



Wir digitalisieren Ihre Geschäftsprozesse

X4 Activities

 **SoftProject**

Die in dieser Dokumentation enthaltenen Informationen und die zugehörigen Programme können ohne besondere Ankündigung geändert werden. Für etwaige Fehler übernimmt SoftProject keine Haftung.

Diese Dokumentation und die zugehörigen Programme dürfen ohne schriftliche Zustimmung der SoftProject GmbH weder ganz noch teilweise kopiert, reproduziert, verändert oder in irgendeine elektronische oder maschinenlesbare Form umgewandelt werden.

Alle genannten Warenzeichen sind Warenzeichen der jeweiligen Eigentümer.

## Kontakt

SoftProject GmbH

Am Erlengraben 3

D-76275 Ettlingen

Website: [www.softproject.de](http://www.softproject.de)

## Vertrieb

Telefon: +49 7243 56175-0

[vertrieb@softproject.de](mailto:vertrieb@softproject.de)

## SoftProject-Support

Telefon: +49 7243 56175-333

[support@softproject.de](mailto:support@softproject.de)

© SoftProject GmbH. Alle Rechte vorbehalten.

Stand: 03.09.2020

# Inhaltsverzeichnis

1	X4 Activities Web Apps.....	10
1.1	Neues Web App-Projekt anlegen und deklarieren .....	10
1.1.1	Neues Web App-Projekt anlegen .....	11
1.1.2	Aufbau eines Web App-Projekts .....	11
1.1.3	Web App-Projekt im Browser anzeigen.....	12
1.1.4	Web App-Definition deklarieren .....	14
1.1.5	Menübreite konfigurieren .....	25
1.1.6	Benutzeranmeldung an der Web App konfigurieren .....	25
1.1.7	Impressum .....	30
1.1.8	Datenschutzerklärung .....	33
1.1.9	Theming .....	36
1.1.10	Responsives Verhalten .....	39
1.2	Lokalisierung .....	41
1.2.1	Eine neue Sprachdatei anlegen .....	41
1.2.2	Übersetzungsschlüssel .....	42
1.3	Benutzerrechte für Web Apps verwalten .....	44
1.3.1	Zugriffsrechte über das X4 Web App Management verwalten .....	45
1.4	Module .....	48
1.5	Datenmodell und Properties-Definition .....	53
1.5.1	Daten aus dem Prozess .....	56
1.5.2	Beispiel für eine einfache Properties-Definition .....	56
1.5.3	Beispiel für eine komplexe Properties-Definition.....	56
1.5.4	Validierung .....	58
1.6	Komponenten .....	62
1.6.1	Attribute für Komponenten .....	64
1.6.2	Detail-Komponente .....	67
1.6.3	List-Komponente .....	69
1.6.4	Kalender-Komponente .....	77
1.6.5	Master/Detail-Komponente.....	87

1.7	Data Binding.....	91
1.8	Layouts.....	93
1.8.1	Flow-Layout.....	99
1.8.2	Grid-Layout.....	104
1.8.3	Registerkarten-Layout: Tab Group.....	107
1.8.4	Registerkarten-Layout: Tab Layout.....	109
1.9	Steuerelemente.....	111
1.9.1	Header.....	115
1.9.2	Label.....	116
1.9.3	TextBox (Web App).....	117
1.9.4	ComboBox.....	120
1.9.5	Checkbox.....	123
1.9.6	ListView.....	125
1.9.7	Radio-Button.....	127
1.9.8	Image.....	128
1.9.9	Datei-Link.....	130
1.9.10	File Upload.....	132
1.9.11	Diagramme.....	134
1.9.12	Maps.....	166
1.9.13	Card.....	181
1.10	Aktionen.....	184
1.10.1	Aktion "Neu".....	189
1.10.2	Aktion "Speichern".....	190
1.10.3	Aktion "Löschen".....	190
1.10.4	Aktion "Auswählen".....	191
1.10.5	Aktion "Datei hochladen".....	191
1.10.6	Aktion "Datei herunterladen".....	193
1.10.7	Benutzerdefinierte Aktion.....	194
1.10.8	Aktion "Neu laden".....	196
1.11	Navigation innerhalb einer Web-Anwendung.....	197
1.11.1	Navigation mit einer Aktion.....	197
1.11.2	Navigation durch Überschreiben der select-Aktion.....	197



1.12	Paging .....	199
1.12.1	Beispielanfrage .....	199
1.12.2	Beispielantwort .....	200
1.12.3	Beispiel für Paging .....	200
1.13	Sortieren .....	202
1.14	Suchen und Filtern .....	203
1.15	Meldungen.....	204
2	X4 Activities Classic .....	207
2.1	X4 Web Designer .....	207
2.2	Benutzeroberfläche des X4 Web Designers .....	207
2.2.1	Menüleiste.....	208
2.2.2	Navigator .....	209
2.2.3	Editor-Bereich .....	212
2.2.4	UI-Editor im X4 Web Designer .....	212
2.3	Tastenkombination im X4 Web Designer .....	218
2.4	Projekte und Ressourcen im X4 Web Designer verwalten.....	219
2.4.1	Struktur eines Web-Portal-Projekts .....	219
2.4.2	Das Kontextmenü im X4 Web Designer .....	221
2.4.3	Neues Web-Portal-Projekt im X4 Web Designer anlegen .....	222
2.4.4	Ordner und Dateien im Navigator verwalten.....	223
2.4.5	Projekt bzw. Web-Anwendung löschen .....	225
2.4.6	Projektname oder Ressourcen-URL Ihrer Web-Anwendung ändern .....	225
2.5	Bestandteile einer X4 Activities-Anwendung.....	225
2.5.1	Application Bar.....	227
2.5.2	Navigator .....	227
2.5.3	Aktionsleiste .....	233
2.5.4	Listenansicht .....	238
2.5.5	Detailmasken .....	250
2.5.6	Main Config .....	252
2.6	Benutzeroberflächen erstellen mit dem X4 Web Designer .....	256
2.6.1	Übersicht der Werkzeuge und Methoden .....	257
2.6.2	Grundlegende Vorgehensweise .....	257

2.6.3	Formulare im UI-Editor definieren .....	258
2.6.4	Widgets im UI-Editor verwenden .....	268
2.6.5	Detailseiten für X4 Activities erstellen .....	379
2.6.6	UI-Controller definieren.....	381
2.6.7	Abhängigkeiten für UI-Elemente definieren.....	383
2.7	X4 Activities-Anwendung im X4 Web Designer konfigurieren.....	383
2.7.1	Datenbank-Konfiguration für Web-Anwendungen .....	383
2.7.2	Application-Config (Main) konfigurieren.....	385
2.7.3	Navigator konfigurieren .....	387
2.7.4	Datasources konfigurieren .....	389
2.7.5	Aktionsleiste konfigurieren .....	393
2.7.6	Detailseiten mit dem UI-Editor erstellen.....	395
2.8	X4 Activities-Anwendung anpassen.....	397
2.8.1	Gestaltung einer X4 Activities-Anwendung anpassen .....	397
2.8.2	Anwendungsspezifische Konfiguration .....	400
2.9	Web-Anwendung neu bauen (Build) .....	405
2.9.1	Einzelne Ordner neu bauen .....	405
2.9.2	Über den X4 Designer komplett neu bauen .....	406
2.10	Web-Anwendung bereitstellen .....	406
2.10.1	Individuelle Web-Anwendung verwenden .....	406
2.10.2	Demo-Web-Anwendung verwenden .....	406
2.11	Anwendung mit dem X4 Web Designer testen und bereitstellen.....	407
2.11.1	Web-Anwendung zu Testzwecken bereitstellen.....	407
2.11.2	Web-Anwendung permanent auf dem Applikations-Server bereitstellen (Deployment).....	408
2.12	Funktionen von X4 Activities .....	409
2.12.1	Berechtigungsprüfung .....	409
2.12.2	Internationalisierung .....	411
2.12.3	Globale Suche .....	415
2.12.4	Dialog.....	416
2.12.5	Selektion von Listeneinträgen .....	417
2.12.6	Mehrfach-Selektion von Listeneinträgen.....	418

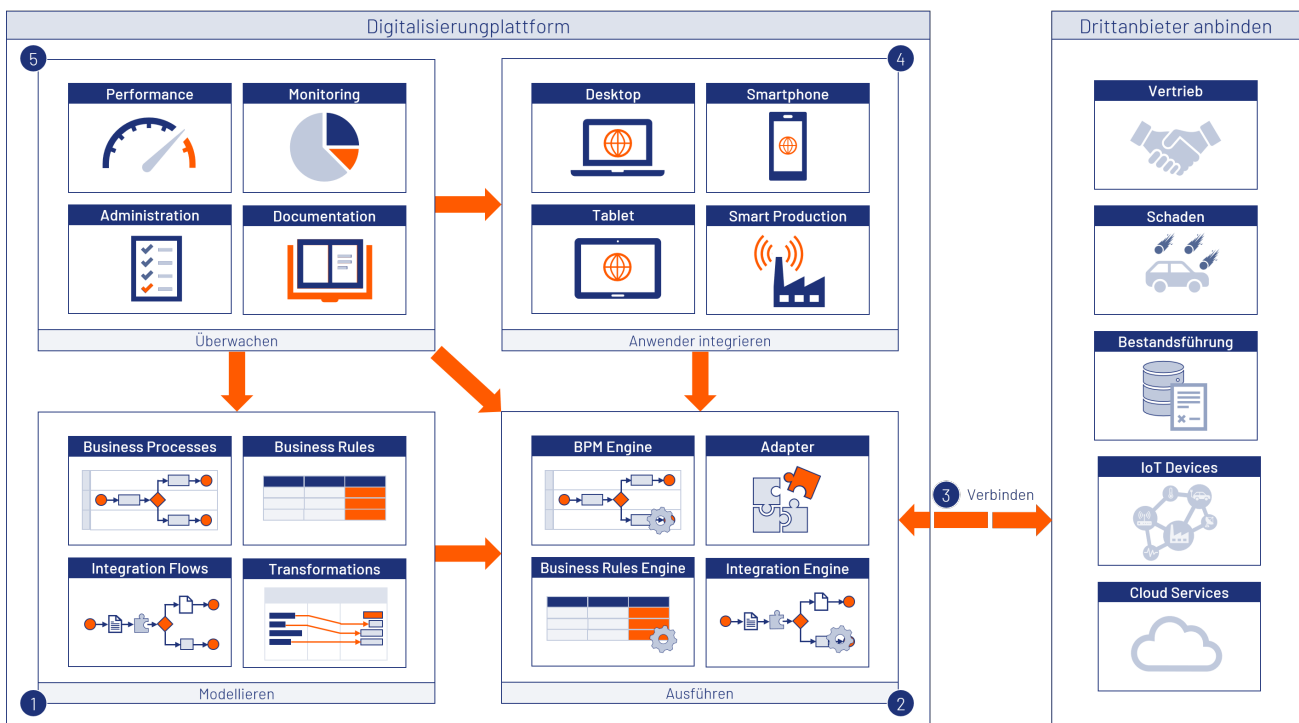
2.12.7	Vor-Selektion von Listeneinträgen .....	419
2.12.8	Marker im Navigator .....	419
2.12.9	Automatische Navigator-Aktualisierung.....	420
2.12.10	Content-only-Applikationen .....	420
2.12.11	Detail-only-Ansicht .....	421
2.12.12	Lister-only-Ansicht .....	422
2.12.13	Offline-Fähigkeit .....	423
2.13	Globale JavaScript-Methoden .....	430
2.13.1	API für die Lister-Tabelle.....	430
2.13.2	Parameter-Injection für die Lister-Tabelle .....	431
2.14	Bibliotheken für X4 Activities .....	431
2.14.1	X4Common-Prozessbibliothek für X4 Activities.....	431
2.14.2	JavaScript-API für Dialoge in X4 Activities .....	434
2.14.3	JavaScript-API für HTML-Tooltips in X4 Activities.....	438
2.14.4	JavaScript-API für den Loading Indicator in X4 Activities .....	440
2.14.5	JavaScript-API für den X4 Web Designer .....	442

## Über die X4 Suite

Digitalisierung braucht eine ganzheitliche Betrachtung, die sich in der einzusetzenden Lösung widerspiegeln muss. Als zentrale Plattform unterstützt Sie die X4 Suite dabei, diese Herausforderungen zu lösen. Im Fokus stehen dabei die Modellierung, Implementierung und Überwachung Ihrer Geschäftsprozesse. Daher enthält die X4 Suite alle benötigten Werkzeuge und ist mit einer Vielzahl an Schnittstellen und Formaten kompatibel. Mit der X4 Suite vermeiden Sie isolierte Informationssilos, produktivitätshemmende Medienbrüche und beschleunigen die Digitalisierung.

Geschäftsprozesse ohne Programmieraufwand zu realisieren, ermöglicht einem großen Anwenderkreis den Einstieg in das Management von Geschäftsprozessen. Das lohnt sich, denn Mitarbeiter der Fachabteilung wissen in der Regel am besten, worauf es bei den jeweiligen Geschäftsabläufen im Kern ankommt. Setzen Sie daher auf die X4 Suite als Plattform, deren Werkzeuge die Komplexität soweit reduzieren, dass sich auch ohne Programmierkenntnisse Geschäftsprozesse analysieren, optimieren, modellieren als auch kontrollieren und dokumentieren lassen. Alle Werkzeuge unterstützen eine integrierte, grafische Prozessmodellierung und -implementierung und erzeugen Prozesse, die von der X4 Suite performant ausgeführt werden.

- **X4 Designer:** Prozesse und Regeln grafisch modellieren
- **X4 Server:** Simulation und Ausführung der Prozesse und Regeln
- **X4 Adapter:** Drittsysteme in Prozesse integrieren
- **X4 Activities:** Web Apps für Mitarbeiter und Kunden bereitstellen
- **X4 Control Center:** Alle Prozesse und Apps überwachen und verwalten



## An wen richtet sich dieses Dokument?

Diese Dokumentation richtet sich an Entwickler, die in kürzester Zeit Web-Anwendungen auf Basis von Web-Technologien und dem X4 ESB mit dessen Webservice-Schnittstelle umsetzen. Dazu werden neben detailliertem fachlichem Wissen der Daten und Arbeitsabläufe, gängige Web-

Technologien wie HTML, CSS und JavaScript auch grundsätzliches Wissen über XML sowie Kenntnisse in XSLT und XPath benötigt.

# 1 X4 Activities Web Apps

Mit der X4 Suite-Komponente X4 Activities Web Apps kann eine Web-Anwendung ohne Programmierung erstellt werden. In diese Web-Anwendungen können direkt Prozesse eingebunden und im Hintergrund Daten verarbeitet werden. Die Web-Anwendungen sind für die Darstellung und Verwendung auf Smartphones, Tablets und klassischen Desktop-PCs optimiert und können mithilfe von Sprachdateien mehrsprachig dargestellt werden. Damit kann X4 Activities Web Apps alternativ zum klassischen X4 Activities genutzt werden.

Um eine Web-Anwendung zu erzeugen, wird der gewünschte Seitenaufbau in XML beschrieben. Dieser Aufbau wird von der X4 Suite automatisch als Web-Anwendung umgesetzt.

Merkmale von X4 Activities Web Apps:

- Zentrale App Definition komplett im X4 Designer
- Einfache Verknüpfung mit technischen Prozessen im X4 Designer
- Dynamische Web-Anwendung, die "on the fly" erweiterbar ist
- Standardmäßig verfügbare Login-Seite
- Responsive Darstellung "out of the box" für Smartphones, Tablets und Desktop-Browser
- Anpassbare Oberflächengestaltung (Farben, Logo, Icons etc.)
- Verwendung mit allen gängigen Desktop-Browser und mobilen Plattformen

Die Komponente X4 Activities Web Apps unterstützt folgende Browser:

- Google Chrome (Desktop/Mobil)
- Apple Safari (Desktop/Mobil)
- Mozilla Firefox (Desktop/Mobil)
- Microsoft Edge (Desktop/Mobil)

Der X4 Designer ermöglicht es, Web App-Projekte mit einer [bereits vorkonfigurierten Projektstruktur](#) anzulegen und diesen mit dem integrierten Theme-Editor eine moderne Optik zu verleihen.

Die vorhandene Business-Logik und statische Ressourcen können einfach in die Web-Anwendungen eingebunden werden. Zusätzlich können rechtliche Angelegenheiten wie ein Impressum und Informationen zum Datenschutz hinterlegt werden.

## 1.1 Neues Web App-Projekt anlegen und deklarieren

In diesem Abschnitt erfahren Sie, wie Sie ein neues Web App-Projekt anlegen und deklarieren.

Ein Web App-Projekt (*WebApp Project*) ist eine Projektart innerhalb der X4 Suite. Es enthält neben einer vordefinierten Ordnerstruktur die zentrale Definitionsdatei `<WebApp>.wad` und wird automatisch als Web-Anwendung im X4 Server registriert.

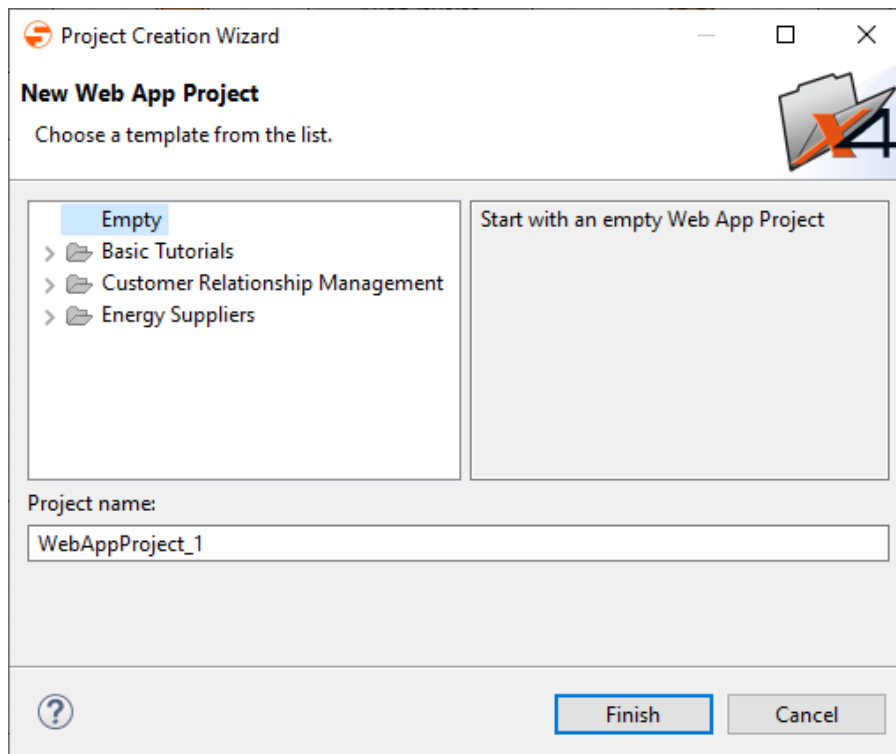


Wenn es bei der Erstellung und Deklaration einer Web-Anwendung zu Fehlern kommen sollte, werden diese nicht nur im Server-Log, sondern auch direkt im Browser angezeigt.



### 1.1.1 Neues Web App-Projekt anlegen

1. Im *X4 Designer* Menü **File > New > WebApp Project** wählen, um den **Project Creation Wizard** zu öffnen.



✓ **Alternativ:** Im Repository rechtsklicken > **New > Web App Project** wählen.

2. In **Project name** den Namen des Projekts eingeben.
3. Auf **Finish** klicken.

Ein neues Web-App-Projekt mit einer vordefinierten [Struktur](#) wird angelegt. Das Web App-Projekt wird automatisch als Web-Anwendung im X4 Server registriert.

### 1.1.2 Aufbau eines Web App-Projekts

Wenn ein Web App-Projekt erstellt wird, wird im X4 Designer folgende Projektstruktur angelegt:

- **Projekt Web App-Projekt**
- **Components:** Separate Definitionen für die einzelnen Komponenten. Der Ordner darf lediglich Definitionen für Detail-Komponenten, List-Komponenten, Kalender-Komponenten und Master/Detail-Komponenten enthalten.
- **Resources:** Alle Ressourcen der Web-Anwendung z. B. Bilder. Kann Unterordner enthalten. Die Inhalte dieses Ordners können über das Internet erreicht werden.
- **Services:**
  - **Adapters:** Adapter, die im Projekt verwendet werden. Kann Unterordner enthalten.
  - **Processes:** Technische Prozesse. Kann Unterordner enthalten.
  - **Resources:** Alle Ressourcen, die in technischen Prozessen verwendet werden. Kann Unterordner enthalten. Kann Text-, XML-, XHTML-, SQL- und Binärdateien enthalten.
  - **TemporaryFiles:** Temporäre Dateien. Kann Unterordner enthalten. Kann Text- und XML-Dateien enthalten.
  - **Transformations:** Mappings für Prozesse. Kann Unterordner enthalten. Kann XSLT- und REP-Dateien enthalten.
- **Translations:** Enthält die einzelnen Sprachdateien für mehrsprachige Oberflächen
- - *Projekt.dpr:* Angaben zum Datenschutz, entweder als Text oder Verlinkung zu einer externen Seite
  - *Projekt.imprint:* Impressum, entweder als Text oder Verlinkung zu einer externen Seite
  - *Projekt.theme:* Theme der Anwendung, kann anhand eines Theme-Designers definiert werden
  - *Projekt.wad* Datei mit der Web App-Definition

Die Ordner Resources, Services und Translations enthalten direkt nach dem Erstellen eines Projektes keine weiteren Dateien. Die Dateien für die Angaben zum Datenschutz und für das Impressum haben zunächst ebenfalls keinen Inhalt und können, wenn sie im Projekt nicht benötigt werden, gelöscht werden.



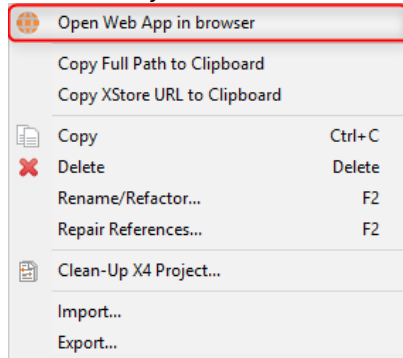
#### Sicherheitshinweis

Beachten Sie, dass alle Datei-Ressourcen innerhalb des Resources-Ordners via HTTP(S) relativ zum Pfad der Web-Anwendung bereitgestellt werden!

### 1.1.3 Web App-Projekt im Browser anzeigen

Das Front-End der Web-Anwendung lässt sich jederzeit, auch während der Bearbeitung, im Browser anzeigen:

1. Auf den Projektordner rechtsklicken und im Kontextmenü **Open Web App in browser** wählen.



Die Login-Seite wird im Browser angezeigt:

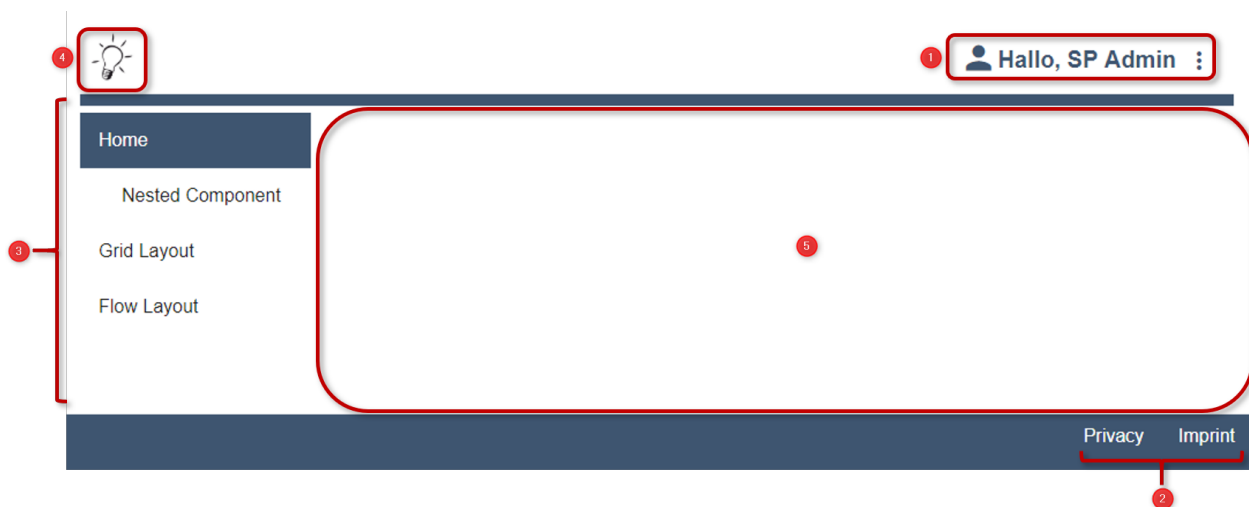
2. Mit den Zugangsdaten einloggen. Wenn keine speziellen Berechtigungen erteilt wurden, dann können alle Benutzer, die im Control Center angelegt sind, auf die Web App zugreifen. Zur Verwaltung von Benutzern bzw. der Rechteverwaltung siehe entsprechende Kapitel [Benutzerrechte für Web Apps verwalten](#).  
Das Web App-Projekt wird angezeigt.

✓ Leerzeichen, die am Ende des Benutzernamens eingefügt werden, z. B. durch eine Autovervollständigung bei der Nutzung am Smartphone, werden automatisch entfernt.

ⓘ Bestimmte Änderungen innerhalb des Web App-Projektes werden erst nach einem erneuten Login angezeigt. Dazu gehören:

- Farben
- Sprache
- Menü

Der Aufbau des Front-Ends einer Web App ist folgender:



1	Begrüßungstext und Benutzermenü
2	Automatisch generierte Links zur Datenschutzerklärung und zum Impressum
3	Automatisch generiertes Menü der Web App. Wird aus den Anzeigenamen der Komponenten gebildet.
4	Logo der Web App
5	Bereich für den Inhalt der Komponente. Die Darstellung hängt von der Art der Komponente, dem verwendeten Layout und den verwendeten Steuerelementen ab.

### 1.1.4 Web App-Definition deklarieren

Web-Anwendungen sind hierarchisch aufgebaut:






- Web-Anwendung
- Modul(e)
- Komponente(n)
- Layout(s)
- Steuerelement(e)

Die Web-Anwendung und die in der Anwendung verfügbaren Module mit ihren jeweiligen Zugriffsrechten werden in der Web App-Definition .wad deklariert. Die einzelnen Komponenten mit ihrem jeweiligen Layout und den verfügbaren Steuerelementen werden innerhalb der jeweiligen Komponente beschrieben und in der Web App-Definition referenziert.

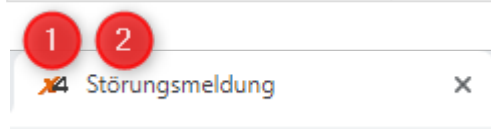
Bei der Deklaration von .wad-Dateien wird die Datei automatisch validiert. So können mögliche Fehlerquellen bereits beim Erstellen von Web-Anwendungen reduziert werden.

Das Wurzelement der .wad-Datei ist <WebApp>. Das Element enthält alle weiteren Elemente der Web App. Über Attribute wird das Aussehen und Verhalten der Web App gesteuert.

#### Mögliche Attribute

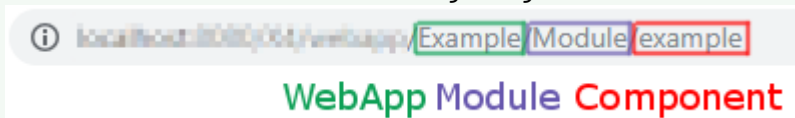
Attribut	Beschreibung
displayName	<p><i>Erforderlich.</i> Anzeigenname der Web App. Wird im Browsertab bzw. der Titelleiste angezeigt.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
faviconUrl	<p>Pfad zur Grafikdatei, die als Favicon der Web App verwendet wird. Die Datei muss im Ordner Resources direkt unterhalb des Web App-Projekts enthalten sein.</p> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p>
logoUrl	<p>Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Logo der Web App verwendet wird.</p> <div data-bbox="547 801 1434 1037"> <p> • Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources.</p> <p>• Das Material Icon muss mit dem Präfix <code>icon:</code>, z. B. <code>icon:&lt;MaterialIconName&gt;</code> angegeben werden.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Zeichenkette (URI), z. B. <code>clock.png</code></li> <li><code>icon:&lt;MaterialIconName&gt;</code>, z. B. <code>icon:extension</code></li> </ul> <div data-bbox="547 1189 1434 1352"> <p> Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren <a href="#">Material Icons</a> abweichen.</p> </div> <div data-bbox="547 1368 1434 1523"> <p> Das Logo wird immer mit einer Höhe von 45 Pixeln angezeigt. Die Grafikdatei, die als Logo verwendet wird, wird automatisch skaliert.</p> </div>
logoColor	<p>Definiert die Farbe des Logos in der Anwendung.</p> <div data-bbox="547 1612 1434 1722"> <p> Diese Einstellung überschreibt die Standardfarbe des Farbschemas!</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Hexadezimaler Farbwert, z. B. <code>ff5a00</code></li> </ul> <div data-bbox="592 1859 1434 1968"> <p> Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts!</p> </div> <ul style="list-style-type: none"> <li>Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. <code>A200</code></li> </ul>

Attribut	Beschreibung
path	<p><i>Erforderlich.</i> Pfad zur Web App. Wird in der Adresszeile des Browsers angezeigt. Muss für jedes Web App-Projekt eindeutig sein.</p> <p><b>Mögliche Werte:</b> Zeichenkette aus alphanumerischen Zeichen (keine Umlaute, Punkte usw.)</p> <div> <p><b>i</b> Der Pfad bildet einen Teil der URL, über die die Web App über den Browser erreichbar ist. Als Standardwert wird für das Attribut path der Name des Web-App-Projekts hinterlegt. Der Pfad kann aber eine beliebige Zeichenkette aus alphanumerischen Zeichen enthalten. Wenn das Web-App-Projekt umbenannt wird, dann wird das Attribut path <b>nicht</b> angepasst und die Web App ist weiterhin über die bisherige URL erreichbar.</p> </div>
showFooter	<p>Definiert, ob die Fußzeile der Web-Anwendung angezeigt wird.</p> <p><b>Mögliche Werte:</b> true (Standard) / false</p> <div> <p><b>w</b> Falls in der angezeigten Komponente Aktionen vorhanden sind, wird die Fußzeile in der mobilen Ansicht unabhängig vom Wert des Attributs angezeigt.</p> </div>
showHeader	<p>Definiert, ob die Kopfzeile der Web-Anwendung angezeigt wird.</p> <p><b>Mögliche Werte:</b> true (Standard) / false</p>



1	Favicon der Web App
2	Anzeigename der Web App

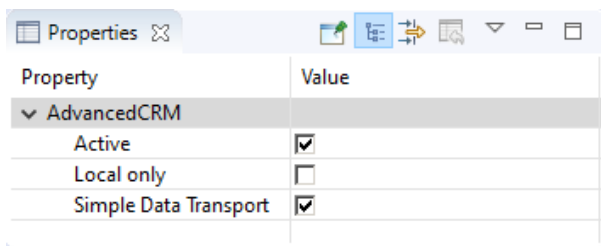
- ✓ Mithilfe der Pfadangaben für das Projekt, die Module und die Komponenten (Attribut path) kann innerhalb der Web-Anwendung navigiert werden.



#### 1.1.4.1 Status des Web App-Projekts festlegen

Ein Web App-Projekt kann unterschiedliche Status haben. Der Status wird in der **Properties**-Sicht unten links angezeigt, sobald das Web App-Projekt ausgewählt ist:





<b>Active</b>	<p>Angabe, ob das Projekt aktiviert ist.</p> <div> <p><b>i</b> Wenn das Projekt nicht aktiv ist, kann die Web-Anwendung im Browser nicht angezeigt werden.</p> </div>
<b>Local only</b>	<p>Angabe, ob das Projekt nur lokal bereitgestellt werden soll.</p> <div> <p><b>i</b> Wenn diese Option aktiviert ist, kann die Web-Anwendung nur im lokalen System im Browser angezeigt werden.</p> </div>
<b>Simple Data Transport</b>	<p>Angabe, ob das Projekt das Simple Data Transport verwendet.</p> <div> <p><b>i</b> Wenn diese Option aktiviert ist, wird das Simple Data Transport, d.h. das seit Release 6.2.0 verfügbare vereinfachte Datenmodell für die Verarbeitung von Daten innerhalb einer Web-Anwendung verwendet.</p> <p>Durch das geänderte Datenmodell müssen Web-App-Projekte, die mit älteren Versionen als 6.2.0 erstellt wurden, manuell an das neue Format angepasst werden, um den Simple Data Transport zu verwenden. Alternativ kann die Option <b>Simple Data Transport</b> für ein Web-App-Projekt deaktiviert werden.</p> </div>

#### 1.1.4.2 Komponenten in der Web App-Definition referenzieren



Jede einzelne in der Web App verwendete **Komponente** wird in einer separaten Definition deklariert. Diese einzelnen Definitionen werden anschließend in der Web App-Definition über das Element `<ComponentReference>` referenziert.




Innerhalb der `<ComponentReference>` können weitere `<Components>` mit ihren jeweiligen Referenzen (`<ComponentReference>`) enthalten sein. Damit kann eine hierarchische Navigation umgesetzt werden.



Das Element `<ComponentReference>` verfügt über folgende Attribute:

Attribut	Beschreibung
default	<p>Bei genau einer Komponente erforderlich. Legt fest, welche Komponente zuerst angezeigt wird ("Startseite").</p> <p><b>Mögliche Werte:</b> <i>true / false</i></p>

Attribut	Beschreibung
displayName	<p>Anzeigenname der Komponente im Menü.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette mit dem Anzeigenamen der Komponente</p>
fontFamily	<p>Legt die Schriftfamilie innerhalb der Komponente fest. Die Eigenschaft wird an alle Steuerelemente und Aktionen der Komponente vererbt.</p> <p><b>Mögliche Werte:</b> Fontcode aus der Font-Palette, z. B. <i>Font04</i></p>
fontSize	<p>Legt die Schriftgröße innerhalb der Komponente fest. Die Eigenschaft wird an alle Steuerelemente und Aktionen der Komponente vererbt.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Schriftgröße in Pixel, z. B. <i>20px</i></li> <li>• Schriftgröße in Punkt, z. B. <i>18pt</i></li> <li>• Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i></li> <li>• Schlüsselwörter, z. B. <i>small</i> oder <i>larger</i></li> </ul>
fontStretch	<p>Legt die Breite der einzelnen Schriftzeichen fest. Die Eigenschaft wird an alle Steuerelemente und Aktionen der Komponente vererbt.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Condensed</li> <li>• Expanded</li> <li>• ExtraCondensed</li> <li>• ExtraExpanded</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• SemiCondensed</li> <li>• SemiExpanded</li> <li>• UltraCondensed</li> <li>• UltraExpanded</li> </ul>
fontStyle	<p>Legt die Neigung der Schrift fest. Die Eigenschaft wird an alle Steuerelemente und Aktionen der Komponente vererbt.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <i>italic</i>: kursive Schrift</li> <li>• <i>normal</i>: normale Schrift (Standard)</li> <li>• <i>oblique</i>: schräg gestellter Schriftstil (berechnet)</li> </ul>

Attribut	Beschreibung
fontWeight	<p>Legt die Schriftstärke fest. Die Eigenschaft wird an alle Steuerelemente und Aktionen der Komponente vererbt.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Black</li> <li>• Bold</li> <li>• DemiBold</li> <li>• ExtraBlack</li> <li>• ExtraBold</li> <li>• ExtraLight</li> <li>• Heavy</li> <li>• Light</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• Regular</li> <li>• SemiBold</li> <li>• Thin</li> <li>• UltraBlack</li> <li>• UltraBold</li> <li>• UltraLight</li> </ul>
iconURL	<p>Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Icon der Komponente verwendet wird.</p> <div data-bbox="547 1193 1439 1429" style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> • Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources.</p> <p>• Das Material Icon muss mit dem Präfix <code>icon:</code>, z. B. <code>icon:&lt;MaterialIconName&gt;</code> angegeben werden.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Zeichenkette (URI), z. B. <code>clock.png</code></li> <li>• <code>icon:&lt;MaterialIconName&gt;</code>, z. B. <code>icon:extension</code></li> </ul> <div data-bbox="547 1581 1439 1742" style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren <b>Material Icons</b> abweichen.</p> </div>

Attribut	Beschreibung
iconColor	<p>Definiert die Farbe des Icons der Komponente.</p> <div>  Diese Einstellung überschreibt die Standardfarbe des Farbschemas!         </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Hexadezimaler Farbwert, z. B. <i>ff5a00</i></li> </ul> <div>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts!         </div> <ul style="list-style-type: none"> <li>Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. <i>A200</i></li> </ul>
name	<p><i>Erforderlich.</i> Eindeutiger Name der Komponente. Wird für Data Binding verwendet.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette mit dem Namen der Komponente</p>
path	<p><i>Erforderlich.</i> Eindeutige URL der Komponente, sichtbar in der Adresszeile des Browsers.</p> <p><b>Mögliche Werte:</b> Zeichenkette aus alphanumerischen Zeichen (keine Umlaute, Punkte usw.)</p>
showInMenu	<p>Legt fest, ob die Komponente im Menü angezeigt wird.</p> <p><b>Mögliche Werte:</b> <code>true</code> (default) / <code>false</code></p> <div>  Diese Einstellung bezieht sich auch auf alle Subkomponenten. Wenn keine Komponente im Menü angezeigt wird, dann wird das gesamte Menü ausgeblendet. Der Inhaltsbereich erstreckt sich dann über die gesamte Breite.         </div>
source	<p>Pfad zur Definitionsdatei relativ zum Ordner Components.</p> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p>
title	<p>Titel der Komponente. Wird in der Komponente als Header angezeigt.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette mit dem Titel der Komponente</p>

Attribut	Beschreibung
titleBackground	<p>Definiert eine Titel-Hintergrundfarbe.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"><li>• Hexadezimaler Farbwert, z. B. ff5a00</li></ul> <div> Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts!</div> <ul style="list-style-type: none"><li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li></ul>
titleForeground	<p>Definiert eine Farbe für den Titel-Vordergrund.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"><li>• Hexadezimaler Farbwert, z. B. ff5a00</li></ul> <div> Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts!</div> <ul style="list-style-type: none"><li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li></ul>

## Beispiel

### Definition des integrierten Beispielprojektes Simple CRM

```
<?xml version="1.0" encoding="UTF-8"?>
<WebApp xmlns="http://softproject.de/webapp/1.0"
  displayName="Customer-Relationship-Management"
  logoUrl="SimpleCRM.png" path="SimpleCRM">
  <Translations>
    <Translation displayName="Deutsch" name="de"/>
    <Translation displayName="English" name="en"/>
  </Translations>
  <Modules>
    <Module displayName="My Module" path="Module">
      <Components>
        <ComponentReference default="true" displayName="$Customers"
          iconUrl="logo.png" name="Home"
          path="Home" source="Customer/Customer.masterdetail"/>
        <ComponentReference displayName="Statistik" iconUrl="Statistic.png"
          path="Statistic" source="Statistic/Statistic.detail"/>
        <ComponentReference displayName="$Administration" iconUrl="Gear.png"
          path="Administration" source="Administration/
Administration.detail">
          <Components>
            <ComponentReference displayName="$Country" iconUrl="Country.p
ng"
              name="Countries" path="Country"
              source="Administration/Country/Country.masterdetail"/>
            <ComponentReference displayName="$Category" iconUrl="Category
.png"
              name="Categories" path="Category"
              source="Administration/Category/Category.masterdetail"/>
            <ComponentReference displayName="$LegalForm" iconUrl="LegalFo
rm.png"
              name="LegalForm" path="LegalForm"
              source="Administration/LegalForm/LegalForm.masterdetail"
/>
            <ComponentReference displayName="$Event" iconUrl="Event.png"
              name="Event" path="Event"
              source="Administration/Event/Event.masterdetail"/>
          </Components>
        </ComponentReference>
      </Components>
    </Module>
  </Modules>
</WebApp>
```

#### 1.1.4.3 Zugriffsrechte für Module definieren

Die Zugriffsrechte auf die einzelnen Module einer Web-Anwendung werden innerhalb der .wad definiert.

Innerhalb des Elementes <AccessRightDefinitions> werden beliebig viele Berechtigungen über ein Element <AccessRightDefinition> angegeben. Diese Berechtigungen werden im [X4 Web App Management](#) verwaltet.



Das Element `<AccessRightDefinition>` verfügt über folgende Attribute:

Attribut	Beschreibung
name	Name der Berechtigung

```
<AccessRightDefinitions>
  <AccessRightDefinition name="Administrator"/>
  <AccessRightDefinition name="Staff"/>
  <AccessRightDefinition name="Customers"/>
</AccessRightDefinitions>
```


Die unter `<AccessRightDefinitions>` definierten Berechtigungen können anschließend den einzelnen Modulen zugewiesen werden. Dabei gibt es verschiedene Möglichkeiten der Zuweisung:

- Als Kindelement von `Module`

```
<Modules>
  <Module>
    <AccessRights>
      <AccessRight name="Administrator"/>
    </AccessRights>
    ...
  </Module>
  ...
</Modules>
```

- Direkt im `Module`-Element über die Attribute `accessRight` und `hideOnAccessDenied`, siehe auch [Module](#)

```
<Modules>
  <Module displayName="Restricted Module" path="RestrictedModule" accessRight="
Staff" hideOnAccessDenied="false">
    ...
  </Module>
  ...
</Modules>
```

 Werden für das Modul keine Zugriffsrechte definiert, ist das Modul für alle Benutzergruppen verfügbar.

## Beispiel

Anhand des folgenden Beispiels soll Zuweisung von Zugriffsberechtigungen erläutert werden.

```

WebApp xmlns="http://softproject.de/webapp/1.0" displayName="MyWebApp"
path="MyWebAppProject">
  <AccessRightDefinitions>
    <AccessRightDefinition name="Administrator"/>
    <AccessRightDefinition name="Staff"/>
    <AccessRightDefinition name="Customers"/>
  </AccessRightDefinitions>
  <Modules>
    <Module displayName="Public Module" path="PublicModule">
      <Components>
        ...
      </Components>
    </Module>
    <Module displayName="Administration Module" path="AdminModule">
      <AccessRights>
        <AccessRight name="Administrator"/>
      </AccessRights>
      <Components>
        ...
      </Components>
    </Module>
    <Module displayName="Restricted Module" path="RestrictedModule" accessRight="
Staff" hideOnAccessDenied="true">
      <Components>
        ...
      </Components>
    </Module>
    <Module displayName="Staff Module" path="StaffModule">
      <AccessRights>
        <AccessRight name="Staff"/>
        <AccessRight name="Administrator"/>
      </AccessRights>
      <Components>
        ...
      </Components>
    </Module>
  </Modules>
</WebApp>

```

### Erläuterung

Für die Web-Anwendung wurden drei verschiedenen Berechtigungen definiert: Administrator, Staff und Customers.

Für die vier Module der Anwendung werden unterschiedliche Zugriffsrechte definiert.


- Für das Modul `Public Module` wurden keine Zugriffsrechte definiert, sodass es für alle Benutzergruppen verfügbar ist.
- Für das Modul `Administration Module` hat lediglich die `Administrator`-Gruppe Zugriffsrechte. Alle anderen Gruppen sehen zwar das Modul, können aber nicht auf dessen Inhalte zugreifen.
- Für das Modul `Restricted Module` wurden die Zugriffsrechte direkt im `Module`-Element definiert. Lediglich die `Staff`-Gruppe kann auf das Modul zugreifen. Mit `hideOnAccessDenied="true"` wurde zudem definiert, dass alle Benutzer, die keinen Zugriff haben, das Modul auch nicht sehen können.

- Auf das Modul `Staff` Module können sowohl die `Administrator`-Gruppe als auch die `Staff`-Gruppe zugreifen. Die `Customers`-Gruppe sieht zwar das Modul, kann aber nicht auf dessen Inhalte zugreifen.

### 1.1.5 Menübreite konfigurieren

Die Breite des Menüs einer Web-Anwendung lässt sich in der `.wad`-Datei über die Attribute des Menu-Elements definieren. Das Menu-Element ist ein direktes Kindelement des `WebApp`-Wurzelements.

#### Mögliche Attribute

Attribut	Beschreibung
<code>unit</code>	<p>Einheit für den in Attribut <code>width</code> angegebenen Wert.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <code>pixels</code>: Angabe in Pixel</li> <li>• <code>percents</code>: Angabe in Prozent</li> </ul> <div>  Wird keine Einheit definiert, wird der in <code>width</code> angegebene Wert als Pixel interpretiert.         </div>
<code>width</code>	<p>Breite des Menüs</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• beliebige ganze Zahl</li> <li>• <code>auto</code>: Automatische Breite, die sich anhand der definierten Menüeinträgen automatisch berechnet</li> </ul>

#### Beispiel-Konfiguration für die Menübreite

```
<WebApp xmlns="http://softproject.de/webapp/1.0" displayName="My Web App">
  ...
  <Menu width="30" unit="percents"/>
  <Modules>
    ...
  </Modules>
</WebApp>
```

### 1.1.6 Benutzeranmeldung an der Web App konfigurieren

Für die Benutzeranmeldung der Web-Anwendung können verschiedene Anpassungen vorgenommen werden. Die Einstellungen werden in der `.wad`-Datei über die Attribute des Login-Elements definiert. Das Login-Element ist ein direktes Kindelement des `WebApp`-Wurzelements.

#### Beispiele für mögliche Anpassungen:

- Icon für die Web-Anwendung

- Optionales Hintergrundbild für die Web-Anwendung
- Anpassung passt zu Smartphone-Bildschirmen (vertikal/horizontal, Tablet, Desktop)
- Anpassbar, ob Passwortwechsel und Remember-me für Nutzer verfügbar sind
- Fehlermeldungen bei falschem Passwort o. Ä.
- etc.

#### Beispiel-Deklaration

```
<WebApp ...>
  <Login changePassword="false"
    forgotPassword="false"
    rememberMe="false"
    loginFailedMessage="$YourCredentialsarenotcorrectorinvalid">
  </Login>
  ...
</WebApp>
```

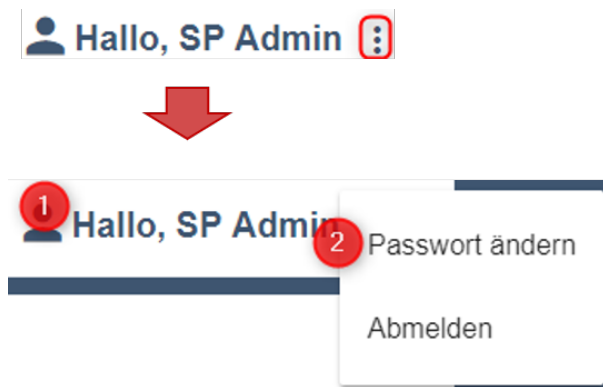
#### 1.1.6.1 Aussehen des Login-Bildschirms



1	Logo. Wird in der <a href="#">Web App-Definition</a> festgelegt.
2	Passwortfeld. Je nach Wert des Attributs <code>passwordFieldType</code> ändert sich das Aussehen.
3	Slider für die Angemeldet-bleiben-Funktion ( <code>rememberMe</code> )
4	Passwort-vergessen-Option ( <code>forgotPassword</code> )

#### 1.1.6.2 Benutzermenü

Das Benutzermenü öffnet sich, wenn in der Desktop-Ansicht auf die drei Punkte oben rechts geklickt wird. In der mobilen Ansicht sind die Einträge des Benutzermenüs in das normale Menü integriert.



- |   |   |
|---|---|
| 1 | Begrüßungstext (greetLabel)             |
| 2 | Passwort-ändern-Option (changePassword) |




### Mögliche Attribute

Attribut	Beschreibung
backgroundImageUrl	<p>Pfad zur Grafikdatei, die als Hintergrundbild der Web App verwendet wird. Die Datei muss im Ordner Resources direkt unterhalb des Web App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources.</p> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p>
changePassword	<p>Ermöglicht dem Benutzer nach der Anmeldung im Benutzermenü sein Passwort zu ändern.</p> <p><b>Mögliche Werte:</b> <i>true</i>(Standard)/ <i>false</i></p>
changePasswordButtonCancel	<p>Beschriftung des Abbrechen-Buttons im Passwort-ändern-Dialog.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
changePasswordButtonOk	<p>Beschriftung des Bestätigen-Buttons im Passwort-ändern-Dialog.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
changepasswordDoNotMatch	<p>Text, der angezeigt wird, wenn die beiden neuen Passwörter im Passwort-ändern-Dialog nicht übereinstimmen.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
changePasswordInvalid	<p>Text, der angezeigt wird, wenn das alte Passwort im Passwort-ändern-Dialog falsch ist.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
changePasswordLabel	<p>Überschrift des Passwort-ändern-Dialogs.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>

Attribut	Beschreibung
changePasswordNewPassword	Beschriftung des Feldes für das neue Passwort im Passwort-ändern-Dialog.  <b>Mögliche Werte:</b> Beliebige Zeichenkette
changePasswordOldPassword	Beschriftung des Feldes für das alte Passwort im Passwort-ändern-Dialog.  <b>Mögliche Werte:</b> Beliebige Zeichenkette
changePasswordRepeat	Beschriftung des Feldes für die Bestätigung des neuen Passworts im Passwort-ändern-Dialog.  <b>Mögliche Werte:</b> Beliebige Zeichenkette
changePasswordReused	Text, der im Passwort-ändern-Dialog angezeigt wird, wenn das Passwort bereits verwendet wurde.  <b>Mögliche Werte:</b> Beliebige Zeichenkette
changePasswordSuccess	Text, der im Passwort-ändern-Dialog angezeigt wird, wenn das Passwort erfolgreich geändert wurde.  <b>Mögliche Werte:</b> Beliebige Zeichenkette
changePasswordWrong	Text, der im Passwort-ändern-Dialog angezeigt wird, wenn das Passwort nicht den Passwortvorgaben entspricht.  <b>Mögliche Werte:</b> Beliebige Zeichenkette
emailLabel	Text, der im Passwort-vergessen-Dialog im Feld für die E-Mail-Adresse angezeigt wird.  <b>Mögliche Werte:</b> Beliebige Zeichenkette
forgotPassword	Legt fest, ob der Passwort-vergessen-Dialog für den Benutzer verfügbar ist.  <b>Mögliche Werte:</b> <i>true</i> (Standard)/ <i>false</i>
forgotPasswordButtonCancel	Beschriftung des Abbrechen-Buttons im Passwort-vergessen-Dialog.  <b>Mögliche Werte:</b> Beliebige Zeichenkette
forgotPasswordButtonOk	Beschriftung des Bestätigen-Buttons im Passwort-vergessen-Dialog.  <b>Mögliche Werte:</b> Beliebige Zeichenkette

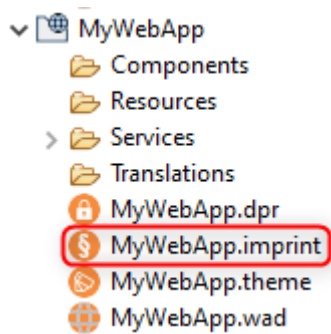


Attribut	Beschreibung
forgotPasswordLabel	Überschrift des Passwort-vergessen-Dialogs. <b>Mögliche Werte:</b> Beliebige Zeichenkette
forgotPasswordRestoreField	Beschriftung des Feldes für die Benutzerinformationen im Passwort-vergessen-Dialog. <b>Mögliche Werte:</b> Beliebige Zeichenkette
greetLabel	Text der Begrüßung. Diese wird zusammen mit dem Benutzernamen oben rechts angezeigt. <b>Mögliche Werte:</b> Beliebige Zeichenkette
loginButtonLabel	Beschriftung des Login-Buttons. <b>Mögliche Werte:</b> Beliebige Zeichenkette
loginFailedMessage	Text, der angezeigt wird, wenn die Anmeldung fehlgeschlagen ist. <b>Mögliche Werte:</b> Beliebige Zeichenkette
loginLabel	Beschriftung des Feldes für den Benutzernamen. <b>Mögliche Werte:</b> Beliebige Zeichenkette
logoutButtonLabel	Beschriftung des Logout-Buttons im Benutzermenü. <b>Mögliche Werte:</b> Beliebige Zeichenkette
passwordFieldType	Anzeige der Eingaben in Passwortfeldern <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• <i>text</i>: Eingabe wird in Klartext angezeigt</li> <li>• <i>password</i>: Eingabe wird als Punkte unkenntlich gemacht</li> <li>• <i>toggle</i>: Eingabe kann umgeschaltet werden zwischen sichtbar und unsichtbar (Default)</li> </ul>
passwordLabel	Beschriftung des Feldes für das Passwort. <b>Mögliche Werte:</b> Beliebige Zeichenkette

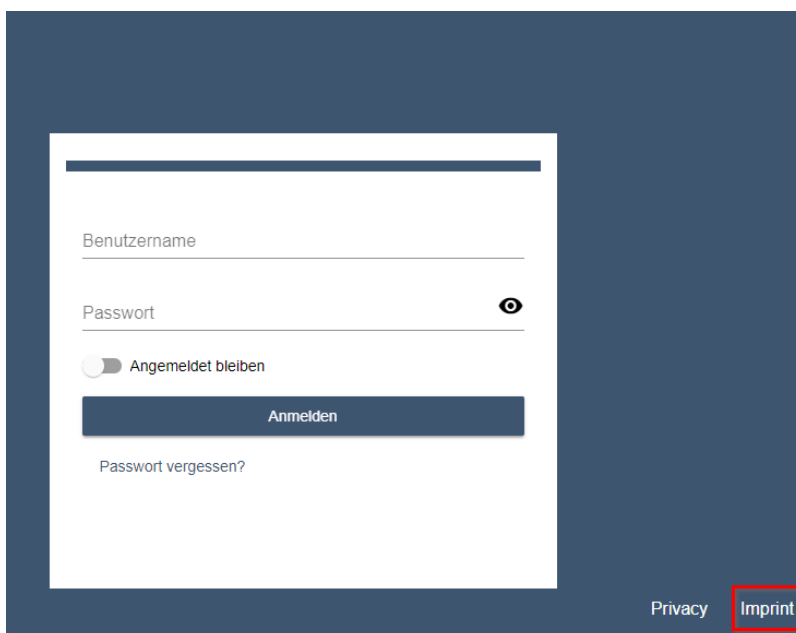
Attribut	Beschreibung
publicAccess	<p>Legt fest, ob ein Login nötig ist, um die Web App zu verwenden.</p> <p><b>Mögliche Werte:</b> <code>true</code> / <code>false</code> (Standard)</p> <div>  Über das Attribut <code>greetLabel</code> keine eine neutrale Begrüßung ohne persönliche Ansprache hinterlegt werden. </div> <div>  Benutzer, die ohne Login auf die Web App zugreifen, werden als <code>ANONYMOUS_USER</code> erkannt. Diese können keine besonderen Rechte erhalten. </div>
rememberMe	<p>Ermöglicht dem Benutzer, angemeldet zu bleiben (Angemeldet-bleiben-Funktion).</p> <p><b>Mögliche Werte:</b> <code>true</code> (Standard) / <code>false</code></p> <div>  Wenn der Benutzer der Web-Anwendung die Angemeldet-bleiben-Funktion aktiviert, dann gilt das nur für diese eine Web-Anwendung. Alle weiteren Web-Anwendungen sind davon nicht betroffen. </div>
rememberMeLabel	<p>Beschriftung der Angemeldet-bleiben-Funktion</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
required	<p>Ändert den standardmäßig hinterlegten Beschriftungstext Pflichtfeld im Passwort-ändern-Dialog</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>

### 1.1.7 Impressum

Das Impressum einer Web-Anwendung, die als Web App-Projekt erstellt wurde, wird automatisch erzeugt. Es enthält zunächst keinen Text und kann daher bearbeitet, manuell gelöscht und erneut erzeugt werden. Die Datei, die das Impressum enthält, heißt immer wie das Web App-Projekt und hat die Dateiendung `.imprint`. Innerhalb eines Web App-Projekts kann nur eine `.imprint`-Datei angelegt werden.



Auf der Login-Seite sowie auf allen Seiten innerhalb der Web-Anwendung wird ein Link zum Impressum angezeigt. Der Link wird nur angezeigt, wenn auch eine .imprint-Datei innerhalb des Web App-Projekts vorhanden ist.



## Eine Seite für das Impressum anlegen

1. Mit der rechten Maustaste auf das Web App-Projekt klicken.  
Das Kontextmenü öffnet sich.
2. Auf **New > Imprint** klicken.  
Die Datei für das Impressum wird erzeugt.

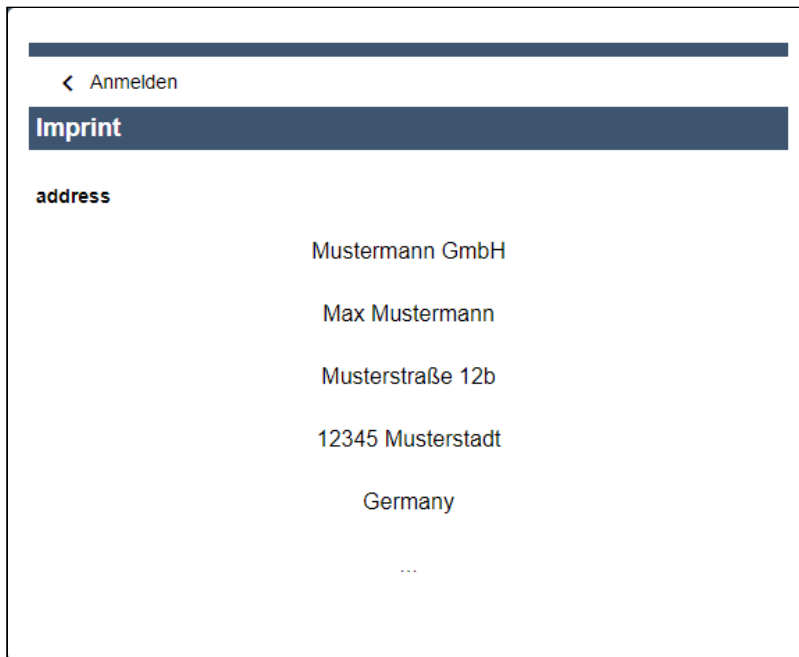
### 1.1.7.1 Inhalt im Impressum hinterlegen

Die Impressumsseite kann ein beliebiges [Layout](#) enthalten. Innerhalb des Layouts kann der Inhalt des Impressums hinterlegt werden:

### Impressum (.imprint)

```
<?xml version='1.0' encoding='UTF-8'?>
<Imprint xmlns="http://softproject.de/webapp/1.0" title="Imprint">
  <FlowLayout>
    <Header value="Imprint" />
    <Header titleLevel="subtitle" value="address" />
    <Label value="Mustermann GmbH" />
    <Label value="Max Mustermann" />
    <Label value="Musterstraße 12b" />
    <Label value="12345 Musterstadt" />
    <Label value="Germany" />
    <Label value="..." />
  </FlowLayout>
</Imprint>
```

Obiger Code erzeugt folgendes Impressum:



#### 1.1.7.2 Externe Seiten einbinden

Es ist auch möglich, externe Seiten als Impressum einzubinden. Dafür muss ein Link hinterlegt werden:

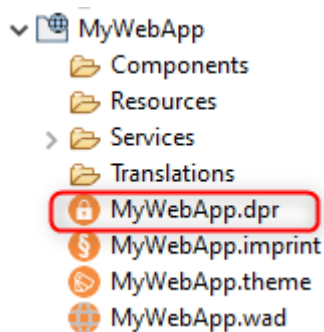
### Externe Seite als Impressum einbinden

```
<?xml version="1.0" encoding="UTF-8" ?>
<Imprint xmlns="http://softproject.de/webapp/1.0" title="Imprint" link="http://www.example.com">
  <FlowLayout />
</Imprint>
```

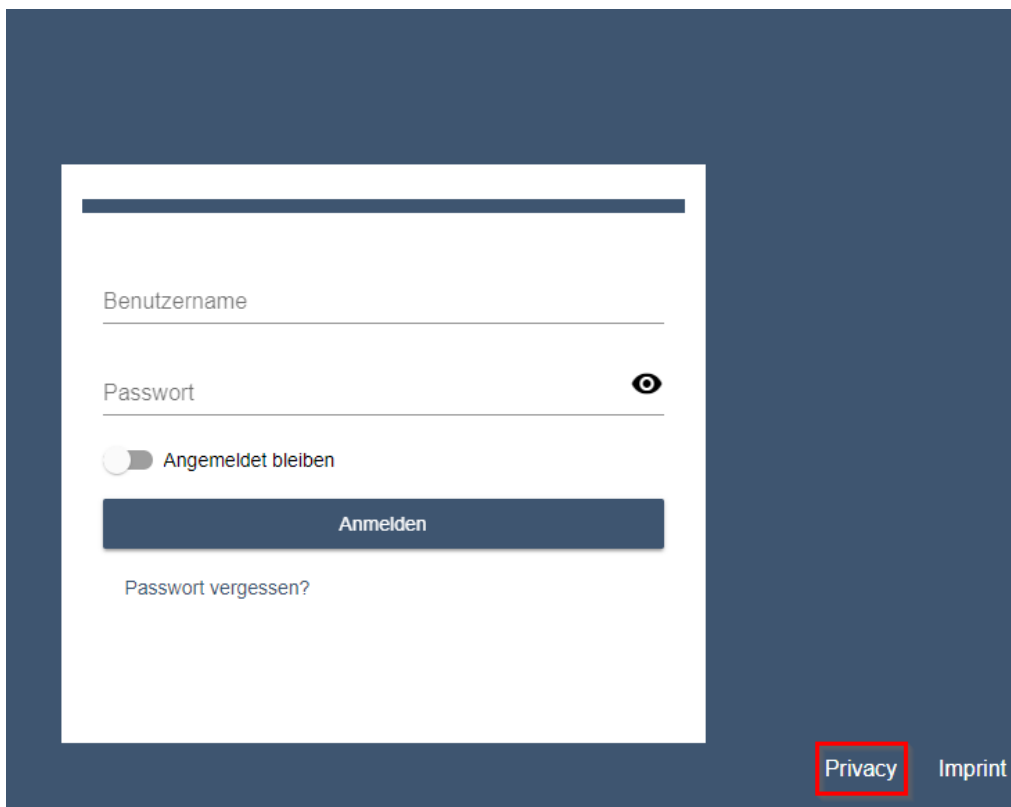
❗ Auch wenn die Impressumsseite hier keinen Inhalt hat, muss trotzdem ein Layout angegeben werden. Wird kein Layout angegeben, kann die Seite nicht gespeichert werden.

### 1.1.8 Datenschutzerklärung

Die Datenschutzerklärung einer Web-Anwendung, die als Web App-Projekt erstellt wurde, wird automatisch erzeugt. Sie enthält zunächst keinen Text und kann daher bearbeitet, manuell gelöscht und erneut erzeugt werden. Die Datei, die die Datenschutzerklärung enthält, heißt immer wie das Web App-Projekt und hat die Dateiendung `.dpr` (Data Protection Statement). Innerhalb eines Web App-Projekts kann nur eine `.dpr`-Datei angelegt werden.



Auf der Login-Seite sowie auf allen Seiten innerhalb der Web-Anwendung wird ein Link zur Datenschutzerklärung angezeigt. Der Link wird nur angezeigt, wenn auch eine `.dpr`-Datei innerhalb des Web App-Projekts vorhanden ist.



#### 1.1.8.1 Eine Seite für die Datenschutzerklärung anlegen

1. Mit der rechten Maustaste auf das Web App-Projekt klicken.  
Das Kontextmenü öffnet sich.
2. Auf **New** > **Data Protection Statement** klicken.  
Die Datei für die Datenschutzerklärung wird erzeugt.

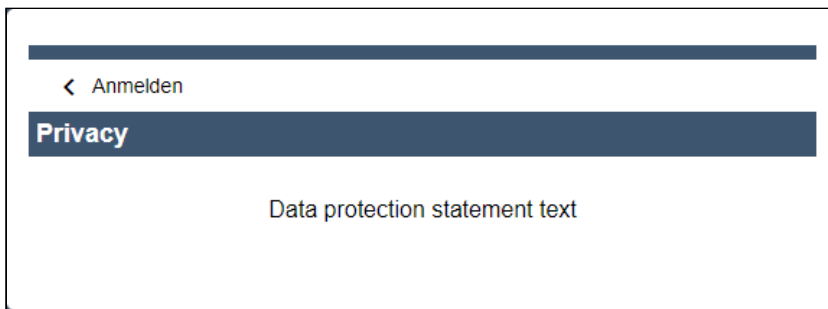
#### 1.1.8.2 Inhalt in der Datenschutzerklärung hinterlegen

Die Datenschutzerklärung kann ein beliebiges [Layout](#) enthalten. Innerhalb des Layouts kann der Inhalt der Datenschutzerklärung hinterlegt werden:

##### Datenschutzerklärung (.dpr)

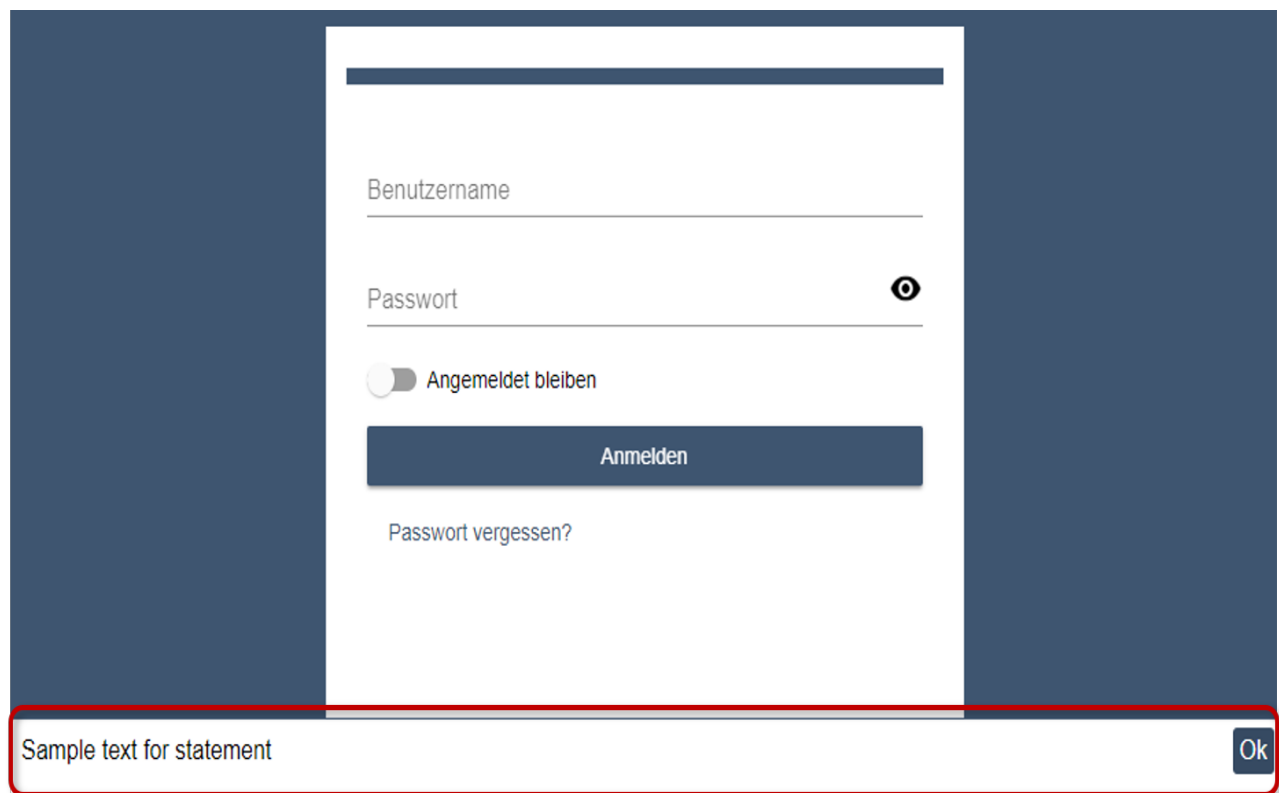
```
<?xml version='1.0' encoding='UTF-8'?>
<DataProtectionStatement xmlns="http://softproject.de/webapp/1.0" title="Privacy">
  <FlowLayout>
    <Header value="Privacy" />
    <Label value="Data protection statement text"/>
  </FlowLayout>
  <Hint text="Sample text for statement" buttonText="Ok"></Hint>
</DataProtectionStatement>
```

Obiger Coder erzeugt folgende Datenschutzerklärung:



A dialog box with a dark blue header bar containing a back arrow and the text "Anmelden". Below the header is a dark blue bar with the word "Privacy" in white. The main area of the dialog is white and contains the text "Data protection statement text".

Das Element `Hint` ist optional. Es erzeugt ein Hinweisfenster im unteren Bereich der Seite. Hier können beispielsweise Informationen zur Verwendung von Cookies hinterlegt werden. Innerhalb des Hinweisfensters kann mit dem Attribut `LinkText` auch auf eine externe Seite verwiesen werden.



A login form with a dark blue header bar. Below the header are input fields for "Benutzername" and "Passwort". The "Passwort" field has a toggle icon to its right. Below the password field is a toggle switch labeled "Angemeldet bleiben". A dark blue button labeled "Anmelden" is below the toggle. Below the button is the text "Passwort vergessen?". At the bottom of the form is a red-bordered hint box containing the text "Sample text for statement" and an "Ok" button.

### 1.1.8.3 Externe Seiten einbinden

Es ist auch möglich, externe Seiten als Datenschutzerklärung einzubinden. Dafür muss ein Link hinterlegt werden:

### Externe Seite als Datenschutzerklärung einbinden

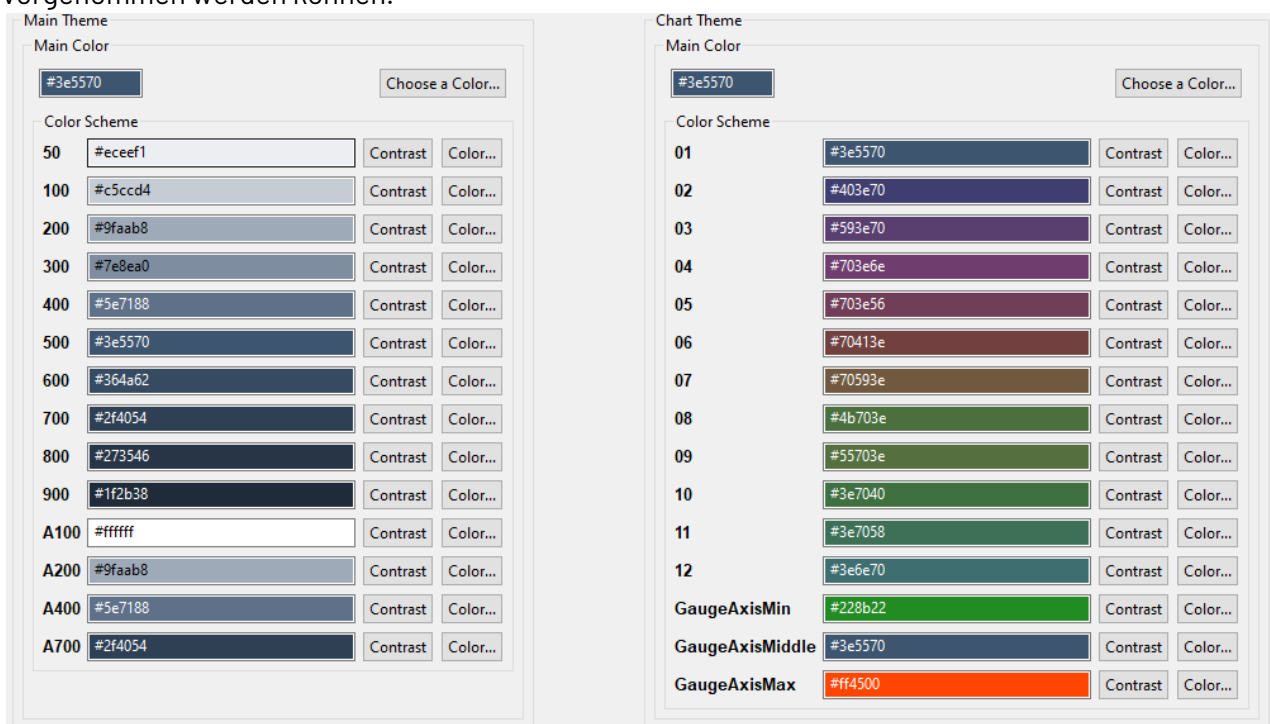
```
<?xml version='1.0' encoding='UTF-8'?>
<DataProtectionStatement xmlns="http://softproject.de/webapp/1.0" title="Privacy"
link="http://www.example.com">
  <FlowLayout>
    <Header value="Privacy" />
    <Label value="Data protection statement text"/>
  </FlowLayout>
  <Hint text="Statement text" buttonText="Button text"></Hint>
</DataProtectionStatement>
```

- ❗ Auch wenn die Seite für die Datenschutzerklärung hier keinen Inhalt hat, muss trotzdem ein Layout angegeben werden. Wird kein Layout angegeben, kann die Seite nicht gespeichert werden.

## 1.1.9 Theming

### 1.1.9.1 Farbschema festlegen

Das Farbschema der Web-Anwendung ist in der <Projektname>.theme-Datei festgelegt. Durch Doppelklicken auf die Datei wird der *Theme Editor* geöffnet, in dem die gewünschten Einstellungen vorgenommen werden können.



Im *Theme Editor* kann sowohl das Farbschema für die Anwendung als auch das Farbschema für Diagramme definiert werden.



- Im *Main Theme Designer* kann eine Hauptfarbe für das Farbschema vorgegeben werden, die weiteren Farben werden automatisch berechnet. Bei Bedarf lassen sich die weiteren Farben jedoch auch manuell einstellen. Als Grundlage wird das Designsystem Material Design ([material.io](https://material.io)) verwendet.

**i** Für den Hover-Effekt bei Buttons und Einträgen im Menü werden ebenfalls Farben aus der Palette verwendet.

- Button: Wechsel von 500 zu 400
- Menüeinträge: Wechsel von A100 zu A200

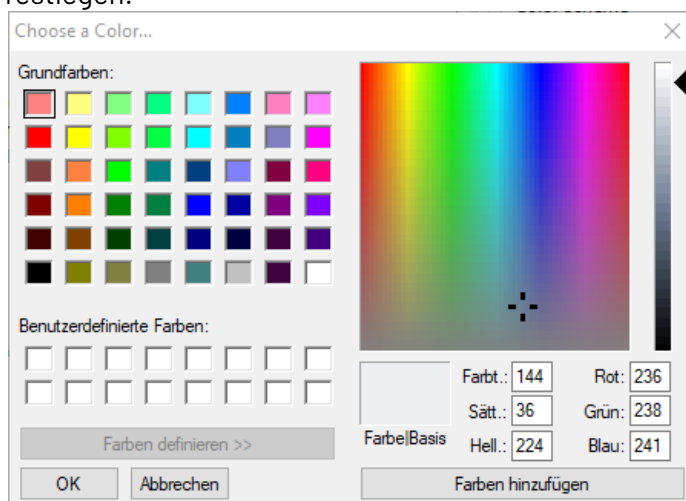
- Unter *Chart Colors* lassen sich die Farben für die verschiedenen Diagramme definieren. Durch Angabe einer Farbe unter *Main Charts Color* lässt sich eine Hauptfarbe definieren. Ausgehend von der Hauptfarbe werden die restlichen Farben automatisch berechnet. Die Farben lassen sich jedoch auch einzeln definieren.

- i**
- Die Farben 01–12 definieren die einzelnen Elemente eines Linien- (Line), Balken- (Bar,) und Kreis/Donut-Diagramms (Pie). Die Farben GaugeAxisMin, GaugeAxisMiddle und GaugeAxisMax definieren die drei Achsen des Tacho-Diagramms (Gauge).
  - Die hier definierten Farben werden verwendet, wenn das Attribut color bzw. GaugeAxisMin, GaugeAxisMiddle und GaugeAxisMax bei den verschiedenen Diagrammtypen nicht explizit definiert wurde.


**i** Die Farbwerte aus dem Farbschema einer Web App können verwendet werden, um bei [Aktionen](#) oder [Steuerelementen](#) benutzerdefinierte Vorder- und Hintergrundfarben zu hinterlegen. Damit können bestimmte Elemente hervorgehoben werden.

#### 1.1.9.1.1 Farben festlegen

1. Den *Theme Editor* durch Doppelklicken auf die <Projektname>.theme-Datei öffnen.
2. Neben der zu verändernden Farbe auf **Color** klicken.
3. Farbe aus der Palette auswählen oder auf **Farben definieren** klicken und eine Farbe individuell festlegen.



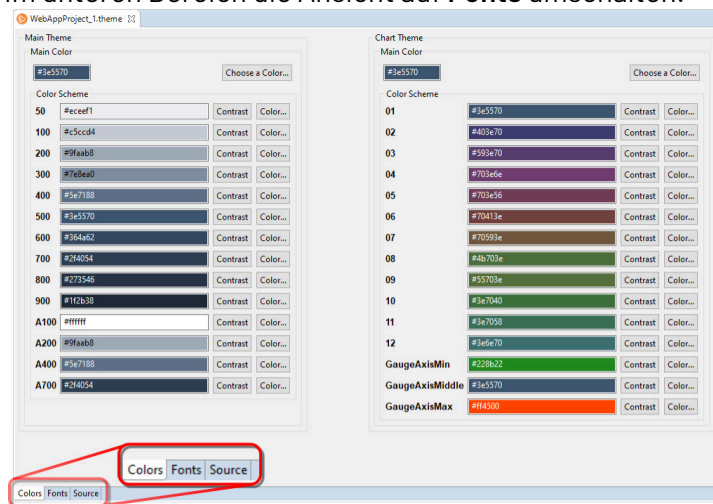
4. **Farbe hinzufügen** klicken und die angegebene Farbe zu den benutzerdefinierten Farben hinzufügen.
5. In **Benutzerdefinierte Farben** die Farbe auswählen und **OK** klicken.  
Die Farbe wurde nun übernommen.

 Mit **Contrast** kann die Schriftfarbe zwischen schwarz und weiß umgeschaltet werden.

### 1.1.9.2 Schriftarten festlegen

Für eine Web App können benutzerdefinierte Schriftarten festgelegt werden. Schriftarten werden über den Theme-Editor definiert.

1. Den *Theme Editor* durch Doppelklicken auf die <Projektname>.theme-Datei öffnen.
2. Im unteren Bereich die Ansicht auf **Fonts** umschalten.

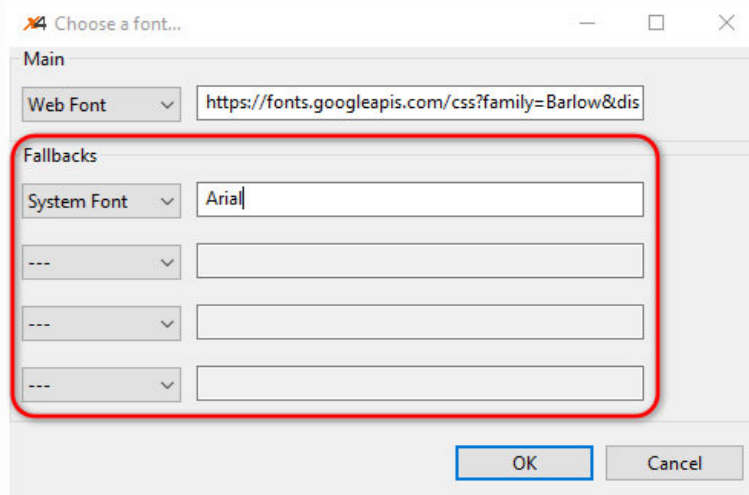


3. Unter **MainFont** wird die Schriftart eingestellt, die als Hauptschriftart der Web App verwendet wird. Hinter **MainFont** auf **Font...** klicken, um die Schriftart festzulegen.
4. Unter **Main** wählen, wo die Schrift zu finden ist.

#### Mögliche Werte:

- Resource Font: Schriftart wird als Datei in der Web App hinterlegt.

- ❗ Es können Dateien vom Typ eot, ttf, woff und woff2 verwendet werden. Die verschiedenen Dateitypen werden aber nicht von allen Browsern unterstützt. Im *Fonts Editor* können bis zu vier Ersatzschriften hinterlegt werden, falls es bei der Darstellung einer Schriftart zu Problemen kommt.



- System Font: Schriftart ist eine Systemschrift des Betriebssystems, z. B. Arial.
- Web Font: Schriftart wird über eine URL abgerufen, z. B.  
`https://fonts.googleapis.com/css?family=Barlow&display=swap`
- Generic: Schriftart ist eine Standardschrift des verwendeten Browsers, z. B. sans-serif.

5. Ggf. unter **Fallbacks** Ersatzschriften festlegen.
6. Mit Klick auf **OK** bestätigen.
7. <Projektname>.theme-Datei speichern.

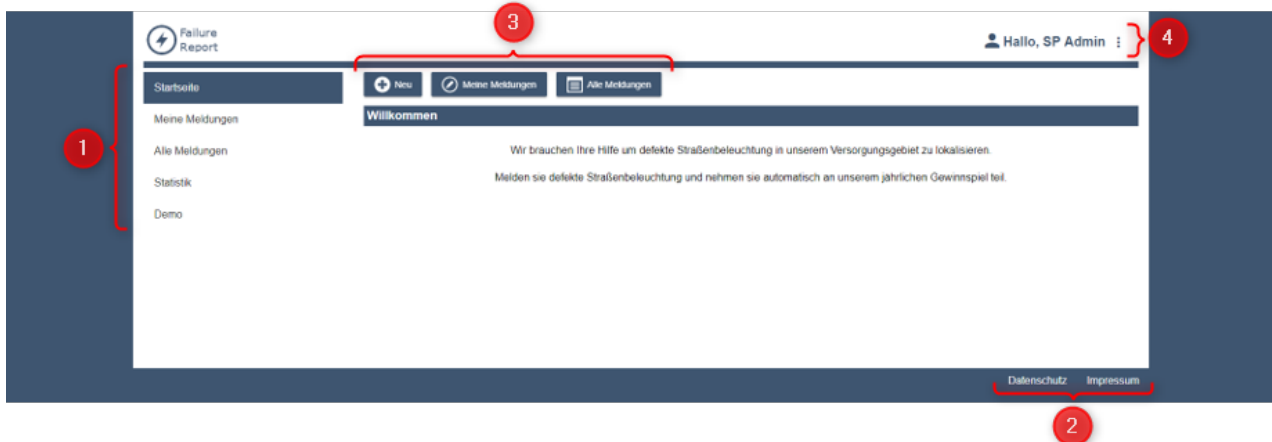
- ✓ Im *Fonts Editor* kann mit den **Custom Fonts** auch eine ganze Font-Palette angelegt werden. Schriften aus der Font-Palette können für einzelne **Steuerelemente** oder **Aktionen** verwendet werden.

### 1.1.10 Responsives Verhalten

Mit Web App erstellte Web-Anwendungen sind für die Darstellung sowohl auf Smartphones und Tablets als auch auf klassischen Desktop-PCs optimiert. Das responsive Verhalten betrifft:

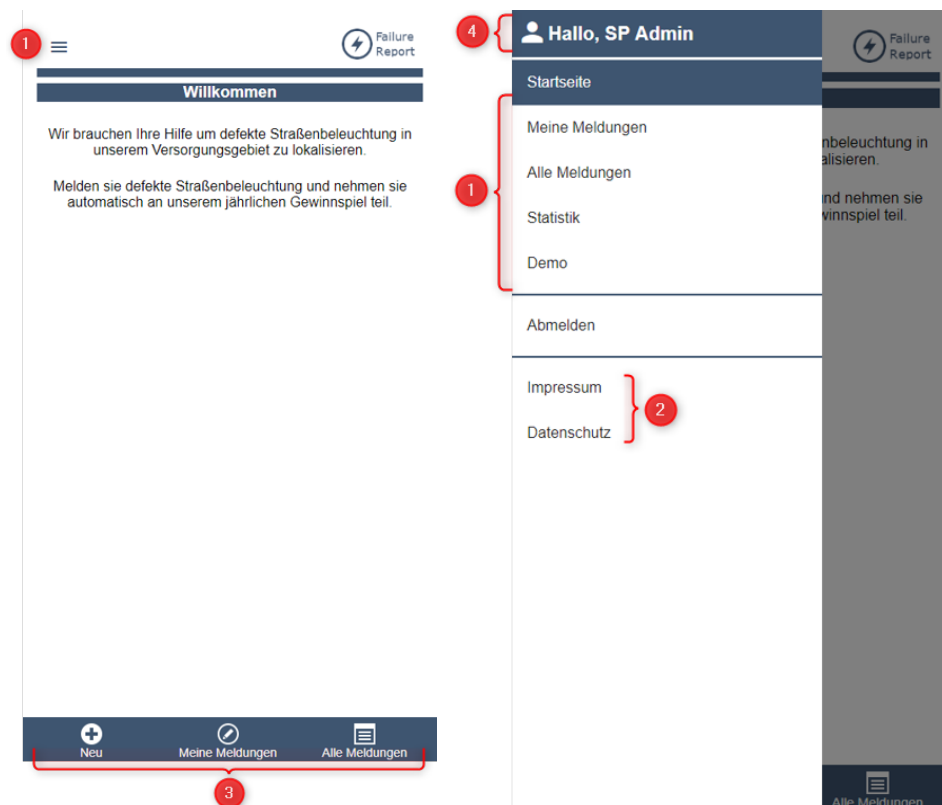
- das Seitenlayout
- das Aussehen von **Checkboxes**
- die Anzeige von Informationen innerhalb einer **List-Komponente**
- die Anzeige von **Master/Detail-Komponenten**

## 1.1.10.1 Seitenlayout am Desktop-PC



1	Menü, wird automatisch mithilfe der <b>Komponenten</b> erzeugt. Jede Komponente erzeugt einen Menüpunkt.
2	Links zu den Datenschutzerklärung bzw. zum Impressum. Links werden automatisch generiert und verweisen auf die Dateien <i>Projektname.dpr</i> bzw. <i>Projektname.imprint</i> . Wenn die Dateien nicht existieren, dann werden hier auch keine Links angezeigt.
3	Buttons für <b>Aktionen</b>
4	Angaben zum angemeldeten Nutzer

## 1.1.10.2 Mobiles Seitenlayout



1	Menü, wird automatisch mithilfe der <b>Komponenten</b> erzeugt. Jede Komponente erzeugt einen Menüpunkt. Das Menü wird im mobilen Modus erst durch Anklicken des Menü-Symbols sichtbar.
---	---

2	Links zu den Datenschutzerklärungen bzw. zum Impressum. Links werden automatisch generiert und verweisen auf die Dateien <i>Projektname.dpr</i> bzw. <i>Projektname.imprint</i> . Wenn die Dateien nicht existieren, dann werden hier auch keine Links angezeigt.
3	Buttons für <a href="#">Aktionen</a>
4	Angaben zum angemeldeten Nutzer

## 1.2 Lokalisierung

Web-Anwendungen, die mit Web Apps erstellt werden, können mehrsprachig zur Verfügung gestellt werden. Dazu muss zu Beginn der *.wad*-Datei festgelegt werden, in welchen Sprachen die Web-Anwendung verfügbar sein soll. Dabei muss der Wert, der für *name* angegeben wird, dem Dateinamen der Sprachdatei ohne Dateiendung entsprechen. Der Wert, der für *displayName* hinterlegt wird, wird bei der Sprachauswahl angezeigt:

```
<?xml version="1.0" encoding="UTF-8" ?>
<WebApp ...>
  <Translations>
    <Translation displayName="Deutsch" name="de"/>
    <Translation displayName="English" name="en"/>
    ...
  </Translations>
</WebApp>
```

Für jede Sprache, in der die Web-Anwendung verfügbar sein soll, muss eine entsprechende Übersetzungsdatei im Ordner *Translations* angelegt werden, die die Zuordnung einer Variable zu einer Benennung in einer Sprache vornimmt. Der Name der Übersetzungsdatei besteht aus dem Namen der Übersetzung (meist einem zweistelligen Sprachenkürzel) und der Dateiendung *translation*, z. B. *de.translation* (deutsche Übersetzungsdatei), *en.translation* (englische Übersetzungsdatei) oder *es.translation* (spanische Übersetzungsdatei).

### 1.2.1 Eine neue Sprachdatei anlegen

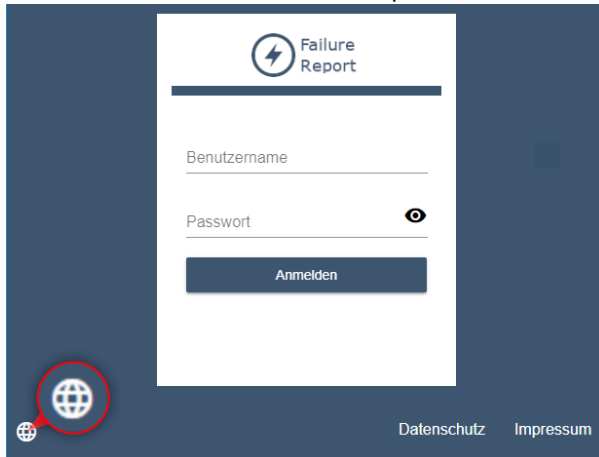
1. Auf den Ordner *Translations* rechtsklicken.  
Das Kontextmenü öffnet sich.
2. Auf **New > Translation** klicken.  
Der Dialog für das Erstellen einer neuen Sprachdatei öffnet sich.
3. Im Feld **File name** den Namen der Sprachdatei eingeben.
4. Auf **Finish** klicken.

Die Sprachdatei wurde angelegt. Die Sprachdatei ist mit vordefinierten Werten in deutscher Sprache befüllt und enthält alle möglichen Schlüssel. Die Sprachdatei ist folgendermaßen aufgebaut:

```
loginLabel = Benutzername
passwordLabel = Passwort
emailLabel = E-Mail
...
```

Die Sprachdatei besteht aus einem Schlüssel (z. B. `loginLabel`) und der zugeordneten Benennung (z. B. `Benutzername`). Viele Oberflächenelemente unterstützen Mehrsprachigkeit. Die entsprechenden Attribute müssen mit einem `$`-Zeichen und dem Schlüssel, beispielsweise `loginLabel="$LoginLabel"` gekennzeichnet werden. Welche Elemente bzw. Attribute Mehrsprachigkeit unterstützen, ist in den entsprechenden Referenzen vermerkt.

**i** Wenn für eine Web App mehrere Sprachen verfügbar sind, dann kann der Benutzer jederzeit zwischen den verschiedenen Sprachen wechseln.



### 1.2.2 Übersetzungsschlüssel

Folgende Übersetzungsschlüssel sind derzeit standardmäßig auf Deutsch hinterlegt und lassen sich bei Bedarf in beliebig viele Sprachen übersetzen:


```
loginLabel = Benutzername
passwordLabel = Passwort
emailLabel = E-Mail
loginButtonLabel = Anmelden
logoutButtonLabel = Abmelden
forgotPasswordLabel = Passwort vergessen?
changePasswordLabel = Passwort ändern
rememberMeLabel = Angemeldet bleiben
wrongPasswordLabel = Der eingegebene Benutzername oder das Passwort sind falsch.
changeButtonLabel = Ändern
required = Pflichtfeld
saveBtn = Speichern
deleteBtn = Löschen
newBtn = Neu
cancelBtn = Abbrechen
backButton=Zurück
notFoundToStart = Zur Startseite
notFoundText = Seite nicht gefunden
notFoundHeader = 404
componentNameNotDeclared = Der Komponententname in der Aktion ist nicht deklariert.
forbiddenHeader = 403
forbiddenText = Sie haben keine Berechtigung für den Zugriff. Wenden Sie sich an den
Administrator.
dataReloaded = Der ausgewählte Datensatz wurde im Hintergrund geändert. Die Seite
wird neu geladen.
changePasswordButtonOk = Speichern
changePasswordButtonCancel = Abbrechen
changePasswordInvalid = Überprüfen Sie Ihr Passwort
changePasswordSuccess = Ihr Passwort wurde geändert
changePasswordWrong = Passwort entspricht nicht den Passwortrichtlinien
changePasswordReused = Passwort wurde bereits verwendet
changePasswordDoNotMatch = Passwörter stimmen nicht überein
changePasswordNewPassword = Neues Passwort
changePasswordOldPassword = Altes Passwort
changePasswordRepeat = Wiederholen
forgotPasswordButtonOk = Senden
forgotPasswordButtonCancel = Zurück
forgotPasswordRestoreField = E-Mail
greetLabel = Hallo,
errorHeader = Fehler
passwordNotChanged = Das Passwort wurde nicht geändert. Bitte überprüfen Sie Ihre
Eingabe.
requestError = Bei der Anfrage ist leider ein Fehler aufgetreten. Bitte versuchen Sie
es zu einem späteren Zeitpunkt nochmals.
passwordEmailSent = Wir haben eine E-Mail mit weiteren Anweisungen verschickt.
leavePageWithUnsavedDataHeader = Möchten Sie diese Seite wirklich verlassen?
leavePageWithUnsavedDataText = Beim Verlassen der Seite werden Ihre Änderungen
verworfen.
leavePageWithUnsavedDataOkButton = Seite verlassen
leavePageWithUnsavedDataCancelButton = Auf Seite bleiben
uploadBtn = Hochladen
downloadBtn = Herunterladen
multipleNavigationHeader = Navigationsfehler
multipleNavigationText = Es kann nur ein Navigationspfad definiert werden.
next=Vor
prev=Zurück
```

```
today=Heute
monthView=Monat
weekView=Wochen
dayView=Tag
monday=Montag
tuesday=Dienstag
wednesday=Mittwoch
thursday=Donnerstag
friday=Freitag
saturday=Samstag
sunday=Sonntag
mondayShort=Mo
tuesdayShort=Di
wednesdayShort=Mi
thursdayShort=Do
fridayShort=Fr
saturdayShort=Sa
sundayShort=So
january=Januar
february=Februar
march=März
april=April
may=Mai
june=Juni
july=Juli
august=August
september=September
october=Oktober
november=November
december=Dezember
start=Start
end=Ende
location=Ort
description=Beschreibung
allDay=Ganztägig
destination=Ziel
search=Suche
routingErrorTitle=Fehler
routingError=Die Route konnte nicht berechnet werden.
languagesLabel=Sprachauswahl
```

## 1.3 Benutzerrechte für Web Apps verwalten

Die Benutzer bzw. Benutzerrechte werden an zwei Stellen verwaltet:

- Im *X4 Control Center* werden die Benutzer und Benutzergruppen angelegt oder gelöscht.

 Nähere Informationen zum Verwalten von Benutzern und Benutzergruppen im X4 Control Center finden sich in den entsprechenden Kapiteln.

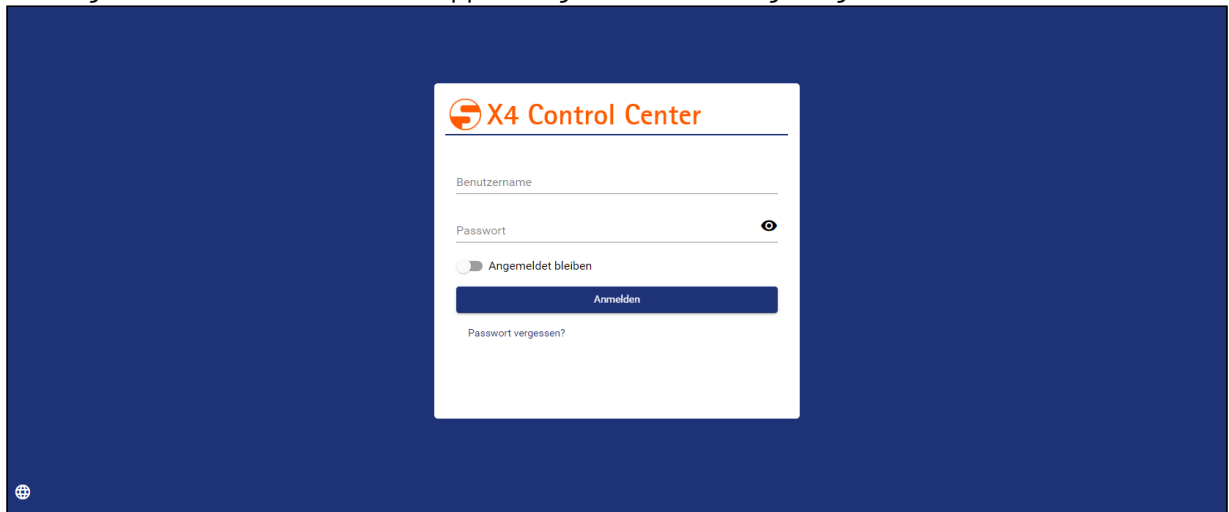
- Im *X4 Web App Management* werden die in der Web App definierten [Zugriffsrechte](#) den Benutzern und Benutzergruppen zugewiesen, siehe [Zugriffsrechte über das X4 Web App Management verwalten](#).




## 1.3.1 Zugriffsrechte über das X4 Web App Management verwalten

### 1.3.1.1 Am X4 Web App Management anmelden

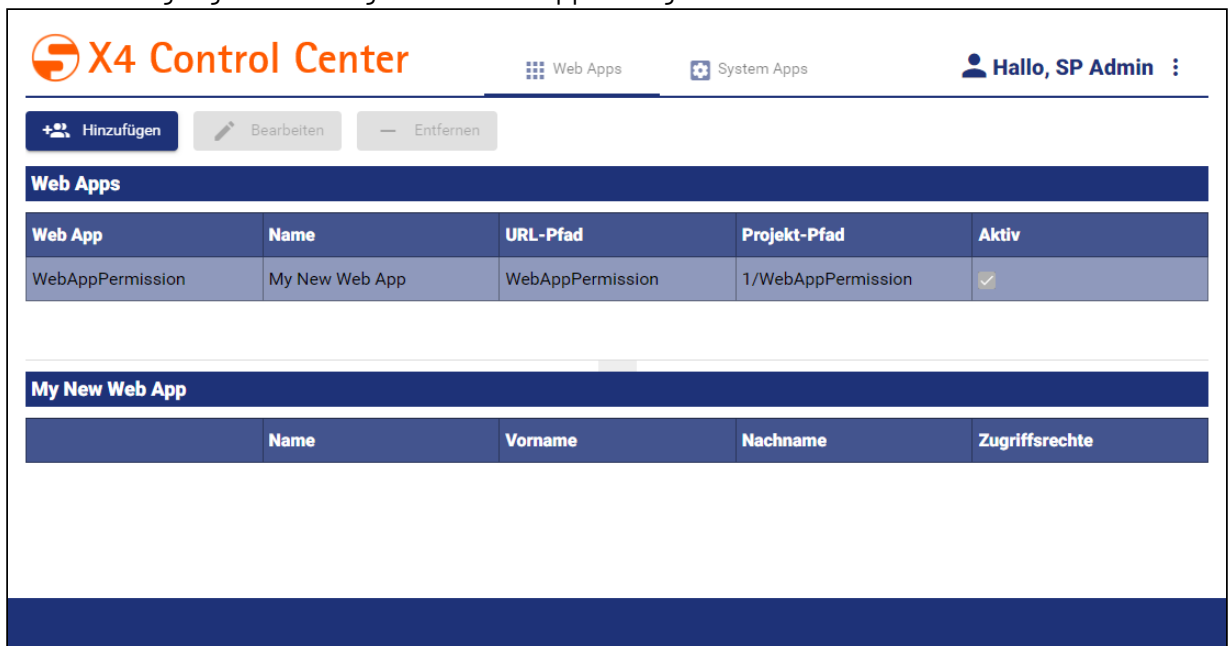
1. Im X4 Designer auf **Tools > X4 Web App Management** klicken.  
Der Login-Bildschirm des X4 Web App Managements wird angezeigt.



2. Mit Benutzername und Passwort anmelden.

 Die Standard-Zugangsdaten sind Benutzername admin und Passwort demo.

3. Die Berechtigungsverwaltung des X4 Web App Managements öffnet sich.



Web App	Name	URL-Pfad	Projekt-Pfad	Aktiv
WebAppPermission	My New Web App	WebAppPermission	1/WebAppPermission	✓

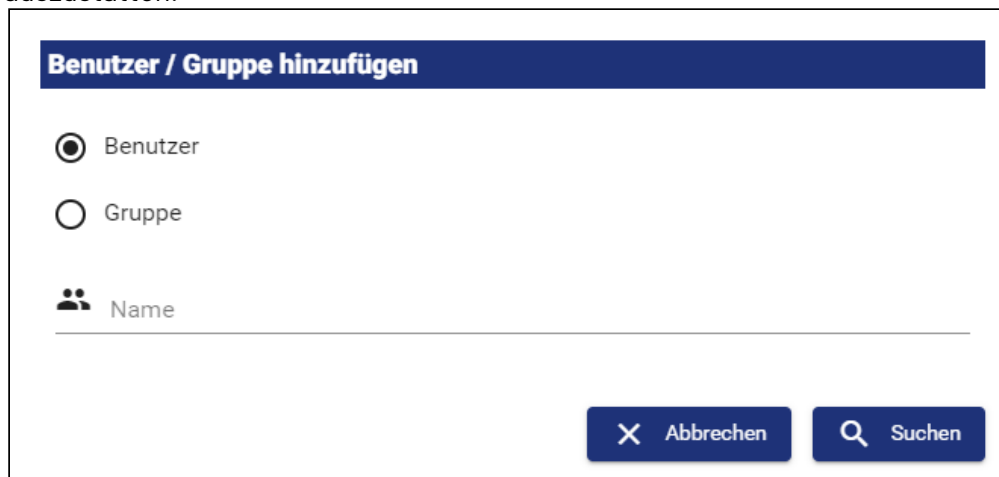
  

My New Web App				
	Name	Vorname	Nachname	Zugriffsrechte

- In der Komponente **Web Apps** werden alle im X4 Repository verfügbaren Web-App-Projekte angezeigt. Hier lassen sich auch die im X4 Control Center verfügbaren Benutzer und Gruppen den einzelnen Projekten zuweisen und mit Zugriffsrechten ausstatten.
- In der Komponente **System Apps** lassen sich bei Bedarf Zugriffsrechte für die System-Web-Apps vergeben.

### 1.3.1.2 Zugriffsrechte an Benutzer oder Gruppe vergeben


1. Die Web-Anwendung, für die Zugriffsrechte vergeben werden sollen, auswählen.  
In der Detailansicht werden die bereits zugewiesenen Benutzer bzw. Gruppen angezeigt.
2. Auf **Hinzufügen** klicken, um einen neuen Benutzer oder eine neue Gruppe mit Zugriffsrechten auszustatten.



**Benutzer / Gruppe hinzufügen**

☒ Benutzer

☐ Gruppe

 Name

3. Angeben, ob ein **Benutzer** oder eine **Gruppe** hinzugefügt werden soll.
4. Den Namen des Benutzers oder der Gruppe eingeben, um nach einem bestimmten Benutzer / einer Gruppe zu suchen.

- Wird kein Benutzer- oder Gruppenname eingegeben, werden alle verfügbaren Benutzer und Gruppen angezeigt.

5. **Suchen** klicken, um die Benutzer bzw. Gruppen anzuzeigen.

Die verfügbaren Benutzer/Gruppen werden angezeigt.


Benutzer / Gruppe hinzufügen				
	Name	Vorname	Nachname	E-Mail
	admin	SP	Admin	
	Administrator	Max	Mustermann	
	StaffMember	Brigitte	Beispiel	
	Customer	Andreas	Custom	

← Zurück
✕ Abbrechen
✓ Auswählen

6. Benutzer bzw. Gruppe, die Zugriffsrechte für die gewählte Web-Anwendung bekommen sollen, auswählen.


7. **Auswählen** klicken.

Der Dialog zum Zuweisen der in der [Web App](#) definierten [Zugriffsrechte](#) wird geöffnet.

Benutzer / Gruppe hinzufügen	
	<b>Zugriffsrechte</b> Auch wenn keine Zugriffsrechte ausgewählt werden, kann sich der Benutzer / die Gruppe an der Web App anmelden.
<input type="checkbox"/>	Administrator
<input type="checkbox"/>	Customer
<input type="checkbox"/>	Staff

← Zurück
✕ Abbrechen
+ Benutzer hinzufügen

8. Ein oder mehrere Zugriffsrechte für den Benutzer / die Gruppe auswählen.

 Der Benutzer kann sich an der Web App anmelden, selbst wenn kein Zugriffsrecht zugewiesen wurde.

9. **Hinzufügen** klicken.

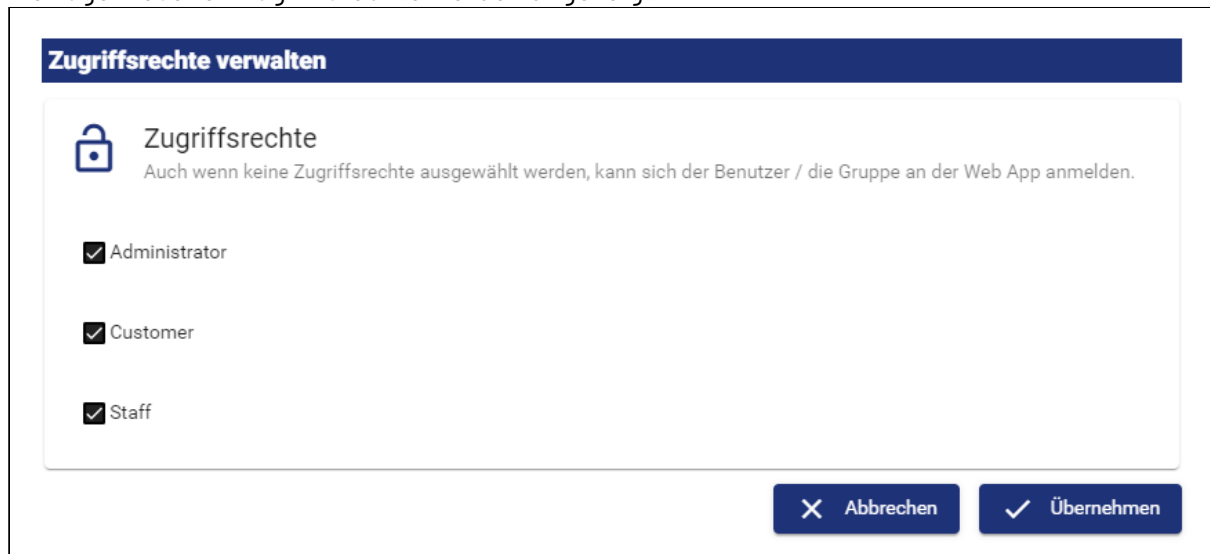
Für den Benutzer bzw. die Benutzergruppe wurden Zugriffsrechte vergeben.

### 1.3.1.3 Zugriffsrechte bearbeiten

1. Die Web-Anwendung, für die die Zugriffsrechte bearbeitet werden sollen, auswählen.  
In der Detailansicht werden die bereits zugewiesenen Benutzer bzw. Gruppen angezeigt.
2. Benutzer bzw. Gruppe in der Detailansicht auswählen.

### 3. **Bearbeiten** klicken.

Die zugewiesenen Zugriffsrechte werden angezeigt.



### 4. Gewünschte Änderungen vornehmen und **Übernehmen** klicken.


Die Änderungen werden nun übernommen.





#### 1.3.1.4 Zugriffsrechte löschen

1. Die Web-Anwendung, für die die Zugriffsrechte gelöscht werden sollen, auswählen.  
In der Detailansicht werden die bereits zugewiesenen Benutzer bzw. Gruppen angezeigt.
2. Zu löschenden Benutzer bzw. Gruppe in der Detailansicht auswählen.
3. **Entfernen** klicken.
4. Den Dialog für das Löschen von Berechtigungen mit **Entfernen** bestätigen.  
Die Zugriffsrechte für den Benutzer bzw. die Benutzergruppe wurden gelöscht.

## 1.4 Module

Web-Anwendungen, die mit Web Apps erstellt wurden, bestehen aus Modulen, die wiederum **Komponenten** enthalten. Wie für die Web-Anwendung selbst können für die einzelnen Module folgende Attribute vergeben werden:


Attribut	Beschreibung
accessRight	<p>Definiert, welche Berechtigung auf das Modul zugreifen können.</p> <p><b>Mögliche Werte:</b> Name einer zuvor definierten Berechtigung</p> <div> <p> Für mehr Informationen, siehe auch <a href="#">Zugriffsrechte für Module definieren</a>.</p> </div>

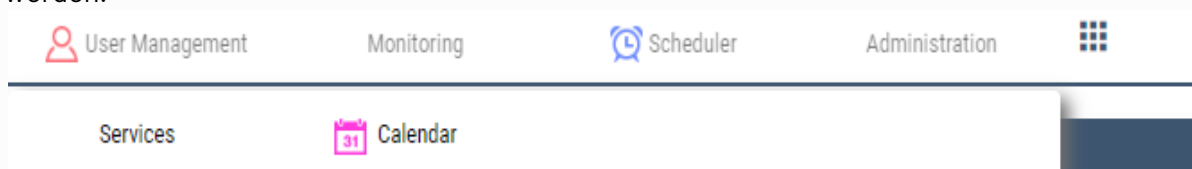
Attribut	Beschreibung
default	<p>Definiert, ob das Modul zuerst geladen und geöffnet werden soll.</p> <div>  Lediglich ein Modul darf den Wert <code>default="true"</code> haben. </div> <p><b>Mögliche Werte:</b> true / false</p>
displayName	<p>Anzeigename des Moduls. Wird im Browsertab angezeigt.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
hideOnAccessDenied	<p>Gibt an, ob Benutzer/Gruppen, die keinen Zugriff auf das Modul haben, das Modul sehen können.</p> <p><b>Mögliche Werte:</b> true / false</p> <div>  Für mehr Informationen, siehe auch <a href="#">Zugriffsrechte für Module definieren</a>. </div>
iconColor	<p>Definiert die Farbe des Icons in der Anwendung.</p> <div>  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Hexadezimaler Farbwert, z. B. <code>ff5a00</code></li> </ul> <div>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! </div> <ul style="list-style-type: none"> <li>Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. <code>A200</code></li> </ul>

Attribut	Beschreibung
iconUrl	<p>Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Icon des Moduls verwendet wird.</p> <div> <p><b>i</b></p> <ul style="list-style-type: none"> <li>Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources.</li> <li>Das Material Icon muss mit dem Präfix icon, z. B. icon:&lt;MaterialIconName&gt; angegeben werden.</li> </ul> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Zeichenkette (URI), z. B. <i>clock.png</i></li> <li>icon:&lt;MaterialIconName&gt;, z. B. <i>icon:extension</i></li> </ul> <div> <p><b>✓</b> Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren <a href="#">Material Icons</a> abweichen.</p> </div>
path	<p>Pfad zum Modul. Wird in der Adresszeile des Browsers angezeigt.</p> <p><b>Mögliche Werte:</b> Zeichenkette aus alphanumerischen Zeichen (keine Umlaute, Punkte usw.)</p>

Für eine Web-Anwendung können mehrere Module definiert werden. Sie werden im Kopfbereich der Anwendung angezeigt.

- ✓** Für die einzelnen Module können auch Zugriffsrechte definiert werden. Wie das geht wird im Abschnitt [Zugriffsrechte für Module definieren](#) erläutert.

- i** Es werden immer die ersten vier Module in der Oberfläche angezeigt. Wurden mehr als vier Module definiert, können diese über das Untermenü  eingeblendet und ausgewählt werden.



## Beispiel

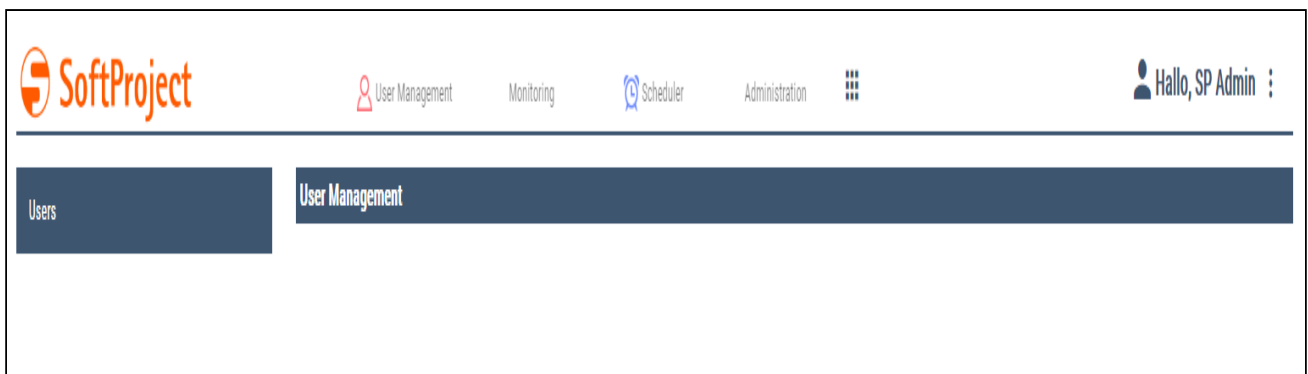
**Beispiel: Web App mit mehreren Modulen**

```

<WebApp xmlns="http://softproject.de/webapp/1.0" displayName="My Web App" path="Modul
es" logoUrl="logo.png">
  <Modules>
    <Module displayName="User Management" path="UserManagement" iconUrl="users.pn
g" default="true" >
      <Components>
        <DetailComponent default="true" displayName="Users" path="Users">
          <FlowLayout>
            <Header value="User Management"/>
          </FlowLayout>
        </DetailComponent>
      </Components>
    </Module>
    <Module displayName="Monitoring" path="Monitoring">
      ...
    </Module>
    <Module displayName="Scheduler" path="Scheduler" iconUrl="clock.png">
      ...
    </Module>
    <Module displayName="Administration" path="Administration">
      ...
    </Module>
    <Module displayName="Services" path="Services">
      ...
    </Module>
    <Module displayName="Calendar" path="Calendar" iconUrl="calendar.png">
      ...
    </Module>
  </Modules>
</WebApp>

```

Das oben aufgeführte Beispiel führt zu folgendem Ergebnis:

**Desktop-Ansicht:****Mobile Ansicht:**

The screenshot displays the X4 Admin web interface. The left sidebar is a dark blue vertical menu with a white header. The header contains a user icon and the text 'Hallo, SP Admin'. Below the header, the menu is organized into sections. The first section is 'User Management', indicated by a red person icon, with a sub-item 'Users' highlighted in a darker blue. The next section is 'Monitoring', with sub-items 'Instances' and 'Scheduler' (marked with a blue clock icon). This is followed by 'Executions'. The 'Administration' section includes 'CustomPlaceholder'. The 'Services' section lists 'Root'. The 'Calendar' section, marked with a pink calendar icon, includes 'Calendar'. At the bottom of the sidebar are two ungrouped items: 'Passwort ändern' and 'Abmelden'. The main content area on the right is a light gray, showing a blurred view of the 'Project' page.

**Hallo, SP Admin**

- User Management
  - Users**
- Monitoring
  - Instances
  - Scheduler
  - Executions
- Administration
  - CustomPlaceholder
- Services
  - Root
- Calendar
  - Calendar
- Passwort ändern
- Abmelden



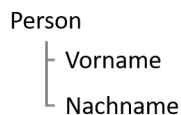
## 1.5 Datenmodell und Properties-Definition

Für jede Komponente können Properties definiert werden. Properties definieren das Datenmodell, das in der Komponente verwendet werden soll. Für **List-Komponenten** (eigenständig oder innerhalb einer **Master/Detail-Komponente**) sind Informationen über das Datenmodell, in dem die Daten von einem Prozess geliefert werden, zwingend erforderlich. Innerhalb einer **Master/Detail-Komponente** werden Daten zwischen den unterschiedlichen Komponenten geteilt (siehe **Properties in Master/Detail**).


Properties werden am Beginn der Komponenten-Deklaration innerhalb von `Properties` definiert:

```
<DetailComponent>
  <Properties>
    <Property name="Person" type="Complex">
      <Property name="Vorname" type="String"/>
      <Property name="Nachname" type="String"/>
    </Property>
  </Properties>
  <FlowLayout>
    <Label value="#Person.Vorname"/>
    <Label value="#Person.Nachname"/>
  </FlowLayout>
</DetailComponent>
```





Das hier definierte Datenmodell entspricht folgendem Schema:



### Mögliche Attribute:

Attribut	Beschreibung
displayName	Anzeigenname der Property. Wird in den Spaltenüberschriften von Master/Detail- und List-Komponenten verwendet. <b>Mögliche Werte:</b> Beliebige Zeichenkette
name	Name der Property. Erforderlich. <b>Mögliche Werte:</b> Beliebige Zeichenkette aus alphanumerischen Zeichen (a-z, A-Z, 0-9) <div> Beachten Sie, dass die Property-Namen <code>Component</code>, <code>OrderBy</code>, <code>Where</code>, <code>FileId</code>, <code>FileName</code>, <code>FileType</code>, <code>FileData</code>, <code>To</code>, <code>From</code> nicht verwendet werden dürfen, da sie bereits vorbelegt sind und daher validiert werden.</div>

Attribut	Beschreibung
readOnly	Schränkt den Zugriff auf Nur-Lesen ein.  <b>Mögliche Werte:</b> <i>true / false</i>

Attribut	Beschreibung
type	<p data-bbox="544 320 932 353">Typ der Property. Erforderlich.</p> <p data-bbox="544 387 756 421"><b>Mögliche Werte:</b></p> <ul data-bbox="560 465 1331 757" style="list-style-type: none"> <li>• Base64: Base64-kodierte Daten</li> <li>• Boolean: Boole'sche Werte</li> <li>• Color: Hexadezimale Farbwerte</li> <li>• Complex: Objekt enthält weitere Objekte (verschachtelte Properties). Wird z. B. für Listeneinträge verwendet.</li> <li>• Date: Datumsangabe im ISO-Format YYYY-MM-DD (z. B. 2020-01-18)</li> <li>• DateTime: Datums- und Zeitangaben im Format YYYY-MM-DD'T'hh:mm:ss.fff (z. B. 2019-01-01T01:00:00.000Z)</li> </ul> <div data-bbox="612 801 1390 869" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p> Das Format der Datums- und Zeitangaben muss konform zu ISO 8601 sein.</p> </div> <div data-bbox="612 936 1401 1294" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p> Wenn ein Wert mit dem Property-Typ DateTime in einer <a href="#">List-Komponente</a> verwendet wird, dann wird der Angabe von Datum und Uhrzeit abhängig von den vorhandenen Werten unterschiedlich angezeigt:</p> <ul style="list-style-type: none"> <li>• Wenn alle Werte den Wert 0 bei Sekunden haben (2020-01-01T00:00:00.000Z), dann werden in der gesamten Spalte keine Sekunden angezeigt.</li> <li>• Wenn mindestens ein Wert einen Wert abweichend von 0 hat, werden die Sekunden in der gesamten Spalte angezeigt.</li> </ul> </div> <div data-bbox="612 1339 1394 1485" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0; background-color: #fff9c4;"> <p> Die Verarbeitung des Property-Typs DateTime wurde mit der Version 6.3 angepasst. Bei allen bisherigen Angaben vom Property-Typ DateTime wird ab Version 6.3 nicht mehr nur das Datum, sondern auch die Uhrzeit angezeigt!</p> </div> <ul data-bbox="560 1529 1358 1630" style="list-style-type: none"> <li>• Decimal: Kommazahlen</li> <li>• Image: Grafik (Base64, Ressource im Ordner <i>Resources</i> oder externe URL)</li> </ul> <div data-bbox="612 1697 1401 1798" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0; background-color: #fff9c4;"> <p> Wenn die Grafik als Base64-kodierte Zeichenkette hinterlegt wird, dann muss zusätzlich das Attribut <code>mediaType="image/*"</code> angegeben werden.</p> </div> <ul data-bbox="560 1843 1401 1989" style="list-style-type: none"> <li>• Integer: Ganzzahlen</li> <li>• List: Enthält Listenelemente. Wird für <a href="#">ComboBox</a> und <a href="#">ListView</a> verwendet.</li> <li>• String: Zeichenkette</li> </ul>

### 1.5.1 Daten aus dem Prozess

Daten können von einem Prozess bereitgestellt werden, um beispielsweise dynamisch Listen zu generieren. Diese Daten müssen dem durch die Properties definierten Format entsprechen. Die Daten werden per [Data Binding](#) an die Oberflächenelemente gebunden. Innerhalb der XML-Datei, die von einem Prozess bereitgestellt wird, entspricht jedes Element einer Property. Der Wert, den das Element enthält, ist der Wert der Property.

```
<Ok>
  <Person>
    <Vorname>Max</Vorname>
    <Nachname>Mustermann</Nachname>
  </Person>
</Ok>
```

Die XML-Daten, die der Prozess liefert, haben als Wurzelement entweder `<Ok>` oder `<OkList>`. `<Ok>` wird für Detail-Komponenten verwendet, `<OkList>` für List-Komponenten.

### 1.5.2 Beispiel für eine einfache Properties-Definition

Folgendes Beispiel zeigt eine Property mit dem Namen *Example* vom Typ *String*.

```
<DetailComponent>
  <Properties>
    <Property name="Example" type="String" />
  </Properties>
  <FlowLayout>
    ...
  </FlowLayout>
</DetailComponent>
```

### 1.5.3 Beispiel für eine komplexe Properties-Definition

Properties können verschachtelt werden. Eine Property, die eine oder mehrere andere Properties enthält, ist immer vom Typ *Complex*.

Das folgende Beispiel zeigt eine komplexe Property:

### Beispiel für eine komplexe Properties-Definition

```
<DetailComponent>
  <Properties>
    <Property name="Example" type="String" />
    <Property name="Example2" type="Complex">
      <Property name="Internal" type="String"/>
      <Property name="Visible" type="Boolean"/>
    </Property>
  </Properties>
  <FlowLayout>
    <Label value="#Example" />
    <Checkbox checked="#Example2.Visible" displayName="Visible"/>
    <Label value="#Example2.Internal" visible="#Example2.Visible"/>
  </FlowLayout>
</DetailComponent>
```

### Beispiel XML-Datei mit komplexen Properties

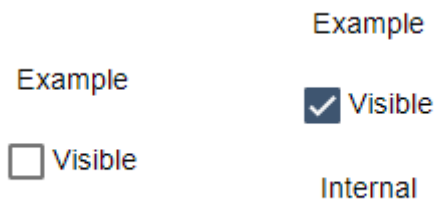
```
<Ok>
  <Example>Example</Example>
  <Example2>
    <Internal>Internal</Internal>
    <Visible>false</Visible>
  </Example2>
</Ok>
```

### Beispiel Data Binding

```
<DetailComponent process="fillDetail.wrf">
  <Properties>
    ...
  </Properties>
  <FlowLayout>
    <Label value="#Example" />
    <Checkbox checked="#Example2.Visible" displayName="Visible" />
    <Label value="#Example2.Internal" visible="#Example2.Visible" />
  </FlowLayout>
</DetailComponent>
```

### Ergebnis

Das Ergebnis dieses Beispiels ist eine Detail-Komponente, die Daten aus `detailData.xml` anzeigt:



**⚠** Mit der Version 6.2.0 der X4 Suite wurde das Datenmodell deutlich vereinfacht. Projekte, die in einer älteren Version erstellt wurden, müssen migriert werden, um den Simple Data Transport zu verwenden. Alternativ kann die Option **Simple Data Transport** für ein Web-App-Projekt deaktiviert werden. Wählen Sie dazu das Projekt im X4 Designer aus und deaktivieren Sie die Option **Simple Data Transport** in den **Properties**.

### 1.5.4 Validierung

Properties unterstützen Validierungen. Diese erlauben Regeln festzulegen, die für die Properties-Werte gelten müssen.

Um Validierungsregeln festzulegen, muss innerhalb des Property-Elements das Validation-Element verwendet werden. Das Validation-Element kann beliebig viele Validation-Elemente mit Validierungsregeln und Meldungen enthalten.

```
<Property name="test" type="Integer">
  <Validations>
    <Validation>
      <Value operator="eq" expected="0" />
      <Message>The entered value is not correct.</Message>
    </Validation>
  </Validations>
</Property>
```

Das obige Beispiel zeigt eine Validierungsregel. Anhand dieser Regel wird geprüft, ob der Wert, der für die Property test angegeben wurde 0 ist. Mit Message wird eine entsprechende Meldung ausgegeben.

**i** Validierungen beeinflussen, ob die **Speichern-Aktion** aktiv ist. Wenn die Validierung nicht bestanden wird, dann ist die Speichern-Aktion innerhalb der Komponente nicht aktiv. Komponenten, die ungültige Eingaben enthalten, können also nicht gespeichert werden.

#### 1.5.4.1 Mögliche Validierungsarten

Folgende Validierungsarten können als Kindelemente innerhalb des Validation-Elements verwendet werden:

Element	Beschreibung
Contains	Prüft, ob ein bestimmter Wert in einem Property-Wert enthalten ist. Mit dem Attribut <code>value</code> wird der entsprechende Wert angegeben.
Email	<i>Keine Attribute erforderlich.</i> Prüft, ob der Property-Wert eine gültige E-Mail-Adresse ist.
EndsWith	Prüft, ob der Property-Wert mit einem bestimmten Wert endet. Mit dem Attribut <code>post fix</code> wird der entsprechende Wert angegeben.
Max	Definiert einen maximalen Wert für den Property-Wert. Mit dem Attribut <code>value</code> wird der entsprechende Wert angegeben.
MaxLength	Definiert eine maximale Zeichenlänge für den Property-Wert. Mit dem Attribut <code>value</code> wird der entsprechende Wert angegeben.
Message	Definiert die Meldung, die als Ergebnis der Validierung angezeigt werden soll.
Min	Definiert einen minimalen Wert für den Property-Wert. Mit dem Attribut <code>value</code> wird der entsprechende Wert angegeben.
MinLength	Definiert eine minimale Zeichenlänge für den Property-Wert. Mit dem Attribut <code>value</code> wird der entsprechende Wert angegeben.
Regex	Prüft, ob der Property-Wert einem bestimmten Muster entspricht. Mit dem Attribut <code>pattern</code> wird das entsprechende Muster angegeben.
Required	<i>Keine Attribute erforderlich.</i> Prüft, dass der Property-Wert nicht leer ist.
StartsWith	Prüft, ob der Property-Wert mit einem bestimmten Wert beginnt. Mit dem Attribut <code>prefix</code> wird der entsprechende Wert angegeben.
Value	Prüft anhand eines Operators (Attribut <code>operator</code> ), ob der eingegebene Wert dem erwarteten Wert ( <code>expected</code> ) entspricht.

#### 1.5.4.2 Mögliche Operatoren

Für Validierungsregeln stehen verschiedene Operatoren zur Verfügung:

Operator	Beschreibung
<code>eq</code>	gleich ( <i>equal</i> )
<code>gt</code>	größer als ( <i>greater than</i> )
<code>gt_eq</code>	größer als oder gleich ( <i>greater than or equal</i> )
<code>lt</code>	kleiner als ( <i>less than</i> )
<code>lt_eq</code>	kleiner oder gleich ( <i>less than or equal</i> )
<code>neq</code>	ungleich ( <i>not equal</i> )

Das expected-Attribut ist der sogenannte Kontrollwert. Dieser Wert wird verwendet, um die Validierung anzuwenden.

*Beispiel:* `<Value operator="gt" expected="0" />` bedeutet, dass der Wert einer Property daraufhin geprüft wird, ob er größer als 0 ist.

#### 1.5.4.3 Beispiel logische Verknüpfungen

Das folgende Beispiel zeigt, wie Validierungen logisch verknüpft werden können. Wenn innerhalb eines Validation-Elements mehrere Regeln definiert sind, dann werden diese mit UND verknüpft. Wenn innerhalb eines Validations-Elements mehrere Validation-Elemente inkl. Regeln enthalten sind, dann werden diese mit ODER verknüpft.

##### Beispiel Validierung

```
<Property name="test" type="Integer">
  <Validations>
    <Validation>
      <Value operator="eq" expected="0" />
    </Validation>
    <Validation>
      <Value operator="gt_eq" expected="10" />
      <Value operator="lt_eq" expected="15" />
    </Validation>
  </Validations>
</Property>
```

In diesem Beispiel ist die Validierung bestanden, wenn der Wert der Property test gleich 0 ODER gleichzeitig größer oder gleich 10 UND kleiner oder gleich 15 ist.

#### 1.5.4.4 Beispiele <Message>

Innerhalb des Message-Elements kann eine Meldung definiert werden. Diese Meldung wird im Eingabe-Steuerelement angezeigt, an das die zu validierende Property gebunden ist.

##### Beachten Sie folgende Vorgaben!

- Wird das Message-Element direkt innerhalb des Validations-Elements definiert, wird die Nachricht immer angezeigt.
- Wird kein Message-Element innerhalb des Validations-Elements definiert, wird die definierte Meldung aus dem zutreffenden Validation-Elementen ausgegeben.
- Wurde kein Message-Element innerhalb des Validations-Elements definiert, müssen für alle Validation-Elemente Meldungen hinterlegt werden.


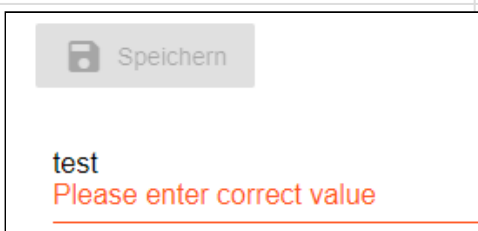
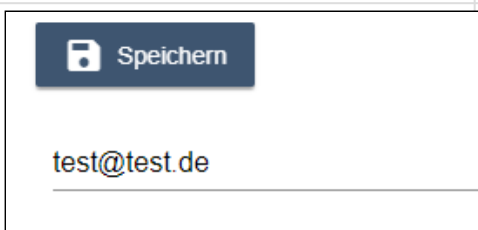
- Die Meldung wird direkt innerhalb des Validations-Elements definiert:



### Beispiel: Message wird direkt innerhalb des Validations-Elements angegeben

```
<Property name="test" type="Integer">
  <Validations>
    <Message>Please enter correct value</Message>
    <Validation>
      <Email/>
    </Validation>
  </Validations>
</Property>
```

Obiges Beispiel ergibt folgende Meldungen:

	Validierung schlägt fehl, da keine Eingabe erfolgt ist.
	Validierung schlägt fehl, da keine gültige E-Mail-Adresse hinterlegt wurde.
	Validierung ist erfolgreich. Es wird keine Meldung angezeigt.

- Für jede Validierungsregel wird eine Meldung hinterlegt:

```
<Validations>
<!-- <Message>Too long</Message> -->
  <Validation>
    <Message>End with fin and contains z</Message>
    <EndsWith postfix="fin" />
    <Contains value="z" />
  </Validation>
  <Validation>
    <Message>Too long</Message>
    <MaxLength value="10" />
  </Validation>
</Validations>
```

## 1.6 Komponenten

Web-Anwendungen bestehen aus verschiedenen **Komponenten**, die jeweils in einer separaten Datei definiert werden. Anschließend werden die Komponenten innerhalb von Components über das Element `<ComponentReference>` referenziert.

Es gibt verschiedene Arten von Komponenten:

Komponente	Tag	Beschreibung
Detail-Komponente	DetailComponent	Zeigt statischen Text an
List-Komponente	ListComponent	Zeigt Daten in Form von dynamisch erzeugten Listen an
Kalender-Komponente	CalendarComponent	Zeigt einen Kalender an
Master/Detail-Komponente	MasterDetailComponent	Besteht aus zwei Strukturelementen (master und detail). Im master-Bereich wird eine dynamisch erzeugte Liste angezeigt. Je nach ausgewähltem Eintrag in dieser Liste, werden die detaillierteren Informationen zu diesem Eintrag im detail-Strukturelement angezeigt.

Die detaillierteren Unterschiede zwischen den Komponenten sind in den jeweiligen Kapiteln erläutert.

```
<WebApp xmlns="http://softproject.de/webapp/1.0" path="BeispielProjekt" displayName="
My New Web App">
  <Modules>
    <Module path="Module" displayName="My Module">
      <Components>
        <!-- Bereich für die Komponenten -->
        <ComponentReference/>
      </Components>
    </Module>
  </Modules>
</WebApp>
```

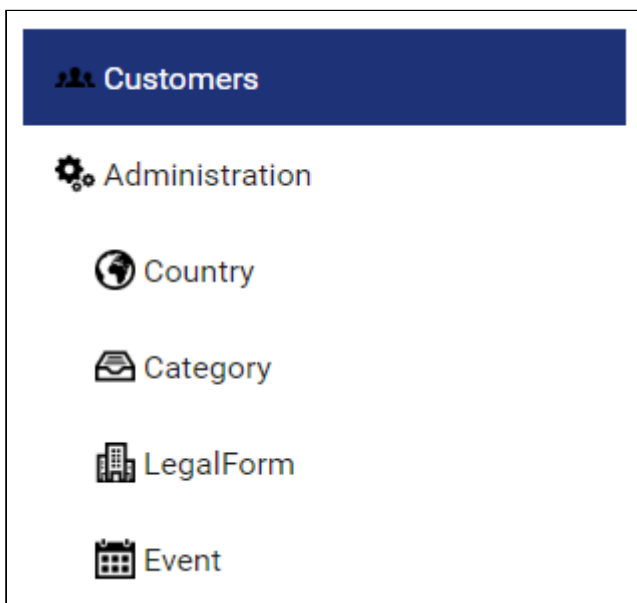
Komponenten können ineinander verschachtelt werden.

```

<WebApp xmlns="http://softproject.de/webapp/1.0" displayName="Advanced CRM">
  <Modules>
    <Module displayName="My Module" path="Module">
      <Components>
        <ComponentReference default="true" displayName="$Customers" iconUrl="
logo.png" name="Home"
          path="Home" source="Customers/Customers.masterdetail"/>
        <ComponentReference displayName="$Administration" iconUrl="Gear.png"
path="Administration"
          source="Administration/Administration.masterdetail">
          <Components>
            <ComponentReference displayName="$Country" iconUrl="Country.p
ng" name="Countries"
              path="Countries" source="Administration/Country/
Country.masterdetail"/>
            ...
            <ComponentReference displayName="$Event" iconUrl="Event.png"
name="Event"
              path="Event" source="Administration/Event/Event.masterdetail/
>
          </Components>
        </ComponentReference>
      </Components>
    </Module>
  </Modules>
</WebApp>

```

Wenn Komponenten verschachtelt werden, werden sie im Menü als Unterpunkt angezeigt:








- ⓘ Bei der Verschachtelung von Komponenten ist darauf zu achten, dass innerhalb der äußeren Komponente zunächst ein Element Components erzeugt wird. Components enthält die untergeordneten Komponenten.

## 1.6.1 Attribute für Komponenten

Komponenten können folgende Attribute besitzen:

Attribut	Beschreibung
fontFamily	<p>Legt die Schriftfamilie innerhalb der Komponente fest. Die Eigenschaft wird an alle Steuerelemente und Aktionen der Komponente vererbt.</p> <p><b>Mögliche Werte:</b> Fontcode aus der Font-Palette, z. B. <i>Font04</i></p>
fontSize	<p>Legt die Schriftgröße innerhalb der Komponente fest. Die Eigenschaft wird an alle Steuerelemente und Aktionen der Komponente vererbt.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <i>20</i>; <i>20.8</i>; <i>.9</i></li> <li>• Schriftgröße in Pixel, z. B. <i>20px</i></li> <li>• Schriftgröße in Punkt, z. B. <i>18pt</i></li> <li>• Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i></li> <li>• Schlüsselwörter: <i>xx-Small</i>, <i>x-small</i>, <i>small</i>, <i>medium</i>, <i>large</i>, <i>x-large</i>, <i>xx-large</i>, <i>smaller</i>, <i>larger</i></li> </ul>
fontStretch	<p>Legt die Breite der einzelnen Schriftzeichen fest. Die Eigenschaft wird an alle Steuerelemente und Aktionen der Komponente vererbt.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Condensed</li> <li>• Expanded</li> <li>• ExtraCondensed</li> <li>• ExtraExpanded</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• SemiCondensed</li> <li>• SemiExpanded</li> <li>• UltraCondensed</li> <li>• UltraExpanded</li> </ul>
fontStyle	<p>Legt die Neigung der Schrift fest. Die Eigenschaft wird an alle Steuerelemente und Aktionen der Komponente vererbt.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <i>italic</i>: kursive Schrift</li> <li>• <i>normal</i>: normale Schrift (Standard)</li> <li>• <i>oblique</i>: schräg gestellter Schriftstil (berechnet)</li> </ul>

Attribut	Beschreibung
fontWeight	<p>Legt die Schriftstärke fest. Die Eigenschaft wird an alle Steuerelemente und Aktionen der Komponente vererbt.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Black</li> <li>• Bold</li> <li>• DemiBold</li> <li>• ExtraBlack</li> <li>• ExtraBold</li> <li>• ExtraLight</li> <li>• Heavy</li> <li>• Light</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• Regular</li> <li>• SemiBold</li> <li>• Thin</li> <li>• UltraBlack</li> <li>• UltraBold</li> <li>• UltraLight</li> </ul>
iconColor	<p>Definiert die Farbe des Icons der Komponente.</p> <div data-bbox="544 1128 1434 1240">  Diese Einstellung überschreibt die Standardfarbe des Farbschemas!         </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Hexadezimaler Farbwert, z. B. <i>ff5a00</i></li> </ul> <div data-bbox="588 1375 1434 1487">  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts!         </div> <ul style="list-style-type: none"> <li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. <i>A200</i></li> </ul>

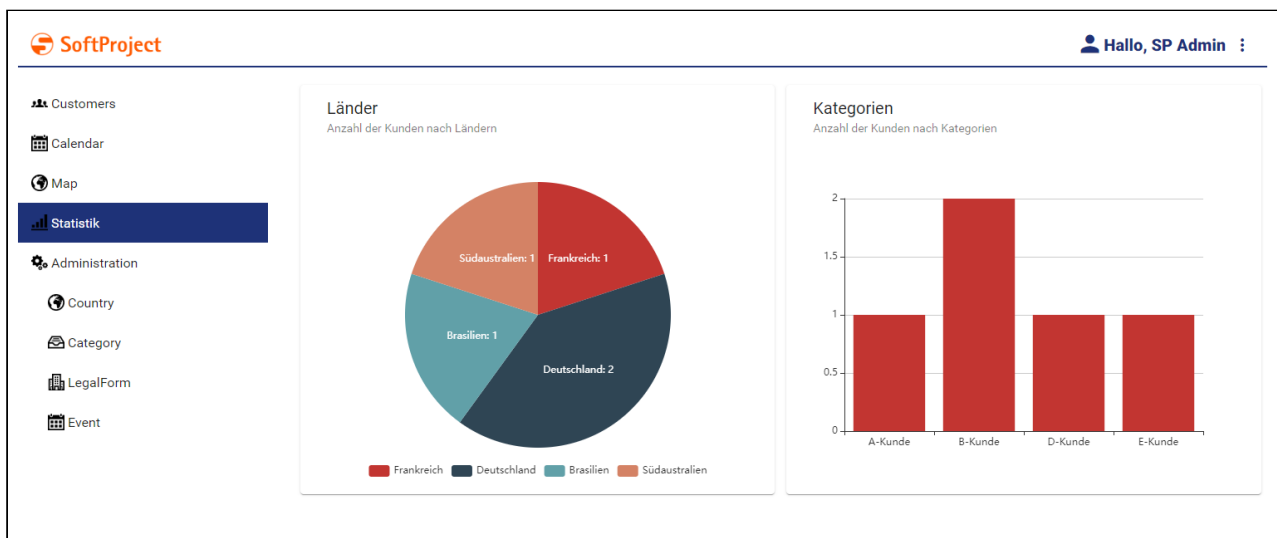
Attribut	Beschreibung
iconURL	<p>Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Icon der Komponente verwendet wird.</p> <div>  <ul style="list-style-type: none"> <li>Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources.</li> <li>Das Material Icon muss mit dem Präfix icon, z. B. icon:&lt;MaterialIconName&gt; angegeben werden.</li> </ul> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Zeichenkette (URI), z. B. <i>clock.png</i></li> <li>icon:&lt;MaterialIconName&gt;, z. B. <i>icon:extension</i></li> </ul> <div>  <p>Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren <a href="#">Material Icons</a> abweichen.</p> </div>
process	<p><i>Erforderlich.</i> Pfad zur .wrf-Datei, die die Daten für die Komponente liefert. Der Prozess muss im Ordner Services/Processes enthalten sein. Pfadangabe relativ zum Ordner Services/Processes.</p> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p>
title	<p>Titel der Komponente. Wird in der Anwendung als Header angezeigt.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
titleBackground	<p>Definiert eine Titel-Hintergrundfarbe.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div>  <p>Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts!</p> </div> <ul style="list-style-type: none"> <li>Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>

Attribut	Beschreibung
titleForeground	<p>Definiert eine Farbe für den Titel-Vordergrund.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div style="border: 1px solid #f0e68c; padding: 5px; margin: 10px 0;"> <p><b>⚠ Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts!</b></p> </div> <ul style="list-style-type: none"> <li>Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>

**i** Komponenten müssen ein [Layout](#) enthalten.

## 1.6.2 Detail-Komponente

Detail-Komponenten zeigen Inhalt seitenbasiert an. Dabei kann es sich um reine Texte, aber z. B. auch um Checkboxes oder statische oder dynamische Listen handeln.



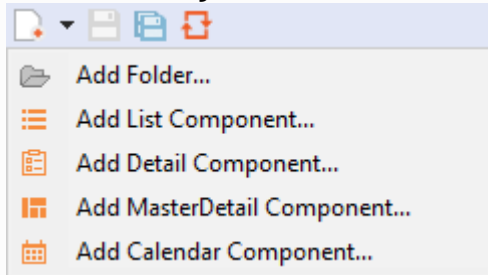
### 1.6.2.1 Detail-Komponente definieren

Eine Detail-Komponente wird innerhalb einer eigenen Definitionsdatei

`<DetailComponentName>.detail` deklariert. Diese wird im Projektordner **Components** angelegt.

**So legen Sie eine neue Definitionsdatei an:**

1. Klicken Sie auf das Menü **New > Add <entsprechende Komponente>**.  
Alternativ lässt sich die Datei auch über das Kontextmenü **New > Add <entsprechende Komponente>** anlegen.
2. Wählen Sie die gewünschte Definitionsdatei.



3. In **File name** den Namen der Definitionsdatei eingeben.
4. Auf **Finish** klicken.  
Eine neue Definitionsdatei mit einer vordefinierten Struktur wird angelegt.

Die Detail-Komponente wird über `<DetailComponent>` erzeugt und kann die Standard-Attribute für **Komponenten** besitzen.

Für eine Detail-Komponente muss ein **Layout** deklariert werden.

### 1.6.2.2 Beispiel Detail-Komponente

```
<DetailComponent process="/WebApp/Map/LoadMapData.wrf" xmlns="http://softproject.de/webapp/1.0">
  <Properties>
    <Property name="Place" type="List">
      <Property name="Company" type="String" />
      <Property name="Address" type="String" />
    </Property>
  </Properties>
  <FlowLayout>
    <Map height="100" vendor="OpenStreetMap">
      <Search />
      <Markers data="#Place" iconUrl="Location.png" name="#Company"
searchString="#Address" />
    </Map>
  </FlowLayout>
</DetailComponent>
```





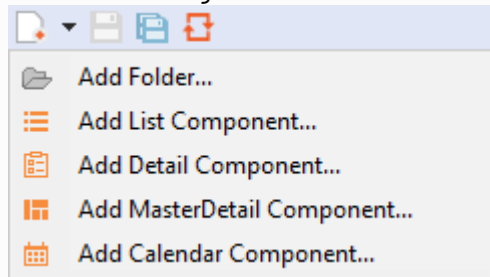
Eine List-Komponente zeigt dynamisch generierte Listen an. Dazu benötigt die Komponente Daten und Informationen darüber, wie diese Daten angezeigt werden sollen.



Eine List-Komponente wird innerhalb einer eigenen Definitionsdatei `<ListName>.list` deklariert. Diese wird im Projektordner **Components** angelegt.

**So legen Sie eine neue Definitionsdatei an:**


1. Klicken Sie auf das Menü **New > Add &ltentsprechende Komponente>**.  
Alternativ lässt sich die Datei auch über das Kontextmenü **New > Add &ltentsprechende Komponente>** anlegen.
2. Wählen Sie die gewünschte Definitionsdatei.




3. In **File name** den Namen der Definitionsdatei eingeben.
4. Auf **Finish** klicken.  
Eine neue Definitionsdatei mit einer vordefinierten Struktur wird angelegt.

Die List-Komponente wird über `<ListComponent>` erzeugt und kann neben den **Standard-Attributen** für **Komponenten** folgende zusätzliche Attribute besitzen:

Attribut	Beschreibung
pageSize	<p>Definiert die Anzahl der Elemente, die auf einmal in die Liste geladen werden (vgl. <a href="#">Paging</a>).</p> <p><b>Mögliche Werte:</b> Ganzzahl größer als 0</p>
rowBackground	<p>Definiert die Hintergrundfarbe der Zeilen in der Liste.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Farbwert aus dem Theme (z. B. A50, A400 oder 09 vgl. <a href="#">Theming</a>)</li> <li>• Hexadezimaler Farbwert</li> </ul> <div style="border: 1px solid #f0e68c; padding: 10px; margin: 10px 0;"> <p><b>⚠</b> Wenn im Data-Binding-Ausdruck ein #-Zeichen verwendet werden soll, muss dieses maskiert werden. Als Maskierungszeichen dient der Backslash (\).</p> </div> <ul style="list-style-type: none"> <li>• Ausdruck für Data Binding (Color)</li> </ul>

Attribut	Beschreibung
rowForeground	<p>Definiert eine Farbe für den Vordergrund (Texte usw.).</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Farbwert aus dem Theme (z. B. A50, A400 oder 09 vgl. <a href="#">Theming</a>)</li> <li>Hexadezimaler Farbwert</li> </ul> <div style="border: 1px solid orange; padding: 10px; margin: 10px 0;"> <p> Wenn im Data-Binding-Ausdruck ein #-Zeichen verwendet werden soll, muss dieses maskiert werden. Als Maskierungszeichen dient der Backslash (\).</p> </div> <ul style="list-style-type: none"> <li>Ausdruck für Data Binding (Color)</li> </ul>

Die Daten, die angezeigt werden sollen, müssen durch einen Prozess zur Verfügung gestellt werden und einem bestimmten Datenmodell entsprechen. In den [Properties](#) muss das Datenmodell abgebildet werden, das durch den Prozess geliefert wird. Für die Verknüpfung der Daten, die durch den Prozess geliefert werden, mit der Darstellung innerhalb der Webanwendung, muss die entsprechende Property definiert und per [Data Binding](#) in der Column-Definition hinterlegt werden.


 Wenn eine Property in der List-Komponente verwendet werden soll, muss sie als *Complex-Type* definiert werden. Diese Property kann weitere Properties enthalten.

### 1.6.3.2 Column-Definition




Innerhalb der List-Komponente müssen die Spalten der Liste definiert werden. Die Spaltendefinition erfolgt über `Column` bzw. `ImageColumn` für Spalten, die Bilder enthalten. Alle Spalten müssen innerhalb von `Columns` definiert werden.


```
<ListComponent process="list.wrf">
  <Properties>
    ...
  </Properties>
  <Columns>
    <Column .../>
    <Column .../>
    <ImageColumn .../>
    <!-- Bereich für ein oder mehrere (Image)Column-Elemente -->
  </Columns>
</ListComponent>
```

**Mögliche Attribute für <Column>:**

Attribut	Beschreibung
horizontalAlign	<p>Legt die horizontale Ausrichtung des Inhalts innerhalb der Spalte fest.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• left: Inhalt linksbündig</li> <li>• center: Inhalt zentriert</li> <li>• right: Inhalt rechtsbündig</li> </ul>
mobilePosition	<p>Legt die Position fest, an der die Spalte in der mobilen Ansicht erscheint. In der mobilen Ansicht können maximal vier Werte angezeigt werden.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• topLeft</li> <li>• topRight</li> <li>• bottomLeft</li> <li>• bottomRight</li> </ul> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p><b>i</b> Wenn dieses Attribut nicht definiert wird, wird die Spalte in der mobilen Ansicht nicht angezeigt.</p> </div> 
sortable	<p>Legt fest, ob nach den Inhalten der Spalte sortiert werden kann. Weitere Informationen zu <a href="#">Sortieren</a>.</p> <p><b>Mögliche Werte:</b> true / false</p>
value	<p>Legt den Inhalt der Spalte per Data Binding fest.</p> <p><b>Mögliche Werte:</b> Zeichenkette (Data Binding)</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p><b>i</b> Wenn für die entsprechende Property kein Attribut displayName vergeben ist, dann wird als Spaltenüberschrift der Data-Binding-Ausdruck verwendet.</p> </div>


**Mögliche Attribute für <ImageColumn>:**

Attribut	Beschreibung
horizontalAlign	<p>Horizontale Ausrichtung der Grafik in der Spalte</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• left: Grafik linksbündig</li> <li>• center: Grafik zentriert</li> <li>• right: Grafik rechtsbündig</li> </ul>
iconColor	<p>Definiert die Farbe des Bildes.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Hexadezimaler Farbwert, z. B. <i>ff5a00</i></li> </ul> <div>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts!         </div> <ul style="list-style-type: none"> <li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. <i>A200</i></li> </ul>
image	<p>Grafik, die in der Spalte angezeigt wird.</p> <p><b>Mögliche Werte:</b> Ausdruck für Data Binding (Image)</p>
imageHeight	<p>Höhe der Grafik</p> <p><b>Mögliche Werte:</b> Ganzzahl</p> <div>  Die Einheit, in der die Höhe angegeben wird, wird mit dem Attribut <code>imageUnits</code> festgelegt.         </div>
imageUnits	<p>Einheit für die Breiten- und Höhenangabe zur Grafik.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• pixels: Pixel</li> <li>• percents: Prozent (Standard)</li> </ul>
imageWidth	<p>Breite der Grafik</p> <p><b>Mögliche Werte:</b> Ganzzahl</p> <div>  Die Einheit, in der die Breite angegeben wird, wird mit dem Attribut <code>imageUnits</code> festgelegt.         </div>


Attribut	Beschreibung
mobilePosition	<p>Legt die Position fest, an der die Spalte in der mobilen Ansicht erscheint. In der mobilen Ansicht können maximal vier Werte angezeigt werden.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• topLeft</li> <li>• topRight</li> <li>• bottomLeft</li> <li>• bottomRight</li> </ul> <div>  Wenn dieses Attribut nicht definiert wird, wird die Spalte in der mobilen Ansicht nicht angezeigt. </div>

### 1.6.3.3 Periodisches Neuladen

Über das Element `ReloadTimer` lässt sich die List-Komponente periodisch neu laden.

 Das periodische Neuladen von Daten kann eine hohe Last verursachen und sollte nur in seltenen Anwendungsfällen verwendet werden.

**Mögliche Attribute für `<ReloadTimer>`:**

Attribut	Beschreibung
refreshTime	<p>Zeit bis zum erneuten Laden in Sekunden</p> <ul style="list-style-type: none"> <li>• Data Binding (Integer) möglich</li> </ul> <p><b>Mögliche Werte:</b> Beliebige ganze Zahl</p>
itemIdentifizier	<p><i>Optional</i>; Eindeutiger Schlüssel für die in den Zeilen dargestellten Daten</p> <div>  <ul style="list-style-type: none"> <li>• Über das Attribut lassen sich Zeilen beim Neuladen wiederfinden und der Neuladeprozess wird verbessert.</li> <li>• Wird das Attribut nicht angegeben wird ein Stringvergleich auf ein Datensatz verwendet, um die Zeile zu identifizieren.</li> </ul> </div> <p><b>Mögliche Werte:</b> Data Binding an eine definierte Property der List-Komponente</p>

### 1.6.3.4 Mobile Ansicht der List-Komponente

Listen werden in der mobilen Ansicht komprimiert mit maximal vier Spalten angezeigt. Die vier Spalten werden in den vier Ecken eines Rechtecks angeordnet, die Anordnung kann mit dem Attribut `mobilePosition` festgelegt werden. Weitere Spalten werden in der mobilen Ansicht nicht angezeigt.

× Company	× ContactPerson	× Category	× Country
Energy Station	Jens Probst	A-Kunde	Deutschland
Patrol Work	Stefan Grunwald	B-Kunde	Frankreich
Brein Innovation	Laura Coley	B-Kunde	Südastralien
OffGrid Promotion	Luiza Silva Castro	E-Kunde	Brasilien
Netelectro	Thor Schuss	D-Kunde	Deutschland

Liste in der Desktop-Ansicht

Energy Station	Jens Probst	>
A-Kunde	Deutschland	>
Patrol Work	Stefan Grunwald	>
B-Kunde	Frankreich	>
Brein Innovation	Laura Coley	>
B-Kunde	Südastralien	>
OffGrid Promotion	Luiza Silva Castro	>
E-Kunde	Brasilien	>
Netelectro	Thor Schuss	>
D-Kunde	Deutschland	>

Liste in der mobilen Ansicht

#### 1.6.3.5 Aktion "Auswählen" innerhalb der List-Komponente

Die Aktion **Auswählen** wird über das Element `<SelectAction/>` innerhalb von `<Actions>` erzeugt und ausgelöst, sobald ein Eintrag aus einer List-Komponente ausgewählt wurde.

Zusätzlich zu den Standardattributen der Aktion "Auswählen" lässt sich folgendes Attribut definieren:

Attribut	Beschreibung
<code>disableAutoselect</code>	<p><i>Nur bei der Verwendung innerhalb der Master/Detail-Komponente.</i> Legt fest, ob der erste Eintrag einer Liste beim Öffnen der Komponente bereits ausgewählt ist.</p> <p><b>Mögliche Werte:</b> <code>true</code> / <code>false</code></p>

#### 1.6.3.6 Beispiel List-Komponente

Das folgende Beispiel zeigt die Deklaration einer List-Komponente.

```
<ListComponent process="list.wrf">
  <Properties>
    <Property name="ListObject" type="Complex">
      <Property name="Caption" type="String" />
    </Property>
  </Properties>
  <Columns>
    <Column value="#ListObject.Caption" />
  </Columns>
</ListComponent>
```

Die Komponente verarbeitet Daten, die in folgendem Format vorliegen.

```
<?xml version="1.0" encoding="UTF-8" ?>
<OkList>
  <ListObject>
    <Caption>Example 1</Caption>
  </ListObject>
  <ListObject>
    <Caption>Example 1</Caption>
  </ListObject>
  <ListObject>
    <Caption>Example 1</Caption>
  </ListObject>
</OkList>
```

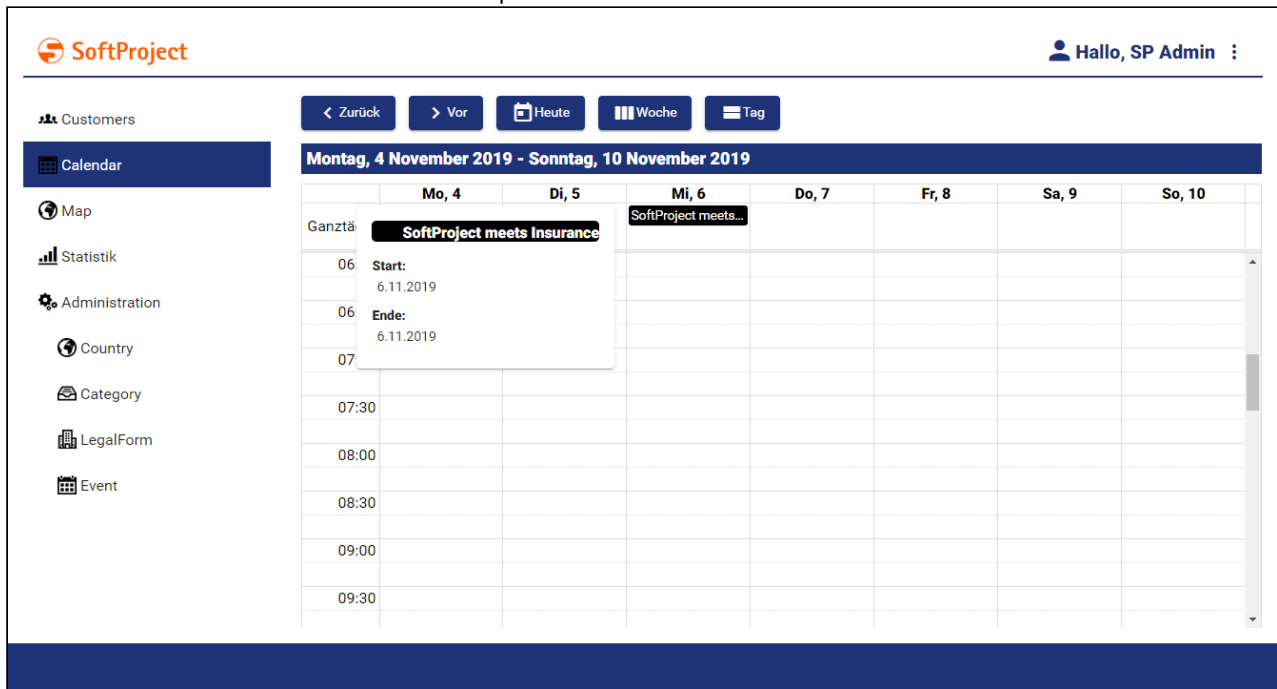
Obiger Code führt zu folgender Ausgabe:

Halo, SP Admin :	
List Component	#ListObject.Caption
	Example 1
	Example 1
	Example 1



### 1.6.4 Kalender-Komponente

Eine Kalender-Komponente zeigen einen Kalender und Ereignisse. Eine Kalender-Komponente kann alleine oder in einer Master/Detail-Komponente verwendet werden.

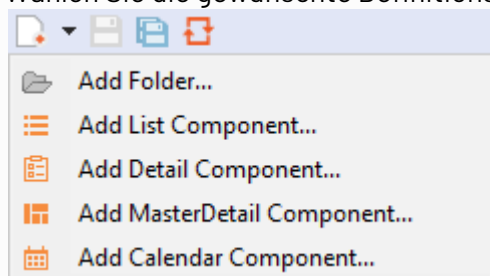


#### 1.6.4.1 Kalender-Komponente definieren

Eine Kalender-Komponente wird innerhalb einer eigenen Definitionsdatei <CalendarName>.calendar deklariert. Diese wird im Projektordner **Components** angelegt.

**i So legen Sie eine neue Definitionsdatei an:**

1. Klicken Sie auf das Menü **New > Add <entsprechende Komponente>**. Alternativ lässt sich die Datei auch über das Kontextmenü **New > Add <entsprechende Komponente>** anlegen.
2. Wählen Sie die gewünschte Definitionsdatei.



3. In **File name** den Namen der Definitionsdatei eingeben.
4. Auf **Finish** klicken.  
Eine neue Definitionsdatei mit einer vordefinierten Struktur wird angelegt.


Die Kalender-Komponente wird über `<CalendarComponent>` erzeugt und kann die Standard-Attribute für Komponenten besitzen.

#### 1.6.4.2 Kalendereinstellungen: Ansichten


Über die Kalendereinstellungen kann festgelegt werden, welche Ansichten des Kalenders durch den Nutzer angezeigt werden können und welche Ansicht die Standardansicht ist. Die Kalendereinstellungen über Attribute des Elements `<CalendarSettings>` vorgenommen.

```
<CalendarComponent ...>
  ...
  <CalendarSettings defaultView="Day" ...>
</CalendarComponent>
```

##### Mögliche Attribute:

Attribut	Beschreibung
dayView	Aktiviert und deaktiviert die Tagesansicht. <b>Mögliche Werte:</b> true (Standard) / false
dayViewSlot	Definiert die Zeitskala in der Tagesansicht. <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• 00:15: 15 Minuten als kleinste Einheit</li> <li>• 00:30: 30 Minuten als kleinste Einheit</li> <li>• 01:00: 1 Stunde als kleinste Einheit</li> </ul>
defaultView	Definiert die Standardansicht des Kalenders. <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• Month: Monatsansicht (Standard)</li> <li>• Week: Wochenansicht</li> <li>• Day: Tagesansicht</li> </ul> <div>  Es kann nur eine aktive Ansicht als Standardansicht definiert werden.         </div>
monthView	Aktiviert und deaktiviert die Monatsansicht. <b>Mögliche Werte:</b> true (Standard) / false
weekView	Aktiviert und deaktiviert die Wochenansicht. <b>Mögliche Werte:</b> true (Standard) / false

Attribut	Beschreibung
weekViewSet	Definiert die Zeitskala in der Wochenansicht.  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• 00:15: 15 Minuten als kleinste Einheit</li> <li>• 00:30: 30 Minuten als kleinste Einheit</li> <li>• 01:00: 1 Stunde als kleinste Einheit</li> </ul>

 Die Kalenderansichten (monthView, weekView, dayView) sind mit Buttons verknüpft, über die der Benutzer die jeweilige Ansicht auswählen kann. Wenn eine Kalenderansicht deaktiviert ist, dann wird auch der entsprechende Button ausgeblendet.



#### 1.6.4.3 Kalendereinstellungen: Lokalisierung

Eine Kalender-Komponente kann lokalisiert werden. Standardmäßig wird eine deutsche Benutzeroberfläche angeboten. Mithilfe des Elements `<Localization>` innerhalb von `<CalendarSettings>` können beliebige Übersetzungen oder benutzerdefinierte Schlüssel für Übersetzungsdateien definiert werden.

```
<CalendarComponent ...>
  ...
  <CalendarSettings ...>
    <Localization march="March" ... />
  </CalendarSettings>
</CalendarComponent>
```

Wenn für die entsprechende Web App eine Übersetzungsdatei angelegt wird, sind alle Übersetzungsschlüssel verfügbar.

#### Vorhandene Übersetzungen:

Schlüssel	Wert
next	Vor
prev	Zurück
today	Heute
monthView	Monat
weekView	Woche
dayView	Tag
monday	Montag
tuesday	Dienstag
wednesday	Mittwoch
thursday	Donnerstag
friday	Freitag

Schlüssel	Wert
saturday	Samstag
sunday	Sonntag
mondayShort	Mo
tuesdayShort	Di
wednesdayShort	Mi
thursdayShort	Do
fridayShort	Fr
saturdayShort	Sa
sundayShort	So
january	Januar
february	Februar
march	März
april	April
may	Mai
june	Juni
july	Juli
august	August
september	September
october	Oktober
november	November
december	Dezember
start	Start
end	Ende
location	Ort
description	Beschreibung
allDay	ganztägig

#### 1.6.4.4 Events

Events werden mit dem Element `<EventConfiguration/>` definiert. Das Element wird verwendet, um Werte aus einer Liste an Events im Kalender zu binden. Die Liste, die die Events enthält, wird von einem Prozess an die Kalender-Komponente geliefert.


```

<CalendarComponent ...>
  ...
  <CalendarSettings ...>
    <Localization ... />
  </CalendarSetting>
  <EventConfiguration />
</CalendarComponent>

```

**Mögliche Attribute:**


Attribut	Beschreibung
allDay	<p>Definiert, ob das Event ganztägig stattfindet.</p> <p><b>Mögliche Werte:</b> true / false oder Ausdruck für Data Binding (Boolean)</p>
background	<p>Definiert die Farbe, mit der das Event angezeigt werden soll.</p> <p><b>Mögliche Werte:</b> HTML-Farbcode, z. B. #FF0000 oder Ausdruck für Data Binding (Color)</p>
begin	<p>Definiert den Startzeitpunkt des Events.</p> <p><b>Mögliche Werte:</b> Ausdruck für Data Binding (DateTime)</p>
description	<p>Gibt die Beschreibung des Events an.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette oder Ausdruck für Data Binding (String)</p>
detailNavigationLinkName	<p>Name der Schaltfläche, die die Detailansicht öffnet, falls das Attribut externalNavigationLink nicht gesetzt ist.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Beliebige Zeichenkette</li> <li>• leer: Der Standardtext <b>Detail view</b> wird angezeigt</li> </ul> <div> <p><b>i</b> Befindet sich der Kalender innerhalb der Master-Komponente einer Master-/Detail-Komponente, kann auf mobilen Geräten nicht auf die Detailansicht zugegriffen werden, da diese durch einen externen Navigationslink geöffnet wird. Hat das Event keinen externen Navigationslink, lässt sich über eine Schaltfläche zur Detailansicht navigieren. Mit dem Attribut detailNavigationLinkName wird der Name der Schaltfläche angegeben.</p> <p>Die Schaltfläche wird nur in der mobilen Ansicht angezeigt, wenn sich die Kalender-Komponente innerhalb einer Master-/Detail-Komponente befindet.</p> </div>
end	<p>Definiert den Endzeitpunkt des Events.</p> <p><b>Mögliche Werte:</b> Ausdruck für Data Binding (DateTime)</p>
externalNavigationLink	<p>Link, der geöffnet werden soll, wenn das Event angeklickt wird.</p> <p><b>Mögliche Werte:</b> Hyperlink (String) oder Ausdruck für Data Binding (String)</p>

Attribut	Beschreibung
externalNavigationLinkName	<p>Text, der bei einem externen Link statt der URL angezeigt wird.</p> <p><b>Mögliche Werte:</b> Zeichenkette</p> <div> <p> Wenn bei einem externen Link das Attribut externalNavigationLinkName nicht definiert ist, dann wird die URL angezeigt, die für das Attribut externalNavigationLink definiert wurde.</p> </div>
externalNavigationLinkTarget	<p>Gibt an, ob der Link (externalNavigationLink) im selben oder in einem neuen Tab geöffnet wird.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• same: Der Link wird im selben Tab geöffnet</li> <li>• new: Der Link wird in einem neuen Tab geöffnet</li> </ul>
fontFamily	<p>Legt die Schriftfamilie fest.</p> <p><b>Mögliche Werte:</b> Fontcode aus der Font-Palette, z. B. <i>Font04</i></p>
fontSize	<p>Legt die Schriftgröße fest.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Schriftgröße in Pixel, z. B. <i>20px</i></li> <li>• Schriftgröße in Punkt, z. B. <i>18pt</i></li> <li>• Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i></li> <li>• Schlüsselwörter, z. B. <i>small</i> oder <i>larger</i></li> </ul>
fontStretch	<p>Legt die Breite der einzelnen Schriftzeichen fest.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Condensed</li> <li>• Expanded</li> <li>• ExtraCondensed</li> <li>• ExtraExpanded</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• SemiCondensed</li> <li>• SemiExpanded</li> <li>• UltraCondensed</li> <li>• UltraExpanded</li> </ul>
fontStyle	<p>Legt die Neigung der Schrift fest.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• italic: kursive Schrift</li> <li>• normal: normale Schrift (Standard)</li> <li>• oblique: schräg gestellter Schriftstil (berechnet)</li> </ul>

Attribut	Beschreibung
fontWeight	<p>Legt die Schriftstärke fest.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Black</li> <li>• Bold</li> <li>• DemiBold</li> <li>• ExtraBlack</li> <li>• ExtraBold</li> <li>• ExtraLight</li> <li>• Heavy</li> <li>• Light</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• Regular</li> <li>• SemiBold</li> <li>• Thin</li> <li>• UltraBlack</li> <li>• UltraBold</li> <li>• UltraLight</li> </ul>
foreground	<p>Definiert eine Farbe für den Event-Namen.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>
location	<p>Gibt den Ort des Events an.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette oder Ausdruck für Data Binding (String)</p>
name	<p>Definiert den Namen des Events.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette oder Ausdruck für Data Binding (String)</p>

#### 1.6.4.5 Externe Links in Events

Innerhalb eines Events lassen sich externe Links einfügen. Diese werden innerhalb des Elementes `<EventConfiguration/>` mithilfe einer `<SelectAction>` umgesetzt.

 Das Element `<SelectAction/>` kann dabei die Standardattribute für Aktionen sowie die für die Aktion "Auswählen" spezifischen Attribute haben.

```
<EventConfiguration>
  <SelectAction externalLink="#event.link" />
</EventConfiguration>
```

#### 1.6.4.6 Daten für die Kalender-Komponente

Die Kalender-Komponente benötigt Daten in einer Liste als Input, um Events anzeigen zu können. Diese Daten werden von einem Prozess geliefert und müssen folgendem Schema entsprechen.

```
<CalendarComponent displayName="Calendar" path="calendar" process="events.wrf">
  <Properties>
    <Property name="List" type="Complex">
      <Property name="Name" type="String" />
      <Property name="AllDay" type="Boolean" />
      <Property name="Begin" type="DateTime" />
      <Property name="End" type="DateTime" />
      <Property name="Color" type="Color" />
      <Property name="Link" type="String" />
    </Property>
  </Properties>
  <EventConfiguration allDay="#List.AllDay" name="#List.Name" begin="#List.Begin"
end="#List.End" color="#List.Color" externalNavigationLink="#List.Link" />
</CalendarComponent>
```

❗ Die Datenstruktur muss in den **Properties** der Komponente angegeben werden!

#### 1.6.4.7 Beispiel

Die Definition einer Kalender-Komponente kann folgendermaßen aussehen.

##### Beispiel CalendarComponent

```
<CalendarComponent
xmlns="http://softproject.de/webapp/1.0" process="calendar.wrf">
  <Properties>
    <Property name="event" type="Complex">
      <Property name="id" type="String"/>
      <Property name="name" type="String"/>
      <Property name="allDay" type="Boolean"/>
      <Property name="begin" type="DateTime"/>
      <Property name="end" type="DateTime"/>
      <Property name="color" type="Color"/>
      <Property name="link" type="String"/>
    </Property>
  </Properties>
  <CalendarSettings defaultView="Week"/>
  <EventConfiguration allDay="#event.allDay" background="#event.color" begin="#even
t.begin" end="#event.end" name="#event.name">
    <SelectAction externalLink="#event.link"/>
  </EventConfiguration>
</CalendarComponent>
```

Der Input wird durch den Prozess `calendar.wrf` geliefert und sieht folgendermaßen aus:



**Beispiel Input Events.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<OkList>
  <event>
    <id>1</id>
    <name>Daily Scrum</name>
    <allDay>false</allDay>
    <begin>2020-03-13T09:30:00Z</begin>
    <end>2020-03-13T09:45:00Z</end>
    <color>0303FF</color>
    <link>https://www.scrum.org/resources/what-is-a-daily-scrum</link>
  </event>
  <event>
    <id>2</id>
    <name>Review</name>
    <allDay>false</allDay>
    <begin>2020-03-13T10:00:00Z</begin>
    <end>2020-03-13T11:00:00Z</end>
    <color>FF3300</color>
    <link></link>
  </event>
  <event>
    <id>3</id>
    <name>Lunch</name>
    <allDay>false</allDay>
    <begin>2020-03-13T12:00:00Z</begin>
    <end>2020-03-13T13:00:00Z</end>
    <color>FF9696</color>
    <link></link>
  </event>
</OkList>
```

Es wird das folgende Ergebnis ausgegeben:



## 1.6.5 Master/Detail-Komponente

Die Master/Detail-Komponente besteht aus den drei Strukturelementen **Master**, **Detail** und **Overlay**.

Element	Beschreibung
Master	Enthält die Komponente, die im oberen Bildschirmbereich angezeigt wird.
Detail	Enthält eine Komponente, die angezeigt wird, wenn eine bestimmte Bedingung erfüllt ist. Standardmäßig wird in diesem Bereich Inhalt angezeigt, wenn ein bestimmter Listeneintrag innerhalb des Master-Elements ausgewählt ist.
Overlay	Zeigt Pop-ups an. Komponenten, die entsprechend deklariert werden, werden im Pop-up auf dem Desktop und als separate Seite auf mobilen Geräten angezeigt.

Alle drei Strukturelemente können eine Referenz zu einer Detail- oder eine List-Komponente enthalten, siehe [Komponenten in der Web App-Definition referenzieren](#). Im Detail- und Overlay-Element kann die Master/Detail-Komponente mehrere referenzierte Komponenten enthalten. Diese Möglichkeit kann genutzt werden, indem Aktionen mit definiertem `componentName`-Attribut und deklarierte Komponenten mit definiertem `name`-Attribut verwendet werden.

The screenshot shows the SoftProject web application interface. On the left is a sidebar with navigation items: Customers, Calendar, Map, Statistik, Administration, Country, Category, LegalForm, and Event (selected). The main area is divided into two sections. The top section, labeled with a red '1', is the Master view, which is a table with columns 'Event', 'Begin', and 'End'. It lists several events, including 'BPA Brunch Bonn', 'BPA Brunch Berlin', 'BPA Brunch München', 'BPA Brunch Ettlingen', 'X4 DeveloperDay', 'SoftProject meets Insurance', 'CONCULECTRA Symposium Netzleittechnik', and 'Messekongress IT für Versicherungen'. The bottom section, labeled with a red '2', is the Detail view, which shows the details for the selected event 'BPA Brunch München'. It includes a checkbox for 'AllDay', a date and time picker for 'Begin' (26.3.2019 00:00), a date and time picker for 'End' (26.3.2019 00:00), a color picker for 'Color' (Schwarz), and a link field with the value 'https://softproject.de/de/softproject/termine/'.

- |   |                        |
|---|------------------------|
| 1 | Master-Strukturelement |
| 2 | Detail-Strukturelement |



Weitere Informationen zu [Komponenten](#), [Aktionen](#), [Properties](#) und [Data Binding](#) finden sich in den entsprechenden Abschnitten.

### 1.6.5.1 Properties in Master/Detail

Die Properties der Master/Detail-Komponente können sowohl in der Master/Detail-Komponente selbst, als auch in den untergeordneten Komponenten innerhalb der Strukturelemente definiert

werden. Aus diesen Properties-Definitionen wird ein gemeinsames Datenmodell für die gesamte Master/Detail-Komponente erzeugt. Das bedeutet auch, dass Konflikte zwischen Properties innerhalb einer Master/Detail-Komponente (z. B. gleicher Name aber unterschiedlicher Datentyp) nicht zulässig sind und zu einem Validierungsfehler führen.

Alle Komponenten innerhalb einer Master/Detail-Komponente arbeiten also mit denselben Daten (bei List-Komponenten bezogen auf den ausgewählten Eintrag). Für Overlays wird eine temporäre Kopie der Daten erstellt, die nach erfolgreicher Bearbeitung zurück synchronisiert wird.

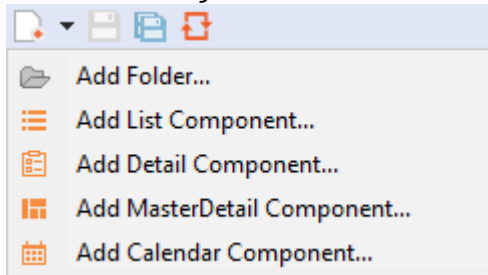
Über `ComponentReference` eingebundene Komponenten können nur auf Properties zugreifen, die auch innerhalb dieser Komponente definiert sind. Die Komponenten arbeiten dennoch auf demselben Datenobjekt, haben aber eine eingeschränkte Sicht auf die Properties, die sie tatsächlich benötigen. Insbesondere bei Prozessaufrufen wird dadurch unnötiger Datentransfer vermieden.

### 1.6.5.2 Master/Detail-Komponente definieren

Eine Master/Detail-Komponente wird innerhalb einer eigenen Definitionsdatei `<MasterDetailComponentName>.masterdetail` deklariert. Diese wird im Projektordner **Components** angelegt.

#### So legen Sie eine neue Definitionsdatei an:

1. Klicken Sie auf das Menü **New > Add <entsprechende Komponente>**. Alternativ lässt sich die Datei auch über das Kontextmenü **New > Add <entsprechende Komponente>** anlegen.
2. Wählen Sie die gewünschte Definitionsdatei.



3. In **File name** den Namen der Definitionsdatei eingeben.
4. Auf **Finish** klicken.  
Eine neue Definitionsdatei mit einer vordefinierten Struktur wird angelegt.

Die Master/Detail-Komponente wird über `<MasterDetailComponent>` erzeugt und kann die Standard-Attributen für **Komponenten** besitzen.

Das folgende Beispiel zeigt den grundsätzlichen Aufbau einer Master/Detail-Komponente.

```

<MasterDetailComponent xmlns="http://softproject.de/webapp/1.0">
  <Actions>
    <NewAction componentName="CustomerDetail" />
  </Actions>
  <Master>
    <ComponentReference source="Customer/Customer.list" name="AllCustomers" />
  </Master>
  <Detail>
    <TabGroup>
      <ComponentReference source="Customer/Tabs/Company.detail" displayName="$Company" name="CustomerDetail" />
      <ComponentReference source="Customer/Tabs/ContactPerson.detail" displayName="$ContactPerson" />
      <ComponentReference source="Customer/Tabs/Event.list" displayName="Events" name="CustomerEvents" />
    </TabGroup>
  </Detail>
  <Overlay>
    <ComponentReference source="Customer/Overlay/AddEvent.list" name="AddToEvent" />
  </Overlay>
</MasterDetailComponent>

```

Erläuterung:

- Innerhalb von Actions werden die verfügbaren **Aktionen** definiert.
- Das Master-Element enthält eine Referenz zu einer **List-Komponente**. Abhängig vom ausgewählten Listen-Eintrag, werden in der Detail-Sicht weitere Informationen angezeigt.
- Im Detail-Element wird definiert, welche Informationen für einen Listen-Eintrag angezeigt werden. In obigen Beispiel werden die Detailinformationen in einer **Tab-Gruppe** angezeigt.

### 1.6.5.3 Master-Strukturelement

Im Master-Strukturelement wird festgelegt, welche Informationen mithilfe welcher Komponente im oberen Bereich des Bildschirms angezeigt werden soll. Das Master-Strukturelement steuert, von welchem Datensatz zusätzliche Informationen im Detail-Bereich angezeigt werden.

Wenn der Benutzer noch keinen Listeneintrag im MasterStrukturelement ausgewählt hat, dann wird in der Desktop-Ansicht automatisch der erste Eintrag gewählt und die dazugehörigen Informationen im Detail-Strukturelement angezeigt.

Das Master-Strukturelement wird als erstes innerhalb der Master/Detail-Komponente definiert. Das Master-Strukturelement muss eine List-Komponente enthalten, die hier referenziert wird.

#### Beispiel für die Definition eines Master-Strukturelements

```

<MasterDetailComponent xmlns="http://softproject.de/webapp/1.0">
  <Actions/>
  <Master>
    <!-- Referenz zu einer List-Komponente-->
    <ComponentReference source="Customer/Customer.list" name="AllCustomers" />
  </Master>
  <Detail/>
  <Overlay/>
</MasterDetailComponent>

```

**i** Der Prozess, der die Daten der Master/Detail-Komponente verarbeitet, wird als Attribut `process` zur List-Komponente angegeben. Dabei muss der Prozess im Ordner `Services/Processes` enthalten sein. Die Pfadangabe erfolgt relativ zum Ordner `Services/Processes`.

#### 1.6.5.4 Detail-Strukturelement

Im Detail-Strukturelement werden weitere Informationen zu dem Datensatz angezeigt, der im Master-Strukturelement ausgewählt ist.

Das Detail-Strukturelement wird innerhalb von `MasterDetailComponent` nach dem Master-Strukturelement definiert. Die einzelnen Komponenten werden hier referenziert.

#### Beispiel für die Definition eines Detail-Strukturelements

```

<MasterDetailComponent xmlns="http://softproject.de/webapp/1.0">
  <Actions/>
  <Master/>
  <Detail>
    <!-- Referenz zu einer oder mehreren Detail- oder List-Komponenten -->
    <ComponentReference source="Administration/Country/Country.detail" name="CountryDetail" displayName="$General"/>
  </Detail>
  <Overlay/>
</MasterDetailComponent>

```

#### 1.6.5.5 Overlay-Strukturelement

Der Inhalt des Overlay-Strukturelements wird in der Web-Anwendung als Pop-up angezeigt und liegt über anderen Elementen. Das Overlay-Strukturelement enthält eine weitere Komponente, die sowohl statischen als auch dynamischen Inhalt anzeigen kann.

Das Overlay-Strukturelement wird innerhalb von MasterDetailComponent nach den Strukturelementen Master und Detail definiert.

Das Overlay-Strukturelement wird nur angezeigt, wenn es über eine Aktion aufgerufen wird.

```
<MasterDetailComponent xmlns="http://softproject.de/webapp/1.0">
  <Actions/>
  <Master/>
  <Detail/>
  <Overlay>
    <!-- Referenz zu einer Komponente in der das Overlay-Strukturelement definiert
ist-->
    <ComponentReference source="Customer/Overlay/AddEvent.list" name="AddToEvent"
/>
  </Overlay>
</MasterDetailComponent>
```

**i** Wenn ein Overlay angezeigt wird, dann wird der Teil der Web App, der noch im Hintergrund zu sehen ist, verschwommen dargestellt.

## 1.7 Data Binding

Data Binding oder Datenbindung ist ein Mechanismus, um Daten zwischen Objekten und Oberflächenelementen zu synchronisieren und so Daten anzuzeigen und das Verhalten von Steuerelementen zu beeinflussen. Normalerweise werden dabei Daten aus einem Datenobjekt (z. B. einer Kundenliste) an ein Steuerelement auf der Benutzeroberfläche (z. B. eine Tabelle)

weitergegeben und umgekehrt. Wenn dieser Mechanismus verwendet wird, muss nicht für jedes Objekt im Datenbestand eine Schleife verwendet werden, sondern das Steuerelement erkennt, wie viele Objekte vorhanden sind und passt sich automatisch an. Wenn im Steuerelement Änderungen durch den Benutzer vorgenommen werden, können diese in den Datenbestand zurückgespielt werden.

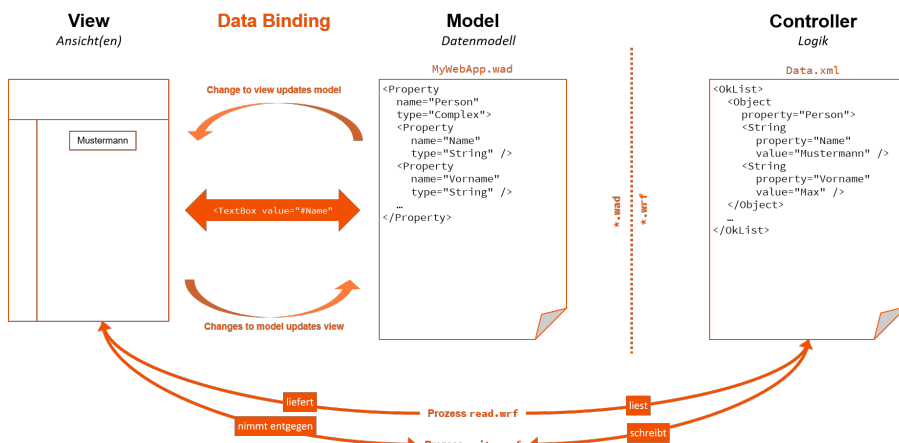
Für das Data Binding werden die name-Attribute der Elemente verwendet.

### Beispiel Data Binding:

```
<DetailComponent name="DetailExample" path="DetailExample" displayName="Detail
example">
  <Properties>
    <Property name="Example" type="String" />
    <Property name="Example2" type="Complex">
      <Property name="Internal" type="String" />
      <Property name="Visible" type="Boolean" />
    </Property>
  </Properties>
  <FlowLayout>
    <Label value="#Example" />
    <Checkbox checked="#Example2.Visible" displayName="Visible" />
    <Label value="#Example2.Internal" visible="#Example2.Visible" />
  </FlowLayout>
</DetailComponent>
```

Das obige Beispiel zeigt die Verwendung von Data Binding. Das value-Attribut des Label-Steuerelements erlaubt Data Binding, daher kann der Name der Eigenschaft mit einem führenden #-Zeichen verwendet werden. Um Zugang zu Subproperties von List- bzw. Complex-Properties zu gewährleisten, wird der Pfad-Mechanismus genutzt. Dabei dient ein Punkt (.) als Trennzeichen zwischen Pfadsegmenten.

Eine der Properties (Example2.Visible) wird verwendet, um zu definieren, ob die Checkbox ausgewählt ist und als Wert für das Attribut visible des zweiten Labels. Damit ist es möglich, die Sichtbarkeit des zweiten Labels mithilfe der Checkbox zu steuern: Wenn der Wert der Checkbox sich ändert (ausgewählt bzw. nicht ausgewählt), wird das durch den Data-Binding-Mechanismus an jeder Stelle aktualisiert.





❗ Wenn im Data-Binding-Ausdruck ein #-Zeichen verwendet werden soll, muss dieses maskiert werden. Als Maskierungszeichen dient der Backslash (\).

**Beispiel:** \#123456

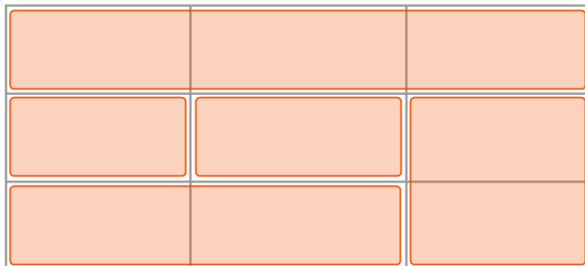
## 1.8 Layouts

Für die Komponenten innerhalb einer Web-Anwendung können verschiedene Layouts vorgegeben werden:

- Flow-Layout



- Grid-Layout



- Registerkarten-Layout (Tab Layout)



Das vorgegebene Layout steuert, wie die Steuerelemente (Text, Radio-Buttons, Checkboxes, Bilder etc.) angezeigt werden.




Die verschiedenen Layouts können ineinander verschachtelt werden.

Layouts können verschiedene **Steuerelemente** enthalten.





Das Layout wird innerhalb einer Komponente deklariert:



```
<DetailComponent>
  <FlowLayout>
    <!-- Inhalt des Layouts -->
  </FlowLayout>
</DetailComponent>
```

Folgende Attribute können für Flow- und Grid-Layouts (FlowLayout, GridLayout) sowie für Registerkarten (<Tab>) festgelegt werden.

Attribut	Beschreibung
background	<p>Definiert eine Hintergrundfarbe.</p> <div>  Diese Einstellung überschreibt die Standardfarbe des Farbschemas!         </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div>  Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts!         </div> <ul style="list-style-type: none"> <li>Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>
border	<p>Legt fest, ob um das Layout ein Rahmen angezeigt wird.</p> <p><b>Mögliche Werte:</b> <i>true</i> / <i>false</i></p>
enabled	<p>Aktiviert das Layout</p> <ul style="list-style-type: none"> <li>Data Binding (Boolean) möglich</li> </ul> <p><b>Mögliche Werte:</b> <i>true</i> / <i>false</i> oder Ausdruck für Data Binding</p>
fontFamily	<p>Legt die Schriftfamilie fest.</p> <div>  <ul style="list-style-type: none"> <li>Mit diesem Attribut wird die Standardschrift der Web App für dieses Steuerelement überschrieben.</li> <li>Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut <code>fontFamily</code> nicht explizit gesetzt haben.</li> <li>Dieses Attribut funktioniert nicht für die Elemente Image und Map.</li> </ul> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>MainFont: hinterlegte Hauptschrift</li> <li>Fontcode aus der Font-Palette, z. B. <i>Font04</i></li> </ul>

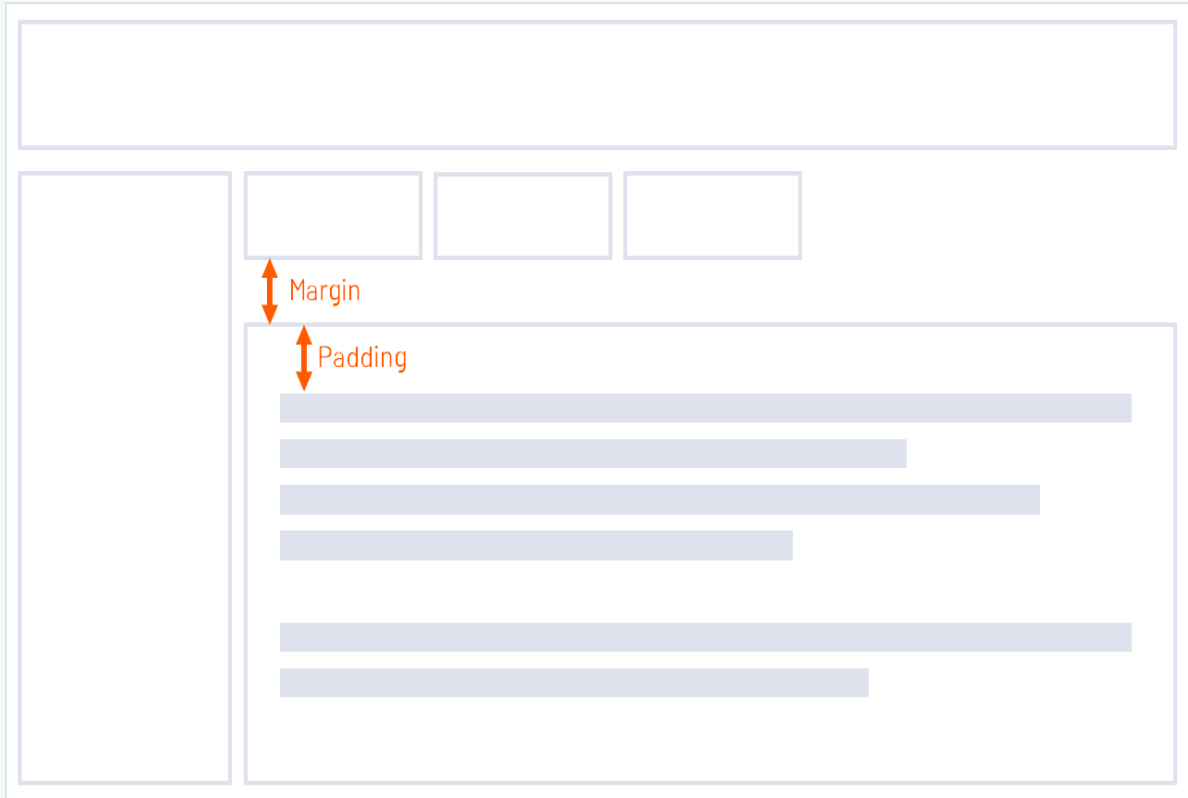
fontSize	<p>Legt die Schriftgröße fest.</p> <div data-bbox="481 340 1433 542"> <p><b>i</b></p> <ul style="list-style-type: none"> <li>• Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Steuerelement überschrieben.</li> <li>• Dieses Attribut funktioniert nicht für die Elemente Image und Map.</li> </ul> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <i>20</i>; <i>20.8</i>; <i>.9</i></li> <li>• Schriftgröße in Pixel, z. B. <i>20px</i></li> <li>• Schriftgröße in Punkt, z. B. <i>18pt</i></li> <li>• Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i></li> <li>• Schlüsselwörter: <i>xx-Small</i>, <i>x-small</i>, <i>small</i>, <i>medium</i>, <i>large</i>, <i>x-large</i>, <i>xx-large</i>, <i>smaller</i>, <i>larger</i></li> </ul>
fontStretch	<p>Legt die Breite der einzelnen Schriftzeichen fest.</p> <div data-bbox="481 987 1433 1099"> <p><b>i</b></p> <p>Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Steuerelement überschrieben.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Condensed</li> <li>• Expanded</li> <li>• ExtraCondensed</li> <li>• ExtraExpanded</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• SemiCondensed</li> <li>• SemiExpanded</li> <li>• UltraCondensed</li> <li>• UltraExpanded</li> </ul> <div data-bbox="481 1574 1433 1702"> <p><b>i</b></p> <p>Dieses Attribut funktioniert nicht für die Elemente Image, Map und Chart.</p> </div>

fontStyle	<p>Legt die Neigung der Schrift fest.</p> <div><p> Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Steuerelement überschrieben.</p></div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"><li>• <code>italic</code>: kursive Schrift</li><li>• <code>normal</code>: normale Schrift (Standard)</li><li>• <code>oblique</code>: schräg gestellter Schriftstil (berechnet)</li></ul> <div><p> Dieses Attribut funktioniert nicht für die Elemente Image und Map.</p></div>
fontWeight	<p>Legt die Schriftstärke fest.</p> <div><p> Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Steuerelement überschrieben.</p></div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"><li>• <code>Black</code></li><li>• <code>Bold</code></li><li>• <code>DemiBold</code></li><li>• <code>ExtraBlack</code></li><li>• <code>ExtraBold</code></li><li>• <code>ExtraLight</code></li><li>• <code>Heavy</code></li><li>• <code>Light</code></li><li>• <code>Medium</code></li><li>• <code>Normal</code> (Standard)</li><li>• <code>Regular</code></li><li>• <code>SemiBold</code></li><li>• <code>Thin</code></li><li>• <code>UltraBlack</code></li><li>• <code>UltraBold</code></li><li>• <code>UltraLight</code></li></ul> <div><p> Dieses Attribut funktioniert nicht für die Elemente Image und Map.</p></div>

foreground	<p>Definiert die Schriftfarbe.</p> <div>  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div>  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! </div> <ul style="list-style-type: none"> <li>Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>
height	<p>Höhe des Layouts</p> <p><b>Mögliche Werte:</b> Ganze Zahlen</p>
horizontalAlign	<p>Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li><i>left</i> (Standard)</li> <li><i>center</i></li> <li><i>right</i></li> </ul>
marginBottom	<p>Unterer Außenabstand</p> <p><b>Mögliche Werte:</b> Beliebige Zahlenwerte</p>
marginLeft	<p>Linker Außenabstand</p> <p><b>Mögliche Werte:</b> Beliebige Zahlenwerte</p>
marginRight	<p>Rechter Außenabstand</p> <p><b>Mögliche Werte:</b> Beliebige Zahlenwerte</p>
marginTop	<p>Oberer Außenabstand</p> <p><b>Mögliche Werte:</b> Beliebige Zahlenwerte</p>
marginUnits	<p>Einheiten für den Außenabstand</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>pixels</li> <li>percents</li> </ul>

paddingBottom	<p>Unterer Innenabstand</p> <p><b>Mögliche Werte:</b> Beliebige Zahlenwerte</p>
paddingLeft	<p>Linker Innenabstand</p> <p><b>Mögliche Werte:</b> Beliebige Zahlenwerte</p>
paddingRight	<p>Rechter Innenabstand</p> <p><b>Mögliche Werte:</b> Beliebige Zahlenwerte</p>
paddingTop	<p>Oberer Innenabstand</p> <p><b>Mögliche Werte:</b> Beliebige Zahlenwerte</p>
paddingUnits	<p>Einheiten für den Innenabstand</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• pixels</li> <li>• percents</li> </ul>
textOverflow	<p>Gibt an, was passieren soll, wenn die Seite voll ist.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <i>ellipsis</i> (Standard): Mit ... zeigen, dass der Text nicht zu Ende ist</li> <li>• <i>hidden</i>: Text abbrechen, auf ganze Wörter achten</li> <li>• <i>wordBreak</i>: innerhalb des Wortes umbrechen</li> <li>• <i>allow</i>: Text zwischen Wörtern umbrechen</li> </ul>
units	<p>Gibt die Einheit an, die für die Größenangaben height und width gilt.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• pixels (Standard)</li> <li>• percents</li> </ul>
visible	<p>Legt fest, ob das Layout sichtbar ist.</p> <ul style="list-style-type: none"> <li>• Data Binding (Boolean) möglich.</li> </ul> <p><b>Mögliche Werte:</b> Boolean oder Zeichenkette für Data Binding</p>
width	<p>Legt die Breite des Layouts fest.</p> <p><b>Mögliche Werte:</b> Ganze Zahlen</p>

- ✓ Unter Padding ist der Innenabstand zu verstehen, also der Abstand zwischen Inhalt (z. B. Text) und Rahmen des Elements.  
Margin bezeichnet den Außenabstand, also den Abstand zwischen dem Element und den Nachbarelementen.



### 1.8.1 Flow-Layout

Mit einem Flow-Layout, werden die Elemente automatisch angeordnet. Dabei fließen die einzelnen Elemente frei über die Seite, ähnlich wie bei einem Text. Dieses Layout bietet sich sehr für Web-Anwendungen an, die auch auf sehr kleinen Bildschirmen gut erkennbar sein sollen, da die Breite über die Anzahl der Spalten für jedes verwendete Endgerät individuell festgelegt werden kann.



#### 1.8.1.1 Mögliche Attribute für das <FlowLayout>

Für das Flow-Layout lassen sich folgende Attribute festlegen:

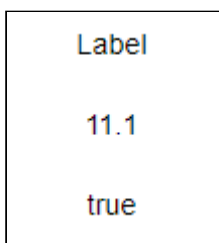
Attribut	Beschreibung
columns	Anzahl Spalten für Desktop-Ansicht  <b>Mögliche Werte:</b> Ganze Zahlen

Attribut	Beschreibung
columnsTablet	Anzahl Spalten für Ansicht am Tablet <b>Mögliche Werte:</b> Ganze Zahlen
columnsMobile	Anzahl Spalten für mobile Ansicht <b>Mögliche Werte:</b> Ganze Zahlen
direction	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• <i>horizontal</i></li> <li>• <i>vertical</i>(Standard)</li> </ul>
useColumns	Legt fest, ob für die Ausrichtung der Elemente Spalten verwendet werden. Nur für horizontale Flussrichtung. Siehe Beispiel useColumns. <b>Mögliche Werte:</b> true / false

ⓘ Wenn für `direction="vertical"` angegeben ist, dann werden die Elemente übereinander von oben nach unten angeordnet. Es erscheint ein Scrollbalken, wenn die Elemente höher sind als das Layout.

### 1.8.1.2 Beispiel für <FlowLayout>

```
<FlowLayout>
  <Label value="Label"/>
  <Label value="11.1"/>
  <Label value="true"/>
</FlowLayout>
```



### 1.8.1.3 Beispiel horizontale Flussrichtung Flow-Layout

Wenn für `direction="horizontal"` angegeben ist, dann werden die Elemente innerhalb des Flow-Layouts nebeneinander von links nach rechts angeordnet. Wenn die entsprechende Zeile voll ist, wird



das nächste Element in der Zeile darunter angezeigt. Das Layout passt sich der Breite des Bildschirms an.

#### Beispiel horizontale Flussrichtung Flow-Layout

```
<FlowLayout direction="horizontal">
  <Label value="Example 1" />
  <TextBox value="Example 2"/>
  <RadioButton optionValue="1" value="1" displayName="Example 3" />
  <Checkbox checked="true" displayName="Example 4"/>
  <Label value="Example 5" />
  <TextBox value="Example 6"/>
  <RadioButton optionValue="1" value="1" displayName="Example 7" />
  <Checkbox checked="false" displayName="Example 8"/>
</FlowLayout>
```

Obiger Code führt zu folgendem Ergebnis:

The screenshot displays a web application interface. At the top right, there is a user greeting "Hallo, SP Admin" next to a user icon and a menu icon. Below this is a dark blue horizontal bar. On the left side, there is a vertical navigation menu with the following items: "Home", "Nested Component", "Grid Layout", and "Flow Layout". The "Flow Layout" item is highlighted with a dark blue background. The main content area shows a horizontal flow layout with the following elements: "Example 1", "Example 2" (in a text box), "Example 3" (with an unselected radio button), "Example 4" (with a checked checkbox), "Example 5", "Example 6" (in a text box), "Example 7" (with an unselected radio button), and "Example 8" (with a checked checkbox). At the bottom right, there are links for "Privacy" and "Imprint".

Example 1

Example 3

Example 5

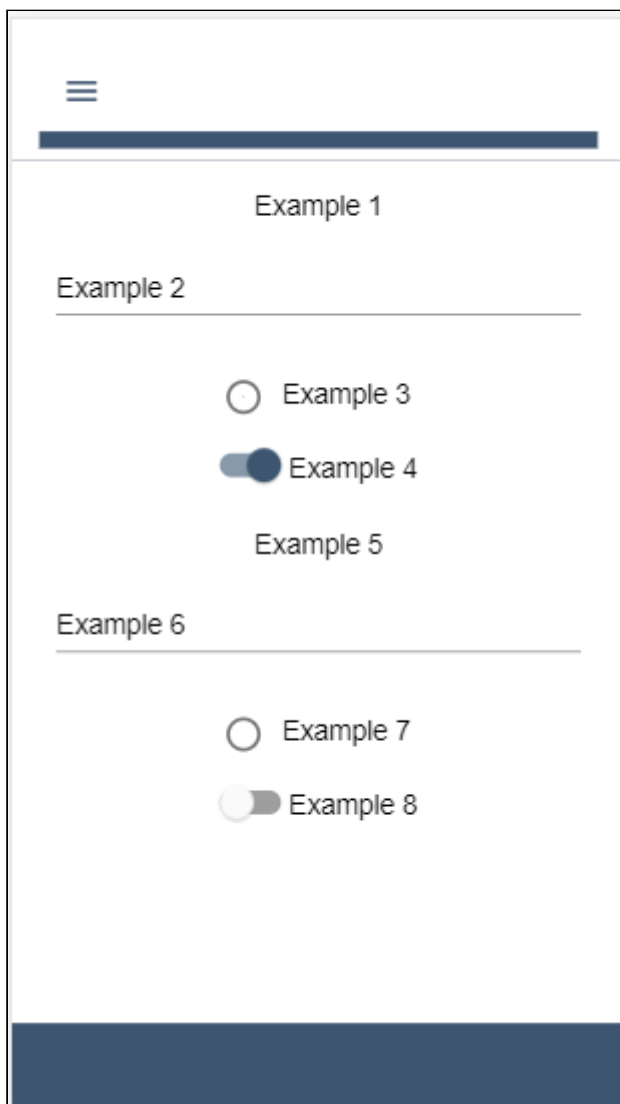
Example 7

Example 2

Example 4

Example 6

Example 8

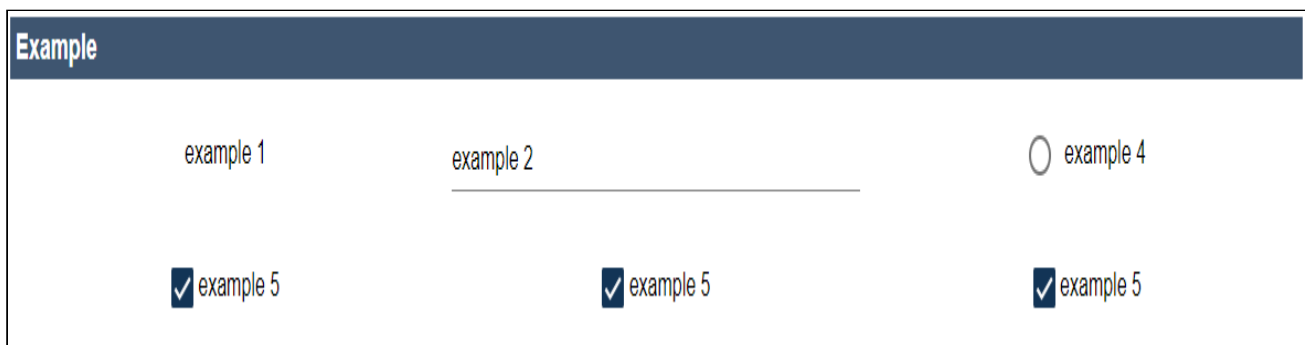


#### 1.8.1.4 Beispiel useColumns

Wenn das Attribut `useColumns` auf `true` gesetzt wird, dann werden die Elemente innerhalb des `FlowLayout` in Spalten angeordnet:

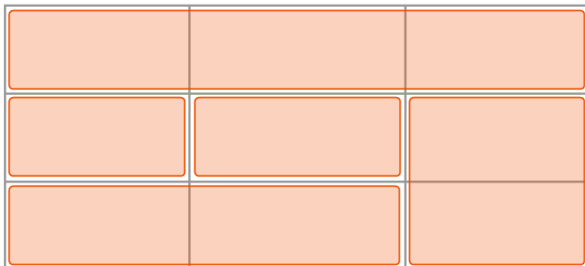
##### Beispiel useColumns Flow-Layout

```
<FlowLayout>
  <Header value="Example" />
  <FlowLayout direction="horizontal" useColumns="true" columns="3">
    <Label value="example 1" />
    <TextBox value="example 2" />
    <RadioButton optionValue="1" value="1" displayName="example 4" />
    <Checkbox checked="true" displayName="example 5" />
    <Checkbox checked="true" displayName="example 5" />
    <Checkbox checked="true" displayName="example 5" />
  </FlowLayout>
</FlowLayout>
```



## 1.8.2 Grid-Layout

Wenn für die Elemente einer Webanwendung das Grid-Layout gewählt wird, dann werden die Elemente an einem Raster ausgerichtet. Dieses Layout sorgt für ein sehr geordnetes Erscheinungsbild. Das Raster wird über die Anzahl Spalten und Zeilen festgelegt und passt sich in der Höhe und Breite dem jeweiligen Endgerät an. Das Grid-Layout wird über `<GridLayout>` erzeugt. Bei der Anzeige des Grid-Layouts auf kleineren Bildschirmen (Smartphones, Tablets) bleibt die Anzahl der Spalten identisch, die Spalten werden nur schmaler angezeigt.



### 1.8.2.1 Mögliche Attribute für `<GridLayout>`

Neben den allgemeinen [Layout-Attributen](#) können für `GridLayout` folgende zusätzliche Attribute definiert werden:

Attribut	Beschreibung
<code>alternatingRows</code>	Legt fest, ob die Zeilen verschiedene, abwechselnde Hintergrundfarben haben.  <b>Mögliche Werte:</b> <i>true / false</i>
<code>columns</code>	Legt fest, aus wie vielen Spalten das Layout-Gitter besteht.  <b>Mögliche Werte:</b> Ganze Zahlen (Standard: 3)
<code>rows</code>	Legt fest, aus wie vielen Zeilen das Layout-Gitter besteht.  <b>Mögliche Werte:</b> Ganze Zahlen

Das Element `GridLayout` kann das Element `ColumnDefinitions` enthalten. In `ColumnDefinitions` kann die Breite der einzelnen Spalten des Grid-Layouts definiert werden:


#### Beispiel ColumnDefinition

```
<GridLayout alternatingRows="false" border="true">
  <ColumnDefinitions>
    <Column width="2"/>
    <Column width="2"/>
    <Column width="1"/>
  </ColumnDefinitions>
  ...
</GridLayout>
```

Wenn die Breite der Spalten über die Spaltendefinition festgelegt wird, dann ist es nicht notwendig, eine Anzahl von Spalten für das Layout anzugeben. Die Anzahl der Spalten wird aus der Spaltendefinition abgeleitet. In obigem Beispiel werden drei Spalten angezeigt, da das Element `Column` dreimal existiert. Dabei sind die ersten beiden Spalten doppelt so breit wie die dritte Spalte.

Für das Element `Column` lassen sich zudem folgende Attribute definieren:

Attribut	Beschreibung
<code>horizontalAlign</code>	Positionierung innerhalb der Spalte  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• <i>left</i> (Standard)</li> <li>• <i>center</i></li> <li>• <i>right</i></li> </ul>
<code>marginBottom</code>	Unterer Außenabstand  <b>Mögliche Werte:</b> Beliebige Zahlenwerte
<code>marginLeft</code>	Linker Außenabstand  <b>Mögliche Werte:</b> Beliebige Zahlenwerte
<code>marginRight</code>	Rechter Außenabstand  <b>Mögliche Werte:</b> Beliebige Zahlenwerte
<code>marginTop</code>	Oberer Außenabstand  <b>Mögliche Werte:</b> Beliebige Zahlenwerte
<code>marginUnits</code>	Einheiten für den Außenabstand  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• <i>pixels</i></li> <li>• <i>percents</i></li> </ul>

Attribut	Beschreibung
paddingBottom	Unterer Innenabstand <b>Mögliche Werte:</b> Beliebige Zahlenwerte
paddingLeft	Linker Innenabstand <b>Mögliche Werte:</b> Beliebige Zahlenwerte
paddingRight	Rechter Innenabstand <b>Mögliche Werte:</b> Beliebige Zahlenwerte
paddingTop	Oberer Innenabstand <b>Mögliche Werte:</b> Beliebige Zahlenwerte
paddingUnits	Einheiten für den Innenabstand <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• pixels</li> <li>• percents</li> </ul>
units	Gibt die Einheit der Breitenangabe width an <div>  Wird explizit das Attribut unit gesetzt, müssen alle Column-Elemente den gleichen Wert für unit besitzen.         </div> <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• pixels (Standard)</li> <li>• percents</li> </ul>
width	Legt die Breite der Spalte fest <b>Mögliche Werte:</b> Ganze Zahlen

### 1.8.2.2 Beispiel <GridLayout>

#### Beispiel Grid-Layout

```
<GridLayout columns="2" alternatingRows="true" border="true">
  <ColumnDefinitions>
    <Column width="1" />
    <Column width="2" />
  </ColumnDefinitions>
  <FlowLayout>
    <Label value="example" />
    <Label value="example" />
  </FlowLayout>
  <Label value="example" />
  <FlowLayout>
    <Label value="example" />
    <Label value="example" />
  </FlowLayout>
  <Label value="example" />
</GridLayout>
```

### 1.8.3 Registerkarten-Layout: Tab Group

Um Registerkarten innerhalb einer **Master/Detail-Komponente** zu verwenden, kann in **Master-**, **Detail-** und **Overlay-Strukturelementen** das Element **<TabGroup>** verwendet werden. Das **<TabGroup>**-Element enthält eine oder mehrere Komponenten, die in jeweils einer Registerkarte angezeigt werden.

Attribut	Beschreibung
description	Beschreibungstext der Komponente. Wird in jedem Tab über der jeweiligen Komponente angezeigt. <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <b>Mögliche Werte:</b> Beliebige Zeichenkette
title	Titel der Gruppe <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <b>Mögliche Werte:</b> Beliebige Zeichenkette

Attribut	Beschreibung
subtitle	Untertitel der Gruppe <ul style="list-style-type: none"><li>• Data Binding möglich</li></ul> <b>Mögliche Werte:</b> Beliebige Zeichenkette

**Beispiel für <TabGroups>**

Invoice Information	Invoices
Name:	<input type="text"/>
First Name:	<input type="text"/>
Address:	<input type="text"/>



**Beispiel für ein**

```

<Detail>
  <TabGroup>
    <DetailComponent default="true" displayName="Invoice Information" name="InvoiceInformation">
      <Properties/>
      <Actions>
        <SaveAction displayName="$saveBtn" process="Save.wrf"/>
      </Actions>
      <Components/>
      <FlowLayout>
        <TextBox iconPosition="prefix" type="text" value="#Invoice.Name"/>
        <TextBox iconPosition="prefix" type="text" value="#Invoice.FirstName"/>
      </FlowLayout>
    </DetailComponent>
    <ListComponent name="Liste" displayName="Invoices" process="InvoiceData.wrf" showInMenu="true" >
      <Actions>
        <SelectAction componentName="InvoiceData" disableAutoSelect="false" displayName="Select" selectionNeeded="true"/>
      </Actions>
      <Columns>
        <Column mobilePosition="topLeft" value="#Invoice.Number"/>
        <Column mobilePosition="topRight" value="#Invoice.Name"/>
      </Columns>
    </ListComponent>
  </TabGroup>
</Detail>

```

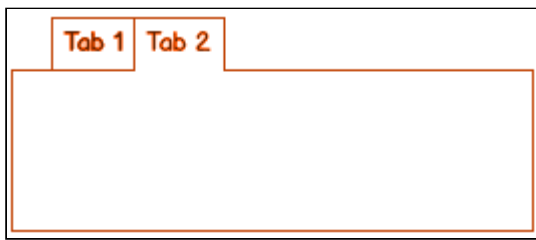
### 1.8.4 Registerkarten-Layout: Tab Layout

Mit dem Registerkarten-Layout können weitere Layouts (Flow-Layout oder Grid-Layout) auf Registerkarten angeordnet werden. Dieses Layout ist gut geeignet, wenn viele Informationen zur Verfügung gestellt werden sollen, die thematisch geordnet werden können, z. B. Formulare.

**i** Das Registerkarten-Layout kann sowohl in Komponenten als auch innerhalb von anderen Layouts verwendet werden.

Ein Registerkarten-Layout wird mit `<TabLayout>` erzeugt. Ein Registerkarten-Layout muss mindestens ein `<Tab>`-Element enthalten.

**i** Ein `<Tab>` muss genau ein `<FlowLayout>` oder ein `<GridLayout>` enthalten.



#### 1.8.4.1 Attribute für <TabLayout>

Zusätzlich zu den in [Layouts](#) genannten Attributen können folgende Attribute vergeben werden:

Attribut	Beschreibung
title	Titel für das Registerkarten-Layout, wird als Header angezeigt. <ul style="list-style-type: none"><li>• Übersetzbar</li><li>• Data Binding möglich</li></ul> <b>Mögliche Werte:</b> Beliebige Zeichenkette

#### 1.8.4.2 Attribute für <Tab>

Zusätzlich zu den in [Layouts](#) genannten Attributen können folgende Attribute vergeben werden:

Attribut	Beschreibung
displayName	Angezeigter Titel für die Registerkarte <ul style="list-style-type: none"><li>• Übersetzbar</li><li>• Data Binding möglich</li></ul> <b>Mögliche Werte:</b> Beliebige Zeichenkette

#### 1.8.4.3 Beispiel für ein <TabLayout>

```
<TabLayout title="Tab layout">
    <Tab displayName="Tab 1">
        <FlowLayout>
            <Label value="tab 1" />
        </FlowLayout>
    </Tab>
    <Tab displayName="Tab 2">
        <GridLayout>
            <Label value="tab 1" />
            <Label value="tab 2" />
            <Label value="tab 3" />
        </GridLayout>
    </Tab>
</TabLayout>
```



## 1.9 Steuerelemente

In diesem Abschnitt erfahren Sie, was Steuerelemente sind und welche Attribute für alle Steuerelemente verwendet werden können.





Mit Steuerelementen können Informationen in der Web-Anwendung ausgegeben und Benutzereingaben verarbeitet werden.

In der folgenden Tabelle sind die Attribute erläutert, die für alle Steuerelemente verwendet werden können.

Attribut	Beschreibung
background	<p>Definiert eine Farbe für den Hintergrund des Steuerelements.</p> <div> <p><b>i</b></p> <ul style="list-style-type: none"> <li>Diese Einstellung überschreibt die Standardfarbe des Farbschemas!</li> <li>Dieses Attribut funktioniert nicht für das Steuerelement Maps!</li> </ul> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div> <p><b>⚠</b> Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts!</p> </div> <ul style="list-style-type: none"> <li>Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>
border	<p>Legt fest, ob um das Steuerelement ein Rahmen angezeigt wird.</p> <p><b>Mögliche Werte:</b> <i>true</i> / <i>false</i></p>

Attribut	Beschreibung
enabled	<p>Legt fest, ob der Nutzer mit dem Steuerelement interagieren kann.</p> <ul style="list-style-type: none"> <li>• Data Binding (Boolean) möglich</li> </ul> <p><b>Mögliche Werte:</b> true/false oder Ausdruck für Data Binding</p> <div> <p><b>i</b> Das Attribut <code>enabled</code> ersetzt das veraltete Attribut <code>disabled</code>. <code>enabled="true"</code> entspricht damit dem veralteten Attribut <code>disabled="false"</code>.</p> </div>
fontFamily	<p>Legt die Schriftfamilie fest.</p> <div> <p><b>i</b></p> <ul style="list-style-type: none"> <li>• Mit diesem Attribut wird die Standardschrift der Web App für dieses Steuerelement überschrieben.</li> <li>• Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut <code>fontFamily</code> nicht explizit gesetzt haben.</li> <li>• Dieses Attribut funktioniert nicht für die Elemente <code>Image</code> und <code>Map</code>.</li> </ul> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <code>MainFont</code>: hinterlegte Hauptschrift</li> <li>• Fontcode aus der Font-Palette, z. B. <code>Font04</code></li> </ul>
fontSize	<p>Legt die Schriftgröße fest.</p> <div> <p><b>i</b></p> <ul style="list-style-type: none"> <li>• Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Steuerelement überschrieben.</li> <li>• Dieses Attribut funktioniert nicht für die Elemente <code>Image</code> und <code>Map</code>.</li> </ul> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code></li> <li>• Schriftgröße in Pixel, z. B. <code>20px</code></li> <li>• Schriftgröße in Punkt, z. B. <code>18pt</code></li> <li>• Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code></li> <li>• Schlüsselwörter: <code>xx-small</code>, <code>x-small</code>, <code>small</code>, <code>medium</code>, <code>large</code>, <code>x-large</code>, <code>xx-large</code>, <code>smaller</code>, <code>larger</code></li> </ul>

Attribut	Beschreibung
fontStretch	<p data-bbox="545 318 1168 353">Legt die Breite der einzelnen Schriftzeichen fest.</p> <div data-bbox="545 385 1434 495"> <p data-bbox="571 407 1342 472"><b>i</b> Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Steuerelement überschrieben.</p> </div> <p data-bbox="545 504 756 539"><b>Mögliche Werte:</b></p> <ul data-bbox="600 571 863 904" style="list-style-type: none"> <li>• Condensed</li> <li>• Expanded</li> <li>• ExtraCondensed</li> <li>• ExtraExpanded</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• SemiCondensed</li> <li>• SemiExpanded</li> <li>• UltraCondensed</li> <li>• UltraExpanded</li> </ul> <div data-bbox="545 969 1434 1099"> <p data-bbox="571 992 1410 1057"><b>i</b> Dieses Attribut funktioniert nicht für die Elemente Image, Map und Chart.</p> </div>
fontStyle	<p data-bbox="545 1124 970 1160">Legt die Neigung der Schrift fest.</p> <div data-bbox="545 1191 1434 1301"> <p data-bbox="571 1214 1342 1279"><b>i</b> Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Steuerelement überschrieben.</p> </div> <p data-bbox="545 1310 756 1346"><b>Mögliche Werte:</b></p> <ul data-bbox="600 1377 1257 1478" style="list-style-type: none"> <li>• <i>italic</i>: kursive Schrift</li> <li>• <i>normal</i>: normale Schrift (Standard)</li> <li>• <i>oblique</i>: schräg gestellter Schriftstil (berechnet)</li> </ul> <div data-bbox="545 1543 1434 1673"> <p data-bbox="571 1565 1426 1630"><b>i</b> Dieses Attribut funktioniert nicht für die Elemente Image und Map.</p> </div>

Attribut	Beschreibung
fontWeight	<p data-bbox="544 320 890 353">Legt die Schriftstärke fest.</p> <div data-bbox="544 387 1434 495"> <p> Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Steuerelement überschrieben.</p> </div> <p data-bbox="544 506 756 539"><b>Mögliche Werte:</b></p> <ul data-bbox="596 573 863 1111" style="list-style-type: none"> <li>• Black</li> <li>• Bold</li> <li>• DemiBold</li> <li>• ExtraBlack</li> <li>• ExtraBold</li> <li>• ExtraLight</li> <li>• Heavy</li> <li>• Light</li> <li>• Medium</li> <li>• Normal(Standard)</li> <li>• Regular</li> <li>• SemiBold</li> <li>• Thin</li> <li>• UltraBlack</li> <li>• UltraBold</li> <li>• UltraLight</li> </ul> <div data-bbox="544 1178 1434 1301"> <p> Dieses Attribut funktioniert nicht für die Elemente Image und Map.</p> </div>
foreground	<p data-bbox="544 1328 1273 1395">Definiert eine Farbe für den Vordergrund (Texte usw.) des Steuerelements.</p> <div data-bbox="544 1429 1434 1626"> <p> • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für das Steuerelement Maps!</p> </div> <p data-bbox="544 1664 756 1697"><b>Mögliche Werte:</b></p> <ul data-bbox="560 1709 1070 1742" style="list-style-type: none"> <li>• Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div data-bbox="592 1753 1434 1899"> <p> Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts!</p> </div> <ul data-bbox="560 1910 1326 1977" style="list-style-type: none"> <li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>

Attribut	Beschreibung
height	Höhe des Steuerelements  <b>Mögliche Werte:</b> Ganzzahl
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration.  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• <i>left</i>(Standard)</li> <li>• <i>center</i></li> <li>• <i>right</i></li> </ul>
textOverflow	Gibt an, was passieren soll, wenn das Steuerelement voll ist.  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist</li> <li>• <i>hidden</i>: Text abbrechen, auf ganze Wörter achten</li> <li>• <i>wordBreak</i>: innerhalb des Wortes abbrechen</li> <li>• <i>allow</i>(Standard): Text zwischen den Wörtern umbrechen</li> </ul>
units	Gibt die Einheit an, die für Größenangaben gilt.  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• <i>pixels</i>(Standard)</li> <li>• <i>percents</i></li> </ul>
visible	Legt fest, ob das Steuerelement sichtbar ist.  <ul style="list-style-type: none"> <li>• Data Binding(Boolean) möglich</li> </ul> <b>Mögliche Werte:</b> <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding
width	Breite des Steuerelements  <b>Mögliche Werte:</b> Ganze Zahlen

### 1.9.1 Header

<Header>-Steuerelemente werden verwendet, um Überschriften zu erzeugen. Ein Header kann nicht nur im oberen Teil eines Layouts platziert werden, sondern überall, wo er benötigt wird.

Attribut	Beschreibung
value	Enthält den Header-Text.  <ul style="list-style-type: none"> <li>• Übersetzbar</li> <li>• Data Binding möglich</li> </ul> <b>Mögliche Werte:</b> Beliebige Zeichenkette

Attribut	Beschreibung
titleLevel	Definiert die Überschriftenebene.  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• <i>title</i>(Standard)</li> <li>• <i>subtitle</i></li> </ul>

### 1.9.1.1 Beispiel <Header>

Im folgenden Beispiel ist die Verwendung des <Header>-Tags gezeigt.

```
<Header value="Header"/>
```

Der obige Code führt zu folgendem Header:

**This is an example.**

## 1.9.2 Label

<Label>-Steuerelemente werden verwendet, um Text anzuzeigen. <Label>-Steuerelemente können nicht durch den Benutzer der Web App editiert werden.

Attribut	Beschreibung
value	Enthält den Label-Text <ul style="list-style-type: none"> <li>• Übersetzbar</li> <li>• Data Binding möglich</li> </ul> <b>Mögliche Werte:</b> Beliebige Zeichenkette

### 1.9.2.1 Beispiel <Label>

Im folgenden Beispiel ist die Verwendung des <Label>-Tags gezeigt. In diesem Beispiel werden die Labels mittels Data Binding befüllt.

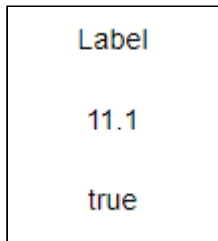
```
<Properties>
  <Property name="number" type="Decimal" />
  <Property name="boolean" type="Boolean" />
</Properties>
<FlowLayout>
  <Label value="Label"/>
  <Label value="#number"/>
  <Label value="#boolean"/>
</FlowLayout>
```

Die Daten, die per Data Binding in den Labels angezeigt werden, müssen von einem Prozess in folgender Form geliefert werden:



```
<?xml version="1.0" encoding="UTF-8" ?>
<number>11.1</number>
<boolean>1</boolean>
```


Der obige Code führt zu folgenden Labels:






### 1.9.3 TextBox (Web App)

<TextBox>-Steuerelemente werden verwendet, um mit Text, Zahlen oder Datumsangaben zu arbeiten. Je nach Typ der TextBox ändert sich ihr Aussehen und Verhalten.

Neben den allgemeinen [Attributen für Steuerelemente](#) lassen sich für Textboxen folgende Attribute definieren:

Attribut	Beschreibung
border	<p>Legt fest, ob um das Steuerelement ein Rahmen angezeigt wird. Funktioniert nur, wenn <code>multiline="true"</code>.</p> <p><b>Mögliche Werte:</b> <i>true / false</i></p>
displayName	<p>Titel der TextBox. Wird klein über dem Ein-/Ausgabefeld angezeigt.</p> <ul style="list-style-type: none"> <li>• Übersetzbar</li> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
iconColor	<p>Definiert die Farbe des TextBox-Icons.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Hexadezimaler Farbwert, z. B. <i>ff5a00</i></li> </ul> <div style="border: 1px solid #ffc107; padding: 5px; margin: 10px 0;"> <p> Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts!</p> </div> <ul style="list-style-type: none"> <li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. <i>A200</i></li> </ul>
iconPosition	<p>Position des Icons innerhalb der TextBox.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <i>prefix</i> (Standard): Icon wird zu Beginn der TextBox angezeigt.</li> <li>• <i>suffix</i>: Icon wird am Ende der TextBox angezeigt.</li> </ul>

Attribut	Beschreibung
iconUrl	<p>Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Icon verwendet wird.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <div> <p><b>i</b></p> <ul style="list-style-type: none"> <li>• Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources.</li> <li>• Das Material Icon muss mit dem Präfix <code>icon:</code>, z. B. <code>icon:&lt;MaterialIconName&gt;</code> angegeben werden.</li> </ul> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Zeichenkette (URI), z. B. <code>clock.png</code></li> <li>• <code>icon:&lt;MaterialIconName&gt;</code>, z. B. <code>icon:extension</code></li> </ul> <div> <p><b>✓</b></p> <p>Mit <code>Strg+Leertaste</code> erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren <b>Material Icons</b> abweichen.</p> </div>
multiline	<p>Legt fest, ob in der TextBox mehrzeilige Ein-/Ausgaben möglich sind. Funktioniert nur beim Typ <code>text</code>.</p> <p><b>Mögliche Werte:</b> <code>true</code> / <code>false</code></p>
rows	<p>Legt die sichtbare Höhe des Textfeldes fest. Funktioniert nur bei mehrzeiligen Textfeldern. Es können mehr Zeilen eingegeben werden, diese sind dann über Scrollen sichtbar.</p> <p><b>Mögliche Werte:</b> Ganze Zahlen (Standard: 5)</p>
showSeconds	<p>Legt fest, ob bei Textboxen des Typs <code>datetime</code> die Sekunden ebenfalls angezeigt werden sollen</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <code>true</code>: Sekunden werden beim Date-/Timepicker ebenfalls angezeigt</li> <li>• <code>false</code>: Sekunden werden beim Date-/Timepicker nicht angezeigt (Standard)</li> </ul>

Attribut	Beschreibung
type	<p>Typ der TextBox, ändert Verhalten und Aussehen.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <i>date</i>: Datum mit Datepicker</li> <li>• <i>datetime</i>: Datum und Zeit mit Date-/Timepicker</li> </ul> <div style="border: 1px solid #add8e6; padding: 5px; margin: 5px 0;"> <p> Mit dem Attribut <code>showSeconds</code> lässt sich zusätzlich definieren, ob der Date-/Timepicker auch Sekundenangaben anzeigt.</p> </div> <div style="border: 1px solid #ffcc00; padding: 5px; margin: 5px 0;"> <p> Das Datum und die Uhrzeit werden entsprechend der Zeitzone des Browsers angezeigt.</p> </div> <ul style="list-style-type: none"> <li>• <i>email</i>: E-Mail-Adresse</li> <li>• <i>month</i>: Monat und Jahr mit Datepicker</li> <li>• <i>number</i>: Zahl mit Wählpfeilen (Spinner)</li> <li>• <i>password</i>: Passwortfeld mit versteckten Zeichen</li> <li>• <i>search</i>: Suchfeld</li> <li>• <i>tel</i>: Telefonnummer</li> <li>• <i>text</i> (Standard): Text</li> <li>• <i>time</i>: Zeit mit Wählpfeilen (Spinner)</li> <li>• <i>url</i>: URL</li> <li>• <i>week</i>: Kalenderwoche und Jahr mit Datepicker und Wählpfeilen (Spinner)</li> </ul> <div style="border: 1px solid #ffcc00; padding: 5px; margin: 5px 0;"> <p> Alle Werte, die ein Datum oder eine Uhrzeit enthalten, müssen in UTC gespeichert und bereitgestellt werden!</p> </div>
value	<p>Inhalt der TextBox.</p> <ul style="list-style-type: none"> <li>• Übersetzbar</li> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>

### 1.9.3.1 Beispiel <TextBox>

Im folgenden Beispiel ist die Verwendung des <TextBox>-Tags gezeigt.

```
<FlowLayout>
  <TextBox value="Example 1"/>
  <TextBox value="Example 2" multiline="true"/>
  <TextBox value="2018-04-12" type="date"/>
</FlowLayout>
```

Der obige Code führt zu folgenden TextBoxen:


Example 1


Example 2

12.4.2018

### 1.9.4 ComboBox

<ComboBox>-Steuerelemente werden verwendet, um Auswahllisten zu erzeugen.

Attribut	Beschreibung
displayName	<p>Titel der ComboBox. Wird klein über dem Ein-/Ausgabefeld angezeigt.</p> <ul style="list-style-type: none"> <li>• Übersetzbar</li> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
displayProperty	<p>Muss angegeben werden, wenn optionsList gesetzt ist. Text, der in der ComboBox als Auswahlmöglichkeit angezeigt wird.</p> <p><b>Mögliche Werte:</b> Zeichenkette, die einer definierten Property entspricht.</p> <div>  Die angegebene Zeichenkette muss einer validen Subproperty entsprechen, die in optionsList verwendet wird. Wenn kein displayProperty angegeben wird, dann wird automatisch die erste Subproperty der Property verwendet, die für optionsList angegeben ist. </div>
optionsList	<p><i>Bei ausgelagerter Optionsliste:</i> Quelle der Liste. Kann nur zusammen mit displayProperty verwendet werden.</p> <ul style="list-style-type: none"> <li>• Wird per Data Binding (Liste) angesteuert.</li> </ul> <p><b>Mögliche Werte:</b> Zeichenkette (Data Binding)</p>
value	<p>Wert, der angezeigt und gespeichert wird.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>

Attribut	Beschreibung
valueProperty	<p>Wert, der technisch verarbeitet wird. Wenn dieses Attribut gesetzt ist, muss auch die für diesen Wert definierte Property so heißen. Wenn dieses Attribut nicht gesetzt wird, wird value verwendet.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p> <div style="border: 1px solid #f0e68c; padding: 10px; margin-top: 10px;"> <p> Die angegebene Zeichenkette muss einer validen Subproperty entsprechen, die in optionsList verwendet wird. Wenn kein valueProperty angegeben wird, dann wird automatisch die der Wert verwendet, der für displayProperty angegeben ist.</p> </div>

ComboBox enthält mindestens ein <Option>-Tag, das eine Auswahloption erzeugt. Die Optionen der Auswahlliste können entweder aus einem Prozess kommen oder manuell eingetragen werden.

Die Daten für eine ausgelagerte Optionenliste, müssen als XML-Daten vorliegen. Dabei müssen die Optionen in einer Liste enthalten sein:

```
<example>
  <listElement>list element 1</listElement>
</example>
<example>
  <listElement>list element 2</listElement>
</example>
```

Die Properties-Definition erfolgt entsprechend der XML-Daten. Dabei ist zu beachten, dass eine zusätzliche Property saveValue angelegt wird, die als initialen Wert und Container für das Speichern der Benutzereingabe verwendet wird.

```
<Property name="example" type="List">
  <Property name="listElement" type="String" />
</Property>
<Property name="saveValue" type="String" />
```

#### 1.9.4.1 Aktion "Auswählen" innerhalb einer ComboBox

Die Aktion **Auswählen** wird über das Element <SelectAction/> innerhalb von <ComboBox> erzeugt und ausgelöst, sobald ein Eintrag in der ComboBox ausgewählt wurde.

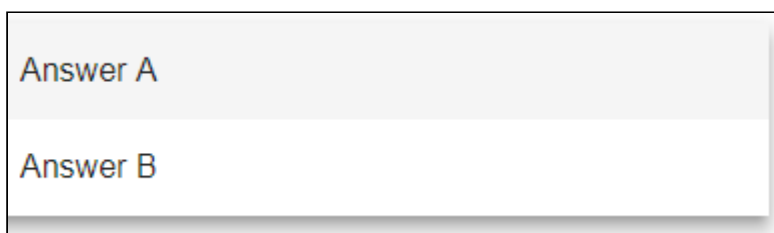
#### 1.9.4.2 Beispiel <ComboBox>

In folgendem Beispiel ist die Verwendung des <ComboBox>-Tags gezeigt.

```
<DetailComponent name="ExampleComboBox" path="ExampleComboBox" displayName="Example
ComboBox" process="ComboBox.wrf">
  <FlowLayout>
    <ComboBox value="#saveValue">
      <Option displayName="Answer A"/>
      <Option displayName="Answer B"/>
    </ComboBox>
  </FlowLayout>
</DetailComponent>
```

Obiger Code führt zu folgender ComboBox:

Answer A ▾



#### 1.9.4.3 Beispiel <ComboBox> mit ausgelagerter Optionsliste

In folgendem Beispiel ist die Verwendung des <ComboBox>-Tags mit ausgelagerter Optionsliste gezeigt.

##### \*.wad-Datei ComboBox

```
<DetailComponent name="ExampleComboBox" path="ExampleComboBox" displayName="Example
ComboBox" process="ComboBox.wrf">
  <Properties>
    <Property name="example" type="List">
      <Property name="listElement" type="String" />
    </Property>
    <Property name="saveValue" type="String" />
  </Properties>
  <FlowLayout>
    <ComboBox value="#saveValue" optionsList="#example" >
    </ComboBox>
  </FlowLayout>
</DetailComponent>
```

**\*.xml-Datei Optionsliste ComboBox**

```
<?xml version="1.0" encoding="UTF-8" ?>
<Ok>
  <example>
    <listElement>list element 1</listElement>
  </example>
  <example>
    <listElement>list element 2</listElement>
  </example>
</Ok>
```

**i** Die XML-Daten, die für die Optionen verwendet werden sollen, müssen von einem Prozess (im Beispiel ComboBox.wrf) zur Verfügung gestellt werden.

Obiger Code führt zu folgender ComboBox:


The image shows a graphical user interface element, a ComboBox. The text 'list element 2' is displayed in the main field of the control. Below this field, a dropdown menu is open, showing a list of two items: 'list element 2' and 'list element 1'. The item 'list element 2' is currently selected and highlighted with a light blue background.



### 1.9.5 Checkbox

<Checkbox>-Steuerelemente werden verwendet, um mit Boole'schen Werten zu arbeiten, z. B. Benutzereingaben im "Ja/Nein"-Format zu erhalten.

Attribut	Beschreibung
checked	<p><b>obligatorisch</b></p> <p>Gibt an, ob die Checkbox standardmäßig aktiviert oder deaktiviert ist.</p> <ul style="list-style-type: none"> <li>• Data Binding (Boolean) möglich</li> </ul> <div> <p><b>i</b> Die Verwendung von Data Binding für den Status der Checkbox ist notwendig, damit die Checkbox ordnungsgemäß funktioniert und den initialen Wert von result bzw. set zu speichern.</p> </div> <p><b>Mögliche Werte:</b> <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding</p>

Attribut	Beschreibung
displayName	Beschriftung. Wird neben der Checkbox angezeigt <ul style="list-style-type: none"><li>• Data Binding möglich</li></ul> <b>Mögliche Werte:</b> Beliebige Zeichenkette
sliderOnDesktop	Legt fest, ob die Checkbox in der Desktopversion als Schieber (Slider) angezeigt wird. <b>Mögliche Werte:</b> <i>true</i> / <i>false</i>

 Checkboxes werden in der Desktop- und in der mobilen Version unterschiedlich angezeigt.

- Anzeige in der Desktopversion: 
- Anzeige in der mobilen Version: 

mit dem Attribut `sliderOnDesktop` kann die Anzeige für die mobile Version auch in der Desktopversion angezeigt werden.


#### 1.9.5.1 Aktion "Auswählen" innerhalb einer Checkbox

Die Aktion `Auswählen` wird über das Element `<SelectAction/>` innerhalb von `<Checkbox>` erzeugt und ausgelöst, sobald diese ausgewählt wurde.

#### 1.9.5.2 Beispiel für `<Checkbox>`

```
<DetailComponent name="Example" path="Example" displayName="Example">
  <FlowLayout>
    <Checkbox checked="true" displayName="Example"/>
  </FlowLayout>
</DetailComponent>
```

Obiger Code erzeugt folgende Checkbox:

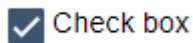
 Example



### 1.9.5.3 Beispiel für <Checkbox> mit Prozess

```
<DetailComponent path="Dashboard"
  displayName="Dashboard" default="true"
  process="/Processes/process_1.wrf">
  <Properties>
    <Property name="boolean" type="Boolean"></Property>
  </Properties>
  <FlowLayout>
    <Checkbox checked="#boolean" displayName="Check box" />
  </FlowLayout>
</DetailComponent>
```

Obiges Beispiel erzeugt folgende Checkbox:



### 1.9.6 ListView

ListView-Steuerelemente werden verwendet, um Datenobjekte (einfach oder komplex) aus Listen darzustellen. Wie ein einzelnes Objekt dargestellt wird, wird als Elementvorlage definiert. Dazu wird die Darstellung innerhalb eines Layouts ([Flow-](#) oder [Grid-Layout](#)) definiert. Diese Darstellung wird für jedes einzelne Objekt übernommen. Da die Datenobjekte, die dargestellt werden sollen, von einem Prozess geliefert werden, muss in der Komponente, die die ListView enthält, ein entsprechender Prozess definiert sein.

Attribut	Beschreibung
elementWidth	Gibt die Breite eines Elements (entspricht einem Listeneintrag in der XML-Datei) an.  <b>Mögliche Werte:</b> Ganzzahl
elementWidthUnits	Gibt die Einheit an, die für Größenangabe des Elements gilt.  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• <i>pixels</i></li> <li>• <i>percents</i> (Standard)</li> </ul>
maxLength	Gibt an, wie viele Einträge maximal angezeigt werden. Sind in der Liste mehr Einträge vorhanden, kann sich der Benutzer diese über einen Ausklapp-Pfeil anzeigen lassen.  <b>Mögliche Werte:</b> Ganzzahl

Attribut	Beschreibung
objectList	<i>Erforderlich.</i> Quelle der Daten. Die Anzahl der Einträge definiert die Anzahl der angezeigten Elemente. <ul style="list-style-type: none"> <li>• Data Binding (Liste) möglich</li> </ul> <b>Mögliche Werte:</b> Beliebige Zeichenkette

### 1.9.6.1 Aktion "Auswählen" innerhalb einer ListView

Die Aktion **Auswählen** wird über das Element `<SelectAction/>` innerhalb von `<ListView>` erzeugt und ausgelöst, sobald ein Eintrag in der ListView ausgewählt wurde.

### 1.9.6.2 Beispiel `<ListView>`

Im folgenden Beispiel ist die Verwendung des `<ListView>`-Tags gezeigt.

```
<DetailComponent name="ExampleListView" path="ExampleListView" displayName="Example
ListView" process="example.wrf">
  <Properties>
    <Property name="ListObject" type="List">
      <Property name="Caption" type="String" />
      <Property name="Name" type="String" />
      <Property name="City" type="String" />
    </Property>
  </Properties>
  <FlowLayout>
    <ListView objectList="#ListObject" border="true">
      <FlowLayout>
        <Label value="#Caption"/>
        <Label value="#Name"/>
        <Label value="#City"/>
      </FlowLayout>
    </ListView>
  </FlowLayout>
</DetailComponent>
```

Die Daten, die vom Prozess geliefert werden, müssen folgendes Format haben:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Ok>
  <ListObject>
    <Caption>Example 1</Caption>
    <Name>Max Mustermann</Name>
    <City>Ettlingen</City>
  </ListObject>
  <ListObject>
    <Caption>Example 2</Caption>
    <Name>John Doe</Name>
    <City>New York</City>
  </ListObject>
  <ListObject>
    <Caption>Example 3</Caption>
    <Name>Kari Nordmann</Name>
    <City>Oslo</City>
  </ListObject>
  ...
</Ok>
```

Obiger Code erzeugt folgende ListView:

Example 1	Example 2	Example 3
Max Mustermann	John Doe	Kari Nordmann
Ettlingen	New York	Oslo

### 1.9.7 Radio-Button

<RadioButton>-Steuerelemente werden verwendet, um eine Gruppe von Buttons zu erzeugen. Der Benutzer kann immer nur einen Radio-Button aus dieser Gruppe gleichzeitig auswählen.

Attribut	Beschreibung
displayName	Beschriftung des Radio-Buttons <ul style="list-style-type: none"> <li>• Übersetzbar</li> <li>• Data Binding möglich</li> </ul> <b>Mögliche Werte:</b> Beliebige Zeichenkette
optionValue	<i>Erforderlich.</i> Technischer Wert, der weiterverarbeitet werden soll. <ul style="list-style-type: none"> <li>• Data Binding (String oder Boolean) möglich</li> </ul> <b>Mögliche Werte:</b> Beliebige Zeichenkette

Attribut	Beschreibung
value	<p><i>Erforderlich.</i> Radio-Buttons mit gleichem value bilden eine Gruppe, von der jeweils nur ein Radio-Button ausgewählt werden kann.</p> <ul style="list-style-type: none"> <li>• Data Binding (String oder Boolean) erforderlich</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>

### 1.9.7.1 Aktion "Auswählen" innerhalb eines Radio-Buttons

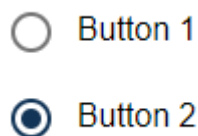
Die Aktion Auswählen wird über das Element `<SelectAction/>` innerhalb von `<RadioButton>` erzeugt und ausgelöst, sobald dieser ausgewählt wurde.

### 1.9.7.2 Beispiel `<RadioButton>`

Im folgenden Beispiel ist die Verwendung des `<RadioButton>`-Tags gezeigt.


```
<DetailComponent name="ExampleRadioButton" path="ExampleRadioButton" displayName="Example RadioButton" process="exampleProcess.wrf">
  <FlowLayout>
    <RadioButton optionValue="true" value="#group" displayName="Button 1" />
    <RadioButton optionValue="false" value="#group" displayName="Button 2"/>
  </FlowLayout>
</DetailComponent>
```

Obiger Code erzeugt folgende Radio-Buttons:



## 1.9.8 Image

`<Image>`-Steuerelemente werden verwendet, um Grafiken einzubinden.

Attribut	Beschreibung
iconColor	<p>Definiert die Farbe des Icons.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Hexadezimaler Farbwert, z. B. <code>ff5a00</code></li> </ul> <div style="border: 1px solid #f0e68c; padding: 10px; margin: 10px 0;"> <p> Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts!</p> </div> <ul style="list-style-type: none"> <li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. <code>A200</code></li> </ul>

Attribut	Beschreibung
value	<p>Grafik, die angezeigt wird.</p> <p>Verweist auf eine Property (vom Typ Base64) innerhalb der Komponente. Wird benötigt, um Grafiken dynamisch anzuzeigen.</p> <p><b>Mögliche Werte:</b> Dateipfad zum Speicherort der Grafik oder Zeichenkette für Data Binding (Image, Base64, String)</p> <div style="border: 1px solid orange; padding: 10px; margin-top: 10px;"> <p>⚠ Wenn eine interne Grafik verwendet werden soll, dann muss die Grafikdatei im Ordner <code>Resources</code> direkt unterhalb des Web-App-Projekts liegen.</p> </div>

- ✓ Die Größe des Bildes kann über die [allgemeinen Attribute](#) `height` und `width` angepasst werden. Die Angabe erfolgt standardmäßig in Pixeln.  
Wenn nur Höhe oder nur Breite angegeben wird, wird das Seitenverhältnis beibehalten. Wenn sowohl Höhe als auch Breite angegeben werden, wird das Bild verzerrt.

- ⚠ Werden weder auf dem Image-Steuerelement noch auf dem übergeordneten Element Größenangaben definiert (über die [allgemeinen Attribute](#) `height` und `width`) kann es in manchen Browsern zu Darstellungsproblemen kommen.

#### 1.9.8.1 Aktion "Auswählen" innerhalb eines Image-Steuerelements

Die Aktion `Auswählen` wird über das Element `<SelectAction/>` innerhalb von `<Image>` erzeugt und ausgelöst, sobald dieses ausgewählt wurde.

#### 1.9.8.2 Beispiel `<Image>` aus statischer Ressource

Im folgenden Beispiel ist die Verwendung des `<Image>`-Tags gezeigt.

```
<DetailComponent name="ExampleImage" path="ExampleImage" displayName="Example Image">
  <FlowLayout>
    <Image value="image.jpg"/>
  </FlowLayout>
</DetailComponent>
```

#### 1.9.8.3 Beispiel `<Image>` mit Data Binding (dynamisch aus einem Prozess)

Im folgenden Beispiel ist die Verwendung des `<Image>`-Tags mit Data Binding gezeigt.



```
<DetailComponent path="ExampleImageWithDataBinding" name="ExampleImageWithDataBinding"
  displayName="Example Image with Data Binding" default="true" process="Image.wrf">
  <Properties>
    <Property name="image" type="Base64" />
  </Properties>
  <FlowLayout>
    <Image value="#image"/>
  </FlowLayout>
</DetailComponent>
```

Die Daten, die vom Prozess geliefert werden, müssen folgendes Format haben:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Ok>
  <image mediaType="image/jpeg">
    <!-- Base64String -->
  </image>
</Ok>
```

### 1.9.9 Datei-Link

FileLink-Steuerelemente werden verwendet, um Dateien für den Download einzubinden. Für diesen Download gibt es zwei Möglichkeiten. Zum einen kann die Datei, die heruntergeladen werden soll, als statische Ressource im Resources-Ordner liegen, zum anderen kann die Datei dynamisch über einen Prozess geladen werden.

Attribut	Beschreibung
data	<div>  Nur verwenden, wenn eine Base64-Datei zum Download angeboten wird! </div> <p>Gibt an, welche Datei heruntergeladen werden soll.</p> <p><b>Mögliche Werte:</b> Zeichenkette für Data Binding</p> <div>  Der Data-Binding-Ausdruck muss auf eine Property vom Typ base64 verweisen! </div>
displayName	<p>Name der Datei, die in der Web-Anwendung angezeigt wird und als Dateiname für die heruntergeladene Datei vergeben wird.</p> <ul style="list-style-type: none"> <li>• Übersetzbar</li> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette inkl. Dateiendung</p>

Attribut	Beschreibung
fileName	Identifikator für den Prozess, der die Daten liefert. <ul style="list-style-type: none"> <li>• Übersetzbar</li> <li>• Data Binding möglich</li> </ul> <b>Mögliche Werte:</b> Beliebige Zeichenkette
process	Prozess, der die Datei liefert. <b>Mögliche Werte:</b> Zeichenkette (URI)

### 1.9.9.1 Beispiel <FileLink> (Dateidownload aus statischer Ressource)

Im folgenden Beispiel ist die Verwendung des <FileLink>-Tags gezeigt. Dabei liegt die Datei 1.pdf, die zum Download bereitgestellt werden soll, direkt im Ordner Resources und wird über den Prozess process\_1.wrf im Ordner Services/Processes geliefert. Sowohl eine Datei als auch ein Prozess müssen für einen Download vorhanden sein.

#### Beispiel FileLink

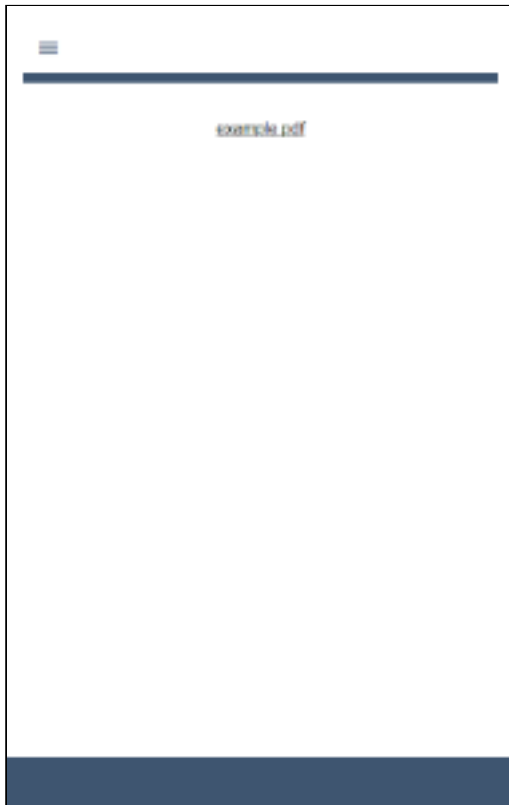
```
<WebApp xmlns="http://softproject.de/webapp/1.0" path="download"
  displayName="File download">
  <Modules>
    <Module path="Module" displayName="My Module">
      <Components>
        <DetailComponent path="Dashboard"
          displayName="Dashboard" default="true">
          <FlowLayout>
            <FileLink process="process_1.wrf" fileName="1.pdf"
displayName="example.pdf" />
          </FlowLayout>
        </DetailComponent>
      </Components>
    </Module>
  </Modules>
</WebApp>
```

In obiger Web App-Definition wird der Datei-Link definiert. Für FileLink wird ein Prozess hinterlegt, der die Datei liefert (process\_1.wrf). Es wird ein Anzeigenamen vergeben (displayName="example.pdf"), der als Text in der Web App und als Dateiname des Downloads verwendet wird. Das Attribut fileName ist in diesem Beispiel nicht relevant, dieses Attribut kann in komplexeren Anwendungen verwendet werden, da es dem verarbeitenden Prozess vom Steuerelement zurückgeliefert wird:

**Output FileLink**


```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<fileId>1.pdf</fileId>
```

Der Code aus dem Beispiel für FileLink erzeugt folgende Ansicht in der Web-Anwendung:




### 1.9.10 File Upload

Mit dem <FileUpload>-Steuerelement können Dateien in eine Web-Anwendung hochgeladen werden, die mit X4 Activities Web Apps erstellt wurde. Das <FileUpload>-Steuerelement kann nur in [Detail-Komponenten](#) oder [Detail-Strukturelementen](#) innerhalb von [Master/Detail-Komponenten](#) verwendet werden. Das <FileUpload>-Steuerelement verwendet den Standarddialog **Datei öffnen** des Browsers, in dem der Anwender die Datei auswählen kann, die hochgeladen werden soll. Nachdem der Nutzer die Datei ausgewählt hat, startet der Upload. Der Prozess, der die hochgeladene Datei entgegennimmt, muss eine Bestätigung oder eine Fehlermeldung zurückgeben. Wenn die Bestätigung eine Nachricht enthält, wird diese Nachricht als Pop-up-Fenster angezeigt. Pop-up-Fenster mit Fehlermeldungen werden in jedem Fall angezeigt.

 Bei Verwendung eines mobilen Endgeräts kann zusätzlich auf die integrierte Kamera zugegriffen werden.



Attribut	Beschreibung
acceptedFileTypes	<p>Dateitypen, die für den Upload akzeptiert werden.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> MIME-Typen, auch mit Wildcards (*), z. B.</p> <ul style="list-style-type: none"> <li>• text/*: Textdateien</li> <li>• image/*: Grafikdateien <ul style="list-style-type: none"> <li>• image/jpg: Nur JPG-Dateien</li> <li>• image/png: Nur PNG-Dateien</li> </ul> </li> <li>• video/*: Videodateien</li> <li>• audio/*: Audiodateien</li> <li>• application/*: Dateien, die an ein bestimmtes Programm gebunden sind</li> <li>• multipart/*: mehrteilige Daten</li> <li>• message/*: Nachrichten</li> <li>• model/*: Dateien, die mehrdimensionale Strukturen repräsentieren</li> </ul>
displayName	<p>Anzeigenname in der Oberfläche.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
fileId	<p>Identifikator, der vom Prozessentwickler verwendet werden kann.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
maxSizeMB	<p>Maximale Dateigröße in Mebibyte.</p> <p><b>Mögliche Werte:</b> Ganzzahl</p>
process	<p><i>Erforderlich.</i> Prozess, der die hochgeladene Datei entgegennimmt.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p> Der Prozess muss inklusive Dateiendung angegeben werden!</p> </div> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p>

### 1.9.10.1 Input-Format für den Prozess

Die Daten werden in einem bestimmten Input-Format an den Prozess übergeben. Das Format ist im Folgenden beschrieben.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<File>
  <FileId>....</FileId>
  <FileName>....</FileName>
  <FileType>....</FileType>
  <FileData>
    ....Base64 data....
  </FileData>
</File>
```

Die Daten im Element <FileData> können z. B. mit dem Base64 Converter dekodiert werden.

### 1.9.10.2 Beispiel <FileUpload>

Im folgenden Beispiel ist die Verwendung des <FileUpload>-Elements gezeigt.

```
<FileUpload
  fileId="id"
  process="process_1.wrf"
  maxSizeMB="1"
  acceptedFileTypes="image/*"
/>
```

Der obige Code führt zu folgendem Steuerelement:



## 1.9.11 Diagramme




In einer X4 Activities Web App können Diagramme („Chart Controls“) eingebunden werden.

Es können verschiedene Arten von Diagrammen eingebunden werden:

- Liniendiagramm
- Balkendiagramm
- Kreis-/Donutdiagramm
- Tacho

Alle Diagramme in einem Layout müssen innerhalb eines <Chart>-Elements definiert werden. Zusätzlich zu den [Standardattributen der Steuerelemente](#) ist folgendes Attribut verfügbar:

Attribut	Beschreibung
title	Titel des Diagramms.
	<b>Mögliche Werte:</b> Beliebige Zeichenkette oder Data Binding

Attribut	Beschreibung
height	<p>Höhe des Diagramms</p> <p><b>Mögliche Werte:</b> Ganzzahl</p> <div> <p> Wird das Attribut <code>height</code> nicht explizit gesetzt, werden Diagramme mit einer Höhe von 400 Pixeln angezeigt.</p> </div>
<div> <p> Die Diagramme sind auf Größenangaben in Pixeln optimiert. Wenn eine ausdrückliche Definition von Höhe und Breite des Diagramms benötigt wird, sollte zuvor das Attribut <code>units</code> auf <code>pixels</code> gesetzt werden.</p> </div>	
<div> <p> Alle Diagramme müssen per Data Binding befüllt werden!</p> </div>	

### 1.9.11.1 Liniendiagramm



Mit einem Liniendiagramm können Trends oder Veränderungen im Zeitablauf dargestellt werden. Dabei werden Datenpunkte durch gerade Liniensegmente verbunden. In einem Liniendiagramm können auch mehrere Zahlenreihen dargestellt werden.





Ein Liniendiagramm kann folgende Elemente enthalten:





- `<Line>`: Enthält alle Elemente, die zum Liniendiagramm gehören.
- `<Axis>`: Definiert die Achsenbeschriftungen.
- `<LineData>`: Definiert, wie die Daten angezeigt werden sollen. Für jeden Datensatz, also jede Linie im Liniendiagramm muss ein `<LineData>`-Element definiert werden.
- `<Tooltip>`: Definiert, wie der Tooltip für die einzelnen Werte im Diagramm angezeigt werden soll.

#### 1.9.11.1.1 Allgemeine Attribute

Folgende Attribute sind für alle Elemente eines Liniendiagramms verfügbar:

fontFamily	<p>Legt die Schriftfamilie fest.</p> <div data-bbox="481 340 1433 667">  <ul style="list-style-type: none"> <li>• Mit diesem Attribut wird die Standardschrift der Web App für dieses Steuerelement überschrieben.</li> <li>• Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut <code>fontFamily</code> nicht explizit gesetzt haben.</li> <li>• Dieses Attribut funktioniert nicht für die Elemente <code>Image</code> und <code>Map</code>.</li> </ul> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <code>MainFont</code>: hinterlegte Hauptschrift</li> <li>• Fontcode aus der Font-Palette, z. B. <code>Font04</code></li> </ul>
fontSize	<p>Legt die Schriftgröße fest.</p> <div data-bbox="481 913 1433 1115">  <ul style="list-style-type: none"> <li>• Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Steuerelement überschrieben.</li> <li>• Dieses Attribut funktioniert nicht für die Elemente <code>Image</code> und <code>Map</code>.</li> </ul> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code></li> <li>• Schriftgröße in Pixel, z. B. <code>20px</code></li> <li>• Schriftgröße in Punkt, z. B. <code>18pt</code></li> <li>• Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code></li> <li>• Schlüsselwörter: <code>xx-small</code>, <code>x-small</code>, <code>small</code>, <code>medium</code>, <code>large</code>, <code>x-large</code>, <code>xx-large</code>, <code>smaller</code>, <code>larger</code></li> </ul>

fontStretch	<p>Legt die Breite der einzelnen Schriftzeichen fest.</p> <div><p> Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Steuerelement überschrieben.</p></div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"><li>• Condensed</li><li>• Expanded</li><li>• ExtraCondensed</li><li>• ExtraExpanded</li><li>• Medium</li><li>• Normal (Standard)</li><li>• SemiCondensed</li><li>• SemiExpanded</li><li>• UltraCondensed</li><li>• UltraExpanded</li></ul> <div><p> Dieses Attribut funktioniert nicht für die Elemente Image, Map und Chart.</p></div>
fontStyle	<p>Legt die Neigung der Schrift fest.</p> <div><p> Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Steuerelement überschrieben.</p></div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"><li>• <i>italic</i>: kursive Schrift</li><li>• <i>normal</i>: normale Schrift (Standard)</li><li>• <i>oblique</i>: schräg gestellter Schriftstil (berechnet)</li></ul> <div><p> Dieses Attribut funktioniert nicht für die Elemente Image und Map.</p></div>

fontWeight	<p>Legt die Schriftstärke fest.</p> <div data-bbox="483 338 1434 450"> <p> Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Steuerelement überschrieben.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Black</li> <li>• Bold</li> <li>• DemiBold</li> <li>• ExtraBlack</li> <li>• ExtraBold</li> <li>• ExtraLight</li> <li>• Heavy</li> <li>• Light</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• Regular</li> <li>• SemiBold</li> <li>• Thin</li> <li>• UltraBlack</li> <li>• UltraBold</li> <li>• UltraLight</li> </ul> <div data-bbox="483 1126 1434 1227"> <p> Dieses Attribut funktioniert nicht für die Elemente Image und Map.</p> </div>
foreground	<p>Definiert die Schriftfarbe.</p> <div data-bbox="483 1312 1434 1440"> <p> Diese Einstellung überschreibt die Standardfarbe des Farbschemas!</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div data-bbox="525 1574 1434 1686"> <p> Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts!</p> </div> <ul style="list-style-type: none"> <li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>

#### 1.9.11.1.2 Line

<Line>: Enthält alle Elemente, die zum Liniendiagramm gehören.


Zusätzlich zu den allgemeinen Attributen kann <Line> folgende Attribute haben:

Attribut	Beschreibung
legend	Gibt an, ob eine Legende angezeigt werden soll.  <b>Mögliche Werte:</b> true(Standard)/ false

### 1.9.11.1.3 Axis

<Axis>: Kindelement von <Line>. Definiert die Achsenbeschriftungen.

Zusätzlich zu den allgemeinen Attributen kann <Axis> folgende Attribute haben:


Attribut	Beschreibung
color	Legt die Farbe der Achsen fest.  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div style="border: 1px solid #f0e68c; padding: 5px; margin: 5px 0;">  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts!         </div> <ul style="list-style-type: none"> <li>Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>
horizontalLabel	Definiert die Beschriftung der x-Achse.  <b>Mögliche Werte:</b> Beliebige Zeichenkette
horizontalMax	Maximalwert der x-Achse  <b>Mögliche Werte:</b> Ganzzahl
horizontalMin	Minimalwert der x-Achse  <b>Mögliche Werte:</b> Ganzzahl
horizontalUnits	Einheit der x-Achse  <b>Mögliche Werte:</b> Ganzzahl
verticalLabel	Definiert die Beschriftung der y-Achse.  <b>Mögliche Werte:</b> Beliebige Zeichenkette
verticalMax	Maximalwert der y-Achse  <b>Mögliche Werte:</b> Ganzzahl
verticalMin	Minimalwert der y-Achse  <b>Mögliche Werte:</b> Ganzzahl

Attribut	Beschreibung
verticalUnits	Einheit der y-Achse  <b>Mögliche Werte:</b> Ganzzahl

#### 1.9.11.1.4 LineData

<LineData>: Kindelement von <Line>. Definiert, wie die Daten angezeigt werden sollen. Für jeden Datensatz, also jede Linie im Liniendiagramm muss ein <LineData>-Element definiert werden.

Zusätzlich zu den allgemeinen Attributen kann <LineData> folgende Attribute haben:

Attribut	Beschreibung
color	Definiert die Farbe der Linie.  <b>Mögliche Werte:</b> Hexadezimaler Farbwert oder Ausdruck für Data Binding (Color)  <div>  Wird das Attribut color nicht definiert, werden die im <a href="#">Theming Editor</a> definierten Farben verwendet. </div>
data	Gibt an, welche Daten im Diagramm dargestellt werden.  <b>Mögliche Werte:</b> Data Binding
labels	Gibt an, ob die Datenpunkte mit dem jeweiligen Wert beschriftet werden.  <b>Mögliche Werte:</b> true / false (Standard)
marker	Gibt an, wie ein Datenpunkt dargestellt wird.  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• circle: Kreis</li> <li>• rect: Rechteck</li> <li>• roundRect: Rechteck mit abgerundeten Ecken</li> <li>• triangle: Dreieck</li> <li>• diamond: Raute</li> <li>• pin: Stecknadel</li> <li>• arrow: Pfeil</li> <li>• none: kein Marker</li> </ul>
markersEnabled	Gibt an, ob ein Datenpunkt markiert werden soll.  <b>Mögliche Werte:</b> true / false
name	Name des Datensatzes, der in der Legende angezeigt wird.  <b>Mögliche Werte:</b> Beliebige Zeichenkette



Attribut	Beschreibung
xValues	Gibt an, welche Daten im Diagramm auf der x-Achse dargestellt werden. Die angegebene Zeichenkette bezieht sich auf eine definierte Property innerhalb der Daten, die für data angegeben werden.  <b>Mögliche Werte:</b> Data Binding
yValues	Gibt an, welche Daten im Diagramm auf der y-Achse dargestellt werden. Die angegebene Zeichenkette bezieht sich auf eine definierte Property innerhalb der Daten, die für data angegeben werden.  <b>Mögliche Werte:</b> Data Binding

#### 1.9.11.1.5 Tooltip

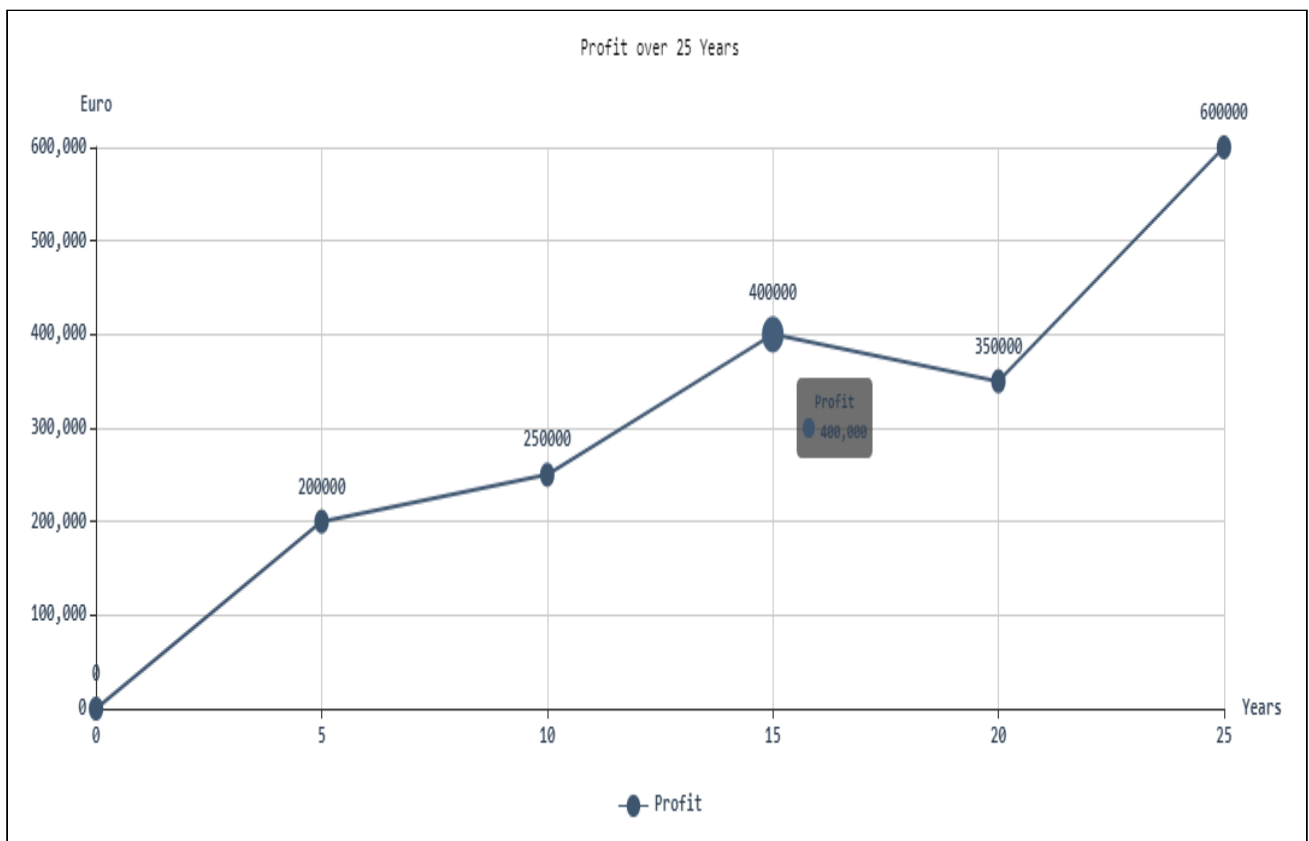
<Tooltip>: Kindelement von <Line>. Definiert, wie der Tooltip für die einzelnen Werte im Diagramm angezeigt werden soll.

Für <Tooltip> lassen sich die allgemeinen Attribute (s.o.) definieren.

#### 1.9.11.1.6 Beispiele

##### Beispiel 1:

```
<Properties>
  <Property name="DataSource" type="Complex">
    <Property name="Profit" type="List">
      <Property name="Years" type="Integer"/>
      <Property name="Money" type="Integer"/>
    </Property>
  </Property>
</Properties>
<FlowLayout>
  <Chart fontFamily="Font01" title="Profit over 25 Years">
    <Line fontSize="" fontStyle="normal" fontFamily="Font02" fontWeight="Black"
foreground="A700" legend="true">
      <Tooltip fontFamily="Font01" fontSize="30"/>
      <Axis fontFamily="Font03" horizontalLabel="Years" verticalLabel="Euro"/>
      <LineData data="#DataSource.Profit" labels="true" name="Profit" xValues="
#Years" yValues="#Money"/>
    </Line>
  </Chart>
</FlowLayout>
```



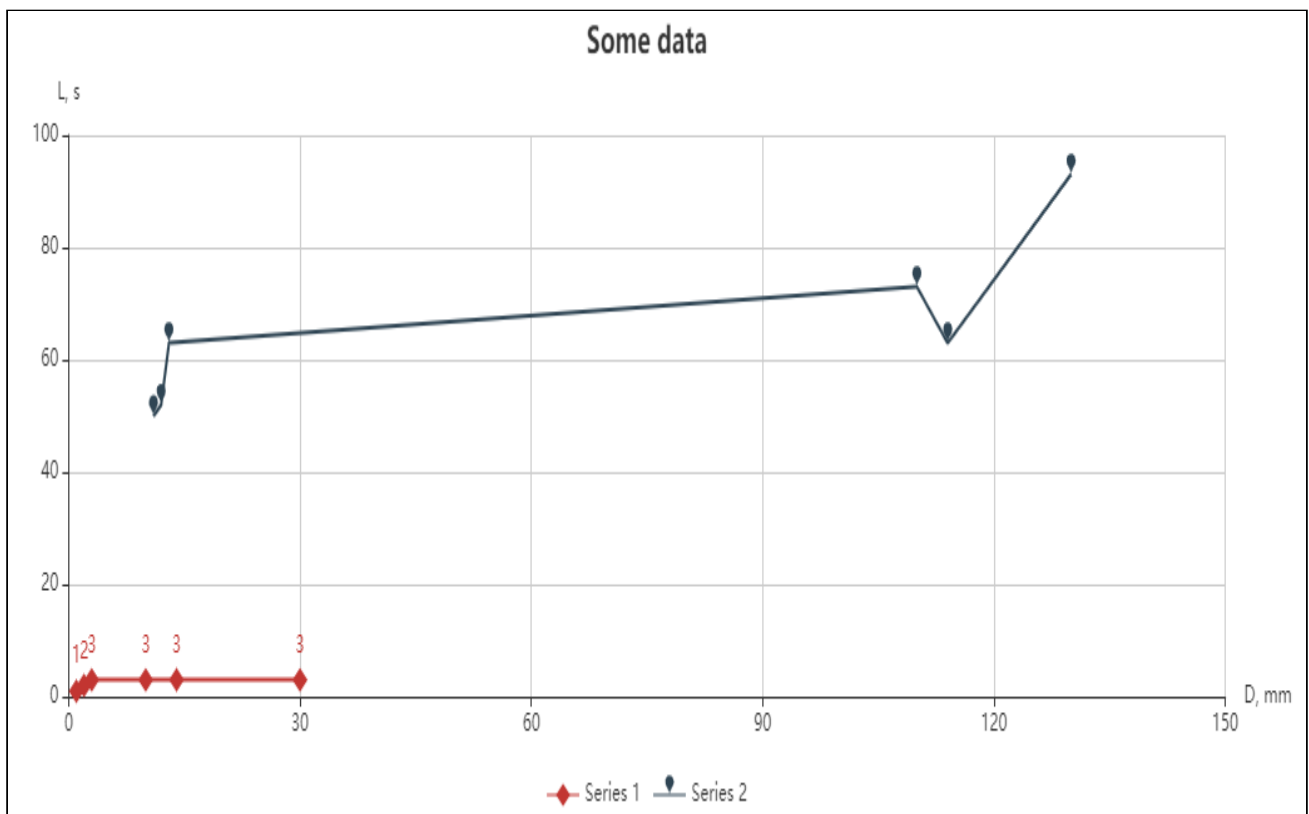
### Beispiel 2

```

<Property name="Datasource" type="Complex">
  <Property name="LineDataSet1" type="List">
    <Property name="xValue" type="Integer"/>
    <Property name="yValue" type="String"/>
  </Property>
  <Property name="LineDataSet2" type="List">
    <Property name="xValue" type="Integer"/>
    <Property name="yValue" type="String"/>
  </Property>
</Property>
...
<Chart title="Some data">
  <Line>
    <Axis horizontalLabel="D, mm" verticalLabel="L, s" />
    <LineData data="#Datasource.LineDataSet1" xValues="#xValue"
      yValues="#yValue" name="Series 1" marker="diamond" labels="true" />
    <LineData data="#Datasource.LineDataSet2" xValues="#xValue"
      yValues="#yValue" name="Series 2" marker="pin" />
  </Line>
</Chart>

```

Um eine Linie im Diagramm darzustellen, wird das Element `LineData` verwendet. Die dargestellten Daten selbst sind in der Property `Datasource` enthalten. Im obigen Beispiel werden die Daten über den Data-Binding-Ausdruck `#Datasource.LineDataSet1` abgerufen. Das obige Beispiel führt zu folgendem Diagramm:



### 1.9.11.2 Balkendiagramm



Mit einem Balkendiagramm können mehrere Variablen mit einem einzelnen Wert verglichen werden. Sie sind hilfreich, um Kategorien zu vergleichen.





Ein Balkendiagramm kann folgende Elemente enthalten:





- `<Bar>`: Enthält alle Elemente, die zum Balkendiagramm gehören. Gibt an, wie das Diagramm angezeigt werden soll.
- `<BarData>`: Definiert, wie die Daten angezeigt werden sollen. Für jede Kategorie im Balkendiagramm muss ein `<BarData>`-Element definiert werden.
- `<Axis>`: Definiert die Achsenbeschriftungen.
- `<Tooltip>`: Definiert, wie der Tooltip für die einzelnen Werte im Diagramm angezeigt werden soll.

#### 1.9.11.2.1 Allgemeine Attribute

Folgende Attribute sind für alle Elemente eines Balkendiagramms verfügbar:

fontFamily	<p>Legt die Schriftfamilie fest.</p> <div data-bbox="481 340 1434 669"> <p> • Mit diesem Attribut wird die Standardschrift der Web App für dieses Steuerelement überschrieben.</p> <p>• Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut <code>fontFamily</code> nicht explizit gesetzt haben.</p> <p>• Dieses Attribut funktioniert nicht für die Elemente <code>Image</code> und <code>Map</code>.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <code>MainFont</code>: hinterlegte Hauptschrift</li> <li>• Fontcode aus der Font-Palette, z. B. <code>Font04</code></li> </ul>
fontSize	<p>Legt die Schriftgröße fest.</p> <div data-bbox="481 913 1434 1117"> <p> • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Steuerelement überschrieben.</p> <p>• Dieses Attribut funktioniert nicht für die Elemente <code>Image</code> und <code>Map</code>.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code></li> <li>• Schriftgröße in Pixel, z. B. <code>20px</code></li> <li>• Schriftgröße in Punkt, z. B. <code>18pt</code></li> <li>• Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code></li> <li>• Schlüsselwörter: <code>xx-small</code>, <code>x-small</code>, <code>small</code>, <code>medium</code>, <code>large</code>, <code>x-large</code>, <code>xx-large</code>, <code>smaller</code>, <code>larger</code></li> </ul>

fontStretch	<p>Legt die Breite der einzelnen Schriftzeichen fest.</p> <div><p> Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Steuerelement überschrieben.</p></div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"><li>• Condensed</li><li>• Expanded</li><li>• ExtraCondensed</li><li>• ExtraExpanded</li><li>• Medium</li><li>• Normal (Standard)</li><li>• SemiCondensed</li><li>• SemiExpanded</li><li>• UltraCondensed</li><li>• UltraExpanded</li></ul> <div><p> Dieses Attribut funktioniert nicht für die Elemente Image, Map und Chart.</p></div>
fontStyle	<p>Legt die Neigung der Schrift fest.</p> <div><p> Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Steuerelement überschrieben.</p></div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"><li>• <i>italic</i>: kursive Schrift</li><li>• <i>normal</i>: normale Schrift (Standard)</li><li>• <i>oblique</i>: schräg gestellter Schriftstil (berechnet)</li></ul> <div><p> Dieses Attribut funktioniert nicht für die Elemente Image und Map.</p></div>

fontWeight	<p>Legt die Schriftstärke fest.</p> <div data-bbox="483 338 1434 450"> <p> Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Steuerelement überschrieben.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Black</li> <li>• Bold</li> <li>• DemiBold</li> <li>• ExtraBlack</li> <li>• ExtraBold</li> <li>• ExtraLight</li> <li>• Heavy</li> <li>• Light</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• Regular</li> <li>• SemiBold</li> <li>• Thin</li> <li>• UltraBlack</li> <li>• UltraBold</li> <li>• UltraLight</li> </ul> <div data-bbox="483 1126 1434 1227"> <p> Dieses Attribut funktioniert nicht für die Elemente Image und Map.</p> </div>
foreground	<p>Definiert die Schriftfarbe.</p> <div data-bbox="483 1312 1434 1440"> <p> Diese Einstellung überschreibt die Standardfarbe des Farbschemas!</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div data-bbox="525 1574 1434 1686"> <p> Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts!</p> </div> <ul style="list-style-type: none"> <li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>

### 1.9.11.2.2 Bar

<Bar>: Enthält alle Elemente, die zum Balkendiagramm gehören. Gibt an, wie das Diagramm angezeigt werden soll.


Zusätzlich zu den allgemeinen Attributen kann <Bar> folgende Attribute haben:

Attribut	Beschreibung
legend	Gibt an, ob eine Legende angezeigt werden soll.  <b>Mögliche Werte:</b> true(Standard) / false
stacked	Gibt an, ob das Balkendiagramm als gestapeltes Diagramm angezeigt werden soll.  <b>Mögliche Werte:</b> true / false(Standard)

#### 1.9.11.2.3 BarData

<BarData>: Kindelement von <Bar>. Definiert, wie die Daten angezeigt werden sollen. Für jede Kategorie im Balkendiagramm muss ein <BarData>-Element definiert werden.


Zusätzlich zu den allgemeinen Attributen kann <BarData> folgende Attribute haben:

Attribut	Beschreibung
category	Gibt an, welcher Kategorie (category) der Wert zugeordnet wird.  <b>Mögliche Werte:</b> Data Binding
color	Farbe des Balkens  <b>Mögliche Werte:</b> Hexadezimaler Farbwert oder Ausdruck für Data Binding (Color)  <div> Wird das Attribut color nicht definiert, werden die im <a href="#">Theming Editor</a> definierten Farben verwendet.</div>
data	Gibt an, welches Datenobjekt der Datenquelle visualisiert wird.  <b>Mögliche Werte:</b> Data Binding
name	Gibt den Namen der Daten an, der in der Legende angezeigt wird.  <b>Mögliche Werte:</b> Beliebige Zeichenkette oder Data Binding
value	Gibt an, welcher Wert des Datenobjekts visualisiert wird.  <b>Mögliche Werte:</b> Data Binding

#### 1.9.11.2.4 Axis

<Axis>: Kindelement von <Bar>. Definiert die Achsenbeschriftungen.

Zusätzlich zu den allgemeinen Attributen kann <Axis> folgende Attribute haben:

Attribut	Beschreibung
color	<p>Legt die Farbe der Achsen fest.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div style="border: 1px solid #fde725; padding: 5px; margin: 5px 0;">  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! </div> <ul style="list-style-type: none"> <li>Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>

### 1.9.11.2.5 Tooltip

<Tooltip>: Kindelement von <Bar>. Definiert, wie der Tooltip für die einzelnen Werte im Diagramm angezeigt werden soll.

Für <Tooltip> lassen sich die allgemeinen Attribute (s.o.) definieren.

### 1.9.11.2.6 Beispiele

#### Beispiel 1:

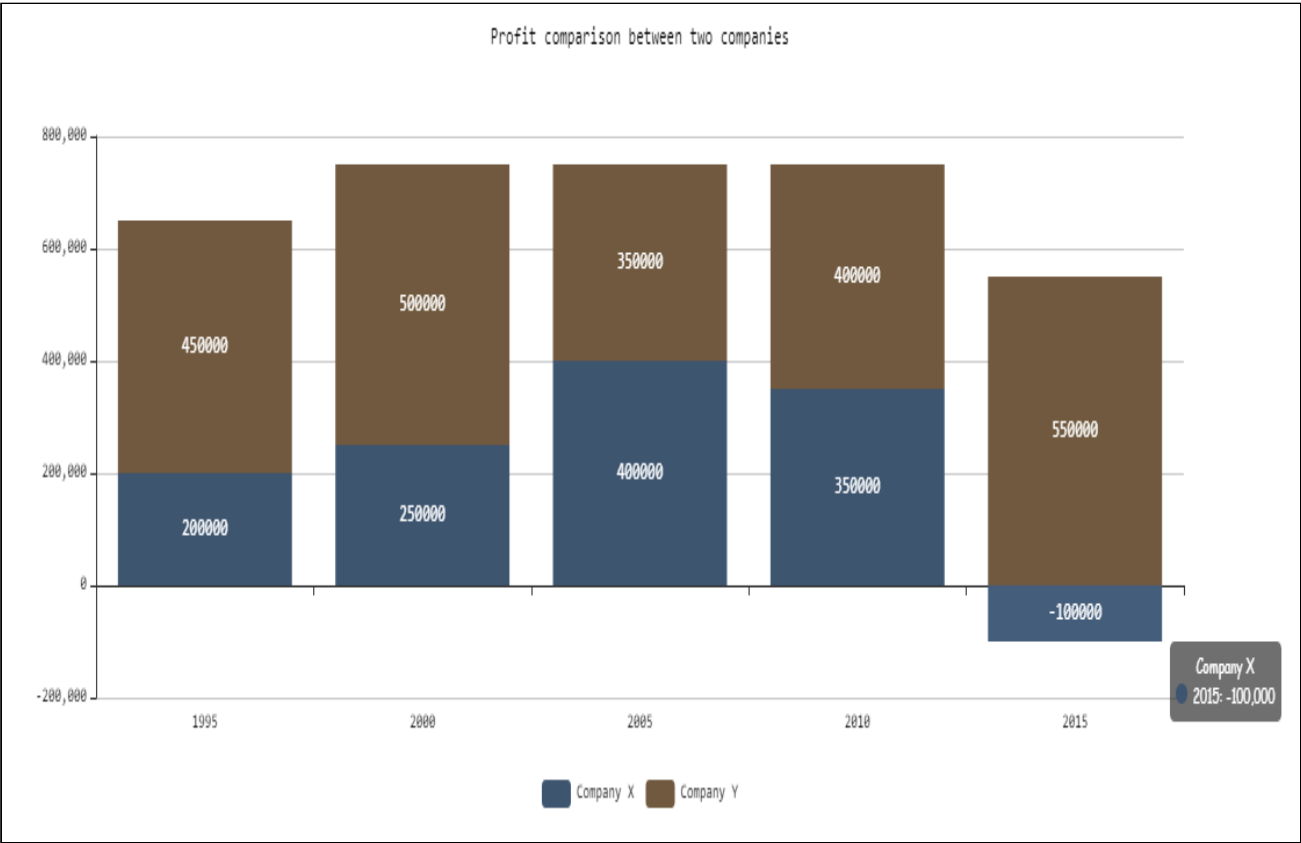
```

<Properties>
  <Property name="DataSource" type="Complex">
    <Property name="CompanyX" type="List">
      <Property name="Category" type="String"/>
      <Property name="Money" type="Integer"/>
    </Property>
    <Property name="CompanyY" type="List">
      <Property name="Category" type="String"/>
      <Property name="Money" type="Integer"/>
    </Property>
  </Property>
</Properties>
<FlowLayout>
  <Chart fontFamily="Font03" title="Profit comparison between two companies">
    <Bar stacked="true">
      <Tooltip fontFamily="Font02" fontSize="10"/>
      <Axis fontFamily="Font03" fontSize="10"/>
      <BarData category="#Category" data="#DataSource.CompanyX" fontFamily="Font03" name="Company X" value="#Money"/>
      <BarData category="#Category" data="#DataSource.CompanyY" fontFamily="Font03" name="Company Y" value="#Money"/>
    </Bar>
  </Chart>
</FlowLayout>

```

Das obige Beispiel führt zu folgendem Diagramm:





**Beispiel 2:**

```

<Property name="Datasource" type="Complex">
  <Property name="Bar1" type="List">
    <Property name="category" type="String"></Property>
    <Property name="value" type="Integer"></Property>
  </Property>
  <Property name="Bar2" type="List">
    <Property name="category" type="String"></Property>
    <Property name="value" type="Integer"></Property>
  </Property>
  <Property name="Bar3" type="List">
    <Property name="category" type="String"></Property>
    <Property name="value" type="Integer"></Property>
  </Property>
  ...
  ...
  ...
</Property>

<Chart title="Test Bar">
  <Bar>
    <BarData data="#Datasource.Bar1" category="#category" value="#value"
      name="Data 1" />
    <BarData data="#Datasource.Bar2" category="#category" value="#value"
      name="Data 2" />
    <BarData data="#Datasource.Bar3" category="#category" value="#value"
      name="Data 3" />
  </Bar>
</Chart>

<Chart title="Stacked Bar">
  <Bar stacked="true">
    <BarData data="#Datasource.Bar1" category="#category" value="#value"
      name="Data 1" />
    <BarData data="#Datasource.Bar2" category="#category" value="#value"
      name="Data 2" />
    <BarData data="#Datasource.Bar3" category="#category" value="#value"
      name="Data 3" />
  </Bar>
</Chart>

```

Um eine Kategorie im Diagramm darzustellen, wird das Element `BarData` verwendet. Die dargestellten Daten selbst sind in der Property `Datasource` enthalten. Das obige Beispiel führt zu folgenden Diagrammen:



### 1.9.11.3 Kreis-/Donutdiagramm



Mit einem Kreisdiagramm kann der Anteil einzelner Daten an der Gesamtheit veranschaulicht werden.





Ein Kreis-/Donutdiagramm kann folgende Elemente enthalten:





- `<Pie>`: Gibt an, wie das Diagramm angezeigt werden soll.
- `<Tooltip>`: Definiert, wie der Tooltip für die einzelnen Werte im Diagramm angezeigt werden soll.

#### 1.9.11.3.1 Allgemeine Attribute

Folgende Attribute sind für alle Elemente eines Kreis-/Donutdiagramms verfügbar:

fontFamily	<p>Legt die Schriftfamilie fest.</p> <div data-bbox="481 340 1433 667"> <p> • Mit diesem Attribut wird die Standardschrift der Web App für dieses Steuerelement überschrieben.</p> <p>• Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut <code>fontFamily</code> nicht explizit gesetzt haben.</p> <p>• Dieses Attribut funktioniert nicht für die Elemente <code>Image</code> und <code>Map</code>.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <code>MainFont</code>: hinterlegte Hauptschrift</li> <li>• Fontcode aus der Font-Palette, z. B. <code>Font04</code></li> </ul>
fontSize	<p>Legt die Schriftgröße fest.</p> <div data-bbox="481 913 1433 1115"> <p> • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Steuerelement überschrieben.</p> <p>• Dieses Attribut funktioniert nicht für die Elemente <code>Image</code> und <code>Map</code>.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code></li> <li>• Schriftgröße in Pixel, z. B. <code>20px</code></li> <li>• Schriftgröße in Punkt, z. B. <code>18pt</code></li> <li>• Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code></li> <li>• Schlüsselwörter: <code>xx-small</code>, <code>x-small</code>, <code>small</code>, <code>medium</code>, <code>large</code>, <code>x-large</code>, <code>xx-large</code>, <code>smaller</code>, <code>larger</code></li> </ul>

fontStretch	<p>Legt die Breite der einzelnen Schriftzeichen fest.</p> <div><p> Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Steuerelement überschrieben.</p></div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"><li>• Condensed</li><li>• Expanded</li><li>• ExtraCondensed</li><li>• ExtraExpanded</li><li>• Medium</li><li>• Normal (Standard)</li><li>• SemiCondensed</li><li>• SemiExpanded</li><li>• UltraCondensed</li><li>• UltraExpanded</li></ul> <div><p> Dieses Attribut funktioniert nicht für die Elemente Image, Map und Chart.</p></div>
fontStyle	<p>Legt die Neigung der Schrift fest.</p> <div><p> Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Steuerelement überschrieben.</p></div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"><li>• <i>italic</i>: kursive Schrift</li><li>• <i>normal</i>: normale Schrift (Standard)</li><li>• <i>oblique</i>: schräg gestellter Schriftstil (berechnet)</li></ul> <div><p> Dieses Attribut funktioniert nicht für die Elemente Image und Map.</p></div>

fontWeight	<p>Legt die Schriftstärke fest.</p> <div data-bbox="481 338 1433 450"> <p> Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Steuerelement überschrieben.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Black</li> <li>• Bold</li> <li>• DemiBold</li> <li>• ExtraBlack</li> <li>• ExtraBold</li> <li>• ExtraLight</li> <li>• Heavy</li> <li>• Light</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• Regular</li> <li>• SemiBold</li> <li>• Thin</li> <li>• UltraBlack</li> <li>• UltraBold</li> <li>• UltraLight</li> </ul> <div data-bbox="481 1126 1433 1227"> <p> Dieses Attribut funktioniert nicht für die Elemente Image und Map.</p> </div>
foreground	<p>Definiert die Schriftfarbe.</p> <div data-bbox="481 1312 1433 1440"> <p> Diese Einstellung überschreibt die Standardfarbe des Farbschemas!</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div data-bbox="525 1574 1433 1686"> <p> Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts!</p> </div> <ul style="list-style-type: none"> <li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>

### 1.9.11.3.2 Pie

<Pie>: Gibt an, wie das Diagramm angezeigt werden soll.

Zusätzlich zu den allgemeinen Attributen kann <Pie> folgende Attribute haben:

Attribut	Beschreibung
color	<p>Farbe des einzelnen Diagrammausschnitts</p> <p><b>Mögliche Werte:</b> Hexadezimaler Farbwert oder Ausdruck für Data Binding (Color)</p> <p> Wird das Attribut <code>color</code> nicht definiert, werden die im <a href="#">Theming Editor</a> definierten Farben verwendet.</p>
data	<p>Gibt an, welche Daten im Diagramm dargestellt werden. Die angegebene Zeichenkette bezieht sich auf eine definierte Property innerhalb der Datenquelle, die im <code>&lt;Chart&gt;</code>-Element angegeben wird.</p> <p><b>Mögliche Werte:</b> Data Binding</p>
donut	<p>Gibt an, ob das Diagramm als Donutdiagramm dargestellt werden soll.</p> <p><b>Mögliche Werte:</b> <code>true</code> / <code>false</code> (Standard)</p>
legend	<p>Gibt an, ob eine Legende angezeigt werden soll.</p> <p><b>Mögliche Werte:</b> <code>true</code> (Standard) / <code>false</code></p>
name	<p>Gibt den Namen an, der dem jeweiligen Wert zugeordnet ist (Datenbeschriftung). Die angegebene Zeichenkette bezieht sich auf eine definierte Property innerhalb der Daten, die für <code>data</code> angegeben werden.</p> <p><b>Mögliche Werte:</b> Data Binding</p>
normals	<p>Gibt an, ob die Werte direkt im Diagramm beschriftet werden sollen.</p> <p><b>Mögliche Werte:</b> <code>true</code> / <code>false</code> (Standard)</p>
value	<p>Gibt an, welcher Wert des Datenobjekts visualisiert wird. Die angegebene Zeichenkette bezieht sich auf eine definierte Property innerhalb der Daten, die für <code>data</code> angegeben werden.</p> <p><b>Mögliche Werte:</b> Data Binding</p>

### 1.9.11.3.3 Tooltip

`<Tooltip>`: Kindelement von `<Pie>`. Definiert, wie der Tooltip für die einzelnen Werte im Diagramm angezeigt werden soll.

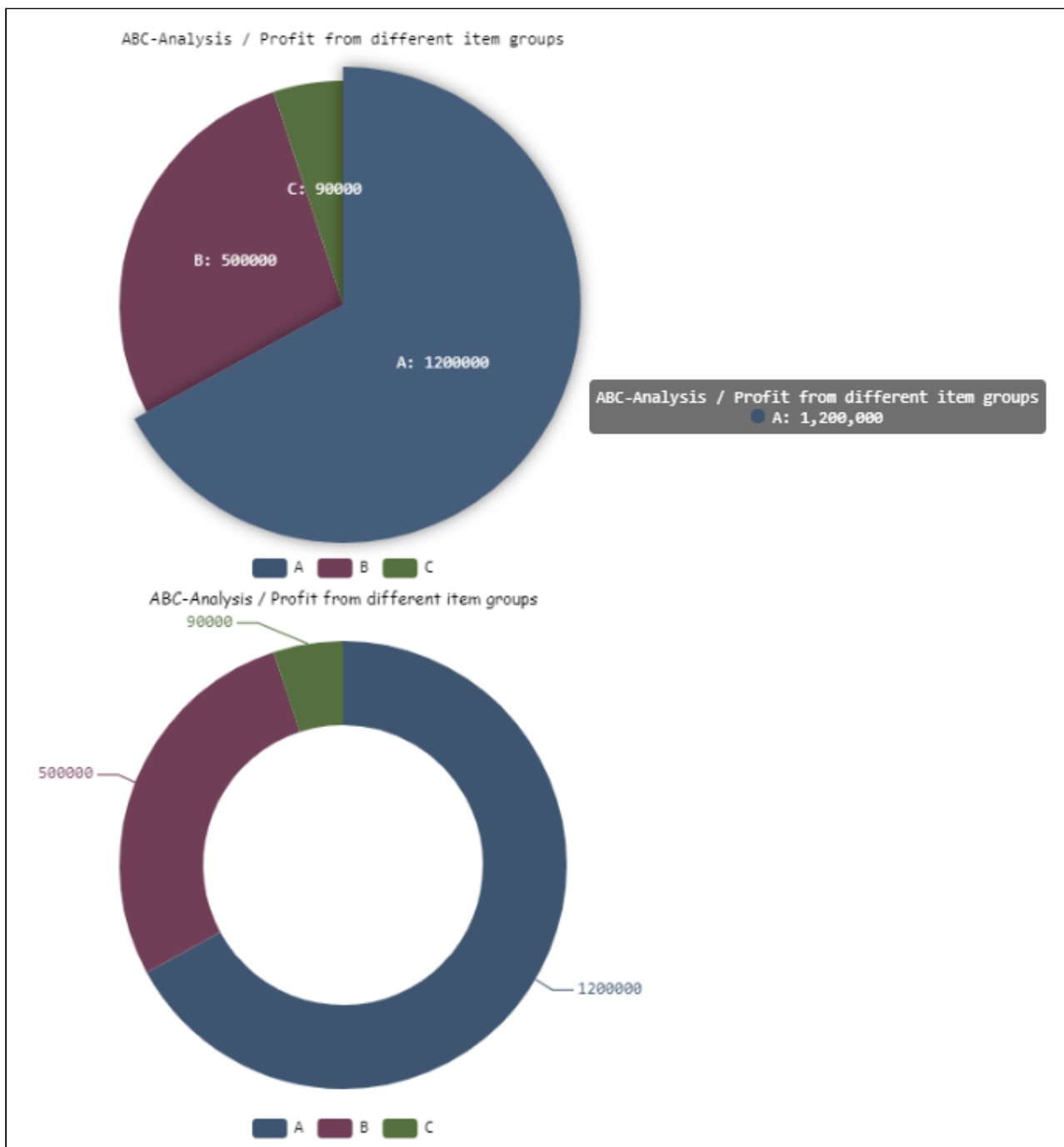
Für `<Tooltip>` lassen sich die allgemeinen Attribute (s.o.) definieren.

### 1.9.11.3.4 Beispiel

```
<Properties>
  <Property name="DataSource" type="Complex">
    <Property name="ABC-Analysis" type="List">
      <Property name="ItemGroup" type="String"/>
      <Property name="Profit" type="Integer"/>
    </Property>
  </Property>
</Properties>
<FlowLayout>
  <Chart fontFamily="Font03" title="ABC-Analysis / Profit from different item
groups">
    <Pie data="#DataSource.ABC-Analysis" fontFamily="Font03" name="#ItemGroup"
normals="false" value="#Profit">
      <Tooltip fontFamily="Font03"/>
    </Pie>
  </Chart>
  <Chart fontFamily="Font02" title="ABC-Analysis / Profit from different item
groups">
    <Pie data="#DataSource.ABC-Analysis" donut="true" fontFamily="Font03" name="#
ItemGroup" normals="true" value="#Profit">
      <Tooltip fontFamily="Font03"/>
    </Pie>
  </Chart>
</FlowLayout>
```

Das obige Beispiel führt zu folgenden Diagrammen:





#### 1.9.11.4 Tachodiagramm



Mit einem Tachodiagramm können Zustände mit unterschiedlichen Bewertungen dargestellt werden. Es ist gut geeignet, um Soll-/Ist-Werte zu vergleichen, z. B. bei Kennzahlen, Kundenzufriedenheits- oder Qualitätsmessungen.





Ein Tachodiagramm kann folgende Elemente enthalten:





- <Gauge>: Gibt an, wie das Diagramm angezeigt werden soll.
- <Tooltip>: Kindelement von <Gauge>. Definiert, wie der Tooltip für die einzelnen Werte im Diagramm angezeigt werden soll.

### 1.9.11.4.1 Allgemeine Attribute

Folgende Attribute sind für alle Elemente eines Tachodiagramms verfügbar:

fontFamily	<p>Legt die Schriftfamilie fest.</p> <div data-bbox="483 483 1436 813"> <p> • Mit diesem Attribut wird die Standardschrift der Web App für dieses Steuerelement überschrieben.</p> <p>• Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut <code>fontFamily</code> nicht explizit gesetzt haben.</p> <p>• Dieses Attribut funktioniert nicht für die Elemente <code>Image</code> und <code>Map</code>.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <code>MainFont</code>: hinterlegte Hauptschrift</li> <li>• Fontcode aus der Font-Palette, z. B. <code>Font04</code></li> </ul>
fontSize	<p>Legt die Schriftgröße fest.</p> <div data-bbox="483 1059 1436 1261"> <p> • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Steuerelement überschrieben.</p> <p>• Dieses Attribut funktioniert nicht für die Elemente <code>Image</code> und <code>Map</code>.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code></li> <li>• Schriftgröße in Pixel, z. B. <code>20px</code></li> <li>• Schriftgröße in Punkt, z. B. <code>18pt</code></li> <li>• Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code></li> <li>• Schlüsselwörter: <code>xx-small</code>, <code>x-small</code>, <code>small</code>, <code>medium</code>, <code>large</code>, <code>x-large</code>, <code>xx-large</code>, <code>smaller</code>, <code>larger</code></li> </ul>

fontStretch	<p>Legt die Breite der einzelnen Schriftzeichen fest.</p> <div><p> Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Steuerelement überschrieben.</p></div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"><li>• Condensed</li><li>• Expanded</li><li>• ExtraCondensed</li><li>• ExtraExpanded</li><li>• Medium</li><li>• Normal (Standard)</li><li>• SemiCondensed</li><li>• SemiExpanded</li><li>• UltraCondensed</li><li>• UltraExpanded</li></ul> <div><p> Dieses Attribut funktioniert nicht für die Elemente Image, Map und Chart.</p></div>
fontStyle	<p>Legt die Neigung der Schrift fest.</p> <div><p> Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Steuerelement überschrieben.</p></div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"><li>• <i>italic</i>: kursive Schrift</li><li>• <i>normal</i>: normale Schrift (Standard)</li><li>• <i>oblique</i>: schräg gestellter Schriftstil (berechnet)</li></ul> <div><p> Dieses Attribut funktioniert nicht für die Elemente Image und Map.</p></div>

fontWeight	<p>Legt die Schriftstärke fest.</p> <div data-bbox="483 338 1434 450"> <p> Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Steuerelement überschrieben.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Black</li> <li>• Bold</li> <li>• DemiBold</li> <li>• ExtraBlack</li> <li>• ExtraBold</li> <li>• ExtraLight</li> <li>• Heavy</li> <li>• Light</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• Regular</li> <li>• SemiBold</li> <li>• Thin</li> <li>• UltraBlack</li> <li>• UltraBold</li> <li>• UltraLight</li> </ul> <div data-bbox="483 1126 1434 1227"> <p> Dieses Attribut funktioniert nicht für die Elemente Image und Map.</p> </div>
foreground	<p>Definiert die Schriftfarbe.</p> <div data-bbox="483 1312 1434 1440"> <p> Diese Einstellung überschreibt die Standardfarbe des Farbschemas!</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div data-bbox="525 1574 1434 1686"> <p> Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts!</p> </div> <ul style="list-style-type: none"> <li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>

#### 1.9.11.4.2 Gauge

<Gauge>: Gibt an, wie das Diagramm angezeigt werden soll.

Zusätzlich zu den allgemeinen Attributen kann <Gauge> folgende Attribute haben:

Attribut	Beschreibung
axisMax	<p>Gibt den Endbereich des Tachodiagramms in Prozent an. Die Farbe des Bereichs wird im <a href="#">Theming Editor</a> über die Farbeigenschaft GaugeAxisMax definiert.</p> <p><b>Mögliche Werte:</b> Ganzzahl zwischen 0 und 100 (Standard: 100)</p> <div> <p> <b>Beispiel</b></p> <p>Die Standardwerte (axisMin="20", axisMiddle="80", axisMax="100") definieren ein Tachodiagramm, bei dem die Bereiche von 0 bis 20 %, 20 bis 80 % und 80 bis 100 % farblich unterschiedlich dargestellt werden.</p> </div>
axisMiddle	<p>Gibt den mittleren Bereich des Tachodiagramms in Prozent an. Die Farbe des Bereichs wird im <a href="#">Theming Editor</a> über die Farbeigenschaft GaugeAxisMiddle definiert.</p> <p><b>Mögliche Werte:</b> Ganzzahl zwischen 0 und 100 (Standard: 80)</p>
axisMin	<p>Gibt den Anfangsbereich des Tachodiagramms in Prozent an. Die Farbe des Bereichs wird im <a href="#">Theming Editor</a> über die Farbeigenschaft GaugeAxisMin definiert.</p> <p><b>Mögliche Werte:</b> Ganzzahl zwischen 0 und 100 (Standard: 20)</p>
endAngle	<p>Gibt den Endpunkt des Tachodiagramms an.</p> <p><b>Mögliche Werte:</b> Ganzzahl zwischen -360 und 360 (Standard: -45)</p> <div> <p> endAngle muss immer kleiner sein als startAngle.</p> </div>
legend	<p>Gibt an, ob eine Legende angezeigt werden soll.</p> <p><b>Mögliche Werte:</b> true (Standard) / false</p>
max	<p>Gibt den Maximalwert des Tachodiagramms an.</p> <p><b>Mögliche Werte:</b> Ganzzahl (Standard: 100)</p>
min	<p>Gibt den Minimalwert des Tachodiagramms an.</p> <p><b>Mögliche Werte:</b> Ganzzahl (Standard: 0)</p> <div> <p> Die Verbindung von min zu max erfolgt immer <b>im Uhrzeigersinn</b>.</p> </div>

Attribut	Beschreibung
startAngle	<p>Gibt den Startpunkt des Tachodiagramms an.</p> <p><b>Mögliche Werte:</b> Ganzzahl zwischen -360 und 360 (Standard: 255)</p> <div> <p><b>i</b> Bei dieser Angabe entsprechen folgende Ganzzahlen folgenden Positionen auf einer Uhr:</p> <ul style="list-style-type: none"> <li>• 0: 3 Uhr</li> <li>• 90: 12 Uhr</li> <li>• 180: 9 Uhr</li> <li>• 270: 6 Uhr</li> </ul> </div>
value	<p>Gibt an, welcher Wert des Datenobjekts visualisiert wird.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette oder Data Binding</p>

#### 1.9.11.4.3 Tooltip

<Tooltip>: Kindelement von <Gauge>. Definiert, wie der Tooltip für die einzelnen Werte im Diagramm angezeigt werden soll.

Für <Tooltip> lassen sich die allgemeinen Attribute (s.o.) definieren.

#### 1.9.11.4.4 Beispiele

##### Beispiel 1:

```

<Properties>
  <Property name="DataSource" type="Complex">
    <Property name="Satisfaction" type="Integer"/>
  </Property>
</Properties>
<FlowLayout>
  <Chart fontFamily="Font03" title="Employee Satisfaction">
    <Gauge max="0" min="100" value="#DataSource.Satisfaction">
      <Tooltip fontFamily="Font02" fontSize="12"/>
    </Gauge>
  </Chart>
</FlowLayout>

```

Das obige Beispiel führt zu folgendem Diagramm:

**Beispiel 2:**

```

<Property name="Datasource" type="Complex">
  <Property name="gauge" type="Integer"/>

  ...
  ...
  ...
</Property>

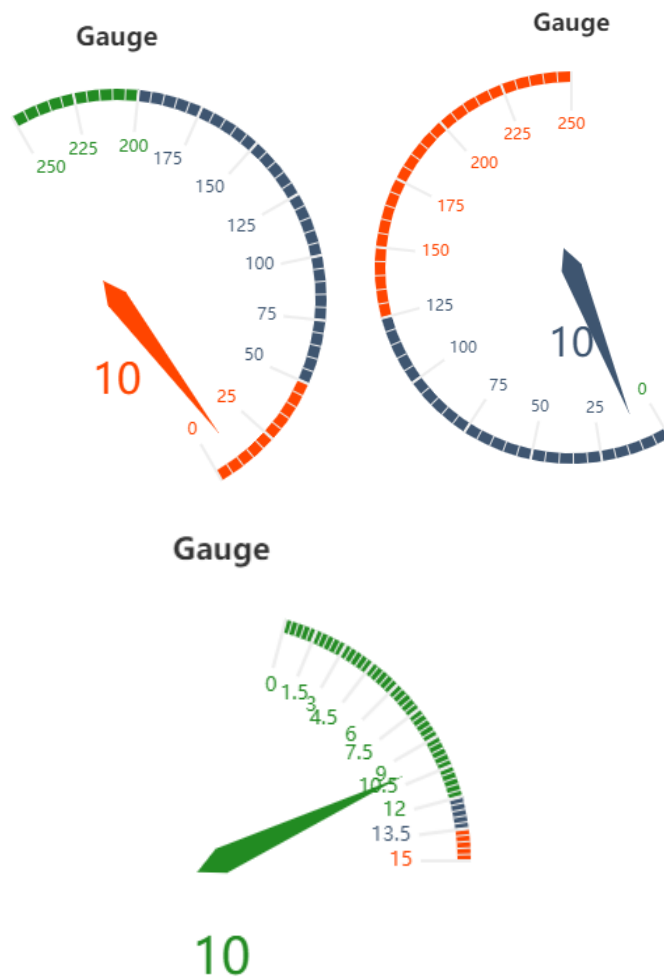
<Chart title="Gauge">
  <Gauge value="#Datasource.gauge" min="0" max="250" startAngle="-60" endAngle="-270" axisMin="0" axisMiddle="50" />
</Chart>

<Chart title="Gauge">
  <Gauge value="#Datasource.gauge" min="0" max="15" startAngle="75" axisMin="80" axisMiddle="90" />
</Chart>

<Chart title="Gauge">
  <Gauge value="#Datasource.gauge" min="-1" max="250" startAngle="120" />
</Chart>

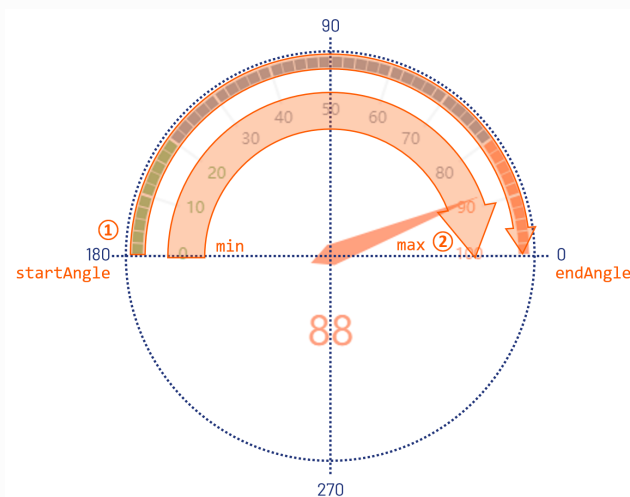
```

Das obige Beispiel führt zu folgenden Diagrammen



#### 1.9.11.4.5 Funktionsweise und weitere Beispiele für Tachodiagramme


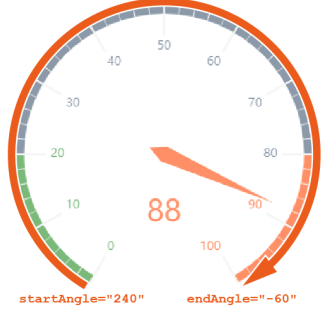
- ① Bei Tachodiagrammen muss folgendes beachtet werden:
1. `endAngle` muss kleiner sein als `startAngle`.
  2. Die Verbindung vom Startwinkel zum Endwinkel erfolgt immer **im Uhrzeigersinn**.
  3. Die Verbindung von min zu max (Skala) erfolgt immer **im Uhrzeigersinn**.





Im Folgenden sind weitere Beispiele für Tachodiagramme mit Erläuterung aufgezeigt, um die Funktionsweise des Start- und Endwinkels zu verdeutlichen.

Definition	Ergebnis	Erläuterung
<pre>&lt;Gauge startAngle="0" endAngle="-180" min="0" max="100" ...&gt;</pre>		
<pre>&lt;Gauge startAngle="90" endAngle="-90" min="0" max="100" ...&gt;</pre>		
<pre>&lt;Gauge startAngle="180" endAngle="0" min="0" max="100" ...&gt;</pre>		
<pre>&lt;Gauge startAngle="-90" endAngle="-270" min="0" max="100" ...&gt;</pre>		

Definition	Ergebnis	Erläuterung
<pre>&lt;Gauge startAngle="240" endAngle="-60" min="0" max="100" ...&gt;</pre>		

### 1.9.12 Maps

Mit dem Maps-Steuerelement können Karten verschiedenen Anbietern, u. a. Google Maps und Openstreetmap, in X4 Activities Web Apps eingebunden werden. Es können Geodaten angezeigt, Standorte gesucht und Routen gefunden werden.



#### 1.9.12.1 Maps-Steuerelement erzeugen

Ein Maps-Steuerelement wird mit dem Element `<Map>` erzeugt. Das Element kann nur im Layout einer [Detail-Komponente](#) verwendet werden.

#### Mögliche Attribute:

Zusätzlich zu den Standardattributen eines Steuerelements kann das Element `<Map>` folgende Attribute haben:

Attribut	Beschreibung
apiKey	<p>Gibt den API-Schlüssel an, falls dieser vom Kartenanbieter verlangt wird.</p> <p><b>Mögliche Werte:</b> API-Schlüssel</p> <div> <p><b>i</b> Für die Verwendung von Openstreetmap wird kein API-Schlüssel benötigt. Falls ein API-Schlüssel eingegeben wird, wenn Openstreetmap als Kartenanbieter verwendet wird, wird der API-Schlüssel ignoriert.</p> <p><b>⚠</b> Wenn Google Maps als Kartenanbieter verwendet wird, müssen folgende APIs aktiviert sein:</p> <ul style="list-style-type: none"> <li>• Maps JS API</li> <li>• Locations (für die Suche und die Routenberechnung)</li> <li>• Directions (für die Routenberechnung)</li> </ul> </div>

Attribut	Beschreibung
initialPointLongitude	<p>Definiert den Längengrad, auf den die Karte zentriert sein soll, wenn sie geladen wird.</p> <ul style="list-style-type: none"> <li>• Data Binding mit konstanten Werten möglich</li> </ul> <p><b>Mögliche Werte:</b> Angabe eines Längengrads in Grad mit Dezimalstellen (z. B. 6.75000)</p>
initialPointLatitude	<p>Definiert den Längengrad, auf den die Karte zentriert sein soll, wenn sie geladen wird.</p> <ul style="list-style-type: none"> <li>• Data Binding mit konstanten Werten möglich</li> </ul> <p><b>Mögliche Werte:</b> Angabe eines Breitengrads in Grad mit Dezimalstellen (z. B. 50.41871)</p>
satelliteView	<p>Legt fest, ob die Satellitenansicht der Karte aktiviert ist.</p> <p><b>Mögliche Werte:</b> true / false (Standard)</p> <div> <p> Die Satellitenansicht kann nicht verwendet werden, wenn OpenStreetMap als Kartenanbieter verwendet wird.</p> </div>
selectedIndex	<p>Ausgewählter Index</p> <div> <p> Das Attribut selectedIndex übernimmt den index-Wert eines einfachen Markers (SingleMarker) oder eine Markergruppe (Markers), sobald diese ausgewählt wurden.</p> </div> <p><b>Mögliche Werte:</b> String Binding</p>
vendor	<p><i>Erforderlich.</i> Spezifiziert den Kartenanbieter.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• OpenStreetMap</li> <li>• Google</li> <li>• Bing</li> </ul>

Attribut	Beschreibung
zoomLevel	<p>Spezifiziert die initiale Zoomstufe der Karte.</p> <div> <p><b>i</b> Die Werte geben die in der Karte angezeigten Meter ungefähr an.</p> <p>Beachten Sie, dass es beim Maßstabsbalken zu Abweichungen kommen kann, da sich der Maßstab in Abhängigkeit von der Entfernung vom Äquator ändert.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• 10</li> <li>• 25</li> <li>• 50</li> <li>• 100</li> <li>• 250</li> <li>• 500</li> <li>• 750</li> <li>• 1000</li> <li>• 2000</li> <li>• 5000</li> <li>• 10000</li> <li>• 25000</li> <li>• 50000</li> <li>• 100000</li> <li>• 2500000</li> <li>• 5000000</li> </ul>

### 1.9.12.2 Suchfunktion

Um einem Web App-Benutzer die Suche nach Orten im Maps-Steuerelement zu ermöglichen, wird das Element `<Search/>` innerhalb von `<Map>` verwendet.

```
<Map>
  <Search/>
</Map>
```

#### Mögliche Attribute:

Das Element `<Search/>` kann folgende Attribute haben:

Attribut	Beschreibung
searchIconHorizontalPosition	<p>Definiert die horizontale Position des Icons relativ zu den gesuchten Koordinaten.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• left: Icon ist links von der Koordinate</li> <li>• center (Standard): Icon ist auf der Koordinate</li> <li>• right: Icon ist rechts von der Koordinate</li> </ul>

Attribut	Beschreibung
searchIconUrl	Pfad zur Grafikdatei, die als Icon verwendet werden soll. Mit dem Icon wird das Suchergebnis markiert. Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources. <b>Mögliche Werte:</b> Zeichenkette (URI)
searchIconVerticalPosition	Definiert die vertikale Position des Icons relativ zu den gesuchten Koordinaten.  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• top: Icon ist über der Koordinate</li> <li>• center (Standard): Icon ist auf der Koordinate</li> <li>• bottom: Icon ist unter der Koordinate</li> </ul>

### 1.9.12.3 Routenberechnung

Um einem Web App-Benutzer die Berechnung einer Route zwischen zwei Orten im Maps-Steuerelement zu ermöglichen, wird das Element `<Routing/>` innerhalb von `<Map>` und ggf. nach `<Search/>` verwendet.

```
<Map ...>
  <Routing/>
</Map>
```

#### Mögliche Attribute:

Das Element `<Routing/>` kann folgende Attribute haben:

Attribut	Beschreibung
destinationIconHorizontalPosition	Definiert die horizontale Position des Zielicons relativ zu den gesuchten Koordinaten.  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• left: Icon ist links von der Koordinate</li> <li>• center (Standard): Icon ist auf der Koordinate</li> <li>• right: Icon ist rechts von der Koordinate</li> </ul>
destinationIconUrl	Pfad zur Grafikdatei, die als Icon für den Zielpunkt verwendet werden soll. Mit dem Icon wird der Zielpunkt der Route markiert. Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources. <b>Mögliche Werte:</b> Zeichenkette (URI)

Attribut	Beschreibung
destinationIconVerticalPosition	<p>Definiert die vertikale Position des Zielicons relativ zu den gesuchten Koordinaten.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• top: Icon ist über der Koordinate</li> <li>• center (Standard): Icon ist auf der Koordinate</li> <li>• bottom: Icon ist unter der Koordinate</li> </ul>
startIconHorizontalPosition	<p>Definiert die horizontale Position des Starticons relativ zu den gesuchten Koordinaten.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• left: Icon ist links von der Koordinate</li> <li>• center (Standard): Icon ist auf der Koordinate</li> <li>• right: Icon ist rechts von der Koordinate</li> </ul>
startIconUrl	<p>Pfad zur Grafikdatei, die als Icon für den Startpunkt verwendet werden soll. Mit dem Icon wird der Startpunkt der Route markiert. Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources.</p> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p>
startIconVerticalPosition	<p>Definiert die vertikale Position des Starticons relativ zu den gesuchten Koordinaten.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• top: Icon ist über der Koordinate</li> <li>• center (Standard): Icon ist auf der Koordinate</li> <li>• bottom: Icon ist unter der Koordinate</li> </ul>

#### 1.9.12.4 Einfacher Marker

Um einen einfachen Marker in einem Maps-Steuerelement anzuzeigen, wird das Element `<SingleMarker/>` innerhalb von `<Map>` und ggf. nach `<Search/>` und `<Routing/>` verwendet.




```
<Map>
...
  <SingleMarker/>
</Map>
```

#### Mögliche Attribute:

Das Element `<SingleMarker/>` kann folgende Attribute haben:

Attribut	Beschreibung
componentName	<p>Gibt den Namen der Komponente für eine interne Navigation an.</p> <p><b>Mögliche Werte:</b> Zeichenkette (Name einer Komponente)</p>

Attribut	Beschreibung
description	<p>Gibt die Beschreibung des Markers an.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Zeichenkette inkl. Leerzeichen</p>
externalLink	<p>Gibt den Link an, der im Pop-up des Markers angezeigt wird.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Zeichenkette (URL)</p>
externalLinkTarget	<p>Gibt an, ob der Link (externalLink) im selben oder in einem neuen Tab geöffnet wird.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• same: Der Link wird im selben Tab geöffnet</li> <li>• new: Der Link wird in einem neuen Tab geöffnet</li> </ul>
iconHorizontalPosition	<p>Definiert die horizontale Position des Icons relativ zu den eingegebenen Koordinaten.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• left: Icon ist links von der Koordinate</li> <li>• center (Standard): Icon ist auf der Koordinate</li> <li>• right: Icon ist rechts von der Koordinate</li> </ul>
iconUrl	<p>Pfad zur Grafikdatei, die als Icon für den Marker verwendet werden soll. Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources.</p> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p>
iconVerticalPosition	<p>Definiert die vertikale Position des Icons relativ zu den eingegebenen Koordinaten.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• top: Icon ist über der Koordinate</li> <li>• center (Standard): Icon ist auf der Koordinate</li> <li>• bottom: Icon ist unter der Koordinate</li> </ul>
index	<p>Index des Markers</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>

Attribut	Beschreibung
latitude	<p><i>Erforderlich, wenn searchString nicht verwendet wird.</i> Gibt den Breitengrad des Markers an.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Angabe eines Breitengrads in Grad mit Dezimalstellen (z. B. 50.41871)</p> <div>  Dieses Attribut kann nicht gleichzeitig mit searchString verwendet werden. </div>
longitude	<p><i>Erforderlich, wenn searchString nicht verwendet wird.</i> Gibt den Längengrad des Markers an.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Angabe eines Längengrads in Grad mit Dezimalstellen (z. B. 6.75000)</p> <div>  Dieses Attribut kann nicht gleichzeitig mit searchString verwendet werden. </div>
name	<p>Gibt den Namen des Markers an.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Zeichenkette aus alphanumerischen Zeichen</p>
navigationDisplayName	<p>Gibt den Text an, der im Pop-up des Markers statt eines externen Links oder eines Komponentennamens angezeigt wird.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Zeichenkette inkl. Leerzeichen</p>
searchString	<p><i>Erforderlich, wenn latitude/longitude nicht verwendet wird.</i> Adresszeile zur Suche nach ausgewählten Diensten des Kartenanbieters.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Zeichenkette für eine Dienstsuche, z. B. Restaurant</p> <div>  Dieses Attribut kann nicht gleichzeitig mit den Attributen latitude/longitude verwendet werden. </div>

### 1.9.12.5 Markergruppe

Um mehrere Marker in einem Maps-Steuerelement anzuzeigen, wird das Element `<Markers/>` innerhalb von `<Map>` und ggf. nach `<Search/>` und `<Routing/>` verwendet.




```

<Map>
  ...
  <Markers/>
</Map>





```

### Mögliche Attribute:

Das Element <Markers/> kann folgende Attribute haben:

Attribut	Beschreibung
cluster	<p>Gibt an, ob Marker zusammengefasst werden sollen, wenn sie auf der Karte sehr dicht beieinander liegen.</p> <p><b>Mögliche Werte:</b> true / false (Standard)</p> <div>  cluster und number können nicht gleichzeitig verwendet werden! </div>
componentName	<p>Gibt den Namen der Komponente für eine interne Navigation an.</p> <p><b>Mögliche Werte:</b> Zeichenkette (Name einer Komponente)</p>
data	<p><i>Erforderlich.</i> Definiert, welche Daten für die Markergruppe verwendet werden.</p> <ul style="list-style-type: none"> <li>• Data Binding erforderlich</li> </ul> <p><b>Mögliche Werte:</b> Data-Binding-Ausdruck</p>
description	<p>Gibt die Beschreibung der Marker an.</p> <ul style="list-style-type: none"> <li>• Data Binding erforderlich</li> </ul> <p><b>Mögliche Werte:</b> Data-Binding-Ausdruck relativ zum Ausdruck in data</p>
externalLink	<p>Gibt den Link an, der im Pop-up des Markers angezeigt wird.</p> <ul style="list-style-type: none"> <li>• Data Binding erforderlich</li> </ul> <p><b>Mögliche Werte:</b> Data-Binding-Ausdruck relativ zum Ausdruck in data</p>
externalLinkTarget	<p>Gibt an, ob der Link (externalLink) im selben oder in einem neuen Tab geöffnet wird.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• same: Der Link wird im selben Tab geöffnet</li> <li>• new: Der Link wird in einem neuen Tab geöffnet</li> </ul>

Attribut	Beschreibung
iconHorizontalPosition	<p>Definiert die horizontale Position des Icons relativ zu den eingegebenen Koordinaten.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• left: Icon ist links von der Koordinate</li> <li>• center (Standard): Icon ist auf der Koordinate</li> <li>• right: Icon ist rechts von der Koordinate</li> </ul>
iconUrl	<p>Pfad zur Grafikdatei, die als Icon für die Marker verwendet werden soll. Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources.</p> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p>
iconVerticalPosition	<p>Definiert die vertikale Position des Icons relativ zu den eingegebenen Koordinaten.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• top: Icon ist über der Koordinate</li> <li>• center (Standard): Icon ist auf der Koordinate</li> <li>• bottom: Icon ist unter der Koordinate</li> </ul>
index	<p>Index der Markergruppe</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
latitude	<p><i>Erforderlich, wenn searchString nicht verwendet wird.</i> Gibt den Breitengrad eines Markers an.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Data-Binding-Ausdruck relativ zum Ausdruck in data</p>
longitude	<p><i>Erforderlich, wenn searchString nicht verwendet wird.</i> Gibt den Längengrad eines Markers an.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Data-Binding-Ausdruck relativ zum Ausdruck in data</p>
name	<p>Gibt den Namen der Marker an.</p> <ul style="list-style-type: none"> <li>• Data Binding erforderlich</li> </ul> <p><b>Mögliche Werte:</b> Data-Binding-Ausdruck relativ zum Ausdruck in data</p>
navigationDisplayName	<p>Gibt den Text an, der im Pop-up des Markers statt eines externen Links oder eines Komponentennamens angezeigt wird.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Zeichenkette inkl. Leerzeichen</p>

Attribut	Beschreibung
number	<p>Gibt an, ob die Marker nummeriert werden sollen.</p> <p><b>Mögliche Werte:</b> true / false (Standard)</p> <div>  cluster und number können nicht gleichzeitig verwendet werden! </div>
searchString	<p><i>Erforderlich, wenn latitude/longitude nicht verwendet wird.</i> Adresszeile zur Suche nach ausgewählten Diensten des Kartenanbieters.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Zeichenkette für eine Dienstsuche, z. B. Restaurant</p> <div>  Dieses Attribut kann nicht gleichzeitig mit den Attributen latitude/longitude verwendet werden. </div>
textHorizontalPosition	<p>Definiert die horizontale Position der Nummerierung relativ zu den eingegebenen Koordinaten.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• left: Nummerierung ist links von der Koordinate (Standard)</li> <li>• center: Nummerierung ist auf der Koordinate</li> <li>• right: Nummerierung ist rechts von der Koordinate</li> </ul> <div>  Bing Maps positioniert die Nummerierung automatisch. Daher hat diese Funktion auf Bing Maps keinen Einfluss. </div>
textVerticalPosition	<p>Definiert die vertikale Position der Nummerierung relativ zu den eingegebenen Koordinaten.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• top: Nummerierung ist über der Koordinate (Standard)</li> <li>• center: Nummerierung ist auf der Koordinate</li> <li>• bottom: Nummerierung ist unter der Koordinate</li> </ul> <div>  Bing Maps positioniert die Nummerierung automatisch. Daher hat diese Funktion auf Bing Maps keinen Einfluss. </div>

### 1.9.12.6 Aktion "Auswählen" innerhalb einer Map

Die Aktion **Auswählen** wird über das Element `<SelectAction/>` innerhalb von `<SingleMarker>` oder `<Markers>` erzeugt und ausgelöst, sobald ein entsprechender Marker ausgewählt wurde.

### 1.9.12.7 Beispiele

In diesem Beispiel werden die Properties definiert und eine Gruppe von Markern eingeblendet.

#### Beispiel 1 Karten-Steuerelement (\*.wad)

```
<Properties>
  <Property name="favorites" type="List">
    <Property name="lat" type="Decimal" />
    <Property name="lng" type="Decimal" />
  </Property>
</Properties>

...

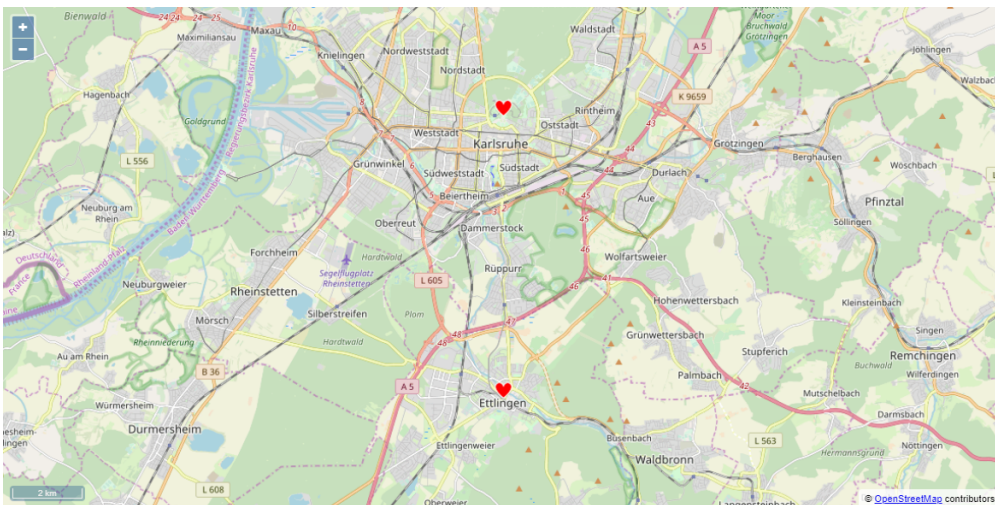
<Map vendor="OpenStreetMap">
  <Markers longitude="#lng" data="#favorites" latitude="#lat" iconUrl="heart.png"
name="Favorites" description="Places, I have visited." iconVerticalPosition="top"/>
</Map>
```

#### Beispiel 1: Durch den Prozess gelieferte Daten (\*.xml)

```
<OkList>
  <favorites>
    <lat>48.94061</lat>
    <lng>8.40471</lng>
  </favorites>

  <favorites>
    <lat>49.01396</lat>
    <lng>8.40445</lng>
  </favorites>
</OkList>
```

Obiges Beispiel führt zu folgendem Ergebnis:



In diesem Beispiel werden zwei Gruppen von Markern eingeblendet sowie die Suche und Routenberechnung aktiviert.

### Beispiel 2 Kartensteuerelement

```
<Properties>
  <Property name="favorites" type="List">
    <Property name="lat" type="Decimal" />
    <Property name="lng" type="Decimal" />
  </Property>
  <Property name="destinations" type="List">
    <Property name="lat" type="Decimal" />
    <Property name="lng" type="Decimal" />
  </Property>
</Properties>

...

<Map vendor="OpenStreetMap">
  <Search/>
  <Routing destinationIconUrl="finish.png" startIconUrl="car.png"
destinationIconVerticalPosition="top" destinationIconHorizontalPosition="center"/>
  <Markers longitude="#lng" data="#favorites" latitude="#lat" iconUrl="heart.png"
name="Favorites" description="Places, I have visited." iconVerticalPosition="top"/>
  <Markers longitude="#lng" data="#destinations" latitude="#lat" iconUrl="star.png"
name="Destinations" description="Places, I want to visit."/>
</Map>
```

### Beispiel 2: Durch den Prozess gelieferte Daten (\*.xml)

```

<OkList>
  <favorites>
    <lat>48.94061</lat>
    <lng>8.40471</lng>
  </favorites>

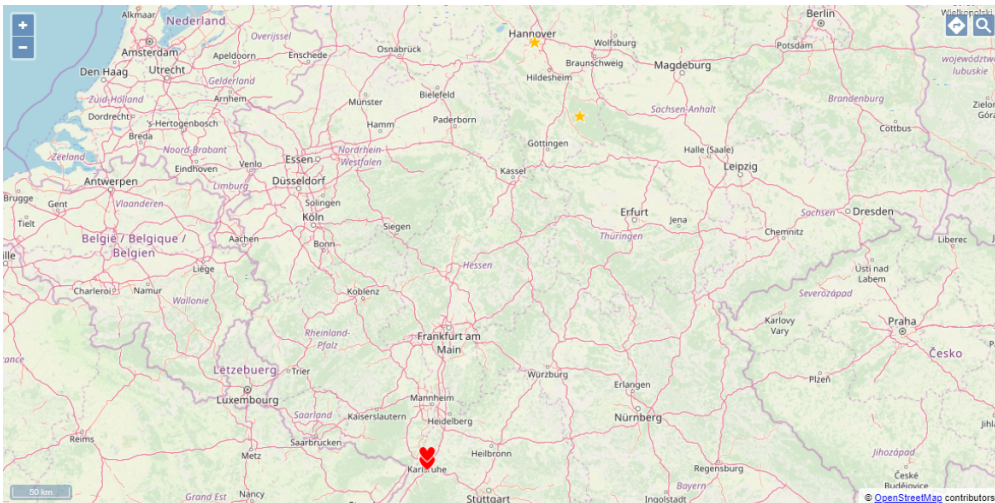
  <favorites>
    <lat>49.01396</lat>
    <lng>8.40445</lng>
  </favorites>

  <destinations>
    <lat>51.8058</lat>
    <lng>10.3343</lng>
  </destinations>

  <destinations>
    <lat>52.3807</lat>
    <lng>9.7706</lng>
  </destinations>
</OkList>

```

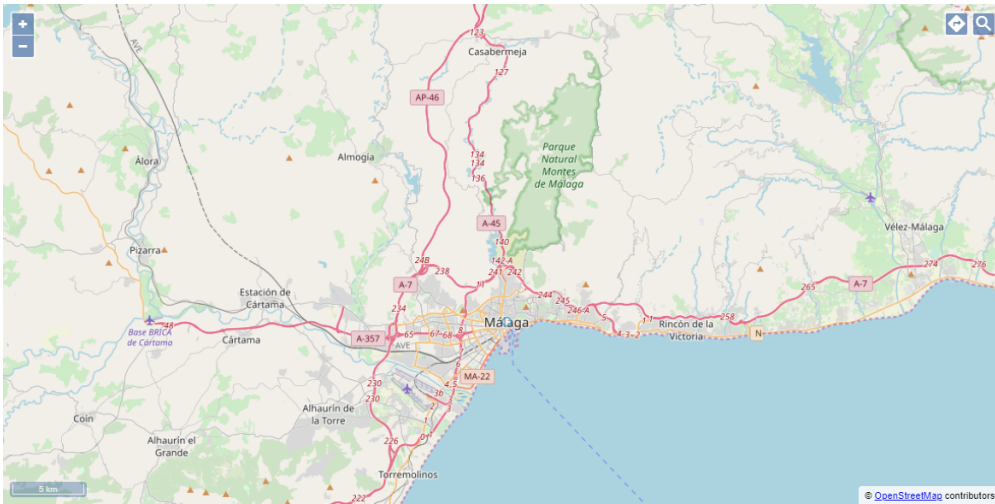
Obiges Beispiel führt zu folgendem Ergebnis:



*Mehrere Gruppen von Markern mit verschiedenen Symbolen (Openstreetmap)*



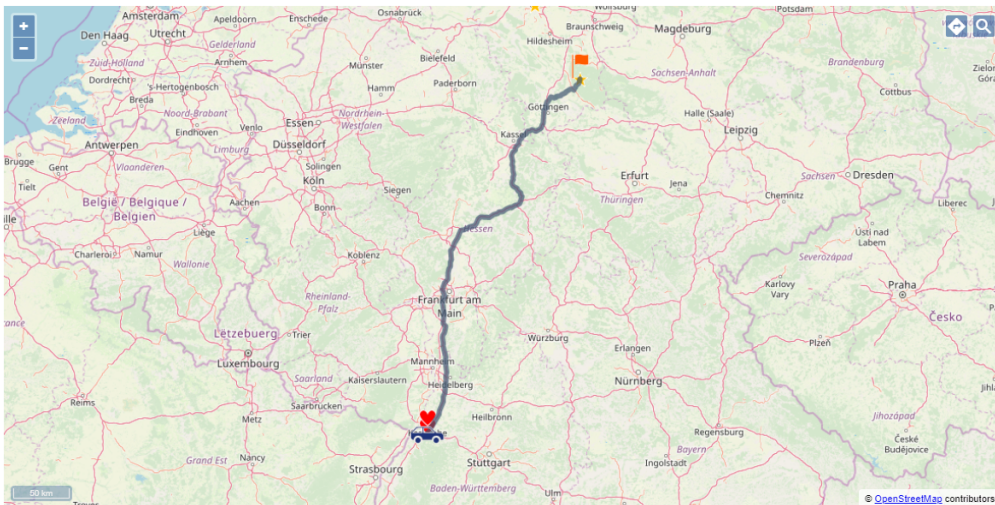
Suche  
Málaga, Andalusien, Spanien



### Suchfunktion (Openstreetmap)

Start  
Ettlingen, Landkreis Karlsruhe, Regierungsbezirk Karlsruhe, Baden-Württemberg, 76275, Deutschland

Ziel  
Clausthal, Clausthal-Zellerfeld, Landkreis Goslar, Niedersachsen, 38678, Deutschland



### Routenberechnung (Openstreetmap)

Weitere Kartenanbieter sind ebenfalls verfügbar. Das folgende Beispiel zeigt ein Map-Steuerelement mit Google Maps.

**Beispiel 2 Kartensteuerelement**

```

<Properties>
  <Property name="favorites" type="List">
    <Property name="lat" type="Decimal" />
    <Property name="lng" type="Decimal" />
  </Property>
  <Property name="destinations" type="List">
    <Property name="lat" type="Decimal" />
    <Property name="lng" type="Decimal" />
  </Property>
</Properties>

...

<Map vendor="Google">
  <Search/>
  <Routing destinationIconUrl="finish.png" startIconUrl="car.png"
destinationIconVerticalPosition="top" destinationIconHorizontalPosition="center"/>
  <Markers longitude="#lng" data="#favorites" latitude="#lat" iconUrl="heart.png"
name="Favorites" description="Places, I have visited." iconVerticalPosition="top"/>
  <Markers longitude="#lng" data="#destinations" latitude="#lat" iconUrl="star.png"
name="Destinations" description="Places, I want to visit."/>
</Map>

```

**Beispiel 2: Durch den Prozess gelieferte Daten (\*.xml)**

```

<OkList>
  <favorites>
    <lat>48.94061</lat>
    <lng>8.40471</lng>
  </favorites>

  <favorites>
    <lat>49.01396</lat>
    <lng>8.40445</lng>
  </favorites>

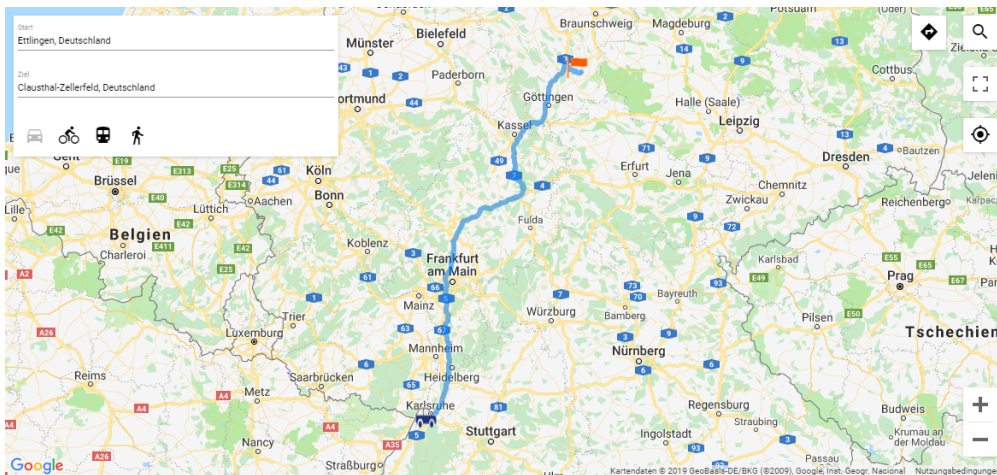
  <destinations>
    <lat>51.8058</lat>
    <lng>10.3343</lng>
  </destinations>

  <destinations>
    <lat>52.3807</lat>
    <lng>9.7706</lng>
  </destinations>
</OkList>

```

Obiges Beispiel führt zu folgendem Ergebnis:





### 1.9.13 Card

Mit dem Card-Steuerelement können Karten in X4 Activities Web Apps eingebunden werden, um beispielsweise Teaser oder Dashboards in Anwendungen zu integrieren. Eine Card kann einen Titel und Untertitel, verschiedene Aktionen sowie ein Layout mit beliebigen Steuerelementen enthalten.

#### 1.9.13.1 Card-Steuerelement erzeugen

Ein Card-Steuerelement wird mit dem Element `<Card>` erzeugt. Das Element kann nur im Layout einer [Detail-Komponente](#) verwendet werden.

#### Mögliche Attribute:

Zusätzlich zu den Standardattributen eines Steuerelements kann das Element `<Card>` folgende Attribute haben:

Attribut	Beschreibung
icon	<p>Icon für das Karten-Steuerelement.</p> <ul style="list-style-type: none"> <li>• Data Binding mit Base64-Werten möglich</li> </ul> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Base64-kodierte Grafikdatei</li> <li>• Pfad zu einer Grafikdatei als Zeichenkette (URI), z. B. <i>clock.png</i></li> <li>• Angabe eines Material Icons, z. B. <i>icon:extension</i></li> </ul> <div style="border: 1px solid green; padding: 10px; margin: 10px 0;"> <p>✔ Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren <a href="#">Material Icons</a> abweichen.</p> </div> <div style="border: 1px solid blue; padding: 10px; margin: 10px 0;"> <p>ℹ</p> <ul style="list-style-type: none"> <li>• Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources.</li> <li>• Das Material Icon muss mit dem Präfix icon, z. B. icon:&lt;MaterialIconName&gt; angegeben werden.</li> </ul> </div> <div style="border: 1px solid orange; padding: 10px; margin: 10px 0;"> <p>⚠ Dieses Attribut kann nicht gleichzeitig mit iconUrl verwendet werden.</p> </div>
iconColor	<p>Definiert die Farbe des Karten-Icons.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Hexadezimaler Farbwert, z. B. <i>ff5a00</i></li> </ul> <div style="border: 1px solid orange; padding: 10px; margin: 10px 0;"> <p>⚠ Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts!</p> </div> <ul style="list-style-type: none"> <li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. <i>A200</i></li> </ul>
subtitle	<p>Untertitel der Karte.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul>
title	<p>Titel der Karte.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul>

### 1.9.13.2 Aktionen

Innerhalb des Elements <Card> können [Aktionen](#) definiert werden.

- ❗ Wenn eine **Auswählen**-Aktion (<SelectAction/>) im <Card>-Element definiert wurde, dann können keine weiteren Aktionen mehr definiert werden und das ganze <Card>-Element kann angeklickt werden. Ist keine Auswählen-Aktion angegeben, darf das <Card>-Element beliebig viele andere Aktionen enthalten.

```
<Card>
  <FlowLayout>
    ...
  </FlowLayout>
  <Actions>
    <CustomAction displayName="Card action"/>
  </Actions>
</Card>
```

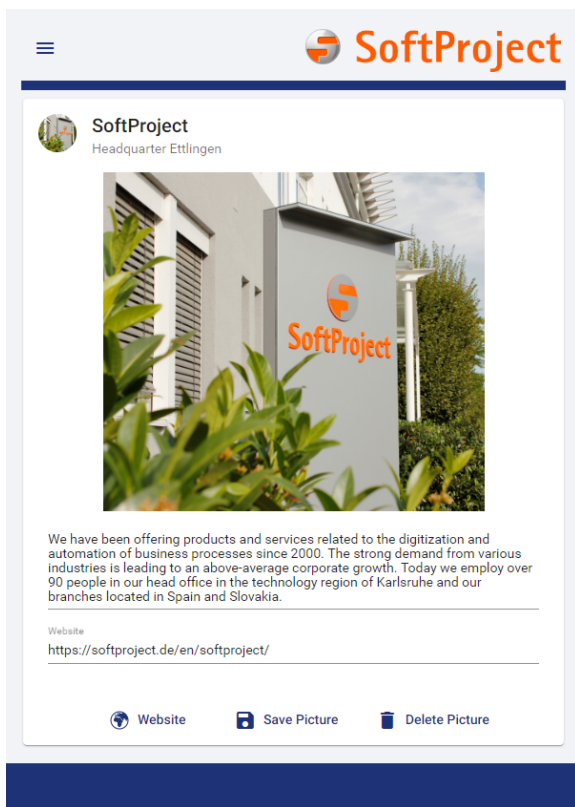
### 1.9.13.3 Beispiele

In diesem Beispiel werden die Attribute sowie die Inhalte des Card-Steuerelements definiert.

#### Beispiel-Struktur

```
<Card border="true" icon="SPHeadquarters.jpg" title="SoftProject" subtitle="Headquarters Ettlingen">
  <FlowLayout>
    <Image resourceUrl="SPHeadquarters.jpg" width="400"/>
    <TextBox border="false" type="text" value="We have been offering products and services related to the digitization and automation of business processes since 2000. The strong demand from various industries is leading to an above-average corporate growth. Today we employ over 90 people in our head office in the technology region of Karlsruhe and our branches located in Spain and Slovakia." multiline="true"/>
    <TextBox type="url" value="https://softproject.de/en/softproject/" displayName="Website"/>
  </FlowLayout>
  <Actions>
    <CustomAction icon="Website.png" iconPosition="left" displayName="Website" externalLink="https://softproject.de/de/">
      <SaveAction displayName="Save Picture"/>
      <DeleteAction displayName="Delete Picture"/>
    </Actions>
</Card>
```

Das obige Beispiel führt zum folgenden Ergebnis:



## 1.10 Aktionen

Innerhalb einer Komponente können Schaltflächen erzeugt werden, die Aktionen auslösen. Diese Schaltflächen werden immer am oberen Rand vor dem Inhalt der Komponente angezeigt. Typische Aktionen sind z. B. das Löschen und Speichern von Inhalten. Aktionen werden innerhalb des `<Action>`-Elements definiert.

**⚠** Es ist nicht möglich, Aktionen auf der Ebene einer **Master/Detail-Komponente** zu definieren. Aktionen können nur innerhalb der Unterkomponenten einer Master/Detail-Komponente definiert werden!

Es können verschiedene Aktionen definiert werden:



- Aktion „Neu“ (`<NewAction/>`)
- Aktion „Speichern“ (`<SaveAction/>`)
- Aktion „Löschen“ (`<DeleteAction/>`)
- Aktion „Auswählen“ (`<SelectAction/>`)
- Aktion „Datei hochladen“ (`<UploadAction/>`)
- Aktion „Datei herunterladen“ (`<DownloadAction/>`)
- Benutzerdefinierte Aktion (`<CustomAction/>`)
- Aktion "Neu laden" (`<ReloadAction/>`)




Alle Aktionen außer der benutzerdefinierten Aktion haben vordefinierte Beschriftungen und Icons. Diese können überschrieben werden, wenn es notwendig ist.

In der folgenden Tabelle sind die Attribute erläutert, die für alle Aktionen verwendet werden können.

Attribut	Beschreibung
background	<p>Definiert eine Farbe für den Hintergrund des Buttons.</p> <div>  Diese Einstellung überschreibt die Standardfarbe des Farbschemas!         </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Hexadezimaler Farbwert, z. B. ff5a00</li> </ul> <div>  Verwenden Sie keine Raute (#) vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts!         </div> <ul style="list-style-type: none"> <li>Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. A200</li> </ul>
componentName	<p>Name der Komponente, zu der am Ende der Aktion navigiert werden soll.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
displayName	<p>Beschriftung des Buttons.</p> <ul style="list-style-type: none"> <li>Übersetzbar</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette oder Sprachschlüssel</p>
enabled	<p>Aktiviert die Aktion</p> <ul style="list-style-type: none"> <li>Data Binding (Boolean) möglich</li> </ul> <p><b>Mögliche Werte:</b> true/false oder Ausdruck für Data Binding</p> <div>  Das Attribut enabled ersetzt das veraltete Attribut disabled. <i>enabled="true"</i> entspricht damit dem veralteten Attribut <i>disabled="false"</i>.         </div>
fontFamily	<p>Legt die Schriftfamilie fest. Mit diesem Attribut wird die Standardschrift der Web App für dieses Steuerelement überschrieben.</p> <p><b>Mögliche Werte:</b> Fontcode aus der Font-Palette, z. B. Font04</p>

Attribut	Beschreibung
fontSize	<p>Legt die Schriftgröße fest. Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Steuerelement überschrieben.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Schriftgröße in Pixel, z. B. <i>20px</i></li> <li>• Schriftgröße in Punkt, z. B. <i>18pt</i></li> <li>• Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i></li> <li>• Schlüsselwörter, z. B. <i>small</i> oder <i>larger</i></li> </ul>
fontStretch	<p>Legt die Breite der einzelnen Schriftzeichen fest. Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Steuerelement überschrieben.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Condensed</li> <li>• Expanded</li> <li>• ExtraCondensed</li> <li>• ExtraExpanded</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• SemiCondensed</li> <li>• SemiExpanded</li> <li>• UltraCondensed</li> <li>• UltraExpanded</li> </ul>
fontStyle	<p>Legt die Neigung der Schrift fest. Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Steuerelement überschrieben.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <i>italic</i>: kursive Schrift</li> <li>• <i>normal</i>: normale Schrift (Standard)</li> <li>• <i>oblique</i>: schräg gestellter Schriftstil (berechnet)</li> </ul>

Attribut	Beschreibung
fontWeight	<p>Legt die Schriftstärke fest. Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Steuerelement überschrieben.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Black</li> <li>• Bold</li> <li>• DemiBold</li> <li>• ExtraBlack</li> <li>• ExtraBold</li> <li>• ExtraLight</li> <li>• Heavy</li> <li>• Light</li> <li>• Medium</li> <li>• Normal (Standard)</li> <li>• Regular</li> <li>• SemiBold</li> <li>• Thin</li> <li>• UltraBlack</li> <li>• UltraBold</li> <li>• UltraLight</li> </ul>
foreground	<p>Definiert eine Farbe für den Vordergrund (Texte, Icons) des Buttons.</p> <div data-bbox="547 1128 1434 1424"> <p> • Diese Einstellung überschreibt die Standardfarbe des Farbschemas!</p> <p>• Wenn für die Aktion mit dem Attribut <code>iconUrl</code> ein benutzerdefiniertes Icon verwendet wird, wird dieses Icon nicht in der benutzerdefinierten Farbe angezeigt. Die Grafikdatei, die als Icon verwendet wird, muss direkt in der gewünschten Farbe hinterlegt werden.</p> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• Hexadezimaler Farbwert, z. B. <code>ff5a00</code></li> </ul> <div data-bbox="592 1561 1434 1702"> <p> Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts!</p> </div> <ul style="list-style-type: none"> <li>• Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. <code>A200</code></li> </ul>

Attribut	Beschreibung
iconColor	<p>Definiert die Farbe des Icons.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Hexadezimaler Farbwert, z. B. <i>ff5a00</i></li> </ul> <div>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts!         </div> <ul style="list-style-type: none"> <li>Farbcode aus der Farbpalette der Web App (vgl. <a href="#">Theming</a>), z. B. <i>A200</i></li> </ul>
iconPosition	<p>Legt die Position des Icons auf dem Button fest.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li><i>left</i>: Das Icon wird links auf dem Button angezeigt (Standard).</li> <li><i>right</i>: Das Icon wird rechts auf dem Button angezeigt.</li> </ul>
iconUrl	<p>Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Icon verwendet wird.</p> <div>  <ul style="list-style-type: none"> <li>Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources.</li> <li>Das Material Icon muss mit dem Präfix <i>icon</i>, z. B. <i>icon:&lt;MaterialIconName&gt;</i> angegeben werden.</li> </ul> </div> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>Zeichenkette (URI), z. B. <i>clock.png</i></li> <li><i>icon:&lt;MaterialIconName&gt;</i>, z. B. <i>icon:extension</i></li> </ul> <div>  Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren <a href="#">Material Icons</a> abweichen.         </div>
process	<p>Pfad zur .wrf-Datei. Der Prozess muss im Ordner Services/Processes enthalten sein. Pfadangabe relativ zum Ordner Services/Processes.</p> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p>
selectionNeeded	<p>Legt fest, ob für die Aktion ein Eintrag aus einer Liste gewählt sein muss. Funktioniert nur in List-Komponenten.</p> <p><b>Mögliche Werte:</b> <i>true</i> / <i>false</i></p>



Attribut	Beschreibung
tooltip	<p>Legt den Text fest, der als Tooltip der Aktion angezeigt wird, sobald der Aktions-Button mit der Maus überfahren wird.</p> <p><b>Mögliche Werte:</b> Zeichenkette</p> <div> <i>i</i> Wenn der Wert des Attributs <code>tooltip</code> leer ist, dann wird kein Tooltip angezeigt. </div>
visible	<p>Legt fest, ob der Button für die Aktion sichtbar ist.</p> <ul style="list-style-type: none"> <li>• Data Binding (Boolean) möglich</li> </ul> <p><b>Mögliche Werte:</b> <code>true</code>/<code>false</code> oder Zeichenkette für Data Binding</p>

### 1.10.1 Aktion "Neu"

Die Aktion **Neu** wird über das Element `<NewAction/>` innerhalb von `<Actions>` erzeugt. Mit der Aktion **Neu** wird ein neuer Eintrag angelegt. Dazu wird im Detail-Strukturelement ein neues Objekt angelegt. Die darin enthaltenen Felder sind leer.

*i* Für die Aktion **Neu** lassen sich die Standardattribute für Aktionen definieren.

*i* Das Attribut `componentName` ist für das Element `<NewAction/>` erforderlich.

#### Beispiel

##### Beispiel Aktion "Neu"

```
<DetailComponent path="Dashboard" displayName="Dashboard" default="true">
  <Actions>
    <NewAction componentName="personDetail"/>
  </Actions>
  <FlowLayout>
    ...
  </FlowLayout>
</DetailComponent>
```

Das obige Beispiel erzeugt folgende Aktion:



### 1.10.2 Aktion "Speichern"

Die Aktion **Speichern** wird über das Element `<SaveAction/>` innerhalb von `<Actions>` erzeugt. Mit der Aktion **Speichern** werden eingegebene Daten gespeichert. Der Status ist abhängig vom Validierungsstatus.

 Für die Aktion **Speichern** lassen sich die Standardattribute für Aktionen definieren.

#### Beispiel

##### Beispiel Aktion "Speichern"

```
<DetailComponent path="Dashboard" displayName="Dashboard" default="true">
  <Actions>
    <SaveAction componentName="personDetail"/>
  </Actions>
  <FlowLayout>
    ...
  </FlowLayout>
</DetailComponent>
```

Das obige Beispiel erzeugt folgende Aktion:



### 1.10.3 Aktion "Löschen"

Die Aktion **Löschen** wird über das Element `<DeleteAction/>` innerhalb von `<Actions>` erzeugt. Mit der Aktion **Löschen** werden die Daten gelöscht.

 Für die Aktion **Löschen** lassen sich die Standardattribute für Aktionen definieren.

#### Beispiel

##### Beispiel Aktion "Löschen"

```
<DetailComponent path="Dashboard" displayName="Dashboard" default="true">
  <Actions>
    <DeleteAction/>
  </Actions>
  <FlowLayout>
    ...
  </FlowLayout>
</DetailComponent>
```

Das obige Beispiel erzeugt folgende Aktion:



### 1.10.4 Aktion "Auswählen"

Die Aktion **Auswählen** wird über das Element `<SelectAction/>` erzeugt. Die Aktion wird ausgelöst, sobald ein Eintrag aus einer [List](#)-Komponente oder aus einem der folgenden Steuerelemente ausgewählt wurde: [ComboBox](#), [Radio-Button](#), [Checkbox](#), [ListView](#), [Image](#), [Maps](#).

Zusätzlich zu den Standardattributen für [Aktionen](#) kann das Element `<SelectAction/>` noch folgende Attribute haben:

Attribut	Beschreibung
externalLink	<p>Ermöglicht die Navigation zu einer externen Website, indem ein externer Link festgelegt wird.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Zeichenkette (URL)</p> <div> <p><b>i</b> Der externe Link kann auch von einem Prozess übergeben werden. Dazu muss mit der Aktion ein Prozess (Attribut <code>process</code>) verknüpft werden, der in seiner Antwort den externen Link enthält:</p> <pre>&lt;ok externalLink="http://www.google.de" /&gt;</pre> </div>
externalLinkTarget	<p>Gibt an, ob der Link (<code>externalLink</code>) im selben oder in einem neuen Tab geöffnet wird.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <code>same</code>: Der Link wird im selben Tab geöffnet</li> <li>• <code>new</code>: Der Link wird in einem neuen Tab geöffnet</li> </ul>

### 1.10.5 Aktion "Datei hochladen"

Die Aktion **Datei hochladen** wird über das Element `<UploadAction/>` innerhalb von `<Actions>` erzeugt. Mit der Aktion **Upload** können Daten in die Web-Anwendung hochgeladen werden.

**i** Das Attribut `process` ist für das Element `<UploadAction/>` erforderlich.

Zusätzlich zu den Standardattributen für [Aktionen](#) kann das Element `<UploadAction/>` noch folgende Attribute haben:

Attribut	Beschreibung
acceptedFileTypes	<p>Dateitypen, die für den Upload akzeptiert werden.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich.</li> </ul> <p><b>Mögliche Werte:</b> MIME-Typen, auch mit Wildcards (*), z. B.</p> <ul style="list-style-type: none"> <li>• text/*: Textdateien</li> <li>• image/*: Grafikdateien <ul style="list-style-type: none"> <li>• image/jpg: Nur JPG-Dateien</li> <li>• image/png: Nur PNG-Dateien</li> </ul> </li> <li>• video/*: Videodateien</li> <li>• audio/*: Audiodateien</li> <li>• application/*: Dateien, die an ein bestimmtes Programm gebunden sind</li> <li>• multipart/*: mehrteilige Daten</li> <li>• message/*: Nachrichten</li> <li>• model/*: Dateien, die mehrdimensionale Strukturen repräsentieren</li> </ul>
fileId	<p>Identifikator, der vom Prozessentwickler verwendet werden kann.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich.</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
maxSizeMB	<p>Maximale Dateigröße in Mebibyte.</p> <p><b>Mögliche Werte:</b> Ganzzahl</p>

### Format der hochgeladenen Daten

Die Daten werden in einem bestimmten Input-Format an den Prozess übergeben, der mit dem Attribut process angegeben wird:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<File>
  <FileId>....</FileId>
  <FileName>....</FileName>
  <FileType>....</FileType>
  <FileData>
    ....Base64 data....
  </FileData>
</File>
```

Die Daten im Element <FileData> können z. B. mit dem Base64 Converter dekodiert werden.

### Beispiel

**Beispiel Aktion "Datei hochladen"**

```
<DetailComponent path="Dashboard" displayName="Dashboard" default="true">
  <Actions>
    <UploadAction />
  </Actions>
  <FlowLayout>
    ...
  </FlowLayout>
</DetailComponent>
```

Das obige Beispiel erzeugt folgende Aktion:



### 1.10.6 Aktion "Datei herunterladen"


Die Aktion **Datei herunterladen** wird über das Element `<DownloadAction/>` innerhalb von `<Actions>` erzeugt. Mit der Aktion **Download** können Daten aus der Web-Anwendung heruntergeladen werden.

ⓘ Die Attribute `fileId` und `fileName` sind für das Element `<DownloadAction/>` erforderlich.

Zusätzlich zu den Standardattributen für **Aktionen** kann das Element `<DownloadAction/>` noch folgende Attribute haben:

Attribut	Beschreibung
data	<div> <span>⚠</span> Nur verwenden, wenn eine Base64-Datei zum Download angeboten wird!         </div> <p>Gibt an, welche Datei heruntergeladen werden soll.</p> <p><b>Mögliche Werte:</b> Zeichenkette für Data Binding</p> <div> <span>ⓘ</span> Der Data-Binding-Ausdruck muss auf eine Property vom Typ <code>base64</code> verweisen!         </div>
fileId	<p><i>Erforderlich.</i> Identifikator, der vom Prozessentwickler verwendet werden kann.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich.</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>

Attribut	Beschreibung
fileName	<p><i>Erforderlich.</i> Name der Datei, die heruntergeladen werden soll. Wird auch als Dateiname der heruntergeladenen Datei beim Benutzer verwendet.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>

-  Die Aktion "Datei herunterladen" unterstützt nicht nur den Download von Daten aus einem Prozess, sondern es können auch Propertys vom Typ Base64 heruntergeladen werden. Statt des Attributs `process` muss das Attribut `data` verwendet werden.

#### Beispiel Base64-Datei herunterladen




```
<Property name="data" type="Base64"/>
...
<DownloadAction fileId="tst" fileName="test.ext" data="#data"/>
```

### 1.10.7 Benutzerdefinierte Aktion

Die benutzerdefinierte Aktion wird über das Element `<CustomAction/>` innerhalb von `<Actions>` erzeugt. Mit einer benutzerdefinierten Aktion können einfache Aktionen ohne vordefinierten Text und vordefiniertes Icon umgesetzt werden, z. B. kann eine Komponente aufgerufen werden.

Zusätzlich zu den Standardattributen für [Aktionen](#) kann das Element `<CustomAction/>` noch folgende Attribute haben:

Attribut	Beschreibung
displayName	<p>Beschriftung des Buttons.</p> <ul style="list-style-type: none"> <li>• Übersetzbar</li> </ul> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette oder Sprachschlüssel</p>

Attribut	Beschreibung
externalLink	<p>Ermöglicht die Navigation zu einer externen Website, indem ein externer Link festgelegt wird.</p> <ul style="list-style-type: none"> <li>• Data Binding möglich</li> </ul> <p><b>Mögliche Werte:</b> Zeichenkette (URL)</p> <div> <p> Der externe Link kann auch von einem Prozess übergeben werden. Dazu muss mit der Aktion ein Prozess (Attribut process) verknüpft werden, der in seiner Antwort den externen Link enthält:</p> <pre>&lt;Ok externalLink="http://www.google.de" /&gt;</pre> </div>
externalLinkTarget	<p>Gibt an, ob der Link (externalLink) im selben oder in einem neuen Tab geöffnet wird.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• same: Der Link wird im selben Tab geöffnet</li> <li>• new: Der Link wird in einem neuen Tab geöffnet</li> </ul>
validated	<p>Ermöglicht eine Validierung der Eingabe</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• true: die Validierung ist aktiviert <ul style="list-style-type: none"> <li> Schlägt die Validierung fehl, wird die Schaltfläche deaktiviert.</li> </ul> </li> <li>• false: die Validierung ist nicht aktiviert (Standard) <ul style="list-style-type: none"> <li> Schlägt die Validierung fehl, wird die Schaltfläche nicht deaktiviert.</li> </ul> </li> </ul>

### Beispiel

#### Beispiel Aktion "CustomAction"


```
<DetailComponent path="Dashboard" displayName="Dashboard" default="true">
  <Actions>
    <CustomAction componentName="CountryList"
      displayName="Save custom"
      validated="true"
      process="/WebApp/Administration/Country/CreateOrUpdateCountry.wrf"/
    >
  </Actions>
  <FlowLayout>
    ...
  </FlowLayout>
</DetailComponent>
```

Das obige Beispiel erzeugt die benutzerdefinierte Aktion *Save customer*:


The screenshot shows a user interface with three buttons at the top: 'Neu' (with a plus icon), 'Löschen' (with a trash icon), and 'Save custom'. Below the buttons is a list of countries: 'Land', 'Amerika', 'Brasilien', 'Deutschland', and 'Dänemark'. The list is scrollable. Below the list is a search bar with the label 'Land' and the text 'Spain' entered.

### 1.10.8 Aktion "Neu laden"


Die Aktion **Neu laden** wird über das Element `<ReloadAction/>` innerhalb von `<Actions>` erzeugt. Mit der Aktion **Neu laden** werden die Properties einer Komponente periodisch neu geladen (über den in `process` angegebenen Prozess) oder eine Navigation ausgelöst (angegeben über `componentName`).

 Das periodische Neuladen von Daten kann eine hohe Last verursachen und sollte nur in seltenen Anwendungsfällen verwendet werden.

Das Element `<ReloadAction/>` kann folgende Attribute haben:

Attribut	Beschreibung
<code>componentName</code>	<p>Name der Komponente, zu der am Ende der Aktion navigiert werden soll.</p> <div style="border: 1px solid #add8e6; padding: 5px; margin: 10px 0;"> <p> Erforderlich, wenn das Attribut <code>process</code> nicht gesetzt wurde.</p> </div> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
<code>enabled</code>	<p>Steuert, ob das Neuladen periodisch gestartet wird oder nicht.</p> <ul style="list-style-type: none"> <li>• Data Binding (Boolean) möglich</li> </ul> <p><b>Mögliche Werte:</b> <code>true/false</code> oder Ausdruck für Data Binding</p>



Attribut	Beschreibung
process	<p>Pfad zur .wrf-Datei. Der Prozess muss im Ordner Services/Processes enthalten sein. Pfadangabe relativ zum Ordner Services/Processes.</p> <div>  Erforderlich, wenn das Attribut componentName nicht gesetzt wurde. </div> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p>
refreshTime	<p>Zeit bis zum erneuten Laden in Sekunden</p> <p><b>Mögliche Werte:</b> Beliebige ganze Zahl</p>

## 1.11 Navigation innerhalb einer Web-Anwendung

Innerhalb einer Web-Anwendung gibt es verschiedene Möglichkeiten zu navigieren, also zwischen verschiedenen Komponenten zu wechseln. Standardmäßig wird aus den Anzeigenamen (`displayName`) der Komponenten das Menü erstellt. Über dieses Menü können Benutzer zu den entsprechenden Komponenten gelangen.

Weitere Möglichkeiten der Navigation sind die Definition einer benutzerdefinierten Aktion und das Überschreiben der `select`-Aktion in der Liste eines Master-Strukturelements innerhalb einer Master/Detail-Komponente. Mit diesen beiden Möglichkeiten kann auch innerhalb einer Master/Detail-Komponente zu Subkomponenten navigiert werden.

### 1.11.1 Navigation mit einer Aktion

Das folgende Beispiel zeigt, wie eine Aktion definiert wird, die zur Komponente mit dem `componentName` `name` navigiert.

```
<CustomAction displayName="Navigate" componentName="name" />
```

Die Aktion in diesem Beispiel navigiert unabhängig davon, ob in der Master/Detail-Komponente etwas ausgewählt ist. Das Attribut `selectionNeeded` kann verwendet werden, um die Aktion an eine Auswahl zu binden:

```
<CustomAction displayName="Navigate" componentName="name" selectionNeeded="true" />
```

### 1.11.2 Navigation durch Überschreiben der `select`-Aktion

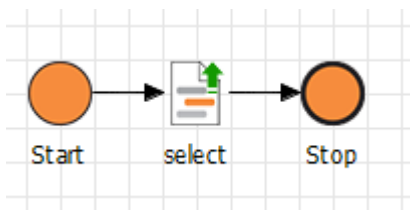
Das folgende Beispiel zeigt eine Master/Detail-Komponente mit überschriebener `select`-Aktion im Master-Strukturelement:

```

<MasterDetailComponent>
  <Properties>
    <Property name="List" type="Complex">
      <Property name="Value" type="String" />
    </Property>
  </Properties>
  <Master>
    <ListComponent name="list" path="list" process="fillListProcess.wrf">
      <Actions>
        <SelectAction process="select.wrf" /> <!-- Überschriebene select-
Aktion -->
      </Actions>
    </ListComponent>
  </Master>
  <Detail>
    <DetailComponent name="Detail" path="Detail">
      <FlowLayout>
        <Label value="#List.Value" />
      </FlowLayout>
    </DetailComponent>
  </Detail>
</MasterDetailComponent>

```

**select.wrf:**



**select.xml:**

```

<?xml version="1.0" encoding="UTF-8" ?>
<ok componentName="Detail2" />

```

In diesem Beispiel wird die Detail2-Komponente immer als Detail-Komponente für die Auswahl in der Master/Detail-Komponente verwendet. Der Prozess, der an die überschriebene select-Aktion gebunden ist, gibt eine Antwort zurück, die das gegebene Attribut componentName enthält. Dieses gibt die Anweisung, zur entsprechenden Komponente zu navigieren.

Dieses Beispiel ist sehr einfach gehalten, es ist möglich komplexe Mappings und Bedingungen in den Prozess einzubauen, der die Antwort zurückgibt, um ein Detail für die Anzeige in jedem einzelnen Fall auszuwählen.

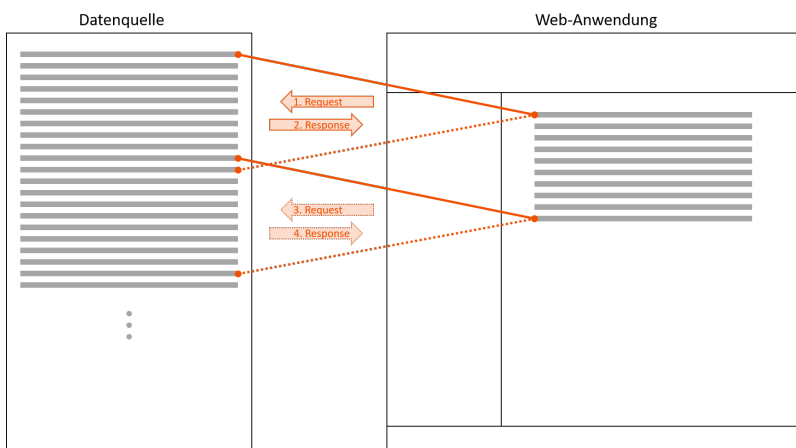
Das Beispiel der [Master/Detail-Komponente mit Projektstruktur](#) findet sich im entsprechenden Abschnitt.

Die [Attribute der Komponenten](#), [Aktionen](#), [Properties](#) und der [Data-Binding-Mechanismus](#) finden sich ebenfalls in den entsprechenden Abschnitten.

## 1.12 Paging

Web-Anwendungen, die mit X4 Activities Web Apps erstellt werden, unterstützen in der Master/Detail- und der List-Komponente automatisch Paging. Deshalb müssen die Prozesse, die für diese Komponenten Daten liefern, ebenfalls Paging unterstützen und Informationen darüber verarbeiten, wie viele Objekte aus welchem Bereich ausgelesen werden sollen.

Beim Paging werden Daten nur in Teilmengen nach und nach aus der Datenquelle geladen. So können große Datenströme vermieden werden, die von der Datenquelle oder dem Browser nur sehr schwer verarbeitet werden können. Im Gegensatz zu herkömmlichen Web-Anwendungen ist es mit Web-Anwendungen, die mit X4 Activities Web Apps erstellt wurden möglich, die neuen Daten während des Scrollens nachzuladen. Der Benutzer muss nicht explizit auf die nächste Seite umblättern.




### 1.12.1 Beispielanfrage

Die Komponente, die mit einem Prozess verknüpft ist, sendet Paging-Anfragen, die verarbeitet werden können. Dabei wird mit `offset` festgelegt, ab welchem Objekt ausgelesen werden sollen. `limit` legt fest, wie viele Objekte abgefragt werden sollen.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Search offset="0" limit="50">
  <OrderBy />
  <Where />
</Search>
```

Attribut	Beschreibung
offset	Gibt den Offset der Datenabfrage an, also ab welchem Datensatz die Datenquelle ausgelesen werden soll.  <b>Mögliche Werte:</b> Ganzzahl

Attribut	Beschreibung
limit	<p>Anzahl der Datensätze, die in der Antwort enthalten sein sollen.</p> <p><b>Mögliche Werte:</b> Ganzzahl größer als 0</p> <div>  Der Wert dieses Attributs wird automatisch vom Attribut pageSize der List-Komponente hinterlegt! </div>

### 1.12.2 Beispielantwort


Die Antwort, die vom Prozess zurückgegeben wird, muss außer den angefragten Objekten zusätzlich Informationen darüber enthalten, wie viele Objekte insgesamt in der Liste enthalten sind (size-Attribut), damit klar ist, ob noch weitere Objekte abgefragt werden müssen.

```
<OkList size="2">
  <List>
    <Description>Value</Description>
    <Id>0</Id>
  </List>
  <List>
    <Description>Value</Description>
    <Id>1</Id>
  </List>
</OkList>
```

Attribut	Beschreibung
size	<p>Anzahl der insgesamt in der Datenquelle vorhandenen Datensätze.</p> <p><b>Mögliche Werte:</b> Ganzzahl</p>

### 1.12.3 Beispiel für Paging

Folgendes Beispiel enthält eine List-Komponente mit einem Prozess und einer Properties-Definition.

 Das folgende Beispiel ist nur ein Beispiel, um das Prinzip von Paging zu veranschaulichen, es darf nicht als Best Practice angesehen werden.

```

<ListComponent name="Paging" path="Paging" displayName="Paging" process="fillListProc
ess.wrf" default="true">
  <Properties>
    <Property name="List" type="Complex">
      <Property name="Description" type="String" displayName="Description"/>
      <Property name="Id" type="Integer" displayName="Id"/>
    </Property>
  </Properties>
  <Columns>
    <Column value="#List.Id" />
    <Column value="#List.Description" />
  </Columns>
</ListComponent>

```

In diesem Beispiel gibt es eine XML-Datei als Datenquelle, die 67 Einträge enthält.:

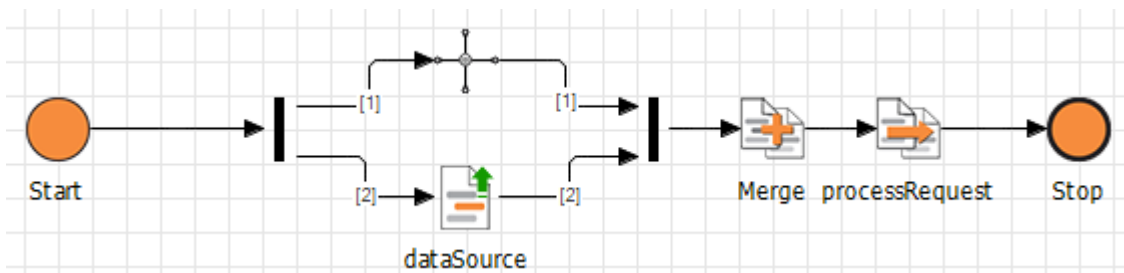
#### dataSource.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<List>
  <Description>Value</Description>
  <Id>0</Id>
  ...
  <Description>Value</Description>
  <Id>66</Id>
</List>

```

Um die Anfrage zu verarbeiten und nur die Einträge zu erhalten, die benötigt werden, muss ein Prozess entworfen werden:



Zunächst werden in diesem Prozess die Datenquelle und die Anfrage zusammengefügt, damit sie im Folgenden in einem Mapping verarbeitet werden können.

Das Mapping sieht folgendermaßen aus:

**processRequest.xsl**

```

<?xml version="1.0" encoding="UTF-8"?>
<?xsl:source ../TemporaryFiles/temp2.xml#Merge?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output media-type="text/xml" method="xml"/>
  <xsl:template match="/">
    <OkList>
      <xsl:attribute name="size" select="count(Merge/List/Id)"/>
      <xsl:for-each select="Merge/List/Id">
        <xsl:if test="number(Id)>=/Merge/Search/@offset and number(Id)< /Merge/
Search/@offset+/Merge/Search/@limit">
          <List>
            <Description>
              <xsl:value-of select="Description"/>
            </Description>
            <Id>
              <xsl:value-of select="Id"/>
            </Id>
          </List>
        </xsl:if>
      </xsl:for-each>
    </OkList>
  </xsl:template>
</xsl:stylesheet>

```

Das Mapping verarbeitet die Angaben `offset` und `limit` aus der Anfrage (`/Merge/Search/@offset` bzw. `/Merge/Search/@limit`) und sortiert die Einträge aus, die diese Bedingungen nicht erfüllen. Für die Einträge, die die Bedingungen erfüllen, wird ein Datenmodell mit `<OkList>` als Wurzelement generiert, die von Web App gelesen werden kann. Die gesamte Anzahl der Einträge in der Liste wird mit dem Attribut `size` zu `<OkList>` hinzugefügt. Wenn nun in der List-Komponente gescrollt wird, weiß die Komponente, dass es mehr Einträge gibt, als angezeigt werden. Diese Einträge werden mit einer neuen Anfrage mit anderem Offset geladen.


- ☐ list?limit=50&offset=0
- ☐ Materialscore-Regular 370ab438796a23ba05...
- ☐ Version:1.0
- ☐ Materialscore-Regular 370ab438796a23ba05...
- ☐ Materialscore-Regular 370ab438796a23ba05...
- ☐ list?limit=50&offset=50

## 1.13 Sortieren

Tabellen innerhalb einer Web-Anwendung, die mit X4 Activities Web Apps erstellt wurde, können nach den Werten innerhalb einer Spalte sortiert werden.

Am Beispiel, das bereits für [Paging](#) verwendet wurde, werden einige Änderungen vorgenommen, um sortieren zu können. Zunächst muss die Spalte in der List-Komponente als sortierbar markiert werden (`sortable="true"`).

```
<ListComponent name="Sorting" path="Sorting" displayName="Paging" process="fillListProcess.wrf" default="true">
  <Properties>
    <Property name="List" type="Complex">
      <Property name="Description" type="String" displayName="Description"/>
      <Property name="Id" type="Integer" displayName="Id"/>
    </Property>
  </Properties>
  <Columns>
    <Column value="#List.Id" sortable="true" />
    <Column value="#List.Description" />
  </Columns>
</ListComponent>
```

Die Spaltenüberschrift sieht nun so aus:  **Id**. Das Zeichen bedeutet, dass der Sortiermechanismus aktiviert ist. Beim Klicken auf die Spaltenüberschrift ändert sich aber noch nichts, die Liste wird nur ohne Sortierung neu geladen.

Für die Sortierung sind neue Anfragen notwendig:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Search offset="0" limit="50">
  <OrderBy>
    <Asc property="#List.Id" />
  </OrderBy>
  <Where />
</Search>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Search offset="0" limit="50">
  <OrderBy>
    <Desc property="#List.Id" />
  </OrderBy>
  <Where />
</Search>
```

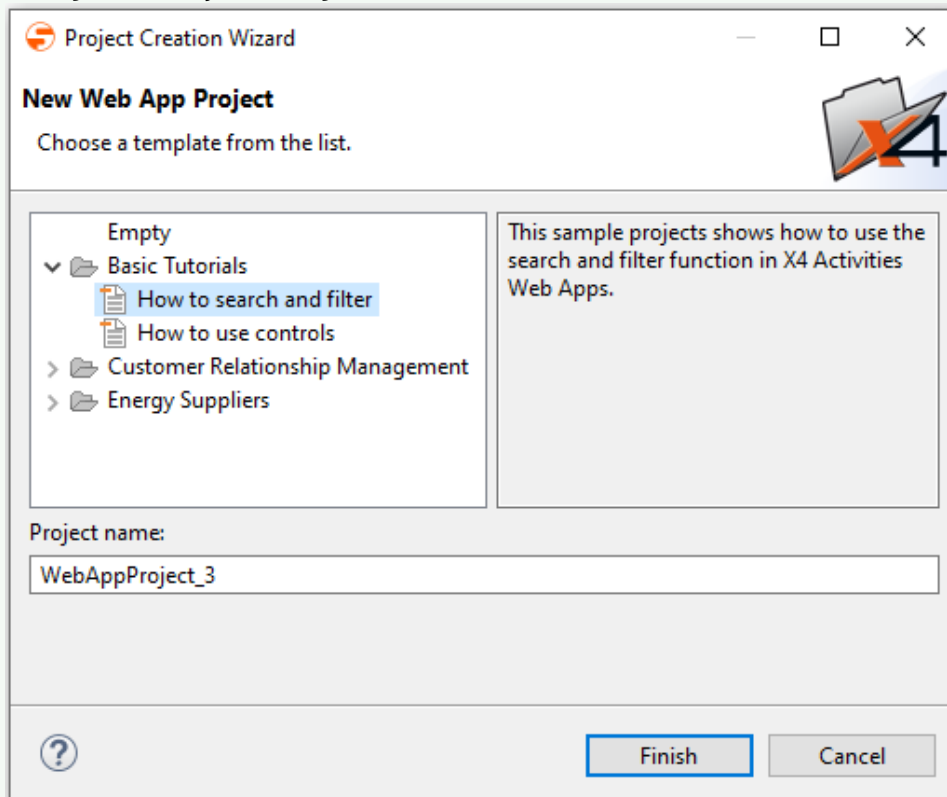
Die Tags <Asc> und <Desc> stehen für aufsteigende (*ascending*) bzw. absteigende (*descending*) Sortierung. Durch ihre Verwendung können Anfragen, Mappings usw. erstellt werden, um Daten so zu bekommen, das sie der Sortiermechanismus verwendet, und so sortierte Listen zu unterstützen.

## 1.14 Suchen und Filtern

In einer Web-Anwendung, die mit X4 Activities Web Apps erstellt wurde, kann eine Such- bzw. Filterfunktion sehr einfach realisiert werden. Damit kann innerhalb von List-Komponenten (allein oder innerhalb einer Master/Detail-Komponente) nach bestimmten Einträgen gesucht bzw. gefiltert werden.

Das Web-Apps-Tutorial **How to search and filter** zeigt die Umsetzung der Such- und Filterfunktion in X4 Activities Web Apps anhand eines einfachen Beispiels.

- ✓ Um das Tutorial zu nutzen, legen Sie ein neues Web-Apps-Projekt an und wählen aus den verfügbaren Projektvorlagen **Basic Tutorial > How to search and filter**.



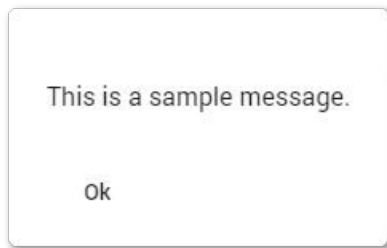
## 1.15 Meldungen

Innerhalb einer Web-Anwendung, die mit X4 Activities Web Apps erstellt wurde, können Meldungen ausgegeben werden. Dazu können in der XML-Datei, die die Antwort enthält Message und Error Elemente verwendet werden. Diese Meldungen werden als Dialogfenster an der entsprechenden Stelle angezeigt.

- ⓘ Die XML-Datei, die die Meldung enthält, muss durch einen Prozess geliefert werden. Dieser Prozess kann z. B. mit einer [benutzerdefinierten Aktion](#) aufgerufen werden.

```
<?xml version="1.0" encoding="UTF-8"?>
<ok>
  <Message>This is a sample message.</Message>
</ok>
```



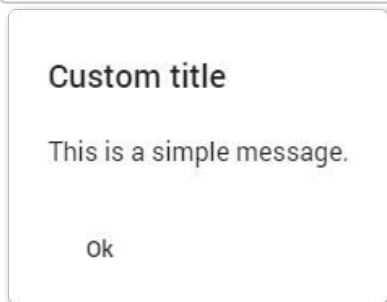


Meldungen können lokalisiert werden, indem als Inhalt der Meldung ein Sprachschlüssel hinterlegt wird:

```
<?xml version="1.0" encoding="UTF-8"?>
<Ok>
  <Message>$translationKey</Message>
</Ok>
```

Mit dem optionalen Attribut title kann dem Dialogfenster, das in der Web-Anwendung ausgegeben wird, ein Titel zugewiesen werden:

```
<?xml version="1.0" encoding="UTF-8"?>
<Ok>
  <Message title="Custom title">This is a simple message.</Message>
</Ok>
```



Wird statt dem Element `<Ok>` das Element `<Error>` verwendet, dann wird die Meldung als Fehlermeldung ausgegeben. Bei Fehlermeldungen wird als Titel „Fehler“ bzw. die entsprechende Übersetzung angezeigt. Für Fehlermeldungen kann kein benutzerdefinierter Titel vergeben werden.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Message>This is a sample error.</Message>
</Error>
```

**Fehler**

This is a simple message.

Ok

## 2 X4 Activities Classic

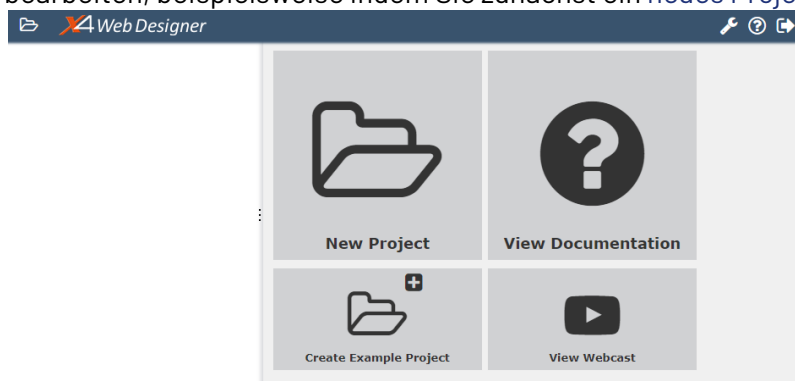
Mit X4 Activities Classic können Web-Anwendungen erstellt werden, mit denen Menschen in Prozesse eingebunden werden können. Der X4 Web Designer ist ein Web-basiertes Werkzeug, um direkt im Browser Web-Anwendungen auf Basis von X4 Activities zu erstellen und Datenmodell, Navigator, und Detailansichten in grafischen Editoren auf Basis von JavaScript, CSS und HTML zu definieren.

### Inhalt

### 2.1 X4 Web Designer

Wie Sie den *X4 Web Designer* im Browser aufrufen, um damit Web-Anwendungen zu entwickeln

1. *X4 Designer* starten.
2. In der Menüleiste des *X4 Designers* auf **Tools > X4 Web Designer** klicken.  
→ Der Browser öffnet sich und der Login-Bildschirm des X4 Web Designers wird angezeigt.
3. Zugangsdaten für die Web-Anwendung *X4 Web Designer* eingeben.  
Nach erfolgreicher Anmeldung können Sie Web-Anwendungen grafisch erstellen und bearbeiten, beispielsweise indem Sie zunächst ein [neues Projekt erstellen](#).

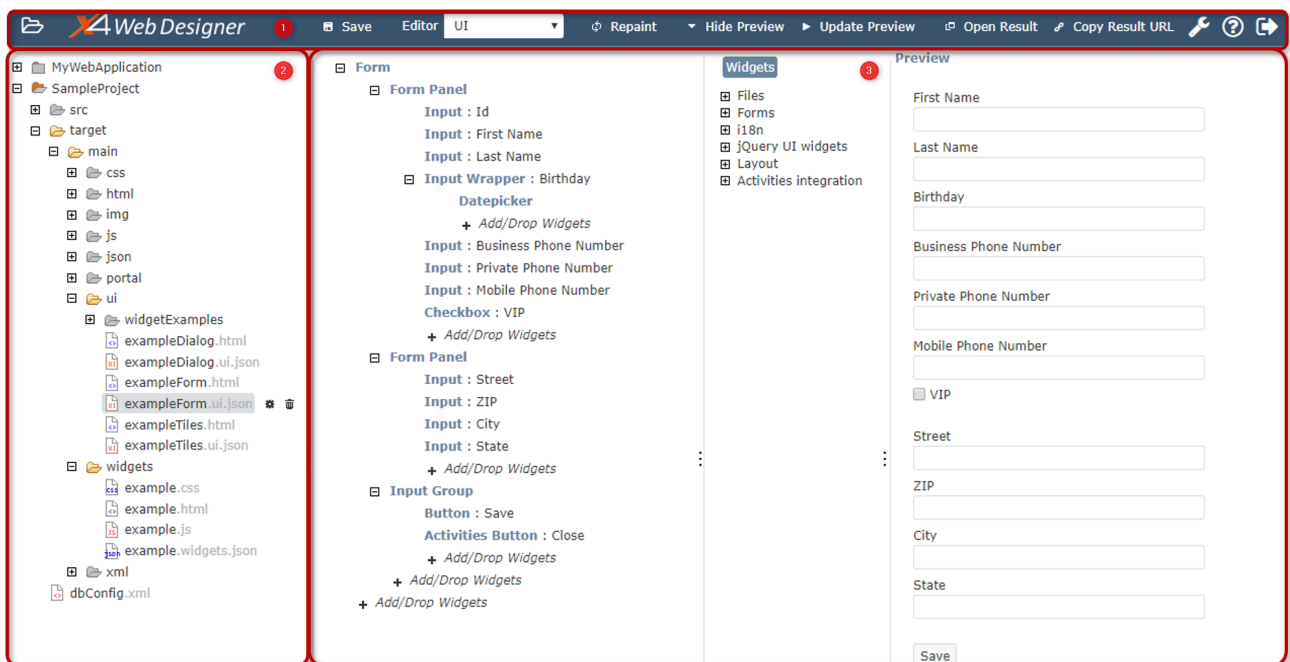


**i** Sie können auf den *X4 Web Designer* nicht zugreifen?

- Stellen Sie sicher, dass *X4 Activities* und der *X4 Web Designer* korrekt installiert sind, siehe [2019-02-20\\_12-05-59\\_X4 Activities installieren](#).
- Prüfen Sie im *X4 Designer* in der Konfigurationsdatei `X4WebIDE_Files/Config/main.xml`, ob der *X4 Web Designer* möglicherweise über eine URL bereitgestellt wurde, die von der Standardeinstellung abweicht, siehe [2019-02-20\\_12-03-58\\_main.xml: URL-Pfad-Konfiguration zum Zugriff auf den X4 Web Designer](#).

### 2.2 Benutzeroberfläche des X4 Web Designers







Wie die Benutzeroberfläche des *X4 Web Designers* aufgebaut ist und welche Editoren enthalten sind. Die Benutzeroberfläche des X4 Web Designers passt sich dem jeweiligen Arbeitskontext an. Grundlegend können jedoch folgende Bereiche unterschieden werden.




## 2.2.1 Menüleiste

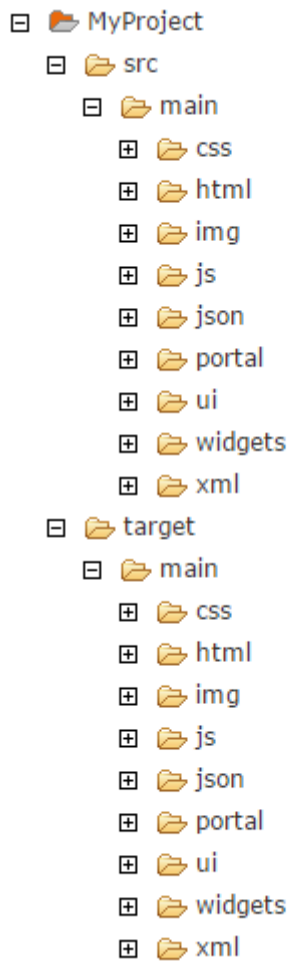
Die **Menüleiste** (1) zeigt zum aktuell geöffneten Editor jeweils kontextabhängige Bedienelemente an.

Element	Titel	Erläuterung
	<b>Show/ hide navigati on</b>	Navigator ein-/ausblenden
	<b>Save</b>	Änderungen des Editors speichern  alternativ Tastenkürzel <b>Strg+S</b> drücken
	<b>Reload</b>	Aktuell im Editor geöffnete Datei neu laden bzw. auf den zuletzt gespeicherten Stand zurücksetzen
	<b>Format</b>	Code formatieren (Einrückungen etc.)
	<b>Run/ Debug</b>	JavaScript-Code ausführen/testen
	<b>Editor</b>	Zwischen verschiedenen Editor-Ansichten wechseln
	<b>Repaint</b>	Darstellung des UI-Editors aktualisieren und Daten der Vorschau zurücksetzen
	<b>Open Preview</b>	Vorschau-Bereich des Editors anzeigen
	<b>Hide Preview</b>	Vorschau-Bereich des Editors ausblenden

	<b>Update Preview</b>	Vorschau aktualisieren (sofern die Vorschau sichtbar ist)  Damit wird die Benutzeroberfläche neu gerendert wird; Daten und Zustandsinformationen bleiben jedoch erhalten.
	<b>Open Result</b>	Ergebnis in einer neuen Browser-Registerkarte zeigen
	<b>Copy Result URL</b>	Ergebnis-URL in die Zwischenablage speichern  <i>Beispiel:</i> Im UI-Editor diese URL kopieren, um sie im <i>Datasource Config Editor</i> als Detailseite zu hinterlegen.
	<b>Open Settings Dialog</b>	X4 Web Designer Einstellungen öffnen, siehe <a href="#">Navigator-Einstellungen</a>
	<b>Open User Manual</b>	Dokumentation zum X4 Web Designer im PDF-Format öffnen
	<b>Logout User</b>	Aktuell angemeldeten Benutzer abmelden













## 2.2.2 Navigator


Der **Navigator (2)** wird auf der linken Seite des *X4 Web Designers* angezeigt, sofern Sie diesen über die Schaltfläche  **Show navigation** in der Menüleiste eingeblendet haben. In einer Baumstruktur werden dann die Inhalte des aktuellen Repository-Ordners dargestellt. Zudem werden zu jedem Navigatorknoten oder Inhalt innerhalb eines Ordners unterschiedliche Funktionen bereitstellt.




Je nach Navigator-Knoten oder Ordner-Inhalt stehen folgende Bearbeitungsoptionen zur Verfügung:

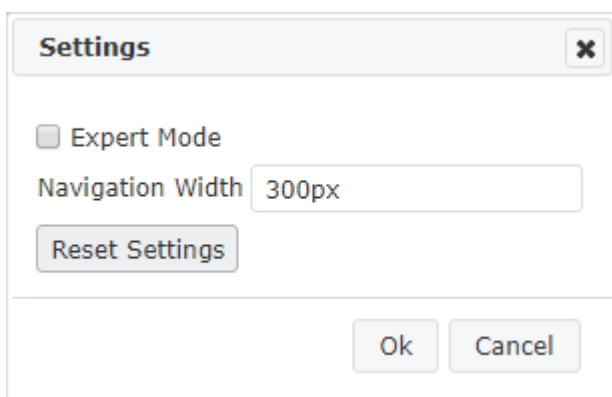
Element	Titel	Erläuterung
	<b>Add file or folder</b>	<p>Neue Datei oder neuen Ordner anlegen:</p> <ul style="list-style-type: none"><li>• <b>Add file:</b> Neue leere Datei im aktuellen Ordner anlegen</li><li>• <b>Add folder:</b> Neuen Ordner anlegen</li></ul> <div> Wenn eine Datei ohne Dateiendung angelegt wird, dann wird automatisch die Dateiendung *.txt an den Dateinamen angehängt.</div>

	<b>Open context menu</b>	Kontextmenü mit weiteren Optionen öffnen: <ul style="list-style-type: none"> <li>•  <b>Refresh:</b> Ordnerinhalte neu laden</li> <li>•  <b>Buid:</b> Ordnerinhalte im entsprechenden target-Ordner neu bauen</li> <li>•  <b>Upload files:</b> Dateien hochladen</li> <li>•  <b>Download:</b> Ordnerinhalt herunterladen</li> <li>•  <b>Rename:</b> Datei oder Ordner umbenennen</li> <li>•  <b>Duplicate:</b> Datei oder Ordner duplizieren</li> <li>•  <b>Copy:</b> Datei oder Ordner kopieren</li> <li>•  <b>Paste:</b> Kopierten Inhalt einfügen</li> <li>•  <b>Open plain:</b> Aktuelle Quelldatei anzeigen</li> <li>•  <b>Open resource:</b> Die ReST-URL, unter der die Datei im Projekt verfügbar ist, öffnen</li> <li>•  <b>Show resource URL:</b> URL der Ressource anzeigen</li> </ul>
	<b>Delete folder/file</b>	Ordner oder Datei löschen

 Für mehr Informationen, wie Sie mit Dateien und Ordnern im Navigator arbeiten, siehe [Ordner und Dateien im Navigator verwalten](#).

### 2.2.2.1 Navigator-Einstellungen

Über  **Open Settings Dialog** im Header des X4 Web Designers lassen sich Einstellungen zur Anzeige des Navigators vornehmen. Diese Einstellungen werden im Local-Storage des Browsers gespeichert und sind somit für alle Fenster/Tabs des Browsers gültig. Nach Bestätigung der vorgenommenen Änderungen wird die aktuelle Seite automatisch neu geladen.



#### 2.2.2.1.1 Expertenmodus

Um die Arbeit mit dem X4 Web Designer zu vereinfachen, werden selten benötigte Inhalte standardmäßig ausgeblendet. So ist im Navigationsbaum unter einem Projekt im Standardmodus nur der Inhalt des Verzeichnisses `src/main` sichtbar, sonstige Verzeichnisse wie `target` werden nicht angezeigt. Zudem werden Dateiendungen ausgeblendet, um den Baum übersichtlicher zu halten. Bei Bedarf kann der vollständige Dateiname aber in einem Tooltip angezeigt werden.

Der Expertenmodus kann mit Hilfe der entsprechenden Checkbox **Expert Mode** im Dialog aktiviert werden.

### 2.2.2.1.2 Breite des Navigationsbereichs

Die Standard-Breite des Navigationsbereichs im X4 Web Designer beträgt 300 Pixel. Mit der Einstellung **Navigation Width** kann dieser Wert durch eine beliebige CSS-Längenangabe ersetzt werden. Dadurch kann das Standard-Layout an individuelle Bedürfnisse abhängig von Projektstrukturen und Bildschirmauflösung angepasst werden.

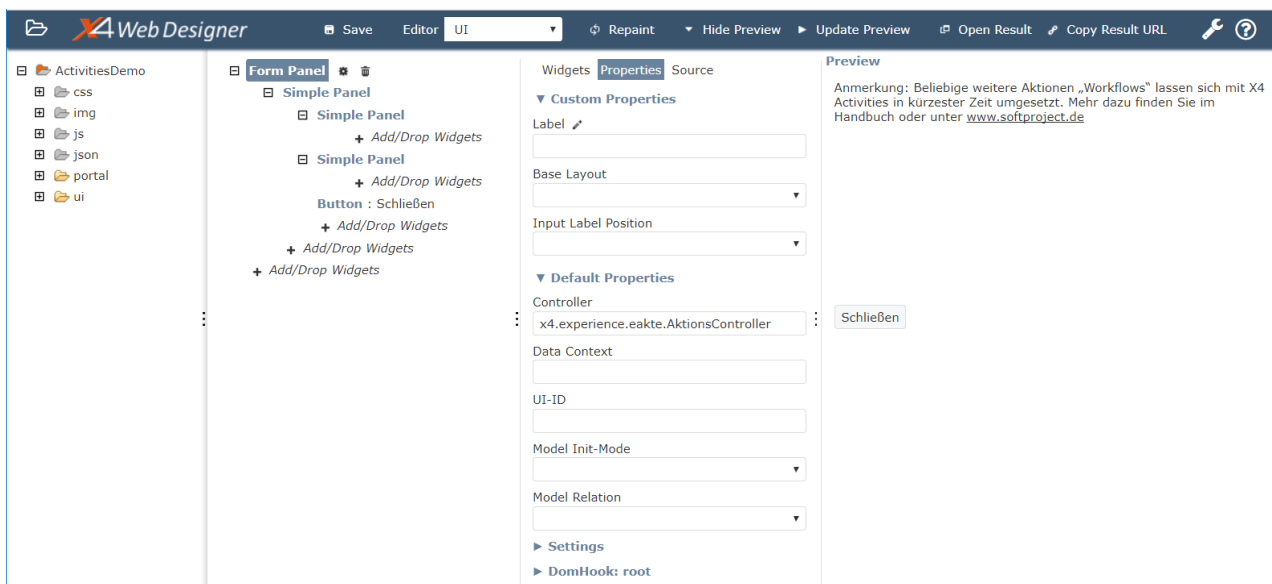
## 2.2.3 Editor-Bereich

Je nach geöffnetem Dateityp stehen im **Editor-Bereich (3)** unterschiedliche grafische Editoren (ggf. mit Vorschaubereich) oder Quelltextansichten zur Verfügung.

## 2.2.4 UI-Editor im X4 Web Designer

Wie der UI-Editor im *X4 Web Designer* aufgebaut ist und welche Funktionen Sie in dessen Benutzeroberfläche nutzen können

### 2.2.4.1 Grafischer Strukturbaum



Der Strukturbaum ist die zentrale Arbeitsfläche im UI-Editor. Hier können Sie UI-Widgets aus der **Widgets**-Palette per Drag & Drop an beliebiger Position einfügen oder per Klick auf **+ Add/Drop Widget** hinzufügen, siehe [UI-Widgets einfügen und verwalten](#).

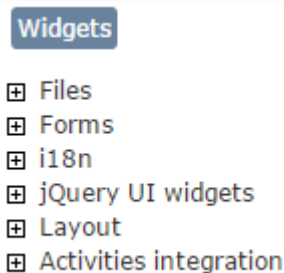



### ✓ Tipps für das Arbeiten mit dem X4 Web Designer

- Um noch mehr Platz für den Strukturbaum zu erhalten:
  - Navigator-Bereich ausblenden: In der Menüleiste auf  **Hide navigation** klicken, siehe [Benutzeroberfläche des X4 Web Designers](#).
  - Vorschau-Bereich ausblenden: In der Menüleiste auf  **Hide Preview** klicken.
- Zusätzliche Menüs bei Auswählen eines Knotens in der Strukturansicht:
  - Über  **Open context menu** lässt sich ein Kontextmenü öffnen, um Knoten zu duplizieren, zu kopieren, auszuschneiden und einzufügen
  - Über  **Remove** lässt sich der ausgewählte Knoten löschen
- Knoten können über Tabs hinweg per Drag & Drop aus einem Editor in einen anderen Editor kopiert werden
- Mit Strg + Drag & Drop können Knoten dupliziert werden

#### 2.2.4.2 Widgets-Palette

Die **Widgets**-Palette bietet Ihnen die Möglichkeit, UI-Widgets per Drag & Drop in den Strukturbaum zu ziehen bzw. einzufügen, siehe [UI-Widgets einfügen und verwalten](#).



UI-Widgets sind in folgende Standardkategorien aufgeteilt, die Sie mit Klick auf  ausklappen:

- **Files:** Enthält alle Widgets, um einen Upload für eine oder mehrere Dateien zu erstellen, siehe [File-Widgets](#)
- **Forms:** Enthält alle Widgets zum Erstellen von Web-Formularen, siehe [Formular-Widgets](#).
- **i18n:** Enthält Widgets für die Konfiguration und Bereitstellung von Lokalisierungs-Inhalten sowie für die Validierung von Formularen, siehe [Internationalisierungs-Widgets](#)
- **jQuery UI:** Enthält zahlreiche Widgets mit Funktionen der JavaScript-Bibliothek *jQuery UI* (z.B. zur Datumsauswahl), siehe [jQuery UI-Widgets](#).
- **Layout:** Enthält alle standardmäßig verfügbaren Layout-Widgets (z.B. Tabellen oder Frames), siehe [Layout-Widgets](#).
- **Activities integration:** Enthält Widgets, die den einfachen Zugriff auf Funktionen eines Activities-Portals innerhalb einer Detail-Seite erlauben, siehe [Activities-Integration-Widgets](#)

#### 2.2.4.3 Properties-Bereich

Wenn Sie im Strukturbaum des UI-Editors ein UI-Widget markiert haben, so werden im **Properties**-Bereich dessen Eigenschaften angezeigt und können bearbeitet werden.

Widgets

Properties

Source

Custom & Default Properties ▼

Label

Action

Default Properties

Widget Mode

FormPart ▼

Controller

Data Context

UI-ID

Model Init-Mode


▼

Model Relation

▼


Im **Properties**-Bereich können Sie über ein Auswahlménü zwischen folgenden Ansichtsoptionen wählen:



Ansichtsoption	Erläuterung
<b>Custom &amp; Default Properties</b>	Alle Eigenschaften des markierten Widgets anzeigen (Standard)
<b>Custom Properties</b>	Nur Eigenschaften anzeigen, die für das markierte Widget spezifisch sind, siehe: <ul style="list-style-type: none"> <li>• <a href="#">File-Widgets</a></li> <li>• <a href="#">Formular-Widgets</a></li> <li>• <a href="#">Internationalisierungs-Widgets</a></li> <li>• <a href="#">jQuery UI-Widgets</a></li> <li>• <a href="#">Layout-Widgets</a></li> <li>• <a href="#">Activities-Integration-Widgets</a></li> </ul>
<b>Default Properties</b>	Nur Standard-Eigenschaften des markierten Widgets anzeigen, siehe <a href="#">Allgemeine Widget-Eigenschaften</a>

Ansichtsoption	Erläuterung
<b>Settings</b>	<p>Inhalte über ein UI-Binding als <code>readOnly</code> definieren.</p> <div> <p> Liefert das UI-Binding <code>true</code>, werden alle Eingabefelder unterhalb dieses UI-Knotens als <code>readOnly</code> festlegt, sofern die Eigenschaft nicht in einem untergeordneten Knoten überschrieben wird. Diese Eigenschaft wird von allen Widgets des X4 Web Designers automatisch ausgewertet. Für andere Elemente oder komplexere Veränderungen an der Seite, z.B. Ausblenden eines Bereiches, kann die Eigenschaft auch explizit per Binding-Syntax ( <code>{#settings.readOnly}</code> ) oder JavaScript-API ( <code>vc.execBinding('settings.readOnly')</code> ) abgerufen werden.</p> </div>
<b>DomHook: Root</b>	Individuelle Anpassungen (z.B. Bedingungen gemäß der fachlichen Logik) für das aktuelle UI-Widget im <i>Document Object Model</i> (DOM) der Benutzeroberfläche hinterlegen, siehe <a href="#">DomHook-Einträge hinterlegen</a>
<b>DomHook: &lt;Element&gt;</b>	Je nach Widget ggf. weitere Eingriffsmöglichkeiten in das <i>Document Object Model</i> (DOM), die sich jedoch ausschließlich auf ein bestimmtes Element beziehen

#### 2.2.4.4 Source-Bereich

Wenn Sie im Strukturbaum des UI-Editors ein Widget markiert haben, so wird im Source-Bereich dessen Konfiguration im JSON-Speicherformat angezeigt und kann dort auch bearbeitet und gespeichert werden.

 Beachten Sie:

- In der **Source**-Ansicht nach Bearbeitungen auf  klicken, um diese zu übernehmen.
-  klicken, um die **Source**-Ansicht neu zu laden, nachdem Sie in der **Properties**-Sicht Änderungen vorgenommen haben.

#### Beispiel

```
{
  "_widget": "x4.forms.RadioItem",
  "_template": "RadioItem",
  "_renderer": "x4.ui.forms.render.InputWrapper",
  "label": "Option Label",
  "value": "Value",
  "groupName": "myRadioGroup"
}
```

#### 2.2.4.5 Preview-Bereich (Vorschau)

Im **Preview**-Bereich können Sie eine Vorschau für Ihre im UI-Editor modellierte Benutzeroberfläche anzeigen:

<b>Show Preview</b>	<b>Preview-Bereich</b> einblenden (falls dieser ausgeblendet war)
<b>Update Preview</b>	Vorschau anhand der aktuellen Baumstruktur aktualisieren
<b>Repaint</b>	Vorschau mit Struktur und Daten aktualisieren, z.B. bei Darstellungsfehlern

#### 2.2.4.6 Weitere Editor-Ansichten

In der Menüleiste können Sie zwischen folgenden Ansichten wechseln, um die aktuell geöffnete Datei zu bearbeiten:

- **UI**: Die grafische Ansicht zum Modellieren von Benutzeroberflächen im *UI Editor*
- **Dependencies**: Ermöglicht die Definition von Abhängigkeiten wie JavaScript-Bibliotheken, CSS-Dateien oder Vorlagen, siehe unten.
- **JSON**: Quelltextansicht der aktuell bearbeiteten JSON-Datei inklusive aller Widgets und definierten Abhängigkeiten (Dependencies)

#### 2.2.4.7 Dependencies-Ansicht

Benutzeroberflächen, die Sie im UI-Editor des *X4 Web Designers* modelliert haben, können Abhängigkeiten enthalten, d.h. Referenzen auf JavaScript-Bibliotheken, CSS-Dateien, Vorlagen o.ä. In der **Dependencies**-Ansicht des UI-Editors können Sie diese Abhängigkeiten verwalten.

Um die **Dependencies**-Ansicht aufzurufen, im UI-Editor in der Menüleiste bei **Editor** den Eintrag *Dependencies* wählen.

**X4 Web Designer** Save Close Editor Dependencies

**Widget Collections**

URL	X4 dependency		Sync	
/X4/httpstarter/ReST/X4WebResources/base-widgets/widgets/main.wid	X4 dependency	1	<input type="checkbox"/>	Sync

+

**Template Collections**

+

**JavaScript Files**

URL	X4 dependency		Sync	
/X4/httpstarter/ReST/Test-project/Resources/js/example.js	X4 dependency	2	<input checked="" type="checkbox"/>	Sync
/X4/httpstarter/ReST/X4WebResources/lib/jquery-ui/i18n/datepicker-de	X4 dependency	3	<input type="checkbox"/>	Sync

+

**CSS Files**

URL	X4 dependency		Sync	
/X4/httpstarter/ReST/Test-project/Resources/css/example.css	X4 dependency	2	<input checked="" type="checkbox"/>	Sync


+

Für jede Kategorie können Sie beliebig viele URLs hinterlegen, die auf Abhängigkeiten verweisen:

Kategorie	Erläuterung
<b>Widget Collections</b>	<p>Ressourcen-URL für eine <i>Widget Collection</i>-Datei, siehe <a href="#">Eigene Widgets erstellen</a></p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• Beliebige gültige relative oder absolute URL, z.B. /X4/httpstarter/ReST/MyWebApp/src/main/widgets/MyWidgetCollection.widgets.json</li> <li>• Widget Collection von <i>X4 Activities</i>: /X4/httpstarter/ReST/X4WebResources/base-widgets/widgets/main.widgets.json (Standard)</li> </ul>
<b>Template Collections</b>	<p>Ressourcen-URL für eine HTML-Seite, die Templates enthält (<i>Template Collection</i>), siehe API-Dokumentation unter <a href="#">Framework-spezifische HTML-Strukturen</a></p>
<b>JavaScript Files</b>	<p>Ressourcen-URL für ggf. einzubindende JavaScript-Bibliotheksdateien</p>

Kategorie	Erläuterung
CSS Files	Ressourcen-URL für ggf. einzubindende Cascading Stylesheets

**Tipps:**

- **+** klicken, um einen neuen Eintrag hinzuzufügen.
- In **Group** ggf. eine Gruppe zur besseren Übersichtlichkeit angeben.
- **Position** angeben, falls Abhängigkeiten untereinander bestehen und sicher gestellt werden soll, dass die Ressourcen stets in der hier definierten Reihenfolge geladen werden.
- **Sync** aktivieren, wenn die Ressource synchron geladen werden soll.
-  klicken, um einen Eintrag zu löschen.
- **Speichern** klicken, um die Änderungen zu speichern.

**Auf dieser Seite:****Siehe auch:**

- [UI-Widgets einfügen und verwalten](#)
- [Formular-Widgets](#)
- [jQuery UI-Widgets](#)
- [Layout-Widgets](#)
- [Tastenkombination im X4 Web Designer](#)

## 2.3 Tastenkombination im X4 Web Designer

Nützliche Tastatur-Kurzbefehle, die Sie im *X4 Web Designer* nutzen können

Tastenkürzel	Erläuterung
<b>Strg+S</b>	Datei speichern
<b>Strg+Z</b>	Letzten Schritt rückgängig machen
<b>Strg+Y</b>	Rückgängig gemachten Schritt wiederherstellen
<b>Strg+F</b>	Suchen
<b>Strg+R</b>	Ersetzen
<b>Strg+Leer</b>	Autovervollständigung aktivieren
<b>Strg+Shift+F</b>	Code formatieren (Zeilen korrekt umbrechen und einrücken)
<b>Strg+D</b>	Zeile löschen
<b>Strg+Q</b>	Markierten Bereich auskommentieren
<b>Strg+7</b>	Kommentar für markierten Bereich entfernen (einkommentieren)
<b>Alt+R</b>	Run (Ausführung starten)
<b>Strg+5</b>	Debug (Debug-Modus starten)

## 2.4 Projekte und Ressourcen im X4 Web Designer verwalten

Erfahren Sie, wie Sie ein neues Web-Anwendungsprojekt auf Basis von *X4 Activities* erstellen und die entsprechenden Dateiressourcen im *X4 Repository* verwalten.

### 2.4.1 Struktur eines Web-Portal-Projekts

Wenn ein neues Web-Portal-Projekt angelegt wird, geschieht dies in einer bestimmten Struktur:

<div> <div>Example</div> <div>css</div> <div>html</div> <div>img</div> <div>js</div> <div>json</div> <div>portal</div> <div> <div>actionbar</div> <div>datasource</div> <div>main</div> <div>navigator</div> </div> <div>ui</div> <div>widgets</div> <div>xml</div> </div>	<ul style="list-style-type: none"> <li>• <i>Projekt</i> <ul style="list-style-type: none"> <li>• <i>css</i>: CSS-Formatvorlagen</li> <li>• <i>html</i>: Statische HTML-Dateien</li> <li>• <i>img</i>: Grafik-Ressourcen (Logos, Icons, ...)</li> <li>• <i>js</i>: JavaScript-Dateien</li> <li>• <i>json</i>: JSON-Dateien</li> <li>• <i>portal</i>: Portalressourcen für Aktionsleiste, Lister, Main-Config-Datei und Navigator <ul style="list-style-type: none"> <li>• <i>actionbar</i>: Enthält die Datei für die Aktionsleiste (siehe <a href="#">Aktionsleiste</a>).</li> <li>• <i>datasource</i></li> <li>• <i>main</i>: Enthält die Main-Config-Datei</li> <li>• <i>navigator</i>: Enthält eine oder mehrere Navigatordatei(en)</li> </ul> </li> <li>• <i>ui</i>: Enthält die Konfigurationsdateien der Web-Anwendung in jeweils einer Datei mit der Endung <i>.ui.json</i>, (<a href="#">Benutzeroberflächen erstellen</a>)</li> <li>• <i>widgets</i>: Enthält benutzerdefinierte UI-Widgets für den <i>X4 Web Designer</i>, (<a href="#">Eigene Widgets erstellen</a>)</li> <li>• <i>xml</i></li> </ul> </li> </ul>
--	--

Im *X4 Designer* wird im Repository folgende Struktur erzeugt:









	<ul style="list-style-type: none"> <li>• <i>Projekt</i>: Projektdatei       <ul style="list-style-type: none"> <li>• Build:           <ul style="list-style-type: none"> <li>• Build.wrf: Technischer Prozess, der die Web-Anwendung (automatisiert oder manuell) zu bauen (siehe <a href="#">Web-Anwendung neu bauen (Build)</a>).</li> </ul> </li> <li>• Portal:           <ul style="list-style-type: none"> <li>• ActionBar</li> <li>• Detail</li> <li>• Lister</li> <li>• Main</li> <li>• Navigator</li> <li>• Portal.wsinc</li> </ul> </li> <li>• WebResources: Enthält die Ressourcen, die im <i>X4 Web Designer</i> zur Verfügung stehen.           <ul style="list-style-type: none"> <li>• src: Enthält jeweils in Unterordnern sämtliche Quelldateien der Web-Anwendung, die im <i>X4 Web Designer</i> erstellt und bearbeitet wird. Der <i>X4 Web Designer</i> greift auf diese Ordnerinhalte zu.               <ul style="list-style-type: none"> <li>• main                   <ul style="list-style-type: none"> <li>• css: CSS-Formatvorlagen</li> <li>• html: Statische HTML-Dateien</li> <li>• img: Grafik-Ressourcen (Logos, Icons, ...)</li> <li>• js: JavaScript-Dateien</li> <li>• json: JSON-Dateien</li> </ul> </li> <li>• portal: Enthält die Konfigurationsdateien der Web-Anwendung, die mit dem <i>X4 Web Designer</i> erstellt wurden.                   <ul style="list-style-type: none"> <li>• actionbar: Enthält die Konfigurationsdatei für die Aktionsleiste (<a href="#">Aktionsleiste</a>)</li> <li>• datasource: Enthält die Konfigurationsdatei für die Datasource (<a href="#">Datasources konfigurieren</a>)</li> <li>• main: Enthält die generelle Konfigurationsdatei für die Web-Anwendung</li> <li>• navigator: Enthält die Konfigurationsdatei für den Navigator</li> </ul> </li> <li>• ui: Enthält die Konfigurationsdateien der Web-Anwendung in jeweils einer Datei mit der Endung <code>.ui.json</code>, (<a href="#">Benutzeroberflächen erstellen</a>)</li> <li>• widgets: Enthält benutzerdefinierte UI-Widgets für den <i>X4 Web Designer</i>, (<a href="#">Eigene Widgets erstellen</a>)</li> <li>• xml</li> </ul> </li> <li>• dbConfig.xml</li> </ul> </li> <li>• <i>Project.winc</i>: Projektspezifische Webservice-Konfiguration, wird automatisch in der zentralen Webservice-Konfiguration des <i>X4 Servers</i> registriert (<a href="#">Projektname oder Ressourcen-URL Ihrer Web-Anwendung ändern</a>).</li> </ul> </li> </ul>
--	--











## 2.4.2 Das Kontextmenü im X4 Web Designer


Im *X4 Web Designer* steht für jedes Element in der Ordnerstruktur ein Kontextmenü zur Verfügung. Das Kontextmenü kann per Klick auf das Zahnradsymbol (⚙️) neben dem entsprechenden Element (Datei oder Ordner) aufgerufen werden. Die Optionen im Kontextmenü hängen von der Datei ab, für die das Kontextmenü aufgerufen wurde.

### 2.4.2.1 Einträge im Kontextmenü für Ordner

 <b>Refresh</b>	Ordner aktualisieren
 <b>Upload Files</b>	Dateien hochladen
 <b>Download</b>	Dateien herunterladen
 <b>Rename</b>	Ordner umbenennen
 <b>Duplicate</b>	Ordner duplizieren
 <b>Copy</b>	Ordner kopieren
 <b>Paste</b>	Datei aus der Zwischenablage in Ordner einfügen
 <b>Build</b>	Ordner für das Portal neu umwandeln.

### 2.4.2.2 Einträge im Kontextmenü für Dateien

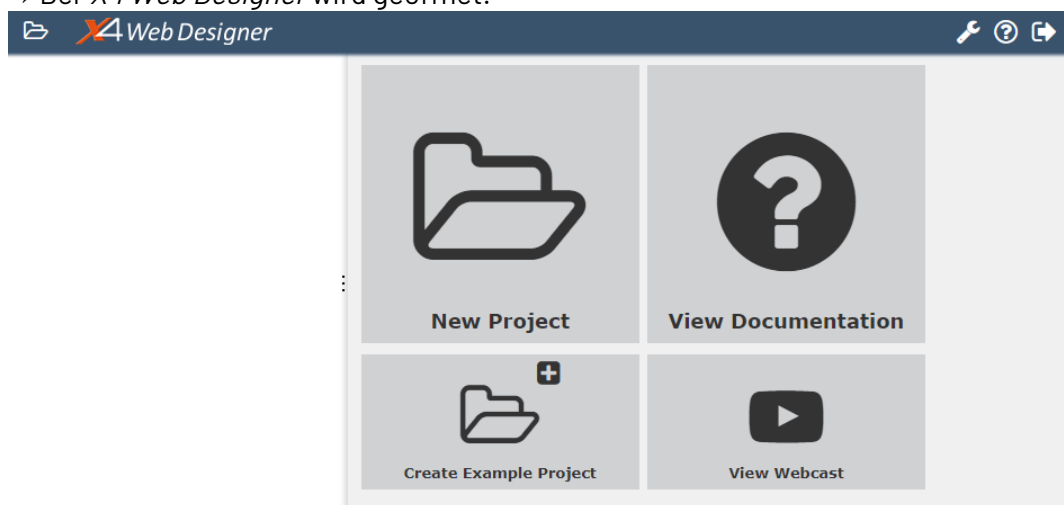
 <b>Open Preview</b>	Vorschau des Portals in einem neuen Browsertab/-fenster öffnen.
 <b>Open plain</b>	Quellcode der Datei in einem neuen Browsertab/-fenster öffnen.
 <b>Open resource</b>	Datei in einem neuen Browsertab/-fenster öffnen.
 <b>Copy resource URL</b>	URL der Ressource in die Zwischenablage kopieren.
 <b>Copy result URL</b>	URL des Ergebnisses in die Zwischenablage kopieren.
 <b>Download</b>	Datei herunterladen
 <b>Rename</b>	Datei umbenennen
 <b>Duplicate</b>	Datei duplizieren

 <b>Copy</b>	Datei kopieren
---	----------------

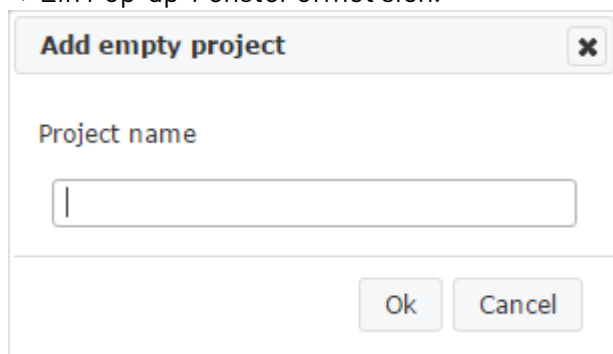
### 2.4.3 Neues Web-Portal-Projekt im X4 Web Designer anlegen

Wie Sie im *X4 Web Designer* ein neues Projekt bzw. eine neue Web-Anwendung auf Basis von X4 Activities erstellen

1. *X4 Web Designer* im Browser aufrufen, siehe [X4 Web Designer](#) .  
→ Der *X4 Web Designer* wird geöffnet.



2. Auf **New Project** klicken.  
→ Ein Pop-up-Fenster öffnet sich.



3. Einen Projektnamen eingeben und OK drücken.


 Für Projektnamen sind weder Umlaut noch Leerzeichen erlaubt.

*Beispiel:*

*MyWebApplication*

→ Ein Projekt mit allen notwendigen Unterordnern wird erzeugt und auf dem Server hinterlegt. Das Projekt ist sowohl im *X4 Web Designer* als auch im *X4 Designer* zu sehen ([Struktur eines Web-Portal-Projekts](#)).


4. Falls der Name der Web-Applikation vom URL-Part abweichen soll, muss die projektspezifische ReST-Konfigurationsdatei angepasst werden, siehe [Projektname oder Ressourcen-URL Ihrer Web-Anwendung ändern](#).


 Nach erfolgreichem Anlegen eines Projekts wird auf dem X4 Server eine Reihe von Ordnern und Dateien angelegt (standardmäßig in Benutzerverzeichnis <X4>/X4Server>/X4DB/1), siehe [Struktur einer X4 Activities-Anwendung](#). Die Webservice-Konfiguration wird automatisch erweitert.


## 2.4.4 Ordner und Dateien im Navigator verwalten

### 2.4.4.1 Ordner und Dateien anlegen


1. Ordner in der Ordnerstruktur auswählen, in dem eine Datei oder ein Ordner angelegt werden soll.  
→ Der Ordner wird ausgewählt und das Menü des *X4 Web Designer* ändert sich.
2. Im Menü des *X4 Web Designer* auf **New File** bzw. **New Folder** klicken.  
→ Ein Pop-up-Fenster öffnet sich.

 Alternativ: Auf das Plus (+) neben dem ausgewählten Ordner klicken und im Kontextmenü auf **Add file** (📄) bzw. **Add folder** (📁) klicken

3. Den Namen der anzulegenden Datei bzw. des anzulegenden Ordners eingeben. Es sind nur Zahlen, Buchstaben, Unterstriche (\_) und Bindestriche (-) erlaubt.  
 Wenn keine Dateiendung angegeben wird, dann wird die Standard-Dateiendung für den gewählten Ordner verwendet (z. B. .html im Ordner html).
4. Auf Ok klicken.  
→ Die Datei bzw. der Ordner wurde erzeugt und wird jetzt in der Ordnerstruktur angezeigt.

 Bereits vorhandene Dateien können dupliziert werden. Dazu:

1. Neben der zu duplizierenden Datei auf das Zahnrad (⚙️) klicken.  
→ Das Kontextmenü öffnet sich.
2. Im Kontextmenü auf **Duplicate** (📄) klicken.  
→ Ein Pop-up-Fenster öffnet sich.
3. Einen Namen für die neue Datei eingeben.
4. Auf **Ok** klicken.

 Die zu duplizierende Datei kann auch mit gedrückter Strg-Taste per Drag & Drop dupliziert werden.

#### 2.4.4.2 Dateien öffnen und verwalten

1. Datei, die geöffnet werden soll, in der Ordnerstruktur des *X4 Web Designer* anklicken.  
→ Die Datei wird geöffnet. Sie wird im Inhaltsbereich des *X4 Web Designer* angezeigt.
2. In der Ordnerstruktur neben der Datei auf das Zahnrad (⚙️) klicken.  
→ Das Kontextmenü öffnet sich.

Im Kontextmenü stehen verschiedene Optionen zur Auswahl:

<b>Open plain</b>	Code-Ansicht der Quelldatei öffnen
<b>Open resource</b>	Code-Ansicht der entsprechenden Datei im Target-Verzeichnis öffnen. Zeigt die Datei im Browser an.
<b>Copy resource URL</b>	URL der Datei in die Zwischenablage speichern
<b>Download</b>	Datei herunterladen
<b>Rename</b>	Datei umbenennen
<b>Duplicate</b>	Datei duplizieren
<b>Copy</b>	Datei kopieren

#### 2.4.4.3 Ordner und Dateien löschen

1. Datei oder Ordner, die bzw. der gelöscht werden soll, in der Ordnerstruktur auswählen.
2. In der Ordnerstruktur neben der Datei bzw. dem Ordner auf das Löschen-Symbol (🗑️) klicken.  
→ Ein Bestätigungsdialog öffnet sich.
3. Um die Datei bzw. den Ordner endgültig zu löschen, auf **Ok** klicken.



- Dieser Vorgang kann nicht rückgängig gemacht werden!
- Damit wird die Ressource auf dem Server im *X4 Repository* gelöscht.
- Ordner können nur gelöscht werden, wenn diese leer sind.

#### 2.4.4.4 Dateien und Ordner umbenennen

1. Datei oder Ordner, die bzw. der umbenannt werden soll, in der Ordnerstruktur auswählen.
2. In der Ordnerstruktur neben der Datei bzw. dem Ordner auf das Zahnrad (⚙️) klicken.  
→ Das Kontextmenü öffnet sich.
3. Im Kontextmenü auf **Rename** (✎️) klicken.  
→ Ein Pop-up-Fenster öffnet sich.
4. Einen neuen Namen für die Datei bzw. den Ordner eingeben.
5. **Ok** klicken.

#### 2.4.4.5 Dateien und Ordner verschieben

Dateien und Ordner können in der Ordnerstruktur des *X4 Web Designers* einfach per Drag & Drop verschoben werden.

### 2.4.5 Projekt bzw. Web-Anwendung löschen

So löschen Sie eine Web-Anwendung vollständig.

1. Im *X4 Designer* den gesamten Projektordner aus dem *X4 Repository* löschen.
2. Im *X4 Designer* auf **Tools > Manage Webservices** klicken.  
→ Die Webservice-Konfiguration öffnet sich.
3. In der Webservice-Konfiguration die Include-Referenz auf die projektspezifische Webservice-Konfigurationsdatei `<Projekt>/Project.wsinc` entfernen.
4. Webservice-Konfiguration speichern.
5. **Reload on server** klicken und im Dialog mit **Save and Reload** bestätigen, um die Änderungen serverweit anzuwenden.

### 2.4.6 Projektname oder Ressourcen-URL Ihrer Web-Anwendung ändern

Wie Sie vorgehen, falls der Projektname vom URL-Part abweicht, unter dem die Web-Ressourcen für Ihre Web-Anwendung via HTTP ReST bereitgestellt wurden, und der URL-Part umbenannt werden soll. Im *X4 Designer* müssen folgende Dateien verändert werden:

- `project.restconfig`  
Die Datei befindet sich im Projektordner der Web-Anwendung
  - `projects.xml`  
Die Datei befindet sich im Ordner `/X4WebIDE_Instance/Config/projects.xml`
  - `Project.wsinc`  
Die Datei befindet sich im Projektordner der Web-Anwendung
1. `Project.wsinc` anpassen:
    - Den Fileservice File unter `Project > Resources > <filePath>` anpassen (**Resource Path**).
    - Datei speichern.
  2. Im *X4 Designer* die globale Konfigurationsdatei des X4 Web Designers in `/X4WebIDE_Instance/Config/projects.xml` anpassen:
    - a. Im entsprechenden `<Project>`-Eintrag den Wert von Attribut `baseUrl` anpassen.  
*Beispiel:* `baseUrl="/X4/httpstarter/ReST/MyWebApp/Resources"`
    - b. Datei speichern.
  3. Die Webservice-Konfiguration auf dem *X4 Server* neu laden:
    - a. Im *X4 Designer* das Menü **Management > Webservice Configuration** aufrufen.
    - b. Auf **Reload on Server** klicken und im Dialog mit **Save and Reload** bestätigen.Die Ressourcen-URL Ihrer Web-Anwendung wurde nun geändert.

## 2.5 Bestandteile einer X4 Activities-Anwendung

Eine Activities-Anwendung besteht aus verschiedenen Komponenten, die im Folgenden näher beschrieben werden.

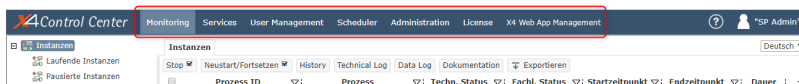
The screenshot shows the X4Control Center interface. The top navigation bar includes 'Monitoring', 'Services', 'User Management', 'Scheduler', 'Administration', 'License', and 'X4 Web App Management'. The left sidebar contains a tree view with 'Instanzen' expanded, showing sub-items like 'Laufende Instanzen', 'Pausierte Instanzen', 'Ausgeführte Instanzen', 'Fehler', 'Meldungen', 'Dashboards', and 'Prozesse'. The main content area is titled 'Instanzen' and features a table of instances. The table has columns: 'Prozess ID', 'Prozess', 'Techn. Status', 'Fachr. Status', 'Startzeitpunkt', 'Endzeitpunkt', and 'Dauer'. The table lists several instances, all with a 'FINISHED' status. Below the table, there are buttons for 'Stop', 'Neustart/Fortsetzen', 'History', 'Technical Log', 'Data Log', 'Dokumentation', and 'Exportieren'. A 'Savepoints' section shows 'Es sind keine Savepoints für diese Instanz vorhanden.' and a 'Neustart/Fortsetzen' button. At the bottom, a 'ComboBox.wrf' diagram shows a flow from 'Start' to 'ComboBox' to 'Stop'.

Komponente	Beschreibung
1. Application Bar	<p>Eine Activities-Anwendung kann aus verschiedenen Applikationen bestehen:</p> <ul style="list-style-type: none"> <li>• <i>Haupt-Applikation</i>: Die Haupt-Applikation wird standardmäßig beim Start der Anwendung geladen. Sie ist mit dem Anwendungslogo verknüpft.</li> <li>• <i>Unter-Applikation</i>: Neben der Haupt-Applikation kann eine Activities-Anwendung beliebig viele Unter-Applikationen enthalten. Diese werden neben dem Logo angezeigt.</li> </ul> <p>Die Application Bar kann auch Verlinkungen zu anderen Anwendungen oder Websites enthalten.</p>
2. Navigator	<p>Der Navigator ist ein Menübaum, der auf der linken Seite der Applikation angezeigt wird. Die einzelnen Knoten des Baumes können mit einer Listenansicht oder einer Detailseite verknüpft werden. Diese werden im Content-Bereich dargestellt.</p> <p>Die Navigator-Konfiguration wird im Abschnitt <a href="#">Navigator</a> beschrieben.</p>
3. Aktionsleiste	<p>Mit der Aktionsleiste können konfigurierbare Buttons über der Listenansicht eingeblendet werden. Damit können Aktionen (ReST-Anfragen) für die Einträge der Liste ausgeführt werden. In den meisten Fällen liefern diese eine Detailseite, die z. B. als Dialog oder Tab angezeigt wird. Diese Detailseite kann entweder eine Rückmeldung der ausgeführten Aktion sein oder ein Formular bereitstellen, um beispielsweise den Datensatz zu bearbeiten oder weitere Informationen zu erfassen.</p> <p>Die Konfiguration der Aktionsleiste wird im Abschnitt <a href="#">Aktionsleiste</a> beschrieben.</p>

Komponente	Beschreibung
4. Liste nansi cht (auch Liste r)	<p>Die Listenansicht zeigt eine generische Liste von Datensätzen an. Diese kann beliebig viele typisierte Spalten enthalten. Jeder Eintrag der Liste kann mit einer Detailseite verknüpft werden. Zudem können über die Aktionsleiste beliebige Aktionen für den ausgewählten Eintrag ausgeführt werden.</p> <p>Die Konfiguration der Listenansicht wird im Abschnitt <a href="#">Listenansicht</a> beschrieben.</p>
5. Detai lansi cht	<p>Die Detailansicht wird verwendet, um Detail-Informationen zu einem Listeneintrag oder einer Aktion darzustellen.</p> <p>Eine Detailseite in Activities kann beliebige Dokumente enthalten, die in einem Browser darstellbar sind (iFrame). In der Regel sind es jedoch HTML-Seiten. Diese können z. B. von einem technischen Prozess bereitgestellt werden.</p> <p>Die Konfiguration der Detailansicht wird im Abschnitt <a href="#">Detailmasken</a> beschrieben.</p>

## 2.5.1 Application Bar

In einer Web-Anwendung, die mit X4 Activities Classic erstellt wurde, können mehrere Applications enthalten sein. Diese werden in der Application Bar am oberen Bildschirmrand angezeigt.



### *Applications in der Application Bar*

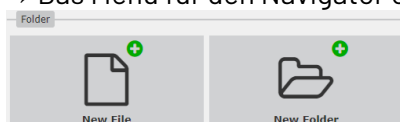
Applikationen können in der [Main Config](#) im Bereich Applications definiert werden.

## 2.5.2 Navigator

Der Navigator wird auf der linken Seite einer Web-Anwendung angezeigt. Der Navigator besteht aus einer Knotenstruktur mit der der Benutzer zwischen den verschiedenen Komponenten der Application navigieren kann.

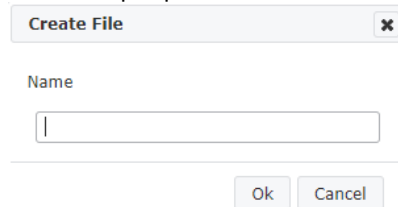
### 2.5.2.1 Einen neuen Navigator anlegen

1. Web Designer öffnen.
2. Web-Anwendungsprojekt öffnen, für das ein Navigator angelegt werden soll.
3. In der Ordnerstruktur den Ordner **navigator (Projektordner > portal > navigator)** auswählen.  
→ Das Menü für den Navigator öffnet sich:



4. Auf **New File** klicken.

→ Ein Pop-up-Fenster öffnet sich:




5. Namen für den Navigator eingeben und auf **Ok** klicken.  
→ Eine neue Navigator-Datei wurde erzeugt. Die Navigator-Datei wird jetzt in der **Dateiübersicht** angezeigt.

### 2.5.2.2 Einen Navigator bearbeiten

1. Web Designer öffnen.
2. Web-Anwendungsprojekt öffnen, für das ein Navigator angelegt werden soll.
3. In der Ordnerstruktur die Navigator-Datei, die bearbeitet werden soll ( **Projektordner** > **portal** > **navigator** > **Navigatordatei** ) auswählen.  
→ Application Config wird im *Application Config Editor* angezeigt. ([Verweis auf verschiedene Editoren](#))

In den Einstellungen der Application Config gibt es mehrere Bereiche:

Bereich	Beschreibung
<i>Navigator ID</i>	<i>Erforderlich.</i> ID des Navigators. Die ID muss innerhalb einer Web-Anwendung eindeutig sein.  <b>Mögliche Werte:</b> Zeichenkette aus alphanumerischen Zeichen
<i>Only Root Node</i>	?  <b>Mögliche Werte:</b> true / false (Standard)
<i>Nodes</i>	Enthält Einstellungen zu den in der Web-Anwendung enthaltenen Applikationen. Siehe <a href="#">Nodes</a> .

4. Gewünschte Änderungen vornehmen.
5. Auf **Save** (  **Save** ) klicken, um zu speichern.  
→ Die Änderungen wurden gespeichert und werden beim nächsten Aufruf der Web-Anwendung verwendet.




### 2.5.2.3 Nodes

Einstellung	Beschreibung
Node ID	



Einstellung	Beschreibung
Node Name	<p>String für den Display-Namen des Knotens (siehe <a href="#">Internationalisierung</a>)</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>
Node Type	<p><i>Erforderlich.</i> Typ des Knotens</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• NODE: Knoten mit Kindknoten</li> <li>• LEAF: Blattknoten ohne Kinder</li> </ul>
Content Type	<p>Typ des Inhalts</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• LISTER_DETAIL: Lister mit Detailansicht (Standard)</li> <li>• LISTER: Lister ohne Detailansicht</li> <li>• DETAIL: Nur Detailansicht</li> <li>• NONE: Inhaltsansicht bleibt leer</li> </ul>
Content URL	<p>URL für den Inhalt, auf den der Knoten verweist.</p> <p><b>Mögliche Werte:</b> Zeichenkette (URL)</p>
Button URL	<p>URL für die Action-Bar-Konfiguration des Knotens</p> <p><b>Mögliche Werte:</b> Zeichenkette (URL)</p>
Node Icon Url	<p>URL für das Knoten-Icon</p> <p><b>Mögliche Werte:</b> Zeichenkette (URL)</p>
Content Header URL	
Content Header Height	
Default Lister Page Size	<p>Legt fest, wie viele Einträge standardmäßig in einem Lister für diesen Knoten angezeigt werden.</p> <p><b>Mögliche Werte:</b> 5, 10, 15 (Standard), 20, 25, 30, 50, 100</p>
Top Content Height	<p>Legt die Höhe (in Pixel) des oberen Bereichs im Haupt-Tab fest (Lister und Content-Header).</p> <p><b>Mögliche Werte:</b> Ganzzahl</p>
Is Default Node	<p><i>Erforderlich für genau einen Knoten pro Navigator.</i> Der Knoten wird beim Laden vorselektiert.</p> <p><b>Mögliche Werte:</b> true / false (Standard)</p>

Innerhalb des Node-Bereichs gibt es drei verschiedene Buttons, mit denen die Application als Ganzes bearbeitet werden kann:

-  Node duplizieren
-  Node löschen
-  Node verschieben (Drag & Drop)

### 2.5.2.3.1 Marker in einem Knoten

In jedem `NavigatorNode`-Element kann durch das optionale Element `Marker` eine Markierung neben dem Label des Baum-Knotens konfiguriert werden. Diese Markierung kann beispielsweise verwendet werden, um die Anzahl neuer Einträge dieses Knotens anzuzeigen. Das Element unterstützt folgende Attribute:

Attribut	Beschreibung
<code>value</code>	<i>Erforderlich.</i> Wert, der in der Markierung angezeigt wird.  <b>Mögliche Werte:</b> Beliebige Zeichenkette oder Zahl
<code>fgColor</code>	Text-Farbe der Markierung. Fehlt dieses Attribut, wird stattdessen eine Standard-Farbe aus dem Theme der Applikation verwendet.  <b>Mögliche Werte:</b> HTML-RGB-Farbcode (z. B. <code>#FFDD22</code> )
<code>bgColor</code>	Hintergrund-Farbe der Markierung. Fehlt dieses Attribut, wird stattdessen eine Standard-Farbe aus dem Theme der Applikation verwendet.  <b>Mögliche Werte:</b> HTML-RGB-Farbcode (z. B. <code>#FFDD22</code> )

#### Beispiel

```
<NavigatorNode ...>
  <Marker value="25" fgColor="#FFFFFF" bgColor="#F05000" />
</NavigatorNode>
```

### 2.5.2.3.2 Content-Header

In jedem NavigatorNode-Element kann optional ein Element ContentHeader definiert werden. Dieses ermöglicht die Aktivierung des Headers im Content-Bereich und benötigt folgende Attribute:

Attribut	Beschreibung
url	<i>Erforderlich.</i> URL auf die im Header darzustellende Seite. <b>Mögliche Werte:</b> Zeichenkette (URI)
height	<i>Erforderlich.</i> Höhe des Headers in Pixel. <b>Mögliche Werte:</b> Ganzzahl

#### Beispiel

```
<NavigatorNode ...>
  <ContentHeader url="/html/myContentHeader.html" height="100" />
</NavigatorNode>
```

### 2.5.2.3.3 Globale Suche

In jedem NavigatorNode-Element kann optional ein Element GlobalSearch definiert werden. Dieses ermöglicht die Aktivierung des Suchfeldes im Applikations-Header, wenn der Baumknoten selektiert ist. Optional kann darin noch ein Hilfe-Text angegeben werden, welcher als Tooltip für das Suchfeld im Browser erscheint.

#### Beispiel

```
<NavigatorNode ...>
  <GlobalSearch enabled="true">
    <helpText>Hier können Sie einen Suchbegriff eingeben.</helpText>
  </GlobalSearch>
</NavigatorNode>
```

## 2.5.2.4 Automatische Aktualisierung von Knoten

Die Anzeige von Navigator-Knoten kann automatisch aktualisiert werden. Dafür können in der Navigator-Konfiguration Update-Intervalle konfiguriert werden. Bei jeder Aktualisierungsanfrage können dabei beliebig viele Knoten (auch keiner) aktualisiert werden.

### 2.5.2.4.1 Konfiguration

Die Konfiguration der Update-Intervalle erfolgt direkt unter dem Wurzel-Element der Navigator-Konfiguration <NavigatorConfig> durch das Element <NavigatorUpdates>. < NavigatorUpdates

kann beliebig viele NavigatorUpdate-Elemente für die einzelnen Intervalle enthalten. Jedes dieser Elemente benötigt folgende Attribute:

Attribut	Beschreibung
url	<i>Erforderlich.</i> URL, von der die Aktualisierungsinformationen geladen werden.  <b>Mögliche Werte:</b> Zeichenkette (URI)
interval	Zeitabstand zwischen den Aktualisierungen in Millisekunden.  <b>Mögliche Werte:</b> Ganzzahl

#### Beispiel

```
<NavigatorConfig navigatorId="myNavigatorID">
  <NavigatorUpdates>
    <NavigatorUpdate url="/URL/fuer/haeufige/Aktualisierung" interval="5000" />
    <NavigatorUpdate url="/URL/fuer/seltene/Aktualisierung" interval="60000" />
  </NavigatorUpdates>
  <NavigatorNode ...>
    ...
  </NavigatorNode>
</NavigatorConfig>
```

#### 2.5.2.4.2 Antwort-Format

Als Antwort auf eine Aktualisierungsanfrage wird ein XML-Dokument erwartet, dessen Wurzel-Element NavigatorUpdate heißt. Damit die Änderungen akzeptiert werden, muss dieses ein targetNavigator-Attribut enthalten, das mit der ID des aktuellen Navigators übereinstimmt. Unterhalb von targetNavigator wird ein UpdateNodes-Element erwartet, in dem für jeden Knoten, der aktualisiert werden soll, ein UpdateNode-Element vorhanden ist. Die Knoten werden dabei über ihre ID im Attribut targetNode referenziert. Dadurch kann aber nur die Anzeige und nicht die Funktionalität oder Struktur eines Knotens geändert werden. Deshalb werden in UpdateNode nur die Attribute nodeName und nodeIconUrl sowie das Marker-Element (keine weiteren Attribute/Elemente von NavigatorNode) unterstützt.

Innerhalb der Antwort ist es auch möglich, das Update-Intervall zu ändern. Dazu kann unterhalb des Wurzel-Elements NavigatorUpdate ein ChangeUpdateConfig-Element definiert werden, in dem das neue Intervall im interval-Attribut konfiguriert werden kann. Zudem kann das Intervall mit dem booleschen Attribut disable komplett abgeschaltet werden. Das interval-Attribut sollte nur angegeben werden, wenn tatsächlich eine Intervall-Änderung gewünscht ist, da dies immer zu einem Neustart des Timers führt.

Wenn ein Attribut/Element in der Update-Struktur fehlt, dann bleibt der entsprechende Wert im Navigator unverändert. Für das Entfernen oder Zurücksetzen eines Wertes muss deshalb explizit ein leerer Eintrag definiert werden.

**Beispiel-Antwort**

```

<NavigatorUpdate targetNavigator="myNavigatorID">
  <UpdateNodes>
    <UpdateNode targetNode="node1" nodeName="Neuer Knoten-Name" nodeIconUrl="newIcon.png">
      <Marker value="23" fgColor="#FFFFFF" bgColor="#FF0000"/>
    </UpdateNode>
    <UpdateNode targetNode="node5"> <!-- Label und Icon bleiben unverändert -->
      <Marker value="" /> <!-- Marker dieses Knotens entfernen -->
    </UpdateNode>
    <UpdateNode targetNode="node6" /> <!-- Knoten bleibt unverändert (selbes Verhalten wenn das Element fehlt) -->
  </UpdateNodes>
</NavigatorUpdate>

```

**Ändern des Intervalls**

```

<NavigatorUpdate targetNavigator="myNavigatorID">
  <ChangeUpdateConfig interval="20000" disable="false" /> <!-- Ändert das Update-Intervall auf 20 Sekunden -->
  <UpdateNodes>
    ...
  </UpdateNodes>
</NavigatorUpdate>

```

## 2.5.3 Aktionsleiste

Die Aktionsleiste ist ein Element, das über der Lister-Tabelle eingeblendet wird, falls eine Konfiguration für Aktionen vorliegt. Sie enthält konfigurierbare Schaltflächen, die mit einem Titel, einem Icon und einer Aktion belegt werden können. Aktionen können allgemein sein (z. B. Erzeugen eines neuen Listenelements) oder sich auf eine Selektion in der Lister-Tabelle beziehen (z. B. gewähltes Element löschen).

Die Aktionsleiste beinhaltet nur Aktionen, die nicht von einem definierten Vorgangstatus abhängig sind. Der Prozess, der sich hinter einem Button verbirgt, muss für alle Vorgänge der Liste anwendbar sein.

Die Aktionsleiste wird pro Ordner im Navigator definiert. Zu jeder Aktion ist in der Konfiguration eine ReST-URL hinterlegt, die beim Anklicken der Schaltfläche aufgerufen wird. Wenn ein Element in der Vorgangsliste ausgewählt ist, wird der Primärschlüssel dieses Vorgangs an den X4-Prozess übertragen.

### 2.5.3.1 Eine Aktionsleiste anlegen

1. *X4 Web Designer* öffnen.
2. Im Web-Anwendungsprojekt, für das die Aktionsleiste erzeugt werden soll, den Ordner *Project > portal > actionbar* in der Ordnerstruktur auswählen.

3. Auf **New File** klicken.  
→ Ein Pop-up-Fenster öffnet sich.
4. Einen Namen für die neue Datei angeben und **Ok** klicken.  
→ Eine neue Aktionsleisten-Datei wird angelegt.

#### 2.5.3.2 Eine Aktionsleiste bearbeiten


1. *X4 Web Designer* öffnen.
2. Datei der Aktionsleiste, die bearbeitet werden soll, in der Ordnerstruktur auswählen.  
→ Die Datei wird geöffnet.


In den Einstellungen der Aktionsleiste gibt es mehrere Bereiche:

Bereich	Beschreibung
Icon Size	Größe der Icons, die auf den Buttons angezeigt werden.  <b>Mögliche Werte:</b> Ganzzahl (Standard: 16)
Buttons	Enthält Einstellungen zu den in der Aktionsleiste enthaltenen Buttons. Siehe <a href="#">Buttons</a> .

#### 2.5.3.3 Buttons

Bereich	Beschreibung
Button ID	<i>Erforderlich.</i> Technische ID der Aktion  <b>Mögliche Werte:</b> Zeichenkette aus alphanumerischen Zeichen
Display Name	<i>Erforderlich.</i> Anzeigename der Aktion (siehe auch <a href="#">Internationalisierung</a> )  <b>Mögliche Werte:</b> Beliebige Zeichenkette

Bereich	Beschreibung
Action Type	<p>Legt fest, wie Primary-Key-Werte der Aktion zum Server übertragen werden.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <b>DEFAULT:</b> Senden per HTTP GET, Name und Wert des Primary Key werden an die URL angehängt.</li> <li>• <b>GET:</b> Senden per HTTP GET, Name und Wert des Primary Key werden als Parameter an die URL angehängt.</li> <li>• <b>POST:</b> Senden per HTTP POST, Name und Wert des Primary Key werden als Parameter im HTTP-Body übertragen (application/x-www-form-urlencoded). Der Name der Parameter ist x4pPrimaryKey.</li> <li>• <b>MULTI_POST:</b> Senden per HTTP POST, Name und Wert des Primary Key werden als Parameter im HTTP-Body übertragen (application/x-www-form-urlencoded). Der Name der Parameter ist x4pPrimaryKey.</li> </ul> <div data-bbox="579 1104 1441 1361" style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> Durch actionType="MULTI_POST" wird automatisch das Selektieren von mehreren Einträgen in der Liste aktiviert (eine Checkbox vor jeder Zeile). Diese Mehrfach-Selektion ist allerdings nur für Schaltflächen mit diesem Aktionstyp relevant, die anderen verwenden weiterhin die einfache Selektion.</p> </div>
Action URL	<p><i>Erforderlich.</i> ReST-URL, zeigt auf den X4-Prozess mit der Implementierung der Aktion</p> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p> <p><i>Beispiel:</i> Der Prozess /X4/httpstarter/ReST/x4p/ListerExport stellt beispielsweise die aktuelle Auswahl als CSV-Download zur Verfügung.</p>
Selection Needed	<p>Legt fest, ob in der Tabelle ein Eintrag ausgewählt sein muss, damit der Button aktiv ist.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <code>true</code>: nur aktiv, wenn eine Selektion in der Tabelle vorliegt</li> <li>• <code>false</code>: auch ohne Selektion aktiv</li> </ul>

Bereich	Beschreibung
Display Type	<p><i>Erforderlich.</i> Steuert die Anzeige des Ergebnisses.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• MASTERDETAIL_VIEW: Ergebnis im Detailbereich auf dem Haupt-Tab anzeigen.</li> <li>• NEWDETAIL_VIEW: Ergebnis in einer Detailansicht in einem neuen Tab anzeigen.</li> <li>• DIALOG_VIEW: Ergebnis in einem Pop-up-Fenster im Browser anzeigen.</li> <li>• GWT_DIALOG: Ergebnis in einem nicht-modalen GWT-Dialog anzeigen.</li> <li>• MODAL_GWT_DIALOG: Ergebnis in einem modalen GWT-Dialog anzeigen.</li> <li>• DOUBLE_CLICK: Simuliert einen Doppelklick auf die selektierte Zeile des Listers.</li> <li>• DOWNLOAD: Ergebnis in einem unsichtbaren iFrame laden.</li> </ul> <div> <p> Hierzu muss der X4-Prozess einen entsprechenden HTTP-Header setzen, so dass das Ergebnis vom Browser als Download angeboten wird.</p> </div> <p>Für mehr Details zu den GWT-Dialogen und -Tabs, siehe <a href="#">JavaScript-API für Dialoge in X4 Activities</a>.</p>
Default Close	<p>Steuert, ob GWT-Dialoge über die Kopfzeile geschlossen werden können.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• AUTO: Wird durch die GWT-Komponente vorgegeben (entspricht: DISABLED).</li> <li>• DISABLED: Dialog kann nicht über die Kopfzeile geschlossen werden.</li> <li>• ENABLED: Dialog kann über die Kopfzeile geschlossen werden.</li> </ul>
Icon URL	<p>URI des Icons für die Aktionsschaltfläche</p> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p> <div> <p> Unter /X4PSupport/images stehen eine Reihe von Standard-Icons zur Verfügung, z. B. download.png für den displayType <i>DOWNLOAD</i>.</p> </div>
Tooltip	<p>Tooltip-Text, der bei MouseOver angezeigt wird.</p> <p><b>Mögliche Werte:</b> Beliebige Zeichenkette</p>



Bereich	Beschreibung
Width	Breite des Dialogs in Pixel oder als CSS-Längenangabe, falls <code>displayType</code> ein Dialog ( <code>DIALOG_VIEW</code> , <code>GWT_DIALOG</code> , <code>MODAL_GWT_DIALOG</code> ) ist.
Height	Höhe des Dialogs in Pixel oder als CSS-Längenangabe, falls <code>displayType</code> ein Dialog ( <code>DIALOG_VIEW</code> , <code>GWT_DIALOG</code> , <code>MODAL_GWT_DIALOG</code> ) ist.

#### 2.5.3.4 Ausblenden/Deaktivieren von Schaltflächen

Schaltflächen in der Aktionsleiste können so konfiguriert werden, dass sie nur eingeblendet/aktiviert werden, wenn die selektierten Einträge in der Listenansicht bestimmte Kriterien erfüllen. Diese Konfiguration erfolgt durch Filter. Nur wenn alle aktuell selektierten Einträge von den definierten Filtern akzeptiert werden, ist die Schaltfläche aktiv/sichtbar. Für die Filter stehen dieselben Filtertypen zur Verfügung, die auch für das [Filtern von Listeneinträgen](#) bereitgestellt werden. Zur Definition der Filter können im Button-Element optional die Elemente `DisplayFiltering` (Ein-/Ausblenden der Schaltfläche) und/oder `ActivationFiltering` (Aktivieren/Deaktivieren der Schaltfläche) verwendet werden.

Die Filter eines Button-Elements mit `selectionNeeded="false"` werden ignoriert, wenn kein Eintrag in der Tabelle selektiert ist.

#### Beispiel

```
<Button buttonId="edit" displayName="Bearbeiten" selectionNeeded="true" ...>
  <DisplayFiltering>
    <FilterEntry>
      <EnumFilter id="State">
        <EnumFilterItem id="OPEN"/>
        <EnumFilterItem id="IN_PROGRESS"/>
      </EnumFilter>
    </FilterEntry>
  </DisplayFiltering>
  <ActivationFiltering>
    <FilterEntry>
      <BooleanFilter id="Blocked" value="false" />
    </FilterEntry>
    <FilterEntry>
      <DatetimeFilter id="CreationDate" min="2015-03-01T00:00:00"/>
    </FilterEntry>
  </ActivationFiltering>
</Button>
```

#### 2.5.3.5 Implementierung

Wenn eine Aktionsfläche aktiviert wurde, sind folgende Aktivitäten notwendig:

1. Wenn keine Eingabe erfolgen muss:
  - Der aufgerufene Prozess führt die gewünschte Aktion aus und liefert eine Meldung („Erfolgreich/nicht erfolgreich ausgeführt“) als HTML-Seite.

- Bei Bedarf Vorgangsliste per JavaScript (siehe [Globale JavaScript-Methoden](#)) aus der HTML-Seite (z. B. per Klick auf einen Schließen-Button oder Load-Event) aktualisieren.
2. Wenn eine Eingabe erfolgen muss:
    - Dialog oder Detailansicht wird mittels HTML-Response vom X4 Server mit einem Formular befüllt.
    - Nach dem Submit sollte die Vorgangsliste aktualisiert werden, um Änderungen in der Listensicht zu erfassen
    - Wenn ein Dialog verwendet wurde, muss dieser noch geschlossen werden (siehe [Globale JavaScript-Methoden](#)).

## 2.5.4 Listenansicht

Wie Sie die Listenansicht konfigurieren und verwenden.

### 2.5.4.1 Datasource-Definition für die Listenansicht

#### 2.5.4.1.1 Datasource-Definition anlegen

1. *X4 Web Designer* öffnen.
2. Im Web-Anwendungsprojekt, für das die Aktionsleiste erzeugt werden soll, den Ordner *Project > portal > datasource* in der Ordnerstruktur auswählen.
3. Auf **New File** klicken.
  - Ein Pop-up-Fenster öffnet sich.
4. Einen Namen für die neue Datei angeben und **Ok** klicken.
  - Eine neue Aktionsleisten-Datei wird angelegt.

#### 2.5.4.1.2 Datasource-Definition bearbeiten

Eine *X4 Activities*-Datasource beschreibt eine Datenzeile, bestehend aus mehreren Datasource-Feldern. Eine Datasource ist immer Teil der Antwort auf eine Datenanfrage (siehe [Datenanfrage für Seiten der Vorgangsliste](#)).

1. *X4 Web Designer* öffnen.
2. Datasource-Definition, die bearbeitet werden soll, in der Ordnerstruktur auswählen.
  - Die Datei wird geöffnet.

In den Einstellungen der Datasource-Definition gibt es mehrere Bereiche:


Bereich	Beschreibung
Database Connection	Enthält Angaben zur Datenbank-Verbindung ( <a href="#">Database Connection</a> ).
Datasource Fields	Enthält Angaben zu den Datenfeldern in der Datenquelle ( <a href="#">Datasource Fields</a> ).
Detail Page	Enthält Angaben zur Detailseite, die angezeigt werden soll ( <a href="#">Detail Page</a> ).


### 2.5.4.1.3 Database Connection

Bereich	Beschreibung
ID	<i>Erforderlich.</i> Technische ID der Datenbank-Verbindung  <b>Mögliche Werte:</b> Zeichenkette aus alphanumerischen Zeichen
Datasource ID	<i>Erforderlich.</i> Technische ID der Quelle  <b>Mögliche Werte:</b> Zeichenkette aus alphanumerischen Zeichen
Source	<i>Erforderlich.</i> Frei verwendbares Kategorisierungsmerkmal.




### 2.5.4.1.4 Datasource Fields

Ein Datasource-Feld beschreibt eine Spalte in der Vorgangstabelle bzw. ein Feld in einer Datenzeile. Felder werden mit einer Reihe von Einstellungen spezifiziert, die sowohl das Aussehen und das Verhalten in der Benutzeroberfläche definieren als auch das Verhalten für die Datasource bestimmen.

Einstellung	Beschreibung
ID	<i>Erforderlich.</i> Technische ID  <b>Mögliche Werte:</b> Zeichenkette aus alphanumerischen Zeichen
Display Name	Anzeigename (siehe auch Internationalisierung)  <b>Mögliche Werte:</b> Beliebige Zeichenkette (Standard: <code>id</code> )
UI Type	<i>Erforderlich.</i> Datentyp des Felds  <b>Mögliche Werte:</b> siehe <a href="#">Feld-Datentypen (UI Type)</a>
Length	<i>Erforderlich.</i> Anzeigelänge des Feldes, wird zur relativen Breitenregelung des Feldes verwendet.  <b>Mögliche Werte:</b> Ganzzahl (Anzahl Zeichen)
Primary Key	Primärschlüssel der Tabelle   Es wird nur ein Primärschlüssel pro Datasource unterstützt.  <b>Mögliche Werte:</b> <code>true</code> / <code>false</code> (Standard)

Einstellung	Beschreibung
Title Column	Feld Displayname liefert Titel für die DataRow, findet Verwendung für Tab-Titel oder Breadcrumb-Titel. Falls nicht definiert, wird der Displayname des Primärschlüsselfelds der Datasource benutzt.  <b>Mögliche Werte:</b> true / false (Standard)
Sortable	Feld ist sortierbar.  <b>Mögliche Werte:</b> true (Standard) / false
Filterable	Feld ist filterbar.  <b>Mögliche Werte:</b> true (Standard) / false
Required	Feldwert muss zwingend gesetzt sein (entspricht <i>NOT NULL</i> bei Datenbanken).  <b>Mögliche Werte:</b> true / false (Standard)
Hidden	Spalte soll in der Tabelle nicht dargestellt werden.  <b>Mögliche Werte:</b> true / false (Standard)
Number Format	Definiert das Zahlenformat für Zahlspalten (INTEGER, FLOAT, DECIMAL), z. B. 0.00 € für einen Betrag in Euro mit zwei Nachkommastellen.  <b>Mögliche Werte:</b> Zahlenformat-Definition (erlaubte Zeichen siehe <a href="#">Erlaubte Zeichen für Zahlenformat-Definitionen</a> )   Dieses Feld wird nur angezeigt, wenn der Datentyp des Feldes ein Zahlenformat ist.

Innerhalb des Datasource-Fields-Bereichs gibt es drei verschiedene Buttons, mit denen das Datasource Field als Ganzes bearbeitet werden kann:

-  Datenfeld duplizieren
-  Datenfeld löschen
-  Datenfeld verschieben (Drag & Drop)

#### 2.5.4.1.4.1 Feld-Datentypen (UI Type)

Folgende Datentypen für Spalten sind für die Einstellung UI Type verfügbar:

- STRING: Text
- INTEGER: Ganzzahl (entspricht Java Long)
- FLOAT: Gleitkommazahl (entspricht Java Double)
- DECIMAL: Gleitkommazahl (entspricht Java BigDecimal)
- BOOLEAN: Ja/Nein bzw. true/false
- DATE: Datum (nach ISO 8601: YYYY-MM-DD, z. B. 2016-03-31)

- DATETIME: Datum mit Zeit (nach ISO 8601: YYYY-MM-DD'T'hh:mm:ss, z. B. 2016-03-31T15:21:37)
- ENUM\_TEXT: textbasierte Aufzählung, Wertdefinition siehe [Aufzählungswerte](#)
- ENUM\_IMAGE: bildbasierte Aufzählung, Wertdefinition siehe [Aufzählungswerte](#)
- IMAGE: URL, die auf ein Bild verweist
- URI: URL, die als Ziel für einen Hyperlink dient. Optional kann durch Whitespace getrennt eine Text- oder eine Icon-URL konfiguriert werden. Andernfalls wird der Default-Wert verwendet, siehe [URI-Spalte](#)

#### 2.5.4.1.4.2 Erlaubte Zeichen für Zahlenformat-Definitionen

Folgende Zeichen sind für Zahlenformat-Definitionen verfügbar:


- 0: Zahl
- #: Zahl, ohne führende 0
- .: Dezimaltrennzeichen oder Trennzeichen für Geldbeträge
- -: Minuszeichen
- ,: Tausendertrennzeichen
- E: Trennt Mantisse und Exponent in naturwissenschaftlichen Darstellungen
- %: mit 100 multiplizieren und als Prozent anzeigen
- ‰: mit 1000 multiplizieren und als Promille anzeigen
- ¤: allgemeines Währungssymbol, wird durch ein Währungssymbol ersetzt

Weitere Information finden sich in den Vorgaben von [GWT NumberFormat](#).

#### 2.5.4.1.4.3 Aufzählungswerte

Bei Feldern vom Aufzählungstyp ENUM\_TEXT oder ENUM\_IMAGE ist eine Auflistung der möglichen Werte erforderlich. In Abhängigkeit vom gewählten Typ müssen die entsprechenden Einstellungen getätigt werden.

Einstellung	Beschreibung
<i>id</i>	<i>Erforderlich.</i> Technische ID  <b>Mögliche Werte:</b> Zeichenkette aus alphanumerischen Zeichen
<i>displayName</i>	internationalisierter Titel  <b>Mögliche Werte:</b> Beliebige Zeichenkette (Standard: <i>id</i> )
<i>fgColor</i>	Schriftfarbe  <b>Mögliche Werte:</b> HTML-RGB-Farbcode (z. B. #FFDD22)
<i>bgColor</i>	Hintergrundfarbe  <b>Mögliche Werte:</b> HTML-RGB-Farbcode (z. B. #FFDD22 )

Einstellung	Beschreibung
<i>imageUri</i>	<p><i>Erforderlich.</i> URI einer Bild-Ressource</p> <p> Damit die Spalte nicht zu groß wird, sollten idealerweise Bilder in der Größe 16x16 Pixel verwendet werden.</p> <p><b>Mögliche Werte:</b> Zeichenkette (URI)</p>

#### 2.5.4.1.4.4 URI-Spalte

Bei Feldern vom Spaltentyp URI wird zur Konfiguration der Links ein Label-Element benötigt. Das Label-Element kann über folgende Attribute verfügen:

Einstellung	Beschreibung
<i>type</i>	<p><i>Erforderlich.</i> Label-Typ</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• TEXT: Der Link wird als Text dargestellt.</li> <li>• ICON: Der Link wird als Icon dargestellt.</li> </ul>
<i>default</i>	Hier kann ein Default-Wert für Labels definiert werden. Bei ICON wird die URL eines Icons erwartet, bei TEXT der entsprechende Link-Text.

#### 2.5.4.1.5 Detail Page

Bereich	Beschreibung
Base URL	Enthält die URL zur Detail-Seite, die angezeigt werden soll.

#### 2.5.4.1.6 Zeilen

Innerhalb des DataSource-Elements können zusätzlich zu den Spalten-Definitionen auch Zeilen-Typen definiert werden, um einzelne Zeilen in der Liste besonders hervorzuheben. Diese werden durch ein RowTypes-Element definiert, das für jeden Zeilentyp ein RowType-Kindelement enthält. Diese Einstellungen können in der JSON-Ansicht des *X4 Web Designers* vorgenommen werden.

##### 2.5.4.1.6.1 Attribute in Element <RowType>

Name	Beschreibung
<i>id</i>	<p><i>Erforderlich.</i> Technische ID</p> <p><b>Mögliche Werte:</b> Zeichenkette aus alphanumerischen Zeichen</p>

Name	Beschreibung
<i>styles</i>	<p>Durch Leerzeichen getrennte Style-Klassen.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• <b>bold</b>: Fett gedruckter Text</li> <li>• <i>italic</i>: Kursiv gedruckter Text</li> <li>• <b>info</b>: Farbliche Hervorhebung als Information</li> <li>• <b>success</b>: Farbliche Hervorhebung als Erfolg</li> <li>• <b>error</b>: Farbliche Hervorhebung als Fehler</li> <li>• <b>warning</b>: Farbliche Hervorhebung als Warnung</li> </ul>

#### 2.5.4.1.6.2 Beispiel

Beispiel
<pre> &lt;DataSource id="zpm02" source="zpm"&gt;   ...   &lt;RowTypes&gt;     &lt;RowType id="new" styles="bold info"/&gt;     &lt;RowType id="changed" styles="italic warning"/&gt;     &lt;RowType id="unchanged" /&gt;   &lt;/RowTypes&gt; &lt;/DataSource&gt; </pre>

#### 2.5.4.2 Datenanfrage für Seiten der Vorgangsliste

Die Vorgangsliste lässt sich in Abhängigkeit von der Tabellensortierung auf einzelne Seiten gruppieren (z. B. 20 pro Seite). Zwischen den einzelnen Seiten kann der Anwender über Schaltflächen navigieren. Einzelne Seiten werden durch Änderungen in der Listenansicht angefordert.

Mögliche Auslöser sind:

- Filter setzen
- Sortierung ändern
- Nächste Seite anfordern

##### 2.5.4.2.1 Seitenanforderung (PageRequest)

Ein PageRequest beinhaltet die Information darüber, welche Seite gezeigt werden soll, welche Spalte sortiert wird und welche Filter gesetzt wurden. Die Definition wird von der Web-Applikation als XML-Dokument geliefert, dessen Struktur dem Schema `pageRequestResponse.xsd` entspricht. Dieses befindet sich im X4Common-Modul unter `X4Common/Activities/schema/datasource/parser`.

**Beispiel für einen PageRequest**

```
<PageRequest>
  <Paging>
    <StartRow>0</StartRow>
    <Size>10</Size>
  </Paging>
  <Sorting>
    <SortField id="AufgabeId" dir="DESC" />
  </Sorting>
  <Filtering>
    <FilterEntry>
      <EnumFilter id="EnumImage">
        <EnumFilterItem id="green" />
        <EnumFilterItem id="yellow" />
      </EnumFilter>
    </FilterEntry>
    <FilterEntry>
      <DecimalFilter id="Dezimal" min="3.23" max="50.45" />
    </FilterEntry>
  </Filtering>
</PageRequest>
```

Folgende Elemente müssen immer vorhanden sein:

1. **Paging**: Steuert die Seitenanzeige (Erforderlich, da sonst keine sinnvolle Antwort möglich ist)
2. **Sorting**: Sortierkriterium (kann leer sein)
3. **Filtering**: Filterkriterium (kann leer sein)

#### 2.5.4.2.2 Seitenantwort

Eine Antwort für den PageRequest beinhaltet immer eine Beschreibung des Formates in Form einer Datasource und die eigentlichen Daten. Dies erlaubt es dem Lister zu prüfen, ob die Antwort zum momentanen Format passt und erlaubt einfaches Parsing, da alle Informationen vorhanden sind.




**Beispiel für eine PageResponse**

```

<PageResponse>
  <DataSource id="zpm01" source="zpm">
    <FieldDefinitions>
      <FieldDef id="AufgabeId" displayName="AufgabeId" type="INTEGER"
length="20" primaryKey="true" required="true" />
      <FieldDef id="Postkorb" displayName="Postkorb" type="STRING"
length="30" primaryKey="false" required="true" />
      <FieldDef id="Fachbereich" displayName="Fachbereich" type="STRING"
length="30" primaryKey="false" required="true" />
      <FieldDef id="IdentNr" displayName="IdentNr" type="STRING"
length="50" primaryKey="false" required="true" />
      <FieldDef id="VorgangId" displayName="VorgangId" type="INTEGER"
length="20" primaryKey="false" required="true" />
      <FieldDef id="Bereitstellungsdatum" displayName="Bereitstellungsdatum"
type="DATE" length="23" required="true" />
      <FieldDef id="Eingangsdatum" type="DATE" length="23" primaryKey="false"
required="true" />
    </FieldDefinitions>
  </DataSource>
  <Paging>
    <StartRow>4</StartRow>
    <Size>2</Size>
    <TotalRows>118</TotalRows>
  </Paging>
  <Sorting>
    <SortField id="AufgabeId" dir="DESC"/>
  </Sorting>
  <DataRows>
    <DataRow detailUri="/X4/httpstarter/ReST/GenericDetail/27">
      <Field id="AufgabeId">27</Field>
      <Field id="Postkorb">1</Field>
      <Field id="Fachbereich" />
      <Field id="IdentNr" />
      <Field id="VorgangId">52</Field>
      <Field id="Bereitstellungsdatum" />
      <Field id="Eingangsdatum">2013-03-17</Field>
    </DataRow>
    <DataRow detailUri="/X4/httpstarter/ReST/GenericDetail/28">
      <Field id="AufgabeId">28</Field>
      <Field id="Postkorb">1</Field>
      <Field id="Fachbereich" />
      <Field id="IdentNr" />
      <Field id="VorgangId">53</Field>
      <Field id="Bereitstellungsdatum" />
      <Field id="Eingangsdatum">2011-11-23</Field>
    </DataRow>
  </DataRows>
</PageResponse>

```

 Es wird keine Filter-Information erwartet, da diese nicht auf die Listenansicht angewendet wird.

### 2.5.4.2.2.1 Element DataRow

Das DataRow-Element fasst die Daten einer Zeile zusammen. Es muss daher für jede Felddefinition in der DataSource einen Field-Eintrag enthalten. Das Element trägt außer den fachlichen auch technische Informationen. Die technischen Informationen steuern, was bei der Selektion (einfacher Klick oder Doppelklick) eines Eintrags in der Liste geschieht.

Für die Liste stehen folgende Selektionsmöglichkeiten zur Verfügung, die sich unterschiedlich belegen lassen:

1. Listeneintrag durch einfaches Anklicken wählen.
2. Listeneintrag per Doppelklick wählen.

Das <DataRow>-Element besitzt folgende Attribute:

Attribut	Beschreibung
<i>detailUri</i>	<p><i>Erforderlich.</i> ReST-URI für die Inhalte der Detailansicht in der Hauptansicht unter der Liste</p> <p><b>Mögliche Werte:</b> String (URI)</p>
<i>dblClickUri</i>	<p>ReST-URI für Inhalte, die nach einem Doppelklick auf den Listeneintrag angezeigt werden sollen. Dieses Attribut kann nur gemeinsam mit dem Attribut <i>dblClickType</i> verwendet werden. Falls <i>dblClickUri</i> nicht vorhanden ist, wird nach Doppelklick die ReST-URI aus <i>detailUri</i> verwendet und in einem neuem Tab geöffnet.</p> <p><b>Mögliche Werte:</b> String (URI)</p>
<i>dblClickType</i>	<p>Aufzählungstyp</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• EDIT_DETAIL_URI (Standard, um Abwärtskompatibilität zu gewährleisten)</li> <li>• BASE_URI: Doppelklick auf einen Listeneintrag löst Initialisieren des Navigators aus. Damit kann ein hierarchischer Abstieg in baumartige Objekte erfolgen.</li> <li>• DISABLED: Doppelklick deaktivieren</li> </ul>
<i>preSelection</i>	<p>Ermöglicht eine Vorselektion des Listeneintrags.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li>• NONE: Keine Selektion (Standard)</li> <li>• SINGLE: Selektiert diese Zeile</li> <li>• MULTI: Setzt den Haken für Multi-Selektion, falls dies in der Tabelle unterstützt wird</li> <li>• ALL: Kombiniert SINGLE und MULTI</li> </ul>

Attribut	Beschreibung
<i>rowType</i>	ID, die auf einen RowType-Eintrag der <i>Datasource</i> verweist. Dadurch ist eine besondere Hervorhebung einzelner Zeilen möglich.

#### 2.5.4.2.2.2 Selektions-Verhaltensmöglichkeiten

Attribut(e) für <b>&lt;DataRow&gt;</b>	Verhalten bei einfachem Klick	Verhalten bei Doppelklick
<i>detailUri</i> ="<URI>"	Eine Detailansicht öffnen.	Eine Detailansicht in einem neuen Tab öffnen.
<i>detailUri</i> ="<URI>" <i>dblClickUri</i> ="<URI>"	Eine nicht-editierbare Detailansicht (definiert in <i>detailUri</i> ) öffnen.	Eine Editiermaske (definiert in <i>dblClickType</i> ) in neuem Tab öffnen.
<i>detailUri</i> ="<URI>" <i>dblClickUri</i> ="<URI>" <i>dblClickType</i> ="BASE_URI"	Eine Detailansicht öffnen, die als Übersicht einen hierarchischen Vorgang mit Untervorgängen anzeigt.	Die Ansicht wechseln und Navigator und Liste durch den hierarchischen Vorgang und dessen Untervorgänge austauschen.

**Auf dieser Seite:**

**Siehe auch:**

- [Filterkriterien](#)

### 2.5.4.3 Filtern von Listeneinträgen

Auf einer Spalte kann ein Filter abhängig vom Spaltentyp eingerichtet werden. Der Tabelleninhalt wird durch die Menge der Filter über alle Spalten bestimmt.

#### 2.5.4.3.1 Filtertypen

In folgender Tabelle werden die möglichen Filter-Werte und -XML-Strukturen der verschiedenen Datentypen beschrieben. Die *id*-Attribute müssen dabei die ID der Spalte enthalten, auf die sich der Filter bezieht. Die Datentypen der Spalten werden in der Datasource-Definition konfiguriert.

Spaltentyp	Filterdefinition	Umsetzung in der XML-Struktur
<b>BOOLEAN</b>	true oder false	<pre>&lt;BooleanFilter id="&lt;ID&gt;" value="true false"/&gt;</pre>

Spaltentyp	Filterdefinition	Umsetzung in der XML-Struktur
<b>STRING</b>	<p>Über das dem Attribut operator die Art der Filterung steuern:</p> <ul style="list-style-type: none"> <li>• <code>startsWith()</code>-Semantik von Java</li> <li>• <code>endsWith()</code>-Semantik von Java</li> <li>• <code>equals()</code>-Semantik von Java</li> <li>• <code>contains()</code>-Semantik von Java</li> </ul>	<pre>&lt;StringFilter id="&lt;ID&gt;" operator="&lt;operator&gt;"&gt;&lt;![CDATA[ &lt;Text&gt; ]]&gt;&lt;/StringFilter&gt;</pre>
<b>DATE</b>	Wertebereich(a .. b)oder(<= k)oder(>= k)	<pre>&lt;DatetimeFilter id="&lt;ID&gt;" min="YYYY-MM-DDThh:mm:ss" max="YYYY-MM-DDThh:mm:ss"/&gt;</pre>
<b>DATETIME</b>	(wie DATE)	(wie DATE)
<b>FLOAT</b>	Wertebereich(a .. b)oder(<= k)oder(>=k)	<pre>&lt;DecimalFilter id="&lt;ID&gt;" min="#.#" max="#.#"/&gt;</pre>
<b>DECIMAL</b>	(wie FLOAT)	(wie FLOAT)
<b>INTEGER</b>	(wie FLOAT)	<pre>&lt;IntegerFilter id="&lt;ID&gt;" min="#" max="#"/&gt;</pre>
<b>ENUM</b>	Auswahl der Werte für den Filter anhand ihrer technischen ID	<pre>&lt;EnumFilter id="&lt;ID&gt;"&gt;   &lt;EnumFilterItem id="&lt;ID 1&gt;" /&gt;   &lt;EnumFilterItem id="&lt;ID 2&gt;" /&gt;   &lt;EnumFilterItem id="&lt;ID n&gt;" /&gt; &lt;/EnumFilter&gt;</pre>

### Volltextsuche

Für (Volltext-)Suche über mehrere Spalten wird, sofern dies in der Navigator-Konfiguration aktiviert ist, ein zusätzliches Element für "globale Suche" bereitgestellt.

### 2.5.4.3.2 Beispiel

```
<Filtering>
  <GlobalSearch><![CDATA[Ein Suchbegriff]]></GlobalSearch>
  <FilterEntry>
```

```
<StringFilter id="text_1" operator="contains">
  <![CDATA[aha!]]>
</StringFilter>
</FilterEntry>
<FilterEntry>
  <BooleanFilter id="boolean_1" value="true"/>
</FilterEntry>
<FilterEntry>
  <DatetimeFilter id="date_1" min="2012-03-13T15:25:55" max="2012-03-13T15:26:05"/>
>
</FilterEntry>
<FilterEntry>
  <EnumFilter id="enum_1">
    <EnumFilterItem id="one"/>
    <EnumFilterItem id="two"/>
    <EnumFilterItem id="four"/>
  </EnumFilter>
</FilterEntry>
<FilterEntry>
  <IntegerFilter id="ganz" min="3" max="15"/>
</FilterEntry>
<FilterEntry>
  <DecimalFilter id="gleit" min="0.1" max="10.1"/>
</FilterEntry>
</Filtering>
```

#### Auf dieser Seite:

#### Siehe auch:

- [Datasource-Definition für die Listenansicht](#)
- [Datenanfrage für Seiten der Vorgangsliste](#)
- [Globale Suche](#)

### 2.5.4.4 Export-Funktion für die Listenansicht

*X4 Activities* bietet die Möglichkeit, die Daten einer Lister-Tabelle auch als CSV-Download zur Verfügung zu stellen.

#### 2.5.4.4.1 Download-Button konfigurieren

In der [Aktionsleiste](#) muss ein entsprechender Download-Button mit folgender Konfiguration hinzugefügt werden:

- **buttonId:** frei wählbar, z. B. *downloadID*
- **displayName:** frei wählbar, z. B. *Exportieren*
- **iconUrl:** */X4PSupport/images/download.png*
- **processUrl:** */X4/httpstarter/ReST/x4p/ListerExport*
- **actionType:** *POST*
- **displayType:** *DOWNLOAD*

#### 2.5.4.4.2 CSV-Inhalt und -Format

Die erzeugte CSV-Datei enthält alle Datensätze des angezeigten Listers unter Berücksichtigung der aktuellen Filter- und Sortier-Einstellungen. Die Datenmenge ist dabei auf maximal 5000 Datensätze beschränkt. Bei Bedarf kann die Beschränkung jedoch über den globalen Platzhalter `LISTEREXPORT_MAX_SIZE` angepasst werden, z. B.

`-Dx4.placeholder.LISTEREXPORT_MAX_SIZE=6000`

Datumsspalten und numerische Spalten werden je nach Spracheinstellung entsprechend formatiert, analog zu den Formaten in der Lister-UI:

Sprache	Datum	Datum und Zeit	Dezimaltrenner
Deutsch	dd.MM.yyyy	dd.MM.yyyy HH:mm	,
Englisch	yyyy-MM-dd	yyyy-MM-dd HH:mm	.

#### 2.5.4.4.3 CSV-Dateiname

Der beim Download verwendete Dateiname lautet standardmäßig `export.csv`. Um einen anderen Namen zu verwenden, kann bei der Angabe der `processUrl` die zusätzliche Pfadangabe `/X4/httpstarter/ReST/x4p/ListerExport/<filename>` genutzt werden.

*Beispiel:* `/X4/httpstarter/ReST/x4p/ListerExport/taskliste.csv`

### 2.5.5 Detailmasken

Im Folgenden wird beschrieben, wie Sie manuell, d. h. ohne den *X4 Web Designer*, Detailseiten für *X4 Activities*-Anwendungen erstellen können, beispielsweise um dynamisch HTML zu generieren. Detailmasken werden als `iFrame` in die Detailansicht der Activities-Anwendung eingebunden. Damit lassen sich neben beliebigen HTML-Seiten auch PDF-Dokumente oder sonstige Web-Inhalte darstellen. In der Regel enthalten Detailmasken Daten zum ausgewählten Listeneintrag als HTML-Formular.

#### 2.5.5.1 Erstellung von HTML-Detailmasken

HTML-Detailmasken können entweder durch X4-Prozesse, über die X4 ReST-Schnittstelle oder durch beliebige andere per URL erreichbare Quellen bereitgestellt werden. Für die Bereitstellung aus Prozessen gibt es zwei unterschiedliche Ansätze:

1. Dynamische Generierung des HTML-Codes mit XSLT
2. Statische HTML-Seite und dynamisches Nachladen der Daten mit JavaScript

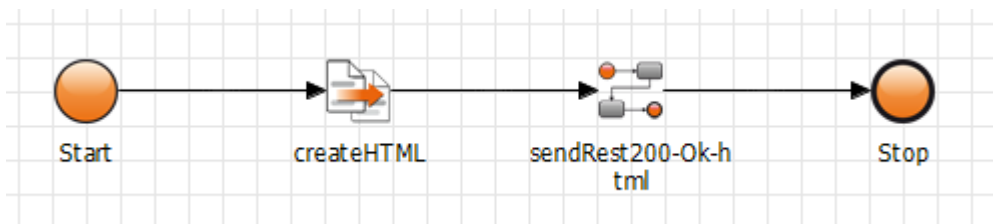
Um aus einem Prozess HTML zu liefern, kann der Prozess `sendRest200-0k-html.wrf` aus dem `X4Common`-Modul verwendet werden, siehe [X4Common-Prozessbibliothek für X4 Activities](#). Vor diesem Prozess kann entweder ein (X)HTML-Dokument (statisches HTML) oder ein XSLT-Mapping (dynamisches HTML) eingebunden werden.

### 2.5.5.1.1 Prozesse für dynamisches HTML

Um Daten direkt in eine HTML-Seite einzufügen, kann das HTML dynamisch per XSLT erzeugt werden. Als Vorlage für ein solches Mapping kann das folgende Beispiel dienen. Allerdings werden in einem realen Detailmasken-Prozess in der Regel vor der HTML-Generierung noch Daten beispielsweise aus einer Datenbank oder einem Fremdsystem geladen.

Bei großen Datenmengen oder komplexen HTML-Seiten ist diese Vorgehensweise nicht empfehlenswert, da das Mapping sehr unübersichtlich werden kann.

**Beispiel:**



#### createHTML.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="html" media-type="text/html"></xsl:output>
  <xsl:variable select="/Request/Query/Parameter" name="urlParams"></xsl:variable>

  <xsl:template match="/">
    <xsl:text disable-output-escaping="yes">&lt;!DOCTYPE html&gt;</xsl:text>
    <html>
      <head>
        <title>My Page</title>
      </head>
      <body>
        <h1>My Page</h1>
        <p>
          <xsl:text>Value of URL parameter 'test': </xsl:text>
          <xsl:value-of select="$urlParams['test']"></xsl:value-of>
        </p>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
<!--Created by X4 Designer, Copyright © SoftProject GmbH. All rights reserved.-->
  
```

### 2.5.5.1.2 Prozesse für statisches HTML

Statt das HTML mithilfe eines Mappings zu erzeugen, kann ein X4 Prozess auch einfach eine HTML-Datei aus der X4DB ausliefern. Das bietet vor allem dann Vorteile, wenn die Seite keine dynamischen Inhalte hat (z. B. nur eine Erfolgsmeldung) oder wenn die Daten in einem separaten Prozess bereitgestellt werden sollen.

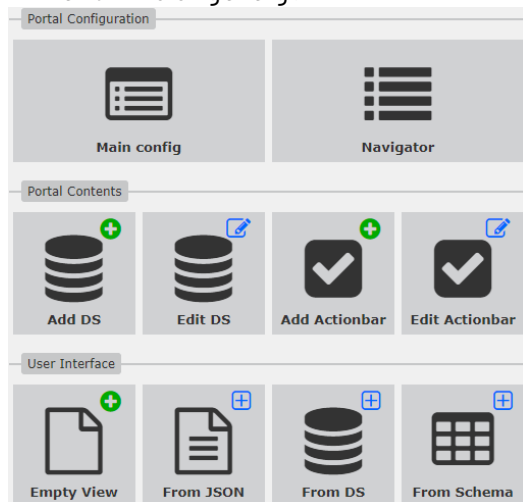
In diesem Fall können die Daten von einem weiteren X4 Prozess als JSON bereitgestellt werden (z. B. mit Hilfe des *JSON Converter 2.0*). Diese können dann mit JavaScript per Ajax abgerufen und in die Seite eingefügt werden.

## 2.5.6 Main Config

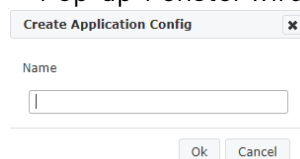
Mit der Hauptkonfiguration *Main Config* wird die grundlegende Konfiguration der Web-Anwendung vorgenommen. Dazu zählt z. B. die URL des Hauptnavigators, die Inhalte der *Application Bar*, das Logo der Hauptanwendung oder spezielle Links.

### Application Config konfigurieren

1. Projekt öffnen  
→ Menü wird angezeigt







2. **Main config** klicken  
→ Pop-up-Fenster wird angezeigt.



3. Name für die Main Config eingeben und **Ok** klicken.  
→ Main Config wird im *Application Config Editor* angezeigt.  
In den Einstellungen der Main Config gibt es mehrere Bereiche:



Bereich	Beschreibung
<i>Main Navigator URL</i>	<p><i>Erforderlich.</i> Enthält die URL für den Hauptnavigator. Der Hauptnavigator wird auf der linken Seite der Web-Applikation angezeigt.</p> <p><b>Mögliche Werte:</b> Zeichenkette (URL)</p> <div style="border: 1px solid green; padding: 10px; margin-top: 10px;"> <p>✓ Die URL wird über <b>Copy Result URL</b> bei der Bearbeitung des <b>Navigators</b> in die Zwischenablage kopiert und muss hier eingefügt werden.</p> </div>
<i>Logo URL</i>	<p>Enthält die URL für das Logo der Hauptapplikation. Das Logo wird in der oberen linken Ecke in der <i>Application Bar</i> angezeigt.</p>  <p><b>Mögliche Werte:</b> Zeichenkette (URL)</p>
<i>Applications</i>	Enthält Einstellungen zu den in der Web-Anwendung enthaltenen Applikationen. Siehe Applications.
<i>External Links</i>	Definiert externe Links. Siehe External Links
<i>Help Links</i>	Definiert Links, die per Klick auf den Hilfe-Button  aufgerufen werden können. Siehe Help Links.
<i>User Links</i>	Definiert Links, die per Klick auf den Benutzer-Button  <b>SP Admin</b> aufgerufen werden können. Siehe User Links.

4. Gewünschte Änderungen vornehmen.
5. Auf **Save** ( **Save**) klicken, um zu speichern.  
→ Die Änderungen wurden gespeichert und werden beim nächsten Aufruf der Web-Anwendung verwendet.




### 2.5.6.1 Applications

Im Bereich *Applications* können Einstellungen für die Applications innerhalb der Web-Anwendung festgelegt werden. Definierte Applications werden als Menüeintrag in der *Application Bar* angezeigt.

Einstellung	Beschreibung
App ID	<p><i>Erforderlich.</i> ID der Application. Die ID muss innerhalb einer Web-Anwendung eindeutig sein.</p> <p>Mögliche Werte: Zeichenkette aus alphanumerischen Zeichen</p>

Einstellung	Beschreibung
Display Name	<i>Erforderlich.</i> Name der Application, der für die Anzeige in der <a href="#">Application Bar</a> verwendet wird. <b>Mögliche Werte:</b> Beliebige Zeichenkette
Is Default App	Legt fest, ob die Application als Standard-Application geladen wird. <b>Mögliche Werte:</b> true / false (Standard)
Content URL	URL zur <a href="#">Navigator-Konfiguration</a> der Anwendung bzw. zur externen Anwendung, die aufgerufen werden soll.
Content Type	Legt fest, wie die angegebene URL geöffnet werden soll. <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• NAVIGATOR (Standard): lädt Navigator-Config</li> <li>• EXTERNAL_LINK: Öffnet die angegebene URL in einem neuen Fenster</li> </ul>
Icon URL	Definiert das Icon, das neben dem Namen der Anwendung angezeigt werden soll (empfohlene Größe: 16x16 Pixel). <b>Mögliche Werte:</b> Zeichenkette (URL)

Innerhalb des Applications-Bereichs gibt es drei verschiedene Buttons, mit denen die Application als Ganzes bearbeitet werden kann:

-  Application duplizieren
-  Application löschen
-  Application verschieben (Drag & Drop)



### 2.5.6.2 External Links

In einer Web-Anwendung können auch externe Links eingebunden werden. Diese werden in der Application Bar angezeigt.

Einstellung	Beschreibung
Display Name	<i>Erforderlich.</i> Name des Links, der für die Anzeige in der <a href="#">Application Bar</a> verwendet wird. <b>Mögliche Werte:</b> Beliebige Zeichenkette
URL	<i>Erforderlich.</i> URL, auf die verlinkt werden soll. <b>Mögliche Werte:</b> Zeichenkette (URL)
Icon URL	URL zu einem Icon, das neben dem Link angezeigt werden soll (empfohlene Größe: 16x16 Pixel). <b>Mögliche Werte:</b> Zeichenkette (URL)

Einstellung	Beschreibung
Link Target	Gibt an, in welcher Form der Link geöffnet werden soll.  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• Default: (Standard)</li> <li>• Self: Öffnet den Link im aktuellen Fenster.</li> <li>• Blank: Öffnet den Link in einem neuen Fenster/Tab.</li> <li>• Dialog: Öffnet den Link in einem GWT-Dialog.</li> <li>• Modal Dialog: Öffnet den Link in einem modalen GWT-Dialog.</li> </ul>
Dialog Width	Breite des Dialogfensters. Wird nur angezeigt, wenn <b>Link Target</b> ein Dialog ist.
Dialog Height	Höhe des Dialogfensters. Wird nur angezeigt, wenn <b>Link Target</b> ein Dialog ist.



Innerhalb des External-Links-Bereichs gibt es zwei verschiedene Buttons, mit denen die Application als Ganzes bearbeitet werden kann:

-  External Link duplizieren
-  External Link löschen

### 2.5.6.3 Help Links

Einstellung	Beschreibung
Display Name	<i>Erforderlich.</i> Name des Links, der für die Anzeige in der <a href="#">Application Bar</a> verwendet wird. <b>Mögliche Werte:</b> Beliebige Zeichenkette
URL	<i>Erforderlich.</i> URL, auf die verlinkt werden soll.  <b>Mögliche Werte:</b> Zeichenkette (URL)
Icon URL	URL zu einem Icon, das neben dem Link angezeigt werden soll (empfohlene Größe: 16x16 Pixel).  <b>Mögliche Werte:</b> Zeichenkette (URL)
Link Target	Gibt an, in welcher Form der Link geöffnet werden soll.  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• Default: (Standard)</li> <li>• Self: Öffnet den Link im aktuellen Fenster.</li> <li>• Blank: Öffnet den Link in einem neuen Fenster/Tab.</li> <li>• Dialog: Öffnet den Link in einem GWT-Dialog.</li> <li>• Modal Dialog: Öffnet den Link in einem modalen GWT-Dialog.</li> </ul>
Dialog Width	Breite des Dialogfensters. Wird nur angezeigt, wenn <b>Link Target</b> ein Dialog ist.
Dialog Height	Höhe des Dialogfensters. Wird nur angezeigt, wenn <b>Link Target</b> ein Dialog ist.



Innerhalb des Help-Links-Bereichs gibt es zwei verschiedene Buttons, mit denen die Application als Ganzes bearbeitet werden kann:

-  Help Link duplizieren
-  Help Link löschen

#### 2.5.6.4 User Links

Einstellung	Beschreibung
Display Name	<i>Erforderlich.</i> Name des Links, der für die Anzeige in der <b>Application Bar</b> verwendet wird. <b>Mögliche Werte:</b> Beliebige Zeichenkette
URL	<i>Erforderlich.</i> URL, auf die verlinkt werden soll. <b>Mögliche Werte:</b> Zeichenkette (URL)
Icon URL	URL zu einem Icon, das neben dem Link angezeigt werden soll (empfohlene Größe: 16x16 Pixel). <b>Mögliche Werte:</b> Zeichenkette (URL)
Link Target	Gibt an, in welcher Form der Link geöffnet werden soll. <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• Default: (Standard)</li> <li>• Self: Öffnet den Link im aktuellen Fenster.</li> <li>• Blank: Öffnet den Link in einem neuen Fenster/Tab.</li> <li>• Dialog: Öffnet den Link in einem GWT-Dialog.</li> <li>• Modal Dialog: Öffnet den Link in einem modalen GWT-Dialog.</li> </ul>
Dialog Width	Breite des Dialogfensters. Wird nur angezeigt, wenn <b>Link Target</b> ein Dialog ist.
Dialog Height	Höhe des Dialogfensters. Wird nur angezeigt, wenn <b>Link Target</b> ein Dialog ist.

Innerhalb des User-Links-Bereichs gibt es zwei verschiedene Buttons, mit denen die Application als Ganzes bearbeitet werden kann:

-  User Link duplizieren
-  User Link löschen

## 2.6 Benutzeroberflächen erstellen mit dem X4 Web Designer

Wie Sie individuelle Benutzeroberflächen für Web-Anwendungen auf Basis von *X4 Activities* mit dem *X4 Web Designer* erstellen und welche Möglichkeiten alternativ zur Verfügung stehen

## 2.6.1 Übersicht der Werkzeuge und Methoden

Mit *X4 Activities* haben Sie sich für ein flexibles Framework für die Entwicklung von Web-Anwendungen auf Basis des *X4 Enterprise Service Bus* (ESB) und dessen ReSTful-Webservice-Schnittstelle entschieden. *X4 Activities* kann für die unterschiedlichsten Anwendungsfälle eingesetzt werden, beispielsweise für die Auftrags- und Aufgabenverwaltung, aber auch für Service-Monitoring oder Administration. Je nach Szenario stehen verschiedene Optionen zur Verfügung, wie Sie Detailseiten in Ihrer *X4 Activities-Anwendung* entwickeln und bereitstellen.

Sie haben folgende Möglichkeiten, um Benutzeroberflächen (Detailseiten) für *X4 Activities* zu erstellen:

- **UI-Editor im *X4 Web Designer*:** Mit dieser Web-Oberfläche können Sie komplexe Benutzeroberflächen grafisch per Drag & Drop in zusammenstellen, siehe [UI-Widgets einfügen und verwalten](#).  
Im *UI-Editor* lässt sich Fachlogik mit Hilfe eines JavaScript-Controllers hinterlegen, siehe [UI-Controller definieren](#).  
Hierfür können Sie die bereitgestellte JavaScript-API nutzen, siehe [JavaScript-API für den \*X4 Web Designer\*](#).  
Die Fachlogik kann über Aktions- und Daten-Bindings mit den UI-Elementen verknüpft werden, siehe [UI-Bindings](#).  
Zudem können Sie Fachlogik zu einzelnen DOM-Elementen verknüpfen, siehe [DomHook-Einträge hinterlegen](#).  
Auch externe JavaScript-Bibliotheken lassen sich als Abhängigkeiten einbinden, siehe [UI-Editor im \*X4 Web Designer\*](#) im Abschnitt *Dependencies*.
- **Statisches HTML und JavaScript mit dem *X4 Web Designer* erzeugen:** Falls Sie für die Oberflächen-Entwicklung nicht die von X4 bereitgestellten JavaScript-Frameworks nutzen möchten, ist dies im *X4 Web Designer* ebenfalls möglich. Mit den integrierten JavaScript- und HTML-Quelltexteditoren des *X4 Web Designers* können Sie direkt HTML und JavaScript mit beliebigen JavaScript-Frameworks und -Bibliotheken erstellen.
- **Mit *X4-Prozessen* dynamisch erzeugte HTML-Detailseiten:** Mit dem *X4 Designer* können Sie *X4-Prozesse* entwickeln, die als Ergebnis fertige HTML-Seiten (ggf. mit Formularen, CSS-Formatierung und JavaScript) zurückliefern. Diese *X4-Prozesse* können Sie über die Webservices-Schnittstelle des *X4 ESB* als ReSTful Webservice bereitstellen und diese als Detailseiten in Ihre *X4 Activities-Anwendung* einbinden, siehe [Datasources konfigurieren](#).
- **HTML-Seite von externen Ressourcen einbinden:** Bereits vorhandene HTML-Seiten, die Sie mit Drittwerkzeugen erzeugt und via HTTP(S) bereitgestellt haben, können Sie ebenfalls als Detailseiten in Ihre *X4 Activities-Anwendung* anhand der Ressourcen-URL einbinden, siehe [Datasources konfigurieren](#).

## 2.6.2 Grundlegende Vorgehensweise

Erforderliche Schritte, um eine Benutzeroberfläche in Ihrer *X4 Activities-Anwendung* verfügbar zu machen:

1. Mit dem Werkzeug Ihrer Wahl eine Benutzeroberfläche erstellen.  
*Beispiel:* Mit dem *UI-Editor* im *X4 Web Designer* grafisch modellieren (siehe [UI-Widgets](#)

[einfügen und verwalten](#)) oder von JSON-Daten erzeugen und konfigurieren (siehe [Formulare auf Basis von JSON-Daten erzeugen](#)).

2. Die URL zur Benutzeroberfläche kopieren und in der Konfiguration zu Ihrer *X4 Activities*-Anwendung hinterlegen:  
*Beispiel:* Im *UI-Editor* des *X4 Web Designers* auf **Show Result URL** klicken, um die entsprechende URL zu ermitteln.
  - **Detailseite zu einem Listeneintrag:** Im *Datasource Config Editor* in Eigenschaft **Base URL** (`BaseUrl`) die URL einfügen und die Konfiguration speichern.
  - **Header-Bereich im Inhaltsbereich zu einem Navigatorknoten:** Im *Navigator Config Editor* im entsprechenden Knoten für Eigenschaft **Content Header URL** (`ContentHeader.url`) die URL einfügen und die Konfiguration speichern.
  - **Inhaltsbereich zu einem Navigatorknoten:** Im *Navigator Config Editor* im entsprechenden Knoten für Eigenschaft **Content URL** (`nodeContentUrl`) die URL einfügen (**Content Type** muss **DETAIL** sein) und die Konfiguration speichern.
3. Vorschau Ihrer Web-Anwendung öffnen und diese testen, siehe [Anwendung mit dem X4 Web Designer testen und bereitstellen](#).

### 2.6.3 Formulare im UI-Editor definieren

Wie Sie Formulare innerhalb von Detailmasken für Web-Anwendungen auf Basis von *X4 Activities* mit dem *X4 Web Designer* erstellen

#### 2.6.3.1 Formulare grafisch modellieren

Wie Sie mit dem grafischen UI-Editor des *X4 Web Designers* komplexe Formulare für Detailseiten aus UI-Widgets zusammenstellen, anpassen und mittels JavaScript Fachlogik hinterlegen können  
Grundlegende Schritte, um mit dem *X4 Web Designer* grafisch Formulare zu erstellen und mit einem Datenmodell zu verknüpfen:

1. Eine Datei vom Typ `.ui.json` anlegen (üblicherweise unter `src/main/ui`) und im *UI-Editor* Widgets einfügen, siehe [UI-Widgets einfügen und verwalten](#).
2. Data Bindings für jedes Formularelement hinterlegen, siehe [Daten-Bindings für Formularelemente hinterlegen](#).
3. Mit Klick auf **Show Result URL** die Ressourcen-URL für die Detailseite kopieren.
4. Die URL der Seite an passender Stelle bereitstellen.  
Beispielsweise in einer Datasource-Konfiguration mit dem *Datasource Config Editor* in Bereich **Detail Page** in Feld **Base URL** (`BaseUrl`) die URL zur Detailseite hinterlegen, wenn das Formular als Detail-Seite für einen Lister verwendet werden soll.


#### 2.6.3.2 Formulare auf Basis von JSON-Daten erzeugen

Wie Sie im UI-Editor des *X4 Web Designers* eine beliebige JSON-Datenstruktur als Datenmodell für ein neues Formular verwenden können und automatisch die entsprechenden Formularelemente erzeugen

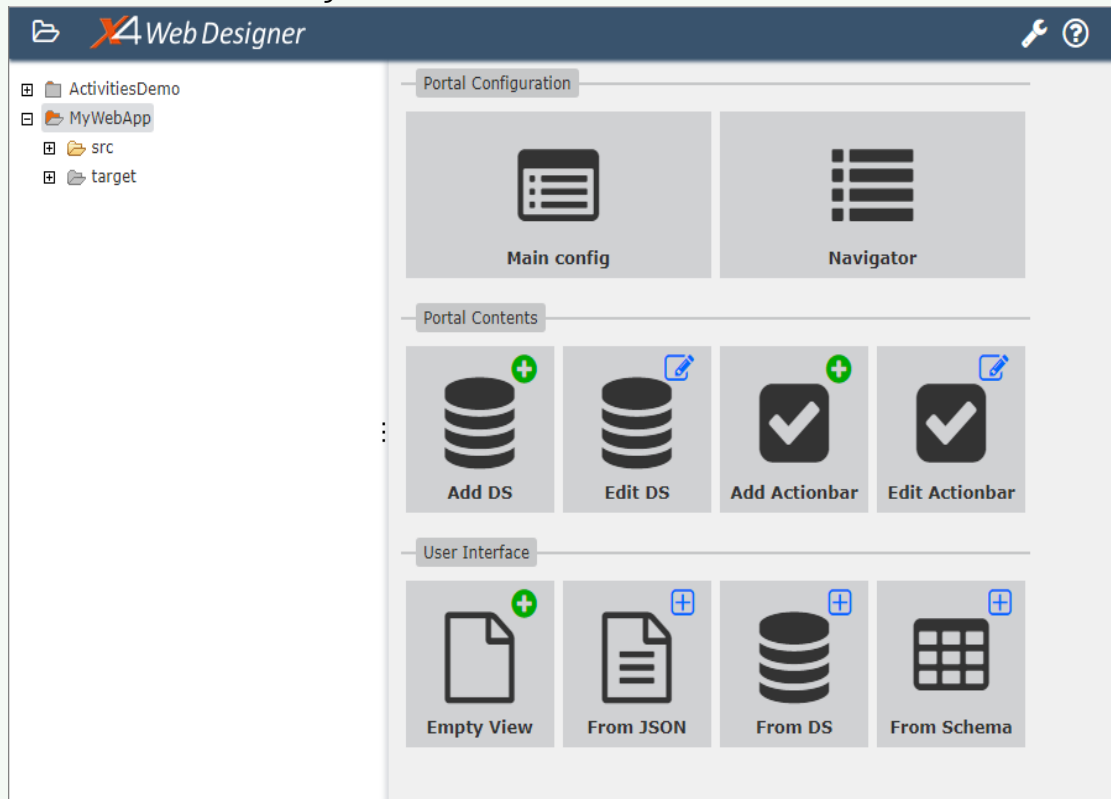
**Voraussetzungen:**

- Die gewünschte Datenstruktur liegt als JSON-Daten vor.
- Die JSON-Daten sind vom *X4 Web Designer* über eine URL erreichbar.

### Schritte, um ein Formular generieren zu lassen:

1. Im UI-Editor an der gewünschten Stelle in der Baumstruktur mit der Maus über *Add/Drop Widgets* fahren und auf  klicken.
2. **Generate Form by JSON Data** wählen.

- ✓ Alternativ lässt das Formular auch direkt über die Projekt-Startseite bzw. über die **ui**-Ordner Startseite anlegen. Dazu auf **From JSON** klicken.



3. Die URL zum JSON-Dokument einfügen und **OK** klicken.  
Auf Basis der JSON-Datenstruktur werden nun hierarchische Formularelemente in der Baumstruktur des UI-Editors wie folgt erzeugt:
  - Anhand des Datentyps wird der Formularfeldtyp bzw. UI-Widget-Typ ermittelt (z.B. *Input* bei Textdaten oder *Form Array Panels* bei Arrays)
  - Data-Bindings werden für jedes Formularfeld automatisch erzeugt und in der jeweiligen Widget-Eigenschaft **Value** hinterlegt, z.B. `{{email}}`.
  - Labels werden automatisch aus den Namen der JSON-Eigenschaften erzeugt.
4. Bei Bedarf die Feldeigenschaften in der **Properties**-Registerkarte anpassen, z.B. die automatisch erzeugten Feldbeschriftungen in Eigenschaft **Label**, siehe [UI-Widgets einfügen und verwalten](#).
5. Binding der Formulardaten an den Lister-Eintrag erstellen, siehe [Binding des Formulars an den Lister-Eintrag erstellen](#).

- ✓ Alternativ empfiehlt es sich, einen Controller zu erstellen, der das Binding des Formulars an den entsprechenden Lister-Eintrag übernimmt, siehe [UI-Controller definieren](#). In der Property für die URL kann dann ein Binding definiert werden, das auf die entsprechende Controller-Methode zugreift, z.B. `{{#controller.getMainUrl}}`.

6. Bei Bedarf noch eine Formularaktion über eine Schaltfläche definieren.  
*Beispiel:* UI-Widget **Data Button** einfügen und in Eigenschaft **Action** (action) eine Aktion (*Load / Save / Remove*) hinzufügen, siehe [UI Widget: Data Button](#).
7. Falls ein *Speichern*-Schaltfläche erstellt wurde mithilfe des UI-Widgets *Data Button* (Aktion *Save*, siehe Schritt 5) und die URL für das Speichern von der Haupt-URL des Formulars abweicht: Das *Form*-Widget wählen und in Eigenschaft **Save URL** eine URL zu einer ReSTful-Webservice-Ressource hinterlegen, welche die zu speichernden Formulardaten verarbeitet.
8. **Save** klicken, um das Formular zu speichern.
9. Die Detailseite in der Datasource-Konfiguration oder an anderer passender Stelle hinterlegen:
  - Im UI-Editor auf **Copy Result URL** klicken, um die Ressourcen-URL für die Detailseite kopieren.
  - In der Datasource-Konfiguration in Bereich **Detail Page** in Feld **Base URL** (*BaseUri*) die URL zur Detailseite hinterlegen.
  - Datasource-Konfiguration speichern.

Das Formular ist nun vollständig definiert.

- ✓ Wenn Sie das Ergebnis prüfen möchten:
  - **Open Result** in der Menüleiste des *UI-Editors* klicken, um die erzeugte HTML-Seite in einer neuen Browser-Registerkarte zu öffnen.
  - **Open Result in Portal** in der Menüleiste des *Application Config Editors* klicken, um eine Vorschau der gesamten Web-Anwendung zu erhalten.

#### Siehe auch:

- [UI-Widgets einfügen und verwalten](#)
- [Binding des Formulars an den Lister-Eintrag erstellen](#)
- [Daten-Bindings für Formularelemente hinterlegen](#)
- [DomHook-Einträge hinterlegen](#)
- [UI Widget: Data Button](#)
- [UI-Controller definieren](#)

### 2.6.3.3 Formulare auf Basis einer Datasource erzeugen


Wie Sie im UI-Editor des *X4 Web Designers* eine Activities-Datasource als Datenmodell für ein neues Formular verwenden können und automatisch die entsprechenden Formularelemente erzeugen.

#### Voraussetzungen:

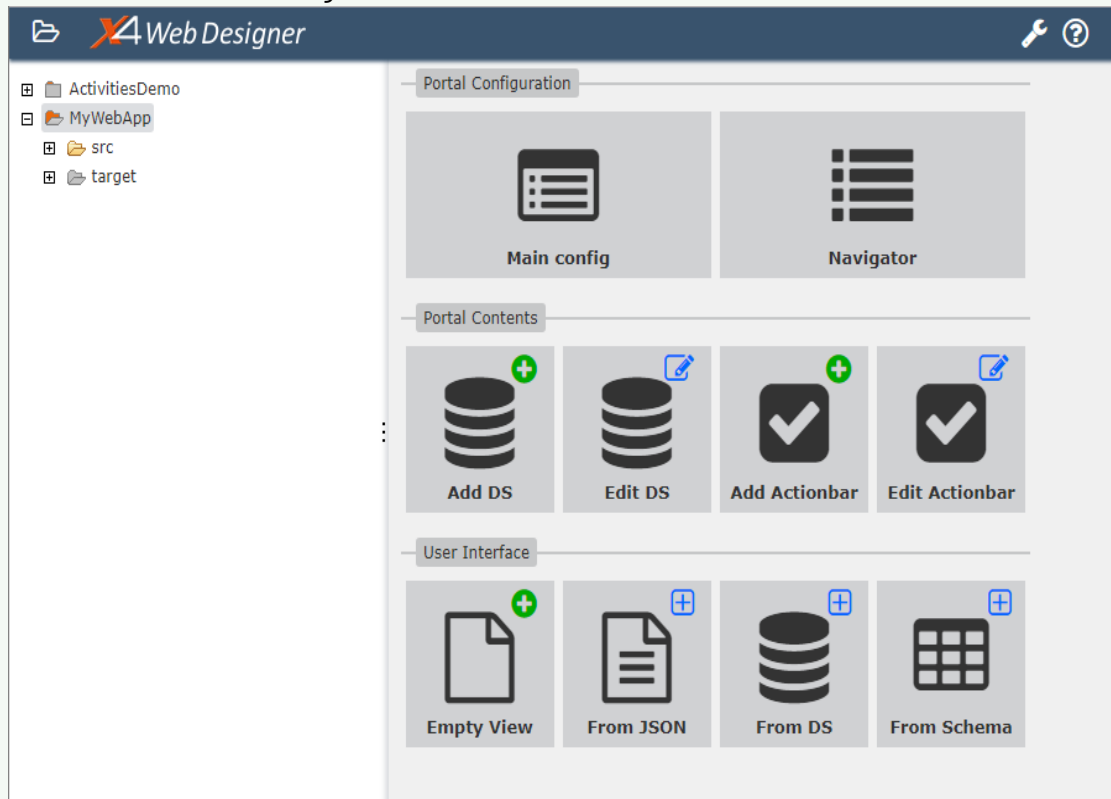
- Die gewünschte Datenstruktur liegt als Activities-Datasource (im JSON-Format) vor.
- Die Activities-Datasource ist vom *X4 Web Designer* über eine URL erreichbar.



**Schritte, um ein Formular generieren zu lassen:**

1. Im UI-Editor an der gewünschten Stelle in der Baumstruktur mit der Maus über *Add/Drop Widgets* fahren und auf  klicken.
2. **Generate Form by Datasource** wählen.

- ✓ Alternativ lässt das Formular auch direkt über die Projekt-Startseite bzw. über die **ui**-Ordner Startseite anlegen. Dazu auf **From DS** klicken.



3. Die URL zur Datasource einfügen und **OK** klicken. Ggf. einen Formularnamen eingeben. Auf Basis der Datasource-Datenstruktur werden nun hierarchische Formularelemente in der Baumstruktur des UI-Editors wie folgt erzeugt:
  - Anhand des Datentyps wird der Formularfeldtyp bzw. UI-Widget-Typ ermittelt (z.B. *Input* für *STRING*, *Checkbox* für *BOOLEAN*)
  - Data-Bindings werden für jedes Formularfeld automatisch erzeugt und in der jeweiligen Widget-Eigenschaft **Value** hinterlegt, z.B. `{{email}}`.
  - Labels werden automatisch aus den Namen der Datasource-Field ID erzeugt.
4. Bei Bedarf die Feldeigenschaften in der **Properties**-Registerkarte anpassen, z.B. die automatisch erzeugten Feldbeschriftungen in Eigenschaft **Label**, siehe [UI-Widgets einfügen und verwalten](#).
5. Binding der Formulardaten an den Lister-Eintrag erstellen. Dazu wird ein **REST-Service** benötigt, welche der **Datasource entsprechende JSON-Daten** liefert. Siehe auch [Binding des Formulars an den Lister-Eintrag erstellen](#).

- ✓ Alternativ empfiehlt es sich, einen Controller zu erstellen, der das Binding des Formulars an den entsprechenden Lister-Eintrag übernimmt, siehe [UI-Controller definieren](#). In der Property für die URL kann dann ein Binding definiert werden, das auf die entsprechende Controller-Methode zugreift, z.B. `{{#controller.getMainUrl}}`.

6. Bei Bedarf noch eine Formularaktion über eine Schaltfläche definieren.  
*Beispiel:* UI-Widget **Data Button** einfügen und in Eigenschaft **Action** (action) eine Aktion (*Load / Save / Remove*) hinzufügen, siehe [UI Widget: Data Button](#).
7. Falls ein *Speichern*-Schaltfläche erstellt wurde mithilfe des UI-Widgets *Data Button* (Aktion *Save*, siehe Schritt 5) und die URL für das Speichern von der Haupt-URL des Formulars abweicht: Das *Form*-Widget wählen und in Eigenschaft **Save URL** eine URL zu einer ReSTful-Webservice-Ressource hinterlegen, welche die zu speichernden Formulardaten verarbeitet.
8. **Save** klicken, um das Formular zu speichern.
9. Die Detailseite in der Datasource-Konfiguration oder an anderer passender Stelle hinterlegen:
  - Im UI-Editor auf **Copy Result URL** klicken, um die Ressourcen-URL für die Detailseite kopieren.
  - In der Datasource-Konfiguration in Bereich **Detail Page** in Feld **Base URL** (*BaseUri*) die URL zur Detailseite hinterlegen.
  - Datasource-Konfiguration speichern.

Das Formular ist nun vollständig definiert.

- ✓ Wenn Sie das Ergebnis prüfen möchten:
  - **Open Result** in der Menüleiste des *UI-Editors* klicken, um die erzeugte HTML-Seite in einer neuen Browser-Registerkarte zu öffnen.
  - **Open Result in Portal** in der Menüleiste des *Application Config Editors* klicken, um eine Vorschau der gesamten Web-Anwendung zu erhalten.

#### Siehe auch:

- [UI-Widgets einfügen und verwalten](#)
- [Binding des Formulars an den Lister-Eintrag erstellen](#)
- [Daten-Bindings für Formularelemente hinterlegen](#)
- [DomHook-Einträge hinterlegen](#)
- [UI Widget: Data Button](#)
- [UI-Controller definieren](#)

#### 2.6.3.4 Formulare auf Basis eines JSON-Schemas erzeugen

Wie Sie im UI-Editor des *X4 Web Designers* ein JSON-Schema als Datenmodell für ein neues Formular verwenden können und automatisch die entsprechenden Formularelemente erzeugen.

##### Voraussetzungen:

- Die gewünschte Datenstruktur liegt als JSON-Schema vor.
- Das JSON-Schema ist vom *X4 Web Designer* über eine URL erreichbar.

### Unterstützte Elemente in JSON-Schematas:


Für die Formulargenerierung wird nicht der komplette Umfang möglicher Deklarationen in JSON-Schematas unterstützt, sondern nur die grundlegenden Strukturen, primitiven Datentypen und Aufzählungswerte (enum). Zudem wird das `description`-Attribut standardmäßig als Tooltip für das entsprechende Eingabefeld verwendet.

Komplexere Definitionen in Schematas sind erlaubt, werden aber bei der Formulargenerierung nicht berücksichtigt. Außerdem werden Zahl-Typen bislang standardmäßig nur als Textfeld ohne Validierung abgebildet.

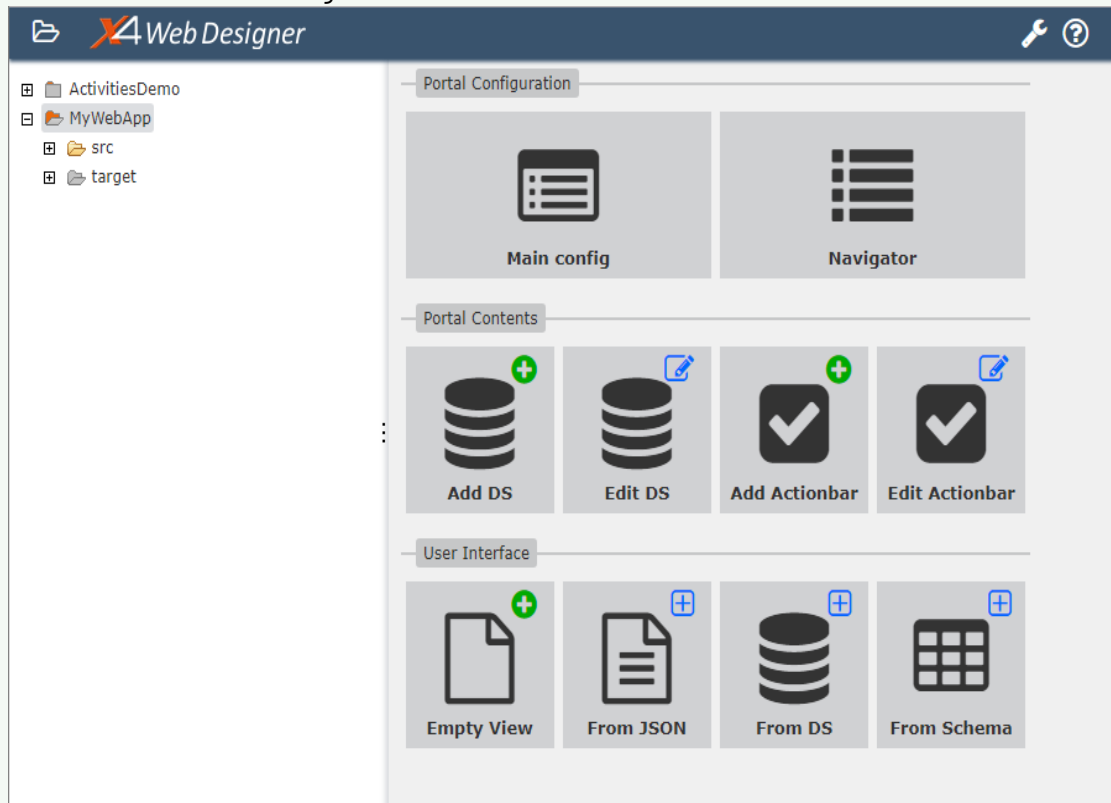
#### Schema-Beispiel

```
{
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "description": "Name is a simple string value"
    },
    "active": {
      "type": "boolean",
      "description": "Active is a simple boolean value"
    },
    "size": {
      "type": "number",
      "description": "Size is a number value"
    },
    "sport": {
      "type": "string",
      "description": "An enumeration type",
      "enum": ["football", "basketball", "tennis"]
    },
    "phoneNumbers": {
      "type": "array",
      "description": "Object in items property defines structure for all array items, could also be an array defining structures for single items",
      "items": {
        "type": "object",
        "properties": {
          "number": { "type": "string" },
          "type": { "type": "string", "enum": ["home", "work", "mobile"] },
          "priority": { "type": "integer" }
        }
      }
    }
  }
}
```

### Schritte, um ein Formular generieren zu lassen:

1. Im UI-Editor an der gewünschten Stelle in der Baumstruktur mit der Maus über *Add/Drop Widgets* fahren und auf  klicken.
2. **Generate Form by JSON Schema** wählen.

- ✓ Alternativ lässt das Formular auch direkt über die Projekt-Startseite bzw. über die **ui**-Ordner Startseite anlegen. Dazu auf **From Schema** klicken.



3. Die URL zum JSON-Schema einfügen und **OK** klicken.  
Auf Basis der JSON-Datenstruktur werden nun hierarchische Formularelemente in der Baumstruktur des UI-Editors wie folgt erzeugt:
  - Anhand des Datentyps wird der Formularfeldtyp bzw. UI-Widget-Typ ermittelt (z.B. *Input* für *string*, *Checkbox* für *boolean* oder *Form Array Panels* bei Arrays)
  - Data-Bindings werden für jedes Formularfeld automatisch erzeugt und in der jeweiligen Widget-Eigenschaft **Value** hinterlegt, z.B. `{{email}}`.
  - Labels werden automatisch aus den Namen der JSON-Eigenschaften erzeugt.
4. Bei Bedarf die Feldeigenschaften in der **Properties**-Registerkarte anpassen, z.B. die automatisch erzeugten Feldbeschriftungen in Eigenschaft **Label**, siehe [UI-Widgets einfügen und verwalten](#).
5. Binding der Formulardaten an den Lister-Eintrag erstellen. Dazu wird ein **REST-Service** benötigt, der dem **Schema entsprechende JSON-Daten** liefert. Siehe auch [Binding des Formulars an den Lister-Eintrag erstellen](#).

- ✓ Alternativ empfiehlt es sich, einen Controller zu erstellen, der das Binding des Formulars an den entsprechenden Lister-Eintrag übernimmt, siehe [UI-Controller definieren](#). In der Property für die URL kann dann ein Binding definiert werden, das auf die entsprechende Controller-Methode zugreift, z.B. `{{#controller.getMainUrl}}`.

6. Bei Bedarf noch eine Formularaktion über eine Schaltfläche definieren.  
*Beispiel:* UI-Widget **Data Button** einfügen und in Eigenschaft **Action** (action) eine Aktion (*Load / Save / Remove*) hinzufügen, siehe [UI Widget: Data Button](#).
7. Falls ein *Speichern*-Schaltfläche erstellt wurde mithilfe des UI-Widgets *Data Button* (Aktion *Save*, siehe Schritt 5) und die URL für das Speichern von der Haupt-URL des Formulars abweicht: Das *Form*-Widget wählen und in Eigenschaft **Save URL** eine URL zu einer ReSTful-Webservice-Ressource hinterlegen, welche die zu speichernden Formularelemente verarbeitet.
8. **Save** klicken, um das Formular zu speichern.
9. Die Detailseite in der Datasource-Konfiguration oder an anderer passender Stelle hinterlegen:
  - Im UI-Editor auf **Copy Result URL** klicken, um die Ressourcen-URL für die Detailseite kopieren.
  - In der Datasource-Konfiguration in Bereich **Detail Page** in Feld **Base URL** (*BaseUri*) die URL zur Detailseite hinterlegen.
  - Datasource-Konfiguration speichern.

Das Formular ist nun vollständig definiert.

- ✓ Wenn Sie das Ergebnis prüfen möchten:
  - **Open Result** in der Menüleiste des *UI-Editors* klicken, um die erzeugte HTML-Seite in einer neuen Browser-Registerkarte zu öffnen.
  - **Open Result in Portal** in der Menüleiste des *Application Config Editors* klicken, um eine Vorschau der gesamten Web-Anwendung zu erhalten.

#### Siehe auch:

- [UI-Widgets einfügen und verwalten](#)
- [Binding des Formulars an den Lister-Eintrag erstellen](#)
- [Daten-Bindings für Formularelemente hinterlegen](#)
- [DomHook-Einträge hinterlegen](#)
- [UI Widget: Data Button](#)
- [UI-Controller definieren](#)

### 2.6.3.5 Binding des Formulars an den Lister-Eintrag erstellen

Wie Sie für ein Formular, das Sie mit dem *UI-Editor* erstellt haben, das Datenmodell mit dem entsprechend selektierten Eintrag im Lister Ihrer *X4 Activities*-Anwendung verknüpfen  
Damit ein Formular innerhalb einer Detailseite die Daten aus dem entsprechenden Datensatz laden und speichern kann, sind folgende Schritte erforderlich:

1. Im *UI-Editor* das entsprechende *Form*-Widget wählen und die **Properties**-Registerkarte aufrufen.

2. In Widget-Eigenschaft **Main URL** (`dataResource:url`) die hinterlegte URL zur Formular-Datenressource erweitern um den Primary-Key des Listeneintrags. Dieser kann durch die JavaScript-Angabe `x4.url.parse().getParam('x4pPrimaryKey')` aus dem dafür vorgesehenen URL-Parameter ausgelesen werden.

*Beispiel für ein komplettes Property-Binding:* `{ '/X4/httpstarter/ReST/MyWebApp/MyDataService/' + x4.url.parse().getParam('x4pPrimaryKey') }`

- ✓ In **Main URL** wurde noch keine URL zur Formular-Datenressource hinterlegt? Ein Standard-Formular erwartet eine zugehörige Daten-Ressource, die über einen ReSTful Webservice erreichbar ist. Wenn das Formular auf Basis JSON-Daten generiert wurde, so wird die URL automatisch hinterlegt, siehe [Formulare auf Basis von JSON-Daten erzeugen](#).

3. **Speichern** klicken.

- ✓ Alternativ empfiehlt es sich, einen Controller zu erstellen, der das Binding des Formulars an den entsprechenden Lister-Eintrag übernimmt, siehe [UI-Controller definieren](#). In der Property für die URL kann dann ein Binding definiert werden, das auf die entsprechende Controller-Methode zugreift, z.B. `{{#controller.getMainUrl}}`.

#### 2.6.3.6 Daten-Bindings für Formularelemente hinterlegen

Wie Sie zu Formularfeldern, die Sie im UI-Editor des *X4 Web Designer* über das entsprechende Widget erstellen, eine Verknüpfung zum Datenobjekt herstellen (Data Binding)

##### 2.6.3.6.1 Einfaches Binding zum JSON-Datenmodell

Um Daten-Bindings zwischen den einzelnen Formularfelder und dem Datenmodell des Formulars herzustellen, den Namen der entsprechenden JSON-Eigenschaft innerhalb von doppelt geschweiften Klammern in der Widget-Eigenschaft **Value** oder einer anderen Eigenschaft die Bindings unterstützt (z.B. auch in allen DOM-Hook Properties) hinterlegen.

*Beispiel:* `{{email}}`

- ✓ Wenn Sie ein Formular auf Basis eines bestehenden JSON-Dokuments erzeugen, so werden die entsprechenden Data-Bindings in den Formularfeld-Widgets automatisch erzeugt, siehe [Formulare auf Basis von JSON-Daten erzeugen](#).

##### 2.6.3.6.2 Erweiterte JavaScript-Bindings

Innerhalb von einfach geschweiften Klammern ist es möglich, JavaScript-Code einzufügen. Innerhalb dieses Codes können dann wieder mit einfach geschweiften Klammern Bindings verwendet werden. Außerdem kann mit `this` auf das Kontext-Objekt der View zugegriffen werden, siehe API-Dokumentation im Abschnitt [UI-Bindings](#).

*Beispiel:* `{ {first_name} + ' ' + {last_name} }`

Mit einer vorangestellten Raute (#) können Sie, neben dem Datenmodell, unter anderem auf folgende Kontext-Eigenschaften zugreifen:

Präfix	Erläuterung	Beispiel
#prop	Widget-Eigenschaft (Property)	{{#prop.label}}
#model	Wert im Datenmodell	{{#model.email}}
#state	Wert im Zustands-Modell des UI	{{#state.adminModeEnabled}}
#controller	Controller-Methode	{{#controller.doSomething}}

 Für eine Übersicht aller verfügbaren Eigenschaften und Elemente sowie weitere Beispiele für UI-Bindings, siehe JavaScript-API des *X4 Web Designers – UI-Bindings*.

#### Auf dieser Seite:

#### Siehe auch:

- [Allgemeine Widget-Eigenschaften](#)
- [Formulare auf Basis von JSON-Daten erzeugen](#)
- [DomHook-Einträge hinterlegen](#)
- [UI-Controller definieren](#)
- [UI-Bindings](#)

### 2.6.3.7 Internationalisierung von Formularen

Für die Internationalisierung von Formularen stehen eine Reihe von Widgets zur Verfügung. Mit diesen lassen sich Lokalisierungs-Inhalte einfach bereitstellen, um diese an beliebigen Stellen im Formular zu verwenden.

Für die Konfiguration und Bereitstellung der Lokalisierungs-Inhalten können folgende Widgets verwendet werden:

- [UI Widget: i18n Initialization](#): Basis-Konfiguration
- [UI Widget: i18n Properties Service](#): Laden der Lokalisierungs-Inhalte von einem ReST-Service
- [UI Widget: i18n Properties File](#): Laden der Lokalisierungs-Inhalte aus einer Datei

Die Verwendung der Lokalisierungs-Inhalte und die sprachspezifische Formatierung von Formularfeldern wird in folgenden Abschnitten näher erläutert:

- [i18n Bindings](#)
- [i18n Standardformate](#)

**Siehe auch:**

- [UI Widget: i18n Initialization](#)
- [UI Widget: i18n Properties Service](#)
- [UI Widget: i18n Properties File](#)
- [i18n Bindings](#)
- [i18n Standardform](#)

### 2.6.3.8 Validierung von Formularen

Für die Validierung von Formularen steht das [UI Widget: i18n Validation Group](#) Widget zur Verfügung. Dieses ermöglicht einfache Validierung und Formatierung von Formularfeld-Inhalten – bei Bedarf auch regionsabhängig.

## 2.6.4 Widgets im UI-Editor verwenden

Wie Sie im UI-Editor des *X4 Web Designers* mit UI-Widgets arbeiten, um Formulare und andere Benutzeroberflächen zu erstellen, die beispielsweise als Detailseiten für Ihre *X4 Activities*-Anwendung dienen

### 2.6.4.1 UI-Widgets einfügen und verwalten

Wie Sie im UI-Editor des *X4 Web Designers* Widgets aus der Palette einfügen, parametrieren und löschen

#### 2.6.4.1.1 Widget hinzufügen

Sie können einzelne Widgets aus der Palette per Drag & Drop auf jeden mit **+ Add/Drop** **Widgets** markierten Bereich hinzufügen. Der Editor ermöglicht dies automatisch in erlaubten Kontexten.



*Tipp:* Um schnell mehrere Widgets desselben Typs hinzuzufügen:


1. In der Widgets-Palette den gewünschten Widget-Typ wählen, z.B. Input.
2. In der Baumstruktur an der gewünschten Stelle wiederholt auf **+ Add/Drop Widgets** klicken.  
Für jeden Klick wird nun ein Widget des gewählten Typs hinzugefügt.

#### 2.6.4.1.2 Widget bearbeiten


Um ein bereits in den Editor-Bereich eingefügtes Widget zu bearbeiten:



1. In der Baumstruktur auf das Element klicken.  
In der Registerkarte **Properties** werden die Widget-spezifischen Eigenschaften (*Custom Properties*) und allgemeinen Eigenschaften (*Default Properties*) angezeigt.
2. In den Properties je nach Bedarf die Feldwerte eingeben. Um direkt die Preview zu aktualisieren können Sie in einem Properties-Feld **Enter** drücken.
3. Abschließend in der Menüleiste **Save** klicken.

- ✓ **Tipp:** Alternativ können Sie Widget-Eigenschaften auch in der JSON-Quelltextansicht bearbeiten:
1. Das Widget in der Baumstruktur markieren und Registerkarte **Source** öffnen.
  2. Änderungen im JSON-Quelltext vornehmen und abschließend auf  klicken, bevor Sie zurück in die grafische Ansicht wechseln.

#### 2.6.4.1.3 Widget löschen

Um ein Widget sofort aus dem Editor-Bereich zu löschen, in der Baumstruktur auf das Element klicken und  (*Remove*) klicken.

##### Auf dieser Seite:

##### Siehe auch:


- [Formulare auf Basis von JSON-Daten erzeugen](#)
- [Formular-Widgets](#)
- [jQuery UI-Widgets](#)
- [Layout-Widgets](#)
- [Allgemeine Widget-Eigenschaften](#)
- [Eigene Widgets erstellen](#)
- [Tastenkombination im X4 Web Designer](#)

#### 2.6.4.2 Allgemeine Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen allgemeine Eigenschaften unter **Default Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Widget Mode</b>	<code>_renderer</code>	Name der zu verwendenden Renderer-Klasse  Wenn weder im HTML noch in den Properties ein Renderer definiert wurde, so wird ein Standard-Renderer verwendet, der lediglich die Elemente des Templates bereitstellt.
	<code>_template</code>	Name des zu verwendenden Templates (sofern dies nicht bereits im HTML angegeben wurde)

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Controller</b>	<code>_controller</code>	Name der zu verwendenden Controller-Klasse Wenn kein Controller definiert ist, wird der nächst übergeordnete Controller verwendet.
<b>Data Context</b>	<code>_dataContext</code>	Model-Referenz relativ zum Data-Model des übergeordneten View-Kontexts  Standard ist <code>#self</code> (kein Kontext-Wechsel).
<b>UI-ID</b>	<code>_uiId</code>	Definiert die UI-ID für die View  Wenn diese Eigenschaft nicht definiert ist oder wenn mehrere Views durch die Konfiguration erzeugt werden (Array-Model), so wird eine ID generiert. Über diese ID können Sie per JavaScript-API oder UI-Binding ( <code>#frame.&lt;id&gt;</code> ) vom Frame-Objekt heraus auf das View-Kontext-Objekt zugreifen.
<b>Model Init-Mode</b>	<code>_modelInitMode</code>	Definiert, wie das Model der View initialisiert wird  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• "auto": automatisch initialisieren (Standard)</li> <li>• "explicit": explizit initialisieren. Die View wird erst gerendert, wenn das Model initialisiert wurde (nicht leer ist), was beispielsweise sinnvoll ist, wenn Daten asynchron geladen werden und eine sinnvolle Darstellung der View ohne Daten nicht möglich ist.</li> </ul>
<b>Model Relation</b>	<code>_modelRelation</code>	Definiert, wie viele Views bei einem Array-Model erzeugt werden sollen  <b>Mögliche Werte:</b> <ul style="list-style-type: none"> <li>• "auto": automatisch mehrere Views erzeugen (Standard); Für jeden Eintrag im Array wird eine View erzeugt und bei jeder Änderung des Arrays wird die View entsprechend angepasst.</li> <li>• "single": unabhängig von den Daten immer nur eine einzige View erzeugen</li> </ul>
<b>State Context</b>	<code>_stateContext</code>	Model-Referenz relativ zum State-Model des übergeordneten View-Kontexts.  <b>Mögliche Werte:</b> Zeichenkette (Standard: <code>#self</code> )

 Ausführliche Informationen zu diesen und weiteren Standard-Eigenschaften finden Sie in der API-Dokumentation zum *X4 Web Designer*, siehe [Standard UI-Properties](#).

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen Verhalten unter **Settings** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Input Change Events</b>	<code>_inputChangeEvents</code>	<p>Definiert, bei welchen Event-Typen Bindings für Input-Elemente aktualisiert werden. Die Event-Typen werden als Array erwartet, wobei jeder Eintrag ein Objekt mit dem Attribut <code>type</code> enthält, das den Namen des nativen Events enthält.</p> <p><b>Mögliche Werte:</b> Array (Standard: "change" und "keyup")</p>
<b>Clear Empty Input Values</b>	<code>_clearEmptyInputValues</code>	<p>Steuert den Umgang mit leeren Formularfeldwerten.</p> <p><b>Mögliche Werte:</b></p> <ul style="list-style-type: none"> <li><code>true</code>: Bindings für Input-Elemente liefern bei einem leeren Wert den Wert <code>undefined</code> (z. B. eine leere Zeichenkette bei Textfeldern oder <code>false</code> bei einer Checkbox). Leere Werte werden nicht in JSON-Daten repräsentiert, was bei großen Formularen mit vielen optionalen Feldern nützlich sein kann.</li> <li><code>false</code> (Standard): Leere Werte werden unverändert vom Formularfeld ins Binding übergeben (z. B. eine leere Zeichenkette bei einem leeren Textfeld).</li> </ul>

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen Verhalten bei bestimmten Ereignissen unter **UI Events** bearbeiten. Für alle Events kann das Verhalten als JavaScript-Code hinterlegt werden.

Bezeichnung	Event	Erläuterung
<b>Prepare</b>	<code>view-prepare</code>	Direkt nach der View-Initialisierung, nachdem Controller und Renderer erzeugt wurden, aber bevor die <code>render</code> -Methode des Renderers aufgerufen und die Standard-Properties für DOM-Hooks ausgewertet werden.

Bezeichnung	Event	Erläuterung
<b>Attach</b>	view-attach	Nachdem die render-Methode des Renderers aufgerufen und die Standard-Properties für DOM-Hooks ausgewertet wurden, aber bevor die Elemente der View in das DOM eingefügt und durch das UI-Framework geparkt werden.
<b>Bind</b>	view-bind	Bevor alle bis dahin bekannten UI-Bindings für das View angewendet werden, die durch das DOM oder die API definiert wurden.
<b>Ready</b>	view-ready	Nachdem die View-Initialisierung abgeschlossen ist.
<b>Unbind</b>	view-unbind	Erstes Event beim Entfernen einer View, bevor Inhalte der View entfernt werden.
<b>Detach</b>	view-detach	Bevor die View und alle untergeordneten Views aus dem DOM entfernt werden.
<b>Cleanup</b>	view-cleanup	Bevor die View aus der UI-Hierarchie gelöscht und alle internen Referenzen entfernt werden. Wird nur beim Entfernen der View, aber nicht bei repaint ausgelöst.
<b>Add Child</b>	view-add-child	<p>Nachdem eine Kind-View in die View eingefügt wurde, bevor das view-prepare-Event der Kind-View ausgelöst wird.</p> <p><b>Data-Attribute:</b></p> <ul style="list-style-type: none"> <li>• uiContext: ID des UI-Kontextes, in dem die Kind-View erzeugt wird.</li> <li>• childContext: View-Kontext-Objekt der Kind-View.</li> </ul>
<b>Remove Child</b>	view-remove-child	<p>Nachdem eine Kind-View entfernt wurde, bevor das view-unbind-Event der Kind-View ausgelöst wird.</p> <p><b>Data-Attribute:</b></p> <ul style="list-style-type: none"> <li>• uiContext: ID des UI-Kontextes, in dem die Kind-View erzeugt wird.</li> <li>• childContext: View-Kontext-Objekt der Kind-View.</li> </ul>

Im **Properties**-Tab des *UI-Editors* können Sie außerdem für ein markiertes Widget folgende Einstellungen tätigen:

Einstellung	Beschreibung
<b>Default Data</b>	Standard-Daten für das Data-Model
<b>Default State</b>	Standard-Daten für das State-Model
<b>Default Local State</b>	Standard-Daten für das lokale State-Model






### 2.6.4.3 Elemente mittels DomHooks anpassen







Mittels sogenannter *DomHooks* lassen sich fachliche Logik (dynamische Werte und Bedingungsprüfungen), spezifische Attribute, Events (Ereignisse) sowie CSS-Klassen und Inline-CSS-Angaben für UI-Widgets hinterlegen.

Im **Properties**-Tab des *UI-Editors* können Sie unter **DomHook: root** DOM-Hooks für jede Widget-Instanz definieren. Damit lässt sich das *Document Object Model* (DOM) entweder für das Wurzel-Element manipulieren (*DomHook: Root*) oder spezifisch für ein bestimmtes DOM-Element des UI-Widgets. So steht beispielsweise für das **Button**-Widget der *DomHook: button* zur Verfügung, welcher direkt auf das `<button>`-Element referenziert.

#### 2.6.4.3.1 DomHook-Eigenschaften



Folgende *DOM-Hook*-Einträge können Sie zu jedem UI-Widget hinterlegen:

Eigenschaft	Beschreibung
<b>Value</b>	<p>Der Wert des DOM-Elements. Bei Eingabefeldern wird dadurch der Wert des Feldes definiert, bei anderen Elementen wird ein Text-Knoten mit diesem Wert als Kind-Knoten des Elements eingefügt.</p> <div style="border: 1px solid red; padding: 5px; margin: 10px 0;">  Kann Kind-Elemente des durch den DomHook referenzierten Elements überschreiben.         </div> <p>Diese Eigenschaft (markiert durch ) kann dynamische Werte innerhalb von doppelt geschweiften Klammern <code>{{...}}</code> enthalten, wie Daten-Bindings bzw. JavaScript-Code.</p> <p><i>Beispiel:</i> <code>{{myInputField}}</code></p>
<b>Show If</b>	<p>Bedingung(en), wann das Widget dem Nutzer der Web-Anwendung angezeigt werden soll</p> <p>Diese Eigenschaft (markiert durch ) kann dynamische Werte innerhalb von doppelt geschweiften Klammern <code>{{...}}</code> enthalten, wie Daten-Bindings bzw. JavaScript-Code.</p>
<b>Render If</b>	<p>Bedingung(en), wann das Widget dem Nutzer der Web-Anwendung gerendert werden soll</p> <p>Diese Eigenschaft (markiert durch ) kann dynamische Werte innerhalb von doppelt geschweiften Klammern <code>{{...}}</code> enthalten, wie Daten-Bindings bzw. JavaScript-Code.</p>
<b>Attributes</b>	<p>Ein oder mehrere Attribute mit jeweils einem (dynamischen) Wert zu dem Element hinzufügen</p> <p>Diese Eigenschaft (markiert durch ) kann dynamische Werte innerhalb von doppelt geschweiften Klammern <code>{{...}}</code> enthalten, wie Daten-Bindings bzw. JavaScript-Code.</p>

Eigenschaft	Beschreibung
Events	<p>Eine oder mehrere Aktionen, die bei nativen JavaScript-Ereignissen (Events) mit jeweils dem Bezeichner und zugehörigem Aktions-Binding, im Element registriert werden</p> <p>Diese Eigenschaft (markiert durch ) kann dynamische Werte innerhalb von doppelt geschweiften Klammern <code>{...}</code> enthalten, wie Daten-Bindings bzw. JavaScript-Code.</p>
CSS Classes	<p>Eine oder mehrere CSS-Klassen mit jeweils einem Klassennamen dem Element zuweisen</p> <div> <p> Wenn der Wert der Eigenschaft eine Zeichenkette (String) ist, wird diese an den definierten Klassennamen angehängt und bei einer Änderung automatisch die Klasse entsprechend umbenannt. Andernfalls wird die Klasse abhängig davon, ob der Wert ein wahrer oder falscher JavaScript-Ausdruck ist, eingefügt oder entfernt.</p> </div> <p>Diese Eigenschaft (markiert durch ) kann dynamische Werte innerhalb von doppelt geschweiften Klammern <code>{...}</code> enthalten, wie Daten-Bindings bzw. JavaScript-Code.</p>
Inline CSS	<p>Eine oder mehrere CSS-Eigenschaften mit jeweils einer Wertzuweisung für das Element definieren (Inline-Style).</p> <p>Diese Eigenschaft (markiert durch ) kann dynamische Werte innerhalb von doppelt geschweiften Klammern <code>{...}</code> enthalten, wie Daten-Bindings bzw. JavaScript-Code.</p> <p><i>Beispiel:</i></p> <div> <p>Inline CSS</p> <p>color  </p> <p>#05113A</p> <p>+</p> </div>



*Tipps zum Bearbeiten von Attributen, Ereignissen, CSS-Klassen und Inline-CSS:*

- + klicken, um einen neuen Eintrag hinzuzufügen
-  Unknown Attachment klicken, um einen Eintrag zu löschen
-  klicken, um einen Eintrag zu bearbeiten

### 2.6.4.3.2 Beispiel für DomHook: Input-Feld bedingt ausblenden

Wenn beispielsweise innerhalb eines Passwort-ändern-Formulars ein Input-Feld `repeatpassword` für die Nutzer so lange ausgeblendet werden soll, bis ein Input-Feld `newpassword` ausgefüllt wurde, so gehen Sie wie folgt vor:

1. Im UI-Editor zwei *Input*-Formularelemente anlegen, siehe [UI-Widgets einfügen und verwalten](#):
  - In der Widget-Eigenschaft **Label** den Wert *New Password* und in **Value** als Daten-Binding `{{newpassword}}` hinterlegen, siehe [Daten-Bindings für Formularelemente hinterlegen](#).
  - In der Widget-Eigenschaft **Label** den Wert *Repeat Password* hinterlegen.
2. Input-Formularelement *Repeat Password* wählen und in Registerkarte **Properties** die Ansicht **DOMHook:Root** aufrufen.
3. In der Widget-Eigenschaft **Show If** den Wert `{{newpassword}}` hinterlegen.  
D.h. nur wenn `newpassword` gesetzt ist, soll dieses Formularelement angezeigt werden.
4. **Update Preview** klicken, um das Verhalten zu testen.
5. **Save** klicken, um die Änderung zu speichern.

#### Beispiel: JSON zum Input-Feld 'New Password'

```
{
  "_widget": "x4.forms.Input",
  "_template": "Input",
  "_renderer": "x4.ui.forms.render.InputWrapper",
  "label": "New Password",
  "value": "{{newpassword}}",
  "_domHooks": {
    "input": {},
    "_root": {}
  }
}
```

#### Beispiel: JSON zum Input-Feld 'Repeat Password'

```
{
  "_widget": "x4.forms.Input",
  "_template": "Input",
  "_renderer": "x4.ui.forms.render.InputWrapper",
  "label": "Repeat Password",
  "_domHooks": {
    "input": {},
    "_root": {
      "showIf": "{{newpassword}}"
    }
  }
}
```

#### 2.6.4.4 Dashboard-Widgets

Welche Dashboard-Widgets für den *UI-Editor* des *X4 Web Designers* verfügbar sind.

##### **Lizenzhinweis**

Bitte beachten Sie, dass für die Verwendung der Dashboard-Widgets eine spezielle Lizenz erforderlich ist.

##### **Beachten Sie**

- Dashboard-Widgets lassen sich mit folgender URL im UI-Editor des Web Designers einbinden: `/X4/httpstarter/ReST/X4WebResources/dashboards-widgets/widgets/main.widgets.json`.
- Wenn nur eine bestimmte Widget-Collection verwendet werden soll, kann die URL aus dem Kapitel der entsprechenden Collection entnommen werden.


#### 2.6.4.4.1 AdminLTE-Widgets

Widgets auf Basis der *AdminLTE-Bibliothek*, die standardmäßig im *UI-Editor* des *X4 Web Designers* in der Widget-Palette unter **AdminLTE widgets** verfügbar sind.

##### 2.6.4.4.1.1 UI-Widget: Box

Mit dem *Box-Widget* lässt sich im *UI-Editor* des *X4 Web Designers* eine farbige Box für beliebigen Inhalt mit Titel und optionalen Aktions-Schaltflächen auf Basis der Bibliothek *AdminLTE* erzeugen. Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Title</b>	<code>title</code>	Titel im Header der Box
<b>Box Type</b>	<code>boxType</code>	<p>Typ (Rahmen-/Header-Farbe) der Box.</p> <p><i>Mögliche Werte:</i> <code>default</code>, <code>primary</code>, <code>info</code>, <code>warning</code>, <code>success</code>, <code>danger</code></p> <p> Je nach gewähltem Typ ändert sich die Rahmen- und Header-Farbe der Box.</p>
<b>Bg Color</b>	<code>bgColor</code>	Durch AdminLTE bereitgestellte Hintergrundfarbe für den Dialog, z. B. <code>blue</code> , <code>red</code> , <code>green</code> , <code>yellow</code> , <code>aqua</code>
<b>Solid</b>	<code>solid</code>	Angabe, ob die Umrandung der Box und der Hintergrund des Headers farbig dargestellt werden.



Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Collapsible</b>	collapsible	Angabe, ob der Inhalt der Box über eine Schaltfläche im Header ausgeblendet werden kann.
<b>Collapsed</b>	collapsed	Angabe, ob der Inhalt der Box initial ausgeblendet ist.
<b>Removable</b>	removable	Angabe, ob die Box über eine Schaltfläche im Header aus dem UI entfernt (geschlossen) werden kann.

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

Datenstruktur des Box-Widgets

Zusätzlich zu den oben genannten Widget-spezifischen und allgemeinen Eigenschaften besitzt das *Box*-Widget eine vordefinierte Datenstruktur mit folgenden Elementen (UI-Kontexte):

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Header Buttons</b>	headerButtons	Innerhalb dieses Arrays können beliebig viele Schaltflächen eingefügt werden, die rechts im Header der Box angezeigt werden.
<b>Main Contents</b>	contents	Innerhalb dieses Arrays können beliebig viele UI-Widgets eingefügt werden, die innerhalb der Box angezeigt werden, z.B. Charts.
<b>Footer Contents</b>	footerContents	Innerhalb dieses Arrays können beliebig viele UI-Widgets eingefügt werden, die in einer Fußzeile der Box angezeigt werden.

Siehe auch:

- [AdminLTE-Widgets](#)

#### 2.6.4.4.1.2 UI-Widget: Info Box

Mit dem *Info Box*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* eine Box mit einem Icon, einem Text und einer Kennzahl, sowie einem optionalen Fortschrittsbalken, auf Basis der Bibliothek *AdminLTE* erzeugen.

Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Text</b>	text	Text, der in der Box angezeigt werden soll

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Number</b>	number	Kennzahl für die Box
<b>IconType</b>	iconType	Die Icon-Collection, die für das Icon der Box verwendet werden soll  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• glyphicon: Icon-Collection <i>GLYPHICONS</i> – <a href="http://glyphicons.com/">http://glyphicons.com/</a></li> <li>• fa: Icon-Collection <i>Font Awesome</i> – <a href="http://fontawesome.io/">http://fontawesome.io/</a></li> </ul>
<b>Icon Name</b>	iconName	Name des Icons, ohne Typ-Präfix
<b>Bg Color</b>	bgColor	Durch AdminLTE bereitgestellte Hintergrundfarbe für die Box, z. B. blue, red, green, yellow, aqua
<b>Icon Bg Color</b>	iconBgColor	Durch AdminLTE bereitgestellte Hintergrundfarbe für den Icon-Bereich, falls abweichend von der Farbe der Box, z. B. blue, red, green, yellow, aqua
<b>Progress Value</b>	progressValue	Optionaler Zahlen-Wert für die Fortschrittanzeige (0 bis 100)
<b>Progress Description</b>	progressDescription	Optionaler Beschreibungstext für die Fortschrittanzeige

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Siehe auch:

- [AdminLTE-Widgets](#)

#### 2.6.4.4.1.3 UI-Widget: Small Box

Mit dem *Small Box*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* eine farbige Box mit beliebigem Haupt-Inhalt, einem Icon und einem Link/Aktions-Bereich auf Basis der Bibliothek *AdminLTE* erzeugen.

Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>IconType</b>	iconType	Die Icon-Collection, die für das Box-Icon verwendet werden soll  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• glyphicon: Icon-Collection <i>GLYPHICONS</i> – <a href="http://glyphicons.com/">http://glyphicons.com/</a></li> <li>• fa: Icon-Collection Font Awesome – <a href="http://fontawesome.io/">http://fontawesome.io/</a></li> </ul>
<b>Icon Name</b>	iconName	Name des Icons, ohne Typ-Präfix.
<b>Link</b>	link	URL, die beim Klick auf den Link-Bereich im Browser-Fenster geöffnet werden soll
<b>Action</b>	action	Aktion als UI-Binding, die beim Klick auf den Link-Bereich ausgeführt werden soll, z.B. <code>{{#controller.boxClicked}}</code>
<b>Bg Color</b>	bgColor	Durch AdminLTE bereitgestellte Hintergrundfarbe für die Box, z. B. blue, red, green, yellow, aqua

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Datenstruktur des Small Box-Widgets

Zusätzlich zu den oben genannten Widget-spezifischen und allgemeinen Eigenschaften besitzt das *Small Box*-Widget eine vordefinierte Datenstruktur mit folgenden Elementen (UI-Kontexte):

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Contents</b>	contents	Innerhalb dieses Arrays können beliebig viele UI-Widgets eingefügt werden, die innerhalb der Box angezeigt werden, beispielsweise Überschrift und Text.
<b>Link Contents</b>	linkContents	Innerhalb dieses Arrays können beliebig viele UI-Widgets eingefügt werden, die in der Linkzeile der Box angezeigt werden, beispielsweise Text und ein Icon.

#### Siehe auch:

- [AdminLTE-Widgets](#)

#### 2.6.4.4.1.4 UI-Widget: Simple Table

Mit dem *Simple Table*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Tabellen auf Basis der Bibliothek *AdminLTE* erzeugen.

Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Bordered</b>	bordered	Angabe, ob die Tabellen-Zellen umrandet werden sollen
<b>Condensed</b>	condensed	Angabe, ob die Innenabstände der Zellen minimiert werden sollen
<b>Striped</b>	striped	Angabe, ob gerade und ungerade Zeilen mit verschiedenen Hintergrundfarben hinterlegt werden sollen
<b>Hover</b>	hover	Angabe, ob Zeilen beim Überfahren mit der Maus hervorgehoben werden sollen

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

Datenstruktur des Simple Table-Widgets

Zusätzlich zu den oben genannten Widget-spezifischen und allgemeinen Eigenschaften besitzt das *Simple Table*-Widget eine vordefinierte Datenstruktur mit folgenden Elementen (UI-Kontexte):

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Columns</b>	columns	Innerhalb dieses Arrays können beliebig viele Spalten mit Titel und optionaler Breite für die Tabelle definiert werden.
<b>Rows</b>	rows	Innerhalb dieses Arrays können beliebig viele Zeilen und in den Zeilen wiederum Zellen ( <i>cells</i> ) mit beliebigem Inhalt definiert werden.  Die Anzahl der Zellen einer Zeile sollte mit der Anzahl der Spalten der Tabelle übereinstimmen.

**Siehe auch:**


- [AdminLTE-Widgets](#)

#### 2.6.4.4.1.5 UI-Widget: Data Table

Mit dem *Data Table*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* eine Tabelle zur Anzeige von Daten mit Seitenverwaltung, Sortierung und Suchfunktionalität auf Basis der Bibliotheken *AdminLTE* und *DataTables*(<https://datatables.net/>) erzeugen.

### Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Data</b>	<code>data</code>	UI-Binding für Daten oder Konfigurationsoptionen, abhängig vom Widget-Modus ( <code>Basic</code> > Daten, <code>Dynamic</code> > Optionen).   Wird nicht benötigt, wenn die Daten über die Ajax-Option (siehe unten) geladen werden.
<b>Ajax - URL</b>	<code>baseOptions.ajax.url</code>	Optionale URL zum Laden von JSON-Daten.
<b>Ajax - Type</b>	<code>baseOptions.ajax.type</code>	HTTP-Methode zum Laden der Daten; Standard ist GET.
<b>Ajax - Data Src</b>	<code>baseOptions.ajax.url</code>	Referenz auf das Daten-Array innerhalb des geladenen JSON-Objekts; Standard ist <code>data</code> .
<b>Server Side</b>	<code>baseOptions.serverSide</code>	Angabe, ob Seitenverwaltung, Sortierung und Suche Server-seitig statt Client-seitig durchgeführt werden. Die entsprechenden Werte werden dann als URL-Parameter an die Ajax-URL angehängt.
<b>Ordering</b>	<code>baseOptions.ordering</code>	Angabe, ob Sortierung für die Tabelle aktiviert wird.
<b>Searching</b>	<code>baseOptions.searching</code>	Angabe, ob Suche für die Tabelle aktiviert wird.
<b>Paging</b>	<code>baseOptions.paging</code>	Angabe, ob Seitenverwaltung für die Tabelle aktiviert wird.
<b>Info</b>	<code>baseOptions.info</code>	Angabe, ob Informationen über den Anzeigebereich der Daten unter der Tabelle angezeigt werden.
<b>Length Change</b>	<code>baseOptions.lengthChange</code>	Angabe, ob die Seitenlänge vom Anwender der Tabelle geändert werden kann.
<b>Page Length</b>	<code>baseOptions.pageLength</code>	Initiale Seitenlänge der Tabelle; Standard ist 10.
<b>Scroll X</b>	<code>baseOptions.scrollX</code>	Angabe, ob horizontales Scrollen bei Bedarf in der Tabelle angeboten wird.
<b>Max. content height</b>	<code>baseOptions.scrollHeight</code>	Definiert eine maximale Höhe für den Inhalt der Tabelle als CSS-Längenangabe. Bei Bedarf wird vertikales Scrollen aktiviert.
<b>Bordered</b>	<code>bordered</code>	Angabe, ob die Zellen der Tabelle umrandet werden sollen.
<b>Condensed</b>	<code>condensed</code>	Angabe, ob die Innenabstände der Zellen minimiert werden sollen.
<b>Striped</b>	<code>striped</code>	Angabe, ob gerade und ungerade Zeilen mit verschiedenen Hintergrundfarben hinterlegt werden sollen.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Hover</b>	hover	Angabe, ob Zeilen beim überfahren mit der Maus hervorgehoben werden sollen.

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

#### Allgemeine Widget-Eigenschaften

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Widget Mode</b>	_template, _renderer	Name des zu verwendenden Templates  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• Basic: In der Data-Property werden dynamische Daten erwartet (Standard)</li> <li>• Dynamic: In der Data-Property wird ein dynamisches Konfigurationsoptionen-Objekt erwartet. Siehe <a href="https://datatables.net/reference/option/">https://datatables.net/reference/option/</a></li> </ul>

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Datenstruktur des Data Table-Widgets

Zusätzlich zu den oben genannten Widget-spezifischen und allgemeinen Eigenschaften besitzt das *Data Table*-Widget eine vordefinierte Datenstruktur mit folgenden Elementen (UI-Kontexte):

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Columns</b>	columns	Innerhalb dieses Arrays können beliebig viele Spalten mit Titel und optionaler Breite für die Tabelle definiert werden. Außerdem kann in der Spalte ein Identifikationsname für die DataTable-API und die Datenverknüpfung für Zeilen-Objekte konfiguriert werden.  Alternativ können die Spalten auch in den Konfigurationsoptionen für die Tabelle definiert werden.

#### Siehe auch:

- [AdminLTE-Widgets](#)

#### 2.6.4.4.1.6 UI-Widget: Label

Mit dem *Label*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* ein farbiges Text-Label auf Basis der Bibliothek *AdminLTE* erzeugen.

#### Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
Value	value	Anzeigetext des Labels
Type	type	Typ (Farbe) des Labels. <i>Mögliche Werte:</i> default, primary, info, warning, success, danger

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Siehe auch:

- [AdminLTE-Widgets](#)

#### 2.6.4.4.1.7 UI-Widget: Badge

Mit dem *Badge*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* ein farbiges, abgerundetes Label auf Basis der Bibliothek *AdminLTE* erzeugen.

Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
Value	value	Anzeigewert
Bg Color	bgColor	Durch AdminLTE bereitgestellte Hintergrundfarbe für das Label, z. B. blue, red, green, yellow, aqua

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Siehe auch:

- [AdminLTE-Widgets](#)

#### 2.6.4.4.1.8 UI-Widget: Progress

Mit dem *Progress*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* ein farbiger Fortschrittsbalken auf Basis der Bibliothek *AdminLTE* erzeugen.

### Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Value</b>	value	Wert des Fortschrittsbalkens von 0 bis 100.
<b>Bg Color</b>	bgColor	Durch AdminLTE bereitgestellte Hintergrundfarbe für den Fortschrittsbalken, z. B. blue, red, green, yellow, aqua
<b>Size</b>	size	Höhe/Dicke des Fortschrittsbalkens  <i>Mögliche Werte:</i> m, sm, xs und xxs

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Siehe auch:

- [AdminLTE-Widgets](#)

### 2.6.4.4.1.9 UI-Widget: Icon

Mit dem *Icon*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* ein Icon auf Basis der Icon-Collections *GLYPHICONS* (<http://glyphicons.com/>) oder *Font Awesome* (<http://fontawesome.io/>) erzeugen.

✓ Die Icons passen sich automatisch der im verwendeten Kontext definierten Text-Größe und -Farbe an. Die Icon-Größe und Farbe lassen sich also auch explizit mit Hilfe der CSS-Eigenschaften `font-size` und `color` beliebig definieren.

### Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.



Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>IconType</b>	iconType	Die Icon-Collection, die für das Icon verwendet werden soll  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• glyphicon: Icon-Collection <i>GLYPHICONS</i> - <a href="http://glyphicons.com/">http://glyphicons.com/</a></li> <li>• fa: Icon-Collection Font Awesome - <a href="http://fontawesome.io/">http://fontawesome.io/</a></li> </ul>
<b>Icon Name</b>	iconName	Name des Icons, ohne Typ-Präfix.

✔ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✔ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Siehe auch:

- [AdminLTE-Widgets](#)

#### 2.6.4.4.2 Chart-Widgets

Chart-Widgets auf Basis der *Chart.js-Bibliothek*, die standardmäßig im *UI-Editor* des *X4 Web Designers* in der Widget-Palette unter **Chart-Widgets** verfügbar sind.

❗ Chart-Widgets lassen sich mit folgender URL im UI-Editor des Web Designers einbinden: / X4/httpstarter/ReST/X4WebResources/dashboards-widgets/widgets/chartjs/charts.widgets.json

##### 2.6.4.4.2.1 UI-Widget: Bar Chart

Mit dem *Bar Chart*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* ein Balkendiagramm auf Basis der Bibliothek *Chart.js* erzeugen (<http://www.chartjs.org/>).

Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Data</b>	data	<p>UI-Binding, das die Daten für das Chart liefert, z. B. <code>{{chartData}}</code> oder <code>{{#controller.getChartData}}</code>.</p> <p><b>i</b> Für Informationen zum erwarteten Datenformat, siehe Dokumentation der verwendeten Bibliothek unter <a href="https://github.com/chartjs/Chart.js/blob/v1.1.1/docs/02-Bar-Chart.md">https://github.com/chartjs/Chart.js/blob/v1.1.1/docs/02-Bar-Chart.md</a></p>
<b>Height</b>	height	Höhe des Charts als CSS-Längenangabe; Kann abweichen, wenn die Responsive-Option aktiviert ist.
<b>Width</b>	width	Breite des Charts als CSS-Längenangabe; Kann abweichen, wenn die Responsive-Option aktiviert ist.
<b>Responsive</b>	options.responsive	Angabe, ob das Chart responsive sein soll, d.h. sich automatische an den verfügbaren Platz im Browser anpasst
<b>Scale Show Grid Lines</b>	options.scaleShowGridLines	Angabe, ob Gitterlinien innerhalb des Charts angezeigt werden sollen

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### Widget-API

Für dieses UI-Widget ist zudem eine JavaScript-API verfügbar. Diese stellt folgende Methoden zur Verfügung.

Methode	Erläuterung
<b>getChart</b>	<p>Diese Methode liefert die native API des Charts.</p> <pre>&lt;vc&gt;.getAPI().getChart(): &lt;chartAPI&gt;</pre>
<b>updateChart</b>	<p>Diese Methode erzeugt/aktualisiert das Chart mit Daten und Konfigurationsoptionen.</p> <pre>&lt;vc&gt;.getAPI().updateChart(data: object, options: object)</pre> <p><b>i</b> Für Details zu Datenformat und verfügbaren Konfigurationsoptionen, siehe <a href="https://github.com/chartjs/Chart.js/blob/v1.1.1/docs/02-Bar-Chart.md">https://github.com/chartjs/Chart.js/blob/v1.1.1/docs/02-Bar-Chart.md</a></p>

Methode	Erläuterung
<b>getOptions</b>	Diese Methode liefert die Konfigurationsoptionen, die für das Widget verwendet werden (Kombination aus Optionen in UI-Properties und Standard-Einstellungen).  <vc>.getAPI().getOptions(): object

Daten-Beispiel

#### Beispiel

```
{
  "labels": [ "January", "February", "March", "April", "May", "June", "July" ],
  "datasets": [
    {
      "label": "Electronics",
      "fillColor": "#d2d6de",
      "strokeColor": "#d2d6de",
      "data": [ 65, 59, 80, 81, 56, 55, 40 ]
    },
    {
      "label": "Digital Goods",
      "fillColor": "#3b8bba",
      "strokeColor": "#3b8bba",
      "data": [ 28, 48, 40, 19, 86, 27, 90 ]
    }
  ]
}
```

#### Siehe auch:

- [Chart-Widgets](#)

#### 2.6.4.4.2.2 UI-Widget: Line Chart

Mit dem *Line Chart*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* ein Liniendiagramm auf Basis der Bibliothek *Chart.js* erzeugen (<http://www.chartjs.org/>).

Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Data</b>	<code>data</code>	<p>UI-Binding, das die Daten für das Chart liefert, z. B. <code>{{chartData}}</code> oder <code>{{#controller.getChartData}}</code>.</p> <p><b>i</b> Für Informationen zum erwarteten Datenformat, siehe Dokumentation der verwendeten Bibliothek unter <a href="https://github.com/chartjs/Chart.js/blob/v1.1.1/docs/01-Line-Chart.md">https://github.com/chartjs/Chart.js/blob/v1.1.1/docs/01-Line-Chart.md</a></p>
<b>Height</b>	<code>height</code>	Höhe des Charts als CSS-Längenangabe; Kann abweichen, wenn die Responsive-Option aktiviert ist.
<b>Width</b>	<code>width</code>	Breite des Charts als CSS-Längenangabe; Kann abweichen, wenn die Responsive-Option aktiviert ist.
<b>Responsive</b>	<code>options.responsive</code>	Angabe, ob das Chart responsive sein soll, d.h. sich automatisch an den verfügbaren Platz im Browser anpasst
<b>Bezier Curve</b>	<code>options.bezierCurve</code>	<p>Angabe, ob die Linien als Kurve dargestellt werden sollen</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>true</code>: Darstellung als Kurven</li> <li>• <code>false</code>: Gerade Linien zwischen zwei Punkten (Standard)</li> </ul>
<b>Dataset Fill</b>	<code>options.datasetFill</code>	Angabe, ob der Bereich unterhalb der Kurven eingefärbt werden soll
<b>Scale Show Grid Lines</b>	<code>options.scaleShowGridLines</code>	Angabe, ob Gitterlinien innerhalb des Charts angezeigt werden sollen

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### Widget-API

Für dieses UI-Widget ist zudem eine JavaScript-API verfügbar. Diese stellt folgende Methoden zur Verfügung.

Methode	Erläuterung
<b>getChart</b>	<p>Diese Methode liefert die native API des Charts.</p> <pre>&lt;vc&gt;.getAPI().getChart(): &lt;chartAPI&gt;</pre>

Methode	Erläuterung
<b>updateChart</b>	<p>Diese Methode erzeugt/aktualisiert das Chart mit Daten und Konfigurationsoptionen.</p> <pre>&lt;vc&gt;.getAPI().updateChart(data: object, options: object)</pre> <p><b>i</b> Für Details zu Datenformat und verfügbaren Konfigurationsoptionen, siehe <a href="https://github.com/chartjs/Chart.js/blob/v1.1.1/docs/01-Line-Chart.md">https://github.com/chartjs/Chart.js/blob/v1.1.1/docs/01-Line-Chart.md</a></p>
<b>getOptions</b>	<p>Diese Methode liefert die Konfigurationsoptionen, die für das Widget verwendet werden (Kombination aus Optionen in UI-Properties und Standard-Einstellungen).</p> <pre>&lt;vc&gt;.getAPI().getOptions(): object</pre>

## Daten-Beispiel

**Beispiel**

```
{
  "labels": [ "January", "February", "March", "April", "May", "June", "July" ],
  "datasets": [
    {
      "label": "Electronics",
      "fillColor": "#d2d6de",
      "strokeColor": "#d2d6de",
      "pointColor": "#d2d6de",
      "data": [ 65, 59, 80, 81, 56, 55, 40 ]
    },
    {
      "label": "Digital Goods",
      "fillColor": "rgba(60,141,188,0.9)",
      "strokeColor": "rgba(60,141,188,0.8)",
      "pointColor": "#3b8bba",
      "data": [ 28, 48, 40, 19, 86, 27, 90 ]
    }
  ]
}
```

**Siehe auch:**


- [Chart-Widgets](#)

## 2.6.4.4.2.3 UI-Widget: Pie Chart

Mit dem *Pie Chart*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* ein Tortendiagramm auf Basis der Bibliothek *Chart.js* erzeugen (<http://www.chartjs.org/>).

Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Data</b>	<code>data</code>	<p>UI-Binding, das die Daten für das Chart liefert, z. B. <code>{{chartData}}</code> oder <code>{{#controller.getChartData}}</code>.</p> <p> Für Informationen zum erwarteten Datenformat, siehe Dokumentation der verwendeten Bibliothek unter <a href="https://github.com/chartjs/Chart.js/blob/v1.1.1/docs/05-Pie-Doughnut-Chart.md">https://github.com/chartjs/Chart.js/blob/v1.1.1/docs/05-Pie-Doughnut-Chart.md</a></p>
<b>Height</b>	<code>height</code>	Höhe des Charts als CSS-Längenangabe; Kann abweichen, wenn die Responsive-Option aktiviert ist.
<b>Width</b>	<code>width</code>	Breite des Charts als CSS-Längenangabe; Kann abweichen, wenn die Responsive-Option aktiviert ist.
<b>Responsive</b>	<code>options.responsive</code>	Angabe, ob das Chart responsive sein soll, d.h. sich automatisch an den verfügbaren Platz im Browser anpasst
<b>Animate Rotate</b>	<code>options.animateRotate</code>	Angabe, ob die Initialisierung des Charts animiert werden soll
<b>Segment Show Stroke</b>	<code>options.segmentShowStroke</code>	Angabe, ob Trennlinien zwischen den einzelnen Bereichen und der Umgebung angezeigt werden sollen


✔ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✔ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Widget-API

Für dieses UI-Widget ist zudem eine JavaScript-API verfügbar. Diese stellt folgende Methoden zur Verfügung.

Methode	Erläuterung
<b>getChart</b>	<p>Diese Methode liefert die native API des Charts.</p> <pre>&lt;vc&gt;.getAPI().getChart(): &lt;chartAPI&gt;</pre>

Methode	Erläuterung
<b>updateChart</b>	<p>Diese Methode erzeugt/aktualisiert das Chart mit Daten und Konfigurationsoptionen.</p> <pre>&lt;vc&gt;.getAPI().updateChart(data: object, options: object)</pre> <p> Für Details zu Datenformat und verfügbaren Konfigurationsoptionen, siehe <a href="https://github.com/chartjs/Chart.js/blob/v1.1.1/docs/05-Pie-Doughnut-Chart.md">https://github.com/chartjs/Chart.js/blob/v1.1.1/docs/05-Pie-Doughnut-Chart.md</a></p>
<b>getOptions</b>	<p>Diese Methode liefert die Konfigurationsoptionen, die für das Widget verwendet werden (Kombination aus Optionen in UI-Properties und Standard-Einstellungen).</p> <pre>&lt;vc&gt;.getAPI().getOptions(): object</pre>

#### Daten-Beispiel

##### Beispiel

```
[
  {
    value: 300,
    color: "#F7464A",
    highlight: "#FF5A5E",
    label: "Red"
  },
  {
    value: 50,
    color: "#46BFBD",
    highlight: "#5AD3D1",
    label: "Green"
  },
  {
    value: 100,
    color: "#FDB45C",
    highlight: "#FFC870",
    label: "Yellow"
  }
]
```

#### Siehe auch:

- [Chart-Widgets](#)

#### 2.6.4.5 File-Widgets

Widgets zur Datei-Verarbeitung, die standardmäßig im *UI-Editor* des *X4 Web Designers* in der Widget-Palette unter **Files** verfügbar sind.

### 2.6.4.5.1 UI-Widget: File Upload Panel

Mit dem *File Upload Panel*-Widget lässt sich im UI-Editor des X4 Web Designers ein Datei-Upload definieren. Dabei lässt sich konfigurieren, ob eine oder mehrere Dateien zum Upload ausgewählt werden können.

- ✓ In Verbindung mit dem *File Button* Widget und dem *File List* Widget können so beispielsweise Dateien selektiert, in einer Liste angezeigt und anschließend hochgeladen werden.

#### 2.6.4.5.1.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Upload URL</b>	uploadURL	URL an welche die Dateien per POST gesendet werden
<b>Multiple</b>	multiple	Definiert, ob eine oder mehrere Dateien für den Upload ausgewählt werden können  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• true: die Auswahl mehrerer Dateien erlauben</li> <li>• false: die Auswahl auf eine Datei beschränken</li> </ul>
<b>Add Model Data</b>	addModelData	Definiert, ob Daten aus dem Datenmodell zur Upload-Anfrage hinzugefügt werden sollen  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• true: Daten aus dem Datenmodell zur Upload-Anfrage hinzufügen</li> <li>• false: Upload-Anfrage ohne Daten aus dem Datenmodell erzeugen</li> </ul>
<b>Model Data Context</b>	modelDataContext	Falls Eigenschaft <i>Add Model Data</i> aktiviert ist: Referenz zum Datenmodell für den Upload-Request (Standard: <code>#self</code> )

- ✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

- ✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### 2.6.4.5.1.2 Events

Durch folgende Events werden im *File Upload Panel* vordefinierte Aktionen ausgeführt



Event	Aktion
file-choose	Öffnet die Datei-Auswahl des Browsers
file-delete	Löscht eine bereits ausgewählte Datei
file-upload	Lädt die ausgewählten Dateien hoch

Folgende Events können im *File Upload Panel* ausgelöst werden, wenn der Datei-Upload erfolgreich bzw. nicht erfolgreich war.

Event	Beschreibung
file-upload-ready	Wird ausgelöst, wenn der Datei-Upload erfolgreich abgeschlossen ist.
file-upload-failed	Wird ausgelöst, wenn der Datei-Upload fehlgeschlagen ist.

#### Siehe auch:

- [UI Widget: File Button](#)
- [UI Widget: File List](#)

### 2.6.4.5.2 UI Widget: File Button

Mit dem *File Button*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* einzelne Schaltflächen zum Auswählen und Hochladen von Dateien innerhalb eines *File Upload Panels* erstellen.



#### Voraussetzung

Das *File Button* Widget kann nur innerhalb eines *File Upload Panels* verwendet werden, siehe [UI-Widget: File Upload Panel](#).



In Verbindung mit dem *File Upload Panel* Widget und dem [File List](#) Widget können so beispielsweise Dateien selektiert, in einer Liste angezeigt und anschließend hochgeladen werden.

### 2.6.4.5.2.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
Label	label	Angezeigter Beschriftungstext (Label) für das Formularfeld

Bezeichnung	JSON-Eigenschaft	Erläuterung
Action	action	<p>Aktion, die bei einem Klick auf den Button ausgeführt werden soll.</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• Choose: Das Event <code>file-choose</code> auslösen, um die Datei-Auswahl des Browsers zu öffnen</li> <li>• Upload: Das Event <code>file-upload</code> auslösen, um den Datei-Upload auszuführen</li> </ul> <div> <i>i</i> Siehe auch Abschnitt Events unter <a href="#">UI-Widget: File Upload Panel</a> </div>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Auf dieser Seite

##### Siehe auch:

- [UI-Widget: File Upload Panel](#)
- [UI Widget: File List](#)

#### 2.6.4.5.3 UI Widget: File List

Mit dem *File List*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* eine Liste von ausgewählten Dateien innerhalb eines *File Upload Panels* anzeigen.

*i* **Voraussetzung**  
Das *File List*-Widget kann nur innerhalb eines *File Upload Panels* verwendet werden, siehe [UI-Widget: File Upload Panel](#).

- ✓
- In Verbindung mit dem *File Upload Panel*-Widget und dem *File Button*-Widget können so beispielsweise Dateien selektiert, in einer Liste angezeigt und anschließend hochgeladen werden.
  - Statt dem *File List*-Widget kann auch eine selbst definierte Dateivorschau erstellt werden, siehe [Eigene Dateivorschau erstellen](#).

### 2.6.4.5.3.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
Name Title	nameTitle	Spaltenüberschrift für die Dateinamen
Size Title	sizeTitle	Spaltenüberschrift für die Dateigröße

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Auf dieser Seite

#### Siehe auch:

- UI-Widget: File Upload Panel
- UI Widget: File Button

### 2.6.4.5.4 Eigene Dateivorschau erstellen

Statt dem *File List Widget* kann auch eine selbst definierte Dateivorschau erstellt werden.

#### 2.6.4.5.4.1 Beispiel

Folgendes Beispiel einer Dateivorschau soll nachgebaut werden:

Struktur im UI-Editor	Fertige Dateivorschau
<div> <div>File Upload Panel</div> <div> <div>Form Panel : File</div> <div> <div>Input : Name</div> <div>Input : Data type</div> <div>Input : File size</div> <div>Input : Formatted size</div> <div>Input : Last change</div> <div>+ Add/Drop Widgets</div> <div>File Button : Select</div> <div>File Button : Upload</div> <div>+ Add/Drop Widgets</div> <div>+ Add/Drop Widgets</div> </div> </div> </div>	<div> <div>File</div> <div> <div>Name X4 Activities.pdf</div> <div>Data type application/pdf</div> <div>File size 1240094</div> <div>Formatted size 1 MB</div> <div>Last change 1481875575780</div> </div> </div> <div> <div>File</div> <div> <div>Name X4 Developer Guide.pdf</div> <div>Data type application/pdf</div> <div>File size 693426</div> <div>Formatted size 677 KB</div> <div>Last change 1481875284392</div> </div> </div> <div> <div>Select</div> <div>Upload</div> </div>

#### 2.6.4.5.4.2 So geht's

Um die Dateivorschau nachzubauen, gehen Sie wie folgt vor:

1. *File Upload Panel* anlegen, siehe auch [UI-Widget: File Upload Panel](#).
2. *Upload Panel* konfigurieren:
  - unter **Upload URL** eine URL eingeben, an welche die Dateien per POST gesendet werden sollen
  - mit **Multiple** definieren, ob eine oder mehrere Dateien für den Upload ausgewählt werden können
3. *Form Panel*/Widget als Unterelement des *File Upload Panels* einfügen, siehe auch [UI Widget: Form Panel](#).  
Damit lässt sich für jede selektierte Datei ein Listeneintrag anzeigen.
4. *Form Panel* konfigurieren:
  - in **Label** einen Namen für das generierte *Fieldset* eingeben, z.B. *File*
  - unter **Data Context** den Wert *files* setzen, um die Werte anzeigen zu können
5. Für jedes anzuzeigende Attribut ein *Input*-Widget innerhalb des *Form Panels* anlegen, siehe [UI Widget: Input](#).
6. *Input*-Widget konfigurieren:
  - in **Label** einen Namen für das Feld eingeben, z.B. *Name*
  - unter **Value** das gewünschte Attribut eingeben (siehe unten, *Attribute*), z.B. `{{name}}`
7. Innerhalb des *File Upload Panels* zwei *File Buttons* für das Selektieren (Action: *Choose*) und Hochladen (Action: *Upload*) der Datei anlegen.
8. *File Buttons* entsprechend konfigurieren, siehe [UI Widget: File Button](#).
9. Einstellungen speichern und mit **Open Result** das Ergebnis öffnen.

#### 2.6.4.5.4.3 Attribute

Für die Darstellung der Dateivorschau können folgende Attribute verwendet werden:

Attribut	Beschreibung
name	Dateiname
type	Dateityp (Content-Type)
size	Dateigröße in Bytes
formattedSize	Formatierte und gerundete Dateigröße
lastModified	Änderungsdatum in Millisekunden

**Siehe auch:**

- [UI-Widget: File Upload Panel](#)
- [UI Widget: Form Panel](#)
- [UI Widget: Input](#)
- [UI Widget: File Button](#)

### 2.6.4.6 Formular-Widgets

Welche Formular-Widgets im *UI-Editor* des *X4 Web Designers* standardmäßig in der Widget-Palette unter **Forms** verfügbar sind.

#### 2.6.4.6.1 UI Widget: Form

Mit dem *Form*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* ein Formular für das Laden, Speichern und Löschen von JSON-Daten einer ReSTful Webservice-Ressource definieren.

##### 2.6.4.6.1.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Auto Load</b>	<code>autoLoad</code>	<p>Legt fest, ob beim Initialisieren des Formulars automatisch Daten vom konfigurierten REST-Service geladen werden sollen.</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• Ja / <code>true</code></li> <li>• Nein / <code>false</code> (Standard)</li> </ul>
<b>Main URL</b>	<code>dataResource.url</code>	<p>ReSTful Webservice-URL für die Daten-Ressource des Formulars</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• Beliebige gültige URL, z.B. <code>/X4/httpstarter/ReST/MyWebApp/Forms/MyForm/Main</code></li> <li>• Property-Binding, das eine gültige URL liefert, z.B. <code>{{#controller.getDataURL}}</code></li> </ul>
<b>Save URL</b>	<code>dataResource.saveURL</code>	URL die zum Speichern verwendet wird, falls abweichend von der Main URL.
<b>Remove URL</b>	<code>dataResource.removeURL</code>	URL die zum Löschen verwendet wird, falls abweichend von der Main URL.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Model Update Method</b>	<code>dataResource.modelUpdateMethod</code>	<p>Methode, wie der Datensatz beim Laden von Daten im Datenmodell bearbeitet werden soll.</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>update</code>: Alle alten Werte im Model entfernen und neue Daten einfügen (Objekt-Strukturen bleiben soweit möglich erhalten)</li> <li>• <code>set</code>: Daten im Model komplett überschreiben</li> <li>• <code>change</code>: Nur die geladenen Daten im Modell ändern, sonstige Daten im Modell bleiben unverändert</li> </ul>
<b>HTTP Save Method</b>	<code>dataResource.httpSaveMethod</code>	<p>Zu verwendende HTTP-Methode, um die Formulardaten zu speichern</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>PUT</code>: HTTP-PUT-Methode zum Speichern verwenden</li> <li>• <code>POST</code>: HTTP-POST-Methode zum Speichern verwenden</li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### 2.6.4.6.1.2 Widget-API

Für dieses UI-Widget ist zudem eine JavaScript-API verfügbar. Diese stellt folgende Methoden zur Verfügung.

Methode	Erläuterung	Rückgabewert
<b>load</b>	<p>Diese Methode startet das Laden von Daten für das Formular.</p> <pre>&lt;vc&gt;.getAPI().load()</pre>	
<b>save</b>	<p>Diese Methode startet das Speichern von Daten aus dem Formular.</p> <pre>&lt;vc&gt;.getAPI().save()</pre>	
<b>remove</b>	<p>Diese Methode startet das Löschen des Formular-Datensatzes.</p> <pre>&lt;vc&gt;.getAPI().remove()</pre>	

Methode	Erläuterung	Rückgabewert
<b>getDataResource</b>	<p>Diese Methode liefert die Daten-Modell-Ressource, die für das Formular verwendet wird, siehe API-Dokumentation für <code>x4.rest.DataModelResource</code> in Abschnitt <a href="#">Asynchrone Aktionen und ReST (async/async.js)</a>.</p> <pre>&lt;vc&gt;.getAPI().getDataResource(): x4.rest.DataModelResource</pre>	Instanz von <code>x4.rest.DataModelResource</code>

### 2.6.4.6.1.3 Widget-Events

Das Laden, Speichern und Löschen der Formulardaten kann durch folgende Events ausgelöst werden.

Event	Erläuterung	Richtung
data-load	Laden von Formulardaten	eingehend
data-save	Speichern der Formulardaten	eingehend
data-remove	Löschen des Datensatzes über den ReSTful Webservice	eingehend
data-load-ready	Wird ausgelöst, wenn das Laden von Formulardaten beendet ist (noch nicht ins Model übernommen)	ausgehend
data-change-ready	Wird ausgelöst, wenn das Laden und Ändern von Formulardaten im Model beendet ist	ausgehend
data-save-ready	Wird ausgelöst, wenn das Speichern von Formulardaten beendet ist	ausgehend
data-remove-ready	Wird ausgelöst, wenn das Löschen von Formulardaten beendet ist	ausgehend



**Tipp:** Das Widget *Data Button* bietet bereits Aktionen zur Auswahl. Formular-Events werden bei Klick auf die Schaltfläche ausgelöst.

#### Auf dieser Seite

##### Siehe auch:

- [UI Widget: Form Panel](#)
- [UI Widget: Form Array Panel](#)
- [UI Widget: Form Array Item](#)
- [UI Widget: Dynamic Form Panel](#)
- [UI-Widgets einfügen und verwalten](#)

### 2.6.4.6.2 UI Widget: Form Panel

Mit dem *Form Panel*-Widget lassen sich Formularbereiche im *UI-Editor* des *X4 Web Designers* definieren.

### 2.6.4.6.2.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Angezeigtes Group Label für den Formularbereich
<b>Base Layout</b>	baseLayout	Anordnung des Formularbereichs  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• <code>vertical</code>: Formularfelder über volle Breite des Bereichs darstellen; Labels werden oberhalb oder unterhalb angezeigt.</li> <li>• <code>horizontal</code>: Formularfelder mit fester Breite darstellen; Labels werden davor oder dahinter angezeigt (falls genügend Platz vorhanden ist).</li> <li>• <code>inline</code>: Formularfelder nebeneinander anordnen (falls genügend Platz vorhanden ist)</li> </ul>
<b>Input Label Position</b>	inputLabelPosition	Position der Feldbeschriftungen (Labels) innerhalb des Formularbereichs  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• <code>left</code>: Labels links neben den Formularfelder anzeigen</li> <li>• <code>right</code>: Labels rechts neben den Formularfeldern anzeigen</li> <li>• <code>none</code>: Keine Labels zu den Formularfeldern anzeigen</li> <li>• Kein Wert (Standard): Labels werden abhängig vom Typ des Formularfeldes angezeigt</li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)




### Auf dieser Seite

#### Siehe auch:

- [UI Widget: Form](#)
- [UI Widget: Form Array Panel](#)
- [UI Widget: Form Array Item](#)
- [UI Widget: Dynamic Form Panel](#)
- [UI-Widgets einfügen und verwalten](#)

### 2.6.4.6.3 UI Widget: Form Array Panel

Mit dem *Form Array Panel*-Widget lassen sich spezielle Formular-Bereiche für Arrays im *UI-Editor* des *X4 Web Designers* definieren. Diese Stellen spezielle Funktionalität für einfaches Bearbeiten von Arrays im Datenmodell bereit.


 Binden Sie das Array an ein untergeordnetes *Form Array Item*-Widget, nicht direkt an dieses Widget!

#### 2.6.4.6.3.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Angezeigtes Group Label für das Formular-Array
<b>Editable</b>	editable	Legt fest, ob Werte zu dem verknüpften Array hinzugefügt werden können oder nicht.  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• Ja / true</li> <li>• Nein / false (Standard)</li> </ul>
<b>Add Item Action</b>	addItemAction	Definiert die Aktion (UI-Binding), die verwendet werden soll um einen Eintrag zu ergänzen.  Beispiel: <code>{{#controller.addItemToModel}}</code>

 *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

 *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### Auf dieser Seite

#### Siehe auch:

- [UI Widget: Form](#)
- [UI Widget: Form Panel](#)
- [UI Widget: Form Array Item](#)
- [UI Widget: Dynamic Form Panel](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.6.4 UI Widget: Form Array Item

Mit dem *Form Array Item*-Widget lassen sich Elemente eines Arrays im *UI-Editor* des *X4 Web Designers* definieren. In Zusammenspiel mit dem *Form Array Panel*-Widget stellt dieses Funktionalität zum Bearbeiten von Arrays im Datenmodell bereit.

- ✓ **Tipp:** Um alle Elemente eines Arrays durch dieses Widget darzustellen kann die relative Referenz auf das Array im Modell (z.B. `myData.myArray`) durch die allgemeine Widget-Eigenschaft *Data Context* (JSON: `_dataContext`) definiert werden.

##### 2.6.4.6.4.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Remove Item Action</b>	<code>removeItemAction</code>	Definiert die Aktion (UI-Binding), die verwendet werden soll um den Eintrag zu löschen  <i>Beispiel:</i> <code>{ this.getModel().remove() }</code>

- ✓ **Tipp:** Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

- ✓ **Tipp:** Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### Auf dieser Seite

#### Siehe auch:

- [UI Widget: Form](#)
- [UI Widget: Form Panel](#)
- [UI Widget: Form Array Panel](#)
- [UI-Widgets einfügen und verwalten](#)

### 2.6.4.6.5 UI Widget: Dynamic Form Panel

Mit dem *Dynamic Form Panel*-Widget lassen sich dynamische Formularbereiche im *UI-Editor* des *X4 Web Designers* definieren. Dabei werden die Felder des Formulars automatisch, abhängig von Daten oder von einem JSON-Schema generiert.

#### 2.6.4.6.5.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Angezeigtes <i>Group Label</i> für den dynamischen Formularbereich
<b>Model Type</b>	modelType	Modell, das für die Generierung der Formularfelder verwendet werden soll  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• data: Daten-Modell</li> <li>• state: UI-State-Modell</li> <li>• prop: UI-Properties</li> </ul>
<b>Content Type</b>	contentType	Definiert, ob in dem Modell für die Generierung der Formularfelder JSON-Daten oder ein JSON-Schema erwartet wird  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• data: JSON-Daten verwenden</li> <li>• schema: JSON-Schema verwenden</li> </ul>
<b>Content Ref</b>	contentRef	Relative Referenz innerhalb des Modells, das durch <i>Model Type</i> festgelegt wurde

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Auf dieser Seite

##### Siehe auch:

- [UI Widget: Form](#)
- [UI Widget: Form Panel](#)
- [UI Widget: Form Array Panel](#)
- [UI Widget: Form Array Item](#)
- [UI Widget: Dynamic Form Panel](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.6.6 UI Widget: Input Wrapper

Mit dem *Input Wrapper*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* Standardfunktionalität für Formularelemente (Label & Layout) auf externe Widgets für Eingabefelder anwenden, z.B. *jQuery UI DatePicker*. Dazu muss das entsprechende externe Widget im Content-Bereich des *Input Wrapper* eingefügt werden.

##### 2.6.4.6.6.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
Label	label	Angezeigtes Label für das Formularfeld

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Auf dieser Seite

##### Siehe auch:

- [UI Widget: Input Group](#)
- [UI Widget: Input](#)
- [UI Widget: Input](#)
- [UI Widget: Textarea](#)
- [UI Widget: Checkbox](#)
- [UI Widget: Select](#)
- [UI Widget: Radio Group](#)
- [UI Widget: Button](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.6.7 UI Widget: Input Group

Mit dem *Input Group*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Formular-Elemente in Gruppen zusammenfassen.

❗ Fügen Sie die Eingabefelder direkt als Inline-Elemente in die Gruppe ein, um die *Input Group* korrekt darzustellen. Für die Standard-Eingabefelder muss dabei der Widget-Modus *Plain* ausgewählt sein.

##### 2.6.4.6.7.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Angezeigtes Group Label für den dynamischen Formularbereich

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### Auf dieser Seite

#### Siehe auch:

- [UI Widget: Input Wrapper](#)
- [UI Widget: Input](#)
- [UI Widget: Textarea](#)
- [UI Widget: Checkbox](#)
- [UI Widget: Select](#)
- [UI Widget: Radio Group](#)
- [UI Widget: Button](#)
- [UI-Widgets einfügen und verwalten](#)

### 2.6.4.6.8 UI Widget: Input

Mit dem *Input*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Texteingabefelder innerhalb von Formularen erstellen.

#### 2.6.4.6.8.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Angezeigter Beschriftungstext (Label) für das Formularfeld
<b>Value</b>	value	Statischer Wert für das Feld oder Binding, z.B. {{name}}

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### Auf dieser Seite

#### Siehe auch:

- [UI Widget: Form](#)
- [UI Widget: Form Panel](#)
- [UI Widget: Form Array Panel](#)
- [UI Widget: Form Array Item](#)
- [UI Widget: Dynamic Form Panel](#)
- [UI Widget: Input Wrapper](#)
- [UI Widget: Input Group](#)
- [UI Widget: Checkbox](#)
- [UI Widget: Select](#)
- [UI Widget: Radio Group](#)
- [UI Widget: Button](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.6.9 UI Widget: Textarea

Mit dem *Textarea*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* mehrzeilige Texteingabefelder innerhalb von Formularen erstellen.

##### 2.6.4.6.9.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Angezeigter Beschriftungstext (Label) für das Formularfeld
<b>Value</b>	value	Statischer Wert für das Feld oder Binding, z.B. {{description}}
<b>Cols</b>	cols	Anzahl Zeichen pro Zeile innerhalb des mehrzeiligen Texteingabefelds
<b>Rows</b>	rows	Anzahl Zeilen innerhalb des mehrzeiligen Texteingabefelds

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

**Auf dieser Seite****Siehe auch:**

- [UI Widget: Input Wrapper](#)
- [UI Widget: Input Group](#)
- [UI Widget: Input](#)
- [UI Widget: Checkbox](#)
- [UI Widget: Select](#)
- [UI Widget: Radio Group](#)
- [UI Widget: Button](#)
- [UI-Widgets einfügen und verwalten](#)

**2.6.4.6.10 UI Widget: Checkbox**

Mit dem *Checkbox*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Kontrollkästchen (Checkboxes) innerhalb von Formularen erstellen.

**2.6.4.6.10.1 Individuelle Widget-Eigenschaften**

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Angezeigter Beschriftungstext (Label) für das Formularfeld
<b>Value</b>	value	Statischer Wert ( true / false) für die Checkbox oder Binding, z.B. {{isEmployee}}

✔ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✔ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

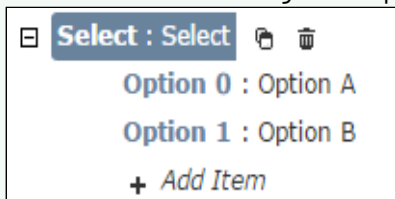
**Auf dieser Seite****Siehe auch:**

- [UI Widget: Input Wrapper](#)
- [UI Widget: Input Group](#)
- [UI Widget: Input](#)
- [UI Widget: Textarea](#)
- [UI Widget: Select](#)
- [UI Widget: Radio Group](#)
- [UI Widget: Button](#)
- [UI-Widgets einfügen und verwalten](#)

### 2.6.4.6.11 UI Widget: Select

Mit dem *Select*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Auswahllisten (Dropdown-Listen bzw. Picklisten) innerhalb von Formularen erstellen.

- ✓ *Tipp:* Wenn Sie ein *Select*-Widget in die Baumstruktur des UI-Editors einfügen, können Sie mit **+ Add Item** beliebig viele Optionseinträge für die Auswahlliste hinzufügen:



Im **Properties**-Tab können Sie zum markierten Optionseintrag in der Eigenschaft **Option** (option) eine Optionsbezeichnung und in **Label** (label) einen Beschriftungstext hinterlegen. Um **Einträge dynamisch aus einem Array** im Daten-Modell zu **erzeugen**, kann in der allgemeinen Widget-Eigenschaft *Data Context* auf das entsprechende Modell verwiesen werden, z.B. `#root.myOptions`. Modell-Bindings in den *Option*- und *Label*-Properties beziehen sich dann auf den Modell-Kontext des jeweiligen Array-Eintrags.

#### 2.6.4.6.11.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Angezeigter Beschriftungstext (Label) für das Formularfeld
<b>Value</b>	value	Statischer Wert für die ausgewählte Option oder Binding, z.B. <code>{{enumValue}}</code>

- ✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

- ✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)



### Auf dieser Seite

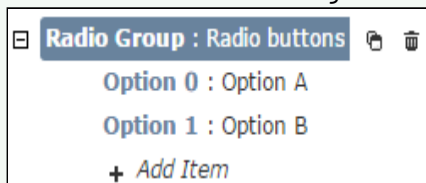
#### Siehe auch:

- [UI Widget: Input Wrapper](#)
- [UI Widget: Input Group](#)
- [UI Widget: Input](#)
- [UI Widget: Textarea](#)
- [UI Widget: Checkbox](#)
- [UI Widget: Radio Group](#)
- [UI Widget: Radio Item](#)
- [UI Widget: Button](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.6.12 UI Widget: Radio Group

Mit dem *Radio Group*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Radio-Button-Gruppen mit beliebig vielen Optionen innerhalb von Formularen erstellen.

- ✓ *Tipp:* Wenn Sie ein *Radio Group*-Widget in die Baumstruktur des UI-Editors einfügen, können Sie mit **+ Add Item** beliebig viele Optionseinträge für die Radio-Button-Gruppe hinzufügen:



Im **Properties**-Tab können Sie zum markierten Optionseintrag in der Eigenschaft **Option** (option) eine Optionsbezeichnung und in **Label** (label) einen Beschriftungstext hinterlegen.

- ✓ *Tipp:* Um Einträge dynamisch aus einem Array im Daten-Modell zu erzeugen, können Sie in der allgemeinen Widget-Eigenschaft *Data Context* auf das entsprechende Modell verweisen, z.B. `#root.myOptions`. Modell-Bindings in den *Option*- und *Label*-Properties beziehen sich dann auf den Modell-Kontext des jeweiligen Array-Eintrags.

- i Alternativ können Sie auch mit dem UI-Widget *Radio Item* einzelne Radio-Buttons erstellen und dieser einer Radio-Button-Gruppe zuweisen, siehe [UI Widget: Radio Item](#).

##### 2.6.4.6.12.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Angezeigter Beschriftungstext (Label) für das Formularfeld
<b>Value</b>	value	Statischer Wert der für die ausgewählte Option oder Binding, z.B. <code>{{enumValue}}</code>

Bezeichnung	JSON-Eigenschaft	Erläuterung
Group Name	groupName	Name der Radio-Button-Gruppe

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Auf dieser Seite

##### Siehe auch:

- [UI Widget: Input Wrapper](#)
- [UI Widget: Input Group](#)
- [UI Widget: Input](#)
- [UI Widget: Textarea](#)
- [UI Widget: Checkbox](#)
- [UI Widget: Select](#)
- [UI Widget: Button](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.6.13 UI Widget: Radio Item

Mit dem *Radio Item*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* einzelne Radio-Buttons innerhalb von Formularen erstellen.

##### 2.6.4.6.13.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
Label	label	Angezeigter Beschriftungstext (Label) für das Formularfeld
Value	value	Feldwert
Group Name	groupName	Name einer zugehörigen Radio-Button-Gruppe (optional)

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)


### Auf dieser Seite

#### Siehe auch:

- [UI Widget: Input Wrapper](#)
- [UI Widget: Input Group](#)
- [UI Widget: Input](#)
- [UI Widget: Textarea](#)
- [UI Widget: Checkbox](#)
- [UI Widget: Select](#)
- [UI Widget: Radio Group](#)
- [UI Widget: Button](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.6.14 UI Widget: Button


Mit dem *Button*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* einzelne Schaltflächen (Buttons) innerhalb von Formularen erstellen, etwa zum Absenden, Zurücksetzen oder für andere Aktionen.

 Zusätzlich zum Formular-Button steht das UI-Widget *Data Button* zur Verfügung, das speziell für das Auslösen der Standard-Formular-Aktionen (Laden, Speichern, Löschen) geeignet ist, siehe [UI Widget: Data Button](#).

##### 2.6.4.6.14.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Angezeigter Beschriftungstext (Label) für das Formularfeld
<b>Action</b>	action	Property-Binding, das bei einem Klick auf den Button ausgeführt werden soll.  Beispiel: <code>{{#controller.validateFormData}}</code>

 *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

 *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### Auf dieser Seite

#### Siehe auch:

- [UI Widget: Input Wrapper](#)
- [UI Widget: Input Group](#)
- [UI Widget: Input](#)
- [UI Widget: Textarea](#)
- [UI Widget: Checkbox](#)
- [UI Widget: Select](#)
- [UI Widget: Radio Group](#)
- [UI Widget: Data Button](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.6.15 UI Widget: Data Button

Mit dem *Data Button*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Schaltflächen (Buttons) zusätzlich zum normalen Formular-Button definieren, die speziell für die Standard-Formular-Aktionen (Laden, Speichern, Löschen) geeignet sind.

##### 2.6.4.6.15.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Angezeigter Beschriftungstext (Label) für das Formularfeld
<b>Action</b>	action	Aktion, die mit den Formulardaten ausgeführt werden soll  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• Load: Daten vom ReSTful Webservice laden und ins Formular schreiben</li> <li>• Save: Daten vom Formular an den ReSTful Webservice senden</li> <li>• Remove: Den Datensatz des Formulars über den ReSTful Webservice löschen</li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

**Auf dieser Seite:****Siehe auch:**


- [UI Widget: Input Wrapper](#)
- [UI Widget: Input Group](#)
- [UI Widget: Input](#)
- [UI Widget: Textarea](#)
- [UI Widget: Checkbox](#)
- [UI Widget: Select](#)
- [UI Widget: Radio Group](#)
- [UI Widget: Button](#)
- [UI-Widgets einfügen und verwalten](#)

### 2.6.4.7 Internationalisierungs-Widgets

Welche Internationalisierungs-Widgets im *UI-Editor* des *X4 Web Designers* standardmäßig in der Widget-Palette unter **i18n** verfügbar sind.

#### 2.6.4.7.1 i18n Bindings

Die Widget Collection für Internationalisierung stellt ein spezielles Binding-Prefix `#i18n` bereit, um in UI-Bindings einfach auf deren Funktionalitäten zugreifen zu können. Dieses stellt verschiedene Aktionen bereit, die anschließend durch Hash (#) gefolgt vom Aktionsnamen definiert werden können.

Aktionsname	Erläuterung	Beispiel
translate	<p>Liefert die Übersetzung für einen Eintrag in den Eigenschaften des aktuell aktiven <i>Locale</i>.</p> <p> Ist die Standardaktion und kann daher auch weggelassen werden.</p>	<ol style="list-style-type: none"> <li><code>#i18n.person.name</code></li> <li><code>#i18n.#translate.person.name</code></li> </ol>
format	Formatiert den ersten an das Binding übergebenen Parameter anhand eines Format-Typs (z. B. <code>date</code> ) und des aktuellen <i>Locale</i> .	<code>#i18n.#format.date</code>
parse	Parst den ersten an das Binding übergebenen Parameter anhand eines Format-Typs (z. B. <code>date</code> ) und des aktuellen <i>Locale</i> .	<code>#i18n.#parse.date</code>
validate	<p>Validiert den ersten an das Binding übergebenen Parameter anhand eines Format-Typs (z. B. <code>date</code>) und des aktuellen <i>Locale</i>.</p> <p>Ist die Validierung erfolgreich, wird <code>true</code> zurückgegeben, andernfalls <code>false</code>.</p>	<code>#i18n.#validate.date</code>
group	Siehe <a href="#">UI Widget: i18n Validation Group</a>	

**Siehe auch:**

- [UI Widget: i18n Validation Group](#)
- [Internationalisierung von Formularen](#)


### 2.6.4.7.2 i18n Standardformate



Standardformate der Widget Collection für Internationalisierung, die für Formatierung und Validierung von Formularwerten abhängig von *Locales* verwendet werden können.

Für die Definition weiterer Formate, siehe [i18n JavaScript-API](#).

#### 2.6.4.7.2.1 Globale Standardformate

Folgende globale Format-Typen werden standardmäßig mitgeliefert.

Format-Typen	Erläuterung und Optionen
<b>required</b>	Legt fest, dass der Wert nicht leer sein darf.
<b>length</b>	<p>Legt eine minimale und/oder eine maximale Länge für eine Text-Eingabe fest</p> <p><i>Mögliche Optionen:</i></p> <ul style="list-style-type: none"> <li>• <b>min</b>: Ganzzahl, welche die minimale Anzahl der Zeichen festlegt</li> <li>• <b>max</b>: Ganzzahl, welche die maximale Anzahl der Zeichen festlegt</li> </ul>
<b>equal</b>	<p>Legt fest, dass der Wert gleich einem anderen Wert sein muss.</p> <p><i>Mögliche Optionen:</i></p> <ul style="list-style-type: none"> <li>• <b>value</b>: Wert, mit dem der zu validierende Wert verglichen wird</li> </ul>
<b>integer</b>	<p>Legt fest, dass der Wert eine Ganzzahl sein muss.</p> <p><i>Mögliche Optionen:</i></p> <ul style="list-style-type: none"> <li>• <b>min</b>: Kleinste gültige Zahl</li> <li>• <b>max</b>: Größte gültige Zahl</li> </ul>
<b>number</b>	<p>Legt fest, dass der Wert eine Zahl sein muss.</p> <p> Als Komma-Zeichen wird ein Punkt (.) verwendet. Für abweichende Formate müssen entsprechende lokalisierte Formatter definiert werden (für deutsch, siehe unten).</p> <p><i>Mögliche Optionen:</i></p> <ul style="list-style-type: none"> <li>• <b>min</b>: Kleinste gültige Zahl</li> <li>• <b>max</b>: Größte gültige Zahl</li> </ul>

Format-Typen	Erläuterung und Optionen
<b>date</b>	<p>Ein gültiges Datum.</p> <p> Das Format ist nach ISO 8601 (YYYY-MM-TT) definiert. Für abweichende Formate müssen entsprechende lokalisierte Formatter definiert werden (für deutsch, siehe unten).</p> <p><i>Mögliche Optionen:</i></p> <ul style="list-style-type: none"> <li>• min: Kleinstes gültiges Datum formatiert nach ISO 8601 (YYYY-MM-TT)</li> <li>• max: Größtes gültiges Datum formatiert nach ISO 8601 (YYYY-MM-TT)</li> </ul>
<b>time</b>	<p>Eine gültige Uhrzeit.</p> <p> Das Format ist nach ISO 8601 (hh-mm-ss) definiert. Für abweichende Formate müssen entsprechende lokalisierte Formatter definiert werden (für deutsch, siehe unten).</p> <p><i>Mögliche Optionen:</i></p> <ul style="list-style-type: none"> <li>• min: Kleinste gültige Zeitangabe formatiert nach ISO 8601 (hh-mm-ss)</li> <li>• max: Größtes gültige Zeitangabe formatiert nach ISO 8601 (hh-mm-ss)</li> </ul>
<b>email</b>	<p>Eine gültige E-Mail-Adresse.</p> <p>Regulärer Ausdruck: <code>/^[a-zA-Z0-9.!#\$%&amp;'*+\/=?^_`{ }~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*\$/</code></p>

Format-Typen	Erläuterung und Optionen
<b>regex</b>	<p>Definiert ein Format mit Hilfe eines Regulären Ausdrucks (<a href="https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/Regular_Expressions">https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/Regular_Expressions</a>). Dabei können alle regulären Ausdrücke entweder direkt als JavaScript RegExp-Instanz oder als String definiert werden.</p> <p><i>Mögliche Optionen:</i></p> <ul style="list-style-type: none"> <li>• <b>regex</b>: Regulärer Ausdruck, der das Format festlegt</li> <li>• <b>formatRegex</b>: Regulärer Ausdruck, der definiert, wie ein nicht formatierter Wert für die Formatierung geparkt werden kann (falls abweichend von <b>regex</b>). Die Bestandteile des Ausdrucks, die im formatierten Ergebnis benötigt werden, müssen in Klammern geschrieben werden.</li> <li>• <b>formatTemplate</b>: Definiert einen String, der das Format für formatierte Werte festlegt. Für jeden geklammerten Wert im Regulären Ausdruck kann ein Platzhalter mit \$ und Position des Wertes definiert werden, beginnend bei 1, z. B. \$1 für den ersten Klammer-Ausdruck.</li> <li>• <b>parseRegex</b>: Regulärer Ausdruck, der definiert, wie ein formatierter Wert geparkt werden kann (falls abweichend von <b>regex</b>). Die Bestandteile des Ausdrucks, die im geparkten Ergebnis benötigt werden, müssen in Klammern geschrieben werden.</li> <li>• <b>parseTemplate</b>: Definiert einen String, der das Format für geparkte Werte festlegt. Für jeden geklammerten Wert im Regulären Ausdruck kann ein Platzhalter mit \$ und Position des Wertes definiert werden, beginnend bei 1, z. B. \$1 für den ersten Klammer-Ausdruck.</li> <li>• <b>validateRegex</b>: Regulärer Ausdruck, der für das Validieren eines formatierten Wertes verwendet wird (falls abweichend von <b>regex</b>).</li> <li>• <b>flags</b>: Ermöglicht die Definition von "Flags" zur Konfiguration der Regulären Ausdrücke, wenn diese als String definiert sind. Für detaillierte Informationen, siehe <a href="https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/Regular_Expressions#Erweiterte_Suche_mit_Options_(Flags)">https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/Regular_Expressions#Erweiterte_Suche_mit_Options_(Flags)</a></li> </ul>



**Beispiel: Regex Options für deutsche Uhrzeit**



```

/*
 * options1 und options2 führt zu gleichem Verhalten;
 * Das Flag "i" führt dazu, dass die Groß-/Kleinschreibung für "Uhr" egal ist
 */
var options1 = {
  formatRegex: /^(\d{2}):(\d{2})$/,
  formatTemplate: "$1.$2 Uhr",
  parseRegex: /^(\d{2})\.(\d{2})\sUhr$/i,
  parseTemplate: "$1:$2",
  validateRegex: /^((?:[0-1][0-9])|(?:2[0-3]))\.([0-5][0-9])\sUhr$/i
};
var options2 = {
  regex: "^(\d{2})\.(\d{2})",
  formatTemplate: "$1.$2 Uhr",
  parseTemplate: "$1:$2",
  validateRegex: "^((?:[0-1][0-9])|(?:2[0-3]))\.([0-5][0-9])\sUhr$",
  flags: "i"
};

```

**2.6.4.7.2.2 Deutsche Standardformate**

Folgende Format-Typen werden standardmäßig für das *Locale* "de" (Deutsch) mitgeliefert.

Format-Typen	Erläuterung und Optionen
<b>number</b>	<p>Eine Zahl in deutschem Format</p> <p> Als Komma-Zeichen wird das Komma-Zeichen (,) verwendet.</p> <p><i>Mögliche Optionen:</i></p> <ul style="list-style-type: none"> <li>min: Kleinste gültige Zahl (nicht in deutschem Format!)</li> <li>max: Größte gültige Zahl (nicht in deutschem Format!)</li> </ul>
<b>date</b>	<p>Ein gültiges Datum in deutschem Format (TT.MM.YYYY).</p> <p> Für nicht formatierte Werte wird das Format (YYYY-MM-TT) nach ISO 8601 erwartet.</p> <p><i>Mögliche Optionen:</i></p> <ul style="list-style-type: none"> <li>min: Kleinstes gültiges Datum formatiert nach ISO 8601 (YYYY-MM-TT)</li> <li>max: Größtes gültiges Datum formatiert nach ISO 8601 (YY-MM-TT)</li> </ul>

**Siehe auch:**

- [i18n JavaScript-API](#)
- [Internationalisierung von Formularen](#)

### 2.6.4.7.3 i18n JavaScript-API


Die Widget Collection für Internationalisierung stellt eine JavaScript-API bereit, durch die zusätzliche *Locales* sowie Formatierungs- und Validierungs-Methoden definiert werden können.


#### 2.6.4.7.3.1 Grundlegende Einstellungen


In dem Objekt `x4.ui.i18n.options` sind grundlegenden Einstellungen der Internationalisierungs-API definiert. Dieses unterstützt aktuell folgende Attribute:



Attribut	Erläuterung
<code>defaultLocaleKey</code>	Bezeichner für das Standard- <i>Locale</i> das verwendet wird, wenn kein <i>Locale</i> angegeben wurde. Standardwert dieser Option ist "default".

#### 2.6.4.7.3.2 Grundlegende API-Funktionen

Methode	Erläuterung	Rückgabewert
<b>getLocaleKey</b>	Liefert den Bezeichner des aktuell aktiven <i>Locale</i> .  <code>x4.ui.i18n.getLocaleKey(): string</code>	<i>Locale</i> -Key als String
<b>setLocaleKey</b>	Ändert das aktive <i>Locale</i> durch dessen Bezeichner.  <code>x4.ui.i18n.setLocaleKey([key: string])</code>  <i>Mögliche Parameter:</i> <ul style="list-style-type: none"> <li>• <code>key</code>: Bezeichner des neu zu aktivierenden <i>Locale</i>, z. B. "de".   Fehlt dieser Parameter, wird das Standard-<i>Locale</i> verwendet.</li> </ul>	

Methode	Erläuterung	Rückgabewert
<b>getLocale</b>	<p>Liefert ein <i>Locale</i>-Objekt.</p> <pre>x4.ui.i18n.getLocale([key: string]): &lt;Locale&gt;</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>key: Bezeichner des gewünschten <i>Locale</i>, z. B. "de".</li> <li> Wird dieser Parameter nicht angegeben, wird das aktuell aktive <i>Locale</i> zurückgegeben. Ist das <i>Locale</i> noch nicht vorhanden, wird automatisch ein neues angelegt.</li> </ul>	Das <i>Locale</i> -Objekt
<b>format</b>	<p>Formatiert einen Wert anhand des aktuell aktiven <i>Locale</i> und einem Format-Typ.</p> <pre>x4.ui.i18n.format(type: string, value [, options]): string</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>type: Bezeichner des Format-Typs, z. B. "date" für ein Datum.</li> <li>value: Der zu formatierende Wert.</li> <li>options: Ein optionales Objekt mit speziellen Zusatzoptionen für den definierten Format-Typ, z. B. Minimal- oder Maximalwerte bei Zahlen.</li> </ul>	Der formatierte Wert
<b>parse</b>	<p>Parst einen Wert anhand des aktuell aktiven <i>Locale</i> und einem Format-Typ.</p> <pre>x4.ui.i18n.parse(type: string, value [, options]): *</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>type: Bezeichner des Format-Typs, z. B. "date" für ein Datum.</li> <li>value: Der zu parsende Wert.</li> <li>options: Ein optionales Objekt mit speziellen Zusatzoptionen für den definierten Format-Typ, z. B. Minimal- oder Maximalwerte bei Zahlen.</li> </ul>	Der geparsete Wert



Methode	Erläuterung	Rückgabewert
<b>validate</b>	<p>Validiert einen Wert anhand des aktuell aktiven <i>Locale</i> und einem Format-Typ.</p> <pre>x4.ui.i18n.validate(type: string, value [, options]): boolean</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"><li>• <b>type</b>: Bezeichner des Format-Typs, z. B. "date" für ein Datum.</li><li>• <b>value</b>: Der zu validierende Wert.</li><li>• <b>options</b>: Ein optionales Objekt mit speziellen Zusatzoptionen für den definierten Format-Typ, z. B. Minimal- oder Maximalwerte bei Zahlen.</li></ul> <p> Eine Standardoption, die für alle Format-Typen angegebene werden kann, ist das boolesche Attribut <b>required</b>– Ist dieses <b>true</b>, werden auch leere Werte validiert (standardmäßig sind leere Werte automatisch valide).</p>	<p>Information, ob die Validierung erfolgreich war (<b>true</b>) oder nicht (<b>false</b>).</p>

Methode	Erläuterung	Rückgabewert
<b>useFormatterConfig</b>	<p>Definiert ein Objekt, das für eine oder mehrere <i>Locales</i> Format-Typen festlegt.</p> <pre>x4.ui.i18n.useFormatterConfig(config: object)</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>• <b>config:</b> Das Objekt, das die Format-Typen enthält.</li> </ul> <div data-bbox="464 741 1018 1402" style="border: 1px solid #add8e6; padding: 10px; margin-top: 10px;"> <p> • Auf oberster Ebene müssen die Attribut-Namen des Objektes dem Bezeichner eines <i>Locale</i> entsprechen. Diese müssen wiederum ein Objekt enthalten, dessen Attribut-Namen den Bezeichnern der Format-Typen entsprechen und die ein <i>Formatter</i>-Objekt enthalten.</p> <p>• Um für ein einzelnes <i>Locale</i> Format-Typen zu konfigurieren, kann auch die Methode <code>useFormatterConfig</code> der entsprechenden <i>Locale</i>-Instanz verwendet werden, z. B. <code>x4.ui.i18n.getLocale('de').useFormatterConfig(config.de)</code>.</p> </div>	
<b>setGlobalFormatter</b>	<p>Definiert einen globalen Format-Typ, der für alle <i>Locales</i> verwendet werden kann.</p> <p> Existiert für das aktuelle <i>Locale</i> ein Format-Typ mit gleichem Bezeichner, wird der lokale anstatt dem globalen verwendet. Damit kann diese Methode verwendet werden, um Standard-Formate ("Fallbacks") und Formate, die unabhängig von der Region sind zu definieren, z. B. E-Mail-Adressen.</p> <pre>x4.ui.i18n.setGlobalFormatter(type: string, formatter: &lt;Formatter&gt;)</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>• <b>type:</b> Bezeichner des Format-Typs.</li> <li>• <b>formatter:</b> Das <i>Formatter</i>-Objekt dieses Format-Typs.</li> </ul>	

### 2.6.4.7.3.3 *Locale*-API

Die Methoden einer *Locale*-Instanz, die beispielsweise von der Methode `x4.ui.i18n.getLocale` bereitgestellt wird.

Methode	Erläuterung	Rückgabewert
<b>getKey</b>	Liefert den Bezeichner des <i>Locale</i> .  <code>&lt;locale&gt;.getKey(): string</code>	<i>Locale</i> -Key als String
<b>format</b>	Formatiert einen Wert für das <i>Locale</i> anhand eines Format-Typs.  <code>&lt;locale&gt;.format(type: string, value [, options]): string</code>  <i>Mögliche Parameter:</i> <ul style="list-style-type: none"> <li>• <code>type</code>: Bezeichner des Format-Typs, z. B. "date" für ein Datum.</li> <li>• <code>value</code>: Der zu formatierende Wert.</li> <li>• <code>options</code>: Ein optionales Objekt mit speziellen Zusatzoptionen für den definierten Format-Typ, z. B. Minimal- oder Maximalwerte bei Zahlen.</li> </ul>	Der formatierte Wert
<b>parse</b>	Parst einen Wert für das <i>Locale</i> anhand eines Format-Typs.  <code>&lt;locale&gt;.parse(type: string, value [, options]): *</code>  <i>Mögliche Parameter:</i> <ul style="list-style-type: none"> <li>• <code>type</code>: Bezeichner des Format-Typs, z. B. "date" für ein Datum.</li> <li>• <code>value</code>: Der zu parsende Wert.</li> <li>• <code>options</code>: Ein optionales Objekt mit speziellen Zusatzoptionen für den definierten Format-Typ, z. B. Minimal- oder Maximalwerte bei Zahlen.</li> </ul>	Der geparsete Wert

Methode	Erläuterung	Rückgabewert
<b>validate</b>	<p>Validiert einen Wert für das <i>Locale</i> anhand eines Format-Typs.</p> <pre>&lt;locale&gt;.validate(type: string, value [, options]): boolean</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>• type: Bezeichner des Format-Typs, z. B. "date" für ein Datum.</li> <li>• value: Der zu validierende Wert.</li> <li>• options: Ein optionales Objekt mit speziellen Zusatzoptionen für den definierten Format-Typ, z. B. Minimal- oder Maximalwerte bei Zahlen.</li> </ul> <p> Eine Standardoption, die für alle Format-Typen angegebene werden kann, ist das boolesche Attribut <code>required</code>– Ist dieses <code>true</code>, werden auch leere Werte validiert (standardmäßig sind leere Werte automatisch valide).</p>	Information, ob die Validierung erfolgreich war ( <code>true</code> ) oder nicht ( <code>false</code> ).
<b>useFormatterConfig</b>	<p>Definiert ein Objekt, das Format-Typen für dieses <i>Locale</i> festlegt.</p> <pre>&lt;locale&gt; .useFormatterConfig(config: object)</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>• config: Das Objekt, das die Format-Typen enthält.</li> </ul> <p> Auf oberster Ebene müssen die Attribut-Namen den Bezeichnern der Format-Typen entsprechen und ein <i>Formatter</i>-Objekt enthalten.</p>	
<b>setFormatter</b>	<p>Definiert einen einzelnen Format-Typ für dieses <i>Locale</i>.</p> <pre>&lt;locale&gt;.setFormatter(type: string, formatter: &lt;Formatter&gt;)</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>• type: Bezeichner des Format-Typs.</li> <li>• formatter: Das <i>Formatter</i>-Objekt dieses Format-Typs.</li> </ul>	


#### 2.6.4.7.3.4 Definition von *Formatter*-Objekten

*Formatter*-Objekte werden verwendet, um Formate für die Internationalisierungs-API zu definieren.

##### Methoden

In einem *Formatter*-Objekt können die Methoden zum Formatieren, Parsen und Validieren von Werten definiert werden. Jede dieser Methoden erhält als ersten Parameter einen Wert und als zweiten Parameter ein Objekt mit Konfigurationsoptionen.

Methoden-Name	Beschreibung
<code>format</code>	Bekommt einen unformatierten Wert als ersten Parameter und liefert den formatierten Wert zurück. <i>Beispiel:</i> Eingabe = 1.5, Ergebnis = "1,5"
<code>parse</code>	Bekommt einen formatierten Wert als ersten Parameter und liefert diesen in der erwarteten technischen Repräsentation zurück. <i>Beispiel:</i> Eingabe = "1,5", Ergebnis = 1.5
<code>validate</code>	Prüft, ob der Wert des ersten Parameters eine gültige Repräsentation des gewünschten Formates ist und liefert einen booleschen Wert zurück ( <code>true</code> wenn der Wert dem Format entspricht, andernfalls <code>false</code> ).

 Jede dieser Methoden ist optional. Fehlt eine Methode, wird der übergebene Wert einfach kopiert bzw. im Fall von `validate` in einen booleschen Wert konvertiert.

##### Standard-Optionen

Über ein `options`-Attribut im *Formatter*-Objekt können Standardwerte für die Konfigurationsoptionen des Formates festgelegt werden.

##### Vererbung

Ein *Formatter*-Objekt kann Funktionalität von einem bestehenden Format "erben", indem der Name des anderen Formates in einem Attribut `type` definiert wird (dabei wird zuerst im aktuellen *Locale* nach einem passenden Format gesucht, danach in den globalen Formaten). Es ist dann möglich im neuen Format abweichende Standard-Optionen zu definieren, oder einzelne Methoden zu überschreiben.



**Beispiel**

```
x4.ui.i18n.setGlobalFormatter('myNumber', {
  options: { // default/fallback options
    min: Number.MIN_VALUE,
    max: Number.MAX_VALUE
  },

  validate: function(value, options){
    if(!/^[0-9]+(,[0-9]+)?$/ .test(value)){
      return false;
    }
    var num = this.parse(value, options);

    return options.min <= num && num <= options.max;
  },
  format: function(value, options){
    return String(value).replace('.', ',');
  },
  parse: function(value, options){
    return parseFloat(value.replace(',', '.'));
  }
});

x4.ui.i18n.setGlobalFormatter('myPosNumber', {
  type: 'myNumber',
  options: {
    min: 1
  }
});
```

#### 2.6.4.7.4 UI Widget: i18n Initialization

Mit dem *i18n Initialization-Widget* lassen sich im *UI-Editor* des *X4 Web Designers* Basis-Einstellungen für die Internationalisierung einer Seite definieren.



Dieses Widget enthält keine UI-Inhalte und sollte für eine gute Übersichtlichkeit möglichst zu Beginn des UI-Baumes auf oberster Ebene definiert werden.

##### 2.6.4.7.4.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Locale</b>	locale	<p>Kürzel der Region, für welche die Seite dargestellt werden soll.</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• default: Standard (Englische Schrift und Standard-Formate)</li> <li>• de: Deutsch/Deutschland</li> </ul> <p><b>? Unknown Attachment</b> Weitere Regionen können über die JavaScript-API ergänzt werden, siehe <a href="#">i18n JavaScript-API</a></p>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Siehe auch:

- [i18n JavaScript-API](#)
- [Internationalisierung von Formularen](#)


### 2.6.4.7.5 UI Widget: i18n Properties Service


Mit dem *i18n Properties Service*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* für die Internationalisierung von Texten Eigenschaften von einem REST-Service laden.


#### 2.6.4.7.5.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Properties URL</b>	url	Legt die Basis-URL des REST-Services fest, von dem die Eigenschaften geladen werden sollen.

Bezeichnung	JSON-Eigenschaft	Erläuterung
URL Locale Parameter	urlLocaleParam	<p>Legt den Namen des URL-Parameters fest, durch den das aktuelle <i>Locale</i>-Kürzel der Seite an den REST-Service übergeben wird (optional).</p> <div>  Das <i>Locale</i> kann alternativ auch direkt in der URL kodiert oder aus sonstigen Meta-Informationen (z.B. Cookies) ermittelt werden. </div>
Properties Type	propDataType	<p>Definiert das erwartete Datenformat der Eigenschaften.</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>xml</code>: Java-Eigenschaften als XML-Format (siehe <a href="#">Internationalisierung</a>)</li> <li>• <code>json</code>: Ein JSON-Objekt</li> </ul>
Base Reference	baseRef	Prefix für die geladenen Eigenschaften, um Konflikte bei der Verwendung von mehreren Properties-Services zu vermeiden.

 *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

 *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Siehe auch:


- [Internationalisierung](#)
- [Internationalisierung von Formularen](#)

### 2.6.4.7.6 UI Widget: i18n Properties File

Mit dem *i18n Properties File*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Eigenschaften für die Internationalisierung von Texten aus einer Datei laden.

#### 2.6.4.7.6.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
Base URL	baseUr1	<p>Legt die Basis-URL der Datei fest (ohne <i>Locale</i> und Dateieindung) in der die Eigenschaften definiert sind.</p> <p> Unknown Attachment Das <i>Locale</i> wird durch Unterstrich getrennt an die URL angehängt. Entspricht das <i>Locale</i> dem Default-Wert, wird keine <i>Locale</i>-Erweiterung an die URL angehängt.</p>
Properties Type	propDataType	<p>Definiert das erwartete Datenformat der Eigenschaften. Dieser Wert wird auch als Dateieindung, durch Punkt getrennt, an die Basis-URL angehängt.</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• xml: Java-Eigenschaften als XML-Format (siehe <a href="#">Internationalisierung</a>)</li> <li>• json: Ein JSON-Objekt</li> </ul>
Base Reference	baseRef	Prefix für die geladenen Eigenschaften, um Konflikte bei der Verwendung von mehreren Properties-Services zu vermeiden.

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Siehe auch:

- [Internationalisierung](#)
- [Internationalisierung von Formularen](#)

#### 2.6.4.7.7 UI Widget: i18n Validation Group

Mit dem *i18n Validation Group*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Validierungs- und Formatierungsregeln für einen Formularbereich definieren.

### 2.6.4.7.7.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Auto Validation</b>	<code>autoValidation</code>	Legt fest, ob die Validierung automatisch beim Laden des Formulars gestartet werden soll.
<b>Validations</b>	<code>validation</code>	<p>Ein Objekt, das die Validierungsregeln dieser Gruppe definiert. Die Attribut-Namen dieses Objektes legen dabei die Namen der einzelnen Regel-Definitionen fest.</p> <p>Jedes dieser Attribute enthält ein Objekt mit folgenden Attributen:</p> <ul style="list-style-type: none"> <li>• <b>Value</b> (<code>value</code>): Der zu validierende Wert. In der Regel sollte hier ein UI-Binding definiert werden, z. B. "<code>{{name}}</code>", um den Eintrag <code>name</code> im Data-Model zu validieren.</li> <li>• <b>Required</b> (<code>required</code>): Information, ob die Angabe eines gültigen Wertes erforderlich ist oder der Wert auch leer sein darf.</li> <li>• <b>Format Type</b> (<code>type</code>): Name der Formatierungsregel, die für die Validierung verwendet werden soll (optional).</li> <li>• <b>Message</b> (<code>message</code>): Die Nachricht, die bei fehlgeschlagener Validierung angezeigt werden soll, optional als UI-Binding.</li> <li>• <b>Options</b> (<code>options</code>): Ein optionales Objekt, das beliebige Konfigurationsparameter für die gewählte Formatierungsregel definieren kann. Dabei kann jedes Attribut bei Bedarf als UI-Binding definiert werden, was eine dynamische Parametrierung der Regeln sehr einfach macht.</li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### 2.6.4.7.7.2 Zugriff auf Regeln in Bindings

❗ In allen Beispielen in diesem Kapitel wird Regel-Name `name` verwendet. Dieser entspricht einem Attributnamen in `validation`-Objekt der *Custom Properties* (siehe oben). Um diese Regel einfach in einem Value-Binding, z. B. in einem Eingabefeld zu referenzieren, kann das Binding `#validation.name` verwendet werden.

Um innerhalb einer Validation Group auf eine Regel der Gruppe zuzugreifen, kann das Binding-Prefix `#i18n.#group` verwendet werden. Dieses stellt verschiedene Aktionen bereit, die danach durch Hash (`#`) gefolgt vom Aktionsnamen definiert werden können. Nach der Aktion folgt der Name der Regel, auf welche die Aktion angewendet werden soll.

Aktionsname	Erläuterung	Beispiel
bind	Ermöglicht das Lesen und Ändern des Wertes und sorgt dafür, dass in beiden Fällen automatisch Validierung und Formatierung des Wertes durchgeführt wird (Validierung nur wenn Auto-Validierung aktiviert ist - siehe <code>startValidation</code> ).  Ist die Standardaktion und kann deshalb auch weggelassen werden.	1. <code>#i18n.#group.name</code> 2. <code>#i18n.#group.#bind.name</code>
get	Liest den formatierten Wert.	<code>#i18n.#group.#get.name</code>
change	Erwartet einen formatierten Wert, um den Wert der Regel zu ändern.	<code>#i18n.#group.#change.name</code>
validate	Validiert den aktuellen Wert der Regel und liefert das boolesche Ergebnis der Validierung zurück.	<code>#i18n.#group.#validate.name</code>
isValid	Prüft, ob die letzte Validierung der Regel erfolgreich war und liefert das Ergebnis als booleschen Wert. Wurde noch keine Validierung durchgeführt, liefert die Aktion <code>undefined</code> , was in den meisten Fällen wie <code>false</code> interpretiert wird.	<code>#i18n.#group.#isValid.name</code>
isInvalid	Prüft, ob die letzte Validierung der Regel <b>nicht erfolgreich</b> war und liefert das Ergebnis als booleschen Wert.	<code>#i18n.#group.#isInvalid.name</code>
message	Liefert die Nachricht der Validierungsregel.	<code>#i18n.#group.#message.name</code>
errorMessage	Liefert die Nachricht der Validierungsregel nur dann, wenn die Validierung nicht erfolgreich war, andernfalls <code>undefined</code> .	<code>#i18n.#group.#errorMessage.name</code>
errorMessages	Liefert die Nachrichten aller Validierungsregeln, deren Validierung nicht erfolgreich war, als Array. Da sich diese Aktion auf die Gruppe bezieht, muss kein Regel-Name angegeben werden.	<code>#i18n.#group.#errorMessages</code>
options	Erlaubt den Zugriff auf ein Attribut des <code>options</code> -Objektes einer Validierungsregel. Dadurch kann z.B. über das <code>required</code> -Attribut geprüft werden, ob es sich bei dem entsprechenden Formularfeld um ein Pflichtfeld handelt (Beispiel 1) oder es können Konfigurationswerte in dynamischen Fehlermeldungen verwendet werden ( <code>min</code> -Attribut in Beispiel 2).	1. <code>#i18n.#group.#options.name.required</code> 2. <code>#i18n.#group.#options.name.min</code>

Aktionsname	Erläuterung	Beispiel
startValidation	Startet automatische Validierung und führt dabei eine Validierung aller Regeln der Gruppe aus. Da sich diese Aktion auf die Gruppe bezieht, muss kein Regel-Name angegeben werden.	#i18n.#group.#startValidation

#### 2.6.4.7.7.3 Widget-API


Für dieses UI-Widget ist zudem eine JavaScript-API verfügbar. Diese stellt folgende Methoden zur Verfügung.


Methode	Erläuterung	Rückgabewert
<b>get</b>	Liest den formatierten Wert einer Validierungsregel.  <code>&lt;vc&gt;.getAPI().get(name: string): string</code>  <i>Mögliche Parameter:</i> <ul style="list-style-type: none"> <li>name: Name der Validierungsregel</li> </ul>	Wert der Validierungsregel
<b>change</b>	Erwartet einen formatierten Wert, um den Wert einer Regel zu ändern.  <code>&lt;vc&gt;.getAPI().change(name: string, value)</code>  <i>Mögliche Parameter:</i> <ul style="list-style-type: none"> <li>name: Name der Validierungsregel</li> <li>value: Neuer Wert für die Validierungsregel</li> </ul>	
<b>validate</b>	Validiert eine Validierungsregel oder die ganze Gruppe.  <code>&lt;vc&gt;.getAPI().validate([name: string]): boolean</code>  <i>Mögliche Parameter:</i> <ul style="list-style-type: none"> <li>name: Name der Validierungsregel. Fehlt dieser Parameter, werden alle Regeln der Gruppe validiert.</li> </ul>	true, wenn die Validierung erfolgreich war, andernfalls false.

Methode	Erläuterung	Rückgabewert
<b>isValid</b>	<p>Prüft, ob die letzte Validierung einer oder aller Regeln erfolgreich war.</p> <pre>&lt;vc&gt;.getAPI().isValid([name: string]): boolean</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>name: Name der Validierungsregel. Fehlt dieser Parameter, werden alle Regeln der Gruppe validiert.</li> </ul>	true, wenn die Validierung erfolgreich war, false wenn sie nicht erfolgreich war und undefined wenn noch keine Validierung durchgeführt wurde.
<b>isInvalid</b>	<p>Prüft, ob die letzte Validierung einer oder aller Regeln <b>nicht erfolgreich</b> war.</p> <pre>&lt;vc&gt;.getAPI().isInvalid([name: string]): boolean</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>name: Name der Validierungsregel. Fehlt dieser Parameter, werden alle Regeln der Gruppe validiert.</li> </ul>	true, wenn die Validierung nicht erfolgreich war, andernfalls false.
<b>getMessage</b>	<p>Liefert die Nachricht einer Validierungsregel.</p> <pre>&lt;vc&gt;.getAPI().getMessage(name: string): string</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>name: Name der Validierungsregel.</li> </ul>	Die Nachricht der Regel.
<b>getErrorMessage</b>	<p>Liefert eine Fehler-Nachricht einer Validierungsregel.</p> <pre>&lt;vc&gt;.getAPI().getErrorMessage(name: string): string</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>name: Name der Validierungsregel, wenn die Validierung nicht erfolgreich war, andernfalls undefined.</li> </ul>	Die Nachricht der Regel im Fehlerfall.



Methode	Erläuterung	Rückgabewert
<b>getErrorMessage</b>	<p>Liefert eine Liste der Nachrichten aller Validierungsregeln, deren Validierung fehlgeschlagen ist.</p> <p><code>&lt;vc&gt;.getAPI().getErrorMessage(): Array</code></p>	Die Liste der Nachrichten als Array.
<b>isAutoValidationEnabled</b>	<p>Prüft, ob die automatische Validierung für diese Gruppe aktiviert ist.</p> <p><code>&lt;vc&gt;.getAPI().isAutoValidationEnabled(): boolean</code></p>	Information, ob die automatische Validierung aktiviert ist als boolescher Wert.
<b>setAutoValidationEnabled</b>	<p>Aktiviert oder deaktiviert die automatische Validierung für diese Gruppe.</p> <p><code>&lt;vc&gt;.getAPI().setAutoValidationEnabled(enabled: boolean)</code></p> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"><li>• <code>enabled</code>: Information, ob die automatische Validierung aktiviert (<code>true</code>) oder deaktiviert (<code>false</code>) wird.</li></ul>	

Methode	Erläuterung	Rückgabewert
<b>setDefaultUIIntegrationHandler</b>	<p>Definiert eine Funktion, die Standardfunktionalität zur Darstellung der Validierung im UI festlegt.</p> <pre>&lt;vc&gt;.getAPI().setDefaultUIIntegrationHandler(handler: Function)</pre> <p><i>Mögliche Parameter:</i></p> <ul style="list-style-type: none"> <li>• handler: Funktion, die folgende Parameter übergeben bekommt: <ul style="list-style-type: none"> <li>a. Name der Validierungsgruppe</li> <li>b. View-Kontext in dem die Regel per bind-Aktion (siehe oben) verwendet wurde</li> </ul> </li> </ul> <p> Unknown Attachment</p> <p>Alternativ kann auch null übergeben werden, um die Standarddarstellung des Widgets ersatzlos zu deaktivieren.</p>	

-  Der Handler wird nur einmal bei der Initialisierung der View ausgeführt und muss nach der Erzeugung, aber vor dem Binding der View (Events view-prepare oder view-attach) definiert werden.

**Beispiel**

```

vc.on('view-prepare', function(){
    vc.getAPI().setDefaultUIIntegrationHandler(function(ruleName, targetVC){
        if(targetVC.hasDomHook('input')){
            targetVC.getDomHook('input').applyBindings({
                attr: {
                    title: "#i18n.#group.#errorMessage." + ruleName
                },
                cssClass: {
                    "x4-error": "#i18n.#group.#isInvalid." + ruleName
                }
            });
            targetVC.getDomHook('_root').applyBindings({
                cssClass: {
                    "x4-required": "!!#i18n.#group.#options." + ruleName +
'.required'
                }
            });
        }
    });
});
};

```

**Siehe auch:**

- [Validierung von Formularen](#)

#### 2.6.4.7.8 UI Widget: i18n Schema Validation Group

Mit dem *i18n Schema Validation Group*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Validierungs- und Formatierungsregeln mit Hilfe eines JSON-Schemas definieren.

##### 2.6.4.7.8.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Auto Validation</b>	autoValidation	Legt fest, ob die Validierung automatisch beim Laden des Formulars gestartet werden soll
<b>Validations</b>	schemaUrl	Die URL, von der das JSON-Schema geladen werden soll

✓ *Tip:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓  *Tipp: Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)*

#### 2.6.4.7.8.2 Zugriff auf Regeln in Bindings

Der Zugriff auf Validierungsregeln in Bindings funktioniert genauso wie auch im [i18n Validation Group](#)-Widget. Die Namen der Regeln werden hierbei durch den Namen der JSON-Property-Keys im Schema definiert. Bei Objekt-Hierarchien werden die Namen der Keys durch Bindestrich/Minus (-) getrennt.

#### 2.6.4.7.8.3 Widget-API

Für dieses UI-Widget ist zudem eine JavaScript-API verfügbar. Sie entspricht der API des [i18n Validation Group](#)-Widgets.

##### Siehe auch:

- [Validierung von Formularen](#)

#### 2.6.4.7.9 UI Widget: i18n Validation

Mit dem *i18n Validation*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* eine Validierungs- und Formatierungsregel für einen Formularbereich definieren. Dieses Widget benötigt ein übergeordnetes [i18n Validation Group](#)-Widget und registriert die Regel automatisch in dieser Gruppe.

##### 2.6.4.7.9.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
Validation ID	id	Optionale ID der Validierungsregel, mit der die Regel in der übergeordneten <i>Validaton Group</i> registriert wird. Fehlt die ID wird automatisch eine ID generiert.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Validations</b>	<code>i18nValidation</code>	<p>Ein Objekt, das die Validierungsregel definiert. Dies erfolgt mit folgenden Attributen:</p> <ul style="list-style-type: none"> <li>• <b>Value</b> (<code>value</code>): Der zu validierende Wert. In der Regel sollte hier ein UI-Binding definiert werden, z. B. "<code>{{name}}</code>", um den Eintrag <code>name</code> im Data-Model zu validieren.</li> <li>• <b>Format Type</b> (<code>type</code>): Name der Formatierungsregel, die für die Validierung verwendet werden soll (optional).</li> <li>• <b>Message</b> (<code>message</code>): Die Nachricht, die bei fehlgeschlagener Validierung angezeigt werden soll, optional als UI-Binding.</li> <li>• <b>Options</b> (<code>options</code>): Ein optionales Objekt, das beliebige Konfigurationsparameter für die gewählte Formatierungsregel definieren kann. Dabei kann jedes Attribut bei Bedarf als UI-Binding definiert werden, was eine dynamische Parametrierung der Regeln sehr einfach macht.</li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### 2.6.4.7.9.2 Zugriff auf Regeln in Bindings

❗ Um diese Regel einfach in einem Value-Binding, z. B. in einem Eingabefeld zu referenzieren, kann das Binding `#i18n.#item` verwendet werden.

Um innerhalb des Validation Widgets auf die Validierung zuzugreifen, kann das Binding-Prefix `#i18n.#item` verwendet werden. Dieses stellt verschiedene Aktionen bereit, die danach durch Hash (`#`) gefolgt vom Aktionsnamen definiert werden können.

Aktionsname	Erläuterung	Beispiel
<code>bind</code>	<p>Ermöglicht das Lesen und Ändern des Wertes und sorgt dafür, dass in beiden Fällen automatisch Validierung und Formatierung des Wertes durchgeführt wird (Validierung nur wenn Auto-Validierung aktiviert ist - siehe <code>startValidation</code>).</p> <p>Ist die Standardaktion und kann deshalb auch weggelassen werden.</p>	<ol style="list-style-type: none"> <li>1. <code>#i18n.#item</code></li> <li>2. <code>#i18n.#item.#bind</code></li> </ol>
<code>get</code>	Liest den formatierten Wert.	<code>#i18n.#item.#get</code>
<code>change</code>	Erwartet einen formatierten Wert, um den Wert der Regel zu ändern.	<code>#i18n.#item.#change</code>

Aktionsname	Erläuterung	Beispiel
validate	Validiert den aktuellen Wert der Regel und liefert das boolesche Ergebnis der Validierung zurück.	#i18n.#item.#validate
isValid	Prüft, ob die letzte Validierung der Regel erfolgreich war und liefert das Ergebnis als booleschen Wert. Wurde noch keine Validierung durchgeführt, liefert die Aktion undefined, was in den meisten Fällen wie false interpretiert wird.	#i18n.#item.#isValid
isInvalid	Prüft, ob die letzte Validierung der Regel <b>nicht erfolgreich</b> war und liefert das Ergebnis als booleschen Wert.	#i18n.#item.#isInvalid
message	Liefert die Nachricht der Validierungsregel.	#i18n.#item.#message
errorMessage	Liefert die Nachricht der Validierungsregel nur dann, wenn die Validierung nicht erfolgreich war, andernfalls undefined.	#i18n.#item.#errorMessage
startValidation	Startet die automatische Validierung und führt dabei eine Validierung aller Regeln der Gruppe aus.	#i18n.#item.#startValidation

#### 2.6.4.7.9.3 Widget-API

Für dieses UI-Widget ist zudem eine JavaScript-API verfügbar. Diese stellt folgende Methoden zur Verfügung.

Methode	Erläuterung	Rückgabewert
<b>get</b>	Liest den formatierten Wert einer Validierungsregel.  <vc>.getAPI().get(): string	Wert der Validierungsregel
<b>change</b>	Erwartet einen formatierten Wert, um den Wert einer Regel zu ändern.  <vc>.getAPI().change(value)	
<b>validate</b>	Validiert eine Validierungsregel oder die ganze Gruppe.  <vc>.getAPI().validate(): boolean	true, wenn die Validierung erfolgreich war, andernfalls false.
<b>isValid</b>	Prüft, ob die letzte Validierung einer oder aller Regeln erfolgreich war.  <vc>.getAPI().isValid(): boolean	true, wenn die Validierung erfolgreich war, false wenn sie nicht erfolgreich war und undefined wenn noch keine Validierung durchgeführt wurde.

Methode	Erläuterung	Rückgabewert
<b>isInvalid</b>	Prüft, ob die letzte Validierung einer oder aller Regeln <b>nicht erfolgreich</b> war.  <code>&lt;vc&gt;.getAPI().isInvalid(): boolean</code>	true, wenn die Validierung nicht erfolgreich war, andernfalls false.
<b>getMessage</b>	Liefert die Nachricht einer Validierungsregel.  <code>&lt;vc&gt;.getAPI().getMessage(): string</code>	Die Nachricht der Regel.
<b>getErrorMessage</b>	Liefert eine Fehler-Nachricht einer Validierungsregel.  <code>&lt;vc&gt;.getAPI().getErrorMessage(): string</code>	Die Nachricht der Regel im Fehlerfall.

**Siehe auch:**

- [Validierung von Formularen](#)

## 2.6.4.8 jQuery UI-Widgets


Welche *jQuery UI*-Widgets im *UI-Editor* des *X4 Web Designers* standardmäßig in der Widget-Palette unter **jQuery UI** verfügbar sind.

### 2.6.4.8.1 UI Widget: Icon (jQuery UI)

Mit dem *Icon*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Symbole (Icons) der Bibliothek *jQuery UI* verwenden.

#### 2.6.4.8.1.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Icon Name</b>	<code>iconName</code>	Name des gewünschten Icons aus der Bibliothek <i>jQuery UI</i> (siehe <a href="https://api.jqueryui.com/theming/icons/">https://api.jqueryui.com/theming/icons/</a> ) ohne Präfix <code>ui-icon-</code>  <i>Beispiel:</i> Um das Symbol  <code>ui-icon-alert</code> einzubinden, in <b>Icon Name</b> lediglich <code>alert</code> hinterlegen.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Wrapped</b>	wrapped	Wenn der Wert ein wahrer JavaScript-Ausdruck ist, wird das Icon mit einem <code>&lt;span&gt;</code> -Element umgeben, das gesondert formatiert werden kann.

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Auf dieser Seite:

#### Siehe auch:

- [UI Widget: DatePicker \(jQuery UI\)](#)
- [UI Widget: Dialog \(jQuery UI\)](#)
- [UI Widget: Tabs \(jQuery UI\)](#)
- [UI Widget: Accordion \(jQuery UI\)](#)
- [UI-Widgets einfügen und verwalten](#)

### 2.6.4.8.2 UI Widget: DatePicker (jQuery UI)


Mit dem *DatePicker*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* ein Datumswähler bzw. ein Kalender auf Basis der Bibliothek *jQuery UI* einbinden.

#### 2.6.4.8.2.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Value</b>	value	Statischer Wert für das Datum oder Binding, z.B. <code>{{date}}</code>



Bezeichnung	JSON-Eigenschaft	Erläuterung
Region	region	<p>Kürzel, welches die Region zur Internationalisierung des DatePicker-Widgets festlegt</p> <p><i>Beispiel: de</i></p> <div style="border: 1px solid #add8e6; padding: 10px; margin: 10px 0;"> <p>Beachten Sie, dass hierfür die entsprechenden Internationalisierungsdaten per JavaScript bereitgestellt werden müssen. Standard ist en (US-Formate).</p> <p>Einige Standard-Regionen werden vom X4 Web Designer mitgeliefert, beispielsweise de für Deutsch über <code>/X4/httpstarter/ReST/X4WebResources/lib/jquery-ui/i18n/datepicker-de.js</code>.</p> <p>Weitere Regionen sind unter <a href="https://github.com/jquery/jquery-ui/tree/master/ui/i18n">https://github.com/jquery/jquery-ui/tree/master/ui/i18n</a> verfügbar.</p> </div> <div style="border: 1px solid #ffcc00; padding: 10px; margin: 10px 0;"> <p> Da für die deutsche Internationalisierungsdatei (unter <code>/X4/httpstarter/ReST/X4WebResources/lib/jquery-ui/i18n/datepicker-de.js</code>) Abhängigkeiten zu jQuery UI in der Standard-Widget-Collection bestehen, ist es erforderlich, diese Abhängigkeiten synchron zu initialisieren und die Reihenfolge festzulegen, siehe <a href="#">Abhängigkeiten für UI-Elemente definieren</a>:</p> <ol style="list-style-type: none"> <li>1. Widget-Collection <code>...main.widgets.json</code> erhält Position 1.</li> <li>2. JavaScript-Abhängigkeit <code>...datepicker-de.js</code> erhält Position 2.</li> </ol> </div>
Model Date Format	modelDateFormat	<p>Datumsformat, das für den Datumswert im Modell/Binding verwendet werden soll.</p> <p>Siehe <a href="https://api.jqueryui.com/datepicker/#utility-formatDate">https://api.jqueryui.com/datepicker/#utility-formatDate</a></p>
Options : Date Format	options.dateFormat	<p>Datumsformat für die Anzeige im Textfeld. Standard: Definiert durch Region</p> <p>Siehe <a href="https://api.jqueryui.com/datepicker/#utility-formatDate">https://api.jqueryui.com/datepicker/#utility-formatDate</a></p>
Options : Change Month	changeMonth	<p>Legt fest, ob ein Dropdown-Menü angezeigt werden soll, um den Monat zu ändern.</p> <p><b>Mögliche Werte:</b> <code>true</code> / <code>false</code> (Standard)</p>
Options : Change Year	changeYear	<p>Legt fest, ob ein Dropdown-Menü angezeigt werden soll, um das Jahr zu ändern.</p> <p><b>Mögliche Werte:</b> <code>true</code> / <code>false</code> (Standard)</p>
Options : Show week	showWeek	<p>Legt fest, ob die Kalenderwoche angezeigt wird.</p> <p><b>Mögliche Werte:</b> <code>true</code> / <code>false</code> (Standard)</p>

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Options : First Day</b>	<code>firstDay</code>	Definiert den ersten Tag der Woche.  <b>Mögliche Werte:</b> Ganzzahl (Standard: 0 (Sonntag))

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### 2.6.4.8.2.2 Widget-API

Für dieses UI-Widget ist zudem eine JavaScript-API verfügbar. Diese stellt folgende Methoden zur Verfügung.

Methode	Erläuterung	Beispiel
<b>open</b>	Öffnet den Datumswähler	<code>&lt;vc&gt;.getAPI().open()</code>
<b>close</b>	Schließt den Datumswähler	<code>&lt;vc&gt;.getAPI().close()</code>

**Auf dieser Seite:**

**Siehe auch:**

- [UI Widget: Icon \(jQuery UI\)](#)
- [UI Widget: Dialog \(jQuery UI\)](#)
- [UI Widget: Tabs \(jQuery UI\)](#)
- [UI Widget: Accordion \(jQuery UI\)](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.8.3 UI Widget: Dialog (jQuery UI)

Mit dem *Dialog*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* ein Dialog (Pop-up-Dialog) auf Basis der Bibliothek *jQuery UI* erzeugen.

##### 2.6.4.8.3.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Title</b>	<code>title</code>	Fenstertitel des Pop-up-Dialogs
<b>Auto Open</b>	<code>options.autoOpen</code>	Wenn der Wert ein wahrer JavaScript-Ausdruck ist, wird der Dialog automatisch bei der UI-Initialisierung geöffnet.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Modal</b>	<code>options.modal</code>	<p>Verhalten des Dialogs, während der Dialog geöffnet ist</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>true</code>: Alle Eingaben außerhalb des Dialogs sperren</li> <li>• <code>false</code>: Eingaben außerhalb des Dialogs weiterhin ermöglichen (Standard)</li> </ul>
<b>Resizable</b>	<code>options.resizable</code>	<p>Nutzer können die Größe des Dialogs durch Ziehen an der unteren rechten Ecke anpassen</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>true</code>: Größenänderung des Dialogs zulassen</li> <li>• <code>false</code>: Größenänderung des Dialogs nicht zulassen (Standard)</li> </ul>
<b>Draggable</b>	<code>options.draggable</code>	<p>Nutzer können den Dialog durch Ziehen an der Titelleiste verschieben</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>true</code>: Verschieben des Dialogs zulassen</li> <li>• <code>false</code>: Verschieben des Dialogs nicht zulassen (Standard)</li> </ul>
<b>Width</b>	<code>options.width</code>	Breite des Dialogfensters in Pixel
<b>Height (only Pixels)</b>	<code>options.height</code>	Höhe des Dialogfensters in Pixel
<b>Opened</b>	<code>_event.dialog-opened</code>	Optionales Property-Binding, das ausgeführt wird, wenn der Dialog geöffnet wurde, z.B. <code>{{#controller.initDialogData}}</code>
<b>Closed</b>	<code>_event.dialog-closed</code>	Optionales Property-Binding, das ausgeführt wird, wenn der Dialog geschlossen wurde, z.B. <code>{{#controller.clearDialogData}}</code>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### 2.6.4.8.3.2 Datenstruktur des Dialog-Widgets

Zusätzlich zu den oben genannten Widget-spezifischen und allgemeinen Eigenschaften besitzt das *Dialog*-Widget eine vordefinierte Datenstruktur mit folgenden Elementen (UI-Kontexte):

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Dialog Contents</b>	contents	Innerhalb dieses Arrays können Sie beliebig viele UI-Widgets einfügen, die innerhalb des Dialogs angezeigt werden sollen, beispielsweise Buttons.
<b>Inline Contents</b>	inlineContents	Innerhalb dieses Arrays können Sie beliebig viele UI-Widgets einfügen, die außerhalb des Dialogs angezeigt werden.

#### 2.6.4.8.3.3 Widget-API

Für dieses UI-Widget ist zudem eine JavaScript-API verfügbar. Diese stellt folgende Methoden zur Verfügung.

Methode	Erläuterung	Beispiel
<b>open</b>	Öffnet den Dialog	<code>&lt;vc&gt;.getAPI().open()</code>
<b>close</b>	Schließt den Dialog	<code>&lt;vc&gt;.getAPI().close()</code>
<b>center</b>	Zentriert den Dialog innerhalb des Dokuments	<code>&lt;vc&gt;.getAPI().center()</code>
<b>isVisible</b>	Prüft ob der Dialog angezeigt wird  <b>Rückgabewert:</b> true wenn der Dialog angezeigt wird, andernfalls false	<code>&lt;vc&gt;.getAPI().isVisible(): boolean</code>

#### 2.6.4.8.3.4 Widget-Events

Das Laden, Speichern und Löschen der Formulardaten kann durch folgende Events ausgelöst werden.

Event	Erläuterung	Richtung
dialog-open	Öffnen des Dialogs	eingehend
dialog-close	Schließen des Dialogs	eingehend
dialog-opened	Wird ausgelöst, wenn der Dialog geöffnet wurde (nicht wenn er schon offen initialisiert wurde)	ausgehend
dialog-closed	Wird ausgelöst, wenn der Dialog geschlossen wurde	ausgehend

**Auf dieser Seite:**

**Siehe auch:**

- [UI Widget: Icon \(jQuery UI\)](#)
- [UI Widget: Datepicker \(jQuery UI\)](#)
- [UI Widget: Tabs \(jQuery UI\)](#)
- [UI Widget: Accordion \(jQuery UI\)](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.8.4 UI Widget: Tabs (jQuery UI)

Mit dem *Tabs*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* eine Registerkarten-Navigation (Tabs) auf Basis der Bibliothek *jQuery UI* erstellen.

##### 2.6.4.8.4.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Height Style</b>	<code>options.heightStyle</code>	<p>Darstellungsmodus für unterschiedlich hohe Registerkarten-Inhalte</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>content</code>: Jede Registerkarte besitzt je nach Inhalt eine individuelle Höhe (Standard)</li> <li>• <code>auto</code>: Es wird für alle Registerkarten die Höhe der Höchsten verwendet</li> <li>• <code>fill</code>: Die Höhe orientiert sich an der verfügbaren Höhe, die das übergeordnete Element bereitstellt</li> </ul>
<b>Collapsible</b>	<code>options.collapsible</code>	<p>Kompakter Darstellungsmodus des Registerkartenbereichs ermöglichen</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>true</code>: Registerkarteninhalte können durch Klick auf die aktive Registerkarte ausgeblendet werden</li> <li>• <code>false</code>: Registerkarteninhalte werden immer angezeigt (Standard)</li> </ul>
<b>Active Panel Index</b>	<code>options.active</code>	<p>Standardmäßig aktive Registerkarte, definiert anhand ihrer Index-Nummer innerhalb der Baumstruktur (Zählung beginnt bei 0)</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• Beliebige gültige Ganzzahl (z.B. 0, 1, ... n-1 Anzahl Registerkarten)</li> <li>• 0: Erste Registerkarte standardmäßig anzeigen (Standard)</li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### 2.6.4.8.4.2 Datenstruktur des Tabs-Widgets

Zusätzlich zu den oben genannten Widget-spezifischen und allgemeinen Eigenschaften besitzt das *Tabs*-Widget eine vordefinierte Datenstruktur, die innerhalb des Strukturelements `items` beliebig viele Registerkarten-Elemente jeweils mit folgender Struktur enthalten kann:

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Title</b>	<code>title</code>	Angezeigter Titel der Registerkarte
(beliebige Widgets)	<code>contents</code>	Innerhalb dieses Arrays können Sie beliebig viele UI-Widgets einfügen, die dann innerhalb der Registerkarte angezeigt werden.
(weitere allgemeine Widget-Eigenschaften)	...	siehe Abschnitt <i>Allgemeine Widget-Eigenschaften</i>  <div style="border: 1px solid #add8e6; padding: 10px; margin-top: 10px;"> <p>Um dynamisch für jeden Eintrag eines Arrays im Daten-Modell eine Registerkarte zu erzeugen, können Sie in der Widget-Eigenschaft <i>Data Context</i> das entsprechende Modell referenzieren, z.B. <code>#root.myDataItems</code>. Modell-Bindings innerhalb dieser Registerkarten – auch die <i>Title</i>-Eigenschaft – beziehen sich dann auf den Modell-Kontext des jeweiligen Array-Eintrags.</p> </div>

**Auf dieser Seite:**

**Siehe auch:**

- [UI Widget: Icon \(jQuery UI\)](#)
- [UI Widget: Datepicker \(jQuery UI\)](#)
- [UI Widget: Dialog \(jQuery UI\)](#)
- [UI Widget: Accordion \(jQuery UI\)](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.8.5 UI Widget: Accordion (jQuery UI)

Mit dem *Accordion*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* eine vertikale Akkordeon-Navigation auf Basis der Bibliothek *jQuery UI* erstellen.

##### 2.6.4.8.5.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Height Style</b>	heightStyle	<p>Darstellungsmodus für unterschiedlich hohe Akkordeon-Elemente</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>content</code>: Jedes Akkordeon-Element besitzt je nach Inhalt eine individuelle Höhe (Standard)</li> <li>• <code>auto</code>: Die Höhe aller Akkordeon-Elemente richtet sich nach der Höhe des höchsten.</li> <li>• <code>fill</code>: Die Höhe orientiert sich an der verfügbaren Höhe, die das übergeordnete Element bereitstellt.</li> </ul>
<b>Collapsible</b>	collapsible	<p>Kompakten Darstellungsmodus des Akkordeon-Elements ermöglichen</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>true</code>: Akkordeon-Element erst beim Anklicken ausklappen</li> <li>• <code>false</code>: Es ist immer ein Akkordeon-Element ausgeklappt (Standard)</li> </ul>
<b>Active Panel Index</b>	active	<p>Standardmäßig aktives Akkordeon-Element, definiert anhand seiner Index-Nummer innerhalb der Baumstruktur (Zählung beginnt bei 0)</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• Beliebige gültige Ganzzahl (0, 1, ... n-1 Anzahl Akkordeon-Elemente)</li> <li>• 0: Erstes Akkordeon-Element standardmäßig aufgeklappt anzeigen</li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### 2.6.4.8.5.2 Datenstruktur des Accordion-Widgets

Zusätzlich zu den oben genannten Widget-spezifischen und allgemeinen Eigenschaften besitzt das *Accordion*-Widget eine vordefinierte Datenstruktur, die innerhalb des Strukturelements `items` beliebig viele Registerkarten-Elemente jeweils mit folgender Struktur enthalten kann:

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Title</b>	title	Angezeigter Titel des Akkordeon-Elements

Bezeichnung	JSON-Eigenschaft	Erläuterung
(beliebige Widgets)	contents	Innerhalb dieses Arrays können Sie beliebig viele UI-Widgets einfügen, die dann innerhalb des Akkordeon-Elements angezeigt werden.
(weitere allgemeine Widget-Eigenschaften)	...	Siehe Abschnitt <i>Allgemeine Widget-Eigenschaften</i>

- ✓ **Tipp:** In der Baumstruktur des grafischen *UI-Editors* auf **+ Add Item** klicken, um ein neues Akkordeon-Element einzufügen.

**Auf dieser Seite:**

**Siehe auch:**

- [UI Widget: Icon \(jQuery UI\)](#)
- [UI Widget: Datepicker \(jQuery UI\)](#)
- [UI Widget: Tabs \(jQuery UI\)](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.8.6 UI Widget: Tooltip Panel (jQuery UI)

Mit dem *Tooltip Panel*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Tooltips für Oberflächen-Elemente auf Basis der Bibliothek *jQuery UI* definieren.

Für alle Widgets im *Tooltip Panel* wird standardmäßig das `title`-Attribut als Tooltip angezeigt.

Alternative lässt sich der Tooltip auch über die Eigenschaft `content` definieren.

##### 2.6.4.8.6.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Content</b>	<code>content</code>	String oder UI-Binding, welches ein Element oder einen String zurückgibt. Dem UI-Binding werden als Parameter der Inhalt des entsprechenden <code>title</code> -Attributs ( <code>title</code> ) und das betroffene DOM-Elemente ( <code>target</code> ) übergeben.



Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>enableHtmlContent</b>	<code>enableHtmlContent</code>	<p>Legt fest, ob der Inhalt des Tooltips in HTML-Format oder als Text dargestellt wird.</p> <div style="border: 1px solid orange; padding: 10px; margin: 10px 0;"> <p><b>⚠ Achtung!</b></p> <p>Darf nur aktiviert werden, wenn sichergestellt wird, dass durch den Inhalt keine ungewollte HTML-Injection ermöglicht wird.</p> </div> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>true</code>: Inhalt des Tooltips in HTML-Format anzeigen</li> <li>• <code>false</code>: Inhalt des Tooltips als Text anzeigen</li> </ul>
<b>Options</b>	<code>options</code> <ul style="list-style-type: none"> <li>• <code>show</code></li> <li>• <code>hide</code></li> <li>• <code>track</code></li> <li>• <code>disabled</code></li> <li>• <code>position</code></li> </ul>	<p>Optionen zur Darstellung des Tooltips</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>Animate Show</code>: Einblendung wird animiert (Standard: <code>false</code>)</li> <li>• <code>Animate Hide</code>: Ausblendung wird animiert (Standard: <code>false</code>)</li> <li>• <code>Follow Mouse</code>: Tooltip wird immer neben dem Mauszeiger angezeigt</li> <li>• <code>Disabled</code>: Deaktiviert Tooltips</li> <li>• <code>Position</code>: Position des Tooltips im Verhältnis zum Oberflächen-Element. Für mögliche Werte, siehe <a href="http://api.jqueryui.com/position/">http://api.jqueryui.com/position/</a></li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Auf dieser Seite:

#### Siehe auch:

- [jQuery UI-Widgets](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.8.7 UI Widget: Spinner (jQuery UI)

Mit dem *Spinner*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Eingabefeld mit Buttons zum Hoch- und Runterzählen auf Basis der Bibliothek *jQuery UI* definieren.

*Beispiel:*

❗ Befindet sich der Mauszeiger im Eingabefeld, kann auch eine Eingabe mit den Pfeiltasten aufwärts und abwärts oder mit den Tasten **Bild auf** bzw. **Bild ab** erfolgen.

#### 2.6.4.8.7.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
Value	value	Statischer Wert für das Feld oder Binding, z.B. {{value}}
Options	options <ul style="list-style-type: none"> <li>• min</li> <li>• max</li> <li>• step</li> <li>• page</li> <li>• numberFormat</li> <li>• culture</li> <li>• icons               <ul style="list-style-type: none"> <li>• down</li> <li>• up</li> </ul> </li> </ul>	Konfigurationsoptionen für den Spinner  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• Min. Value: Minimaler angezeigter Wert, z.B. 0</li> <li>• Max. Value: Maximaler angezeigter Wert, z.B. 100</li> <li>• Step Size: Schrittgröße beim Hoch- und Runterzählen, z.B. 2, um in 2-er Schritten hoch und runter zu zählen</li> <li>• Page Size: Anzahl der Schritte, die beim Hoch- und Runterzählen durchgeführt werden</li> <li>• Number Format: Zahlenformat, z.B. n für Dezimalzahlen oder C für Währung Siehe auch <a href="http://api.jqueryui.com/spinner/#option-numberFormat">http://api.jqueryui.com/spinner/#option-numberFormat</a></li> <li>• Culture: Länderkürzel, der beim Parsen und Formatieren des Wertes herangezogen werden soll, z.B. fr Siehe auch <a href="http://api.jqueryui.com/spinner/#option-culture">http://api.jqueryui.com/spinner/#option-culture</a></li> <li>• Icons: Alternative CSS-Klassen für die Buttons zum Hoch und Runterzählen, siehe auch <a href="http://api.jqueryui.com/theming/icons/">http://api.jqueryui.com/theming/icons/</a> <ul style="list-style-type: none"> <li>• Down: alternativer Klassenname</li> <li>• Up: alternativer Klassenname</li> </ul> </li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

**Auf dieser Seite:**

**Siehe auch:**

- [jQuery UI-Widgets](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.8.8 UI Widget: Progressbar (jQuery UI)

Mit dem *Progressbar*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Fortschrittsbalken auf Basis der Bibliothek *jQuery UI* definieren. Ein Fortschrittsbalken nimmt dabei einen Wert und einen maximalen Wert entgegen und zeigt dazu einen prozentualen Status an.

##### 2.6.4.8.8.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Value</b>	value	Statischer Wert für das Feld oder Binding, z.B. {{value}}
<b>Options</b>	options <ul style="list-style-type: none"> <li>• max</li> </ul>	Konfigurationsoptionen für den Fortschrittsbalken  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• Max. Value: Maximaler Wert, z.B. 100</li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

**Auf dieser Seite:**

**Siehe auch:**

- [jQuery UI-Widgets](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.8.9 UI Widget: Slider (jQuery UI)

Mit dem *Slider*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Slider auf Basis der Bibliothek *jQuery UI* definieren.

### 2.6.4.8.9.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Value</b>	value	Statischer Wert für das Feld oder Binding, z.B. {{value}}
<b>Options</b>	options <ul style="list-style-type: none"> <li>• min</li> <li>• max</li> <li>• step</li> <li>• orientation</li> </ul>	Konfigurationsoptionen für den Slider  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• Min. Value: Minimaler Wert der eingestellt werden kann</li> <li>• Max. Value: Maximaler Wert der eingestellt werden kann</li> <li>• Step: Schrittgröße beim Schieben</li> <li>• Orientation: Ausrichtung des Sliders               <ul style="list-style-type: none"> <li>• horizontal: Slider horizontal ausrichten</li> <li>• vertical: Slider vertikal ausrichten</li> </ul> </li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

**Auf dieser Seite:**

**Siehe auch:**

- [jQuery UI-Widgets](#)
- [UI-Widgets einfügen und verwalten](#)

### 2.6.4.8.10 UI Widget: Range Slider (jQuery UI)

Mit dem *Range Slider*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Wertebereiche innerhalb eines Sliders auf Basis der Bibliothek *jQuery UI* definieren.

#### 2.6.4.8.10.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Min Value</b>	minValue	Statischer minimaler Wert für das Feld oder Binding, z.B. {{minvalue}}

Bezeichnung	JSON-Eigenschaft	Erläuterung
Max Value	maxValue	Statischer maximaler Wert für das Feld oder Binding, z.B. {{maxvalue}}
Options	options <ul style="list-style-type: none"> <li>• min</li> <li>• max</li> <li>• step</li> <li>• orientation</li> </ul>	Konfigurationsoptionen für den Slider  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• Min. Value: Minimaler Wert der eingestellt werden kann</li> <li>• Max. Value: Maximaler Wert der eingestellt werden kann</li> <li>• Step: Schrittgröße beim Schieben</li> <li>• Orientation: Ausrichtung des Sliders               <ul style="list-style-type: none"> <li>• horizontal: Slider horizontal ausrichten</li> <li>• vertical: Slider vertikal ausrichten</li> </ul> </li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

**Auf dieser Seite:**

**Siehe auch:**

- [jQuery UI-Widgets](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.8.11 UI Widget: Autocomplete (jQuery UI)

Mit dem *Autocomplete*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* eine Autovervollständigung für Eingabefelder auf Basis der Bibliothek *jQuery UI* definieren.

##### 2.6.4.8.11.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
Source Type	sourceType	Definiert, wie die Eigenschaft Source interpretiert werden soll  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• csv: Komma-separierte Werte</li> <li>• array: JavaScript-Array, das die Werte enthält</li> <li>• url: URL zum Abrufen der Werte</li> </ul>

Bezeichnung	JSON-Eigenschaft	Erläuterung
Source	source	Konfiguration der Werte; Entsprechend dem definierten Typ, als String oder UI-Binding.
Value	value	Wert des Feldes
Options	options <ul style="list-style-type: none"> <li>• disabled</li> <li>• autoFocus</li> <li>• delay</li> <li>• minLength</li> <li>• position</li> </ul>	Optionen zur Konfiguration der Autovervollständigung  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• Disabled: Autovervollständigung aktivieren und deaktivieren</li> <li>• Auto Focus: Autofokus aktivieren und deaktivieren; Mit dem Wert <i>true</i> wird das erste Element automatisch fokussiert, wenn das Menü angezeigt wird.</li> <li>• Search Delay: Verzögerung in Millisekunde für die Initialisierung der Suche nach jedem Tastenschlag</li> <li>• Search Min. Length: Minimale Eingabelänge ab welcher mit der Suche begonnen werden soll</li> <li>• Position: Position der vorgeschlagenen Werte im Verhältnis zum Input-Feld. Für mögliche Werte, siehe <a href="http://api.jqueryui.com/position/">http://api.jqueryui.com/position/</a></li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Auf dieser Seite:

#### Siehe auch:

- [jQuery UI-Widgets](#)
- [UI-Widgets einfügen und verwalten](#)

### 2.6.4.8.12 UI Widget: Menu (jQuery UI)

Mit dem *Menu*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Menüs auf Basis der Bibliothek *jQuery UI* definieren. Dabei können Menüs mit bis zu vier Ebenen definiert werden, wobei jedes Menü mit einem *titel*, *icon* und einer Aktion *onclick* versehen werden kann.

#### 2.6.4.8.12.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Width</b>	<code>width</code>	Breite des Menüs als CSS-Längenangabe
<b>Options</b>	<code>options</code> <ul style="list-style-type: none"> <li>• <code>disabled</code></li> <li>• <code>icons &gt; submenu</code></li> <li>• <code>position</code></li> </ul>	Optionen zur Konfiguration des Menüs  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• <b>Disabled:</b> Menü aktivieren und deaktivieren</li> <li>• <b>Icons:</b> Icon für das Untermenü, siehe auch <a href="http://api.jqueryui.com/theming/icons/">http://api.jqueryui.com/theming/icons/</a></li> <li>• <b>Position:</b> Position des Untermenüs im Verhältnis zum übergeordneten Menü. Für mögliche Werte, siehe <a href="http://api.jqueryui.com/position/">http://api.jqueryui.com/position/</a></li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### 2.6.4.8.12.2 Datenstruktur des Menu-Widgets

Zusätzlich zu den oben genannten Widget-spezifischen und allgemeinen Eigenschaften besitzt das *Menu*-Widget eine vordefinierte Datenstruktur, die innerhalb des Strukturelements `items` beliebig viele Menu-Elemente jeweils mit folgender Struktur enthalten kann:

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Title</b>	<code>title</code>	Angezeigter Titel des Menu-Elements
<b>Icon Name</b>	<code>iconName</code>	Icon-Name
<b>Action</b>	<code>action</code>	Hinterlegte Aktion
<b>Disabled</b>	<code>disabled</code>	Untermenü aktivieren und deaktivieren
(weitere allgemeine Widget-Eigenschaften)	...	Siehe Abschnitt <i>Allgemeine Widget-Eigenschaften</i>

✓ In der Baumstruktur des grafischen *UI-Editors* auf **+ Add Item** klicken, um ein neues Menu-Element einzufügen.

**Auf dieser Seite:****Siehe auch:**

- [jQuery UI-Widgets](#)
- [UI-Widgets einfügen und verwalten](#)

### 2.6.4.9 Layout-Widgets

Welche Layout-Widgets im *UI-Editor* des *X4 Web Designers* standardmäßig in der Widget-Palette unter **Layout** verfügbar sind.

#### 2.6.4.9.1 UI Widget: Simple Panel

Mit dem *Simple Panel*-Widget lassen sich im *UI-Editor* des *X4 Web Designers*, je nach Widget-Modus, ein einfaches Block- oder Inline-Element zur Strukturierung der Seite definieren.

- ✓ *Tipp:* Innerhalb dieses Elements können Sie beliebig viele UI-Widgets einfügen, beispielsweise Buttons.

##### 2.6.4.9.1.1 Individuelle Widget-Eigenschaften

Dieses Widget besitzt keine individuellen Eigenschaften, die Sie im **Properties**-Tab des *UI-Editors* unter **Custom Properties** bearbeiten könnten.

- ✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

- ✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

**Auf dieser Seite****Siehe auch:**

- [UI Widget: Split Panel](#)
- [UI Widget: Table](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.9.2 UI Widget: Split Panel

Mit dem *Split Panel*-Widget lassen sich Layout-Bereiche im *UI-Editor* des *X4 Web Designers* definieren, die nebeneinander oder untereinander angeordnet werden. Das Widget definiert drei Bereiche. Die äußeren Bereiche besitzen eine festgelegte Größe (z.B. für Header-/ Navigationsbereiche); Der mittlere Bereich nimmt den verbleibenden Raum ein.



### 2.6.4.9.2.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Vertical</b>	<code>vertical</code>	Legt fest, ob die Panels untereinander ( <code>true</code> ) oder nebeneinander ( <code>false</code> / Standard) angeordnet werden
<b>First Panel Size</b>	<code>firstPanelSize</code>	Größe des oberen/linken Panels als CSS-Angabe  Auch relative Angaben sind möglich, z.B. 20%.
<b>Last Panel Size</b>	<code>lastPanelSize</code>	Größe des unteren/rechten Panels als CSS-Angabe  Auch relative Angaben sind möglich, z.B. 20%.
<b>First Panel Resizable</b>	<code>firstPanelResizable</code>	Definiert, ob neben dem ersten Panel ein Slider angezeigt wird ( <code>true</code> ), um die Größe des Panels mit der Maus verändern zu können. Mit Hilfe eines UI-Bindings kann diese Eigenschaft auch dynamisch aktiviert/deaktiviert werden.
<b>Last Panel Resizable</b>	<code>lastPanelResizable</code>	Definiert, ob neben dem letzten Panel ein Slider angezeigt wird ( <code>true</code> ), um die Größe des Panels mit der Maus verändern zu können. Mit Hilfe eines UI-Bindings kann diese Eigenschaft auch dynamisch aktiviert/deaktiviert werden.

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### 2.6.4.9.2.2 Datenstruktur des *Split Panel*-Widgets

Zusätzlich zu den oben genannten Widget-spezifischen und allgemeinen Eigenschaften besitzt das *Split Panel*-Widget eine vordefinierte Datenstruktur mit folgenden Elementen:

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>First Panel</b>	<code>firstPanel</code>	Innerhalb dieses Bereichs für das erste Panel lassen sich beliebig viele UI-Widgets einfügen.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Main Panel</b>	contents	Innerhalb dieses Bereichs für die Hauptinhalte lassen sich beliebig viele UI-Widgets einfügen.
<b>Last Panel</b>	lastPanel	Innerhalb dieses Bereichs für das letzte Panel lassen sich beliebig viele UI-Widgets einfügen.

#### 2.6.4.9.2.3 Widget-API

Für dieses UI-Widget ist zudem eine JavaScript-API verfügbar. Diese stellt folgende Methoden zur Verfügung.

Methode	Erläuterung	Rückgabewert
<b>firstPanelSize</b>	Ändert (neuer Wert als Parameter) oder liest (kein Parameter) die Größe des ersten Panels.  <vc>.getAPI().firstPanelSize([size: string]): string	Größe des ersten Panels
<b>lastPanelSize</b>	Ändert (neuer Wert als Parameter) oder liest (kein Parameter) die Größe des letzten Panels.  <vc>.getAPI().firstPanelSize([size: string]): string	Größe des letzten Panels
<b>vertical</b>	Ändert (neuer Wert als Parameter) oder liest (kein Parameter) ob die Panels nebeneinander (false) oder untereinander (true) dargestellt werden sollen.  <vc>.getAPI().vertical([vertical: boolean]): boolean	true, wenn die Panels untereinander dargestellt werden, andernfalls false
<b>firstPanelResizable</b>	Aktiviert/Deaktiviert den Slider für das erste Panel (boolescher Wert als Parameter), oder prüft ob dieser aktiv ist (kein Parameter).  <vc>.getAPI().firstPanelResizable([enabled: boolean]): boolean	Information, ob der Slider für das erste Panel aktiviert ist
<b>lastPanelResizable</b>	Aktiviert/Deaktiviert den Slider für das letzte Panel (boolescher Wert als Parameter), oder prüft ob dieser aktiv ist (kein Parameter).  <vc>.getAPI().lastPanelResizable([enabled: boolean]): boolean	Information, ob der Slider für das letzte Panel aktiviert ist

Methode	Erläuterung	Rückgabewert
<b>fitFirstPanelSize</b>	Passt die Größe des ersten Panels an dessen Inhalt an.  <vc>.getAPI().fitFirstPanelSize() ( )	-
<b>fitLastPanelSize</b>	Passt die Größe des unteren/ rechten Panels an dessen Inhalt an.  <vc>.getAPI().fitLastPanelSize()	-

#### Auf dieser Seite

#### Siehe auch:

- [UI Widget: Simple Panel](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.9.3 UI Widget: Table

Mit dem *Table*-Widget lassen sich (Layout-)Tabellen im *UI-Editor* des *X4 Web Designers* definieren.

##### 2.6.4.9.3.1 Individuelle Widget-Eigenschaften

Dieses Widget besitzt keine individuellen Eigenschaften, die Sie im **Properties**-Tab des *UI-Editors* unter **Custom Properties** bearbeiten könnten.

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

##### 2.6.4.9.3.2 Datenstruktur des Table-Widgets

Zusätzlich zu den oben genannten Widget-spezifischen und allgemeinen Eigenschaften besitzt das *Table*-Widget eine vordefinierte Datenstruktur, welche die Definition von Zeilen-Elementen (*rows*) und darin definierten Zellen-Elementen (*cells*) ermöglicht.

❗ Die Zellen- und Inhalts-Elemente besitzen keine individuellen Eigenschaften, die Sie im **Properties**-Tab des *UI-Editors* unter **Custom Properties** bearbeiten könnten. Sie können jedoch die allgemeinen Widget-Eigenschaften verwendet werden, z.B. zum Styling über DOM-Hooks.

Bezeichnung	JSON-Eigenschaft	Erläuterung
Cell 0-n	rows	Innerhalb dieses Arrays für ein Zeilen-Elemente (entspricht jeweils einer Zeile) können Sie beliebig viele Zell-Elemente mit Klick auf <b>+ Add Item</b> in der Baumstruktur einfügen.
Row 0-n	cells	Innerhalb dieses Arrays für ein Inhalts-Element können Sie beliebig viele UI-Widgets einfügen, beispielsweise Buttons.

#### Auf dieser Seite

#### Siehe auch:

- [UI Widget: Simple Panel](#)
- [UI Widget: Split Panel](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.9.4 UI Widget: Frame

Mit dem *Frame*-Widget lassen sich anderweitig erzeugte Oberflächen-Elemente im *UI-Editor* des *X4 Web Designers* einbinden.

- ✓ *Tipp:* Um komplett unabhängig erzeugte Oberflächen (z.B. durch Drittanwendungen bereitgestellte Web-Seiten) als echtes HTML-iFrame einzubinden, können Sie das Widget *Native iFrame* verwenden, siehe [UI Widget: Native iFrame](#).

##### 2.6.4.9.4.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
UI Configuration URL	url	URL eines JSON-Dokuments, das die UI-Konfiguration/UI-Properties für den Frame enthält

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Init-Properties Model Type</b>	<code>initPropModelType</code>	<p>Model-Typ im Kontext der View, dessen Daten als Eigenschaften (Properties) für den Frame verwendet werden sollen</p> <p>Sie können auch nur ein Teil des Models verwenden, siehe <code>initPropModelContext</code>.</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• "prop"</li> <li>• "model"</li> <li>• "data"</li> <li>• "state"</li> <li>• "localState"</li> <li>• "none" (Standard)</li> </ul>
<b>Init-Properties Model Context</b>	<code>initPropModelContext</code>	<p>Kontext, der für die Properties des Frames verwendet werden soll</p> <p>Dieser wurde innerhalb des Models durch <code>initPropModelType</code> definiert.</p> <p><i>Standardwert:</i> "#self"</p>
<b>Model Type</b>	<code>modelType</code>	<p>Model, welches für den Frame verwendet werden soll</p> <p>Hier sind die gleichen Werte wie für <code>initPropModelType</code> erlaubt.</p> <p><i>Standardwert:</i> "none": ein neues, leeres Model für den Frame erzeugen</p>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

i Für weiterführende Informationen zum Einbinden von Frames über die JavaScript-API für den *X4 Web Designer*, siehe [Frames \(ui/frames.js\)](#).

**Auf dieser Seite****Siehe auch:**

- API-Dokumentation: Frames (ui/frames.js)
- UI Widget: Simple Panel
- UI Widget: Split Panel
- UI Widget: Table
- UI Widget: Native iFrame
- UI-Widgets einfügen und verwalten

### 2.6.4.9.5 UI Widget: Native iFrame

Mit dem *Native iFrame*-Widget lassen sich komplett unabhängig erzeugte Oberflächen (z.B. durch Drittanwendungen bereitgestellte Web-Seiten) als echtes HTML-iFrame im *UI-Editor* des *X4 Web Designers* einbinden.

#### 2.6.4.9.5.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
URL	url	URL zur iFrame-Ressource  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• Beliebige gültige relative URL, z.B. / X4/httpstarter/ReST/MyWebApp/iFrames/MyiFrame.html</li> <li>• Beliebige gültige absolute URL, z.B. http://localhost:8080/X4/httpstarter/ReST/MyWebApp/iFrames/MyiFrame.html</li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

**Auf dieser Seite****Siehe auch:**

- UI Widget: Simple Panel
- UI Widget: Split Panel
- UI Widget: Table
- UI Widget: Frame
- UI-Widgets einfügen und verwalten

#### 2.6.4.9.6 UI Widget: Tree

Mit dem *Tree*-Widget lassen sich Baumstrukturen mit auf-/zuklappbaren Knoten im *UI-Editor* des *X4 Web Designers* definieren.

##### 2.6.4.9.6.1 Individuelle Widget-Eigenschaften

Dieses Widget besitzt keine individuellen Eigenschaften, die Sie im **Properties**-Tab des *UI-Editors* unter **Custom Properties** bearbeiten könnten.

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

##### 2.6.4.9.6.2 Datenstruktur des Tree-Widgets

Zusätzlich zu den oben genannten Widget-spezifischen und allgemeinen Eigenschaften besitzt das *Tree*-Widget, sowie jeder Knoten des Widgets eine vordefinierte Datenstruktur mit folgenden Elementen:

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Node Contents</b>	contents	Innerhalb dieses Bereichs für die Inhalte des Knotens lassen sich beliebig viele UI-Widgets einfügen.
<b>Sub Nodes</b>	nodes	Innerhalb dieses Bereichs für Kindknoten lassen sich beliebig viele Knoten einfügen.

#### Auf dieser Seite

#### Siehe auch:

- [Layout-Widgets](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.9.7 UI Widget: Navigation Tree

Mit dem *Navigation Tree*-Widget lassen sich Baumstrukturen mit auf- und zuklappbaren Knoten definieren, deren Knoten standardmäßig ein Navigationselement mit Label und/oder Icon enthalten. Auf den Navigationselementen können URLs oder JavaScript-Aktionen definiert werden.

##### 2.6.4.9.7.1 Individuelle Widget-Eigenschaften

Dieses Widget besitzt keine individuellen Eigenschaften, die Sie im **Properties**-Tab des *UI-Editors* unter **Custom Properties** bearbeiten könnten.

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tip:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### 2.6.4.9.7.2 Datenstruktur des Navigation Tree-Widgets

Zusätzlich zu den oben genannten Widget-spezifischen und allgemeinen Eigenschaften besitzt das *Navigation Tree*-Widget, sowie jeder Knoten des Widgets eine vordefinierte Datenstruktur mit folgenden Elementen:

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Node Contents</b>	contents	Innerhalb dieses Bereichs für zusätzliche Inhalte des Knotens lassen sich beliebig viele UI-Widgets einfügen. Diese werden hinter Label und Icon eingefügt.
<b>Sub Nodes</b>	nodes	Innerhalb dieses Bereichs für Kindknoten lassen sich beliebig viele Knoten einfügen.

#### 2.6.4.9.7.3 Individuelle Knoten-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für einen markierten Knoten des Widgets dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Label des Knotens.
<b>Icon URL</b>	iconUrl	Optionale URL, die auf ein Icon verweist.
<b>Link</b>	link	Optionale URL, die bei Klick auf den Knoten aufgerufen werden soll.
<b>Action</b>	action	JavaScript-Aktion als UI-Binding, die bei Klick auf den Knoten aufgerufen werden soll.

#### Auf dieser Seite

#### Siehe auch:

- [Layout-Widgets](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.9.8 UI Widget: Marker

Mit dem *Marker*-Widget lässt sich ein Label mit farblicher Hervorhebung definieren.

##### 2.6.4.9.8.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Value</b>	value	Statischer Wert für das Feld oder Binding, z.B. <code>{{markerText}}</code>



Bezeichnung	JSON-Eigenschaft	Erläuterung
Color	color	Hintergrundfarbe des Markers (optional).
Text Color	textColor	Text-Farbe des Markers (optional).

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### Auf dieser Seite

##### Siehe auch:

- [Layout-Widgets](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.9.9 UI Widget: Tiles

Mit dem *Tiles*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* beliebige Elemente in unterschiedlich großen Kacheln anordnen.

##### 2.6.4.9.9.1 Individuelle Widget-Eigenschaften

Dieses Widget besitzt keine individuellen Eigenschaften, die Sie im **Properties**-Tab des *UI-Editors* unter **Custom Properties** bearbeiten könnten.

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)



##### 2.6.4.9.9.2 Datenstruktur des Tiles-Widgets

Zusätzlich zu den oben genannten Eigenschaften besitzt das *Tiles*-Widget, sowie jeder Knoten des Widgets eine vordefinierte Datenstruktur mit folgenden Elementen:

Bezeichnung	JSON-Eigenschaft	Erläuterung
Slide Contents	slideContainer	Inhalte, die auf der Kachel angezeigt werden.
Slide-Over Contents	slideOverContainer	Inhalte, die beim Bewegen der Maus über die Kachel angezeigt werden.

### 2.6.4.9.9.3 Individuelle Knoten-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für einen markierten Knoten des Widgets dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Title</b>	<code>title</code>	Angezeigter Titel der Kachel. Dieser wird links unten in der Kachel angezeigt.
<b>Icon URL</b>	<code>iconUrl</code>	<p>URL, die auf ein Icon verweist. Dieses wird mittig auf der Kachel angezeigt.</p> <p> <b>Unknown Attachment</b> Bei der Aktivierung von <code>Display Child Widgets</code> ist die Eingabe optional.</p>
<b>Text</b>	<code>text</code>	<p>Beliebiger Text, der auf der zweiten animierten Kachel angezeigt wird.</p> <p> <b>Unknown Attachment</b> Ist keine Animation aktiviert oder <code>Display Child Widgets</code> aktiv, ist der Text optional.</p>
<b>Action</b>	<code>action</code>	JavaScript-Aktion als UI-Binding, die bei Klick auf die Kachel aufgerufen werden soll.
<b>Size</b>	<code>size</code>	<p>Definiert die Größe der Kachel.</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>small</code>: ca. 10x10 em</li> <li>• <code>small-wide</code>: ca. 10x20 em</li> <li>• <code>small-extra-wide</code>: ca. 10x30 em</li> <li>• <code>medium</code>: ca. 20x20 em</li> <li>• <code>medium-wide</code>: ca. 20x30 em</li> <li>• <code>large</code>: ca. 30x30 em</li> </ul>
<b>Animation</b>	<code>animation</code>	<p>Definiert, ob und wie die zweite Folie bei einem <i>Mouse-Over</i> angezeigt wird.</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>disabled</code>: zweite Folie wird nicht angezeigt</li> <li>• <code>slide-up</code>: zweite Folie wird von unten nach oben eingeblendet</li> <li>• <code>slide-down</code>: zweite Folie wird von oben nach unten eingeblendet.</li> <li>• <code>slide-left</code>: zweite Folie wird von rechts nach links eingeblendet.</li> <li>• <code>slide-right</code>: zweite Folie wird von links nach rechts eingeblendet.</li> </ul>

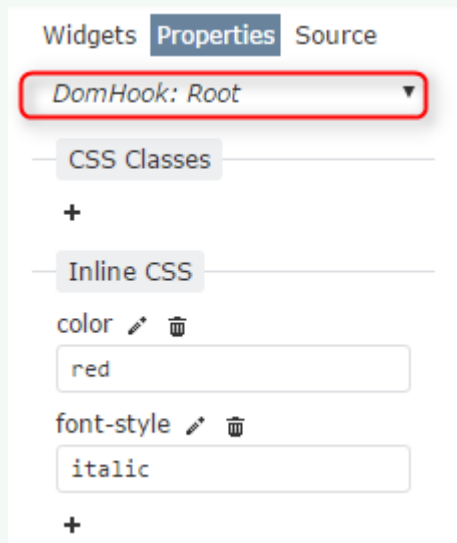
<b>Background-Color</b>	<code>color</code>	<p>Gültige Zeichenkette, die eine Hintergrundfarbe definiert.</p> <p><i>Beispiele:</i></p> <ul style="list-style-type: none"> <li>• <code>red</code></li> <li>• <code>#FF0000</code></li> <li>• <code>rgba(255, 0, 0, 0.7)</code></li> </ul>
<b>Default Font-Color</b>	<code>textColor</code>	<p>Gültige Zeichenkette, die eine Schriftfarbe definiert. Diese Farbe wird als Standardfarbe für die Kachel verwendet</p> <p><i>Beispiele:</i></p> <ul style="list-style-type: none"> <li>• <code>red</code></li> <li>• <code>#FF0000</code></li> <li>• <code>rgba(255, 0, 0, 0.7)</code></li> </ul>
<b>Display Child Widgets</b>	<code>useWidgetContent</code>	Definiert ob die Widgets aus dem Kontext Slide Contents und Slide-Over Contents anstelle der Icon URL und des Textes verwendet werden sollen.
<b>Slide-Over Properties</b>	<b><code>slideOver</code></b>	
<b>Background-Color</b>	<code>slideOver.color</code>	<p>Gültige Zeichenkette, die eine Hintergrundfarbe für die zweite Folie definiert.</p> <p><i>Beispiele:</i></p> <ul style="list-style-type: none"> <li>• <code>red</code></li> <li>• <code>#FF0000</code></li> <li>• <code>rgba(255, 0, 0, 0.7)</code></li> </ul>
<b>Default Font-Color</b>	<code>slideOver.textColor</code>	<p>Gültige Zeichenkette, die eine Schriftfarbe für die zweite Folie definiert. Diese Farbe wird als Standardfarbe für die Kachel verwendet</p> <p><i>Beispiele:</i></p> <ul style="list-style-type: none"> <li>• <code>red</code></li> <li>• <code>#FF0000</code></li> <li>• <code>rgba(255, 0, 0, 0.7)</code></li> </ul>
<b>Mode</b>	<code>slideOver.mode</code>	<p>Definiert wie die zweite Folie angezeigt werden soll.</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>solid</code>: erste Folie wird aus dem sichtbaren Bereich geschoben.</li> <li>• <code>transparent</code>: zweite Folie wird über die erste Folie geschoben. Bei einem Transparentem Hintergrund bleibt die erste Folie sichtbar.</li> </ul>

## Auf dieser Seite

### 2.6.4.9.10 UI Widget: Heading

Mit dem *Heading*-Widget lassen sich Überschriften mit mehreren Ebenen definieren.

- ✓ *Tipp:* Um die Darstellung der Überschriften anzupassen, lassen sich im **Properties**-Tab unter **DOM-Hook: Root** CSS-Klassen oder direkt beliebige CSS-Eigenschaften für das Heading-Element definieren:



#### 2.6.4.9.10.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Value</b>	value	Statischer Wert für die Überschrift oder Binding, z.B. {{heading}}
<b>Level</b>	level	Überschriften-Ebene  <i>Mögliche Werte: Ebene 1–6</i>

- ✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

- ✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### Auf dieser Seite

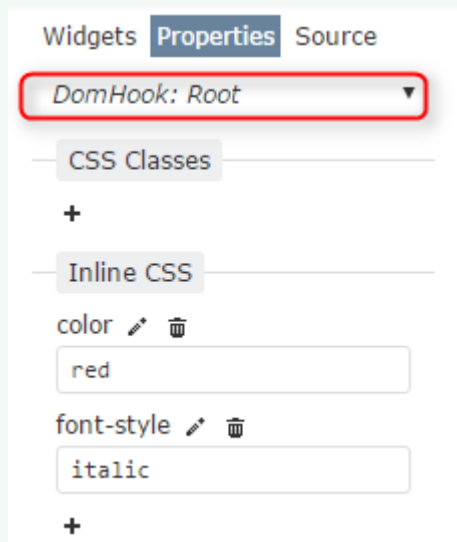
#### Siehe auch:

- [Layout-Widgets](#)
- [UI-Widgets einfügen und verwalten](#)

#### 2.6.4.9.11 UI-Widget: Text

Mit dem *Text*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Textblöcke definieren.

- ✓ *Tipp:* Um die Darstellung des Textes anzupassen, lassen sich im **Properties**-Tab unter **DOM-Hook: Root** CSS-Klassen oder direkt beliebige CSS-Eigenschaften für das Text-Element definieren:



##### 2.6.4.9.11.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
Value	value	Statischer Wert für den Text oder Binding, z.B. {{text}}

- ✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

### 2.6.4.9.11.2 Allgemeine Widget-Eigenschaften

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Widget Mode</b>	<code>_template</code>	Name des zu verwendenden Templates  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• <b>Block</b>: Der Text wird als eigener Absatz dargestellt. (Standard)</li> <li>• <b>Inline</b>: Der Text wird an der eingefügten Stelle dargestellt.</li> </ul> <p><b>i</b> Mehrere nacheinander eingefügte Text-Widgets mit der Eigenschaft <i>Inline</i>, werden nebeneinander als Fließtext angezeigt</p>

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### 2.6.4.10 Activities-Integration-Widgets

Welche Widgets zur Activities-Unterstützung im *UI-Editor* des *X4 Web Designers* standardmäßig in der Widget-Palette unter **Activities Integration** verfügbar sind.

#### 2.6.4.10.1 Activities Integration API

Die Activities Integration API erlaubt den einfachen Zugriff auf Funktionen eines Activities-Portals innerhalb einer Detail-Seite.

##### 2.6.4.10.1.1 Verwendung

Über das globale Objekt `x4.portal.api` kann auf folgende API-Methoden zugegriffen werden.

##### 2.6.4.10.1.2 Lister-API

Methode	Beschreibung	Parameter
<code>updateLister()</code>	Führt einen Reload für den aktuell angezeigten Activities-Lister aus.	keine
<code>getPrimaryKey()</code>	Liefert den Primärschlüssel des selektierten Lister-Eintrags. Wenn kein Lister-Eintrag selektiert ist, wird der Aufruf ignoriert.	keine

##### 2.6.4.10.1.3 Dialog-API

Methode	Beschreibung	Parameter
<code>closeDialog()</code>	Schließt den aktuell geöffneten Activities-Dialog.	keine
<code>updateListerAndCloseDialog()</code>	Führt einen Reload für den aktuell angezeigten Activities-Lister aus und schließt den aktuell geöffneten Activities-Dialog.	keine

Methode	Beschreibung	Parameter
isDialog()	Liefert <code>true</code> , wenn ein Activities-Dialog verfügbar ist, andernfalls <code>false</code> .	keine
updateListerAndCloseDialog(force: <boolean undefiend>)	Führt einen Reload für den aktuell angezeigten Activities-Lister aus und schließt den aktuell geöffneten Activities-Dialog.	<ul style="list-style-type: none"> <li>• <code>true</code>: erzwingt das Schließen des Dialogs, auch wenn ein CloseHandler registriert ist.</li> <li>• <code>false</code>: der CloseHandler wird aufgerufen.</li> </ul>
closeDialog(force: <boolean undefiend>)	Schließt den aktuell geöffneten Activities-Dialog.	<ul style="list-style-type: none"> <li>• <code>true</code>: erzwingt das Schließen des Dialogs, auch wenn ein CloseHandler registriert ist.</li> <li>• <code>false</code>: der CloseHandler wird aufgerufen.</li> </ul>
getDialogId()	Liefert die ID des aktuellen Dialogs.	keine
registerDialogConfirmCloseHandler(callback: <function undefiend>)	Registriert einen CloseHandler, der beim Schließen des Dialogs aufgerufen wird. Die Callback-Funktion muss einen Boolean-Wert zurückgeben und entscheidet somit, ob ein Dialog geschlossen wird oder nicht.	<i>callback</i> : JavaScript-Funktion mit einem Boolean-Wert als Rückgabewert.

#### 2.6.4.10.1.4 Cookie-API

Methode	Beschreibung	Parameter
getLocale()	Liefert den Wert für den aktuell aktiven Locale aus dem Cookie zurück.	keine
getUsername()	Liefert den Anzeigenamen für den aktuell angemeldeten Benutzer aus dem Cookie zurück.	keine
getLogin()	Liefert den Login-Namen für den aktuell angemeldeten Benutzer aus dem Cookie zurück.	keine

#### 2.6.4.10.1.5 Loading Indicator API

Methode	Beschreibung	Parameter
start()	Loading Indicator einblenden und Animation starten.	keine
stop()	Loading Indicator ausblenden und die Animation stoppen. Wenn der Loading Indicator nicht aktiv ist, wird der Aufruf ignoriert.	keine
isRunning()	Gibt <code>true</code> zurück, wenn der Loading Indicator aktiv ist, sonst <code>false</code> .	keine

**Siehe auch:**

- [Listenansicht](#)
- [Dialog](#)

### 2.6.4.10.2 UI-Widget: Activities Button

Mit dem *Activities Button*-Widget lassen sich im *UI-Editor* des *X4 Web Designers* Schaltflächen (Buttons) erstellen, um X4 Activities-spezifische Funktionen wie das Schließen eines Dialogs oder das Neuladen des Listers durchzuführen.

#### 2.6.4.10.2.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Label</b>	label	Angezeigter Beschriftungstext für die Schaltfläche
<b>Action</b>	action	<p>Aktion, die durch die Schaltfläche ausgeführt werden soll</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>Close Dialog</code>: Schließt den aktuell geöffneten Activities-Dialog.</li> <li>• <code>Update Lister</code>: Führt einen Reload für den aktuell angezeigten Activities-Lister aus.</li> <li>• <code>Update Lister &amp; Close Dialog</code>: Führt einen Reload für den aktuell angezeigten Activities-Lister aus und schließt den aktuell geöffneten Activities-Dialog.</li> </ul>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

#### 2.6.4.10.2.2 Widget-API

Für dieses UI-Widget ist zudem eine JavaScript-API verfügbar. Diese stellt folgende Methoden zur Verfügung:

Methode	Erläuterung
<b>closeDialog</b>	<p>Schließt den aktuell geöffneten Activities-Dialog.</p> <p><code>x4.portal.api.closeDialog()</code></p>




Methode	Erläuterung
<b>isDialog</b>	Liefert <code>true</code> , wenn ein Activities-Dialog verfügbar ist, andernfalls <code>false</code> .  <code>x4.portal.api.isDialog()</code>
<b>updateLister</b>	Führt einen Reload für den aktuell angezeigten Activities-Lister aus.  <code>x4.portal.api.updateLister()</code>
<b>updateListerAndCloseDialog</b>	Führt einen Reload für den aktuell angezeigten Activities-Lister aus und schließt den aktuell geöffneten Activities-Dialog.  <code>x4.portal.api.updateListerAndCloseDialog()</code>

### 2.6.4.10.3 UI-Widget: Loading Indicator

Mit dem *Loading Indicator*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* der X4 Activities Loading Indicator integrieren.

#### 2.6.4.10.3.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Delay (ms)</b>	<code>delay</code>	Das Starten der Ladeanzeige um die Angabe (in Millisekunden) verzögern.  <div>  Unknown Attachment </div> Mit der Angabe einer Verzögerung kann die Anzeige des Loading Indicators für kurze Ladevorgänge verhindert werden.
<b>Auto Start</b>	<code>autoStart</code>	Das automatische Starten beim Seitenaufruf steuern.  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li><code>true</code>: Der Loading Indicator wird beim Seitenaufruf automatisch gestartet.</li> <li><code>false</code>: Der Loading Indicator wird beim Seitenaufruf nicht automatisch gestartet. Dieser muss bei Bedarf über die API getriggert werden. (Standard)</li> </ul>



*Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

### 2.6.4.10.3.2 Allgemeine Widget-Eigenschaften

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Widget Mode</b>	<code>_template</code>	<p>Name des zu verwendenden Templates (sofern dies nicht bereits im HTML angegeben wurde)</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• <code>Overlay</code>: Der Loading Indicator überdeckt die gesamte Seite. (Standard)</li> <li>• <code>Inline</code>: Der Loading Indicator wird an der eingefügten Stelle dargestellt.</li> </ul>

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### 2.6.4.10.3.3 Widget-API

Für dieses UI-Widget ist zudem eine JavaScript-API verfügbar. Diese stellt folgende Methoden zur Verfügung:

Methode	Erläuterung
<b>start</b>	<p>Loading Indicator einblenden und Animation starten.</p> <pre>&lt;vc&gt;.getAPI().start()</pre>
<b>stop</b>	<p>Loading Indicator ausblenden und die Animation stoppen. Wenn der Loading Indicator nicht aktiv ist, wird der Aufruf ignoriert.</p> <pre>&lt;vc&gt;.getAPI().stop()</pre>
<b>isRunning</b>	<p>Gibt <code>true</code> zurück, wenn der Loading Indicator aktiv ist, sonst <code>false</code>.</p> <pre>&lt;vc&gt;.getAPI().isRunning()</pre>

## 2.6.4.11 Eigene Widgets erstellen

Wie Sie eigene Widgets als wiederverwendbare Oberflächenelemente zur Bibliothek der UI-Widgets im *X4 Web Designer* hinzufügen


### 2.6.4.11.1 Widget Collections

Alle Quelldateien zu selbst definierten Widgets lassen sich über den *X4 Web Designer* erstellen und werden im Projektordner `<Projekt>/src/main/widgets` abgelegt. Jedes Widget besteht aus einer sogenannten *Widget Collection*, die jeweils standardmäßig folgende vier Dateien enthält:

Typ	Dateiendung	Erläuterung
Widget-Konfiguration	.widgets.json	<p>Konfigurationsdatei zum selbst definierten UI-Widget im JSON-Format</p> <p>In dieser Datei sind der Name und Titel des Widgets anpassbar. Ansonsten enthält die Konfigurationsdatei ein Array mit einer beliebigen Zahl von Widgets, jeweils mit Name und Titel, siehe API-Dokumentation <a href="#">Widgets im UI-Framework</a>.</p> <p>Im Array modes lassen sich verschiedene Modi für das Widget hinterlegen, die sich im UI-Editor in der Widget-Konfiguration auswählen lassen.</p>
CSS-Stylesheet	.css	Enthält alle Widget-spezifischen <i>Cascading Style Sheets</i>
HTML-Vorlage	.html	<p>Beachten Sie:</p> <ul style="list-style-type: none"> <li>• <code>&lt;script&gt;</code>-Elemente lassen sich über das Attribut <code>x4-template="&lt;Vorlagenname&gt;"</code> als Vorlage deklarieren.</li> <li>• Spezifische Attribute wie z.B. <code>x4-value</code>, können Sie mittels Tastenkombination <code>Strg+Leer</code> auswählen und hinzufügen. Diese können Referenzierungen mittels <code>#</code> enthalten, z.B. <code>#prop.label</code>. Für Details siehe API-Dokumentation <a href="#">Framework-spezifische HTML-Strukturen</a>.</li> <li>• Mit dem spezifischen Attribut <code>x4-dom-hook</code> können Sie ein HTML-Element als DOM-Hook-Einstiegspunkt definieren für spätere Ergänzungen, z.B. für Verknüpfungen mit Fachlogik, Formatierungen usw., siehe <a href="#">DomHook-Einträge hinterlegen</a>.</li> </ul>
JavaScript-Code	.js	Enthält den gesamten Widget-spezifischen JavaScript-Code, der zum Rendern des Widgets erforderlich ist, sowie Event-Handler etc., siehe API-Dokumentation <a href="#">JavaScript-API für den X4 Web Designer</a> .

#### 2.6.4.11.2 Selbst definierte UI-Widget mit dem X4 Web Designer erstellen

1. Im Navigator unter `<Projekt>/src/main/widgets` eine neue Widget-Konfigurationsdatei ohne Dateiendung anlegen, siehe [Ordner und Dateien im Navigator anlegen und verwalten](#).  
*Beispiel: MyWidgets*  
Dadurch wird automatisch eine Konfigurationsdatei `.widgets.json` und gleichnamige `.css`-, `.js`-Dateien erstellt.

- html und .js-Dateien in demselben Ordner erstellt (siehe oben).
2. Die Quelldateien per Klick im Navigator öffnen und im jeweiligen Editor die entsprechenden Inhalte für Ihr UI-Widget hinterlegen, siehe [JavaScript-API für den X4 Web Designer](#).
  3. Die URL zur Widgets-Konfigurationsdatei kopieren. Dazu im *Navigator* bei Ihrer .widgets.json-Datei auf  klicken und **Show resource URL** wählen, um die HTTP-ReST-URL zur Datei anzuzeigen.
  4. Im *UI-Editor* des *X4 Web Designers* die entsprechende Oberflächenkonfiguration im .ui.json-Format öffnen und in die **Dependencies**-Ansicht wechseln, siehe [UI-Editor im X4 Web Designer](#).
  5. In der **Dependencies**-Ansicht unter **Widget Collections** die kopierte URL zur benutzerdefinierten Widget-Konfiguration hinzufügen.



*Tip:* Mit der Option Sync werden die verknüpften Ressourcen synchron geladen. Falls Abhängigkeiten untereinander bestehen, kann mit dieser Option sicher gestellt werden, dass die Ressourcen stets in der hier definierten Reihenfolge geladen werden.

6. **Save** klicken und im UI-Editor in die UI-Ansicht wechseln.  
Ihr neu erstelltes UI-Widget steht nun in der Widgets-Palette zur Verfügung.

#### Auf dieser Seite:

#### Siehe auch:

- [Widgets im UI-Editor verwenden](#)
- [Ordner und Dateien im Navigator verwalten](#)
- [JavaScript-API für den X4 Web Designer](#)
- [DomHook-Einträge hinterlegen](#)
- [UI-Editor im X4 Web Designer](#)
- [JavaScript-API für den X4 Web Designer](#)

#### 2.6.4.12 Bing Maps Widget

Mit dem *Bing Maps*-Widget lässt sich im *UI-Editor* des *X4 Web Designers* eine Karte des Microsoft-Kartendienstes Bing Maps einbinden. In einer solchen Karte können unter anderem über eine JavaScript-API Orte markiert und Informationen zu diesen Orten eingeblendet werden.



Um die *Bing Maps API* nutzen zu können, wird ein *Bing Maps API Key* benötigt.



Um das Widget in alten Detailseiten, die vor Version 6.0 erstellt wurden, verwenden zu können, muss die Widget-Collection mit folgender URL in den Dependencies des UI Editors eingetragen werden:  
`/X4/httpstarter/ReST/X4WebResources/maps-widgets/widgets/maps.widgets.json`

### 2.6.4.12.1 Individuelle Widget-Eigenschaften

Im **Properties**-Tab des *UI-Editors* können Sie für ein markiertes Widget dessen individuelle Eigenschaften unter **Custom Properties** bearbeiten.

Bezeichnung	JSON-Eigenschaft	Erläuterung
<b>Bing Maps Key</b>	bingMapsKey	Enthält einen Bing Maps API Key
<b>Lang</b>	lang	Legt die Sprache für Beschriftungen in der Karte fest.  <b>Mögliche Werte:</b> Sprachkürzel, z. B. en
<b>MKT</b>	mkt	Legt den Standort des Anwenders fest.  <b>Mögliche Werte:</b> Kürzel einer Sprachvariante, z. B. en-US

### 2.6.4.12.2 Widget-API

Methode	Beschreibung
getMap	Liefert die native JavaScript-API der <i>Bing Maps</i> Karte ( <a href="https://msdn.microsoft.com/en-us/library/mt712547.aspx">https://msdn.microsoft.com/en-us/library/mt712547.aspx</a> ). <code>&lt;vc&gt;.getAPI().getMap(): &lt;BingMap&gt;</code>
getMarkers	Liefert alle Marker ( <code>Microsoft.Maps.Pushpin</code> ), die über die Widget-API erzeugt wurden ( <a href="https://msdn.microsoft.com/en-us/library/mt712674.aspx">https://msdn.microsoft.com/en-us/library/mt712674.aspx</a> ).  <code>&lt;vc&gt;.getAPI().getMarkers(): array</code>
createMarker	Erstellt einen Marker ( <code>Microsoft.Maps.Pushpin</code> ) und fügt diesen in die Karte ein.  <code>&lt;vc&gt;.getAPI().createMarker(data: object)</code>  Das übergebene Objekt muss folgende Attribute enthalten: <ul style="list-style-type: none"> <li>• <code>id</code>: <i>Erforderlich</i>. ID des Markers</li> <li>• <code>latitude</code>: <i>Erforderlich</i>. Breitengrad der Koordinate</li> <li>• <code>longitude</code>: <i>Erforderlich</i>. Längengrad der Koordinate</li> <li>• <code>options</code>: Optionen des Bing Maps Pushpin (<a href="https://msdn.microsoft.com/en-us/library/mt712673.aspx">https://msdn.microsoft.com/en-us/library/mt712673.aspx</a>)</li> </ul>
createMarkers	Erstellt mehrere Marker ( <code>Microsoft.Maps.Pushpin</code> ) aus einem Array. Jeder Eintrag des Arrays muss dem Objektformat entsprechen, das auch von <code>createMarker</code> erwartet wird.  <code>&lt;vc&gt;.getAPI().createMarkers(list: array)</code>

Methode	Beschreibung
getInfoWindow	<p>Liefert alle Info-Fenster ( <code>Microsoft.Maps.Infobox</code> ), die über die Widget-API erzeugt wurden (<a href="https://msdn.microsoft.com/en-us/library/mt712643.aspx">https://msdn.microsoft.com/en-us/library/mt712643.aspx</a>).</p> <pre>&lt;vc&gt;.getAPI().getInfoWindows(): array</pre>
createInfoWindow	<p>Erstellt ein Info-Fenster ( <code>Microsoft.Maps.Infobox</code> ) für einen bestehenden Marker. Bei Klick auf den Marker wird das Fenster eingeblendet.</p> <pre>&lt;vc&gt;.getAPI().createInfoWindow(config: object)</pre> <p>Das übergebene Objekt muss folgende Attribute enthalten:</p> <ul style="list-style-type: none"> <li>• <i>id</i>: <i>Erforderlich</i>. ID des Markers, an den das Info-Fenster gebunden werden soll.</li> <li>• <i>content</i>: <i>Erforderlich</i>. Inhalt, der im Info-Fenster angezeigt werden soll (als HTML).</li> <li>• <i>options</i>: Optionen der Bing Maps Infobox (<a href="https://msdn.microsoft.com/en-us/library/mt712658.aspx">https://msdn.microsoft.com/en-us/library/mt712658.aspx</a>)</li> </ul>
createInfoWindows	<p>Erstellt mehrere Info-Fenster ( <code>Microsoft.Maps.Infobox</code> ) aus einem Array. Jeder Eintrag des Arrays muss dem Objektformat entsprechen, das auch von <code>createInfoWindow</code> erwartet wird.</p> <pre>&lt;vc&gt;.getAPI().createInfoWindows(list: array)</pre>
showInfoWindow	<p>Blendet ein Info-Fenster, das über die Widget-API erstellt wurde, anhand der ID des zugehörigen Markers ein. Wenn als zweiter Parameter nicht <code>false</code> übergeben wird, werden alle anderen Info-Fenster in der Karte geschlossen.</p> <pre>&lt;vc&gt;.getAPI().showInfoWindow(id: string [, closeAllOthers: boolean])</pre>
showAllMarkers	<p>Wählt den sichtbaren Bereich der Karte so, dass alle Marker, die über die Widget-API erstellt wurden, darin enthalten sind.</p> <pre>&lt;vc&gt;.getAPI().showAllMarkers()</pre>

✓ *Tipp:* Für weitere Eigenschaften von UI-Widgets siehe [Allgemeine Widget-Eigenschaften](#).

✓ *Tipp:* Für DomHook-Eigenschaften von UI-Widgets, siehe [Elemente mittels DomHooks anpassen](#)

### 2.6.4.12.3 Events

Folgende Events können im *Bing Maps Widget* ausgelöst werden:

Event	Beschreibung
map-ready	Wird ausgelöst, wenn die Karte fertig geladen und die API bereit zur Verwendung ist.

## 2.6.5 Detailseiten für X4 Activities erstellen

Mit *X4 Activities* haben Sie sich für ein flexibles Framework für die Entwicklung von Web-Anwendungen auf Basis des *X4 Enterprise Service Bus* (ESB) und dessen ReSTful-Webservice-Schnittstelle entschieden. *X4 Activities* kann für die unterschiedlichsten Anwendungsfälle eingesetzt werden, beispielsweise für die Auftrags- und Aufgabenverwaltung, aber auch für Service-Monitoring oder Administration. Je nach Szenario stehen verschiedene Optionen zur Verfügung, wie Sie Detailseiten in Ihrer *X4 Activities-Anwendung* entwickeln und bereitstellen.

### 2.6.5.1 Übersicht der Werkzeuge und Methoden

Sie haben folgende Möglichkeiten, um Benutzeroberflächen (Detailseiten) für *X4 Activities* zu erstellen:

- **UI-Editor im *X4 Web Designer*:** Mit dieser Web-Oberfläche können Sie komplexe Benutzeroberflächen schnell und grafisch per Drag & Drop in zusammenstellen, siehe [UI-Widgets einfügen und verwalten](#). Im *UI-Editor* können Sie Fachlogik zu den einzelnen Elementen hinterlegen (siehe [DomHook-Einträge hinterlegen](#)) und hierfür die bereitgestellte JavaScript-API nutzen, die sich speziell an Entwickler aus dem Java-Umfeld richtet (siehe [Java Script-API für den X4 Web Designer](#)).
- **Statisches HTML und JavaScript mit dem *X4 Web Designer* erzeugen:** Falls Sie für die Oberflächen-Entwicklung bereits Frameworks wie z.B. Angular nutzen möchten, ist dies im *X4 Web Designer* ebenfalls möglich. Definieren Sie dazu die entsprechenden Bibliotheken als Abhängigkeiten, siehe [UI-Editor im X4 Web Designer](#) unter "Dependencies" und verwenden die integrierten JavaScript- und HTML-Quelltexteditoren des *X4 Web Designers*.
- **Mit technischen Prozessen dynamisch erzeugte HTML-Detailseiten:** Mit dem *X4 Designer* können Sie technische Prozesse entwickeln, die als Ergebnis fertige HTML-Seiten (ggf. mit Formularen, CSS-Formatierung und JavaScript) zurückliefern. Diese technischen Prozesse können Sie über die Webservices-Schnittstelle des *X4 ESB* als ReSTful Webservice bereitstellen und diese als Detailseiten in Ihre *X4 Activities-Anwendung* einbinden, siehe [Datasources konfigurieren](#).
- **HTML-Seite von externen Ressourcen einbinden:** Bereits vorhandene HTML-Seiten, die Sie mit Drittwerkzeugen erzeugt und via HTTP(S) bereitgestellt haben, können Sie ebenfalls in Ihre *X4 Activities-Anwendung* als Detailseiten über deren URL einbinden, siehe [Datasources konfigurieren](#).

### 2.6.5.2 Grundlegende Vorgehensweise

Um eine Benutzeroberfläche in Ihrer *X4 Activities-Anwendung* verfügbar zu machen, wie folgt vorgehen:

1. Mit dem Werkzeug Ihrer Wahl eine Benutzeroberfläche erstellen.  
*Beispiel:* Mit dem *UI-Editor* im *X4 Web Designer* grafisch modellieren (siehe [UI-Widgets einfügen und verwalten](#)) oder von JSON-Daten erzeugen und konfigurieren (siehe [Formulare auf Basis von JSON-Daten erzeugen](#)).

2. Die URL zur Benutzeroberfläche kopieren und in der Konfiguration zu Ihrer *X4 Activities*-Anwendung hinterlegen:  
*Beispiel:* Im *UI-Editor* des *X4 Web Designers* auf **Show Result URL** klicken, um die entsprechende URL zu ermitteln.
  - **Detailseite zu einem Listeneintrag:** Im *Datasource Config Editor* in Eigenschaft **Base URL** (`BaseUri`) die URL einfügen und die Konfiguration speichern.
  - **Header-Bereich im Inhaltsbereich zu einem Navigatorknoten:** Im *Navigator Config Editor* im entsprechenden Knoten für Eigenschaft **Content Header URL** (`ContentHeader.url`) die URL einfügen und die Konfiguration speichern.
  - **Inhaltsbereich zu einem Navigatorknoten:** Im *Navigator Config Editor* im entsprechenden Knoten für Eigenschaft **Content URL** (`nodeContentUrl`) die URL einfügen und die Konfiguration speichern.

 Der **Content Type** muss **DETAIL** sein.
3. Vorschau Ihrer Web-Anwendung öffnen und diese testen, siehe [Anwendung mit dem X4 Web Designer testen und bereitstellen](#).


### 2.6.5.3 Tipps zum Arbeiten mit dem UI-Editor

#### 2.6.5.3.1 Schließen von Dialogen, die per Actionbar-Button geöffnet wurden

Per Actionbar-Button geöffnete Dialoge lassen sich am einfachsten über den JavaScript-Befehl `parent.x4p.dialogs.closeDialog(x4.url.parse().getParam('x4pDialogId'))` schließen. Um diesen Befehl als direktes Binding im UI-Editor, beispielsweise im Action-Feld eines Buttons zu verwenden, werden zusätzlich geschweifte Klammern am Anfang und Ende benötigt.

*Beispiel:* {


```
parent.x4p.dialogs.closeDialog(x4.url.parse().getParam('x4pDialogId')) }
```

 Für die vollständige JavaScript-API der Dialoge, siehe auch *X4 Activities Handbuch – JavaScript-API für Dialoge in X4 Activities*.

#### 2.6.5.3.2 Lister aus einem Formular heraus aktualisieren

Über den JavaScript-Befehl `parent.x4p.lister.update()` kann der Lister des Activities-Portals aktualisiert werden. Um diesen Befehl als direktes Binding im UI-Editor zu verwenden, werden zusätzlich geschweifte Klammern am Anfang und Ende benötigt.

*Beispiel:* { `parent.x4p.lister.update()` }

 Für die vollständige JavaScript-API des Activities Listers, siehe auch *X4 Activities Handbuch – Globale JavaScript-Methoden*.



**Siehe auch:**

- [UI-Widgets einfügen und verwalten](#)
- [DomHook-Einträge hinterlegen](#)
- [JavaScript-API für den X4 Web Designer](#)
- [UI-Editor im X4 Web Designer](#)
- [Datasources konfigurieren](#)
- [UI-Widgets einfügen und verwalten](#)
- [Formulare auf Basis von JSON-Daten erzeugen](#)
- [Anwendung mit dem X4 Web Designer testen und bereitstellen](#)
- [JavaScript-API für Dialoge in X4 Activities](#)
- [Globale JavaScript-Methoden](#)

## 2.6.6 UI-Controller definieren

Wie Sie Controller für Ihre Web-Anwendung erstellen, um fachliche Logik zu kapseln. Hierzu bietet *X4 Activities* ab Version 5.0 ein Framework, das die JavaScript-Entwicklung im *X4 Web Designer* speziell auch für Entwickler mit Java-Hintergrund erleichtert.

Code für Controller etc. wird typischerweise in eine externe JavaScript-Datei ausgelagert. Um einen Controller zu erstellen, gehen Sie wie folgt vor:

1. In Ordner `<Projekt>/main/js/` eine neue Datei anlegen und öffnen, siehe [Ordner und Dateien im Navigator verwalten](#).  
*Beispiel:* `MyWebApp/main/js/MyController.js`
2. Im JavaScript-Editor des *X4 Web Designers* die Tastenkombination `Strg+Leer` drücken.

✓ Über die Tastenkombination `Strg+Leer` innerhalb des JavaScript-Editors erhalten Sie eine Vorschlagsliste von aktuell erlaubten Syntaxelementen bzw. Autovervollständigung.

3. Bei Bedarf ein Namespace für den Controller definieren.
4. `<X4 UI Controller>` auswählen und **Enter** drücken. Das Grundgerüst für den UI Controller wird dann automatisch eingefügt.

**Beispiel**


```
x4.Namespace('my.example');


my.example.MyController = x4.Class({

  INIT: function(vc) {
    this.vc = vc;

    this.vc.on('view-ready', function(){
      // do something when view rendering is ready
    });
  },

  PUBLIC: {
    setData: function(data){
      this.vc.getModel().set(data);
    },
    fireAnEvent: function() {
      this.vc.fire('my-custom-event');
    }
  }
});
```

5. Erweiterungen und Anpassungen am JavaScript-Quellcode hinterlegen, siehe [JavaScript-API für den X4 Web Designer](#).
6. **Speichern** klicken.
7. Im Navigator neben Ihrer JavaScript-Datei auf  klicken und **Show result URL** aus dem Kontextmenü wählen und den Wert kopieren.

✓ Der Navigatorbereich auf der linken Seite des Editors ist nicht sichtbar? In der Menüleiste auf  (Show navigation) klicken.

8. Im UI-Editor die neue JavaScript-Ressource verfügbar machen:
  - a. In der Menüleiste im Auswahlfeld **Editor** auf *Dependencies* wechseln.  
Der Abhängigkeiten-Editor wird geöffnet, siehe [UI-Editor im X4 Web Designer](#).
  - b. In **JavaScript Files** auf **+** klicken.
  - c. In **URL** die URL der JavaScript-Datei-Ressource (siehe Schritt 6) einfügen und die Datei speichern.
9. Über die Menüleiste im Auswahlfeld **Editor** auf die *UI*-Ansicht zurück wechseln.
10. Zum entsprechenden Widget in der **Properties**-Registerkarte in Standard-Eigenschaft **Controller** den Namen des Controllers eingeben und speichern.  
*Beispiel: my.example.MyController* zu einem Formular-Widget MyForm zuweisen.
11. Nach Bedarf kann nun in Property-Bindings auf Methoden des Controllers zugegriffen werden.  
*Beispiel: {{#controller.fireAnEvent}}*  
oder *{ this.getController().setData(event.target.value) }*

Der Controller wurde nun erstellt und mit dem entsprechenden UI-Widget (beispielsweise ein Formular) verknüpft.

## 2.6.7 Abhängigkeiten für UI-Elemente definieren

Benutzeroberflächen, die Sie im **UI-Editor** des *X4 Web Designers* modelliert haben, können Abhängigkeiten enthalten, d.h. Referenzen auf JavaScript-Bibliotheken, CSS-Dateien, Vorlagen o.ä. In der **Dependencies**-Ansicht des UI-Editors können Sie diese Abhängigkeiten verwalten.

Um eine Abhängigkeit zu definieren, gehen Sie wie folgt vor:

1. Im UI-Editor die **Dependencies**-Ansicht aufrufen, indem Sie in der Menüleiste bei **Editor** den Eintrag *Dependencies* wählen.
2. Im entsprechenden Bereich auf **+** klicken, um einen neuen Eintrag hinzuzufügen.
  - **Widget Collections:** Ressourcen-URL für eine *Widget Collection*-Datei, siehe [Eigene Widgets erstellen](#)
  - **Template Collections:** Ressourcen-URL für eine HTML-Seite, die Templates enthält (*Template Collection*), siehe API-Dokumentation unter [Framework-spezifische HTML-Strukturen](#)
  - **JavaScript Files:** Ressourcen-URL für ggf. einzubindende JavaScript-Bibliotheksdateien
  - **CSS Files:** Ressourcen-URL für ggf. einzubindende Cascading Stylesheets
3. Falls Abhängigkeiten synchron, also in einer bestimmten Reihenfolge nacheinander, geladen werden sollen:
  - Für alle relevanten Einträge die Option **Sync** aktivieren.
  - Für alle relevanten Einträge in **Position** die gewünschte Reihenfolge hinterlegen, in der die Abhängigkeiten (Widget Collections, Template Collections, JavaScript oder CSS) geladen werden sollen, beginnend bei *1*.
  - Bei Bedarf können Sie zusammengehörige Abhängigkeiten zudem über **Group** anhand einer Gruppen-ID gruppieren.
4. **Save** klicken, um die Änderungen zu speichern.


## 2.7 X4 Activities-Anwendung im X4 Web Designer konfigurieren

Erfahren sie, wie Sie mit dem *X4 Web Designer* im Browser eine neue Web-Anwendung auf Basis von *X4 Activities* einrichten.

### 2.7.1 Datenbank-Konfiguration für Web-Anwendungen

In der Konfigurationsdatei `<Projekt>/WebResources/dbConfig.xml` lassen sich direkt im Web-Designer-Projekt die verwendeten Datenbankeinstellungen wie JNDI-Name und Datenbanktyp konfigurieren. Um die Datei zu editieren, muss der sogenannte **Expert Mode** aktiviert sein. Innerhalb von Wurzel-Element `<WebIDE_DB_Config>` können Sie in Element `<dbConnections>` mit beliebig vielen Kindelementen `<dbConnection .../>` jeweils eine Datenbankverbindung mit folgenden Parametern konfigurieren:

Attribut	Beschreibung
id	Eindeutiger Name der Datenbankverbindung  <i>Beispiel: X4BAM</i>


Attribut	Beschreibung
name	Anzeigename der Datenbank <i>Beispiel: X4 BAM</i>
type	Datenbankverbindungs-Typ <i>Beispiel: JDBC</i>
jndi	URL der zugehörigen JNDI-Ressource <i>Beispiel: java:/X4BAM_DS</i>
vendor	Datenbanktyp bzw. -hersteller <i>Beispiel: postgres, oracle, mssql, elasticsearch</i>  Bitte hierbei die genaue Schreibweise beachten.

#### Beispiel für eine Datenbankkonfiguration

```
<?xml version="1.0" encoding="UTF-8"?>
<WebIDE_DB_Config>
  <dbConnections>
    <dbConnection id="X4BAMConnection" name="X4 BAM" type="JDBC"
      jndi="java:/X4BAM_DS" vendor="mssql"/>
    <dbConnection id="X4Experience" name="X4 Experience" type="JDBC"
      jndi="java:/X4Experience_DS" vendor="postgres"/>
  </dbConnections>
</WebIDE_DB_Config>
```

#### 2.7.1.1 Konfiguration einer Elasticsearch-Datasource

In der Konfigurationsdatei `dbConfig.xml` kann auch Elasticsearch als Datenbankverbindung definiert werden. Aufrufe **Generate Fields** und **Generate Lister** funktionieren dabei wie gewohnt.

 Bei der Definition einer Elasticsearch-Datasource können auch **nestedObjects** als Felder wie folgt beliebig tief referenziert werden: `nestedObject.nestedObject.value`. Bei Arrays wird **immer der/das erste Wert/Object angezeigt**. Die interne ID des Dokuments kann über `_id` referenziert werden.

**Beispiel: Elasticsearch-Datenbankkonfiguration**

```
<?xml version="1.0" encoding="UTF-8"?>
<WebIDE_DB_Config>
  <dbConnections>
    <dbConnection id="X4Elasticsearch" name="X4 Elasticsearch" type="elastic"
protocol="http" host="localhost" port="9200"/>
  </dbConnections>
</WebIDE_DB_Config>
```

## 2.7.2 Application-Config(Main) konfigurieren

Wie Sie beim Erstellen einer neuen Web-Anwendung mit dem *X4 Web Designer* in *main* (entspricht der Application-Config von herkömmlichen *X4 Activities*-Anwendungen) grundlegende Einstellungen hinterlegen

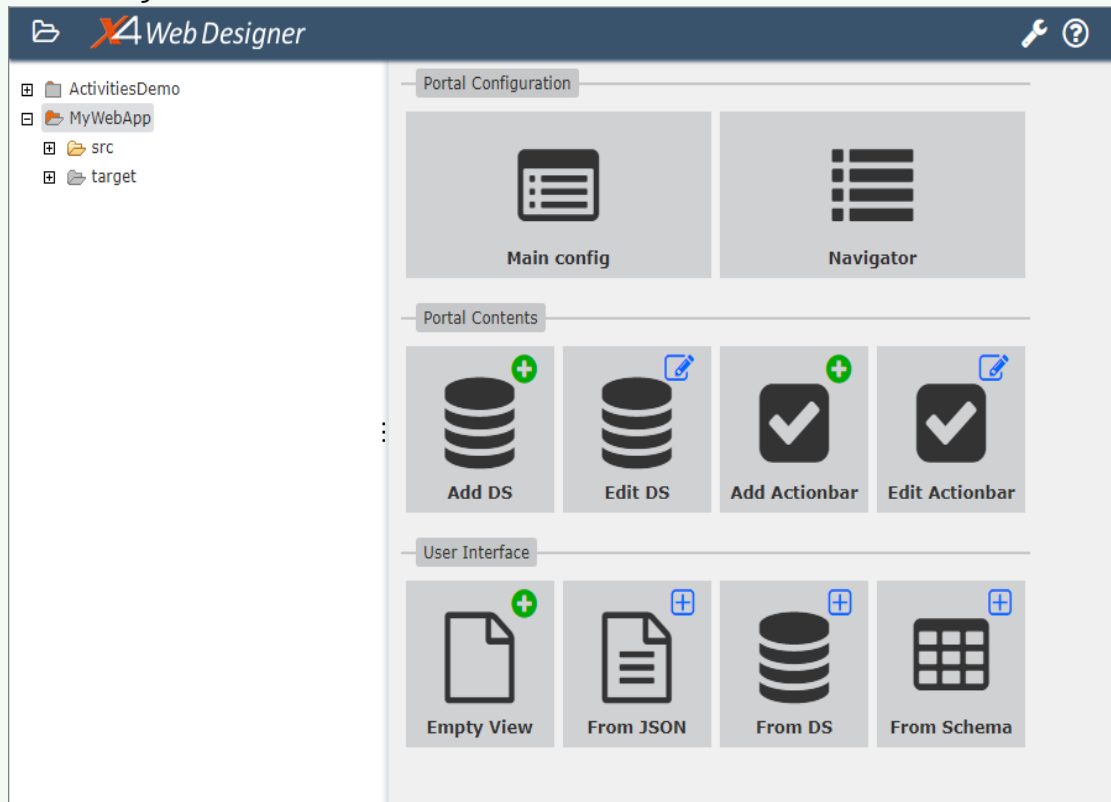
1. Im *X4 Web Designer* zum Ordner `<Projektname>/src/main/portal/main/` navigieren.
2. Eine bestehende Datei vom Typ `.main.x4p.json` öffnen oder mit Klick auf **+** eine neue erstellen, siehe [Ordner und Dateien im Navigator verwalten](#).

Der *Application Config Editor* wird geöffnet.

The screenshot shows the 'Application Config Editor' window in the X4 Web Designer. The title bar includes 'X4 Web Designer', 'Save', 'Editor', 'Application Config', 'Open Preview', and 'Copy Result URL'. The main area is divided into several sections:

- Main Navigator URL**: A text input field.
- Logo URL**: A text input field.
- Applications**: A section with fields for 'App ID', 'Display Name', 'Is Default App' (checkbox), 'Content URL', 'Content Type' (dropdown menu with 'Navigator' selected), and 'Icon URL'. There is a '+' button to add new applications.
- External Links**: A section with fields for 'Display Name', 'URL', 'Icon URL', and 'Link Target' (dropdown menu with 'Default' selected). There is a '+' button to add new external links.
- Help Links**: A section with fields for 'Display Name', 'URL', 'Icon URL', and 'Link Target' (dropdown menu with 'Default' selected). There is a '+' button to add new help links.
- User Links**: A section with fields for 'Display Name', 'URL', 'Icon URL', and 'Link Target' (dropdown menu with 'Default' selected). There is a '+' button to add new user links.

- ✓ Alternativ lässt sich die Application-Config auch direkt über die Projekt-Startseite anlegen.  
Dazu auf **Main config** klicken, einen Namen für die Application-Config eingeben und mit **OK** bestätigen.



3. Konfiguration anpassen und/oder mit + neue Einträge für *Applications* und *ExternalLinks* hinzufügen.

- ✓ **Tipp:** In der Menüleiste können Sie im Auswahlfeld **Editor** zwischen dem grafischen Editor und der JSON-Code-Ansicht wechseln, um die tatsächlich erzeugte JSON-Datenstruktur einzusehen und ggf. zu editieren.  
Der JSON-Editor bietet den vollen Komfort eines JavaScript-Editors mit Syntaxhighlighting, Validierung und Autovervollständigung (mittels Strg+Leertaste).



4. **Save** klicken oder mit **Strg+S** speichern.  
Nach dem Speichern wird die Datei automatisch neu im target-Ordner gebaut.

#### Siehe auch:

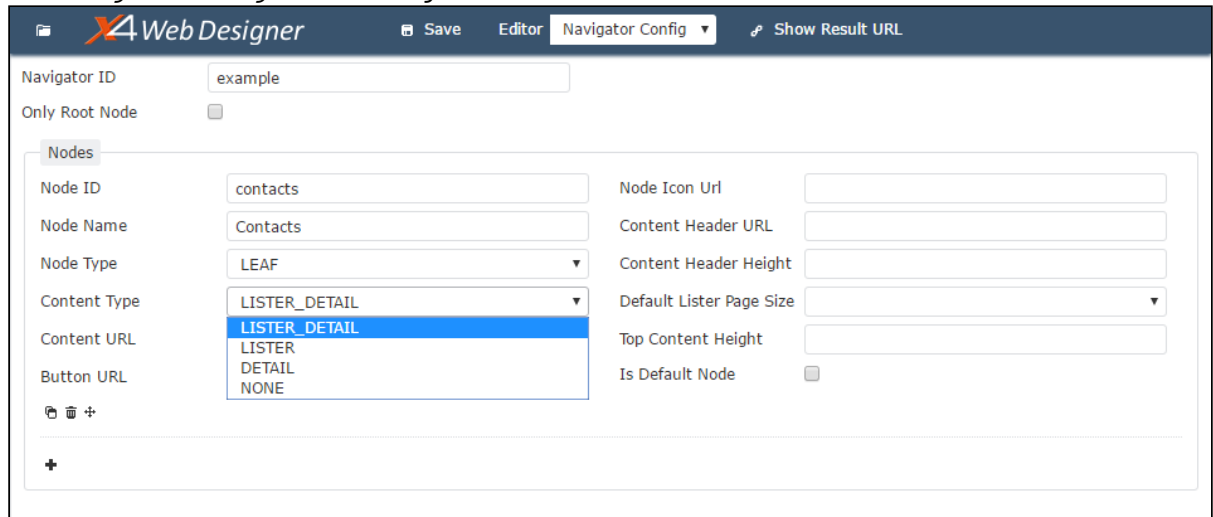
- [Ordner und Dateien im Navigator verwalten](#)
- [Tastenkombination im X4 Web Designer](#)
- [Main Config](#)

### 2.7.3 Navigator konfigurieren

Wie Sie im *X4 Web Designer* für Ihre Web-Anwendung mindestens einen Navigatorknoten definieren

1. Im *X4 Web Designer* zum Ordner <Projektname>/src/main/portal/navigator/ navigieren.
2. Eine bestehende Datei vom Typ `.nav.x4p.json` öffnen oder mit Klick auf   eine neue erstellen, siehe [Ordner und Dateien im Navigator verwalten](#).

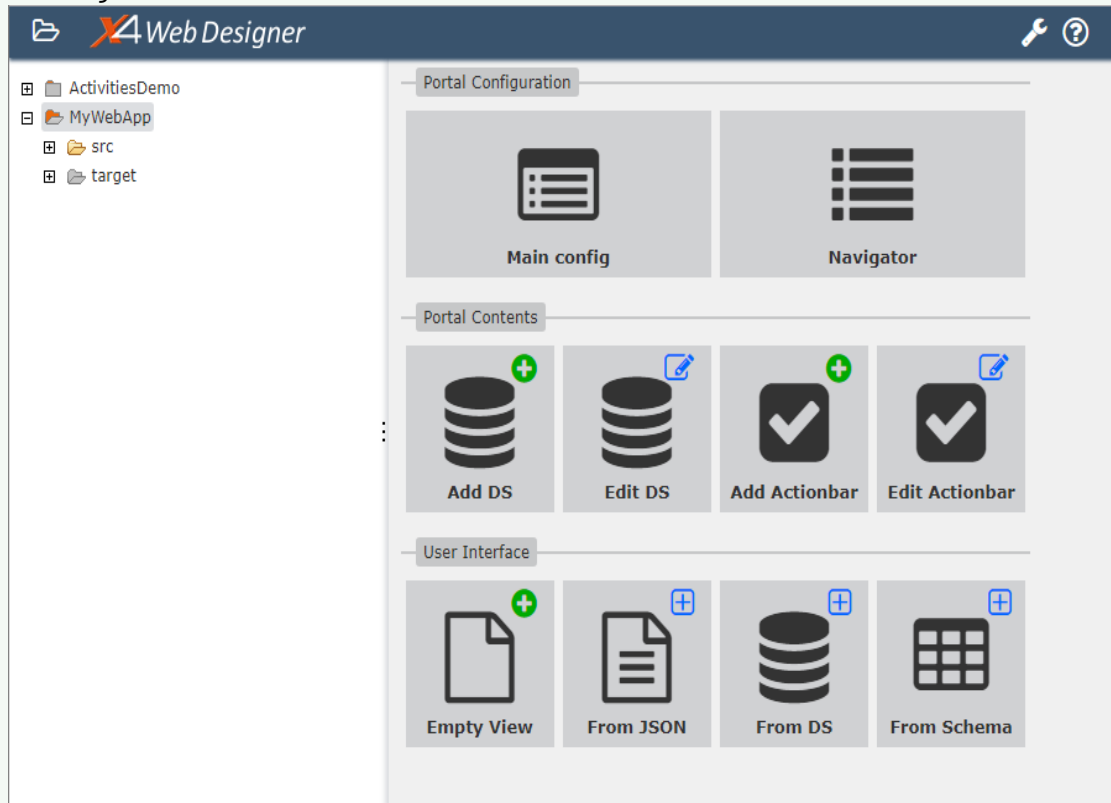
Der *Navigator Config Editor* wird geöffnet.



The screenshot shows the 'Navigator Config' window in X4 Web Designer. The interface includes a header bar with 'X4 Web Designer', 'Save', 'Editor', 'Navigator Config', and 'Show Result URL'. Below the header, the 'Navigator ID' is set to 'example' and 'Only Root Node' is unchecked. The 'Nodes' section contains a table with fields for Node ID, Node Name, Node Type, Content Type, Content URL, Button URL, Node Icon Url, Content Header URL, Content Header Height, Default Lister Page Size, Top Content Height, and Is Default Node. The 'Content Type' dropdown is open, showing options: LISTER\_DETAIL (selected), LISTER, and NONE. At the bottom, there are icons for adding, deleting, and refreshing nodes, and a plus sign to add a new node.

Field	Value
Navigator ID	example
Only Root Node	<input type="checkbox"/>
<b>Nodes</b>	
Node ID	contacts
Node Name	Contacts
Node Type	LEAF
Content Type	LISTER_DETAIL
Content URL	
Button URL	
Node Icon Url	
Content Header URL	
Content Header Height	
Default Lister Page Size	
Top Content Height	
Is Default Node	<input type="checkbox"/>

- ✓ Alternativ lässt sich der Navigator auch direkt über die Projekt-Startseite anlegen. Dazu auf **Navigator** klicken, einen Namen für den Navigator eingeben und mit **OK** bestätigen.



3. Konfiguration anpassen und/oder mit + eine neue Navigatorknoten-Konfiguration hinzufügen.

- ✓ *Tip:* In der Menüleiste können Sie im Auswahlfeld **Editor** zwischen dem grafischen Editor und der JSON-Code-Ansicht wechseln, um die tatsächlich erzeugte JSON-Datenstruktur einzusehen und ggf. zu editieren. Der JSON-Editor bietet den vollen Komfort eines JavaScript-Editors mit Syntaxhighlighting, Validierung und Autovervollständigung (mittels Strg+Leertaste).

4. **Save** klicken oder mit **Strg+S** speichern.  
Nach dem Speichern wird die Datei automatisch neu im target-Ordner gebaut.
5. **Show Result URL** klicken, um Ressourcen-URL zur Navigator-Konfiguration zu kopieren.  
Um die Navigator-Konfiguration für Ihre Web-Anwendung verfügbar zu machen, können Sie diese URL im *Application Config Editor* in **Navigation URL** (`baseUrl`) hinterlegen, siehe [Application-Config \(Main\) konfigurieren](#).

#### Siehe auch:



- [Ordner und Dateien im Navigator verwalten](#)
- [Tastenkombination im X4 Web Designer](#)



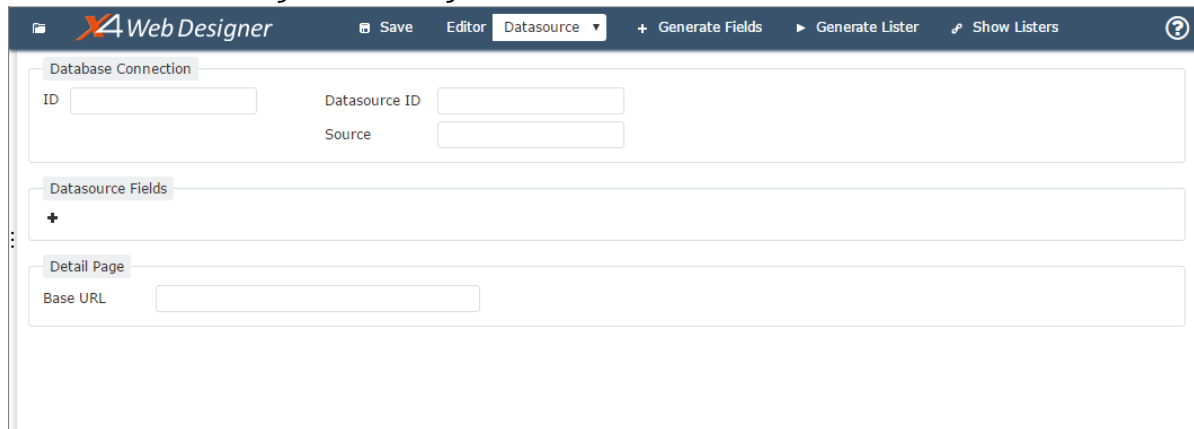
## 2.7.4 Datasources konfigurieren

Wie Sie im *X4 Web Designer* beliebige Daten aus Datenbankschemas mittels sogenannter *Datasources* bereitstellen, sodass Sie diese für Lister und Detailseiten Ihrer Web-Anwendung verwenden können

### 2.7.4.1 Datasource-Konfiguration anlegen und bearbeiten

1. Im *X4 Web Designer* zum Ordner `<Projektname>/src/main/portal/datasource/` navigieren.
2. Eine bestehende Datei vom Typ `.ds.x4p.json` öffnen oder mit Klick auf   eine neue erstellen, siehe [Ordner und Dateien im Navigator verwalten](#).

Der *Datasource Config Editor* wird geöffnet.

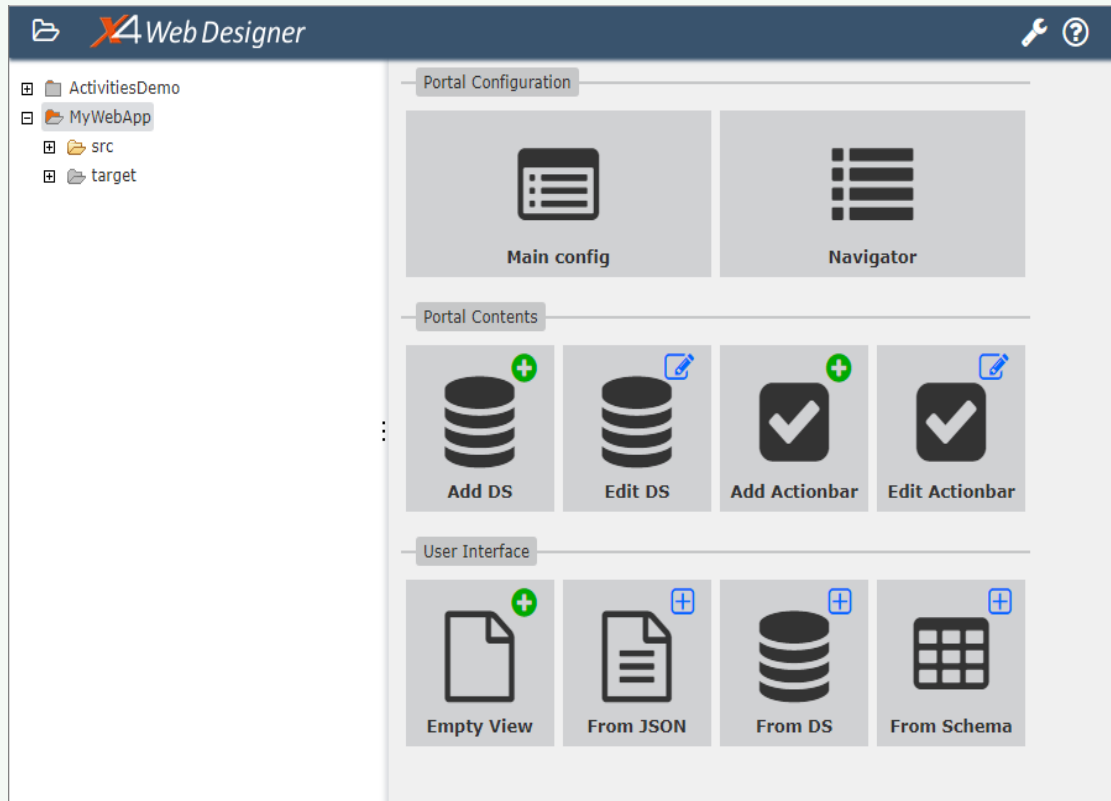


The screenshot shows the 'X4 Web Designer' application window. The title bar includes 'Save', 'Editor', and a dropdown menu set to 'Datasource'. Action buttons for '+ Generate Fields', '► Generate Lister', and 'Show Listers' are visible. The main content area is divided into three sections: 'Database Connection' with fields for 'ID', 'Datasource ID', and 'Source'; 'Datasource Fields' with a '+' button; and 'Detail Page' with a 'Base URL' field.

- ✓ Alternativ lässt sich die Datasource auch direkt über die Projekt-Startseite anlegen und öffnen.

Dazu auf **Add DS** klicken, einen Namen für die Datasource eingeben und mit **OK** bestätigen.

Zum Bearbeiten der Datasource auf **Edit DS** klicken.



3. Die Datenbank-Verbindung konfigurieren:
  - In **ID** die Technische ID der verwendeten Datenbankverbindung eingeben
  - Unter **Datasource ID** und **Source** einen eindeutigen Bezeichner sowie einen Namen für die Datenquelle eingeben
4. Entweder die Feldkonfiguration aus einer verfügbaren Datenbankverbindung generieren (siehe [Feldkonfiguration aus Datenbankverbindung generieren](#)) oder manuell definieren, indem Sie mit **+** bei Bedarf eine weitere Feldkonfigurationen hinzufügen.
5. In **Base URL** die URL zu einer entsprechenden Detailseite hinterlegen.  
Detailseiten können Sie beispielsweise mit dem *UI-Editor* grafisch erzeugen und dort über **Show Result URL** kopieren, siehe [Detailseiten mit dem UI-Editor erstellen](#). Hier können Sie jedoch jede beliebige URL zu einer Detailseite hinterlegen.

- ✓ *Tipp:* In der Menüleiste können Sie im Auswahlfeld **Editor** zwischen dem grafischen Editor und der JSON-Code-Ansicht wechseln, um die tatsächlich erzeugte JSON-Datenstruktur einzusehen und ggf. zu editieren.  
Der JSON-Editor bietet den vollen Komfort eines JavaScript-Editors mit Syntaxhighlighting, Validierung und Autovervollständigung (mittels Strg+Leertaste).

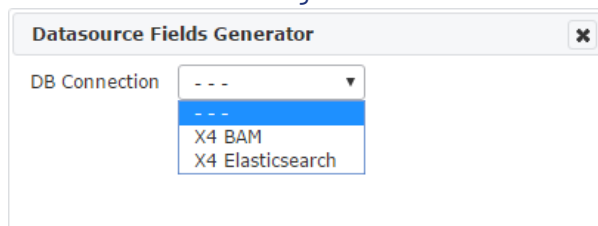
6. **Save** klicken oder mit **Strg+S** speichern.  
Nach dem Speichern wird die Datei automatisch neu im target-Ordner gebaut.

### 2.7.4.2 Feldkonfiguration aus Datenbankverbindung generieren

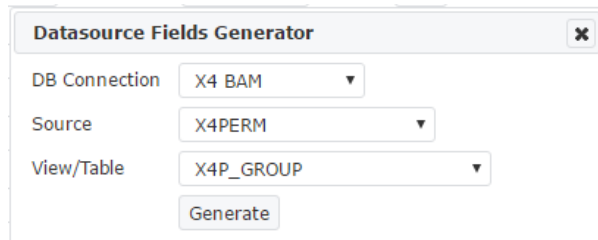
**Voraussetzung:** Der *X4 Web Designer* muss bereits Zugriff auf die zu verwendende Datenbank besitzen. Definieren Sie dazu mit dem *X4 Designer* ggf. zunächst eine entsprechende Datasource in der Konfigurationsdatei `dbTypes.xml`, siehe [Datenbank-Konfiguration für Web-Anwendungen](#).

1. In der Menüleiste des *Datasource Editors* auf **Generate Fields** klicken.

Ein Dialog wird geöffnet, in dem alle Datenbankverbindungen angezeigt werden, die dem *X4 Web Designer* in der Konfigurationsdatei `dbTypes.xml` bekannt gemacht wurden, siehe [Datenbank-Konfiguration für Web-Anwendungen](#).

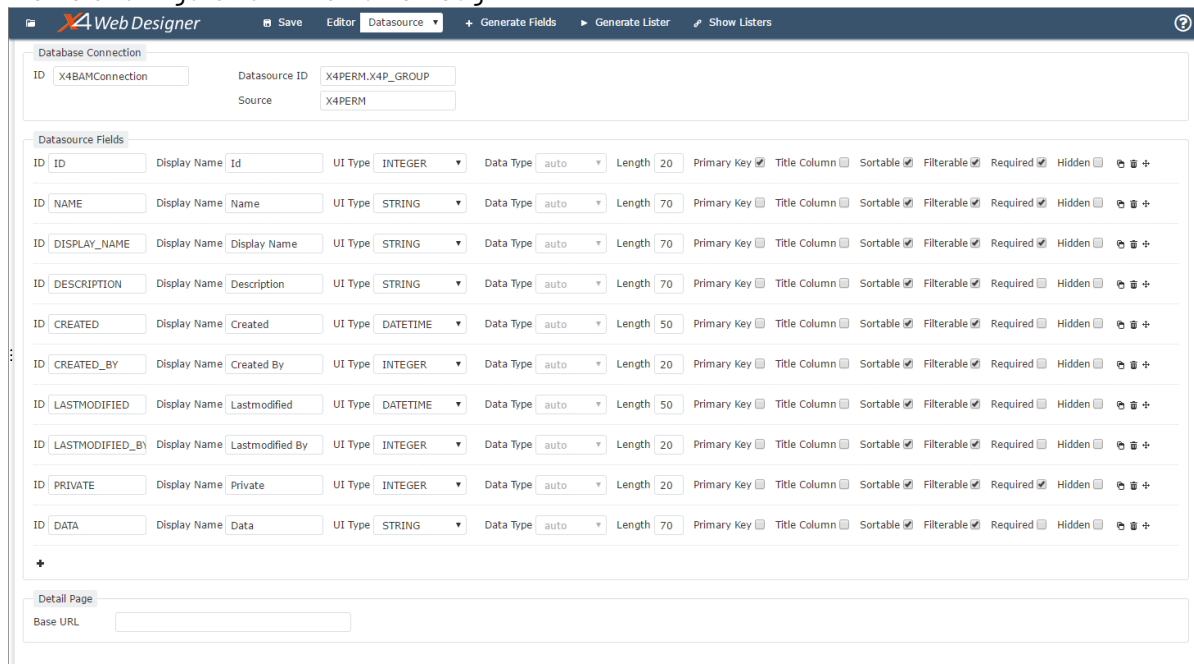


2. Eine Datenbankverbindung sowie das gewünschte Schema und eine View/Tabelle wählen. Zur Datenbankverbindung werden nach der Auswahl der Datenbankverbindung die verfügbaren Schemas sowie nach Auswahl des Schemas alle verfügbaren Tabellen angezeigt.




3. **Generate** klicken.


Die Feldkonfiguration wird nun erzeugt.



4. Ein geeignetes Feld als **Primary Key** markieren und die Konfigurationsdatei speichern. Beispiel: CUSTOMERID.

✓ **Tipp**

Mit  können Sie eine Zeile mit der Feldkonfiguration duplizieren.

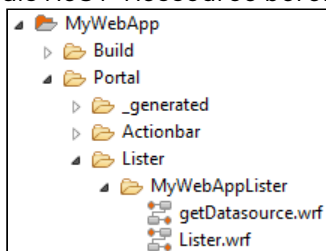
Mit  können Sie eine Zeile löschen.

Im JSON-Editor können Sie bei Bedarf weitere Anpassungen direkt im Quellcode der Konfigurationsdatei vornehmen.

### 2.7.4.3 Lister-Konfiguration aus Datasource generieren

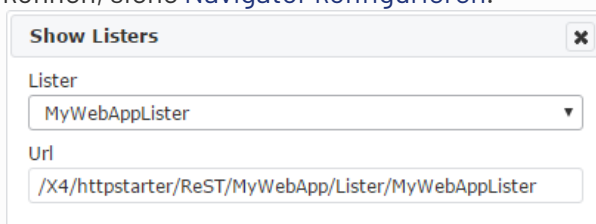
Nachdem Sie Ihre Datasource konfiguriert haben, können Sie bei Bedarf direkt eine oder mehrere Lister-Konfiguration(en) für Ihre Web-Anwendung generieren.

1. In der Menüleiste des *Datasource Editors* auf **Generate Lister** klicken.
2. Den gewünschten Lister-Namen im Dialog eingeben und **OK** klicken.  
Der Lister wird nun als Ressource angelegt. Hierzu wird im *X4 Repository* u.a. ein Lister-Prozess `<Projekt>/Portal/Lister/<Lister-Name>/Lister.wrf` erzeugt und automatisch als ReST-Ressource bereitgestellt, beispielsweise:



3. In der Menüleiste des *Datasource Editors* auf **Show Lister** klicken.  
Ein Dialog wird geöffnet, in dem alle bereits in der ReST-Konfiguration definierten Lister angezeigt werden.

ℹ Wenn Sie einen Lister wählen, wird dessen zugehörige Ressourcen-URL angezeigt, die Sie in der Navigator-Konfiguration als Content URL (`nodeContentUrl`) hinterlegen können, siehe [Navigator konfigurieren](#).



4. Bei Bedarf Fachlogik (Berechtigungsprüfungen, Internationalisierung etc.) für den/die Lister hinzufügen. Hierzu im *X4 Designer* den entsprechenden technischen Prozess `<Projekt>/Portal/Lister/<Lister-Name>/getDatasource.wrf` anpassen/erweitern.  
Dieser technische Prozess wird zur Laufzeit in den eigentlichen Lister-Prozess eingebunden und verwendet Ihre Datasource-Konfiguration.

**Auf dieser Seite:**

**Siehe auch:**

- [Datenbank-Konfiguration für Web-Anwendungen](#)
- [Tastenkombination im X4 Web Designer](#)
- [X4 Activities – Datasource-Definition für die Listenansicht](#)

## 2.7.5 Aktionsleiste konfigurieren

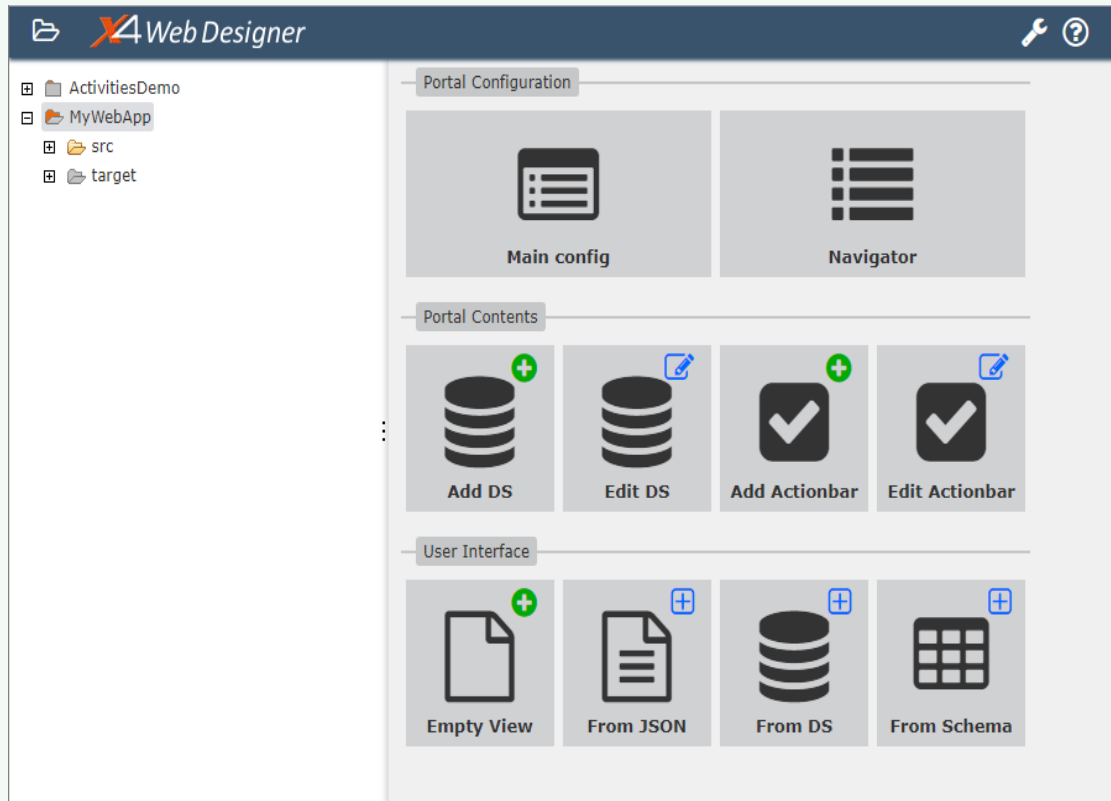
Wie Sie im *X4 Web Designer* eine Aktionsleiste mit einem oder mehreren Schaltflächen (Aktions-Buttons) für Ihre Web-Anwendung definieren

1. Im *X4 Web Designer* zum Ordner `<Projektname>/src/main/portal/actionbar/` navigieren.
2. Eine bestehende Datei vom Typ `.ab.x4p.json` öffnen oder mit Klick auf **+** eine neue erstellen, siehe [Ordner und Dateien im Navigator verwalten](#).

Der *Actionbar Config Editor* wird geöffnet.

The screenshot shows the 'Actionbar Config' editor in the X4 Web Designer. The interface includes a top toolbar with 'Save', 'Editor', 'Actionbar Config', and 'Copy Result URL' buttons. Below the toolbar, there is a section for 'Icon Size' with an input field. The main configuration area is divided into two columns. The left column contains fields for 'Button ID', 'Display Name', 'Action Type' (set to 'GET'), 'Action URL', and a 'Selection Needed' checkbox. The right column contains fields for 'Display Type' (set to 'Dialog'), 'Default Close' (set to 'AUTO'), 'Icon URL', 'Tooltip', 'Width', and 'Height'. At the bottom left, there are icons for deleting, adding, and a plus sign for creating new buttons.

- ✓ Alternativ lässt sich die Aktionsleiste auch direkt über die Projekt-Startseite anlegen und öffnen.  
Dazu auf **Add ActionBar** klicken, einen Namen für die Aktionsleiste eingeben und mit **OK** bestätigen.  
Zum Bearbeiten der Aktionsleiste auf **Edit ActionBar** klicken.



3. Konfiguration anpassen und/oder mit + einen neuen Aktions-Button hinzufügen.

- ✓ *Tip:* In der Menüleiste können Sie im Auswahlfeld **Editor** zwischen dem grafischen Editor und der JSON-Code-Ansicht wechseln, um die tatsächlich erzeugte JSON-Datenstruktur einzusehen und ggf. zu editieren.  
Der JSON-Editor bietet den vollen Komfort eines JavaScript-Editors mit Syntaxhighlighting, Validierung und Autovervollständigung (mittels Strg+Leertaste).


4. **Save** klicken oder mit Strg+S speichern.  
Nach dem Speichern wird die Datei automatisch neu im target-Ordner gebaut.
5. **Show Result URL** klicken, um Ressourcen-URL zur Aktionsleisten-Konfiguration zu kopieren.  
Um die Aktionsleiste für einen bestimmten Navigatorknoten verfügbar zu machen, können Sie diese URL in der Navigator-Konfiguration in **Button URL** (nodeButtonUrl) hinterlegen, siehe [Navigator konfigurieren](#).

**Siehe auch:**

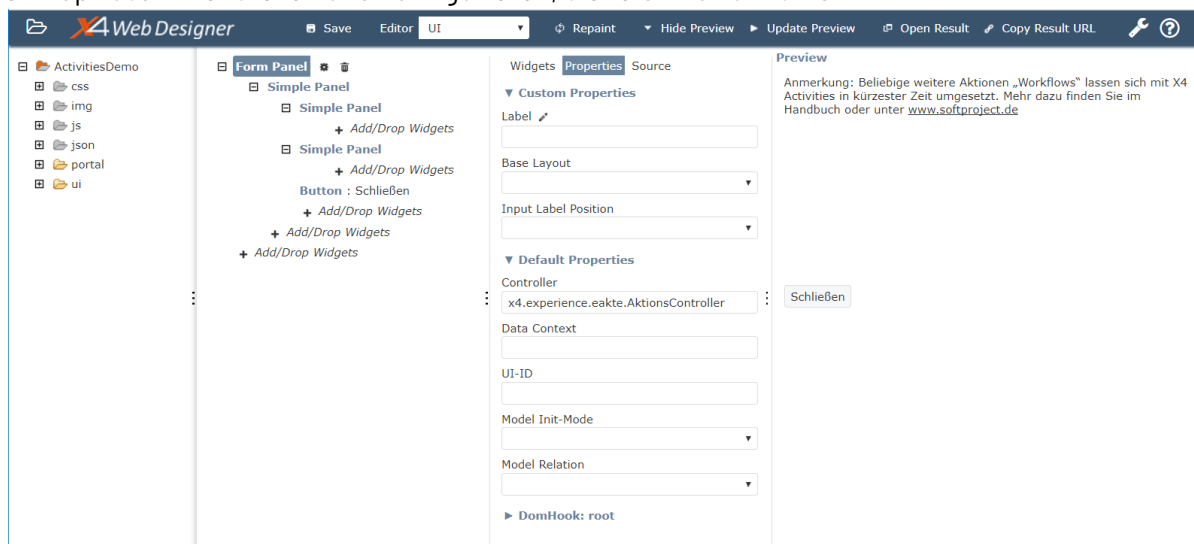
- Navigator konfigurieren
- Tastenkombination im X4 Web Designer
- X4 Activities – Aktionsleiste

## 2.7.6 Detailseiten mit dem UI-Editor erstellen

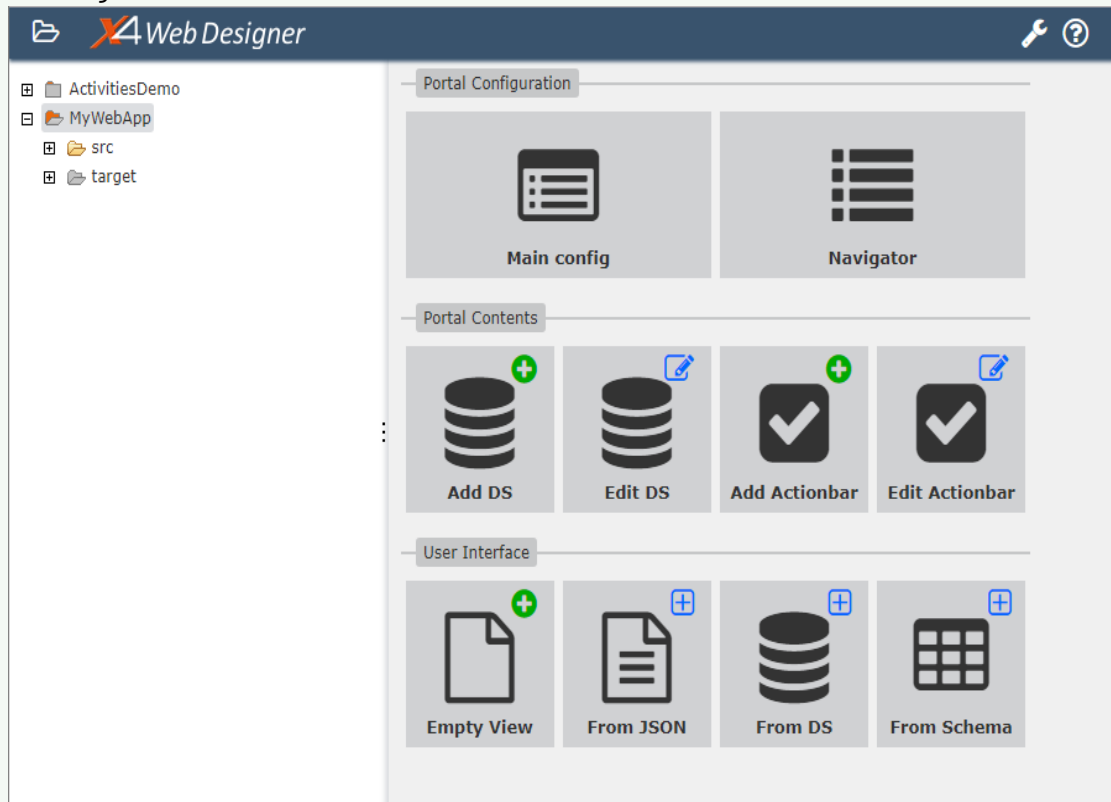
Wie Sie im *X4 Web Designer* im grafischen UI-Editor beliebig komplexe Detailseiten bzw. intelligente Formulare für Ihre Web-Anwendung definieren können.

1. Im *X4 Web Designer* zum Ordner `<Projektname>/src/main/ui/` navigieren.
2. Eine bestehende Datei vom Typ `.ui.json` öffnen oder mit Klick auf **+**  eine neue erstellen, siehe [Ordner und Dateien im Navigator verwalten](#).

Der *UI Editor* wird geöffnet. Hier können Sie Benutzeroberflächen wie z.B. Formulare per Drag & Drop zusammenstellen und konfigurieren, siehe [UI-Editor nutzen](#).



- ✓ Alternativ lässt sich eine neue Detailseite auch direkt über die Projekt-Startseite anlegen.  
Dazu auf **Empty View** klicken, einen Namen für die Detailseite eingeben und mit **OK** bestätigen.



3. UI-Widgets per Drag&Drop aus der **Widgets**-Palette einfügen, siehe [UI-Widgets einfügen und verwalten](#).

ⓘ Standardmäßig verfügbar sind [Formular-Widgets](#), [jQuery UI-Widgets](#) und [Layout-Widgets](#). Bei Bedarf können Sie zudem eigene UI-Widgets erstellen, siehe [Eigene Widgets erstellen](#).

4. Falls erforderlich in die JSON-Code-Ansicht wechseln, indem Sie in der Menüleiste **Editor > JSON** wählen.

✓ *Tipp:* In der Menüleiste können Sie im Auswahlfeld **Editor** zwischen dem grafischen Editor und der JSON-Code-Ansicht wechseln, um die tatsächlich erzeugte JSON-Datenstruktur einzusehen und ggf. zu editieren.  
Der JSON-Editor bietet den vollen Komfort eines JavaScript-Editors mit Syntaxhighlighting, Validierung und Autovervollständigung (mittels Strg+Leertaste).

5. **Save** klicken oder mit **Strg+S** speichern.  
Nach dem Speichern wird die Datei automatisch neu im **target**-Ordner gebaut.



6. **Show Result URL** klicken, um Ressourcen-URL zur Detailseite zu kopieren.

Im nächsten Schritt können Sie diese URL in die Datasource-Konfiguration in Eigenschaft *Base URL* (*BaseUri*) einfügen, siehe [Datasources konfigurieren](#).

**Siehe auch:**

- [UI-Editor im X4 Web Designer](#)
- [UI-Widgets einfügen und verwalten](#)
- [Datasources konfigurieren](#)
- [UI-Controller definieren](#)
- [Tastenkombination im X4 Web Designer](#)

## 2.8 X4 Activities-Anwendung anpassen

### 2.8.1 Gestaltung einer X4 Activities-Anwendung anpassen

Hier erfahren Sie, wie Sie die Gestaltung (Styling) einer X4 Activities-Anwendung ändern können.

#### 2.8.1.1 Anwendungs-Logo konfigurieren

Für das Logo einer X4 Activities-Anwendung kann in der Basis-Konfiguration im Parameter `x4p.application.iconUrl` (siehe [Anwendungsspezifische Konfiguration](#)) eine URL hinterlegt werden. Alternativ kann das Logo auch dynamisch über das Attribut `logoUrl` im Main-Knoten der [Main Config](#) definiert werden.

Das Logo kann direkt im Maven-Projekt im Verzeichnis `src/main/webapp/static` abgelegt und dann über die URL `/<Projektname>/static/<Bild-Dateiname>` eingebunden werden. Alternativ kann eine beliebige externe URL oder auch ein File-Service für die Bereitstellung des Logos verwendet werden.

#### 2.8.1.2 Login-/Logout-Maske konfigurieren

Für die Login- und Logout-Maske lässt sich bei Bedarf eine Hintergrundbild hinterlegen.

Die Formatierung des Hintergrundbildes lässt sich über folgende css-Eigenschaften steuern:

"background-size: auto;"	Bild wird in Originalgröße angezeigt (Standard)
"background-size: contain;"	Bild wird skaliert, um den Hintergrund möglichst abzudecken; Der Hintergrund kann aber teilweise frei bleiben.
"background-size: cover;"	Bild wird skaliert, so dass der Hintergrund vollständig abgedeckt wird; Das Bild kann teilweise abgeschnitten sein.

❗ Für weitere Styling-Möglichkeiten wie die Skalierung in Prozent oder z.B. "background-repeat: no-repeat;", siehe [https://www.w3schools.com/cssref/css3\\_pr\\_background-size.asp](https://www.w3schools.com/cssref/css3_pr_background-size.asp)

Um das Hintergrundbild zur Verfügung zu stellen, sind folgende Schritte notwendig:

1. Das gewünschte Hintergrundbild zum war-Archiv des Portals unter dem Pfad `static/img/background.png` hinzufügen.
2. Die Datei `<Portalname>.css` editieren und folgendes Styling hinzufügen:

```
#mainView-contentLayout:empty
{
    background-image:url(static/img/background.png);
    background-size: cover;
    background-repeat: no-repeat;
}
```

3. Das war-Archiv anschließend neu deployen.

### 2.8.1.3 Icons für Navigator-Knoten und Schaltflächen

X4 Activities stellt aktuell folgende Standard-Icons für Navigator-Knoten und Schaltflächen in der Aktionsleiste zur Verfügung:

+	/X4PSupport/images/common/Add.png
⊗	/X4PSupport/images/common/Cancel.png
✓	/X4PSupport/images/common/Check.png
✕	/X4PSupport/images/common/Delete.png
✚	/X4PSupport/images/common/Duplicate.png
⇩	/X4PSupport/images/download.png

❗ Weitere Icons müssen bei Bedarf projektspezifisch bereitgestellt werden. Hierzu bietet sich z. B. der File-Service des X4 Servers an.

### 2.8.1.4 Styling-Anpassung

#### 2.8.1.4.1 Basis-Theme

Um das Theme (Farben, Schriftarten, Schriftgröße) anzupassen, wird im Maven-Projekt einer X4 Activities-Anwendung eine JavaScript-Datei (`src/main/webapp/static/Theme.js`) bereitgestellt, die beim Laden von X4 Activities das Theme initialisiert. In der Datei ist ein JSON-Objekt definiert, das die Werte des Themes enthält. Die Werte können bei Bedarf auch hier angepasst werden. Zudem kann auch (synchrone!) Logik zum dynamischen Initialisieren des Themes definiert werden, z. B. abhängig von URL-Parametern oder Werten im Local-Storage des Browsers.

Das Basis-Theme stellt folgende Eigenschaften bereit:

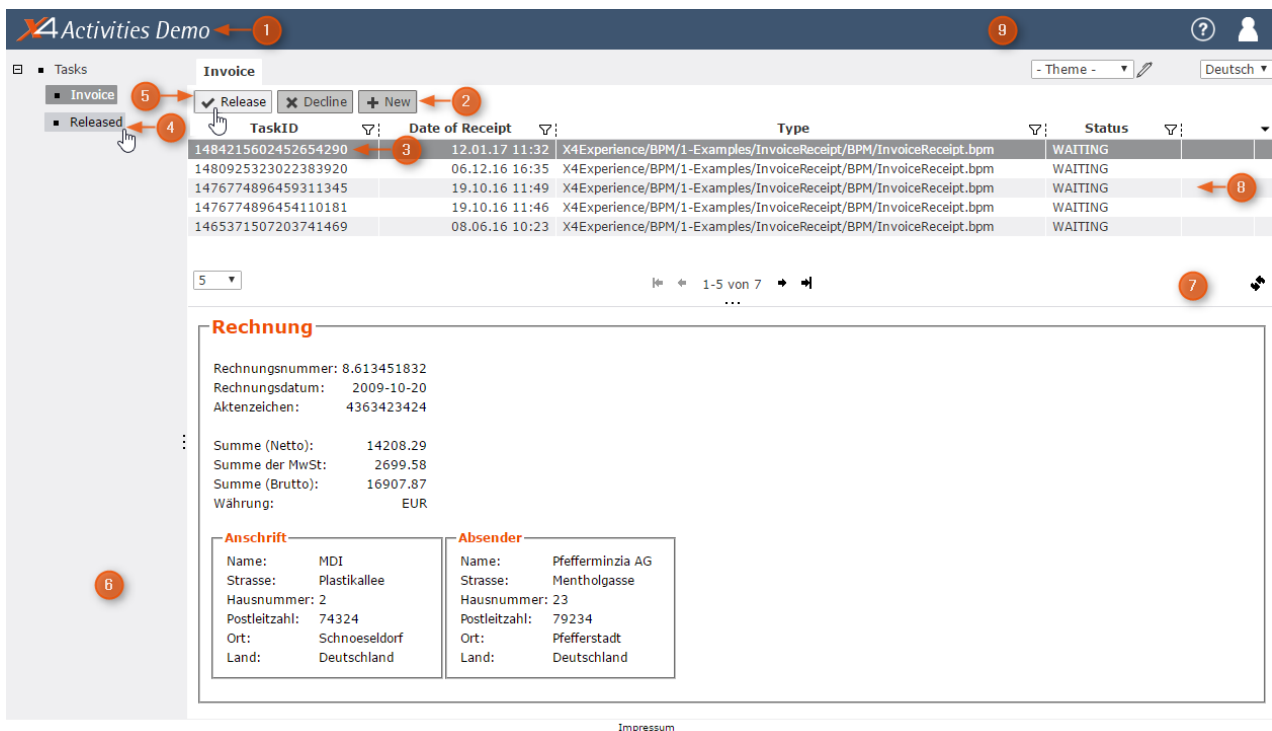
Eigenschaft	Beschreibung
fontFamily	Schriftart
fontSize	Schriftgröße
textColor	Textfarbe
color	Hintergrundfarbe der Anwendung
color2	Hintergrundfarbe des Content-Bereichs
color3	Hintergrundfarbe für Schaltflächen
borderColor	Linien-Farbe für die Haupt-Panels der Anwendung
borderColor2	Linien-Farbe für Content-Panels der Anwendung
borderColor3	Linien-Farbe für Schaltflächen und Eingabefelder
selectedColor	Hintergrundfarbe für selektierte Elemente
selectedTextColor	Textfarbe für selektierte Elemente
hoverColor	Hintergrundfarbe beim Bewegen der Maus über ein selektierbares Element
hoverColor2	Hintergrundfarbe beim Bewegen der Maus über eine Schaltfläche
hoverBorderColor	Linien-Farbe beim Bewegen der Maus über eine Schaltfläche oder ein Eingabefeld
inputTextColor	Textfarbe von Eingabefeldern
inputColor	Hintergrundfarbe von Eingabefeldern
borderRadius	Definiert die Rundung von Ecken als CSS-Längenangabe
layoutShadowsEnabled	Boolescher Wert, der festlegt, ob zwischen den Haupt-Bereichen der Anwendung Schatten angezeigt werden sollen (true) oder nicht (false)
header.fontSize	Schriftgröße im Header
header.textColor	Textfarbe im Header
header.color	Hintergrundfarbe im Header
header.hoverTextColor	Textfarbe beim Bewegen der Maus über eine Schaltfläche im Header
header.hoverColor	Hintergrundfarbe beim Bewegen der Maus über eine Schaltfläche im Header
table.oddRowColor	Hintergrundfarbe, in der jede zweite Zeile von Tabellen eingefärbt wird.

#### 2.8.1.4.2 Erweitertes Styling

Für erweiterte Styling-Möglichkeiten stehen unter `src\main\resources\de\softproject\portal\style` verschiedene CSS-Dateien bereit, in denen GWT-Konstanten für die X4 Activities-Stylesheets geändert werden können.

#### 2.8.1.5 Beispiel

Folgende Beispiel-Anwendung auf Basis von X4 Activities soll einige Anpassungsmöglichkeiten verdeutlichen:



### Erläuterungen:


1. Anwendungs-Logo
2. Schaltflächen
  - Schaltflächen-Icons
  - color3: Hintergrundfarbe für Schaltflächen
  - borderColor3: Linien-Farbe für Schaltflächen und Eingabefelder
3. Selektierte Elemente
  - selectedColor: Hintergrundfarbe für selektierte Elemente
  - selectedTextColor: Textfarbe für selektierte Elemente
4. Mouse-Over bei Elementen
  - hoverColor: Hintergrundfarbe beim Bewegen der Maus über ein selektierbares Element
  - selectedTextColor: Textfarbe für selektierte Elemente
5. Mouse-Over bei Schaltflächen
  - hoverColor2: Hintergrundfarbe beim Bewegen der Maus über eine Schaltfläche
  - hoverBorderColor: Linien-Farbe beim Bewegen der Maus über eine Schaltfläche oder ein Eingabefeld
6. color: Hintergrundfarbe der Anwendung
7. color2: Hintergrundfarbe des Content-Bereichs
8. table.oddRowColor: Hintergrundfarbe, in der jede zweite Zeile von Tabellen eingefärbt wird.
9. header.color: Hintergrundfarbe im Header

## 2.8.2 Anwendungsspezifische Konfiguration

Für jede Web-Anwendung, die als .war-Datei im Applikations-Server bereitgestellt wird, sind anwendungsspezifische Einstellungen verfügbar. Diese Konfiguration ist Bestandteil des Projekts und wird in der web.xml-Datei des Projekts abgelegt. Wenn ein Konfigurationsattribut fehlt, wird eine entsprechende Fehlermeldung erzeugt und in server.log protokolliert.

### 2.8.2.1 Applikations-Einstellungen

Eigenschaft	Beschreibung
x4p.applicationConfig.uri	<p>URL zur Application-Config (Konfiguration der Unteranwendungen), z.B.</p> <pre>&lt;context-param&gt;   &lt;param-name&gt;x4p.applicationConfig.uri&lt;/param-name&gt;   &lt;param-value&gt; /X4/httpstarter/ReST/Applikationsname/config/   ApplicationConfig &lt;/param-value&gt; &lt;/context-param&gt;</pre>
x4p.application.name	<p>Name der Web-Applikation, z.B.</p> <pre>&lt;context-param&gt;   &lt;param-name&gt;x4p.application.name&lt;/param-name&gt;   &lt;param-value&gt;Applikationsname&lt;/param-value&gt; &lt;/context-param&gt;</pre> <p> Der Name erscheint als Titel im Browser.</p>
x4p.application.iconUrl	<p>URL zum Logo der Web-Applikation, z.B.</p> <pre>&lt;context-param&gt;   &lt;param-name&gt;x4p.application.iconUrl&lt;/param-name&gt;   &lt;param-value&gt;Logo.png&lt;/param-value&gt; &lt;/context-param&gt;</pre>
x4p.application.embeddedMode	<p>Legt fest, wie Header und Footer der Web-Anwendung angezeigt werden. Wenn dieser Parameter auf true gesetzt wird, dann werden Header und Footer ausgeblendet.</p> <p><b>Mögliche Werte:</b> true / false (Standard)</p> <div> <p><b>Beispiel</b></p> <pre>&lt;context-param&gt;   &lt;param-name&gt;x4p.application.embeddedMode&lt;/param-name&gt;   &lt;param-value&gt;&gt;false&lt;/param-value&gt; &lt;/context-param&gt;</pre> </div>


Eigenschaft	Beschreibung
x4p.login.disableAutocomplete	<p>Legt fest, ob die Autovervollständigung für Login-Eingabefelder aktiviert ist.</p> <p><b>Mögliche Werte:</b> <code>true</code> (Autovervollständigung inaktiv)/ <code>false</code> (Autovervollständigung aktiv, Standard)</p> <div>  Diese Einstellung hat keinen Einfluss auf das Verhalten von in den Browsern integrierten Passwortmanagern. </div>

### 2.8.2.2 RPC Timeout

Eigenschaft	Beschreibung
x4p.rpc.timeout	<p>(optional) Zeitbeschränkung für RPC-Requests vom Client an den GWT-Server</p> <p>Folgende Werte können für diesen optionalen Context-Parameter in der <code>web.xml</code> des Moduls eingetragen werden:</p> <ul style="list-style-type: none"> <li>• Beliebiger Wert in Millisekunden</li> <li>• 170000: 17 Sekunden (Standard)</li> </ul> <p> Bei allen Requests, die vor Abruf der <i>ClientConfiguration</i> erfolgen, wird der Standard-Timeout verwendet.</p> <div> <p><b>Beispiel</b></p> <pre>&lt;context-param&gt;   &lt;param-name&gt;x4p.rpc.timeout&lt;/param-name&gt;   &lt;param-value&gt;17000&lt;/param-value&gt; &lt;/context-param&gt;</pre> </div>

### 2.8.2.3 Lister-Einstellungen

Für die Speicherung der Lister Preferences (Sortierung, Filterung, Spaltenbreiten/-Sichtbarkeit) gibt es folgende optionale Konfigurationsparameter in der `web.xml`, die im Folgenden erläutert werden.

Eigenschaft	Beschreibung
x4p.listerPref .autosaveEnabled	<p>Aktiviert/deaktiviert die automatische Speicherung bei einer Änderung der Spaltenkonfiguration. Automatisch gespeicherte Daten werden immer im <i>Local Storage</i> des Browsers abgelegt.</p> <ul style="list-style-type: none"> <li>• true: Spaltenkonfiguration automatisch speichern</li> <li>• false: Daten nicht im <i>Local Storage</i> des Browsers ablegen (Standard)</li> </ul> <p> Bei Browsern, die keinen <i>Local Storage</i> unterstützen, funktioniert die automatische Speicherung nicht.</p> <div data-bbox="676 719 1441 1025"> <p><b>Beispiel</b></p> <pre>&lt;context-param&gt;   &lt;param-name&gt; x4p.listerPref.autosaveEnabled &lt;/param-name&gt;   &lt;param-value&gt;true&lt;/param-value&gt; &lt;/context-param&gt;</pre> </div>
x4p.listerPref .namedEntriesEnabled	<p>Aktiviert/deaktiviert, ob der Anwender benannte Tabellenkonfigurationen speichern kann. Diese Einträge werden entweder im <i>Local Storage</i> des Browsers oder in einer Datenbank des <i>X4 Servers</i> gespeichert.</p> <ul style="list-style-type: none"> <li>• true: benannte Tabellenkonfigurationen speichern</li> <li>• false: keine benannte Tabellenkonfigurationen speichern (Standard)</li> </ul> <div data-bbox="676 1346 1441 1648"> <p><b>Beispiel</b></p> <pre>&lt;context-param&gt;   &lt;param-name&gt; x4p.listerPref.namedEntriesEnabled &lt;/param-name&gt;   &lt;param-value&gt;false&lt;/param-value&gt; &lt;/context-param&gt;</pre> </div>

Eigenschaft	Beschreibung
x4p.listerPref.remoteStorageEnabled	<p>Aktiviert/Deaktiviert das Speichern benannter Tabellenkonfigurationen auf dem <i>X4 Server</i></p> <ul style="list-style-type: none"><li>• true: Daten in der Datenbank auf dem Server speichern</li><li>• false: Daten im <i>Local Storage</i> des Browsers speichern (Standard)</li></ul> <div><b>Beispiel</b></div> <pre>&lt;context-param&gt;   &lt;param-name&gt; x4p.listerPref.remoteStorageEnabled &lt;/param-name&gt;   &lt;param-value&gt;false&lt;/param-value&gt; &lt;/context-param&gt;</pre>
x4p.listerPref.maxNamedEntries	<p>Maximale Anzahl benannter Speichereinträge, die ein Benutzer pro Navigator-Config speichern darf. Dieser Wert gilt sowohl bei Speicherung im <i>Local Storage</i> als auch für die Datenbank.</p> <ul style="list-style-type: none"><li>• Beliebige ganze Zahl</li><li>• 30 (Standard)</li></ul> <div><b>Beispiel</b></div> <pre>&lt;context-param&gt;   &lt;param-name&gt; x4p.listerPref.maxNamedEntries &lt;/param-name&gt;   &lt;param-value&gt;30&lt;/param-value&gt; &lt;/context-param&gt;</pre>
x4p.listerPref.defaultName	<p>Legt einen Standardnamen für benannte Tabellenkonfigurationen fest.</p> <div><b>Beispiel</b></div> <pre>&lt;context-param&gt;   &lt;param-name&gt;x4p.listerPref.defaultName&lt;/param-name&gt;   &lt;param-value&gt;Default&lt;/param-value&gt; &lt;/context-param&gt;</pre>



Eigenschaft	Beschreibung
x4p.listerPref .autoloadByDefaultNameEnabled	<p>Aktiviert/deaktiviert das automatische Laden einer mit dem Standardnamen benannten Tabellenkonfiguration, falls eine solche vorhanden ist. Ist das automatische Speichern aktiviert, haben unbenannte Einträge im <i>LocalStorage</i> Vorrang.</p> <ul style="list-style-type: none"> <li>• true: Automatisches Laden aktivieren</li> <li>• false: Automatisches Laden deaktivieren (Standard)</li> </ul> <div> <b>Beispiel</b> <pre> &lt;context-param&gt;   &lt;param-name&gt; x4p.listerPref/autoloadByDefaultNameEnabled&lt;/ param-name&gt;   &lt;param-value&gt;false&lt;/param-value&gt; &lt;/context-param&gt; </pre> </div>

#### 2.8.2.4 Senden von HTTP-Headern vom Server an den Client

Eigenschaft	Beschreibung
x4p.copy.header	<p>Gibt die HTTP-Header-Namen an, die vom Server an den Client gesendet werden. Sollen mehrere Header gesendet werden, so werden die Header-Namen durch Kommas getrennt angegeben.</p> <div> <b>Beispiel</b> <pre> &lt;context-param&gt;   &lt;param-name&gt;x4p.copy.header&lt;/param-name&gt;   &lt;param-value&gt;Set-Cookie,Origin,Referer&lt;/param-value&gt; &lt;/context-param&gt; </pre> </div>


## 2.9 Web-Anwendung neu bauen (Build)

Dieser Abschnitt beschreibt Möglichkeiten, um Ihr Projekt auf Basis der von Ihnen erstellten Quelldateien neu zu erzeugen.

 Wenn Sie im *X4 Web Designer* eine Datei speichern, wird sie automatisch neu gebaut.

### 2.9.1 Einzelne Ordner neu bauen

Sie können im *X4 Web Designer* bei Bedarf auch gezielt einzelne Ordner im Navigator neu bauen lassen:

Dazu im Navigator neben dem Ordner auf  **Open context menu** klicken und ► **Build** aus dem Kontextmenü wählen.

Im entsprechenden target-Unterordner wird nun der gewünschte Ressourcen-Ordner für die Web-Anwendung neu erzeugt.

## 2.9.2 Über den X4 Designer komplett neu bauen

Auch wenn Sie sich in der Eclipse-basierten Entwicklungsumgebung X4 Designer befinden, können Sie den Build für Ihre Web-Anwendung anstoßen.

1. Im **Repository Navigator** den technischen Prozess <Projektname>/Build/Build.wrf finden.  
*Beispiel:* MyWebApp/Build/Build.wrf:
2. Im **Repository Navigator** per Rechtsklick auf den technischen Prozess im Kontextmenü z.B. **Run Process Asynchronously** wählen.  
Nun wird der komplette target-Ordner für das Projekt neu erzeugt, z.B. MyWebApp/Portal/WebResources/target.  
Dieser Vorgang kann einige Sekunden dauern.

✓ Sie können diesen technischen Prozess über den Scheduler im X4 Control Center automatisiert starten (siehe *Handbuch X4 Control Center – Scheduler*) oder z.B. via ReSTful Webservice in Ihre *Continuous Integration*-Umgebung einbinden.

## 2.10 Web-Anwendung bereitstellen

### 2.10.1 Individuelle Web-Anwendung verwenden

Für jede Activities-Anwendung ist eine eigene Web-Anwendung erforderlich. Web-Applikationen werden dabei als .war-Datei im Application Server bereitgestellt (z. B. unter <X4>\Server\wildfly\standalone\deployments).

✓ Auf Wunsch wird Ihnen eine individuell konfigurierte Web-Anwendung bereitgestellt. Neben der technischen Konfiguration passen wir dabei gerne auch die Gestaltung der Oberfläche gemäß Ihrer CI-Richtlinien für Sie an.

### 2.10.2 Demo-Web-Anwendung verwenden

Alternativ können Sie für Test- und Entwicklungszwecke die im Installationspaket mitgelieferte Demo-Web-Anwendung (X4PDemonstrator.war) verwenden. Um Ihre Web-Anwendung auf diese Weise auf dem Applikations-Server bereitzustellen, sind folgende Schritte erforderlich, die hier beispielhaft beschrieben wurden anhand des Applikations-Servers *Wildfly 10.1.0*:

1. Relative URL zur Application-Config ermitteln, siehe [Anwendungsspezifische Konfiguration](#).
2. Ein Web-Archiv (.war-Datei) für *X4 Activities*-Anwendungen kopieren und modifizieren:

- Beispielsweise Datei `X4PDemonstrator.war` im Ordner `<X4>/Server/wildfly/standalone/deployments` in ein lokales Verzeichnis kopieren und mit einem ZIP-Werkzeug entpacken.
- In der Datei `WEB-INF/web.xml` zum Konfigurationsparameter `x4p.applicationConfig.uri` als Wert die URL zur Application-Config hinterlegen (siehe Schritt 1).

**Beispiel**

```
<context-param>
  <param-name>x4p.applicationConfig.uri</param-name>
  <param-value>/X4/httpstarter/ReST/MyWebApp/Resources</param-value>
</context-param>
```

- Das entpackte Web-Archiv mit einem ZIP-Werkzeug wieder als `.war`-Datei packen.
- Die modifizierte `.war`-Datei entsprechend umbenannt auf dem Applikations-Server bereitstellen.


*Beispiel: MyWebApp.war*

3. Konfigurationsdatei `WEB-INF/jboss-web.xml` anpassen, um eine Namenskollision mit der bestehenden Demonstrator-Anwendung zu vermeiden:

- Den Namen Ihres Web-Archivs (ohne Dateierdung `.war`) ermitteln, z.B. *MyWebApp*.
- In Element `<context-root>` den entsprechenden eindeutigen Namen eingeben und `jboss-web.xml` speichern.

**Beispiel**

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-web>
  <context-root>/MyWebApp</context-root>
</jboss-web>
```

 Wenn `jboss-web.xml` nicht vorhanden ist, wird stattdessen der Name des Web-Archivs (`.war`) verwendet.

Ihre Web-Anwendung wurde nun als eigenständige Applikation auf dem Applikations-Server bereitgestellt.

## 2.11 Anwendung mit dem X4 Web Designer testen und bereitstellen

Wie Sie Ihre Web-Anwendung, die Sie mit dem *X4 Web Designer* erstellt haben, temporär zum Test oder produktiv auf dem Applikations-Server bereitstellen

### 2.11.1 Web-Anwendung zu Testzwecken bereitstellen

Um die von Ihnen entwickelte Web-Anwendung beispielsweise für Testzwecke anderen Nutzern zur Verfügung zu stellen, können Sie diese und auch bereits bestehende *X4 Activities*-Web-

Anwendungen testweise über die Demo-Portal-Anwendung in `X4PDemonstrator.war` über den URL-Part `/X4PDemo` (z.B. `http://localhost:8080/X4PDemo/`) bereitstellen.

Hierzu steht ein spezieller URL-Parameter `mainUrl` zur Verfügung:

1. Im *X4 Web Designer* die entsprechende Application-Config `<Projektname>/src/main/portal/main/<Dateiname>.main.x4p.json` öffnen.  
*Beispiel:* `MyWebApp/src/main/portal/main/example.main.x4p.json`
2. In der Menüleiste des *X4 Web Designers* auf **Open Result URL in Portal** klicken.  
Ihre Web-Anwendung wird nun in einer neuen Browser-Registerkarte aufgerufen. Diese Portal-Anwendung ist verfügbar unter einer URL nach folgendem Muster:  
`<Host>/X4PDemo/?mainUrl=<URL-encodierte ReST-URL der Web-Anwendung>`

#### Beispiel

```
http://localhost:8080/X4PDemo/?mainUrl=%2FX4%2Fhttpstarter%2FReST%2FMyWebApp%2FResources%2Fportal%2Fmain%2Fexample.main.x4p.xml
```

## 2.11.2 Web-Anwendung permanent auf dem Applikations-Server bereitstellen (Deployment)

Für den Testbetrieb Ihrer mit dem *X4 Web Designer* erstellten Web-Anwendung wird standardmäßig die mitgelieferte Demo-Web-Anwendung (`X4PDemonstrator.war`) verwendet. Um Ihre Web-Anwendung für den produktiven Betrieb bereitzustellen, gehen Sie wie folgt vor:

1. Die Anweisungen aus dem Handbuch *X4 Activities* – [Web-Anwendung bereitstellen](#) befolgen.
2. Den *X4 Web Designer* vor der Produktivnahme deaktivieren, um unbefugte Modifikationen auszuschließen:
  - a. Im *X4 Designer* die Webservice-Konfiguration über Menü **Management > Webservice Configuration** öffnen.
  - b. Include-Knoten `X4WebIDE` markieren und die `Entf`-Taste drücken, um den Eintrag zu löschen.
  - c. Auf **Reload on Server** klicken und im Dialog mit **Save and Reload** bestätigen.
3. Den *X4 Server* absichern, siehe Handbuch *X4 Server* – Zugriff auf das *X4 Control Center* absichern.

Ihre Web-Anwendung auf Basis von *X4 Activities* kann nun produktiv genutzt werden.

#### Auf dieser Seite:

#### Siehe auch:

- [Application-Config \(Main\) konfigurieren](#)
- [Anwendungsspezifische Konfiguration](#)
- [Web-Anwendung bereitstellen](#)

## 2.12 Funktionen von X4 Activities

### 2.12.1 Berechtigungsprüfung

In den meisten *X4 Activities*-Anwendungen müssen die Berechtigungen der jeweiligen Benutzer überprüft werden, um unbefugten Zugriff auf Daten zu vermeiden. Daher sollten sämtliche technischen Prozesse, die u. a. über *X4 ReST* erreicht werden können, mit einer Berechtigungsprüfung ausgestattet werden. Ein Benutzer, der auf eine bestimmte Funktion nicht zugreifen darf, sollte eine entsprechende Antwort des *X4 Servers* erhalten. Hierfür bietet sich der HTTP-Statuscode 403 *Forbidden* an.

Die Berechtigungen werden bei einer Anfrage über *X4 ReST* automatisch ausgewertet und der Anfrage in einer `<Security>`-Elementstruktur hinzugefügt.

#### 2.12.1.1 Aufbau einer Berechtigung

Eine Berechtigung enthält diverse Datenfelder mit unterschiedlichen Bedeutungen:

```
<Security realm="X4Portal">
  <Group class="de.softproject.jaas.principal.PrincipalGroup"
        name="Permissions">
    <Principal class="de.softproject.jaas.principal.PermissionPrincipal"
              name="X4 URM Administer Users" object_type="user"
              module="URM" parameter="*" action="administer" />
  </Group>
</Security>
```

Sämtliche Berechtigungen erscheinen im *X4 ReST* HTTP-Request innerhalb einer Gruppe mit dem Namen `Permissions`. Einzelne Berechtigungen werden jeweils als `<Principal>`-Elemente mit folgenden Attributen abgebildet:

- `class`: Java-Klasse, mit der die Berechtigung abgebildet wird (nicht für die Auswertung der Berechtigungen relevant)
- `name`: Eine Freitextbezeichnung der Berechtigung (darf nicht für die Auswertung der Berechtigungen relevant sein)
- `module`: Anwendung, auf die sich die Berechtigung bezieht. Die Anwendung ist eine einzelne *X4 Activities*-Anwendung, auf welche die Berechtigung wirken soll, z.B. `BAM`.
- `object_type`: Objekt, auf das sich die Berechtigung bezieht. Ein Objekt kann beispielsweise ein Benutzer, eine Rechnung oder ein Bericht sein.
- `action`: Aktion, die auf dem Objekt ausgeführt werden darf, also beispielsweise betrachten, editieren oder löschen.
- `parameter`: optionale Parameter für bestimmte Einschränkungen. Somit könnte beispielsweise die Berechtigung nur auf bestimmte Objekte (z. B. mit einem bestimmten Attribut) wirken.

Zur Prüfung einer Berechtigung werden folgende Attribute verwendet:

- `module`
- `object_type`

- action
- parameter (optional)

 Eine Prüfung auf die Attribute `class` oder `name` ist nicht empfehlenswert.

### 2.12.1.2 Im technischen Prozess auf eine Berechtigung prüfen

Eine Berechtigung wird anhand des Vorhandenseins einer Berechtigung im Request überprüft. Dazu wird ein XSL-Mapping erstellt, das den Request prüft und auswertet, ob die benötigten Berechtigungen vorhanden sind. Ansonsten muss die Anfrage mit einer entsprechenden Fehlermeldung, z. B. per `reST return` mit dem HTTP-Statuscode 403 `Forbidden` beantwortet werden. Ob die Berechtigung vorhanden ist, kann z. B. mit `<xsl:if>` oder `<xsl:choose>` getestet werden.

#### Beispiel: XSL-Mapping zur Berechtigungsprüfung

```

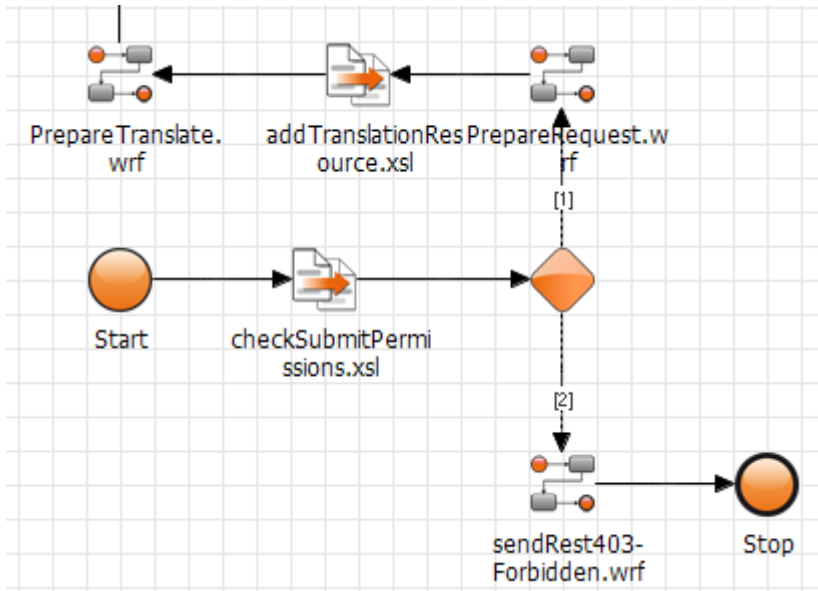
1  <?xml version="1.0" encoding="UTF-8"?><xsl:stylesheet xmlns:xsl="http://
2  www.w3.org/1999/XSL/Transform" version="1.0">
3  <xsl:output method="xml" media-type="text/xml" />
4  <xsl:variable select="/Request/Security/Group[@name='Permissions']/
5  Principal[@module='URM' and @object_type='user' and @action='administer']"
6  name="userAdminister" />
7  <xsl:template match="/">
8  <xsl:choose>
9  <xsl:when test="$userAdminister">
10 <xsl:processing-instruction name="ConditionState">1</
11 xsl:processing-instruction>
12 <xsl:copy-of select="/Request" />
13 </xsl:when>
14 <xsl:otherwise>
15 <xsl:processing-instruction name="ConditionState">0</
16 xsl:processing-instruction>
17 <Data></Data>
18 </xsl:otherwise>
19 </xsl:choose>
20 </xsl:template>
21 </xsl:stylesheet>

```

#### Erläuterung des XSL-Mapping-Beispiels:

- In Zeile 3 wird in diesem Beispiel eine Variable erzeugt, die die Prüfung auf die entsprechende Berechtigung enthält. Es ist wichtig, den kompletten Pfad zur Berechtigung, also `/Request/Security/Group[@name='Permissions']` zu verwenden, um keine Dopplungen o. Ä. zu erhalten. Es wird ein `Principal` gesucht, der im Attribut `module` den Wert `URM`, als `object_type` den Wert `user` und als `action` den Wert `administer` enthält. Der Parameter wird in diesem Fall ignoriert.
- In Zeile 6 wird dies dann in einem `<xsl:when>`-Block ausgewertet und eine entsprechende *Processing Instruction* erzeugt, mit der ein nachgelagerter *Condition*-Prozessbaustein eine entsprechende Weiterverarbeitung oder die Ausgabe einer Fehlermeldung veranlassen kann.

Ein entsprechender Prozess könnte wie folgt aussehen:



#### Erläuterung des Prozess-Beispiels:

- Zweig 1 hinter dem Condition-Baustein leitet eine weitere Verarbeitung ein
- Zweig 2 sendet eine Fehlermeldung mit HTTP-Statuscode 403 Forbidden zurück.

**Auf dieser Seite:**

## 2.12.2 Internationalisierung

*X4 Activities*-Anwendungen müssen oftmals in verschiedenen Sprachen bereitgestellt werden. Eine einfache Möglichkeit zur Internationalisierung von XML-Inhalten in technischen Prozessen sind die Translation-Prozesse für *X4 Activities*. Diese technischen Prozesse nehmen einen beliebigen XML-Input an und übersetzen diesen in die gewünschte Sprachversion, sofern entsprechende Übersetzungsdateien vorhanden sind.

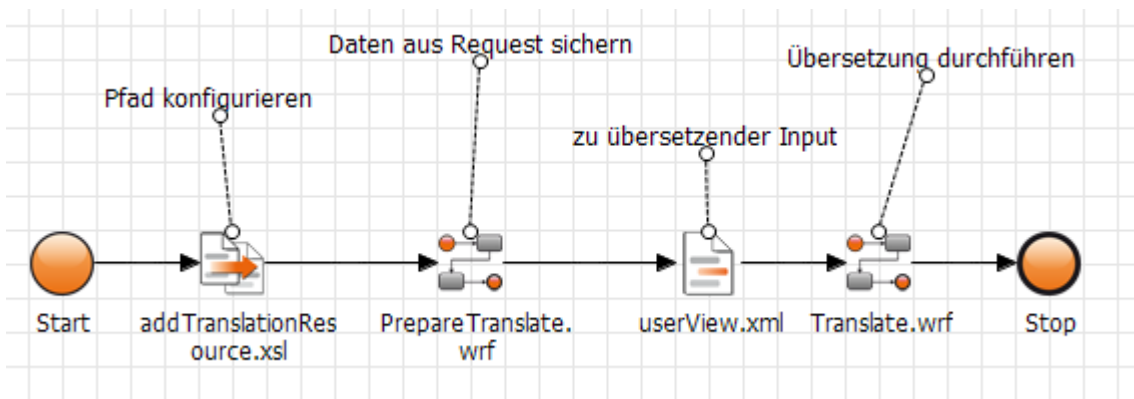
**i** Die technischen Prozesse, die zur Internationalisierung benötigt werden, befinden sich im Prozesspaket *X4Common*. Dieses wird also zwingend zur Benutzung der entsprechenden Prozesse benötigt.

### 2.12.2.1 XML-Inhalte internationalisieren

Um die Übersetzung in einem technischen Prozess einzusetzen, folgende Schritte durchführen:

1. Pfad zur Übersetzungsdatei konfigurieren.
2. Cookies aus dem HTTP-Request speichern (zur Bestimmung der Sprache).
3. Übersetzung durchführen.

Ein entsprechender technischer Prozess kann wie folgt aufgebaut sein:



### 2.12.2.1.1 Schritt 1: Pfad zur Übersetzungsdatei konfigurieren

Die Konfiguration des Pfades erfolgt über ein XSL-Mapping, das eine *Processing Instruction* namens `sp-i18n-resource` in den HTTP-Request einfügt. Der Rest des Requests wird kopiert. Dies dient dazu, den Pfad dynamisch konfigurierbar zu machen. Es ist somit möglich, pro Prozess eine eigene Übersetzungsdatei anzugeben.

Das XSL-Mapping, im obigen Bild `addTranslationResource.xml`, beinhaltet folgenden Code:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/
  Transform">
3    <xsl:output media-type="text/xml" method="xml"/>
4
5    <xsl:template match="/">
6      <xsl:processing-instruction name="sp-i18n-resource">
7        <xsl:text>/URM/Common/translation/urm</xsl:text>
8      </xsl:processing-instruction>
9      <xsl:apply-templates/>
10   </xsl:template>
11
12   <!-- standard copy template -->
13   <xsl:template match="@*|node()">
14     <xsl:copy>
15       <xsl:apply-templates select="@*" />
16       <xsl:apply-templates/>
17     </xsl:copy>
18   </xsl:template>
19 </xsl:stylesheet>
  
```


#### Erläuterungen zum XSL-Mapping:

- In Zeile 5–10 wird der Pfad zur Übersetzungsdatei gesetzt, relativ zum aktuellen *X4DB*-Benutzerordner (der `%USER_COLLECTION%`). Der letzte Teil des Pfades (im Beispiel: `urm`) ist der Präfix des Names der Übersetzungsdatei. Der Dateiname wird im *X4*-Prozess nach dem Muster **<Präfix>\_<Sprachkürzel>.xml** zusammengebaut, z.B. `urm_de.xml` bei einer deutschen Sprachvariante und dem Pfad `/URM/Common/translation/urm_de.xml`.
- In Zeile 13–18 wird der HTTP-Request mit allen Elementen und Attributen kopiert (Identitätstransformation), damit er in späteren *X4*-Prozessen und XSL-Mappings verwendet werden kann.



### 2.12.2.1.2 Schritt 2: Cookies aus dem Request speichern

Im nächsten Schritt werden die Cookies sowie die *Processing Instructions* aus dem HTTP-Request kopiert und in einer Prozessvariable *spi18n* gespeichert.

 Diese Variable darf im Verlauf des Prozesses nicht überschrieben werden, da sie zur Bestimmung der Sprache und des Dateipfads zur Übersetzungsdatei benötigt wird.

Als Ausgabe erfolgt wiederum der Request, der in den Prozess eingelesen wurde.

Diese Schritte werden im Prozess `X4Common/Translation/process/PrepareTranslate.wrf` durchgeführt.

### 2.12.2.1.3 Schritt 3: Übersetzung durchführen

Wurde nun der zu übersetzende Input eingelesen, so wird dieser im Prozess `Translate.wrf` übersetzt. Dieser bestimmt dazu zuerst den Dateipfad zur entsprechenden Übersetzungsdatei anhand der Konfiguration sowie der im Cookie enthaltenen Sprache. Hierfür wird zuerst versucht, die Datei anhand des konfigurierten Pfads und der Abkürzung der Sprache zu finden. Wird die Datei nicht gefunden, so fällt der Prozess automatisch auf die Standardsprache Englisch (Sprachkürzel `en`) zurück. Diese Übersetzungsdatei (im Beispiel `urm_en.xml`) muss also auf jeden Fall existieren, da der Prozess sonst mit einer Fehlermeldung abbricht.

Die Übersetzungsdatei wird im technischen Prozess eingelesen und mit einem entsprechenden XSL-Mapping übersetzt. Die Ausgabe des XSL-Mappings ist dabei immer eine XML-Datei mit dem MIME-Typ `text/xml`. Ist ein andere MIME-Typ gewünscht, so muss dieser mit einem Adapter oder einem weiteren XSL-Mapping gesetzt werden. Dies wäre z.B. für die Bereitstellung einer HTML-Datei für die Darstellung einer Web-Oberfläche notwendig.

## 2.12.2.2 Format der Übersetzungsdatei

Die Datei, welche die Übersetzungen enthält, muss als XML-Repräsentation einer *Java Properties*-Datei vorliegen. Die Übersetzungen bestehen dabei jeweils aus Schlüssel-Wert-Paaren. Für jede unterstützte Sprache muss eine eigene Datei existieren, welche die jeweiligen Übersetzungen für diese Sprache enthält. Die Schlüssel müssen in der jeweils zu übersetzenden Datei verwendet werden.

**Beispiel**

```

1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <!-- <!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd"
   -->
3  <properties>
4      <!-- navigator config -->
5      <entry key="nodeName.users">Benutzer</entry>
6      <entry key="nodeName.groups">Gruppen</entry>
7      <entry key="nodeName.roles">Rollen</entry>
8      <entry key="nodeName.permissions">Berechtigungen</entry>
9      <entry key="nodeName.pwpolicies">Passwortrichtlinien</entry>
10 </properties>

```

Im Beispiel existiert genau ein Element `<properties>`, das wiederum beliebig viele Elemente `<entry>` enthält. Jedes Element `<entry>` besitzt ein Attribut `key`, das den Schlüssel, mit dem der entsprechende Übersetzungsstring referenziert wird, enthält. Der Schlüssel muss innerhalb der Datei eindeutig sein. Element `<entry>` enthält die Übersetzung, im Beispiel für die deutsche Sprachversion.

### 2.12.2.3 Aufbau der zu übersetzenden Datei

Eine Übersetzungsdatei muss den Namespace `sp-i18n` der Übersetzungsfunktion besitzen. Sie muss zudem die Schlüssel aus der Übersetzungsdatei referenzieren. Im folgenden Beispiel wird der Namespace in einer HTML-Struktur gesetzt. Dieser Namespace muss in jeder zu übersetzenden Datei entsprechend konfiguriert werden.

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:sp-i18n="/sp/translation/"></html>
```

**i** Die Übersetzung von Attributen und die Übersetzung von Textinhalten müssen bei der Internationalisierung unterschiedlich behandelt werden.

#### 2.12.2.3.1 Übersetzung von Attributen

Soll ein Attribut übersetzt werden, so wird dieses im Namespace `sp-i18n` definiert. Der Wert des Attributs ist der Schlüssel für die Übersetzungsdatei, anhand dessen die Übersetzung bestimmt wird. Das Attribut wird dann im Standard-Namespace mit dem übersetzten Wert erzeugt.

**Input**

```
<input type="text" name="first_name" id="first_name" sp-
i18n:placeholder="user.first_name.placeholder" />
```

Dieses Beispiel erzeugt ein Element vom Typ `input`, das ein Attribut `placeholder` erhält. Dieses Attribut wird mit der Zeichenkette aus der Übersetzungsdatei ersetzt.

**Ausgabe**

```
<input type="text" name="first_name" id="first_name" placeholder="Vorname des Benutzers" />
```

### 2.12.2.3.2 Übersetzung von Textinhalten

Ein für sich allein stehender Text wird mit einem dem Element `<sp-i18n:text>` (mit dem Namespace `sp-i18n`) übersetzt. Es besitzt lediglich ein Attribut `key`, das dem Schlüssel aus der Übersetzungsdatei entspricht. Das Element wird bei der Übersetzung durch den entsprechenden Textknoten ersetzt.

**Input**

```
<p><sp-i18n:text key="user.description"/><p>
```

Die Ausgabe würde in diesem Beispiel (für eine deutsche Übersetzung) so aussehen:

**Ausgabe**

```
<p>Ermöglicht die Erzeugung eines Benutzers, der zum Login in der Anwendung verwendet werden kann.<p>
```

**Auf dieser Seite:**

## 2.12.3 Globale Suche

Die **globale Suche** wird in der `NavigatorConfig` pro Folder konfiguriert und benötigt angepasste Lister-Prozesse für die Umsetzung der Funktionalität. *X4 Activities* stellt somit nur Oberflächenelemente für eine Suche bereit – die Umsetzung erfolgt fachspezifisch in den Lister-Prozessen.

### 2.12.3.1 Aktivieren der Suchbox

Die Suchbox wird pro Folder im [Navigatorbaum](#) eingeschaltet. Sie wird also je nach ausgewähltem Ordner sichtbar oder verschwindet. Dazu kann ein XML-Element `GlobalSearch` optional in jedem Element einer Navigator-Config definiert werden. Damit die Suche aktiviert ist, muss noch das boolesche Attribut `enabled` mit dem Wert `true` angegeben werden. Innerhalb des `GlobalSearch`-Elements kann optional ein weiteres Element `helpText` definiert werden, um einen Hilfetext für die Suche bereitzustellen, der als Tooltip für das Suchfeld angezeigt wird. In diesem Hilfetext kann beispielsweise beschrieben werden, auf welche Inhalte des Listers sich die Suche bezieht oder ob/wie "Wildcards" verwendet werden können.

**Beispiel**

```
<GlobalSearch enabled="true">
  <helpText>Hier wird beschrieben, wie das Suchfeld im Kontext des aktuellen
  Folders verwendet wird.</helpText>
</GlobalSearch>
```

### 2.12.3.2 Umsetzung der Suche in Listerprozessen

Für die Umsetzung der Suchfunktionalität in einem Lister-Prozess, wird der vom Benutzer eingegebene Such-String als zusätzliches XML-Element in den Filter-Informationen des Prozess-Inputs bereitgestellt. Dieser String kann dann verwendet werden, um eine Suche über mehrere Spalten hinweg durchzuführen. Für welche Spalten es technisch und fachlich sinnvoll ist sie in diese Suche mit einzubeziehen, muss pro Anwendungsfall geprüft werden. Die konkrete Realisierung der Suche ist abhängig von der Technologie, die die Daten für den Lister bereitstellt. Die meisten relationalen Datenbanksysteme stellen beispielsweise Unterstützung für Volltextsuche bei textuellen Spalten bereit, welche für diesen Zweck verwendet werden kann. Ob/wie dabei Wildcards oder Ähnliches verwendet werden können, hängt von der konkreten Implementierung ab und sollte dem Benutzer ggf. mit Hilfe eines Hilfetextes mitgeteilt werden.

Die Ergebnismenge des Listerprozesses darf nur Einträge enthalten, die in mindestens einer Spalte Treffer für den Such-String enthalten. Die [Seiteneinteilung](#) der Prozess-Antwort muss auf Grundlage dieser eingeschränkten Ergebnismenge erfolgen.

**Beispiel**

```
<Filtering>
  <GlobalSearch><![CDATA[Ein Suchbegriff]]></GlobalSearch>
  <!-- Hier folgen ggf. die spaltenspezifischen Filter -->
</Filtering>
```

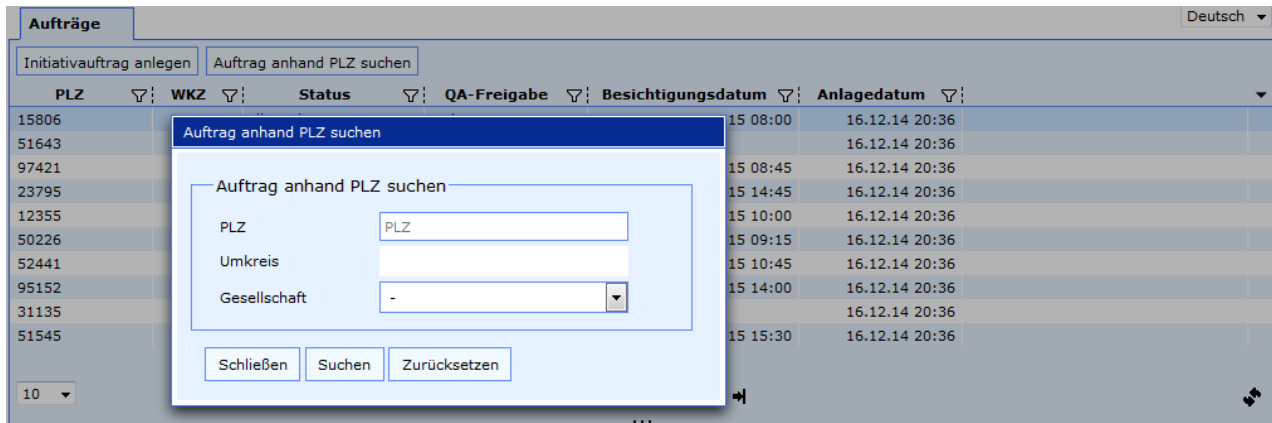
**Auf dieser Seite:****Siehe auch:**

- [Navigator](#)
- [Filtern von Listeneinträgen](#)
- [Datenanfrage für Seiten der Vorgangsliste](#)

### 2.12.4 Dialog

Dialoge einer Activities-Anwendung können entweder über die [Aktionsleiste](#) (`displayType`-Attribut) oder eine [JavaScript API](#) geöffnet werden. Dabei können Dialoge nur Meldungen (Informationen/Fehlermeldungen etc.) oder Detailmasken enthalten.

Detailmasken-Dialoge sind oft mit Listeneinträgen verknüpft und werden verwendet, um Daten eines Eintrags zu bearbeiten. Sie können aber auch unabhängig von der Listenansicht, z. B. in einer [Detail-only-Ansicht](#) angezeigt werden.



#### Siehe auch:

- [JavaScript-API für Dialoge in X4 Activities](#)
- [Aktionsleiste](#)
- [Detail-only-Ansicht](#)

## 2.12.5 Selektion von Listeneinträgen

In der Listenansicht einer Activities-Applikation hat der Anwender die Möglichkeit, Einträge zu selektieren. Zu einem selektierten Eintrag kann eine Detailmaske angezeigt werden. Diese wird in der Antwort auf eine Seitenanfrage pro Eintrag konfiguriert, siehe [Datenanfrage für Seiten der Vorgangsliste](#). Wie eine solche Detailmaske erstellt werden kann, ist im Abschnitt [Detailmasken](#) beschrieben.

Auf selektierten Einträgen können mit Hilfe von Buttons der [Aktionsleiste](#) unterschiedliche Aktionen durchgeführt werden. Wie diese Aktion durchgeführt wird, ist durch das `actionType`-Attribut des entsprechenden Buttons definiert. Mit dem Attribut `displayType` kann zusätzlich angegeben werden, wie das Ergebnis einer Aktion angezeigt wird, z. B. als Dialog oder Tab.

The screenshot displays the 'Rollen' (Roles) management section of the X4ControlCenter. The top navigation bar includes 'BAM', 'Services', 'URM', 'X4 Admin', and 'License'. The left sidebar contains links for 'Benutzer', 'Gruppen', 'Rollen', 'Berechtigungen', and 'Passwortrichtlinien'. The main area shows a table of roles with columns: Name, Metadaten, Anwendung, Letzter Bearbeiter, Zuletzt bearbeitet, Ersteller, and Erstellt am. The 'X4 URM list' role is selected. Below the table, the 'Rolleigenschaften' (Role Properties) section shows details for 'X4 URM list', including its description, technical identifier, and application. The 'Zugewiesene Berechtigungen' (Assigned Permissions) section shows a list of permissions for the 'URM' application, with buttons to assign, remove, or select all permissions.

Name	Metadaten	Anwendung	Letzter Bearbeiter	Zuletzt bearbeitet	Ersteller	Erstellt am
X4 URM administer		URM		31.03.15 08:01		31.03.15 08:01
<b>X4 URM list</b>		<b>URM</b>		<b>31.03.15 08:01</b>		<b>31.03.15 08:01</b>
X4BAM Admin		X4BAM	admin	31.03.15 08:01	admin	31.03.15 08:01
X4BAM User		X4BAM	admin	31.03.15 08:01	admin	31.03.15 08:01

**Rolleigenschaften**

Name: X4 URM list

Beschreibung: Role for listing all users, groups, roles and permissions

Technischer Bezeichner: X4 URM list

Metadaten: Metadaten der Rolle

Anwendung: URM

**Zugewiesene Berechtigungen**

Verfügbar:

- X4 URM Administer Roles
- X4 URM Administer Users
- X4 URM Administer Passwords
- X4 URM Administer Groups
- X4 URM Administer Permissions

Zugewiesen:

- X4 URM List Permissions
- X4 URM List Roles
- X4 URM List Groups
- X4 URM List Users
- X4 URM List Password Policies
- X4 URM List All Modules

Buttons: Zuweisen >, Entfernen <, Alle >>, Alle <<, Speichern

**i** Für die Selektion von mehreren Einträgen, siehe [Mehrfach-Selektion von Listeneinträgen](#).

#### Siehe auch:

- [Datenanfrage für Seiten der Vorgangsliste](#)
- [Detailmasken](#)
- [Aktionsleiste](#)

## 2.12.6 Mehrfach-Selektion von Listeneinträgen

In der Listenansicht einer Activities-Applikation können mehrer Einträge selektiert werden. Dazu muss es mindestens einen Button in der Aktionsleiste geben, dessen Aktion auf mehrere Einträge angewendet werden kann.

Diese Funktion kann über das `actionType`-Attribut (`actionType="MULTI_POST"`) aktiviert werden, siehe [Aktionsleiste](#).

The screenshot shows the X4ControlCenter interface. The left sidebar contains navigation items: Instances, Dashboards, Errors, Messages, Processes, and Scheduler. The main area is titled 'Running instances' and contains a table with columns: Process ID, Process, Techn. state, Prof. state, and Starttime. Three instances are listed, all with a 'PAUSED' technical state and 'WAITING' professional state. Below the table, there is a 'Savepoints' section with a dropdown menu and a 'Restart/Resume' button. A diagram titled 'Create Save Point and halt process execution' illustrates a process flow from 'Start' to 'In' (Reading an XML document), then to 'Out' (Writing an XML document), and finally to 'Stop'. A note explains that a transition with a 'SavePoint' property stops the process execution, which must be continued via X4 BAM. A warning icon indicates that savepoints do not work in debug mode and suggests pressing F9 to execute the process.

Process ID	Process	Techn. state	Prof. state	Starttime
142847387520072047310[1]	x4db:/1/X4Experience/BPM/1-Examples/	PAUSED	WAITING	2015-04-08 09:52
142890532779687986910[1]	x4db:/1/X4Experience/ESB/3-Reference/	PAUSED	WAITING	2015-04-13 13:25
142918939329366397410[1]	x4db:/1/X4Experience/BPM/1-Examples/	PAUSED	WAITING	2015-04-16 15:30

Savepoints: Please select a savepoint. Restart/Resume

SavePoint.wrf

**Create Save Point and halt process execution**

Transition has a SavePoint (specified in property "Savepoint"): Process instance execution stops and must be continued via X4 BAM

Reading an XML document

Writing an XML document

Start In Out Stop

(c) SoftProject GmbH

Savepoints do not work in debug mode! Press e.g. the F9 key to execute the process.

Impressum

#### Siehe auch:

- [Selektion von Listeneinträgen](#)
- [Aktionsleiste](#)

## 2.12.7 Vor-Selektion von Listeneinträgen

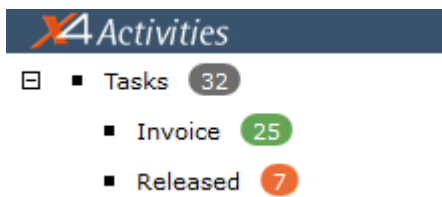
Beim Laden des Listers ist es möglich, automatisch Einträge zu selektieren, sofern noch keine Selektion vorhanden ist. Dies wird in der Seitenantwort konfiguriert (preSelection-Attribut im DataRow-Element), siehe [Datenanfrage für Seiten der Vorgangsliste](#).

#### Siehe auch:

- [Datenanfrage für Seiten der Vorgangsliste](#)

## 2.12.8 Marker im Navigator

Im [Navigator](#) können konfigurierbare Marker in den einzelnen Knoten angezeigt werden. Diese können beispielsweise verwendet werden, um die Anzahl neuer Einträge im Lister darzustellen. Dabei lassen sich die Marker automatisch aktualisieren, siehe [Automatische Navigator-Aktualisierung](#).

**Siehe auch:**

- [Navigator](#)
- [Automatische Navigator-Aktualisierung](#)

## 2.12.9 Automatische Navigator-Aktualisierung

Teile des Navigators können in konfigurierbaren Zeitintervallen automatisch aktualisiert werden, siehe *Automatische Aktualisierung von Knoten* im Abschnitt [Navigator](#). Diese Funktion wird hauptsächlich zur Aktualisierung der [Marker](#) eingesetzt.

**Siehe auch:**

- [Marker im Navigator](#)
- [Navigator](#)

## 2.12.10 Content-only-Applikationen

Eine Applikation kann auch ohne [Navigator](#) dargestellt werden. Dies ist der Fall, wenn in der Navigator Config nur der ROOT-Knoten definiert ist. Der Content-Bereich unterstützt weiterhin die gleichen Inhalte, die auch in der Standard-Ansicht verfügbar sind (z. B. Lister, Detailansicht etc.).





### Siehe auch:

- [Navigator](#)

## 2.12.11 Detail-only-Ansicht

Im Content-Bereich einer Activities-Anwendung kann anstelle der Listenansicht auch nur eine [Detailansicht](#) eingeblendet werden. Dies wird über das `contentType`-Attribut des [Navigators](#) konfiguriert.

**X4ControlCenter** BAM Services URM X4 Admin License ? SP Admin

Benutzer Gruppen Rollen Berechtigungen **Passwortrichtlinien** Passwortrichtlinien testen

**Passwortrichtlinien** Deutsch

**Zeichenkonfiguration**

Minimale Länge: 8

☐ Ziffer erforderlich

☐ Kleinbuchstabe erforderlich

☐ Großbuchstabe erforderlich

☐ Sonderzeichen erforderlich

**Passwortänderungen**

☒ Passwortänderung wird benötigt nach 365 Tagen

Alte Passwörter, von denen sich das Neue unterscheiden muss: 0

**Accountspernung**

Anzahl Fehlversuche bis zur Accountspernung: 10

☒ Automatisch entsperren nach 10 Stunden

☐ Passwortänderung für alle Benutzer erzwingen

Speichern

[Impressum](#)

**Siehe auch:**

- [Navigator](#)
- [Detailmasken](#)

## 2.12.12 Lister-only-Ansicht

Im Content-Bereich einer Activities-Anwendung kann die Listenansicht auch ohne Detailansicht eingeblendet werden. Dies wird über das `contentType`-Attribut des [Navigators](#) konfiguriert.

Prozess ID	Prozess	Techn. Status	Startzeitpunkt	Endzeitpunkt	Dauer
14284738751876058030[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	8
14284738751885567220[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	5
14284738751897823800[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	141
14284738751904261480[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	44
1428473875191035887[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	73
1428473875192368940[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	6
1428473875193788960[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	44
1428473875194462506[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	50
1428473875195250907[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	3
1428473875196906452[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	44
1428473875197605526[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	46
1428473875198494216[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	3
1428473875199933014[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	40
1428473875200720473[0]1	x4db:/1/X4Experience/BPM/1-Examples/	PAUSED	08.04.15 09:52	08.04.15 09:52	893
1428473875201583977[0]1	x4db:/1/X4Experience/BPM/1-Examples/	FINISHED	08.04.15 09:52	08.04.15 09:52	138

## 2.12.13 Offline-Fähigkeit

### 2.12.13.1 Voraussetzungen für die Offline-Fähigkeit

Um eine Web-Anwendung auf Basis von X4 Activities auch im Offline-Modus bereitstellen zu können, müssen folgende Voraussetzungen erfüllt sein:

- es besteht die Möglichkeit, statische Ressourcen im Browser zu cachen
- eine Client-seitige Datenspeicherung im Browser in Form des HTML5 Web Storage ist gegeben

Für das Cachen von statischen Ressourcen, wie beispielsweise HTML-Seiten, Bilder und Skript-Dateien im Web-Browser, bietet die HTML5-Spezifikation *Application Cache*. Dabei handelt es sich um ein Speichermechanismus, mit dem sich statische Ressourcen für den Offline-Betrieb vorrätig halten lassen.

### 2.12.13.2 Zu Cachende Inhalte definieren

Welche Ressourcen vorrätig zu halten sind, kann beim Entwickeln der Web-Anwendung mit Hilfe einer Cache-Manifest-Datei festgelegt werden. Die Cache-Manifest-Datei ist eine einfache Textdatei, in der die URLs der zu cachenden Ressourcen aufgelistet werden. Dabei können die Ressourcen entweder absolut oder relativ zur Manifest-Datei angegeben werden.

Die Manifest-Datei wird üblicherweise direkt in der WAR-Datei der Activities-Anwendung (z.B. <X4-Installationsordner>\Server\wildfly\standalone\deployments\<Activities-Portal.war>) bereitgestellt und ist wie Folgt aufgebaut:


**Beispiel einer Manifest-Datei cache.manifest**

```
CACHE MANIFEST
CACHE:
index.html
css/main.css
img/logo.png
NETWORK:
login.html
FALLBACK:
login.html offline.html
```


**Erläuterungen:**

Die erste Zeile einer Cache-Manifest-Datei muss immer mit `CACHE MANIFEST` beginnen und besteht aus folgenden drei Abschnitten:

- **CACHE:** Hier werden alle Ressourcen aufgelistet, die vom *Application Cache* explizit heruntergeladen und gecached werden sollen.

 Folgen die Ressourcen direkt unter `CACHE MANIFEST`, kann das Schlüsselwort `CACHE` weggelassen werden.

- **NETWORK:** Hier werden alle Ressourcen aufgelistet, die **nie** vom *Application Cache* oder Browser gecached werden dürfen und immer eine Verbindung zum Server benötigen.

 Falls der Benutzer im Offline-Modus ist, wird keine gecachte Version dieser Ressource verwendet. Bei fehlender Internetverbindung wird dem Anwender in diesem Fall eine Fehlermeldung (404 Not Found) angezeigt. Um diese Fehlermeldung zu vermeiden, lassen sich Ersatz-Inhalte definieren.

- **FALLBACK:** Hier werden Ersatz-Inhalte definiert, für den Fall, dass mangels Erreichbarkeit nicht auf eine Ressource zugegriffen werden kann. Jede Zeile besteht deshalb aus zwei URLs. Die zweite URL verweist dabei auf die Ersatz-Ressource, die verwendet werden soll, falls nicht auf die URL der ersten Ressource zugegriffen werden kann.

### 2.12.13.3 Application Cache aktivieren

Um den *Application Cache* für eine Web-Anwendung zu aktivieren, muss im HTML-Element das `manifest`-Attribut mit dem Pfad zur Cache-Manifest-Datei angegeben werden. Diese ist ebenfalls innerhalb der WAR-Datei der Activities-Anwendung (z.B. `<X4-Installationsordner>\Server\wildfly\standalone\deployments\<Activities-Portal.war>`) abgelegt.

```
<!DOCTYPE HTML>
<html manifest="cache.manifest">
<body>
...
</body>
</html>
```

Anschließend muss im Browser explizit bekanntgemacht werden, dass es den content type *text/cache-manifest* gibt, damit die *.manifest*-Datei für den Browser verständlich ausgeliefert wird. Dazu muss in der *.htaccess*-Datei folgende Zeile ergänzt werden:

```
AddType text/cache-manifest .manifest
```

**i** Ist noch keine *.htaccess*-Datei vorhanden, kann diese über einen Texteditor angelegt, als *.htaccess*-Datei gespeichert und auf gleicher Ebene wie die *.manifest*-Datei abgelegt werden.

*Beispiel:* `<X4-`

`Installationsordner>\Server\wildfly\standalone\deployments\<Activities-Portal.war>`

#### 2.12.13.4 Daten Laden und Speichern

Das **UI Widget: Form** bietet verschiedene Methoden für das Laden und Speichern von Daten. Für den Offline-Modus kann ein Controller bereitgestellt werden, dessen Methoden abhängig davon, ob der Offline-Modus aktiv ist oder nicht, automatisch die Standard-API des Form-Widgets oder die Offline-API des Activities Listers für das Laden und Ändern der Formulardaten verwenden.

**Beispiel-Controller für ein Detail-Formular**

```

x4.Namespace('x4.offlineExample');

x4.offlineExample.DetailFormController = x4.Class({

  INIT: function(vc){
    this.vc = vc;
    var self = this;
    vc.on('view-ready', function(){
      self.load(); // auto-load data on form initialization
    });
  },

  PUBLIC: {
    load: function(){
      if(this.listerAPI().isOffline()){
        this.vc.getModel().change(this.listerAPI().getDataRow(this.getId()));
      } else {
        this.vc.getAPI().load();
      }
    },
    save: function(){
      if(this.listerAPI().isOffline()){
        this.listerAPI().setDataRow(this.getId(), this.vc.getModel().get());
        x4.portal.api.updateListerAndCloseDialog();
      } else {
        var resource = this.vc.getAPI().getDataResource();
        resource.save(function(response) {
          if (response.ok()) {
            x4.portal.api.updateListerAndCloseDialog();
          }
        });
      }
    },
    getId: function(){
      return x4.portal.api.getPrimaryKey();
    }
  },

  PRIVATE: {
    listerAPI: function(){
      return x4.portal.api.getMainApp().lister;
    }
  }
});

```

Der Wechsel in den Offline- bzw. Online-Modus findet üblicherweise über eine Schaltfläche statt. Die dafür benötigte JavaScript-Logik kann zum Beispiel wie folgt als UI-Controller spezifiziert werden:

**Beispiel: Wechsel in den Offline- und Online-Modus**

```

x4.Namespace('x4.offlineExample');

x4.offlineExample.MainFormController = x4.Class({

    INIT: function(vc){
        this.vc = vc;
    },

    PUBLIC: {
        offline: function() {
            var self = this;
            var selectedkeys = this.listerAPI().getMultiSelectionKeys();
            x4.rest.sendJSON("/X4/httpstarter/ReST/X4PTest/BaseTests/employees/
offline", selectedkeys, "POST", function(response){
                self.listerAPI().setOfflineData(JSON.parse(response.getText()));
                parent.location.reload();
            });
        },
        synchronize : function() {
            var self = this;
            var dirtyOfflineData = this.listerAPI().getDirtyOfflineData();

            x4.rest.sendJSON("/X4/httpstarter/ReST/X4PTest/BaseTests/employees/
synchronize", dirtyOfflineData, "POST", function(response){
                if(response.ok()){
                    for (var i=0; i<dirtyOfflineData.length; i++ ) {
                        self.listerAPI().setDataRow(dirtyOfflineData[i].id,
dirtyOfflineData[i], false);
                    }
                    self.listerAPI().reload();
                }
                else{
                    self.dialogAPI().openErrorDialog(null, 'Synchronization failed.
Please check your server connection.');
```

#### 2.12.13.5 Lister-API

Mit Hilfe einer API lässt sich der Offline-Modus in X4 Activities-Oberflächen aktivieren und deaktivieren.

Auf die im folgenden beschriebenen Methoden, kann über das globale Objekt `x4p.Lister` zugegriffen werden.



Methode	Beschreibung	Parameter
setOfflineData(data)	<p>Setzt die Daten für den Offline-Modus.</p> <p>Die Schlüssel innerhalb des JSON-Objekts müssen den Lister-Spalten ID's entsprechen. Zusätzlich muss jedes JSON-Objekt den Schlüssel "_detailUri" beinhalten und auf die in der Datasource konfigurierten Detail-URI verweisen.</p> <div><p><b>Beispiel</b></p><pre>var data = [     {         "id": "1",         "firstName": "Thomas",         "lastName": "Schmidt",         "birthday": "1958-12-06",         "gender": "male",         "phoneNumber": "100",         "department": "ceo",         "freelancer": false,         "salary": "7500.95",         "sportActivity": "soccer",         "_detailUri": "/X4/httpstarter/ReST/X4PTest/BaseTests/Resources/ui/other/employee_form.html"     },     {         "id": "3",         "firstName": "Andreas",         "lastName": "Fischer",         "birthday": "1982-03-06",         "gender": "male",         "phoneNumber": "110",         "department": "it",         "freelancer": false,         "salary": "3600.75",         "sportActivity": "biking",         "_detailUri": "/X4/httpstarter/ReST/X4PTest/BaseTests/Resources/ui/other/employee_form.html"     } ];  x4p.lister.setOfflineData(JSON.parse(data));</pre></div>	data: JSON-Array
isOffline()	Gibt true zurück, wenn der Offline-Modus aktiv ist, sonst false.	keine
getOfflineData()	Liefert alle offline Datensätze, die im local Storage gespeichert sind.	keine

Methode	Beschreibung	Parameter
<code>getDirtyOfflineData()</code>	Liefert alle Datensätze, die im Offline-Modus geändert wurden und im Status ("offlineStatus") <i>changed</i> sind.	keine
<code>hasDirtyOfflineData()</code>	Gibt <i>true</i> zurück, wenn im Offline-Modus Datensätze mit dem Status <i>changed</i> vorhanden sind, sonst <i>false</i> .	keine
<code>getDataRow(pk)</code>	Liefert den Datensatz anhand des übergebenen Primary-Key zurück.	pk: Primärschlüssel des Datensatzes
<code>setDataRow(pk, data, isDirty)</code>	Überschreibt den Datensatz im local Storage mit den übergebenen Daten anhand des Primary-Key.  Liefert <i>true</i> zurück, wenn der Datensatz erfolgreich überschrieben wurde, sonst <i>false</i> .	<ul style="list-style-type: none"> <li>pk: Primärschlüssel des Datensatzes</li> <li>data: Daten als JSON-Objekt</li> <li>isDirty: boolean-Flag (optional); <i>false</i> Datensatz wird als synchron markiert (<i>offlineStatus</i> = <i>sync</i>), <i>true</i> Datensatz wird als nicht synchron markiert (<i>offlineStatus</i> = <i>changed</i>)</li> </ul>
<code>removeOfflineData(force)</code>	Gibt <i>true</i> zurück, wenn die Offline-Daten erfolgreich aus dem local Storage gelöscht wurden, sonst <i>false</i> .	<b>force(optional):</b> <ul style="list-style-type: none"> <li><i>true</i>: erzwingt das Löschen der Daten aus dem local Storage, auch wenn sich nicht synchrone Daten im local Storage befinden</li> <li><i>false</i> (default): es wird geprüft, ob sich im Offline-Modus geänderte Daten befinden. Ist dies der Fall, liefert die Methode <i>false</i>, sonst <i>true</i>.</li> </ul>

## 2.13 Globale JavaScript-Methoden

X4 Activities bietet ein JavaScript-API, die im Folgenden beschrieben wird.

### 2.13.1 API für die Lister-Tabelle

Für die Lister-Tabelle wird über `parent.x4p.lister` eine JavaScript-API bereitgestellt. Diese definiert folgende Methoden:

- `update()`: Refresh/Reload der Daten in der Tabelle
- `reload()`: Reload der kompletten Tabelle (auch Spaltenkonfiguration)
- `isMultiSelectionEnabled()`: liefert *true*, wenn Multi-Selection für den Lister aktiviert ist, andernfalls *false*
- `getSelectedKey()`: liefert die ID des im Lister selektierten Eintrags

- `getMultiSelectionKeys()`: liefert die IDs aller selektierten Einträge als Array, wenn Multi-Selection aktiviert ist

```
parent.x4p.lister.update();
```

## 2.13.2 Parameter-Injection für die Lister-Tabelle

Der Methodenaufruf erwartet eine Map mit Key/Value-Paaren (JavaScript Daten-Objekt), wobei alle Values Strings sein müssen. Die Methode löst einen Reload des Listers aus, bei dem die Key/Value-Paare der Map als Query-Parameter in der URL des Requests zum Server geschickt werden. Dort müssen diese von einem geeigneten Lister-Prozess ausgewertet werden.

Dieser Mechanismus ermöglicht beispielsweise komplexere Filter-/Sortierungskriterien in einem Dialog zu definieren, z. B. Sortierung mehrerer Spalten.

### Beispiel für den Aufruf

```
var pMap = {  
    key1: "value1",  
    key2: "value2"  
};  
  
parent.injectListParameters(pMap);
```

**Beispiel für den Zugriff auf einen Wert in einem XSL-Mapping des Lister-Prozesses:**

```
<xsl:value-of select="/Request/Query/Parameter[@name='key1']" />
```

**Auf dieser Seite:**

## 2.14 Bibliotheken für X4 Activities

### 2.14.1 X4Common-Prozessbibliothek für X4 Activities

Das Prozessmodul X4Common stellt eine Reihe von Basisfunktionen für die Prozessentwicklung für X4 Activities-Anwendungen bereit.

#### 2.14.1.1 X4 ReST-Hilfsprozesse

Pfad: /Common/process

Prozess	Beschreibung
<i>PrepareRequest.wrf</i>	Dekodiert ein Base64-kodiertes <body>-Element im Request-Dokument, und liefert das gesamte Dokument mit Klartext-Body zurück.
<i>sendRest200-Ok.wrf</i>	Kodiert das Input-Dokument als Base64, umgibt es mit einem <Data>-Element und erzeugt eine entsprechende http-Response-Struktur. Caching wird unterdrückt. Es wird kein Content-type Header gesetzt.  HTTP-Status ist 200.
<i>sendRest200-Ok- application-json.wrf</i>	Siehe <i>sendRest200-Ok.wrf</i> , nur mit Header <Header name="Content-type">application/json</Header>
<i>sendRest200-Ok- application-octet.wrf</i>	Siehe <i>sendRest200-Ok.wrf</i> , nur mit Header <Header name="Content-type">application/octet</Header>
<i>sendRest200-Ok- application-pdf.wrf</i>	Siehe <i>sendRest200-Ok.wrf</i> , nur mit Header <Header name="Content-type">application/pdf</Header>
<i>sendRest200-Ok- html.wrf</i>	Siehe <i>sendRest200-Ok.wrf</i> , nur mit Header <Header name="Content-type">text/html</Header>
<i>sendRest204- NoData.wrf</i>	Sendet HTTP-Status 204 ohne Daten.
<i>sendRest400- BadRequest.wrf</i>	Kodiert das Input-Dokument als Base64, umgibt es mit einem <Data>-Element und erzeugt eine entsprechende http-Response-Struktur. Caching wird unterdrückt. Es wird kein Content-type Header gesetzt.  HTTP-Status ist 400.
<i>sendRest401- Authenticate.wrf</i>	Sendet HTTP-Code 401 mit WWW-Authenticate-Header.
<i>sendRest403- Forbidden.wrf</i>	Sendet HTTP-Code 403.
<i>sendRest403- Forbidden-html.wrf</i>	Sendet HTML-Seite mit Inhalt <i>Zugriff verboten</i> in Englisch (HTML 403) als Ersatz für JBoss 403-Seite.  Sendet HTTP-Code 200.
<i>sendRest404- NotFound.wrf</i>	Sendet HTTP-Code 404.
<i>sendRest404- NotFound-html.wrf</i>	Sendet HTML-Seite mit Inhalt <i>Seite nicht gefunden</i> in Englisch (HTML 404) als Ersatz für JBoss 403-Seite.  Sendet HTTP-Code 200.

Prozess	Beschreibung
<i>sendRest409- Conflict.wrf</i>	Kodiert das Input-Dokument als Base64, umgibt es mit einem <code>&lt;Data&gt;</code> -Element und erzeugt eine entsprechende <code>http-Response-Struktur</code> . Caching wird unterdrückt. Es wird kein <code>Content-type Header</code> gesetzt.  HTTP-Status ist 409.
<i>sendRest500.wrf</i>	Sendet HTTP-Code 500.

### 2.14.1.2 JSON-Konvertierung und -Fehlerbehandlung

Pfad: /Common/process

Prozess	Beschreibung
<i>Convert JSONToXML.wrf</i>	Konvertiert ein JSON-Dokument in ein XML-Dokument.  Der technische Prozess erwartet als Input ein ReST-Request, bei dem das <code>&lt;body&gt;</code> -Element Base64-kodiert wurde und das JSON enthält. Das Input wird nach XML gewandelt und in eine Variable <i>jsonData</i> abgelegt. Der technische Prozess gibt das XML-Dokument als Ergebnis zurück.
<i>Convert ToJSONandSend.wrf</i>	Konvertiert ein beliebiges XML-Dokument als Input in ein JSON-Dokument. Das Ergebnis wird Base64-kodiert, in ein <code>&lt;data&gt;</code> -Element eingefügt und mit einer <code>http-Response-Struktur</code> versehen, die den korrekten Mime-Type <code>application/json</code> setzt.
<i>PrepareErrors.wrf</i>	Versieht ein beliebiges Input-Dokument mit einem XML-Element <code>&lt;ERROR&gt;</code> und verpackt das Ergebnis in eine XML-Struktur, die nach JSON gewandelt werden kann.

### 2.14.1.3 Generischer E-Mail-Versand

Pfad: /Common/process

Prozess	Beschreibung
<i>SendMail.wrf</i>	<p>Versendet eine E-Mail-Nachricht.</p> <p>Die Input-XML-Struktur enthält den E-Mail-Body und dynamische Parameter für den Versand.</p> <div> <p><b>Beispiel</b></p> <pre> &lt;Mail&gt;   &lt;Parameters&gt;     &lt;to&gt;dummy@sp.de&lt;/to&gt;     &lt;sender&gt;x4test@softproject.de&lt;/sender&gt;     &lt;from&gt;x4test@softproject.de&lt;/from&gt;     &lt;login&gt;dummy@sp.de&lt;/login&gt;     &lt;password&gt;*****&lt;/password&gt;     &lt;smtp&gt;smtp.lund1.de&lt;/smtp&gt;     &lt;smtpPort /&gt;     &lt;subject&gt;       Ihr Passwort wurde geändert.     &lt;/subject&gt;     &lt;body&gt;       Es wurde ein neues Passwort für Sie erstellt.        Ihr X4 Portal Team     &lt;/body&gt;   &lt;/Parameters&gt; &lt;/Mail&gt; </pre> </div>

#### 2.14.1.4 Translation (Internationalisierung)

Für eine Beschreibung des Translation-Subsystems siehe [Internationalisierung](#).

### 2.14.2 JavaScript-API für Dialoge in X4 Activities

Diese Seite beschreibt die Dialog-API von Activities für Detail-Seiten und deren Verwendung. Dabei wird sowohl das Öffnen von Detailmasken in Dialogen, als auch das Öffnen von Dialogen aus Detailmasken behandelt.

#### 2.14.2.1 Dialog-API

Für das Öffnen und Schließen von GWT-Dialogen aus Detailmasken stellt *Activities* eine JavaScript-API bereit. Alle Methoden der API sind im `window`-Objekt der Hauptfensters unter `window.x4p.dialogs` verfügbar. Jede Methode, die einen Dialog öffnet, liefert als Rückgabewert eine ID. Diese ID wird benötigt, um den Dialog wieder schließen zu können.

#### 2.14.2.1.1 Methode `openHtmlDialog()`

Öffnet einen Dialog, der die angegebene Seite in einem `IFrame` enthält. Kann auch zum Anzeigen von Detail-Ansichten verwendet werden. Dabei wird der gleiche Dialog verwendet, der auch für die GWT-Dialoge der [Aktionsleiste](#) zum Einsatz kommt.

*Parameter:*

- `caption`: Titel des Dialogs (erforderlich)
- `url`: URL der Seite, die in dem Dialog angezeigt werden soll (erforderlich)
- `options`: Ein JavaScript-Objekt, das folgende optionale Attribute enthalten kann, um den Dialog zu konfigurieren (optional):

Name	Beschreibung	Default
<code>width</code>	Breite des Dialogs in Pixel als Zeichenkette, die eine Ganzzahl enthält <i>Beispiel: width: "800px"</i>	480
<code>height</code>	Höhe des Dialogs in Pixel als Zeichenkette, die eine Ganzzahl enthält <i>Beispiel: height: "550px"</i>	300
<code>modal</code>	Modal( <code>true</code> ) oder nicht modal( <code>false</code> )	<code>false</code>

#### 2.14.2.1.2 Methode `openInfoDialog()`

Öffnet einen einfachen modalen Dialog, zur Anzeige einer Info-Meldung.

*Parameter:*

- `caption`: Titel des Dialogs
- `message`: Meldung, die im Dialog angezeigt werden soll

#### 2.14.2.1.3 Methode `openErrorDialog()`

Öffnet einen einfachen modalen Dialog, zur Anzeige einer Fehler-Meldung.

*Parameter:*

- `caption`: Titel des Dialogs
- `message`: Meldung, die im Dialog angezeigt werden soll

#### 2.14.2.1.4 Methode `openConfirmDialog()`

Öffnet einen einfachen modalen Dialog, der eine Text-Meldung enthält (i.d.R. eine Frage) und darunter zwei Buttons ("Ok" und "Abbrechen") zur Verfügung stellt.

*Parameter:*


- `caption`: Titel des Dialogs
- `message`: Meldung, die im Dialog angezeigt werden soll


- `callback`: Eine Funktion, die nach dem Klick des Benutzers auf einen Button aufgerufen wird und einen Parameter definieren muss, welcher entweder `true` ("Ok") oder `false` ("Abbrechen") ist.

#### 2.14.2.1.5 Methode `closeDialog()`

Schließt einen beliebigen Dialog mit Hilfe dessen ID. Der Rückgabewert ist `true` wenn der Dialog geschlossen wurde und `false` wenn kein Dialog mit der angegebenen ID gefunden wurde.

*Parameter:* `id`: Die ID des Dialogs

 Der `caption`-Parameter von *Info*-, *Error*- und *Confirm*-Dialogen darf auch `null` sein. In diesem Fall werden internationalisierte Standard-Titel entsprechend dem Dialog-Typ verwendet.

 Die Methode `closeDialog()` kann auch zum Schließen von GWT-Tabs verwendet werden.

#### 2.14.2.2 Öffnen von Detailmasken in Dialogen

Um eine Detailmaske in einem Dialog zu öffnen, kann die ReST-URL der Detailmaske an die Methode `openHtmlDialog()` übergeben werden. Alternativ kann der Dialog auch mit Hilfe der [Aktionsleiste](#) (`GWT_DIALOG/MODAL_GWT_DIALOG` als `displayType`) erzeugt werden. Damit der Dialog aus der Detailmaske geschlossen werden kann, wird automatisch die ID des Dialogs als GET-Parameter (`x4pDialogId`) in die URL eingefügt.

##### Öffnen des Dialogs

```
var dialogs = window.parent.x4p.dialogs;
dialogs.openHtmlDialog('Eine Seite', '/X4/httpstarter/ReST/AnyProcess');
```

##### Schließen des Dialogs (Auszug aus XSL-Mapping eines X4-Prozesses)

```
var dialogs = window.parent.x4p.dialogs;
dialogs.closeDialog(<xsl:value-of select="/Request/Query/
Parameter[@name='x4pDialogId']" />);
```



### 2.14.2.3 Öffnen von Dialogen aus Detailmasken

#### Beispiel 1 (Info-Dialog)

```
var dialogs = window.parent.x4p.dialogs;
dialogs.openInfoDialog(null, 'Das ist eine Information.');
```

#### Beispiel 2 (Confirm-Dialog)

```
var dialogs = window.parent.x4p.dialogs;
dialogs.openConfirmDialog(null, 'Wollen Sie diese Aktion wirklich ausführen?',
function(ok){
    if(ok){
        doSomething();
    } else {
        // Code der beim Abbrechen ausgeführt werden soll
    }
});
```

#### Beispiel 3 (HTML-Dialog)

```
var dialogs = window.parent.x4p.dialogs;
var id = dialogs.openHtmlDialog('Eine Seite', 'http://example.org/', {
    width: "500px",
    height: "300px",
    modal: true
});
function closeDialog() {
    if(dialogs.closeDialog(id)){
        dialogs.openInfoDialog('Erfolgreich geschlossen', 'Der Dialog wurde
erfolgreich geschlossen.');
```

Auf dieser Seite:

## 2.14.3 JavaScript-API für HTML-Tooltips in X4 Activities

### 2.14.3.1 Tooltip-Objekt

Für Tooltips wird im `X4PSupport.war (/X4PSupport/js-libs/sp_utils/htmlTooltip.js)` der JavaScript-Konstruktor `X4HTMLTooltip` bereitgestellt. Damit erzeugte Tooltips werden automatisch relativ zu einem definierten DOM-Element positioniert. Die Positionierung erfolgt in einer vorgegebenen Richtung unter Berücksichtigung des Tooltip-Inhalts. Innerhalb eines solchen Tooltips kann eine einzelne Text-Meldung, eine Liste von Meldungen oder auch beliebiger HTML-Code dargestellt werden.

#### 2.14.3.1.1 Konstruktor

Der Konstruktor für die Tooltip-Objekte erwartet folgende Parameter:

- **targetElement**: Das Zielelement des Tooltips als jQuery-Objekt. Relativ zu diesem Element wird die Position des Tooltips berechnet. Außerdem werden implizite Trigger-Aktionen auf diesem Element definiert.
- **options**: Optionales Konfigurationsobjekt. Die möglichen Attribute sind in folgenden Tabelle aufgelistet.

Attribut	Beschreibung	Mögliche Werte	Default
<i>width</i>	Breite des Tooltips in Pixel.	Positive Ganzzahl	250
<i>spacing</i>	Abstand zwischen Zielelement und Tooltip in Pixel.	Positive Ganzzahl	0
<i>placement</i>	Richtung der Positionierung ausgehend vom Zielelement.	"left", "right", "top", "bottom"	"right"
<i>triggerAction</i>	Aktion, durch die der Tooltip ein-/ausgeblendet wird. Dies kann ein Klick oder Hover-Effekt sein. Bei <code>null</code> wird der Tooltip nicht implizit angezeigt, sondern muss explizit über die dafür bereitgestellten Methoden ein-/ausgeblendet werden (s.u.).	"click", null, "mousemove",	null
<i>cssClass</i>	Name einer CSS-Klasse, die direkt auf dem Tooltip-Element definiert wird. Dadurch kann der Tooltip beliebig gestylt werden. Wird das Attribut nicht oder als <code>null</code> definiert, werden Default-Inline-Styles auf dem Tooltip definiert. Einzelne CSS-Angaben können stattdessen auch dynamisch mit Hilfe der <code>css()</code> -Methode (s.u.) überschrieben werden.	beliebiger CSS-Klassenname oder null	null

### 2.14.3.1.2 Methoden

- `show()`: Blendet den Tooltip ein

**Parameter:**

Name	Beschreibung	Modus
content	<ul style="list-style-type: none"> <li>• Inhalt des Tooltips (vgl. <code>setContent()</code>).</li> <li>• <code>null</code>, wenn der Content erhalten werden und dennoch eine Farbe übergeben werden soll</li> </ul>	optional
color	Text-Farbe für den Dialog als CSS-Farb-String	optional

- `success()`: Blendet den Tooltip ein und setzt die Schriftfarbe auf grün

**Parameter:**

Name	Beschreibung	Modus
content	Inhalt des Tooltips (vgl. <code>setContent()</code> ).	optional

- `error()`: Blendet den Tooltip ein und setzt die Schriftfarbe auf rot

**Parameter:**

Name	Beschreibung	Modus
content	Inhalt des Tooltips (vgl. <code>setContent()</code> ).	optional

- `hide()`: Blendet den Tooltip aus
- `toggle()`: Wechselt die Sichtbarkeit des Tooltips
- `setContent()`: Fügt Inhalt in den Tooltip ein

**Parameter:**

Name	Beschreibung	Modus
content	Einfache Text-Meldung, beliebiger HTML-Code oder eine Liste (Array) von Meldungen	erforderlich

- `setMessages()`: Fügt eine Liste (Array) von Meldungen in den Tooltip ein

**Parameter:**

Name	Beschreibung	Modus
msgs	Das Array	erforderlich

- `addMessages()`: Fügt eine Liste (Array) von Meldungen in den Tooltip sein, ohne bereits vorhandene Inhalte zu entfernen

**Parameter:**

Name	Beschreibung	Modus
msgs	Das Array	erforderlich

- `css()`: Setzt ein CSS-Attribut direkt auf dem Tooltip-Element

**Parameter:**

Name	Beschreibung	Modus
attrName	Name des Attributs	erforderlich
value	Wert des Attributs	erforderlich

- `clear()`: Entfernt den Inhalt des Tooltips
- `remove()`: Entfernt den Tooltip aus dem DOM. Nach Aufruf dieser Methode können keine anderen Aktionen mehr auf dem Tooltip durchgeführt werden.
- `computeAndSetPosition()`: Berechnet die Position des Tooltips anhand seiner Größe sowie der Größe und Position des Zielelements. Ein direkter Aufruf dieser Methode ist in der Regel nicht notwendig, da diese implizit von den relevanten Methoden des Objekts aufgerufen wird.

## 2.14.3.1.3 Beispiel

**Beispiel für die Verwendung eines Tool**

```
// Tooltip, der automatisch bei einem Klick auf das Element mit der ID
// "triggerElement" ein-/ausgeblendet wird
var tooltip = new X4PhtmlTooltip($('#triggerElement'), {
    spacing: 10,
    triggerAction: 'click',
    cssClass: 'my-custom-tooltip'
});
tooltip.setContent(['Info 1', 'Info 2']);
// Tooltip, der Standard-Options verwendet und explizit bei Eintritt einer
// bestimmten Bedingung eingeblendet wird
var tooltip2 = new X4PhtmlTooltip($('#anElement'));
function showOrHideTooltip() {
    if(/* beliebige Bedingung */){
        tooltip2.show('Das ist eine Information.');
```

Auf dieser Seite:

## 2.14.4 JavaScript-API für den Loading Indicator in X4 Activities

Für die Einbindung der Ladeanzeige in ein iFrame oder in ein von *X4 Activities* geöffnetes Detail-Fenster stellt `X4PSupport.war` den JavaScript-Konstruktor `X4PLoadingIndicator` bereit.

X4PLoadingIndicator kann aus einem *X4 Activities*-Modul über die URL `/X4PSupport/js-libs/sp_utils/loadingIndicator.js` eingebunden werden. Eine Ladeanzeige, die durch ein solches Objekt erzeugt wird, besteht immer aus der animierten Grafik und einem Hintergrund-`<div>`-Element.

Der Konstruktor von X4PLoadingIndicator definiert folgende optionale Parameter:

Parameter	Beschreibung	Modus
parent	Erwartet ein jQuery-Objekt, in welches die Ladeanzeige eingefügt wird.  Wird kein Parameter oder <i>null</i> übergeben, wird der Body des Dokuments verwendet.	optional
fixed	Boolescher Parameter, der definiert, ob die CSS-Position "fixed" ( <i>true</i> ) oder "absolute" ( <i>false</i> ) ist; Standard ist <i>true</i> .	optional
bgStyleClass	Erwartet eine CSS-Klasse, die das Styling des Hintergrund- <code>&lt;div&gt;</code> -Elements festlegt.  Wenn dieser Parameter fehlt, ist der Hintergrund dunkelgrau-halbtransparent.	optional

### Beispiele

```
// Standardeinstellungen
var li = new X4PLoadingIndicator();

// Absolut positionierte Ladeanzeige, die in das Element mit der ID "wrapper"
// eingefügt wird
// und die CSS-Klasse "myBackgroundStyle" für das Hintergrund-<div>
li = new X4PLoadingIndicator($("#wrapper"), false, "myBackgroundStyle");


// Loading Indicator starten
li.start();

// Hier wird auf irgendetwas gewartet

// Loading Indicator beenden
li.stop();
```

Ein mit diesem Konstrukt erzeugtes Objekt stellt folgende Methoden bereit:

Methode	Beschreibung
<i>start(delay)</i>	<p>Ladeanzeige einblenden und Animation starten. Mit dem optionalen Parameter <i>delay</i> kann das Starten der Ladeanzeige verzögert werden (Angabe in Millisekunden). Damit kann die Anzeige des Loading Indicators für kurze Ladevorgänge verhindert werden.</p> <p>Wenn die Ladeanzeige bereits aktiv ist, wird der Aufruf ignoriert.</p>
<i>stop()</i>	<p>Ladeanzeige ausblenden und die Animation stoppen.</p> <p>Wenn die Ladeanzeige nicht aktiv ist, wird der Aufruf ignoriert.</p>
<i>isRunning()</i>	Gibt <code>true</code> zurück, wenn die Ladeanzeige aktiv ist, sonst <code>false</code> .
<i>getId()</i>	Gibt die ID des <code>&lt;div&gt;</code> -Elements zurück, welches das <code>&lt;img&gt;</code> -Element der Ladeanzeige enthält.
<i>setId(id)</i>	<p>Setzt die ID des <code>&lt;div&gt;</code>-Elements, welches das <code>&lt;img&gt;</code>-Element der Ladeanzeige enthält.</p> <p>Wird nur benötigt, wenn mehrere Ladeanzeigen gleichzeitig in demselben Dokument verwendet werden. Andernfalls wird die Standard-ID <code>X4PLoadingIndicator</code> verwendet. Dies kann bei Bedarf auch geändert werden, während die Ladeanzeige aktiv ist.</p>
<i>setFrameProviderWindow(window)</i>	<p>Erwartet das <code>window</code>-Objekt einer <i>X4 Activities</i>-Hauptapplikation.</p> <p>Wird nur benötigt, wenn dies nicht <code>window.parent</code>, <code>window.top</code> oder <code>window.opener</code> ist. Diese Methode darf nach dem ersten Aufruf von <code>start()</code> nicht mehr aufgerufen werden.</p>

 Auf Wunsch kann eine individuelle Loading-Indicator-Grafik in Ihre Web-Applikation integriert werden.

### 2.14.5 JavaScript-API für den X4 Web Designer

Erfahren Sie, wie Sie die JavaScript-Bibliotheken aus dem X4-Modul *X4WebResources* verwenden. Das *X4WebResources*-Modul ist standardmäßig im Installationspaket `wildfly9_with_activities.zip` der X4 Suite enthalten und vorkonfiguriert. Die darin enthaltenen JavaScript-Dateien sind über die folgende Basis-URL abrufbar. Die Datei-Referenzen auf den folgenden Seiten werden relativ zu dieser Basis-URL angegeben.

Relative Basis-URL des Moduls: `/X4/httpstarter/ReST/X4WebResources/core/js/`

Im *X4WebResources*-Modul sind folgende Dateien enthalten:

- `core.js`, siehe [Objektorientierte JavaScript-API \(core.js\)](#)
- `model/model.js`, siehe [Data Model API \(model/model.js\)](#)
- `async/async.js`, siehe [Asynchrone Aktionen und ReST \(async/async.js\)](#)
- `ui/main.js`, siehe [UI-Framework für den X4 Web Designer](#)
- `xml/xml.js`, siehe [XML-Parser/Serializer \(xml/xml.js\)](#)
- `main.js`: Vereinigung der oben stehenden Dateien

### 2.14.5.1 Objektorientierte JavaScript-API (core.js)

Die Javascript-Datei `core.js` ist Bestandteil der JavaScript-API für den *X4 Web Designer* und enthält eine API für die objektorientierte Arbeitsweise mit JavaScript sowie grundlegende Hilfsmethoden. Mit der objektorientierten API (im Folgenden *OO-API* genannt) lassen sich typische Objektorientierungs-Konzepte nach dem Vorbild Java annähernd in nativem JavaScript-Code abbilden. Dazu wurden entsprechende Konfigurations-Objekte in native JavaScript-Konstruktoren überführt, die im folgenden als *Klassen* bezeichnet werden. Die dabei erzeugten Klassen können Sie mit bestehenden nativen Bibliotheken kombinieren. So kann beispielsweise eine solche Klasse ohne Anpassungen Eigenschaften von einem "normalen" JavaScript-Konstruktor erben und umgekehrt.

#### 2.14.5.1.1 Namensräume (Namespaces)

Diese Methode erzeugt eine JavaScript-Objekt-Hierarchie, falls diese noch nicht existiert, und gibt das innerste Objekt zurück.

`x4.Namespace(reference: string [, baseObj: Object]): Object`

Parameter	Erläuterung
<code>reference</code>	JavaScript-Referenz, d.h. durch Punkte getrennte Bezeichner
<code>baseObj</code>	Basis-Objekt, relativ zu welchem die Referenz aufgelöst wird <ul style="list-style-type: none"><li>• Im Kontext des Skripts ist dies standardmäßig <code>self</code></li><li>• Bei Browsern ist dies standardmäßig <code>window</code>.</li></ul>

#### Beispiel

```
x4.Namespace('my.test.namespace');  
  
my.test.namespace.Test = something;
```

#### Beispiel 2

```
var ns = x4.Namespace('my.test.namespace', base);  
  
ns === base.my.test.namespace; // => true
```

#### 2.14.5.1.2 Klassen

Diese Methode erzeugt einen JavaScript-Konstruktor auf Basis eines Konfigurationsobjektes.

```
x4.Class(conf: <ClassConfig>): <class>
```

**ClassConfig**

Attribut	Erläuterung
EXTENDS	<p>Erwartet eine Klasse (Super-Klasse), von der die neue Klasse erben soll</p> <p><i>Standard:</i> Keine Super-Klasse</p>
INNER_CLASS_NAMES	<p>Ein Array, das die Namen aller inneren Klassen (siehe unten) definiert.</p> <p>Wenn dieses Attribut fehlt oder das Array leer ist, können keine inneren Klassen für diese Klasse erzeugt werden.</p>
INIT	<p>Konstruktor-Methode für eine Instanz der Klasse</p> <p>Die Methode erhält alle Parameter, die dem nativen Konstruktor übergeben werden.</p> <p><i>Standard:</i> Die Instanz wird ohne expliziten Konstruktor initialisiert und alle Parameter werden an den Konstruktor der Super-Klasse (falls vorhanden) weitergereicht.</p>
PUBLIC	<p>Objekt, das alle öffentlichen Methoden der Klasse definiert.</p> <p>Der Methodenname entspricht dem Attributnamen.</p>
INNER	<p>Objekt, das die <i>inneren</i> Methoden der Klasse definiert</p> <p>Die inneren Methoden sind nur innerhalb der Klasse und in inneren Klassen (über die <code>this.__outer</code>-Referenz) sichtbar.</p>
PRIVATE	<p>Objekt, das alle privaten Methoden der Klasse definiert</p> <p>Der Methodenname entspricht dem Namen des Attributs.</p>



**Beispiel**

```
x4.Namespace('my.test');

my.test.AnyType = x4.Class({
  INIT: function(a, b){
    this.a = a;

    this.setB(b);
  },

  PUBLIC: {
    /** Get or set b */
    foo: function(b){
      if('undefined' == typeof b){
        return this.getB(); // or this.b
      }
      this.setB(b);
    }
  },

  PRIVATE: {
    getB: function(){
      return this.b;
    },
    setB: function(b){
      this.b = b;
    }
  }
});

var obj = new my.app.AnyType(a, b);
obj.foo(); // => b
obj.b; // => undefined
obj.getB(); // error
```

### 2.14.5.1.3 Singletons

Diese Methode funktioniert analog zu Klassen (siehe `x4.Class`), jedoch existiert nur eine Instanz der Klasse und diese wird direkt erzeugt und zurückgegeben.

```
x4.Singleton(conf: <ClassConfig>): <instance>
```

**Beispiel**

```
my.test.MySingleton = x4.Singleton({
  PUBLIC: {
    foo: function(){
      // do something
    }
  }
});

my.test.MySingleton.foo();
```

#### 2.14.5.1.4 Innere Klassen

Mit dieser Methode können Sie Klassen definieren, die nur innerhalb einer bestimmten übergeordneten Klasse instanziiert werden können. Dies ermöglicht, Funktionalitäten aus einer Klasse auszulagern, ohne dass öffentliche Schnittstellen davon betroffen sind. Dadurch lässt sich der Quellcode übersichtlicher strukturieren.

```
<class>.__InnerClass(qualifiedName: String, conf: <ClassConfig>)
```

**Beispiel**

```
my.app.OuterClass = x4.Class({
  INNER_CLASS_NAMES: ['Parser'],

  PUBLIC: {
    parseValueByInnerClass: function(value, options){
      var parser = this.__new('Parser', [options]);

      parser.parse(value);
    },

    getLatestResult: function(){
      return this.latestResult;
    }
  },

  INNER: { // only visible in this class & inner classes
    setLatestResult: function(result){
      this.latestResult = result;
    }
  }
});

my.app.OuterClass.__InnerClass('Parser', {
  INIT: function(options){
    this.__outer(); // init this.__outer reference

    this.options = options;
  },

  PUBLIC: { // only visible in this class & outer class
    parse: function(value){
      var result = this.doParse(value);

      this.__outer.setLatestResult(result);
    }
  },

  PRIVATE: { // only visible in this class
    doParse: function(value){
      // parse value using this.options
      // and return result
    }
  }
});

// Um in 'Parser' eine Innere Klasse 'Helper' zu definieren:
// my.app.OuterClass.__InnerClass('Parser.Helper', {...});
```

### 2.14.5.1.5 Vererbung

Mit dem EXTENDS-Attribut können Sie es einer Klasse ermöglichen, eine bestehende Klasse zu erweitern. Die zu erweiternde Klasse kann ein beliebiger JavaScript-Konstruktor sein, sofern dessen relevante Methoden korrekt im Prototyp definiert sind.

Rufen Sie dazu im Konstruktor der Klasse über `this.__super` den Konstruktor der Super-Klasse auf und übergeben ggf. benötigte Parameter. Anschließend können Sie über `this.__super` auf die Super-Instanz zugreifen.

Sie können Methoden der Super-Instanz auch direkt mit `this.<Methodenname>` aufrufen, sofern diese nicht in der untergeordneten Instanz überschrieben wurden.

#### Beispiel

```
my.app.AnyType = x4.Class({
  EXTENDS: AnySuperType,

  INIT: function(a, b){
    this.__super(a); // Aufruf des Super-Konstruktors

    // Call super method doSomething
    // -> same as this.__super.doSomething()
    this.doSomething();
    // Call own method doSomethingWithB
    this.doSomethingWithB(b);
  },

  PUBLIC: {
    doSomethingWithB: function(){
      alert(b);

      // call super method
      this.__super.doSomethingWithB(b);
    }
  }
});
```

#### Auf dieser Seite:

#### Siehe auch:

- [Data Model API \(model/model.js\)](#)
- [Asynchrone Aktionen und ReST \(async/async.js\)](#)
- [UI-Framework für den X4 Web Designer](#)
- [XML-Parser/Serializer \(xml/xml.js\)](#)

### 2.14.5.2 Data Model API(model/model.js)

Die Javascript-Datei `model/model.js` ist Bestandteil der JavaScript-API für den *X4 Web Designer* und enthält eine API für *Data Models*. Data Models unterstützen Sie bei der Verwaltung von JavaScript-Datenstrukturen.

#### 2.14.5.2.1 Model-Objekt

Ein Model-Objekt repräsentiert ein natives Daten-Objekt, -Array oder einen primitiven Wert und stellt Methoden für den Zugriff und die Manipulation der Daten sowie die Benachrichtigung über Änderungen bereit.

#### 2.14.5.2.2 Model-Referenzen

Einige Methoden von Model-Instanzen arbeiten mit Model-Referenzen (beispielsweise `getRef`, `find`, `findModel`). Eine Referenz ist eine Zeichenkette (String), bestehend aus durch Punkten getrennte Bezeichner. Als Bezeichner sind gültige JavaScript-Bezeichner und positive Ganzzahlen (für Array-Indizes) erlaubt.

Folgende Meta-Bezeichner werden zusätzlich unterstützt:

Bezeichner	Erläuterung
<code>#root</code>	Verweis auf das Wurzel- <i>Model</i>
<code>#parent</code>	Nächstes übergeordnete <i>Model</i>
<code>#self</code>	Aktuelles <i>Model</i> , wenn eine Referenz erforderlich, aber kein Kontext-Wechsel erwünscht ist

#### 2.14.5.2.3 Grundlegende API-Funktionen

Hilfsmethoden, die unabhängig von einer bestimmten Instanz verwendet werden können

##### 2.14.5.2.3.1 create

Diese Methode erzeugt eine *Model*-Instanz.

`x4.model.create([data]): <Model>`

Parameter	Erläuterung
<code>data</code>	Ein natives Daten-Objekt, -Array oder ein primitiver Wert, mit dem das Model initialisiert werden soll  Wenn der Wert fehlt, wird das Model mit dem Wert <code>undefined</code> erzeugt.

**Beispiel**

```
var emptyModel = x4.model.create();  
  
var numberModel = x4.model.create(5);  
  
var textModel = x4.model.create('my.test.namespace');  
  
var objectModel = x4.model.create({ key: "value" });  
  
var arrayModel = x4.model.create(['bla', 5, true]);
```

### 2.14.5.2.3.2 destroy

Diese Methode löscht die Daten sowie alle Event-Handler und interne Referenzen einer *Model*-Instanz. Danach ist die Instanz nicht mehr verwendbar.

Mit der *destroy*-Methode wird dem Browser die Freigabe von nicht mehr benötigter Speicherbereiche erleichtert.

Wenn alle Event-Handler mit externen Referenzen explizit entfernt werden, ist der Einsatz der *destroy*-Methode nicht erforderlich.

```
x4.model.destroy(model: <Model>)
```


Parameter	Erläuterung
<code>data</code>	Natives Daten-Objekt, -Array oder ein primitiver Wert, mit dem das Model initialisiert werden soll  Wenn der Wert fehlt, wird das Model mit dem Wert <code>undefined</code> erzeugt.

**Beispiel**

```
var model = x4.model.create();  
  
model.get(); // ok  
  
x4.model.destroy(model);  
  
model.get(); // Fehler
```

### 2.14.5.2.3.3 typeOf

Diese Methode ermittelt, ob ein nativer Wert im Model als Objekt ( "object" ), Array ( "array" ) oder primitiver Wert ( "simple" ) repräsentiert wird.

 **Achtung:** `null` wird – im Gegensatz zum nativen `typeof` – nicht als Objekt, sondern als primitiver Wert behandelt!

`x4.model.typeOf(data): string`

Parameter	Erläuterung
<code>data</code>	Beliebiger JavaScript-Wert

#### Beispiel

```
x4.model.typeOf(1); // => "simple"
x4.model.typeOf(true); // => "simple"
x4.model.typeOf('value'); // => "simple"
x4.model.typeOf(null); // => "simple"
x4.model.typeOf({}); // => "object"
x4.model.typeOf([]); // => "array"
```

#### 2.14.5.2.3.4 isModel

Diese Methode prüft, ob der übergebene Wert eine *Model*-Instanz ist.

`x4.model.isModel(model): boolean`

Parameter	Erläuterung
<code>model</code>	Ein beliebiger JavaScript-Wert

#### Beispiel

```
x4.model.isModel(x4.model.create()); // => true
x4.model.isModel(null); // => false
x4.model.isModel({}); // => false
```

#### 2.14.5.2.3.5 isSimpleModel

Diese Methode prüft, ob die übergebene *Model*-Instanz einen primitiven Wert enthält.

`x4.model.isSimpleModel(model: <Model|null|undefined>): boolean`

Parameter	Erläuterung
<code>model</code>	Model-Instanz, null oder undefined

**Beispiel**

```
x4.model.isSimpleModel(x4.model.create()); // => true
x4.model.isSimpleModel(x4.model.create('value')); // => true
x4.model.isSimpleModel(x4.model.create({})); // => false
x4.model.isSimpleModel(null); // => false
x4.model.isSimpleModel({}); // FEHLER
```

**2.14.5.2.3.6 isObjectModel**

Diese Methode prüft, ob die übergebene *Model*-Instanz ein Objekt enthält.

`x4.model.isObjectModel(model: <Model|null|undefined>): boolean`

Parameter	Erläuterung
<code>model</code>	Model-Instanz, null oder undefined

**Beispiel**

```
x4.model.isObjectModel(x4.model.create({})); // => true
x4.model.isObjectModel(x4.model.create('value')); // => false
x4.model.isObjectModel(null); // => false
x4.model.isObjectModel({}); // FEHLER
```

**2.14.5.2.3.7 isArrayModel**

Diese Methode prüft, ob die übergebene *Model*-Instanz ein Array enthält.

`x4.model.isArrayModel(model: <Model|null|undefined>): boolean`

Parameter	Erläuterung
<code>model</code>	<i>Model</i> -Instanz, null oder undefined

**Beispiel**

```
x4.model.isArrayModel(x4.model.create([])); // => true
x4.model.isArrayModel(x4.model.create('value')); // => false
x4.model.isArrayModel(null); // => false
x4.model.isArrayModel({}); // FEHLER
```

**2.14.5.2.3.8 isArrayItem**

Diese Methode prüft, ob die übergebene *Model*-Instanz ein direkter Eintrag eines *Array-Models* ist.



`x4.model.isArrayItem(model: <Model|null|undefined>): boolean`

Parameter	Erläuterung
<code>model</code>	<i>Model</i> -Instanz, null oder undefined

#### Beispiel

```
var list = ['test', 'bla'],
    arrayModel = x4.model.create(list);

x4.model.isArrayItem(arrayModel.getModel(0)); // => true
x4.model.isArrayItem(arrayModel); // => false
x4.model.isArrayItem(null); // => false
x4.model.isArrayItem(list[0]); // FEHLER
```

#### 2.14.5.2.3.9 isEmpty

Diese Methode prüft, ob eine *Model*-Instanz leer ist, also den Wert `undefined` enthält.

`x4.model.isEmpty(model: <Model>): boolean`

Parameter	Erläuterung
<code>model</code>	Eine <i>Model</i> -Instanz

#### Beispiel

```
x4.model.isEmpty(x4.model.create()); // => true
x4.model.isEmpty(x4.model.create('value')); // => false
x4.model.isEmpty(x4.model.create(null)); // => false
x4.model.isEmpty({}); // FEHLER
x4.model.isEmpty(undefined); // FEHLER
x4.model.isEmpty(null); // FEHLER
```

#### 2.14.5.2.3.10 getRootOf

Diese Methode liefert die Wurzel-Instanz (äußerste Instanz) der *Model*-Hierarchie, in der sich das übergebene *Model* befindet.

`x4.model.getRootOf(model: <Model>): <Model>`

Parameter	Erläuterung
<code>model</code>	Eine <i>Model</i> -Instanz

**Beispiel**

```
var model = x4.model.create({ key: "value" });

x4.model.getRootOf(model) === model; // => true
x4.model.getRootOf(model.getModel('key')) === model; // => true
```

#### 2.14.5.2.3.11 initModelData

Diese Methode initialisiert eine *Model*-Instanz mit Daten.

Wenn das *Model*/leer (undefined) ist, werden die Daten komplett ins *Model*/übernommen.

Wenn das *Model*/bereits Daten enthält, so wird versucht, die bestehenden Daten entsprechend zu erweitern. Bestehende Werte werden dabei nicht verändert.

```
x4.model.initModelData(model: <Model>, data)
```

Parameter	Erläuterung
<code>model</code>	<i>Model</i> -Instanz

**Beispiel**

```
var model = x4.model.create();

x4.model.initModelData(model, { key: "value" });
model.get(); // => { key: "value" }

x4.model.initModelData(model, { key2: "value2" });
model.get(); // => { key: "value", key2: "value2" }

x4.model.initModelData(model, 'test');
model.get(); // => { key: "value", key2: "value2" }
```

#### 2.14.5.2.4 Model-Instanz-API

Methoden für den Zugriff auf eine *Model*-Instanz

##### 2.14.5.2.4.1 getId

Diese Methode liefert die ID der *Model*-Instanz. Diese wird beim Anlegen des *Models* automatisch generiert, um dieses eindeutig identifizieren zu können.

```
<model>.getId(): string
```

#### 2.14.5.2.4.2 getRef

Diese Methode liefert die absolute Referenz (durch Punkte getrennte Bezeichner) des Models innerhalb der Model-Hierarchie.

```
<model>.getRef(): string
```

#### 2.14.5.2.4.3 getKey

Diese Methode liefert den Key/Namen des Models im übergeordneten Model.

```
<model>.getKey(): string
```

#### 2.14.5.2.4.4 get

Diese Methode liefert die Daten des *Models* oder eines direkten Kind-*Models*.

Wenn das Kind-*Model* noch nicht existiert, wird automatisch ein leeres *Model* erzeugt.

```
<model>.get([key: string]): <PlainData>
```

Parameter	Erläuterung
<code>key</code>	Key/Name eines Kind-Models

**Rückgabewert:** Die Daten des referenzierten Models

#### Beispiel

```
var model = x4.model.create({
    num: 3,
    str: "test"
});

model.get(); // => { num: 3, str: "asdf" }
model.get('num'); // => 2
model.get('str'); // => "test"
```

#### 2.14.5.2.4.5 find

Diese Methode löst eine *Model-Referenz* auf und liefert die Daten des referenzierten *Models*.

Wenn das referenzierte *Model* noch nicht existiert, wird automatisch ein leeres *Model* inklusive der kompletten dafür benötigten Objekt-Hierarchie erzeugt.

```
<model>.find(ref: string): <PlainData>
```

Parameter	Erläuterung
<code>ref</code>	Referenz innerhalb des Models

**Rückgabewert:** Die Daten des referenzierten Models

#### Beispiel

```
var model = x4.model.create({
  num: 3,
  str: "test",
  obj: {
    bool: true
  }
});

model.find('num'); // => 2
model.find('obj.bool'); // => true
model.find('#self'); // => { num: 3, str: "test", obj: { bool: true} }

model.find('obj.test'); // => undefined
model.get(); // => { num: 3, str: "test", obj: { bool: true, test: undefined} }
```

#### 2.14.5.2.4.6 search

Diese Methode löst eine Model-Referenz auf und liefert die Daten des referenzierten Models.

Wenn das referenzierte Model existiert, wird `undefined` zurückgegeben. Die Struktur des Models bleibt unverändert.

`<model>.search(ref: string): <PlainData>`

Parameter	Erläuterung
<code>ref</code>	Referenz innerhalb des Models

**Rückgabewert:** Die Daten des referenzierten Models

**Beispiel**

```

var model = x4.model.create({
  num: 3,
  str: "test",
  obj: {
    bool: true
  }
});

model.search('num'); // => 2
model.search('obj.bool'); // => true
model.search('#self'); // => { num: 3, str: "test", obj: { bool: true } }

model.search('obj.test'); // => undefined
model.get(); // => { num: 3, str: "test", obj: { bool: true } }

```

**2.14.5.2.4.7 set**

Diese Methode ändert die Daten des *Models* oder eines direkten Kind-*Models*. Dabei wird die Struktur des Models an die neuen Daten angepasst, d.h. Einträge, die in den Daten nicht vorhanden sind, werden aus dem *Model* entfernt.

<model>.set([key: string, ] value: <PlainData>): <model>

Parameter	Erläuterung
key	Referenz innerhalb des Models
value	Neue Daten

**Rückgabewert:** Das *Model*, auf dem die Methode aufgerufen wurde

**Beispiel**

```

var model = x4.model.create({
  num: 3,
  str: "test",
  obj: {
    bool: true
  }
});

model.set('num', 5).get(); // => { num: 5, str: "test", obj: { bool: true } }
model.set('str', ['test']).get(); // => { num: 5, str: ["test"], obj: { bool: true } }
model.set('obj', {test: null}).get(); // => { num: 5, str: ["test"], obj: { test: null } }

model.set({ num: 11 }).get(); // => { num: 11 }
model.set(['test', 5]).get(); // => [ "test", 5 ]

```

#### 2.14.5.2.4.8 change

Diese Methode ändert die Daten des *Models* oder eines direkten Kind-*Models*. Dabei wird die Struktur des *Models* soweit möglich beibehalten und nur so weit geändert, um die neuen Daten abzubilden. Nur wenn Daten geändert wurden, wird ein Change-Event ausgelöst (im Gegensatz zu `set`).

```
<model>.change([key: string, ] value: <PlainData>): <model>
```

Parameter	Erläuterung
key	Referenz innerhalb des Models
value	Die neuen Daten

**Rückgabewert:** Das Model, auf dem die Methode aufgerufen wurde

##### Beispiel

```
var model = x4.model.create({
  num: 3,
  str: "test",
  obj: {
    bool: true
  }
});

model.change('num', 5).get(); // => { num: 5, str: "test", obj: { bool: true } }
model.change('str', ['test']).get(); // => { num: 5, str: ["test"], obj: { bool: true } }
model.change('obj', {test: null}).get(); // => { num: 5, str: ["test"], obj: { bool: true, test: null } }

model.change({ num: 11 }).get(); // => { num: 11, str: ["test"], obj: { bool: true, test: null } }
model.change(['test', 5]).get(); // => [ "test", 5 ]
```

#### 2.14.5.2.4.9 update

Diese Methode ändert die Daten des *Models* oder eines direkten Kind-*Models* und löscht die Werte aller anderen Einträge des Models. Dabei wird die Objekt-Struktur des *Models* soweit möglich beibehalten und nur so weit geändert, um die neuen Daten abzubilden. Führt zum gleichen Ergebnis wie `clear` gefolgt von `change`.

```
<model>.update([key: string, ] value: <PlainData>): <model>
```

Parameter	Erläuterung
key	Referenz innerhalb des Models
value	Die neuen Daten

**Rückgabewert:** Das Model, auf dem die Methode aufgerufen wurde

#### Beispiel

```
var model = x4.model.create({
  num: 3,
  str: "test",
  obj: {
    bool: true
  },
  list: [1, 5, 8]
});

model.update('num', 5).get(); // => { num: 5, str: "test", obj: { bool: true }, list:
[1, 5, 8] }

model.update({ num: 11 }).get(); // => { num: 11, str: undefined, obj: { bool:
undefined }, list: [] }
```

#### 2.14.5.2.4.10 clear

Diese Methode löscht alle Werte eines *Models*, aber erhält Objekt-Strukturen.

`<model>.clear([key: string])`

Parameter	Erläuterung
key	Key eines Kind- <i>Models</i>

**Rückgabewert:** Das *Model*, auf dem die Methode aufgerufen wurde

#### Beispiel

```
var model = x4.model.create({
  num: 3,
  str: "test",
  list: [4, 7, 11, "test"],
  obj: {
    bool: true
  }
});

model.clear('obj').get(); // => { num: 3, str: "test", list: [4, 7, 11, "test"], obj:
{ bool: undefined} }

model.clear().get(); // => { num: undefined, str: undefined, list: [], obj: { bool:
undefined} }
```

#### 2.14.5.2.4.11 remove

Diese Methode löscht ein *Model*.

`<model>.remove([key: string])`

Parameter	Erläuterung
key	Key eines Kind-Models

#### Beispiel

```
var model = x4.model.create({
  num: 3,
  str: "test",
  list: [4, 7, 11, "test"],
  obj: {
    bool: true
  }
});

model.remove('obj');
model.get(); // => { num: 3, str: "test", list: [4, 7, 11, "test"] }

model.getModel('list').remove();
model.get(); // => { num: 3, str: "test" }
```

#### 2.14.5.2.4.12 length

Diese Methode ermittelt die Länge des *Models*.

`<model>.length(): integer|boolean`

**Rückgabewert:** Bei Array-Models die Anzahl an Einträgen, ansonsten false.



**Beispiel**

```

var model = x4.model.create({
  num: 3,
  list: [4, 7, 11, "test"],
  list2: [],
  obj: {
    bool: true
  }
});

model.getModel('num').length(); // => false
model.getModel('list').length(); // => 4
model.getModel('list2').length(); // => 0
model.getModel('obj').length(); // => false

```

**Beispiel 2**

```

// check if model contains array and has entries
if(anyModel.length())
  doSomethingWithArrayItems(anyModel);

```

**2.14.5.2.4.13 getType**

Diese Methode liefert den Typ des Arrays.

`<model>.getType(): string`

**Rückgabewert:** Typ-String des Models ("simple", "object" oder "array")

**2.14.5.2.4.14 getModel**

Diese Methode liefert ein Kind-*Model* des *Models*.

Wenn das Kind-Model noch nicht existiert, wird automatisch ein leeres *Model* erzeugt.

`<model>.getModel(key: string): <model>`

Parameter	Erläuterung
<code>key</code>	Key/Name eines Kind- <i>Models</i>

**Rückgabewert:** Das referenzierte *Model*

**Beispiel**

```
model.getModel('test').get(); // same as: model.get('test')

model.getModel(0).get(); // same as: model.get(0)
```

**2.14.5.2.4.15 findModel**

Diese Methode löst eine *Model*-Referenz auf und liefert das referenzierte *Model*. Existiert das *Model* noch nicht, wird automatisch ein leeres *Model* erzeugt.

```
<model>.findModel(ref: string): <Model>
```

Parameter	Erläuterung
<code>ref</code>	Key/Name eines Kind- <i>Models</i>

**Rückgabewert:** Das referenzierte *Model*

**Beispiel**

```
model.findModel('test') === model.getModel('test'); // => true

model.findModel('test.bool') === model.getModel('test').getModel('bool'); // => true

model.findModel('#self') === model; // => true
model.findModel('#parent.' + model.getKey()) === model; // => true
model.findModel(model.getRef()) === model; // => true
```

**2.14.5.2.4.16 searchModel**

Diese Methode löst eine *Model*-Referenz auf und liefert das referenzierte *Model*, wenn es existiert.

```
<model>.searchModel(ref: string): <Model>
```

Parameter	Erläuterung
<code>ref</code>	Key/Name eines Kind- <i>Models</i>

**Rückgabewert:** Das referenzierte *Model* oder null, wenn es nicht existiert

**2.14.5.2.4.17 getParent**

Diese Methode liefert das übergeordnete *Model*.

```
<model>.getParent(): <Model>
```

**Rückgabewert:** Das übergeordnete *Model* oder null, im Falle des Wurzel-*Models*

#### 2.14.5.2.4.18 each

Diese Methode iteriert über alle Kind-*Models* oder bei primitiven Typen das *Model* selbst.

```
<model>.each(iterator: Function): boolean
```

Parameter	Erläuterung
<code>iterator</code>	<p>Funktion, der pro Aufruf ein Kind-<i>Model</i> und dessen Key als Parameter übergeben werden. Die <code>this</code>-Referenz im Iterator verweist auf das <i>Model</i>, auf dem <code>each</code> aufgerufen wurde.</p> <p>Das Iterieren kann abgebrochen werden, wenn die Iterator-Funktion <code>false</code> zurückgibt.</p>

**Rückgabewert:** `false` wenn das Iterieren abgebrochen wurde, andernfalls `true`

#### Beispiel

```
var model = x4.model.create({
  num: 3,
  list: [4, 7, 11],
  obj: {
    bool: true,
    num: 11
  }
});

function iterator(child, key){
  if(arguments.length < 2){
    console.log("simple model");
  }
  console.log(child.get(), key);
}

model.getModel('num').each(iterator); // log: "simple model" | 3,undefined
model.getModel('list').each(iterator); // log: 4,0 | 7,1 | 11,2
model.getModel('obj').each(iterator); // log: true,"bool" | 11,"num"
```

#### 2.14.5.2.4.19 add

Diese Methode fügt einen Eintrag am Ende eines Arrays hinzu.

Die `add`-Methode ist nur auf *Array-Models* und leeren *Models* (`undefined` als Wert) verfügbar. Ein leeres *Model* wird automatisch vor dem Einfügen in ein leeres Array konvertiert.

```
<model>.add(value: <PlainData>): <model>
```

Parameter	Erläuterung
value	Daten für den neuen Array-Eintrag

**Rückgabewert:** Das *Model*, auf dem die Methode aufgerufen wurde

#### Beispiel

```
var model = x4.model.create();

model.add(5).get(); // => [ 5 ]
model.add({ bool: true }).get() // [ 5, { bool: true } ]
```

#### 2.14.5.2.4.20 insert

Diese Methode fügt einen Eintrag an einer bestimmten Position zu einem Array hinzu. Nachfolgende Einträge werden nach hinten verschoben.

Die insert-Methode ist nur für Array-*Models* verfügbar.

```
<model>.insert(pos: integer, value: <PlainData>): <model>
```

Parameter	Erläuterung
pos	Position des neuen Eintrags (Zählung beginnt bei 0)
value	Daten für den neuen Array-Eintrag

**Rückgabewert:** Das *Model*, auf dem die Methode aufgerufen wurde

#### Beispiel

```
var model = x4.model.create(['c']);

model.insert(1, 'x').get(); // => [ "c", "x" ]
model.insert(0, 'a').get(); // => [ "a", "c", "x" ]
model.insert(1, 'b').get(); // => [ "a", "b", "c", "x" ]
model.insert(4, 'z').get(); // => [ "a", "b", "c", "x", "z" ]

model.insert(6, 'fail').get(); // FEHLER - ungültiger Index
```

#### 2.14.5.2.4.21 subscribe

Diese Methode registriert eine Funktion im *Model*, die über Änderungen des *Models* benachrichtigt wird.

```
<model>.subscribe(handler: Function): <handler>
```

Parameter	Erläuterung
<code>handler</code>	Eine Funktion, die bei jeder Änderung des <i>Models</i> (auch untergeordnete <i>Models</i> ) aufgerufen wird. Als Parameter wird eine <i>ChangeEvent</i> -Objekt übergeben. Die <i>this</i> -Referenz verweist auf das <i>Model</i> , auf dem die Methode aufgerufen wurde.

**Rückgabewert:** Die Funktion, die als Handler übergeben wurde

#### Beispiel

```
var model = x4.model.create(5);

model.subscribe(function(event){
    console.log(this.get());
});

model.change(5); // kein Aufruf
model.set(5); // log: 5

model.change(7); // log: 7
model.change(true); // log: true
model.change({ str: "test" }); // log: { str: "test" }
```

#### 2.14.5.2.4.22 unsubscribe

Diese Methode iteriert über alle Kind-Models oder bei primitiven Typen das Model selbst.

`<model>.subscribe(handler: Function): <handler>`

Parameter	Erläuterung
<code>handler</code>	Eine Funktion, die bei jeder Änderung des Models (auch untergeordnete Models) aufgerufen wird. Als Parameter wird eine <i>ChangeEvent</i> -Objekt übergeben. Die <i>this</i> -Referenz verweist auf das Model, auf dem die Methode aufgerufen wurde.

**Rückgabewert:** Die Funktion, die als Handler übergeben wurde

**Beispiel**

```
var model = x4.model.create(5);

var subscription = model.subscribe(function(event){
    console.log(this.get());
});

model.change(7); // log: 7

model.unsubscribe(subscription);

model.change(10); // kein Aufruf

// expliziter Aufruf
subscription.call(model); // log: 10
```

**Auf dieser Seite:****Siehe auch:**

- [Objektorientierte JavaScript-API \(core.js\)](#)
- [Asynchrone Aktionen und ReST \(async/async.js\)](#)
- [UI-Framework für den X4 Web Designer](#)
- [XML-Parser/Serializer \(xml/xml.js\)](#)

### 2.14.5.3 Asynchrone Aktionen und ReST (async/async.js)

Die Javascript-Datei `async/async.js` ist Bestandteil der JavaScript-API für den *X4 Web Designer* und unterstützt bei der asynchronen Verarbeitung von Daten.

#### 2.14.5.3.1 Asynchrone Aktionen synchronisieren (x4.async)

##### 2.14.5.3.1.1 runActions

Diese Methode führt mehrere Funktionen asynchron aus und benachrichtigt, wenn alle Aktionen abgeschlossen sind.

`x4.async.runActions(actions: Array<Function>, ready: Function)`

Parameter	Erläuterung
<code>actions</code>	<p>Array von Funktionen, die asynchrone Aktionen ausführen</p> <p>Jede dieser Funktionen erhält als einzigen Parameter eine Callback-Funktion, die sie aufrufen muss, sobald die Aktion abgeschlossen ist. Wenn der Callback-Funktion Parameter übergeben werden, so werden die Werte der Parameter aller Aufrufe gesammelt und am Ende als Ergebnis-Array bereitgestellt.</p>

Parameter	Erläuterung
ready	Funktion, die aufgerufen wird, wenn alle Aktionen abgeschlossen sind. Diese Funktion erhält die Ergebnisse der Aktionen (sofern vorhanden) als Array übergeben.

### Beispiel

```

var actions = [];

actions.push(function(ready){
    loadExternalData(function(successful, data){
        if(successful) ready(data);
        else ready(); // no data found
    });
});

actions.push(function(ready){
    waitForUserInput(function(inputData){
        ready(inputData);
    });
});

x4.async.runActions(actions, function(results){
    if(results.length < 2) {
        handleSingleData(results[0]);
    } else {
        handleMultiData(results);
    }
});

```

#### 2.14.5.3.1.2 processData

Diese Methode führt eine asynchrone Aktion für eine Liste von Datensätzen aus und benachrichtigt, wenn alle Aktionen abgeschlossen sind.

```
x4.async.processData(items: Array, action: Function, ready: Function)
```

Parameter	Erläuterung
items	Array von Datensätzen, die asynchron verarbeitet werden sollen.
action	<p>Funktion, die für jeden Datensatz in <code>items</code> einmal aufgerufen wird.</p> <p>Als ersten Parameter erhält sie den Datensatz.</p> <p>Als zweiter Parameter wird eine Callback-Funktion übergeben. Diese muss aufgerufen werden, sobald die Aktion abgeschlossen ist. Wenn Sie dieser Callback-Funktion Parameter übergeben, so werden die Werte der Parameter aller Aufrufe gesammelt und am Ende als Ergebnis-Array bereitgestellt.</p>
ready	Funktion, die aufgerufen wird, wenn alle Aktionen abgeschlossen sind. Diese Funktion erhält die Ergebnisse der Aktionen (sofern vorhanden) als Array übergeben.

**Beispiel**

```

var action = function(url, ready){
    loadDataFromURL(url, function(successful, data){
        if(successful) ready(data);
        else ready();
    });
};

x4.async.processData([ url1, url2, url3 ], action, function(results){
    if(results.length < 3) alert( (3 - results.length) + ' of 3 loads failed' );

    handleSuccessfulResults(results);
});

```

### 2.14.5.3.2 Event-Management (x4.customEvents)

#### 2.14.5.3.2.1 on

Diese Methode registriert einen Event-Handler.

x4.customEvents.on(baseObj: Object, eventType: string, handler: Function):  
<handler>

Parameter	Erläuterung
baseObj	Objekt, in dem das Event registriert werden soll. Dazu wird in dem Objekt ein Attribut mit dem eventType als Key angelegt, in dem ein Array mit allen Event-Handler-Funktionen dieses Types verwaltet wird.
eventType	Beliebige Zeichenkette, die den Event-Typ definiert
handler	Funktion, die beim Auslösen des Events ausgeführt wird

**Rückgabewert:** Funktion, die als Handler übergeben wurde

#### 2.14.5.3.2.2 off

Diese Methode entfernt einen Event-Handler oder alle Handler eines Typs.

x4.customEvents.off(baseObj: Object, eventType: string [, handler: Function]):  
boolean

Parameter	Erläuterung
baseObj	Objekt, in dem der/die Handler registriert wurde(n)
eventType	Beliebige Zeichenkette, die den Event-Typ definiert
handler	Handler-Funktion, die entfernt werden soll

**Rückgabewerte:** true wenn das Entfernen erfolgreich war, andernfalls false (z.B. wenn der Handler bereits entfernt wurde)



### 2.14.5.3.2.3 fire

Diese Methode löst ein Event aus.

```
x4.customEvents.fire(baseObj: Object, eventType: string [, args: Array [,  
thisObj: Object]])
```

Parameter	Erläuterung
<a href="#">baseObj</a>	Objekt, in welchem die Event-Handler verwaltet werden
eventType	Beliebige Zeichenkette, die den Event-Typ definiert
args	Array, dessen Werte den Handler-Funktionen als Parameter übergeben werden
thisObj	Objekt, das in den Handler-Funktionen als this-Referenz verfügbar sein soll

**Beispiel**

```
/**
 * Eigene Klasse mit Events.
 */
var MyEventManager = function(){
    this.events = {};
};
MyEventManager.prototype = {
    on: function(type, handler){
        return x4.customEvents.on(this.events, type, handler);
    },
    off: function(type, handler){
        return x4.customEvents.off(this.events, type, handler);
    },
    removeAllHandlers: function(type){
        if(type) {
            x4.customEvents.off(this.events, type);
        } else {
            this.events = {};
        }
    },
    change: function(data){
        this.data = data;

        x4.customEvents.fire(this.events, 'change', [data], this);
    }
};

// Verwendung einer Instanz

var em = new MyEventManager();

var handler = em.on('change', function(data){
    console.log(data);
});
em.on('change', function(data){
    alert(data);
});
em.change(5); // log: 5; alert: 5

em.off('change', handler); // => true
em.change(7); // alert: 7
em.off('change', handler); // => false

em.off('change'); // => true
em.change(11); // kein Aufruf

handler(em.data); // log: 11
```

### 2.14.5.3.3 URL-Verwaltung (x4.url)

#### 2.14.5.3.3.1 parse

Diese Methode parst eine URL und stellt ein Objekt für den Zugriff auf die URL-Bestandteile bereit.

Wenn Sie keine URL übergeben, wird automatisch die aktuelle URL des Browser-Fensters verwendet (`window.location.href`).

```
x4.url.parse([url: string]): <Url>
```

Parameter	Erläuterung
<code>url</code>	Zu parsende URL

**Rückgabewert:** Ein URL-Objekt

### 2.14.5.3.4 URL-Instanz-Verwaltung

#### 2.14.5.3.4.1 get

Diese Methode liefert die vollständige, absolute URL als Zeichenkette.

```
<url>.get(): string
```

**Rückgabewert:** Die URL als Zeichenkette

#### 2.14.5.3.4.2 getParam

Diese Methode liefert den Wert eines URL-/Query-Parameters.

```
<url>.getParam(key: string): string
```

Parameter	Erläuterung
<code>key</code>	URL-Parameter Key/Name

**Rückgabewert:** Den Parameter-Wert als Zeichenkette

#### 2.14.5.3.4.3 getParams

Diese Methode liefert alle URL-/Query-Parameter als Objekt.

```
<url>.getParams(): Object
```

**Rückgabewert:** Ein Objekt, das für jeden URL-Parameter ein Attribut enthält. Der Attribut-Name entspricht dabei jeweils dem dekodierten Parameter-Namen. Der Attribut-Wert entspricht dem dekodierten Parameter-Wert.

#### 2.14.5.3.4.4 getHash

Diese Methode liefert ein Hash/Fragment (dekodiert) der URL.

```
<url>.getHash(): string
```

**Rückgabewert:** Hash/Fragment als dekodierte Zeichenkette ohne Hash-Prefix (#)

#### 2.14.5.3.4.5 getNativeLocation

Diese Methode liefert eine native Repräsentation der URL.

```
<url>.getNativeLocation(): Location
```




**Rückgabewert:** URL als `Location`-Objekt

#### 2.14.5.3.5 HTTP-Request (x4.http.Request)

##### 2.14.5.3.5.1 Konstruktor

Diese Methode erzeugt ein Objekt, um einen HTTP-Request auszuführen mit Hilfe einer `XMLHttpRequest`-Instanz (siehe <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>).

```
new x4.http.Request(url: string [, method: string [, settings: Object]])
```

Parameter	Erläuterung				
<code>url</code>	URL für den Request				
<code>method</code>	HTTP-Methode des Requests.  <i>Standard: 'GET'</i>				
<code>settings</code>	Objekt, das Einstellungen für den Request definiert. <table> <tr> <th>Option</th><th>Erläuterung</th></tr> <tr> <td><code>timeout</code></td><td> Zeitbeschränkung (Timeout) für die Anfrage in Millisekunden   <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>0 oder null bedeutet, dass es keinen Timeout gibt (Standard).</li> <li>Beliebige ganze Zahl</li> </ul>  Durch das Konfigurationsattribut <code>x4.http.options.defaultTimeout</code> kann eine Standard-Zeitbeschränkung für alle Requests definiert werden, für die nicht explizit eine Zeitbeschränkung definiert wurde. </td></tr> </table>	Option	Erläuterung	<code>timeout</code>	Zeitbeschränkung (Timeout) für die Anfrage in Millisekunden  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>0 oder null bedeutet, dass es keinen Timeout gibt (Standard).</li> <li>Beliebige ganze Zahl</li> </ul>  Durch das Konfigurationsattribut <code>x4.http.options.defaultTimeout</code> kann eine Standard-Zeitbeschränkung für alle Requests definiert werden, für die nicht explizit eine Zeitbeschränkung definiert wurde.
Option	Erläuterung				
<code>timeout</code>	Zeitbeschränkung (Timeout) für die Anfrage in Millisekunden  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>0 oder null bedeutet, dass es keinen Timeout gibt (Standard).</li> <li>Beliebige ganze Zahl</li> </ul>  Durch das Konfigurationsattribut <code>x4.http.options.defaultTimeout</code> kann eine Standard-Zeitbeschränkung für alle Requests definiert werden, für die nicht explizit eine Zeitbeschränkung definiert wurde.				


##### 2.14.5.3.5.2 send

Diese Methode startet einen HTTP-Request.

```
<httpRequest>.send([data [, contentType: string]]): <httpRequest>
```

Parameter	Erläuterung
<code>data</code>	Daten, die per HTTP gesendet werden sollen  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• Beliebige Zeichenkette</li> <li>• Beliebiger Datentyp, der von der nativen <code>send</code>-Methode akzeptiert wird (siehe <a href="https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest#send">https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest#send</a>); <i>Beispiel:</i> <code>FormData</code> (siehe <a href="https://developer.mozilla.org/en-US/docs/Web/API/FormData">https://developer.mozilla.org/en-US/docs/Web/API/FormData</a>).</li> <li>• (ohne Daten)</li> </ul>
<code>contentType</code>	Wert für den HTTP-Header <code>Content-Type</code> (MIME-Typ der Daten).  Wenn dieser Parameter fehlt und kein MIME-Type definiert wurde, so wird der HTTP-Request ohne <code>Content-Type</code> -Header gesendet.

**Rückgabewert:** Das HTTP-Request-Objekt, auf dem die Methode aufgerufen wurde

 Nach Beendigung des HTTP-Requests werden automatisch alle internen Referenzen und Event-Handler entfernt, wodurch die Instanz unbrauchbar wird.

#### 2.14.5.3.5.3 `ready`

Diese Methode registriert eine Funktion, die aufgerufen wird, sobald ein zuvor ausgeführter HTTP-Request beendet wurde.

Sie können beliebig viele `ready`-Handler registrieren, die in der Reihenfolge ihrer Registrierung nacheinander ausgeführt werden.

```
<httpRequest>.ready(handler: Function): <httpRequest>
```

Parameter	Erläuterung
<code>handler</code>	Funktion, die nach Abschluss des HTTP-Requests aufgerufen wird  Diese Funktion erhält als einzigen Parameter ein HTTP-Response-Objekt (siehe unten). Sie können über die <code>this</code> -Referenz auf das Objekt zugreifen, auf dem <code>ready</code> aufgerufen wurde.

**Rückgabewert:** Das Request-Objekt, auf dem die Methode aufgerufen wurde

#### 2.14.5.3.5.4 `getXHR`

Diese Methode stellt das native `XMLHttpRequest`-Objekt (siehe <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>) bereit, um komplexere Operationen zu ermöglichen, beispielsweise zum Ergänzen zusätzlicher HTTP-Header.

```
<httpRequest>.getXHR(): XMLHttpRequest
```

**Rückgabewert:** Das native  
XMLHttpRequest-Objekt

### 2.14.5.3.6 HTTP-Response (x4.http.Response)

#### 2.14.5.3.6.1 Konstruktor

Diese Methode erzeugt ein Objekt, mit dem Sie die Antwort einer  
XMLHttpRequest-Instanz (siehe <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>) auswerten können.

```
new x4.http.Response(xhr: XMLHttpRequest)
```

Parameter	Erläuterung
xhr	XMLHttpRequest-Instanz



Beachten Sie:

- Verwenden Sie Methoden der Instanz erst nach Abschluss des HTTP-Requests.
- Wenn Sie `x4.http.Request` verwenden, wird automatisch eine `x4.http.Response`-Instanz für die `ready-Handler` erzeugt und muss nicht initialisiert werden.

#### 2.14.5.3.6.2 ok

Diese Methode prüft, ob der HTTP-Request erfolgreich war.

```
<httpResponse>.ok(): boolean
```

**Rückgabewert:** `true`, wenn der HTTP-Antwortstatus zwischen 200 und 299 ist, andernfalls `false`

#### 2.14.5.3.6.3 getText

Diese Methode liefert die Response-Daten eines HTTP-Requests als Zeichenkette.

```
<httpResponse>.getText(): string
```

**Rückgabewert:** Daten als Zeichenkette

#### 2.14.5.3.6.4 getXML

Diese Methode liefert die Antwort-Daten als XML-Dokument.

```
<httpResponse>.getXML(): Document
```

**Rückgabewert:** Daten als XML-Dokument (siehe <https://developer.mozilla.org/en-US/docs/Web/API/Document>)

#### 2.14.5.3.6.5 getContentType

Diese Methode liefert den Daten-Typ der HTTP-Antwort als Zeichenkette.

```
<httpResponse>.getContentType(): string
```

**Rückgabewert:** Zeichenkette mit dem Content-Type HTTP-Header (MIME-Type der Daten)

#### 2.14.5.3.6.6 getXHR

Diese Methode stellt das native XMLHttpRequest-Objekt bereit, um komplexere Operationen zu ermöglichen, wie beispielsweise das Auslesen beliebiger HTTP-Header.

```
<httpResponse>.getXHR(): XMLHttpRequest
```

**Rückgabewert:** Natives XMLHttpRequest-Objekt (siehe <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>)

#### Beispiel

```
new x4.http.Request('test.xml').ready(function(response){
    if(response.ok()){
        var doc = response.getXML();
        // select XML content with jQuery
        var name = $(doc).find('name').text();

        alert( 'Response Name: ' + name);
    } else {
        alert('HTTP-Request failed');
    }
}).send();
```

#### 2.14.5.3.7 ReST(x4.rest)

Die hier bereitgestellte ReST-API in JavaScript verwendet intern für alle Requests `x4.http.Request`-Instanzen.

##### 2.14.5.3.7.1 request

Diese Methode führt einen HTTP-Request aus. Hierzu sind zwei Zeilen Code nach folgendem Muster erforderlich:

1. `x4.rest.request(url: string, settings: Object): x4.http.Request`
2. `x4.rest.request(settings: Object): x4.http.Request(URL in Settings)`

Parameter	Erläuterung
url	Die URL für den Request

Parameter	Erläuterung												
settings	Objekt, welches Einstellungen für den Request definiert: <table> <tr> <th>Einstellung</th><th>Erläuterung</th></tr> <tr> <td>url</td><td>URL für den HTTP-Request, sofern nicht als separater Parameter in Zeile 1 definiert</td></tr> <tr> <td>type</td><td>HTTP-Methode (Standard: "GET")</td></tr> <tr> <td>requestData</td><td>Daten für den HTTP-Request, siehe send in <code>x4.http.Request</code></td></tr> <tr> <td>contentType</td><td>Content-Type für den HTTP-Request</td></tr> <tr> <td>ready</td><td>ready-Handler-Funktion (siehe ready in <code>x4.http.Request</code>)</td></tr> </table>	Einstellung	Erläuterung	url	URL für den HTTP-Request, sofern nicht als separater Parameter in Zeile 1 definiert	type	HTTP-Methode (Standard: "GET")	requestData	Daten für den HTTP-Request, siehe send in <code>x4.http.Request</code>	contentType	Content-Type für den HTTP-Request	ready	ready-Handler-Funktion (siehe ready in <code>x4.http.Request</code> )
Einstellung	Erläuterung												
url	URL für den HTTP-Request, sofern nicht als separater Parameter in Zeile 1 definiert												
type	HTTP-Methode (Standard: "GET")												
requestData	Daten für den HTTP-Request, siehe send in <code>x4.http.Request</code>												
contentType	Content-Type für den HTTP-Request												
ready	ready-Handler-Funktion (siehe ready in <code>x4.http.Request</code> )												

**Rückgabewert:** `x4.http.Request`-Objekt

#### 2.14.5.3.7.2 get

Diese Methode lädt Textdaten als HTTP-GET-Request.

`x4.rest.get(url: string, success: Function): x4.http.Request`

Parameter	Erläuterung
url	URL für den Request
success	Funktion, die aufgerufen wird, wenn der HTTP-Request erfolgreich beendet wurde und als einzigen Parameter die Antwortdaten der HTTP-Response als Text erhält.  Sie können über die <code>this</code> -Referenz auf das <code>x4.http.Request</code> -Objekt zugreifen.

**Rückgabewert:** `x4.http.Request`-Objekt

#### 2.14.5.3.7.3 getJSON

Diese Methode funktioniert analog zu Methode `get`, jedoch werden die Daten für die `success`-Funktion als JSON geparkt und als native JavaScript-Daten (JSON) bereitgestellt.

#### 2.14.5.3.7.4 getXML

Diese Methode funktioniert analog zu Methode `get`, jedoch werden die Daten für die `success`-Funktion als XML-Dokument bereitgestellt.

#### 2.14.5.3.7.5 send

Diese Methode sendet Daten als HTTP-Request.

`x4.rest.send(url: string, data, contentType: string, method: string, ready: Function): x4.http.Request`



Parameter	Erläuterung
<code>url</code>	URL für den HTTP-Request
<code>data</code>	Daten für den HTTP-Request, siehe <code>send</code> in <code>x4.http.Request</code>
<code>contentType</code>	Content-Type für den HTTP-Request
<code>method</code>	HTTP-Methode (Standard: "POST")
<code>ready</code>	Ready-Handler-Funktion, siehe <code>ready</code> in <code>x4.http.Request</code>

**Rückgabewert:** `x4.http.Request`-Objekt

#### 2.14.5.3.7.6 `sendJSON`

Diese Methode sendet JSON-Daten als HTTP-Request.

```
x4.rest.send(url: string [, data: <PlainData> [, method: string [, ready:
Function]]]): x4.http.Request
```

Parameter	Erläuterung
<code>url</code>	URL für den HTTP-Request
<code>data</code>	Native JavaScript-Daten, die vor dem HTTP-Request automatisch als JSON-Daten serialisiert werden
<code>method</code>	HTTP-Methode (Standard: "POST")
<code>ready</code>	Ready-Handler-Funktion, siehe <code>ready</code> in <code>x4.http.Request</code>

**Rückgabewert:** `x4.http.Request`-Objekt

#### Beispiel für Laden und Speichern von JSON-Daten

```
x4.rest.getJSON('person.json', function(data){
    data.name = prompt('Insert New Name');

    x4.rest.sendJSON('person.json', data, null, function(response){
        alert(response.ok() ? 'Name erfolgreich gespeichert' : 'Fehler beim
        Speichern');
    });
});
```

**Beispiel für die Senden und Laden von JSON-Daten mit einem Request**

```
x4.rest.sendJSON('myURL', requestData, 'POST', function(response){
    if(response.ok()){
        var responseData = JSON.parse(response.getText());
        // do something with data of response
    } else {
        alert('Fehler!');
    }
});
```

#### 2.14.5.3.8 Laden und Speichern von *Model*-Daten (x4.rest.DataModelResource)

Mit Klasse `x4.rest.DataModelResource` wird eine ReST-Ressource definiert, die das Laden und Speichern von Daten eines Data-Models vereinfacht, siehe [Data Model API \(model/model.js\)](#). Beim Laden werden automatisch JSON-Daten von einer konfigurierten URL geladen und in das Data-Model geschrieben. Beim Speichern werden die Daten des Models als JSON serialisiert und an eine definierte URL gesendet.

##### 2.14.5.3.8.1 Konstruktor

Diese Methode erzeugt ein Objekt, mit dem ein HTTP-Request mit Hilfe einer XMLHttpRequest-Instanz (siehe <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>) ausgeführt werden kann.

```
new x4.rest.DataModelResource(model: <Model>, config: Object)
```

Parameter	Erläuterung
<code>model</code>	<i>Model</i> -Instanz

Parameter	Erläuterung	
config	Objekt, das Konfigurations-Optionen für die Ressource definiert.	
	Attribut	Erläuterung
	url	Haupt-URL für die ReST-Ressource zum Laden der Daten
	saveURL	URL zum Speichern der Ressource  Wenn diese URL nicht definiert ist, wird stattdessen die Haupt-URL verwendet.
	removeURL	URL zum Löschen der Ressource  Wenn diese URL nicht definiert ist, wird stattdessen die Haupt-URL verwendet.
	httpSaveMethod	HTTP-Methode zum Speichern (Standard: "PUT")
	modelUpdateMethod	Methodenname der <i>Model</i> -Instanz, die zum Ändern der Daten im <i>Model</i> verwendet wird  <i>Mögliche Werte:</i> <ul style="list-style-type: none"> <li>• "set"</li> <li>• "change"</li> <li>• "update" (Standard)</li> </ul>

#### 2.14.5.3.8.2 load

Diese Methode lädt Daten und ändert das *Model* entsprechend.

```
<resource>.load([ready: Function]): x4.http.Request
```

```
<resource>.load([params: object]): x4.http.Request
```

Parameter	Erläuterung
<code>ready</code> or <code>params.loadReady</code>	<p>Funktion, die nach Abschluss des HTTP-Requests, aber vor Änderung des Models, aufgerufen wird</p> <p>Die Funktion erhält als einzigen Parameter die geladenen JSON-Daten.</p> <p>Beachten Sie:</p> <ul style="list-style-type: none"> <li>• Wenn die Funktion <code>false</code> zurückgibt, werden die Daten nicht ins <i>Model</i> übernommen, wodurch die geladenen Daten validiert werden können.</li> <li>• Änderungen an den Daten sind innerhalb dieser Funktion möglich.</li> <li>• Über die <code>this</code>-Referenz können Sie auf das <code>x4.http.Request</code>-Objekt des HTTP-Requests zugreifen.</li> </ul>
<code>params.changeReady</code>	<p>Funktion, die nach Abschluss der Änderung des Models aufgerufen wird.</p> <p>Die Funktion erhält als einzigen Parameter die Daten, die in das Model eingetragen wurden.</p>

**Rückgabewert:** `x4.http.Request`-Objekt, welches für den HTTP-Request verwendet wird

#### 2.14.5.3.8.3 save

Diese Methode speichert die Daten des *Models*.

`<resource>.save([ready: Function]): x4.http.Request`

Parameter	Erläuterung
<code>ready</code>	Standardmäßige <code>ready</code> -Handler-Funktion, siehe <code>ready</code> in <code>x4.http.Request</code>

**Rückgabewert:** `x4.http.Request`-Objekt, welches für den HTTP-Request verwendet wird

#### 2.14.5.3.8.4 remove

Diese Methode löscht die ReST-Ressource. Dazu wird die konfigurierte URL und mit der HTTP-Methode DELETE aufgerufen.

`<resource>.save([ready: Function]): x4.http.Request`

Parameter	Erläuterung
<code>ready</code>	Standardmäßige <code>ready</code> -Handler-Funktion, siehe <code>ready</code> in <code>x4.http.Request</code>

**Rückgabewert:** `x4.http.Request`-Objekt, welches für den HTTP-Request verwendet wird

#### 2.14.5.3.8.5 changeConfig

Diese Methode ändert die Konfigurations-Optionen der Instanz.

`<resource>.changeConfig(newConfig: Object)`

Parameter	Erläuterung
<code>newConfig</code>	Objekt, welches die zu ändernden Attribute enthält  Attribute, die in diesem Objekt nicht vorhanden sind, bleiben unverändert.

#### 2.14.5.3.8.6 getConfig

Diese Methode liefert eine Kopie der aktuellen Konfigurations-Optionen der Instanz.

`<resource>.getConfig(): Object`

Parameter	Erläuterung
<code>ready</code>	Standardmäßige ready-Handler-Funktion, siehe <code>ready</code> in <code>x4.http.Request</code>

**Rückgabewert:** Objekt mit Konfigurations-Optionen

#### 2.14.5.3.8.7 getModel

Diese Methode liefert das verwendete *Model*.

`<resource>.getModel(): <Model>`

**Rückgabewert:** Model-Instanz

#### Beispiel

```
var resource = new x4.rest.DataModelResource(x4.model.create(), {
    url: "person.json"
});
// Automatisches Speichern nach jeder Model-Änderung
resource.getModel().subscribe(function(){
    resource.save();
});
resource.load({
    loadReady: function(data){
        // Model nur ändern, wenn die Daten ein Objekt sind
        return data !== null && typeof data === 'object';
    },
    changeReady: function(data){
        alert('Daten von ' + data.name + ' wurden erfolgreich aktualisiert');
    }
});
```

#### 2.14.5.3.9 Verwenden von Cookies (x4.cookie)

Mit der Funktion `x4.cookie.get` kann der Wert eines Cookies ausgelesen werden.

#### 2.14.5.3.9.1 get

Diese Methode liest den Wert eines Cookies aus und gibt ihn zurück. Wenn ein Cookie nicht vorhanden ist, wird null zurückgegeben.

```
x4.cookie.get(name: string): string
```

Parameter	Erläuterung
name	Name des Cookies der ausgelesen werden soll.

**Auf dieser Seite:**

**Siehe auch:**

- [Objektorientierte JavaScript-API \(core.js\)](#)
- [Data Model API \(model/model.js\)](#)
- [UI-Framework für den X4 Web Designer](#)
- [XML-Parser/Serializer \(xml/xml.js\)](#)

#### 2.14.5.4 UI-Framework für den X4 Web Designer

Die Javascript-Dateien im Ordner `ui` sind Bestandteil der JavaScript-API für den *X4 Web Designer* und stellen ein Framework für die Entwicklung von Benutzeroberflächen dar.

##### 2.14.5.4.1 Einführung zum UI-Framework

Welche grundlegenden Eigenschaften im UI-Framework für den *X4 Web Designer* verfügbar sind und wie Sie das UI-Framework verwenden, um Benutzeroberflächen zu erstellen

##### 2.14.5.4.1.1 Zugehörige JavaScript-Bibliotheken

Im Ordner `ui` des Framework-Moduls befinden sich folgende Bibliotheken:

- `ui.js`: UI-Framework Core, siehe [UI-Framework Core \(ui/ui.js\)](#)
- `frames.js`: Standard-Renderer und Standard-Template für UI-Frames, siehe [Frames \(ui/frames.js\)](#)
- `widgets.js`: API zur Verwaltung von Widget-Collections, siehe [Widgets \(ui/widgets.js\)](#)
- `dependencies.js`: API zur Verwaltung von Abhängigkeiten zwischen UI-Ressourcen, siehe [Dependencies \(ui/dependencies.js\)](#)
- `main.js`: Vereinigung der oben stehenden Dateien

##### 2.14.5.4.1.2 Voraussetzungen für die Nutzung

Um das UI-Framework nutzen zu können, ist es erforderlich, dass in der erstellten HTML-Seite die JavaScript-Bibliothek *jQuery* in Version 1.8 oder höher eingebunden ist.

##### 2.14.5.4.1.3 UI-Struktur

Ein Benutzeroberfläche auf Basis dieses UI-Frameworks ist durch folgenden Elemente strukturiert:

- **Page:** umfasst alle UI-Elemente innerhalb eines HTML-Dokuments, die durch das UI-Framework erzeugt/verwaltet werden.
- **Frame:** enthält einen Baum von Views, wobei es immer eine einzige Wurzel-View gibt.
- **View:** besteht aus DOM-Elementen, die in der Regel durch ein *Template* definiert sind und einer optionalen JavaScript-API, die durch eine *Renderer*-Klasse festgelegt wird.

Durch einen Renderer können die HTML-Elemente der Templates um JavaScript-Logik ergänzt werden, was die Verwendung externer Bibliotheken und Frameworks ermöglicht.

#### View-Kontext

Zu jeder View gehört ein View-Kontext-Objekt. Dieses Objekt stellt die Verbindung zwischen der *View* und dem zugehörigen *Model*(oder zu mehreren Models) sowie zu einem optionalen *Controller* her.

#### UI-Properties

Ein View-Kontext wird mit Hilfe von *UI-Properties*(in JSON) konfiguriert. Diese Eigenschaften definieren u.a.:

- Das zu verwendende Template
- Den Renderer
- Einen Controller
- Custom-Properties: Diese können Sie im Template, Renderer und Controller für dort entsprechende Konfigurationen/Bindings verwenden.

#### UI-Kontext

In einer View können Sie beliebig viele UI-Kontexte definieren. Jeder UI-Kontext kann eine Liste von Views enthalten. Jeder UI-Kontext muss einen innerhalb der View eindeutigen Namen besitzen und definiert eine bestimmte Position im DOM, an welcher die Kind-Views eingefügt werden.

In einem UI-Kontext können Sie die View Konfiguration entweder direkt in den Properties der übergeordneten View hinterlegen oder über die JavaScript-API des View-Kontext-Objekts. Wenn Sie Properties für die Konfiguration verwenden, ist eine Property erforderlich, deren Key/Name dem Namen des UI-Kontexts entspricht und ein Array der Property-Objekte für die Kind-Views enthält.

**Auf dieser Seite:**

**Siehe auch:**

#### 2.14.5.4.1.4 UI-Bindings

Mit UI-Bindings können Sie mit dem UI-Framework als Bestandteil der JavaScript-API des *X4 Web Designers* beliebige Daten und Funktionalität eines View-Kontexts mit UI-Elementen verbinden. Diese Bindings bestehen in der Regel aus einer Kontext-Referenz und/oder einer Content-Referenz, die gemeinsam eine Binding-Referenz ergeben. und aus Bezeichnern bestehen.

#### Bezeichner für UI-Bindings

Für Bezeichner gelten folgende Regeln:

- Bezeichner sind durch Punktzeichen (.) getrennt.
- Erlaubt sind alle gültigen JavaScript-Bezeichner.
- Zusätzlich dürfen Minus (-) und Hash (#) enthalten sein.
- Das Hash-Zeichen (#) markiert Kontext-Referenzen.

- Vor dem ersten Bezeichner kann ein Ausrufezeichen (!) definiert werden, um den Wert des Bindings zu negieren (liefert immer einen booleschen Wert)

#### Kontext-Referenzen und verfügbare Kontext-Variablen

Eine Kontext-Referenz beginnt immer mit einem Hash-Zeichen (#). Sie ermöglicht den Zugriff auf eine Kontext-Variable der View.

Verfügbare Kontext-Variablen:

Variable	Erläuterung
model  (Standard)	Haupt-Model der View, äquivalent zu data  Hierbei handelt es sich um den Standard-Kontext, wenn keine Kontext-Referenz am Anfang eines Bindings angegeben ist.
prop	Properties-Model der View
settings	Ein Model, das Einstellungen enthält, welche innerhalb der View-Hierarchie nach unten vererbt werden
data	Data-Model im Kontext der View
state	State-Model im Kontext der View  Das State-Model speichert UI-Zustandsinformationen, die für die fachlichen Daten der Benutzeroberfläche nicht relevant sind.
localState	Lokales State-Model der View  Das lokale State-Model speichert lokale Zustandsinformationen, die nur im Kontext der aktuellen View relevant sind.
controller	Controller-Instanz im Kontext der View  Hierbei handelt es sich um den nächst übergeordneten Kontext, wenn kein eigener vorhanden ist.
api	Instanz der Renderer-Klasse für diese View
fire	Kurzschreibweise für das Auslösen eines UI-Events im Kontext der View
self	Aktuelles View-Kontext-Objekt  Die self-Variable ist hilfreich, wenn eine Binding-Referenz erforderlich ist, aber kein Kontext-Wechsel gewünscht wird.
parent	Kontext-Wechsel zum nächst übergeordneten View
root	Kontext-Wechsel zur Wurzel-View
frame	Frame, in welchem sich die View befindet  Als Content-Referenz können Sie die UI-ID einer anderen View im gleichen Frame angeben, siehe Beispiel 5



Variable	Erläuterung
window	Liefert das <code>window</code> -Objekt des Browsers. Dadurch kann in Bindings direkt auf globale JavaScript-Methoden der Seite zugegriffen werden.

- i** Welche Art von Folge-Referenzen nach einer Kontext-Referenz unterstützt werden, ist vom Kontext abhängig. Beispielsweise sind nach einer Referenz, die wieder auf einen View-Kontext verweist, wieder alle Standard-Kontext-Referenzen verfügbar (jedoch in anderem Kontext). Bei einem Model stehen nur Model-Referenzen und bei einem Controller nur ein Name einer Controller-Methode zur Verfügung.

### Wertezuweisung in UI-Bindings

In einem UI-Binding kann zwischen zwei Referenzen ein Pfeil bestehend aus Gleichheitszeichen und Größer-Zeichen (`=>`) verwendet werden, um den Wert des ersten Bindings an das zweite Binding zu übergeben. Dies kann bei Bedarf beliebig oft wiederholt werden, z. B. `ref1=>ref2=>ref3`.

Für die Übergabe des Wertes wird der Standard-Parameter `_value` verwendet. `vc.execBinding('source=>target')` ist also identisch zu `vc.execBinding('target', { _value: vc.execBinding('source') })`

### Beispiele für UI-Bindings

1	Zugriff aufs Data-Model	<code>#model.person.name</code> alternativ: <code>person.name</code>
2	Aufruf einer Controller-Funktion	<code>#controller.doSomething</code>
3	Auslesen eines Property-Wertes	<code>#prop.myProperty</code>
4	Zugriff auf einen Property-Wert des übergeordneten Kontexts	<code>#parent.#prop.myProperty</code>
5	Zugriff auf eine View mit ID <i>anyOtherView</i> im gleichen Frame und deren Daten	<code>#frame.anyOtherView.#model.person.name</code> alternativ: <code>#frame.anyOtherView.person.name</code>
6	Auslösen eines UI-Events <i>my-custom-event</i>	<code>#fire.my-custom-event</code>
7	Einen Wert des Data-Models in einem Browser-Dialog anzeigen	<code>person.name=&gt;#window.alert</code>
8	Eine URL aus dem Data-Model in einem neuen Browser-Fenster/-Tab öffnen	<code>url=&gt;#window.open</code>
9	Negieren des Wertes <i>visible</i> im State-Model	<code>!</code> <code>#state.visible=&gt;#state.visible</code>

### Dynamische Bindings für Properties

UI-Properties können nicht nur statische Werte, sondern auch dynamische Bindings enthalten. Hierfür können Sie Platzhalter in geschweiften Klammern hinterlegen. Diese Platzhalter werden ausgewertet, wenn auf die Property in einem Binding-Kontext zugegriffen wird, beispielsweise über ein HTML-Binding-Attribut (siehe [Framework-spezifische HTML-Strukturen](#)) oder über die

Methoden `applyBindings` und `execBinding` eines View-Kontext-Objekts (siehe [UI-Framework Core \(ui/ui.js\)](#)).

*Hinweise zur Platzhalter-Notation:*

- UI-Binding: doppelt geschweifte Klammern, z.B. `{{person.name}}`
- JavaScript-Binding: einfach geschweifte Klammern mit einem beliebigen JavaScript-Ausdruck
- Innerhalb eines solchen JavaScript-Ausdrucks können wieder mit einfachen geschweiften Klammern Platzhalter definiert werden, die ein UI-Binding enthalten.
- `this` ermöglicht innerhalb eines JavaScript-Bindings den Zugriff auf das View-Kontext-Objekt (siehe [UI-Framework Core \(ui/ui.js\)](#)).
- Zugriff auf Kontextvariablen-Werte in den JavaScript-Bindings (abhängig vom Binding-Kontext) mit `_value` (z.B. wenn durch ein Value-Binding ein Wert geändert werden soll).
- Kontext-Variable `_parameters` ist immer verfügbar und enthält alle Kontext-Variablen als Objekt-Properties (Schlüssel: Variablenname, Wert: Variablenwert).

Beispiele für Property-Bindings

1	Zugriff aufs Data-Model	<code>{{#model.person.name}}</code>  alternativ: <code>{{person.name}}</code>
2	Aufruf einer Controller-Funktion	<code>{{#controller.doSomething}}</code>
3	JavaScript-Binding, das zwei Model-Werte zusammenführt:	<code>{ {person.firstName} + ' ' + {person.lastName} }</code>  alternativ: <code>{ this.getModel().get('person.firstName') + ' ' + this.getModel().get('person.lastName') }</code>
4	JavaScript-Binding, das per Komma-Operator ein Event auslöst und eine Controller-Funktion aufruft	<code>{ ( {#fire.my-custom-event}, {#controller.doSomething} ) }</code>  alternativ: <code>{ ( this.fire('my-custom-event'), this.getController().doSomething() ) }</code>
5	Zugriff auf Kontext-Variable in einem Value-Binding	<code>{ _parameters.hasOwnProperty('_value') ? this.getModel().set('value', _value) : this.getModel().get('value') }</code>
6	Einen Wert aus einem Eingabe-Dialog mit dynamischem Titel ins Data-Model schreiben	<code>{{title=&gt;#window.prompt=&gt;result}}</code>

**Auf dieser Seite:**

**Siehe auch:**

- [Binding des Formulars an den Lister-Eintrag erstellen](#)
- [Daten-Bindings für Formularelemente hinterlegen](#)
- [UI-Controller definieren](#)

### 2.14.5.4.1.5 Framework-spezifische HTML-Strukturen

Mit dem UI-Framework als Bestandteil der JavaScript-API des *X4 Web Designers* stehen eine Reihe von Attributen und weiteren HTML-Konstrukten zur Verfügung, um das JavaScript-Framework mit den Inhalten der HTML-Seiten dynamisch zu verknüpfen.

#### HTML-Attribute

Alle HTML-Attribute, die durch das Framework definiert werden, können Sie entweder mit oder ohne `data-`-Präfix verwenden. Eine Präfix wird benötigt, um die HTML5-Spezifikation einzuhalten. Alle gängigen Browser unterstützen jedoch auch die Variante ohne Präfix. Im Folgenden wird daher auf die Angabe von Präfixe verzichtet.

#### Grundlegende HTML-Struktur

Um das UI-Framework in einer HTML-Seite nutzen zu können, sind folgende Angaben erforderlich:

- *jQuery* und das Haupt-Skript muss eingebunden werden.
- Über das HTML-Attribut `x4-init="ui"` auf dem `body`-Element oder die JavaScript-Methode `x4.ui.initPage` (siehe [UI-Framework Core \(ui/ui.js\)](#)) muss das UI-Framework initialisiert werden.

#### Beispiel: Grundlegende HTML-Struktur


```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1"> <!-- Für
mobile Seiten -->
    <title>Title</title>
    <script type="text/javascript" src="/X4/httpstarter/ReST/X4WebResources/lib/
jquery/jquery-2.x.js"></script>
    <script type="text/javascript" src="/X4/httpstarter/ReST/X4WebResources/core/
js/main.js"></script>
  </head>
  <body x4-init="ui">
  </body>
</html>
```

#### HTML-Bindings

Um UI-Bindings auf HTML-Elemente herzustellen, bietet das UI-Framework folgende Attribute.

- ❗ Einige dieser Attribute erwarten eine Liste von Bindings anstelle eines UI-Bindings. Für jedes Binding definieren Sie dann den Name der nativen Eigenschaft (Attribut, Event, CSS-Property oder CSS-Klasse), einen Doppelpunkt, gefolgt vom UI-Binding, das mit dieser Eigenschaft verknüpft werden soll. Die einzelnen Bindings werden dabei durch Semikolons ( ; ) getrennt.  
*Beispiel:* `x4-css="color: #state.myColor; font-size: #settings.myFontSize"`

HTML-Attribut	JavaScript-Bezeichnung	Erläuterung	Typ
---------------	------------------------	-------------	-----

<b>x4-value</b>	value	<p>UI-Binding für den Wert des Elements</p> <p>Bei Input-Elementen entspricht dies dem Eingabewert; bei sonstigen Elementen dem Textinhalt.</p> <p>Wenn das Binding keinen initialen Wert besitzt, wird der Wert des Elements verwendet.</p> <div style="border: 1px solid red; padding: 5px; margin-top: 10px;"> <p> <i>Achtung:</i> x4-value kann Kind-Knoten des Elements überschreiben!</p> </div>	UI-Binding
<b>x4-attr</b>	attr	<p>UI-Bindings für dynamische Attribute</p> <p>Wenn der Wert des Bindings <code>false</code>, <code>null</code> oder <code>undefined</code> ist, wird das Attribut entfernt.</p> <p>Bei <code>true</code> wird der Name des Attributs als Wert verwendet. Andere Werte werden in eine Zeichenkette umgewandelt.</p>	UI-Binding-List
<b>x4-event</b>	event	<p>UI-Bindings für DOM-Events</p> <p>Ein Binding wird ausgeführt, sobald das entsprechende Event ausgelöst wurde. Das Binding erhält das native Event-Objekt als <code>event</code>-Parameter übergeben.</p>	UI-Binding-List
<b>x4-css</b>	css	<p>UI-Bindings für dynamische CSS-Eigenschaften</p> <p>Wenn der Wert des Bindings <code>false</code>, <code>null</code>, <code>undefined</code> oder eine leere Zeichenkette ist, wird die Eigenschaft entfernt. Andere Werte werden in eine Zeichenkette umgewandelt.</p>	UI-Binding-List

<b>x4-css-class</b>	cssClass	<p>UI-Bindings für dynamische CSS-Klassen</p> <p>Wenn der Wert eine nicht-leere Zeichenkette ist, werden der angegebene Klassenname und der Wert des Bindings zusammengeführt, um dynamische Klassennamen zu erzeugen. Dies ist vor allem hilfreich, wenn durch verschiedene Endungen von Klassennamen unterschiedliche Darstellungsmodi eines Elements konfiguriert werden, beispielsweise bei <i>jQuery UI Icons</i>.</p> <p>Wenn der Wert keine Zeichenkette oder leer ist, wird die Klasse hinzugefügt, falls der Wert ein wahrer JavaScript-Ausdruck ist, andernfalls wird die Klasse entfernt.</p>	UI-Binding-List
<b>x4-show-if</b>	showIf	<p>UI-Binding zum dynamischen Ein-/Ausblenden des Elements. Ist der Wert ein wahrer JavaScript-Ausdruck wird das Element angezeigt, andernfalls nicht. Das Ein-/Ausblenden erfolgt über die CSS <code>display</code>-Eigenschaft.</p>	UI-Binding
<b>x4-render-if</b>	renderIf	<p>UI-Binding zum dynamischen Einfügen/Entfernen des Elements im DOM</p> <p>Wenn der Wert kein wahrer JavaScript-Ausdruck ist, wird das Element entfernt, andernfalls bleibt das Element erhalten oder wird wieder eingefügt.</p> <p>Untergeordnete Elemente, auch UI-Kontexte und DOM-Hooks, werden erst ausgewertet, wenn das Element zum ersten Mal gerendert wird. Dadurch können Sie Teile einer Benutzeroberfläche verzögert initialisieren.</p>	UI-Binding
<b>x4-dom-hook</b>	-	<p>Definiert das Element als benannten <i>DOM-Hook</i>, um Zugriff auf das Element per UI-Properties und JavaScript-API zu unterstützen</p> <p>Innerhalb einer View muss der Hook-Name eindeutig sein. <code>_root</code> ist für die Wurzel-Elemente der View reserviert und wird automatisch für jede View erzeugt.</p>	Name

**Beispiel**

```

<h1 x4-value="#prop.title">Add Message</h1>
<!-- Heading could be overwritten by UI-Property title -->

<div>
  <label>
    Message <input x4-value="myMessage" value="This is a default message">
  </label>
  <label>
    Color
    <select x4-value="myColor">
      <option>blue</option>
      <option>green</option>
      <option>red</option>
    </select>
  </label>
  <label>
    Description <textarea x4-value="myDescription"></textarea>
  </label>
  <button x4-event="click: #controller.submitMessage">Submit</button>
</div>

<div x4-show-if="myMessage">
  <h3>Message Preview:</h3>
  <span x4-value="myMessage" x4-attr="title: myDescription" x4-css="color:
myColor"></span>
</div>

```

**Definition von HTML-Code-Templates**

Ein Template kapselt HTML-Code zur Wiederverwendung und wird durch ein Script-Element mit MIME-Typ `text/html` definiert, das den Namen des Templates mit Hilfe des HTML-Attributes `x4-template` definiert. Die Definition befindet sich entweder in der gleichen HTML-Seite, in welcher das Template verwendet werden soll, oder in einer externen HTML-Datei innerhalb einer Template-Collection.

**Beispiel**

```

<script type="text/html" x4-template="MyButton">
  <button x4-value="#prop.label" x4-event="click: #prop.action">Default Label</
button>
</script>

```

**Verwenden von HTML-Code-Templates**

Templates können Sie durch beliebige Platzhalter-Elemente (außer `script`) in die HTML-Struktur der Seite einbinden. Geben Sie dazu den Namen des zu verwendeten Templates in Attribut `x4-template` an. Beim Parsen ersetzt das Framework das Platzhalter-Element durch den Inhalt des Templates.

Beachten Sie, dass die Seite unterschiedlich dargestellt wird, bevor das UI-Framework initialisiert wurde, je nachdem, welchem Element als Platzhalter verwendet wird. Folgende Template-Referenzen erzeugen das gleiche Ergebnis, verhalten sich jedoch vor der Ersetzung unterschiedlich:

**Beispiel**

```
<div x4-template="MyButton"></div>

<span x4-template="MyButton"></span>

<button x4-template="MyButton" onclick="alert('Bitte warten bis die Initialisierung abgeschlossen ist')">Button Platzhalter</button>
```

Definieren eines UI-Kontext

Den DOM-Bezugspunkt für einen UI-Kontext (siehe [UI-Bindings](#)) definieren Sie durch ein beliebiges HTML-Element mit einem `x4-ui-context`-Attribut, welches den Namen des UI-Kontext angibt. Sie können einen UI-Kontext im HTML-Code entweder in einem Template (Standardverhalten) oder inline definieren.

**Beispiel: Dynamischer UI-Kontext**

```
<div x4-ui-context="whatever"></div>
```

UI-Kontext mit Template definieren

Standardmäßig wird ein UI-Kontext für HTML-Element im `template`-Modus definiert, eine Angabe von Attribut `x4-ui-type="template"` zusätzlich zum `x4-ui-context`-Attribut ist also nicht erforderlich.

Folgende Attribute stehen zusätzlich zur Verfügung:

- `x4-template`: Template statisch als HTML-Attribut definieren oder
- `_template`: Template alternativ dynamisch als UI-Property definieren
- `x4-renderer`: optionale Angabe einer Renderer-Klasse (alternativ über UI-Property `_renderer`)

**i** Einen Renderer können Sie, im Gegensatz zum Template, jedoch auch im HTML und als UI-Property definieren, wobei das UI-Property Vorrang besitzt. Dazu muss diese JavaScript-Klasse zum Initialisierungszeitpunkt in der Seite verfügbar sein (z.B. durch Einbinden der entsprechenden JavaScript-Datei).

**Beispiel: UI-Kontext mit festem Template**

```
<div x4-ui-context="button1" x4-template="MyButton"></div>
```

UI-Kontext inline definieren

Um einen UI-Kontext inline zu definieren, hinterlegen Sie das Attribut `x4-ui-type="inline"`.

**Beispiel: Inline UI-Kontext**

```
<button x4-ui-context="button2" x4-ui-type="inline" x4-value="#prop.label"
x4-event="click: #prop.action">Default Label</button>
```

**Einbinden von Template-Collections**

Template-Collections können über ein `<link>`-Element innerhalb von `<head>` eingebunden werden. Beim Initialisieren des UI-Frameworks werden Template Collections automatisch geladen und geparkt. Der Inhalt einer Template-Collection ist eine HTML-Seite. Alle anderen Inhalte außer den Template-Scripts werden ignoriert.

**Beispiel**

```
<link x4-templates type="text/html" href="MyTemplates.html" rel="prefetch" />
```

**Einbindung von UI-Properties**

UI-Properties können Sie per JavaScript-API über `x4.ui.initPage` oder die View-Kontext-API mit `addContent` (siehe [UI-Framework Core \(ui/ui.js\)](#)), aber auch direkt in der HTML-Seite konfigurieren.

Um UI-Properties ins HTML einzubinden, wird ein `<script>`-Element mit MIME-Typ `application/json` benötigt. Dieses Element gibt mit Hilfe eines HTML-Attributs `x4-ui-property` die Model-Referenz an, unter welcher die JSON-Daten des Skripts in das Properties-Model eingefügt werden sollen (`#root`, wenn das JSON die komplette UI-Konfiguration enthält).

Sie können das JSON entweder im Skript definieren oder per `src`-Attribut von einer URL laden.

**Beispiel: Konfiguration von 2 Buttons**

```
<script x4-ui-property="button1" type="application/json">
  {
    "label": "Mein Button",
    "action": "{{#controller.doSomething}}"
  }
</script>

<script x4-ui-property="button2" type="application/json" src="external/button/
config.json">
</script>
```

**Bereitstellen von Controllern**

Einen Controller definieren Sie als JavaScript-Klasse. Diese wird automatisch beim Initialisieren eines View-Kontext instanziiert. Als einzigen Parameter wird dem Konstruktor das View-Kontext-Objekt übergeben, für welches die Controller-Instanz erzeugt wird.

Um einen Controller mit einem View-Kontext zu verknüpfen, in der UI-Property `_controller` die voll qualifizierte JavaScript-Referenz auf die Controller-Klasse als Zeichenkette hinterlegen.



**Beispiel: Native Definition eines Controllers**

```
var MyController = function(vc) {  
    this.vc = vc;  
  
    vc.on('view-ready', function(){  
        alert('View fertig initialisiert');  
    });  
};  
  
MyController.prototype = {  
    showData: function(){  
        alert( JSON.stringify( this.vc.getModel().get() ) );  
    }  
};
```

**Beispiel: Definition mit X4 OO-API**

```
var MyController = x4.Class({  
  
    INIT: function(vc) {  
        this.vc = vc;  
  
        vc.on('view-ready', function(){  
            alert('View fertig initialisiert');  
        });  
    },  
  
    PUBLIC: {  
        showData: function(){  
            alert( JSON.stringify( this.vc.getModel().get() ) );  
        }  
    }  
});
```

**Auf dieser Seite:****Siehe auch:**

- [UI-Bindings](#)
- [Standard UI-Properties](#)
- [UI-Events](#)
- [Widgets im UI-Framework](#)
- [UI-Framework Core \(ui/ui.js\)](#)

### 2.14.5.4.1.6 Standard UI-Properties

Welche UI-Properties im UI-Framework als Bestandteil der JavaScript-API des *X4 Web Designers* standardmäßig verfügbar sind und wie diese verwendet werden können

UI-Properties: View

In der folgenden Tabelle sind die UI-Properties beschrieben, die vom Framework direkt ausgewertet werden und für jede View verwendet werden können.

In diesen Properties werden keine Property-Bindings unterstützt, außer in `_event`, `_domHooks` und `_settings`.

Eigenschaft	Erläuterung	Typ
<code>_uiId</code>	Definiert die UI-ID für die View  Wenn diese Eigenschaft nicht definiert ist oder wenn mehrere Views durch die Konfiguration erzeugt werden (Array-Model), so wird eine ID generiert. Über diese ID können Sie per JavaScript-API oder UI-Binding ( <code>#frame.&lt;id&gt;</code> ) vom Frame-Objekt heraus auf das View-Kontext-Objekt zugreifen.	String
<code>_template</code>	Name des zu verwendenden Templates (sofern dies nicht bereits im HTML angegeben wurde)	String
<code>_renderer</code>	Name der zu verwendenden Renderer-Klasse  Wenn weder im HTML noch in den Properties ein Renderer definiert wurde, so wird ein Standard-Renderer verwendet, der lediglich die Elemente des Templates bereitstellt.	String
<code>_controller</code>	Name der zu verwendenden Controller-Klasse  Wenn kein Controller definiert ist, wird der nächst übergeordnete Controller verwendet.	String
<code>_dataContext</code>	Model-Referenz relativ zum Data-Model des übergeordneten View-Kontexts  Standard ist <code>#self</code> (kein Kontext-Wechsel).	String
<code>_stateContext</code>	Model-Referenz relativ zum State-Model des übergeordneten View-Kontexts  Standard ist <code>#self</code> (kein Kontext-Wechsel)	String
<code>_event</code>	Definiert Bindings für UI-Events in einem Objekt: <ul style="list-style-type: none"><li>• Jeder Schlüssel des Objekts definiert einen Event-Namen.</li><li>• Der zugehörige Wert definiert ein Property-Binding zur Behandlung dieses Events.</li></ul>	Object

Eigenschaft	Erläuterung	Typ
_domHooks	<p>Definiert Bindings für DOM-Hooks in einem Objekt</p> <ul style="list-style-type: none"> <li>• Jeder Schlüssel des Objekts definiert eine Hook-ID.</li> <li>• Der zugehörige Wert ist ein Objekt, das die Bindings für den Hook definiert. Für die hier unterstützten Attribute, siehe <a href="#">Framework-spezifische HTML-Strukturen</a> in Abschnitt "HTML-Bindings" die Spalte <i>JS-Name</i>.</li> </ul> <p>Die Hook-ID <code>_root</code> ist für alle Views verfügbar und verweist auf die Wurzel-Elemente der Views.</p>	Object
_settings	Objekt mit Konfigurations-Properties, die von untergeordneten Views geerbt werden, sofern sie dort nicht überschrieben werden.	Object
_data	Default-Daten für das Data-Model	*
_state	Default-Daten für das State-Model	*
_localState	Default-Daten für das lokale State-Model	*
_modelInitMode	<p>Definiert, wie das Model der View initialisiert wird</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• "auto": automatisch initialisieren (Standard)</li> <li>• "explicit": explizit initialisieren. Die View wird erst gerendert, wenn das Model initialisiert wurde (nicht leer ist), was beispielsweise sinnvoll ist, wenn Daten asynchron geladen werden und eine sinnvolle Darstellung der View ohne Daten nicht möglich ist.</li> </ul>	String
_modelRelation	<p>Definiert, wieviele Views bei einem Array-Model erzeugt werden sollen</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li>• "auto": automatisch mehrere Views erzeugen (Standard); Für jeden Eintrag im Array wird eine View erzeugt und bei jeder Änderung des Arrays wird die Views entsprechend angepasst.</li> <li>• "single": unabhängig von den Daten immer nur eine einzige View erzeugen</li> </ul>	String

UI-Properties: Frame

Folgende zusätzliche UI-Properties werden vom Framework lediglich auf Ebene eines Frames interpretiert:

Eigenschaft	Erläuterung	Typ
<code>_contents</code>	<p>Standardmäßiger UI-Kontext für die Wurzel-View des Frames</p> <p>Wenn dieser UI-Kontext nicht explizit im DOM des Frames vorhanden ist, wird er automatisch als erster Kind-Knoten des Wurzel-Elements eingefügt (beim Haupt-Frame als erstes Kind des <code>&lt;body&gt;</code>-Elements).</p>	Object
<code>_dependencies</code>	<p>Definiert Abhängigkeiten für den Frame (siehe <a href="#">Dependencies (ui/dependencies.js)</a>)</p> <p>Die Dependencies werden automatisch geladen, sobald der Frame initialisiert wird, falls diese nicht bereits von einem anderen Frame geladen wurden. So können Sie sicherstellen, dass die Ressourcen eines Frames erst bei der ersten Anzeigen und nicht bereits beim Initialisieren der Seite geladen werden.</p>	Object

#### Standard-Attribute: Settings

In *Settings* werden folgende Standard-Attribute unterstützt, die direkt vom Framework ausgewertet werden.

Eigenschaft	Erläuterung	Typ
<code>_clearEmptyInputValues</code>	<p>Steuert den Umgang mit leeren Formularfeldwerten</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"> <li><code>true</code>: Bindings für Input-Elemente liefern bei einem "leeren Wert" den Wert <code>undefined</code>, beispielsweise eine leere Zeichenkette bei Textfeldern, <code>false</code> bei Checkbox) Leere Werte werden nicht in JSON-Daten repräsentiert, was bei großen Formularen mit vielen optionalen Feldern nützlich sein kann.</li> <li><code>false</code>: Leere Werte werden unverändert vom Formularfeld ins Binding übergeben, beispielsweise eine leere Zeichenkette bei einem leeren Textfeld (Standard)</li> </ul>	Boolean
<code>_inputChangeEvents</code>	<p>Definiert, bei welchen Events-Typen Bindings für Input-Elemente aktualisiert werden.</p> <p>Die Event-Typen werden als Array erwartet, wobei jeder Eintrag ein Objekt mit dem Attribut <code>type</code> enthält, das den Namen eines nativen Events enthält.</p> <p>Standard sind <code>"change"</code> und <code>"keyup"</code>.</p>	Array

**Auf dieser Seite:****Siehe auch:**

- [UI-Framework Core \(ui/ui.js\)](#)
- [Frames \(ui/frames.js\)](#)
- [Dependencies \(ui/dependencies.js\)](#)

#### 2.14.5.4.1.7 UI-Events

UI-Events werden mit dem UI-Framework (als Bestandteil der JavaScript-API des *X4 Web Designers*) im Kontext einer View ausgelöst. Neben der Möglichkeit anwendungsspezifische Events auszulösen (siehe [UI-Framework Core \(ui/ui.js\)](#)), werden folgende Standard-Events bereitgestellt.

Event	Auslösezeitpunkt	Data-Attribute
view-prepare	Direkt nach der View-Initialisierung (nachdem Controller und Renderer erzeugt wurden), aber bevor die render-Methode des Renderers aufgerufen und die Standard-Properties für DOM-Hooks ausgewertet wurden	–
view-attach	Bevor die Elemente der View in das DOM eingefügt und durch das UI-Framework geparkt werden, jedoch nachdem die render-Methode des Renderers aufgerufen und die Standard-Properties für DOM-Hooks ausgewertet wurden	–
view-bind	Bevor alle bis dahin bekannten UI-Bindings (die durch den DOM oder API definiert wurden) für die View angewendet werden	–
view-ready	Nachdem die View-Initialisierung abgeschlossen ist	–
view-unbind	Erstes Event beim Entfernen einer View, bevor Inhalte der View entfernt werden	–
view-detach	Bevor die View und alle untergeordneten Views aus dem DOM entfernt werden	–
view-cleanup	Bevor die View aus der UI-Hierarchie gelöscht und alle internen Referenzen entfernt werden (wird nur beim Entfernen der View, nicht bei repaint, ausgelöst)	–
view-add-child	Wenn eine Kind-View in die View eingefügt wird, bevor das view-prepare-Event der Kind-View ausgelöst wird	<b>uiContext:</b> ID des UI-Kontexts, in welchem die Kind-View erzeugt wird  <b>childContext:</b> View-Kontext-Objekt der Kind-View
view-remove-child	Wenn eine Kind-View entfernt wird, direkt bevor das view-unbind-Event der Kind-View ausgelöst wird	siehe view-add-child

#### 2.14.5.4.1.8 Widgets im UI-Framework

Ein Widget im UI-Framework (als Bestandteil der JavaScript-API des *X4 Web Designers*) dient als Vorlage für die Konfiguration einer View. Es besteht aus mindestens einem Template und optionalen Renderer-Klassen. Außerdem gehört zu einem Widget ein JSON-Schema. Dieses konfiguriert UI-Properties, die von Template und Renderern unterstützt werden.

Ein Widget kann unterschiedliche Darstellungen/Funktionalitäten durch verschiedene Modi bereitstellen, wobei jeder Modus eine andere Kombination aus einem Template und einem optionalen Renderer definieren kann.

Im JSON-Schema und den Modus-Objekten können Beschreibungstexte in einem `description`-Attribut hinterlegt werden. Diese werden dann im UI-Editor des *X4 Web Designers* als Tooltip angezeigt.

Widgets können per JavaScript-API oder über eine Widget-Collection konfiguriert werden.

##### Widget-Collections

Mehrere Widgets können zu einer Widget-Collection zusammengefasst werden, um diese einfacher verwalten zu können. Eine Widget-Collection enthält eine Liste von Widgets und die für die Widgets benötigten Dependencies (siehe unten). In den Dependencies können sowohl die HTML und JavaScript-Dateien, die die Templates und Renderer für die Widgets enthalten, als auch externe Bibliotheken/Frameworks definiert werden. Der Dependency-Manager sorgt automatisch dafür, dass auch Abhängigkeiten, die in mehreren Collections definiert sind, pro Seite nur einmal geladen werden.

Widget-Collections können per JavaScript-API oder durch JSON-Konfigurationen, als `script`-Elemente im HTML, bereitgestellt werden.

Beispiel für eine Widget-Definition

**Beispiel in HTML**

```
<script x4-widgets type="application/json">
{
  "name": "my.demo",
  "title": "My Demo",
  "widgets": [
    {
      "name": "MyWidget",
      "title": "My Widget",
      "baseSchema": {
        "type": "object",
        "description": "This is a super fancy widget!",
        "properties": {
          "myTitle": {
            "type": "string",
            "description": "Defines a title for the widget content."
          },
          ...
        }
      },
      "modes": [
        {
          "name": "Default",
          "template": "MyTemplate",
          "renderer": "my.demo.Renderer",
          "description": "Simple default presentation"
        }
      ]
    },
    ...
  ],
  "dependencies": {
    "templates": [{ "url": "/Resources/widgets/demo.html" }],
    "js": [{ "url": "/Resources/widgets/demo.js" }],
    "css": [{ "url": "/Resources/widgets/demo.css" }]
  }
}
</script>

<script x4-widgets type="application/json" src="external/widgets/my.widgets.json">
</script>
```

**Auf dieser Seite:**

**Siehe auch:**

#### 2.14.5.4.2 UI-Framework Core (ui/ui.js)

Die Javascript-Datei `ui/ui.js` ist Bestandteil der JavaScript-API für den *X4 Web Designer* und stellt die grundlegende API für dessen UI-Framework bereit.

### 2.14.5.4.2.1 Grundlegende API-Funktionen des UI-Frameworks

#### initPage

Diese Methode initialisiert das UI-Framework für die HTML-Seite, in welcher die Methode aufgerufen wird. Dazu zählt das Erzeugen und Initialisieren eines Frames (*Main-Frame*), dessen Wurzel-View das body-Element der Seite ist. Eine weitere Aktion ist das Parsen des Document Object Models (DOM) und eine entsprechende Behandlung der Custom-Attribute des UI-Frameworks.

**i** Beachten Sie:

- Rufen Sie diese Methode erst auf, wenn der HTML-DOM der HTML-Seite bereit ist.
- Wenn auf dem body-Element der Seite das Attribut `x4-init="ui"` definiert ist, wird die Methode automatisch aufgerufen, wenn der DOM erzeugt wurde.

```
x4.ui.initPage([properties: Object [, data: <PlainData>]])
```

Parameter	Erläuterung
<code>properties</code>	Optionales Objekt, welches die UI-Properties für den Main-Frame definiert  Alternativ können Sie die Properties auch über ein Properties-Skript konfigurieren.
<code>data</code>	Definiert die initialen Daten für das Data-Model des Frames  Wenn dieser Parameter fehlt, wird zunächst ein leeres Model erzeugt, das Sie bei Bedarf nachträglich füllen können.

#### Beispiel mit jQuery


```
$(document).ready(function(){  
    x4.ui.initPage(myProperties, myData);  
});
```

#### addInitAction

Diese Methode registriert eine Funktion, die während der Initialisierung des UI-Frameworks aufgerufen wird, nachdem der DOM geparkt wurde, aber bevor die Views initialisiert werden. Damit können beliebige Vorbereitungen für die Initialisierung der Views durchgeführt werden, beispielsweise das Laden von Abhängigkeiten. Wenn die Initialisierung zum Registrierungszeitpunkt bereits abgeschlossen ist, wird die Funktion sofort aufgerufen.

```
x4.ui.addInitAction(action: Function)
```



Parameter	Erläuterung
action	<p>Die Funktion, die während der Initialisierungs-Phase ausgeführt werden soll</p> <div style="border: 1px solid red; padding: 10px; margin-top: 10px;"> <p> <b>Achtung:</b> Die registrierte Funktion erhält als einzigen Parameter ein Callback-Funktion, die aufgerufen werden muss, sobald die Aktion abgeschlossen ist. Andernfalls wird die Initialisierung der UI-Frameworks nicht fortgeführt.</p> </div>

### Beispiel: Dynamisches Erzeugen eines Templates

```
x4.ui.addInitAction(function(ready){
    var $button = $(x4.ui.getTemplate('Button').provideElements());
    if(useSuperButton){
        $button.addClass('super-button');
    }
    x4.ui.addTemplate(x4.ui.createTemplate('MyButton', $button.toArray()));

    ready();
});
```

ready

Diese Methode registriert eine Funktion, die während der Initialisierung des Frameworks aufgerufen wird, nachdem der DOM geparkt wurde, aber bevor die Views initialisiert werden. Damit können beliebige Vorbereitungen für die Initialisierung der Views durchgeführt werden, beispielsweise das Laden von Abhängigkeiten. Wenn die Initialisierung zum Registrierungszeitpunkt bereits abgeschlossen ist, wird die Funktion sofort aufgerufen.

x4.ui.ready(handler: Function)

Parameter	Erläuterung
handler	Funktion, die aufgerufen werden soll, wenn das Framework fertig initialisiert ist

### Beispiel

```
x4.ui.ready(function(){
    alert('Fertig!');
});
```

getMainFrame

Diese Methode liefert den Main-Frame, der bei Initialisierung des Frameworks erzeugt wurde (siehe API-Beschreibung unten).

x4.ui.getMainFrame(): <Frame>

**Rückgabewert:** Das Objekt des Main-Frames

createFrame

Diese Methode erzeugt ein neues Frame-Objekt.


`x4.ui.createFrame(element: HTMLElement): <Frame>`

Parameter	Erläuterung
element	Das DOM-Element, welches als Wurzel-Element für die Wurzel-View des Frames verwendet werden soll.

**Rückgabewert:** Das erzeugte Frame-Objekt

createTemplate

Diese Methode erzeugt ein neues Template-Objekt.

 Das Template-Objekt wird nicht automatisch registriert, siehe addTemplate.

`x4.ui.createTemplate(name: string, elements: Array): <Template>`

Parameter	Erläuterung
name	Name des Templates
elements	Array von DOM-Elementen, die den Inhalt des Templates definieren

**Rückgabewert:** Erzeugtes Template-Objekt

addTemplate

Diese Methode registriert ein Template, sodass Sie dieses in UI-Properties oder über die API (siehe getTemplate) verwenden können.

`x4.ui.addTemplate(template: <Template>)`

Parameter	Erläuterung
template	Template-Objekt

getTemplate

Diese Methode ermöglicht den Zugriff auf ein registriertes Template.

`x4.ui.getTemplate(name: string): <Template>`

Parameter	Erläuterung
name	Name des Templates

**Rückgabewert:** Template-Objekt

getTemplateNames

Diese Methode liefert die Namen aller registrierten Templates.

`x4.ui.getTemplateNames(): Array`

**Rückgabewert:** Array, das die Namen aller registrierten Templates enthält

#### 2.14.5.4.2.2 Template-API

getName

Diese Methode liefert den Namen des Templates.

```
<template>.getName(): string
```

**Rückgabewert:** Template-Name als Zeichenkette

provideElements

Diese Methode stellt eine Kopie der Elemente des Templates bereit.

```
<template>.provideElements(): Array
```

**Rückgabewert:** DOM-Elemente des Templates als Array

#### 2.14.5.4.2.3 Frame-API

init

Diese Methode lädt die Abhängigkeiten (Dependencies) des Frames und erzeugt die Wurzel-View.

 Der Wurzel-View wird nicht automatisch in das DOM eingefügt, siehe attach.

```
<frame>.init(properties: <Model>, dataModel: <Model>, ready: Function)
```

Parameter	Erläuterung
properties	Model-Instanz, welche die UI-Properties für den Frame enthält
dataModel	Model-Instanz, welche die Daten für den Frame enthält
ready	Funktion, die aufgerufen wird, wenn die Initialisierung des Frames abgeschlossen ist

getViewContext

Diese Methode ermöglicht den Zugriff auf das Kontext-Objekt eines Views innerhalb des Frames.

```
<frame>.getViewContext(id: string): <ViewContext>
```

Parameter	Erläuterung
id	UI-ID der View (wird durch UI-Properties konfiguriert)

**Rückgabewert:** Kontext-Objekt der View

getProperties

Diese Methode liefert das Properties-Model des Frames.

```
<frame>.getProperties(): <Model>
```

**Rückgabewert:** *Model*-Objekt, welches die UI-Properties des Frames enthält

getModel

Diese Methode liefert das Data-Model des Frames.

```
<frame>.getModel(): <Model>
```

**Rückgabewert:** *Model*-Objekt, welches die Daten des Frames enthält

getStateModel

Diese Methode liefert das State-Model des Frames.

```
<frame>.getStateModel(): <Model>
```

**Rückgabewert:** Das *Model*-Objekt, welches die UI-Zustandsdaten des Frames enthält

**attach**

Diese Methode initialisiert die Views des Frames und fügt sie in das DOM ein.

```
<frame>.attach()
```

**detach**

Diese Methode entfernt alle Views des Frames.

```
<frame>.detach()
```

**repaint**

Diese Methode aktualisiert sämtliche Frame-Inhalte, indem alle Views entfernt und neu erzeugt werden. Dies entspricht einem detach-Aufruf gefolgt von attach.

```
<frame>.repaint()
```

**cleanup**

Diese Methode löscht die internen Referenzen des Frames.

```
<frame>.cleanup()
```



**Achtung:** Dadurch wird der Frame unbrauchbar!

Führen Sie zunächst detach aus, um ein korrektes Entfernen der Views sicherzustellen.

#### 2.14.5.4.2.4 View-Kontext-API

**getViewId**

Diese Methode liefert die UI-ID der View.

```
<vc>.getViewId(): string
```

**Rückgabewert:** ID der View als Zeichenkette

**getFrame**

Diese Methode liefert den Frame, in dem sich die View befindet.

```
<vc>.getFrame(): <Frame>
```

**Rückgabewert:** Frame-Objekt

**getElements**

Diese Methode liefert die DOM-Elemente der View.

```
<vc>.getElements(): Array
```

**Rückgabewert:** DOM-Elemente als Array

**getTemplate**

Diese Methode liefert das Template, welches für die Erzeugung der View verwendet wird/wurde.

```
<vc>.getTemplate(): <Template>
```

**Rückgabewert:** Template-Objekt

**getAPI**

Diese Methode liefert die Renderer-Instanz, die für die Erzeugung dieser View verwendet wird/wurde. Diese Renderer-Instanz kann Methoden für UI-Aktionen bereitstellen (beispielsweise das Öffnen eines Dialogs oder Ergänzen einer Registerkarte in einem entsprechenden Panel).

```
<vc>.getAPI(): Object
```

**Rückgabewert:** Renderer-Objekt

getProperties

Diese Methode liefert das Properties-Model der View.

```
<vc>.getProperties(): <Model>
```

**Rückgabewert:** *Model*-Objekt, das die UI-Properties des Frames enthält

getSettings

Diese Methode liefert das Settings-Model mit Einstellungen der View. Einstellungen werden vom übergeordneten View vererbt. Sie können diese jedoch überschreiben. Nach dem ersten Zugriff auf das Settings-Model werden Änderungen an den Einstellungen der übergeordneten Views nicht mehr berücksichtigt. Settings werden wie Properties behandelt. Es sind also auch Property-Bindings in Settings-Einträgen möglich.

```
<vc>.getSettings(): <Model>
```

**Rückgabewert:** Model-Objekt, das die UI-Properties des Frames enthält

hasSettings

Diese Methode prüft, ob für die View eigene Einstellungen definiert wurden, oder ob nur übergeordnete vererbt werden.

```
<vc>.hasSettings(): boolean
```

**Rückgabewert:** true, wenn eigene Settings vorhanden sind, andernfalls false

searchInSettings

Diese Methode sucht nach einem Settings-Eintrag unter einer bestimmten Model-Referenz.

```
<vc>.searchInSettings(ref: string): <PlainData>
```

Parameter	Erläuterung
ref	Model-Referenz

**Rückgabewert:** Daten des Settings-Eintrags oder undefined, wenn kein Eintrag gefunden wurde

getModel

Diese Methode liefert das Data-Model im Kontext der View.

```
<vc>.getModel(): <Model>
```

**Rückgabewert:** Model-Objekt, welches die Daten des View-Kontext enthält

getStateModel

Diese Methode liefert das State-Model im Kontext der View.

```
<vc>.getStateModel(): <Model>
```

**Rückgabewert:** Model-Objekt, welches die UI-Zustandsdaten im Kontext der View enthält

getLocalStateModel

Diese Methode liefert das lokale State-Model der View.

```
<vc>.getLocalStateModel(): <Model>
```

**Rückgabewert:** Model-Objekt, welches die lokalen UI-Zustandsdaten der View enthält

getController

Diese Methode liefert die Controller-Instanz des View-Kontext. Wenn kein Controller definiert ist, wird der direkt übergeordnete Controller in der UI-Hierarchie zurückgegeben. Wenn in der kompletten übergeordneten Hierarchie nirgends ein Controller definiert wurde, so liefert die Methode null.

```
<vc>.getController(): string
```

**Rückgabewert:** Controller-Objekt oder null

getParent

Diese Methode liefert den übergeordneten View-Kontext.

```
<vc>.getParent(): <ViewContext>
```

**Rückgabewert:** View-Kontext-Objekt

getRoot

Diese Methode liefert den Wurzel-View-Kontext des Frames.

```
<vc>.getRoot(): <ViewContext>
```

**Rückgabewert:** View-Kontext-Objekt

getChildren

Diese Methode liefert die View-Kontexte der Views, welche sich direkt in einem bestimmten UI-Kontext befinden.

```
<vc>.getChildren(uiContextKey: string [, filter: Function]): Array
```

Parameter	Erläuterung
uiContextKey	Name des UI-Kontext
filter	<p>Optionale Filter-Funktion, die als einzigen Parameter die zu prüfenden View-Kontext-Objekte erhält.</p> <p>Wenn die Funktion true als JavaScript-Wert zurückgibt, so wird der entsprechende View-Kontext in das Ergebnis-Array aufgenommen, andernfalls nicht.</p> <p>Wenn die Funktion false zurückgibt, so wird die Suche abgebrochen.</p>

**Rückgabewert:** Array von View-Kontext-Objekten

findAncestors

Diese Methode liefert die View-Kontexte übergeordneter Views.

```
<vc>.findAncestors([filter: Function]): Array
```

Parameter	Erläuterung
filter	<p>Optionale Filter-Funktion, die als einzigen Parameter die zu prüfenden View-Kontext-Objekte erhält.</p> <p>Wenn die Funktion <code>true</code> als JavaScript-Wert zurückgibt, so wird der entsprechende View-Kontext in das Ergebnis-Array aufgenommen, andernfalls nicht.</p> <p>Wenn die Funktion <code>false</code> zurückgibt, so wird die Suche abgebrochen.</p>

**Rückgabewert:** Array von View-Kontext-Objekten

findDescendants

Liefert die View-Kontexte untergeordneter Views.

`<vc>.findDescendants([filter: Function]): Array`

Parameter	Erläuterung
filter	<p>Optionale Filter-Funktion, die als einzigen Parameter die zu prüfenden View-Kontext-Objekte erhält.</p> <p>Wenn die Funktion <code>true</code> als JavaScript-Wert zurückgibt, so wird der entsprechende View-Kontext in das Ergebnis-Array aufgenommen, andernfalls nicht.</p> <p>Wenn die Funktion <code>false</code> zurückgibt, so wird die Suche abgebrochen.</p>

**Rückgabewert:** Array von View-Kontext-Objekten

attach

Diese Methode fügt die View in das DOM ein. Diese müssen Sie nicht explizit aufrufen, es sei denn, zuvor wurde ein `detach`-Aufruf durchgeführt.

`<vc>.attach()`

detach

Diese Methode entfernt die View und alle untergeordneten Views aus dem DOM.

`<vc>.detach()`

repaint

Diese Methode aktualisiert die View, indem alle Views entfernt und neu erzeugt werden. Dies entspricht einem `detach`-Aufruf gefolgt von `attach`.

`<vc>.repaint()`

remove

Diese Methode entfernt die View und sämtliche verknüpften Kontext-Ressourcen.

`<vc>.remove()`

on

Diese Methode registriert einen Handler für ein UI-Event.

```
<vc>.on(eventType: string, handler: Function): <handler>
```

Parameter	Erläuterung
eventType	Beliebige Zeichenkette, die den Event-Typ definiert
handler	Funktion, die ausgeführt wird, wenn das Event ausgelöst wurde

**Rückgabewert:** Funktion, die als Handler übergeben wurde

off

Diese Methode entfernt einen Event-Handler oder alle Handler eines Typs.

```
<vc>.off(eventType: string [, handler: Function]): boolean
```

Parameter	Erläuterung
eventType	Beliebige Zeichenkette, die den Event-Typ definiert
handler	Handler-Funktion, die entfernt werden soll

**Rückgabewert:** true, wenn das Entfernen erfolgreich war, andernfalls false (beispielsweise wenn der Handler bereits entfernt wurde)

fire

Diese Methode löst ein Event aus.

```
<vc>.fire(eventType: string [, settings: Object])
```

Parameter	Erläuterung	
eventType	Beliebige Zeichenkette, die den Event-Typ definiert	
settings	Objekt, das Konfiguration für das Event enthält	
	Attribut	Erläuterung
	bubbles	Legt fest, ob das Event auch an übergeordnete Views weitergereicht wird  <i>Mögliche Werte:</i> <ul style="list-style-type: none"><li>• true(Standard)</li><li>• false</li></ul>
	data	Beliebige Daten, die über das Event-Objekt abrufbar sein sollen

addContent

Diese Methode fügt Inhalte (Views) in einen bestimmten UI-Kontext ein. Die Funktion sollte erst aufgerufen werden, wenn die View fertig initialisiert wurde.

```
<vc>.addContent(uiContextKey: string, properties: Object|<Model> [, index: integer])
```

Parameter	Erläuterung
uiContextKey	Name des UI-Kontexts



Parameter	Erläuterung
properties	<p>UI-Properties, die den einzufügenden Inhalt konfigurieren</p> <p>Sie können entweder ein einzelnes Properties-Objekt oder ein Array von Properties-Objekten übergeben.</p> <p>Die Properties können Sie entweder als Daten-Objekt oder als Model-Instanz übergeben.</p>
index	<p>Optionale Indexangabe, an welcher View-Position der neue Inhalt eingefügt werden soll.</p> <p>Wenn die Indexangabe fehlt, wird der Inhalt am Ende angefügt.</p>

removeContent

Diese Methode entfernt eine View aus einem UI-Kontext.

```
<vc>.removeContent(uiContextKey: string, index: integer)
```

Parameter	Erläuterung
uiContextKey	Name des UI-Kontexts
index	<p>Index der zu entfernenden View</p> <p>Diesen können Sie auf dem View-Kontext-Objekt der zu entfernenden View durch getIndexInParentContent ermitteln.</p>

countContents

Diese Methode liefert die Anzahl an Views in einem bestimmten UI-Kontext.

```
<vc>.countContents(uiContextKey: string): integer
```

Parameter	Erläuterung
uiContextKey	Name des UI-Kontexts

**Rückgabewert:** Anzahl der Views

getIndexInParentContent

Diese Methode liefert die Position der View (beginnend bei 0), innerhalb des UI-Kontexts, in dem sie sich befindet.

```
<vc>.getIndexInParentContent(): integer
```

**Rückgabewert:** Die 0-basierte Position

getDomHook

Diese Methode ermöglicht den Zugriff auf ein DOM-Hook-Objekt der View.

```
<vc>.getDomHook(id: string): <DomHook>
```

Parameter	Erläuterung
id	ID des DOM-Hooks innerhalb der View.  Mit "_root" können Sie auf den Standard-Hook für die Wurzel-Elemente der View zugreifen.

**Rückgabewert:** DOM-Hook-Objekt

removeDomHook

Diese Methode entfernt einen DOM-Hook der View.

```
<vc>.removeDomHook(id: string)
```

Parameter	Erläuterung
id	ID des DOM-Hooks innerhalb der View

parseElements

Diese Methode parst DOM-Elemente im Kontext der View. Dadurch können Sie auch Elemente, die sich im DOM an anderer Position befinden oder nach dem Rendern der View dynamisch ergänzt werden sollen, im Kontext dieser View binden. Darin können auch UI-Kontext- und DOM-Hook-Elemente enthalten sein.

```
<vc>.parseElements(elements: Array)
```

Parameter	Erläuterung
elements	Array von DOM-Elementen

resetElements

Diese Methode dient als Gegenstück zu parseElements und entfernt die durch das Parsen erzeugten UI-Kontexte, DOM-Hooks, Event-Handler usw.

```
<vc>.resetElements(elements: Array)
```

Parameter	Erläuterung
elements	Array von DOM-Elementen, das zuvor für parseElements verwendet wurde

execBinding

Diese Methode ermöglicht die einmalige Ausführung eines UI-Bindings.

```
<vc>.execBinding(binding: string [, params: Object]): *
```

Parameter	Erläuterung
binding	Zeichenkette des UI-Bindings
params	Objekt, das Parameter für die Binding-Ausführung definiert  Diese Parameter werden in Binding-Funktionen als Funktionsparameter bereitgestellt. Der Attribut-Name definiert dabei den Parameter-Namen und der Attribut-Wert den Parameter-Wert.  Durch den Parameter _value können Sie den Wert eines Model-Bindings ändern.

**Rückgabewert:** Ergebnis der Binding-Ausführung

applyBinding

Diese Methode registriert eine Handler-Funktion, die den Wert eines Bindings als Parameter erhält. Diese Handler-Funktion wird dann automatisch bei jeder Ausführung des Bindings aufgerufen (beispielsweise durch Anpassung von Model-Einträgen, auf die das Binding zugreift). Bei der Registrierung wird die Funktion einmalig aufgerufen und erhält den aktuellen Wert des Bindings.

```
<vc>.applyBinding(binding: string, valueHandler: Function [, params: Object]):  
string
```

Parameter	Erläuterung
binding	Zeichenkette des UI-Bindings
valueHandler	Funktion, die den Wert des Bindings als ersten Parameter erhält. Als zweiter Parameter wird einen boolescher Wert übergeben, der angibt ob, es sich um den initialen Aufruf handelt ( true ) oder nicht ( false ).
params	Siehe params-Beschreibung von execBinding

**Rückgabewert:** ID, unter welcher das Binding registriert wurde

triggerBinding

Diese Methode ermöglicht die explizite Ausführung eines Bindings, das mit applyBinding registriert wurde.

```
<vc>.triggerBinding(id: string)
```

Parameter	Erläuterung
id	ID des Bindings

removeBinding

Diese Methode entfernt ein Binding, das mit applyBinding registriert wurde.

```
<vc>.removeBinding(id: string)
```

Parameter	Erläuterung
id	ID des Bindings

get

Diese Methode ermöglicht den Zugriff auf eine Kontext-Variable der View. Für die verfügbaren Kontext-Variablen siehe [UI-Bindings](#) im Abschnitt *Kontext-Referenzen*.

```
<vc>.get(contextKey: string): *
```

Parameter	Erläuterung
contextKey	Name der Kontext-Variable

**Rückgabewert:** Wert der Kontext-Variable


access

Diese Methode ermöglicht den Zugriff auf den Inhalt einer Kontext-Variable der View.

```
<vc>.access(contextKey: string, contentRef: string [, params: Object]): *
```

Parameter	Erläuterung
contextKey	Name der Kontext-Variable
contentRef	Inhalts-Referenz
params	Optionale Parameter, siehe params-Beschreibung von execBinding

**Rückgabewert:** Wert der Kontext-Variable

 Diese Methode wird zur Ausführung von UI-Bindings verwendet.  
*Beispiel:* `vc.execBinding('#model.test.data')` entspricht `vc.access('model', 'test.data')`.

#### 2.14.5.4.2.5 UI-Event-API

`getType`

Diese Methode liefert den Event-Typ.

```
<uiEvent>.getType(): string
```


**Rückgabewert:** Event-Typ als Zeichenkette

`getData`

Diese Methode liefert Daten, die beim Auslösen des Events definiert wurden.

```
<uiEvent>.getData([key: string]): *
```

Parameter	Erläuterung
key	Name eines Attributs im Daten-Objekt  Wenn dieses Attribut fehlt, werden die kompletten Event-Daten zurückgegeben.

 Der Aufruf `event.getData(key)` entspricht `event.getData()[key]`, jedoch wird in der ersten Variante keine Exception ausgelöst, wenn für das Event kein Daten-Objekt definiert wurde.

**Rückgabewert:** Event-Daten

`getSource`

Diese Methode liefert den initialen View-Kontext des Events.

```
<uiEvent>.getSource(): <ViewContext>
```


**Rückgabewert:** View-Kontext-Objekt, von welchem das Event ausgelöst wurde

`getSourceElement`

Diese Methode liefert – wenn vorhanden – das DOM-Element, durch das das Event ausgelöst wurde.

```
<uiEvent>.getSourceElement(): <DomElement>
```

**Rückgabewert:** DOM-Element, von dem aus das Event ausgelöst wurde.

 Das DOM-Element ist nur verfügbar, wenn das Event durch ein direktes Event-UI-Binding auf einem DOM-Element ausgelöst wurde. Dabei wird automatisch das native Event in den Daten des UI-Events unter `_parameters.event` abgelegt.

**bubbles**

Diese Methode prüft, ob das Event an übergeordnete Views weitergeleitet wird.

```
<uiEvent>.bubbles(): boolean
```

**Rückgabewert:** true, wenn das Event an übergeordnete Views weitergeleitet wird, andernfalls false

**stopPropagation**

Diese Methode verhindert, dass das Event an weitere übergeordnete Views weitergeleitet wird.

```
<uiEvent>.stopPropagation()
```

#### 2.14.5.4.2.6 DOM-Hook-API

**getId**

Diese Methode liefert die ID des DOM-Hooks.

```
<domHook>.getId(): string
```

**Rückgabewert:** Die ID als Zeichenkette

**bind**

Diese Methode bindet DOM-Elemente an den Hook. Die Elemente und Sub-Elemente werden automatisch im Kontext der View, zu der dieser Hook gehört, geparkt (Binding-Attribute, UI-Kontexte, weitere DOM-Hooks).

Verwenden Sie diese Methode nur, wenn in der View kein Hook-Element für den Hook definiert wurde.

```
<domHook>.bind(elements: Array)
```

Parameter	Erläuterung
elements	Array von DOM-Elementen, die an den Hook gebunden werden sollen

**unbind**

Diese Methode entfernt die Elemente des Hooks. Dabei werden auch alle Bindings, UI-Kontexte usw. entfernt, die durch den Hook erzeugt wurden.

Anschließend können Sie bei Bedarf ein erneutes bind mit anderen Elementen durchführen, wobei zuvor definierte Bindings wieder auf die neuen Elemente angewendet werden.

```
<domHook>.unbind()
```

**getElements**

Diese Methode liefert die Elemente des Hooks.

```
<domHook>.getElements(): Array
```

**Rückgabewert:** DOM-Elemente des Hooks als Array

value

Diese Methode setzt oder liest den Wert der DOM-Elemente. Bei Input-Elementen wird der Wert des Input-Feldes gesetzt – abhängig vom Typ. Bei sonstigen DOM-Elementen wird der Wert als Textknoten eingefügt, welcher ggf. andere Kindknoten ersetzt. Beim Auslesen des Wertes wird nur der Wert des ersten Elements gelesen.

```
<domHook>.value([value]): string|undefined
```

Parameter	Erläuterung
value	Neuer Wert für die Elemente des Hooks

**Rückgabewert:** Wert des Elements oder undefined, wenn keine Elemente vorhanden sind, oder wenn die Methode als *Setter* verwendet wird

attr

Diese Methode setzt oder liest den Wert eines Attributs der DOM-Elemente.

```
<domHook>.attr(attrName: string [, value]): string|undefined
```

Parameter	Erläuterung
attrName	Name des Attributs
value	Neuer Attributwert  Beachten Sie: <ul style="list-style-type: none"> <li>Bei null, undefined oder false wird das Attribut entfernt.</li> <li>Bei true wird der Attributname als Attributwert verwendet.</li> </ul>

**Rückgabewert:** Wert des Attributs oder undefined, wenn keine Elemente vorhanden sind, oder wenn die Methode als *Setter* verwendet wird

css

Diese Methode setzt oder liest eine CSS-Eigenschaft der DOM-Elemente.

```
<domHook>.css(cssProp: string [, value]): string|undefined
```

Parameter	Erläuterung
cssProp	Name der CSS-Eigenschaft
value	Neuer CSS-Wert  Durch null, undefined, false oder eine leere Zeichenkette wird die CSS-Eigenschaft entfernt.

**Rückgabewert:** CSS-Wert oder undefined, wenn keine Elemente vorhanden sind, oder wenn die Methode als *Setter* verwendet wird

cssClass

Diese Methode fügt eine CSS-Klasse zu den DOM-Elementen hinzu, oder prüft deren Existenz.

`<domHook>.cssClass(baseName: string [, value]): boolean|undefined`

Parameter	Erläuterung
baseName	(Basis-)Name der CSS-Klasse
value	Wenn der Wert eine nicht-leere Zeichenkette ist, wird der Wert an den Klassennamen angehängt und ggf. zuvor gesetzte Klassen mit gleichem Basis-Namen entfernt.  Andernfalls wird die Klasse durch einen wahren Wert eingefügt, durch einen nicht-wahren Wert entfernt.

**Rückgabewert:** `true` wenn die Klasse vorhanden ist, `false` wenn sie nicht vorhanden ist oder `undefined`, wenn die Methode als *Setter* verwendet wird

`showIf`

Diese Methode blendet die Elemente des Hooks ein oder aus.

`<domHook>.showIf(value)`

Parameter	Erläuterung
value	Wenn der Wert wahr ist, werden die Elemente angezeigt, oder andernfalls mittels <code>display</code> -Eigenschaft von CSS ausgeblendet

`renderIf`

Diese Methode fügt die Elemente des Hooks in das DOM ein oder entfernt diese.

`<domHook>.renderIf(value)`

Parameter	Erläuterung
value	Wenn der Wert wahr ist, werden die Elemente eingefügt, andernfalls werden Elemente entfernt.

`on`

Diese Methode registriert einen nativen Event-Handler für die Hook-Elemente.

`<domHook>.on(eventType: string, handler: Function): Function`

Parameter	Erläuterung
eventType	Name des nativen DOM-Events
handler	Event-Handler-Funktion

**Rückgabewert:** Handler-Funktion, die an die Methode übergeben wurde

`off`

Diese Methode entfernt einen oder mehrere Event-Handler.

Beachten Sie:

- Wenn Sie die Methode ohne Parameter aufrufen, werden sämtliche Event-Handler aller Event-Typen entfernt.
- Wenn Sie nur einen Event-Typ-Parameter übergeben, werden alle Handler dieses Typs entfernt.
- Wenn Sie zusätzlich noch die Funktion eines bestimmten Handlers übergeben, wird nur dieser entfernt.


```
<domHook>.off([eventType: string [, handler: Function]])
```

Parameter	Erläuterung
eventType	Name des nativen DOM-Events
handler	Zu entfernende Event-Handler-Funktion

fire

Diese Methode ruft die Event-Handler eines bestimmten Event-Typs für jedes Wurzel-Element des Hooks auf.

Über die `this`-Referenz können Sie innerhalb der Handler auf das DOM-Element zugreifen, für welches der Handler aufgerufen wurde.

 **Achtung:** Dadurch wird kein natives Event ausgelöst, sondern nur Handler aufgerufen, die über die DOM-Hook-API registriert wurden.

```
<domHook>.fire(eventType: string)
```

Parameter	Erläuterung
eventType	Name des nativen DOM-Events
handler	Zu entfernende Event-Handler-Funktion

applyBindings

Diese Methode registriert Bindings für die Wurzel-Elemente des Hooks.

```
<domHook>.applyBindings(bindings: Object)
```



Parameter	Erläuterung		
bindings	Objekt, das Attribute für die verschiedenen Binding-Typen enthalten kann		
	Binding	Ziel-Funktion	Erläuterung
	value	value	UI-Binding-Zeichenkette
	attr	attr	Objekt, bei dem jedes Attribut ein UI-Binding für ein DOM-Attribut definiert
	event	on	Objekt, bei dem jedes Attribut ein UI-Binding für ein DOM-Event definiert
	css	css	Objekt, bei dem jedes Attribut ein UI-Binding für eine CSS-Eigenschaft definiert
	cssClass	cssClass	Objekt, bei dem jedes Attribut ein UI-Binding für eine CSS-Klasse definiert
	showIf	showIf	UI-Binding-Zeichenkette
	renderIf	renderIf	UI-Binding-Zeichenkette

### Beispiel

```
vc.getDomHook('myInput').applyBindings({
  value: "#prop.value",
  showIf: "#frame.otherInput.#prop.value",
  attr: {
    disabled: "#state.readonly"
  },
  event: {
    change: "#controller.myInputChanged"
  },
  css: {
    color: "#settings.inputTextColor"
  },
  cssClass: {
    "static-class": true,
    "my-input-": "#state.myInputClassExtension"
  }
});
```

#### 2.14.5.4.2.7 Utils

##### accessModelProperty

Ermöglicht Zugriff auf eine Objekt-Proprety eines Data Models in einem UI Binding. Dabei werden im Gegensatz zu der normalen Binding-Syntax auch Property-Namen/Schlüssel unterstützt, die nicht der in der Binding-Syntax erwarteten Namenskonvention entsprechen und z. B. Sonderzeichen oder Umlaute enthalten.

❗ Es wird dennoch davon abgeraten, Namen für Datenfelder zu verwenden, die keine gültigen JavaScript-Bezeichner sind.

```
x4.ui.utils.accessModelProperty(model: <Model>, key: string [, params: object]):  
<PlainData>
```

Parameter	Erläuterung
model	Die Data-Model-Instanz, auf die das Binding angewandt wird.
key	Name/Schlüssel der Property
params	Optionales Objekt, das die Kontextparameter des Bindings enthält. Wenn die Eigenschaft <code>_value</code> vorhanden ist (auch <code>undefined</code> ), dann wird der Aufruf als Schreibzugriff ausgeführt und der darin enthaltene Wert in das Model geschrieben. Wenn die Eigenschaft <code>_value</code> oder das ganze Objekt nicht vorhanden ist, dann wird der Aufruf als Lesezugriff ausgeführt.

#### Rückgabewert:

- *Lesezugriff*: Wert der Model-Eigenschaft
- *Schreibzugriff*: `undefined`

#### Beispiele

##### Beispiel 1: Einfacher Name

```
// Access key "simpleName" in model -> same as {simpleName} or {#model.simpleName}  
x4.ui.utils.accessModelProperty(this.getModel(), 'simpleName', _parameters)
```

##### Beispiel 2: Komplexer Name

```
// Access key "@complexName!" in model -> not possible with Bindng Syntax  
x4.ui.utils.accessModelProperty(this.getModel(), '@complexName!', _parameters)
```

##### Beispiel 3: Deutscher Name

```
// Access key "straße" in model -> not possible with Binding Syntax  
x4.ui.utils.accessModelProperty(this.getModel(), 'straße', _parameters)
```

**Auf dieser Seite:****Siehe auch:**

- [Einführung zum UI-Framework](#)
- [Frames\(ui/frames.js\)](#)
- [Widgets\(ui/widgets.js\)](#)
- [Dependencies\(ui/dependencies.js\)](#)

### 2.14.5.4.3 Frames(ui/frames.js)

Die Javascript-Datei `ui/frames.js` ist Bestandteil der JavaScript-API für den *X4 Web Designer* und definiert ein Template sowie einen Renderer für die Einbindung eines Frames in eine View.

#### 2.14.5.4.3.1 Template

Das Standard-Template ("Frame") für Frames definiert ein `div`-Element mit der CSS-Klasse `x4-frame-wrapper`. In dieses Element werden durch einen *Renderer* die Elemente des Frames eingefügt.

#### 2.14.5.4.3.2 Renderer

Der Standard-Renderer für Frames (`x4.ui.render.Frame`) verwendet intern die Basis-API des UI-Frameworks (`x4.ui.createFrame`) um einen Frame zu erzeugen und in die View einzubinden, die diesen Renderer verwendet. Als Einstiegspunkt für die Inhalte des Frames wird der UI-Kontext `_contents` verwendet. Wenn dieser nicht im Template der View existiert (beispielsweise beim Standard-Template), so wird der UI-Kontext automatisch ins erste Element des Templates eingefügt.

#### UI-Properties

Um den Renderer zu konfigurieren, können folgende UI-Properties für die View definiert werden.

Name	Beschreibung	Typ
<code>url</code>	URL, von der die UI-Konfiguration/UI-Properties für den Frame als JSON-Daten geladen werden kann	Property-Binding
<code>initPropModelType</code>	<p>Typ des Models im Kontext der View, dessen Daten als Eigenschaften (Properties) für den Frame verwendet werden sollen</p> <p>Sie können auch nur ein Teil des Models verwenden, siehe <code>initPropModelContext</code>.</p> <p><i>Mögliche Werte:</i></p> <ul style="list-style-type: none"><li>• <code>"prop"</code></li><li>• <code>"model"</code></li><li>• <code>"data"</code></li><li>• <code>"state"</code></li><li>• <code>"localState"</code></li><li>• <code>"none"</code> (Standard)</li></ul>	Model-Typ

Name	Beschreibung	Typ
initPropModelContext	<p>Kontext, der für die Properties des Frames verwendet werden soll</p> <p>Dieser wurde innerhalb des Models durch initPropModelType definiert.</p> <p><i>Standardwert:</i> "#self"</p>	Model-Referenz
modelType	<p>Model, welches für den Frame verwendet werden soll</p> <p>Hier sind die gleichen Werte wie für initPropModelType erlaubt.</p> <p><i>Standardwert:</i> "none": ein neues, leeres Model für den Frame erzeugen</p>	Model-Typ

**i** Wenn Sie die Eigenschaften url und initPropModelType kombinieren, so werden die Eigenschaften des konfigurierten Models und die externen Eigenschaften kombiniert. Bei Konflikten überschreiben die externen Eigenschaften die internen.

### Beispiel

```
{
  "_template": "Frame",
  "_renderer": "x4.ui.render.Frame",
  "url": "/my/external.ui.json",
  "initPropModelType": "prop",
  "initPropModelContext": "frame",
  "modelType": "data",
  "frame": {
    /* Properties, die mit den Properties von "/my/external.ui.json" kombiniert
    werden */
  }
}
```

### Methoden

#### getFrame

Diese Methode liefert den Frame, der in die View eingebunden ist.

```
<frameRenderer>.getFrame(): <Frame>
```

**Rückgabewert:** Frame-Objekt oder null, wenn der Frame noch nicht initialisiert wurde

#### render

Diese Methode wertet die Konfigurations-Properties aus und erzeugt und initialisiert dementsprechend nach dem view-ready-Event ein Frame. Als Basis-Element für den Frame wird das erste Element aus dem übergebenen Array verwendet.

```
<frameRenderer>.render(elements: Array): Array
```

**Rückgabewert:** Das Array von DOM-Elementen, das der Methode übergeben wurde.

UI-Events

Der Renderer löst bei der Frame-Erzeugung folgende UI-Events aus:

Event-Name	Zeitpunkt des Auslösens
frame-attach	Direkt bevor die Wurzel-View des Frames in den DOM eingefügt wird
frame-ready	Wenn die Frame-Initialisierung abgeschlossen ist

**Auf dieser Seite:**

**Siehe auch:**

- [Einführung zum UI-Framework](#)
- [UI-Framework Core \(ui/ui.js\)](#)
- [Widgets \(ui/widgets.js\)](#)
- [Dependencies \(ui/dependencies.js\)](#)

#### 2.14.5.4.4 Widgets (ui/widgets.js)

Die Javascript-Datei `ui/widgets.js` ist Bestandteil der JavaScript-API für den *X4 Web Designer* und enthält die API für die Registrierung und Verwendung von Widget-Collections.

##### 2.14.5.4.4.1 Grundlegende Funktionen der Widget-API

`createManager`

Diese Methode erzeugt einen neuen Widget-Collection-Manager.

`x4.ui.widgets.createManager(): <WidgetColManager>`

**Rückgabewert:** Instanz des Widget-Collection-Managers

`getMainManager`

Diese Methode liefert den Widget-Collection-Manager, der für die Haupt-Widget-Collections der Seite verwendet wird.

`x4.ui.widgets.getMainManager(): <WidgetColManager>`

**Rückgabewert:** Instanz des Widget-Collection-Managers

##### 2.14.5.4.4.2 Widget-Collection-Manager API

`initByDOM`

Diese Methode lädt die im DOM definierten Widget-Collections.

`<manager>.initByDOM([ready: Function])`

Parameter	Erläuterung
<code>ready</code>	Funktion die aufgerufen wird, wenn alle Widget-Collections des Dokuments geladen wurden

**Rückgabewert:** Instanz des Widget-Collection-Managers

`create`

Diese Methode erzeugt ein neues Widget, registriert das Widget jedoch nicht automatisch im Manager, siehe add.

```
<manager>.create(name: string, title: string|null, baseSchema: Object, modes: Array): <Widget>
```

Parameter	Erläuterung																				
name	Name (technische Bezeichner) des Widgets																				
title	Anzeigename des Widgets  Wenn kein Titel übergeben wird, so wird der Name auch als Titel verwendet.																				
baseSchema	JSON-Schema, das die erwarteten UI-Properties des Widgets definiert																				
modes	Definiert die verfügbaren Modi für das Widget  Jeder Modus muss als Objekt innerhalb des Arrays definiert werden. Mögliche Attribute für ein Modus-Objekt sind: <table><tr><th>Attribut</th><th>Erläuterung</th><th>Typ</th><th>Erforderlich</th></tr><tr><td>name</td><td>Name (technischer Bezeichner) des Modus</td><td>String</td><td>Ja</td></tr><tr><td>title</td><td>Anzeigename) des Modus</td><td>String</td><td>Nein</td></tr><tr><td>template</td><td>Name des zu verwendenden Templates</td><td>String</td><td>Ja</td></tr><tr><td>renderer</td><td>Name der zu verwendenden Renderer-Klasse</td><td>String</td><td>Nein</td></tr></table>	Attribut	Erläuterung	Typ	Erforderlich	name	Name (technischer Bezeichner) des Modus	String	Ja	title	Anzeigename) des Modus	String	Nein	template	Name des zu verwendenden Templates	String	Ja	renderer	Name der zu verwendenden Renderer-Klasse	String	Nein
Attribut	Erläuterung	Typ	Erforderlich																		
name	Name (technischer Bezeichner) des Modus	String	Ja																		
title	Anzeigename) des Modus	String	Nein																		
template	Name des zu verwendenden Templates	String	Ja																		
renderer	Name der zu verwendenden Renderer-Klasse	String	Nein																		

**Rückgabewert:** Widget-Instanz

add

Diese Methode fügt ein Widget zu der Collection hinzu.

```
<manager>.add(widget: <Widget>)
```

Parameter	Erläuterung
widget	Widget-Objekt

get

Diese Methode liefert ein im Manager registriertes Widget.

```
<manager>.get(name: string): <Widget>
```

Parameter	Erläuterung
name	Name (technischer Bezeichner) des Widgets

**Rückgabewert:** Widget-Instanz

getNames

Diese Methode liefert die Namen aller im Manager registrierten Widgets.

```
<manager>.getNames(): Array
```

**Rückgabewert:** Namen als Array

useConfig

Diese Methode erzeugt und registriert ein Widget im Widget-Manager auf Basis eines Konfigurations-Objekts.

```
<manager>.useConfig(config: Object)
```

Parameter	Erläuterung
config	Objekt, welches das Widget konfiguriert  Dieses Objekt enthält die Attribute name, title, baseSchema und mode, die inhaltlich den Parametern der create-Methode entsprechen.

createCollection

Diese Methode erzeugt eine neue Widget-Collection, registriert diese jedoch nicht automatisch im Manager, siehe addCollection.

```
<manager>.createCollection(name: string, title: string|null, widgetNames: Array, dependencies: Object): <WidgetCol>
```

Parameter	Erläuterung
name	Name (technischer Bezeichner) der Widget-Collection
title	Anzeigename der Widget-Collection  Wenn kein Titel übergeben wird, so wird der Name auch als Titel verwendet.
widgetNames	Namen der Widgets, die zu der Collection gehören, als Array
dependencies	Objekt, das die Abhängigkeiten für die Widget-Collection definiert, siehe <a href="#">Dependencies (ui/dependencies.js)</a>

**Rückgabewert:** Instanz der Widget-Collection

addCollection

Diese Methode fügt ein Widget zu einer Collection hinzu.

```
<manager>.addCollection(collection: <WidgetCol>)
```

Parameter	Erläuterung
collection	Widget-Collection-Objekt

getCollection

Diese Methode liefert eine Widget-Collection, die im Manager registriert ist.

```
<manager>.getCollection(name: string): <WidgetCol>
```

Parameter	Erläuterung
name	Name (technischer Bezeichner) der Widget-Collection

**Rückgabewert:** Instanz der Widget-Collection

getCollectionNames

Diese Methode liefert die Namen aller Widget-Collections, die im Manager registriert sind.

```
<manager>.getCollectionNames(): Array
```

**Rückgabewert:** Namen als Array

useCollection

Diese Methode erzeugt und registriert eine Widget-Collection, inklusive Widgets, im Manager auf Basis eines Konfigurations-Objekts.

```
<manager>.useCollection(config: Object): <WidgetCol>
```

Parameter	Erläuterung
data	Objekt, das die Widget-Collection konfiguriert. Folgende Attribute beschreiben eine Collection:
</	

**Rückgabewert:** Instanz der Widget-Collection

#### 2.14.5.4.4.3 Widget-API

getName

Diese Methode liefert den Namen des Widgets.

```
<widget>.getName(): string
```

**Rückgabewert:** Name des Widgets

getTitle

Diese Methode liefert den Titel des Widgets.

```
<widget>.getTitle(): string
```

**Rückgabewert:** Titel des Widgets

getBaseSchema

Diese Methode liefert das JSON-Schema für die UI-Properties, welche das Widget erwartet.

```
<widget>.getBaseSchema(): Object
```

**Rückgabewert:** JSON-Schema-Objekt



getMode

Diese Methode liefert einen Modus des Widgets.

```
<widget>.getMode(name: string): <WidgetMode>
```

Parameter	Erläuterung
name	Name des Modus

**Rückgabewert:** Widget-Instanz

getModeNames

Diese Methode liefert die Namen aller Modi des Widgets.

```
<widget>.getModeNames(): Array
```

**Rückgabewert:** Namen als Array

#### 2.14.5.4.4 Widget-Mode-API

getName

Diese Methode liefert den Namen des Modus.

```
<widgetMode>.getName(): string
```

**Rückgabewert:** Name des Modus

getTitle

Diese Methode liefert den Titel des Modus.

```
<widgetMode>.getTitle(): string
```

**Rückgabewert:** Titel des Modus

getTemplateName

Diese Methode liefert den Namen des Templates des Modus.

```
<widgetMode>.getTemplateName(): string
```

**Rückgabewert:** Name des Templates

getRendererClassName

Diese Methode liefert den Namen der Renderer-Klasse zum Modus.

```
<widgetMode>.getRendererClassName(): string
```

**Rückgabewert:** Vollqualifizierter Name der JavaScript-Klasse, die als Renderer verwendet werden soll oder null, wenn kein Renderer für diesen Modus definiert ist

getPropertiesSchema

Diese Methode liefert das JSON-Schema für die UI-Properties, welche das Widget in diesem Modus erwartet.

```
<widgetMode>.getPropertiesSchema(): Object
```

**Rückgabewert:** JSON-Schema-Objekt

#### 2.14.5.4.5 Widget-Collection-API

getName

Diese Methode liefert den Namen der Widget-Collection.

```
<widgetCol>.getName(): string
```

**Rückgabewert:** Name der Widget-Collection

getTitle

Diese Methode liefert den Titel der Widget-Collection.

```
<widgetCol>.getTitle(): string
```

**Rückgabewert:** Titel der Widget-Collection

getWidgetNames

Diese Methode liefert die Namen aller Widgets zur Collection.

```
<widgetCol>.getWidgetNames(): Array
```

**Rückgabewert:** Namen als Array

getDependencies

Diese Methode liefert sämtliche Abhängigkeiten der Widget-Collection.

```
<widgetCol>.getDependencies(): Object
```

**Rückgabewert:** Objekt, welches die Dependencies konfiguriert, siehe [Dependencies \(ui/dependencies.js\)](#)

**Auf dieser Seite:**

**Siehe auch:**

- [Einführung zum UI-Framework](#)
- [UI-Framework Core \(ui/ui.js\)](#)
- [Frames \(ui/frames.js\)](#)
- [Dependencies \(ui/dependencies.js\)](#)

#### 2.14.5.4.5 Dependencies (ui/dependencies.js)

Die Javascript-Datei `ui/dependencies.js` ist Bestandteil der JavaScript-API für den *X4 Web Designer* und stellt Funktionalität für das Laden und Verwalten von Abhängigkeiten (Dependencies) bereit.

##### 2.14.5.4.5.1 Grundlegende Funktionen der Dependencies-API

createManager

Diese Methode erzeugt einen neuen Dependencies-Manager für die definierten Dependency-Typen.

```
x4.ui.dependencies.createManager([types: Object]): <DependenciesManager>
```

Parameter	Erläuterung				
<code>types</code>	<p>Objekt, welches die Abhängigkeits-Typen konfiguriert</p> <p>Jeder Abhängigkeits-Typ wird durch ein Attribut des Objekts konfiguriert, wobei der Attribut-Name den Typ und das enthaltene Objekt das Lade-Verhalten steuert. Wenn der Parameter-Wert fehlt, werden die Standard-Typen des UI-Frameworks verwendet.</p> <p><b>Attribute für den Dependency-Typ:</b></p> <table> <tr> <th>Attribut</th><th>Erläuterung</th></tr> <tr> <td><code>loadFile</code></td><td> <p>Funktion, die beim Laden einer Ressource dieses Typs folgende Parameter übergeben bekommt:</p> <ul style="list-style-type: none"> <li>• <code>url</code>: Die URL, von der geladen werden soll</li> <li>• <code>ready</code>: Eine Funktion, die aufgerufen werden muss, wenn das Laden und Bereitstellen der Ressource abgeschlossen ist</li> <li>• <code>manager</code>: Die Instanz des Dependencies-Managers</li> </ul> </td></tr> </table>	Attribut	Erläuterung	<code>loadFile</code>	<p>Funktion, die beim Laden einer Ressource dieses Typs folgende Parameter übergeben bekommt:</p> <ul style="list-style-type: none"> <li>• <code>url</code>: Die URL, von der geladen werden soll</li> <li>• <code>ready</code>: Eine Funktion, die aufgerufen werden muss, wenn das Laden und Bereitstellen der Ressource abgeschlossen ist</li> <li>• <code>manager</code>: Die Instanz des Dependencies-Managers</li> </ul>
Attribut	Erläuterung				
<code>loadFile</code>	<p>Funktion, die beim Laden einer Ressource dieses Typs folgende Parameter übergeben bekommt:</p> <ul style="list-style-type: none"> <li>• <code>url</code>: Die URL, von der geladen werden soll</li> <li>• <code>ready</code>: Eine Funktion, die aufgerufen werden muss, wenn das Laden und Bereitstellen der Ressource abgeschlossen ist</li> <li>• <code>manager</code>: Die Instanz des Dependencies-Managers</li> </ul>				

**Rückgabewert:** Die Instanz des Dependencies-Managers

`getMainManager`

Diese Methode liefert den Dependencies-Manager, welcher für das Laden der Haupt-Ressourcen des Dokuments verwendet wird.

`x4.ui.dependencies.getMainManager(): <DependenciesManager>`

**Rückgabewert:** Die Instanz des Dependencies-Managers

#### 2.14.5.4.5.2 API des Dependencies-Managers

`addType`

Diese Methode fügt einen Abhängigkeits-Typ hinzu.

`<manager>.addType(type: string, config: Object)`

Parameter	Erläuterung
<code>type</code>	Name des Abhängigkeits-Typs
<code>config</code>	Konfiguration des Abhängigkeits-Typs

`getType`

Diese Methode liefert die Konfiguration eines Abhängigkeits-Typs.

`<manager>.getType(type: string): Object`

Parameter	Erläuterung
<code>type</code>	Name des Abhängigkeits-Typs

**Rückgabewert:** Konfigurations-Objekt**load**

Diese Methode lädt Abhängigkeiten. Bei mehreren Aufrufen, die gleiche Dependencies enthalten (gleiche URL und gleicher Typ), werden diese nur einmal geladen und beim Ready-Handling entsprechend berücksichtigt.

```
<manager>.load(depsConf: Object, ready: Function)
```

Parameter	Erläuterung
<code>depsConf</code>	<p>Objekt, das die zu ladenden Abhängigkeiten konfiguriert</p> <p>Alle Abhängigkeiten eines Typs werden durch ein Array in einem Attribut des Objekts konfiguriert, wobei der Attribut-Name den Typ definiert. Jeder Eintrag in einem dieser Arrays muss ein Objekt sein, das folgende Attribute enthält:</p> <ul style="list-style-type: none"> <li>• <code>url</code>: Die URL von der geladen werden soll</li> <li>• <code>sync</code>: Gibt an, ob die Ressource synchron mit anderen Ressourcen (<code>true</code>) oder asynchron (<code>false</code>/Standard) geladen werden soll. Synchrone Ressourcen werden nacheinander in der Reihenfolge geladen, in der sie in der Objekt-Struktur definiert sind (Typ-übergreifend).</li> </ul>
<code>ready</code>	Funktion die aufgerufen wird, wenn das Laden aller Abhängigkeiten abgeschlossen ist

**isLoading**

Diese Methode prüft, ob zum Zeitpunkt des Aufrufs Abhängigkeiten geladen werden.

```
<manager>.isLoading(): boolean
```

**Rückgabewert:** `true`, wenn gerade Dependencies geladen werden, andernfalls `false`

**getAll**

Diese Methode liefert alle Abhängigkeiten, die von dem Manager geladen werden oder wurden.

```
<manager>.getAll(): Object
```

**Rückgabewert:** Objekt, das für jeden Abhängigkeits-Typ ein Attribut enthält, welches ein Array der Abhängigkeiten dieses Typs enthält. Jedes dieser Dependency-Objekte enthält folgende Attribute:

- `url`: Die URL, von der geladen wird/wurde
- `sync`: Gibt an, ob die Ressource synchron mit anderen Ressourcen (`true`) oder asynchron (`false`/undefined) geladen wird oder wurde
- `state`: Lade-Zustand der Ressource, siehe `getStateOf`

**getAllOfType**

Diese Methode verhält sich wie `getAll`, jedoch werden die Abhängigkeiten eines bestimmten Typs geliefert.

```
<manager>.getAllOfType(type: string): Array
```

Parameter	Erläuterung
<code>type</code>	Name des Dependency-Typs

**Rückgabewert:** Array, das die Abhängigkeiten des definierten Typs enthält

getStateOf

Diese Methode liefert den Lade-Zustand einer Abhängigkeit. Mögliche Zustände sind:

- INITIAL: Laden noch nicht gestartet
- LOADING: Laden ist aktiv
- READY: Laden ist abgeschlossen

Die Zustände sind im Objekt `x4.ui.dependencies.states` hinterlegt und werden als aufsteigende numerische Werte definiert. Somit sind Prüfungen wie `state < x4.ui.dependencies.states.READY` möglich.

```
<manager>.getStateOf(type: string, url: string): integer
```

Parameter	Erläuterung
<code>type</code>	Name des Abhängigkeits-Typs
<code>url</code>	URL der Abhängigkeits-Ressource

**Rückgabewert:** Zahl, die den aktuellen Lade-Zustand der Ressource repräsentiert

**Auf dieser Seite:**

**Siehe auch:**

- [Einführung zum UI-Framework](#)
- [UI-Framework Core \(ui/ui.js\)](#)
- [Frames \(ui/frames.js\)](#)
- [Widgets \(ui/widgets.js\)](#)

#### 2.14.5.5 XML-Parser/Serializer (xml/xml.js)

Die Javascript-Datei `xml/xml.js` ist Bestandteil der JavaScript-API für den *X4 Web Designer* und enthält in Klasse `x4.xml` eine API, um XML-Daten zu parsen und zu serialisieren.

##### 2.14.5.5.1 parse

Diese Methode liest (parst) einen XML-String als XML-DOM-Objekt.

```
x4.xml.parse(xml: string): <DOM>
```

Parameter	Beschreibung
<code>xml</code>	XML als Zeichenkette

**Rückgabewert:** XML-DOM, abhängig von der Implementierung des Browsers

##### 2.14.5.5.2 serialize

Diese Methode erzeugt (serialisiert) ein XML *Document Object Model* (XML-DOM) oder einen Knoten aus diesem.

```
x4.xml.serialize(dom: <DOM>): string
```

Parameter	Beschreibung
dom	XML-DOM oder ein Knoten eines XML-DOMs

**Rückgabewert:** XML-Dokument als Zeichenkette

**Auf dieser Seite:**

**Siehe auch:**

- [Objektorientierte JavaScript-API \(core.js\)](#)
- [Data Model API \(model/model.js\)](#)
- [Asynchrone Aktionen und ReST \(async/async.js\)](#)
- [UI-Framework für den X4 Web Designer](#)