

Geschäftsprozesse erfolgreich digitalisieren

Die digitale Transformation durch die Low-Code-Plattform X4 BPMS beschleunigen

X4 Web Apps

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

Vertrieb

Telefon: +49 7243 56175-0

vertrieb@softproject.de

SoftProject-Support

Telefon: +49 7243 56175-333

support@softproject.de

© SoftProject GmbH. Alle Rechte vorbehalten.

Inhaltsverzeichnis

1	Aufbau eines Web App-Projekts	13
1.1	Back End	13
1.2	Front End	14
2	Neues Web App-Projekt anlegen und deklarieren	15
2.1	Templates	15
2.1.1	ESB Project	15
2.1.2	BPM Project	15
2.1.3	Web App Project	16
2.2	Neues Web App-Projekt anlegen	16
2.3	Web App-Projekt anzeigen	17
2.3.1	Auf mobilen Endgeräten öffnen	18
2.3.2	Im Browser öffnen	18
2.3.3	URL der Web App kopieren	19
2.4	Web App-Definition deklarieren	19
2.4.1	Components in der Web App-Definition referenzieren	20
2.5	Menübreite konfigurieren	25
2.6	Impressum	26
2.6.1	Inhalt im Impressum hinterlegen	27
2.7	Datenschutzerklärung	28
2.7.1	Eine Seite für die Datenschutzerklärung anlegen	29
2.7.2	Inhalt in der Datenschutzerklärung hinterlegen	30
2.8	Responsives Verhalten	31
2.8.1	Seitenlayout am Desktop-PC	31
2.8.2	Mobiles Seitenlayout	32
3	Theming	33
3.1	Farbschema festlegen	33
3.2	Farben	33
3.2.1	Main Theme	33

3.2.2	Chart Theme	36
3.2.3	Farben festlegen	37
3.3	Schriftarten	37
3.3.1	Schriftarten festlegen	38
4	Configuration	40
4.1	Neue Configuration anlegen	40
4.2	Zugriff auf Web Apps verwalten.....	40
4.3	Configuration deklarieren.....	40
4.3.1	<General>	41
4.3.2	<Access>.....	43
4.3.3	<Status>.....	46
4.3.4	<Logout>.....	48
4.3.5	<Notifications>.....	48
4.3.6	<HTTPSecurityHeader>	50
4.3.7	<HeaderIncludes>	52
5	Lokalisierung	53
5.1	Eine neue Sprachdatei anlegen	53
5.2	Übersetzungsschlüssel	54
6	Anmeldeseite	58
6.1	Anmeldeseite anlegen	59
6.1.1	<Login>	59
6.2	Passwort vergessen-Schaltfläche erzeugen	60
6.2.1	Beispiel	60
7	Module	61
8	Datenmodell und Properties-Definition	65
8.1	Daten aus dem Technical Process	68
8.1.1	URL	68
8.2	Beispiel für eine einfache Properties-Definition	69
8.3	Beispiel für eine komplexe Properties-Definition.....	69
8.4	Validierung	71
8.4.1	Mögliche Validierungsarten	71

8.4.2	Mögliche Operatoren	72
8.4.3	Beispiel logische Verknüpfungen.....	73
8.4.4	Beispiele <Message>	73
9	Components.....	76
9.1	Attribute für Komponenten	78
9.2	Detail Component.....	81
9.2.1	Detail Component definieren	81
9.2.2	Beispiel Detail Component	82
9.3	List Component.....	83
9.3.1	List Component definieren	83
9.3.2	Column-Definition	86
9.3.3	Periodisches Neuladen.....	90
9.3.4	Mobile Ansicht der List Component	91
9.3.5	Aktion "Auswählen" innerhalb der List Component	91
9.3.6	Beispiel List Component	92
9.3.7	Paging.....	93
9.4	Calendar Component	97
9.4.1	Calendar Component definieren	97
9.4.2	Calendar Settings: Ansichten	98
9.4.3	Calendar Settings: Lokalisierung.....	99
9.4.4	Events.....	101
9.4.5	Externe Links in Events	104
9.4.6	Daten für die Calendar Component	105
9.4.7	Beispiel	105
9.5	Master/Detail Component	109
9.5.1	Properties in Master/Detail.....	110
9.5.2	Master/Detail Component definieren.....	110
9.5.3	Component aus- und einblenden	111
9.5.4	Master-Strukturelement.....	112
9.5.5	Detail-Strukturelement	113
9.5.6	Overlay-Strukturelement.....	113

9.6	ExternalWeb Component.....	116
9.6.1	Beispiel	118
9.7	Grid-Komponente.....	119
9.7.1	Grid Component definieren	119
9.7.2	Beispiel einer Grid-Komponente.....	128
9.7.3	Grid Settings	128
9.7.4	GridColumn-Definition.....	131
9.7.5	Periodisches Neuladen.....	132
9.7.6	"Select" Action innerhalb der Grid Component.....	133
9.7.7	"Save" Action innerhalb der Grid Component	133
9.7.8	Paging.....	133
10	Layouts	136
10.1	Box-Layout	138
10.1.1	Beispiele	138
10.1.2	<BoxLayout>	139
10.2	Flow-Layout	144
10.2.1	Beispiele	145
10.2.2	<FlowLayout>.....	145
10.3	Grid-Layout.....	149
10.3.1	Beispiele	150
10.3.2	<GridLayout>.....	150
10.3.3	<ColumnDefinitions>	154
10.3.4	<Column>	155
10.4	Registerkarten-Layout: Tab Group	156
10.4.1	Beispiele	157
10.4.2	<TabGroup>.....	158
10.5	Registerkarten-Layout: Tab Layout.....	161
10.5.1	Beispiele	162
10.5.2	<TabLayout>	162
10.5.3	<Tab>.....	166
10.6	Responsive-Layout	170

10.6.1	Beispiele	171
10.6.2	<ResponsiveLayout>	172
10.6.3	<Cell>	177
11	Controls	181
11.1	Button.....	181
11.1.1	Beispiel	182
11.2	Charts	182
11.2.1	Liniendiagramm	183
11.2.2	Balkendiagramm	192
11.2.3	Kreis-/Donutdiagramm	202
11.2.4	Tachodiagramm	210
11.3	Checkbox	219
11.3.1	Aktion "Auswählen" innerhalb einer Checkbox.....	224
11.3.2	Beispiel für <Checkbox>	224
11.3.3	Beispiel für <Checkbox> mit einem Technical Process.....	225
11.4	ComboBox.....	225
11.4.1	Beispiele	232
11.5	ComboBoxAutocomplete	237
11.5.1	Beispiel	243
11.6	DateTimePicker	245
11.7	File Link	249
11.7.1	Beispiel <FileLink> (Dateidownload aus statischer Ressource)	254
11.8	File Upload	255
11.8.1	Input-Format für den Technical Process.....	261
11.8.2	Beispiel <FileUpload>	261
11.9	Header	262
11.9.1	Beispiel <Header>	266
11.10	HTML Document.....	267
11.10.1	Beispiel für <HtmlDocument>	269
11.11	Image.....	272
11.11.1	Aktion "Auswählen" innerhalb eines Image Controls	276

11.11.2 Beispiel <Image> aus statischer Ressource	276
11.11.3 Beispiel <Image> mit Data Binding (dynamisch aus einem Technical Process)	276
11.12 Card	278
11.12.1 Card-Control erzeugen	278
11.12.2 Aktionen.....	282
11.12.3 Beispiele	282
11.13 Label	284
11.13.1 Beispiel <Label>	288
11.14 Link.....	289
11.14.1 Beispiel	290
11.15 ListView	290
11.15.1 Select Action innerhalb einer ListView	295
11.15.2 Beispiele	295
11.16 Maps	298
11.16.1 Maps Controls erzeugen	298
11.16.2 Suchfunktion.....	302
11.16.3 Routenberechnung.....	303
11.16.4 Einfacher Marker	304
11.16.5 Markergruppe	307
11.16.6 Aktion "Auswählen" innerhalb einer Map.....	310
11.16.7 Beispiele	310
11.16.8 Marker per Klick in der Karte setzen.....	315
11.16.9 Standort des Benutzers verwenden	316
11.17 PasswordBox	317
11.18 Radio Button	321
11.18.1 Aktion "Auswählen" innerhalb eines Radio-Buttons	325
11.18.2 Beispiel <RadioButton>	326
11.19 TextBlock.....	326
11.19.1 Paragraph	326
11.19.2 Span.....	330
11.19.3 Break	334

11.19.4 Binding.....	334
11.19.5 Translation.....	335
11.19.6 Link.....	335
11.20 TextBox.....	336
11.20.1 Beispiel <TextBox>	342
11.21 Video.....	343
11.21.1 Data Binding für value.....	346
12 Actions.....	347
12.1 Action "New".....	352
12.2 Action "Save"	353
12.3 Action "Delete"	353
12.4 Action "Upload"	354
12.5 Action "Download"	356
12.6 Action "Custom".....	357
12.7 Action "Reload"	359
12.8 Action "Cancel"	360
12.9 Action "Select"	361
12.9.1 Beispiel	362
12.10 Action "ForgotPassword"	362
12.10.1 Beispiel	363
12.11 Action "LogoutAction"	363
12.11.1 Beispiel	363
13 HeaderSection	364
13.1 HeaderSection anlegen	364
13.1.1 <HeaderSection>.....	364
14 JavaScript in Web Apps einbinden	365
14.1 Mit Web App Control Projects arbeiten	365
14.1.1 Web App Control Project anlegen	365
14.1.2 Benutzerdefiniertes Control in eine Web App einbinden	366
14.1.3 Verfügbare API-Funktionen	369
15 Data Binding.....	371

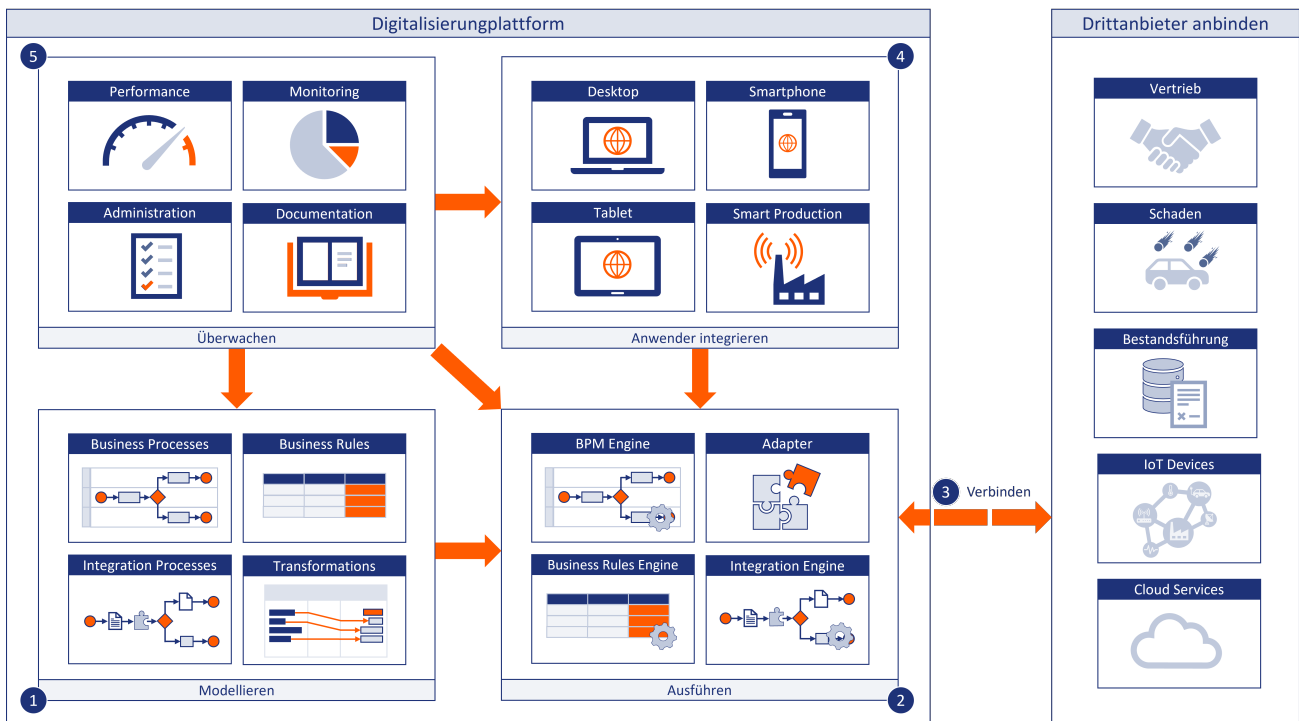
16	Navigation innerhalb einer Web-Anwendung	373
16.1	Navigation mit einer Action	373
16.2	Navigation durch Überschreiben der select Action	373
17	Sortieren	375
18	Meldungen.....	376
19	Profiling	378
19.1	Profiling-Datei erstellen	378
19.2	Profiling konfigurieren.....	378
19.3	Beispiel einer Profiling-Konfigurationsdatei	379

Über die X4 BPMS

Digitalisierung braucht eine ganzheitliche Betrachtung, die sich in der einzusetzenden Lösung widerspiegeln muss. Als zentrale Plattform unterstützt Sie die X4 BPMS dabei, diese Herausforderungen zu lösen. Im Fokus stehen dabei die Modellierung, Implementierung und Überwachung Ihrer Geschäftsprozesse. Daher enthält die X4 BPMS alle benötigten Werkzeuge und ist mit einer Vielzahl an Schnittstellen und Formaten kompatibel. Mit der X4 BPMS 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 BPMS 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 BPMS 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 Web Apps:** Web Apps für Mitarbeiter und Kunden bereitstellen



Mit der X4 Suite-Komponente X4 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.

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 Web Apps:

- Zentrale App Definition komplett im X4 Designer
- Einfache Verknüpfung mit Technical Processes 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 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.

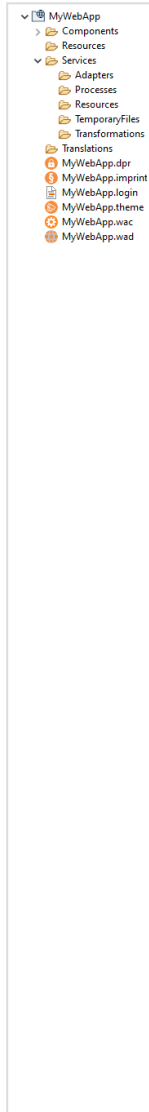
An wen richtet sich dieses Dokument?

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 Aufbau eines Web App-Projekts

1.1 Back End

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

 The screenshot shows a hierarchical tree view for a project named 'MyWebApp'. The tree structure is as follows: - MyWebApp - Components - Resources - Services - Adapters - Processes - Resources - TemporaryFiles - Transformations - Translations - MyWebApp.dpr - MyWebApp.imprint - MyWebApp.login - MyWebApp.theme - MyWebApp.wac - MyWebApp.wad	• <i>Projekt</i> Web App-Projekt • Components: Separate Definitionen für die einzelnen Components. Der Ordner darf lediglich Definitionen für Detail Components, List Components, Grid Components, Calendar Component und Master/Detail Components 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: Technical Processes. 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 • <i>Projekt.dpr</i>: Angaben zum Datenschutz, entweder als Text oder Verlinkung zu einer externen Seite • <i>Projekt.headersection</i>: Definition eines eigenen Headers, der anstelle des Standard-Headers angezeigt wird. • <i>Projekt.imprint</i>: Impressum, entweder als Text oder Verlinkung zu einer externen Seite • <i>Projekt.theme</i>: Theme der Anwendung, kann anhand eines Theme-Designers definiert werden • <i>Projekt.wad</i>: Datei mit der Web App-Definition • <i>Projekt.wac</i>: Datei mit der Web App-Konfiguration
--	--

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.

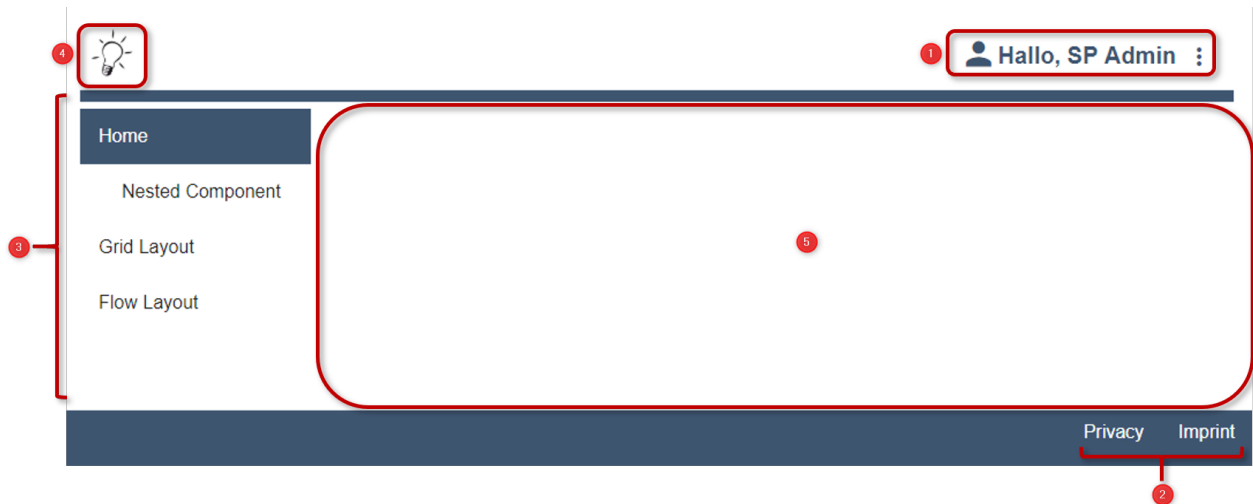


Sicherheitshinweis

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

1.2 Front End

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 Components gebildet.
4	Logo der Web App
5	Bereich für den Inhalt der Component. Die Darstellung hängt von der Art der Component, dem verwendeten Layout und den verwendeten Controls ab.

2 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 BPMS. 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.

2.1 Templates

Im X4 Designer stehen Ihnen verschiedene Templates beim Erstellen eines Projects zur Auswahl.

2.1.1 ESB Project

Kategorie	Template	Beschreibung
Basics	ESB Introduction	Dieses Projekt enthält eine kurze Einführung in die Modellierung Technical Processes.
	XSLT Introduction	Dieses Projekt enthält eine kurze Einführung in die Erstellung von XML-Datentransformationen.
Database	Create Data	Dieses Projekt enthält ein Beispiel für die Erstellung von Daten in einer Datenbank unter Verwendung des JDBC Connector Adapters.
	Delete Data	Dieses Projekt enthält ein Beispiel für das Löschen von Daten aus einer Datenbank unter Verwendung des JDBC Connector Adapters.
	Read Data	Dieses Projekt enthält ein Beispiel für das Lesen von Daten aus einer Datenbank unter Verwendung des JDBC Connector Adapters.
	Update Data	Dieses Projekt enthält ein Beispiel für die Aktualisierung von Daten in einer Datenbank unter Verwendung des JDBC Connector Adapters.
E-Mail	Send e-mails with the POP/SMTP adapter	Dieses Projekt enthält ein Beispiel für das Versenden von E-Mails unter Verwendung des POP/SMTP-Adapters.

2.1.2 BPM Project

Kategorie	Template	Beschreibung
Basics	BPMN Introduction	Dieses Projekt enthält eine kurze Einführung in die Modellierung von Business Processes.

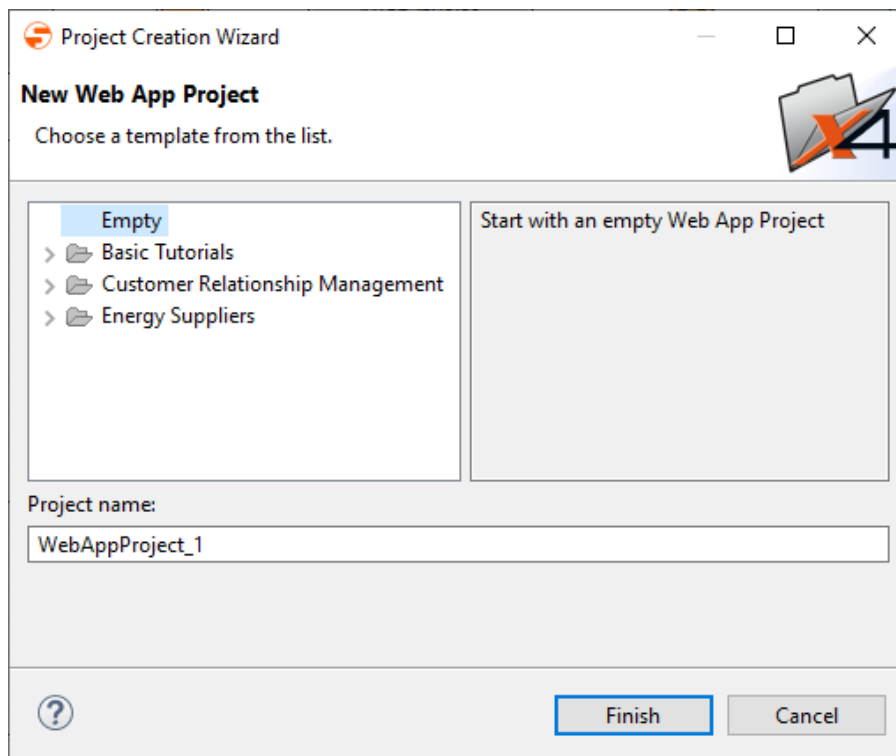
Demos	Approve electronic invoice	Diese Demo enthält einen Genehmigungsprozess für elektronische Rechnungen. Es umfasst Business Processes, Geschäftsregeln (DMN) und Human Tasks.
--------------	----------------------------	--

2.1.3 Web App Project

Kategorie	Template	Beschreibung
Components	Calendar Component	Diese Web App ist ein Beispiel für eine Calendar Component mit Terminen.
	Grid Component	Diese Web App ist ein Beispiel für die Anzeige von Daten in tabellarischer Form in der Grid Component.
Controls	How to use Controls	Dieses Projekt enthält Beispiele für die Verwendung der gängigsten Controls in X4 Web Apps.
	TextBlock	Dieses Projekt enthält Beispiele für die Verwendung des TextBlock Controls in X4 Web Apps.
Demos	Manage your customer data	In dieser Demo können Sie Ihre Kunden, deren Adressdaten sowie Ansprechpartner verwalten.
	Report a street lighting malfunction	In dieser Demo kann ein Kunde eine Störung im Straßenbeleuchtungssystem melden. Die Daten werden in einer Datenbank gespeichert.
Layouts	How to use layouts	Dieses Projekt enthält Beispiele für die Verwendung von Layouts in X4 Web Apps.
Properties	Data Context	Dieses Projekt enthält ein Beispiel für die Verwendung von Properties desselben Kontexts in mehreren Components.

2.2 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.

2.3 Web App-Projekt anzeigen

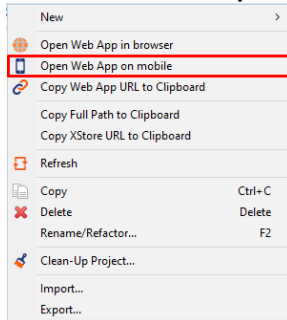
Das Frontend der Web-Anwendung lässt sich jederzeit, auch während der Bearbeitung, sowohl im Browser als auch auf einem mobilen Endgerät anzeigen.

- ❗ Bestimmte Änderungen innerhalb des Web App-Projektes werden erst nach einem erneuten Log-in angezeigt. Dazu gehören:
- Farben
 - Sprache
 - Menü

✓ Weitere Informationen finden Sie im Abschnitt

2.3.1 Auf mobilen Endgeräten öffnen

1. Mit der rechten Maustaste auf das Web App-Projekt klicken.
2. Im Kontextmenü **Open Web App on mobile** wählen.



Ein QR-Code wird generiert und angezeigt.

3. QR-Code mit dem mobilen Endgerät scannen.
4. Mit den Zugangsdaten einloggen.

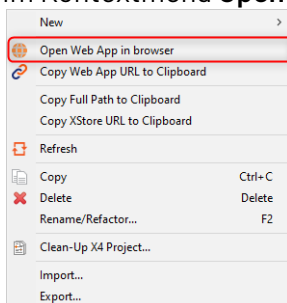
⚠ Wenn der Zugriff auf die Web App nicht eingeschränkt wird, kann jeder Benutzer auf die Web App zugreifen.

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

✓ Weitere Informationen finden Sie im Abschnitt

2.3.2 Im Browser öffnen

1. Mit der rechten Maustaste auf das Web App-Projekt klicken.
2. Im Kontextmenü **Open Web App in browser** wählen.



Die Web App wird im Standard-Browser geöffnet.

3. Mit den Zugangsdaten einloggen.

⚠ Wenn der Zugriff auf die Web App nicht eingeschränkt wird, kann jeder Benutzer auf die Web App zugreifen.

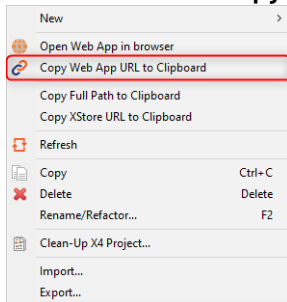
- ✓ Leerzeichen, die am Ende des Benutzernamens eingefügt werden, z. B. durch eine Autovervollständigung bei der Nutzung am Smartphone, werden automatisch entfernt.

- ✓ Weitere Informationen finden Sie im Abschnitt

2.3.3 URL der Web App kopieren

Wenn Sie die Web App in einem anderen als dem Standard-Browser öffnen oder z. B. die URL versenden möchten, dann können Sie die URL der Web App auch direkt in die Zwischenablage kopieren.

1. Mit der rechten Maustaste auf das Web App-Projekt klicken.
2. Im Kontextmenü **Copy Web App URL to Clipboard** wählen.



- ✓ Weitere Informationen finden Sie im Abschnitt

2.4 Web App-Definition deklarieren

Web-Anwendungen sind hierarchisch aufgebaut:

- Web-Anwendung
- Modul(e)
- Components
- Layout(s)
- Controls

Die Web-Anwendung und die in der Anwendung verfügbaren Module mit ihren jeweiligen Zugriffsrechten werden in der Web App-Definition `.wad` deklariert. Die einzelnen Components mit ihrem jeweiligen Layout und den verfügbaren Controls werden innerhalb der jeweiligen Component beschrieben und in der Web App-Definition referenziert.

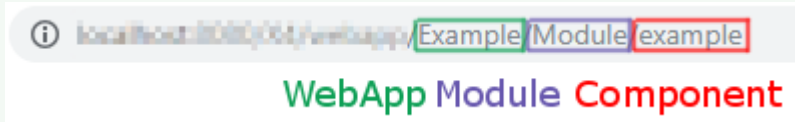
Bei der Deklaration von `.wad`-Dateien wird die Datei automatisch validiert. So können mögliche Fehlerquellen bereits beim Erstellen von Web-Anwendungen reduziert werden.

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
path	<i>Erforderlich.</i> Pfad zur Web App. Wird in der Adresszeile des Browsers angezeigt. Muss für jedes Web App-Projekt eindeutig sein. Mögliche Werte: Zeichenkette aus alphanumerischen Zeichen (keine Umlaute, Punkte usw.) i 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 nicht angepasst und die Web App ist weiterhin über die bisherige URL erreichbar.

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



2.4.1 Components in der Web App-Definition referenzieren

Jede einzelne in der Web App verwendete **Component** 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	Bei genau einer Component erforderlich. Legt fest, welche Component zuerst angezeigt wird ("Startseite"). Mögliche Werte: <i>true / false</i>

Attribut	Beschreibung
displayName	Anzeigenamen der Component im Menü. Mögliche Werte: Beliebige Zeichenkette mit dem Anzeigenamen der Component
fontFamily	Legt die Schriftfamilie innerhalb der Component fest. Die Eigenschaft wird an alle Controls und Actions der Component vererbt. Mögliche Werte: Fontcode aus der Font-Palette, z. B. <i>Font04</i>
fontSize	Legt die Schriftgröße innerhalb der Component fest. Die Eigenschaft wird an alle Controls und Actions der Component vererbt. Mögliche Werte: • Schriftgröße in Pixel, z. B. <i>20px</i> • Schriftgröße in Punkt, z. B. <i>18pt</i> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i> • Schlüsselwörter, z. B. <i>small</i> oder <i>larger</i>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. Die Eigenschaft wird an alle Controls und Actions der Component vererbt. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded
fontStyle	Legt die Neigung der Schrift fest. Die Eigenschaft wird an alle Controls und Actions der Component vererbt. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest. Die Eigenschaft wird an alle Controls und Actions der Component vererbt. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight
iconURL	Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Icon der Component verwendet wird.  • Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources. • Das Material Icon muss mit dem Präfix <code>icon:</code>, z. B. <code>icon:<MaterialIconName></code> angegeben werden. Mögliche Werte: • Zeichenkette (URI), z. B. <code>clock.png</code> • <code>icon:<MaterialIconName></code>, z. B. <code>icon:extension</code>  Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren Material Icons abweichen.

Attribut	Beschreibung
iconColor	Definiert die Farbe des Icons der Component.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: - Hexadezimaler Farbwert, z. B. <i>ff5a00</i>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>
name	Eindeutiger Name der Component. Mögliche Werte: Beliebige Zeichenkette mit dem Namen der Component
path	<i>Erforderlich.</i> Eindeutige URL der Component, sichtbar in der Adresszeile des Browsers. Mögliche Werte: Zeichenkette aus alphanumerischen Zeichen (keine Umlaute, Punkte usw.)
showInMenu	Legt fest, ob die Component im Menü angezeigt wird.  Dieses Attribut hat keine Auswirkungen auf eine Master/Detail Component. Mögliche Werte: <code>true</code> (default) / <code>false</code>  Diese Einstellung bezieht sich auch auf alle Subcomponents. Wenn keine Component im Menü angezeigt wird, dann wird das gesamte Menü ausgeblendet. Der Inhaltsbereich erstreckt sich dann über die gesamte Breite.
source	Pfad zur Definitionsdatei relativ zum Ordner Components. Mögliche Werte: Zeichenkette (URI)

Attribut	Beschreibung
title	Titel der Component. Wird in der Component als Header angezeigt. • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette mit dem Titel der Component
titleBackground	Definiert eine Titel-Hintergrundfarbe. Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
titleForeground	Definiert eine Farbe für den Titel-Vordergrund. Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Beispiel

Definition des integrierten Beispielprojektes Simple CRM

```
<?xml version="1.0" encoding="UTF-8"?>
<WebApp xmlns="http://softproject.de/webapp/1.0"
  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>
```

2.5 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

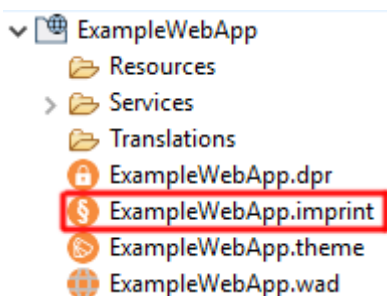
Attribut	Beschreibung
unit	Einheit für den in Attribut width angegebenen Wert. Mögliche Werte: • pixels: Angabe in Pixel • percents: Angabe in Prozent  Wird keine Einheit definiert, wird der in width angegebene Wert als Pixel interpretiert.
width	Breite des Menüs Mögliche Werte: • beliebige ganze Zahl • auto: Automatische Breite, die sich anhand der definierten Menüeinträgen automatisch berechnet

Beispiel-Konfiguration für die Menübreite

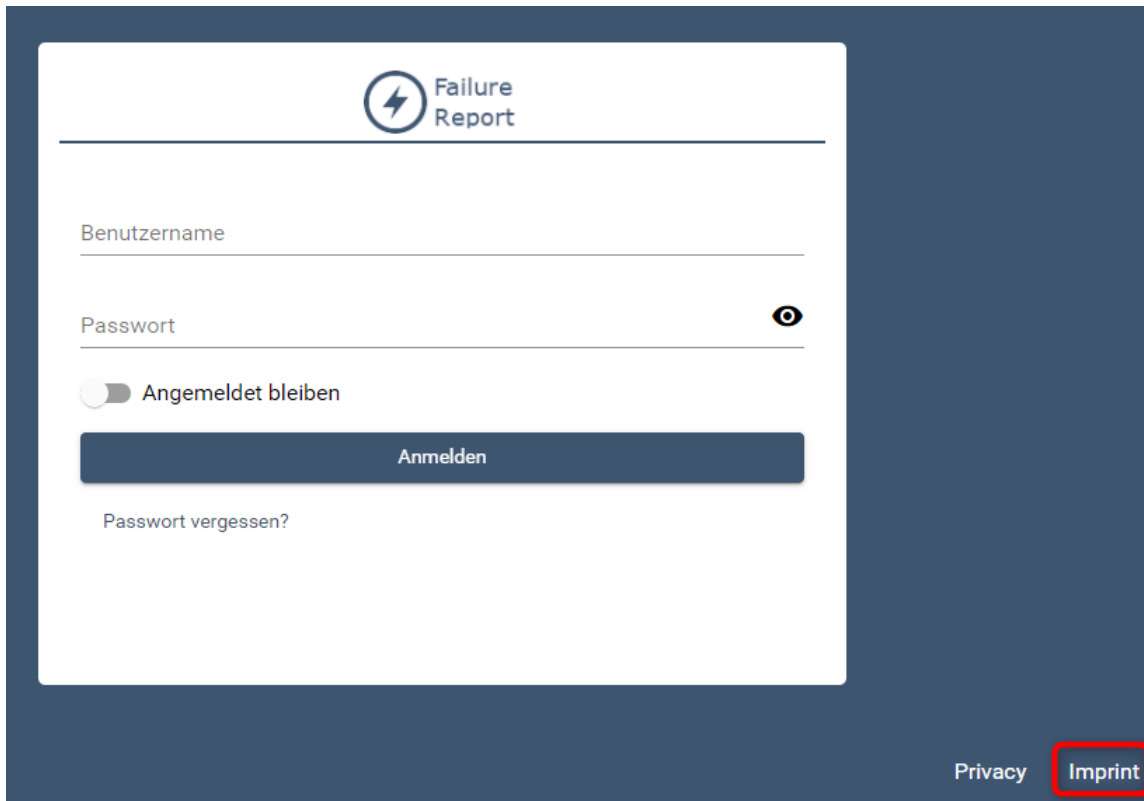
```
<WebApp xmlns="http://softproject.de/webapp/1.0">
  ...
  <Menu width="30" unit="percents"/>
  <Modules>
    ...
  </Modules>
</WebApp>
```

2.6 Impressum

Das Impressum einer Web-Anwendung 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.



⚠ Es ist auch möglich, externe Seiten als Impressum einzubinden. Legen Sie die Verlinkung in der [Web App-Konfiguration](#) `.wac` fest.

Eine Seite für das Impressum anlegen

⚠ Der Wert des Attributs `title` wird ebenfalls im Footer-Bereich der Web App angezeigt.

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.

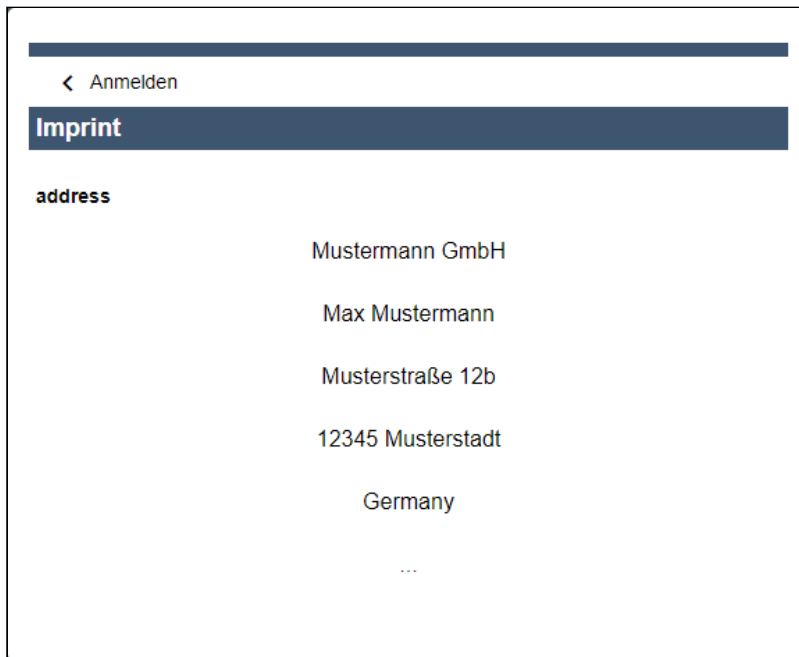
2.6.1 Inhalt im Impressum hinterlegen

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

Impressum (.imprint)

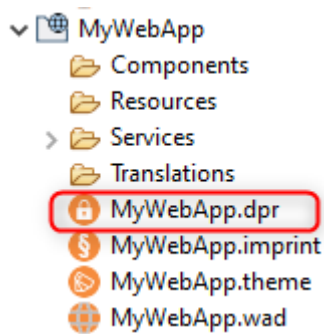
```
<?xml version='1.0' encoding='UTF-8'?>
<Imprint xmlns="http://softproject.de/webapp/1.0" title="Imprint">
  <FlowLayout>
    <Header value="Imprint" />
    <Header titleLevel="subtitle" value="address" />
    <Label value="Mustermann GmbH" />
    <Label value="Max Mustermann" />
    <Label value="Musterstraße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

Obiger Code erzeugt folgendes Impressum:

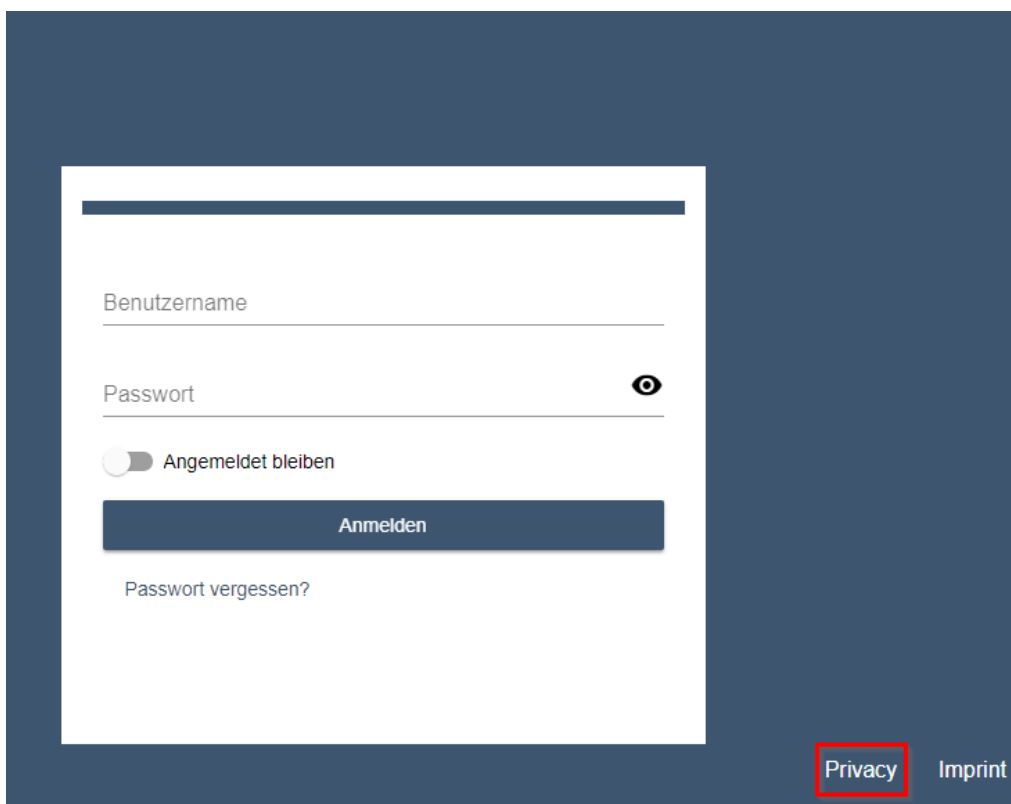


2.7 Datenschutzerklärung

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 Dateiergung `.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.



⚠ Es ist auch möglich, externe Seiten als Datenschutzerklärung einzubinden. Legen Sie die Verlinkung in der [Web App-Konfiguration](#) .wac fest.

2.7.1 Eine Seite für die Datenschutzerklärung anlegen

⚠ Der Wert des Attributs `title` wird ebenfalls im Footer-Bereich der Web App angezeigt.

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.

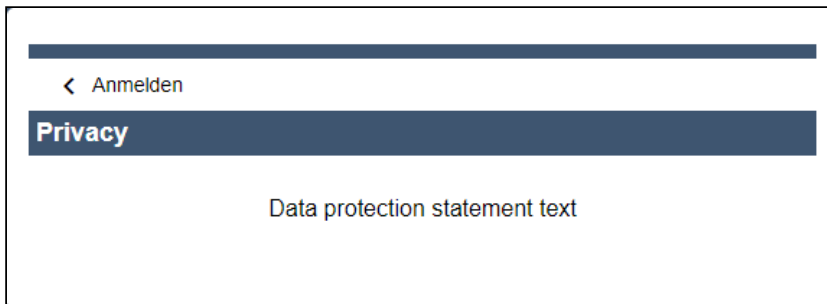
2.7.2 Inhalt in der Datenschutzerklärung hinterlegen

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

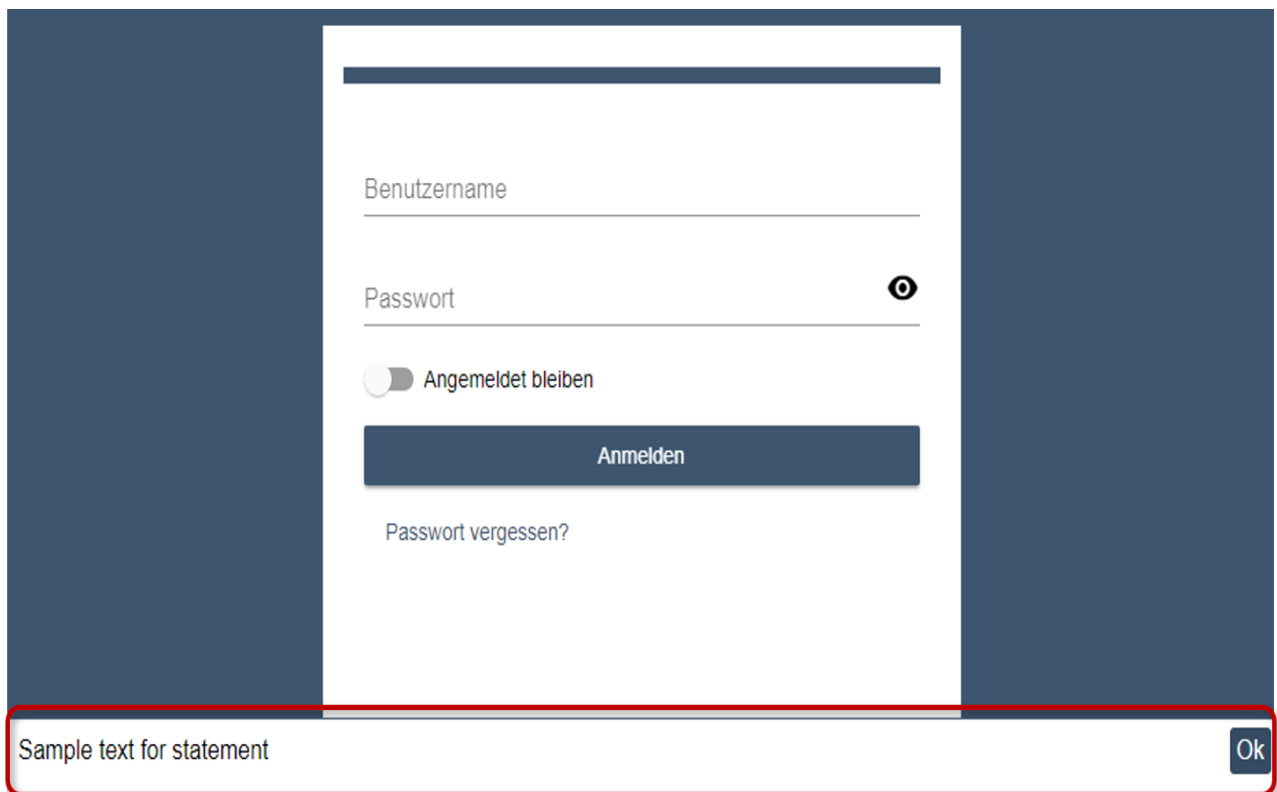
Datenschutzerklärung (.dpr)

```
<?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:



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 background. The form is centered and contains the following elements:

- A horizontal line at the top.
- A text input field labeled "Benutzername".
- A text input field labeled "Passwort" with an eye icon to its right.
- A toggle switch labeled "Angemeldet bleiben".
- A dark blue button labeled "Anmelden".
- A link labeled "Passwort vergessen?".

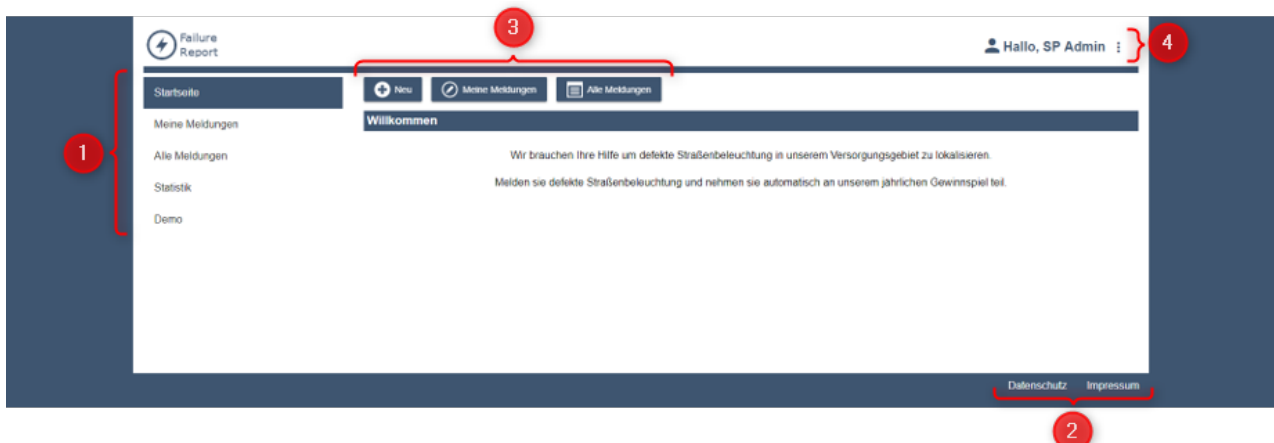
Below the form, there is a white box with a red border containing the text "Sample text for statement" and an "Ok" button.

2.8 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:

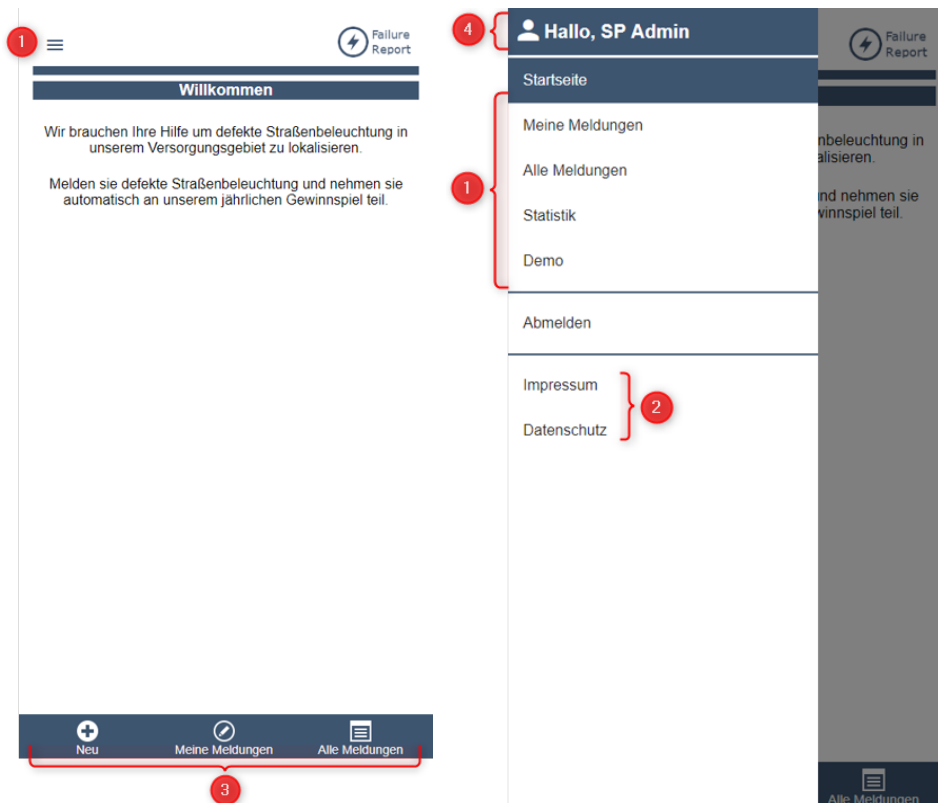
- das Seitenlayout
- das Aussehen von [Checkboxes](#)
- die Anzeige von Informationen innerhalb einer [List Component](#) / [Grid Component](#)
- die Anzeige von [Master/Detail Components](#)

2.8.1 Seitenlayout am Desktop-PC



1	Das Menü wird automatisch mithilfe der Components erzeugt. Jede Component erzeugt einen Menüpunkt.
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 Actions
4	Angaben zum angemeldeten Nutzer

2.8.2 Mobiles Seitenlayout

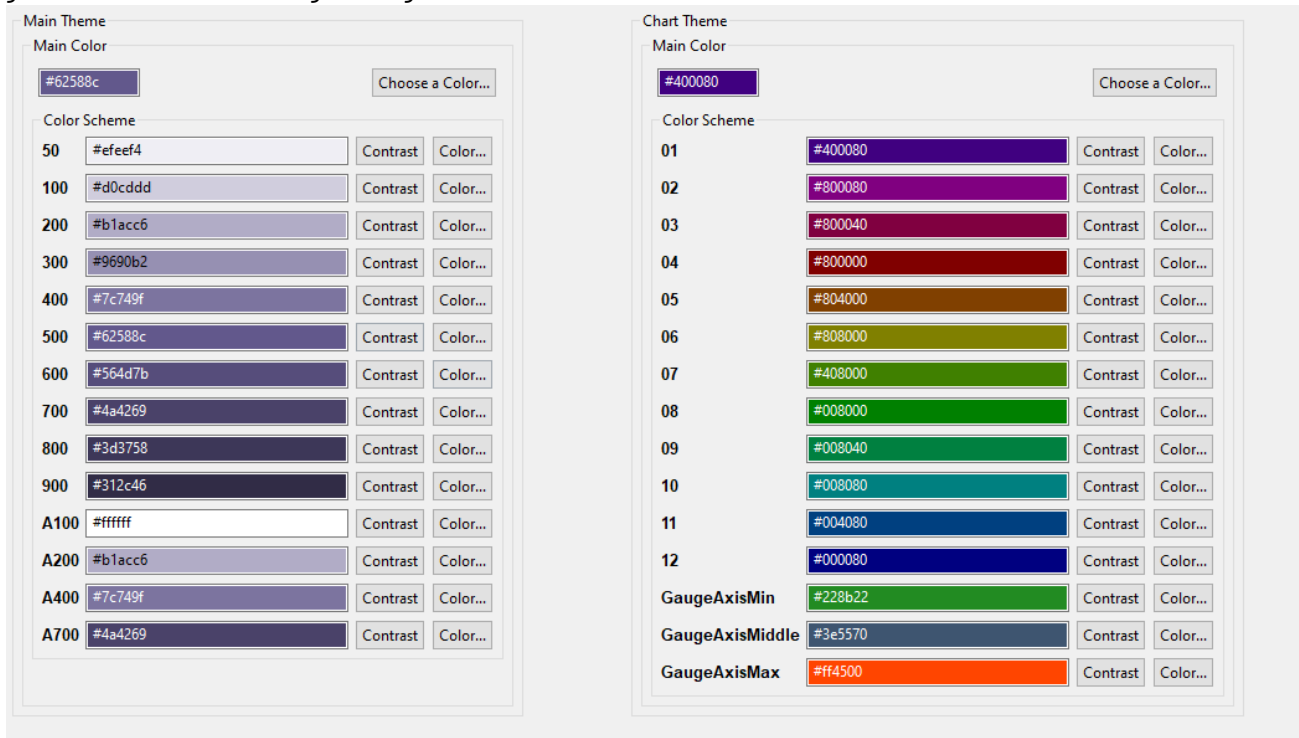


1	Menü, wird automatisch mithilfe der Component erzeugt. Jede Component erzeugt einen Menüpunkt. Das Menü wird im mobilen Modus erst durch Anklicken des Menü-Symbols sichtbar.
2	Links zu den Datenschutzerklärungen bzw. zum Impressum. Links werden automatisch generiert und verweisen auf die Dateien <i>Projektname.dpr</i> bzw. <i>Projektname.imprint</i> . Wenn die Dateien nicht existieren, dann werden hier auch keine Links angezeigt.
3	Buttons für Actions
4	Angaben zum angemeldeten Nutzer

3 Theming

3.1 Farbschema festlegen

Das Farbschema und die Schriftarten der Web-Anwendung werden 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.

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

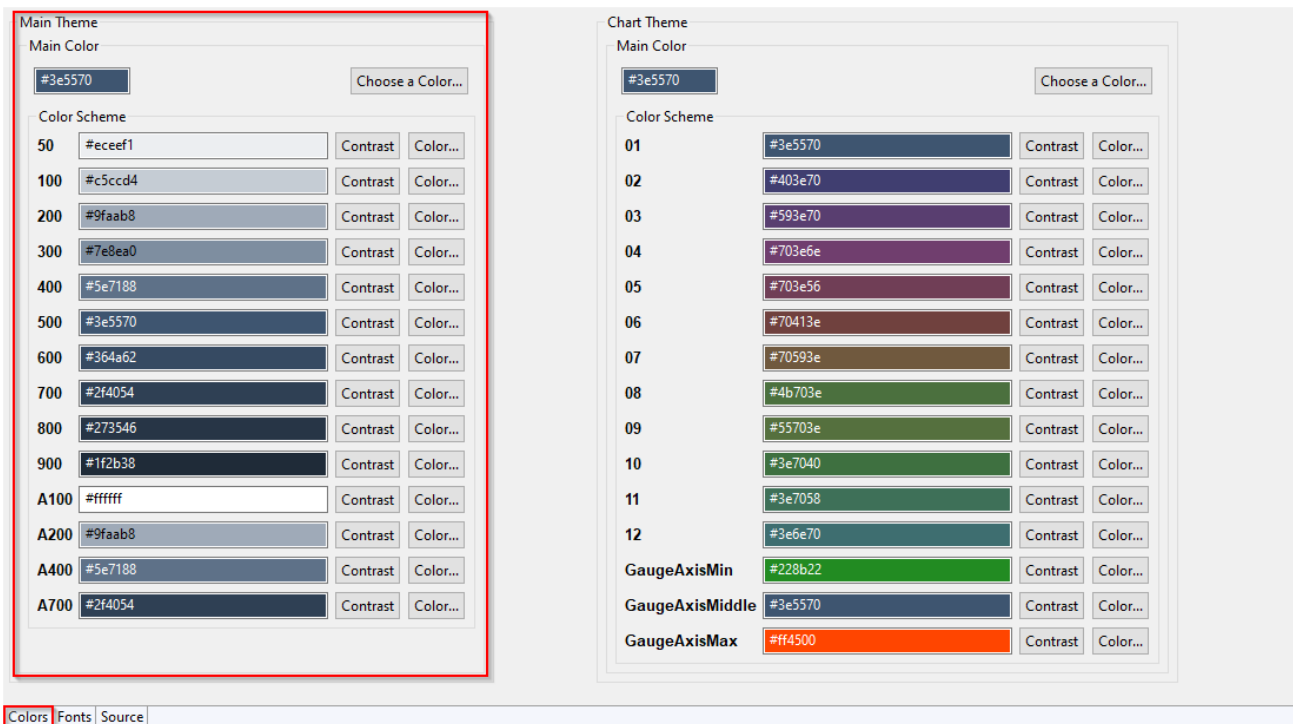
3.2 Farben

Farben werden in Web Apps mithilfe der **Color Scheme** (Farbschema) angewendet. Jedes Color Scheme hat eine eindeutige Bezeichnung. Die Bezeichnung des Color Scheme kann in Attributen angegeben werden, um das Color Scheme auf ein Element anzuwenden.

3.2.1 Main Theme

Unter Main Theme lassen sich die Farben für die Darstellung der Web App festlegen. Durch Angabe einer Farbe unter Main Color lässt sich eine Hauptfarbe definieren. Ausgehend von der Hauptfarbe

werden die restlichen Farben automatisch berechnet. Die Farben lassen sich jedoch auch einzeln definieren.



In der folgenden Tabelle sehen Sie die standardmäßige Verwendung der für die im Farbschema definierten Farbe (**Color**) und Kontrast (**Contrast**):

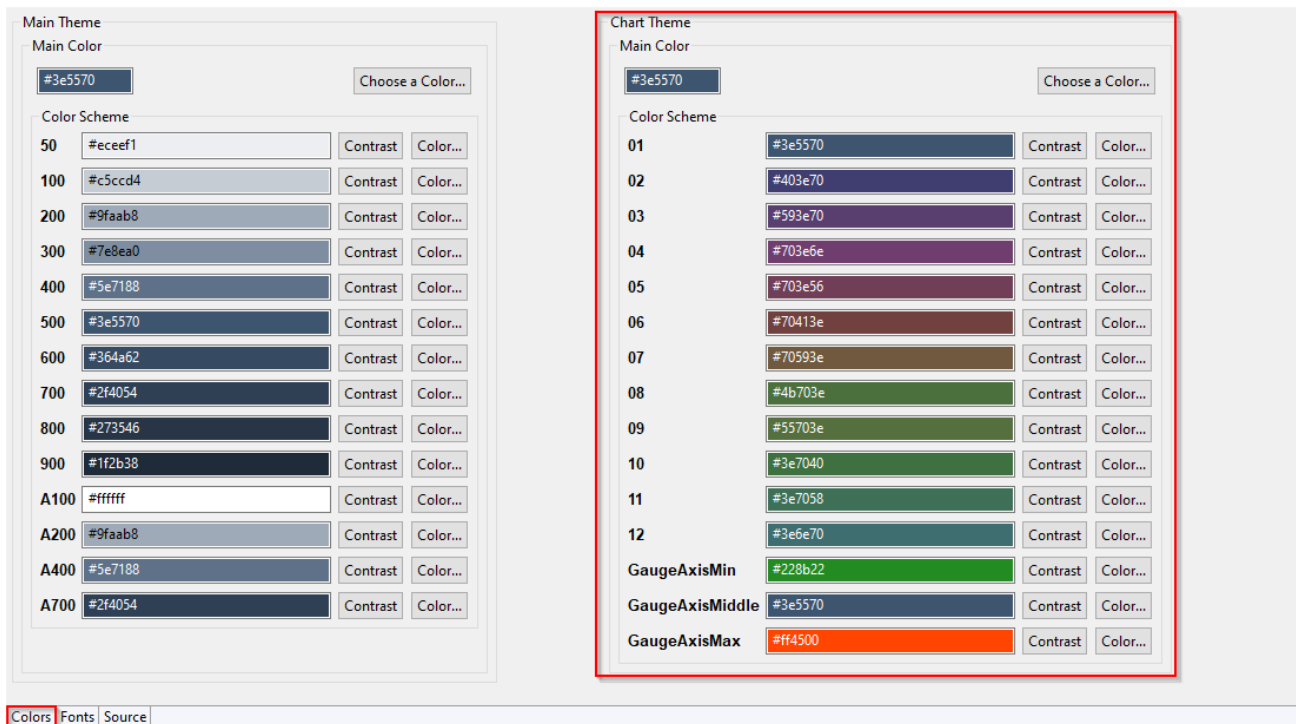
i Mit **Contrast** kann zwischen schwarz und weiß umgeschaltet werden.

Color Scheme	Color	Contrast
50	Zebra-Tabelle Hintergrund der geraden Zeile	Zebra-Tabelle Schriftfarbe der geraden Zeile
100	Zebra-Tabelle Hintergrund der ungeraden Zeile	Zebra-Tabelle Schriftfarbe der ungeraden Zeile
200	Zebra-Tabelle Hintergrund bei Maus-Hover	Zebra-Tabelle Schriftfarbe bei Maus-Hover
400	Button Hintergrund bei Maus-Hover	Button Schriftfarbe bei Maus-Hover

Color Scheme	Color	Contrast
500	Webseite • Hintergrund Header • Hintergrund Menü • Hintergrund des gewählten Menü-Eintrags TextBox • Labels der aktiven TextBox • gewähltes Datum im DatePicker Button • Hintergrund ComboBox • gewählter Eintrag bei geöffneter Liste DatePicker • gewähltes Datum Login • Hintergrund TabGroup • Aktiver Tab	Webseite • Schriftfarbe Header • Schriftfarbe Menü • Schriftfarbe des gewählten Menü-Eintrags Button • Schriftfarbe
A100	Web App • Hintergrund • RadioButton • CheckBox	
A200	Menü: • Hintergrund bei Maus-Hover Zebra-Tabelle • Hintergrund der gewählten Zeile	Zebra-Tabelle • Schriftfarbe der gewählten Zeile
A400	Zebra-Tabelle • Hintergrund der Tabellenkopfzeile Link • Schriftfarbe	Zebra-Tabelle • Schriftfarbe der Tabellenkopfzeile

3.2.2 Chart Theme

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



In der folgenden Tabelle sehen Sie die standardmäßige Verwendung der für die im Farbschema definierten Farbe (**Color**) und Kontrast (**Contrast**):

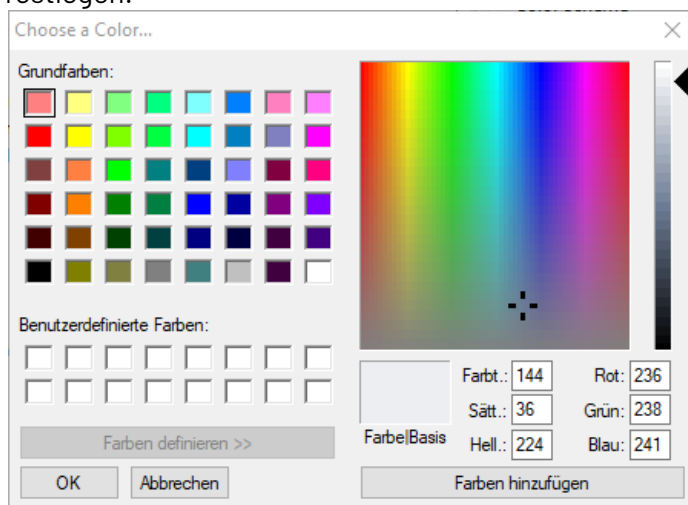
i Mit **Contrast** kann zwischen schwarz und weiß umgeschaltet werden.

Color Scheme	Color	Contrast
01 - 12	- Line - Liniendiagramm - Bar - Balkendiagramm - Pie - Kreis/Donut-Diagramm	- Line - Beschriftung - Bar - Beschriftung - Pie - Beschriftung
GaugeAxisMin	Gauge Anfangsbereich des Tachodiagramms	
GaugeAxisMiddle	Gauge mittleren Bereich des Tachodiagramms	

Color Scheme	Color	Contrast
GaugeAxisMax	Gauge • Endbereich des Tachodiagramms	

3.2.3 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.

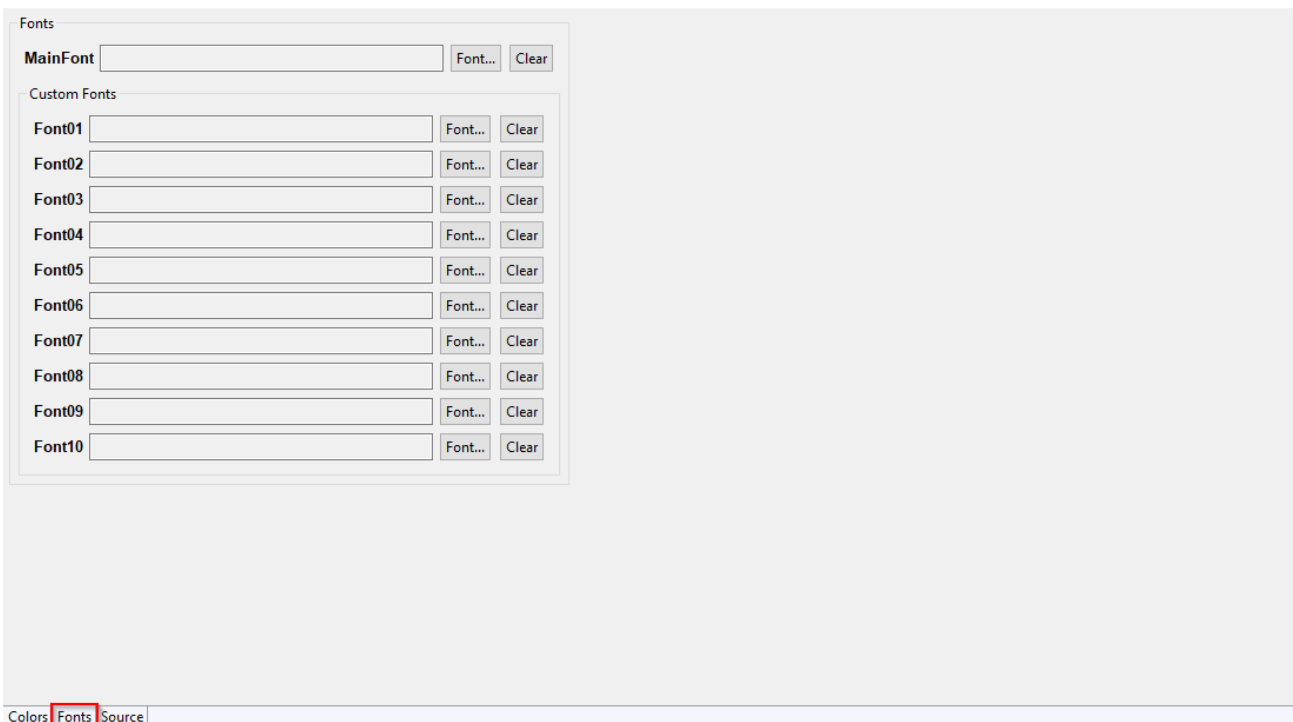


4. **Farbe hinzufügen** klicken und die angegebene Farbe zu den benutzerdefinierten Farben hinzufügen.
5. In **Benutzerdefinierte Farben** die Farbe auswählen und **OK** klicken.
Die Farbe wurde nun übernommen.

 Die Dokumentation der ReST-Schnittstelle finden Sie unter <http://<server>:8080/X4/X4Api/>.

3.3 Schriftarten

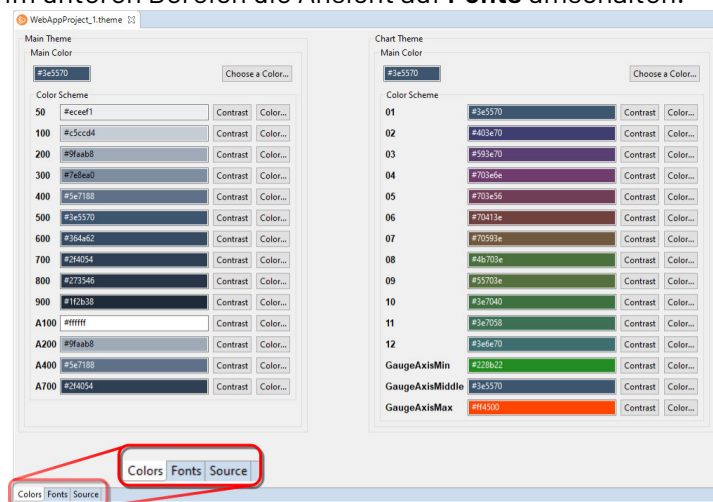
Die Hauptschriftart der Web App wird mithilfe des **MainFont** festgelegt. Mit Custom Fonts können weitere Schriftarten festgelegt werden. Jeder Schriftart hat eine eindeutige Bezeichnung. Die Bezeichnung des Fonts kann in Attributen angegeben werden, um die Schriftart auf ein Element anzuwenden.



3.3.1 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.

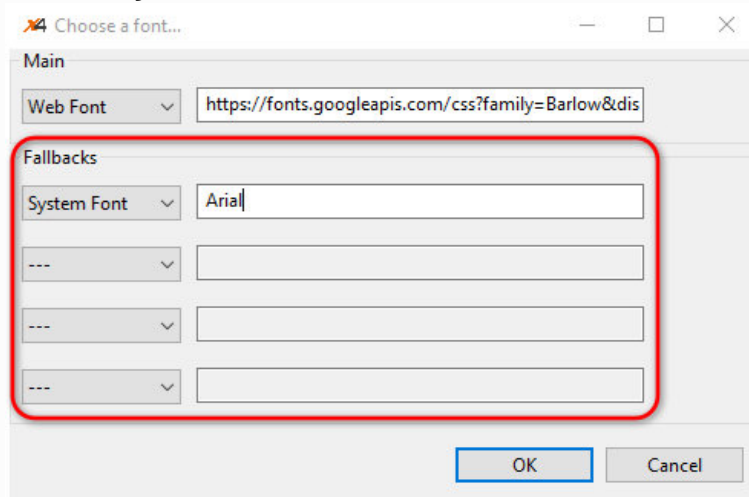


3. Unter **MainFont** wird die Schriftart eingestellt, die als Hauptschriftart der Web App verwendet wird. Hinter **MainFont** auf **Font...** klicken, um die Schriftart festzulegen.
4. Unter **Main** wählen, wo die Schrift zu finden ist.

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.

- ⓘ Die Dokumentation der ReST-Schnittstelle finden Sie unter <http://localhost:8080/X4/X4Api/>.

4 Configuration

Das Verhalten der Web-Anwendungen, die mit Web Apps erstellt werden, wird in der Configuration `.wac` definiert. In der `.wac`-Datei werden unter anderem allgemeine Einstellungen der Web App, wie z. B. das Logo, und das Verhalten beim Login definiert.

Das Wurzelement der `.wac`-Datei ist `<Configuration>`. Innerhalb des Wurzelements wird mit den Elementen `General`, `Access` und `Status` das Aussehen und Verhalten der Web App gesteuert. Bei der Deklaration von `.wac`-Dateien wird die Datei automatisch validiert. So können mögliche Fehlerquellen bereits beim Erstellen von Web-Anwendungen reduziert werden.

4.1 Neue Configuration anlegen

1. Im *X4 Designer* ein Web App Projekt rechtsklicken.
2. Im Kontextmenü **New > Configuration** wählen.

Eine Configuration `.wac` wird angelegt und geöffnet.

4.2 Zugriff auf Web Apps verwalten

Der Zugriff auf eine Web App wird mithilfe der Rolle aus Keycloak verwaltet. In Keycloak kann jedem Benutzer eine Rolle zugewiesen werden. Mithilfe der zugewiesenen Rollen kann der Zugriff auf eine Web App gesteuert werden.

In der Configuration `.wac` kann für die Web App festgelegt werden, welche Rolle einem Benutzer zugewiesen sein muss, damit der Benutzer auf die Web App zugreifen kann.

✔ Weitere Informationen finden Sie im Abschnitt [Configuration deklarieren](#).

4.3 Configuration deklarieren

Das Wurzelement der `.wac`-Datei ist `<Configuration>`. Das Element enthält alle Konfigurationen zur Web App. Über Attribute wird die Darstellung und das Verhalten der Web App gesteuert.

Das Element `<Configuration>` kann folgende Elemente enthalten:

Mögliche Elemente

Element	Beschreibung
General	Enthält Konfigurationsdefinitionen zu der allgemeinen Darstellung der Web App.
Access	Enthält Konfigurationsdefinitionen zum Zugriff auf die Web App.

Element	Beschreibung
Status	Enthält Konfigurationsdefinitionen zu den internen Statuswerten der Web App.
MapAPIKey	Enthält den API-Schlüssel, falls dieser vom Kartenanbieter verlangt wird. i 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. ⚠ Wenn Google Maps als Kartenanbieter verwendet wird, müssen folgende APIs aktiviert sein: • Maps JS API • Locations (für die Suche und die Routenberechnung) • Directions (für die Routenberechnung)
Logout	Enthält Konfigurationsdefinitionen zum Logout.
Notifications	Enthält Konfigurationsdefinitionen zum Infobanner.
HTTPSecurityHeader	Enthält Konfigurationsdefinitionen zum HTTP-Security-Header.
HeaderIncludes	Enthält Konfigurationsdefinitionen zum Header. Im Header können Sie zum Beispiel eigenen Javascript-Code einfügen.

i Die Dokumentation der ReST-Schnittstelle finden Sie unter <http://localhost:8080/X4/X4Api/>.

✓ Weitere Informationen finden Sie im Abschnitt

4.3.1 <General>

Das Element `General` enthält Konfigurationsdefinitionen zu der allgemeinen Darstellung der Web App, wie zum Beispiel das Favicon, der Anzeigename oder ob Kopf- und Fußzeilen angezeigt werden.

Mögliche Attribute

Attribut	Beschreibung
displayName	<i>Erforderlich.</i> Anzeigename der Web App. Wird im Browsertab bzw. der Titelleiste angezeigt. Mögliche Werte: Beliebige Zeichenkette

Attribut	Beschreibung
<code>enableOverlayBlurredBackground</code>	Angabe, ob der Hintergrund bei der Anzeige von Overlay-Strukturelementen verschwommen oder klar dargestellt werden soll Mögliche Werte: • <code>true</code> (Standard): Bei Overlays wird der Teil der Web App, der noch im Hintergrund zu sehen ist, verschwommen dargestellt. • <code>false</code>: Bei Overlays wird der Teil der Web App, der noch im Hintergrund zu sehen ist, klar dargestellt.
<code>faviconUrl</code>	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. Mögliche Werte: Zeichenkette (URI)
<code>logoUrl</code>	Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Logo der Web App verwendet wird. i • Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources. • Das Material Icon muss mit dem Präfix <code>icon:</code>, z. B. <code>icon:<MaterialIconName></code> angegeben werden. Mögliche Werte: • Zeichenkette (URI), z. B. <code>clock.png</code> • <code>icon:<MaterialIconName></code>, z. B. <code>icon:extension</code> ✓ Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren Material Icons abweichen. i Das Logo wird immer mit einer Höhe von 45 Pixeln angezeigt. Die Grafikdatei, die als Logo verwendet wird, wird automatisch skaliert.

Attribut	Beschreibung
logoColor	Definiert die Farbe des Logos in der Anwendung.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: - Hexadezimaler Farbwert, z. B. <i>ff5a00</i>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>
showFooter	Definiert, ob die Fußzeile der Web-Anwendung angezeigt wird. Mögliche Werte: <code>true</code> (Standard) / <code>false</code>  Falls in der angezeigten Component Actions vorhanden sind, wird die Fußzeile in der mobilen Ansicht unabhängig vom Wert des Attributs angezeigt.
showHeader	Definiert, ob die Kopfzeile der Web-Anwendung angezeigt wird. Mögliche Werte: <code>true</code> (Standard) / <code>false</code>
imprintLink	Verlinkt externe Seiten als Impressum.
dprLink	Verlinkt externe Seiten als Datenschutzerklärung.
enableCookieConsent	Definiert, ob das Pop-up mit der Cookie-Zustimmung angezeigt wird. Mögliche Werte: <code>true</code> (Standard) / <code>false</code>



Weitere Informationen finden Sie im Abschnitt

4.3.2 <Access>

Das Element Access enthält Konfigurationsdefinitionen zum Zugriff auf die Web App.

Mögliche Attribute

Source	Beschreibung
accessRight	Definiert, welche Rolle auf das Modul zugreifen kann. Mögliche Werte: Name einer zuvor definierten Rolle  Die Rollen werden in der Administrationskonsole von Keycloak verwaltet.

Source	Beschreibung
authorizationFlow	Definiert, welcher Autorisierungsablauf verwendet wird. Wenn keine Anmeldung erforderlich sein soll, verwenden Sie <code>Public Access</code>.  Um die Single-Sign-On-Funktionalität zu verwenden, müssen Sie den Autorisierungsablauf <code>Authorization Code Flow</code> verwenden. Mögliche Werte: • <code>Public Access</code>  Benutzer, die ohne Login auf die Web App zugreifen, werden als <code>ANONYMOUS_USER</code> erkannt. Diese können keine besonderen Rechte erhalten. • <code>Ressource Owner Password Flow</code>  Wenn eine Web App den Autorisierungsablauf <code>Ressource Owner Password Flow</code> verwendet, kann sich ein Benutzer mit einem temporären Passwort nicht an dieser Web App anmelden. Wenn Sie temporäre Passwörter einsetzen möchten, verwenden Sie den Autorisierungsablauf <code>Authorization Code Flow</code>. <code>Ressource Owner Password Flow</code> wird mit der nächsten Version abgekündigt. Aus Sicherheitsgründen empfehlen wir den <code>Authorization Code Flow</code> zu verwenden. • <code>Authorization Code Flow</code>  Keycloak muss entsprechend dem gewählten Autorisierungsablaufs konfiguriert werden.  Weitere Informationen finden Sie in der offiziellen Auth0® Inc. Webseite. • https://auth0.com/docs/authorization/flows/resource-owner-password-flow • https://auth0.com/docs/authorization/flows/authorization-code-flow

Source	Beschreibung
realm	Definiert, welcher Realm aus Keycloak verwendet wird.  Die Angabe des Realms wird benötigt, wenn der Autorisierungsablauf Authorization Code Flow oder Resource Owner Password Flow verwendet wird und dabei ein vom Standard abweichender Realm verwendet werden soll.
client	Definiert, welcher Client aus Keycloak verwendet wird.  Die Angabe des Clients wird benötigt, wenn der Autorisierungsablauf Authorization Code Flow oder Resource Owner Password Flow verwendet wird und dabei ein vom Standard abweichender Client verwendet werden soll.
clientSecret	Client Secret des definierten Clients, der aus Keycloak ausgelesen werden kann.  Die Angabe des Client Secrets wird benötigt, wenn der Autorisierungsablauf Authorization Code Flow oder Resource Owner Password Flow verwendet wird und dabei ein vom Standard abweichender Client Secret verwendet werden soll.
managementUser	Benutzername des Benutzers, der die Keycloak ReST-API ansteuert.  Die Angabe der Benutzerdaten wird benötigt, wenn der Autorisierungsablauf Resource Owner Password Flow verwendet wird.
managementUserPassword	Passwort des Benutzers, der die Keycloak ReST-API ansteuert.  Die Angabe der Benutzerdaten wird benötigt, wenn der Autorisierungsablauf Resource Owner Password Flow verwendet wird.

✔ Weitere Informationen finden Sie im Abschnitt

4.3.3 <Status>

Das Element Status enthält Konfigurationsdefinitionen zu den internen Statuswerten der Web App.

Mögliche Attribute

Attribut	Beschreibung
active	Angabe, ob das Projekt aktiviert ist. Mögliche Werte: true / false (Standard)  Wenn das Projekt nicht aktiv ist, kann die Web-Anwendung im Browser nicht angezeigt werden.
localOnly	Angabe, ob das Projekt nur lokal bereitgestellt werden soll. Mögliche Werte: true / false (Standard)  Wenn diese Option aktiviert ist, kann die Web-Anwendung nur im lokalen System im Browser angezeigt werden.
dataFormatVersion	Angabe, welches Datenformat verwendet wird. Mögliche Werte: • 1.0 • 2.0 (Standard)  Der Wert des Attributs kann nicht während der Laufzeit geändert werden.

4.3.3.1 dataFormatVersion

Technical Processes in der X4 BPMS können Daten in verschiedenen Formaten ausgeben. Um die Daten in X4 Web Apps zu verwenden, muss in der Konfiguration der Web App festgelegt werden, in welchem Datenformat die Daten vom Technical Process ausgegeben werden.

Das verwendete Datenformat wird im Attribut dataFormatVersion festgelegt.

4.3.3.1.1 Beispiel

Dieses Beispiel zeigt die Struktur der Datenformat-Version 1.0.

```
<Object property="Container">
  <String value="test" property="Text" />
</Object>
```

Dieses Beispiel zeigt die Struktur der Datenformat-Version 2.0.

```
<Container>
  <Text>test</Text>
</Container>
```

✔ Weitere Informationen finden Sie im Abschnitt

4.3.4 <Logout>

Das Element Logout enthält Konfigurationsdefinitionen zum Logout.

Mögliche Attribute

Attribut	Beschreibung
redirectUrl	Weiterleitung der Anwender nach dem Logout auf die angegebene URL. Uniform Resource Locator (URL) mit beliebigem gültigen Protokoll (http://, https://, etc.) Mögliche Werte: Beliebige gültige URL (z.B. https://www.softproject.de/)

✔ Weitere Informationen finden Sie im Abschnitt

4.3.5 <Notifications>

Das Element Notifications erzeugt ein Infobanner in der Web App.

⚠ Der Inhalt des Infobanners wird alle 30 Minuten automatisch neu geladen. Um den Inhalt des Infobanners manuell neu zu laden, aktualisieren Sie Ihr X4 Repository.

Mögliche Attribute

Attribut	Beschreibung
infoBannerProcess	<i>Erforderlich.</i> Pfad zur .wrf-Datei, die die Daten liefert. Der Technical Process muss im Ordner Services/Processes enthalten sein. Mögliche Werte: Zeichenkette

4.3.5.1 Technical Process

Damit ein Infobanner angezeigt werden kann, muss ein Technical Process die Daten in einer vorgegebener XML-Struktur ausgeben.

```
<Alerts>
  <Alert>
    <Background></Background>
    <Foreground></Foreground>
    <Icon></Icon>
    <Message></Message>
  </Alert>
</Alerts>
```

Element	Beschreibung
Background	Definiert die Hintergrundfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas. Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
Foreground	Definiert die Schriftfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas. Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Element	Beschreibung
Icon	Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Icon der Component verwendet wird.  - Ein Material Icon muss mit dem Präfix <code>icon:</code>, z. B. <code>icon:<MaterialIconName></code> angegeben werden. Mögliche Werte: - Zeichenkette (URI), z. B. <code>clock.png</code> <code>icon:<MaterialIconName></code>, z. B. <code>icon:extension</code> - Beliebige URL, z. B. <code>http://mysite.com/logo.png</code>
Message	Nachricht des Infobanners. Mögliche Werte: Beliebige Zeichenkette

4.3.5.1.1 Beispiel

```

<Alerts>
  <Alert>
    <Background>A700</Background>
    <Foreground>900</Foreground>
    <Icon>icon:3d_rotation</Icon>
    <Message>Upcoming maintenance</Message>
  </Alert>
</Alerts>

```

4.3.6 <HTTPSecurityHeader>

Das Element `HTTPSecurityHeader` erzeugt einen HTTP-Security-Header in der Web App.

Element	Beschreibung	Mögliche Werte
<code>HTTPSecurityHeader</code>	Der HTTP-Security-Header ist ein HTTP-Antwort-Header. Der HTTP-Security-Header wird verwendet, um die Web App vor unerwünschten Angriffen zu schützen.	Das Element enthält folgende Elemente: <code>strictTransportSecurity</code>
<code>strictTransportSecurity</code>	Der HTTP-Strict-Transport-Security-Antwort-Header informiert die Browser darüber, dass der Zugriff auf die Website nur über HTTPS erfolgen sollte und dass alle zukünftigen Zugriffsversuche über HTTP automatisch in HTTPS umgewandelt werden sollten.	

4.3.6.1 HTTPSecurityHeader

Mögliche Attribute

Attribut	Beschreibung	Mögliche Werte
XFrameOptions	Die X-Frame-Options im HTTP-Antwort-Header kann verwendet werden, um zu bestimmen, ob ein aufrufender Browser die Zielseite in einem <frame>, <iframe>, <embed> oder <object> einbetten darf.	• DENY: Die Zielseite darf nicht eingebettet werden. • SAMEORIGIN: Die Zielseite darf eingebettet werden.

4.3.6.2 strictTransportSecurity

Mögliche Attribute

Attribut	Beschreibung	Mögliche Werte
enabled	Legt den Status des HTTP-Strict-Transport-Security-Antwort-Headers fest.	• true: Der Strict-Transport-Security-Antwort-Header ist aktiviert. • false: Strict-Transport-Security-Antwort-Header ist deaktiviert.
includeSubDomains	Legt fest, ob die Einstellungen zum HTTP-Strict-Transport-Security-Antwort-Header auch für die Subdomänen der Web App gelten.	• true: Die Einstellungen gelten auch für die Subdomänen. • false: Die Einstellungen gelten nicht für die Subdomänen.
maxAge	Die Zeit in Sekunden, in der sich der Browser merken soll, dass auf die Web App nur über HTTPS zugegriffen werden darf.	• Beliebige Ganzzahl

Attribut	Beschreibung	Mögliche Werte
preLoad	Legt fest, ob der HTTP-Strict-Transport-Security-Antwort-Header vorgeladen wird.	• true: Der HTTP-Strict-Transport-Security-Antwort-Header wird vorgeladen • false: Der HTTP-Strict-Transport-Security-Antwort-Header wird nicht vorgeladen.

4.3.7 <HeaderIncludes>

Das Element `HeaderIncludes` enthält Konfigurationsdefinitionen zum HTML Head Element der Web App. Im Head Element können Sie zum Beispiel eigenen Javascript-Code einfügen.

Element	Beschreibung	Mögliche Werte
HeaderIncludes		Das Element enthält folgende Elemente: • Include
Include	Fügt Inhalte aus einer Datei in den HTML Head als Metaelemente ein.	

4.3.7.1 Include

Mögliche Attribute

Attribut	Beschreibung	Mögliche Werte
path	Pfad zu einer Datei im Projekt  Die Datei muss sich im Verzeichnis Ressources befinden.	• Beliebige Zeichenkette (z. B. <i>File.txt</i>)

✓ Weitere Informationen finden Sie im Abschnitt

5 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.

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).

Hinweis

Der Wert für name entspricht dem Dateinamen der Sprachdatei mit Dateiendung.
Der Wert für displayName wird bei der Sprachauswahl angezeigt:

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

5.1 Eine neue Sprachdatei anlegen

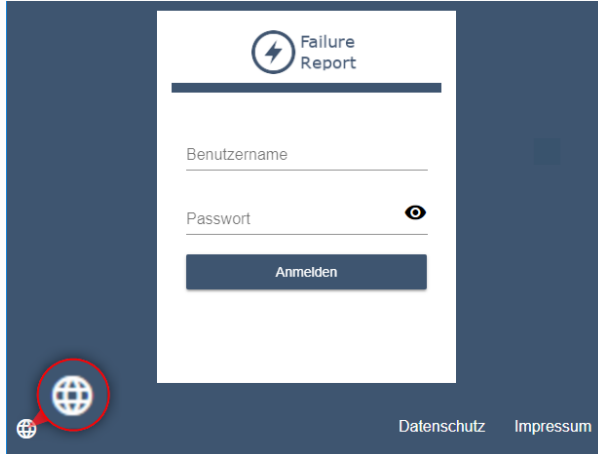
1. Auf den Ordner Translations rechtsklicken.
Das Kontextmenü öffnet sich.
2. Auf **New > Translation** klicken.
Der Dialog für das Erstellen einer neuen Sprachdatei öffnet sich.
3. Im Feld **File name** den Namen der Sprachdatei eingeben.
4. Auf **Finish** klicken.

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.

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



5.2 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=Woche
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
start=Start
destination=Ziel
search=Suche
routingErrorTitle=Fehler
routingError=Die Route konnte nicht berechnet werden.
languagesLabel=Sprachauswahl
scrollingDesktopNonMacWarning=Verwenden Sie Strg + Scrollen zum Zoomen der Karte
scrollingDesktopMacWarning=Verwenden Sie \u2318 + Scrollen zum Zoomen der Karte
scrollingMobileWarning=Verwenden Sie zwei Finger zum Bewegen der Karte
registerLabel = Registrieren
titleLabel = Ihre Privatsphäre ist uns wichtig
defaultCookieConsentText = Einige Cookies sind für die einwandfreie Nutzung der
Webseite unerlässlich, während andere uns helfen, Ihr Erlebnis auf dieser Webseite zu
optimieren und zu verbessern.
dprLabel = Datenschutz
imprintLabel = Impressum
externalMediaCookiesTitle = Externe Medien
```

```
externalMediaCookiesDescription = Inhalte von Videoplattformen und Social-Media-  
Plattformen sind standardmäßig blockiert. Wenn ExternalMedia-Cookies akzeptiert  
werden, ist der Zugriff auf diese möglich.  
preferencesLabel = Vorlieben  
necessaryCookiesBtn = Nur notwendige Cookies zulassen  
allowSelectionBtn = Auswahl zulassen  
preferencesTitle = Cookie-Einstellungen  
preferencesText = Cookies sind kleine Textdateien, die von Webseiten verwendet werden  
können, um Interaktionen zu analysieren und zu messen und Ihr Erlebnis auf dieser  
Webseite zu optimieren und zu verbessern. Diese Webseite verwendet verschiedene Arten  
von Cookies. Einige Cookies sind für den Betrieb dieser Webseite unerlässlich; für  
alle anderen Arten von Cookies benötigen wir Ihre Zustimmung. Einige Cookies werden  
von Diensten Dritter gesetzt, die auf unseren Webseiten erscheinen.  
allowAllCookiesBtn = Alle Cookies zulassen  
essentialCookiesTitle = Essentielle  
essentialCookiesDescription = Essentielle Cookies sind erforderlich, um grundlegende  
Funktionen zu nutzen und um das reibungslose Funktionieren der Website zu  
gewährleisten.  
sixMonthText = 6 Monate  
youtubePurposeText = Wird verwendet, um YouTube-Inhalte zu entsperren.  
bingPurposeText = Wird verwendet, um Bing-Inhalte zu entsperren.  
showCookiesDetailsLabel = Details zu Cookies anzeigen  
acceptLabel = Akzeptieren  
nameLabel = Name  
providerLabel = Anbieter  
purposeLabel = Aufgabe  
hostLabel = Host(s)  
cookieNameLabel = Cookie Name  
expiryLabel = Cookie Ablauf  
waitTitle = Bitte warten  
waitText = Sie werden in Kürze weitergeleitet.  
waitButton = Zur Anmeldung
```

6 Anmeldeseite

⚠ Wenn in der Configuration .wac das Attribut `authorizationFlow=Authorization Code Flow` festgelegt wurde, wird die Anmeldeseite des Authentifizierungsproviders Keycloak verwendet.

Die Anmeldeseite wird angezeigt, sobald sich ein Benutzer beim Aufrufen der X4 Web App anmelden muss. Die Login-Maske ist der zentrale Bestandteil der Anmeldeseite.

Das Diagramm zeigt die Struktur der Login-Maske mit vier nummerierten Elementen:

- 1**: Logo (X4 Web Apps Logo)
- 2**: Passwortfeld (mit einem Auge-Symbol für das Umschalten der Sichtbarkeit)
- 3**: Slider für die Funktion "Angemeldet bleiben"
- 4**: Link "Passwort vergessen?"

Unter den Elementen befindet sich ein dunkelblauer Button mit der Aufschrift "Anmelden".

1	Logo. Wird in der Configuration 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>)

Sie können die Login-Maske anpassen oder eine auf Sie zugeschnittene Anmeldeseite erstellen.

Um die Login-Maske anzupassen, muss eine `.login`-Datei im Web App-Projekt erstellt werden.

Die `.login`-Datei enthält das Element `<Login>`. Mit den Attributen des Elements kann die Login-Maske konfiguriert werden.

Um eine individuelle Login-Maske zu erstellen, muss innerhalb des Elements `<Login>` eine Detail Component deklariert werden. In der Detail Component können alle Layouts und Controls der X4 Web Apps verwendet werden.

6.1 Anmeldeseite anlegen

Die Anmeldeseite wird nach dem Web App-Projekt benannt und hat die Dateiendung .login. Innerhalb eines Web App-Projekts kann nur eine .login-Datei angelegt werden.

Das Element `<Login>` erzeugt die standardmäßige Login-Maske. Mit den Attributen des Elements kann die Login-Maske konfiguriert werden.

Um eine individuelle Login-Maske zu erstellen, muss innerhalb des Elements `<Login>` eine **Detail Component** deklariert werden. In der Detail Component können alle Layouts und Controls der X4 Web Apps verwendet werden.

 Die Detail Component überschreibt die standardmäßige Login-Maske. Wenn Sie eine Detail Component deklarieren, müssen Sie in der Detail Component eine individuelle Login-Maske definieren.

1. Mit der rechten Maustaste auf das Web App-Projekt klicken.
2. Auf **New > Login** klicken.
Die Datei für die Anmeldeseite wird erzeugt.

6.1.1 `<Login>`

 Das Element `<Login>` kann folgendes Element enthalten:

- `DetailComponent`

Die in der `DetailComponent` gesetzten Attribute werden ignoriert und haben keine Auswirkungen auf die Anmeldeseite.

Folgende Attribute lassen sich für das Element `<Login>` definieren:

Attribut	Beschreibung
<code>backgroundImageUrl</code>	Legt eine Grafik als Hintergrund fest.  Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts liegen.
<code>changePassword</code>	Erzeugt im Benutzermenü der X4 Web App die Passwort ändern -Schaltfläche.

Attribut	Beschreibung
loginMaskSizeDesktop	Definiert die Größe der Login-Maske, wenn die X4 Web App auf Desktop-Computern angezeigt wird. Mögliche Werte: • <code>Third</code>: Die Login-Maske nimmt ein Drittel der Anmeldeseite ein. • <code>Half</code>: Die Login-Maske nimmt die Hälfte der Anmeldeseite ein. • <code>Full</code>: Die Login-Maske nimmt die gesamte Anmeldeseite ein.
loginMaskSizeTablet	Definiert die Größe der Login-Maske, wenn die X4 Web App auf Tablets angezeigt wird. Mögliche Werte: • <code>Third</code>: Die Login-Maske nimmt ein Drittel der Anmeldeseite ein. • <code>Half</code>: Die Login-Maske nimmt die Hälfte der Anmeldeseite ein. • <code>Full</code>: Die Login-Maske nimmt die gesamte Anmeldeseite ein.

6.2 Passwort vergessen-Schaltfläche erzeugen

Um auf der Anmeldeseite eine **Passwort vergessen**-Schaltfläche zu erzeugen, muss in der Datei `.login` im Element `<Login>` ein [Button Control](#) eingefügt werden.

In dem [Button Control](#) muss anschließend eine [Passwort vergessen-Action](#) eingefügt werden.

6.2.1 Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<Login
  xmlns="http://softproject.de/webapp/1.0">
  <DetailComponent>
    <FlowLayout>
      <Button>
        <ForgotPasswordAction displayName="$forgotPasswordLabel"/>
      </Button>
    </FlowLayout>
  </DetailComponent>
</Login>
```

7 Module

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

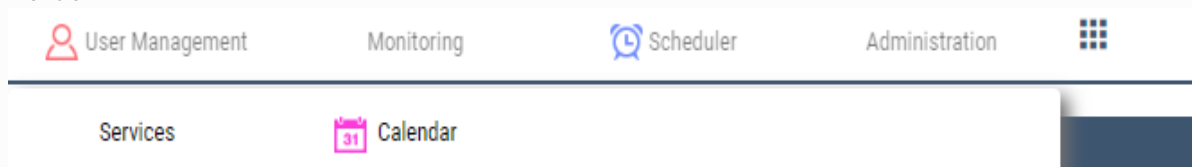
Attribut	Beschreibung
<code>accessRight</code>	Definiert, welche Rolle auf das Modul zugreifen kann. Mögliche Werte: Name einer zuvor definierten Rolle  Die Rollen werden in der Administrationskonsole von Keycloak verwaltet.
<code>default</code>	Definiert, ob das Modul zuerst geladen und geöffnet werden soll.  Lediglich ein Modul darf den Wert <code>default="true"</code> haben. Mögliche Werte: <code>true</code> / <code>false</code>
<code>displayName</code>	Anzeigename des Moduls. Wird im Browsertab angezeigt. Mögliche Werte: Beliebige Zeichenkette
<code>hideOnAccessDenied</code>	Gibt an, ob Benutzer/Gruppen, die keinen Zugriff auf das Modul haben, das Modul sehen können. Mögliche Werte: <code>true</code> / <code>false</code>  Für mehr Informationen, siehe auch Zugriffsrechte für Module definieren.
<code>iconColor</code>	Definiert die Farbe des Icons in der Anwendung.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: - Hexadezimaler Farbwert, z. B. <code>ff5a00</code>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <code>A200</code>

Attribut	Beschreibung
iconUrl	Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Icon des Moduls verwendet wird. i - Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources. - Das Material Icon muss mit dem Präfix icon, z. B. icon:<MaterialIconName> angegeben werden. Mögliche Werte: - Zeichenkette (URI), z. B. <i>clock.png</i> - icon:<MaterialIconName>, z. B. <i>icon:extension</i> ✓ Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren Material Icons abweichen.
path	Pfad zum Modul. Wird in der Adresszeile des Browsers angezeigt. Mögliche Werte: Zeichenkette aus alphanumerischen Zeichen (keine Umlaute, Punkte usw.)

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 funktioniert, 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

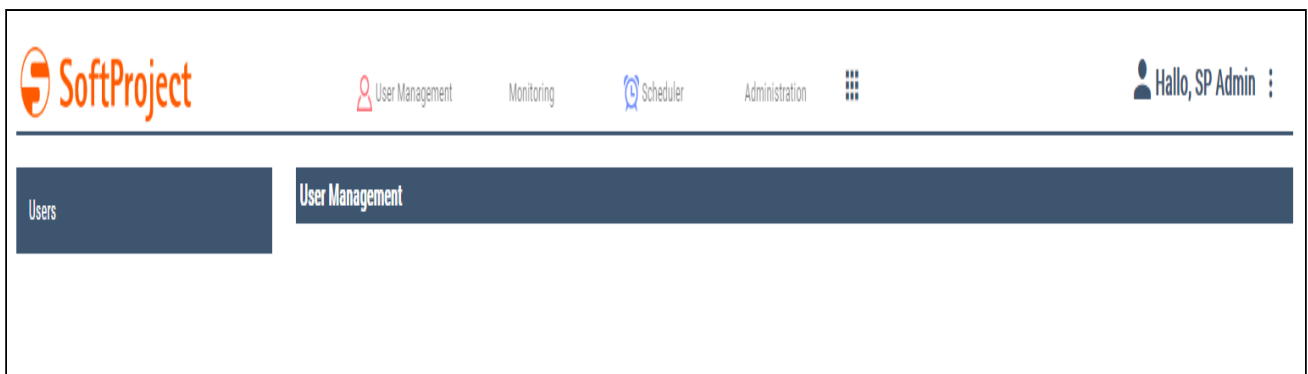
Beispiel: Web App mit mehreren Modulen

```

<WebApp xmlns="http://softproject.de/webapp/1.0" path="Modules">
  <Modules>
    <Module displayName="User Management" path="UserManagement" iconUrl="users.png" 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:**

 Hallo, SP Admin

 User Management

Users

Monitoring

Instances

 Scheduler

Executions

Administration

CustomPlaceholder

Services

Root

 Calendar

Calendar

Passwort ändern

Abmelden

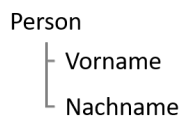
8 Datenmodell und Properties-Definition

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

Properties werden am Beginn der Componentsdeklaration 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	Anzeigename der Property. Wird in den Spaltenüberschriften von Master/Detail Components, List Components und Grid Components verwendet. Mögliche Werte: Beliebige Zeichenkette

Attribut	Beschreibung
name	Name der Property. Erforderlich. Mögliche Werte: Beliebige Zeichenkette aus alphanumerischen Zeichen (a-z, A-Z, 0-9)  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.
readOnly	Schränkt den Zugriff auf Nur-Lesen ein. Mögliche Werte: <code>true</code> / <code>false</code>

Attribut	Beschreibung
type	Typ der Property. Erforderlich. Mögliche Werte: • Base64: Base64-kodierte Daten • Boolean: Boole'sche Werte • Color: Hexadezimale Farbwerte • Complex: Objekt enthält weitere Objekte (verschachtelte Properties). Wird z. B. für Listeneinträge verwendet. • Date: Datumsangabe im ISO-Format YYYY-MM-DD (z. B. 2020-01-18) • DateTime: Datums- und Zeitangaben im Format YYYY-MM-DD'T'hh:mm:ss.fff (z. B. 2019-01-01T01:00:00.000Z)  Das Format der Datums- und Zeitangaben muss konform zu ISO 8601 sein.  Wenn bei einer Zeitangabe keine Zeitzone angegeben wird, dann wird die Zeitzone des Servers verwendet. Serverseitig wird dann diese Zeitzone ergänzt.  Wenn ein Wert mit dem Property-Typ DateTime in einer List Component verwendet wird, dann wird der Angabe von Datum und Uhrzeit abhängig von den vorhandenen Werten unterschiedlich angezeigt: • Wenn alle Werte den Wert 0 bei Sekunden haben (2020-01-01T00:00:00.000Z), dann werden in der gesamten Spalte keine Sekunden angezeigt. • Wenn mindestens ein Wert einen Wert abweichend von 0 hat, werden die Sekunden in der gesamten Spalte angezeigt.  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! • Decimal: Kommazahlen • Image: Grafik (Base64, Ressource im Ordner <i>Resources</i> oder externe URL)  Wenn die Grafik als Base64-kodierte Zeichenkette hinterlegt wird, dann muss zusätzlich das Attribut <code>mediaType="image/*"</code> angegeben werden.

Attribut	Beschreibung
	Wenn die Grafik als externe URL hinterlegt wird, dann muss zusätzlich das Attribut <code>type="Image"</code> angegeben werden. • Video: Video (Base64, Ressource im Ordner <i>Resources</i> oder externe URL) • Integer: Ganzzahlen • List: Enthält Listenelemente. Wird für ComboBox und ListView verwendet. • String: Zeichenkette • URL: URL  Die XML-Daten, die der Technical Process liefert, müssen zusätzlich das Attribut <code>type</code> und <code>externLinkTarget</code> enthalten.

8.1 Daten aus dem Technical Process

Daten können von einem Technical Process 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 Technical Process 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 Technical Process liefert, haben als Wurzelement entweder `<Ok>` oder `<OkList>`. `<Ok>` wird für Detail Components verwendet, `<OkList>` für List Components und Grid Components.

8.1.1 URL

 Die XML-Daten, die der Technical Process liefert, müssen zusätzlich das Attribut `type` und `externLinkTarget` enthalten.

Attribut	Beschreibung
displayName	Anzeigenname, der statt dem Datensatz angezeigt wird. Mögliche Werte: Beliebige Zeichenkette
type	Legt fest, ob die URL eine interne oder externe Verlinkung ist. Mögliche Werte: • internal • external
externLinkTarget	Legt fest, in welchem Fenster der Datentyp URL geöffnet wird. Mögliche Werte: • same: Die URL wird im selben Fenster geöffnet. • new: Die URL wird in einem neuen Fenster geöffnet.

8.2 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>
```

8.3 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 Component, die Daten aus `detailData.xml` anzeigt:



8.4 Validierung

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

Um Validierungsregeln festzulegen, muss innerhalb des Property-Elements das `Validations`-Element verwendet werden. Das `Validations`-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 [Save Action](#) aktiv ist. Wenn die Validierung nicht bestanden wird, dann ist die Save Action innerhalb der Component nicht aktiv. Components, die ungültige Eingaben enthalten, können also nicht gespeichert werden.

8.4.1 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.

Element	Beschreibung
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.

8.4.2 Mögliche Operatoren

Für Validierungsregeln stehen verschiedene Operatoren zur Verfügung:

Operator	Beschreibung
eq	gleich (<i>equal</i>)
gt	größer als (<i>greater than</i>)
gt_eq	größer als oder gleich (<i>greater than or equal</i>)
lt	kleiner als (<i>less than</i>)
lt_eq	kleiner oder gleich (<i>less than or equal</i>)
neq	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.

8.4.3 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.

8.4.4 Beispiele <Message>

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

Beachten Sie folgende Vorgaben!

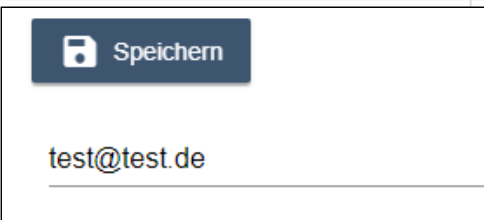
- Wird das `Message`-Element direkt innerhalb des `Validations`-Elements definiert, wird die Nachricht immer angezeigt.
- Wird kein `Message`-Element innerhalb des `Validations`-Elements definiert, wird die definierte Meldung aus dem zutreffenden `Validation`-Elementen ausgegeben.
- Wurde kein `Message`-Element 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>
```

9 Components

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

 Die Vererbung von Properties über die Navigation ist nicht möglich.

Es gibt verschiedene Arten von Components:

Komponente	Tag	Beschreibung
Detail Component	DetailComponent	Zeigt Text an
List Component	ListComponent	Zeigt Daten in Form von dynamisch erzeugten Listen an
Calendar Component	CalendarComponent	Zeigt einen Kalender an
Master/Detail Component	MasterDetailComponent	Kann aus den Strukturelementen Master, Detail und Overlay bestehen. 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.
ExternalWeb Component	ExternalWebComponent	Bettet eine externe Webseite in eine Web App ein
Grid Component	GridComponent	Zeigt Daten in einer tabellarischen Darstellung an

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

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

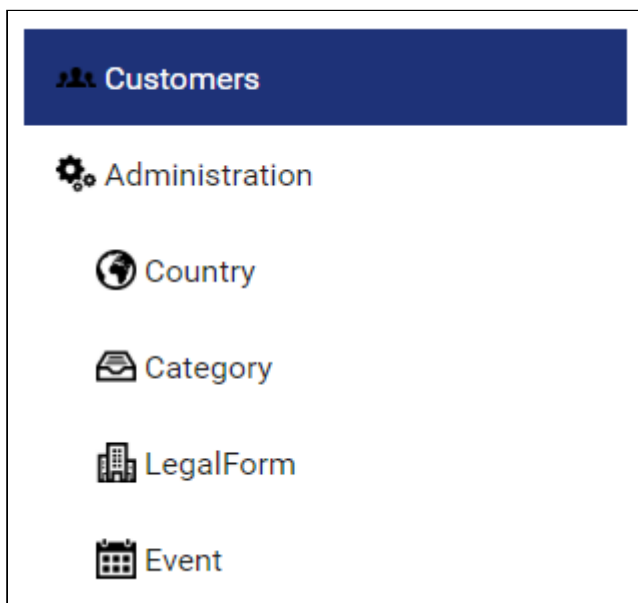
Komponenten können ineinander verschachtelt werden.

```

<WebApp xmlns="http://softproject.de/webapp/1.0">
  <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 Components verschachtelt werden, werden sie im Menü als Unterpunkt angezeigt:



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

9.1 Attribute für Komponenten

Components können folgende Attribute besitzen:

Attribut	Beschreibung
fontFamily	Legt die Schriftfamilie innerhalb der Component fest. Die Eigenschaft wird an alle Controls und Aktionen der Component vererbt. Mögliche Werte: Fontcode aus der Font-Palette, z. B. <i>Font04</i>
fontSize	Legt die Schriftgröße innerhalb der Component fest. Die Eigenschaft wird an alle Controls und Aktionen der Component vererbt. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <i>20</i>; <i>20.8</i>; <i>.9</i> • Schriftgröße in Pixel, z. B. <i>20px</i> • Schriftgröße in Punkt, z. B. <i>18pt</i> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i> • 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>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. Die Eigenschaft wird an alle Controls und Aktionen der Component vererbt. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded
fontStyle	Legt die Neigung der Schrift fest. Die Eigenschaft wird an alle Controls und Aktionen der Components vererbt. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest. Die Eigenschaft wird an alle Controls und Aktionen der Components vererbt. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight
iconColor	Definiert die Farbe des Icons der Component.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. <i>ff5a00</i>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>

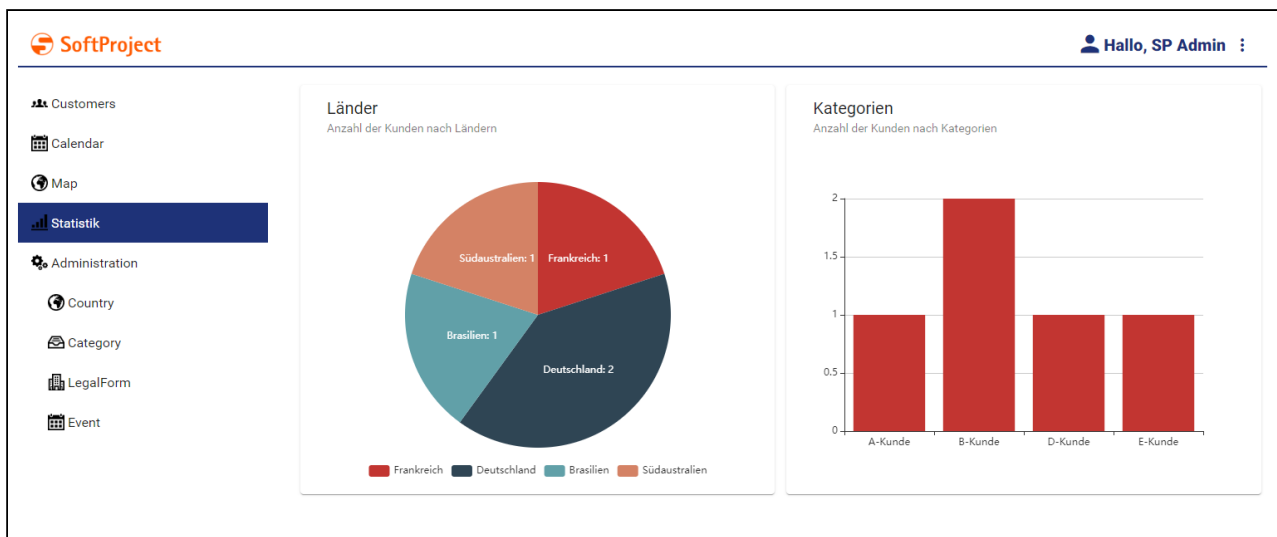
Attribut	Beschreibung
iconURL	Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Icon der Component verwendet wird.  - Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources. - Das Material Icon muss mit dem Präfix icon, z. B. icon:<MaterialIconName> angegeben werden. Mögliche Werte: - Zeichenkette (URI), z. B. <i>clock.png</i> - icon:<MaterialIconName>, z. B. <i>icon:extension</i>  Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Eine weitere Übersicht ist auch online verfügbar. Die Auswahl kann von den tatsächlich verfügbaren Material Icons abweichen.
process	<i>Erforderlich.</i> Pfad zur .wrf-Datei, die die Daten für die Component liefert. Der Technical Process muss im Ordner Services/Processes enthalten sein. Pfadangabe relativ zum Ordner Services/Processes. Mögliche Werte: Zeichenkette (URI)
title	Titel der Komponente. Wird in der Anwendung als Header angezeigt. Mögliche Werte: Beliebige Zeichenkette
titleBackground	Definiert eine Titel-Hintergrundfarbe. Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
titleForeground	Definiert eine Farbe für den Titel-Vordergrund. Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00 ⚠ Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

i Components müssen ein [Layout](#) enthalten.

9.2 Detail Component

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

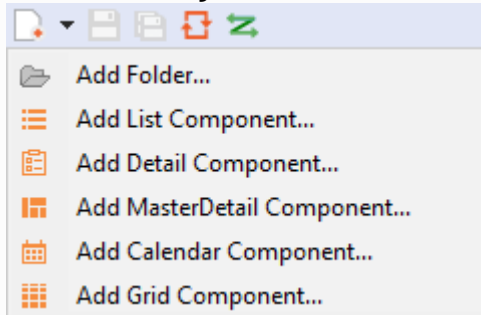


9.2.1 Detail Component definieren

Eine Detail Component 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.



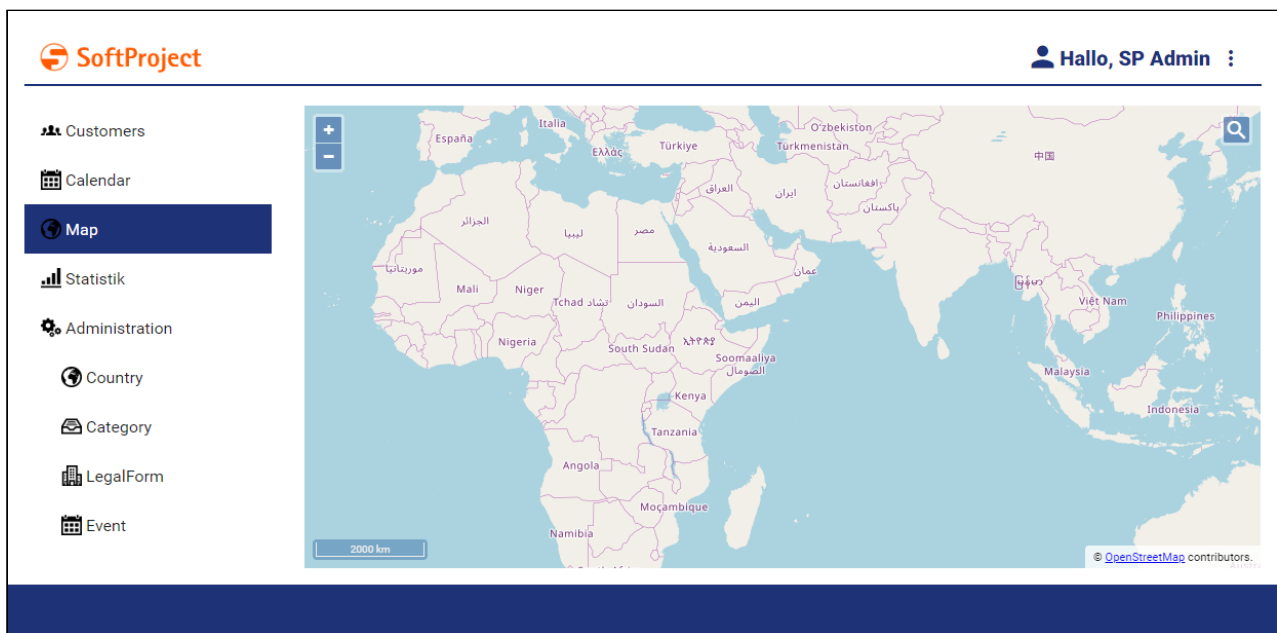
3. In **File name** den Namen der Definitionsdatei eingeben.
4. Auf **Finish** klicken.
Eine neue Definitionsdatei mit einer vordefinierten Struktur wird angelegt.

Die Detail Component wird über `<DetailComponent>` erzeugt und kann die Standard-Attribute für [Components](#) besitzen.

Für eine Detail Component muss ein [Layout](#) deklariert werden.

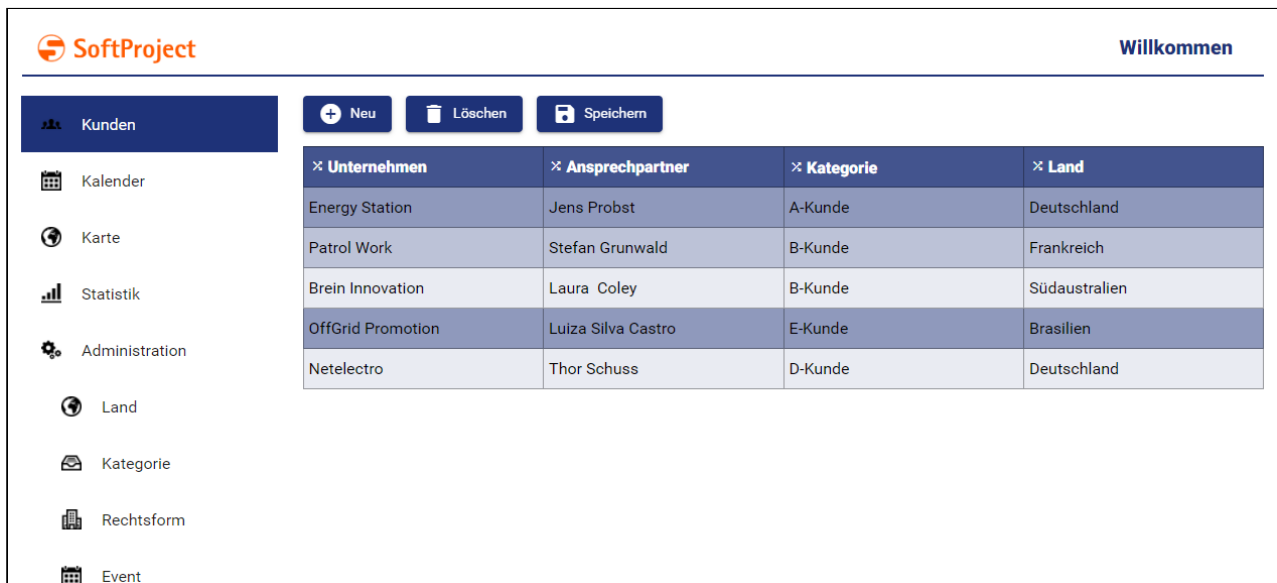
9.2.2 Beispiel Detail Component

```
<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 vendor="OpenStreetMap">
      <Search />
      <Markers data="#Place" iconUrl="Location.png" name="#Company"
searchString="#Address" />
    </Map>
  </FlowLayout>
</DetailComponent>
```



9.3 List Component

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

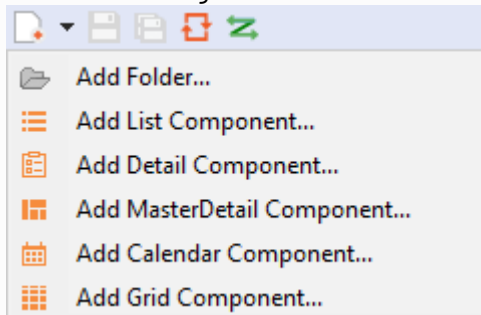


9.3.1 List Component definieren

Eine List Component 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 <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 List Component wird über `<ListComponent>` erzeugt und kann neben den Standard-Attributen für Components folgende zusätzliche Attribute besitzen:

Attribut	Beschreibung
pageSize	Definiert die Anzahl der Elemente, die auf einmal in die Liste geladen werden (vgl. Paging). Mögliche Werte: Ganzzahl größer als 0
rowBackground	Definiert die Hintergrundfarbe der Zeilen in der Liste. Mögliche Werte: • Farbwert aus dem Theme (z. B. A50, A400 oder 09 vgl. Theming) • Hexadezimaler Farbwert ⚠ Wenn im Data-Binding-Ausdruck ein #-Zeichen verwendet werden soll, muss dieses maskiert werden. Als Maskierungszeichen dient der Backslash (\). • Ausdruck für Data Binding (Color)

Attribut	Beschreibung
rowForeground	Definiert eine Farbe für den Vordergrund (Texte usw.). Mögliche Werte: - Farbwert aus dem Theme (z. B. A50, A400 oder 09 vgl. Theming) - Hexadezimaler Farbwert  Wenn im Data-Binding-Ausdruck ein #-Zeichen verwendet werden soll, muss dieses maskiert werden. Als Maskierungszeichen dient der Backslash (\). - Ausdruck für Data Binding (Color)
layout	Definiert den Layout-Modus der Liste. Möglich Werte: fixed: (Standard) Die Spaltenbreite width aller Spalten ist identisch, solange keine Spaltenbreite festgelegt wird. Wenn eine Spaltenbreite festgelegt wird, wird diese Spaltenbreite eingehalten. optimized: Die Spaltenbreite width wird basierend auf dem Tabelleninhalt und den optional gesetzten Attributen dynamisch kalkuliert, um auf dem verfügbaren Platz alle Inhalte anzuzeigen.  Wenn Zeilenumbrüche verhindert werden, kann das bei Bedarf vertikale Scrollbalken erzeugen.
textOverflow	Gibt an, was passieren soll, wenn überlaufende Texte nicht angezeigt werden können. Mögliche Werte: ellipsis: Mit ... zeigen, dass der Text nicht zu Ende ist. Ein Textumbruch wird verhindert. hidden: (Standard, wenn layout="optimized") Text abbrechen, auf ganze Wörter achten. Ein Textumbruch wird verhindert. wordBreak: (Standard, wenn layout="fixed") Innerhalb des Wortes umbrechen allow: Text zwischen Wörtern umbrechen

Attribut	Beschreibung
borderColor	Definiert die Farbe des Rahmens in einer List-Komponente. Mögliche Werte: - Hexadezimaler Farbwert, z. B. <i>ff5a00</i>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>

Die Daten, die angezeigt werden sollen, müssen durch einen Technical Process zur Verfügung gestellt werden und einem bestimmten Datenmodell entsprechen. In den [Properties](#) muss das Datenmodell abgebildet werden, das durch den Technical Process geliefert wird. Für die Verknüpfung der Daten, die durch den Technical Process 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.

9.3.2 Column-Definition

Innerhalb der List Component 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 .../>
  </Columns>
</ListComponent>
```

Mögliche Attribute für `<Column>`:

Attribut	Beschreibung
displayName	Titel der Spalte. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
horizontalAlign	Legt die horizontale Ausrichtung des Inhalts innerhalb der Spalte fest. Mögliche Werte: • left: Inhalt linksbündig • center: Inhalt zentriert • right: Inhalt rechtsbündig
mobilePosition	Legt die Position fest, an der die Spalte in der mobilen Ansicht erscheint. In der mobilen Ansicht können maximal vier Werte angezeigt werden. Mögliche Werte: • topLeft • topRight • bottomLeft • bottomRight i Wenn dieses Attribut nicht definiert wird, wird die Spalte in der mobilen Ansicht nicht angezeigt. 
sortable	Legt fest, ob nach den Inhalten der Spalte sortiert werden kann. Weitere Informationen zu Sortieren. Mögliche Werte: true / false
value	Legt den Inhalt der Spalte per Data Binding fest. Mögliche Werte: Zeichenkette (Data Binding) i Wenn für die entsprechende Property kein Attribut displayName vergeben ist, dann wird als Spaltenüberschrift der Data-Binding-Ausdruck verwendet.

Attribut	Beschreibung
width	Legt die Spaltenbreite fest. Mögliche Werte: • Ganze Zahlen • auto: Die Breite passt sich dem Inhalt an. Kann nur im Layout-Modus optimized verwendet werden.  Die Einheit, in der die Spaltenbreite angegeben wird, wird mit dem Attribut widthUnits festgelegt.
widthUnits	Legt die Einheit fest, in der die Spaltenbreite width angegeben wird. Mögliche Werte: • pixels: Die Spaltenbreite wird in Pixeln angegeben. • percents: Die Spaltenbreite wird in Prozent angegeben.
textOverflow	Gibt an, was passieren soll, wenn überlaufende Texte nicht angezeigt werden können. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist. Ein Textumbruch wird verhindert. • <i>hidden</i>: (Standard, wenn layout="optimized") Text abbrechen, auf ganze Wörter achten. Ein Textumbruch wird verhindert. • <i>wordBreak</i>: (Standard, wenn layout="fixed") Innerhalb des Wortes umbrechen • allow: Text zwischen Wörtern umbrechen  Das Attribut in <Column> hat eine höhere Priorität und überschreibt das Attribut in <ListComponent>.

Mögliche Attribute für <ImageColumn>:

Attribut	Beschreibung
displayName	Titel der Spalte. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette

Attribut	Beschreibung
horizontalAlign	Horizontale Ausrichtung der Grafik in der Spalte Mögliche Werte: • left: Grafik linksbündig • center: Grafik zentriert • right: Grafik rechtsbündig
iconColor	Definiert die Farbe des Bildes. Mögliche Werte: • Hexadezimaler Farbwert, z. B. <i>ff5a00</i>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>
image	Grafik, die in der Spalte angezeigt wird. Mögliche Werte: Ausdruck für Data Binding (Image)
imageHeight	Höhe der Grafik Mögliche Werte: Ganzzahl  Die Einheit, in der die Höhe angegeben wird, wird mit dem Attribut <code>imageUnits</code> festgelegt.
imageUnits	Einheit für die Breiten- und Höhenangabe zur Grafik. Mögliche Werte: • pixels: Pixel • percents: Prozent (Standard)
imageWidth	Breite der Grafik Mögliche Werte: Ganzzahl  Die Einheit, in der die Breite angegeben wird, wird mit dem Attribut <code>imageUnits</code> festgelegt.

Attribut	Beschreibung
mobilePosition	Legt die Position fest, an der die Spalte in der mobilen Ansicht erscheint. In der mobilen Ansicht können maximal vier Werte angezeigt werden. Mögliche Werte: • topLeft • topRight • bottomLeft • bottomRight i Wenn dieses Attribut nicht definiert wird, wird die Spalte in der mobilen Ansicht nicht angezeigt.

9.3.3 Periodisches Neuladen

Über das Element `ReloadTimer` lässt sich die List Component 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	Zeit bis zum erneuten Laden in Sekunden • Data Binding (Integer) möglich Mögliche Werte: Beliebige ganze Zahl
itemIdentifizier	<i>Optional</i>; Eindeutiger Schlüssel für die in den Zeilen dargestellten Daten i • Über das Attribut lassen sich Zeilen beim Neuladen wiederfinden und der Neuladeprozess wird verbessert. • Wird das Attribut nicht angegeben wird ein Stringvergleich auf ein Datensatz verwendet, um die Zeile zu identifizieren. Mögliche Werte: Data Binding an eine definierte Property der List-Komponente

9.3.4 Mobile Ansicht der List Component

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.

⌘ Unternehmen	⌘ Ansprechpartner	⌘ Kategorie	⌘ Land
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

9.3.5 Aktion "Auswählen" innerhalb der List Component

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

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

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

9.3.6 Beispiel List Component

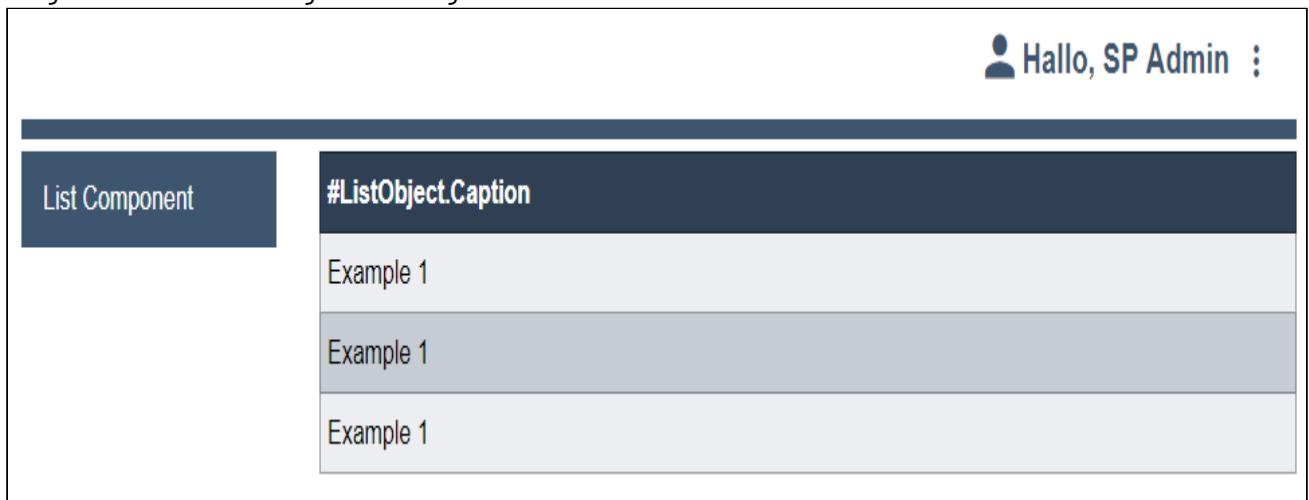
Das folgende Beispiel zeigt die Deklaration einer List Component.

```
<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 Component verarbeitet Daten, die in folgendem Format vorliegen.

```
<?xml version="1.0" encoding="UTF-8" ?>
<OkList size="30">
  <ListObject>
    <Caption>Example 1</Caption>
  </ListObject>
  <ListObject>
    <Caption>Example 1</Caption>
  </ListObject>
  <ListObject>
    <Caption>Example 1</Caption>
  </ListObject>
</OkList>
```

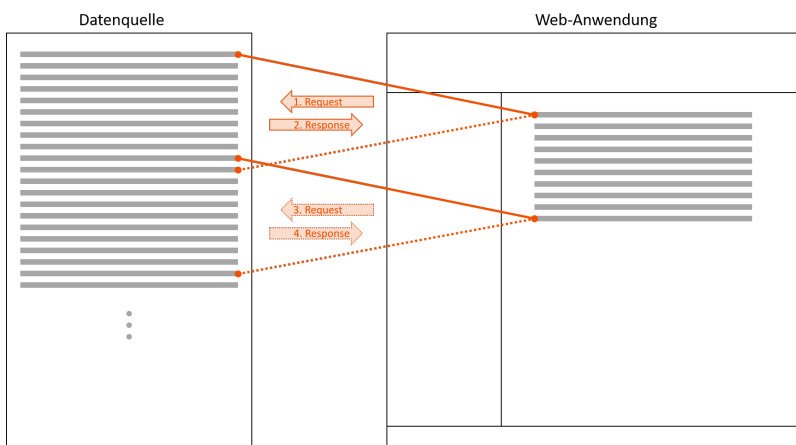
Obiger Code führt zu folgender Ausgabe:



9.3.7 Paging

Web-Anwendungen, die mit X4 Web Apps erstellt werden, unterstützen in der Master/Detail und der List Component automatisch Paging. Deshalb müssen die Prozesse, die für diese Components 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 Web Apps erstellt wurden möglich, die neuen Daten während des Scrollens nachzuladen. Der Benutzer muss nicht explizit auf die nächste Seite umblättern.



9.3.7.1 Beispielanfrage

Die Component, die mit einem Technical Process 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. Mögliche Werte: Ganzzahl
limit	Anzahl der Datensätze, die in der Antwort enthalten sein sollen. Mögliche Werte: Ganzzahl größer als 0  Der Wert dieses Attributs wird automatisch vom Attribut pageSize der List Component hinterlegt!

9.3.7.2 Beispielantwort

Die Antwort, die vom Technical Process 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	Anzahl der insgesamt in der Datenquelle vorhandenen Datensätze. Mögliche Werte: Ganzzahl

9.3.7.3 Beispiel für Paging

Folgendes Beispiel enthält eine List Component mit einem Technical Process 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.

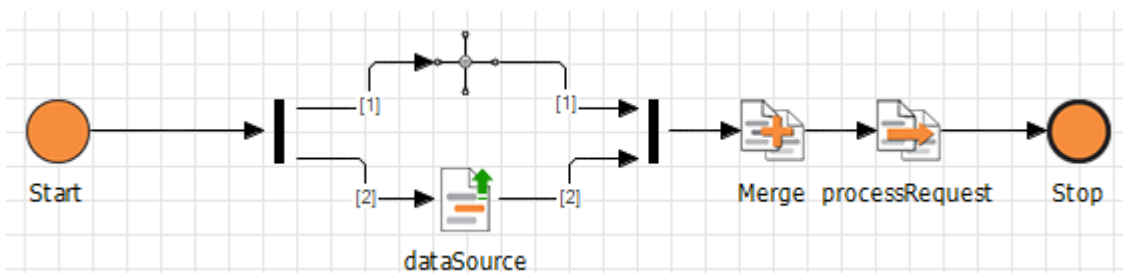
```
<ListComponent name="Paging" path="Paging" displayName="Paging" process="fillListProcess.wrf" default="true">
  <Properties>
    <Property name="List" type="Complex">
      <Property name="Description" type="String" displayName="Description"/>
      <Property name="Id" type="Integer" displayName="Id"/>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



Zunächst werden in diesem Technical Process 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: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>

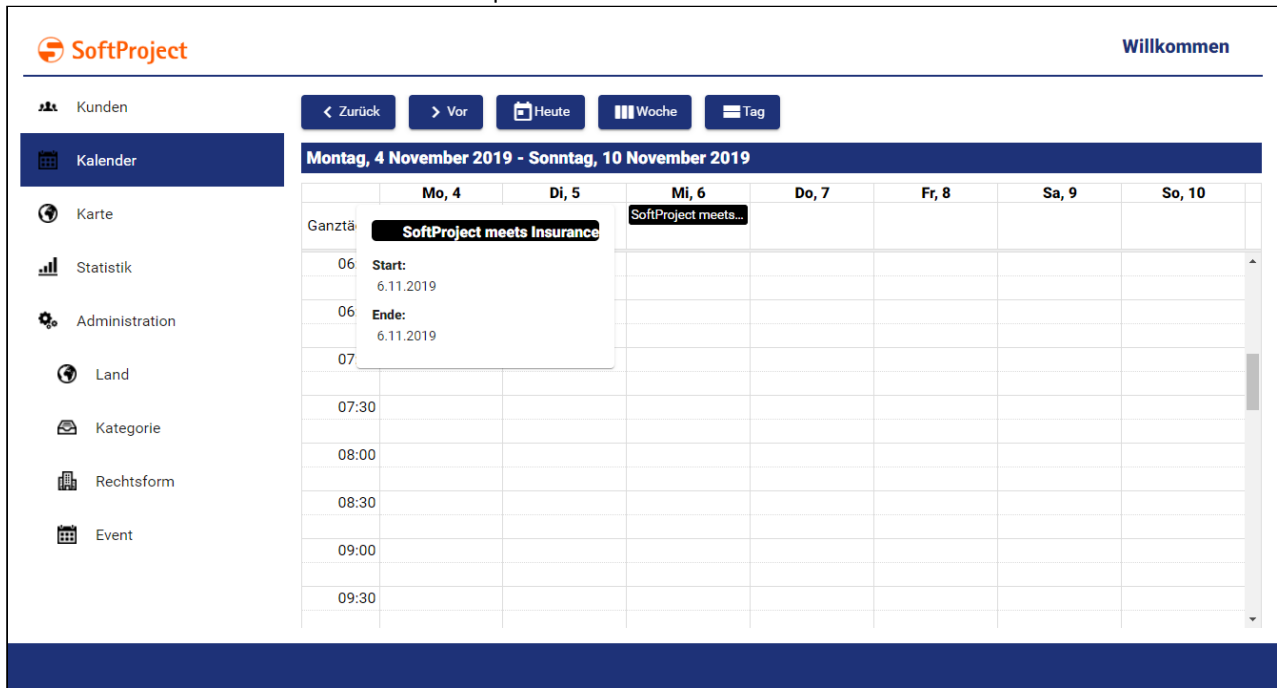
```

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 Component gescrollt wird, weiß die Component, 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 370ab438f96a234a65...
- ☐ Services-Log
- ☐ Materialscore-Regular 370ab438f96a234a65...
- ☐ Materialscore-Regular 370ab438f96a234a65...
- ☐ list?limit=50&offset=50

9.4 Calendar Component

Eine Calendar Component zeigen einen Kalender und Ereignisse. Eine Calendar Component kann alleine oder in einer Master/Detail Components verwendet werden.

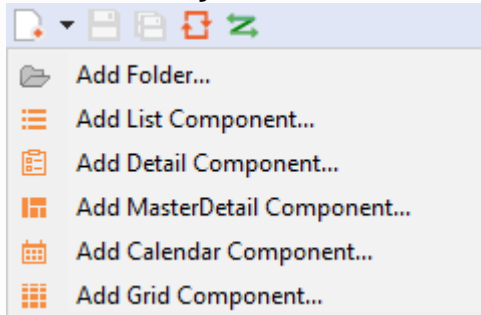


9.4.1 Calendar Component definieren

Eine Calendar Component wird innerhalb einer eigenen Definitionsdatei `<CalendarName>.calendar.d` 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 Calendar Component wird über `<CalendarComponent>` erzeugt und kann die Standard-Attribute für Komponenten besitzen.

9.4.2 Calendar Settings: Ansichten

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

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

Mögliche Attribute:

Attribut	Beschreibung
dayView	Aktiviert und deaktiviert die Tagesansicht. Mögliche Werte: true (Standard) / false
dayViewSlot	Definiert die Zeitskala in der Tagesansicht. Mögliche Werte: • 00:15: 15 Minuten als kleinste Einheit • 00:30: 30 Minuten als kleinste Einheit • 01:00: 1 Stunde als kleinste Einheit

Attribut	Beschreibung
defaultView	Definiert die Standardansicht des Kalenders. Mögliche Werte: • Month: Monatsansicht (Standard) • Week: Wochenansicht • Day: Tagesansicht <i>i</i> Es kann nur eine aktive Ansicht als Standardansicht definiert werden.
monthView	Aktiviert und deaktiviert die Monatsansicht. Mögliche Werte: true (Standard) / false
weekView	Aktiviert und deaktiviert die Wochenansicht. Mögliche Werte: true (Standard) / false
weekViewSlot	Definiert die Zeitskala in der Wochenansicht. Mögliche Werte: • 00:15: 15 Minuten als kleinste Einheit • 00:30: 30 Minuten als kleinste Einheit • 01:00: 1 Stunde als kleinste Einheit
<i>i</i> 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.	

9.4.3 Calendar Settings: Lokalisierung

Eine Calendar Component 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
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

Schlüssel	Wert
september	September
october	Oktober
november	November
december	Dezember
start	Start
end	Ende
location	Ort
description	Beschreibung
allDay	ganztägig

9.4.4 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 Technical Process an die Kalender-Komponente geliefert.

```

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

```

Mögliche Attribute:

Attribut	Beschreibung
allDay	Definiert, ob das Event ganztägig stattfindet. Mögliche Werte: true / false oder Ausdruck für Data Binding (Boolean)
background	Definiert die Farbe, mit der das Event angezeigt werden soll. Mögliche Werte: HTML-Farbcode, z. B. #FF0000 oder Ausdruck für Data Binding (Color)
begin	Definiert den Startzeitpunkt des Events. Mögliche Werte: Ausdruck für Data Binding (DateTime)

Attribut	Beschreibung
description	Gibt die Beschreibung des Events an. Mögliche Werte: Beliebige Zeichenkette oder Ausdruck für Data Binding (String)
detailNavigationLinkName	Name der Schaltfläche, die die Detailansicht öffnet, falls das Attribut externalNavigationLink nicht gesetzt ist. Mögliche Werte: • Beliebige Zeichenkette • leer: Der Standardtext Detail view wird angezeigt i Befindet sich der Kalender innerhalb der Master Component einer Master/Detail Component, 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. Die Schaltfläche wird nur in der mobilen Ansicht angezeigt, wenn sich die Calendar Component innerhalb einer Master/Detail Component befindet.
end	Definiert den Endzeitpunkt des Events. Mögliche Werte: Ausdruck für Data Binding (DateTime)
externalNavigationLink	Link, der geöffnet werden soll, wenn das Event angeklickt wird. Mögliche Werte: Hyperlink (String) oder Ausdruck für Data Binding (String)
externalNavigationLinkName	Text, der bei einem externen Link statt der URL angezeigt wird. Mögliche Werte: Zeichenkette i Wenn bei einem externen Link das Attribut externalNavigationLinkName nicht definiert ist, dann wird die URL angezeigt, die für das Attribut externalNavigationLink definiert wurde.

Attribut	Beschreibung
externalNavigationLinkTarget	Gibt an, ob der Link (externalNavigationLink) im selben oder in einem neuen Tab geöffnet wird. Mögliche Werte: • same: Der Link wird im selben Tab geöffnet • new: Der Link wird in einem neuen Tab geöffnet
fontFamily	Legt die Schriftfamilie fest. Mögliche Werte: Fontcode aus der Font-Palette, z. B. <i>Font04</i>
fontSize	Legt die Schriftgröße fest. Mögliche Werte: • Schriftgröße in Pixel, z. B. <i>20px</i> • Schriftgröße in Punkt, z. B. <i>18pt</i> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i> • Schlüsselwörter, z. B. <i>small</i> oder <i>larger</i>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded
fontStyle	Legt die Neigung der Schrift fest. Mögliche Werte: • italic: kursive Schrift • normal: normale Schrift (Standard) • oblique: schräg gestellter Schriftstil (berechnet)

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight
foreground	Definiert eine Farbe für den Event-Namen. Mögliche Werte: • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
location	Gibt den Ort des Events an. Mögliche Werte: Beliebige Zeichenkette oder Ausdruck für Data Binding (String)
name	Definiert den Namen des Events. Mögliche Werte: Beliebige Zeichenkette oder Ausdruck für Data Binding (String)

9.4.5 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>
```

9.4.6 Daten für die Calendar Component

Die Calendar Component benötigt Daten in einer Liste als Input, um Events anzeigen zu können. Diese Daten werden von einem Technical Process 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!

9.4.7 Beispiel

Die Definition einer Calendar Component 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="#event.begin" end="#event.end" name="#event.name">
    <SelectAction externalLink="#event.link"/>
  </EventConfiguration>
</CalendarComponent>
```

Der Input wird durch den Technical Process 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:



 User Management

 Monitoring

 Scheduler

 Administration



Hallo, SP Admin

Calendar

< Zurück

> Vor

 Heute

 Monat

 Woche

 Tag

Montag, 9 März 2020 - Sonntag, 15 März 2020

	Mo, 9	Di, 10	Mi, 11	Do, 12	Fr, 13	Sa, 14	So, 15
Ganztägig							
09:00							
09:30							
10:00							
10:30					10:30 - Daily Scrum		
11:00					11:00 - 12:00 Review		
11:30							
12:00							
12:30							
13:00					13:00 - 14:00 Lunch		
13:30							
14:00							
14:30							
15:00							
15:30							
16:00							

9.5 Master/Detail Component

Die Master/Detail Component besteht aus [Actions](#) sowie den Strukturelementen [Master](#), [Detail](#) und [Overlay](#).

Element	Beschreibung
Master	Enthält die Component, die im oberen Bildschirmbereich angezeigt wird.
Detail	Enthält eine Component, 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. Components, die entsprechend deklariert werden, werden im Pop-up auf dem Desktop und als separate Seite auf mobilen Geräten angezeigt.

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

The screenshot shows the SoftProject web application. On the left is a sidebar with navigation items: Kunden, Kalender, Karte, Statistik, Administration, Land, Kategorie, Rechtsform, and Event (selected). The main area displays a table of events (Master component). Below the table, the details of the selected event 'BPA Brunch Bonn' are shown (Detail component). An overlay form (Overlay component) is open for editing the event, showing fields for Startzeitpunkt, Endzeitpunkt, Farbe, and Link. At the top of the main area, there are action buttons: 'Neu', 'Löschen', and 'Speichern' (Actions component).

1	Aktionen
2	Master-Strukturelement
3	Separator
4	Detail-Strukturelement



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

9.5.1 Properties in Master/Detail

⚠ Für jede Component kann ein Technical Process zum Laden der Daten definiert werden. Das Datenmodell kann wiederverwendet werden.

Die Properties der Master/Detail Component können sowohl in der Master/Detail Component selbst, als auch in den untergeordneten Components innerhalb der Strukturelemente definiert werden. Aus diesen Properties-Definitionen wird ein gemeinsames Datenmodell für die gesamte Master/Detail Component erzeugt. Das bedeutet auch, dass Konflikte zwischen Properties innerhalb einer Master/Detail Component (z. B. gleicher Name aber unterschiedlicher Datentyp) nicht zulässig sind und zu einem Validierungsfehler führen.

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

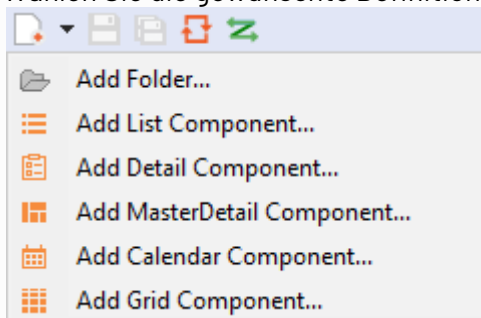
Über ComponentReference eingebundene Components können nur auf Properties zugreifen, die auch innerhalb dieser Component definiert sind. Die Components arbeiten dennoch auf demselben Datenobjekt, haben aber eine eingeschränkte Sicht auf die Properties, die sie tatsächlich benötigen. Insbesondere bei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

9.5.2 Master/Detail Component definieren

Eine Master/Detail Component wird innerhalb einer eigenen Definitionsdatei `<MasterDetailComponentName>.masterdetail` 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 Master/Detail Component wird über `<MasterDetailComponent>` erzeugt und kann die Standard-Attributen für Components besitzen.

Attribut	Beschreibung
defaultSeparatorPosition	Horizontale Standardposition des Separators in Prozent Mögliche Werte: • 30 - 70

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

```
<MasterDetailComponent xmlns="http://softproject.de/webapp/1.0"
defaultSeparatorPosition="60">
  <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 Component**. 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.

9.5.3 Component aus- und einblenden

Bereiche einer Master/Detail Component (Master-Strukturelement, Detail-Strukturelement und Overlay) können per Data Binding dynamisch aus- und eingeblendet werden. Das kann über das Attribut `visible` gesteuert werden:

Komponente mit dynamischen Aus-/Einblenden

```
<CalendarComponent displayName="CalendarComp with visible attribute" visible="#visibility">
```

Komponentenreferenz mit dynamischen Aus-/Einblenden

```
<ComponentReference source="detail_1.detail" displayName="CompRef with visible attribute" visible="#visibility"/>
```

✓ Weitere Informationen finden Sie im Abschnitt [Navigation innerhalb einer Web-Anwendung](#).

9.5.4 Master-Strukturelement

Im Master-Strukturelement wird festgelegt, welche Informationen mithilfe welcher Component 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 Component definiert. Das Master-Strukturelement muss eine List Component 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>
```

❗ Der Technical Process, der die Daten der Master/Detail Component verarbeitet, wird als Attribut process zur List Component angegeben. Dabei muss der Technical Process im Ordner Services/Processes enthalten sein. Die Pfadangabe erfolgt relativ zum Ordner Services/Processes.

9.5.5 Detail-Strukturelement

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

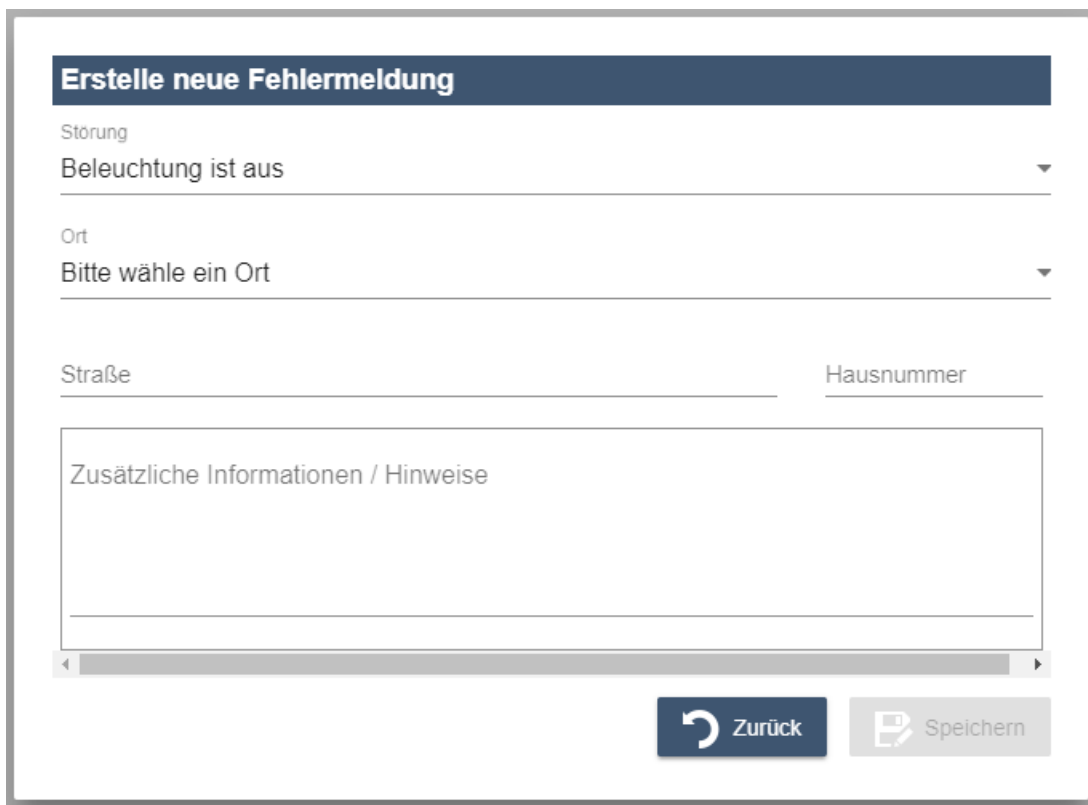
Das Detail-Strukturelement wird innerhalb von `MasterDetailComponent` nach dem Master-Strukturelement definiert. Die einzelnen Components werden hier referenziert.

Beispiel für die Definition eines Detail-Strukturelements

```
<MasterDetailComponent xmlns="http://softproject.de/webapp/1.0">
  <Actions/>
  <Master/>
  <Detail>
    <ComponentReference source="Administration/Country/Country.detail" name="CountryDetail" displayName="$General"/>
  </Detail>
  <Overlay/>
</MasterDetailComponent>
```

9.5.6 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 Component, 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 angezeigt, wenn es über eine Aktion aufgerufen wird.

Innerhalb der `MasterDetailComponent` lassen sich beliebig viele Overlay-Strukturelemente anlegen, die mit folgenden Attributen definiert werden können:

Attribut	Beschreibung
<code>width</code>	Breite des Overlays i Die Angabe des <code>width</code>-Attributs ist optional. Wird das Attribut nicht angegeben, wird die Standard-Breite angewendet. Mögliche Werte: - Ganzzahl > 0 bei Pixelangaben - Ganzzahl zwischen 0 und 100 bei Prozentangaben

Attribut	Beschreibung
height	Höhe des Overlays i Die Angabe des height-Attributs ist optional. Wird das Attribut nicht angegeben, wird die Standard-Höhe angewendet. Mögliche Werte: • Ganzzahl > 0 bei Pixelangaben • Ganzzahl zwischen 0 und 100 bei Prozentangaben
units	Gibt die Einheit an, die für Größenangaben gilt. Mögliche Werte: • pixels • percents (Standard)



- Werden die Attribute width und height mit unzulässigen Werten bei Prozentangabe gefüllt, wird ein Validierungsfehler ausgegeben.
- Werden die Attribute width und height nicht gesetzt oder mit 0 angegeben, werden Overlays in der Standardgröße angezeigt.



Mit dem Attribut enableOverlayBlurredBackground innerhalb der [Web-App-Definition](#) (.wad) kann angegeben werden, ob bei Overlays der Teil der Web App, der noch im Hintergrund zu sehen ist, verschwommen dargestellt werden soll oder nicht.

Beispiel-Definition

```
<MasterDetailComponent xmlns="http://softproject.de/webapp/1.0">
  <Actions/>
  <Master/>
  <Detail/>
  <Overlay width="100" height="100" units="percents">
    <ComponentReference source="Customers/AddToEvent.list" name="AddToEvent"></
ComponentReference>
  </Overlay>
  <Overlay width="200" height="300" units="pixels">
    <ComponentReference source="Customers/AddToEvent.list" name="DifferentEvent">
</ComponentReference>
  </Overlay>
</MasterDetailComponent>
```

9.6 ExternalWeb Component

Eine ExternalWeb Component bettet eine externe Webseite in eine Web App ein. Eine ExternalWeb Component wird mit dem Element `<ExternalWebComponent>` erzeugt.



- Wenn der Eigentümer der Webseite das Einbetten untersagt, z. B. durch Einbinden eines entsprechenden HTTP-Antwort-Headers, kann diese Webseite nicht in eine Web App eingebettet werden.
- Beachten Sie, dass eingebettete Webseiten Cookies auf dem Endgerät speichern und verwenden können.

Für das Element `<ExternalWebComponent>` lassen sich folgende Attribute festlegen:

Attribut	Beschreibung
displayName	Anzeigenname der Component im Menü. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
path	<i>Erforderlich.</i> Eindeutige URL der Component, sichtbar in der Adresszeile des Browsers. Mögliche Werte: Zeichenkette aus alphanumerischen Zeichen (keine Umlaute, Punkte usw.)
allowForms	Erlaubt eingebetteten Inhalten Formulare zu übermitteln. Mögliche Werte: • true • false (Standard)
allowModals	Erlaubt eingebetteten Inhalten einen modalen Dialog anzuzeigen. Mögliche Werte: • true • false (Standard)
allowPointerLock	Erlaubt eingebetteten Inhalten Mausbewegungen direkt als Eingabemethode zu interpretieren. Mögliche Werte: • true • false (Standard)

Attribut	Beschreibung
allowPopups	Erlaubt eingebetteten Inhalten eine Webseite in einem neuen Fenster oder Registerkarte zu öffnen. Mögliche Werte: • true • false(Standard)
allowSameOrigin	Erlaubt eingebetteten Inhalten gemeinsamen Speicherplatz zu verwenden, wenn der Ursprung der eingebetteten Inhalte derselbe ist wie der Ursprung der Host-Webanwendung. Mögliche Werte: • true • false(Standard)
allowScripts	Erlaubt eingebetteten Inhalten JavaScript auszuführen. Mögliche Werte: • true • false(Standard)
allowTopNavigation	Erlaubt eingebetteten Inhalten eine Webseite im selben Fenster oder Registerkarte zu öffnen. Mögliche Werte: • true • false(Standard)
process	Repository-Pfad zu einem Technical Process Mögliche Werte: Beliebiger Pfad im X4 Repository ohne führenden Schrägstrich (z.B. <i>Project/Folder/Process.wrf</i>)
url	Uniform Resource Locator (URL) mit beliebigem gültigen Protokoll (http://, https://, etc.) Mögliche Werte: Beliebige gültige URL (z.B. https://www.softproject.de/)
name	Eindeutiger Name, der die Component referenziert. Mögliche Werte: Beliebige Zeichenkette
default	Bestimmt, ob die Ansicht auf eine Component als Voreinstellung ausgewählt ist. Mögliche Werte: true / false (maximal eine Component darf das Attribut default="true" besitzen)

Attribut	Beschreibung
showInMenu	Bestimmt, ob die ExternalWeb Component in der Navigationsleiste sichtbar ist. Mögliche Werte: true / false

Input

Wenn der Parameter process gesetzt wurde, wird eine XML-Datei mit der URL als Input in folgender Struktur erwartet:

```
<?xml version="1.0" encoding="UTF-8"?>
<Ok
  url="URL"/>
```

9.6.1 Beispiel

9.6.1.1 URL in ExternalWeb

Das folgende Beispiel in der .wad erzeugt eine ExternalWeb Component:

```
<ExternalWebComponent
  path="ExternalWebUrl"
  displayName="SoftProject Website"
  url="https://www.softproject.de" />
```

9.6.1.2 URL über Technical Process

Das folgende Beispiel in der .wad erzeugt eine ExternalWeb Component:

```
<ExternalWebComponent
  path="ExternalWebUrl"
  displayName="SoftProject Website"
  process="provideURL.wrf" />
```

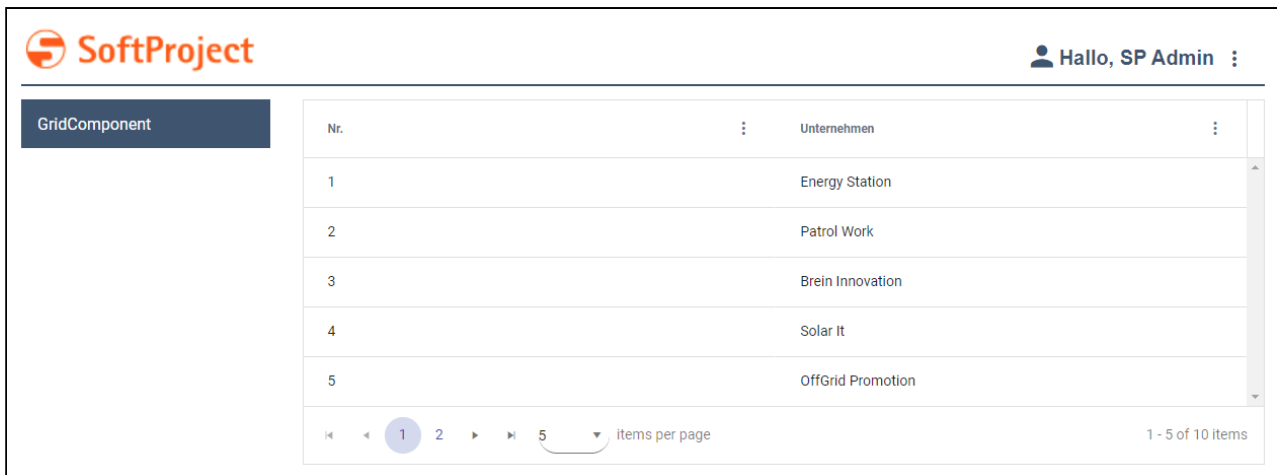
Der Technical Process provideURL.wrf liefert folgenden Output, der die URL festsetzt:

```
<?xml version="1.0" encoding="UTF-8"?>
<Ok
  url="https://www.softproject.de"/>
```

9.7 Grid-Komponente

Eine Grid Component eignet sich ideal, um Daten in einer tabellarischen Darstellung abzubilden. In den Grid Settings kann durch das Setzen einzelner Elemente das Verhalten der Grid Component festgelegt werden.

 Die Grid Component wird in einer zukünftigen Version der X4 BPMS für die Anzeige auf mobilen Geräten optimiert.



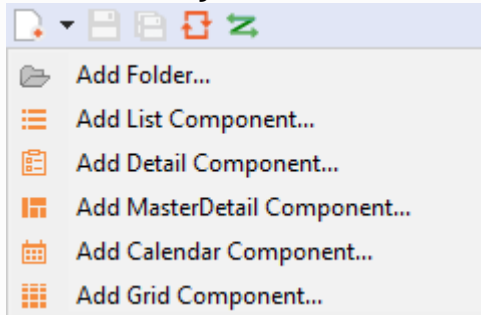
Nr.	Unternehmen
1	Energy Station
2	Patrol Work
3	Brein Innovation
4	Solar It
5	OffGrid Promotion

9.7.1 Grid Component definieren

Eine Grid Component wird innerhalb einer eigenen Definitionsdatei `<GridName>.grid` deklariert. Diese wird im Projektordner **Components** angelegt.

So legen Sie eine neue Definitionsdatei an:

1. Klicken Sie auf das Menü **New > Add Grid Component**.
Alternativ lässt sich die Datei auch über das Kontextmenü **New > Add Grid Component** 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 Daten, die angezeigt werden sollen, müssen durch einen Technical Process zur Verfügung gestellt werden und einem bestimmten Datenmodell entsprechen.

In den [Properties](#) muss das Datenmodell abgebildet werden, das durch den Technical Process geliefert wird. Für die Verknüpfung der Daten, die durch den Technical Process geliefert werden, mit der Darstellung innerhalb der Webanwendung, muss die entsprechende Property definiert und per [Data Binding](#) in der GridColumn-Definition hinterlegt werden.

- Wenn eine Property in der Grid Component verwendet werden soll, muss sie als *Complex*-Typ definiert werden. Diese Property kann weitere Properties enthalten.**

Die Grid Component wird über `<GridComponent>` erzeugt und kann folgende Attribute besitzen:

Attribut	Beschreibung
background	Definiert eine Hintergrundfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas. Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
displayName	Anzeigename der Component im Menü. - Übersetzbar - Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
fontFamily	Legt die Schriftfamilie innerhalb der Component fest. Die Eigenschaft wird an alle Controls und Actions der Component vererbt. Mögliche Werte: Fontcode aus der Font-Palette, z. B. <i>Font04</i>
fontSize	Legt die Schriftgröße innerhalb der Component fest. Die Eigenschaft wird an alle Controls und Actions der Component vererbt. Mögliche Werte: - Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. 20; 20.8; .9 - Schriftgröße in Pixel, z. B. 20px - Schriftgröße in Punkt, z. B. 18pt - Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. .8em oder 120% - Schlüsselwörter: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. Die Eigenschaft wird an alle Controls und Actions der Component vererbt. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded
fontStyle	Legt die Neigung der Schrift fest. Die Eigenschaft wird an alle Controls und Actions der Component vererbt. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)
fontWeight	Legt die Schriftstärke fest. Die Eigenschaft wird an alle Controls und Actions der Component vererbt. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight

Attribut	Beschreibung
foreground	Definiert die Schriftfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas. Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
headerBackground	Definiert die Hintergrundfarbe der Kopfzeile  Diese Einstellung überschreibt die Standardfarbe des Farbschemas. Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
headerFontFamily	Legt die Schriftfamilie der Kopfzeile fest. Mögliche Werte: Fontcode aus der Font-Palette, z. B. <i>Font04</i>

Attribut	Beschreibung
headerFontWeight	Legt die Schriftstärke der Kopfzeile fest. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight
headerForeground	Definiert die Schriftfarbe der Kopfzeile.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas. Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  • Verwenden Sie keine Raute vor dem Farbwert. • Verwenden Sie keine verkürzte Schreibweise des Farbwerts. • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
iconColor	Definiert die Farbe des Icons der Component.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas. Mögliche Werte: - Hexadezimaler Farbwert, z. B. <i>ff5a00</i>  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>
iconUrl	Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Icon der Component verwendet wird.  - Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources. - Das Material Icon muss mit dem Präfix <i>icon</i>, z. B. <i>icon:<MaterialIconName></i> angegeben werden. Mögliche Werte: - Zeichenkette (URI), z. B. <i>clock.png</i> <i>icon:<MaterialIconName></i>, z. B. <i>icon:extension</i>  Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren Material Icons abweichen.
itemsPerPage	Definiert die Anzahl der pro Seite angezeigten Reihen. Mögliche Werte: 5 (default) 10 20 50 100
name	Eindeutiger Name, der die Component referenziert. Mögliche Werte: Beliebige Zeichenkette

Attribut	Beschreibung
rowHighlightBackground	Definiert die Hintergrundfarbe der gewählten Zeile.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas. Mögliche Werte: - Hexadezimaler Farbwert, z. B. <code>ff5a00</code>  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. Hexadezimaler Farbwert, z. B. <code>ff5a00</code> - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <code>A200</code>
rowHighlightForeground	Definiert die Schriftfarbe der gewählten Zeile.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas. Mögliche Werte: - Hexadezimaler Farbwert, z. B. <code>ff5a00</code>  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. Hexadezimaler Farbwert, z. B. <code>ff5a00</code> - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <code>A200</code>
process	<i>Erforderlich.</i> Pfad zur <code>.wrf</code>-Datei, die die Daten für die Component liefert. Der Prozess muss im Ordner <code>Services/Processes</code> enthalten sein. Pfadangabe relativ zum Ordner <code>Services/Processes</code>. Mögliche Werte: Zeichenkette (URI)
title	Titel der Component. Wird in der Anwendung als Header angezeigt. Mögliche Werte: Beliebige Zeichenkette

Attribut	Beschreibung
titleBackground	Definiert eine Titel-Hintergrundfarbe. Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  • Verwenden Sie keine Raute vor dem Farbwert. • Verwenden Sie keine verkürzte Schreibweise des Farbwerts. • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
titleForeground	Definiert eine Farbe für den Titel-Vordergrund. Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  • Verwenden Sie keine Raute vor dem Farbwert. • Verwenden Sie keine verkürzte Schreibweise des Farbwerts. • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
visible	Legt die Sichtbarkeit fest. • Data Binding (Boolean) möglich. Mögliche Werte: Boolean oder Zeichenkette für Data Binding

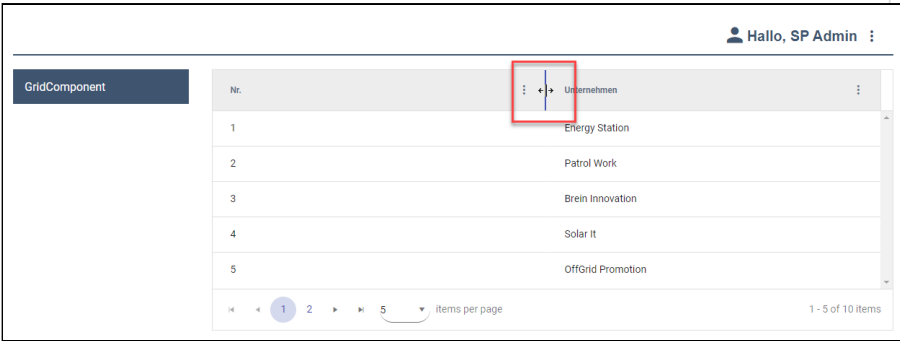
9.7.2 Beispiel einer Grid-Komponente

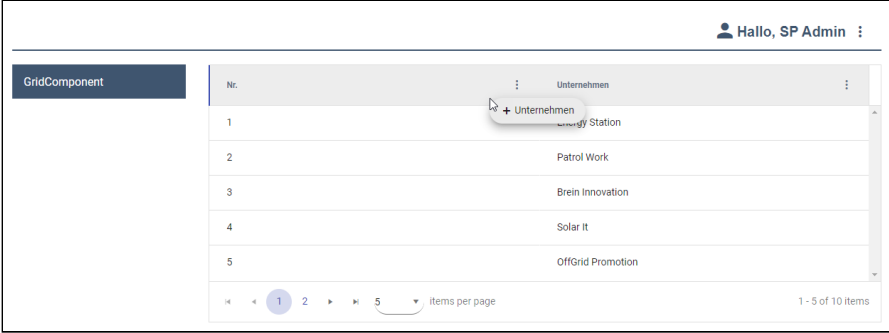
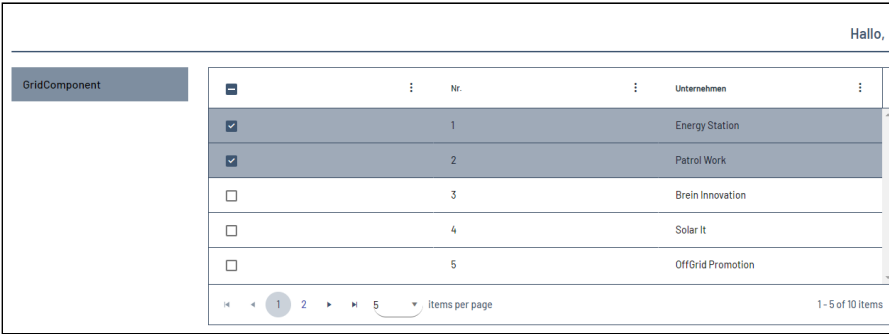
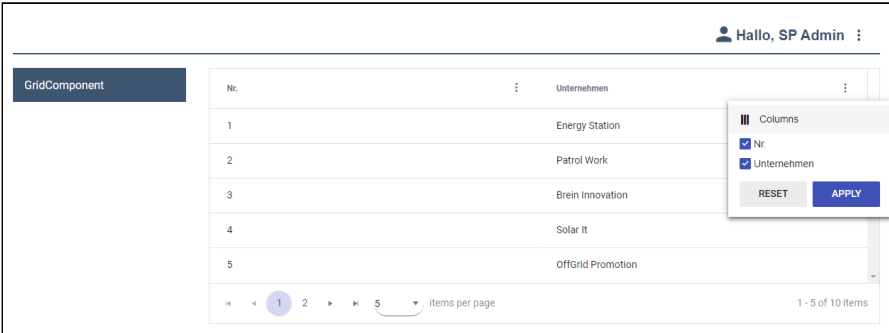
```
<?xml version="1.0" encoding="UTF-8"?>
<GridComponent xmlns="http://softproject.de/webapp/1.0" process="loadGrid.wrf" >
  <Properties>
    <Property name="list" type="Complex">
      <Property name="id" type="Integer" displayName="Nr." />
      <Property name="company" type="String" displayName="Unternehmen" />
    </Property>
  </Properties>
  <GridSettings>
    <Resize/>
    <Reorder/>
    <ColumnVisibility/>
    <Filtering/>
    <Multiselect checkboxSelection="true"/>
    <Autoselection/>
  </GridSettings>
  <GridColumns>
    <GridColumn value="#list.id" sortable="true"/>
    <GridColumn value="#list.company"/>
  </GridColumns>
</GridComponent>
```

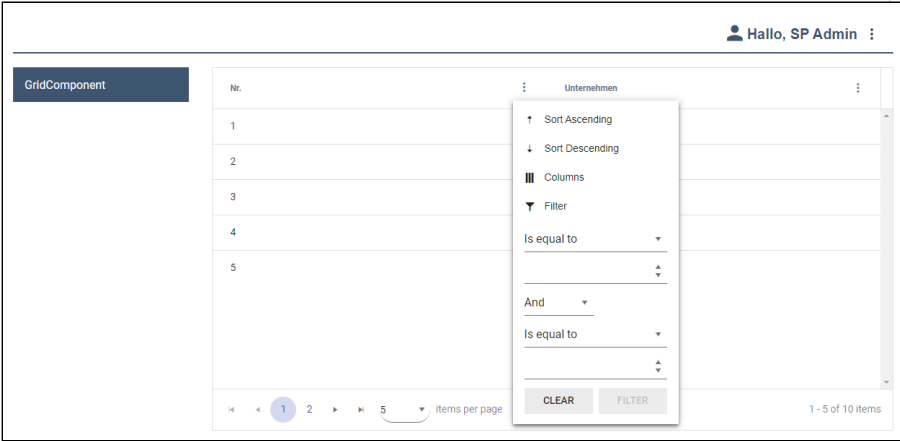
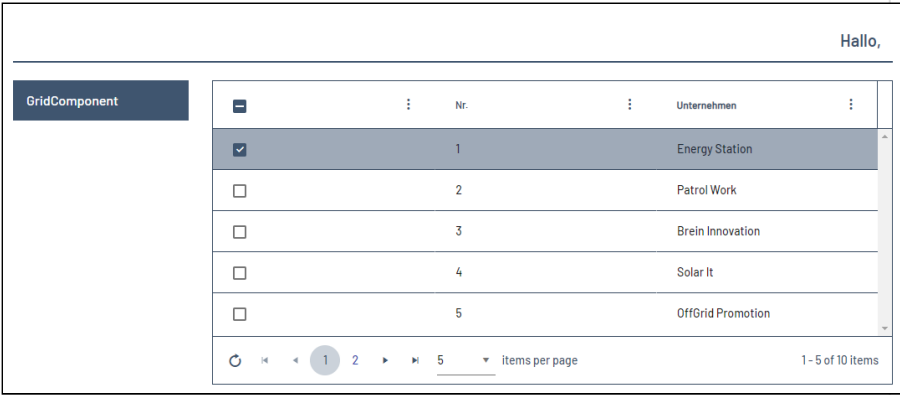
9.7.3 Grid Settings

Innerhalb des Elements `GridSettings` können die unten aufgelisteten Elemente deklariert werden, um das Verhalten der Grid Component festzulegen:

Mögliche Elemente

Element	Beschreibung
Resize	Mit dem Element <code>Resize</code> können die Übergänge der Spaltenüberschriften angeklickt und die Breite der jeweiligen Spalten mit gedrückter Maustaste verändert werden. 

Element	Beschreibung
Reorder	Mit dem Element Reorder können die Spaltenüberschriften angeklickt und per Drag and Drop an eine andere Position verschoben werden. 
Multiselect	Mit dem Element Multiselect wird ermöglicht, dass mehrere Einträge gewählt werden können. Bei Verwendung des Elements Multiselect, kann das Attribut <code>checkboxSelection</code> gesetzt werden und folgende Werte enthalten: Mögliche Werte: <code>true</code> / <code>false</code> 
ColumnVisibility	Mit dem Element ColumnVisibility können einzelne Spalten ein- und ausgeblendet werden. 

Element	Beschreibung
Filtering	Mit dem Element Filtering können Bedingungen festgelegt werden, nach denen gefiltert werden soll. 
Autoselection	Das Element Autoselection legt fest, dass die erste Zeile der Tabelle automatisch ausgewählt wird, wenn die Grid Component geöffnet wird. 
Editing	Das Element Editing hat nur eine Funktion, wenn im Element GridColumn das Attribut <code>editable="true"</code> gesetzt wurde. Bei Verwendung des Elements Editing kann das Attribut <code>reloadOnSave</code> gesetzt werden und folgende Werte enthalten: Mögliche Werte: <code>true</code> : Beim Speichern mithilfe einer Save Action wird die Grid Component neu geladen. <code>false</code> : Beim Speichern mithilfe einer Save Action wird die Grid Component nicht neu geladen.

9.7.4 GridColumn-Definition

Innerhalb der Grid Component müssen die Spalten definiert werden. Die Spaltendefinition erfolgt über `GridColumn`. Alle Spalten müssen innerhalb von `GridColumn`s definiert werden.

Mögliche Attribute

Attribut	Beschreibung
<code>horizontalAlign</code>	Legt die horizontale Ausrichtung des Inhalts innerhalb der Spalte fest. Mögliche Werte: • <code>left</code>: Inhalt linksbündig • <code>center</code>: Inhalt zentriert • <code>right</code>: Inhalt rechtsbündig
<code>format</code>	Legt die Anzeige des Datumsformats mit Minuten, Sekunden oder Millisekunden fest. Mögliche Werte: • <code>Datetime with Minutes</code>: Das Datum wird im Format <code>hh:mm</code> angegeben. (Standard) • <code>Datetime with Seconds</code>: Das Datum wird im Format <code>hh:mm:ss</code> angegeben. • <code>Datetime with Milliseconds</code>: Das Datum wird im Format <code>hh:mm:ss.sss</code> angegeben.
<code>sortable</code>	Legt fest, ob nach den Inhalten der Spalte sortiert werden kann. Weitere Informationen zu Sortieren. Mögliche Werte: <code>true</code> / <code>false</code>
<code>editable</code>	Legt fest, ob die Inhalte der Spalte editierbar sind. Mögliche Werte: <code>true</code> / <code>false</code> (Standard)
<code>value</code>	Legt den Inhalt der Spalte per Data Binding fest. Mögliche Werte: Zeichenkette (Data Binding)  Wenn für die entsprechende Property kein Attribut <code>displayName</code> vergeben ist, dann wird als Spaltenüberschrift der Data-Binding-Ausdruck verwendet.

Attribut	Beschreibung
width	Legt die Spaltenbreite fest. Mögliche Werte: • Ganze Zahlen  Die Einheit, in der die Spaltenbreite angegeben wird, wird mit dem Attribut widthUnits festgelegt.
widthUnits	Legt die Einheit fest, in der die Spaltenbreite width angegeben wird. Mögliche Werte: • pixels: Die Spaltenbreite wird in Pixeln angegeben. • percents: Die Spaltenbreite wird in Prozent angegeben.
textOverflow	Gibt an, was passieren soll, wenn überlaufende Texte nicht angezeigt werden können. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist. Ein Textumbruch wird verhindert. • <i>hidden</i>: (Standard, wenn layout="optimized") Text abbrechen, auf ganze Wörter achten. Ein Textumbruch wird verhindert. • <i>wordBreak</i>: (Standard, wenn layout="fixed") Innerhalb des Wortes umbrechen • <i>allow</i>: Text zwischen Wörtern umbrechen Das Attribut in <GridColumn> hat eine höhere Priorität und überschreibt das Attribut in <GridComponent>.

9.7.5 Periodisches Neuladen

Über das Element `ReloadTimer` lässt sich die Grid Component periodisch neu laden.



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

Mögliche Attribute

Attribut	Beschreibung
refreshTime	Zeit bis zum erneuten Laden in Sekunden • Data Binding (Integer) möglich Mögliche Werte: Beliebige ganze Zahl

Attribut	Beschreibung
itemIdentifizier	<i>Optional</i>; Eindeutiger Schlüssel für die in den Zeilen dargestellten Daten  - Über das Attribut lassen sich Zeilen beim Neuladen wiederfinden und der Neuladeprozess wird verbessert. - Wird das Attribut nicht angegeben wird ein Stringvergleich auf ein Datensatz verwendet, um die Zeile zu identifizieren. Mögliche Werte: Data Binding an eine definierte Property der Grid Component

9.7.6 "Select" Action innerhalb der Grid Component

Die **Select** Action wird über das Element `<SelectAction/>` innerhalb von `<Actions>` erzeugt und ausgelöst, sobald ein Eintrag aus einer Grid Component ausgewählt wurde.

9.7.7 "Save" Action innerhalb der Grid Component

 Wenn die **Save** Action definiert ist, erhält der Technical Process die Datenobjekte der bearbeiteten Zeilen mit einem neuen Indexattribut, das den Index einer Zeile enthält. Die Nummerierung beginnt bei 0.

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

9.7.8 Paging

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

9.7.8.1 Clientseitiges Paging

⚠ Eine hohe Anzahl an darzustellenden Datensätzen erfordert eine hohe Rechenleistung. Wenn mehr als 10.000 Datensätze dargestellt werden sollen, wird empfohlen, auf das serverseitige Paging umzustellen.

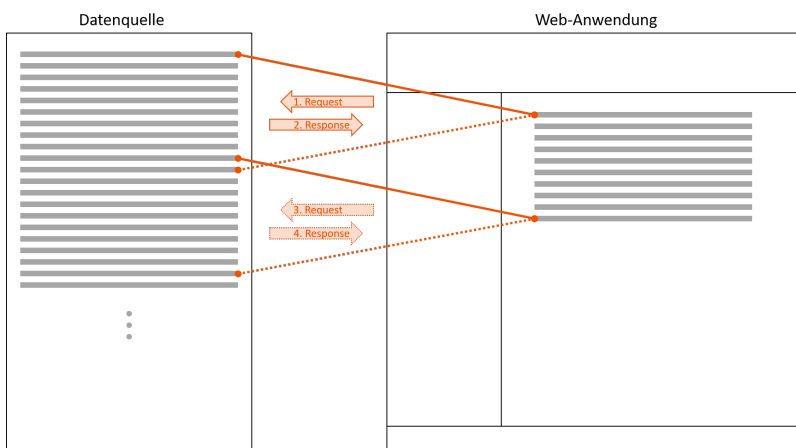
Beim clientseitigem Paging übernimmt der Browser das Paging. Das entspricht dem Standardverhalten der Grid Component.

Die Daten müssen vom Technical Process in folgender Form geliefert werden:

```
<OkList>
  <List>
    ...
  </List>
</OkList>
```

9.7.8.2 Serverseitiges Paging

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 Web Apps erstellt wurden möglich, die neuen Daten während des Scrollens nachzuladen. Der Benutzer muss nicht explizit auf die nächste Seite umblättern.



Die Antwort, die vom Technical Process 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.

Die Daten müssen vom Technical Process in folgender Form geliefert werden:

```
<OkList size="">
  <List>
    ...
  </List>
</OkList>
```

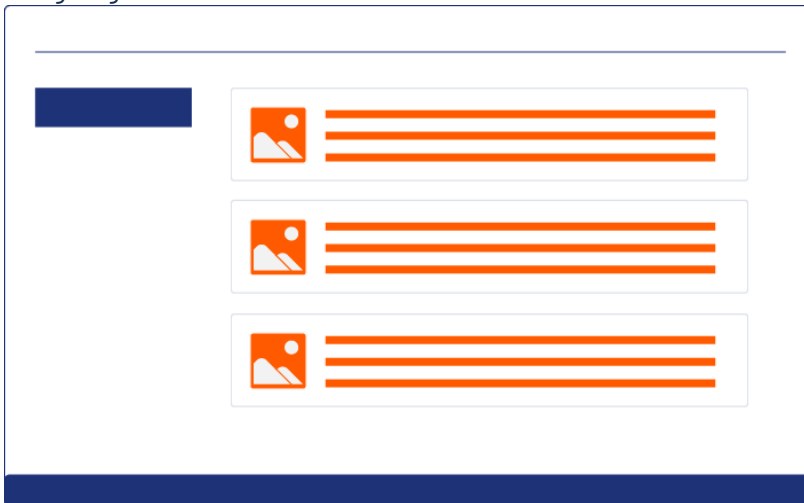
Attribut	Beschreibung
size	Anzahl der insgesamt in der Datenquelle vorhandenen Datensätze. Mögliche Werte: Ganzzahl

10 Layouts

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

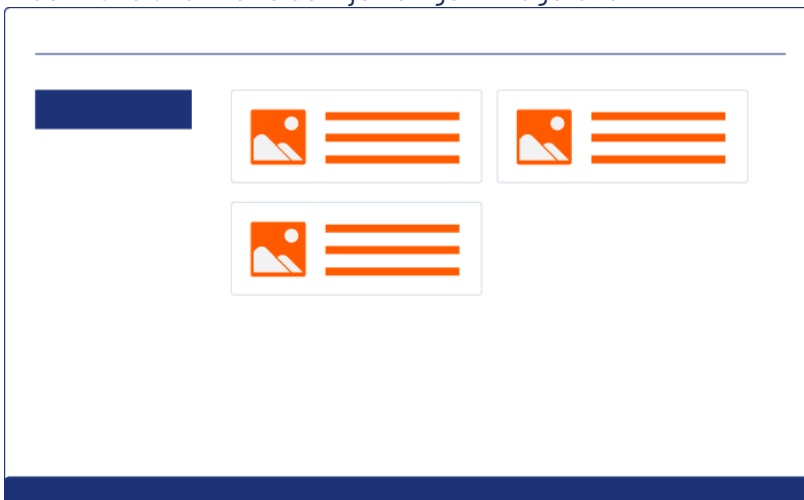
- Flow-Layout

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.

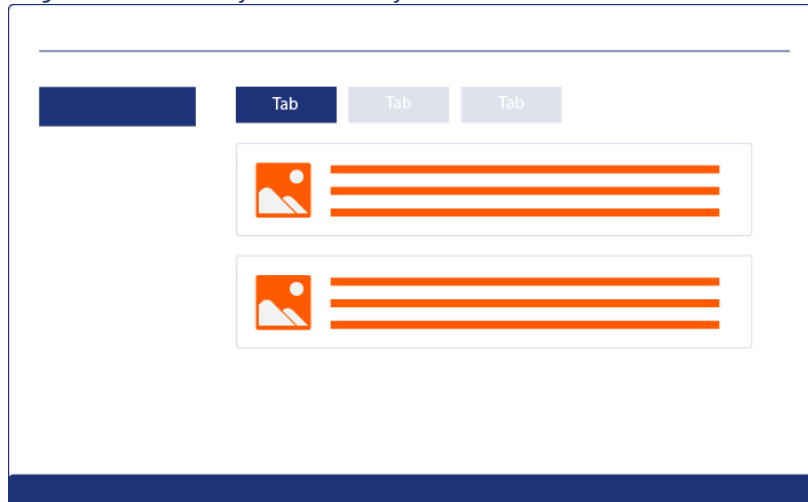


- Grid-Layout

Wenn für die Elemente einer Web App 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.

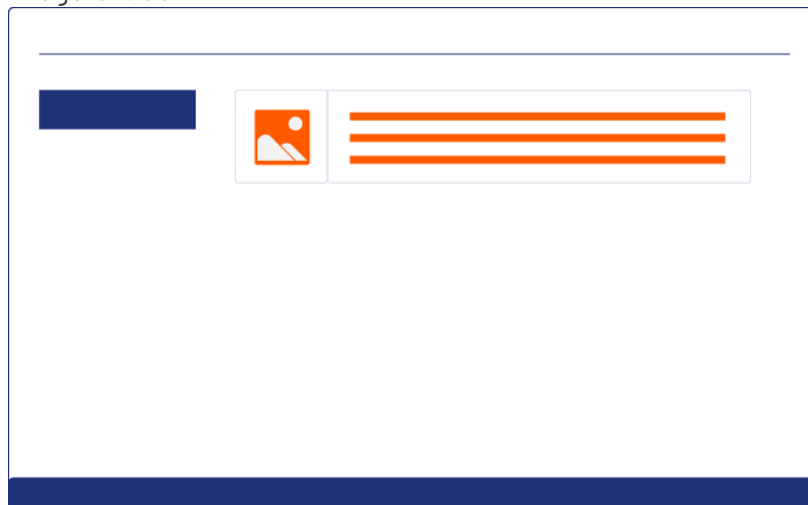


- Registerkarten-Layout (Tab Layout)



- Responsive-Layout

Das Responsive-Layout bietet sich für Web-Anwendungen an, die auf unterschiedlichen Endgeräten angezeigt werden sollen. Das Responsive-Layout ist in 12 virtuelle Spalten unterteilt. Wie viel Platz einer Spalte zur Verfügung steht, hängt von der Auflösung des Endgeräts ab.



- Box-Layout

Das Box-Layout ist ein einfaches Layout mit nur einem Inhaltselement. Höhe, Breite, Rahmen, Margin und Padding können nur für das Box-Layout festgelegt werden.

Das vorgegebene Layout steuert, wie die Controls (Text, Radio-Buttons, Checkboxes, Bilder etc.) angezeigt werden. Die verschiedenen Layouts können ineinander verschachtelt werden. Layouts können verschiedene [Controls](#) enthalten.

Das Layout wird innerhalb einer Component deklariert:

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

10.1 Box-Layout

Das Box-Layout ist ein einfaches Layout mit nur einem Inhaltselement. Höhe, Breite, Rahmen, Margin und Padding können nur für das Box-Layout festgelegt werden.

Das Element `<BoxLayout>` erzeugt ein Box-Layout.

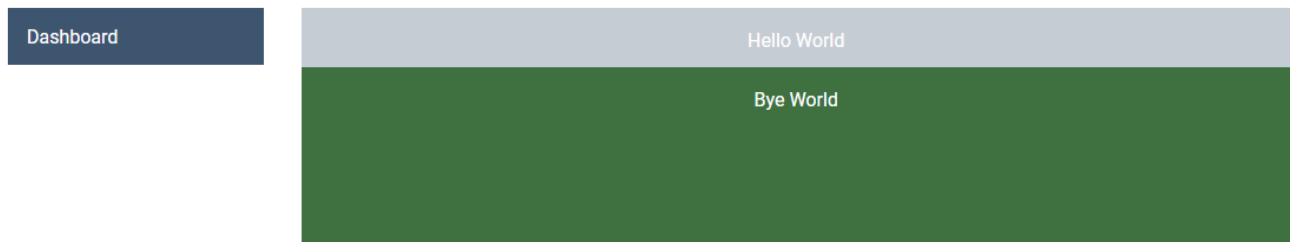
- ✓ Weitere Informationen finden Sie im Abschnitt
 - `<BoxLayout>`

10.1.1 Beispiele

10.1.1.1 BoxLayout in Kombination mit FlowLayout

```
<DetailComponent
  default="true"
  displayName="Dashboard"
  path="Dashboard">
  <BoxLayout
    background="50"
    foreground="A100">
    <FlowLayout>
      <BoxLayout
        background="100"
        height="50"
        units="pixels">
        <Label value="Hello World" />
      </BoxLayout>
      <BoxLayout
        background="10"
        height="150"
        units="pixels"
        horizontalAlign="center">
        <Label value="Bye World" />
      </BoxLayout>
    </FlowLayout>
  </BoxLayout>
</DetailComponent>
```

10.1.1.1.1 Desktop-Ansicht



10.1.2 <BoxLayout>

Für das Element <BoxLayout> lassen sich folgende Attribute festlegen:

Attribut	Beschreibung
background	Definiert eine Hintergrundfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas. Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
border	Legt fest, ob um das Layout ein Rahmen angezeigt wird. Mögliche Werte: <i>true</i> / <i>false</i>
border-left	Legt fest, ob ein linker Rand angezeigt wird Mögliche Werte: <i>true</i> / <i>false</i>
border-right	Legt fest, ob ein rechter Rand angezeigt wird Mögliche Werte: <i>true</i> / <i>false</i>
border-top	Legt fest, ob ein oberer Rand angezeigt wird Mögliche Werte: <i>true</i> / <i>false</i>

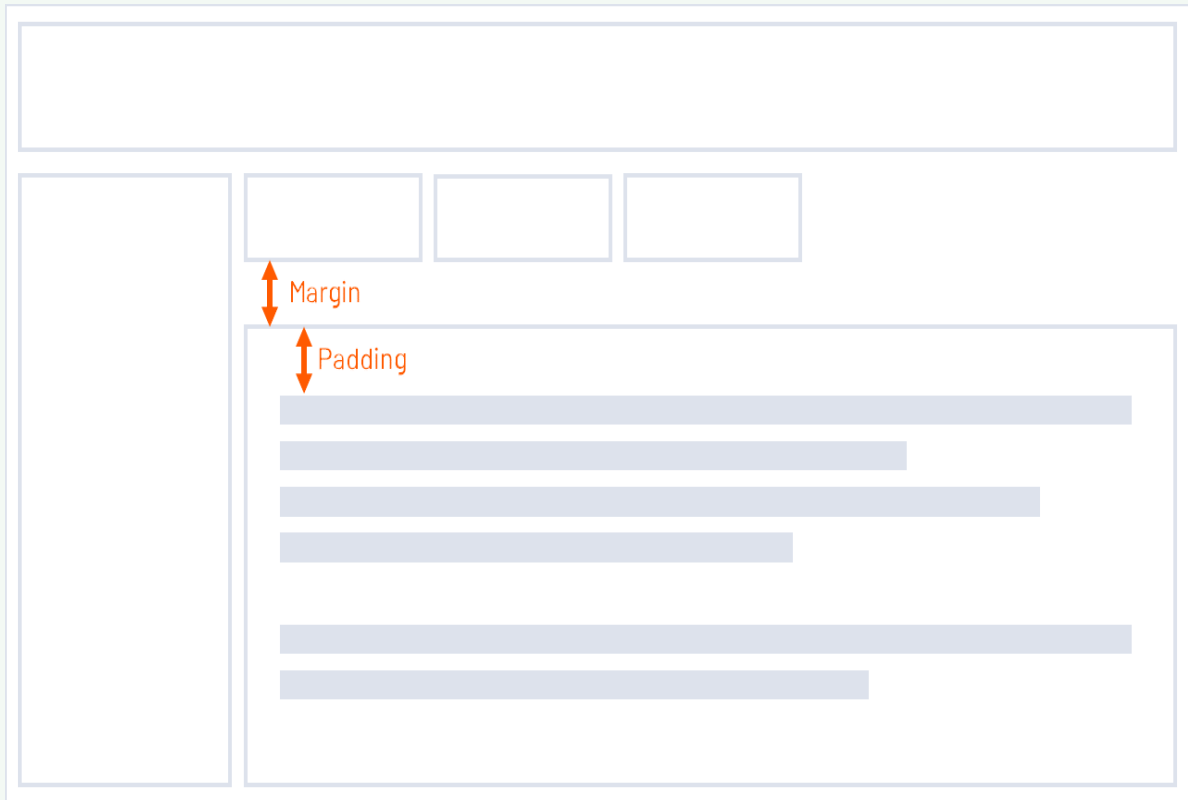
Attribut	Beschreibung
border-bottom	Legt fest, ob ein unterer Rand angezeigt wird Mögliche Werte: <i>true</i> / <i>false</i>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight
foreground	Definiert die Schriftfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  • Verwenden Sie keine Raute vor dem Farbwert. • Verwenden Sie keine verkürzte Schreibweise des Farbwerts. • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
height	Höhe des Layouts Mögliche Werte: Ganze Zahlen

Attribut	Beschreibung
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i>(Standard) • <i>center</i> • <i>right</i>
paddingBottom	Unterer Innenabstand Mögliche Werte: Beliebige Zahlenwerte
paddingLeft	Linker Innenabstand Mögliche Werte: Beliebige Zahlenwerte
paddingRight	Rechter Innenabstand Mögliche Werte: Beliebige Zahlenwerte
paddingTop	Oberer Innenabstand Mögliche Werte: Beliebige Zahlenwerte
paddingUnits	Einheiten für den Innenabstand Mögliche Werte: • <i>pixels</i> • <i>percents</i>
marginBottom	Unterer Außenabstand Mögliche Werte: Beliebige Zahlenwerte
marginLeft	Linker Außenabstand Mögliche Werte: Beliebige Zahlenwerte
marginRight	Rechter Außenabstand Mögliche Werte: Beliebige Zahlenwerte
marginTop	Oberer Außenabstand Mögliche Werte: Beliebige Zahlenwerte

Attribut	Beschreibung
marginUnits	Einheiten für den Außenabstand Mögliche Werte: • pixels • percents
textOverflow	Gibt an, was passieren soll, wenn das Layout voll ist. Mögliche Werte: • ellipsis (Standard): Mit ... zeigen, dass der Text nicht zu Ende ist • hidden: Text abbrechen, auf ganze Wörter achten • wordBreak: innerhalb des Wortes abbrechen • allow: Text zwischen den Wörtern umbrechen
units	Gibt die Einheit an, die für die Größenangaben height und width gilt. Mögliche Werte: • pixels (Standard) • percents
visible	Legt fest, ob das Layout sichtbar ist. • Data Binding (Boolean) möglich. Mögliche Werte: Boolean oder Zeichenkette für Data Binding
width	Legt die Breite des Layouts fest. Mögliche Werte: Ganze Zahlen

- ✓ **Padding** bezeichnet den Innenabstand, 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.



10.2 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.

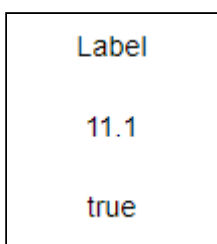
Das Element `<FlowLayout>` erzeugt ein Flow-Layout.



- ✓ Weitere Informationen finden Sie im Abschnitt
- `<FlowLayout>`

10.2.1 Beispiele

```
<FlowLayout>
  <Label value="Label"/>
  <Label value="1"/>
  <Label value="true"/>
</FlowLayout>
```



10.2.2 `<FlowLayout>`

Für das Element `<FlowLayout>` lassen sich folgende Attribute festlegen:

Attribut	Beschreibung
direction	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>horizontal</i> • <i>vertical</i> (Standard)
background	Definiert eine Hintergrundfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
enabled	Aktiviert das Layout • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding
fontFamily	Legt die Schriftfamilie fest.  • Mit diesem Attribut wird die Standardschrift der Web App für dieses Layout überschrieben. • Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut fontFamily nicht explizit gesetzt haben. Mögliche Werte: • MainFont: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. Font04

Attribut	Beschreibung
fontSize	Legt die Schriftgröße fest.  - Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Layout überschrieben. Mögliche Werte: - Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <i>20</i>; <i>20.8</i>; <i>.9</i> - Schriftgröße in Pixel, z. B. <i>20px</i> - Schriftgröße in Punkt, z. B. <i>18pt</i> - Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i> - 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>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: - Condensed - Expanded - ExtraCondensed - ExtraExpanded - Medium - Normal (Standard) - SemiCondensed - SemiExpanded - UltraCondensed - UltraExpanded
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: <i>italic</i>: kursive Schrift <i>normal</i>: normale Schrift (Standard) <i>oblique</i>: schräg gestellter Schriftstil (berechnet)

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Layouts.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
textOverflow	Gibt an, was passieren soll, wenn das Layout voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen

Attribut	Beschreibung
visible	Legt fest, ob das Layout sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

i 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.

10.3 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 Element `<GridLayout>` erzeugt ein Grid-Layout. Bei der Anzeige des Grid-Layouts auf kleineren Bildschirmen (Smartphones, Tablets) bleibt die Anzahl der Spalten identisch, die Spalten werden nur schmaler angezeigt.



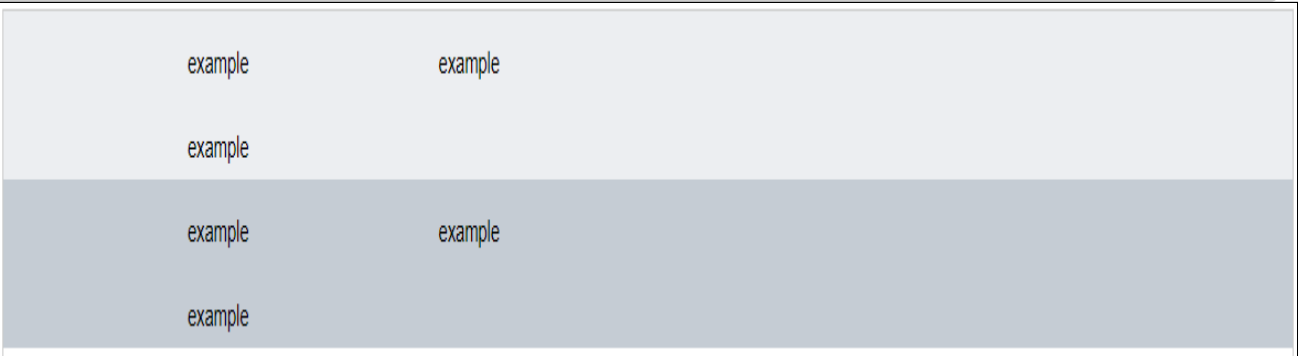
✓ Weitere Informationen finden Sie im Abschnitt

- `<GridLayout>`
- `<ColumnDefinitions>`
- `<Column>`

10.3.1 Beispiele

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>
```



10.3.2 <GridLayout>

Für das Element <GridLayout> lassen sich folgende Attribute festlegen:

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

Attribut	Beschreibung
rows	Legt fest, aus wie vielen Zeilen das Layout-Gitter besteht. Mögliche Werte: Ganze Zahlen
background	Definiert eine Hintergrundfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
enabled	Aktiviert das Layout Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding
fontFamily	Legt die Schriftfamilie fest.  - Mit diesem Attribut wird die Standardschrift der Web App für dieses Layout überschrieben. - Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut fontFamily nicht explizit gesetzt haben. Mögliche Werte: - MainFont: hinterlegte Hauptschrift - Fontcode aus der Font-Palette, z. B. Font04

Attribut	Beschreibung
fontSize	Legt die Schriftgröße fest.  - Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Layout überschrieben. Mögliche Werte: - Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <i>20</i>; <i>20.8</i>; <i>.9</i> - Schriftgröße in Pixel, z. B. <i>20px</i> - Schriftgröße in Punkt, z. B. <i>18pt</i> - Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i> - 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>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: - Condensed - Expanded - ExtraCondensed - ExtraExpanded - Medium - Normal (Standard) - SemiCondensed - SemiExpanded - UltraCondensed - UltraExpanded
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: <i>italic</i>: kursive Schrift <i>normal</i>: normale Schrift (Standard) <i>oblique</i>: schräg gestellter Schriftstil (berechnet)

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Layouts.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>

Attribut	Beschreibung
textOverflow	Gibt an, was passieren soll, wenn das Layout voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Layout sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

✓ Weitere Informationen finden Sie im Abschnitt

- [<ColumnDefinitions>](#)
- [<Column>](#)

10.3.3 <ColumnDefinitions>

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">
  <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.

✓ Weitere Informationen finden Sie im Abschnitt

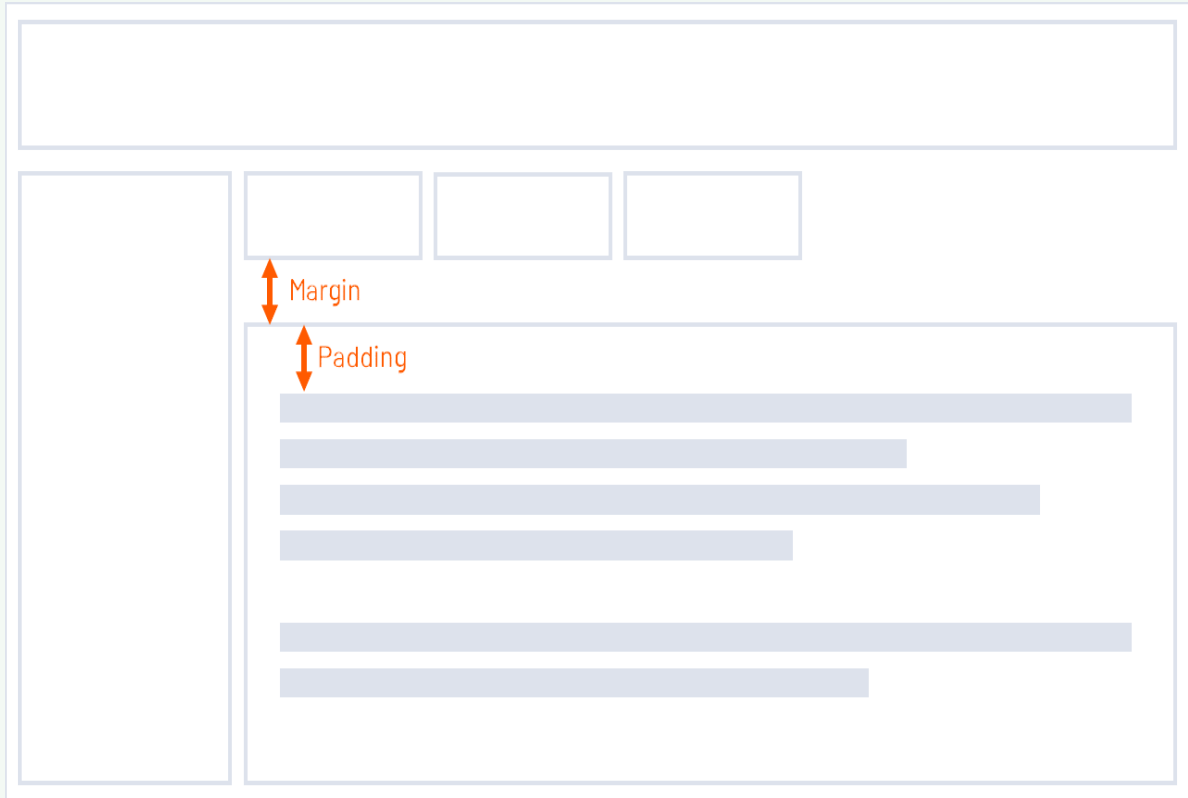
- [<GridLayout>](#)
- [<Column>](#)

10.3.4 <Column>

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

Attribut	Beschreibung
<code>horizontalAlign</code>	Positionierung innerhalb der Spalte Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
<code>paddingBottom</code>	Unterer Innenabstand Mögliche Werte: Beliebige Zahlenwerte
<code>paddingLeft</code>	Linker Innenabstand Mögliche Werte: Beliebige Zahlenwerte
<code>paddingRight</code>	Rechter Innenabstand Mögliche Werte: Beliebige Zahlenwerte
<code>paddingTop</code>	Oberer Innenabstand Mögliche Werte: Beliebige Zahlenwerte
<code>paddingUnits</code>	Einheiten für den Innenabstand Mögliche Werte: • <i>pixels</i> • <i>percents</i>

- ✓ **Padding** bezeichnet den Innenabstand, 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.



- ✓ Weitere Informationen finden Sie im Abschnitt
 - `<GridLayout>`
 - `<ColumnDefinitions>`

10.4 Registerkarten-Layout: Tab Group

Um Registerkarten innerhalb einer [Master/Detail Component](#) zu verwenden, kann in [Master-](#), [Detail-](#) und [Overlay-Strukturelementen](#) das Element `<TabGroup>` verwendet werden. Das `<TabGroup>`-Element enthält eine oder mehrere Components, die in jeweils einer Registerkarte angezeigt werden.

- ✓ Weitere Informationen finden Sie im Abschnitt
 - `<TabGroup>`

10.4.1 Beispiele

```

<MasterDetailComponent
  default="true"
  path="Companies"
  displayName="Companies">
  <Master defaultComponentName="overview">
    <TabGroup>
      <DetailComponent
        displayName="Overview"
        name="overview">
        <ResponsiveLayout defaultCellSizeMobile="6">
          <Cell cellSizeDesktop="3">
            <FlowLayout>
              <TextBox
                displayName="Unternehmen"
                value="SoftProject GmbH" />
            </FlowLayout>
          </Cell>
          <Cell cellSizeDesktop="3">
            <GridLayout columns="2">
              <TextBox
                displayName="Straße"
                value="Am Erlengraben 3" />
              <TextBox
                displayName="Land"
                value="Deutschland" />
              <TextBox
                displayName="Postleitzahl"
                value="76275" />
              <TextBox
                displayName="Ort"
                value="Ettlingen" />
            </GridLayout>
          </Cell>
          <Cell
            cellSizeDesktop="6"
            cellSizeTablet="12">
            <Map vendor="OpenStreetMap">
              <SingleMarker
                name="Marker"
                searchString="Am Erlengraben 3, 76275 Ettlingen" />
            </Map>
          </Cell>
        </ResponsiveLayout>
      </DetailComponent>
      <DetailComponent
        displayName="Detail"
        name="detail">
        <FlowLayout>
          <!-- Content -->
        </FlowLayout>
      </DetailComponent>
    </TabGroup>
  </Master>
</MasterDetailComponent>

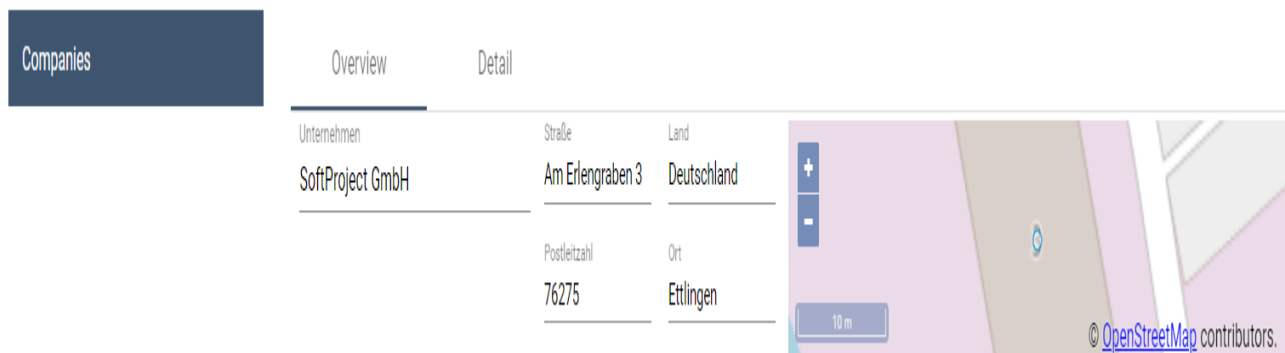
```

```

        </DetailComponent>
    </TabGroup>
</Master>
<Detail>
    <!-- Content -->
</Detail>
</MasterDetailComponent>

```

10.4.1.1 Desktop-Ansicht



10.4.2 <TabGroup>

Für das Element <TabGroup> lassen sich folgende Attribute festlegen:

Attribut	Beschreibung
background	Definiert eine Hintergrundfarbe. ⚠ Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00 ⚠ Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
description	Beschreibungstext der Component. Wird in jedem Tab über der jeweiligen Component angezeigt. Data Binding möglich Mögliche Werte: Beliebige Zeichenkette

Attribut	Beschreibung
fontFamily	Legt die Schriftfamilie fest. • Mit diesem Attribut wird die Standardschrift der Web App für dieses Layout überschrieben. • 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. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest. • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Layout überschrieben. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: • <code>Condensed</code> • <code>Expanded</code> • <code>ExtraCondensed</code> • <code>ExtraExpanded</code> • <code>Medium</code> • <code>Normal (Standard)</code> • <code>SemiCondensed</code> • <code>SemiExpanded</code> • <code>UltraCondensed</code> • <code>UltraExpanded</code>

Attribut	Beschreibung
fontStyle	Legt die Neigung der Schrift fest. Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: • <code>italic</code>: kursive Schrift • <code>normal</code>: normale Schrift (Standard) • <code>oblique</code>: schräg gestellter Schriftstil (berechnet)
fontWeight	Legt die Schriftstärke fest. Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: • <code>Black</code> • <code>Bold</code> • <code>DemiBold</code> • <code>ExtraBlack</code> • <code>ExtraBold</code> • <code>ExtraLight</code> • <code>Heavy</code> • <code>Light</code> • <code>Medium</code> • <code>Normal</code> (Standard) • <code>Regular</code> • <code>SemiBold</code> • <code>Thin</code> • <code>UltraBlack</code> • <code>UltraBold</code> • <code>UltraLight</code>
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Layouts. • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. <code>ff5a00</code> Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <code>A200</code>

Attribut	Beschreibung
title	Titel der Gruppe • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
subtitle	Untertitel der Gruppe • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette

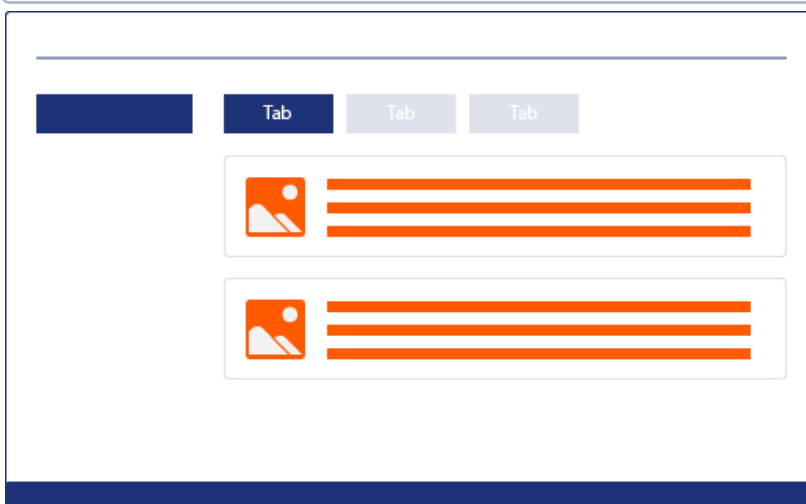
10.5 Registerkarten-Layout: Tab Layout

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

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



10.5.1 Beispiele

```
<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>
```



10.5.2 <TabLayout>

Für das Element <TabLayout> lassen sich folgende Attribute festlegen:

Attribut	Beschreibung
background	Definiert eine Hintergrundfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
enabled	Aktiviert das Layout • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding
fontFamily	Legt die Schriftfamilie fest.  • Mit diesem Attribut wird die Standardschrift der Web App für dieses Layout überschrieben. • 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. Mögliche Werte: • MainFont: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <i>Font04</i>
fontSize	Legt die Schriftgröße fest.  Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Layout überschrieben. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <i>20</i>; <i>20.8</i>; <i>.9</i> • Schriftgröße in Pixel, z. B. <i>20px</i> • Schriftgröße in Punkt, z. B. <i>18pt</i> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i> • 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>

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Layouts.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert. Verwenden Sie keine verkürzte Schreibweise des Farbwerts. • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>

Attribut	Beschreibung
textOverflow	Gibt an, was passieren soll, wenn das Layout voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
title	Titel für das Registerkarten-Layout wird als Header angezeigt. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
visible	Legt fest, ob das Layout sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

10.5.3 <Tab>

Für das Element <Tab> lassen sich folgende Attribute festlegen:

Attribut	Beschreibung
background	Definiert eine Hintergrundfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert und keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
displayName	Angezeigter Titel für die Registerkarte • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette

Attribut	Beschreibung
enabled	Aktiviert das Layout • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding
fontFamily	Legt die Schriftfamilie fest.  • Mit diesem Attribut wird die Standardschrift der Web App für dieses Layout überschrieben. • Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut fontFamily nicht explizit gesetzt haben. Mögliche Werte: • MainFont: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <i>Font04</i>
fontSize	Legt die Schriftgröße fest.  • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Layout überschrieben. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <i>20</i>; <i>20.8</i>; <i>.9</i> • Schriftgröße in Pixel, z. B. <i>20px</i> • Schriftgröße in Punkt, z. B. <i>18pt</i> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i> • 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>

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)

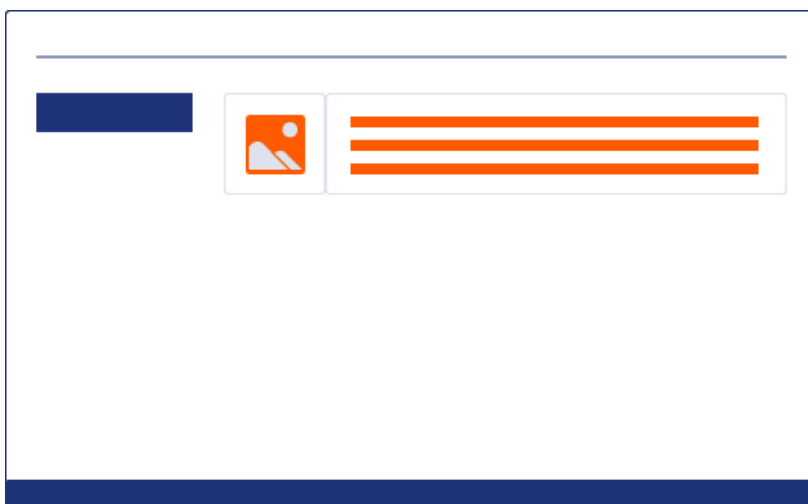
Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Layout überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Layouts.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>

Attribut	Beschreibung
<code>textOverflow</code>	Gibt an, was passieren soll, wenn das Layout voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
<code>visible</code>	Legt fest, ob das Layout sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

10.6 Responsive-Layout

Das Responsive-Layout bietet sich für Web-Anwendungen an, die auf unterschiedlichen Endgeräten angezeigt werden sollen. Das Responsive-Layout ist in 12 virtuelle Spalten unterteilt. Wie viel Platz einer Spalte zur Verfügung steht, hängt von der Auflösung des Endgeräts ab.

Das Element `<ResponsiveLayout>` erzeugt ein Responsive-Layout. Innerhalb des Responsive-Layout `<ResponsiveLayout>` muss mindestens ein Element `<Cell>` angegeben werden. Für jedes Element `<Cell>` kann die Anzahl der Spalten angegeben werden, dass das Element auf dem entsprechenden Endgerät einnimmt. Im Element `<Cell>` können weitere Layout-Typen verschachtelt werden. Beim Responsive-Layout werden die Inhalte innerhalb des Elements `<Cell>` an den verfügbaren Bildschirmplatz angepasst und bei Bedarf umbrochen.



✔ Weitere Informationen finden Sie im Abschnitt

- `<ResponsiveLayout>`
- `<Cell>`

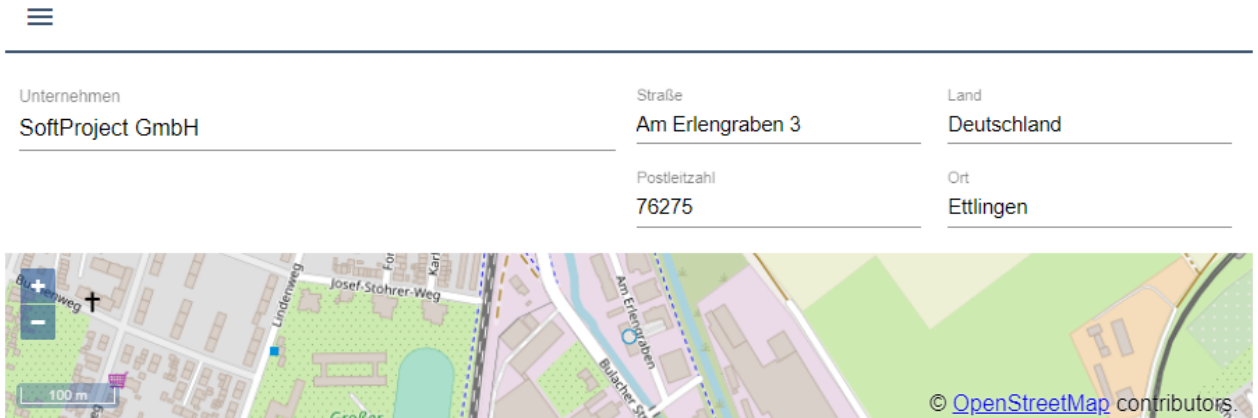
10.6.1 Beispiele

```
<DetailComponent
  default="true"
  displayName="Dashboard"
  path="Dashboard">
  <ResponsiveLayout defaultCellSizeTablet="6">
    <Cell cellSizeTablet="3">
      <FlowLayout>
        <TextBox
          displayName="Unternehmen"
          value="SoftProject GmbH" />
        </FlowLayout>
      </Cell>
      <Cell cellSizeDesktop="3">
        <GridLayout columns="2">
          <TextBox
            displayName="Straße"
            value="Am Erlengraben 3" />
          <TextBox
            displayName="Land"
            value="Deutschland" />
          <TextBox
            displayName="Postleitzahl"
            value="76275" />
          <TextBox
            displayName="Ort"
            value="Ettlingen" />
        </GridLayout>
      </Cell>
      <Cell
        cellSizeDesktop="6"
        cellSizeTablet="12">
        <Map vendor="OpenStreetMap">
          <SingleMarker
            name="Marker"
            searchString="Am Erlengraben 3, 76275 Ettlingen" />
          </Map>
        </Cell>
      </ResponsiveLayout>
    </DetailComponent>
```

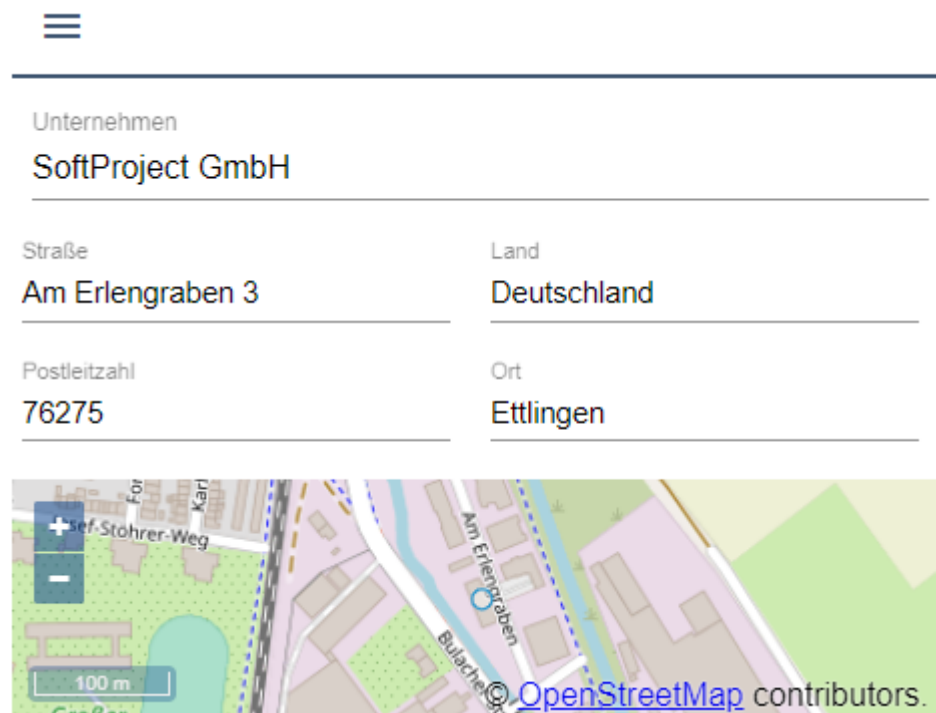
Desktop-Ansicht



Tablet-Ansicht



Mobile-Ansicht



10.6.2 <ResponsiveLayout>

Für das Element <ResponsiveLayout> lassen sich folgende Attribute festlegen:

Attribut	Beschreibung
defaultCellSizeDesktop	Legt die Anzahl der Spalten fest, die das Element <code><Cell></code> bei der Anzeige auf Desktop-Computern standardmäßig einnimmt. Mögliche Werte: Ganze Zahlen zwischen 1-12 (Standard: 4)  Der hier festgelegte Standard wird überschrieben, wenn das Attribut <code>cellSizeDesktop</code> im untergeordneten Element <code><Cell></code> festgelegt wird.
defaultCellSizeTablet	Legt die Anzahl der Spalten fest, die das Element <code><Cell></code> bei der Anzeige auf Tablets standardmäßig einnimmt. Mögliche Werte: Ganze Zahlen zwischen 1-12 (Standard: 6)  Der hier festgelegte Standard wird überschrieben, wenn das Attribut <code>cellSizeTablet</code> im untergeordneten Element <code><Cell></code> festgelegt wird.
defaultCellSizeMobile	Legt die Anzahl der Spalten fest, die das Element <code><Cell></code> bei der Anzeige auf Smartphones standardmäßig einnimmt. Mögliche Werte: Ganze Zahlen zwischen 1-12 (Standard: 12)  Der hier festgelegte Standard wird überschrieben, wenn das Attribut <code>cellSizeMobile</code> im untergeordneten Element <code><Cell></code> festgelegt wird.
background	Definiert eine Hintergrundfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas. Mögliche Werte: - Hexadezimaler Farbwert, z. B. <code>ff5a00</code>  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <code>A200</code>

Attribut	Beschreibung
fontFamily	Legt die Schriftfamilie fest. i • Mit diesem Attribut wird die Standardschrift der Web App überschrieben. • 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. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest. i Mit diesem Attribut wird die Standardschriftgröße der Web App überschrieben. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • Schlüsselwörter: <code>xx-small</code>, <code>x-small</code>, <code>small</code>, <code>medium</code>, <code>large</code>, <code>xx-large</code>, <code>smaller</code>, <code>larger</code>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. i Mit diesem Attribut wird die Standardbreite der Schriftzeichen überschrieben. Mögliche Werte: • <code>Condensed</code> • <code>Expanded</code> • <code>ExtraCondensed</code> • <code>ExtraExpanded</code> • <code>Medium</code> • <code>Normal (Standard)</code> • <code>SemiCondensed</code> • <code>SemiExpanded</code> • <code>UltraCondensed</code> • <code>UltraExpanded</code>

Attribut	Beschreibung
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen überschrieben. Mögliche Werte: • <code>italic</code>: kursive Schrift • <code>normal</code>: normale Schrift (Standard) • <code>oblique</code>: schräg gestellter Schriftstil (berechnet)
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen überschrieben. Mögliche Werte: • <code>Black</code> • <code>Bold</code> • <code>DemiBold</code> • <code>ExtraBlack</code> • <code>ExtraBold</code> • <code>ExtraLight</code> • <code>Heavy</code> • <code>Light</code> • <code>Medium</code> • <code>Normal(Standard)</code> • <code>Regular</code> • <code>SemiBold</code> • <code>Thin</code> • <code>UltraBlack</code> • <code>UltraBold</code> • <code>UltraLight</code>

Attribut	Beschreibung
foreground	Definiert die Schriftfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: <i>left</i> (Standard) <i>center</i> <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Layout voll ist. Mögliche Werte: <i>ellipsis</i> (Standard): Mit ... zeigen, dass der Text nicht zu Ende ist <i>hidden</i>: Text abbrechen, auf ganze Wörter achten <i>wordBreak</i>: innerhalb des Wortes abbrechen <i>allow</i>: Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Layout sichtbar ist. Data Binding (Boolean) möglich. Mögliche Werte: Boolean oder Zeichenkette für Data Binding

 Das Element <ResponsiveLayout> muss folgende Elemente enthalten:

- Cell

 Weitere Informationen finden Sie im Abschnitt

- <Cell>

10.6.3 <Cell>

Das Element <Cell> wird verwendet, um einen Inhaltsbereich im Element <ResponsiveLayout> zu erstellen.

Für das Element <Cell> lassen sich folgende Attribute festlegen:

Attribut	Beschreibung
cellSizeDesktop	Legt die Anzahl der Spalten fest, die das Element <Cell> bei der Anzeige auf Desktop-Computern einnimmt. Mögliche Werte: Ganze Zahlen zwischen 1-12  Der hier festgelegte Wert überschreibt das Attribut <code>defaultCellSizeDesktop</code> im Eltern-Element <ResponsiveLayout>.
cellSizeTablet	Legt die Anzahl der Spalten fest, die das Element <Cell> bei der Anzeige auf Tablets einnimmt. Mögliche Werte: Ganze Zahlen zwischen 1-12  Der hier festgelegte Wert überschreibt das Attribut <code>defaultCellSizeTablet</code> im Eltern-Element <ResponsiveLayout>.
cellSizeMobile	Legt die Anzahl der Spalten fest, die das Element <Cell> bei der Anzeige auf Smartphones einnimmt. Mögliche Werte: Ganze Zahlen zwischen 1-12  Der hier festgelegte Wert überschreibt das Attribut <code>defaultCellSizeMobile</code> im Eltern-Element <ResponsiveLayout>.
background	Definiert eine Hintergrundfarbe. Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
fontFamily	Legt die Schriftfamilie fest. i • Mit diesem Attribut wird die Standardschrift der Web App überschrieben. • 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. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest. i Mit diesem Attribut wird die Standardschriftgröße der Web App überschrieben. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • Schlüsselwörter: <code>xx-small</code>, <code>x-small</code>, <code>small</code>, <code>medium</code>, <code>large</code>, <code>xx-large</code>, <code>smaller</code>, <code>larger</code>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. i Mit diesem Attribut wird die Standardbreite der Schriftzeichen überschrieben. Mögliche Werte: • <code>Condensed</code> • <code>Expanded</code> • <code>ExtraCondensed</code> • <code>ExtraExpanded</code> • <code>Medium</code> • <code>Normal (Standard)</code> • <code>SemiCondensed</code> • <code>SemiExpanded</code> • <code>UltraCondensed</code> • <code>UltraExpanded</code>

Attribut	Beschreibung
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen überschrieben. Mögliche Werte: • <code>italic</code>: kursive Schrift • <code>normal</code>: normale Schrift (Standard) • <code>oblique</code>: schräg gestellter Schriftstil (berechnet)
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen überschrieben. Mögliche Werte: • <code>Black</code> • <code>Bold</code> • <code>DemiBold</code> • <code>ExtraBlack</code> • <code>ExtraBold</code> • <code>ExtraLight</code> • <code>Heavy</code> • <code>Light</code> • <code>Medium</code> • <code>Normal(Standard)</code> • <code>Regular</code> • <code>SemiBold</code> • <code>Thin</code> • <code>UltraBlack</code> • <code>UltraBold</code> • <code>UltraLight</code>

Attribut	Beschreibung
foreground	Definiert die Schriftfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  - Verwenden Sie keine Raute vor dem Farbwert. - Verwenden Sie keine verkürzte Schreibweise des Farbwerts. - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
minHeight	Minimale Höhe des Layouts Mögliche Werte: Ganze Zahlen
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: <i>left</i> (Standard) <i>center</i> <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Layout voll ist. Mögliche Werte: <i>ellipsis</i> (Standard): Mit ... zeigen, dass der Text nicht zu Ende ist <i>hidden</i>: Text abbrechen, auf ganze Wörter achten <i>wordBreak</i>: innerhalb des Wortes abbrechen <i>allow</i>: Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Layout sichtbar ist. Data Binding (Boolean) möglich. Mögliche Werte: Boolean oder Zeichenkette für Data Binding

- ✓ Weitere Informationen finden Sie im Abschnitt
- [<ResponsiveLayout>](#)

11 Controls

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

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

11.1 Button

<Button>-Controls werden verwendet, um Schaltflächen zu erzeugen. Die Schaltfläche kann in der Web App frei platziert werden. Wenn Sie eine Aktions-Schaltfläche immer am oberen Rand vor dem Inhalt der Component anzeigen möchten, erzeugen Sie eine Schaltfläche mit [Actions](#).

i Das Element <Button> kann folgende Aktionen enthalten:

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

Folgende zusätzlichen Attribute lassen sich für das Element <Button> definieren:

Attribut	Beschreibung
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • left (Standard) • center • right

Attribut	Beschreibung
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • ellipsis: Mit ... zeigen, dass der Text nicht zu Ende ist • hidden: Text abbrechen, auf ganze Wörter achten • wordBreak: innerhalb des Wortes abbrechen • allow(Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true / false</i> oder Zeichenkette für Data Binding

11.1.1 Beispiel

```

1  <DetailComponent default="true" displayName="Dashboard" path="Dashboard">
2    <FlowLayout>
3      <Header value="Welcome to my new Web App!"/>
4      <TextBlock>Upload your file here</TextBlock>
5      <Button horizontalAlign="center">
6        <UploadAction displayName="Upload" iconUrl="icon:file_upload"/
7      >
8        </Button>
9    </FlowLayout>
  </DetailComponent>

```

11.2 Charts

In einer X4 Web App können Chart eingebunden werden.

Es können verschiedene Arten von Charts eingebunden werden:

- Liniendiagramm
- Balkendiagramm
- Kreis-/Donutdiagramm
- Tacho

Alle Diagramme in einem Layout müssen innerhalb eines <Chart>-Elements definiert werden.

Attribut	Beschreibung
title	Titel des Charts. Mögliche Werte: Beliebige Zeichenkette oder Data Binding

Attribut	Beschreibung
subtitle	Untertitel des Charts. Mögliche Werte: Beliebige Zeichenkette oder Data Binding
saveAsImage	Ermöglicht das Herunterladen des Charts als Bilddatei im SVG-Format. Mögliche Werte: • Beliebiger String: Legt den Dateinamen fest. • <code>TIMESTAMP</code>: Dateiname besteht aus dem String "Chart_" und einem Zeitstempel
minHeight	Minimale Höhe des Charts Mögliche Werte: Ganzzahl <i>i</i> Wird das Attribut <code>minHeight</code> nicht explizit gesetzt, werden Charts mit einer Höhe von 400 Pixeln angezeigt.
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen

i Die Charts sind auf Größenangaben in Pixeln optimiert. Wenn eine ausdrückliche Definition von Höhe und Breite des Charts benötigt wird, sollte zuvor das Attribut `units` auf *pixels* gesetzt werden.

! Alle Charts müssen per Data Binding befüllt werden!

11.2.1 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.

11.2.1.1 Allgemeine Attribute

Folgende Attribute sind für alle Elemente eines Liniendiagramms verfügbar:

fontFamily	Legt die Schriftfamilie fest. i • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded  Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)  • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert die Schriftfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

11.2.1.2 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. Mögliche Werte: true(Standard)/ false

11.2.1.3 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. Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
horizontalLabel	Definiert die Beschriftung der x-Achse. Mögliche Werte: Beliebige Zeichenkette
horizontalMax	Maximalwert der x-Achse Mögliche Werte: Ganzzahl
horizontalMin	Minimalwert der x-Achse Mögliche Werte: Ganzzahl
horizontalUnits	Einheit der x-Achse Mögliche Werte: Ganzzahl
verticalLabel	Definiert die Beschriftung der y-Achse. Mögliche Werte: Beliebige Zeichenkette
verticalMax	Maximalwert der y-Achse Mögliche Werte: Ganzzahl

Attribut	Beschreibung
verticalMin	Minimalwert der y-Achse Mögliche Werte: Ganzzahl
verticalUnits	Einheit der y-Achse Mögliche Werte: Ganzzahl

11.2.1.4 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. Mögliche Werte: Hexadezimaler Farbwert oder Ausdruck für Data Binding (Color)  Wird das Attribut color nicht definiert, werden die im Theming Editor definierten Farben verwendet.
data	Gibt an, welche Daten im Diagramm dargestellt werden. Mögliche Werte: Data Binding
labels	Gibt an, ob die Datenpunkte mit dem jeweiligen Wert beschriftet werden. Mögliche Werte: true / false (Standard)
marker	Gibt an, wie ein Datenpunkt dargestellt wird. Mögliche Werte: • circle: Kreis • rect: Rechteck • roundRect: Rechteck mit abgerundeten Ecken • triangle: Dreieck • diamond: Raute • pin: Stecknadel • arrow: Pfeil • none: kein Marker

Attribut	Beschreibung
markersEnabled	Gibt an, ob ein Datenpunkt markiert werden soll. Mögliche Werte: true / false
name	Name des Datensatzes, der in der Legende angezeigt wird. Mögliche Werte: Beliebige Zeichenkette
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. Mögliche Werte: 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. Mögliche Werte: Data Binding

11.2.1.5 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.

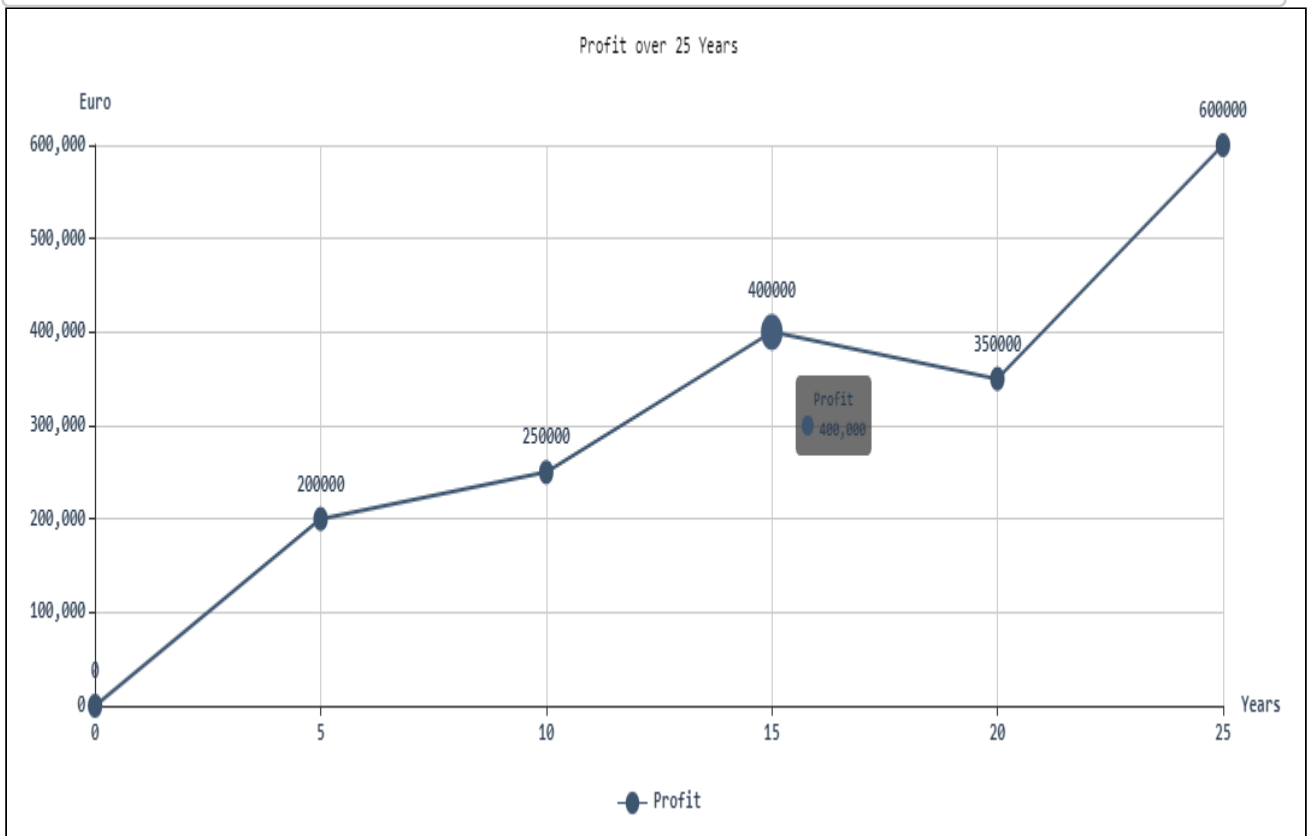
11.2.1.6 Beispiele

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>

```



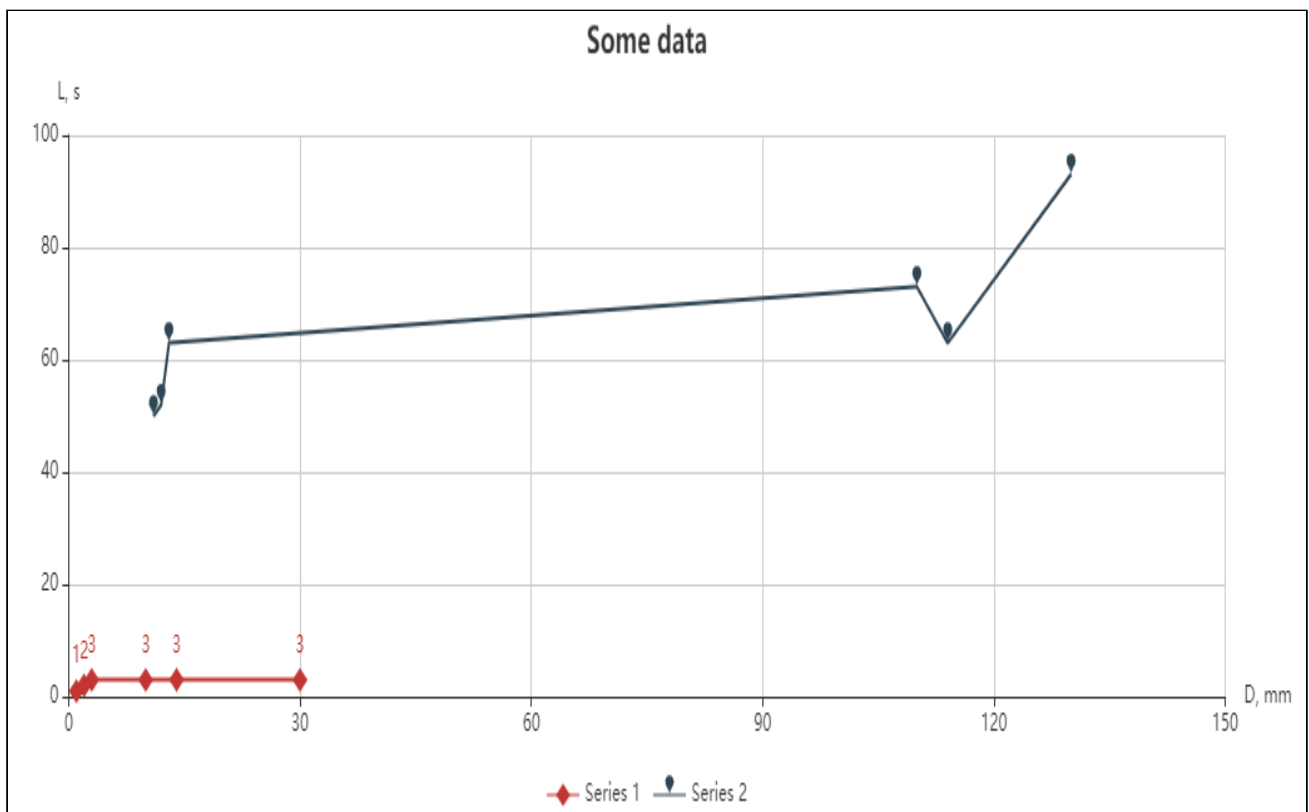
Beispiel 2

```

<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:



11.2.2 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.

11.2.2.1 Allgemeine Attribute

Folgende Attribute sind für alle Elemente eines Balkendiagramms verfügbar:

fontFamily	Legt die Schriftfamilie fest.  • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest.  • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded  Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)  • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert die Schriftfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

11.2.2.2 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. Mögliche Werte: true (Standard) / false
stacked	Gibt an, ob das Balkendiagramm als gestapeltes Diagramm angezeigt werden soll. • Data Binding (Boolean) möglich Mögliche Werte: true / false (Standard)

11.2.2.3 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. Mögliche Werte: Data Binding
color	Farbe des Balkens Mögliche Werte: Hexadezimaler Farbwert oder Ausdruck für Data Binding (Color)  Wird das Attribut color nicht definiert, werden die im Theming Editor definierten Farben verwendet.
data	Gibt an, welches Datenobjekt der Datenquelle visualisiert wird. Mögliche Werte: Data Binding
name	Gibt den Namen der Daten an, der in der Legende angezeigt wird. Mögliche Werte: Beliebige Zeichenkette oder Data Binding
value	Gibt an, welcher Wert des Datenobjekts visualisiert wird. Mögliche Werte: Data Binding

11.2.2.4 Axis

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

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

Attribut	Beschreibung
color	Legt die Farbe der Achsen fest. Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

11.2.2.5 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.

11.2.2.6 Daten leeren

Wenn Sie weitere Daten mithilfe eines Prozesses nachladen, wird das bereits geladene Chart nicht entfernt. Wenn Sie das bereits geladene Chart mit den neuen Daten überschreiben möchten, muss in den Daten das Attribut `empty="true"` deklariert werden. Das Attribut leert die vorhandenen Daten.

11.2.2.6.1 Input 1

Beispiel 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ok>
  <b2 empty="true"/>
  <b1>
    <a>1</a>
    <b>1</b>
  </b1>
</ok>
```

11.2.2.6.2 Input 2

Beispiel 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<Ok>
  <b1 empty="true"/>
  <b2>
    <a>1</a>
    <b>1</b>
  </b2>
</Ok>
```

11.2.2.6.3 Chart in der Component

Beispiel 1:

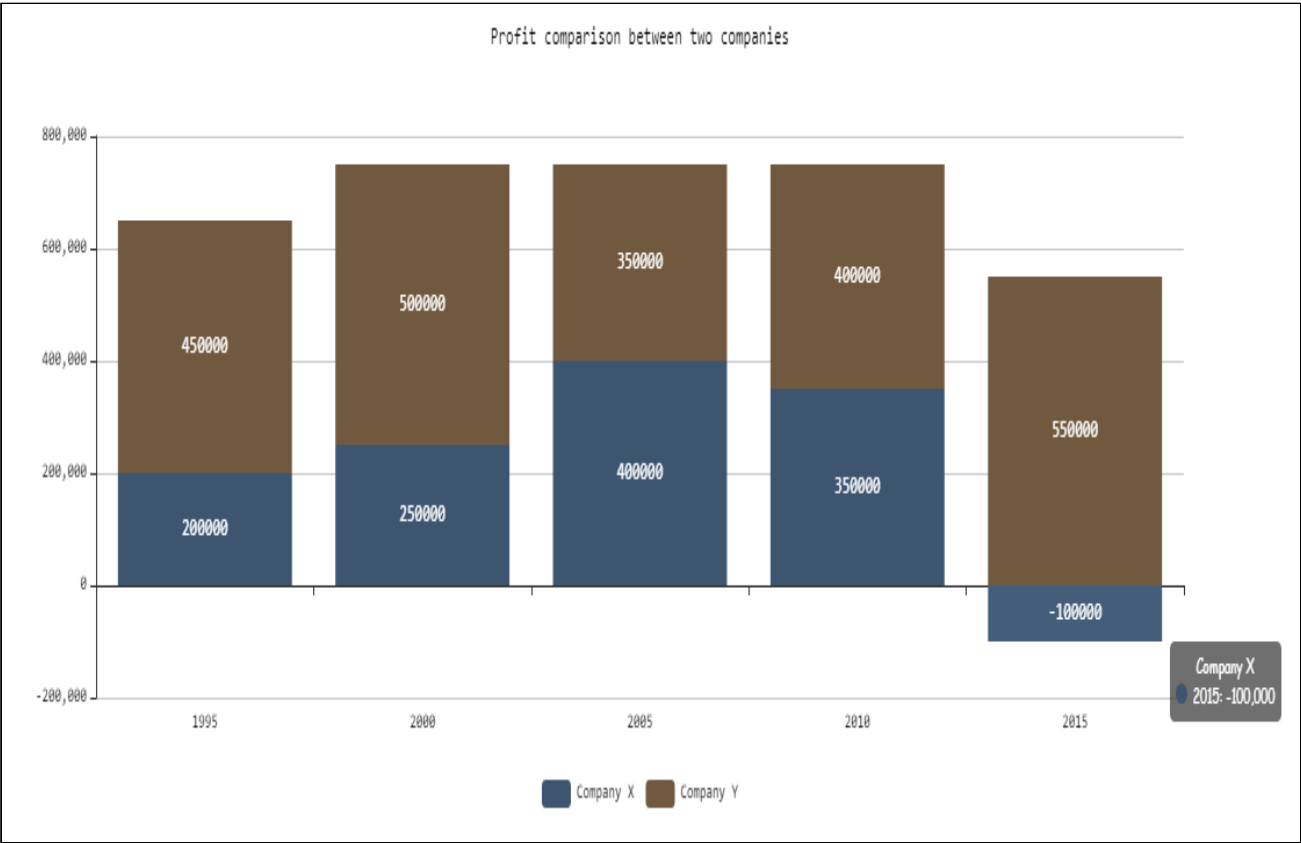
```
<Chart>
  <Bar>
    <BarData
      category="#a"
      data="#b1"
      value="#b" />
    <BarData
      category="#a"
      data="#b2"
      value="#b" />
  </Bar>
</Chart>
```

11.2.2.7 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>
  ...
  ...
  ...

<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:



11.2.3 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.

11.2.3.1 Allgemeine Attribute

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

fontFamily	Legt die Schriftfamilie fest.  • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest.  • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded  Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)  • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert die Schriftfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

11.2.3.2 Pie

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

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

Attribut	Beschreibung
color	Farbe des einzelnen Diagrammausschnitts Mögliche Werte: Hexadezimaler Farbwert oder Ausdruck für Data Binding (Color)  Wird das Attribut <code>color</code> nicht definiert, werden die im Theming Editor definierten Farben verwendet.
data	Gibt an, welche Daten im Diagramm dargestellt werden. Die angegebene Zeichenkette bezieht sich auf eine definierte Property innerhalb der Datenquelle, die im <code><Chart></code>-Element angegeben wird. Mögliche Werte: Data Binding
donut	Gibt an, ob das Diagramm als Donutdiagramm dargestellt werden soll. Mögliche Werte: <code>true</code> / <code>false</code> (Standard)
legend	Gibt an, ob eine Legende angezeigt werden soll. Mögliche Werte: <code>true</code> (Standard) / <code>false</code>
name	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. Mögliche Werte: Data Binding
normals	Gibt an, ob die Werte direkt im Diagramm beschriftet werden sollen. Mögliche Werte: <code>true</code> / <code>false</code> (Standard)
value	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. Mögliche Werte: Data Binding

Attribut	Beschreibung
labelColor	Legt die Schriftfarbe der Beschriftung fest.  Überschreibt die Farbe aller Beschriftungen des Charts. Mögliche Werte: - Hexadezimaler Farbwert, z. B. <i>ff5a00</i>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>
labelWidth	Legt die Breite der Beschriftung in Pixeln fest. Mögliche Werte: Beliebige Ganzzahl
labelOverflow	Gibt an, was passieren soll, wenn die Beschriftung länger als die festgelegte Breite der Beschriftung ist. Mögliche Werte: - none: Verhindert einen Textumbruch (Standard) - truncate: Mit ... zeigen, dass der Text nicht zu Ende ist - break: Text zwischen den Wörtern umbrechen - breakAll: Innerhalb des Wortes abbrechen
avoidLabelOverlap	Legt fest, ob das Überlappen von Beschriftungen verhindert werden soll. Mögliche Werte: - True: Das Überlappen von Beschriftungen wird verhindert. - False: Das Überlappen von Beschriftungen wird nicht verhindert.
radius	Legt den Radius in Pixeln fest. Mögliche Werte: Beliebige Ganzzahl
radiusOuterDonut	Legt den inneren Radius in Pixeln fest.  Nur relevant, wenn das Attribut donut="true" gesetzt ist. Mögliche Werte: Beliebige Ganzzahl

Attribut	Beschreibung
<code>centerHorizontal</code>	Legt die horizontale Position des Diagramms fest. Mögliche Werte: • Beliebige Ganzzahl
<code>centerVertical</code>	Legt die vertikale Position des Diagramms fest. Mögliche Werte: • Beliebige Ganzzahl
<code>tooltipPercentage</code>	Legt fest, ob zusätzlich zum Wert eine Prozentangabe in der Beschriftung und im Tooltip angezeigt wird. Mögliche Werte: • <code>True</code>: Prozentangabe wird angezeigt. • <code>False</code>: Prozentangabe wird nicht angezeigt.

11.2.3.3 Tooltip

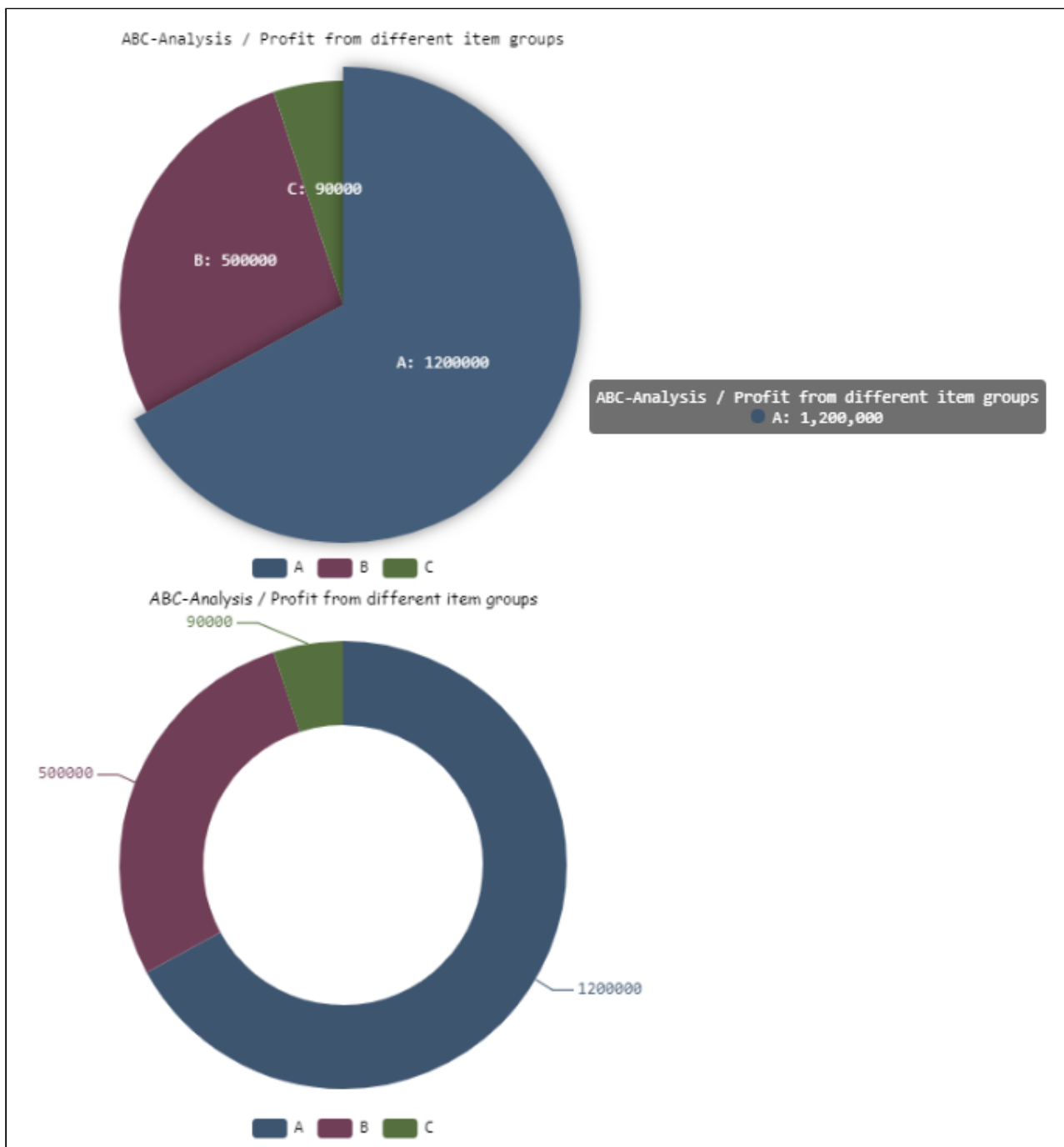
`<Tooltip>`: Kindelement von `<Pie>`. Definiert, wie der Tooltip für die einzelnen Werte im Diagramm angezeigt werden soll.

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

11.2.3.4 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:



11.2.4 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.

11.2.4.1 Allgemeine Attribute

Folgende Attribute sind für alle Elemente eines Tachodiagramms verfügbar:

fontFamily	Legt die Schriftfamilie fest.  • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest.  • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded  Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)  • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert die Schriftfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

11.2.4.2 Gauge

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

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

Attribut	Beschreibung
axisMax	Gibt den Endbereich des Tachodiagramms in Prozent an. Die Farbe des Bereichs wird im Theming Editor über die Farbeigenschaft GaugeAxisMax definiert. Mögliche Werte: Ganzzahl zwischen 0 und 100 (Standard: 100)  Beispiel 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.
axisMiddle	Gibt den mittleren Bereich des Tachodiagramms in Prozent an. Die Farbe des Bereichs wird im Theming Editor über die Farbeigenschaft GaugeAxisMiddle definiert. Mögliche Werte: Ganzzahl zwischen 0 und 100 (Standard: 80)
axisMin	Gibt den Anfangsbereich des Tachodiagramms in Prozent an. Die Farbe des Bereichs wird im Theming Editor über die Farbeigenschaft GaugeAxisMin definiert. Mögliche Werte: Ganzzahl zwischen 0 und 100 (Standard: 20)
endAngle	Gibt den Endpunkt des Tachodiagramms an. Mögliche Werte: Ganzzahl zwischen -360 und 360 (Standard: -45)  endAngle muss immer kleiner sein als startAngle.
legend	Gibt an, ob eine Legende angezeigt werden soll. Mögliche Werte: true (Standard) / false
max	Gibt den Maximalwert des Tachodiagramms an. Mögliche Werte: Ganzzahl (Standard: 100)
min	Gibt den Minimalwert des Tachodiagramms an. Mögliche Werte: Ganzzahl (Standard: 0)  Die Verbindung von min zu max erfolgt immer im Uhrzeigersinn.

Attribut	Beschreibung
startAngle	Gibt den Startpunkt des Tachodiagramms an. Mögliche Werte: Ganzzahl zwischen -360 und 360 (Standard: 255) i Bei dieser Angabe entsprechen folgende Ganzzahlen folgenden Positionen auf einer Uhr: • 0: 3 Uhr • 90: 12 Uhr • 180: 9 Uhr • 270: 6 Uhr
value	Gibt an, welcher Wert des Datenobjekts visualisiert wird. Mögliche Werte: Beliebige Zeichenkette oder Data Binding

11.2.4.3 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.

11.2.4.4 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"/>

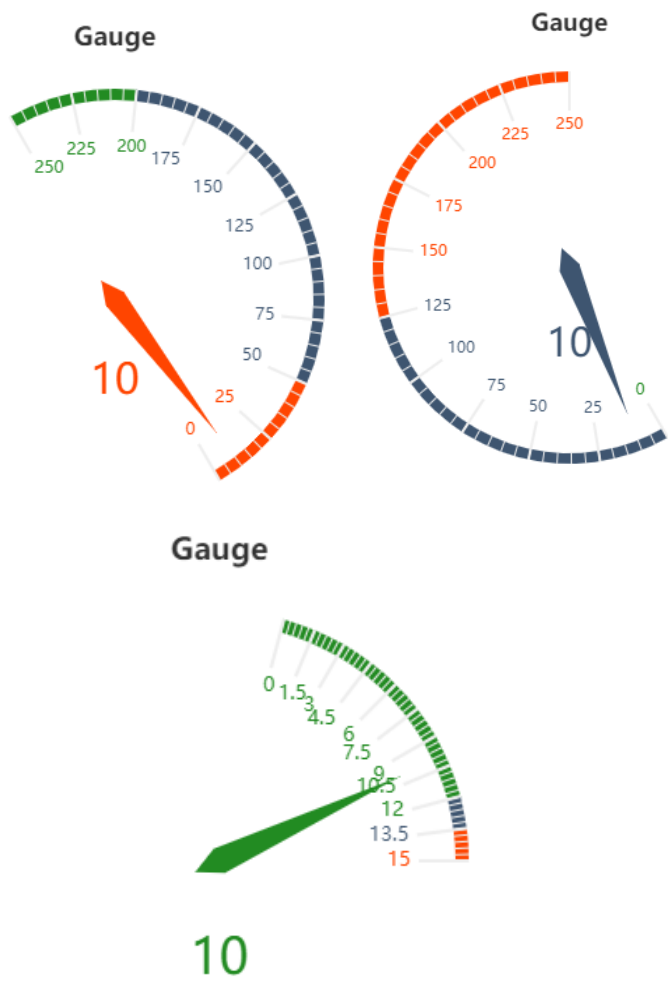
  ...
  ...
  ...
</Property>

<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>

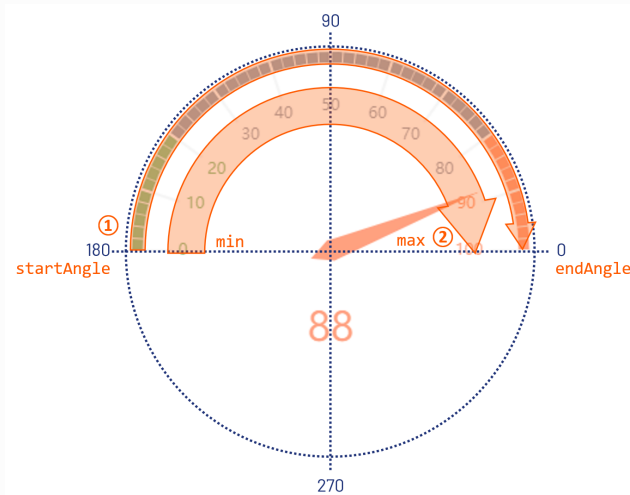
<Chart title="Gauge">
  <Gauge value="#Datasource.gauge" min="-1" max="250" startAngle="120" />
</Chart>
```

Das obige Beispiel führt zu folgenden Diagrammen



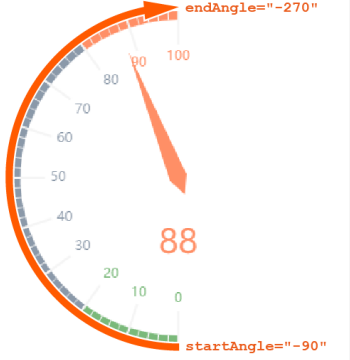
11.2.4.5 Funktionsweise und weitere Beispiele für Tachodiagramme

- 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><Gauge startAngle="0" endAngle="-180" min="0" max="100" ...></pre>		
<pre><Gauge startAngle="90" endAngle="-90" min="0" max="100" ...></pre>		

Definition	Ergebnis	Erläuterung
<code><Gauge startAngle="180" endAngle="0" min="0" max="100" ...></code>		
<code><Gauge startAngle="-90" endAngle="-270" min="0" max="100" ...></code>		
<code><Gauge startAngle="240" endAngle="-60" min="0" max="100" ...></code>		

11.3 Checkbox

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

Attribut	Beschreibung
checked	obligatorisch Gibt an, ob die Checkbox standardmäßig aktiviert oder deaktiviert ist. • Data Binding (Boolean) möglich  Die Verwendung von Data Binding für den Status der Checkbox ist notwendig, damit die Checkbox ordnungsgemäß funktioniert und den initialen Wert von <code>result</code> bzw. <code>set</code> zu speichern. Mögliche Werte: <code>true</code> / <code>false</code> oder Zeichenkette für Data Binding
displayName	Beschriftung. Wird neben der Checkbox angezeigt • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
sliderOnDesktop	Legt fest, ob die Checkbox in der Desktopversion als Schieber (Slider) angezeigt wird. Mögliche Werte: <code>true</code> / <code>false</code>
background	 • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Maps und <code>HtmlDocument</code>! Mögliche Werte: • Hexadezimaler Farbwert, z. B. <code>ff5a00</code>  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <code>A200</code>

Attribut	Beschreibung
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding i 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>
fontFamily	Legt die Schriftfamilie fest. i • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. i Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded i Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.
fontStyle	Legt die Neigung der Schrift fest. i Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet) i Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal(Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controlle Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

 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.

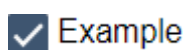
11.3.1 Aktion "Auswählen" innerhalb einer Checkbox

Die Aktion **Auswählen** wird über das Element `<SelectAction/>` innerhalb von `<Checkbox>` erzeugt und ausgelöst, sobald die Checkbox aktiviert oder deaktiviert wird.

11.3.2 Beispiel für `<Checkbox>`

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

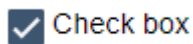
Obiger Code erzeugt folgende Checkbox:



11.3.3 Beispiel für <Checkbox> mit einem Technical Process

```
<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:



11.4 ComboBox

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

Um die Action "Select" bei Auswahl eines ComboBox-Eintrags auszulösen, kann innerhalb des ComboBox-Controls eine `SelectAction`-Aktion eingefügt werden.

Neben den allgemeinen Attributen für Controls lassen sich folgende zusätzlichen Attribute für das Element <ComboBox> definieren:

Attribut	Beschreibung
displayName	Titel der ComboBox. Wird klein über dem Ein-/Ausgabefeld angezeigt. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
displayProperty	Muss angegeben werden, wenn optionsList gesetzt ist. Text, der in der ComboBox als Auswahlmöglichkeit angezeigt wird. Mögliche Werte: Zeichenkette, die einer definierten Property entspricht.  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.

Attribut	Beschreibung
iconColorProperty	Name der Property, die die Farbangaben des Icons enthält. i Dieses Attribut wird nur für Material Icons verwendet. Dabei kann die Farbangabe ein Hexadezimaler Farbwert, z. B. <code>ff5a00</code> oder ein Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <code>A200</code> sein. • Data Binding möglich Mögliche Werte: Zeichenkette, die einer definierten Property entspricht. i Für relative Bindings muss ein Hashtag # als Präfix für die Property verwendet werden, z. B. <code>#Country.Id</code>.
iconProperty	Name der Property, die die URL der Icon-Ressource enthält. i Beim Icon kann es sich entweder um eine Ressource aus dem Resources-Ordner, einem Material Icon oder einem über eine externe URL eingebundenen Icon handeln. Wird das Icon über einen externen Link eingebunden, muss in der Property ein zusätzliches Attribut <code>type="external"</code> angegeben werden. • Data Binding möglich Mögliche Werte: Zeichenkette, die einer definierten Property entspricht. i Für relative Bindings muss ein Hashtag # als Präfix für die Property verwendet werden, z. B. <code>#Country.Id</code>.
optionsAutoWidth	Automatische Breite für die Optionsliste aktivieren Mögliche Werte: • <code>true</code> (Standard): Die Breite der Optionsliste passt sich automatisch der Breite der Optionswerte an • <code>false</code>: Die Breite der Optionsliste richtet sich nach der Breite der ComboBox
optionsList	<i>Bei ausgelagerter Optionsliste:</i> Quelle der Liste. Kann nur zusammen mit <code>displayProperty</code> verwendet werden. • Wird per Data Binding (Liste) angesteuert. Mögliche Werte: Zeichenkette (Data Binding)

Attribut	Beschreibung
optionsMaxWidth	Maximale Breite der Optionsliste in Prozent oder Pixel, wobei Prozent sich auf die Bildschirmbreite bezieht Mögliche Werte: Beliebige ganze Zahl
optionsUnits	Definiert die Einheit für das Attribut optionsMaxWidth Mögliche Werte: • percents • pixels (Standard)
value	Wert, der angezeigt und gespeichert wird. • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
valueProperty	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. Mögliche Werte: Beliebige Zeichenkette  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.
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Maps und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding i 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>
fontFamily	Legt die Schriftfamilie fest. i • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. i Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded i Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument .
fontStyle	Legt die Neigung der Schrift fest. i Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet) i Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

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

Das Element <Option> kann folgende Attribute haben:

Attribut	Beschreibung
iconColor	Farbe des Icons  Dieses Attribut wird nur für Material Icons verwendet. Mögliche Werte: • Hexadezimaler Farbwert, z. B. <i>ff5a00</i> Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>

Attribut	Beschreibung
iconUrl	Pfad zum verwendeten Icon  Beim Icon kann es sich entweder um eine Ressource aus dem Resources-Ordner, einem Material Icon oder einem über eine externe URL eingebundenen Icon handeln. Mögliche Werte: • Zeichenkette (URI), z. B. <i>clock.png</i> • Angabe eines Material Icons nach dem Muster <i>icon:<MaterialIconName></i>, z. B. <i>icon:extension</i>  Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren Material Icons abweichen. • Externe URL

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" />
```

11.4.1 Beispiele

11.4.1.1 Beispiel <ComboBox>

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

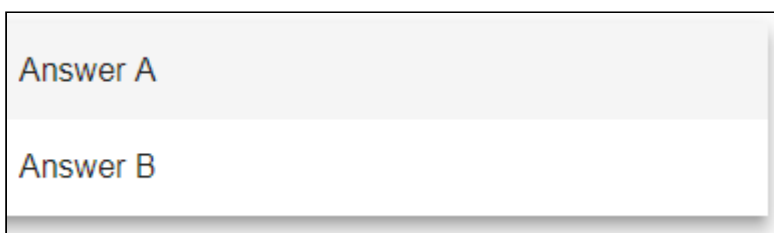
```

<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 ▾



11.4.1.2 Beispiel <ComboBox> mit ausgelagerter Optionsliste

In folgendem Beispiel ist die Verwendung des <ComboBox>-Elements 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>

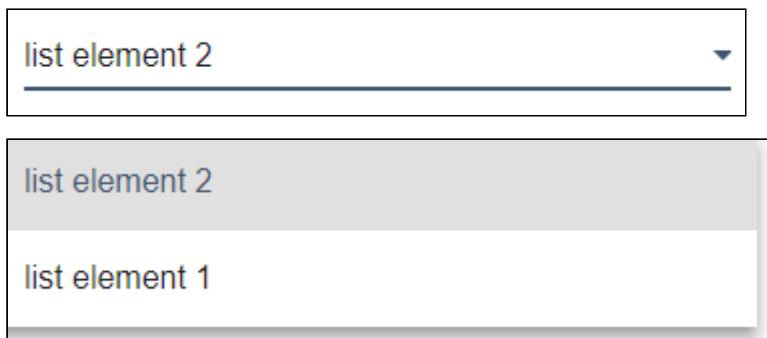
```

***.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>
```

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

Obiger Code führt zu folgender ComboBox:



The image shows a visual representation of the XML data in a ComboBox. The dropdown menu is open, displaying two items: 'list element 2' at the top and 'list element 1' below it. The text 'list element 2' is highlighted, indicating it is the selected or hovered option.

11.4.1.3 Beispiel ComboBox mit Icons

In folgendem Beispiel ist die Verwendung des <ComboBox>-Elements mit Icons und einer ausgelagerten Optionsliste gezeigt.

Folgende Optionen werden über einen Technical Process(hier Options.wrf) als Auswahloption zur Verfügung gestellt:

Optionsliste

```
<?xml version="1.0" encoding="UTF-8"?>
<OkList>
  <Country>
    <Id>6</Id>
    <Name>Worldwide</Name>
    <icon>icon:public</icon>
    <color>ff5a00</color>
  </Country>
  <Country>
    <Id>2</Id>
    <Name>Italy</Name>
    <icon>italy.png</icon>
  </Country>
  <Country>
    <Id>5</Id>
    <Name>Germany</Name>
    <icon>germany.png</icon>
  </Country>
  <Country>
    <Id>3</Id>
    <Name>Portugal</Name>
    <icon>portugal.png</icon>
  </Country>
  <Country>
    <Id>4</Id>
    <Name>Spain</Name>
    <icon>spain.png</icon>
  </Country>
  <selected>6</selected>
</OkList>
```

Die zwei beispielhaften ComboBoxen werden wie folgt definiert:

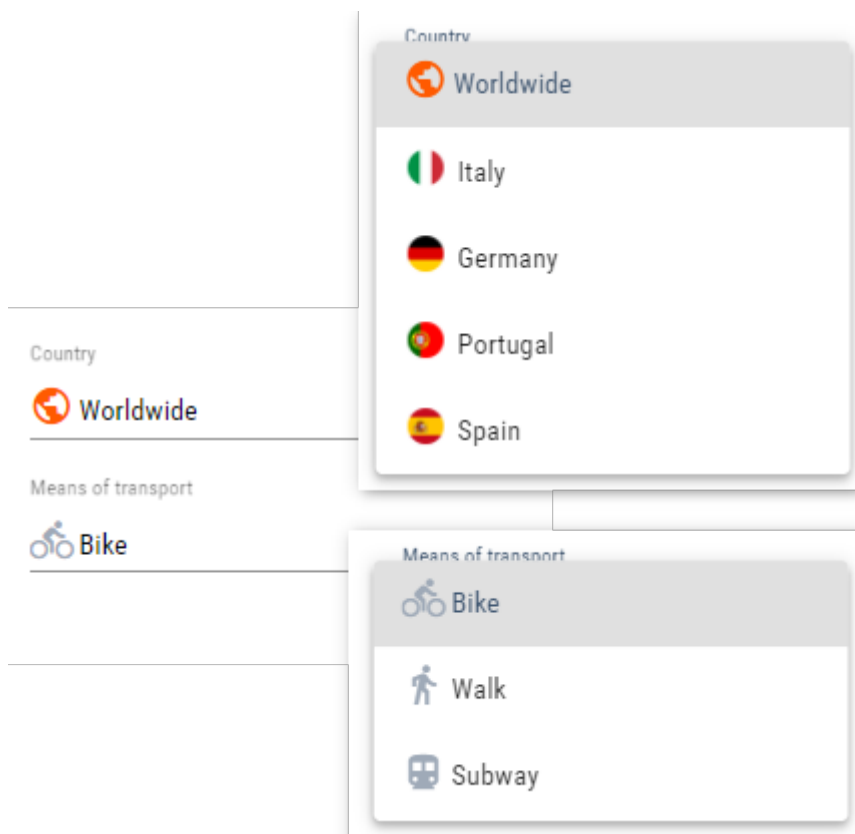
Ausschnitt aus der Definitionsdatei

```

...
<DetailComponent default="true" displayName="Dashboard" path="Dashboard"
    process="folder_1/Options.wrf">
    <Properties>
        <Property name="selected" type="Integer"/>
        <Property name="direction" type="Integer"/>
        <Property name="Country" type="List">
            <Property name="Id" type="Integer"/>
            <Property name="Name" type="String"/>
            <Property name="icon" type="Image"/>
            <Property name="color" type="Color"/>
        </Property>
        <Property name="image" type="Image"/>
    </Properties>
    <FlowLayout>
        <ComboBox displayName="$Country" displayProperty="#Name"
            iconColorProperty="#color"
            iconProperty="#icon"
            optionsList="#Country"
            value="#selected"
            valueProperty="#Id"/>
        <ComboBox displayName="Means of transport" value="#direction">
            <Option displayName="Bike" iconColor="A200" iconUrl="icon:directions_bike
" value="1"/>
            <Option displayName="Walk" iconColor="A200" iconUrl="icon:directions_walk
" value="4"/>
            <Option displayName="Subway" iconColor="A200" iconUrl="icon:directions_su
bway" value="3"/>
        </ComboBox>
    </FlowLayout>
</DetailComponent>
...

```

Das Ergebnis sieht wie folgt aus:



11.5 ComboBoxAutocomplete

<ComboBoxAutocomplete>-Controls werden verwendet, um Auswahllisten mit einer automatischen Vervollständigung zu erzeugen.

Um die Action "Select" bei Auswahl eines ComboBox-Eintrags auszulösen, kann innerhalb des ComboBox-Controls eine SelectAction-Aktion eingefügt werden.

Folgende zusätzlichen Attribute lassen sich für das Element <ComboBoxAutocomplete> definieren:

Attribut	Beschreibung
displayName	Titel der ComboBox. Wird klein über dem Ein-/Ausgabefeld angezeigt. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
optionsList	<i>Bei ausgelagerter Optionsliste:</i> Quelle der Liste. Kann nur zusammen mit displayProperty verwendet werden. • Wird per Data Binding (Liste) angesteuert. Mögliche Werte: Zeichenkette (Data Binding)

Attribut	Beschreibung
value	Wert, der angezeigt und gespeichert wird. • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
valueProperty	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. Mögliche Werte: Beliebige Zeichenkette
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Maps und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding  Das Attribut enabled ersetzt das veraltete Attribut disabled. <i>enabled="true"</i> entspricht damit dem veralteten Attribut <i>disabled="false"</i>

Attribut	Beschreibung
fontFamily	Legt die Schriftfamilie fest. i • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. i Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded i Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument .
fontStyle	Legt die Neigung der Schrift fest. i Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet) i Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal(Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

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

Das Element <Option> kann folgende Attribute haben:

Attribut	Beschreibung
iconColor	Farbe des Icons  Dieses Attribut wird nur für Material Icons verwendet. Mögliche Werte: • Hexadezimaler Farbwert, z. B. <i>ff5a00</i> Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>

Attribut	Beschreibung
iconUrl	Pfad zum verwendeten Icon  Beim Icon kann es sich entweder um eine Ressource aus dem Resources-Ordner, einem Material Icon oder einem über eine externe URL eingebundenen Icon handeln. Mögliche Werte: - Zeichenkette (URI), z. B. <i>clock.png</i> - Angabe eines Material Icons nach dem Muster <code>icon:<MaterialIconName></code>, z. B. <i>icon:extension</i>  Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren Material Icons abweichen. - Externe URL

11.5.1 Beispiel

In diesem Beispiel wird eine ausgelagerte Optionenliste verwendet. Der angezeigte Name und der zugehörige technische Wert unterscheiden sich.

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

```
<?xml version="1.0" encoding="UTF-8"?>
<Ok>
  <ComboBoxValue></ComboBoxValue>
  <OptionList>
    <OptionValue>Red</OptionValue>
    <TechnicalValue>#FF0000</TechnicalValue>
  </OptionList>
  <OptionList>
    <OptionValue>Green</OptionValue>
    <TechnicalValue>#008000</TechnicalValue>
  </OptionList>
  <OptionList>
    <OptionValue>Blue</OptionValue>
    <TechnicalValue>#0000FF</TechnicalValue>
  </OptionList>
</Ok>
```

Die Properties-Definition erfolgt entsprechend der XML-Daten.

```

<Properties>
  <Property
    name="ComboBoxValue"
    type="String"></Property>
  <Property
    name="OptionList"
    type="List">
    <Property
      name="OptionValue"
      type="String"></Property>
    <Property
      name="TechnicalValue"
      type="String"></Property>
  </Property>
</Properties>

```

Das ComboBoxAutocomplete Control in der Component enthält eine SelectAction. Die SelectAction sendet bei Auswahl einer Option aus der ComboBoxAutocomplete die Daten an den definierten Technical Process. In der ComboBoxAutocomplete werden die Daten aus der ausgelagerten Optionenliste verwendet. Angezeigt werden die Elemente OptionValue, im Hintergrund ausgewählt werden jedoch die technischen Daten aus dem Element TechnicalValue.

```

<?xml version="1.0" encoding="UTF-8"?>
<DetailComponent
  xmlns="http://softproject.de/webapp/1.0"
  process="LoadData.wrf">
  <Properties>
    <Property
      name="ComboBoxValue"
      type="String"></Property>
    <Property
      name="OptionList"
      type="List">
      <Property
        name="OptionValue"
        type="String"></Property>
      <Property
        name="TechnicalValue"
        type="String"></Property>
    </Property>
  </Properties>
  <FlowLayout>
    <Header value="Welcome to my new Web App!" />
    <ComboBoxAutocomplete
      value="#ComboBoxValue"
      optionsList="#OptionList" valueProperty="#TechnicalValue">
      <SelectAction process="SaveData.wrf" />
    </ComboBoxAutocomplete>
  </FlowLayout>
</DetailComponent>

```

11.6 DateTimePicker

<DateTimePicker>-Controls werden verwendet, um eine Datums- und Uhrzeitauswahl zu erstellen.

Folgende zusätzlichen Attribute lassen sich für das Element <DateTimePicker> definieren:

Attribut	Beschreibung
value	Wert, der angezeigt und gespeichert wird. • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
format	Format, in dem die Uhrzeit angezeigt und gespeichert wird. Mögliche Werte: • Date Time With Minutes: Uhrzeit mit Angabe der Stunden und Minuten im Format hh:mm. • Date Time With Seconds: Uhrzeit mit Angabe der Stunden, Minuten und Sekunden im Format hh:mm:ss. • Date Time With Milliseconds: Uhrzeit mit Angabe der Stunden, Minuten, Sekunden und Millisekunden im Format hh:mm:ss.sss.
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Maps und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
displayName	Titel des Controls. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette

Attribut	Beschreibung
<code>enabled</code>	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: <code>true/false</code> oder Ausdruck für Data Binding i 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>.
<code>fontFamily</code>	Legt die Schriftfamilie fest. i • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
<code>fontSize</code>	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. i Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded i Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.
fontStyle	Legt die Neigung der Schrift fest. i Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet) i Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest. i Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight i Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls. i • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00 ⚠ Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
<code>horizontalAlign</code>	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
<code>textOverflow</code>	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
<code>visible</code>	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

11.7 File Link

FileLink-Controls 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 Technical Process geladen werden.

Attribut	Beschreibung
<code>data</code>	 Nur verwenden, wenn eine Base64-Datei zum Download angeboten wird! Gibt an, welche Datei heruntergeladen werden soll. Mögliche Werte: Zeichenkette für Data Binding  Der Data-Binding-Ausdruck muss auf eine Property vom Typ <code>base64</code> verweisen!

Attribut	Beschreibung
displayName	Name der Datei, die in der Web-Anwendung angezeigt wird und als Dateiname für die heruntergeladene Datei vergeben wird. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette inkl. Dateieindung
fileName	Identifikator für den Technical Process, der die Daten liefert. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
process	Technical Process, der die Datei liefert. Mögliche Werte: Zeichenkette (URI)
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Maps und HTMLDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
fileId	Identifikator, der vom Prozessentwickler verwendet werden kann. • Data Binding möglich. Mögliche Werte: Beliebige Zeichenkette

Attribut	Beschreibung
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding i 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>
fontFamily	Legt die Schriftfamilie fest. i • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. i Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded i Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument .
fontStyle	Legt die Neigung der Schrift fest. i Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet) i Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

11.7.1 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 Technical Process process_1.wrf im Ordner Services/Processes geliefert. Sowohl eine Datei als auch ein Technical Process müssen für einen Download vorhanden sein.

Beispiel FileLink

```

<WebApp xmlns="http://softproject.de/webapp/1.0" path="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>

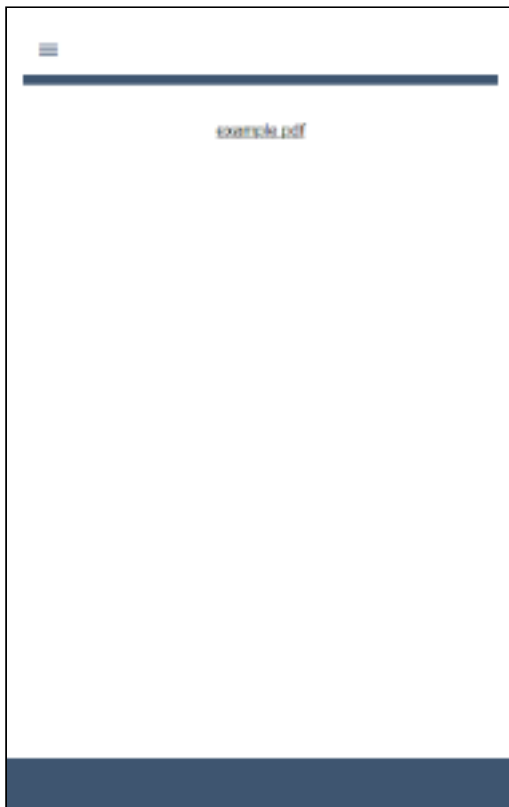
```

In obiger Web App-Definition wird der Datei-Link definiert. Für `FileLink` wird ein Technical Process hinterlegt, der die Datei liefert (`process_1.wrf`). Es wird ein Anzeigename 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 Technical Process vom Control zurückgeliefert wird:

Input 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:



11.8 File Upload

Mit dem `<FileUpload>`-Control können Dateien in eine Web-Anwendung hochgeladen werden, die mit X4 Web Apps erstellt wurde. Das `<FileUpload>`-Control kann nur in [Detail Component](#) oder [Detail-Strukturelementen](#) innerhalb von [Master/Detail Component](#) verwendet werden. Das `<FileUpload>`-Control 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 Technical Process, 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	Dateityp, nach dem im Auswahldialog gefiltert wird. • Data Binding möglich  Dieses Attribut filtert nur nach Dateitypen. Der Filter kann vom Benutzer zurückgesetzt werden, um alle Dateitypen anzuzeigen und auszuwählen. Mögliche Werte: MIME-Typen, auch mit Wildcards (*), z. B. • text/*: Textdateien • image/*: Grafikdateien • image/jpg: Nur JPG-Dateien • image/png: Nur PNG-Dateien • video/*: Videodateien • audio/*: Audiodateien • application/*: Dateien, die an ein bestimmtes Programm gebunden sind • multipart/*: mehrteilige Daten • message/*: Nachrichten • model/*: Dateien, die mehrdimensionale Strukturen repräsentieren
displayName	Anzeigename in der Oberfläche. Mögliche Werte: Beliebige Zeichenkette
fileId	Identifikator, der vom Prozessentwickler verwendet werden kann. • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
maxSizeMB	Maximale Dateigröße in Mebibyte. Mögliche Werte: Ganzzahl

Attribut	Beschreibung
process	<i>Erforderlich.</i> Technical Process, der die hochgeladene Datei entgegennimmt.  Der Technical Process muss inklusive Dateiendung angegeben werden! Mögliche Werte: Zeichenkette (URI)
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Maps und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding  Das Attribut enabled ersetzt das veraltete Attribut disabled. <code>enabled="true"</code> entspricht damit dem veralteten Attribut <code>disabled="false"</code>

Attribut	Beschreibung
fontFamily	Legt die Schriftfamilie fest. i • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded  Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument .
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

11.8.1 Input-Format für den Technical Process

Die Daten werden in einem bestimmten Input-Format an den Technical Process 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.

11.8.2 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 Control:



11.9 Header

<Header>-Controls werden verwendet, um Überschriften zu erzeugen. Ein Header kann nicht nur im oberen Teil eines Layouts platziert werden, sondern überall, wo er benötigt wird. In der folgenden Tabelle sind die Attribute erläutert, die verwendet werden können.

Attribut	Beschreibung
value	Enthält den Header-Text. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
titleLevel	Definiert die Überschriftenebene. Mögliche Werte: • <i>title</i>(Standard) • <i>subtitle</i>

Attribut	Beschreibung
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Maps und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding  Das Attribut enabled ersetzt das veraltete Attribut disabled. <i>enabled="true"</i> entspricht damit dem veralteten Attribut <i>disabled="false"</i>
fontFamily	Legt die Schriftfamilie fest.  • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut fontFamily nicht explizit gesetzt haben. • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument. Mögliche Werte: • MainFont: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. Font04

Attribut	Beschreibung
fontSize	Legt die Schriftgröße fest. i - Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. - Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument. Mögliche Werte: - Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <i>20</i>; <i>20.8</i>; <i>.9</i> - Schriftgröße in Pixel, z. B. <i>20px</i> - Schriftgröße in Punkt, z. B. <i>18pt</i> - Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i> - 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>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. i Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: - Condensed - Expanded - ExtraCondensed - ExtraExpanded - Medium - Normal (Standard) - SemiCondensed - SemiExpanded - UltraCondensed - UltraExpanded i Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.

Attribut	Beschreibung
fontStyle	Legt die Neigung der Schrift fest. i Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <code>italic</code>: kursive Schrift • <code>normal</code>: normale Schrift (Standard) • <code>oblique</code>: schräg gestellter Schriftstil (berechnet) i Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>.
fontWeight	Legt die Schriftstärke fest. i Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <code>Black</code> • <code>Bold</code> • <code>DemiBold</code> • <code>ExtraBlack</code> • <code>ExtraBold</code> • <code>ExtraLight</code> • <code>Heavy</code> • <code>Light</code> • <code>Medium</code> • <code>Normal (Standard)</code> • <code>Regular</code> • <code>SemiBold</code> • <code>Thin</code> • <code>UltraBlack</code> • <code>UltraBold</code> • <code>UltraLight</code> i Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>.

Attribut	Beschreibung
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

11.9.1 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.

11.10 HTML Document

`<HtmlDocument>`-Controls werden verwendet, um statisches HTML (Mark-up-Text) anzuzeigen, z. B. E-Mails. Es werden alle gängigen HTML5-Elemente, die vom Browser angezeigt werden können, unterstützt.

⚠ Die Anzeige von HTML ist als iFrame im Sandbox-Modus umgesetzt. Daher wird JavaScript standardmäßig nicht ausgeführt. Wenn eine eingebundene Ressource (z. B. Video, Grafik oder Audio), JavaScript enthält, wird diese Ressource nicht angezeigt.

⚠ Die Elemente `<video>` und `<audio>` werden von Safari nicht unterstützt.
Die Elemente `<video>`, `<picture>` und `<source>` werden auf einem iPad nicht angezeigt.

✓ Die Anzeige kann mit einem `<style>`-Tag im `<head>`-Element des HTML-Dokuments formatiert werden. Lokale CSS-Dateien werden nicht unterstützt.

Für das Element `<HtmlDocument>` lassen sich folgende Attribute festlegen:

Attribut	Beschreibung
process	<i>Erforderlich.</i> Technical Process, der das HTML-Dokument liefert. Mögliche Werte: Zeichenkette (URI) ⚠ Der aufgerufene Technical Process muss ein vollständiges HTML-Dokument zurückliefern!
allowSameOrigin	Erlaubt eingebetteten Inhalten, gemeinsamen Speicherplatz zu verwenden, wenn der Ursprung der eingebetteten Inhalte derselbe ist wie der Ursprung der Host-Webanwendung. Mögliche Werte: • true (Standard) • false
allowScript	Erlaubt den Inhalten, Javascript zu verwenden. Mögliche Werte: • true • false (Standard)

Attribut	Beschreibung
allowForms	Erlaubt eingebetteten Inhalten Formulare zu übermitteln. Mögliche Werte: • true • false (Standard)
allowPointerLock	Erlaubt eingebetteten Inhalten Mausbewegungen direkt als Eingabemethode zu interpretieren. Mögliche Werte: • true • false (Standard)
allowTopNavigation	Erlaubt eingebetteten Inhalten eine Webseite im selben Fenster oder Registerkarte zu öffnen. Mögliche Werte: • true • false (Standard)
allowModals	Erlaubt eingebetteten Inhalten einen modalen Dialog anzuzeigen. Mögliche Werte: • true • false (Standard)
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding Das Attribut enabled ersetzt das veraltete Attribut disabled. <i>enabled="true"</i> entspricht damit dem veralteten Attribut <i>disabled="false"</i>.
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • left (Standard) • center • right

Attribut	Beschreibung
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

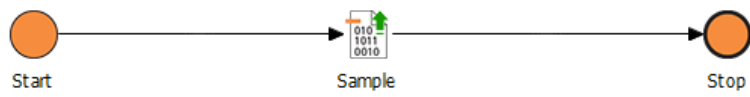
11.10.1 Beispiel für <HtmlDocument>

In der Web-App-Definition (.wad) wird das Control <HtmlDocument> mit dem Technical Process `Sample.wrf` verknüpft.

```
<WebApp xmlns="http://softproject.de/webapp/1.0"
  path="HtmlDocument">
  <Modules>
    <Module displayName="My Module" path="Module">
      <Components>
        <DetailComponent default="true"
          displayName="Dashboard"
          path="Dashboard"
          process="load.wrf">
          <Properties>
            <Property name="string" type="String"/>
          </Properties>
          <FlowLayout>
            <HTMLDocument process="Sample.wrf" units="pixels"/>
          </FlowLayout>
        </DetailComponent>
      </Components>
    </Module>
  </Modules>
</WebApp>
```

Der Technical Process `Sample.wrf` liefert ein HTML-Dokument:

Sample



Das vom Technical Process gelieferte HTML-Dokument sieht folgendermaßen aus:

```

<html>
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-strict.dtd">
  <html xmlns="http://www.w3.org/1999/xhtml">
    <head>
      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
      <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0" />
      <style>
        body{
          font-family: Arial, sans-serif;
          padding: 10px;
        }

        table, td, th{
          border: 1px solid grey;
          padding: 5px;
          border-collapse: collapse;
        }

        th{
          background-color: grey;
          color: white;
        }
      </style>
    </head>
    <body>
      <h1>This is a sample for displaying HTML content in Web Apps</h1>
      <p>Most default elements of HTML5 are supported like <i>lists</i>
(ordered and unordered), <i>horizontal lines</i> and <i>tables</i>.</p>
      <hr/>
      <h1>Samples</h1>
      <h2>Example for an ordered list</h2>
      <p>This is an example for an ordered list with a few items:</p>
      <ol>
        <li>
          Item 1
        </li>
        <li>
          Item 2
        </li>
      </ol>
      <h2>Example for an unordered list</h2>
      <p>This is an example for an unordered list with a few items:</p>
      <ul>
        <li>
          Item 1
        </li>
        <li>
          Item 2
        </li>
      </ul>
      <h2>Tables</h2>
      <p>This is an example for a table with some columns:</p>

```

```

<table>
  <thead>
    <th>Header 1</th>
    <th>Header 2</th>
    <th>Header 3</th>
    <th>Header 4</th>
  </thead>
  <tr>
    <td>A</td>
    <td>B</td>
    <td>C</td>
    <td>D</td>
  </tr>
</table>
</body>
</html>

```

Das HTML-Dokument wird in der Web App folgendermaßen angezeigt:

Hallo, [Logout](#)

Dashboard

This is a sample for displaying HTML content in Web Apps

Most default elements of HTML5 are supported like *lists* (ordered and unordered), *horizontal lines* and *tables*.

Samples

Example for an ordered list

This is an example for an ordered list with a few items:

- Item 1
- Item 2

Example for an unordered list

This is an example for an unordered list with a few items:

- Item 1
- Item 2

Tables

This is an example for a table with some columns:

Header 1	Header 2	Header 3	Header 4
A	B	C	D

[Privacy](#)
[Imprint](#)

11.11 Image

<Image>-Controls werden verwendet, um Grafiken einzubinden.

Attribut	Beschreibung
iconColor	Definiert die Farbe des Icons. Mögliche Werte: - Hexadezimaler Farbwert, z. B. <i>ff5a00</i>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>
value	Grafik, die angezeigt wird. Verweist auf eine Property (vom Typ Base64 oder Image) innerhalb der Component. Wird benötigt, um Grafiken dynamisch anzuzeigen. Mögliche Werte: Dateipfad zum Speicherort der Grafik oder Zeichenkette für Data Binding (Image, Base64, String, URL)  Wenn eine interne Grafik verwendet werden soll, dann muss die Grafikdatei im Ordner Resources direkt unterhalb des Web-App-Projekts liegen. Wenn auf die Grafik über eine externe URL zugegriffen werden soll, dann muss beim Input das Attribut type="external" gesetzt werden.
background	Definiert eine Farbe für den Hintergrund des Controls.  - Diese Einstellung überschreibt die Standardfarbe des Farbschemas! - Dieses Attribut funktioniert nicht für die Controls Maps und HtmlDocument! Mögliche Werte: - Hexadezimaler Farbwert, z. B. <i>ff5a00</i>  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>

Attribut	Beschreibung
border	Legt fest, ob um das Control ein Rahmen angezeigt wird. Mögliche Werte: <i>true / false</i>
border-left	Legt fest, ob ein linker Rand angezeigt wird Mögliche Werte: <i>true / false</i>
border-right	Legt fest, ob ein rechter Rand angezeigt wird Mögliche Werte: <i>true / false</i>
border-top	Legt fest, ob ein oberer Rand angezeigt wird Mögliche Werte: <i>true / false</i>
border-bottom	Legt fest, ob ein unterer Rand angezeigt wird Mögliche Werte: <i>true / false</i>
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: <i>true / false</i> oder Ausdruck für Data Binding  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>

Attribut	Beschreibung
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
height	Höhe des Controls Mögliche Werte: Ganzzahl
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
units	Gibt die Einheit an, die für Größenangaben gilt. Mögliche Werte: • <i>pixels</i> (Standard) • <i>percents</i>
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

Attribut	Beschreibung
width	Breite des Controls Mögliche Werte: Ganze Zahlen

✓ 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 Control noch auf dem übergeordneten Element Größenangaben definiert (über die [allgemeinen Attribute](#) height und width) kann es in manchen Browsern zu Darstellungsproblemen kommen.

11.11.1 Aktion "Auswählen" innerhalb eines Image Controls

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

11.11.2 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>
```

11.11.3 Beispiel `<Image>` mit Data Binding (dynamisch aus einem Technical Process)

11.11.3.1 Grafiken im Base64-Format

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 Technical Process `Image.wrf` geliefert werden, müssen folgendes Format haben:

```

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

```

11.11.3.2 Grafiken aus einer externen URL

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="Image" />
  </Properties>
  <FlowLayout>
    <Image value="#image"/>
  </FlowLayout>
</DetailComponent>

```

Die Daten, die vom Technical Process `Image.wrf` geliefert werden, müssen folgendes Format haben:

```

<?xml version="1.0" encoding="UTF-8" ?>
<Ok>
  <image mediaType="image/png" type="external">
    <![CDATA[https://softproject.de/de/wp-content/uploads/sites/2/2019/01/
LogoSoftProject.png]]>
  </image>
</Ok>

```

11.12 Card

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

11.12.1 Card-Control erzeugen

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

Mögliche Attribute:

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

Attribut	Beschreibung
icon	Icon für das Card Control. • Data Binding mit Base64-Werten möglich Mögliche Werte: • Base64-kodierte Grafikdatei • Pfad zu einer Grafikdatei als Zeichenkette (URI), z. B. <i>clock.png</i> • Angabe eines Material Icons, z. B. <i>icon:extension</i> ✓ Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren Material Icons abweichen. ⓘ • Die Grafikdatei muss im Ordner <code>Resources</code> direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner <code>Resources</code>. • Das Material Icon muss mit dem Präfix <code>icon</code>, z. B. <code>icon:<MaterialIconName></code> angegeben werden. ⚠ Dieses Attribut kann nicht gleichzeitig mit <code>iconUrl</code> verwendet werden.

Attribut	Beschreibung
iconColor	Definiert die Farbe des Karten-Icons. Mögliche Werte: - Hexadezimaler Farbwert, z. B. <i>ff5a00</i>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>
subtitle	Untertitel der Karte. Data Binding möglich
title	Titel der Karte. Data Binding möglich
background	Definiert eine Farbe für den Hintergrund des Controls.  - Diese Einstellung überschreibt die Standardfarbe des Farbschemas! - Dieses Attribut funktioniert nicht für die Controls Maps und HtmlDocument! Mögliche Werte: - Hexadezimaler Farbwert, z. B. <i>ff5a00</i>  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <i>A200</i>
border	Legt fest, ob um das Control ein Rahmen angezeigt wird. Mögliche Werte: <i>true</i> / <i>false</i>
border-left	Legt fest, ob ein linker Rand angezeigt wird Mögliche Werte: <i>true</i> / <i>false</i>
border-right	Legt fest, ob ein rechter Rand angezeigt wird Mögliche Werte: <i>true</i> / <i>false</i>

Attribut	Beschreibung
border-top	Legt fest, ob ein oberer Rand angezeigt wird Mögliche Werte: <i>true</i> / <i>false</i>
border-bottom	Legt fest, ob ein unterer Rand angezeigt wird Mögliche Werte: <i>true</i> / <i>false</i>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded  Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)  • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

11.12.2 Aktionen

Innerhalb des Elements `<Card>` können **Aktionen** definiert werden.

- i** Wenn eine **Select** Action (`<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>

```

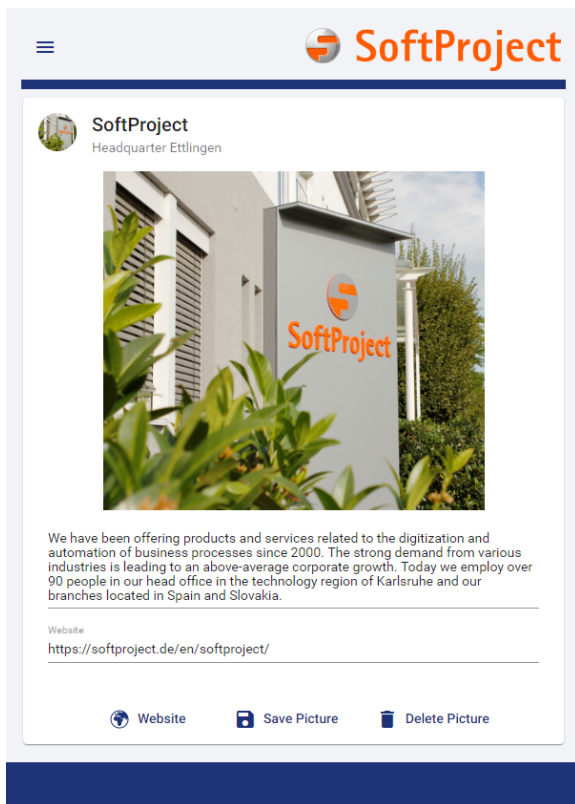
11.12.3 Beispiele

In diesem Beispiel werden die Attribute sowie die Inhalte des Card Control 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:



11.13 Label

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

Attribut	Beschreibung
value	Enthält den Label-Text • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Maps und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding  Das Attribut enabled ersetzt das veraltete Attribut disabled. <i>enabled="true"</i> entspricht damit dem veralteten Attribut <i>disabled="false"</i>

Attribut	Beschreibung
fontFamily	Legt die Schriftfamilie fest. i • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded  Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal(Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

11.13.1 Beispiel <Label>

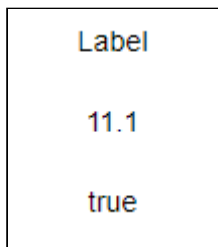
Im folgenden Beispiel ist die Verwendung des <Label>-Tags gezeigt. In diesem Beispiel werden die Labels mittels Data Binding befüllt.

```
<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 Technical Process 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:



11.14 Link

<Link>-Controls werden verwendet, um Hyperlinks zu erzeugen. Der Hyperlink kann in der Web App frei platziert werden.

❗ Das Element <Link> kann folgende Aktionen enthalten:

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

Folgende zusätzlichen Attribute lassen sich für das Element <Link> definieren:

Attribut	Beschreibung
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • left (Standard) • center • right

Attribut	Beschreibung
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • ellipsis: Mit ... zeigen, dass der Text nicht zu Ende ist • hidden: Text abbrechen, auf ganze Wörter achten • wordBreak: innerhalb des Wortes abbrechen • allow(Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true / false</i> oder Zeichenkette für Data Binding

11.14.1 Beispiel

```

1  <DetailComponent
2      default="true"
3      path="Dashboard">
4      <GridLayout>
5          <Link>
6              <CustomAction
7                  externalLink="https://www.softproject.de"
8                  externalLinkTarget="new"
9                  displayName="Open website" />
10         </Link>
11     </GridLayout>
12 </DetailComponent>

```

11.15 ListView

ListView-Controls 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 Technical Process geliefert werden, muss in der Component, die die ListView enthält, ein entsprechender Technical Process definiert sein.

Attribut	Beschreibung
elementWidth	Gibt die Breite eines Elements (entspricht einem Listeneintrag in der XML-Datei) an. Mögliche Werte: Ganzzahl

Attribut	Beschreibung
elementWidthUnits	Gibt die Einheit an, die für Größenangabe des Elements gilt. Mögliche Werte: • <i>pixels</i> • <i>percents</i>(Standard)
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. Mögliche Werte: Ganzzahl
objectList	<i>Erforderlich.</i> Quelle der Daten. Die Anzahl der Einträge definiert die Anzahl der angezeigten Elemente. • Data Binding (Liste) möglich Mögliche Werte: Beliebige Zeichenkette
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Maps und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding  Das Attribut enabled ersetzt das veraltete Attribut disabled. <i>enabled="true"</i> entspricht damit dem veralteten Attribut <i>disabled="false"</i>

Attribut	Beschreibung
fontFamily	Legt die Schriftfamilie fest. i • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded  Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal(Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

11.15.1 Select Action innerhalb einer ListView

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

 Das ausgewählte Element wird im Output mit dem Attribut `selected="true"` gekennzeichnet.

11.15.2 Beispiele

11.15.2.1 Liste ohne Aktionen

Input

```

<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">
      <FlowLayout>
        <Label value="#Caption"/>
        <Label value="#Name"/>
        <Label value="#City"/>
      </FlowLayout>
    </ListView>
  </FlowLayout>
</DetailComponent>

```

Output

Die Daten, die vom Technical Process geliefert werden, haben folgendes Format:

```

<?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

11.15.2.2 Liste mit Aktionen

Input

```

<DetailComponent
  path="Dashboard"
  displayName="Dashboard"
  process="DataInput.wrf"
  default="true">
  <Properties>
    <Property
      name="ListObject"
      type="List">
      <Property
        name="name"
        type="String" />
    </Property>
  </Properties>
  <FlowLayout>
    <ListView objectList="#ListObject">
      <SelectAction process="Response.wrf" />
      <FlowLayout>
        <TextBox value="#name" />
      </FlowLayout>
    </ListView>
  </FlowLayout>
</DetailComponent>

```

Output

Die Daten, die vom Technical Process geliefert werden, haben folgendes Format:

- ① Das ausgewählte Element wird im Output mit dem Attribut `selected="true"` gekennzeichnet.
Damit alle Einträge der ListView vollständig gelöscht werden können, muss im Output das Attribut `empty="true"` gesetzt werden.

```
<?xml version="1.0" encoding="UTF-8"?>
<Action>
  <ListObject selected="true">
    <name>Ettlingen</name>
  </ListObject>
  <ListObject>
    <name>New York</name>
  </ListObject>
  <ListObject>
    <name>Oslo</name>
  </ListObject>
</Action>
```

11.16 Maps

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

11.16.1 Maps Controls erzeugen

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

Mögliche Attribute:

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

Attribut	Beschreibung
apiKey	Gibt den API-Schlüssel an, falls dieser vom Kartenanbieter verlangt wird. Mögliche Werte: API-Schlüssel  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.  Wenn Google Maps als Kartenanbieter verwendet wird, müssen folgende APIs aktiviert sein: • Maps JS API • Locations (für die Suche und die Routenberechnung) • Directions (für die Routenberechnung)  Das Attribut apiKey überschreibt das Element MapAPIKey in der Web App-Konfiguration .wac.
initialPointLongitude	Definiert den Längengrad, auf den die Karte zentriert sein soll, wenn sie geladen wird. • Data Binding mit konstanten Werten möglich Mögliche Werte: Angabe eines Längengrads in Grad mit Dezimalstellen (z. B. 6.75000)
initialPointLatitude	Definiert den Längengrad, auf den die Karte zentriert sein soll, wenn sie geladen wird. • Data Binding mit konstanten Werten möglich Mögliche Werte: Angabe eines Breitengrads in Grad mit Dezimalstellen (z. B. 50.41871)
restrictScrolling	Scrolling-Verhalten in Maps einschränken Mögliche Werte: • true (Standard): Beim Scrollen mit der Maus bzw. mit dem Finger auf mobilen Geräten wird eine Meldung eingeblendet, die über das alternative Scrollen informiert • false: alternatives Scrolling-Verhalten ist deaktiviert

Attribut	Beschreibung
satelliteView	Legt fest, ob die Satellitenansicht der Karte aktiviert ist. Mögliche Werte: true / false (Standard)  Die Satellitenansicht kann nicht verwendet werden, wenn OpenStreetMap als Kartenanbieter verwendet wird.
selectedIndex	Ausgewählter Index  Das Attribut selectedIndex übernimmt den index-Wert eines einfachen Markers (SingleMarker) oder eine Markergruppe (Markers), sobald diese ausgewählt wurden. Mögliche Werte: String Binding
vendor	<i>Erforderlich.</i> Spezifiziert den Kartenanbieter. Mögliche Werte: • OpenStreetMap • Google • Bing

Attribut	Beschreibung
zoomLevel	Spezifiziert die initiale Zoomstufe der Karte. i Die Werte geben die in der Karte angezeigten Meter ungefähr an. Beachten Sie, dass es beim Maßstabsbalken zu Abweichungen kommen kann, da sich der Maßstab in Abhängigkeit von der Entfernung vom Äquator ändert. Mögliche Werte: • 10 • 25 • 50 • 100 • 250 • 500 • 750 • 1000 • 2000 • 5000 • 10000 • 25000 • 50000 • 100000 • 2500000 • 5000000
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding i 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>
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>

Attribut	Beschreibung
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

11.16.2 Suchfunktion

Um einem Web App-Benutzer die Suche nach Orten im Maps-Control 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	Definiert die horizontale Position des Icons relativ zu den gesuchten Koordinaten. Mögliche Werte: • <i>left</i>: Icon ist links von der Koordinate • <i>center</i> (Standard): Icon ist auf der Koordinate • <i>right</i>: Icon ist rechts von der Koordinate
searchIconUrl	Pfad zur Grafikdatei, die als Icon verwendet werden soll. Mit dem Icon wird das Suchergebnis markiert. Die Grafikdatei muss im Ordner <code>Resources</code> direkt unterhalb des Web App-Projekts enthalten sein. Pfadangabe relativ zum Ordner <code>Resources</code> . Mögliche Werte: Zeichenkette (URI)

Attribut	Beschreibung
searchIconVerticalPosition	Definiert die vertikale Position des Icons relativ zu den gesuchten Koordinaten. Mögliche Werte: • top: Icon ist über der Koordinate • center (Standard): Icon ist auf der Koordinate • bottom: Icon ist unter der Koordinate

11.16.3 Routenberechnung

Um einem Web App-Benutzer die Berechnung einer Route zwischen zwei Orten im Maps-Control 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. Mögliche Werte: • left: Icon ist links von der Koordinate • center (Standard): Icon ist auf der Koordinate • right: Icon ist rechts von der Koordinate
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 <code>Resources</code> direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner <code>Resources</code>. Mögliche Werte: Zeichenkette (URI)
destinationIconVerticalPosition	Definiert die vertikale Position des Zielicons relativ zu den gesuchten Koordinaten. Mögliche Werte: • top: Icon ist über der Koordinate • center (Standard): Icon ist auf der Koordinate • bottom: Icon ist unter der Koordinate

Attribut	Beschreibung
startIconHorizontalPosition	Definiert die horizontale Position des Starticons relativ zu den gesuchten Koordinaten. Mögliche Werte: • left: Icon ist links von der Koordinate • center (Standard): Icon ist auf der Koordinate • right: Icon ist rechts von der Koordinate
startIconUrl	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 <code>Resources</code> direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner <code>Resources</code>. Mögliche Werte: Zeichenkette (URI)
startIconVerticalPosition	Definiert die vertikale Position des Starticons relativ zu den gesuchten Koordinaten. Mögliche Werte: • top: Icon ist über der Koordinate • center (Standard): Icon ist auf der Koordinate • bottom: Icon ist unter der Koordinate

11.16.4 Einfacher Marker

Um einen einfachen Marker in einem Maps-Control 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	Gibt den Namen der Component für eine interne Navigation an. Mögliche Werte: Zeichenkette (Name einer Component)
description	Gibt die Beschreibung des Markers an. • Data Binding möglich Mögliche Werte: Zeichenkette inkl. Leerzeichen

Attribut	Beschreibung
externalLink	Gibt den Link an, der im Pop-up des Markers angezeigt wird. • Data Binding möglich Mögliche Werte: Zeichenkette (URL)
externalLinkTarget	Gibt an, ob der Link (externalLink) im selben oder in einem neuen Tab geöffnet wird. Mögliche Werte: • same: Der Link wird im selben Tab geöffnet • new: Der Link wird in einem neuen Tab geöffnet
iconHorizontalPosition	Definiert die horizontale Position des Icons relativ zu den eingegebenen Koordinaten. Mögliche Werte: • left: Icon ist links von der Koordinate • center (Standard): Icon ist auf der Koordinate • right: Icon ist rechts von der Koordinate
iconUrl	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. Mögliche Werte: Zeichenkette (URI)
iconVerticalPosition	Definiert die vertikale Position des Icons relativ zu den eingegebenen Koordinaten. Mögliche Werte: • top: Icon ist über der Koordinate • center (Standard): Icon ist auf der Koordinate • bottom: Icon ist unter der Koordinate
index	Index des Markers • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette

Attribut	Beschreibung
latitude	<i>Erforderlich, wenn searchString nicht verwendet wird.</i> Gibt den Breitengrad des Markers an. • Data Binding möglich Mögliche Werte: Angabe eines Breitengrads in Grad mit Dezimalstellen (z. B. 50.41871)  Dieses Attribut kann nicht gleichzeitig mit searchString verwendet werden.
longitude	<i>Erforderlich, wenn searchString nicht verwendet wird.</i> Gibt den Längengrad des Markers an. • Data Binding möglich Mögliche Werte: Angabe eines Längengrads in Grad mit Dezimalstellen (z. B. 6.75000)  Dieses Attribut kann nicht gleichzeitig mit searchString verwendet werden.
name	Gibt den Namen des Markers an. • Data Binding möglich Mögliche Werte: Zeichenkette aus alphanumerischen Zeichen
navigationDisplayName	Gibt den Text an, der im Pop-up des Markers statt eines externen Links oder eines Componentnamens angezeigt wird. • Data Binding möglich Mögliche Werte: Zeichenkette inkl. Leerzeichen
searchString	<i>Erforderlich, wenn latitude/longitude nicht verwendet wird.</i> Adresszeile zur Suche nach ausgewählten Diensten des Kartenanbieters. • Data Binding möglich Mögliche Werte: Zeichenkette für eine Dienstsuche, z. B. Restaurant  Dieses Attribut kann nicht gleichzeitig mit den Attributen latitude/longitude verwendet werden.

11.16.5 Markergruppe

Um mehrere Marker in einem Maps-Control 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	Gibt an, ob Marker zusammengefasst werden sollen, wenn sie auf der Karte sehr dicht beieinander liegen. Mögliche Werte: true / false (Standard)  cluster und number können nicht gleichzeitig verwendet werden!
componentName	Gibt den Namen der Component für eine interne Navigation an. Mögliche Werte: Zeichenkette (Name einer Component)
data	<i>Erforderlich.</i> Definiert, welche Daten für die Markergruppe verwendet werden. • Data Binding erforderlich Mögliche Werte: Data-Binding-Ausdruck
description	Gibt die Beschreibung der Marker an. • Data Binding erforderlich Mögliche Werte: Data-Binding-Ausdruck relativ zum Ausdruck in data
externalLink	Gibt den Link an, der im Pop-up des Markers angezeigt wird. • Data Binding erforderlich Mögliche Werte: Data-Binding-Ausdruck relativ zum Ausdruck in data

Attribut	Beschreibung
externalLinkTarget	Gibt an, ob der Link (externalLink) im selben oder in einem neuen Tab geöffnet wird. Mögliche Werte: • same: Der Link wird im selben Tab geöffnet • new: Der Link wird in einem neuen Tab geöffnet
iconHorizontalPosition	Definiert die horizontale Position des Icons relativ zu den eingegebenen Koordinaten. Mögliche Werte: • left: Icon ist links von der Koordinate • center (Standard): Icon ist auf der Koordinate • right: Icon ist rechts von der Koordinate
iconUrl	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. Mögliche Werte: Zeichenkette (URI)
iconVerticalPosition	Definiert die vertikale Position des Icons relativ zu den eingegebenen Koordinaten. Mögliche Werte: • top: Icon ist über der Koordinate • center (Standard): Icon ist auf der Koordinate • bottom: Icon ist unter der Koordinate
index	Index der Markergruppe • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
latitude	<i>Erforderlich, wenn searchString nicht verwendet wird.</i> Gibt den Breitengrad eines Markers an. • Data Binding möglich Mögliche Werte: Data-Binding-Ausdruck relativ zum Ausdruck in data
longitude	<i>Erforderlich, wenn searchString nicht verwendet wird.</i> Gibt den Längengrad eines Markers an. • Data Binding möglich Mögliche Werte: Data-Binding-Ausdruck relativ zum Ausdruck in data

Attribut	Beschreibung
name	Gibt den Namen der Marker an. • Data Binding erforderlich Mögliche Werte: Data-Binding-Ausdruck relativ zum Ausdruck in data
navigationDisplayName	Gibt den Text an, der im Pop-up des Markers statt eines externen Links oder eines Componentnamens angezeigt wird. • Data Binding möglich Mögliche Werte: Zeichenkette inkl. Leerzeichen
number	Gibt an, ob die Marker nummeriert werden sollen. Mögliche Werte: true / false (Standard)  cluster und number können nicht gleichzeitig verwendet werden!
searchString	<i>Erforderlich, wenn latitude/longitude nicht verwendet wird.</i> Adresszeile zur Suche nach ausgewählten Diensten des Kartenanbieters. • Data Binding möglich Mögliche Werte: Zeichenkette für eine Dienstsuche, z. B. Restaurant  Dieses Attribut kann nicht gleichzeitig mit den Attributen latitude/longitude verwendet werden.
textHorizontalPosition	Definiert die horizontale Position der Nummerierung relativ zu den eingegebenen Koordinaten. Mögliche Werte: • left: Nummerierung ist links von der Koordinate (Standard) • center: Nummerierung ist auf der Koordinate • right: Nummerierung ist rechts von der Koordinate  Bing Maps positioniert die Nummerierung automatisch. Daher hat diese Funktion auf Bing Maps keinen Einfluss.

Attribut	Beschreibung
textVerticalPosition	Definiert die vertikale Position der Nummerierung relativ zu den eingegebenen Koordinaten. Mögliche Werte: • top: Nummerierung ist über der Koordinate (Standard) • center: Nummerierung ist auf der Koordinate • bottom: Nummerierung ist unter der Koordinate  Bing Maps positioniert die Nummerierung automatisch. Daher hat diese Funktion auf Bing Maps keinen Einfluss.

11.16.6 Aktion "Auswählen" innerhalb einer Map

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

11.16.7 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>

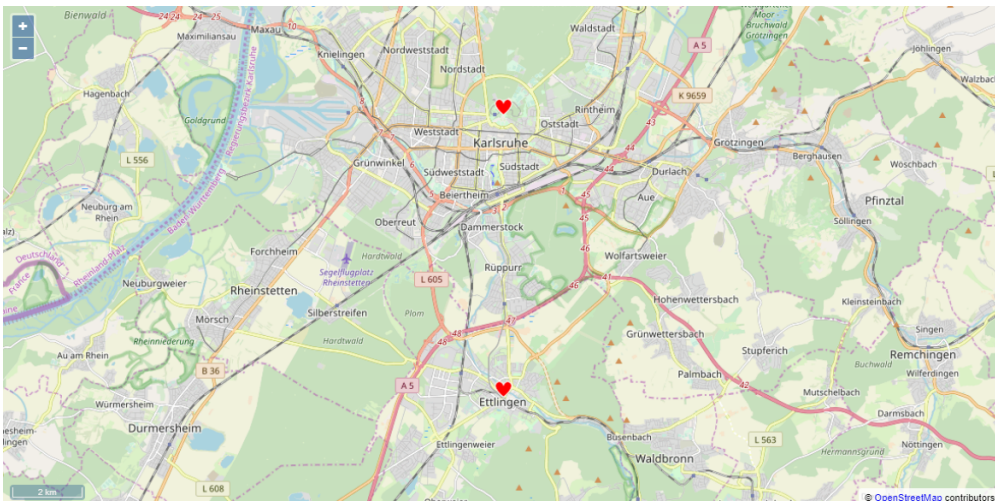
```

Beispiel 1: Durch den Technical Process 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 Technical Process gelieferte Daten (*.xml)

```

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

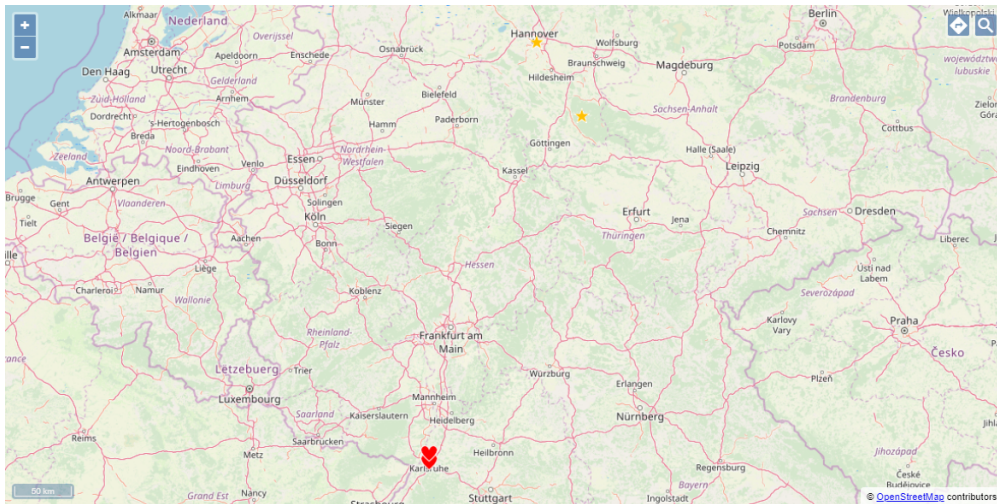
  <favorites>
    <lat>49.01396</lat>
    <lng>8.40445</lng>
  </favorites>

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

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

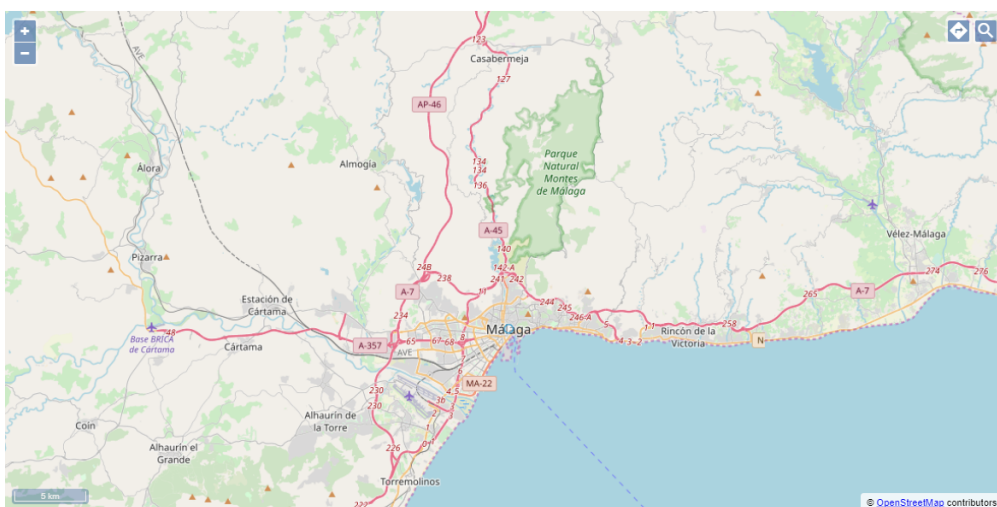
```

Obiges Beispiel führt zu folgendem Ergebnis:



Mehrere Gruppen von Markern mit verschiedenen Symbolen (Openstreetmap)

Suche
Malaga, Andalusien, Spanien|



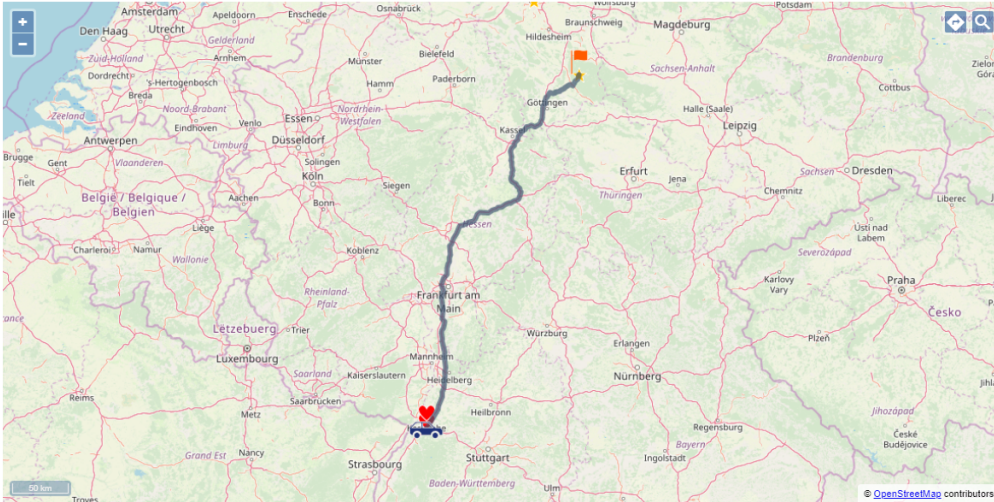
Suchfunktion (Openstreetmap)

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-Control 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 Technical Process gelieferte Daten (*.xml)

```

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

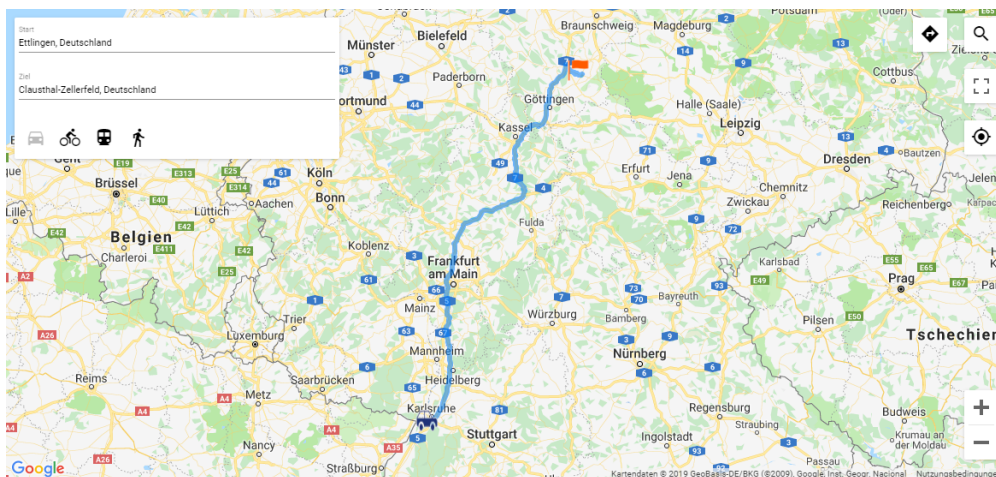
  <favorites>
    <lat>49.01396</lat>
    <lng>8.40445</lng>
  </favorites>

  <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:



11.16.8 Marker per Klick in der Karte setzen

Das Element `<SelectedLocationMarker>` ermöglicht dem Benutzer einer Web App, eine Markierung zu setzen, in dem er in der Karte auf eine bestimmte Stelle klickt. Die vom Benutzer markierte Stelle kann durch einen Technical Process weiterverarbeitet werden.

```

<Map>
  ...
  <SelectedLocationMarker .../>
</Map>

```

Mögliche Attribute:

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

Attribut	Beschreibung
longitude	<i>Erforderlich.</i> Längengrad der Markierung Mögliche Werte: Property ⚠ Als Wert für das Attribut muss eine Property hinterlegt werden.
latitude	<i>Erforderlich.</i> Breitengrad der Markierung Mögliche Werte: Property ⚠ Als Wert für das Attribut muss eine Property hinterlegt werden.
iconUrl	Pfad zur Grafikdatei, die als Icon verwendet werden soll. Mit dem Icon wird der gewählte Ort markiert. Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources. Mögliche Werte: Zeichenkette (URI)

11.16.9 Standort des Benutzers verwenden

Innerhalb des Elements `<Map>` kann das Element `<Geolocation>` verwendet werden, um den Standort des Benutzers abzufragen. Wenn das Element `<Geolocation>` verwendet wird, dann wird in der Karte ein Button angezeigt. Per Klick auf den Button wird der Standort des Benutzers abgefragt und auf der Karte markiert.

Mögliche Attribute:

Attribut	Beschreibung
iconUrl	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. Mögliche Werte: Zeichenkette (URI)
iconHorizontalPosition	Horizontale Position des Icons
iconVerticalPosition	Verticale Position des Icons

11.17 PasswordBox

<PasswordBox>-Controls werden verwendet, um ein Passwort-Textfeld zu erzeugen. Je nach Definition des Controls ändert sich das Aussehen.

Für das Element <PasswordBox> lassen sich folgende Attribute festlegen:

Attribut	Beschreibung
displayName	Titel der PasswordBox. Wird klein über dem Ein-/Ausgabefeld angezeigt. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
value	Inhalt der PasswordBox. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Maps und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
displayToggle	Legt fest, ob das Passwortfeld umgeschaltet werden kann, um das Passwort anzuzeigen. Mögliche Werte: • true: Das Passwortfeld kann umgeschaltet werden, um das Passwort anzuzeigen. • false: Das Passwortfeld kann nicht umgeschaltet werden. Das Passwort bleibt verborgen.

Attribut	Beschreibung
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding i 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>
fontFamily	Legt die Schriftfamilie fest. i • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • <code>MainFont</code>: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <code>Font04</code>
fontSize	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <code>20</code>; <code>20.8</code>; <code>.9</code> • Schriftgröße in Pixel, z. B. <code>20px</code> • Schriftgröße in Punkt, z. B. <code>18pt</code> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <code>.8em</code> oder <code>120%</code> • 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>

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest.  Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded  Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.
fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.

Attribut	Beschreibung
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal(Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight  Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument.
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200

Attribut	Beschreibung
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

11.18 Radio Button

<RadioButton>-Controls 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 • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette
optionValue	<i>Erforderlich.</i> Technischer Wert, der weiterverarbeitet werden soll. • Data Binding (String oder Boolean) möglich Mögliche Werte: Beliebige Zeichenkette
value	<i>Erforderlich.</i> Radio-Buttons mit gleichem <code>value</code> bilden eine Gruppe, von der jeweils nur ein Radio-Button ausgewählt werden kann. • Data Binding (String oder Boolean) erforderlich Mögliche Werte: Beliebige Zeichenkette

Attribut	Beschreibung
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Maps und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding  Das Attribut enabled ersetzt das veraltete Attribut disabled. <i>enabled="true"</i> entspricht damit dem veralteten Attribut <i>disabled="false"</i>
fontFamily	Legt die Schriftfamilie fest.  • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut fontFamily nicht explizit gesetzt haben. • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument. Mögliche Werte: • MainFont: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. Font04

Attribut	Beschreibung
fontSize	Legt die Schriftgröße fest. i - Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. - Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument. Mögliche Werte: - Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. 20; 20.8; .9 - Schriftgröße in Pixel, z. B. 20px - Schriftgröße in Punkt, z. B. 18pt - Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. .8em oder 120% - Schlüsselwörter: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. i Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: - Condensed - Expanded - ExtraCondensed - ExtraExpanded - Medium - Normal (Standard) - SemiCondensed - SemiExpanded - UltraCondensed - UltraExpanded i Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.

Attribut	Beschreibung
fontStyle	Legt die Neigung der Schrift fest. i Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <code>italic</code>: kursive Schrift • <code>normal</code>: normale Schrift (Standard) • <code>oblique</code>: schräg gestellter Schriftstil (berechnet) i Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>.
fontWeight	Legt die Schriftstärke fest. i Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <code>Black</code> • <code>Bold</code> • <code>DemiBold</code> • <code>ExtraBlack</code> • <code>ExtraBold</code> • <code>ExtraLight</code> • <code>Heavy</code> • <code>Light</code> • <code>Medium</code> • <code>Normal</code> (Standard) • <code>Regular</code> • <code>SemiBold</code> • <code>Thin</code> • <code>UltraBlack</code> • <code>UltraBold</code> • <code>UltraLight</code> i Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>.

Attribut	Beschreibung
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

11.18.1 Aktion "Auswählen" innerhalb eines Radio-Buttons

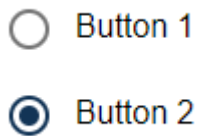
Die Aktion **Auswählen** wird über das Element `<SelectAction/>` innerhalb von `<RadioButton>` erzeugt und ausgelöst, sobald dieser ausgewählt wurde.

11.18.2 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:



11.19 TextBlock

<TextBlock>-Controls werden verwendet, um Textblöcke innerhalb einer Web-Anwendung zu formatieren.

Das TextBlock-Control wird über die allgemeinen [Attribute für Controls](#) definiert.

Das Element <TextBlock> kann folgende Elemente enthalten:

- Freitext
- <Paragraph>
- <Binding>
- <Break>
-
- <Translation>
- <Link>

11.19.1 Paragraph

Das Element <Paragraph> wird verwendet, um Texte in verschiedene Absätze zu unterteilen und kann folgende Elemente enthalten:

- Freitext
- <Binding>
- <Break>
-
- <Translation>

- <Link>

Ein Paragraph kann über folgende Attribute definiert werden:

Attribut	Beschreibung
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für das Control Maps! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
fontFamily	Legt die Schriftfamilie fest.  • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • 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. • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument. Mögliche Werte: • MainFont: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. <i>Font04</i>

fontSize	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <i>20</i>; <i>20.8</i>; <i>.9</i> • Schriftgröße in Pixel, z. B. <i>20px</i> • Schriftgröße in Punkt, z. B. <i>18pt</i> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i> • 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>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. i Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded i Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.

fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <code>italic</code>: kursive Schrift • <code>normal</code>: normale Schrift (Standard) • <code>oblique</code>: schräg gestellter Schriftstil (berechnet)  • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>.
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <code>Black</code> • <code>Bold</code> • <code>DemiBold</code> • <code>ExtraBlack</code> • <code>ExtraBold</code> • <code>ExtraLight</code> • <code>Heavy</code> • <code>Light</code> • <code>Medium</code> • <code>Normal (Standard)</code> • <code>Regular</code> • <code>SemiBold</code> • <code>Thin</code> • <code>UltraBlack</code> • <code>UltraBold</code> • <code>UltraLight</code>  Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>.

foreground	Definiert die Schriftfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
justifyText	Legt die Textausrichtung innerhalb des Textblocks fest. Mögliche Werte: - justify (Standard): Blocksatz - center: Zentriert - left: Linksbündig - right: Rechtsbündig
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: - ellipsis (Standard): Mit ... zeigen, dass der Text nicht zu Ende ist - hidden: Text abbrechen, auf ganze Wörter achten - wordBreak: innerhalb des Wortes abbrechen - allow: Text zwischen den Wörtern umbrechen

Beispiel

```
<Paragraph justifyText="justify" textOverflow="allow">
  Paragraph
  <Binding value="#test" />
</Paragraph>
```

11.19.2 Span

Das Element `<Span>` wird verwendet, um Textabschnitte oder einzelne Textelemente zu formatieren und kann folgende Elemente enthalten:

- Freitext
- `<Binding>`
- `<Break>`
- `<Span>`

- <Translation>
- <Link>

Das Element Span kann über folgende Attribute definiert werden:

Attribut	Beschreibung
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für das Control Maps! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
fontFamily	Legt die Schriftfamilie fest.  • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut fontFamily nicht explizit gesetzt haben. • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument. Mögliche Werte: • MainFont: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. Font04

fontSize	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. <i>20</i>; <i>20.8</i>; <i>.9</i> • Schriftgröße in Pixel, z. B. <i>20px</i> • Schriftgröße in Punkt, z. B. <i>18pt</i> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i> • 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>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. i Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded i Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.

fontStyle	Legt die Neigung der Schrift fest.  Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <code>italic</code>: kursive Schrift • <code>normal</code>: normale Schrift (Standard) • <code>oblique</code>: schräg gestellter Schriftstil (berechnet)  • Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>.
fontWeight	Legt die Schriftstärke fest.  Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <code>Black</code> • <code>Bold</code> • <code>DemiBold</code> • <code>ExtraBlack</code> • <code>ExtraBold</code> • <code>ExtraLight</code> • <code>Heavy</code> • <code>Light</code> • <code>Medium</code> • <code>Normal (Standard)</code> • <code>Regular</code> • <code>SemiBold</code> • <code>Thin</code> • <code>UltraBlack</code> • <code>UltraBold</code> • <code>UltraLight</code>  Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>.

foreground	Definiert die Schriftfarbe.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: - Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! - Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
------------	--

Beispiel

```
<Span fontWeight="ExtraBold" foreground="GaugeAxisMiddle">
  Span
    <Binding value="#test" />
</Span>
```

11.19.3 Break

Das Element `<Break>` wird verwendet, um Zeilenumbrüche innerhalb eines Textabschnitts einzufügen.

Beispiel

```
<Paragraph>
  SoftProject GmbH
  <Break/>
  Am Erlengraben 3
  <Break/>
  D-76275 Ettlingen
  <Break />
  E-Mail: info@softproject.de
</Paragraph>
```

11.19.4 Binding

Das Element `<Binding>` wird für [Data Binding](#) verwendet.

Das Element `Binding` wird über folgendes Attribut definiert:

Attribut	Beschreibung
value	Bindet eine gültige Data-Binding-Zeichenkette.  Beachten Sie, dass keine Konstanten erlaubt sind.

Beispiel

```
<Span>
  Span with binding
  <Break />
  value = '<Binding value="#test" />'
</Span>
```

11.19.5 Translation

Das Element `<Translation>` wird verwendet, um für mehrsprachige Oberflächen einen Übersetzungsschlüssel zu hinterlegen.

Das Element `Translation` wird über folgendes Attribut definiert:

Attribut	Beschreibung
value	Gibt den Übersetzungsschlüssel an.

Beispiel

```
<Span>
  Span with translation
  <Break />
  value = '<Translation value="$TranslationKey" />'
</Span>
```

11.19.6 Link

Das Element `<Link>` wird verwendet, um eine Verlinkung zu erstellen. Um die Verlinkung zu generieren, muss eine Action definiert werden.

Folgende Actions werden dabei unterstützt:

- Upload Action
- Download Action
- Select Action
- CustomAction

Beispiel

```

<?xml version="1.0" encoding="UTF-8"?>
<DetailComponent xmlns="http://softproject.de/webapp/1.0">
  <FlowLayout>
    <Header value="Welcome to my new Web App!" />
    <TextBlock>
      <Link>
        <SelectAction
          process="myProcess.wrf"
          displayName="Click here"></SelectAction>
      </Link>
    </TextBlock>
  </FlowLayout>
</DetailComponent>

```

11.20 TextBox

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

Um die Action "Select" nach dem Abwählen eines TextBox-Controls aufzurufen, kann innerhalb des TextBox-Controls eine **SelectAction-Aktion** eingefügt werden.

Für das Element <TextBox> lassen sich folgende Attribute festlegen:

Attribut	Beschreibung
border	Legt fest, ob um das Control ein Rahmen angezeigt wird. Funktioniert nur, wenn multiline="true". Mögliche Werte: <i>true / false</i>
border-left	Legt fest, ob ein linker Rand angezeigt wird. Funktioniert nur, wenn multiline="true". Mögliche Werte: <i>true / false</i>
border-right	Legt fest, ob ein rechter Rand angezeigt wird. Funktioniert nur, wenn multiline="true". Mögliche Werte: <i>true / false</i>
border-top	Legt fest, ob ein oberer Rand angezeigt wird. Funktioniert nur, wenn multiline="true". Mögliche Werte: <i>true / false</i>
border-bottom	Legt fest, ob ein unterer Rand angezeigt wird. Funktioniert nur, wenn multiline="true". Mögliche Werte: <i>true / false</i>

Attribut	Beschreibung
iconUrl	Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Icon verwendet wird. • Data Binding möglich i • Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources. • Das Material Icon muss mit dem Präfix <code>icon:</code>, z. B. <code>icon:<MaterialIconName></code> angegeben werden. Mögliche Werte: • Zeichenkette (URI), z. B. <code>clock.png</code> • <code>icon:<MaterialIconName></code>, z. B. <code>icon:extension</code> ✓ Mit <code>Strg+Leertaste</code> erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren Material Icons abweichen.
multiline	Legt fest, ob in der TextBox mehrzeilige Ein-/Ausgaben möglich sind. Funktioniert nur beim Typ <code>text</code>. Mögliche Werte: <code>true</code> / <code>false</code>
rows	Legt die sichtbare Höhe des Textfeldes fest. Funktioniert nur bei mehrzeiligen Textfeldern. Es können mehr Zeilen eingegeben werden, diese sind dann über Scrollen sichtbar. Mögliche Werte: Ganze Zahlen (Standard: 5)
showSeconds	Legt fest, ob bei Textboxen des Typs <code>datetime</code> die Sekunden ebenfalls angezeigt werden sollen Mögliche Werte: • <code>true</code>: Sekunden werden beim Date-/Timepicker ebenfalls angezeigt • <code>false</code>: Sekunden werden beim Date-/Timepicker nicht angezeigt (Standard)

Attribut	Beschreibung
type	Typ der TextBox, ändert Verhalten und Aussehen. Mögliche Werte: • <i>date</i> : Datum mit Datepicker • <i>datetime</i>: Datum und Zeit mit Date-/Timepicker  Mit dem Attribut showSeconds lässt sich zusätzlich definieren, ob der Date-/Timepicker auch Sekundenangaben anzeigt.  Das Datum und die Uhrzeit werden entsprechend der Zeitzone des Browsers angezeigt. • <i>email</i> : E-Mail-Adresse • <i>month</i> : Monat und Jahr mit Datepicker • <i>number</i> : Zahl mit Wählpfeilen (Spinner) • <i>search</i> : Suchfeld • <i>tel</i> : Telefonnummer • <i>text</i> (Standard): Text • <i>time</i> : Zeit mit Wählpfeilen (Spinner) • <i>url</i> : URL • <i>week</i> : Kalenderwoche und Jahr mit Datepicker und Wählpfeilen (Spinner)  Alle Werte, die ein Datum oder eine Uhrzeit enthalten, müssen in UTC gespeichert und bereitgestellt werden!
value	Inhalt der TextBox. • Übersetzbar • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette

Attribut	Beschreibung
background	Definiert eine Farbe für den Hintergrund des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Maps und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
enabled	Legt fest, ob der Nutzer mit dem Control interagieren kann. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding  Das Attribut enabled ersetzt das veraltete Attribut disabled. <i>enabled="true"</i> entspricht damit dem veralteten Attribut <i>disabled="false"</i>
fontFamily	Legt die Schriftfamilie fest.  • Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. • Wird das Attribut auf einem Element definiert, so wird die Schriftfamilie an die Kindelemente des Elementes vererbt, sofern die Kindelemente das Attribut fontFamily nicht explizit gesetzt haben. • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument. Mögliche Werte: • MainFont: hinterlegte Hauptschrift • Fontcode aus der Font-Palette, z. B. Font04

Attribut	Beschreibung
fontSize	Legt die Schriftgröße fest. i • Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. • Dieses Attribut funktioniert nicht für die Elemente Image, Map und HtmlDocument. Mögliche Werte: • Beliebige ganze Zahl oder Dezimalzahl mit einem Punkt als Dezimaltrennzeichen, z. B. 20; 20.8; .9 • Schriftgröße in Pixel, z. B. 20px • Schriftgröße in Punkt, z. B. 18pt • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. .8em oder 120% • 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>
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. i Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded i Dieses Attribut funktioniert nicht für die Elemente Image, Map, Chart und HtmlDocument.

Attribut	Beschreibung
fontStyle	Legt die Neigung der Schrift fest. i Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <code>italic</code>: kursive Schrift • <code>normal</code>: normale Schrift (Standard) • <code>oblique</code>: schräg gestellter Schriftstil (berechnet) i Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>.
fontWeight	Legt die Schriftstärke fest. i Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <code>Black</code> • <code>Bold</code> • <code>DemiBold</code> • <code>ExtraBlack</code> • <code>ExtraBold</code> • <code>ExtraLight</code> • <code>Heavy</code> • <code>Light</code> • <code>Medium</code> • <code>Normal</code> (Standard) • <code>Regular</code> • <code>SemiBold</code> • <code>Thin</code> • <code>UltraBlack</code> • <code>UltraBold</code> • <code>UltraLight</code> i Dieses Attribut funktioniert nicht für die Elemente <code>Image</code>, <code>Map</code> und <code>HtmlDocument</code>.

Attribut	Beschreibung
foreground	Definiert eine Farbe für den Vordergrund (Texte usw.) des Controls.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • Dieses Attribut funktioniert nicht für die Controls Map und HtmlDocument! Mögliche Werte: • Hexadezimaler Farbwert, z. B. ff5a00  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. A200
horizontalAlign	Richtung, in die die Elemente fließen. Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <i>left</i> (Standard) • <i>center</i> • <i>right</i>
textOverflow	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <i>ellipsis</i>: Mit ... zeigen, dass der Text nicht zu Ende ist • <i>hidden</i>: Text abbrechen, auf ganze Wörter achten • <i>wordBreak</i>: innerhalb des Wortes abbrechen • <i>allow</i> (Standard): Text zwischen den Wörtern umbrechen
visible	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <i>true</i> / <i>false</i> oder Zeichenkette für Data Binding

11.20.1 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:



11.21 Video

<Video>-Controls werden verwendet, um Videos einzubetten.

Neben den allgemeinen Attributen für Controls lassen sich folgende zusätzlichen Attribute für das Element <Video> definieren:

Attribut	Beschreibung
type	Definiert die Art des Videos, das eingebunden wird. Mögliche Werte: • YouTube: Video von YouTube mit der Video-ID einbetten • File: Video vom Dateisystem einbinden (Data Binding möglich)  Hierfür wird das HTML-Element <video> verwendet.  Der Browser muss den Dateityp des Videos unterstützen!

Attribut	Beschreibung
value	Video, das eingebettet wird. Mögliche Werte: • YouTube-ID des Videos (falls type="YouTube") • URI zu einer Videodatei (falls type="File"). Data Binding ist möglich.
autoplay	Definiert das Autoplay-Verhalten des Videos. Mögliche Werte: • true (Standard): Das Video startet, sobald die Seite geladen ist. • false: Das Video startet nicht automatisch.  Manche Browser/-einstellungen verhindern Autoplay!
aspectRatioX aspectRatioY	X-Anteil bzw. Y-Anteil des Seitenverhältnisses.  Für das Seitenverhältnis müssen sowohl aspectRatioX als auch aspectRatioY angegeben werden!  Wenn height und width angegeben werden, dann werden aspectRatioX und aspectRatioY ignoriert!  YouTube-Videos haben standardmäßig ein Seitenverhältnis von 16:9. Bei Videos vom Dateisystem wird das Seitenverhältnis der Datei übernommen.

Beispiele

Eingebundenes YouTube-Video

```
<DetailComponent displayName="YouTube" path="YouTube" default="true">
  <FlowLayout>
    <Video type="YouTube" value="feY13KgCMow" />
  </FlowLayout>
</DetailComponent>
```

Eingebundene statische Ressource

```
<DetailComponent displayName="Resource File" path="File">
  <FlowLayout>
    <Video type="File" value="video.mp4" />
  </FlowLayout>
</DetailComponent>
```

Eingebundenes Video vom Dateisystem mit Data Binding

```
<DetailComponent displayName="Resource Binding" path="Binding">
  <Properties>
    <Property name="VideoPath" type="String" />
  </Properties>
  <FlowLayout>
    <ComboBox value="#VideoPath">
      <Option displayName="Video 1" value="video1.mp4" />
      <Option displayName="Video 2" value="video2.mp4" />
    </ComboBox>
    <Video value="#VideoPath" type="File" autoplay="true" aspectRatioX="16"
aspectRatioY="9" />
  </FlowLayout>
</DetailComponent>
```

Video von einer externen URL

```
<DetailComponent displayName="External Video" path="External" process="ExternalVideo.
wrf">
  <Properties>
    <Property name="ExternalVideo" type="Video" />
  </Properties>
  <FlowLayout>
    <Video value="#ExternalVideo" type="File" aspectRatioX="4" aspectRatioY="3" /
>
  </FlowLayout>
</DetailComponent>
```

- ⚠ Für das Datenelement, das durch den Technical Process bereitgestellt wird, muss mit dem Attribut `type="external"` festgelegt werden, dass eine externe URL verwendet wird. Ansonsten wird eine interne URL erwartet und es kommt zu einem Fehler.

```
<Ok>
  <ExternalVideo type="external">
    <!-- Video URL -->
  </ExternalVideo>
</Ok>
```

11.21.1 Data Binding für value

Das Attribut `value` des Elements `<Video>` kann per Data Binding befüllt werden. Dafür muss die Property den Typ `String`, `Base64` oder `Video` haben.

Wenn ein YouTube-Video (`type="YouTube"`) eingebunden wird, dann muss per Data Binding immer eine gültige YouTube-Video-ID bereitgestellt werden.

Wenn Videodateien eingebunden werden (`type="File"`), dann werden folgende Inhalte unterstützt:

- Property-Typ **String**: Die Property muss einen Dateipfad zu einer Videodatei enthalten. Die Videodatei muss direkt im Ordner `Resources` des Web App-Projekts liegen.
- Property-Typ **Base64**: Die Property muss eine Base64-kodierte Videodatei enthalten. Der Medientyp des Videos muss im Datenelement definiert werden, das vom Technical Process als XML zurückgegeben wird.

```
<MyVideo mediaType="video/mp4">
  <!-- Base64 -->
</MyVideo>
```

- Property-Typ **Video**: Die Property kann entweder einen Dateipfad zu einer Videodatei, eine Base64-kodierte Videodatei oder eine externe URL enthalten. Was bereitgestellt wird, wird über das Attribut `type` im Datenelement des antwortenden Technical Processes definiert. Für `type` können die Werte `resource` (Standard), `base64` oder `external` angegeben werden.

- ⚠ Safari unterstützt keine Videos, die von Dateisystemen eingebunden werden. Verwenden Sie stattdessen externe URLs, die Buffering unterstützen!

- ✓ Videos von Dateisystemen und Base64-kodierte Videos unterstützen kein Buffering, sondern werden auf einmal geladen. Verwenden Sie Videos von Dateisystemen und Base64-kodierte Videos daher nur bei kleinen Videos. Verwenden Sie für größere Videos externe URLs, die Buffering unterstützen.

12 Actions

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

Folgende Actions können definiert werden:

- Action „Neu“: `<NewAction>`
- Action „Speichern“: `<SaveAction>`
- Action „Löschen“: `<DeleteAction>`
- Action „Datei hochladen“: `<UploadAction>`
- Action „Datei herunterladen“: `<DownloadAction>`
- Action „Custom“: `<CustomAction>`
- Action „Neu laden“: `<ReloadAction>`
- Action „Abbrechen“: `<CancelAction>`
- Action „Auswählen“: `<SelectAction>`

 Die Action `SelectAction` erzeugt keine Schaltfläche.

Alle Actions außer der benutzerdefinierten Action (`CustomAction`) und der `SelectAction` haben vordefinierte Beschriftungen und Icons. Diese können überschrieben werden, wenn es notwendig ist.

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

Attribut	Beschreibung
background	Definiert eine Farbe für den Hintergrund des Controls.  Diese Einstellung überschreibt die Standardfarbe des Farbschemas! Mögliche Werte: • Hexadezimaler Farbwert, z. B. <code>ff5a00</code>  Verwenden Sie keine Raute (#) vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <code>A200</code>

Attribut	Beschreibung
componentName	Name der Component, zu der am Ende der Action navigiert werden soll. Mögliche Werte: Beliebige Zeichenkette  Dieses Attribut kann für die Aktion "Abbrechen" nicht definiert werden und führt zu einem Validierungsfehler, falls das Attribut trotzdem gesetzt wird.
displayName	Beschriftung des Controls. • Übersetzbar Mögliche Werte: Beliebige Zeichenkette oder Sprachschlüssel
enabled	Aktiviert die Action • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding  Das Attribut enabled ersetzt das veraltete Attribut disabled. <i>enabled="true"</i> entspricht damit dem veralteten Attribut <i>disabled="false"</i>
fontFamily	Legt die Schriftfamilie fest. Mit diesem Attribut wird die Standardschrift der Web App für dieses Control überschrieben. Mögliche Werte: Fontcode aus der Font-Palette, z. B. Font04
fontSize	Legt die Schriftgröße fest. Mit diesem Attribut wird die Standardschriftgröße der Web App für dieses Control überschrieben. Mögliche Werte: • Schriftgröße in Pixel, z. B. <i>20px</i> • Schriftgröße in Punkt, z. B. <i>18pt</i> • Schriftgröße im Vergleich zur Schriftgröße des Elternelements, z. B. <i>.8em</i> oder <i>120%</i> • Schlüsselwörter, z. B. <i>small</i> oder <i>larger</i>

Attribut	Beschreibung
fontStretch	Legt die Breite der einzelnen Schriftzeichen fest. Mit diesem Attribut wird die Standardbreite der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Condensed • Expanded • ExtraCondensed • ExtraExpanded • Medium • Normal (Standard) • SemiCondensed • SemiExpanded • UltraCondensed • UltraExpanded
fontStyle	Legt die Neigung der Schrift fest. Mit diesem Attribut wird die Standardneigung der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • <i>italic</i>: kursive Schrift • <i>normal</i>: normale Schrift (Standard) • <i>oblique</i>: schräg gestellter Schriftstil (berechnet)
fontWeight	Legt die Schriftstärke fest. Mit diesem Attribut wird die Standardstärke der Schriftzeichen für dieses Control überschrieben. Mögliche Werte: • Black • Bold • DemiBold • ExtraBlack • ExtraBold • ExtraLight • Heavy • Light • Medium • Normal (Standard) • Regular • SemiBold • Thin • UltraBlack • UltraBold • UltraLight

Attribut	Beschreibung
foreground	Definiert eine Farbe für den Vordergrund (Texte, Icons) des Buttons.  • Diese Einstellung überschreibt die Standardfarbe des Farbschemas! • 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. Mögliche Werte: • Hexadezimaler Farbwert, z. B. <code>ff5a00</code>  Verwenden Sie keine Raute vor dem Farbwert! Verwenden Sie keine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <code>A200</code>
iconColor	Definiert die Farbe des Icons. Mögliche Werte: • Hexadezimaler Farbwert, z. B. <code>ff5a00</code>  Verwenden Sie keine Raute vor dem Farbwert oder eine verkürzte Schreibweise des Farbwerts! • Farbcode aus der Farbpalette der Web App (vgl. Theming), z. B. <code>A200</code>
iconPosition	Legt die Position des Icons auf dem Button fest. Mögliche Werte: • <code>left</code>: Das Icon wird links auf dem Button angezeigt (Standard). • <code>right</code>: Das Icon wird rechts auf dem Button angezeigt.

Attribut	Beschreibung
iconUrl	Pfad zu einer Grafikdatei oder Angabe eines Material Icons, der als Icon verwendet wird. i - Die Grafikdatei muss im Ordner Resources direkt unterhalb des Web-App-Projekts enthalten sein. Pfadangabe relativ zum Ordner Resources. - Das Material Icon muss mit dem Präfix <code>icon:</code>, z. B. <code>icon:<MaterialIconName></code> angegeben werden. Mögliche Werte: - Zeichenkette (URI), z. B. <code>clock.png</code> <code>icon:<MaterialIconName></code>, z. B. <code>icon:extension</code> ✓ Mit Strg+Leertaste erhalten Sie eine Übersicht der verfügbaren Icons. Die Auswahl kann von den tatsächlich verfügbaren Material Icons abweichen.
process	Pfad zur <code>.wrf</code>-Datei. Der Technical Process muss im Ordner <code>Services/Processes</code> enthalten sein. Pfadangabe relativ zum Ordner <code>Services/Processes</code>. Mögliche Werte: Zeichenkette (URI) i Dieses Attribut kann für die Aktion "Abbrechen" nicht definiert werden und führt zu einem Validierungsfehler, falls das Attribut trotzdem gesetzt wird.
selectionNeeded	Legt fest, ob für die Aktion ein Eintrag aus einer Liste gewählt sein muss. Funktioniert nur in List Component. Mögliche Werte: <code>true / false</code>
tooltip	Legt den Text fest, der als Tooltip der Aktion angezeigt wird, sobald der Aktions-Button mit der Maus überfahren wird. Mögliche Werte: Zeichenkette i Wenn der Wert des Attributs <code>tooltip</code> leer ist, dann wird kein Tooltip angezeigt.
visible	Legt fest, ob der Button für die Aktion sichtbar ist. Data Binding (Boolean) möglich Mögliche Werte: <code>true / false</code> oder Zeichenkette für Data Binding

Jede der oben genannten Actions kann mit dem Element `<Parameters>` erweitert werden. Innerhalb des Elements `<Parameters>` können Sie mit dem Element `<Parameter>` weitere Definitionen vornehmen.

Das Element `<Parameter>` enthält die folgenden obligatorischen Attribute:

Attribut	Beschreibung
key	Name des Parameters • Data Binding (Boolean) möglich Mögliche Werte: Zeichenausdruck oder Ausdruck für Data Binding
value	Parameterwert • Data Binding (Boolean) möglich Mögliche Werte: Zeichenkette oder Ausdruck für Data Binding

12.1 Action "New"

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

 Für die Action **New** lassen sich die Standardattribute für Actions definieren.

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

Beispiel

Beispiel Action "Neu"

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

Das obige Beispiel erzeugt folgende Action:



12.2 Action "Save"

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

 Für die Action **Save** lassen sich die Standardattribute für Actions definieren.

Beispiel

Beispiel Action "Speichern"

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

Das obige Beispiel erzeugt folgende Action:



12.3 Action "Delete"

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

 Für die Action **Delete** lassen sich die Standardattribute für Actions definieren.

Beispiel

Beispiel Action "Löschen"

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

Das obige Beispiel erzeugt folgende Action:



12.4 Action "Upload"

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

Der Technical Process, 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.

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

Zusätzlich zu den Standardattributen für `Actions` kann das Element `<UploadAction/>` noch folgende Attribute haben:

Attribut	Beschreibung
acceptedFileTypes	Dateityp, nach dem im Auswahldialog gefiltert wird. • Data Binding möglich  Dieses Attribut filtert nur nach Dateitypen. Der Filter kann vom Benutzer zurückgesetzt werden, um alle Dateitypen anzuzeigen und auszuwählen. Mögliche Werte: MIME-Typen, auch mit Wildcards (*), z. B. • text/*: Textdateien • image/*: Grafikdateien • image/jpeg: Nur JPG-Dateien • image/png: Nur PNG-Dateien • video/*: Videodateien • audio/*: Audiodateien • application/*: Dateien, die an ein bestimmtes Programm gebunden sind • multipart/*: mehrteilige Daten • message/*: Nachrichten • model/*: Dateien, die mehrdimensionale Strukturen repräsentieren
fileId	Identifikator, der vom Prozessentwickler verwendet werden kann. • Data Binding möglich. Mögliche Werte: Beliebige Zeichenkette
maxSizeMB	Maximale Dateigröße in Megabyte. Mögliche Werte: Ganzzahl

Format der hochgeladenen Daten

Die Daten werden in einem bestimmten Input-Format an den Technical Process 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 Action:



- ✓ Weitere Informationen finden Sie im Abschnitt
 - [Meldungen](#)

12.5 Action "Download"

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

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

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

Attribut	Beschreibung
data	 Nur verwenden, wenn eine Base64-Datei zum Download angeboten wird! Gibt an, welche Datei heruntergeladen werden soll. Mögliche Werte: Zeichenkette für Data Binding  Der Data-Binding-Ausdruck muss auf eine Property vom Typ base64 verweisen!
fileId	<i>Erforderlich.</i> Identifikator, der vom Prozessentwickler verwendet werden kann. • Data Binding möglich. Mögliche Werte: Beliebige Zeichenkette
fileName	<i>Erforderlich.</i> Name der Datei, die heruntergeladen werden soll. Wird auch als Dateiname der heruntergeladenen Datei beim Benutzer verwendet. • Data Binding möglich Mögliche Werte: Beliebige Zeichenkette

-  Die Aktion "Download" unterstützt nicht nur den Download von Daten aus einem Technical Process, 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"/>
```

12.6 Action "Custom"

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

Zusätzlich zu den Standardattributen für `Actions` kann das Element `<CustomAction/>` noch folgende Attribute haben:

Attribut	Beschreibung
displayName	Beschriftung des Buttons. • Übersetzbar Mögliche Werte: Beliebige Zeichenkette oder Sprachschlüssel
externalLink	Ermöglicht die Navigation zu einer externen Website, indem ein externer Link festgelegt wird. • Data Binding möglich Mögliche Werte: Zeichenkette (URL)  Der externe Link kann auch von einem Technical Process übergeben werden. Dazu muss mit der Aktion ein Technical Process (Attribut process) verknüpft werden, der in seiner Antwort den externen Link enthält: <pre><ok externalLink="http://www.google.de" /></pre>
externalLinkTarget	Gibt an, ob der Link (externalLink) im selben oder in einem neuen Tab geöffnet wird. Mögliche Werte: • same: Der Link wird im selben Tab geöffnet • new: Der Link wird in einem neuen Tab geöffnet
validated	Ermöglicht eine Validierung der Eingabe Mögliche Werte: • true: die Validierung ist aktiviert  Schlägt die Validierung fehl, wird die Schaltfläche deaktiviert. • false: die Validierung ist nicht aktiviert (Standard)  Schlägt die Validierung fehl, wird die Schaltfläche nicht deaktiviert.
url	Ermöglicht die Navigation zu einer externen Website, indem ein externer Link festgelegt wird. • Data Binding

Beispiel

Beispiel Action "Custom"

```

<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 Action *Save customer*:

The screenshot shows a web application interface. At the top, there are three buttons: 'Neu' (with a plus icon), 'Löschen' (with a trash icon), and 'Save custom'. Below these buttons is a table with a dark blue header labeled 'Land'. The table contains four rows with the following values: 'Amerika', 'Brasilien', 'Deutschland', and 'Dänemark'. Below the table is a form with a dark blue header labeled 'Land' and a text input field containing 'Spain'.

12.7 Action "Reload"

Die Action **Reload** wird über das Element `<ReloadAction/>` innerhalb von `<Actions>` erzeugt. Mit der Action **Reload** werden die Properties einer Component periodisch neu geladen (über den in process angegebenen Technical Process) 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
componentName	Name der Component, zu der am Ende der Action navigiert werden soll. i Erforderlich, wenn das Attribut process nicht gesetzt wurde. Mögliche Werte: Beliebige Zeichenkette
enabled	Steuert, ob das Neuladen periodisch gestartet wird oder nicht. • Data Binding (Boolean) möglich Mögliche Werte: true/false oder Ausdruck für Data Binding
process	Pfad zur .wrf-Datei. Der Technical Process muss im Ordner Services/Processes enthalten sein. Pfadangabe relativ zum Ordner Services/Processes. i Erforderlich, wenn das Attribut componentName nicht gesetzt wurde. Mögliche Werte: Zeichenkette (URI)
refreshTime	Zeit bis zum erneuten Laden in Sekunden Mögliche Werte: Beliebige ganze Zahl

12.8 Action "Cancel"

Die Action **Cancel** wird über das Element `<CancelAction/>` innerhalb von `<Actions>` erzeugt und führt dazu, dass ein geöffnetes [Overlay-Strukturelement](#) oder eine [Grid Component](#) ohne Speichern und erneutes Laden der Daten geschlossen werden kann.

i Für die Action **Cancel** lassen sich die Standardattribute für [Actions](#) definieren. Die Attribute `componentName` und `process` können für diese Action nicht definiert werden und führen zu Validierungsfehlern, falls die Attribute trotzdem gesetzt werden.

Beispiel

Beispiel Action "Cancel"

```

<DetailComponent>
  <Actions>
    <CancelAction displayName="Abbrechen"/>
  </Actions>
  <FlowLayout>
    ...
  </FlowLayout>
</DetailComponent>

```

Das obige Beispiel erzeugt folgende Action:



12.9 Action "Select"

Die Action **Select** wird über das Element `<SelectAction/>` erzeugt und kann zum Beispiel einen Technical Process oder eine externe Webseite aufrufen.

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

Attribut	Beschreibung
externalLink	Ermöglicht die Navigation zu einer externen Website, indem ein externer Link festgelegt wird. • Data Binding möglich Mögliche Werte: Zeichenkette (URL) i Der externe Link kann auch von einem Technical Process übergeben werden. Dazu muss mit der Action ein Technical Process (Attribut process) verknüpft werden, der in seiner Antwort den externen Link enthält: <pre><Ok externalLink="http://www.google.de" /></pre>
externalLinkTarget	Gibt an, ob der Link (externalLink) im selben oder in einem neuen Tab geöffnet wird. Mögliche Werte: • same: Der Link wird im selben Tab geöffnet • new: Der Link wird in einem neuen Tab geöffnet

Attribut	Beschreibung
url	Ermöglicht die Navigation zu einer externen Website, indem ein externer Link festgelegt wird. • Data Binding

12.9.1 Beispiel

```

<DetailComponent
  xmlns="http://softproject.de/webapp/1.0">
  <Properties>
    <Property name="text" type="Integer"></Property>
    <Property name="date" type="DateTime"></Property>
  </Properties>
  <FlowLayout>
    <Header value="Welcome to my new Web App!" />
    <TextBox value="#text" type="number">
      <SelectAction process="sendData.wrf"></SelectAction>
    </TextBox>
    <TextBox value="#date" type="datetime">
      <SelectAction process="sendData.wrf"></SelectAction>
    </TextBox>
  </FlowLayout>
</DetailComponent>

```

12.10 Action "ForgotPassword"

Die Action **ForgotPassword** wird über das Element `<ForgotPasswordAction/>` erzeugt und ruft eine Seite auf, um die E-Mail-Adresse für das Zusenden des Passworts einzutragen.

⚠ Die Action **ForgotPassword** kann nur innerhalb der Anmeldeseite `.login` verwendet werden.

Passwort vergessen?

12.10.1 Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<Login
  xmlns="http://softproject.de/webapp/1.0">
  <DetailComponent>
    <FlowLayout>
      <Button>
        <ForgotPasswordAction displayName="$ForgotPassword"/>
      </Button>
    </FlowLayout>
  </DetailComponent>
</Login>
```

12.11 Action "LogoutAction"

Die Action **LogoutAction** wird über das Element `<LogoutAction>` erzeugt und führt zum Ausloggen aus der Web App. Die Web App-Seite wird neu geladen.

 Für die Action **LogOut** lassen sich die Standardattribute für Actions definieren.

12.11.1 Beispiel

```
<DetailComponent>
  <FlowLayout>
    <TextBox value="#TEST"/>
    <Button>
      <LogoutAction process="logout.wrf"/>
    </Button>
  </FlowLayout>
</DetailComponent>
```

13 HeaderSection

Mit der HeaderSection-Datei können Sie den Header einer Web App individuell gestalten. Dabei stehen Ihnen alle Layouts und Controls der X4 Web Apps zur Verfügung.

13.1 HeaderSection anlegen

Die HeaderSection wird nach dem Web App-Projekt benannt und hat die Dateiendung .headersection. Innerhalb eines Web App-Projekts kann nur eine .headersection-Datei angelegt werden.

Das Element `<HeaderSection>` erzeugt den Header.

Im Header können alle Layouts und Controls der X4 Web Apps verwendet werden.

 Der individuelle Header überschreibt den standardmäßigen Header.

1. Mit der rechten Maustaste auf das Web App-Projekt klicken.
2. Auf **New > HeaderSection** klicken.
Die Datei für den Header wird erzeugt.

13.1.1 `<HeaderSection>`

Folgende Attribute lassen sich für das Element `<HeaderSection>` definieren:

Attribut	Beschreibung
process	Pfad zur .wrf-Datei, die die Daten für den Header liefert. Der Prozess muss im Ordner Services/Processes enthalten sein. Mögliche Werte: Zeichenkette (URI)

14 JavaScript in Web Apps einbinden

Mit der Komponente **Web App Control Project** können X4 BPMS-Entwickler X4 Web Apps auf Basis von Web-Technologien und dem X4 ESB mit dessen Webservice-Schnittstelle um eigene JavaScript-Funktionen erweitern, sodass der Zugriff auf die Daten und Aktionen der Web App über JavaScript-API-Funktionen erfolgen kann. 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.

14.1 Mit Web App Control Projects arbeiten

Mit dem Projekttyp **Web App Control Project** können Sie X4 Web Apps um eigene JavaScript-Funktionen erweitern, sodass der Zugriff auf die Daten und Aktionen der Web App über JavaScript-API-Funktionen erfolgen kann. Mit `<CustomControls>`, also benutzerdefinierten Controls, können Sie eigenen JavaScript-Code in eine Web App einbinden und Aktionen aufrufen, um mit der Anwendung zu interagieren. Außerdem können Sie das Layout durch das Setzen von Eigenschaftsparametern beeinflussen. Um ein benutzerdefiniertes Control in einer Web App zu verwenden, müssen Sie lediglich das Web App Control Project in einem Web App Project referenzieren. In einem Web App Control Project werden die Inhalte hinterlegt, die in der Web App angezeigt werden.

14.1.1 Web App Control Project anlegen

So legen Sie ein neues Web App Control Project an:

1. Klicken Sie im X4 Designer in der Sicht **Projects** mit der rechten Maustaste, und wählen Sie im Kontextmenü den Eintrag **New**.
2. Wählen Sie **Web App Control Project....**

✓ Alternativ können Sie im Menü **File** auf **New** klicken und **Web App Control Project...** auswählen.

3. Geben Sie im Dialogfenster **Project Creation Wizard** einen Namen für Ihr Projekt ein.
4. Wählen Sie **Finish**.

Web App Control Projects haben eine vordefinierte und nicht veränderbare Ordnerstruktur, die beim Anlegen eines neuen Web App Control Projects automatisch angelegt wird.

Sources	Dieser Ordner enthält nach dem Anlegen eines Projekts drei Dateien: • index.html: Diese Datei enthält eine grundlegende HTML-Struktur, die Sie um die gewünschten Elemente erweitern können. • x4-webapp-api.d.ts: Diese Datei enthält die aktuellen API-Deklarationen (diese werden von TypeScript-Entwicklern benötigt). • x4-webapp-api.js: Diese Datei enthält das aktuelle JavaScript API. Eine Auflistung der verfügbaren API-Funktionen mit ihrer Beschreibung finden Sie unter Verfügbare API-Funktionen.  Der automatisch angelegte Ordner Sources kann nicht gelöscht, verschoben oder umbenannt werden. Im Ordner Sources können Sie z. B. auch eine <code>.css</code>-Datei anlegen.
<Projekt>.webcontrol	Projektbezogene Datei, die automatisch beim Anlegen des Projektes angelegt und nach dem Projekt benannt wird. Über diese Datei können Sie bei Bedarf eine Definition zur Cookie-Einwilligung hinzufügen. Sie können beliebigen Text in die Felder eingeben oder Angaben zu einem Drittanbieter-Service machen. Jedes Web App Control Project enthält genau eine <code>.webcontrol</code>-Datei. Wenn Sie in einem Web App Project mehr als ein Web App Control Project referenzieren möchten, legen Sie im Web App Project im Ordner Components eine neue <code>.detail-Component</code> für jedes weitere Web App Control Project an.

Hinweis:

Sie können alle Dateien aus einem Web App Control Project über die Entwicklertools im Browser ohne Authentifizierung anzeigen.

14.1.2 Benutzerdefiniertes Control in eine Web App einbinden

Voraussetzung

Sie verfügen über ein Web App Project, das mindestens einen Technical Process enthält. Informationen zum Anlegen von Web App Projects finden Sie unter [Neues Web App-Projekt anlegen](#).

In den folgenden Abschnitten wird erläutert, wie Sie ein Web App Control Project mit einem Web App Project verbinden, um ein oder mehrere benutzerdefinierte Controls in der Web App zu verwenden.

So binden Sie ein benutzerdefiniertes Control ein:

1. Öffnen Sie das Web App Project, und doppelklicken Sie auf die Datei **<Projekt>.wac**. Wählen Sie im Abschnitt **HTTP Security Header** in der Dropdown-Liste **X-Frame-Options** den Wert **SAMEORIGIN** aus.
2. Fügen Sie in der Datei **Dashboard.detail** im Ordner **Components** die gewünschten Properties ein.

An dieser Stelle fügen Sie die gewünschten Custom Controls über das Element `<CustomControl>` ein und referenzieren das zuvor angelegte Web App Control Project über das obligatorische Attribut `project`.

 Sie können im Ordner **Components** weitere Detail Components anlegen, in denen Sie z. B. Layouteigenschaften festlegen.

Außerdem können Sie folgende Attribute für das Element `<CustomControl>` definieren:

Attribut	Beschreibung
<code>horizontalAlign</code>	Richtung, in die die Elemente fließen. Die Reihenfolge der Elemente entspricht ihrer Deklaration. Mögliche Werte: • <code>left</code>(Standard) • <code>center</code> • <code>right</code>
<code>textOverflow</code>	Gibt an, was passieren soll, wenn das Control voll ist. Mögliche Werte: • <code>ellipsis</code>: Mit ... zeigen, dass der Text nicht zu Ende ist • <code>hidden</code>: Text abbrechen, auf ganze Wörter achten • <code>wordBreak</code>: innerhalb des Wortes abbrechen • <code>allow</code> (Standard): Text zwischen den Wörtern umbrechen
<code>visible</code>	Legt fest, ob das Control sichtbar ist. • Data Binding (Boolean) möglich Mögliche Werte: <code>true/false</code> oder Zeichenkette für Data Binding

Innerhalb des Elements `<CustomControl>` legen Sie über das Element `<APIAction>` die API-Funktionen fest, die in die Web App übernommen werden sollen. Über das

obligatorische Attribut name geben Sie den Namen der aufzurufenden Aktion an.

Außerdem können Sie folgende Attribute für das Element `<APIAction>` definieren:

Attribut	Beschreibung
componentName	Name der Component, zu der am Ende der Action navigiert werden soll. Mögliche Werte: Beliebige Zeichenkette
externalLink	Eingabe einer URL. Mögliche Werte: Beliebige URL
externalLinkTarget	Gibt einen externen Link an. Mögliche Werte: • new(Standard): Die Seite wird auf einer neuen Registerkarte geöffnet. • same: Die Seite wird auf derselben Registerkarte geöffnet.
process	Pfad zur .wrf-Datei. Der Technical Process muss im Ordner Services/Processes enthalten sein. Pfadangabe relativ zum Ordner Services/Processes. Mögliche Werte: Zeichenkette (URI)

3. Passen Sie im Web App Control Project die Datei **index.html** im Ordner **Sources** an, und fügen Sie die gewünschten Elemente hinzu.
4. Öffnen Sie die Web App im Browser. Informationen zu den unterstützten Browsern finden Sie [hier](#).

14.1.3 Verfügbare API-Funktionen

In jedem Web App Control Project sind im Ordner **Sources** die Dateien **x4-webapp-api.d.ts** und **x4-webapp-api.js** enthalten. Für die darin deklarierte Klasse X4API sind die folgenden Funktionen verfügbar:

Funktion	Beschreibung
<code>init(): Promise<boolean>;</code>	Initialisiert die Verbindung zur übergeordneten Custom Control Component. Rückgabewert: • Promise mit einem booleschen Wert, der angibt, ob JavaScript API von der aktuellen Version der Web App unterstützt wird
<code>getPropertyDefinitions() : Promise<any[]>;</code>	Rückgabewert: • Promise mit einem Array-Wert von Eigenschaftsdefinitionen
<code>subscribeToDataModelChanges(func: (v: any) => void): string;</code>	Mit dieser Funktion können Entwickler auf Modelländerungen reagieren, indem sie die Funktion so konfigurieren, dass eingehende Änderungen verarbeitet werden. Rückgabewert: • ID, die zu einem späteren Zeitpunkt für die Unsubscribe-Funktion verwendet werden kann
<code>subscribeToPropertyChanges(propertyName: string, func: (v: object[]) => void): string;</code>	Mit dieser Funktion können Entwickler auf Eigenschaftsänderungen reagieren, indem sie die Funktion so konfigurieren, dass eingehende Änderungen verarbeitet werden. Rückgabewert: • ID, die zu einem späteren Zeitpunkt für die Unsubscribe-Funktion verwendet werden kann
<code>getPropertyValue(v: string): Promise<string>;</code>	Rückgabewert: • Promise mit einer String-Darstellung eines Eigenschaftswerts

Funktion	Beschreibung
<code>getPropertyValidationErrors(v: string): Promise<string[]>;</code>	Rückgabewert: • Promise mit einem Array von Validierungsmeldungen (wenn keine Fehler vorliegen, ist das Array leer)
<code>setProperty(n: string, v: string): void;</code>	Legt den Eigenschaftswert nach Namen fest.
<code>unsubscribeDataModelChanges(sid: string): void;</code>	Beendet die Subscription von Modelländerungen anhand der Subscription-ID.
<code>getAllPropertyValues(): Promise<any>;</code>	Rückgabewert: • Promise mit allen Eigenschaften, die in der Component verfügbar sind
<code>unsubscribePropertyValueChanges(sid: string): void;</code>	Beendet die Subscription von Modelländerungen anhand der Subscription-ID.
<code>setControlWidth(value: string): void;</code>	Legt die gewünschte Breite des Controls fest. Mögliche Werte: • CSS-Werte für Breite/Höhe (z. B. 150px, 33%, 73vw)
<code>setControlHeight(value: string): void;</code>	Legt die gewünschte Höhe des Controls fest. Mögliche Werte: • CSS-Werte für Breite/Höhe (z. B. 150px, 33%, 73vw)
<code>getControlWidth(): Promise<any>;</code>	Rückgabewert: • Promise mit der Höhe des Controls
<code>getControlHeight(): Promise<any>;</code>	Rückgabewert: • Promise mit der Breite des Controls
<code>getActions(): Promise<any>;</code>	Rückgabewert: • Promise mit dem Action-Namen
<code>executeAction(name: string, parameters?: any): Promise<any>;</code>	Akzeptiert den Action-Namen und ein optionales benutzerdefiniertes Parameter-Array vom Typ [{key: 'test', value: 'test'},]. Rückgabewert: • Promise mit Action Response

15 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 Controls zu beeinflussen. Normalerweise werden dabei Daten aus einem Datenobjekt (z. B. einer Kundenliste) an ein Control 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 Control erkennt, wie viele Objekte vorhanden sind und passt sich automatisch an. Wenn im Control Änderungen durch den Benutzer vorgenommen werden, können diese in den Datenbestand zurückgespielt werden.

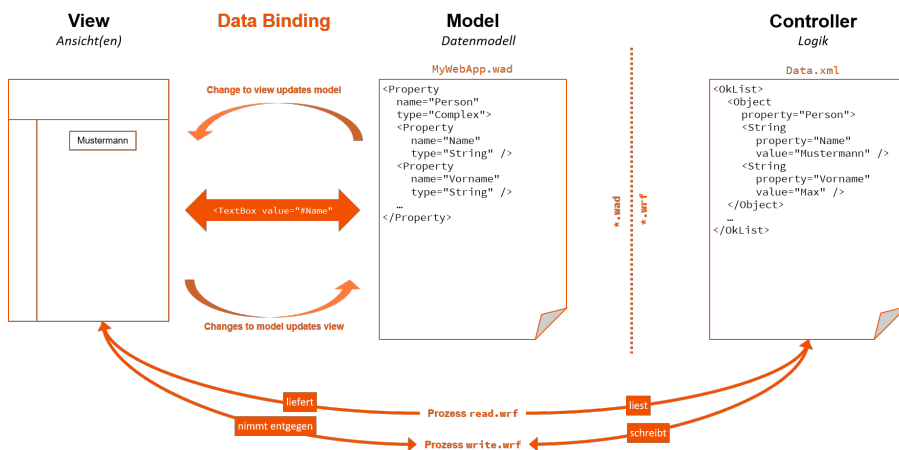
Für das Data Binding werden die `name`-Attribute der Elemente verwendet.

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` Controls 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.



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

Beispiel: \#123456

16 Navigation innerhalb einer Web-Anwendung

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

Weitere Möglichkeiten der Navigation sind die Definition einer Custom Action und das Überschreiben der `select` Action in der Liste eines Master-Strukturelements innerhalb einer Master/Detail Component. Mit diesen beiden Möglichkeiten kann auch innerhalb einer Master/Detail Component zu Subcomponents navigiert werden.

16.1 Navigation mit einer Action

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

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

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

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

16.2 Navigation durch Überschreiben der `select` Action

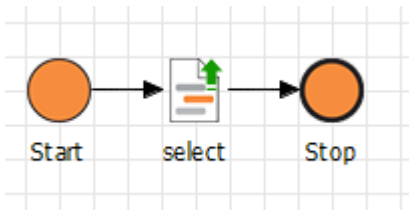
Das folgende Beispiel zeigt eine Master/Detail Component mit überschriebener `select` Action im Master-Strukturelement:

```

<MasterDetailComponent xmlns="http://softproject.de/webapp/1.0">
  <Properties>
    <Property name="List" type="Complex">
      <Property name="Value" type="String" />
    </Property>
  </Properties>
  <Master>
    <ListComponent name="list" process="fillListProcess.wrf">
      <Actions>
        <SelectAction process="select.wrf" />
      </Actions>
    </ListComponent>
  </Master>
  <Detail>
    <DetailComponent name="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 Component immer als Detail Component für die Auswahl in der Master/Detail Component verwendet. Der Technical Process, der an die überschriebene select Action gebunden ist, gibt eine Antwort zurück, die das gegebene Attribut componentName enthält. Dieses gibt die Anweisung, zur entsprechenden Component zu navigieren.

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

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

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

17 Sortieren

Tabellen innerhalb einer Web-Anwendung, die mit X4 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 Component 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.

18 Meldungen

Innerhalb einer Web-Anwendung, die mit X4 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.

i Die XML-Datei, die die Meldung enthält, muss durch einen Technical Process geliefert werden. Dieser Technical Process kann z. B. mit einer [Custom Action](#) aufgerufen werden.

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

This is a sample message.

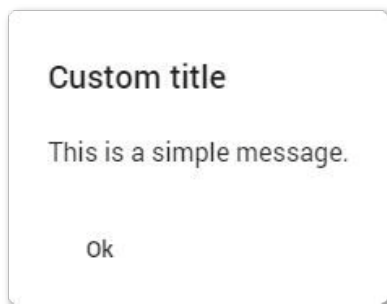
Ok

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>
```

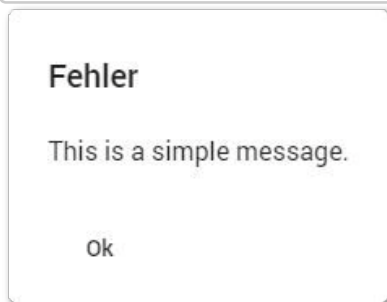
Mit dem optionalen Attribut `title` kann dem Dialogfenster, das in der Web-Anwendung ausgegeben wird, ein Titel zugewiesen werden:

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



19 Profiling

In einer Profiling-Datei (`.profiling`) können Sie festlegen, ob Prozesse innerhalb Ihres Projekts zur Laufzeit protokolliert werden sollen. In der Profiling-Datei können Sie dabei genau definieren, welche Prozesse von der Protokollierung erfasst oder ausgeschlossen werden.

Die Profiling-Datei muss nur angelegt werden, wenn sie bestimmte Prozesse von der Protokollierung ausschließen wollen.

19.1 Profiling-Datei erstellen

 In einem Projekt kann nur eine `.profiling`-Datei erstellt werden.

1. Klicken Sie in der Sicht **Projects** mit der rechten Maustaste auf ein Projekt.
2. Wählen Sie im Kontextmenü **New> Profiling**.
Die Profiling-Konfigurationsdatei wird erstellt und geöffnet.

19.2 Profiling konfigurieren

Das Wurzelement der Profiling-Datei ist `<Profiling>`. Das Element kann einzelne Prozesse des Projekts enthalten, die von einer Protokollierung explizit erfasst oder ausgeschlossen werden sollen.

Das `<Profiling>`-Element kann folgendes Attribut besitzen:

Attribut	Beschreibung
<code>active</code>	Legt fest, ob die Protokollierung von Prozessen aktiviert oder deaktiviert ist. Mögliche Werte: • <code>true</code>: Die Protokollierung der Prozesse ist aktiviert. (Standard) • <code>false</code>: Die Protokollierung der Prozesse ist deaktiviert.

Das `<Profiling>`-Element enthält das Kind-Element `<Processes>`, das folgendes Attribut enthalten kann:

Attribut	Beschreibung
filterType	Legt fest, ob die hinterlegten Prozesse von der Protokollierung erfasst oder ausgenommen werden sollen. Mögliche Werte: • <i>include</i>: Die hinterlegten Prozesse werden von der Protokollierung explizit erfasst. • <i>exclude</i>: Die hinterlegten Prozesse werden von der Protokollierung explizit ausgeschlossen.

Das <Processes>-Element kann <Process>-Kindelemente enthalten, sofern ein oder mehrere Prozesse explizit von der Protokollierung erfasst oder ausgenommen werden sollen.

Das Process-Element enthält folgendes Attribut:

Attribut	Beschreibung
path	<i>Erforderlich.</i> Enthält den Pfad zum Technical Process, der erfasst oder ausgeschlossen werden soll. Mögliche Werte: Repository-Pfad zu einem Technical Process (z.B. Processes/Process.wrf)

19.3 Beispiel einer Profiling-Konfigurationsdatei

Im nachfolgenden Beispiel werden die Technical Processes Prozess_B.wrf und Process_C.wrf von der Protokollierung ausgeschlossen.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Profiling>
  <Processes filterType="exclude">
    <Process path="Processes/Process_B.wrf"/>
    <Process path="Processes/Process_C.wrf"/>
  </Processes>
</Profiling>
```