





## Reading Sample

This sample chapter provides you with an introduction to each type of SAP Fiori elements floorplan: overview pages, list reports, analytical list pages, worklists, and object pages. You'll walk through an example of each floorplan type and then review their common uses cases. Finally, you'll see how generic annotations can be used to control and customize each type of floorplan.

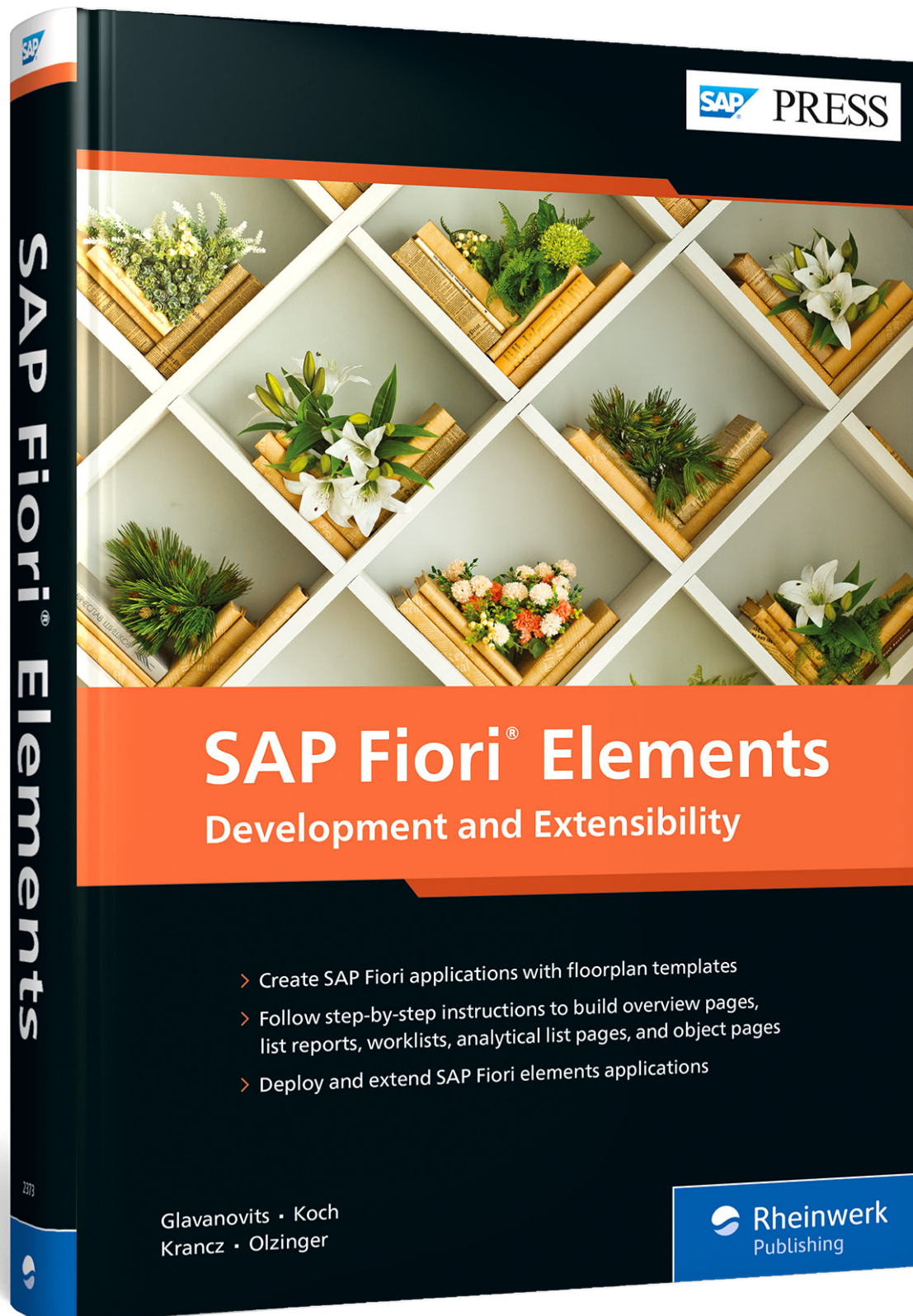
-  "Introduction to Floorplans and Application Scenarios"
-  Contents
-  Index
-  The Authors

Glavanovits, Koch, Krancz, Olzinger

### SAP Fiori Elements Development and Extensibility

413 pages | 03/2023 | \$89.95 | ISBN 978-1-4932-2373-2

 [www.sap-press.com/5641](http://www.sap-press.com/5641)



## Chapter 6

# Introduction to Floorplans and Application Scenarios

*In the development of business apps, there are always use cases that require the representation of data in similar formats. Whether lists, tables, forms, or graphics, SAP Fiori elements provides suitable floorplans are available for all.*

From the most diverse requirements for business apps, comparable use cases crystallize again and again in which data must be presented in a certain way. To implement these use cases as simply as possible and with relatively little development effort, you can fall back on *floorplans*. Admittedly, you're more flexible if you develop your own freestyle SAPUI5 apps. However, freestyle development is more complex and requires in-depth knowledge of SAPUI5 development. If you rely on SAP Fiori elements instead, you can create apps with little knowledge of SAPUI5. At the same time, you automatically achieve a consistent user interface (UI) design, and your apps comply with the SAP Fiori design guidelines.

In this chapter, we give you an overview of the floorplans available with SAP Fiori elements and their app scenarios. To this end, we'll first discuss the individual floorplans in Section 6.1. Then, in Section 6.2, we'll introduce you to their use cases. Finally, in Section 6.3, you'll learn about some annotations that can be used generically for all floorplans.

### 6.1 Available Floorplans

SAP Fiori elements offers you a range of floorplans. Whether you want to create an overview page with key figures or simply display a large amount of data in list form, there is a floorplan for every common use case. In addition to a list display, you can also create a detailed view for the individual objects within this list, which you can jump to when you select an object. You can use this detailed view to create new objects or edit existing ones. And statistical data can also be displayed in various ways with a floorplan.

We briefly introduce the available floorplans and their functions in the following sections. We'll then show a detailed development example for each floorplan type.



### OData Version

The individual floorplans sometimes differ depending on which OData version you're using. Versions 2 and 4 are available, so it could be that certain functions or annotations are only available in one of the two versions or work differently in the two versions. In the following examples we use OData V2.

#### 6.1.1 Overview Page

If you want to display an overview or dashboard for a specific business area or topic, it's best to use the floorplan overview page for this purpose. The *overview page* is particularly suitable for displaying the following information:

- Key tasks to be completed
- Various notes on a task area
- Various key performance indicators (KPIs)

In addition, you can also display diagrams and graphics on an overview page. The data displayed on the overview page is usually compiled from a variety of sources and is intended to provide the user with an overall view. For example, you could show data on all your products, customers, suppliers, or similar on an overview page.

Figure 6.1 shows what an overview page might look like. Cards are used to display the data. *Cards* are design elements on which the most important information about an object can be clearly displayed. Similar to tiles, cards allow a UI that brings together information on several objects or from several apps to be structured dynamically and clearly. This floorplan is also characterized by this design element. What kind of content you place on the individual cards is up to you. You can display tables, lists, diagrams, or tiles of any kind within the individual cards.

Basically, an overview page consists of two areas:

- Dynamic page header
- Dynamic page content

In the upper area of the app is the *dynamic page header* (Figure 6.1 ❶), which contains general information about the app. Here, for instance, the title describes the area for which the app provides data, such as purchasing or maintenance. You'll also find some user-specific settings in the header area. For example, you can call up and manage your user profile here. It's also possible to manage the displayed cards here.

If you want to further restrict the data displayed, you can do so using *smart filter* ❷, which is located below the title in the dynamic page header. For example, you can filter the data displayed so that only entries that match your filter criteria are shown, such as only certain date values, totals, and so on.

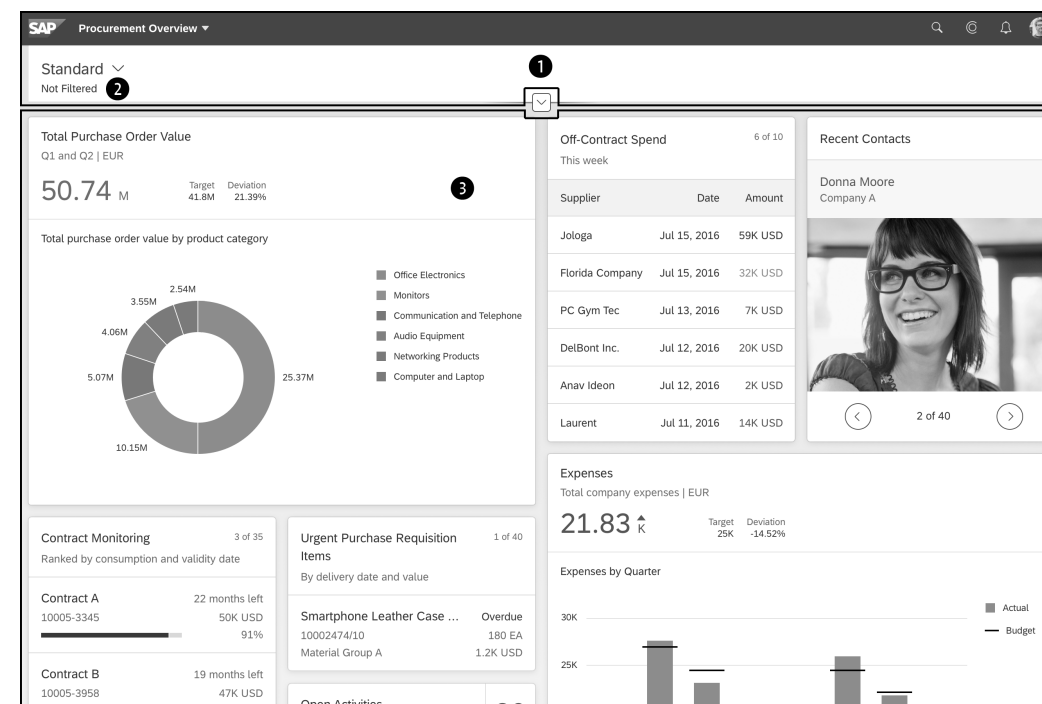


Figure 6.1 SAP Fiori Elements Overview Page

The actual content is represented by the area with the cards, also called *dynamic page content* ❸. Cards belong to the components of the UI that can be configured via annotations. This makes them very versatile, as data can be displayed in many different ways. All kinds of charts, tables, or lists can be displayed within a card. The cards are displayed in five dynamic columns, whereby a card can also extend over several columns. They can be arranged as desired via drag and drop.

Note that a card is always bound to exactly one entity set. For example, if you want to display a product list on a card, this can be defined in a JavaScript Object Notation (JSON) object, as shown in Listing 6.1. This object can be found in the *manifest.json* file of your SAP Fiori elements app. When adding a card via one of the development environments, the corresponding definition is added at the code level.

```
"sap.ovp": {
  "cards": {
    "card01": {
      "model": "ZCP_PRODUCTS_SRV",
      "template": "sap.ovp.cards.list",
      "settings": {
        "sortBy": "Price",
        "sortOrder": "descending",
        "listFlavor": "bar",
```

```

    "annotationPath": "com.sap.vocabularies.UI.v1.LineItem#bar",
    "category": "{{card01_category}}",
    "entitySet": "ProductSet"
  }
}
}
}

```

Listing 6.1 Definition of a Card in manifest.json

### 6.1.2 Floorplans for Lists

With the list floorplans, you have the option of displaying data in list or table form. Unlike the overview page, three variants of the list floorplan are available here, which are used depending on the app:

- List report
- Analytical list page
- Worklist

All three floorplans are ideal for presenting data in the form of lists or tables. However, there are some differences in their use, which we'll now discuss briefly.

#### List Report

The *list report* is mainly used when large amounts of data need to be displayed and the individual entries need to be handled. The presentation doesn't necessarily have to be in tabular or list form, and a diagram is also possible. In addition, this floorplan enables users to prepare the result set accordingly by means of filters, sorting, and grouping. In this way, large amounts of data can be reduced to the relevant records.

However, you should not use the list report if you want to see the details of a record. In this case, the object page would be suitable (Section 6.1.3).

The list report is structured as follows:

- Dynamic page header
- Content area
- Footer toolbar

At the top of the page, you'll find the dynamic page header mentioned earlier (Figure 6.2 ❶). This header contains the title of the report and tells you what information is displayed with the app. In addition, you'll also find the filter line and buttons for general actions in the dynamic page header. These actions allow you to expand or compress the header. You can also pin or unpin it so that it's always displayed or only at the top when scrolling through the list.

The *content area* ❷ displays the actual content in table, list, or chart form. You can display data in different ways in this area. In addition, a title and an *icon tab bar* are displayed above the list. The latter can contain text buttons and/or icons.

In the *footer toolbar*, you'll find overarching actions that relate to the entire data overview, for example, saving or starting a simulation, as well as a notification icon, if applicable. This icon allows you to view notifications of errors that have occurred in a floating menu (popover). The footer toolbar is optional.

Figure 6.2 shows what a list report might look like. In this example, however, no footer toolbar is used—only the dynamic page header with the filter functions and the content area.

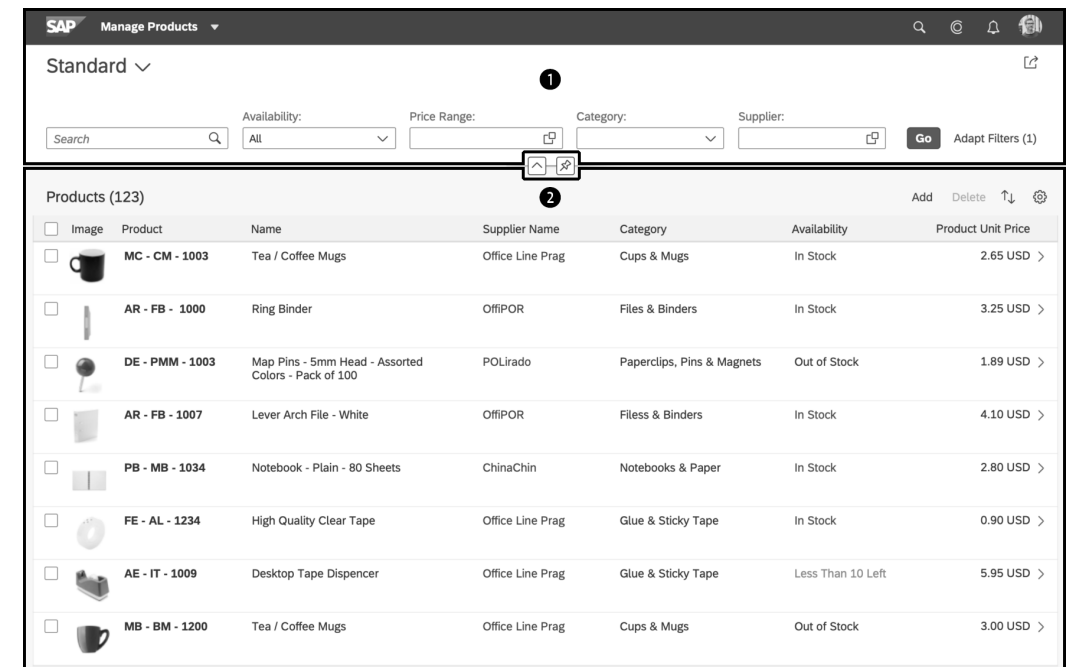


Figure 6.2 SAP Fiori Elements List Report

#### Worklist

The *worklist* is very similar to the list report in some respects. This floorplan is also used when large amounts of data need to be presented in list or table form. However, the data processed in a worklist differs from that of a list report. A list report displays all data. A worklist displays data on tasks that need to be completed by the user. A worklist can be used in an approval/rejection process, for example. As a rule, users can choose whether to complete a *work item* themselves or to delegate it.

The structure of the worklist basically corresponds to that of the list report. You'll find the following sections here as well:



- Header title
- Content area
- Footer toolbar

Figure 6.3 shows an example of a worklist. The header title and header tool are visible in the upper area, and the content area with several tabs is directly below. The *header title* ❶ for the worklist is the counterpart to the dynamic page header of the list report. However, KPIs can also be displayed here. In the content area ❷, you'll find the table with the work items. In addition, a *tab bar* ❸ can be inserted to filter the displayed data via different tabs. The footer toolbar is used in the same way as the list report. Here, too, closing functions and a floating menu (popover) for notifications are available.

Document Number	Company	Contact Person	Posting Date	Amount (Local Currency)
10223882001981	Jologa	Denise Smith	11/15/2019	12,897.00 EUR
10223882001982	DelBont Industries	Richard Wilson	11/15/2019	234,197.00 EUR
10223882001983	Jologa	Denise Smith	11/15/2019	11,865.99 EUR
10223882001984	DelBont Industries	Richard Wilson	11/15/2019	12,897.00 EUR
10223882001985	Jologa	Denise Smith	11/15/2019	12,897.00 EUR
10223882001986	DelBont Industries	Richard Wilson	11/15/2019	12,897.00 EUR
10223882001987	DelBont Industries	Richard Wilson	11/15/2019	234,197.00 EUR
10223882001988	Jologa	Denise Smith	11/15/2019	12,897.00 EUR
10223882001989	DelBont Industries	Richard Wilson	11/15/2019	12,897.00 EUR
10223882001990	DelBont Industries	Richard Wilson	11/15/2019	234,197.00 EUR
10223882001991	DelBont Industries	Richard Wilson	11/15/2019	234,197.00 EUR
10223882001992	DelBont Industries	Richard Wilson	11/15/2019	12,897.00 EUR
10223882001993	Jologa	Denise Smith	11/15/2019	11,865.99 EUR
10223882001994	Jologa	Denise Smith	11/15/2019	12,897.00 EUR
10223882001995	DelBont Industries	Richard Wilson	11/15/2019	234,197.00 EUR

Figure 6.3 SAP Fiori Elements Worklist

### Analytical List Page

The third list floorplan is the *analytical list page*, which offers you the possibility to display a wide variety of data in a unique way. For example, you can break down data into individual items and analyze them in different ways. To do this, you don't have to navigate back and forth, but can stay on the same page. In addition to a simple table view, the analytical list page allows you to display selected data in charts and more detailed lists. This allows users to see all the data sets that are relevant to them at first glance.

Highest and lowest values in particular, as well as deviations, can be displayed especially well.

Unlike the other two list floorplans, the analytical list page consists of only two elements:

- Analytical list page header
- Analytical list page content area

The *analytical list page header* (Figure 6.4 ❶) contains filter operations in addition to a title. You'll also find the option to expand or compress the filter bar here. You can display data in a formatted form within the header using KPIs. The *analytical list page content area* ❷ contains the content of the analytical list page. This can be displayed using a wide variety of UI elements. It's up to you whether you display the data using a table, diagram, or both.

The analytical list page is particularly useful when the app needs to provide key information about a subject or topic area at a glance. It's often used to gain insight into the current situation and to understand how this data came about. To do this, it's often necessary to break down data to several levels, for which the analytical list page is very well suited. If, on the other hand, it's only a question of displaying an arbitrary list of data or the detailed view of an object, this floorplan should not be used.

Figure 6.4 shows the hybrid view of an analytical list page.

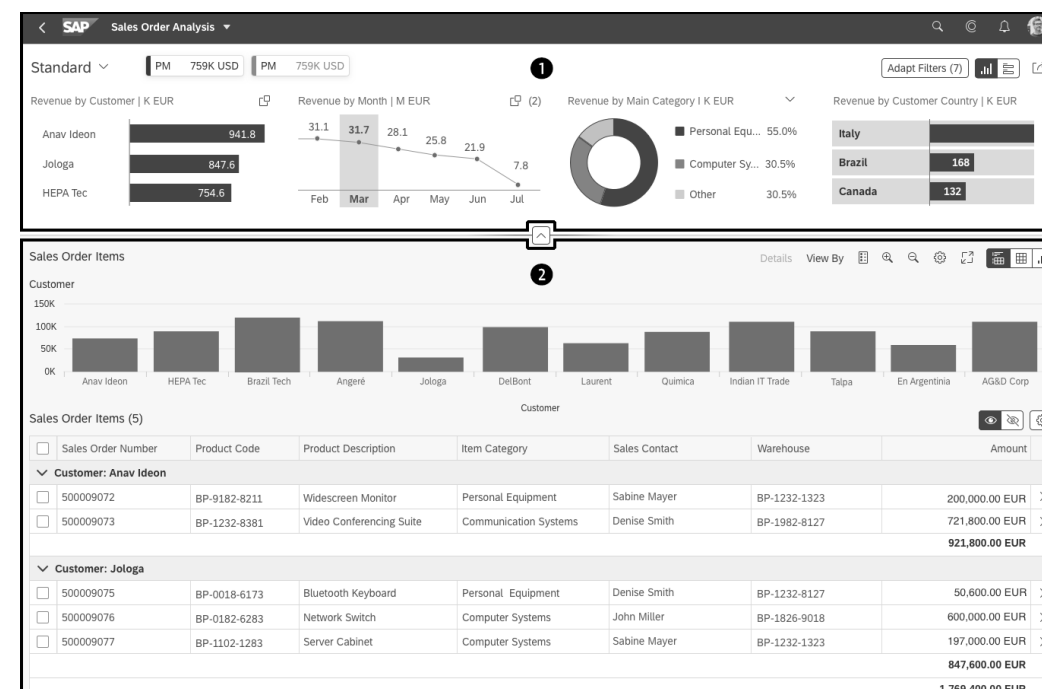


Figure 6.4 SAP Fiori Elements Analytical List Page

In addition to a tabular representation, this also contains diagrams. The tables and the charts are based on the same data. However, in the table view the data is broken down in more detail. While in the diagram, you see the **Sales Orders** cumulated per customer, the tables show the individual orders of each **Customer**. This view provides an improved overview of this data.

### 6.1.3 Object Page

In the previous sections, we've looked at floorplans that allow data to be displayed in tables or charts. What we're still missing is the detailed view of a business object. This is made possible by the object page. However, the functions of the *object page* go beyond those of a detailed view. You can also use it to create new objects, edit existing ones, or delete them. Depending on how complex your objects are, you can display the data in simple forms or in tables.

The object page is made up of the following components:

- Dynamic page header
- Navigation bar (optional)
- Content area
- Footer toolbar (optional)

The dynamic page header (Figure 6.5 ①) contains key information about the displayed object. You'll also find an **Edit** button and a **Delete** button here. In addition to the title, the header also contains a subtitle or breadcrumbs.

You can use the *navigation bar* ② to control which content is displayed in the content area. This can be used in two different ways:

- **Anchor bar**  
An anchor bar consists of a series of links that refer to different subsections. Clicking on such a link takes you to the corresponding subsection. The anchor bar is mainly used when the contents of the individual sections belong together.
- **Tab bar**  
The tab bar also contains links. However, these refer to other pages within the app. This form of navigation is used when different contents are displayed on the individual pages.

The available tabs are displayed in the form of horizontally arranged links. Each of these links can be used to jump to a section or subsection.

The content area displays the detailed data of an object. this representation is done by means of *sections* ③ and *subsections* ④. A section can be seen as a kind of container that contains several subsections. Each section has a title. In the subsections, you'll find the actual contents. Each subsection also has a title and a toolbar. Whether you display tables, forms, or diagrams in the subsections is up to you.

Below the content area is the footer toolbar, which contains, if available, final actions such as saving, canceling, or accepting changes.

Figure 6.5 shows what an object page might look like. In the header, some important KPIs are shown here. You'll also find actions such as editing or deleting an object, but also more specific functions. In the content area, the details of the selected object are displayed in forms and tables.

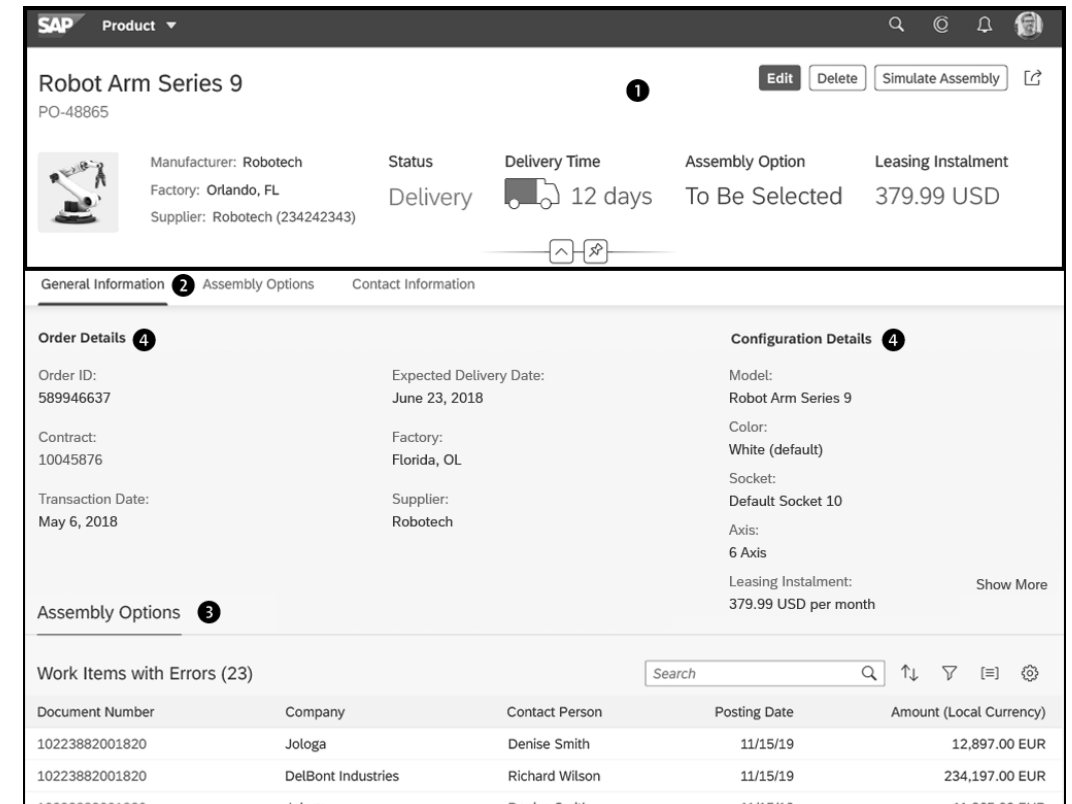


Figure 6.5 SAP Fiori Elements Object Page

The object page should be used if you want to allow users to view, create, and edit business objects. It's also very practical to get an overview of an object and to use some interaction possibilities.

## 6.2 Use Cases for the Floorplans

With the help of the individual floorplans, all conceivable scenarios can be mapped. The individual use cases have already been mentioned in the previous sections. Here, we compare them once again:

### ■ Overview page

The aim of the overview page is to provide users with an overview of a subject area or a business division. For example, they can get an overview of sales figures, earnings, or other information.

An overview page is therefore often classified as a launchpad, but there are significant differences: while the classic launchpad exists only once for each user, an overview page can have different views for a user. In addition, the contents of an overview page are context-related, whereas with the launchpad, you access completely different apps depending on your role.

### ■ List report

You should use the list report especially if you want to enable filtering, sorting, and grouping of large amounts of data. The display is in table or list form. List reports can also be ideally combined with the object page to display the details of a list object.

### ■ Worklist

The scope of the worklist is very similar to that of the list report. One difference is the data displayed. In the worklist, data records, called work items, are displayed that are to be processed by a user. In addition, a worklist contains actions that complete a process or workflow, such as releases.

### ■ Analytical list page

The analytical list page is mainly used when it comes to displaying analytical data and breaking it down to several levels. For example, you can display the turnover of customers in chart form. By selecting an individual customer, you get an exact breakdown of how this total turnover came about.

### ■ Object page

The object page is basically the detailed view of a single object. This floorplan gives you a detailed overview of the respective object. It can be used not only to display and edit the objects but also as a form to create new objects.

## 6.3 Generic Annotations

When you create apps using SAP Fiori elements, you'll notice that a lot is controlled by annotations within these apps. You can determine what content is displayed and how it's displayed. But functions such as filters can also be controlled via annotations.

The annotations can come from the OData service in the backend system, or they can be stored locally in the SAPUI5 app in a separate file. To simplify the maintenance of this file for SAP Fiori elements apps, the *annotation modeler* is provided. This is available in SAP Web IDE.

In the following sections, we'll present specific UI annotations for each of the floorplans presented, which you can use to influence the display on the interface. The annotations

listed in this section, on the other hand, are generic and can thus be applied to any floorplan of SAP Fiori elements.

### 6.3.1 UI.HeaderInfo

The following annotation is used to control the display of the respective headers:

#### ■ UI.HeaderInfo.typeName

- Datatype: String(60)
- Example: UI.HeaderInfo.typeName: 'Order'

This string, which can contain up to 60 characters, is displayed in the header of the called app and indicates the name of an object type. This annotation must be obligatory if the annotation `EndUserText.label` isn't used instead. This annotation is typically defined in a Core Data Services (CDS) view and contains a title for the CDS view that is as meaningful as possible.

#### ■ UI.HeaderInfo.typeNamePlural

- Datatype: String(60)
- Example: UI.HeaderInfo.typeNamePlural: 'Orders'

This represents the title of a list and is mandatory.

#### ■ UI.HeaderInfo.typeImageUrl

- Datatype: String
- Example: UI.HeaderInfo.typeName: '/img/order'

This contains the URL of an image used to represent the object type and can be specified optionally.

### 6.3.2 UI.selectionField

With these annotations, you can determine which filter criteria you want to provide. The following variants are available:

#### ■ UI.selectionField.qualifier

- Datatype: String(120)
- Example: UI.selectionField.qualifier: 'Ordernumber'

With this string of up to 120 characters, a unique name for a selection field can be specified.

#### ■ UI.selectionField.position

- Datatype: Decimal
- Example: UI.selectionField.position: 16

This property controls the order of the individual filter criteria. The individual criteria are sorted in ascending order.

### 6.3.3 UI.lineItem

The annotation `UI.lineItem` allows you to control the presentation of data fields in tables and lists. For example, you can specify which columns should be present, how they should be arranged, and also what the column heading should look like. The following specifications are possible here:

- **UI.lineItem.position**

- Datatype: `Decimal`
- Example: `UI.lineItem.position: 16`

The position of a column can be determined by the value of this annotation. This annotation must be obligatory.

- **UI.lineItem.importance**

- Datatype: `String`
- Possible values: `HIGH, MEDIUM, LOW`
- Example: `UI.lineItem.importance: HIGH`

This indicates the priority of a column. This specification plays a role if not all columns can be displayed due to the display size.

- **UI.lineItem.label**

- Datatype: `String(60)`
- Example: `UI.lineItem.label: 'Ordernumber'`

This indicates the title of the respective column in the table.

### 6.3.4 UI.hidden

With this annotation you can control the visibility of UI elements. Possible values for this are `true` or `false`. A corresponding specification could therefore look as follows:

`UI.hidden: false`

## 6.4 Summary

In this chapter, we provided a rough overview of the available floorplans, their structures, and their possible uses. We'll look at how these are implemented in the following chapters. In addition, you've also learned about annotations in this chapter, which can be used in any case regardless of the floorplan used.



# Contents

Preface .....	15
---------------	----

## Part I Basics

### **1 Why SAP Fiori Elements?** 21

---

<b>1.1 Historical Development of User Interfaces at SAP</b> .....	22
1.1.1 Emergence of Web Interfaces .....	23
1.1.2 Implementation of SAPUI5 and SAP Fiori .....	24
1.1.3 Templates for Recurring Requirements .....	27
<b>1.2 Requirements for User Interfaces in SAP On-Premise and Cloud Products</b> .....	32
1.2.1 On-Premise and Cloud Products .....	32
1.2.2 Consistent User Experience .....	32
<b>1.3 Low-Code Development</b> .....	35
<b>1.4 Example Application</b> .....	36
<b>1.5 Summary</b> .....	37

### **2 Overview of SAPUI5 and SAP Fiori** 39

---

<b>2.1 Introduction to SAPUI5</b> .....	39
2.1.1 Comparison with Other Frameworks .....	40
2.1.2 Release Cycle .....	40
2.1.3 Responsive Design .....	40
2.1.4 Model View Controller .....	40
<b>2.2 Standardized Development with SAP Fiori</b> .....	43
2.2.1 SAPUI5 or SAP Fiori? .....	44
2.2.2 Versions of SAP Fiori .....	46
2.2.3 SAP Fiori Design Guidelines .....	49
2.2.4 SAP Fiori Launchpad .....	51
2.2.5 Extension Concept .....	58
<b>2.3 Summary</b> .....	63

<b>3</b>	<b>OData: The Basis of All SAP Fiori Elements Applications</b>	<b>65</b>
<b>3.1</b>	<b>Introduction to the OData Standard</b>	<b>66</b>
<b>3.2</b>	<b>CRUDQ Methods</b>	<b>70</b>
3.2.1	Query	71
3.2.2	Read	73
3.2.3	Delete	76
3.2.4	Create	76
3.2.5	Update	78
<b>3.3</b>	<b>Filtering, Sorting, and Other Functions</b>	<b>78</b>
3.3.1	Filtering	79
3.3.2	Sorting	83
3.3.3	Expand	83
3.3.4	Pagination	84
3.3.5	Formatting the Response	84
3.3.6	Select	84
3.3.7	Inlinecount and Count	85
<b>3.4</b>	<b>Managing OData Services in the SAP System</b>	<b>85</b>
<b>3.5</b>	<b>OData and Core Data Services Annotations</b>	<b>92</b>
<b>3.6</b>	<b>Summary</b>	<b>95</b>
<b>4</b>	<b>Programming Models</b>	<b>97</b>
<b>4.1</b>	<b>Why Do We Need New Programming Models for the Cloud?</b>	<b>98</b>
<b>4.2</b>	<b>SAP Cloud Application Programming Model</b>	<b>100</b>
<b>4.3</b>	<b>ABAP RESTful Application Programming Model</b>	<b>110</b>
<b>4.4</b>	<b>Summary</b>	<b>115</b>
<b>5</b>	<b>Development Tools</b>	<b>117</b>
<b>5.1</b>	<b>Cloud Development Environments</b>	<b>117</b>
5.1.1	SAP Web IDE Full-Stack	117
5.1.2	SAP Business Application Studio	125

<b>5.2</b>	<b>Local Development Environments</b>	<b>134</b>
5.2.1	SAP Web IDE, Personal Edition	135
5.2.2	Eclipse	140
5.2.3	Visual Studio Code	145
<b>5.3</b>	<b>Summary</b>	<b>147</b>
<b>Part II Developing Applications with Floorplan Templates</b>		
<b>6</b>	<b>Introduction to Floorplans and Application Scenarios</b>	<b>151</b>
<b>6.1</b>	<b>Available Floorplans</b>	<b>151</b>
6.1.1	Overview Page	152
6.1.2	Floorplans for Lists	154
6.1.3	Object Page	158
<b>6.2</b>	<b>Use Cases for the Floorplans</b>	<b>159</b>
<b>6.3</b>	<b>Generic Annotations</b>	<b>160</b>
6.3.1	UI.HeaderInfo	161
6.3.2	UI.selectionField	161
6.3.3	UI.lineltem	162
6.3.4	UI.hidden	162
<b>6.4</b>	<b>Summary</b>	<b>162</b>
<b>7</b>	<b>Overview Page</b>	<b>163</b>
<b>7.1</b>	<b>Introduction</b>	<b>163</b>
7.1.1	Structure of an Overview Page	163
7.1.2	Using Overview Pages	169
<b>7.2</b>	<b>Annotations for Overview Pages</b>	<b>172</b>
<b>7.3</b>	<b>Core Data Services Views for Overview Pages</b>	<b>173</b>
<b>7.4</b>	<b>Developing an Overview Page in SAP Business Application Studio</b>	<b>175</b>
7.4.1	Create a Project	175
7.4.2	Add a Table Card	178
7.4.3	Add an Analytical Card	183
7.4.4	Add a List Card	188
7.4.5	Add a Stack Card	191

<b>7.5</b>	<b>Developing an Overview Page in SAP Web IDE</b>	193
7.5.1	Create a Project	193
7.5.2	Define Cards	196
<b>7.6</b>	<b>Summary</b>	198
<b>8</b>	<b>List Report and Worklist</b>	199
<b>8.1</b>	<b>Introduction</b>	199
8.1.1	Dynamic Page Header	200
8.1.2	Content Area	202
8.1.3	Footer Toolbar	208
<b>8.2</b>	<b>Annotations for List Reports and Worklists</b>	209
8.2.1	Annotations for List Reports	210
8.2.2	Annotations for Worklists	210
<b>8.3</b>	<b>Development of a List Report in SAP Web IDE</b>	211
8.3.1	Create a Project	211
8.3.2	Maintaining Annotations in the Core Data Services View	215
8.3.3	Maintaining Annotations Locally	219
<b>8.4</b>	<b>Developing a List Report in SAP Business Application Studio</b>	222
8.4.1	Create a Project	222
8.4.2	Maintain Annotations	226
<b>8.5</b>	<b>Developing a Worklist in SAP Business Application Studio</b>	227
8.5.1	Create a Project	228
8.5.2	Maintaining Annotations Locally	230
<b>8.6</b>	<b>Summary</b>	235
<b>9</b>	<b>Analytical List Page</b>	237
<b>9.1</b>	<b>Introduction</b>	238
9.1.1	Overview	238
9.1.2	Header	241
9.1.3	Content Area	244
<b>9.2</b>	<b>Annotations for Analytical List Pages</b>	245
<b>9.3</b>	<b>Core Data Services Views for Analytical List Page</b>	249

<b>9.4</b>	<b>Developing Analytical List Pages in SAP Business Application Studio</b>	252
9.4.1	Create a Project	252
9.4.2	Define a Visual Filter	255
9.4.3	Display Key Performance Indicators	259
<b>9.5</b>	<b>Developing an Analytical List Page in SAP Web IDE</b>	264
<b>9.6</b>	<b>Further Development of the Analytical List Page</b>	267
<b>9.7</b>	<b>Summary</b>	270
<b>10</b>	<b>Object Page</b>	271
<b>10.1</b>	<b>Introduction</b>	271
10.1.1	Header	272
10.1.2	Content Area	274
10.1.3	Footer Toolbar	276
<b>10.2</b>	<b>Annotations for the Object Pages</b>	277
10.2.1	Header Annotations	277
10.2.2	Annotations for Facets in the Header and Content Area Sections	280
10.2.3	Semantic Annotations	285
<b>10.3</b>	<b>Core Data Services View for an Object Page</b>	288
<b>10.4</b>	<b>Development of an Object Page in SAP Web IDE</b>	289
<b>10.5</b>	<b>Development of an Object Page in SAP Business Application Studio</b>	297
<b>10.6</b>	<b>Summary</b>	302

## Part III Additional Concepts

<b>11</b>	<b>Deployment</b>	305
<b>11.1</b>	<b>Continuous Integration and Continuous Delivery</b>	305
<b>11.2</b>	<b>Deployment Options</b>	307
11.2.1	On-Premise Deployment	308
11.2.2	Cloud Deployment	311
11.2.3	Mobile Deployment	314

<b>11.3</b>	<b>Configuring SAP Fiori Launchpad</b> .....	316
11.3.1	On-Premise Configuration .....	317
11.3.2	Configuration in the SAP BTP, Neo Environment .....	327
11.3.3	Configuration in the SAP BTP, Cloud Foundry Environment .....	332
<b>11.4</b>	<b>Summary</b> .....	337
<b>12</b>	<b>Extensions</b> .....	339
<hr/>		
<b>12.1</b>	<b>Developing with Guided Development</b> .....	340
<b>12.2</b>	<b>Extending Overview Pages</b> .....	345
12.2.1	Adding a Custom Filter .....	345
12.2.2	Developing a User-Specific Card .....	351
<b>12.3</b>	<b>Extending List Reports</b> .....	355
12.3.1	Adding a User-Specific Action .....	356
12.3.2	Adding a User-Specific Filter .....	359
<b>12.4</b>	<b>Extending Object Pages</b> .....	361
<b>12.5</b>	<b>Extending Analytical List Pages</b> .....	364
12.5.1	Adding Default Filters .....	364
12.5.2	Adding a User-Specific Filter .....	366
<b>12.6</b>	<b>Implementing an Adaption Project</b> .....	370
12.6.1	Overview .....	370
12.6.2	Creating an Adaption Project .....	371
12.6.3	Building Extensions in the Graphical Editor .....	374
12.6.4	Deploying an Adaption Project .....	379
<b>12.7</b>	<b>Summary</b> .....	383
<b>13</b>	<b>Navigation</b> .....	385
<hr/>		
<b>13.1</b>	<b>Internal Navigation</b> .....	385
13.1.1	General Annotations for Navigation .....	386
13.1.2	Methods of Declaring Navigation .....	387
<b>13.2</b>	<b>External Navigation</b> .....	391
13.2.1	General Annotations for Navigation .....	391
13.2.2	Methods of Declaring Navigation .....	393
13.2.3	Open Navigation Target in a New Browser Tab .....	395
<b>13.3</b>	<b>Summary</b> .....	396

<b>Appendices</b> .....	397	
<hr/>		
<b>A</b>	<b>Important Annotations</b> .....	397
<b>B</b>	<b>The Authors</b> .....	403
Index .....	405	

# Index

/UI2/FLPD_CUST .....	317	Analytical list page (Cont.)	
\$count .....	85	<i>extend</i> .....	364
<b>A</b>		<i>further development</i> .....	267
ABAP .....	97, 100, 110	<i>header</i> .....	157, 241
<i>programming model for SAP Fiori</i> .....	110	<i>OData entity</i> .....	266
<i>repository</i> .....	120	<i>overview</i> .....	238
ABAP Developer Tools (ADT) .....	135	<i>usage scenarios</i> .....	241
<i>install</i> .....	140	<i>visual filter</i> .....	266
ABAP Development Tools (ADT) .....	111, 114,	Anchor bar .....	158
173, 213, 250, 288, 289		Angular.js .....	39, 42, 43
ABAP Dictionary (DDIC) .....	41	Annotation .....	28, 106, 397
ABAP RESTful application programming		<i>@OData.publish</i> .....	173
<i>model</i> .....	100, 110, 273	<i>@UI.presentationVariant</i> .....	251
<i>business object</i> .....	112	<i>analytics</i> .....	399
ABAP-managed database procedures		<i>Analytics.dataCategory</i> .....	245
(AMDP) .....	111	<i>Analytics.query</i> .....	245
Accept .....	276	<i>Analytics.writeBack.className</i> .....	246
Action .....	317	<i>attribute-specific</i> .....	107
<i>event handlers</i> .....	358	<i>external</i> .....	94
<i>user-specific</i> .....	356	<i>imports</i> .....	267
Action ID field .....	357	<i>line item</i> .....	106
Action Position field .....	357	<i>local</i> .....	95, 364
Adaption project		<i>maintain</i> .....	226
<i>build extension</i> .....	374	<i>maintain locally</i> .....	219, 230
<i>create</i> .....	371	<i>metadata</i> .....	93
<i>deploy</i> .....	379	<i>search.defaultSearchElement</i> .....	210
<i>implement</i> .....	370	<i>search.searchable</i> .....	210
<i>name field</i> .....	380	<i>selection.fields</i> .....	106
<i>overview</i> .....	370	<i>selectionPresentationVariant</i> .....	254
Adaptivity .....	315	<i>semantics.address</i> .....	285
Address .....	275	<i>semantics.amount</i> .....	286
Agile project management .....	26	<i>semantics.businessDate</i> .....	286
Alibaba Cloud .....	311	<i>semantics.calendar</i> .....	286
Amazon Web Services .....	311	<i>semantics.calendItem</i> .....	286
Analytic Manager .....	399	<i>semantics.contact</i> .....	286
Analytical card .....	170	<i>semantics.currencyCode</i> .....	286
<i>add</i> .....	183, 186	<i>semantics.eMail</i> .....	286
<i>definition</i> .....	186	<i>semantics.fiscal</i> .....	286
<i>preview</i> .....	188	<i>semantics.geoLocation</i> .....	286
Analytical list page .....	29, 156, 160, 237	<i>semantics.language</i> .....	286
<i>annotations</i> .....	245	<i>semantics mimeType</i> .....	286
<i>CDS views</i> .....	249	<i>semantics.name</i> .....	286
<i>content area</i> .....	157, 244	<i>semantics.organization</i> .....	287
<i>create project</i> .....	252	<i>semantics.quantity</i> .....	287
<i>develop</i> .....	252	<i>semantics.systemDate</i> .....	287
<i>develop in SAP Web IDE</i> .....	264	<i>semantics.telephone</i> .....	287
		<i>semantics.text</i> .....	287
		<i>semantics.time</i> .....	287



Annotation (Cont.)  
*semantics.unitOfMeasure* ..... 287  
*semantics.url* ..... 287  
*semantics.user* ..... 287  
 UI  
   *metadata-driven* ..... 275  
   *vocabulary* ..... 94  
*UI.Chart* ..... 173  
*UI.chart* ..... 247  
*UI.dataPoint* ..... 249  
*UI.facet.id* ..... 282  
*UI.facet.label* ..... 281  
*UI.facet.parentID* ..... 282  
*UI.facet.position* ..... 282  
*UI.facet.purpose* ..... 281  
*UI.facet.targetQualifier* ..... 282  
*UI.facet.type* ..... 281  
*UI.Facets* ..... 173  
*UI.FieldGroup* ..... 172  
*UI.fieldGroup.semanticObjectAction* ..... 387  
*UI.HeaderInfo* ..... 161, 172, 210  
*UI.headerInfo.description.criticality* ..... 280  
*UI.headerInfo.description.iconURL* ..... 280  
*UI.headerInfo.description.label* ..... 279  
*UI.headerInfo.description.targetElement* ..... 280  
*UI.headerInfo.description.type* ..... 280  
*UI.headerInfo.description.value* ..... 279  
*UI.headerInfo.title.criticality* ..... 279  
*UI.headerInfo.title.iconURL* ..... 279  
*UI.headerInfo.title.label* ..... 278  
*UI.headerInfo.title.targetElement* ..... 279  
*UI.headerInfo.title.type* ..... 278  
*UI.headerInfo.title.url* ..... 279, 387  
*UI.headerInfo.title.value* ..... 278  
*UI.headerInfo.typeImageUrl* ..... 278  
*UI.headerInfo.typeName* ..... 277  
*UI.headerInfo.typeNamePlural* ..... 278  
*UI.hidden* ..... 162  
*UI.Identification* ..... 210  
*UI.identification.semanticObjectAction* ..... 387  
*UI.LineItem* ..... 210  
*UI.lineItem* ..... 162, 216, 217  
*UI.lineItem.semanticObjectAction* ..... 387  
*UI.presentationVariant* ..... 247  
*UI.selectionField* ..... 161, 218  
*UI.SelectionFields* ..... 210  
*UI.selectionPresentationVariant* ..... 246, 250  
*UI.SelectionVariant* ..... 172  
*UI.selectionVariant* ..... 247, 250  
*UI.statusInfo.semanticObjectAction* ..... 387

Apache Tomcat ..... 99  
 App  
   *hybrid* ..... 315  
   *launch* ..... 54  
 App directory ..... 108, 109  
 App Finder ..... 53  
 AppConfiguration service ..... 390  
 Application lifecycle management ..... 35  
 Application modeler ..... 132  
 Application programming interface (API) ... 98  
 Area  
   *edit* ..... 376  
 Arithmetic operator ..... 80  
 Aspect ..... 103  
 Association ..... 112  
 Avatar ..... 273  
 Azure ..... 311

**B**

Backend development ..... 25  
 Bar chart ..... 242  
   *list cards* ..... 170  
 Behavior Definition Language ..... 112  
 Bookmark service ..... 390  
 Breadcrumb ..... 272  
 BSP repository ..... 308  
 Build ..... 306  
 Business Object Processing Framework ..... 273  
 Button Text field ..... 357

**C**

CAPEX ..... 98  
 Card ..... 152  
   *custom* ..... 171  
   *define* ..... 196  
   *types* ..... 169  
   *user-specific* ..... 351  
   *XML fragment* ..... 354  
 Catalog  
   *SAP Cloud Portal service* ..... 329  
   *SAP launchpad service* ..... 334  
 Change directory ..... 371  
 Chart ..... 244, 251, 260  
   *entry point* ..... 256  
   *type* ..... 269  
 Chart-only view ..... 244  
 Citizen developer ..... 35  
 Classic ABAP programming model ..... 110  
 Close and save ..... 276  
 Cloud Connector ..... 312

Cloud development environment ..... 117  
 Codeline ..... 306  
 Collection Path field ..... 257  
 Combo box ..... 370  
 Common annotations ..... 397, 400  
 Common Data Model ..... 314  
 Common patterns ..... 49  
 Common.ValueList ..... 248  
 Common.ValueListParameterIn ..... 248  
 Common.ValueListParameterInOut ..... 249  
 Common.ValueListParameterOut ..... 248  
 Compact filter ..... 242  
 Complex control ..... 50  
 Component ..... 351  
   *controller* ..... 354  
 Component configuration ..... 58  
 Component.js file ..... 351, 354, 389  
 Composition ..... 112  
 Configuration service ..... 390  
 Consistency ..... 33  
 Consumer app ..... 316  
 Consumption annotation ..... 386, 397, 399  
 Consumption view ..... 112, 113  
 Container service ..... 389  
 Content area ..... 155, 158, 199, 202, 244, 271, 274  
 Content federation ..... 314  
 Content provider ..... 314  
 Continuous delivery ..... 305  
 Continuous integration ..... 305  
   *build* ..... 307  
 Continuous integration and continuous delivery (CI/CD) ..... 305  
 Continuous working mode ..... 206  
 Control ID field ..... 345, 366  
 Controller ..... 369  
 Controller hook ..... 59  
 controller.js file ..... 352  
 Control-level variant management ..... 200  
 Controls ..... 50  
 Conversational UX ..... 45  
 Core data services (CDS) ..... 86, 100, 110, 111, 237  
   *metadata annotations* ..... 277  
 Core data services (CDS) views ..... 91, 213, 215, 249, 271, 284, 388  
   *create* ..... 143  
   *definition* ..... 144  
   *development environment* ..... 91  
   *overview pages* ..... 173  
   *publish* ..... 251  
 Corporate design ..... 57

Count ..... 85  
 Cross navigation ..... 206  
 Cross-origin resource sharing (CORS) ... 308, 309  
 Cross-platform support ..... 35  
 CRUDQ methods ..... 70, 105  
   *create* ..... 76  
   *delete* ..... 76  
   *query* ..... 71  
   *read* ..... 73  
   *update* ..... 78  
 CSS ..... 43, 57  
 CSS3 ..... 25  
 CSV import ..... 104  
 Custom filter  
   *add* ..... 345  
   *build* ..... 348  
   *groups* ..... 350  
   *JavaScript* ..... 349, 350  
   *logic* ..... 346  
   *XML fragment* ..... 348  
 Custom Filter key field ..... 345, 366  
 Custom Filter name field ..... 345, 366  
 Custom Filter Property field ..... 346

**D**

Data binding ..... 78  
 Data Definition Language (DDL) ..... 144  
 Data Provider Class (DPC) ..... 86  
 Date functions ..... 82  
 Demilitarized zone (DMZ) ..... 310  
 Deployment ..... 34, 305  
 Deployment option ..... 307  
 Design principle ..... 44  
 Design-led development process ..... 45  
 Desktop devices ..... 239  
 Destination ..... 137  
 Detail page ..... 343  
 Detail theming ..... 57  
 Detailed description ..... 342  
 Dev space ..... 127  
   *create* ..... 127  
   *overview* ..... 128  
   *types* ..... 127  
 Developer customization ..... 62  
 Development environment ..... 44  
 Development guidelines ..... 27  
 Donut chart ..... 242  
 Draft concept ..... 49, 202  
 Draft mode ..... 276  
 Drilldown navigation ..... 206  
 Dynamic page ..... 50

- Dynamic page content ..... 152, 164, 166
  - Dynamic page header ..... 152, 158, 164, 199, 200, 271
    - variants* ..... 165
  - Dynamic page layout ..... 239
- ## E
- Eclipse ..... 25, 111, 118, 125, 134, 140, 250
    - benefits and limitations* ..... 145
    - connect to SAP* ..... 141
    - data preview* ..... 145
  - ECMAScript 2015 ..... 42
  - ECMAScript 6 ..... 42
  - Element
    - edit* ..... 376
  - Embedded deployment ..... 85, 310
  - End user customization ..... 60
  - Entity ..... 67
    - create new* ..... 76
    - display bundle* ..... 69
    - navigation attributes* ..... 75
    - read-only access* ..... 105
    - sets* ..... 79, 368
    - type* ..... 68
  - Entity Set field ..... 357
  - Entity Type field ..... 257, 346
  - Entity Type Property field ..... 257, 346
  - Event ..... 98
  - Express Framework ..... 101
  - Extension ..... 339
    - concept* ..... 58
    - options* ..... 33
    - point* ..... 59, 135
    - project* ..... 59, 135
  - External navigation ..... 385
    - annotations* ..... 391
    - declare method* ..... 393
    - open in new tab* ..... 395
- ## F
- Facet ..... 273, 283
  - Filter ..... 176
    - add default* ..... 364
    - add field* ..... 364
    - annotations* ..... 247
    - local annotations* ..... 360
    - user-specific* ..... 359, 366
    - XML fragment* ..... 366
  - Filtering ..... 79
  - Fixed card layout ..... 166, 167
- Floorplan ..... 29, 47, 50, 151
    - annotations* ..... 160
    - for lists* ..... 154
    - use cases* ..... 159
  - Footer content ..... 208
  - Footer toolbar ..... 155, 158, 199, 208, 271
    - annotation* ..... 287
    - object page* ..... 276
  - Form ..... 275
  - Fragment
    - add* ..... 377
  - Fragment File Name field ..... 345, 366
  - Freestyle SAPUI5 ..... 146, 151
  - Frontend development ..... 25
  - Frontend server ..... 308
  - Full-stack app ..... 100
  - Function Name field ..... 356
- ## G
- Gateway Service Builder ..... 86
  - Git ..... 118, 305
    - Git repository ..... 120, 121, 129
      - clone app* ..... 129
  - Google Cloud ..... 311
  - Graphic element ..... 273
  - Graphical display ..... 203
  - Graphical editor ..... 374
    - UI* ..... 375
  - Graphical user interface (GUI) ..... 342
  - Grid tables ..... 205
  - Group
    - SAP Cloud Portal service* ..... 329
    - SAP launchpad service* ..... 334
  - Grouping ..... 80
  - Growing ..... 208
  - Guided development ..... 131, 232, 339
    - add visual filter* ..... 256
    - define annotations locally* ..... 256
    - developing apps* ..... 340
    - extend overview pages* ..... 345
    - navigation* ..... 344
    - table card* ..... 178
- ## H
- H2 Database ..... 101
  - Header ..... 272
  - Header title ..... 156
  - High contrast themes ..... 56
  - Homepage ..... 51
  - Hook ..... 59

- HTML ..... 41
  - HTML5 ..... 24, 25, 43
  - HTTP request ..... 76, 78, 90
  - HTTP response ..... 77
  - Hub deployment ..... 85, 309
  - Hybrid app ..... 118
  - Hybrid view ..... 244
  - Hyperscale provider ..... 311
- ## I
- i18n ..... 41
  - i18n.properties file ..... 361, 363
  - Icon tab bar ..... 155, 205
  - ID ..... 75
    - assign* ..... 379
  - Identity Authentication ..... 34
  - Identity Provider ..... 34
  - Inbound navigation ..... 385
  - Inlinecount ..... 85
  - Input elements for filters ..... 201
  - Integrated development environment (IDE) ..... 44
  - Integration ..... 33
  - Intelligence ..... 33
  - Intent ..... 54–56, 317
  - Intent-based navigation ..... 317, 385, 390, 397
  - Internal navigation ..... 385
    - annotations* ..... 386
    - declare method* ..... 387
  - International Electrotechnical Commission (IEC) ..... 65
  - International Organization for Standardization (ISO) ..... 65
  - Internet Communication Framework (ICF) ..... 87
- ## J
- Java ..... 99
  - JavaScript ..... 25, 41, 43, 99, 100, 388
  - JavaScript controller ..... 368
  - JavaScript Object Notation (JSON) ... 71, 77, 153
  - JSON ..... 41
- ## K
- Key performance indicator (KPI) ..... 152, 243, 249, 272, 281
    - display* ..... 259
    - output* ..... 260
  - Key user customization ..... 61
- ## L
- Language parameters ..... 296
  - Launchpad target resolution service ..... 55
  - Layered repository (LREP) ..... 60, 382
  - Layout ..... 50
  - LESS ..... 57
  - Line chart ..... 242
  - Line-item navigation ..... 206
  - List card ..... 170
    - add* ..... 188
    - definition* ..... 190
    - link* ..... 170
  - List report ..... 29, 154, 160, 199, 209, 271
    - annotations* ..... 210
    - content area* ..... 202
    - develop in SAP Business Application Studio* ..... 222
    - develop in SAP Web IDE* ..... 211
    - extension* ..... 355
  - ListReportExt.controller.js ..... 363
  - Live update mode ..... 164
  - Local Data Property field ..... 257
  - Local development environment ..... 134
  - Logical operator ..... 79
  - Low-code development ..... 21, 35
- ## M
- Managed artifact ..... 103
  - manifest.json file ..... 153, 260, 261, 353, 357, 361, 368
    - cards* ..... 166
  - Math function ..... 82
  - Menu button ..... 209
  - Merge ..... 307
  - Message popover ..... 208
  - Message toast ..... 356, 358
  - Metadata ..... 28, 35
    - extension* ..... 111
  - Micro chart ..... 274
  - Mobile deployment ..... 314
  - Mobile device ..... 240
    - management* ..... 316
  - Mobile first ..... 46
  - Mobile-native app ..... 45
  - Model field ..... 346
  - Model Provider Class (MPC) ..... 86
  - Model View Controller (MVC) ..... 23, 40, 185
  - MongoDB ..... 101
  - Multiple content layout ..... 204
    - table types* ..... 205

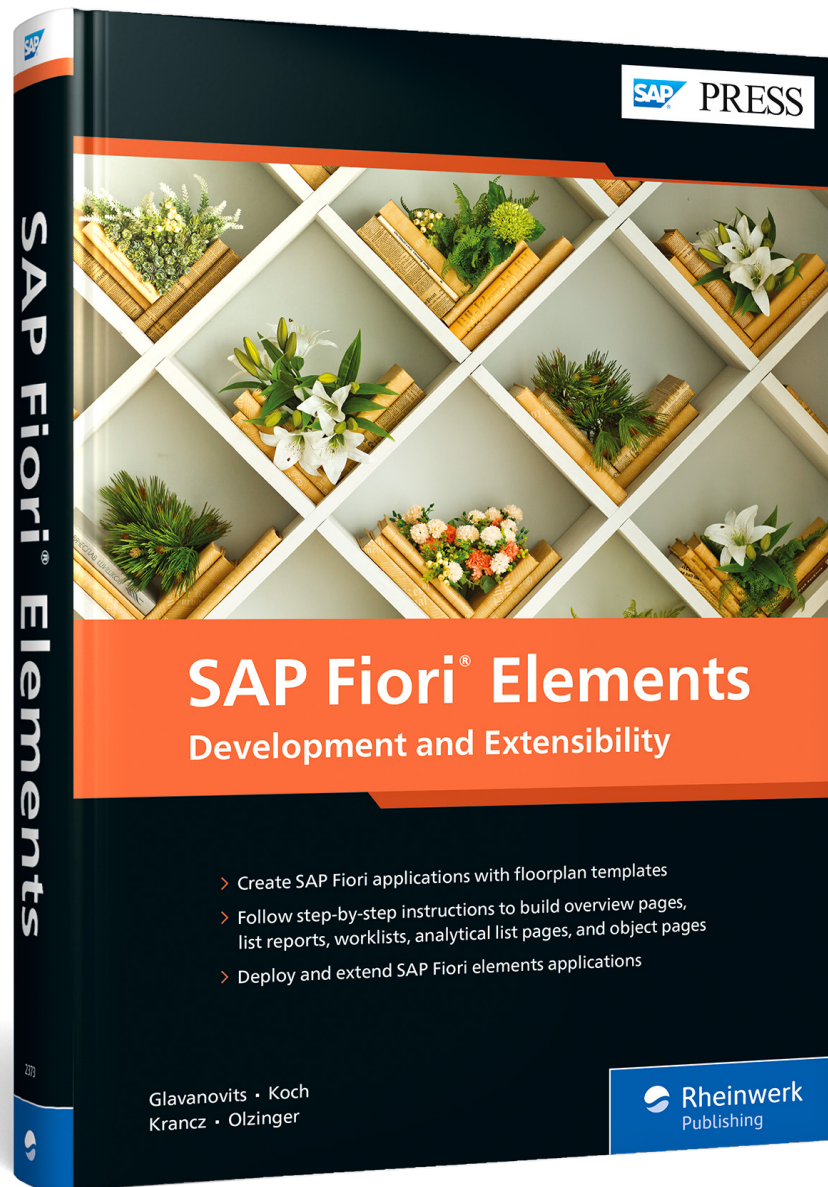
- Multiple view layout ..... 203
- My Inbox ..... 33, 51
- N**
- Namespace ..... 103
- Navigation ..... 385
  - basic* ..... 388
  - consuming parameters* ..... 389
  - CrossApplicationNavigation service* ..... 389
  - entity attributes with fixed URL* ..... 394
  - entity attributes with link* ..... 393
  - external* ..... 391
  - fixed URL and fixed text* ..... 395
  - intent-based* ..... 55
  - link entity attribute directly* ..... 394
  - output fixed text* ..... 393
- Navigation bar ..... 158
- Network graph ..... 31
- No-code development ..... 35
- Node Package Manager ..... 101, 225
- Node.js ..... 99–101
- NoSQL ..... 101
- Notification
  - list* ..... 56
  - service* ..... 390
- O**
- Object marker ..... 273
- Object page ..... 29, 158, 160
  - CDS views* ..... 288
  - custom action* ..... 361
  - develop in SAP Business Application Studio* ..... 297
  - develop in SAP Web IDE* ..... 289
  - facet annotations* ..... 280
  - header annotations* ..... 277
- Object page footer ..... 287
- Object-oriented programming (OOP) ..... 30
- OData ..... 30, 41, 65, 211, 308
  - annotations* ..... 94
  - development* ..... 86
  - expand* ..... 83
  - metadata* ..... 93, 114
  - standard* ..... 66
  - version* ..... 152
- OData annotation ..... 199
- OData model
  - definition* ..... 185
  - naming* ..... 197
- OData service ..... 28, 104, 112, 114, 123, 183, 184, 219, 252–254
  - manage* ..... 85
  - register* ..... 88
- Onboarding ..... 316
- Online analytical processing (OLAP) ..... 239
- On-premise ..... 32
  - deployment* ..... 307, 308
- OpenUI5 ..... 40, 41
- OPEX ..... 98
- Organization for the Advancement of Structured Information Standards (OASIS) ..... 65
- orion.exe file ..... 135
- Outbound navigation ..... 385
- Overview page ..... 29, 152, 160, 163
  - annotations* ..... 172
  - CDS views* ..... 173
  - data update* ..... 171
  - design best practices* ..... 171
  - develop in SAP Business Application Studio* ..... 175
  - develop in SAP Web IDE* ..... 193
  - extend* ..... 345
  - register cards* ..... 180
  - structure* ..... 163
  - usage* ..... 169
- overviewpageext.controller.js file ..... 349
- P**
- Package field ..... 380
- Page field ..... 356
- Page-level variant management ..... 200
- Pagination ..... 84
- Personalization ..... 60, 61, 309
  - service* ..... 390
- Prebuilt module ..... 35
- Presentation Variant Qualifier field ..... 257
- Production system ..... 307
- Programming model ..... 97
- Project creation wizard ..... 123
- Project Explorer ..... 143, 178, 195, 196, 255, 340
- Projection ..... 105
  - layer* ..... 112
- Push notification ..... 316
- Q**
- Query option ..... 78
- Quick theming ..... 57
- Quick view card ..... 170

- R**
- React ..... 39, 42, 43
- Reporting and monitoring ..... 35
- Request
  - primary keys* ..... 73
- Resizable card layout ..... 166, 169
- Response formatting ..... 84
- Responsive design ..... 40
- REST
  - APIs* ..... 66
  - client* ..... 90
- REST services ..... 41
- Reuse component ..... 50
- Rich client ..... 43
- Role
  - assignment* ..... 323
  - collection* ..... 335
  - SAP Cloud Portal service* ..... 330
  - SAP launchpad service* ..... 334
- Role-based access ..... 51
- Row Selection field ..... 357
- RSS feed ..... 319
- S**
- SADL ..... 88
- Safe Mode setting ..... 376
- Same Origin Policy ..... 309
- SAML ..... 313
- SAML 2.0 ..... 34
- SAP BTP, Cloud Foundry environment ..... 297, 311, 332
- SAP BTP, Neo environment ..... 311, 316, 327
- SAP Build Apps ..... 35, 36
- SAP Business Application Studio ..... 44, 117, 125, 215, 222, 228, 297, 370
  - adaption project* ..... 371
  - add extension* ..... 133
  - benefits and limitations* ..... 134
  - create list report app* ..... 222
  - deployment wizard* ..... 379
  - dev spaces* ..... 127
  - develop analytical list page* ..... 252
  - develop overview page* ..... 175
  - extension* ..... 62
  - extension types* ..... 132
  - functions* ..... 131
  - guided development* ..... 340
  - import apps* ..... 129
  - import from SAPUI5 ABAP repository* ..... 130
  - open* ..... 126
- SAP Business Application Studio (Cont.)
  - start page* ..... 129
- SAP Business ByDesign ..... 98
- SAP Business Suite ..... 310
- SAP Business Technology Platform (SAP BTP) ..... 21, 40, 99, 117, 118, 122, 183, 194, 289, 311, 327
- SAP Business Warehouse ..... 239
- SAP Business Workflow ..... 207
- SAP Cloud Application Programming Model ..... 21, 99, 100, 109
- SAP Cloud Platform ..... 99
- SAP Cloud Portal service ..... 311, 327
- SAP Fiori ..... 21, 26, 39, 43, 44, 118
  - 1.0* ..... 46
  - 2.0* ..... 47
  - 3.0* ..... 47
  - apps reference library* ..... 59
  - catalog* ..... 316
  - client* ..... 315
  - extension guide* ..... 342
  - for Android* ..... 47
  - for iOS* ..... 47
  - group* ..... 316
  - versions* ..... 46
- SAP Fiori design guidelines ..... 26, 44, 49, 151, 163, 199, 238, 271
- SAP Fiori elements ..... 28, 30, 35, 101, 146, 151
  - generated app* ..... 106
  - object page* ..... 271
- SAP Fiori launchpad ..... 28, 36, 44, 45, 51, 87, 308, 315, 316
  - catalog* ..... 52, 316
  - designer* ..... 317
  - group* ..... 52, 316, 322
  - Neo environment* ..... 327
  - personalization* ..... 52
  - role* ..... 316
  - spaces* ..... 51
- SAP Gateway ..... 66, 123
  - client* ..... 90
- SAP GUI ..... 22, 51
  - for HTML* ..... 314
- SAP HANA ..... 101
  - Extended Application Services*
    - Advanced Model* ..... 99
    - XSJS* ..... 99
- SAP HANA Cloud Platform ..... 99
- SAP HANA Extended Application Services ..... 99
- SAP Jam ..... 201
- SAP launchpad service ..... 332
- SAP Mobile Platform SDK ..... 316

- SAP NetWeaver ..... 23, 308
  - SAP NetWeaver Application Server ..... 97
  - SAP Quartz theme ..... 56
  - SAP S/4HANA ..... 22, 47, 310
  - SAP Screen Personas ..... 42
  - SAP Store ..... 24
  - SAP Web IDE ..... 44, 117, 122, 160, 211, 219
    - benefits and limitations* ..... 125
    - create list report project* ..... 211
    - create new project* ..... 120
    - develop analytical list page* ..... 264
    - develop object page* ..... 289
    - develop overview page* ..... 193
    - extensions* ..... 124
    - full stack* ..... 117
    - functions* ..... 123
    - import apps* ..... 120
    - software development kit (SDK)* ..... 123
    - start* ..... 118
  - SAP Web IDE, personal edition ..... 135
    - benefits and limitations* ..... 140
    - floorplans* ..... 136
    - import projects* ..... 136
    - install* ..... 135
    - integrating SAPUI5 library* ..... 138
    - OData services* ..... 137
  - SAPUI5 ..... 24, 35, 39, 118, 121, 160, 314
    - ABAP repository* ..... 87, 121
    - display width* ..... 50
    - release cycle* ..... 40
    - version* ..... 137, 310
    - vs. SAP Fiori* ..... 44
  - SAPUI5 flexibility ..... 59, 309
    - availability* ..... 60
  - SAPUI5 freestyle ..... 391
  - SAPUI5 library ..... 344
  - SAPUI5 Visual Editor ..... 62
  - SASS ..... 57
  - Save ..... 276
  - Scrum ..... 26
  - Search ..... 201
  - Section ..... 274, 281
  - Security Assertion Markup
    - Language (SAML 2.0) ..... 43, 313
  - Segmented button ..... 203
  - Select box ..... 203
  - Select query ..... 84
  - Select System field ..... 380
  - Semantic action ..... 385
  - Semantic annotation ..... 285, 397, 400
  - Semantic object ..... 54, 317, 385
  - Semantic object navigation ..... 317
  - Service Adaption Definition
    - Language (SADL) ..... 88, 251
  - Service binding ..... 113
  - Service document ..... 67
  - Service metadata document ..... 67
  - Service modeler ..... 131
  - service.cds file ..... 104
  - Share menu ..... 201
  - Shell ..... 51
  - Simple content layout ..... 202
  - Simple control ..... 50
  - Single sign-on (SSO) ..... 34, 313
  - Single- sign-on (SSO) ..... 43
  - Site manager ..... 311
  - Smart chart ..... 275
  - Smart control ..... 27, 28, 50
  - Smart filter ..... 152
  - Smart filter bar ..... 164, 165, 201
  - Smart link contact ..... 274
  - Software as a service (SaaS) ..... 23, 43, 101, 307
  - Software development kit (SDK) ..... 101
  - Sorting ..... 83
  - Source code management ..... 306
  - Spring framework ..... 101
  - SQLite ..... 101
  - Stack card ..... 170
    - add* ..... 191
    - annotation* ..... 191
    - definition* ..... 191
    - preview* ..... 193
  - Start-up parameter ..... 389
  - Sticky ..... 206
  - String function ..... 80
  - Subsection ..... 274
  - Subtitle ..... 272
- ## T
- Tab bar ..... 156, 158
  - Table ..... 244, 275
    - toolbar* ..... 206
    - types* ..... 245
  - Table card ..... 170
    - add* ..... 178
    - definition* ..... 181
  - Table-only view ..... 244
  - Tablet ..... 240
  - Tabular display ..... 203
  - Target mapping ..... 54, 55, 320
  - Template ..... 27
  - Template Wizard ..... 108
  - Theme ..... 33

- Theming ..... 33
  - Tile ..... 51, 317
    - create* ..... 319
    - dynamic* ..... 319
    - news* ..... 319
    - static* ..... 319
    - type* ..... 319
  - Title ..... 272
  - Transaction
    - /IWFND/MAINT\_SERVICE* ..... 87, 88, 92, 289
    - /UI2/FLP* ..... 326
    - PFCG* ..... 323
    - RSA1* ..... 239
    - SE11* ..... 144
    - SEGW* ..... 86
    - SUI\_SUPPORT* ..... 382
  - Transport layer security (TLS) ..... 313
  - Transport management ..... 307
  - Transport request number field ..... 380
- ## U
- UI5 Web Components ..... 42
  - Uniform Resource Identifier (URI) syntax ..... 66
  - Update mode ..... 201
  - User experience (UX) ..... 24, 32, 33
  - User interface (UI) ..... 344
    - add-on* ..... 308
    - annotation* ..... 36, 386, 397
    - annotations* ..... 397
    - control* ..... 40
    - framework* ..... 39
    - history* ..... 22
    - requirements* ..... 32
    - service* ..... 114
    - theme designer* ..... 33, 57
    - vocabulary* ..... 112
- ## V
- Value help ..... 243, 399, 400
  - Value List Property field ..... 257
  - Variant ..... 62
  - Variant management ..... 165, 200, 204, 275
  - Vendor lock-in ..... 24
- ## Version control
- Version control ..... 306
  - View extension ..... 59
  - Visual filter ..... 238, 242, 269
    - annotations* ..... 247
    - define* ..... 255
    - local annotations* ..... 256, 258
  - Visual modeling tools ..... 35
  - Visual Studio Code ..... 44, 102, 108, 118, 125, 132, 134, 145
    - benefits and limitations* ..... 147
    - create SAP Fiori elements app* ..... 146
    - SAP Fiori extensions* ..... 146
  - Vocabulary ..... 92
  - Voter build ..... 306
  - Vue ..... 42
  - Vue.js ..... 39, 43
- ## W
- Web app ..... 43
  - Web Dynpro ..... 23
    - for ABAP* ..... 51, 314
  - Web framework ..... 43
  - Web interface ..... 23
  - Web technology platform ..... 45
  - WebIDEUsage parameter ..... 138
  - Work item ..... 155
  - Worklist ..... 29, 155, 160, 199, 209, 227
    - annotations* ..... 210
    - content area* ..... 207
    - create* ..... 228
    - develop in SAP Business Application Studio* ..... 227
    - mode* ..... 206
- ## X
- XML ..... 41
  - XML fragment ..... 368
  - XML response ..... 71, 73
- ## Z
- ZAUTHOR\_CV ..... 226





Glavanovits, Koch, Krancz, Olzinger

## SAP Fiori Elements

### Development and Extensibility

413 pages | 03/2023 | \$89.95 | ISBN 978-1-4932-2373-2

 [www.sap-press.com/5641](http://www.sap-press.com/5641)



**Rene Glavanovits** is an SAP consultant and developer at CloudDNA GmbH, an SAP partner in Austria. He specializes in the latest SAP technologies, specifically in the development of full-stack applications with SAP Fiori, SAPUI5, OData, core data services (CDS), and SAP Cloud Application Programming Model.



**Martin Koch** is the managing director of CloudDNA GmbH, an SAP partner in Austria. His focus is on SAP Business Technology Platform (SAP BTP), user experience, integration, and security. In addition to his SAP-related work, he is an active champagne importer and dealer in Austria.



**Daniel Krancz** is a software developer and consultant at CloudDNA GmbH. His focus is full-stack development with SAPUI5/SAP Fiori, OData, SAP Cloud Application Programming Model, and ABAP RESTful application programming model, as well as mobile development. He has been teaching SAP trainings since 2019 and has been named a certified trainer and consultant for his achievements by SAP.



**Maximilian Olzinger** is a software developer and consultant at CloudDNA GmbH. He is an SAP Certified Development Associate and has handled projects for companies in all industries with great success. He conducts trainings in the areas of SAP Fiori, ABAP, and web development. He has developed several SAP standard trainings for SAP Fiori.

*We hope you have enjoyed this reading sample. You may recommend or pass it on to others, but only in its entirety, including all pages. This reading sample and all its parts are protected by copyright law. All usage and exploitation rights are reserved by the author and the publisher.*