

UX403

SAP Fiori Elements Development

PARTICIPANT HANDBOOK INSTRUCTOR-LED TRAINING

Course Version: 02
Course Duration: 2 Day(s)
e-book Duration: 5 Hours 50 Minutes
Material Number: 50144132

SAP Copyrights and Trademarks

© 2018 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see <http://global12.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

Typographic Conventions

American English is the standard used in this handbook.

The following typographic conventions are also used.

This information is displayed in the instructor's presentation	
Demonstration	
Procedure	
Warning or Caution	
Hint	
Related or Additional Information	
Facilitated Discussion	
User interface control	<i>Example text</i>
Window title	<i>Example text</i>

Contents

vii Course Overview

1 Unit 1: SAP UX Strategy

2	Lesson: Describing SAP User Experience Strategy
17	Lesson: Explaining SAP User Experience Tools and Technologies
20	Lesson: Describing SAP User Experience Use Case for Building Fiori-like Apps

34 Unit 2: SAP Fiori Elements, Overview

36	Lesson: Explaining the Architecture of Fiori Elements
44	Lesson: Explaining Templates for Fiori Elements
49	Lesson: Exploring the Development Environment
51	Lesson: Exploring the Basic Process of Building Fiori Elements Application
57	Lesson: Using the Core Data Services (CDS) View
63	Lesson: Using the Service Adaption Definition Language (SADL)
66	Lesson: Explaining Metadata Extension
68	Lesson: Learning Scenarios of Fiori Elements Implementation

77 Unit 3: List Report

78	Lesson: Explaining Basic Annotations for List Report
83	Lesson: Using Searching and Filtering Data
88	Lesson: Providing the Value Help
91	Lesson: Explaining Variant Management

99 Unit 4: Object Page

100	Lesson: Using Basic Annotations for Object Pages
102	Lesson: Using Header Facets for Object Pages
108	Lesson: Using Sections and Facets in Object Pages

118 Unit 5: Advanced Topics of List Report and Object Page

119	Lesson: Explaining Navigation Concept and Annotations
124	Lesson: Using Data Visualization
127	Lesson: Creating Charts
130	Lesson: Performing CURD operations with BOPF

138 Unit 6: Overview Page

139	Lesson: Getting an Overview of the Overview Page (OVP)
-----	--

146 Unit 7: Analytical List Page**147** Lesson: Getting an Overview of the Analytical List Page

Course Overview

TARGET AUDIENCE

This course is intended for the following audiences:

- Application Consultant
- Developer
- Development Consultant
- Solution Architect
- User Experience Designer



The slide features a dark grey header section on the left containing the text "UNIT 1" in white. To its right is a large yellow section containing the text "SAP UX Strategy" in black.

Lesson 1

Describing SAP User Experience Strategy 2

Lesson 2

Explaining SAP User Experience Tools and Technologies 17

Lesson 3

Describing SAP User Experience Use Case for Building Fiori-like Apps 20

UNIT OBJECTIVES

- Describe SAP User Experience Strategy
- Explain SAP User Experience Tools and Technologies
- Describe SAP User Experience Use Case for Building Fiori-like Apps

Unit 1

Lesson 1

Describing SAP User Experience Strategy



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Describe SAP User Experience Strategy

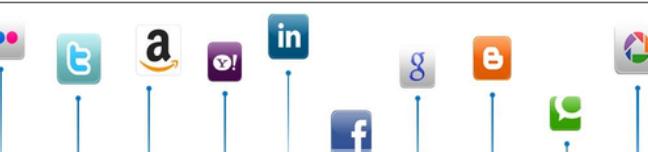
User Experience Strategy

Scenario

As a Developer, you are requested by the business process owners to permanently improve the user experience.

Creating user interfaces is a critical aspect in the user experience. In this training, you will extend your knowledge in developing a great SAPUI5 user interfaces.

The World is Changing



Consumer user experience is the new standard for enterprise applications ...

Transform the enterprise experience

Complex and feature-rich experience must be replaced by a simple, intuitive, and mobile experience.

Figure 1: The World is Changing

Take for example, purchasing something personal online like a television. That same user would expect if they needed to order something from work, say a new cell phone that the process would be the same in terms of ease and simplicity. This idea has set the bar for expectations when using business applications.

Consumerization is the growing tendency for new information technology to emerge first in the consumer market and then spread into business and government organizations.

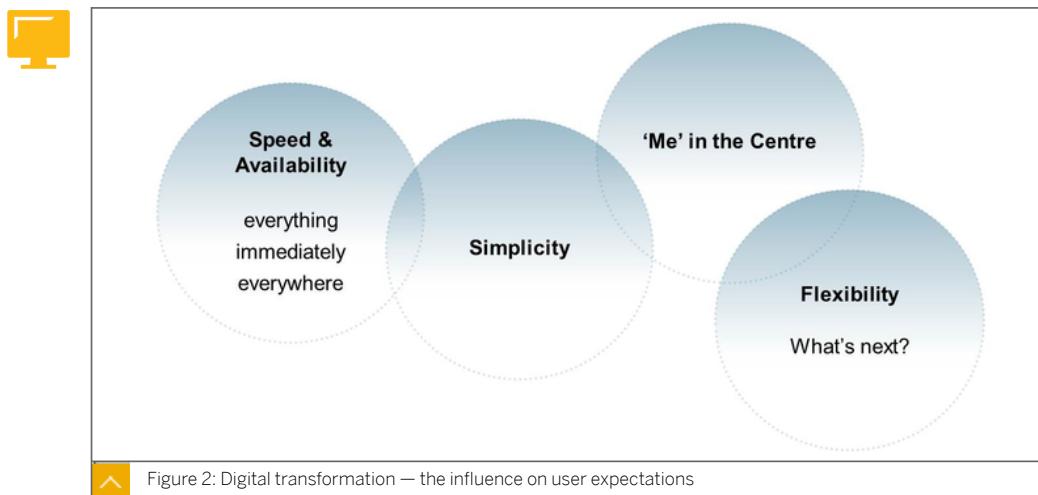
This is evidenced today more so than ever before. People are used to the application user experience they interact with their daily applications such as google and social media apps.

This experience has been embraced and accepted by people to the point where it is now the new standard. People want to have the same easy, feature rich experience they have with their personal sites in the business place.

SAP has recognized this change in society and made it our goal to meet this new standard.

Trying to transform the user experience

Digital transformation — the influence on user expectations



SAP developed a user experience strategy consisting of 3 main components: New, Renew and Enable.

During our research phase we realized that most users still use the SAP GUI to access applications.

The GUI contains approximately 300,000 screens and consist of a vast number of functionalities.

We looked at all the functionality offered in our GUI and developed a list of the most frequently used applications, namely manager and employee functions such as leave request or travel expenses. We decided it was time to renew these top scenarios to make good on our mission.

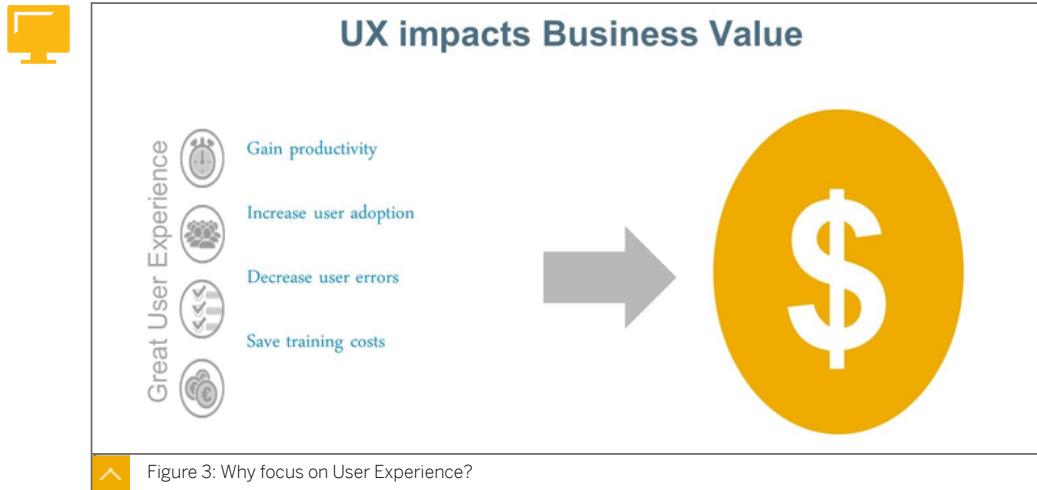
It was from this idea that SAP Fiori was born.

We also decided to renovate business suite programs and enable customers to improve their user experience on their own.

For example, SAP developed SAP Screen Personas which allows customers to optimize and simplify any screen in the GUI. While developing SAP Fiori we also decided to renovate our business suite programs and provide enablement tools to enable customers to improve their user experience on their own - take for example, SAP screen personas that allows customers to optimize and simplify any screen in the GUI.

While we were working on renewing our existing solutions and enabling our customers we also continue to develop new applications. All the while SAP is continuing to create new innovative applications to fulfill customer demands and needs.

Not looking at creating a new business process, looking at creating a new user experience without changing the backend ("lipstick on a pig").

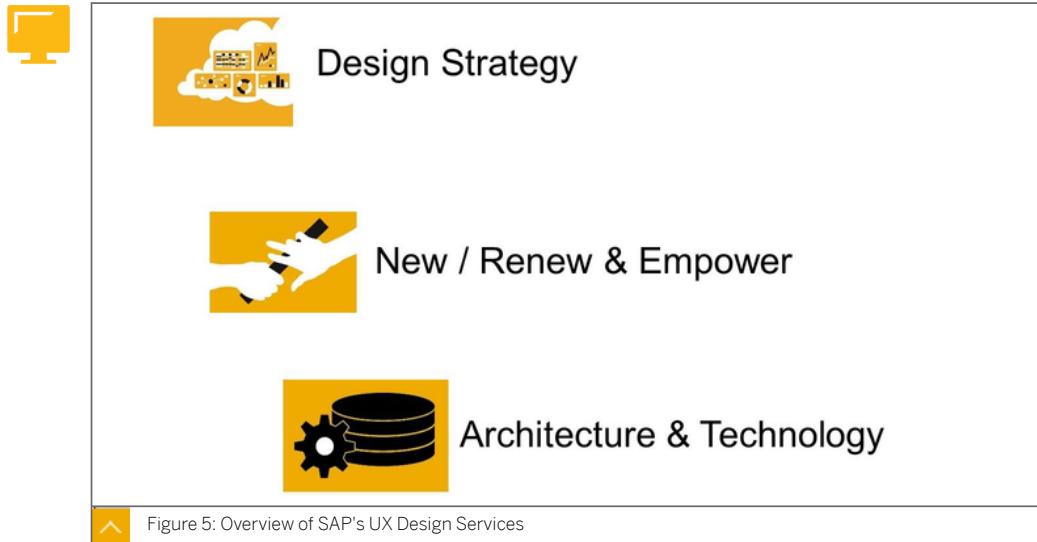
Why focus on User Experience?

The figure shows the influences on monetary and human value.

SAP User Experience Design Services

Finally, we will look deeper into the User Experience design services component.

Overview of SAP's UX Design Services



SAP Identified several leads from our customers and they asked us for advice to understand the SAP strategy and translate it to their reality: they asked for services to realize: implement, adapt and optimize the user experience of existing software.

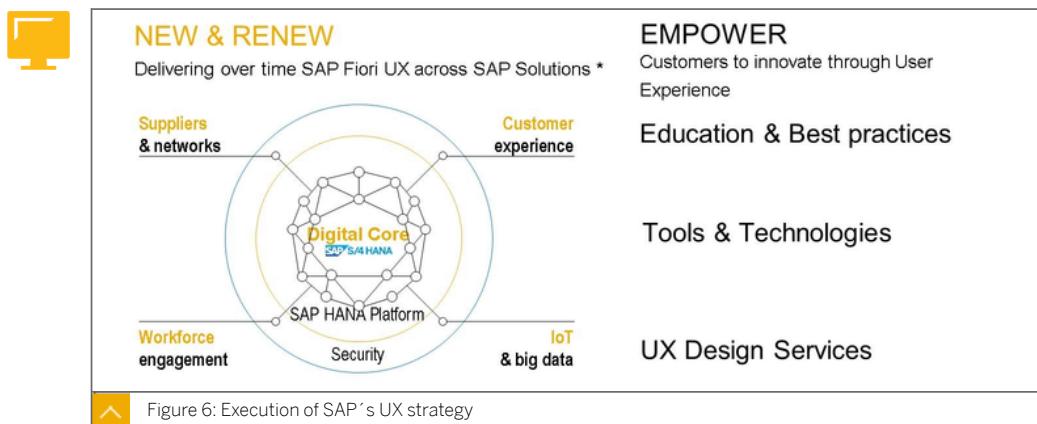
If you achieved results such as with screen personas or Fiori the next level is to have customers build up skills on their own or empower their organization for a user experience strategy.

The last level is for the customer to become really innovative. This is the final goal of all customers, to be more innovative and they can achieve that by designing new products, looking for new services, etc.

All of this is powered by design thinking.

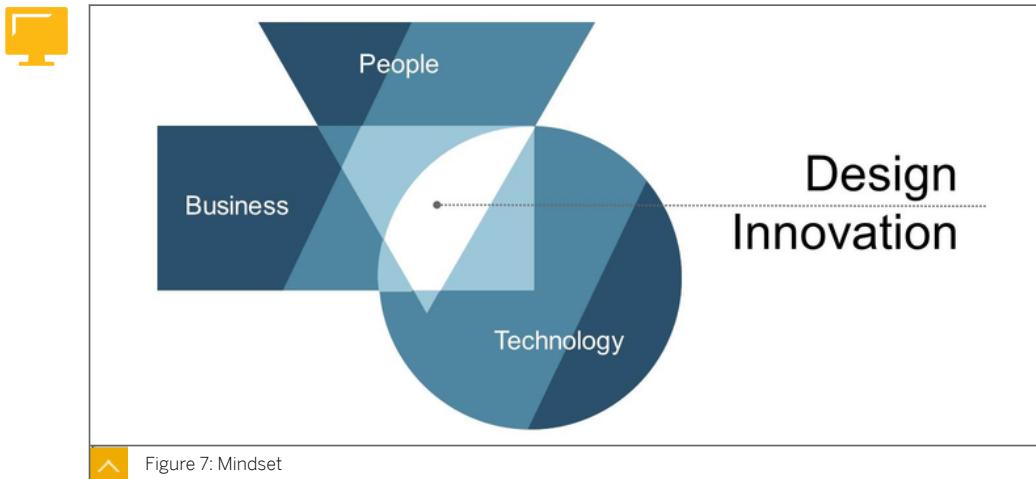
In Summary: the design services that SAP offers are to advise about the strategy and the value of a solid user experience.

Execution of SAP's UX strategy

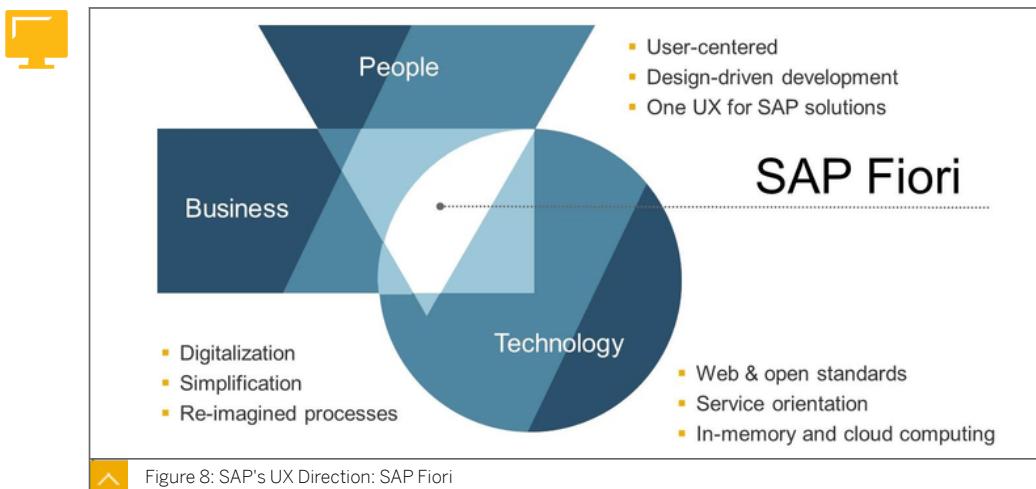


Unit 1: SAP UX Strategy

The figure shows the execution of SAP's UX strategy

Mindset

SAP expanded their research to include the most commonly used functionality across all Lines of Business. With the first set of Fiori apps we focused on the most commonly used business functions and those focused on the HR, workflow and SRM lines of business. Here you can gain an overview of the other business lines we expanded to such as research and development and finance.

SAP's UX Direction: SAP Fiori

SAP Fiori is a collection of apps that represents the new user experience from SAP.

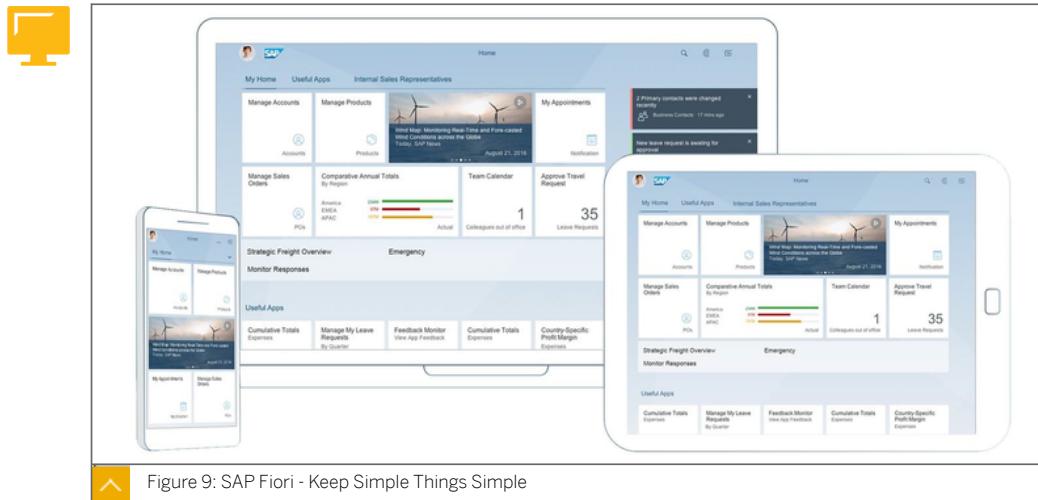
At Sapphire 2013 a set of 25 apps targeting the main business functions of managers and employees, such as approvals and requests were released.

SAP Fiori is 100% geared toward business users. We wanted to provide a consistent user experience. SAP Fiori assures that people, both employees and managers have consistent,

Lesson: Describing SAP User Experience Strategy

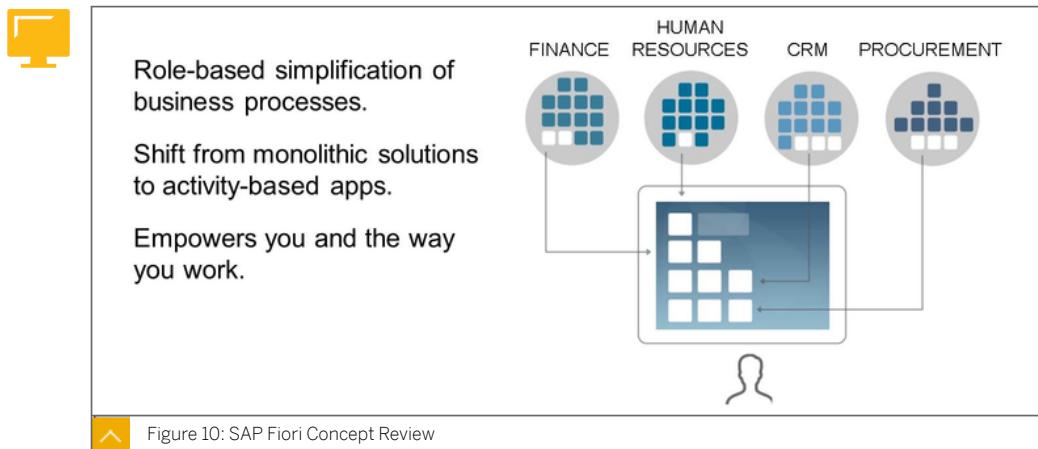
coherent, simple and intuitive user experience across multiple devices which allows them to work smarter, more efficiently and deliver on business objectives.

SAP Fiori - Keep Simple Things Simple



If you look at the tablet in this image you'll see the detail list on the left side and the main pane. Now look to the mobile phone and notice only the main pane is visible. The detail list can be accessed by swiping the phone but both panes will not fit at once. Responsive design is credited for automatically completing the look on our UI framework.

SAP Fiori Concept Review



The figure shows the entry to the SAP Fiori Concept.

Idea of SAP Fiori



- SAP Fiori is more than just a collection of apps, it represents the new SAP User Experience paradigm based on SAPUI5 framework
- Like SAP Fiori, SAPUI5 offers the developer opportunities to develop various business roles into simple, easy-to-use experiences for SAP software functions, and works seamlessly on desktop, tablet, or smartphone

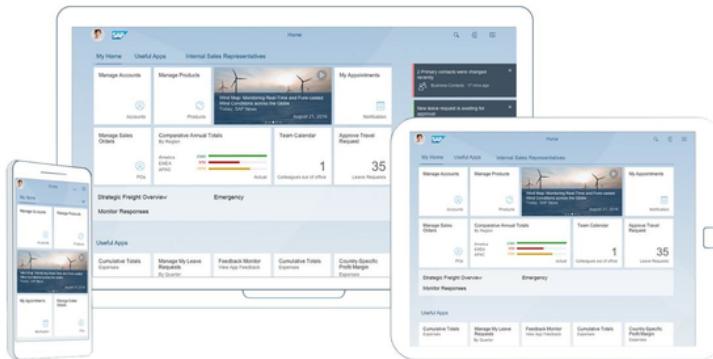


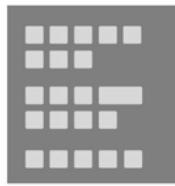
Figure 11: Idea of SAP Fiori

The figure shows examples of SAP Fiori on various devices.

SAP Fiori, Concept Review



Single point of access
for end users



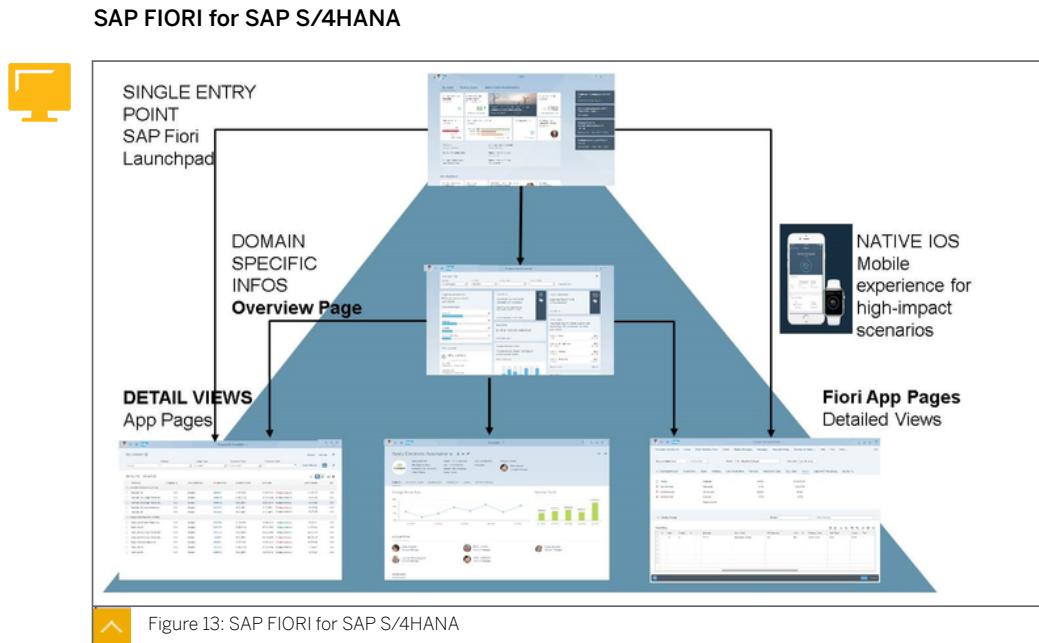
SAP delivered software

Custom built
and 3rd party
software



Figure 12: SAP Fiori, Concept Review

SAP Applications can be provided on-premise or as cloud applications. The user gets access to these applications using a single point of entry. This single point of entry is the SAP Fiori Launchpad (FLP).

**Level 1**

Full role scope accessible via FLP, but mainly integration of legacy UIs.

Example: Fiori Launchpad, Search, ...

Level 2

Domain specific info & action - All key business objects or domain areas.

Example: List Reports, Overview Pages, Work Lists, ...

Level 3

Fiori UX for key use cases.

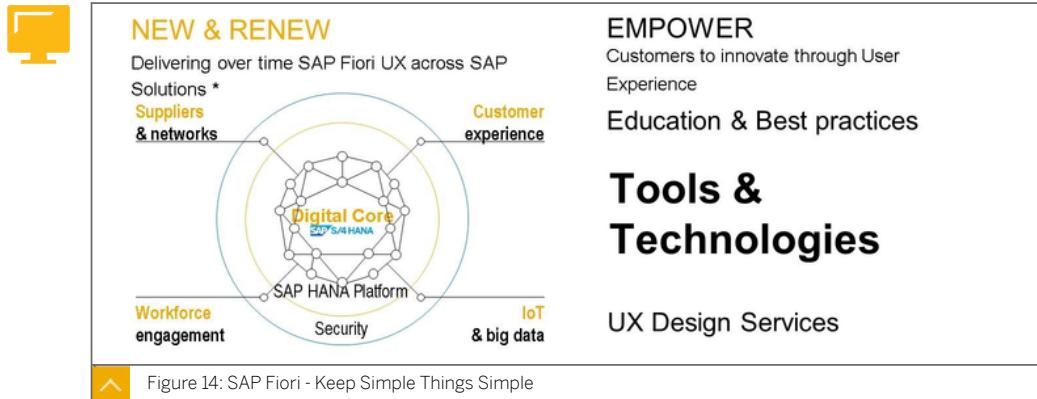
Example: Object Pages, Edit Pages...

Level 4

Fiori Apps, WebGUI with SAP Screen Personas, WebDynpros, 3rd Party UIs, Partner UIs...

Example: Fiori apps für all use cases, Visually harmonized classic UIs

SAP Fiori - Keep Simple Things Simple



SAP offers a new and broad portfolio of UX design services that guide organizations into a user centered design perspective. We help you define and execute the best UX strategy for your business using proven design methodologies like design thinking and user-centered design.

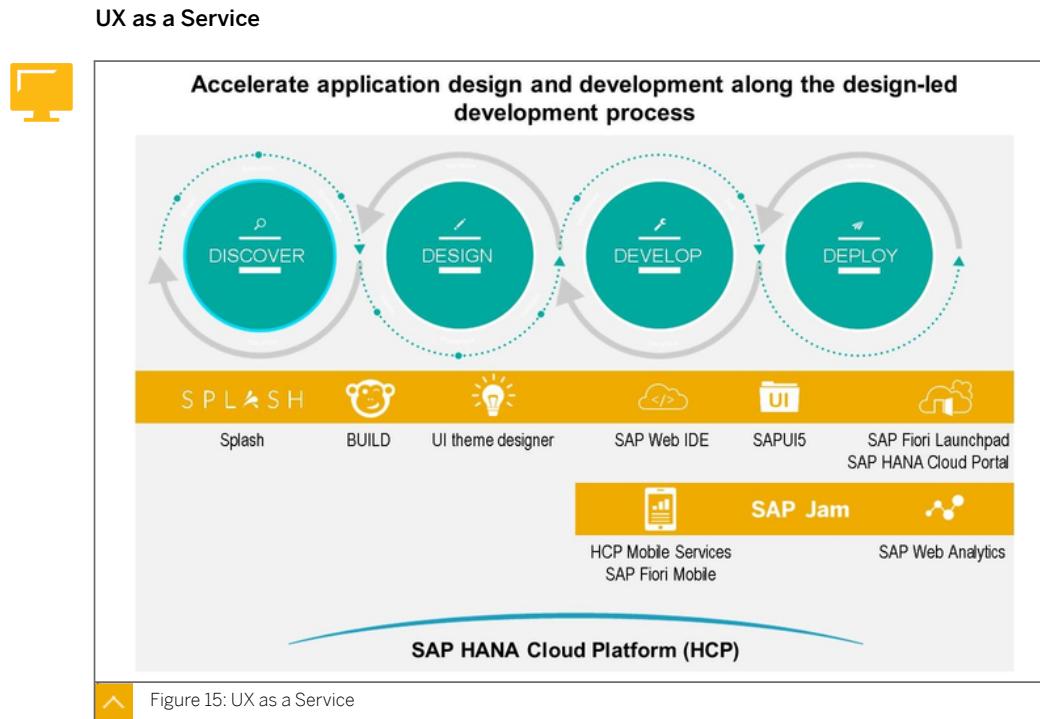
SAP Fiori is the way we renewed the most widely used scenarios.

SAP Fiori uses HTML5 and SAPUI5 technology and it can run on all devices.

Depending on the device, SAP Fiori adapts visualization to the device specifically and uses responsive design.

If you look at the tablet in this image you'll see the detail list on the left side and the main pane. Now look to the mobile phone and notice only the main pane is visible. The detail list can be accessed by swiping the phone but both panes will not fit at once. Responsive design is credited for automatically completing the look on our UI framework.

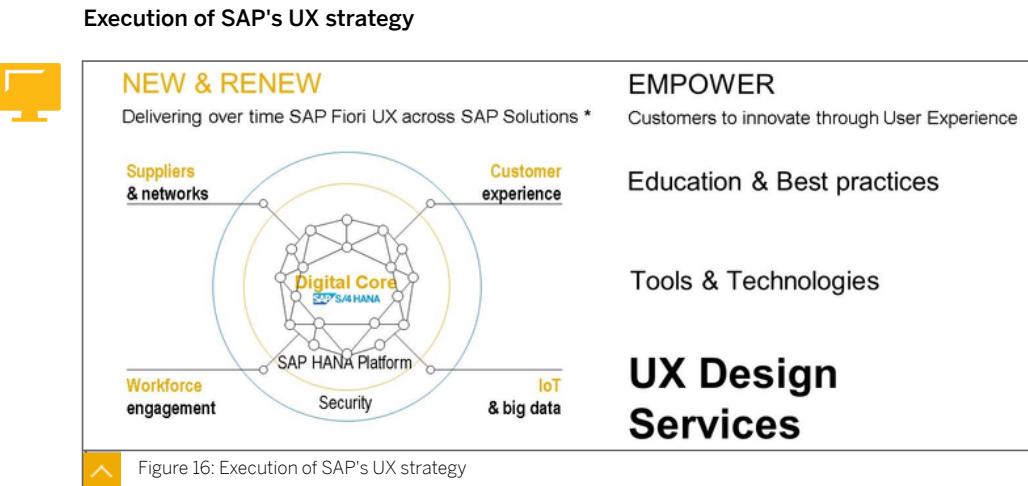
An important thing to note with SAP Fiori is it can be deployed in the customers existing landscape. SAP customers running ECC 6.0 or Suite on Hana will need gateway to the backend system and add-ons making it an easy to deploy and use solution.



SAP is committed to designing role-based applications that address the needs of our end users across all lines of business, tasks, and devices. We believe this is the key to a great user experience. But how do we guarantee a solid and consistent design for our customers and end users? The answer is **SAP's design-led development process**.

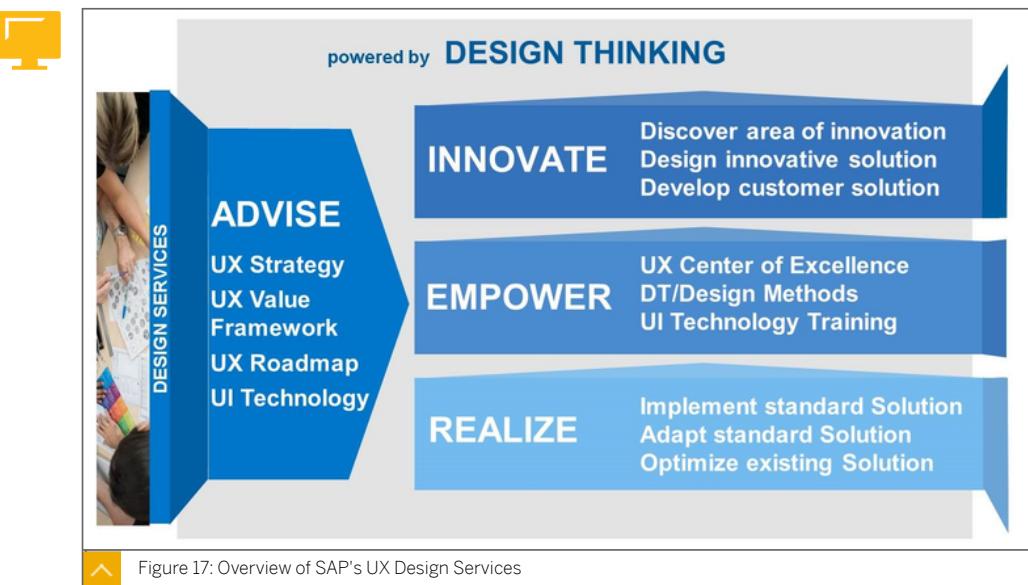
Design-led development takes advantage of proven design thinking methods to achieve an optimal user experience. The process spans the entire development lifecycle, is simple and easy to follow, and provides a solid basis for scaling design as a whole. It fosters unity between designers and developers, while ensuring that the needs of the end user are addressed at every step along the way. You can get more details under <https://experience.sap.com/fiori-design-web/design-led-development-process-external/>

The design-led development process is supported by various tools and technologies of SAP. Like Splash, SAP BUILD, UI theme designer, SAP Web IDE and SAPUI5.



SAP offers a new and broad portfolio of UX design services that guide organizations into a user centered design perspective. We help you define and execute the best UX strategy for your business using proven design methodologies like design thinking and user-centered design.

Overview of SAP's UX Design Services

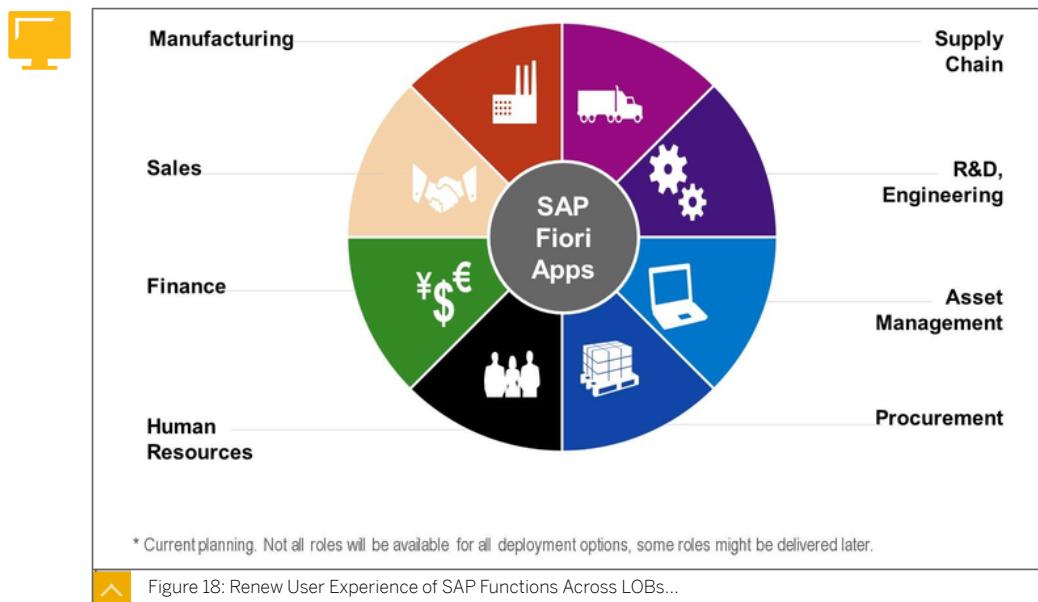


SAP has Identified several leads from our customers and they asked us for advice to understand the SAP strategy and translate it to their reality:

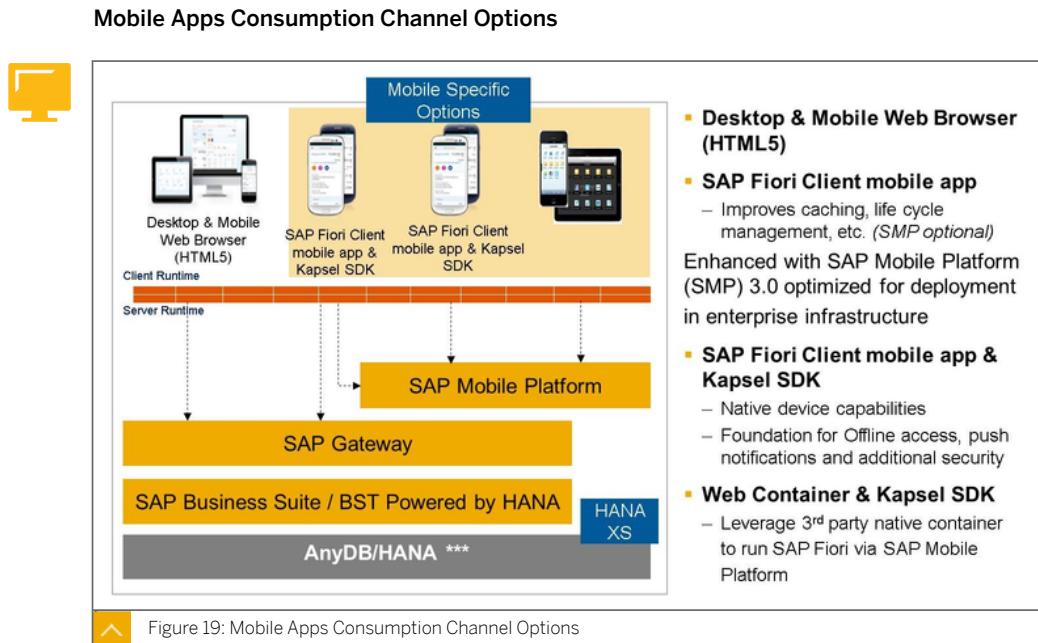
- Asked for services to realize: implement, adapt and optimize the user experience of existing software

- If you achieved results such as with screen personas or Fiori the next level is to have customers build up skills on their own or empower their organization for a user experience strategy
- The last level is for the customer to become really innovative. This is the final goal of all customers, to be more innovative and they can achieve that by designing new products, looking for new services etc.
- All of this is powered by design thinking.
- In Summary: the design services that SAP offers are to advise about the strategy and the value of a solid user experience.

Renew User Experience of SAP Functions Across LOB



The figure shows the various areas for which the renewed User Experience of SAP Functions Across LOBs is available.



Using SAP HANA allows Fiori to have all the same benefits of SAP HANA and allowed us to offer some of the awesome new features such as analytical data and KPIs.

Depending on the apps that are being used some simply run on the SAP HANA platform while others leverage the XS engine part of HANA - we will dig into this topic deeper in a later portion of this class.

Gateway is the plumbing. Can link Odata services to HTML 5 with some work. Is not difficult just tedious.

There has been a huge interest in the field for integration of the SAP Mobile Platform and SAP Fiori.

With SMP 3.0 we have included the option to integrate Fiori using Kapsel/Web Container:

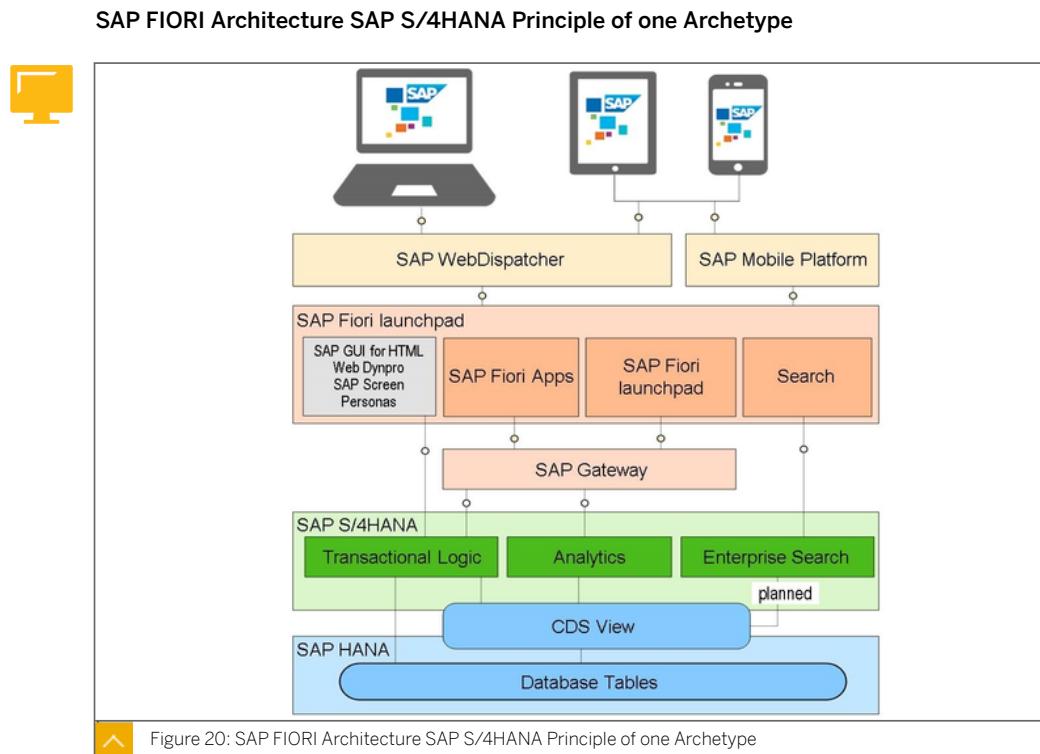
- This allows for native device capabilities
- Offline access, push notifications and additional security

Using iOS and Android with Kapsel (written in Cordova) can view documents. Android uses a back button to navigate. iOS, you have to leave Fiori App, open SAP Mobile Documents, then allow "mutil-use"

Right now Fiori client and Kapsel are separate plugins, (wave 2) will be part of Kapsel SDK eventually.

No business data is stored within the Fiori client.

SAP Mobile Platform integration is limited right now because Fiori is web based. When users get an app to launch (in the mobile device scenario) push notification and offline accessibility will be needed and SMP will apply this. Will also be used for security.



SAP Fiori Architecture for SAP S/4HANA consists of only one archetype for transactional, analytical and search:



Fiori technology components

- SAP Fiori launchpad
- Metadata driven UIs - Smart Controls & Smart Templates

ABAP infrastructure components

- Draft Infrastructure for transactional Logic
- SADL for CDS read access
- Analytical Engine (embedded BW) for analytical CDS access
- SAP Gateway for OData exposure

CDS Views (ABAP managed)

- Uniform Business Object Modelling
- Central repository for Metadata

Key Enablement Tools

SAP Screen Personas
(for SAP GUI screens)

Floorplan Manager
(for FPM screens*)

FOCUS OF THIS COURSE

NWBC & Side Panel
(all screens)

Theme Designer
(all screens)

SAPUI5 Application Development Tools

further tools available

* for CRM Web UI screens use CRM WebUI

Figure 21: Key Enablement Tools

We just reviewed SAP Screen Personas.

Floorplan Manager which is based on ABAP web dynpro and allows customers to build new screens and adapt floorplan manager screens.

SAP UI5 application development tools are available so customers can build, or adapt fiori apps on their own or build their own ui5 applications.

NW business client, side panel and we offer a theme designer which is a tool that allows to adapt the branding of customer specific branding (colors, fonts, logos, change stylesheets etc.).



LESSON SUMMARY

You should now be able to:

- Describe SAP User Experience Strategy

Unit 1 Lesson 2

Explaining SAP User Experience Tools and Technologies

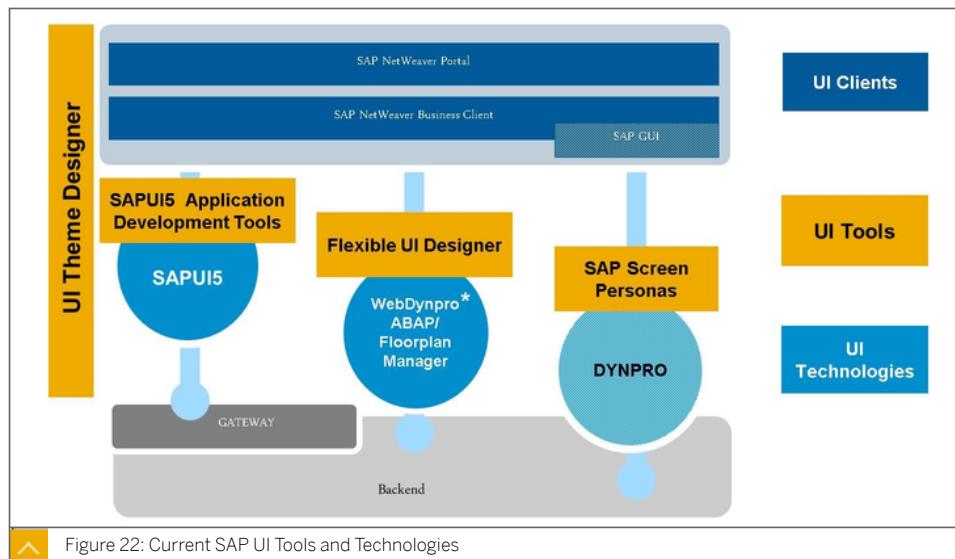


LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain SAP User Experience Tools and Technologies

User Experience Tools and Technologies



* Harmonized in Run & Design Time with WebUIF

Overview of key UI technologies currently (as of the release of this course) offered by SAP:

SAP SAPUI5 and UI5 application development tools to change adapt or develop new applications.

Web dynpro ABAP and Floorplan manager - tool that can be used for adoption or creating new apps.

SAP dynpro - which includes SAP screen personas to optimize SAP GUI screens. Also notice the connections to the backend - dynpro screens have a close relationship to the backend, they are very interwoven. SAP Screen personas for example, is only a layer on the dynpro where you can reduce fields, merge tops, combine fields but you cannot add business functionality - this only works on the UI level.

Unit 1: SAP UX Strategy

With SAP UI5 with gateway, there is a clear separation between UI and business logic. This setup enables you to be much more flexible in the future. Gateway does not need to run on the same machine as the backend.

Theme designer running up the right side of the screen that allows for the branding of UIs.

Basic fundamental architectural change is that SAP decoupled UI and business logic which makes it easier for developers to react to change in UI technology. SAP's UI technologies will continue to evolve and improve into new UI technologies as they make it to the marketplace in the future.

Goal of the SAP SAPUI5 Framework

The goals of the SAP SAPUI5 Framework are:



- the UI development toolkit for HTML5 (a.k.a. SAP SAPUI5) is a user interface technology that is used to build and adapt client applications
- the SAPUI5 runtime is a client-side HTML5 rendering library with a rich set of standard and extension controls
- it provides a lightweight programming model for desktop and mobile applications
- based on JavaScript, it supports RIA like client-side features:
 - SAPUI5 complies with OpenAjax and can be used together with standard JavaScript libraries
- it supports CSS3, which allows you to adapt themes to your company's branding in an effective manner
- it is based on an extensibility concept regarding custom controls & uses the open source jQuery library as a foundation

SAPUI5 is a UI technology that provides everything you need to build enterprise-ready web apps. It comes with all main SAP platforms but can also be used outside the SAP ecosystem because a large part of SAPUI5 had been open sourced with OpenUI5.

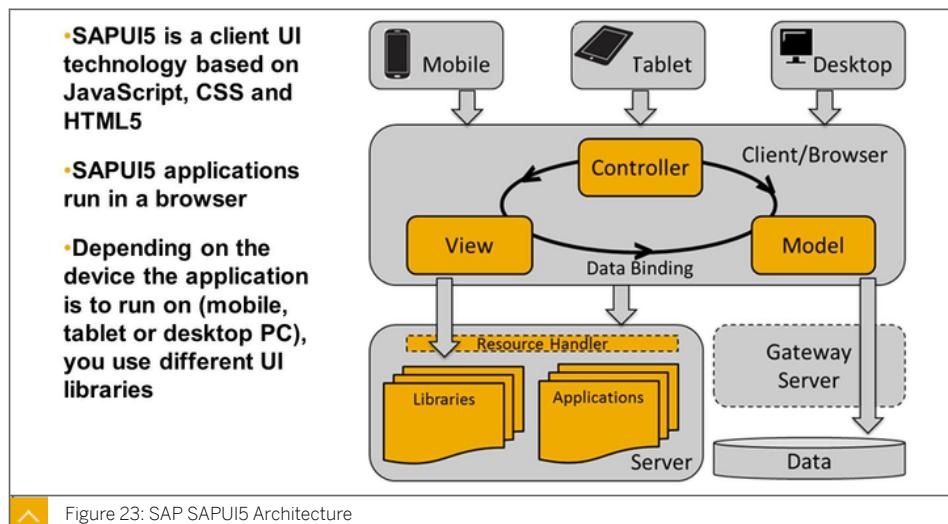


Figure 23: SAP SAPUI5 Architecture

Servers come into play for deploying your applications, storing the SAPUI5 libraries and connecting to a database. Depending on the environment in which SAPUI5 is used, the libraries or your applications are stored on an SAP NetWeaver Application Server or an SAP HANA Cloud platform, for instance. the preferred way to access business data for your application is using the OData model through a SAP NetWeaver Gateway.

When users access an SAPUI5 application from their device, a request is sent to the respective server to load the application into the browser. The view accesses the relevant libraries. Usually the model is also instantiated and business data is fetched from the database.

SAPUI5 Provides Predefined Models

SAPUI5 provides predefined models. These are the benefits:



The JSON model can be used to bind controls to JavaScript object data, which is usually serialized in the JSON format.

- The JSON model is a client-side model and, therefore, intended for small datasets, which are completely available on the client
- The JSON model supports two-way binding

The XML model is a client-side model intended for small datasets, which are completely available on the client.

- The XMLModel does not contain mechanisms for server-based paging or loading of deltas

The Resource model is designed to handle data in resource bundles, mainly to provide texts in different languages.

The OData model enables binding of controls to data from OData services

- The OData model is a server-side model: the dataset is only available on the server and the client only knows the currently visible rows and fields
- This also means that sorting and filtering on the client is not possible
- For this, the client has to send a request to the server
- The OData model currently supports OData version 2.0

SAPUI5 supports different types of models. A model in the Model View Controller concept holds the data and provides methods to retrieve the data from the database and to set and update data.



LESSON SUMMARY

You should now be able to:

- Explain SAP User Experience Tools and Technologies

Unit 1

Lesson 3

Describing SAP User Experience Use Case for Building Fiori-like Apps



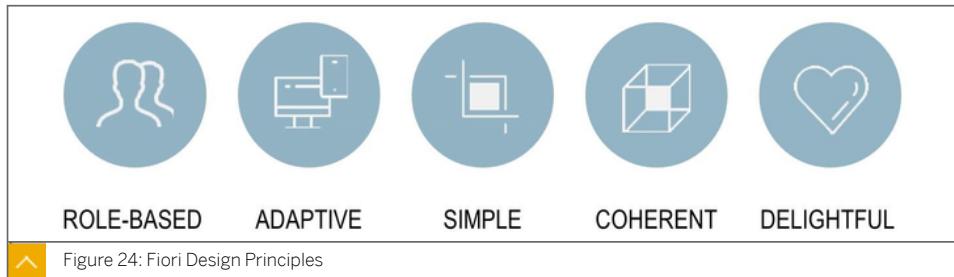
LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Describe SAP User Experience Use Case for Building Fiori-like Apps

SAP User Experience Use Case for Building Fiori-like Apps

Fiori Design Principles



The SAP Fiori Design Principles are

Role based:

Designed for you, your needs, and how you work.

Adaptive:

Adapts to multiple use cases and devices.

Simple:

Only what is necessary.

Coherent:

Provides one fluid user experience.

Delightful:

Makes an emotional connection.

Use Case - Design Process for Fiori-Like Apps using SAPUI5

The Design Process for Fiori-Like Apps using SAPUI5 is:



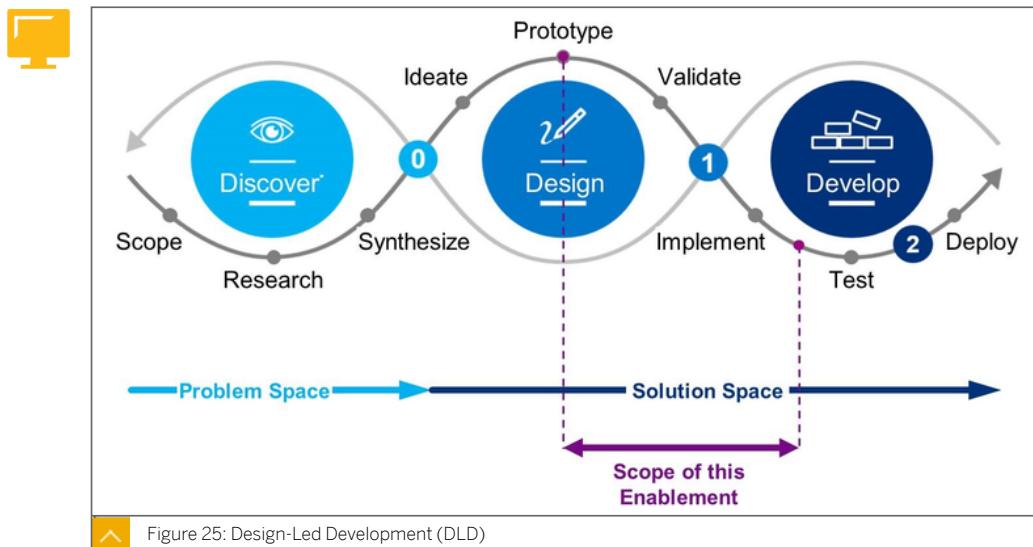
1. Observe how users are currently using the application to complete a business process.
2. Facilitate a Design Thinking workshop with users and designers.

3. Create wireframes to represent the agreed upon use case.
4. Validate the wireframes with the users to ensure that their use case was understood correctly.
5. Iterate through the wireframe scenarios & user validation until agreement is met.
6. Make proposal of finalized wireframes to all stakeholders.
7. Begin the development & test cycles of the SAPUI5 application.



Note:
When implementing SAP Fiori apps you have to go new ways.

Design-Led Development (DLD)



How UX is achieved: through iterative approach in the early stages of Software development.

Unit 1: SAP UX Strategy

Use case - Financial Dashboard Outcome

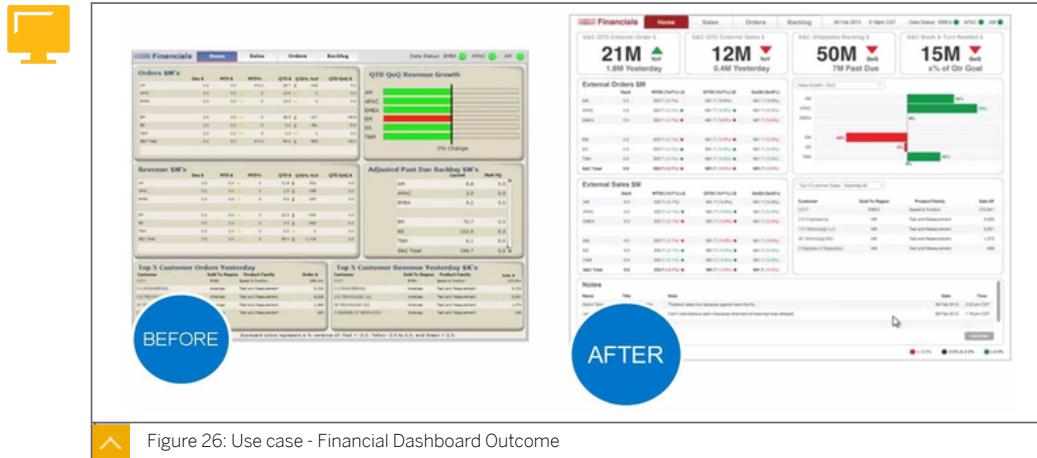


Figure 26: Use case - Financial Dashboard Outcome

Business Problem: standard implementation of a financial dashboard was cumbersome and difficult to use by users: see the **Before** bubble.

Solution: Via the user experience driven design process, a user-centric financial dashboard cockpit was developed that focused on the core 4 KPIs across all users. see the **After** bubble.

Online SAP SAPUI5 Application Demonstrations

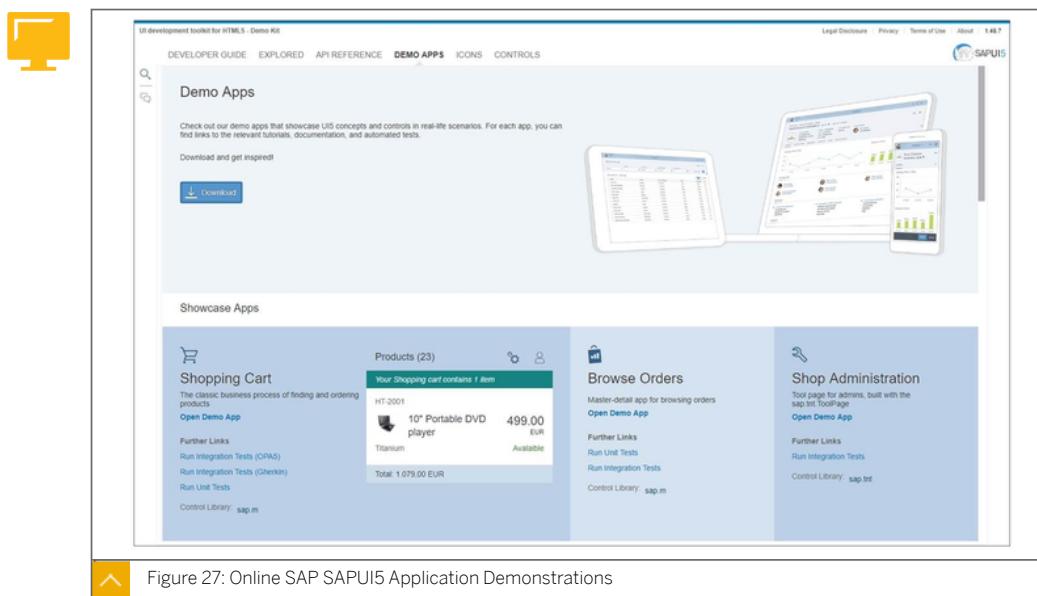


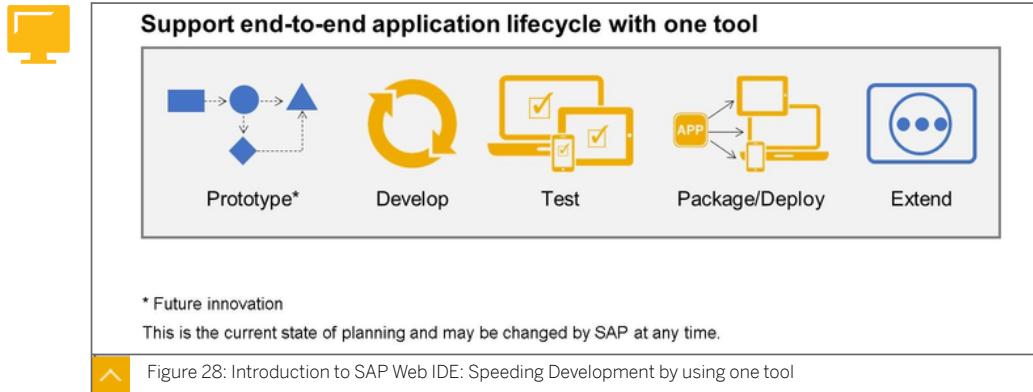
Figure 27: Online SAP SAPUI5 Application Demonstrations

The figure shows the Demo entry page.

Take a moment to view the sample application that SAP has made available online at: <https://sapui5.netweaver.ondemand.com/sdk/#demoapps.html>

SAP WEB IDE, Introduction

Introduction to SAP Web IDE: Speeding Development by using one tool

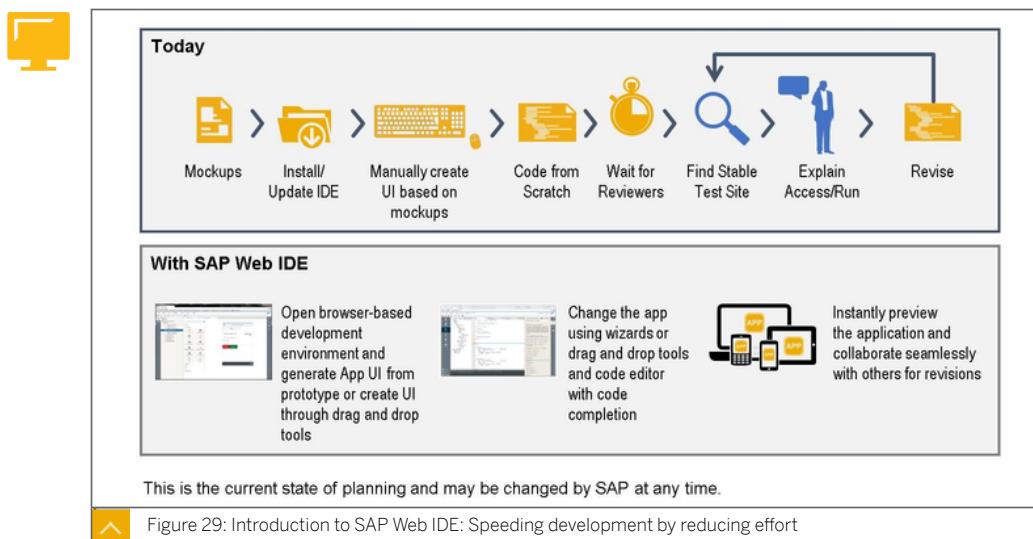


To implement SAP Fiori applications it is recommended to use the SAP Web IDE. SAP Web IDE is a browser-based tool that empowers developers, business experts and designers to build new user interfaces that work with SAP applications.

SAP Web IDE intends to simplify the end-to-end application lifecycle: prototyping, development, packaging, deployment, and customer extensions for SAPUI5 applications.

- Prototyping
- Developing
- Packaging
- Deploying
- Extending (customer extensions for SAPUI5 applications)

Introduction to SAP Web IDE: Speeding development by reducing effort



Unit 1: SAP UX Strategy

Delivering a web-based development tool designed to support the End-to-End application lifecycle for SAPUI5:

Prototyping

Developing

Packaging

Deploying

Extending

SAP Web IDE is a browser-based tool that empowers developers, business experts and designers to build new user interfaces that work with SAP applications. SAP Web IDE intends to simplify the end-to-end application lifecycle: prototyping, development, packaging, deployment, and customer extensions for SAPUI5 applications.

Introduction to SAP Web IDE: Product Benefits



- Develop once, deploy everywhere, & run on any device – mobile, tablet, desktop
- Reduces cost, complexity and effort through cloud-based offering (zero installation)
- Increases developer productivity by simplifying development to create apps faster
- Profit from code templates and SAP best practices to rapidly build applications
- Improves team productivity with the ability to code and collaborate anywhere

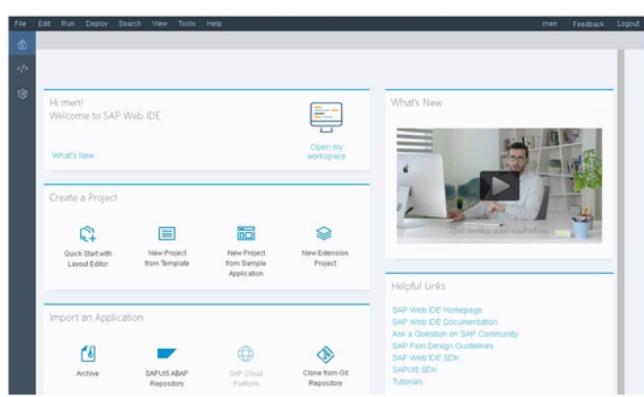
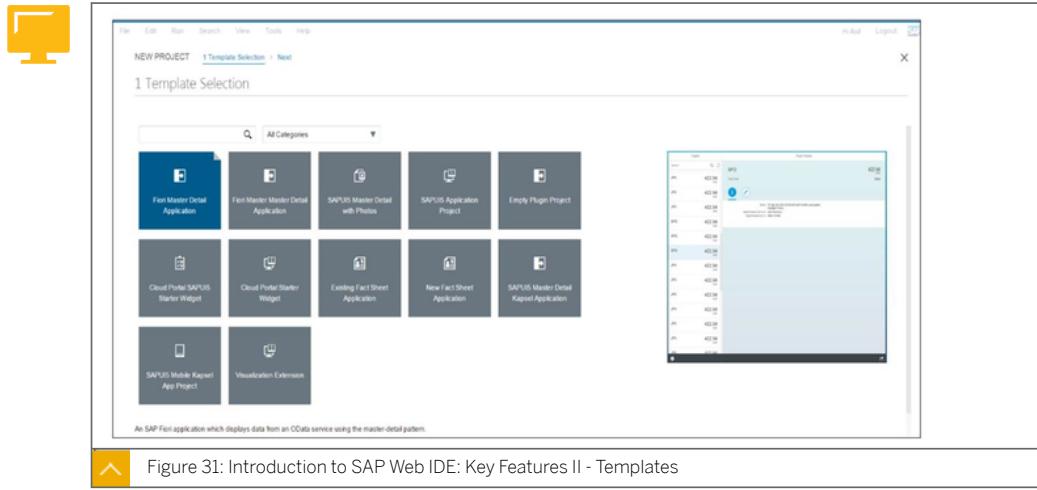


Figure 30: Introduction to SAP Web IDE: Product Benefits

High efficient development environment:

- Source code editor with SAPUI5-specific code completion (configurable for any UI5 version)
- to allow new project creation based on SAPUI5 or Fiori templates
- Extending existing Fiori/UI5 application easily
- WYSIWYG tooling which is a graphical environment to build the UI and to modify it later
- Simple Project persistency (Git)
- Mock data support for
 - testing Decoupling frontend development from backend
 - purposes
- Instant preview in browser
- Comprehensive search capabilities

Introduction to SAP Web IDE: Key Features II - Templates



Templates for creating new project:

- SAP Fiori freestyle application templates
- SAP Fiori smart application templates

Templates for Fiori Libraries:

- Create new library
- Extend an existing library

Fiori Reference Applications integration:

- Enable developers to easily ramp up Fiori development by audited ready-to-run end to end samples
- Enable designers to demonstrate design and interaction patterns in real life
- Full coding with comprehensive comments

Introduction to SAP WEB IDE: SAP Web IDE Plugins

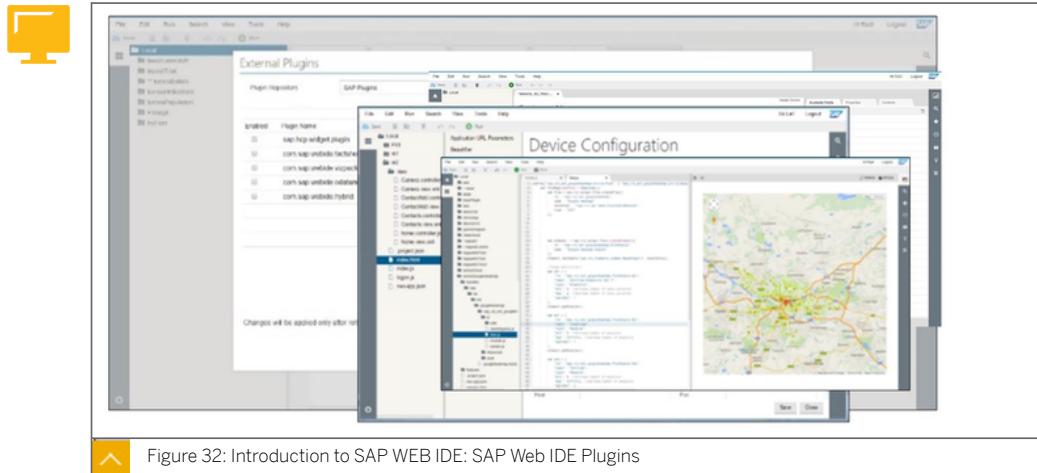


Figure 32: Introduction to SAP WEB IDE: SAP Web IDE Plugins

The figure shows various examples of SAP WEB IDE Plugins.

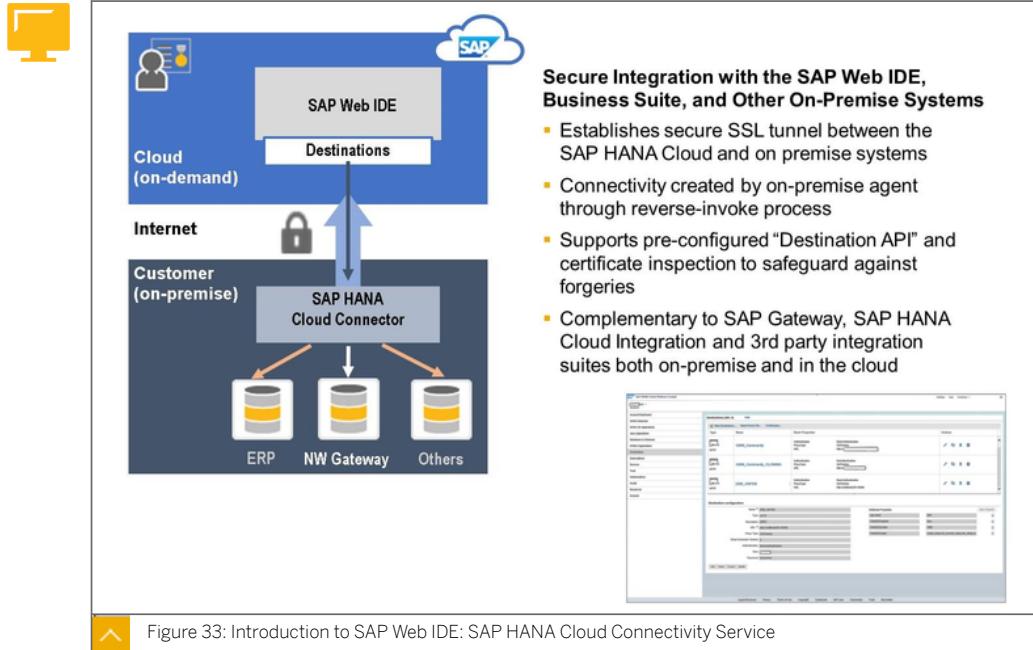
SAP offers out-of-the-box plug-ins and templates that leverage SAP Web IDE's modular and extensible framework:

- Hybrid Application Toolkit
- Create and deploy Apache Cordova hybrid apps
- Templates, code completion, testing, build and deploy
- OData Model Editor
- Define OData models
- Create OData models from scratch or import from file
- Edit OData models quickly using code assist, auto complete and beautify
- Validation of the code on the fly

UI Annotations modeler:

- Create new UI annotations
- Edit existing annotations

Introduction to SAP Web IDE: SAP HANA Cloud Connectivity Service



If you want to connect to an ABAP system, you need to specify this in the destination details.

Let's suppose you want to connect to the GM6 ABAP system (<https://wdflbmt2291:55080>) in order to consume some OData for creating a new Fiori app or for extending some existing ones.

Furthermore you want also to execute some other SAPUI5 applications. We need to pass this information to the SAP Web IDE and the way to do it is through the destination's additional property WebIDEUsage; we need to specify in this property the three usages we want to have with this destination:

- odata_abap (for consuming OData),
- dev_abap (for extending existing Fiori apps) and
- ui5_execute_abap (for executing SAPUI5 apps).

Please refer to the official SAP Web IDE documentation to get more information on this.



LESSON SUMMARY

You should now be able to:

- Describe SAP User Experience Use Case for Building Fiori-like Apps

Unit 1

Learning Assessment

1. What impact does UX have on monetary values?

Choose the correct answers.

- A Increase user satisfaction
- B Gain productivity and data quality
- C Strengthen relationship between customers
- D Save training costs
- E Decrease change requests and user errors

2. What are the answers of SAP in UX strategy?

Choose the correct answers.

- A Design Strategy
- B New / Renew / Enablement
- C New / Renew / Empower
- D Architecture and Technology
- E SAP Screen Personas

3. What impact does SAP Fiori have on Business?

Choose the correct answers.

- A Digitalization
- B Simplification
- C Web & open standards
- D User-centered
- E Re-imagine processes

4. What are the current SAP UI Tools?

Choose the correct answers.

- A** SAPUI5 Application Development Tools
- B** SAP Screen Personas
- C** SAP NetWeaver Portal
- D** Flexible UI Designer
- E** SAP NetWeaver Gateway

5. What are the current UI Technologies of SAP?

Choose the correct answers.

- A** Business Server Pages
- B** SAPUI5
- C** Java Server Pages
- D** Web Dynpro ABAP / Floorplan Manager
- E** Dynpro

6. What goals does the SAPUI5 framework have?

Choose the correct answers.

- A** User Interface technology for building and adapting client applications.
- B** User Interface technology for building and adapting server based applications.
- C** Providing a lightweight programming model for desktop only applications.
- D** Providing an extensible framework for building desktop and mobile applications.

7. What are the SAP Fiori principles?

Choose the correct answers.

- A** Rolebased
- B** Adaptive
- C** Creative
- D** Coherent
- E** Complex

Unit 1: Learning Assessment

8. What steps are part of the discover phase in the DLD?

Choose the correct answers.

- A** Scope
- B** Test
- C** Implement
- D** Research
- E** Synthesize

9. What steps are part of the design phase in the DLD?

Choose the correct answers.

- A** Test
- B** Validate
- C** Prototype
- D** Scope
- E** Ideate

Unit 1

Learning Assessment - Answers

1. What impact does UX have on monetary values?

Choose the correct answers.

- A Increase user satisfaction
- B Gain productivity and data quality
- C Strengthen relationship between customers
- D Save training costs
- E Decrease change requests and user errors

2. What are the answers of SAP in UX strategy?

Choose the correct answers.

- A Design Strategy
- B New / Renew / Enablement
- C New / Renew /Empower
- D Architecture and Technology
- E SAP Screen Personas

3. What impact does SAP Fiori have on Business?

Choose the correct answers.

- A Digitalization
- B Simplification
- C Web & open standards
- D User-centered
- E Re-imagine processes

Unit 1: Learning Assessment - Answers

4. What are the current SAP UI Tools?

Choose the correct answers.

- A SAPUI5 Application Development Tools
- B SAP Screen Personas
- C SAP NetWeaver Portal
- D Flexible UI Designer
- E SAP NetWeaver Gateway

5. What are the current UI Technologies of SAP?

Choose the correct answers.

- A Business Server Pages
- B SAPUI5
- C Java Server Pages
- D Web Dynpro ABAP / Floorplan Manager
- E Dynpro

6. What goals does the SAPUI5 framework have?

Choose the correct answers.

- A User Interface technology for building and adapting client applications.
- B User Interface technology for building and adapting server based applications.
- C Providing a lightweight programming model for desktop only applications.
- D Providing an extensible framework for building desktop and mobile applications.

7. What are the SAP Fiori principles?

Choose the correct answers.

- A** Rolebased
- B** Adaptive
- C** Creative
- D** Coherent
- E** Complex

8. What steps are part of the discover phase in the DLD?

Choose the correct answers.

- A** Scope
- B** Test
- C** Implement
- D** Research
- E** Synthesize

9. What steps are part of the design phase in the DLD?

Choose the correct answers.

- A** Test
- B** Validate
- C** Prototype
- D** Scope
- E** Ideate



UNIT 2

SAP Fiori Elements, Overview

Lesson 1

Explaining the Architecture of Fiori Elements	36
---	----

Lesson 2

Explaining Templates for Fiori Elements	44
---	----

Lesson 3

Exploring the Development Environment	49
---------------------------------------	----

Lesson 4

Exploring the Basic Process of Building Fiori Elements Application	51
--	----

Lesson 5

Using the Core Data Services (CDS) View	57
---	----

Lesson 6

Using the Service Adaption Definition Language (SADL)	63
---	----

Lesson 7

Explaining Metadata Extension	66
-------------------------------	----

Lesson 8

Learning Scenarios of Fiori Elements Implementation	68
---	----

UNIT OBJECTIVES

- Explain the concept of Fiori Elements
- Explain the concept of annotations of OData services
- Explain the concept of Smart Control
- Explain Templates for Fiori Elements

- Explore the Development Environment
- Explore the basic process of building Fiori Elements applications
- Use the CDS View and SADL
- Use SADL
- Explain Metadata Extension
- Learn about scenarios of Fiori Elements implementation

Unit 2

Lesson 1

Explaining the Architecture of Fiori Elements



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the concept of Fiori Elements
- Explain the concept of annotations of OData services
- Explain the concept of Smart Control

Fiori Elements, Concept

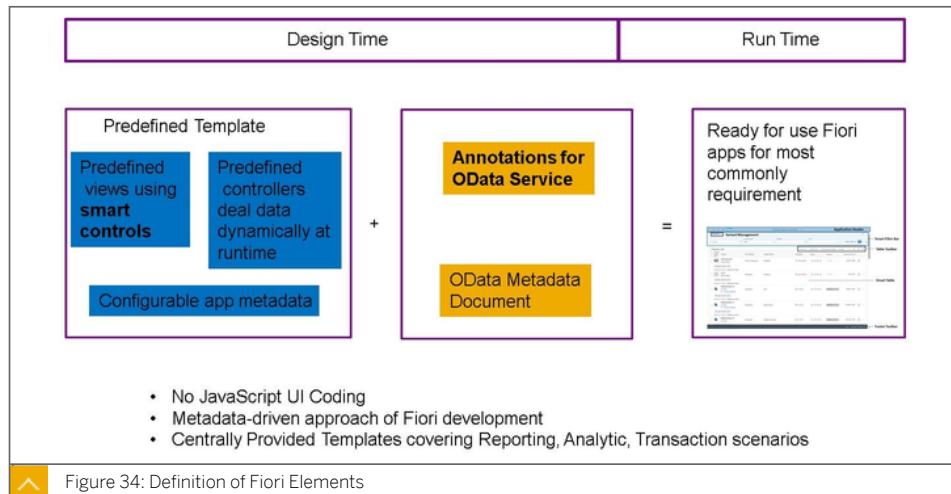


Figure 34: Definition of Fiori Elements

SAP Fiori elements provide designs for UI patterns and predefined templates for commonly used application patterns. App developers can use SAP Fiori elements to create SAP Fiori applications based on OData services and annotations requiring no JavaScript UI coding. The resulting app uses predefined views and controllers that are provided centrally, so no application-specific view instances are required. The SAPUI5 runtime interprets metadata and annotations of the underlying OData service and uses the corresponding views for the SAP Fiori app at startup.

The predefined views and controllers ensure UI design consistency across similar apps. Also the metadata-driven development model significantly reduces the amount of front-end code for each app, so the developer can focus on the business logic.

Fiori Elements — Definition

No JavaScript UI Coding.

Not like traditional SAPUI5 development, detailed JavaScript knowledge is not mandatory for developing Fiori Elements applications. Since JavaScript is a flexible and totally dynamic language and no compile time code check, code written by JavaScript has more chance of runtime error. Fiori Elements will save you lots of effort of studying, writing and debugging JavaScript code. This approach of developing Fiori apps can also your overall quality, stability, and maintainability of your Fiori apps, especially when you need lots of fiori apps.

Metadata-driven approach of Fiori development.

As you have learned from previous course, Fiori application rely on OData service, which provide backend logic. A metadata document of each OData service describes information of the service including entities, properties for each entity, collections, association and all stuffs for the service consumer. The approach of Fiori Elements development is to add more vivid descriptions to the OData metadata document. Those descriptions let the consumer not only know information of data, but also know how to represent the data. We call the descriptions added to OData metadata document as OData Annotation.

Centrally Provided Templates covering Reporting, Analytic, Transaction scenarios.

SAP is responsible for providing templates for Fiori Elements. Each template targets one common requirement of enterprise application. Currently, we have List Report, Overview Page and Analytic List Page, targeting to daily reporting, analytic and transaction scenarios. SAP will continually add more templates to make Fiori Elements covers most of common requirements.

Concept of Annotations of OData Services

In Place Annotations
in OData V2

```
<EntityType Name="Z_C_0211_DemoType" sap:label="Sales Order Report" sap:content-version="1">
  <Key>
    <propertyRef Name="SalesOrderUUID"/>
  </Key>
  <EntityType Name="SalesOrderLineItem" sap:label="Sales Order Line Item" sap:display-format="UpperCase">
    <Property Name="LineIndex" Type="Edm.String" MaxLength="5" sap:label="Line Index" sap:display-format="UpperCase" sap:label="LineIndex"/>
    <Property Name="SalesOrderID" Type="Edm.String" Nullable="true" sap:label="Sales Order ID" sap:display-format="UpperCase" sap:label="SalesOrderID"/>
    <Property Name="OverdueStatus" Type="Edm.String" MaxLength="1" sap:label="Overdue Status" sap:display-format="UpperCase" sap:label="OverdueStatus"/>
    <Property Name="CustomerName" Type="Edm.String" MaxLength="80" sap:label="Customer Name" sap:display-format="UpperCase" sap:label="CustomerName"/>
    <Property Name="NetAmount" Type="Edm.Decimal" Precision="16" Scale="3" sap:label="Net Amount" sap:display-format="Currency" sap:label="NetAmount"/>
    <Property Name="TaxAmount" Type="Edm.Decimal" Precision="16" Scale="3" sap:label="Tax Amount" sap:display-format="Currency" sap:label="TaxAmount"/>
    <Property Name="GrossAmount" Type="Edm.Decimal" Precision="17" Scale="3" sap:label="Gross Amount" sap:display-format="Currency" sap:label="GrossAmount"/>
    <Property Name="Currency" Type="Edm.String" MaxLength="5" sap:label="Currency Code" sap:value-list="standard" sap:display-format="UpperCase" sap:label="Currency"/>
    <Property Name="BillingStatus" Type="Edm.String" MaxLength="1" sap:label="Billing Status" sap:display-format="UpperCase" sap:label="BillingStatus"/>
    <Property Name="DeliverStatus" Type="Edm.String" MaxLength="1" sap:label="Delivery Status" sap:display-format="UpperCase" sap:label="DeliverStatus"/>
  </EntityType>
```

Ex-Place Annotation
in OData V4

```
<Annotations Target="sap.com:cds:z_c_0211_demo1.Z_C_0211_DemoType">
  <Annotation Term="UI.HeaderInfo">
    <Record>
      <PropertyValue Property="TypeKeyName" String=""/>
      <PropertyValue Property="TypeKeyNamePlural" String="Sales Orders"/>
    </Record>
  </Annotation>
  <Annotation Term="UI.LineItem">
    <Collection>
      <Record Type="UI.Datafield">
        <PropertyValue Property="Value" Path="SalesOrderID"/>
      </Record>
      <Record Type="UI.Datafield">
        <PropertyValue Property="Value" Path="CustomerName"/>
      </Record>
      <Record Type="UI.Datafield">
        <PropertyValue Property="Value" Path="NetAmount"/>
      </Record>
      <Record Type="UI.Datafield">
        <PropertyValue Property="Value" Path="GrossAmount"/>
      </Record>
      <Record Type="UI.Datafield">
        <PropertyValue Property="Label" String="Tax Amount"/>
        <PropertyValue Property="Value" Path="TaxAmount"/>
      </Record>
      <Record Type="UI.Datafield">
        <PropertyValue Property="Value" Path="Currency"/>
      </Record>
    </Collection>
  </Annotation>
</Annotations>
```

Figure 35: Annotations for OData Service

Annotations are descriptive information for OData Service.

In OData version 2.0, annotation is stored in XML tag of OData metadata, to describe entity or fields, like datatype, nullable and so on. SAP has expanded the standard OData annotation to satisfy more complex usage. All sap specified annotation are started with 'sap:', for example, 'sap:createable', 'sap:updateable'. Those annotations are not really affect your SAPUI5 application. But if you create your project from wizard like '\CRUD Application', the code generated is affected by those annotations. For example: The editable property of fields in the form generated for updating a record is determined by 'sap:updatable' annotation of corresponding field.

Unit 2: SAP Fiori Elements, Overview

In OData version 4.0, annotations are separated from the main part of OData metadata document. It can be a stand alone part of OData metadata document apart from main part, or store in a separate file. That help us to build more complex annotations, or write more than one annotations for one OData services to satisfy different requirements.



Reference to vocabularies in annotation file	Example of vocabulary
<pre><edmx:Edm xmlns:edmx="http://docs.oasis-open.org/odata/ns/edm" Version="4.0" <edmx:Reference Uri="/JwNfD/CATALOGSERVICE/v2/Vocabularies(TechnicalName='%2F%1BEP%2FVOC_
%2FCommon', Alias="Common")"/> <edmx:Reference Uri="/JwNfD/CATALOGSERVICE/v2/Vocabularies(TechnicalName='%2F%1BEP%2FVOC_
%2FUI', Alias="UI")"/> <edmx:Include Namespace="com.sap.vocabularies.Common.v1" Alias="Common"/> <edmx:Reference Uri="/JwNfD/CATALOGSERVICE/v2/Vocabularies(TechnicalName='%2F%1BEP%2FVOC_
%2FCommunication', Alias="Communication")"/> <edmx:Include Namespace="com.sap.vocabularies.UI.v1" Alias="Communication"/> <edmx:Reference Uri="/JwNfD/CATALOGSERVICE/v2/Vocabularies(TechnicalName='%2F%1BEP%2FVOC_
%2FCommunication.v1', Alias="Communication.v1")"/> <edmx:Include Namespace="com.sap.vocabularies.Communication.v1" Alias="Communication.v1"/> <edmx:Reference Uri="/SAP/Z_C_01L_DEMO_CDS/Metadata"/> <edmx:Include Namespace="Z_C_01L_DEMO_CDS" Alias="SAP"/> <edmx:Reference></pre>	<pre><!--Annotations--> <Annotation Target="UI.Facets"> <Annotations> <Annotation Term="UI.Facets"> <Collection> <Record Type="UI.CollectionFacet"> <PropertyValue Property="ID" String="@11Bnagt:0GeneralInformation"/> <PropertyValue Property="Label" String="@11Bnagt:0GeneralInformation"/> <PropertyValue Property="Facets"> <Collection> <Record Type="UI.ReferenceFacet"> <PropertyValue Property="Label" String="@11Bnagt:0GeneralInformation"/> <PropertyValue Property="Target" AnnotationPa...> </Record> </Collection> </PropertyValue> </Record> </Collection> </Annotations> </Annotation> </Annotations></pre>

Figure 36: Annotation Concepts(1): Vocabulary

Vocabulary define how can you add annotations to an OData service. All annotation file must start from references and giving an alias to some vocabularies.

The form of vocabulary is an XML document contains information of Terms and Types.

In Fiori Elements scenario, the most important vocabulary is UI vocabulary, it defines most of annotations you can use in Fiori Elements. Other vocabularies, like Common, Communication, define annotations in specific domain and used in several scenarios.



Annotation targets to an Entity Type	Annotation has no target
<pre><Annotation Target="sap.com:z_c_01l_demo1.SIEM_I_BusinessPartnerType"> <Annotations> <Annotation Term="UI.FieldGroup"> <Record Type="UI.FieldGroupType"> <PropertyValue Property="Label" String="#@11Bnagt:0CUSTOMER_INFORMATION"/> <PropertyValue Property="Data"> <Collection> <Record Type="UI.DataField"> <PropertyValue Property="Value" Path="BusinessPartner"/> </Record> <Record Type="UI.DataField"> <PropertyValue Property="Value" Path="CompanyName"/> </Record> <Record Type="UI.DataField"> <PropertyValue Property="Value" Path="LegalForm"/> </Record> <Record Type="UI.DataField"> <PropertyValue Property="Value" Path="PhoneNumber"/> </Record> </Collection> </PropertyValue> </Record> </Annotations></pre>	<pre><Annotations Target=""> <Annotation Term="UI.Facets"> <Collection> <Record Type="UI.CollectionFacet"> <PropertyValue Property="ID" String="@11Bnagt:0GeneralInformation"/> <PropertyValue Property="Label" String="@11Bnagt:0GeneralInformation"/> <PropertyValue Property="Facets"> <Collection> <Record Type="UI.ReferenceFacet"> <PropertyValue Property="Label" String="@11Bnagt:0GeneralInformation"/> <PropertyValue Property="Target" AnnotationPa...> </Record> </Collection> </PropertyValue> </Record> </Collection> </Annotations></pre>

Figure 37: Annotation Concepts(2): Target

Annotations must grouped by targets. Target associate a group of annotations to something in OData service. Due to most annotations are data relevant, most annotations target to an entity type or property. In some cases, annotations targets to Association, Function Import or other objects in OData service.

Some annotations are nothing about data, or not binding with particular entity type, those annotation do not have a target, it should contained in a group of annotations with a blank "" target.



```

<Annotations Target="sap.com.cds_z_c_u4l1_demo1.Z_C_U4L1_SalesOrderItemType">
  <Annotation Term="UI.FieldGroup" Qualifier="DG1">
    <Record Type="UI.FieldGroupType">
      <PropertyValue Property="Data">
        .....
      </PropertyValue>
    </Record>
  </Annotation>
  <Annotation Term="UI.FieldGroup" Qualifier="DG2">
    <Record Type="UI.FieldGroupType">
      .....
    </Record>
  </Annotation>
</Annotations>



- Target + Term + Qualifier is the unique identifier for an annotation
- Qualifier can be omitted if Target + Term is unique

```

Figure 38: Annotation Concepts(3):Term and Qualifier

All annotation must have a term. A term determines the meaning of an annotation. For example: term UI.FieldGroup defines a group of fields.

If you have two annotations with same term for same target a qualifier must be added to make your annotation unique.



```

Singular Term

<Term Name="HeaderInfo" Type="UI.HeaderInfoType" AppliesTo="EntityType">
  <Annotation Term="UI.ThingPerspective"/>
  <Annotation Term="Core.Description" String="Information for the header area"/>
</Term>

Collection Term

<Term Name="LineItem" Type="Collection(UI.DataFieldAbstract)" Nullable="false" AppliesTo="EntityType">
  <Annotation Term="Core.Description" String="Collection of data fields for representation in a table or list"/>
  <Annotation Term="UI.ThingPerspective"/>
</Term>

```

Figure 39: Annotation Concepts(4):Term Definition in Vocabulary

Terms are defined in vocabulary. Information in a term definition is:

- Term Name: The name of the term.
- Data Type: If there is only one record in the term , write the name of the type. If the term may have multiply records, the Type attribute needs to be 'Collection(<Data Type>)'
- Applies To: Defines which kinds of elements in OData service can be target of this term.
- Nullable: Default is true. Most annotations are nullable.
- Annotations to this term, used for development tools.

Unit 2: SAP Fiori Elements, Overview



```
<ComplexType Name="HeaderInfoType">
  <Property Name="TypeName" Type="Edm.String" Nullable="false">
    <Annotation Term="Core.IsLanguageDependent"/>
    <Annotation Term="Core.Description" String="Name of the main entity type"/>
  </Property>
  <Property Name="TypeNamePlural" Type="Edm.String" Nullable="false">
    <Annotation Term="Core.IsLanguageDependent"/>
    <Annotation Term="Core.Description" String="Plural form of the name of the main entity type"/>
  </Property>
  <Property Name="Title" Type="UI.DataField" Nullable="false">
    <Annotation Term="Core.Description" String="Title, e.g. for overview pages"/>
  </Property>
  <Property Name="Description" Type="UI.DataField" Nullable="true">
    <Annotation Term="Core.Description" String="Description, e.g. for overview pages"/>
  </Property>
  <Property Name="ImageUrl" Type="Edm.String" Nullable="true">
    <Annotation Term="Core.Description" String="Image URL for an instance of the entity type. If the property is available or not valid the property TypeImageUrl can be used instead."/>
  </Property>
  <Property Name="TypeImageUrl" Type="Edm.String" Nullable="true">
    <Annotation Term="Core.Description" String="Image URL for the entity type"/>
  </Property>
</ComplexType>
```

Figure 40: Annotation Concepts(4): Complex Type

Most term described by a complex type. A complex type can have properties, which are further described by other complex type, or OData built in type (Edm.*) like string,int32,boolean.



```
<ComplexType Name="DataFieldAbstract" Abstract="true">
  <Property Name="Label" Type="Edm.String" Nullable="true">
    <Annotation Term="Core.Description" String="A short, human-readable text suitable for labels and captions in UIs"/>
    <Annotation Term="Core.IsLanguageDependent"/>
  </Property>
  <Property Name="Criticality" Type="UI.CriticalityType" Nullable="true">
    <Annotation Term="Core.Description" String="Criticality of the data field value"/>
  </Property>
  <Property Name="CriticalityRepresentation" Type="UI.CriticalityRepresentationType" Nullable="true">
    <Annotation Term="Core.Description" String="Decides if criticality is visualized in addition by means of an icon"/>
  </Property>
  <Property Name="IconUrl" Type="Edm.String" Nullable="true">
    <Annotation Term="Core.Description" String="Optional icon to decorate the value"/>
    <Annotation Term="Core.IsURL"/>
  </Property>
</ComplexType>

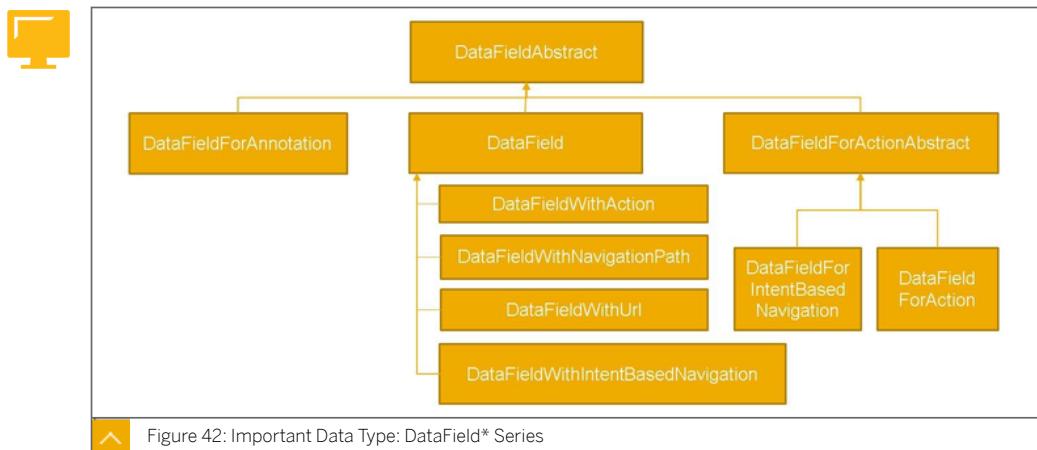
<ComplexType Name="DataFieldWithAction" BaseType="UI.DataField">
  <Property Name="Action" Type="Common.QualifiedName" Nullable="false">
    <Annotation Term="Core.Description" String="Qualified name of an action"/>
  </Property>
</ComplexType>
```

↑
Type "DataFieldWithAction" has all properties of "DataFieldAbstract" and additional property "Action"

Figure 41: Annotation Concept(5): Type Inheritance

For better reusability, inheritance of types are supported.

In this example, DataFieldAbstract is an abstract data type (Abstract = true). And type DataFieldWithAction has a base type of DataFieldAbstract, so the latter have all properties of the base type, with additional properties defined for itself.



A collection of data type starting of DataField are foundation of UI annotations. Most terms and complex types have relationship with those datatypes since data fields are what end user see, click and search. All types are inherit from DataFieldAbstract.

Here are definitions for those types:

DataFieldAbstract

Abstract type for all DataField* types, define common properties for describe a field.

DataField

A normal data field in screen, display value from a field of an entity type of the OData service.

DataFieldForAnnotation

The field displayed on screen is determined by another annotation.

DataFieldWithAction

Rendered as a button or link, the actual action is relevant to value of the field.

DataFieldForAction

Rendered as a button or link, the action is not relevant to value of the field in contrast to DataFieldWithAction.

DataFieldWithNavigationPath

The data source of the field is a field in other entity types which can access through by an association.

DataFieldWithURL

The data field is rendered as hyper link, jumps to a url when clicked by user.

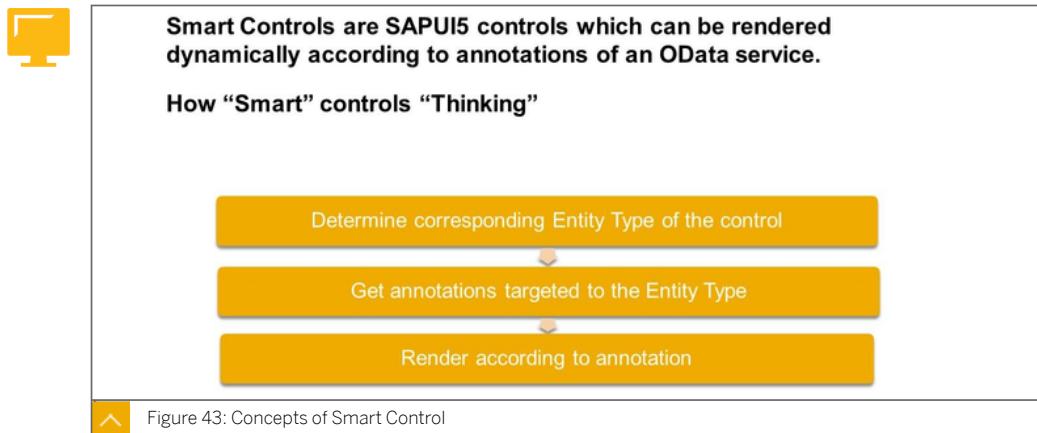
DataFieldWithIntentBasedNavigation

The data field is rendered as hyperlink, jumps to an intent defined in Fiori Launchpad, the target is relevant to value of the field.

DataFieldForIntentBasedNavigation

The data field is rendered as hyperlink, jumps to an intent defined in Fiori Launchpad, the target is not relevant to value of the field in contrast to DataFieldWithIntentBasedNavigation.

Smart Control, Concept

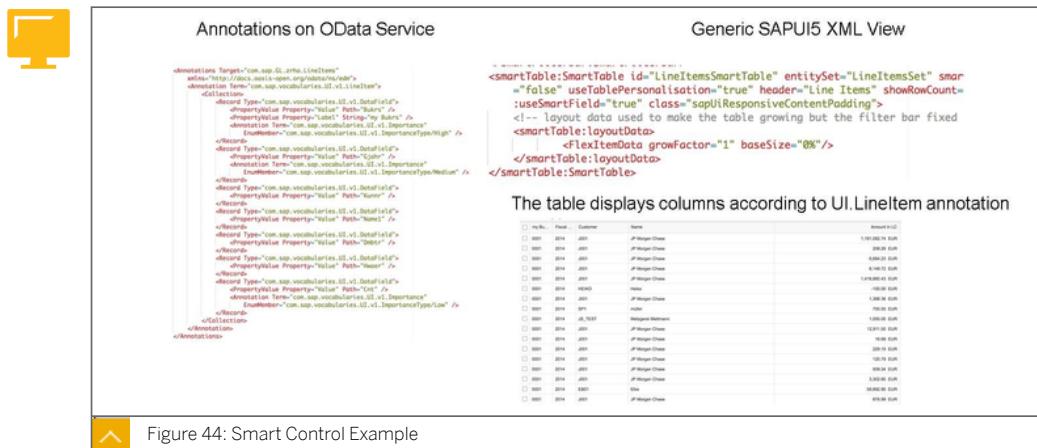


Smart Controls is a collection of controls developed before Fiori Elements. The initial purpose is to replace basic controls in SAPUI5 in most common use case.

Then there was an idea about build whole application by smart controls, that was Smart Template, the predecessor of Fiori Elements. Developers can use those control standalone, or use Smart Template to build smart applications. The development and maintenance of Smart Controls and Fiori Elements are separate.

Currently, Smart Control is mainly for Fiori Elements, and not recommend for standalone usage. All new features of smart controls will be determined by requirements of Fiori Elements.

Smart Control Example



The figure, Smart Control Example, shows an example of a smart control.



LESSON SUMMARY

You should now be able to:

- Explain the concept of Fiori Elements
- Explain the concept of annotations of OData services

Lesson: Explaining the Architecture of Fiori Elements

- Explain the concept of Smart Control

Unit 2

Lesson 2

Explaining Templates for Fiori Elements



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain Templates for Fiori Elements

Templates of Fiori Elements

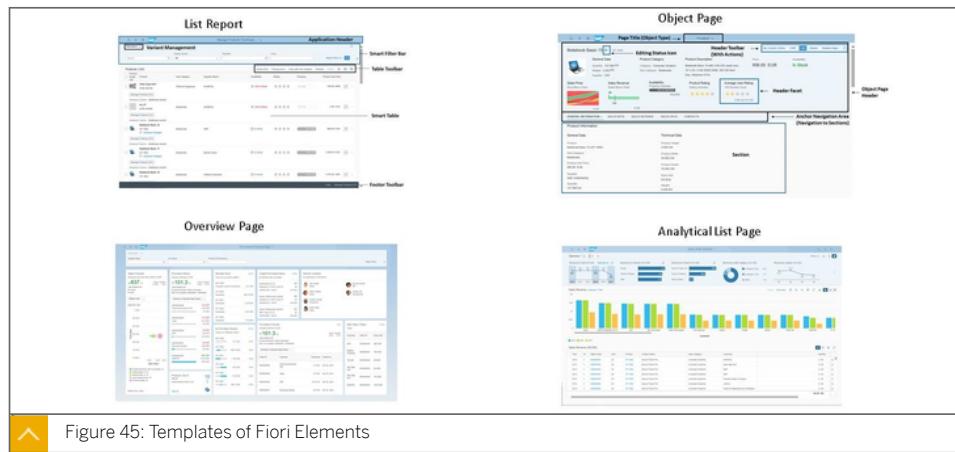
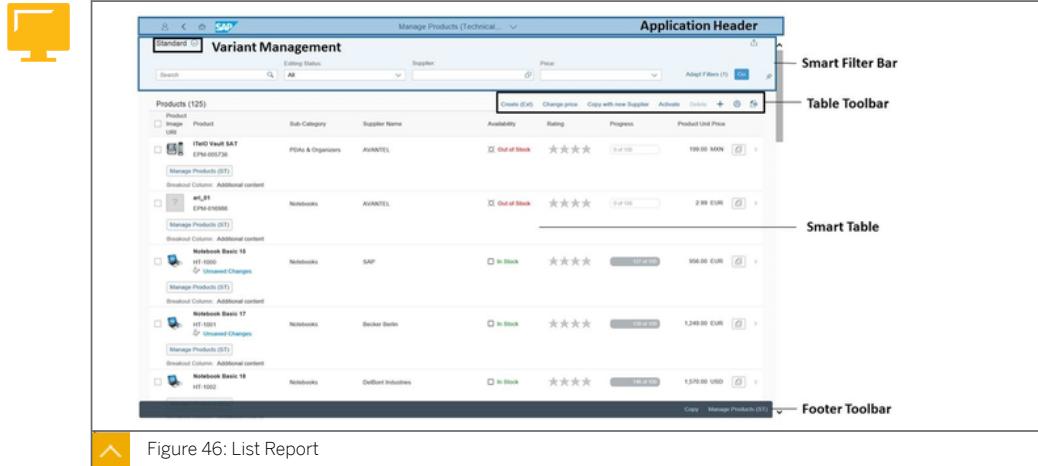


Figure 45: Templates of Fiori Elements

Currently there are four templates for Fiori Elements. They are List Report, Object Page, Analytic List Page, Overview Page.

List Report



The list report allows the user to work with a large list of items. It combines powerful functions for filtering large lists with different ways of displaying the resulting item list.

The list report view includes the following main elements:

Application header

Smart filter bar with variant management and the generic Share menu that includes the following actions: Send Email.

Save as Tile

Share in SAP Jam (if integration with SAP Jam is configured).

By default, the smart filter bar is displayed, as shown above, when users launch an app. When choosing Go, the content of the list report is displayed. If users have set Execute on Select for their default variant, the content of the list report is displayed immediately upon launching the app, and the smart filter bar is collapsed.

Smart table

Footer toolbar

Can include optional actions.

A List Report can display data not only as plain text. It displays data in various forms to improve user experience. It also supports editability and can leverage create, update and delete operations to correspond to the backend code.

From that, we can use the List Report as replacement of ALV in SAP GUI. Lots of ABAP queries created by SQVI and ALV-based ABAP reports can be replaced by List Report.

Object Page

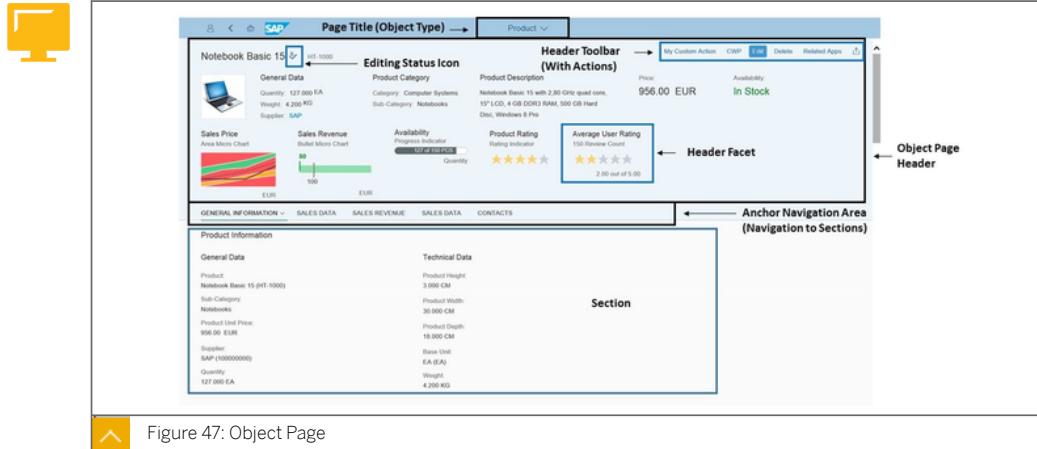


Figure 47: Object Page

The Object page displays information of a single business entity. Major entry of object page is to click a row in list report. So this page is combined in List Report template. When you create a Fiori Elements application using List Report, you have already got an Object Page.

The object page is made up of the following elements:

- Page title which is set to the object type, product, for example, product
- Object header including the following:
 - Title and subtitle
 - Editing status icon
- Header toolbar, containing generic actions (in Display mode)
- Optional elements, which can include the following:
 - A description
 - An image of the object
 - Buttons in the header toolbar for use case-specific actions, for example, Edit and Delete
- Header facets to showcase important information for the object. Header facets can contain, for example:
 - Label-field pairs, to show, for example, price or availability. We recommend using a maximum of five label-field pairs.
 - Smart controls, to show, for example, a micro chart detailing sales revenue, or a rating indicator to visualize the average of all user-assigned ratings
- Anchor navigation area that lets users navigate to the individual content area sections
- Content area, in which data is organized into sections that can contain field groups or a table
- Footer bar (not shown), in which actions and the Show Messages button are available (if applicable). The footer bar of sub-item object pages also includes the Apply button, in

create and edit mode. This action concludes the current create or edit activity, saves the draft, and navigates one step up in the object hierarchy. A toast message is displayed if the operation was successful.

Overview Page

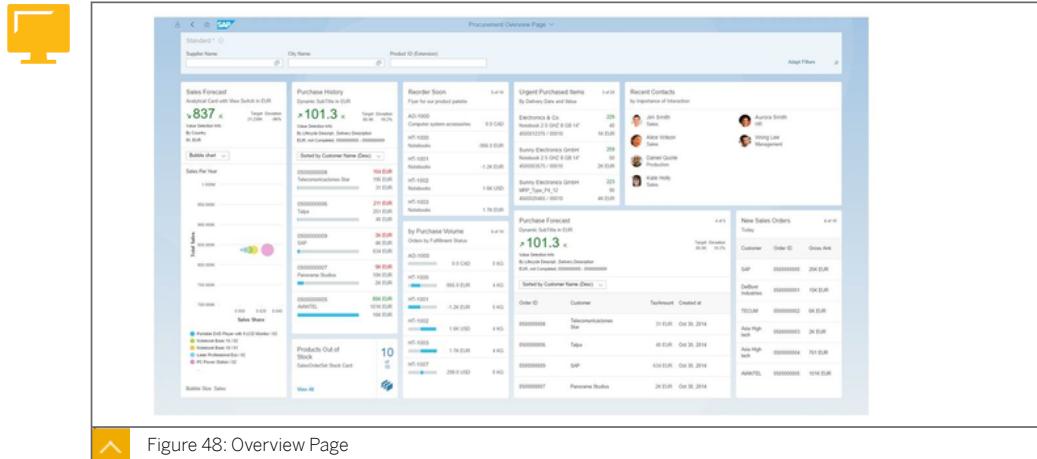


Figure 48: Overview Page

An Overview Page is a data-driven SAP Fiori application built using SAPUI5 technology, OData services, and annotations for organizing large amounts of information.

The Overview pages provides a quick access to vital business information at a glance, in the form of visual, actionable cards. The user-friendly experience makes viewing, filtering, and acting upon data quick and simple. While simultaneously presenting the big picture at a glance, business users can focus on the most important tasks enabling faster decision making as well as immediate action.

The application lets you create several cards for different types of content that helps in visualizing information in an attractive and efficient way. You can create overview pages and add cards to the page using the overview page wizard in SAP Web IDE.

The displayed data is fully interactive, with clickable areas for easy navigation to relevant applications. Based on SAP Fiori, overview pages organize action items with a fully responsive user interface. Users can access overview pages from SAP Fiori launchpad and narrow down the information displayed, or opt to hide cards to focus on a particular topic.

Overview Page — Components

The overview page application contains the following main components:

Application header:

Provides a description of the area for which this application provides an overview (for example, procurement or sales). From the header area, users can change user account settings and manage cards (show/hide).

Smart filter:

Provides application-level filters for changing the levels of data displayed in the cards. For example, you could use the filter to display only transactions larger than \$10,000, only items lighter than 50kg, and so on.

Cards:

A card is a smart component that uses UI annotation to render its content. Each card is bound to a single entity set in a data source. A card may display a donut or bar chart, or a table. Stack cards contain a set of quick view cards which can be viewed in an object stream. Cards are displayed on the overview page in up to five responsive columns and can be rearranged by dragging and dropping.

Analytical List Page

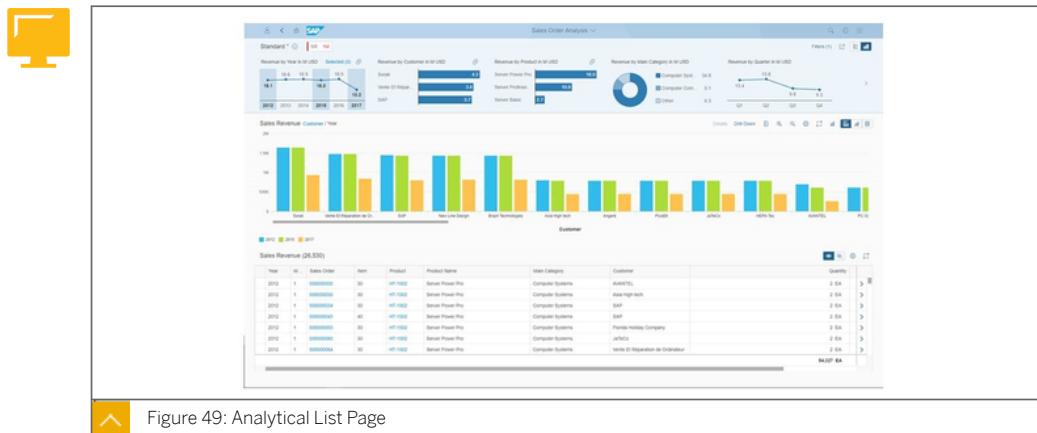


Figure 49: Analytical List Page

The Analytical List Page (ALP) is a SAP Fiori elements application for detailed analytics.

This is a brand new template introduced in S/4 HANA 1709 and ABAP 7.52.

It lets you analyze data from different perspectives, to investigate a root cause, and to act on transactional content. You can identify relevant areas within data sets or significant single instances using data visualization and business intelligence. All this can be done seamlessly within one page.

The combination of transactional and analytical data using chart and table visualization lets you to view relevant data more quickly. This hybrid view of data allows an interesting interplay between the chart and table representations.

Configure ALP to include the following use cases seamlessly within one page:

- Related KPIs (key performance indicators) on the header area as KPI tags.
- These KPI tags further allow a progressive disclosure and navigation through KPI cards.
- Filter data sets used for the main content area through different filter modes.
- For example, visual filters provide an intuitive way of choosing filter values from an associated measure value.
- Seamless navigation to applications from the content area and KPI card area.
- Customizing and sharing ALP as a page variant with other users.



LESSON SUMMARY

You should now be able to:

- Explain Templates for Fiori Elements

Unit 2

Lesson 3

Exploring the Development Environment

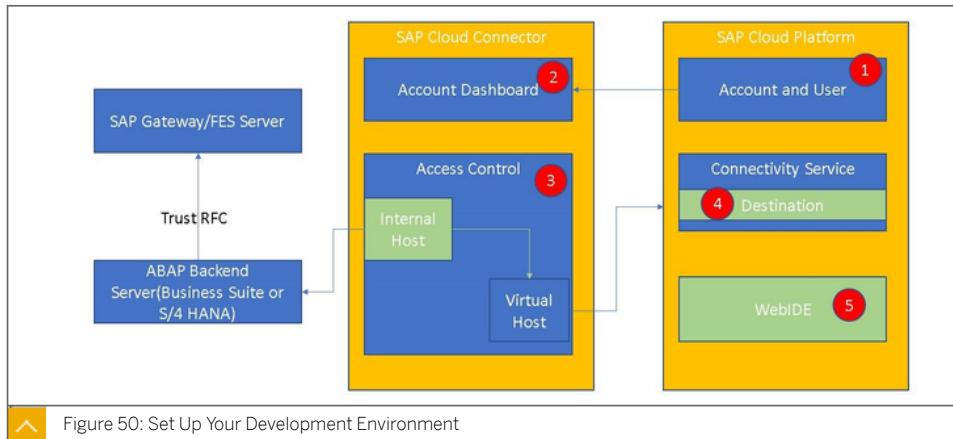


LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explore the Development Environment

The Development Environment



To set up development for Fiori Elements developments, you need:

- An SAP cloud platform account to run development tools -- WebIDE.
- A Server plays the role of OData backend, in this course, we have to ABAP server, one for backend logic, another SAP Gateway server for publishing OData service and Fiori application.
- SAP Cloud Connector to set up a secure connection between your Gateway server and SAP Cloud Platform.

Major steps of configuration are:

1. Register an account in SAP Cloud Platform.
2. Add your SAP CP account into SAP Cloud Connector.
3. Configure access control information in SAP Cloud Connector to connect your internal SAP Gateway server and expose it through a virtual host name/port.
4. Create a destination in connectivity service of SAP CP. Use this destination to connect to your internal SAP Gateway through virtual host name defined in SAP Cloud Connector

5. Test your configuration by running the wizard of WebIDE. Select the destination you created and see if a list of OData services are shown up.



LESSON SUMMARY

You should now be able to:

- Explore the Development Environment

Unit 2

Lesson 4

Exploring the Basic Process of Building Fiori Elements Application

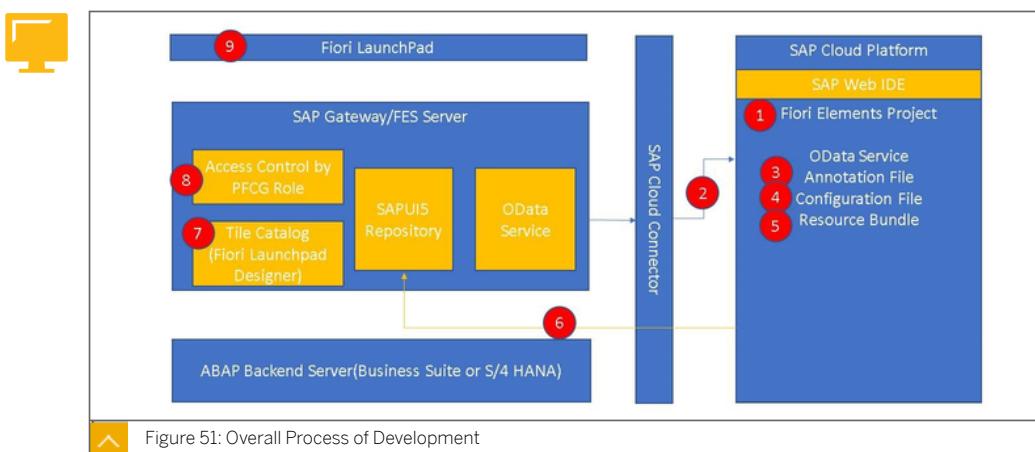


LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explore the basic process of building Fiori Elements applications

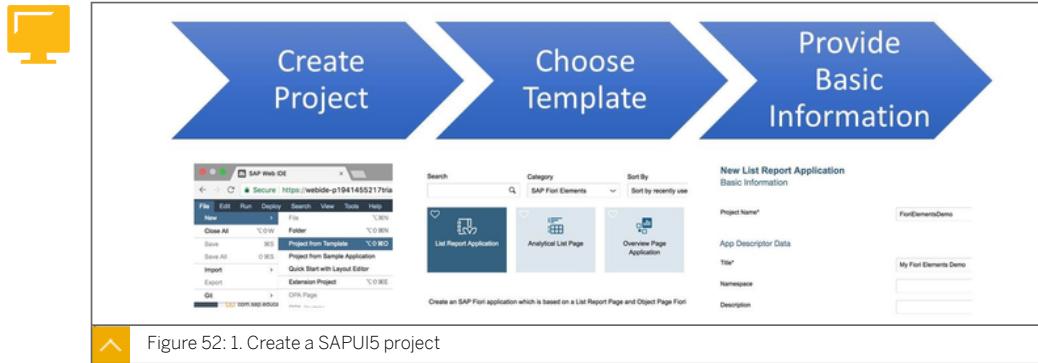
The Process of Building Fiori Elements Applications



- 1.Create a SAPUI5 project from one of Fiori Elements template
- 2.Choose an OData service as data provider of your project
- 3.Create an annotation file associated to the OData service and write annotation in it
- 4.Modify configuration file to customize your application(Optional)
- 5.Translate resource bundles (i18n* files) to make your application support multiple languages(Optional)
- 6.Deploy your Fiori Elements application to a web server (In this course, we use SAPUI5 repository in SAP Gateway server)
- 7.Create tile catalog, target mapping, tiles and other configuration for Fiori Launchpad
- 8.Add catalog to user menu in a PFCG role , then grant the role to user
- 9.Logon to Fiori Launchpad, add the tile to Launchpad.



1. Create a SAPUI5 project



To create Fiori Elements project, you need to create a SAPUI5 project from template. To have a quick view of all fiori elements templates, you can set the category to 'SAP Fiori Elements'.

Then you need to provide basic information for your project.

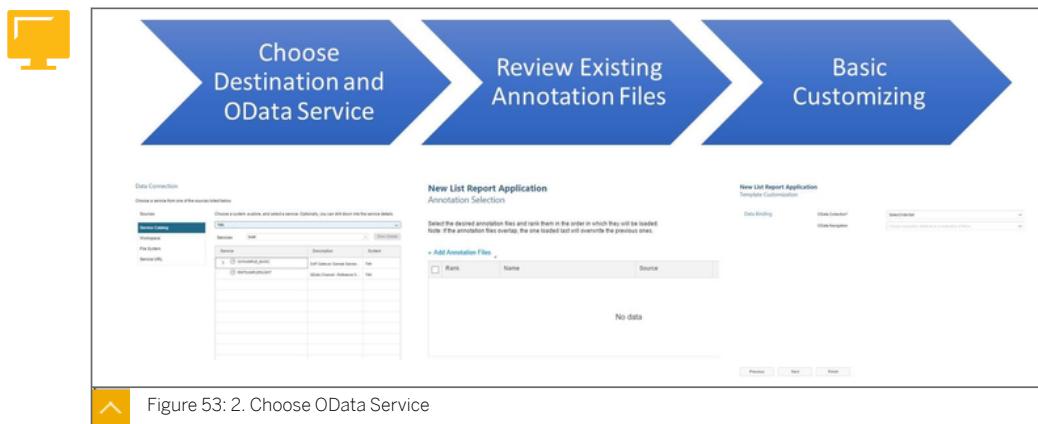
Project Name: The name of project folder in your WebIDE workspace. It will also be the namespace of your project.

Title: Title displayed in your fiori elements application

Namespace(Optional): Namespace of your project. If you fill this field, the final namespace will be followed by content in title field.

Description(Optional): Project description

2. Choose OData Service



To bind your Fiori Elements to an OData Service. You should finish following steps.

1. Choose Destination and OData Service

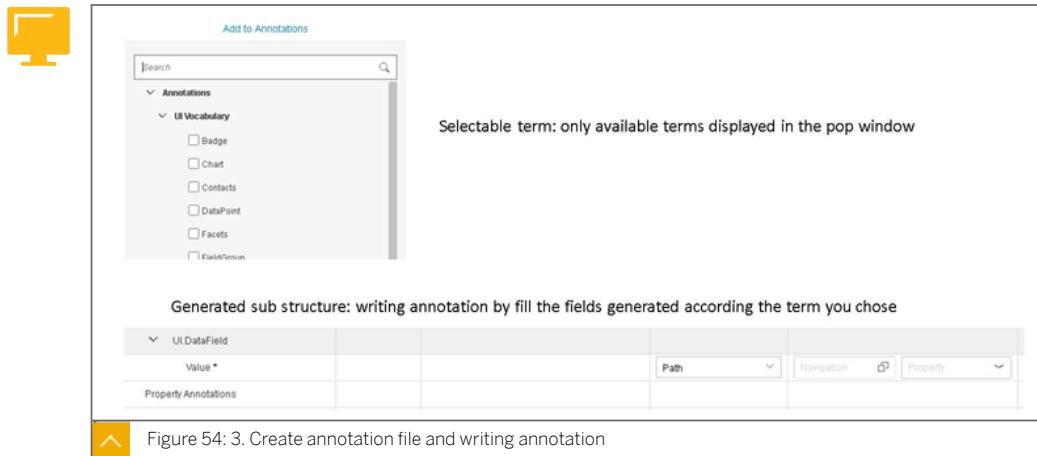
Lesson: Exploring the Basic Process of Building Fiori Elements Application

2. If an annotation file for this OData service is already exist, you can see it Review Existing Annotation Files step. You can also add annotation file here , or in later step.

SAP provided you the most flexibility for creating annotation file, you can create it either in backend (External Annotation) or frontend (Local Annotation).

3. In step Basic Customizing, you need to provide basic information for template. The content depends on the template you choose.

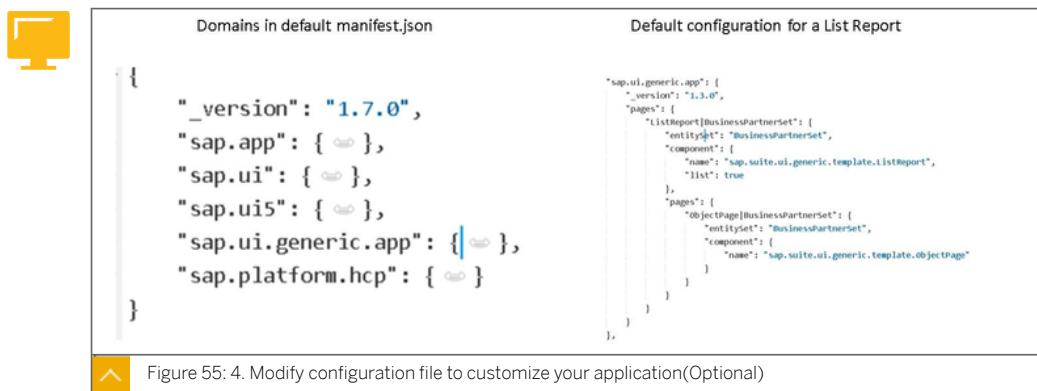
3. Create annotation file and writing annotation



In SAP Web IDE, you can create new annotation file for specified OData service by using context menu.

SAP Web IDE provide an annotation editor to make the writing of annotation easier. You can choose term from a pop window, and its substructure will automatically generated according to data type. Then you can choose options and fill values accordingly.

4. Modify configuration file



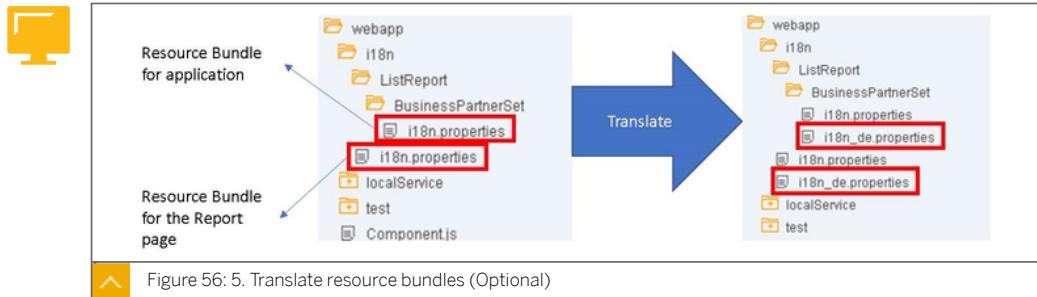
All information you provided in the wizard will result as configuration information in manifest.json. The file is located in folder webapp in your project. You can navigate detailed information by double click it and choose Code Editor.

Unit 2: SAP Fiori Elements, Overview

The configuration are divided into several domains. Domain "sap.app" has the configuration information generally used by all SAPUI5 application, like name, title, OData service and location of its annotations. Based on the template you choose, it may have other specific domains contain other information. For example, domain "sap.ui.generic.app" contains configuration information about ListReport and ObjectPage.

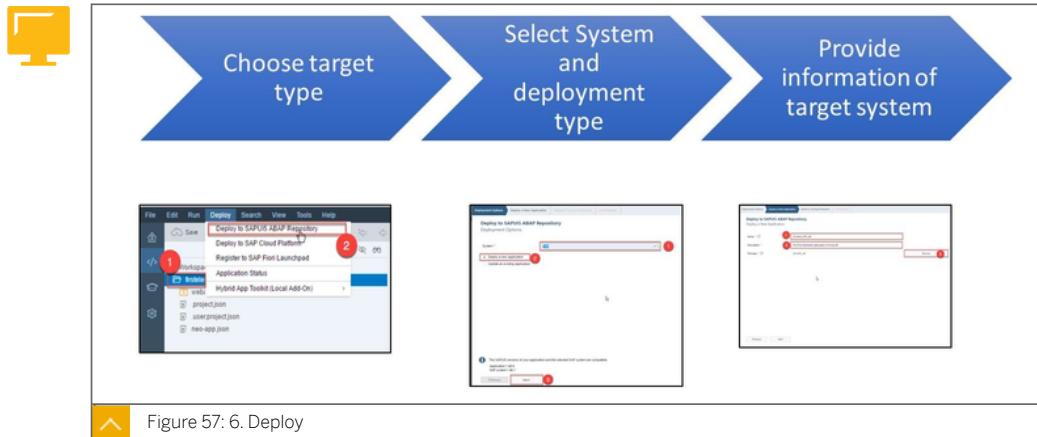
When you need to change some basic information, like OData service name or entity set name, you can modify this file directly. In some cases, advanced features also need some adjustments to this file.

5. Translate resource bundles



For the need of internationalization of static text in your application. It's best to store static text content as key-value pair in a file. In fiori elements, the template will create a default resource bundle file (i18n.properties) for the application and each page in it. You can translate each file into any language by create a i18n_<language iso code>.properties in same position of the resource bundle you want to translate.

6. Deploy

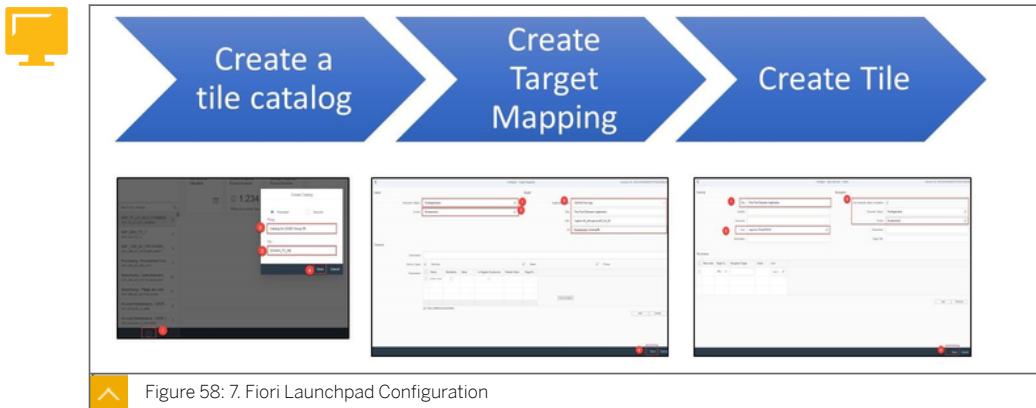


To deploy your Fiori Elements to a server. You should finish following steps:

1. Choose which platform do you need to run the application(In this course, we use ABAP Repository).
2. Choose the system by destination, then decide to create a new application or update existing one.

3. Provide information, like program name, description, package.

7. Fiori Launchpad Configuration



If the Fiori Elements application is deployed on Netweaver ABAP, you need to enable the application by the following steps:

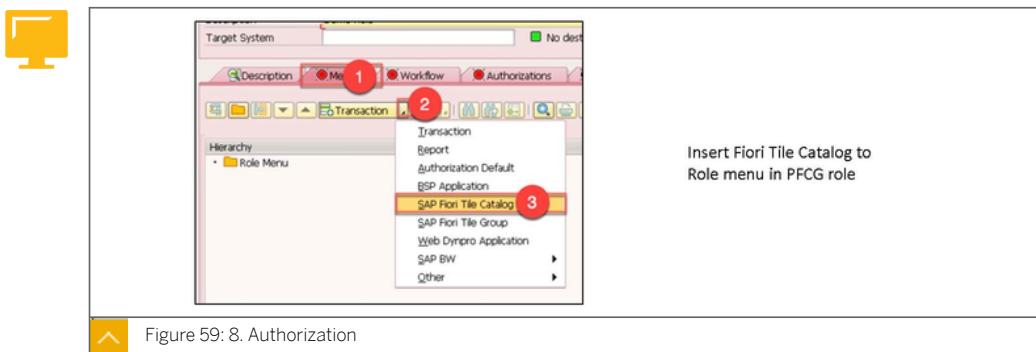
1. Open Fiori Launchpad Designer, create a tile catalog for the application.
2. Create a target mapping. Target mapping should have detailed technical information for the access of the application, and expose the application as an intent.(Semantic Object - Action).
3. Create a tile based on the target mapping.



Note:

Detailed information of Fiori Launchpad configuration is described in UX100.

8. Authorization



To make user have privilege to access the fiori application. Several application specific authorization is needed.

For front-end user:

The user must have a role which have the tile catalog in its role menu.

For back-end user:

The user must have authorization object to access the OData service and business data.

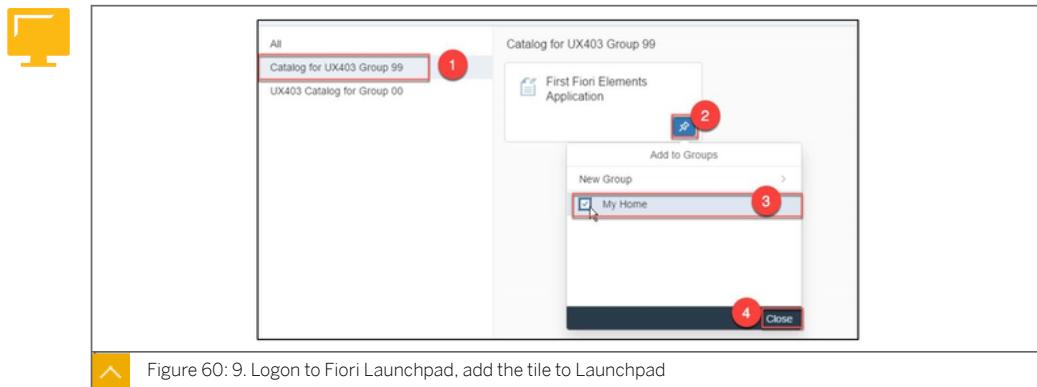
9. Logon to Fiori Launchpad, add the tile to Launchpad

Figure 60: 9. Logon to Fiori Launchpad, add the tile to Launchpad

User can logon to Fiori Launchpad and add tiles from catalog to their home page.

**Note:**

Detailed information of Fiori Launchpad usage is described in UX100.

- 1
- 2
- 3

To Create a Fiori Elements Application

1. Create a SAPUI5 project from one of Fiori Elements template.
2. Choose an OData service as data provider of your project.
3. Create an annotation file associated to the OData service and write annotation in it.
4. Modify configuration file to customize your application(Optional).
5. Translate resource bundles (i18n* files) to make your application support multiple languages(Optional).
6. Deploy your Fiori Elements application to a web server (In this course, we use SAPUI5 repository in SAP Gateway server).
7. Create tile catalog, target mapping, tiles and other configuration for Fiori Launchpad.
8. Add catalog to user menu in a PFCG role , then grant the role to user.
9. Logon to Fiori Launchpad, add the tile to Launchpad.

**LESSON SUMMARY**

You should now be able to:

- Explore the basic process of building Fiori Elements applications

Unit 2

Lesson 5

Using the Core Data Services (CDS) View



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Use the CDS View and SADL

The Core Data Services (CDS) View

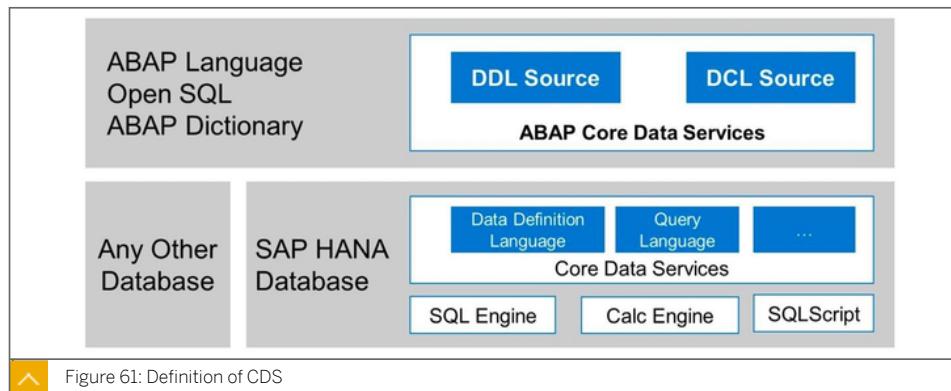


Figure 61: Definition of CDS



Note:

The purpose of this lesson is to let you have basic understand of CDS.

For detailed knowledge of CDS, please refer to course S4D430

The ABAP Core Data Services (ABAP CDS) are the platform-independent implementation of the general CDS concept for AS ABAP. The ABAP CDS make it possible to define semantic data models on the standard AS ABAP database. Unlike the SAP HANA-specific variant HANA CDS, the ABAP CDS are independent of the database system. The entities of the models defined in ABAP CDS provide enhanced access functions compared with existing database tables and views defined in ABAP Dictionary, making it possible to optimize Open SQL-based applications. This is particularly clear when an AS ABAP uses a SAP HANA database, since its in-memory characteristics can be implemented in an optimum manner.

Unit 2: SAP Fiori Elements, Overview



SQLView Name
Data source
Projection list

```

1 @AbapCatalog.sqlViewName: 'ZSAPUX403DVAR'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #CHECK
4 @EndUserText.label: 'List Report with Variant Man
5 @Data: (
6   publish: true
7 )
8 define view ZSAP_UX403D_VAR
9 as select from SEPM I SalesOrder {
10   @UI.lineItem.position: 10
11   key SalesOrder as SalesOrderID,
12   @UI.lineItem.position: 20
13   Customer.BusinessPartner as CustomerID,
14   @UI.lineItem.position: 30
15   Customer.CompanyName as CustomerName,
16   @UI.lineItem.position: 40
17   GrossAmountInTransacCurrency,
18   @UI.lineItem.position: 50
19   OverallStatus.SalesOrderOverallStatus
20 }
21
22 }
```

Figure 62: Basic Syntax of CDS

A basic CDS defines a view, the definition of CDS use SQL like syntax.

SQL View Name: For compatibility with ABAP dictionary, each CDS must have a name in Old ABAP dictionary format(Max length:16).

Data source: A cds view select data from one or several database table or cds views. If data come from more than one data source, SQL JOIN syntax can be used to select data from multiply data sources.

Projection List: The most important part of a CDS view is the projection list. In projection list, complex expression can be used to create calculated field



Note:

The process of creating a CDS view is described in the Exercise [Create a List Report using CDS with Annotation](#).

Annotations for CDS Views



```

@AbapCatalog.sqlViewName: 'ZSAPUX403DVAR'
@AbapCatalog.compiler.compareFilter: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'List Report with Variant Man
@Data: (
  publish: true
)
define view ZSAP_UX403D_VAR
as select from SEPM I SalesOrder {
  @UI.lineItem.position: 10
  key SalesOrder as SalesOrderID,
  @UI.lineItem.position: 20
  Customer.BusinessPartner as CustomerID,
  @UI.lineItem.position: 30
  Customer.CompanyName as CustomerName,
  @UI.lineItem.position: 40
  GrossAmountInTransacCurrency,
  @UI.lineItem.position: 50
  OverallStatus.SalesOrderOverallStatus
}

```

Figure 63: Annotation on CDS

- CDS view define the logic of reading data, with semantic information extract from ABAP DDIC
- CDS Annotation add additional information for a data field or the view itself
- CDS core annotation contain most common information use by all clients
- CDS domain-specific annotation contain information for special usage
- How to write annotation for a CDS view depends on the usage of the CDS view

CDS annotations are extra information to describe the data, it let the consumer of CDS view know how to use data from the CDS view.

Lesson: Using the Core Data Services (CDS) View



Domain	Usage
UI Annotations	Most important domain for Fiori Elements. Nearly one-to-one relationship between CDS annotation and OData annotation term
Consumption Annotations	Control visibility of a data field or the whole entity. Useful when defining value help for a field.
Object Model Annotations	Provide definitions of structural as well as transactional related aspects of the business data model
Semantics Annotations	Enrichment of semantic information for data field.

Figure 64: Domain Specific Annotation related to Fiori Elements

Annotations are grouped according to usage, this figure shows most relevant groups of association for Fiori Elements.



1. Go to <http://help.sap.com>
 2. Search "ABAP Programming Model for SAP Fiori" and press Enter
 3. Choose the correct result according your Kernel version
 4. Click "Reference" on left bottom of the screen.

The screenshot shows the SAP Documentation interface. In the search results, there is a link to 'ABAP Programming Model for SAP Fiori'. On the right, under 'CDS Annotations', it says 'The following list summarizes all SAP annotation frameworks and can be either ABAP annotations'. It lists several categories: ABAP Catalog Annotations, AccessControl Annotations, ClientDependent Annotations, DataLayer Annotations, EntityAnnotations, Environment Annotations, MappingAnnotations, Metadata Annotations, and Semantics Annotations. A 'Tip' button is also visible.

Figure 65: CDS Annotation Reference

To have full knowledge of CDS annotations, you can access CDS annotation document by searching **ABAP Programming Model for AP Fiori** on sap help portal.

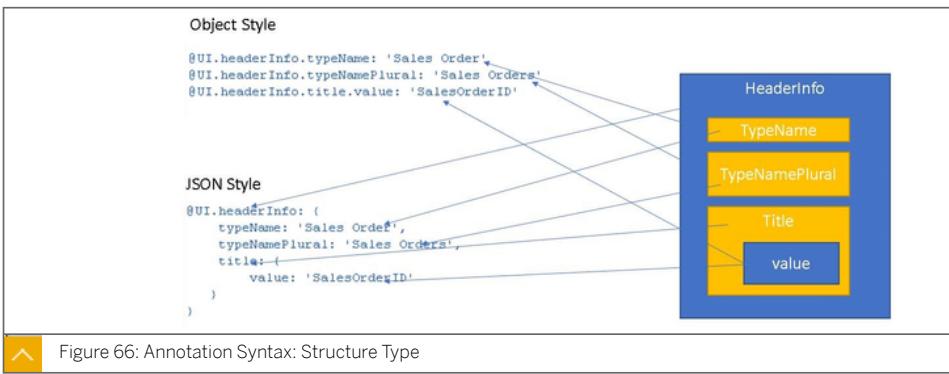


Figure 66: Annotation Syntax: Structure Type

If an annotation term represent a structure, two option is available for writing the annotation.

The object style access properties of annotation like attributes of a class, use '.' to access sub level properties.

The JSON style access properties like a JSON object, key feature of this syntax are:

- "{}" represent a structure
- Use ":" to separate property name and its value
- Use "," to separate properties

Both syntax have same meaning and can use as a combination, such as @UI.headerInfo.title:
{ value: 'SalesOrderID' }

The diagram illustrates the mapping of JSON array syntax to a table structure. On the left, a snippet of JSON code defines two line items with qualifiers, positions, and labels. Arrows point from the 'qualifier', 'position', and 'label' properties in the JSON to the corresponding columns in a table on the right. The table is titled 'UI.lineItem' and has four columns: 'qualifier', 'position', 'label', and '.....'. Two rows are shown: one for 'qualifier1' with position 10 and label 'label1', and another for 'qualifier2' with position 20 and label 'label2'.

UI.lineItem			
qualifier	position	label
qualifier1	10	label1
qualifier2	20	label2

Figure 67: Annotation Syntax: Array Type

If an annotation term represent an array, the only option is to use a JSON array.

Key feature of this syntax are:

- "{}" represent a structure
- "[]" represent an array
- Use ":" to separate property name and its value
- Use "," to separate properties
- In most cases, each line should have a property called "qualifier" to identify it

The diagram illustrates the mapping of JSON single-line array syntax to a table structure. On the left, a snippet of JSON code defines a line item with position, label, and importance properties. Arrows point from these properties to the corresponding columns in a table on the right. The table is titled 'UI.lineItem' and has four columns: 'position', 'label', 'importance', and '.....'. One row is shown: position 10, label 'label1', and importance '#HIGH'.

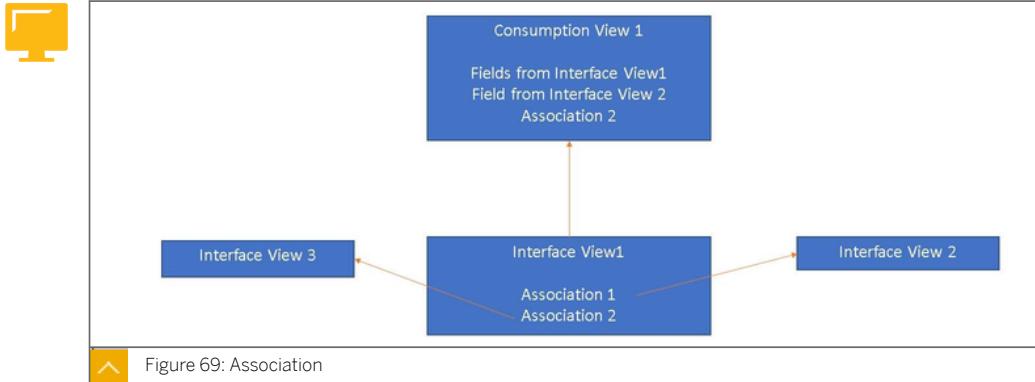
UI.lineItem			
position	label	importance
10	label1	#HIGH

Figure 68: Annotation Syntax: Single line Array Type

In some cases, a term is self is an array type, but most of its usage only have single line of record.

In this situation, you can use object style to simplify the input.

CDS Associations



A CDS view can define some associations. Association is a declare of relationship of views which can be consumed by itself or by the consumer of this cds view.

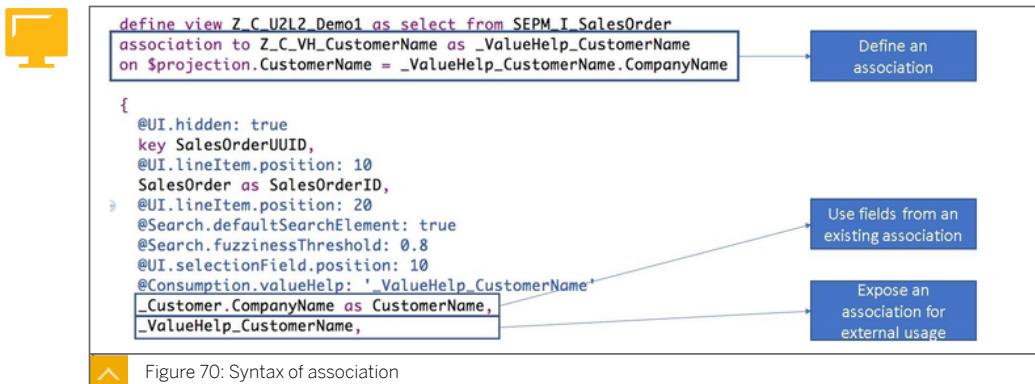
Usually you need to define some reusable CDS views, which defines associations as a represent of business entity. Then you can write a consumption CDS view , which uses interface view as its data source, to consume the data and associations.

In this diagram:

Interface View 1 have two associations connect to Interface View 3 and Interface View2

When consumption view 1 access data of interface view 1, it can:

- Access fields of Interface View1
- Access fields of Interface View 2 through association 1 of interface view 1
- Expose Association 2 of Interface View1 again, to let consumer of this view have the ability to access field of Interface View3



The figure, Syntax of association, shows the 3 main areas of an association.

These are:

Define an association:

The define of association is quite like SQL JOIN syntax. A easy way to reference fields in current project list is \$projection.

Use fields from an existing association:

You can access fields from associated CDS view by connect association name and fields name with a period '..'. The association can be defined either in current CDS view or the data source view.

Expose an association for external usage:

To expose an association for consumer of this view, write the name of association in the projection list. Since we can not distinguish association and fields, usually we use '_' as first char for association name.



LESSON SUMMARY

You should now be able to:

- Use the CDS View and SADL

Unit 2

Lesson 6

Using the Service Adaption Definition Language (SADL)



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Use SADL

Service Adaptation Definition Language (SADL)

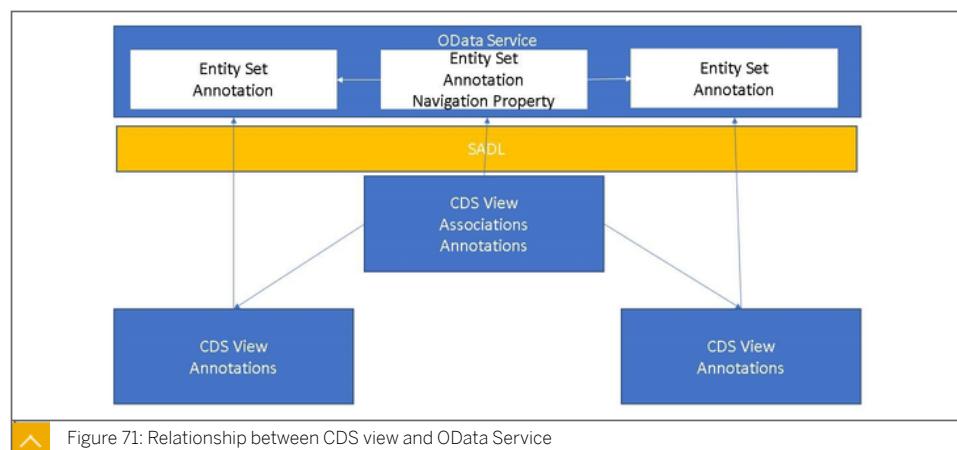


Figure 71: Relationship between CDS view and OData Service

A frame work called SADL can help you publish a group of associated CDS views to an OData service. Each CDS view will be an Entity Set, and their associations will translated to OData association. The association name will reflect in OData entity as a Navigation Property.

And also, all Fiori Elements related annotations will translated to OData annotations. In this way , you can combine your business logic, Odata service development and annotation into one CDS view.

CDS as an OData Service



```

(Multiple markers at this line:
  - Service ZCDS_ODATA_GEN_1_CDS created. Activation to be done manually (/IWFND/MAINT_SERVICE) [OData Exposure]
  - View has generated Objects)

@AbapCatalog.sqlViewName: 'ZCDSODATAGEN1'
@AbapCatalog.compiler.compareFilter: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'Generate OData Service from CDS View Demo_1'

@OData.publish:=true
define view ZCDS_ODATA_GEN_1 as select from SEPM_I_SalesOrder {
  CreatedByUser,
  CreationDateTime,
  CustomerUUID,
  GrossAmountInTransacCurrency,
  IsCreatedByBusinessPartner.
}

Limitations:
Can not change Name of entity set
Only one level association is taken into consideration
Customized ABAP implementation in SAP Gateway is not allowed

```

Figure 72: Expose CDS as an OData Service

If a CDS view has annotation "@Odata.Publish:true", an OData service is generated automatically in your backend system. The only thing you need to do is to activate it in your frontend system.

The CDS view you are working on will be transformed to an EntitySet with identical name of the generated OData service.

All CDS view associated to this view will also be transformed to an EntitySet. The association relationship will be transformed to Association and Navigation Property in OData service.

Creating an OData service by referencing a CDS



In SAP Gateway Service Builder, you can reference an existing CDS view. Then you can add associated CDS views selectively, even associations of associated view can be added to your OData Service.

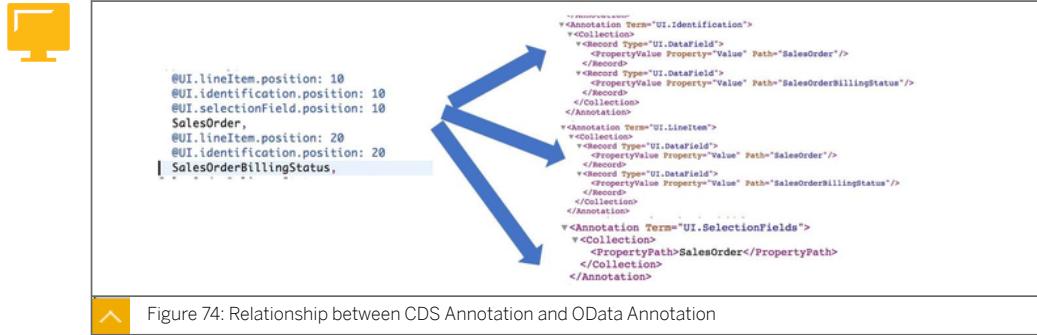
You can then override MPC or DPC's standard implementation to add write support and advanced OData features for the OData Service.



Note:

For detailed information of how to create an OData service by reference an CDS, please refer to GW100.

Relationship between CDS Annotation and OData Annotation



Like OData Annotations, CDS Annotations can apply on an EntitySet or a data field.

In most cases, there is an 1:1 relationship between OData annotation term to CDS annotation, with similar name.

CDS Annotations are grouped by target, annotations for the same target stay together.

OData Annotations do not have a fixed position. It is usually grouped by terms.

When translate to an OData annotation, the sequence of CDS annotations belonging to one term is determined by a position property.



LESSON SUMMARY

You should now be able to:

- Use SADL

Unit 2

Lesson 7

Explaining Metadata Extension



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain Metadata Extension

Metadata Extension

Metadata Extension



```

@Metadata.layer: #CUSTOMER

@UI.headerInfo.typeName: 'SalesOrder'
@UI.headerInfo.typeNamePlural: 'SalesOrders'
@UI.headerInfo.title.value: 'SalesOrderID'
annotate view ZCDS_UX403_META_00

with
{
    @UI.lineItem:{{(position: 10)}}
    SalesOrderID;
    @UI.lineItem:{{(position: 20)}}
    CustomerID;
    @UI.lineItem:{{(position: 30)}}
    CustomerName;
    @UI.lineItem:{{(position: 40)}}
    GrossAmount;
}

CDS
define view ZCDS_UX403_META_00 as select from SEPM_I_SalesOrder
key SalesOrder as SalesOrderID,
    _Customer.BusinessPartner as CustomerID,
    _Customer.CompanyName as CustomerName,
    GrossAmountInTransacCurrency as GrossAmount
;

```

Metadata Extension

Figure 75: Metadata Extension

The syntax of CDS will grow if complex annotation are need, that may reduce the readability of CDS.

A metadata extension separate annotation from business logic.

To implement a metadata extension, you need to:

- Add an annotation of `@metadata.allExtensions:true` to the CDS view
- Create a Metadata extension, write annotations for the view and its fields
- Use ":" to separate fields
- Metadata extension only allow annotation of JSON style
- Some annotations relevant to the activation of CDS, like `@odata.publish`, `@AbapCatalog.sqlViewName` are not supported in a metadata extension

Note: If there is more than one metadata extension annotation for the same CDS, please reference: https://help.sap.com/doc/abapdocu_752_index_htm/7.52/en-US/abencds_meta_data_extension_eval.htm for how to determine which metadata extension is used.



LESSON SUMMARY

You should now be able to:

- Explain Metadata Extension

Unit 2

Lesson 8

Learning Scenarios of Fiori Elements Implementation



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Learn about scenarios of Fiori Elements implementation

Scenarios

System Requirements for Fiori Elements Development with SADL



	NetWeaver	Web IDE	Related S/4 HANA Version
List Report	>= 7.50 SP01	>= 1.17	S/4 HANA 1511
Object Page	>= 7.50 SP01	>= 1.17	S/4 HANA 1511
Overview Page	>= 7.51	>= 1.17	S/4 HANA 1511
Analytic List Page	>= 7.52	>= 1.17	S/4 HANA 1709

*NetWeaver 7.50 support only few annotations, most advanced annotations are supported in 7.51
*Some features, like analytical report, are only work with HANA database



Figure 76: System Requirements for Fiori Elements Development with SADL

To use Fiori Elements in NetWeaver AS ABAP, the minimum version of NetWeaver is 7.50 SP01, otherwise the SADL will not work.

Lots of new annotations are added in each upgrade of NetWeaver, Fiori Elements will be more powerful if you choose the newest version of NetWeaver.

In general, CDS works on any database. But SAP HANA is needed when good performance is needed, due to CDS push down the calculation to the DB layer.

System Requirements for Fiori Elements Development with Local Annotation



	SAPUI5/OpenUI5	SAP_UI
List Report	>1.38	SAP_UI 7.50 and UI add-on 2.0
Object Page	>1.38	SAP_UI 7.50 and UI add-on 2.0
Overview Page	>1.44	SAP_UI 7.51
Analytic List Page	>1.48	SAP_UI 7.52

Figure 77: System Requirements for Fiori Elements Development with Local Annotation

When writing annotation as a local file using SAP WebIDE, Fiori Elements works with any OData service, even from non-SAP backend.

If SAPUI5 runs on non-ABAP platform, like SAP CP, or you're using OpenUI5, Fiori Elements only requires the version of SAPUI5/OpenUI5.

If SAPUI5 runs on ABAP platform, Fiori Elements require version of SAP_UI.

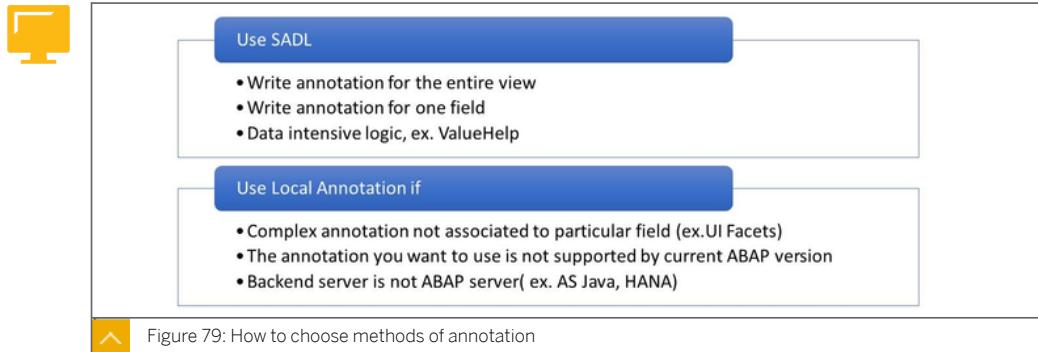
SADL vs Local Annotation



<ul style="list-style-type: none"> Advantage of SADL <ul style="list-style-type: none"> Annotation and data stay together Syntax is simple Reusable
<ul style="list-style-type: none"> Advantage of Local Annotation <ul style="list-style-type: none"> More Flexible Can handle complex annotation which is not bound to a single field or entity Vocabulary list update faster than CDS annotation Can override CDS Annotation

Figure 78: SADL vs Local Annotation

In most cases, UI annotations in CDS view can be translated to OData annotation by SADL framework. But they are not 100% same and have their own advantages.

Suggestions, to choose the correct method of annotation

The figure, How to choose methods of annotation, lists some factors for choosing the methods of annotation.

**LESSON SUMMARY**

You should now be able to:

- Learn about scenarios of Fiori Elements implementation

Unit 2

Learning Assessment

1. What are the features of Fiori Elements?

Choose the correct answers.

- A No JavaScript UI Coding.
- B Metadata-driven approach of Fiori development.
- C A replacement for traditional free style SAPUI5 programming, can satisfy all customer needs in a brand new approach.
- D Centrally Provided Templates covering Reporting, Analytic, Transaction scenarios.

2. Which of the following information about an oData service should be provided as an annotation?

Choose the correct answer.

- A The entities of an oData service.
- B Properties of an Entity Set/Collection.
- C The position for each field in a list report.
- D Data type for each property in an Entity Set/ Connection.

3. The combination of Term/Target is unique, that means for an Entity or a field, every term can be used only once.

Determine whether this statement is true or false.

- True
- False

Unit 2: Learning Assessment

4. Which of following templates display only one business entity?

Choose the correct answer.

- A List Report
- B Object Page
- C Overview Page
- D Analytic List Page

5. Using S/4 HANA 1610 with ABAP 7.51, which of following templates can you use for Fiori Element?

Choose the correct answers.

- A List Report
- B Object Page
- C Overview Page
- D Analytic List Page

6. Which info will be used for creating a destination in SAP Cloud Platform for Fiori Elements development?

Choose the correct answer.

- A Virtual name in SAP Cloud Connector.
- B Internal name in SAP Could Connector.
- C External name in SAP Cloud Connector.
- D Internal address of SAP Backend Server.

7. When creating a CDS view, the SQL view name and view name for CDS must be identical.

Determine whether this statement is true or false.

- True
- False

8. Which of following steps are needed to create an association in CDS and expose it?

Choose the correct answers.

- A** Declare an association using “association to” statement.
- B** Declare an association using “left outer join” statement.
- C** Expose the association by writing its name in projection list.
- D** Expose fields in the association by writing each field in projection list.

9. What are the limitations of publishing CDS as oData service by adding a @OData.publish:true?

Choose the correct answers.

- A** Can not expose associations.
- B** Can not change names of entity sets.
- C** Only 1 level is taken into consideration when exposing associations.
- D** No customized ABAP code in SAP Gateway.

10. In which cases, is a local annotation better than a CDS annotation?

Choose the correct answers.

- A** UI with data intensive.
- B** Annotations is for 1 field.
- C** Complex UI relevant annotations.
- D** You want to use annotations which are not support by your current ABAP version.

Unit 2

Learning Assessment - Answers

1. What are the features of Fiori Elements?

Choose the correct answers.

- A No JavaScript UI Coding.
- B Metadata-driven approach of Fiori development.
- C A replacement for traditional free style SAPUI5 programming, can satisfy all customer needs in a brand new approach.
- D Centrally Provided Templates covering Reporting, Analytic, Transaction scenarios.

2. Which of the following information about an oData service should be provided as an annotation?

Choose the correct answer.

- A The entities of an oData service.
- B Properties of an Entity Set/Collection.
- C The position for each field in a list report.
- D Data type for each property in an Entity Set/ Connection.

3. The combination of Term/Target is unique, that means for an Entity or a field, every term can be used only once.

Determine whether this statement is true or false.

- True
- False

4. Which of following templates display only one business entity?

Choose the correct answer.

- A List Report
- B Object Page
- C Overview Page
- D Analytic List Page

5. Using S/4 HANA 1610 with ABAP 7.51, which of following templates can you use for Fiori Element?

Choose the correct answers.

- A List Report
- B Object Page
- C Overview Page
- D Analytic List Page

6. Which info will be used for creating a destination in SAP Cloud Platform for Fiori Elements development?

Choose the correct answer.

- A Virtual name in SAP Cloud Connector.
- B Internal name in SAP Could Connector.
- C External name in SAP Cloud Connector.
- D Internal address of SAP Backend Server.

7. When creating a CDS view, the SQL view name and view name for CDS must be identical.

Determine whether this statement is true or false.

- True
- False

Unit 2: Learning Assessment - Answers

8. Which of following steps are needed to create an association in CDS and expose it?

Choose the correct answers.

- A Declare an association using “association to” statement.
- B Declare an association using “left outer join” statement.
- C Expose the association by writing its name in projection list.
- D Expose fields in the association by writing each field in projection list.

9. What are the limitations of publishing CDS as oData service by adding a @*OData.publish:true*?

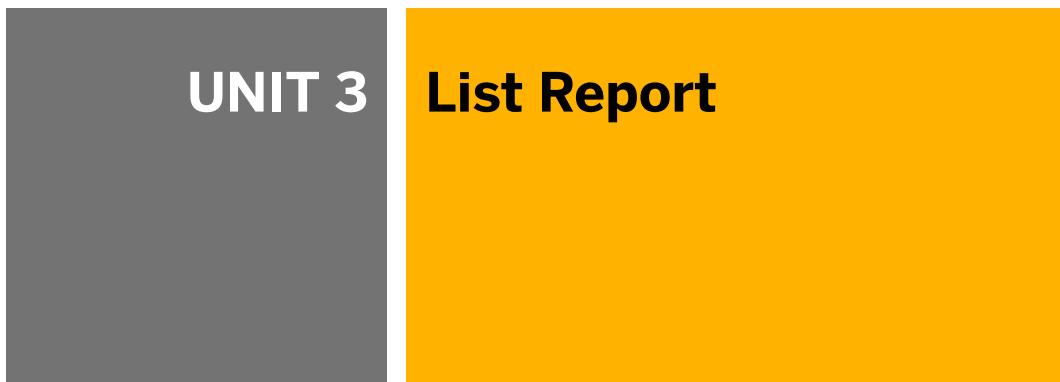
Choose the correct answers.

- A Can not expose associations.
- B Can not change names of entity sets.
- C Only 1 level is taken into consideration when exposing associations.
- D No customized ABAP code in SAP Gateway.

10. In which cases, is a local annotation better than a CDS annotation?

Choose the correct answers.

- A UI with data intensive.
- B Annotations is for 1 field.
- C Complex UI relevant annotations.
- D You want to use annotations which are not support by your current ABAP version.



The slide features a dark grey header section on the left containing the text "UNIT 3" in white. To its right is a large yellow section containing the text "List Report" in black.

Lesson 1

Explaining Basic Annotations for List Report 78

Lesson 2

Using Searching and Filtering Data 83

Lesson 3

Providing the Value Help 88

Lesson 4

Explaining Variant Management 91

UNIT OBJECTIVES

- Explain basic annotations for creating a list report
- Describe options for adjusting the display of columns
- Use Searching and Filtering data
- Provide the value help
- Explain Variant Management

Unit 3 Lesson 1

Explaining Basic Annotations for List Report



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain basic annotations for creating a list report
- Describe options for adjusting the display of columns

Basic Annotations for Creating a List Report

Mandatory Annotations

Annotation in CDS

Generated XML

Annotation Modeler View

```

5
6 @OData.publish: true
7 //@UI.headerInfo.typeNamePlural: 'Sales Orders'
8
9
10 define view Z_C_02L1_Demo1 as select from SEPM_I_SalesOrder {
11   //@UI.hidden: true
12   key SalesOrderUUID,
13   //@UI.lineItem.position: 10
14 }
```

Annotation Terms="UI.HeaderInfo">

```

<Record Type="UI.HeaderInfoType">
  <PropertyValue Property="TypeName" String="Basic List Report"/>
  <PropertyValue Property="TypeNamePlural" String="Sales Orders"/>
</Record>
</Annotation>
```

UI.HeaderInfo	Type Name
Type Name *	String
Type Name Plural *	String

Sales Orders (0) Standard

Figure 80: Mandatory Annotations

The following are typical mandatory annotations:

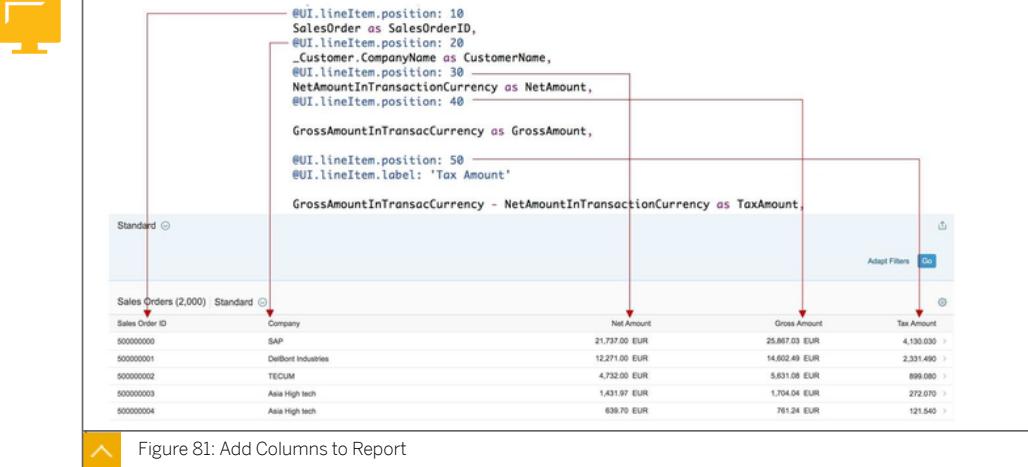
@OData.publish:true

Auto exposes the CDS view as OData Service.

@UI.headerInfo.typeNamePlural

Determines the text displayed on the up-left corner of the list report.

Adding Columns to Report



The screenshot shows the SAP List Report interface. At the top, there is XML code defining annotations for columns:

```

@UI.lineItem.positions: 10
SalesOrder as SalesOrderID,
@UI.lineItem.position: 20
_Customer.CompanyName as CustomerName,
@UI.lineItem.position: 30
NetAmountInTransactionCurrency as NetAmount,
@UI.lineItem.position: 40
GrossAmountInTransacCurrency as GrossAmount,
@UI.lineItem.position: 50
@UI.lineItem.label: 'Tax Amount'
GrossAmountInTransacCurrency - NetAmountInTransactionCurrency as TaxAmount

```

The report table displays data for Sales Orders (2,000). The columns are labeled: Sales Order ID, Company, Net Amount, Gross Amount, and Tax Amount. Arrows point from the XML annotations to the corresponding columns in the report table.

Figure 81: Add Columns to Report

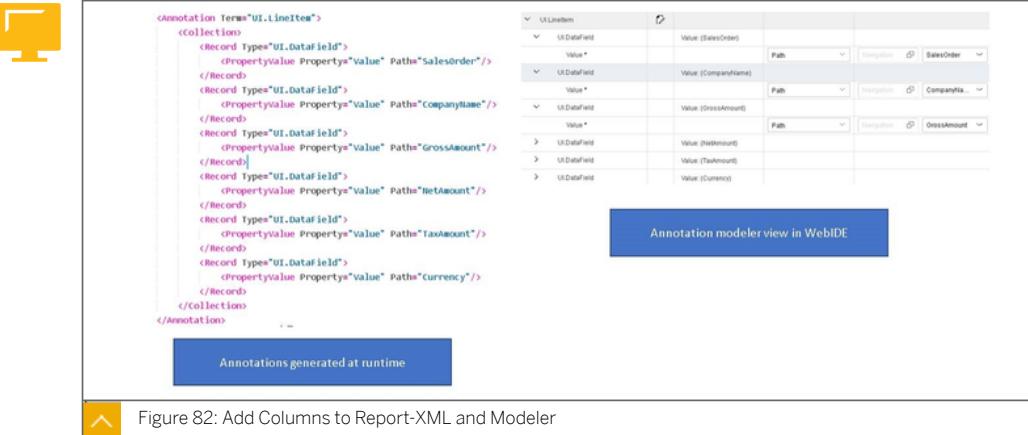
Annotation `@UI.lineItem` is used to set default columns for list report.

The Data types for `@UI.lineItem` is a collection of Data Fields, but in most cases only one `@UI.lineItem` exist for a field.

The mandatory property is **position**, which determines the sequence of columns. The label for each field comes from field label of data element for a fields.

You can also add calculated field to list report. Since the fields do not reference a data element, you need to either assign it a label by using `@UI.lineItem.label`, or convert the data type to a data element by using function cast.

Add Columns to Report-XML and Modeler



The screenshot shows the SAP Annotation Modeler view in WebIDE. On the left, the XML code for annotations is shown:

```

<Annotation Terms="UI.lineItem">
<Collection>
<Record Type="UI.DataField">
<PropertyValue Property="Value" Path="SalesOrder"/>
</Record>
<Record Type="UI.DataField">
<PropertyValue Property="Value" Path="CompanyName"/>
</Record>
<Record Type="UI.DataField">
<PropertyValue Property="Value" Path="GrossAmount"/>
</Record>
<Record Type="UI.DataField">
<PropertyValue Property="Value" Path="NetAmount"/>
</Record>
<Record Type="UI.DataField">
<PropertyValue Property="Value" Path="TaxAmount"/>
</Record>
<Record Type="UI.DataField">
<PropertyValue Property="Value" Path="Currency"/>
</Record>
</Collection>
</Annotation>

```

On the right, the "Annotations generated at runtime" section shows the generated annotations:

UI.lineItem	UI.DataField	Value	Path	Navigation	Label
	Value *	Value (SalesOrder)	SalesOrder		
	Value *	Value (CompanyName)	CompanyName		
	Value *	Value (GrossAmount)	GrossAmount		
	Value *	Value (NetAmount)	NetAmount		
	Value *	Value (TaxAmount)	TaxAmount		
	Value *	Value (Currency)	Currency		

Figure 82: Add Columns to Report-XML and Modeler

The OData annotation generated as XML format looks like the left part of the figure. You can also open it using annotation modeler in WebIDE.

Semantic information for Amount/Quantity fields

The screenshot illustrates the addition of semantic information for Amount/Quantity fields. It shows two code snippets: 'Before' and 'After'. In the 'Before' snippet, there is no semantic information for currency. In the 'After' snippet, annotations like `@UI.lineItem.position: 50`, `@UI.lineItem.label: 'Tax Amount'`, `@Semantics.amount.currencyCode: 'Currency'`, and `@Semantics.currencyCode: true` are added. Below the code, a red arrow points down to a UI preview where the currency codes 'EUR' are now displayed next to the numerical values in a table.

Figure 83: Add semantic information for Amount/Quantity fields

Information about adding semantic Information for Amount/Quantity fields:

- For amount and quantity fields, you should point out where to find its currency/unit fields.
- Fields selecting from DDIC, or other CDS entities may already have this information. If it is defined correctly, the currency/unit will be displayed with the amount/quantity fields.

If it's not displayed correctly, you should add semantic information in your CDS view, using

- `@semantics.currencycode/@semantics.unitofmeasure` to annotate for currency/unit field.
- `@semantics.amount.currencycode/@semantics.quantity.unitofmeasure` to annotate amount/quantity field and tell them which field has the information of currencyCode/ unit of measure.

You should always annotate semantics information for calculated fields you declared in your CDS view.

Options for Adjusting the Display of Columns

Importance of Columns

The screenshot shows how column importance is handled across different devices. Annotations like `@UI.lineItem.importance: #MEDIUM` and `@UI.lineItem.importance: #LOW` are used to control visibility. The 'Desktop' view shows all columns. The 'Tablet' view shows a subset of columns. The 'Phone' view shows only the most important columns. Red boxes highlight the annotated code and the columns that are displayed on each device.

Figure 84: Importance of Columns

Controls used in list report is designed for all clients.

Different clients have different screen width, the fields displayed in a Smart Phone should less then Desktop Browser.

Using `@UI.lineitem.importance` to determine in which clients the field should display:

- #HIGH: Default value, display in all clients
- #MEDIUM: Only display in desktop browser or tablet
- #LOW: Only display in desktop browser

Importance of Columns-XML and Modeler

The screenshot shows the SAP WebIDE Annotation Modeler interface. It displays two `UI.DataField` entries. The first entry has a value of `(TaxAmount)` and an importance of `Medium`. The second entry has a value of `(Currency)` and an importance of `Low`. Below the interface, the generated OData annotation XML is shown:

```

<Record Type="UI.DataField">
  <PropertyValue Property="Value" Path="NetAmount"/>
  <Annotation Term="UI.Importance" EnumMember="UI.ImportanceType/Medium"/>
</Record>
<Record Type="UI.Datafield">
  <PropertyValue Property="Value" Path="TaxAmount"/>
  <Annotation Term="UI.Importance" EnumMember="UI.ImportanceType/Low"/>
</Record>

```

▲ Figure 85: Importance of Columns-XML and Modeler

The OData annotation generated as XML format looks like the upper part of the figure. You can also open it using annotation modeler in WebIDE.

Hiding Fields

The screenshot shows the SAP Fiori View Settings dialog. It has a list of fields under the `Columns` tab. The fields listed are: Sales Order ID, Company, Net Amount, Gross Amount, Tax Amount, Currency Status, Currency Code, Lifecycle Status, Ordering Status, Overall Status, and Sales Order UUID. The `Currency Code` checkbox is highlighted with a red rectangle. At the top, there are buttons for `View Settings`, `Sort`, and `Group`.

▲ Figure 86: Hide fields from Customizing of Columns by User

Some fields are not suitable for displaying on the report. You can use `@UI.hidden` to hide them in the table settings dialog.



LESSON SUMMARY

You should now be able to:

- Explain basic annotations for creating a list report
- Describe options for adjusting the display of columns

Unit 3 Lesson 2

Using Searching and Filtering Data



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Use Searching and Filtering data

Searching and Filtering Data



The screenshot shows a SAP Fiori List Report interface. On the left, there is a sidebar with two sections: "Search Field" and "Selection Field". The "Search Field" section contains three bullet points: "Find a keyword in multiply fields", "Support fuzzy search", and "No value help dialog". The "Selection Field" section contains two bullet points: "1 : 1 relationship between selection field and OData properties" and "Value help dialog can be used to search data". To the right of the sidebar, there is a main content area. At the top of this area, there is a "Search Field" input field with a magnifying glass icon. Below it, there is a "Sales Orders (0) | Standard" button. Underneath the button, there are two input fields: "Sales Order ID" and "Company Name", each with a small "Standard" button next to it. A blue callout box labeled "Search Field" points to the search input field, and another blue callout box labeled "Selection Fields" points to the "Customer UID:" input field.

Figure 87: Search Field and Selection Field

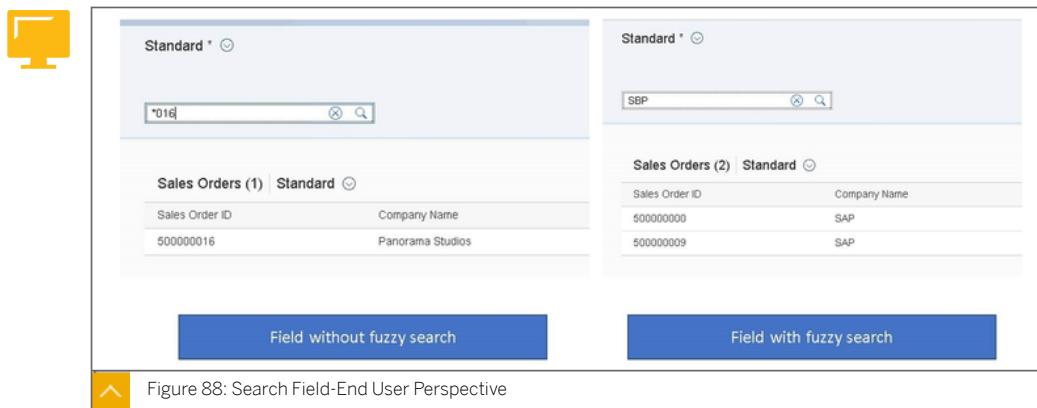
Searching and filtering data are fundamental functions for a report.

The List Report supports two ways of search and filtering:

Search Field is a field in the top left corner of a list report. User can search data by simply enter a keyword in the search field. The report will search data across several fields according the rule defined by developer.

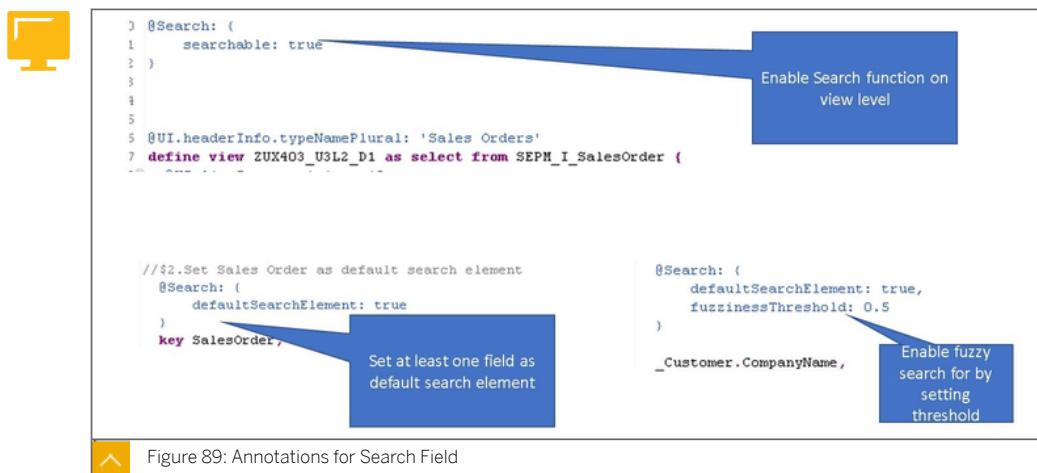
Selection Fields are some input fields with value help dialogs on top of a list report. Each selection fields represent a filter for one property of the backend OData service. User can even customize selection fields for his/her own and save it as a variant.

Unit 3: List Report



For fields without fuzzy search function. User must input the exactly same keyword as the data, or use wildcard characters (*) and (?) to match data.

For fields with fuzzy search function, even user has typo in his/her key word, similar results will appear. In this example, the key word is 'SBP', but SAP appears in the result.



To add search field for a list report, developer must unable it on CDS view level by adding a @Search.searchable:true annotation.

At least one field in the will should be set as default search element. If a field needs fuzzy search function, a threshold value must be set.

The threshold value is a number between 0 and 1. The search result will be more accurate as well as less record when the value increase. And vice versa.

Generally a suggest value in common usage is 0.8



```

<EntityContainer Name="ZUX403_U3L2_D1_CDS_Entities" m:IsDefaultEntityContainer="true"
    sap:supported-formats="atom json xlsx">
    <EntitySet Name="ZUX403_U3L2_D1" EntityType="ZUX403_U3L2_D1_CDS.ZUX403_U3L2_D1Type"
        sap:createable="false" sap:updatable="false" sap:deletable="false" sap:searchable="true"
        sap:content-version="1"/>
</EntityContainer>

```

Generated OData metadata is affected by the search annotation

Figure 90: Search Field in OData Metadata

Not like other annotations. Annotations for search fields only affect metadata of OData service.

It is an OData V2 annotation on the entity set generated by the CDS view.

Default search elements and fuzzy search related annotations are not relevant with UI, so that it will not appear in OData annotation.



A value help button is displayed on the right of a search field

A dialog with search and select capability will help user choose values

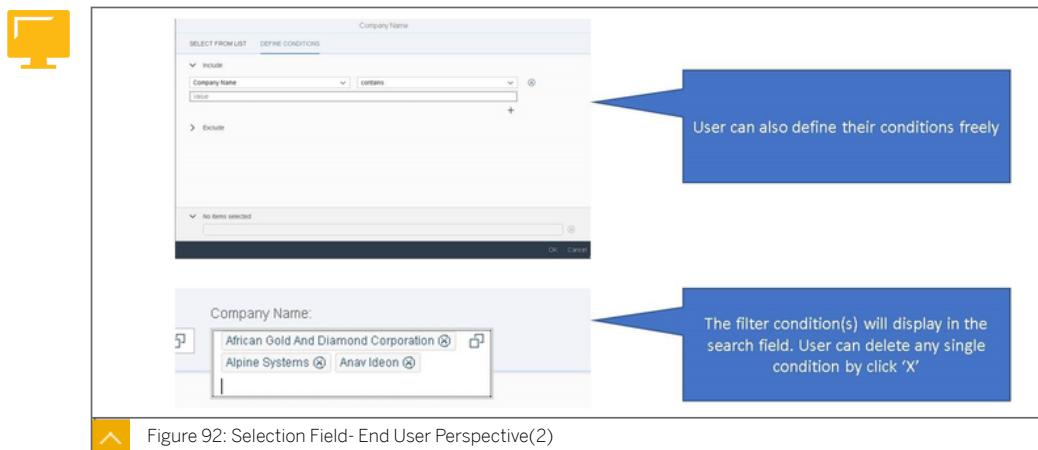
Figure 91: Selection Field- End User Perspective(1)

For each search field, a value help button is displayed on the right.

After click the button, a dialog with search and select capability will help user choose values.

It is developer's responsibility to provide a value help dialog for end user.

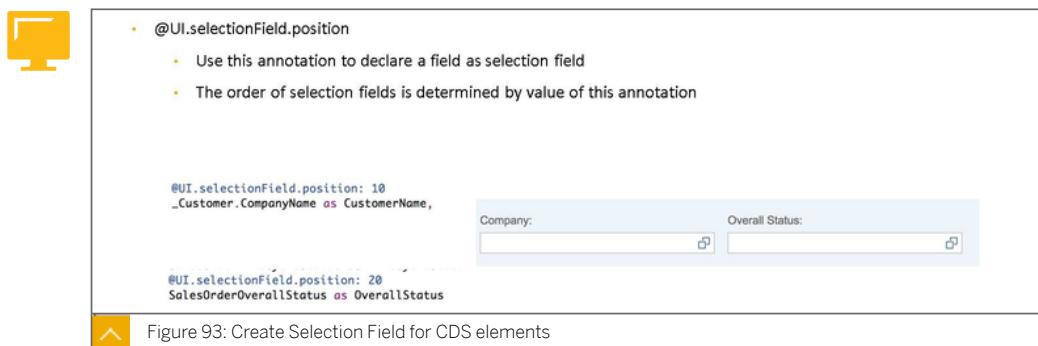
Unit 3: List Report



If choose value from a list is not the case user wanted, it also support free style input of conditions.

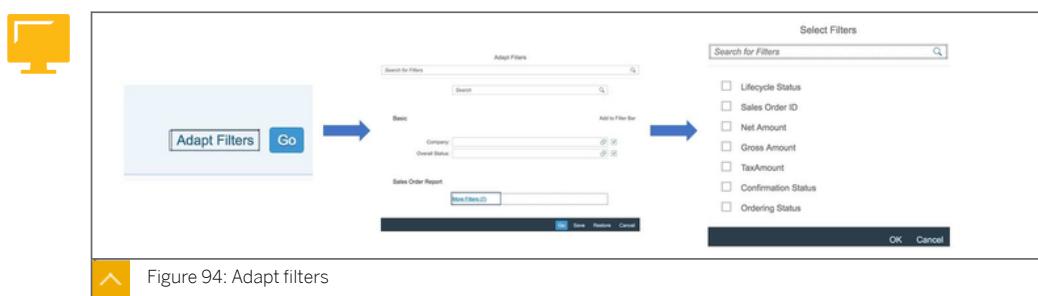
In case the developer hadn't provided a value help dialog, only free style input is supported.

When the condition input is done, all conditions will display in the search field. User can delete any single condition by click 'X' on the right



To enable selection field, use @UI.selectionField.position as element annotation on the field needs to be a filter.

Even fields does not display as a column can be a filter condition



However, at runtime, end user can choose any fields as a filter by click 'Adapt Filters'.



LESSON SUMMARY

You should now be able to:

- Use Searching and Filtering data

Unit 3

Lesson 3

Providing the Value Help



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Provide the value help

Value Help



The screenshot shows a SAP Fiori application interface. At the top, there is a code snippet:

```
@ObjectModel.foreignKey.association: '_OverallStatus'  
SalesOrderOverallStatus as OverallStatus,  
//expose the association  
_OverallStatus
```

Below the code, a blue callout box points to the word "OverallStatus" in the code and the "Overall Status" field in the dialog, containing the text: "Define value help dialog by reference an association with foreign key relationship".

The dialog itself has a title bar "Overall Status" and two tabs: "SELECT FROM LIST" (selected) and "DEFINE CONDITIONS". It contains two input fields: "Overall Status:" and "Short Descript.". Below these are two sections: "Items" and "List". The "Items" section shows a table with columns "Overall Status" and "Short Descript.", listing items D (Delivered), I (In Progress), N (New), P (Paid), and X (Canceled). The "List" section shows a table with columns "Overall Status" and "Short Descript.", listing items D (Delivered), I (In Progress), N (New), P (Paid), and X (Canceled).

Figure 95: Providing Value Help with Foreign Key Association

If there is an association for a field, you can use this association as a check table for the field. Then you can provide a value help for user. The steps are:

- Find appropriate association for value help
- Annotate the selection field with `@ObjectModel.foreignKey.association`
- Expose the association

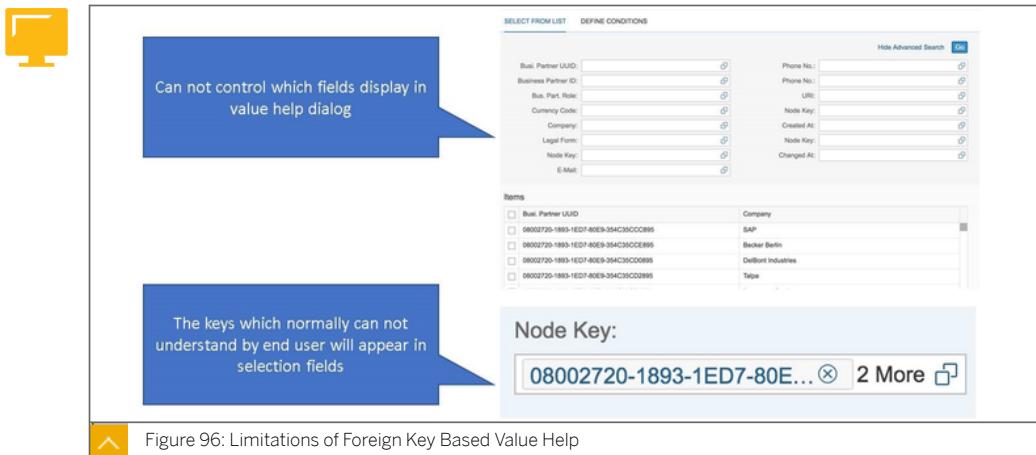


Figure 96: Limitations of Foreign Key Based Value Help

Although it's easy to declare a foreign key based value help, it has some limitations:

- Normally, only technical key fields of a business entity have the association to other views, which are fields we want to hide from end user.
- In the search help window, only key field and one field annotated with `@semantics.text:true` can be displayed in the search result window
- All fields from check view will be present as selection fields of the value help screen
- Here is a bad example of a selection field for Company UUID field

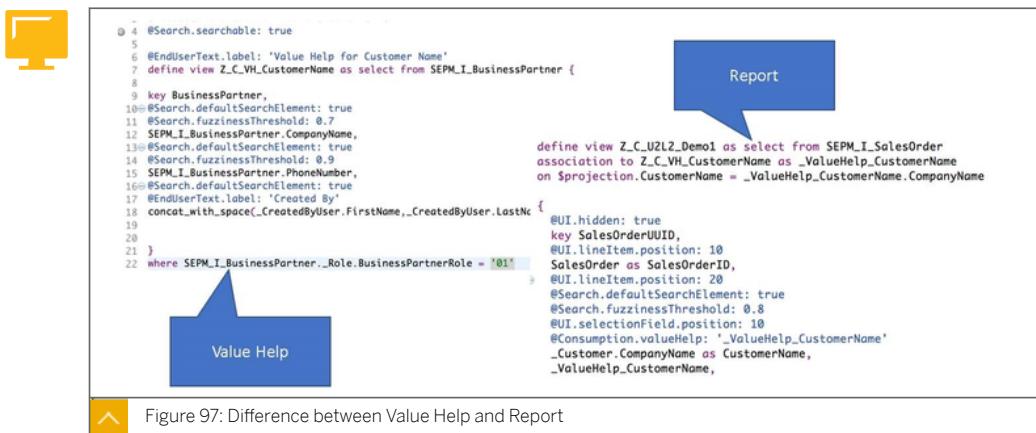


Figure 97: Difference between Value Help and Report

For better user experience, you can create a modeled value help view as following steps:

- Create a CDS view, select data from existing table or cds view, expose only fields you want to display in value help
- You can add additional where conditions and calculated fields for your value help view
- Create an association from your report CDS view to the value help view, associate them with selection field, which is understandable by end users
- Annotate the selection field with `@Comsuption.valuehelp:'<association_name>`

Unit 3: List Report

- Expose the association
- You can add full text fuzzy search function for your value help by add @search annotation in your value help CDS view



Note:

- Do not use calculated fields as search condition if unnecessary, it may leads to performance issue
- For calculated fields in value help view, use @EndUserText.label to set a label
- Redeploy your report CDS view after you changed and activated the value help view

When running a list report, users can only filter data using fields provided as selection fields by developer.

True

False



LESSON SUMMARY

You should now be able to:

- Provide the value help

Unit 3

Lesson 4

Explaining Variant Management



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain Variant Management

Variant Management



I personalize apps for myself.



I adapt apps for my team.



I develop new apps for my company.



Figure 98: SAPUI5 Flexibility Services

SAPUI5 flexibility services enable functions for different user groups to personalize SAP Fiori apps, adapt their user interface at runtime, and develop new apps.

The service store user-specific data and client-wide data in a special repository on abap server, which is called layered repository.

Pre-request of SAPUI5 Flexibility Service

The following are prerequisites of the SAPUI5 flexibility service:



- Server: ABAP front-end server and SAP NetWeaver 7.31 SP 11 or higher
- Following icf node must be activated:
 - /SAP/BC/UI5_UI5
 - /SAP/BC/LREP
- For creating shared variants
 - ABAP authorization object /UIF/FLEX with authorization field /UIF/KEYU = 'X' is required

Unit 3: List Report

- A transport request with task assigned to the user is required

Currently, Fiori support variant management only when the backend system is NetWeaver AS ABAP.

Variant Selection

Default (radio button) Filtered By (1): Overall Status

Variants

Standard
Default
ee
test

Manage Save Save As

Manage Variants

Name	Type	Default	Execute on Select	Author
Standard	Private	<input type="radio"/>	<input checked="" type="checkbox"/>	SAP
Default	Shared	<input checked="" type="radio"/>	<input checked="" type="checkbox"/>	DEVELOPER
ee	Private	<input type="radio"/>	<input checked="" type="checkbox"/>	DEVELOPER
test	Private	<input type="radio"/>	<input type="checkbox"/>	DEVELOPER

OK Cancel

Figure 99: Selection of Variant

The figure, Selection of Variant shows 2 screenshots of two different way to select variants.

The Selection Variant stores filter conditions for users.

After filtering the data, a user can save the condition as a variant. These are options for a selection variant:

- Name: Name of the variant
- Default: Select this variant when user entering the report
- Execute on Select: Execute query immediately after user select the variant
- Shared: Shared variant, a transport request must assigned when set a variant to shared.

The user can manage its variants by clicking the *Manage* button.

Presentation Variant

Orders (1,400) Default (radio)

Variants

Standard
ByCompany
Default

Manage Save Save As

Manage Variants

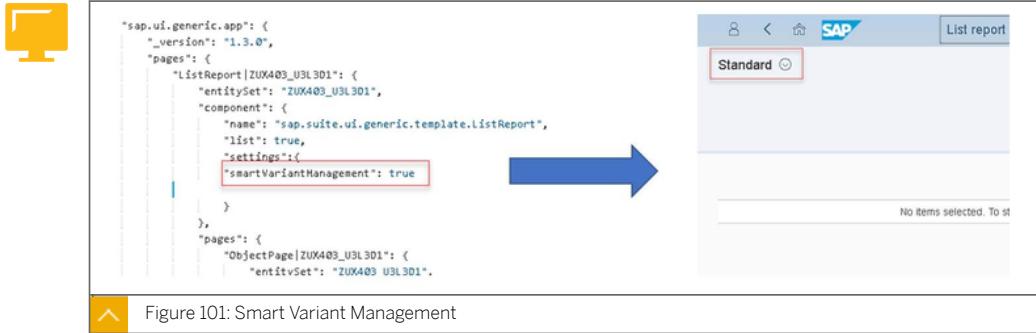
Name	Type	Default	Author
Standard	Private	<input type="radio"/>	SAP
ByCompany	Private	<input type="radio"/>	DEVELOPER
Default	Shared	<input checked="" type="radio"/>	DEVELOPER

OK Cancel

Figure 100: Presentation Variant

The figure, Presentation Variant shows 2 screenshots of two different ways, how variants are presented.

Smart Variant Management



To combine selection variant and presentation variant into one, smart variant management can be used.

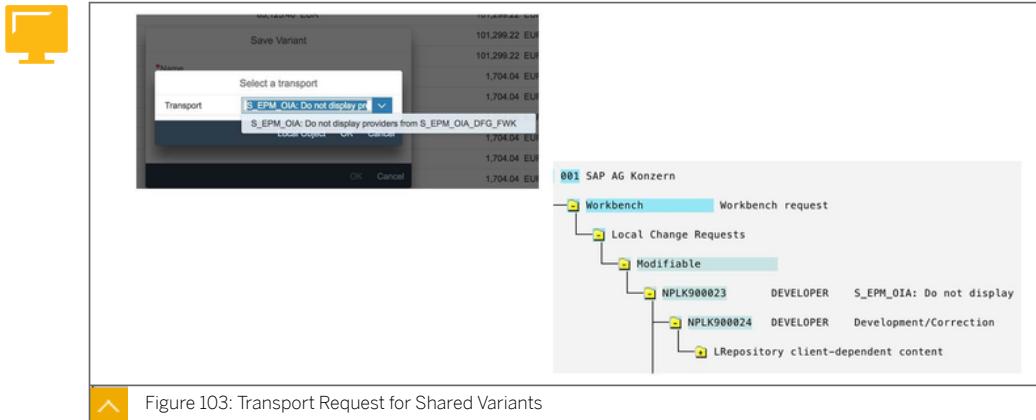
To enable smart variant management, edit manifest.json in WebIDE, locate domain "sap.ui.generic.app", modify code like the figure above.

Disable Variant Management



To disable variant management function, edit manifest.json in WebIDE, locate domain "sap.ui.generic.app", modify code like the figure above.

Transport Request for Shared Variants



Unit 3: List Report

A transport request, with a task assigned to the user, should be prepared before creating a shared variant. In *SE10*, you can find *LRepository* content in your task.

LRepository objects are client dependent, so either a workbench request or a customizing request is required.



LESSON SUMMARY

You should now be able to:

- Explain Variant Management

Unit 3

Learning Assessment

1. For a list report, which annotation is used to describe the name of business entity displayed on the report?

Choose the correct answer.

- A @UI.lineItem.title
- B @UI.headerInfo.type
- C @UI.headerInfo.typeName
- D @UI.headerInfo.typeNamePlural

2. How can you hide some columns when the list report is accessed by a mobile phone?

Choose the correct answer.

- A Put all fields, which are not important at the end of the report, when screen gets smaller, the fields will hide automatically.
- B Add @UI.hidden for fields. not import.
- C Prepare a different version of @UI.lineItem and assign them as a different qualifier.
- D Set UI.lineItem.importance for those fields as #LOW or @Medium.

3. What is the use of the annotation @UI.hidden?

Choose the correct answer.

- A Prevent a column from display on the UI.
- B Not expose these fields as a property of OData service.
- C Prevent a column selected by user when customizing table settings.
- D Create a invisible column to save the value in a hidden control of HTML.

Unit 3: Learning Assessment

4. Which of following description about search field is NOT true?

Choose the correct answer.

- A The Search field is searching on more than one data field.
- B There is only 1 search field per list report.
- C Search field searches for only 1 data field.
- D Search field support fuzzy search.

5. Choose available options for creating a value help for a selection field.

Choose the correct answers.

- A By adding a foreign key annotation.
- B By adding a value help annotation.
- C If the domain which associates to the field, has a fixed value, the value help will be generated automatically.
- D By adding annotations to list all possible in source code of CDS.

6. When running a list report, users can only filter data using fields provided as selection fields by developer.

Determine whether this statement is true or false.

- True
- False

7. What steps needs to be done in *manifest.json* if you need to hide variant function in a list report?

Choose the correct answers.

- A Add setting to enable smart Variant Management.
- B Add setting to disable smart Variant Management.
- C Set setting *variantManagementHidden* to true.
- D Set setting *showVariantManagement* to false.

Unit 3

Learning Assessment - Answers

1. For a list report, which annotation is used to describe the name of business entity displayed on the report?

Choose the correct answer.

- A @UI.lineItem.title
- B @UI.headerInfo.type
- C @UI.headerInfo.typeName
- D @UI.headerInfo.typeNamePlural

2. How can you hide some columns when the list report is accessed by a mobile phone?

Choose the correct answer.

- A Put all fields, which are not important at the end of the report, when screen gets smaller, the fields will hide automatically.
- B Add @UI.hidden for fields. not import.
- C Prepare a different version of @UI.lineItem and assign them as a different qualifier.
- D Set UI.lineItem.importance for those fields as #LOW or @Medium.

3. What is the use of the annotation @UI.hidden?

Choose the correct answer.

- A Prevent a column from display on the UI.
- B Not expose these fields as a property of OData service.
- C Prevent a column selected by user when customizing table settings.
- D Create a invisible column to save the value in a hidden control of HTML.

Unit 3: Learning Assessment - Answers

4. Which of following description about search field is NOT true?

Choose the correct answer.

- A The Search field is searching on more than one data field.
- B There is only 1 search field per list report.
- C Search field searches for only 1 data field.
- D Search field support fuzzy search.

5. Choose available options for creating a value help for a selection field.

Choose the correct answers.

- A By adding a foreign key annotation.
- B By adding a value help annotation.
- C If the domain which associates to the field, has a fixed value, the value help will be generated automatically.
- D By adding annotations to list all possible in source code of CDS.

6. When running a list report, users can only filter data using fields provided as selection fields by developer.

Determine whether this statement is true or false.

- True
- False

7. What steps needs to be done in *manifest.json* if you need to hide variant function in a list report?

Choose the correct answers.

- A Add setting to enable smart Variant Management.
- B Add setting to disable smart Variant Management.
- C Set setting *variantManagementHidden* to true.
- D Set setting *showVariantManagement* to false.

A horizontal banner divided into two sections. The left section is dark grey with the text "UNIT 4" in white. The right section is yellow with the text "Object Page" in black.

UNIT 4

Object Page

Lesson 1

Using Basic Annotations for Object Pages	100
--	-----

Lesson 2

Using Header Facets for Object Pages	102
--------------------------------------	-----

Lesson 3

Using Sections and Facets in Object Pages	108
---	-----

UNIT OBJECTIVES

- Use Basic Annotations for Object Pages
- Use Header Facets for Object Pages
- Use Sections and Facets in Object Pages

Unit 4

Lesson 1

Using Basic Annotations for Object Pages



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Use Basic Annotations for Object Pages

Basic Annotations for Object Pages



General Information	
Sales Order ID:	500000000
Business Partner ID:	100000000
Company Name:	SAP
Total Gross Amount:	24,481.03
Total Net Amount:	21,737.03
Overall Status:	Open
Payment Method:	Bank Transfer

Figure 104: A simple object page

An object page is displayed when end user press a row in list report.

A basic object page contains a header area and a body.

In the top, a name describes the type of object displayed in the center.

The header area contains title, description, image of the record, the value should bind to a property of the OData service

In the body area, by default, fields annotated with `@UI.identification` will displayed.

Annotations for basic object page

Essential Annotations for basic object pages are:



`@UI.headerinfo.typeName`

Indicates the type name of the business object.

`@UI.headerinfo.title.value`

Points to the ID data field.

`@UI.headerinfo.description.value`

Point to the Description data field.

`@UI.identification`

Like `UI.LineItem`, fields with this annotation will be displayed in the *General Information* section of the page.

Example of a Business Object Page

The screenshot shows a computer monitor icon on the left and a screenshot of an SAP Business Object Page on the right. The page has a header with annotations: `@UI.headerInfo.typeName: 'Sales Order'`, `@UI.headerInfo.title.value: 'SalesOrderID'`, and `@UI.headerInfo.description.value: 'CustomerName'`. Below the header is a search bar labeled `@Search.searchable: true`. The main content area contains a code snippet defining a view named `Z_C_U3L1_Demo1` with associations and projections. A specific line item is highlighted with annotations: `@UI.lineItem.position: 30`, `@UI.identification.position: 20`, and `NetAmountInTransactionCurrency as NetAmount`. This line item is connected by a line to a detailed view of the Net Amount field in the General Information section, which shows values 21,737.00 and 25,867.03. Other sections visible include Company (SAP), Net Amount, Gross Amount, and various status indicators like Confirmation Status (P), Ordering Status (O), Lifecycle Status (C), and Overall Status (Paid (P)).

Figure 105: Example of a Business Object Page

The figure, Example of a Business Object Page, displays an example for a simple object page. From this figure you can see, how `@UI.headerInfo` and `@UI.identification` are displayed on the Object Page.



LESSON SUMMARY

You should now be able to:

- Use Basic Annotations for Object Pages

Unit 4

Lesson 2

Using Header Facets for Object Pages



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Use Header Facets for Object Pages

Header Facets for Object Pages

Define Header Facets



Header Facets

Pieces of information placed on header part of object page

Figure 106: Define Header Facets

Header facets are pieces of information placed on header part of an Object Page.

Generally, key information and status information should put in this area

Types of Header Facets



General Information

Quantity: 400.000 KG
Weight: 400.000 KG
Supplier: SAP

Product Description

Notebook Basic 15 with 2.80 GHz quad core, 15" LCD, 4 GB DDR3 RAM, 500 GB Hard Disc, Windows 8 Pro

Field Group Facet

Plain Text Facet

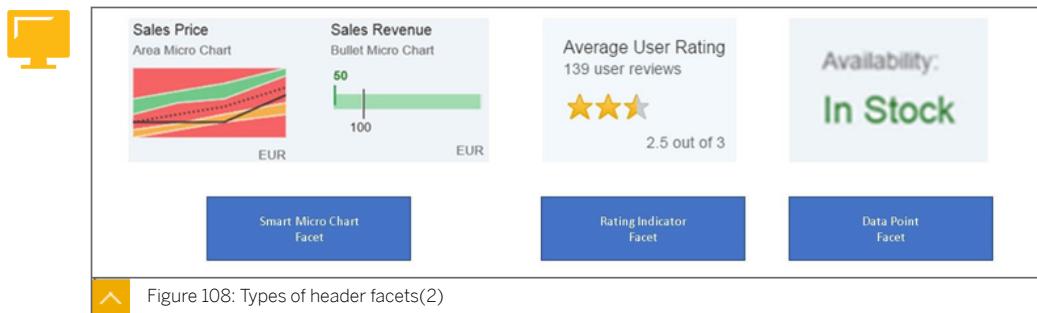
Communication Facet

Figure 107: Types of Header Facets(1)

Field Group Facet display a group of fields with a label.

Plain Text Facet display description information from exactly one field.

Communication Facet generate a hyper link, which user can access communication information(like phone, email ...) by click the link.



Smart Micro Chart Facet displays a micro chart to help user get most important KPI information.

Rating Indicator facet displays rating information as stars.

Data Point facet display a key figure or status information in different color or style to help user get the meaning in a glance.



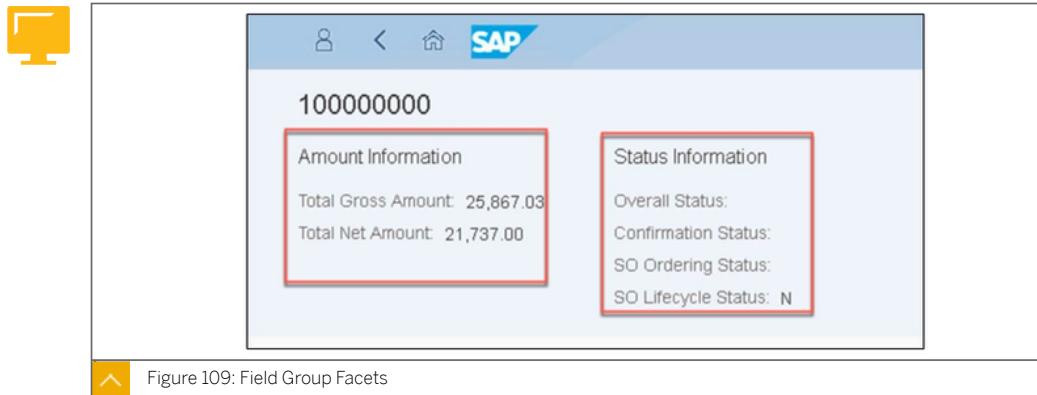
Note:

Smart Micro Chart Facet, Rating Indicator Facet and Data Point Facet will be covered in later chapter of this course.

For a full list of types of header facets, open SAPUI5 SDK and navigate to following path.

Documentation->Developing Apps with SAP Fiori Elements->List Report and Object Page->Configuring Object Page Features->Setting up the Object Page Header->Header Facets

Field Group Facets

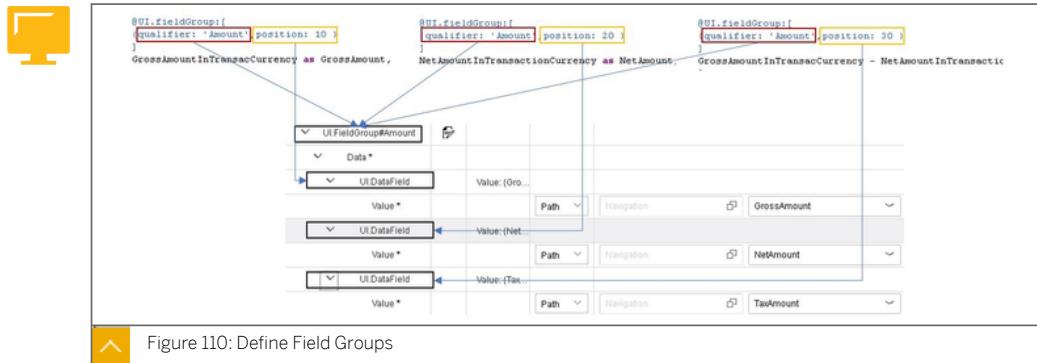


A field group defines the fields displayed within a specific reference facet. For example, the figure shows the following fields within the Amount Information and Status Information:

- Amount Information

Unit 4: Object Page

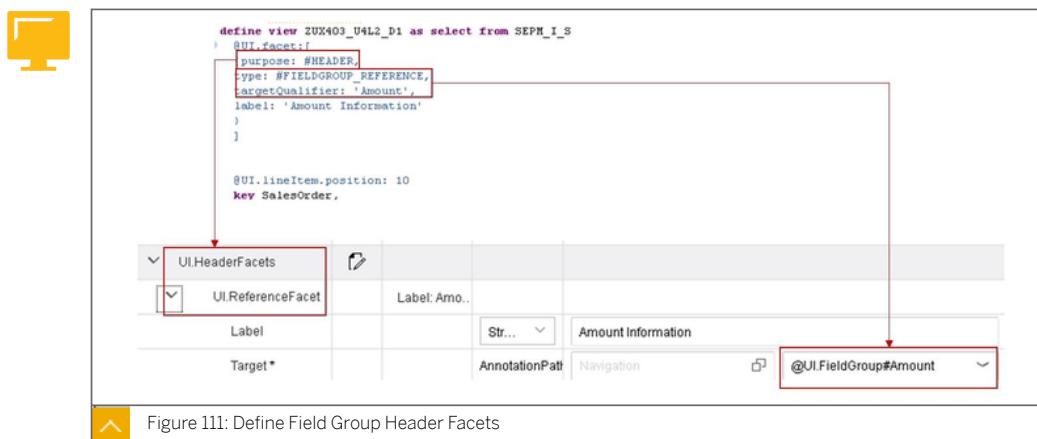
- Total Gross Amount
- Total Net Amount
- Status Information
- Overall Status
 - Confirmation Status
 - SO Ordering Status
 - SO Lifecycle Status

Define Field Groups

Term `UI.FieldGroup` is used to define field groups. Because in most cases a CDS view has more than 1 field groups, a qualifier should be assigned to each field group to distinguish among those field groups.

A field can belong to different field groups, so that the `UI.fieldGroup` annotation for each field is an array. The annotation should use JSON format for easier modification in the future.

When translating field groups annotation to OData annotation, all `@UI.fieldGroup` annotation have same qualifier will be combined to one term. The sequence of fields in a field group will be determined by `@UI.fieldGroup.position` defined in CDS view.

Define Field Group Header Facets

To define a field group header facet, a `@UI.facet` annotation should be added for the first field of your CDS view. The `@UI.facet` annotation is an array and should contain all facet definitions.

For Field Group Header Facet, it should:

- Set purpose to `#HEADER`
- Set type to `#FIELDGROUP_REFERENCE`
- Set targetQualifier to the qualifier of corresponding field group
- Set position and label

The translated OData annotation will have a term called `UI.HeaderFacet` and with some `UI.ReferenceFacet` in it.

A more common way is to define `UI.fieldGroup` in CDS view since field group is data intensive information but define `UI.HeaderFacet` annotation as Local Annotation while facets are more relevant with UI layer. Another reason for this is `@UI.facet` annotation in CDS view is only supported by NetWeaver AS ABAP 7.52.

Defining a Plain Text Header Facet

Figure 112: Defining a Plain Text Header Facet

The Plain Text Header Facet is quite like field group header facet.

To define a plain text header facet, you should:

- Define a field group, containing only one field
- Set the field as a multiline text

Definition of the Header Facet Communication

Figure 113: Define Communication Header Facet

Technically a communication header facet is a Field Group header facet. The only difference is at least one field of the field group should reference to a *vCard.Contact* annotation.

Currently, communication header facet can only be created by add local annotation through annotation modeler in SAP WebIDE.

The steps are:

Add a *vCard.Contact* term, bind properties to corresponding fields. Mention that for properties like phone, email, a type property is needed to clarify what is the usage of the phone number, or email address(ex. Home, work).

Add a *UI.FieldGroup* term with a *DataFieldForAnnotation* item, reference to the *vCard.Contact* annotation.

Add a *UI.HeaderFacet* term then add a *ReferenceFacet* in it. Reference the facet to the field group.

Arrangement of Header Facets

Header Facet 1 Header Facet 2 Header Facet 3 Header Facet 4 Header Facet 5 Header Facet 6 Header Facet 7
 Net Amount: 21,737.00 Net Amount: 21,737.00
 Gross Amount: 25,867.03 Gross Amount: 25,867.03
 : 4,130.03 : 4,130.03 : 4,130.03 : 4,130.03 : 4,130.03 : 4,130.03
 Header Facet 8
 Net Amount: 21,737.00
 Gross Amount: 25,867.03
 : 4,130.03

**Header Facets will arrange in a row.
When the space is not enough, it will start a new row**

Figure 114: Arrangement of Header Facets

If there is more than one Header Facet, they will arrange in row and will start a new row if there is not enough room for it. Here is an example of the arrangement for 8 Header Facets.



LESSON SUMMARY

You should now be able to:

- Use Header Facets for Object Pages

Unit 4 Lesson 3

Using Sections and Facets in Object Pages



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Use Sections and Facets in Object Pages

Sections and Facets in Object Pages



The areas of the Object Page below the header facets are some facets, grouped by sections. There is also a navigation bar to help user navigation to specific section.

Types of Facets in Sections

The screenshot displays a SAP Fiori application interface. At the top left is a navigation bar with 'GENERAL INFORMITICS' and 'Contacts'. Below it is a contact list with entries for Sally Spring, Maria Hicks, and Paul Burke. A 'Contact Details' card is open for Sally Spring, showing her name, email (sally.spring@erp.com), phone number (+1234 567890), and address. To the right of the contact list is a 'Product Description' box containing a detailed text about a Notebook Professional 17. Below the contact list are four facets: 'Contact Facet' (a list of contacts), 'Plain Text Facet' (a table of general and technical data), 'Smart Chart Facet' (a bar chart showing values for different categories), and 'Field Group Facet' (a list of field groups).

Figure 116: Types of Facets in sections

The following are further explanations about types of facets:

Contact Facet

Displays contacts information from a 1 to * association.

Plain Text Facet

Displays a long description.

Smart Chart Facet

Displays a chart. The data usually come from a 1 to * association.

Field Group Facet

Display field groups in a section.



Note:

Detailed explanations for smart chart facet is given in a later lesson of this course.

Detailed information about Section, Indicator Facet and Data Point Facet will be given in later chapter of this course.

For a full list of types of header facets, open the SAPUI5 SDK and navigate to following path: *Documentation* → *Developing Apps with SAP Fiori Elements* → *List Report and Object Page* → *Configuring Object Page Features* → *Defining and Adapting Sections*.

Unit 4: Object Page

Field Group Facets-CDS annotation

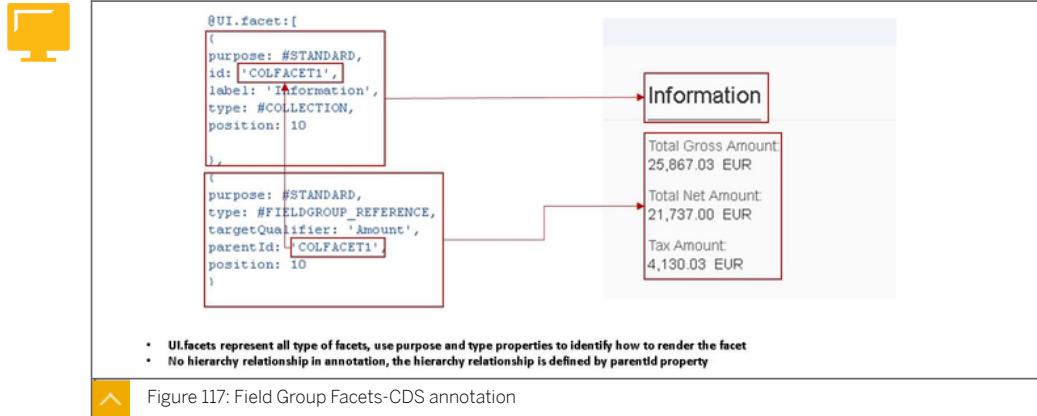


Figure 117: Field Group Facets-CDS annotation

A Field group facet must be put into a section.

To create a section, use `UI.facet` annotation to create a facet with type `#COLLECTION` and assign it an id.

To create a field group facet, add a second record in `UI.facet` with type `@FIELDGROUP_REFERENCE` and set `targetQualifier` to **qualifier of the field group**. Then reference the reference to the collection facet by `parentId`.



Note:

Define facets in CDS view is only supported by ABAP 7.52.

Field Group Facets-Local Annotation

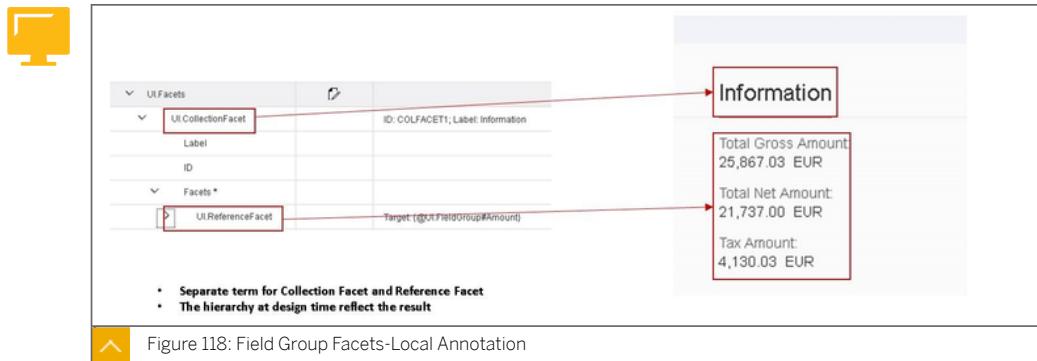


Figure 118: Field Group Facets-Local Annotation

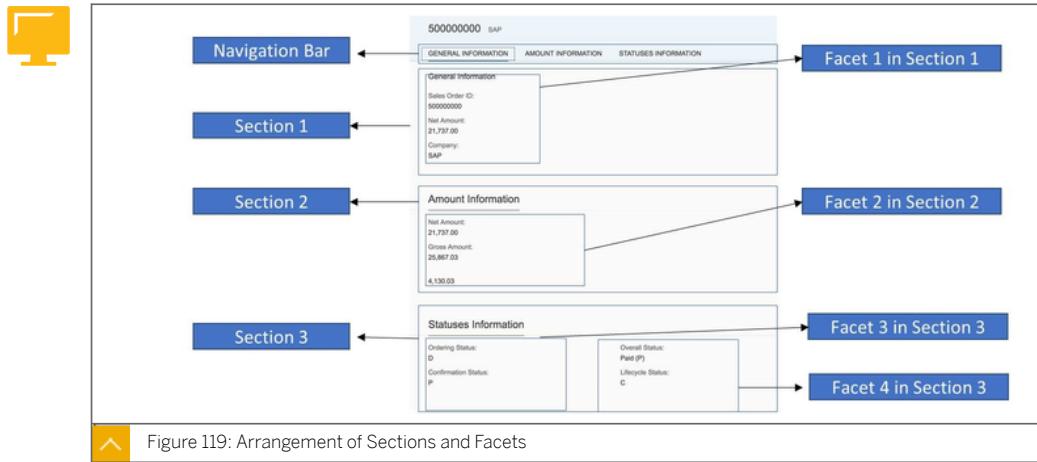
Back end systems do not support the definition of facets on backend. Local annotation can be used to define facets.

There are different terms for collection facet and reference facet. The steps are:

1. Create a UI.CollectionFacet

2. Create a UI.ReferenceFacet under the collection facet and reference it to field group annotation

Arrangement of sections and facets



All sections arrange vertically.

All facets in a section arrange in row.

Display Data from Sub-Entity

The screenshot shows a table titled "Items" with columns: Item Position, Product ID, and Total Net Amount. The table lists 100 items with their respective product IDs and total net amounts. A note at the bottom states: "For facet display as a table, use Reference Facet directly as a section".

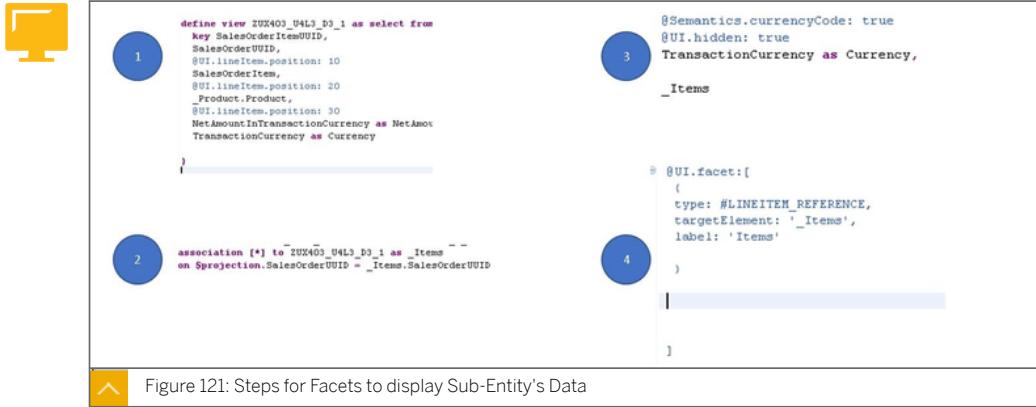
Item Position	Product ID	Total Net Amount
10	HT-1000	956.00 EUR
20	HT-1001	2,496.00 EUR
30	HT-1002	3,140.00 USD
40	HT-1003	3,300.00 EUR
50	HT-1007	897.00 USD
60	HT-1010	3,998.00 EUR
70	HT-1011	459,800 JPY
80	HT-1020	460.00 EUR
90	HT-1031	855.00 GBP
100	HT-1032	1,035.00 EUR

Figure 120: Display Data from Sub-Entity

In case you need to display data from a 1 to * association as a table in object page, you should use a reference facet to create a section, since the section can only contain one facet.

Unit 4: Object Page

Steps for Facets to display Sub-Entity's Data



Steps for facets to display sub-entity's data are:

1. Create a CDS view for detailed data. (e.g.. Sales order items data)
2. In CDS view for the list report, create a to * association to the CDS view created in step 1
3. Expose the association by write it's name just like a field
4. Create a facet with type #LINEITEM_REFERENCE and link it to the association by setting the *targetElement* property

Display of Sub-Entity's Data - Local Annotation



For local annotation. You need to create *UI.ReferenceFacet* as a direct child of *UI.Facets*(No collection facet).

For the reference facet you need to set the target by referencing the association and annotation.

Note, that when exposing CDS view as an OData service, SAP added "to" before CDS association name as OData association name.

Default UI.Facet generated by WebIDE**Default UI.Facet generated by WebIDE are:**

- For CDS views, which do not contain any *UI.Facet* annotation, a default *UI.Facet* local annotation is created automatically when Fiori Elements application was created by the wizard in WebIDE.
- The default Facet contains a collection facet and a reference facet In it reference to *UI.identification*.
- IF *UI.Facet* annotation is added in CDS view after the SAPUI5 application was created, developer should delete the default local annotation manually, otherwise it will override *UI.Facet* in CDS view.

**LESSON SUMMARY**

You should now be able to:

- Use Sections and Facets in Object Pages

Unit 4

Learning Assessment

1. How to set a field as the title for a business entity?

Choose the correct answer.

- A Add @UI.title annotation for that field.
- B Add @UI.headerInfo.title annotation for that field.
- C Add @UI.headerInfo.title.value for the CDS view and reference it to the field.
- D Add @UI.headerInfo.title for the CDS view and reference it to the field.

2. Which types are supported as header facet for object page?

Choose the correct answers.

- A Field Group Facet
- B Plan Text Facet
- C Smart Chart Facet
- D Rating Indicator Facet

3. For adding a header facet, you should use @UI.HeaderFacet annotation and put it before define statement of your CDS view.

Determine whether this statement is true or false.

- True
- False

4. The CDS annotation @UI.facet is used to:

Choose the correct answers.

- A Create a Header Facet.
- B Create a collection facet, which is displayed as section.
- C Create a reference under collection facet.
- D Create content, like field groups, or charts for a Facet.

5. Which of the following properties are relevant to the type of facet?

Choose the correct answers.

- A purpose
- B targetQualifier
- C parentId
- D type

6. If your backend system is based on ABAP 7.50, which are available ways of using Facets?

Choose the correct answers.

- A Declare both: field groups and facets in CDS view.
- B Declare both: field group and facet as local annotation using WebIDE.
- C Declare field group in CDS view, declare facets as local annotation using WebIDE.
- D It is not possible to use facets in ABAP 7.50.

Unit 4

Learning Assessment - Answers

1. How to set a field as the title for a business entity?

Choose the correct answer.

- A Add @UI.title annotation for that field.
- B Add @UI.headerInfo.title annotation for that field.
- C Add @UI.headerInfo.title.value for the CDS view and reference it to the field.
- D Add @UI.headerInfo.title for the CDS view and reference it to the field.

2. Which types are supported as header facet for object page?

Choose the correct answers.

- A Field Group Facet
- B Plan Text Facet
- C Smart Chart Facet
- D Rating Indicator Facet

3. For adding a header facet, you should use @UI.HeaderFacet annotation and put it before define statement of your CDS view.

Determine whether this statement is true or false.

- True
- False

4. The CDS annotation @UI.facet is used to:

Choose the correct answers.

- A Create a Header Facet.
- B Create a collection facet, which is displayed as section.
- C Create a reference under collection facet.
- D Create content, like field groups, or charts for a Facet.

5. Which of the following properties are relevant to the type of facet?

Choose the correct answers.

- A purpose
- B targetQualifier
- C parentId
- D type

6. If your backend system is based on ABAP 7.50, which are available ways of using Facets?

Choose the correct answers.

- A Declare both: field groups and facets in CDS view.
- B Declare both: field group and facet as local annotation using WebIDE.
- C Declare field group in CDS view, declare facets as local annotation using WebIDE.
- D It is not possible to use facets in ABAP 7.50.



UNIT 5

Advanced Topics of List Report and Object Page

Lesson 1

Explaining Navigation Concept and Annotations	119
---	-----

Lesson 2

Using Data Visualization	124
--------------------------	-----

Lesson 3

Creating Charts	127
-----------------	-----

Lesson 4

Performing CURD operations with BOPF	130
--------------------------------------	-----

UNIT OBJECTIVES

- Explain the navigation concept and annotations
- Describe options of external navigation
- Use data visualization
- Create Charts
- Perform CURD operations with BOPF

Unit 5

Lesson 1

Explaining Navigation Concept and Annotations



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain the navigation concept and annotations
- Describe options of external navigation

Navigation Concept and Annotations



- Internal navigation is the navigation with a Fiori Elements App
- The default internal navigation mode is List Report to Object Page
- Internal navigation is configured in manifest.json

The screenshot shows a Fiori Elements app interface with a list report on the left and an object page on the right. A red arrow points from the list report to the manifest.json code on the right, highlighting the connection between the visual representation and the configuration file.

```

    "sap.ui.generic.app": {
      "_version": "1.3.0",
      "pages": [
        "ListReport|ZUX400_USL1_01": {
          "entitySet": "ZUX400_USL1_01",
          "component": {
            "name": "sap.suite.ui.generic.template.ListReport",
            "list": true
          }
        },
        "pages": [
          "ObjectPage|ZUX400_USL1_01": {
            "entitySet": "ZUX400_USL1_01",
            "component": {
              "name": "sap.suite.ui.generic.template.ObjectPage"
            }
          }
        ]
      }
    }
  
```

Figure 123: Internal Navigation-default mode

In the *manifest.json* file, you define which pages are available in the app. At the top level, an array "pages" is defined. This array should have only one point of entry, which is the main entry point to the app and should always be a list report. Each page can have an array "pages" containing all sub-pages of the given page. Only one sub-page is allowed in the list report- this should be an object page for the same entity set. By default, the object page does not contain any sub pages.

Internal Navigation-List Report Only

The screenshot shows the SAP Fiori List Report Only configuration interface. It includes a table of business partners and their details, and a JSON code snippet for the ListReport configuration.

Table Data:

Business Partner ID	Company Name	Bus. Part. Role
SAP (100000000)	SAP	01
Becker Berlin (1000000001)	Becker Berlin	02
DeBont Industries (100000002)	DeBont Industries	01
Talpa (100000003)	Talpa	01
Panorama Studios (100000004)	Panorama Studios	01
TECUM (100000005)	TECUM	01
Asia High tech (100000006)	Asia High tech	01
I am (100000007)	I am	01

JSON Configuration:

```

Standard {
  "sap.ui.generic.app": {
    "_version": "1.3.0",
    "pages": [
      "ListReport[ZUX403_051_01]: {
        "entitySet": "ZUX403_051_01",
        "component": {
          "name": "sap.suite.ui.generic.template.ListReport",
          "list": true
        }
      }"
    ]
  }
}
  
```

Annotations:

- A blue callout points to the right side of the configuration area with the text "No Navigation link".
- A blue callout points to the bottom left of the configuration area with the text "Sub pages for list report has been deleted".

Figure 124: Internal Navigation-List Report Only

If you want to display list report without object page, you can simply delete **pages** and all properties in it under *ListReport*→ <Your EntitySetName> .

Subpages for Object Page

The screenshot illustrates the addition of subpages to an object page. It shows three stages of configuration:

- List Report:** Shows a standard list report view.
- Object Page with a report as a section:** Shows the list report integrated as a section within an object page frame.
- Object Page as subpage:** Shows the list report as a separate subpage, indicated by a blue arrow pointing from the previous stage.

Figure 125: Adding a subpage for Object Page

You can add further page as sub page to your object page, that let your application can navigate into very detail level of information,

Adding a subpage for Object Page-manifest.json

```

    "sap.ui.generic.app": {
      "_version": "1.3.0",
      "pages": [
        "listReport|ZUX403_USL1_D1": {
          "entitySet": "ZUX403_USL1_D1",
          "component": {
            "name": "sap.suite.ui.generic.template.ListReport",
            "list": true
          }
        },
        "pages": [
          "ObjectPage|ZUX403_USL1_D1": {
            "entitySet": "ZUX403_USL1_D1",
            "component": {
              "name": "sap.suite.ui.generic.template.ObjectPage"
            },
            "pages": [
              "ObjectPage|to_SalesOrders": {
                "navigationProperty": "to_SalesOrders",
                "entitySet": "ZUX403_USL1_D1_2",
                "component": {
                  "name": "sap.suite.ui.generic.template.ObjectPage"
                }
              }
            ]
          }
        ]
      ]
    }
  
```

Figure 126: Adding a subpage for Object Page-manifest.json

Under the node of *ObjectPage*, you can add a *pages* sub node further and add sub page for each OData entity set.

Options of External Navigation

Sales Order ID	Company Name	Country Name	Tax Amount
500000000	SAP	SAP	4,130.03 EUR >
500000001	DelBont Industries	DelBont Industries	2,331.49 EUR >
500000002	TECUM	TECUM	899.08 EUR >
500000003	Asia High tech	Asia High tech	272.07 EUR >
500000004	Asia High tech	Asia High tech	121.54 EUR >
500000005	AVANTEL	AVANTEL	16,173.82 EUR >
500000006	Talpa	Talpa	40.03 EUR >

Figure 127: External Navigation

In the Fiori Elements app, you can navigate to other addresses by using the External Navigation.

This function works both on list report and object page.

When creating external navigation, you can choose from two alternatives:

- Using a URL: Navigates to any URL
- Using Semantic Object navigation

For the semantic object navigation, you can choose between:

- display as a link, or
- a button on the toolbar.

Unit 5: Advanced Topics of List Report and Object Page

Note:
For detailed information of semantic object and intent based navigation, please refer to UX100.

The screenshot shows a SAP Fiori List Report titled "Customers (20) | Standard". A table lists customer records with columns for Sales Order ID and Company Name. The "Company Name" column contains entries like "SAP", "DelBont Industries", and "TECUM". A red box highlights the "SAP" entry. An arrow points from this entry to a screenshot of a Google search results page for "SAP". The search results page shows a snippet for "SAP Software Solutions | Business Applications an" with a link to "https://www.sap.com/index.html". Below the search results, a privacy reminder from Google is displayed. The SAP Fiori interface includes a navigation bar with links for All, News, Images, Maps, Videos, and More, along with a search bar and a reminder message.

```

@UI.lineItem.type: #WITH_URL
@UI.lineItem.position:20
@UI.lineItem.url: 'url'
_Customer.CompanyName,
concat('http://google.com/search?q=',_Customer.CompanyName) as url,

```

Figure 128: External Navigation-Using a URL

To create a external navigation using a URL, you need to:

- Create a field represent as URL
- For the field you want to add a link, set the type to `#WITH_URL` and point to the URL field using the `url` property

The screenshot shows a SAP Fiori List Report titled "Sales Orders (20) | Standard". It displays a table with columns for Sales Order ID and Company Name. A red box highlights the "Company Name" column. Another red box highlights the "Navigation" button in the top right corner of the table header. A callout bubble points to this button with the text "Display as a button". A second callout bubble points to the "Company Name" column with the text "Display as a link". The table lists various companies including SAP, DelBont Industries, TECUM, and Asia High Tech. To the right of the table, a separate table shows "Tax Amount" for each row. Arrows point from the "Navigation" button and the "Company Name" column to their respective code snippets in the SAP Fiori interface.

```

@Consumption.semanticObject: 'Action'
@UI.lineItem:[{
position: 30,
type: #WITH_INTENT_BASED_NAVIGATION,
semanticObjectAction: 'toappnavsample2'
}]
_Customer.CompanyName as CompanyName2,

```

```

@Consumption.semanticObject: 'Action'
@UI.lineItem:[{
label: 'Navigation',
position: 40,
type: #FOR_INTENT_BASED_NAVIGATION,
semanticObjectAction: 'toappnavsample2'
}]
_Customer.CompanyName as CompanyName3,

```

Figure 129: External Navigation-Semantic Object

For navigation to semantic object, you need to:

- Add a `@consumption.semanticObject` before the field
- Set `UI.lineItem.type` to `#WITH_INTENT_BASED_NAVIGATION` (as a link) or `#FOR_INTENT_BASED_NAVIGATION` (as a button)
- Set `@UI.lineItem.semanticObjectAction`

The generated URL will contain the semantic object and action, all values in the entity will pass as parameters.

Here is an example:

`https://.....#Product-display?CompanyName=SAP&CompanyName2=SAP&Currency=EUR&NetAmount=21737.00&SalesOrder=500000000&TaxAmount=4130.03`



LESSON SUMMARY

You should now be able to:

- Explain the navigation concept and annotations
- Describe options of external navigation

Unit 5 Lesson 2

Using Data Visualization



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Use data visualization

Data Visualization



Figure 130: Ways of Visualization

There are several methods of visualizing your data in list report and object page.

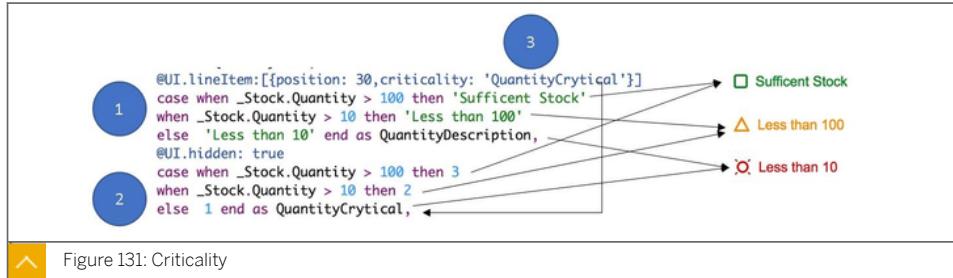
To display a value in different state (Neutral, Negative, Critical, Positive) with different colors, you can use the concept of criticality to visualize it.

For data representing a rating or progress, rating indicator and progress bar can be used to display them in a more vivid way.

For header of an Object Page, you can add an area micro chart or a bullet micro chart as a header facet to display relationship among actual value, target value, predictive value, max value and min value.

In the section area of object page, you can add a smart chart as a section to get insight from an entity set with 0..* association.

Criticality



Criticality can be used in any data field displayed on list report and object page. It also works with other annotations with DataField as datatype like UI.fieldGroup and UI.identification.

If you are using local annotation, criticality is a optional attribute for DataField, you can either set it to a static value or reference it to a property.

Steps to use criticality in list report by writing CDS view are:

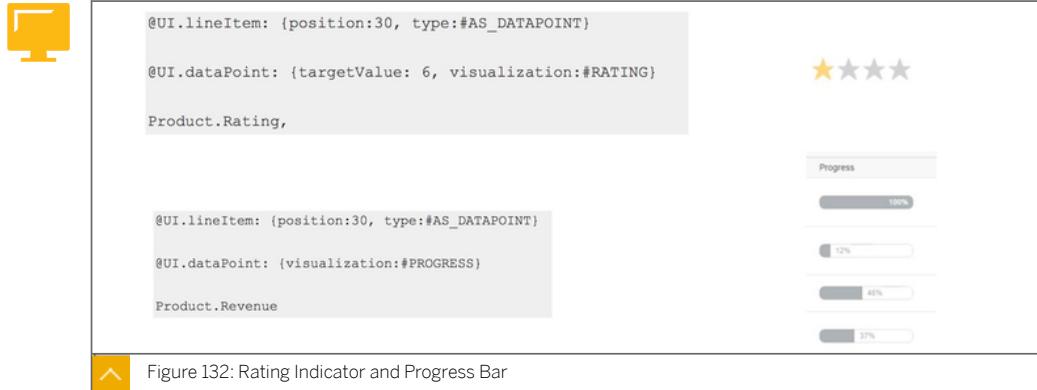
1. You have a Data Field to display some text as Critically Description Field, the output may be different for each criticality status.
2. Create another integer Data Field as Critically Value Field having values between 0 and 3, the meanings of values are.
 - 0 - Neutral
 - 1 - Negative
 - 2 - Critical
 - 3 - Positive
 - Normally, this field is calculated based on quantity or status fields
 - For preventing user display the number directly, the field should be set to hidden
3. Writing Annotation for Criticality Description Field have been created in a previous step. Point critical information to the Critically Value Field, using criticality property of @UI.lineitem.
 - Default colors for status: Neutral = Black, Negative = Red, Critical = Yellow, Positive = Green.



Note:

You can change these default colors by adjusting the theme you are using by Theme Designer. For detailed information for Theme Designer, refer to UX100.

Rating Indicator and Progress Bar



For rating and progress data, you can use rating indicator and progress bar.

To use those controls, set your @UI.lineItem (also apply to @UI.fieldGroup , @UI.identification and other annotation refer to a DataField) to the type of #AS_DATAPOINT. Then add a @UI.dataPoint annotation, set visualization property to @RATING or #PROGRESS.

By default, the rating indicator has 5 stars and progress bar set 100 as the target. You can customize those further by setting the value of the *targetValue* property.



LESSON SUMMARY

You should now be able to:

- Use data visualization

Unit 5 Lesson 3

Creating Charts



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Create Charts

Charts

OData Service with Analytic Function

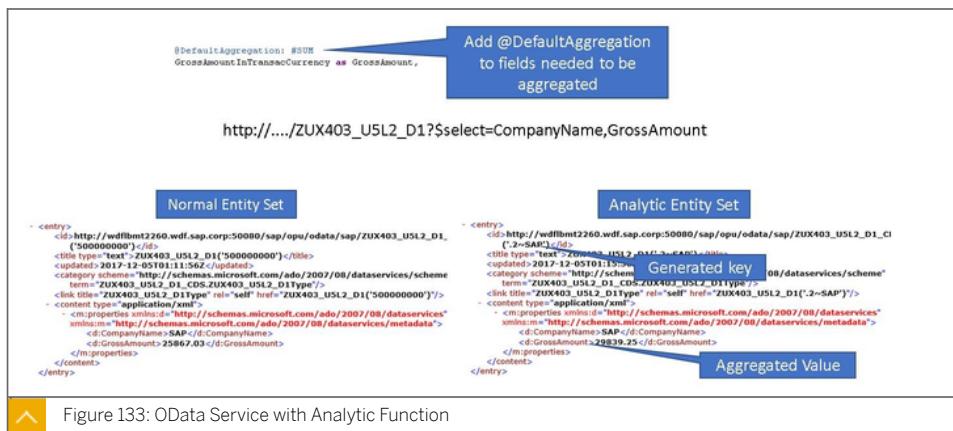


Figure 133: OData Service with Analytic Function

A smart chart control send an aggregate request to the backend service by sending the \$select OData parameter.

A normal entity set will return all records only with fields in \$select, but an analytic entity set will return an aggregated result considering the fields in \$select.

To define an analytic entity set, add a @defaultAggregation annotation to fields you needs to aggregate (normally are amount or quantity fields).

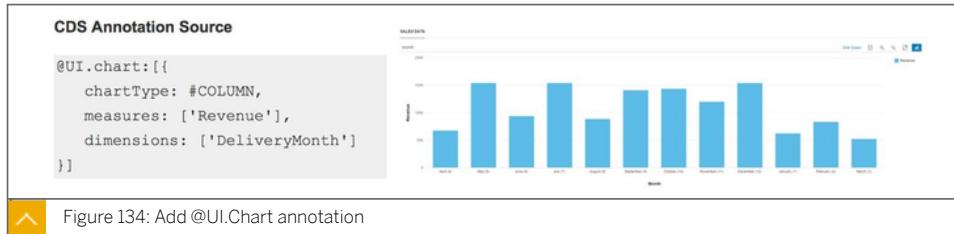
The generated entity set has the following differences compared to a normal entity set:

- A generated ID field is used as primary key instead of key of the CDS view.
- When accessed by \$select, the engine try to determine fields in \$select are attributes or measures according to @defaultAggregation , then caculate the aggregated measures grouped by attributes in \$select.

A typical scenario is define the CDS view for list report and object page as normal entity set and define the CDS view represent detailed level entry as analytic entity set. Then create a 1 to * association from master level entity to detail level entity. Then the user can get summarized detail data from object page.

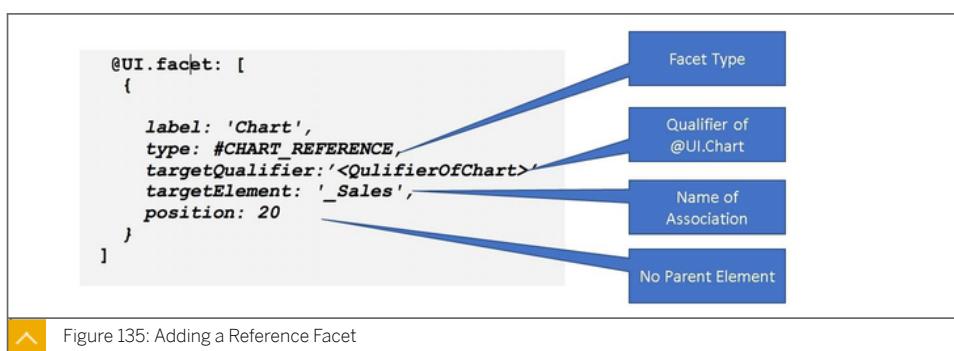
**Note:**

Another method of creating analytic OData service is Analytic Query, for detail information of Analytic Query, refer to course S4H410.

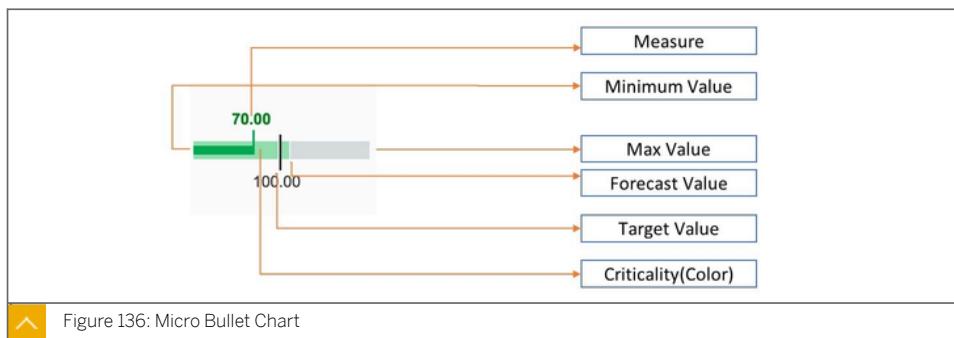
Add @UI.Chart annotation

A `@UI.Chart` annotation is used to define a smart chart. In the `chartType` property, you can specify the type of chart. Then you need to specify measures and dimensions.

The `@UI.Chart` term is also available in local annotation. Normally a `@UI.Chart` is defined in a analytic CDS view.

Adding a Reference Facet

A reference fact should be created as a direct child of `@UI.facet` with type `#CHART_REFERENCE` and reference to `@UI.Chart` annotation through an association.

Micro Bullet Chart

A Micro Bullet chart is used to evaluate a key figure by put its actual value, minimum value, max value, forecast value and target value together, then show the criticality by the color. It's useful to get insight among target, plan and actual and widely used in analytic scenarios.

The bullet chart can be used as header facet in object page.

Example of a Bullet Chart



```

@UI.chart: {
    title: 'Sales Revenue',
    description: 'Bullet Micro Chart',
    chartType: '#BULLET',
    measures: [ 'BulletChartRevenue' ],
    measureAttributes: [
        { measure: 'BulletChartRevenue', role: '#AXIS_1', asDataPoint: true }
    ]
}
@UI.dataPoint: {
    title: 'Sales Revenue',
    targetValueElement: 'TargetRevenue',
    forecastValue: 'ForecastRevenue',
    minValue: 100,
    maxValue: 300,
    criticality: 'Criticality'
}
ProductSalesRevenue.Revenue as BulletChartRevenue

```

Figure 137: Example of a Bullet Chart

Like smart charts, the creation of micro chart involves `@UI.facet` and `@UI.chart`.

In contrast to smart chart, a micro chart has more detailed insight to a point of data. So that a `@UI.dataPoint` is involved in the creation of micro chart as `@UI.chart` does not contain all detailed information for a point of data. The evaluated value should be the only member in the `@UI.chart.measure` and it must links to a `@UI.datapoint` annotation through `measureAttributes`. All other values (max,min,target,forecast ...) are assigned in `@UI.dataPoint`.



To Create a Chart

1. Create an OData entity set with analytic function.
2. Create `@UI.Chart` annotation on the analytic OData entity set.
3. Create a Reference Facet to display the Chart.



LESSON SUMMARY

You should now be able to:

- Create Charts

Unit 5

Lesson 4

Performing CURD operations with BOPF



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Perform CURD operations with BOPF

Business Object Processing Framework (BOPF)

Create and Delete Data with Fiori Elements



The screenshot illustrates the Fiori Elements interface for creating and deleting data. On the left, a list report titled "Sales Order for Transactional app" shows three sales orders. A blue box labeled "Create Button in List Report" points to the "New" button in the toolbar. An arrow points to the right, leading to a detailed "Sales Order" creation form. A blue box labeled "Render Object Page as a Create Form" highlights the "Save" button on the form.

Figure 138: Create and Delete Data with Fiori Elements

You can use the List Report and an Object to create a new object.

By adding a '+' button in the toolbar, you can navigate to the object page in which all fields are editable to create a new object.

You can also delete a record by choose the radio button on the left of the record and click Delete button on the toolbar.

Updating a Record

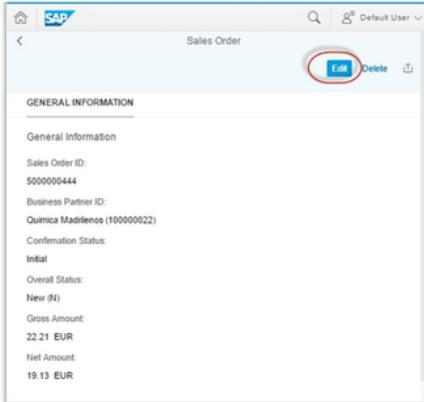


Figure 139: Update a Record using Fiori Elements

When you are in an object page, you can use the *Edit* button to switch between display mode and edit mode. You can also delete this record by using *Delete* button.

Draft Control

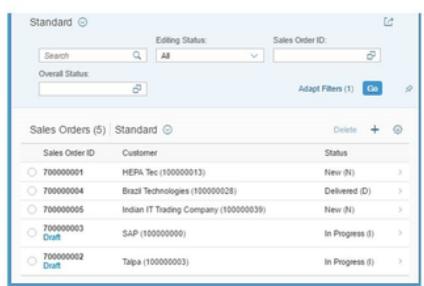
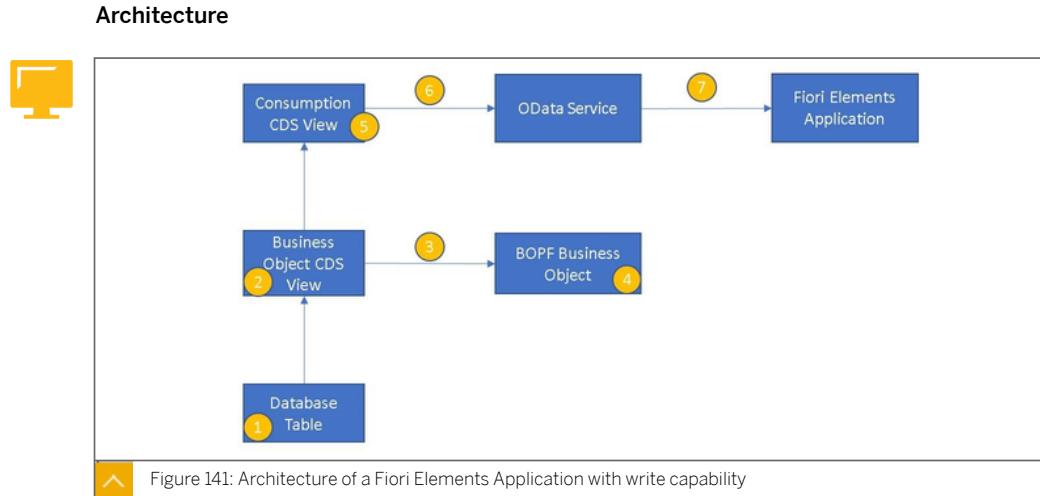


Figure 140: Draft Control when creating or changing a record

Fiori elements can also handle the status of a record.

For example, when User A want to update a record, a draft will be created for this record. If User B wants to update this record at the same time, he will be locked.



This is the architecture for Fiori Elements application with write capability:

1. Create a database table.

A database table is need to store records. The table must include structure "/bobf/s_lib_admin_data".

2. Create a business object CDS view.

Create a CDS view which select data from the table you've created and create associations to connect with relative business entities.

3. Generate a BOPF business object.

In BOPF, a business object is represented as a hierarchical tree of nodes. A single node includes a set of semantically related attributes and the corresponding business logic. For each node, several types of entities can be defined to describe the specific business logic part of the business object.

You can generate BOPF simply by adding annotations in Business Object CDS View.

4. Add additional business logic in BOPF.

Additional business logic written in ABAP code can be add to BOPF. For example, you can add validation code which can be executed before the record be saved in data base.

5. Create a consumption CDS view and write UI annotation on it.

Then you can start with your CDS view for the fiori elements application and writing UI annotation on it.

6. Generate OData Service

Perform this step as you've learned in this course.

7. Generate Fiori Elements application.

Perform this step as you've learned in this course.



Note:

For Detailed information of how to create transactional list report and object page.

Please proceed as follows:

Open URL https://help.sap.com/viewer/p/SAP_NETWEAVER_AS_ABAP_752,
then navigate to Development → Development Information → Application
Development on AS ABAP → SAP - ABAP Programming Model for SAP
Fiori → Develop → Developing New Transactional Apps



LESSON SUMMARY

You should now be able to:

- Perform CURD operations with BOPF

Unit 5

Learning Assessment

1. What are options for external navigation?

Choose the correct answers.

- A Navigation to URL (As a link).
- B Navigation to URL (As a button).
- C Navigation to Intent (As a link).
- D Navigation to Intent(As a button).

2. What can be done by editing *manifest.json*?

Choose the correct answers.

- A Disable object page.
- B Add object pages as sub page under an object page.
- C Define facets on Object Page.
- D Disable List Report.

3. The field indicate criticality should be hidden because it means nothing to end user.

Determine whether this statement is true or false.

- True
- False

4. Which steps are used for display a field as rating indicator?

Choose the correct answers.

- A Set type of line item to #AS_DATAPOINT .
- B Add a @UI.datapoint to the field and set visualization to #RATING .
- C Add a @UI.chart annotation to the field and set visualization to #RATING .
- D Set the visualization of line item to #RATING .

5. What are differences between analytical entity set and normal entity set from oData consumer perspective?

Choose the correct answers.

- A The Analytical entity set, generates a new field for primary key.
- B The Analytical entity set, returns all data, needed to analyze the client.
- C The Analytical entity set, returns summarized results according to the \$select parameter.
- D Analyzes the use of the database and runs faster when SAP HANA is used as database compared to a normal entity set.

6. Fiori Elements supports write operations, if the backend service is written by CDS working with BOPF.

Determine whether this statement is true or false.

- True
- False

Unit 5

Learning Assessment - Answers

1. What are options for external navigation?

Choose the correct answers.

- A Navigation to URL (As a link).
- B Navigation to URL (As a button).
- C Navigation to Intent (As a link).
- D Navigation to Intent(As a button).

2. What can be done by editing *manifest.json*?

Choose the correct answers.

- A Disable object page.
- B Add object pages as sub page under an object page.
- C Define facets on Object Page.
- D Disable List Report.

3. The field indicate criticality should be hidden because it means nothing to end user.

Determine whether this statement is true or false.

- True
- False

4. Which steps are used for display a field as rating indicator?

Choose the correct answers.

- A Set type of line item to #AS_DATAPOINT .
- B Add a @UI.datapoint to the field and set visualization to #RATING .
- C Add a @UI.chart annotation to the field and set visualization to #RATING .
- D Set the visualization of line item to #RATING .

5. What are differences between analytical entity set and normal entity set from oData consumer perspective?

Choose the correct answers.

- A The Analytical entity set, generates a new field for primary key.
- B The Analytical entity set, returns all data, needed to analyze the client.
- C The Analytical entity set, returns summarized results according to the \$select parameter.
- D Analyzes the use of the database and runs faster when SAP HANA is used as database compared to a normal entity set.

6. Fiori Elements supports write operations, if the backend service is written by CDS working with BOPF.

Determine whether this statement is true or false.

- True
- False

UNIT 6**Overview Page****Lesson 1**

Getting an Overview of the Overview Page (OVP)

139

UNIT OBJECTIVES

- Get an overview of the Overview Page (OVP)

Unit 6

Lesson 1

Getting an Overview of the Overview Page (OVP)



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Get an overview of the Overview Page (OVP)

Overview of the Overview Page (OVP)

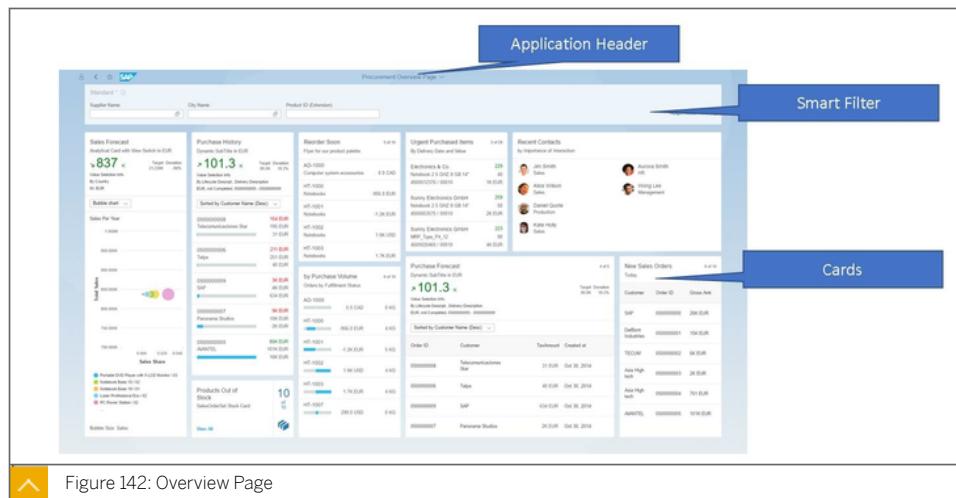


Figure 142: Overview Page

Overview pages provide quick access to vital business information at a glance, in the form of visual, actionable cards. The user-friendly experience makes viewing, filtering, and acting upon data quick and simple. While simultaneously presenting the big picture at a glance, business users can focus on the most important tasks enabling faster decision making as well as immediate action.

The application lets you create several cards for different types of content that helps in visualizing information in an attractive and efficient way. You can create overview pages and add cards to the page using the overview page wizard in SAP Web IDE.

The displayed data is fully interactive, with clickable areas for easy navigation to relevant applications. Based on SAP Fiori, overview pages organize action items with a fully responsive user interface. Users can access overview pages from SAP Fiori launchpad and narrow down the information displayed, or opt to hide cards to focus on a particular topic.

Main Components of the Overview Page Application

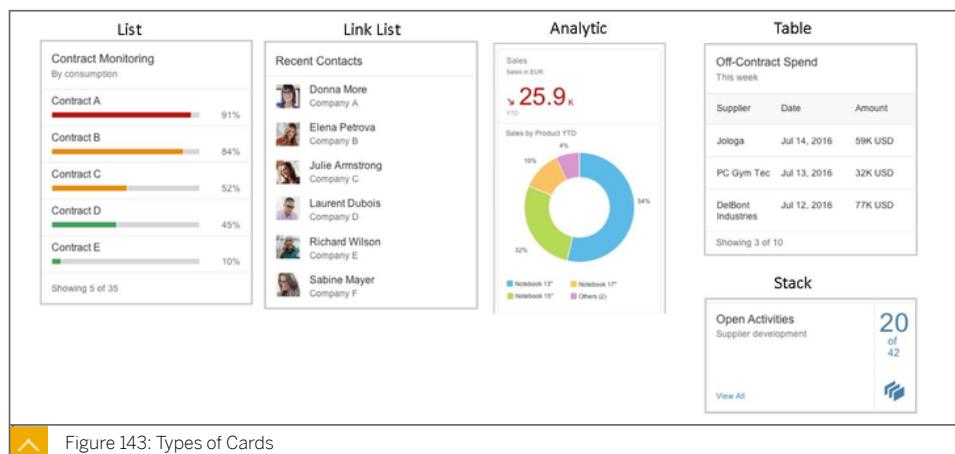
The overview page application contains the following main components:



- Application header: Provides a description of the area for which this application provides an overview (for example, procurement or sales). From the header area, users can change user account settings and manage cards (show/hide).
- Smart filter: Provides application-level filters for changing the levels of data displayed in the cards. For example, you could use the filter to display only transactions larger than \$10,000, only items lighter than 50kg, and so on.
- Cards: A card is a smart component that uses UI annotation to render its content. Each card is bound to a single entity set in a data source. A card may display a donut or bar chart, or a table. Stack cards contain a set of quick view cards which can be viewed in an object stream. Cards are displayed on the overview page in up to five responsive columns and can be rearranged by dragging and dropping.

Overview page application instances consist of a UI component that extends the overview page application component and a manifest file that contains the application configuration.

Types of Cards



The following are explanations about the various types of cards:

List Cards

List cards display lists of records according to the configuration in the com.sap.vocabularies.UI.v1.LineItem term. List cards display up to six fields of data in each list item.

Link List Cards

Allows you to view a list of links with title, picture, icon, or subtitle.

Analytic Cards

Analytical cards lets you view data in a variety of chart formats. The card is divided into two areas (header and chart).

Table Cards

A table card displays a list of records according to the configuration in the com.sap.vocabularies.UI.v1.LineItem term. A table card displays data in a 3-column table layout. (Optional) You can configure smart links in table cards to access quick links.

Stack Cards

Stack cards aggregate a set of cards of the same type, which are based on a common topic or action. When clicked, up to 20 stacked cards can be displayed in the object stream.



Note:

For more information of type of cards, please access website: <https://sapui5.netweaver.ondemand.com/sdk/#/topic> and navigate to *Developing Apps with SAP Fiori Elements* → *Overview Pages* → *Overview Page Card* → *Types of Cards*.

- 1
- 2
- 3

To Create an Overview Page

1. Prepare backend service for global filter.
2. Prepare backend service for cards.
3. Create Fiori application using Overview Page template.
4. Add cards for Overview Page.

For more details, refer to the following slides.

Overview Page Creation

1st Step



```

@UI.selectionField: [
    position: 10
]
@ObjectModel.foreignKey: {
    association: '_ProductCategory'
}

key ProductCategory,
@UI.selectionField: [
    position: 20
]
@ObjectModel.foreignKey: {
    association: '_ProductType'
}
key ProductType,
_ProductCategory,
_ProductType

```

All fields should be key field

All fields should be a selection field

Provide value help for all fields



Figure 144: 1. Prepare Backend Service for Global Filter

For most overview pages, there is an OData service supporting the global filter on the top of the page.

The OData service for global filter should only contains attribute values and should expose as key fields. All fields should been exposed as an selection field. And also a value help created either by foreign key association or modeled value help should assigned to the selection field.

Unit 6: Overview Page

2nd Step

The screenshot shows a code editor with two main sections. On the left, a CDS view definition for 'C_SAP_UX403_OVF_01' includes annotations for cards: '@UI.lineItem' for SalesOrder and SalesOrderItem, and '@UI.identification' for Product, ProductCategory, and ProductType. A callout box labeled 'Required annotations by cards' points to these annotations. On the right, another CDS view definition for 'C_SAP_IX403_OVF_03' includes annotations for a global filter: '@UI.lineItem' for SalesOrder and SalesOrderItem, and '@UI.filter' for SalesOrder, SalesOrderItem, Product, ProductCategory, and ProductType. A callout box labeled 'Identical field name with global filter' points to these annotations. The code uses SAP's Fiori conventions for annotations.

Figure 145: 2. Prepare Backend Services for Cards

You can provide backend service in the same OData service of the global filter or in standalone OData service.

Generally, for global filter and each card in overview page, you need a group of cds views (entity sets) to serve the main query as well as value help or other associated data access. Cards in an overview page usually involves lots of different base cds views. So in most cases we create separate OData service for each card, unless for cards have same backend logic.

Different type of card requires different annotations on CDS view. For example, table cards require @UI.lineItem, analytic cards require @UI.Chart and @UI.identification. For detailed information please refer to detailed information for card types.

Cards and global filter connected by fields. So If you want to filter data on card by setting global filter, the card need to have fields with identical name fields in global filter. If the names don't match, the filter will be ignored.

3rd Step

The screenshot shows the SAP Fiori application creation process. It starts with a 'Choose Template' step where 'Overview Page' is selected under the 'SAP Fiori Elements' category. An arrow points to the 'Set Global Filter' step, which is shown in a configuration dialog for 'New Overview Page Application'. The dialog has tabs for 'Template Selection', 'Basic Information', 'Data Connection', 'Annotation Selection', 'Template Customization', and 'Finalization'. In the 'Template Selection' tab, 'Enable Live Filter' is checked. The 'General' section contains fields for 'App Title' and 'App Description'.

Figure 146: 3. Create Fiori application using Overview Page template

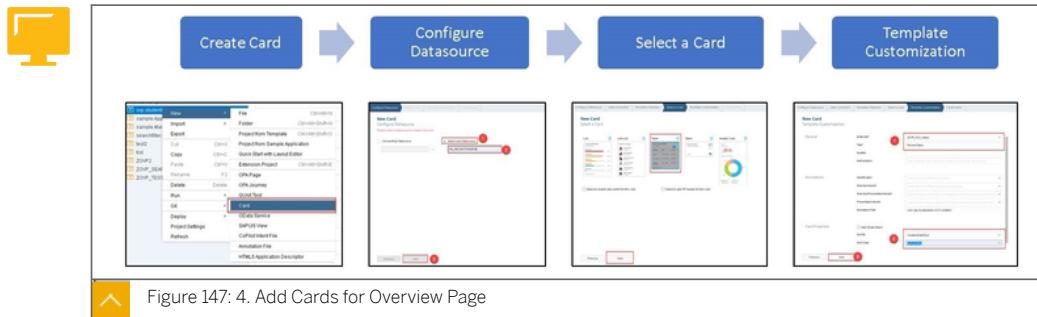
The process of creating an Overview Page is similar with list report.

There are only 2 different places:

1. When selecting template, choose Overview Page under category SAP Fiori Elements

2. In the *Template Customizing* step, you need to assign a entity set for global filter

4th Step



To add a card to overview page, you need to :

1. Add a card by choosing New –Card in the context menu of your overview page product.
2. In step configure data source, you can either select an existing OData service or add a new OData service. If you choose to add new service, some steps for select an OData service from backend system will be performed.
3. Choose a card type to add.
4. In the *template customization* step, you need to help the fiori application find the annotation it needs. Especially when annotations in CDS view have qualifier.



LESSON SUMMARY

You should now be able to:

- Get an overview of the Overview Page (OVP)

Unit 6

Learning Assessment

1. Which of the following types are types of cards in an Overview Page?

Choose the correct answers.

- A List Cards
- B Link List Cards
- C Analytic Cards
- D Table Cards
- E Stack Cards

2. In an overview page, all entity set for cards should have association with the entity set for global filter, otherwise the filter will not work on the card.

Determine whether this statement is true or false.

- True
- False

Unit 6

Learning Assessment - Answers

1. Which of the following types are types of cards in an Overview Page?

Choose the correct answers.

- A List Cards
- B Link List Cards
- C Analytic Cards
- D Table Cards
- E Stack Cards

2. In an overview page, all entity set for cards should have association with the entity set for global filter, otherwise the filter will not work on the card.

Determine whether this statement is true or false.

- True
- False

UNIT 7**Analytical List Page****Lesson 1**

Getting an Overview of the Analytical List Page

147

UNIT OBJECTIVES

- Get an overview of the Analytical List Page

Unit 7 Lesson 1

Getting an Overview of the Analytical List Page



LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Get an overview of the Analytical List Page

Overview of the Analytical List Page

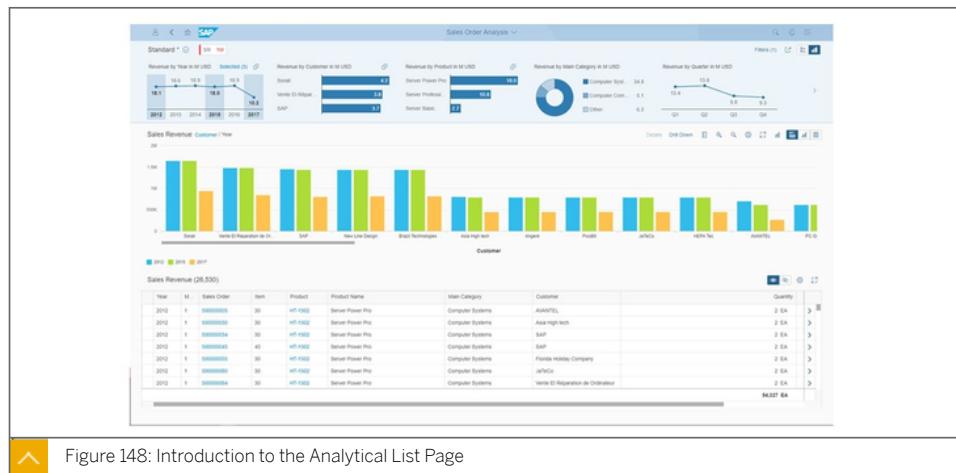


Figure 148: Introduction to the Analytical List Page

Analytical List Page (ALP) is a SAP Fiori elements application for detailed analytics.

It lets you analyze data from different perspectives, to investigate a root cause, and to act on transactional content. You can identify relevant areas within data sets or significant single instances using data visualization and business intelligence. All this can be done seamlessly within one page.

The combination of transactional and analytical data using chart and table visualization lets you to view relevant data more quickly. This hybrid view of data allows an interesting interplay between the chart and table representations.

Configure ALP to include the following use cases seamlessly within one page:

- Related KPIs (key performance indicators) on the header area as KPI tags. These KPI tags further allow a progressive disclosure and navigation through KPI cards.
- Filter data sets used for the main content area through different filter modes. For example, visual filters provide an intuitive way of choosing filter values from an associated measure value.
- Seamless navigation to applications from the content area and KPI card area.

Unit 7: Analytical List Page

- Customizing and sharing ALP as a page variant with other users.

Title Area

Figure 149: Title Area

In the header area you can view information related to the Key Performance Indicator (KPI) or choose any of the following built-in Fiori elements features to:

- Define or manage page variants
- Choose filter modes (compact or visual)
- Customize filter area
- Share analytical list page

Visual Filters

Figure 150: Visual Filters

Visual Filter is an intuitive way of choosing filter values from an associated measure value. It supports line, bar, and donut chart types.

Use visual filter to combine measures or item counts with filter values. Chart visualization increases the joy of use and the faster perception of relevant data. Selecting one or several chart data points allows quick analysis of the data set. For example, to choose the country with the highest sales, you can visualize the graphic and make a selection on it. The default view of visual filter bar is based on the filter fields defined in the *SelectionFields* annotation for which a visual filter is defined.

Content Area

Figure 151: Content Area

Lesson: Getting an Overview of the Analytical List Page

Users can interact with both the chart and the table. The initial view of the chart, visualize the most important aspects of the whole data set. Selecting a dimension within a chart area automatically filters all relevant information in the table area.

For example, if a chart selection is Country=ABC, then all records associated with this country selection are filtered in the table.

Note: For detailed information about the Analytical List Page, please access website: <https://sapui5.netweaver.ondemand.com/sdk/#/topic> and navigate to *Developing Apps with SAP Fiori Elements*→ *Analytical List Page*.



LESSON SUMMARY

You should now be able to:

- Get an overview of the Analytical List Page

Unit 7

Learning Assessment

1. Which areas are part of an Analytic List Page?

Choose the correct answers.

- A Title Area
- B Visual Filters
- C Selection Fields
- D Content Area

2. In an Analytic List Page, the user can filter data by clicking some points in the chart.

Determine whether this statement is true or false.

- True
- False

Unit 7

Learning Assessment - Answers

1. Which areas are part of an Analytic List Page?

Choose the correct answers.

- A Title Area
- B Visual Filters
- C Selection Fields
- D Content Area

2. In an Analytic List Page, the user can filter data by clicking some points in the chart.

Determine whether this statement is true or false.

- True
- False