UX410

Developing SAP Fiori UIs

PARTICIPANT HANDBOOK
INSTRUCTOR-LED TRAINING

Course Version: 19
Course Duration: 5 Day(s)
e-book Duration: 11 Hours 10 Minutes
Material Number: 50151674

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	Example text
Window title	Example text

Contents

Course Overview

TARGET AUDIENCE
This course is intended for the following audiences:

- Development Consultant
- Developer

UNIT 1

Introduction to User Experience (UX)

Lesson 1

UNIT OBJECTIVES

- Explain the differences between UX, usability, and UI

Unit 1
Lesson 1

Explaining UX versus Usability versus UI

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Explain the differences between UX, usability, and UI

Introduction to UX

The goldfish is ahead of man - at least as far as his attention is concerned. Thanks to modern media usage, our human attention span is now limited to eight seconds. That's a whole second less than the shiny fish with nine seconds.

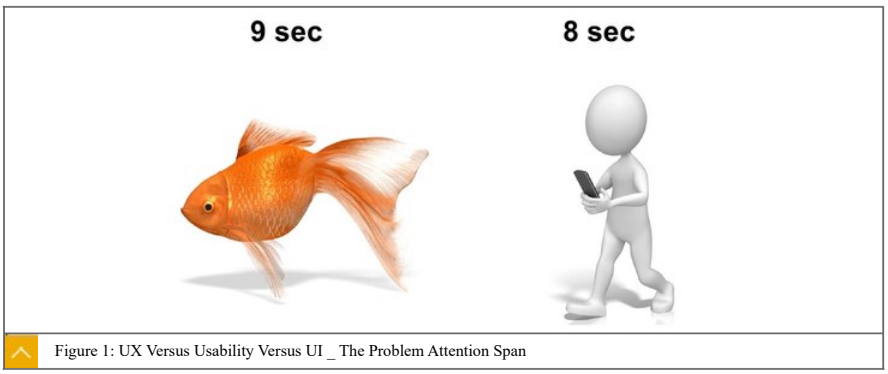


Figure 1: UX Versus Usability Versus UI _ The Problem Attention Span

A use interface is like a joke, if you have to explain it, it is not that good.

The international standard on *ergonomics of human system interaction*, ISO 9241-210, defines user experience as "a person's perceptions and responses that result from the use or anticipated use of a product, system or service". According to the ISO definition, user experience includes all the users' emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviors and accomplishments that occur before, during and after use. The ISO also list three factors that influence user experience: system, user and the context of use.

Figure 2: User Experience Definition

Definition of usability - ISO 9241: The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.

Effectiveness: the accuracy and completeness with which specified users can achieve specified goals in particular environments.

Efficiency: the resources expended in relation to the accuracy and completeness of goals achieved.

Satisfaction: the comfort and acceptability of the work system to its users and other people affected by its use.

The user interface, in the industrial design field of human–machine interaction, is the space where interactions between humans and machines occur. The goal of this interaction is to allow effective operation and control of the machine from the human end, whilst the machine simultaneously feeds back information that aids the operators' decision making process.

Figure 3: Ui Usability Definition

In the next unit, we discuss how to achieve a good user experience.

LESSON SUMMARY
You should now be able to:

- Explain the differences between UX, usability, and UI

Unit 1

Learning Assessment

1. What is described as the person's perceptions and response that result from the use or anticipated use of a product, system, or service?

   Choose the correct answer.

   - ☐ **A** User interface
   - ☐ **B** User experience
   - ☐ **C** Usability
   - ☐ **D** User acceptance

2. What does effectiveness mean in the relation to the user interface?

   Choose the correct answer.

   - ☐ **A** The resources expended in relation to the accuracy and completeness of goals achieved
   - ☐ **B** The accuracy and completeness with which specified users can achieve specified goals in specific environments
   - ☐ **C** The comfort and acceptability of the work system to its users and other people affected by its use

3. Which of the following can be used for usability assessment?

   Choose the correct answers.

   - ☐ **A** Heuristic evaluation
   - ☐ **B** User testing
   - ☐ **C** User observation
   - ☐ **D** Questionnaires

Unit 1

Learning Assessment - Answers

1. What is described as the person's perceptions and response that result from the use or anticipated use of a product, system, or service?

   Choose the correct answer.

   - [ ] **A** User interface
   - [X] **B** User experience
   - [ ] **C** Usability
   - [ ] **D** User acceptance

   You are correct!. User experience describes a person’s perceptions and response that result from the use or anticipated use of a product, system or service.

2. What does effectiveness mean in the relation to the user interface?

   Choose the correct answer.

   - [ ] **A** The resources expended in relation to the accuracy and completeness of goals achieved
   - [X] **B** The accuracy and completeness with which specified users can achieve specified goals in specific environments
   - [ ] **C** The comfort and acceptability of the work system to its users and other people affected by its use

   You are correct! Effectiveness describes the accuracy and completeness with which specified users can achieve specified goals in specific environments.

3. Which of the following can be used for usability assessment?

   Choose the correct answers.

   - [X] **A** Heuristic evaluation
   - [X] **B** User testing
   - [ ] **C** User observation
   - [X] **D** Questionnaires

   You are correct! To access the usability of a product you can use heuristic evaluation, user testing and questionnaires.

UNIT 2 SAP UX strategy

Lesson 1

Introducing the SAP UX Strategy 7

UNIT OBJECTIVES

- Describe the SAP UX strategy

Unit 2
Lesson 1

Introducing the SAP UX Strategy

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe the SAP UX strategy

SAP UX Strategy

Why is focusing on the user experience important?

Because having a great user experience has an impact on business value.

Many people believe user experience is simply colors and fonts. However we've learned throughout this presentation that is simply not true. If a great user experience is achieved, people gain productivity and can work faster and more efficiently, increase adoption, decrease errors, and save training costs. All of these are positive outcomes that immediately impact overall business value.

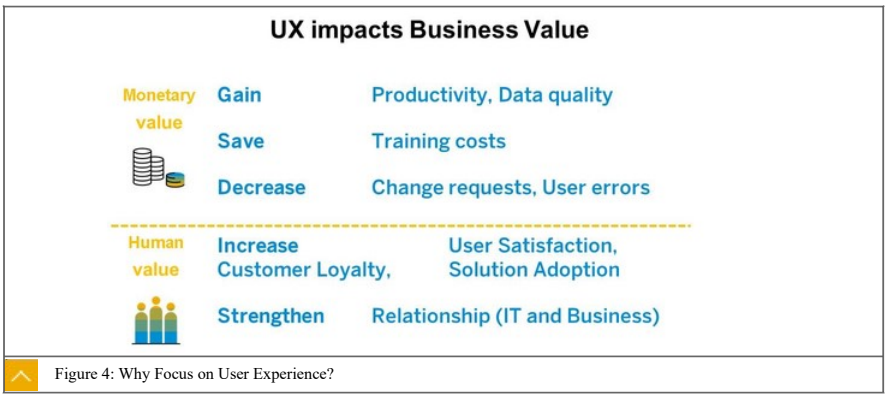


Figure 4: Why Focus on User Experience?

The way we experience the world is evolving.

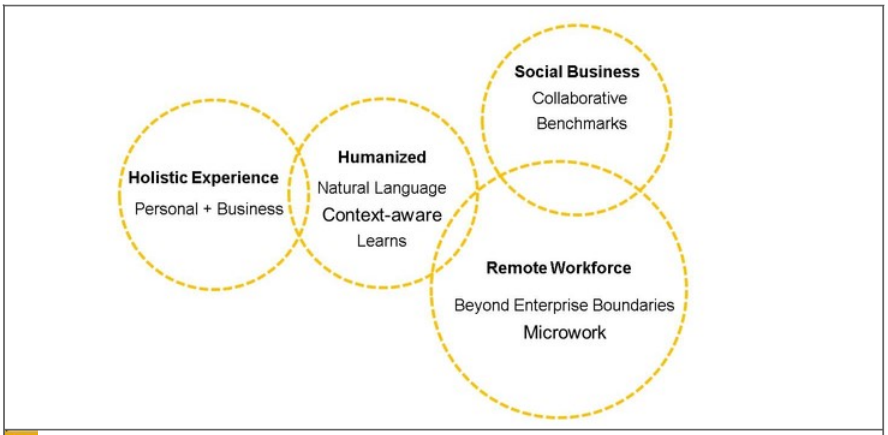


Figure 5: The Future Of UX

SAP has adopted a strategy to enhance user experience in an efficient manner.

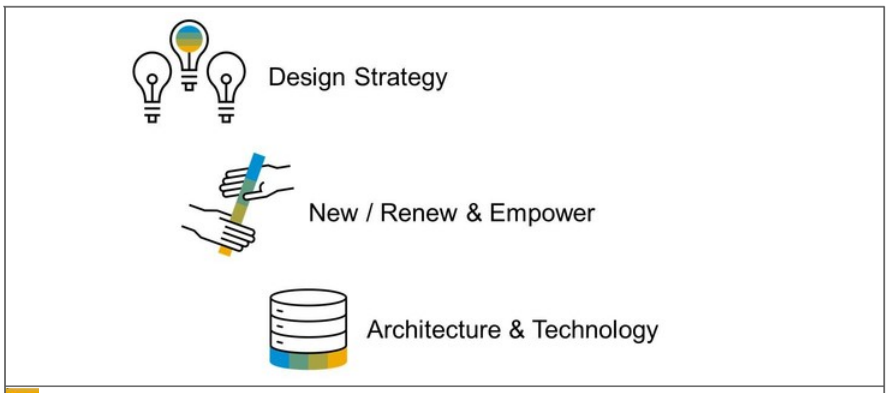


Figure 6: SAP UX Strategy

To implement the strategy SAP has developed tools and practices for this purpose.

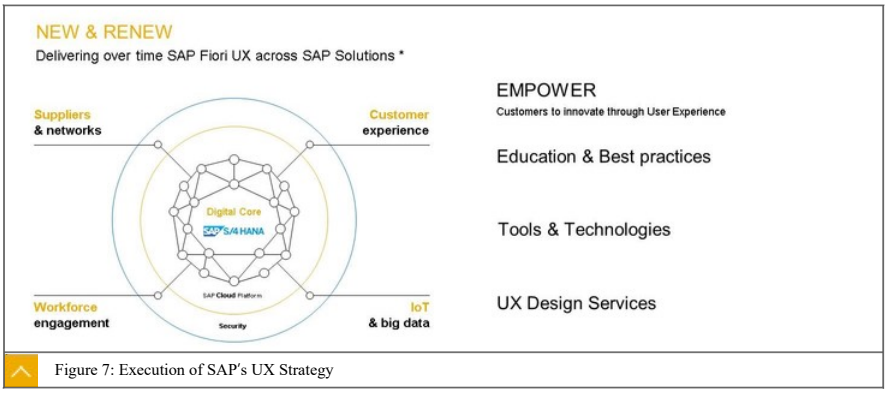


Figure 7: Execution of SAP's UX Strategy

It involves people, business, and technology.

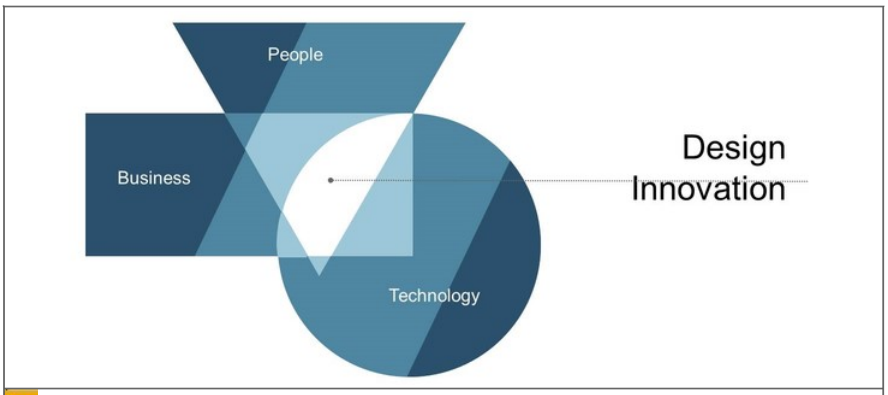


Figure 8: Design Innovation Mindset

SAP Fiori is the new platform to deliver an enhanced user experience.

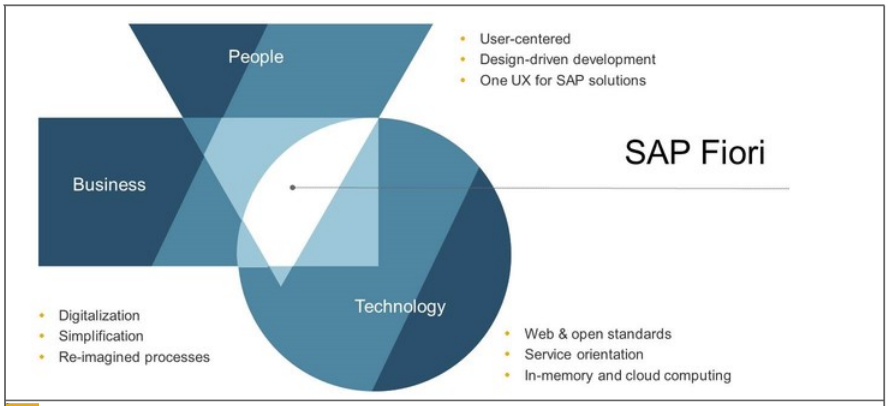


Figure 9: SAP's UX Direction-SAP Fiori

SAP Fiori keeps things simple.

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.

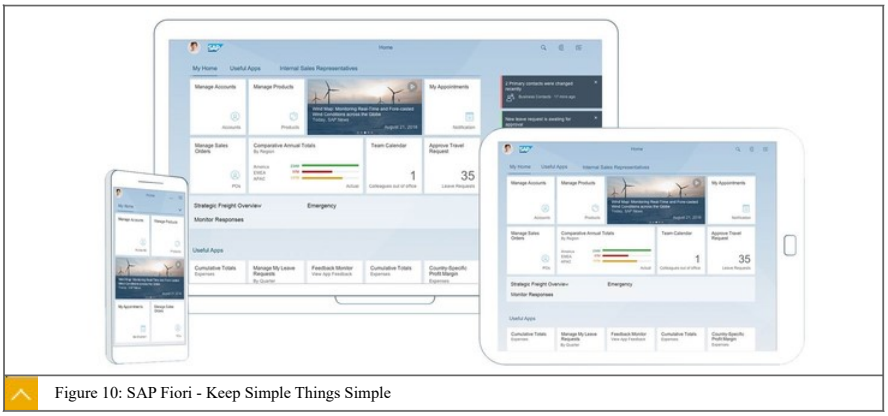


Figure 10: SAP Fiori - Keep Simple Things Simple

If you look at the image of the tablet, you will 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 do not fit at once. Responsive design is credited for automatically completing the look of our UI framework.

One important thing to note with SAP Fiori, is that it can be deployed in the customers existing landscape. SAP customers running ECC 6.0 or Suite on SAP Hana use a gateway to the back-end system and add-ons, making it an easy to deploy and use solution.

SAP Fiori introduces a new concept in consumer interaction.

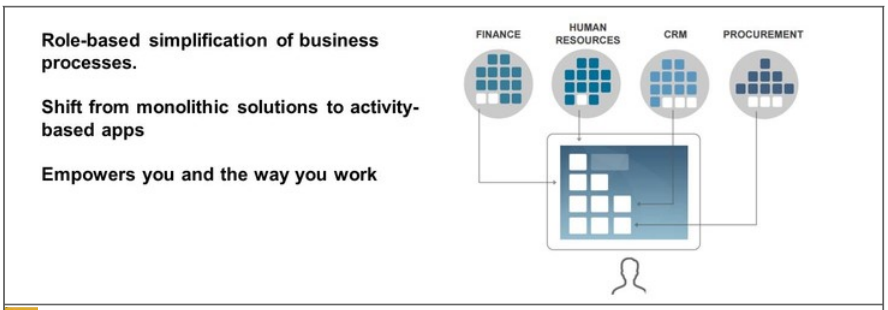


Figure 11: SAP Fiori Concept Review

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

SAP Fiori is 100% geared toward business users. To provide a consistent user experience. SAP Fiori ensures that people, both employees and managers, have a consistent, coherent, simple, and intuitive user experience across multiple devices. This allows everyone to work smarter, more efficiently, and deliver on business objectives.

Due to positive feedback from customers, this user experience is being applied across multiple platforms. Some aren't even SAP Fiori applications, but have that look and feel.

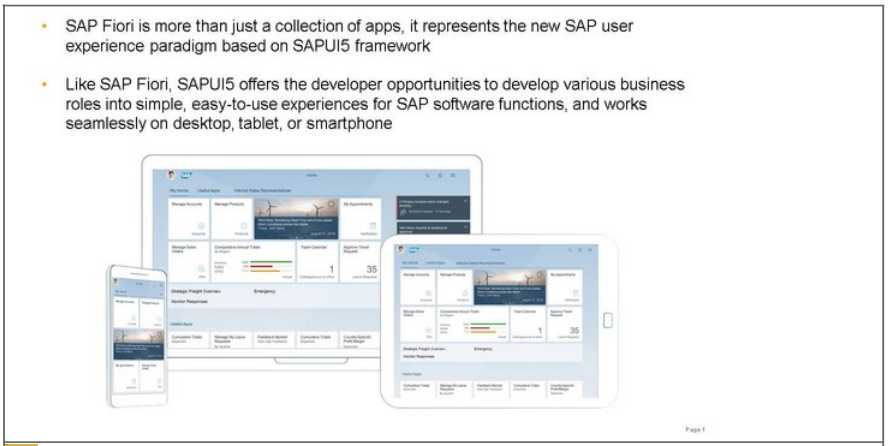


Figure 12: Why Design Apps Like SAP Fiori?

To understand how SAP Fiori is implemented, we review the concept.

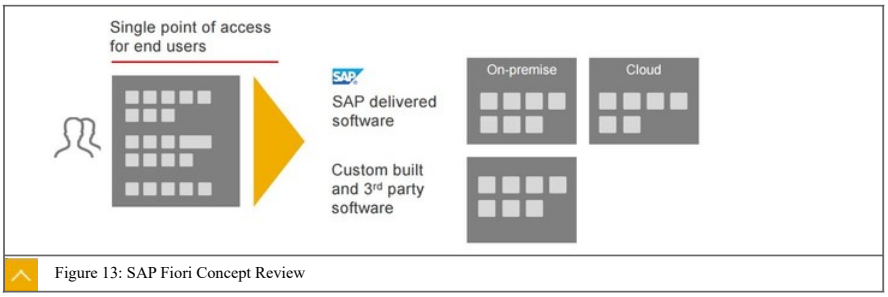


Figure 13: SAP Fiori Concept Review

Level 1

A full role scope is accessible from the SAP Fiori launchpad, but mainly with integration of legacy user interfaces.

Example: SAP Fiori launchpad, Search, ...

Level 2

Domain-specific information and action - All key business objects or domain areas

Example: List Reports, Overview Pages, Work Lists

Level 3

SAP Fiori UX for key use cases

Example: Object Pages, Edit Pages

Level 4

SAP Fiori Apps, web graphic user interfaces with SAP Screen Personas, WebDynpros, 3rd party user interfaces, partner user Interfaces

Example: SAP Fiori apps for all use cases, visually-harmonized classic user interfaces

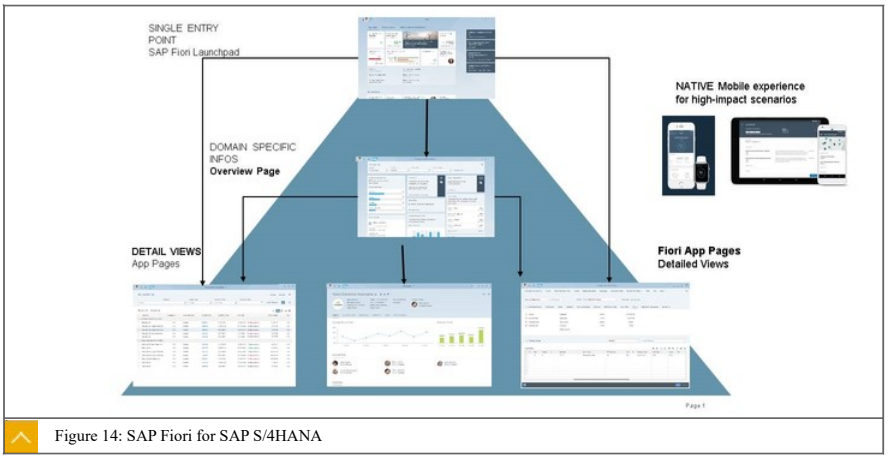


Figure 14: SAP Fiori for SAP S/4HANA

SAP offers a new and broad portfolio of UX tools and technologies that guide organizations into a user-centered design perspective. SAP helps define and execute the best UX strategy for business, using proven design methodologies like design thinking and user-centered design.

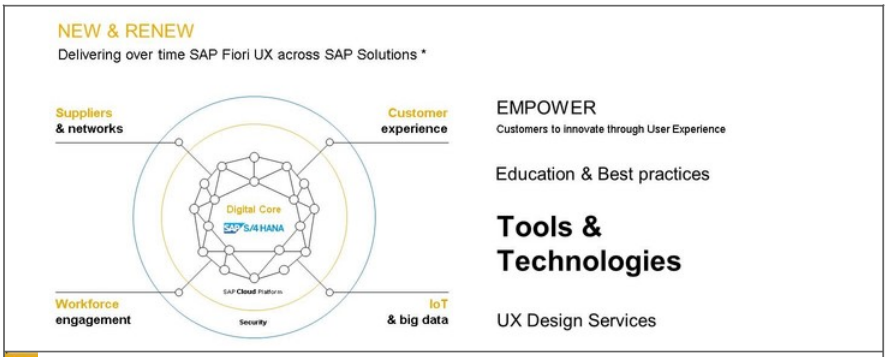


Figure 15: Execution of SAP's UX Strategy

User experience design can be adopted for various business scenarios. Design development can be accelerated using the SAP user experience design process.

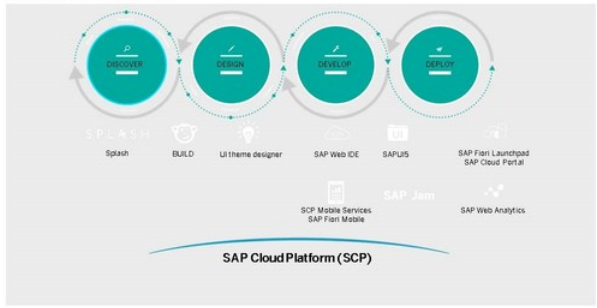


Figure 16: UX as a Service

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

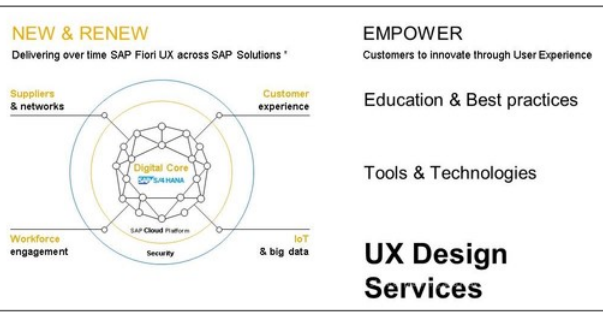


Figure 17: Execution of SAP's UX Strategy

SAP identified several leads from customers who asked for advice to understand the SAP strategy and translate it to their reality.

Customers requested services to realize: implement, adapt, and optimize the user experience of existing software.

When results are achieved with SAP Screen Personas, or SAP Fiori, the next level is to have customers build up skills on their own or empower their organization to have a user experience strategy.

The ultimate level is for the customer to become innovative. A goal for all customers is to be more innovative. They can achieve that by designing new products and looking for new services.

All of this is powered by design thinking.

In summary, the design service that SAP offers is to provide advice about the strategy and the benefits of a solid user experience.

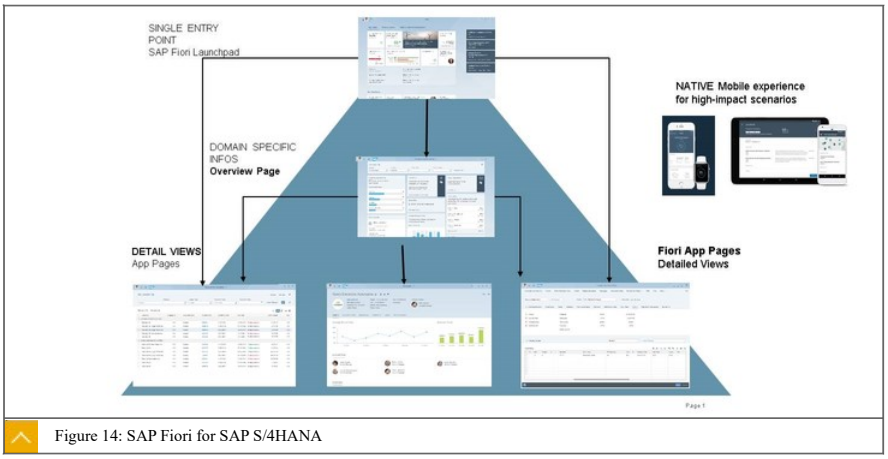


Figure 14: SAP Fiori for SAP S/4HANA

SAP offers a new and broad portfolio of UX tools and technologies that guide organizations into a user-centered design perspective. SAP helps define and execute the best UX strategy for business, using proven design methodologies like design thinking and user-centered design.

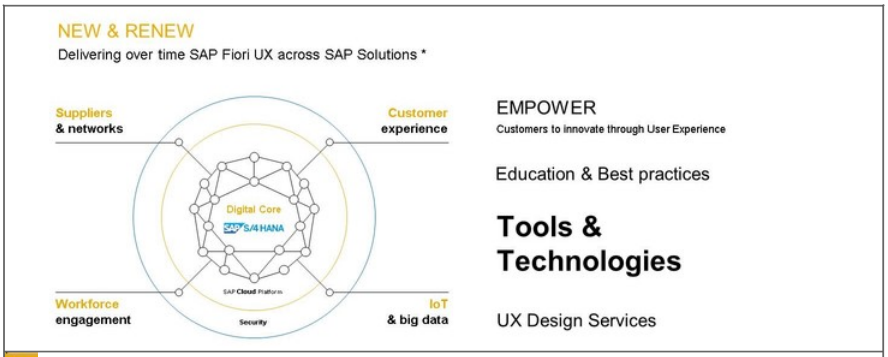


Figure 15: Execution of SAP's UX Strategy

User experience design can be adopted for various business scenarios. Design development can be accelerated using the SAP user experience design process.

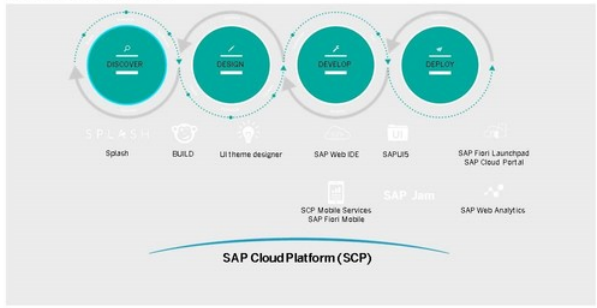


Figure 16: UX as a Service

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

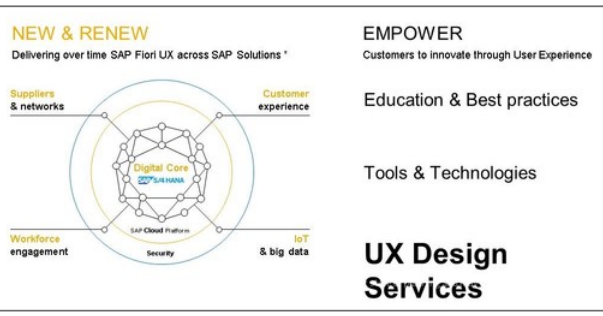


Figure 17: Execution of SAP's UX Strategy

SAP identified several leads from customers who asked for advice to understand the SAP strategy and translate it to their reality.

Customers requested services to realize: implement, adapt, and optimize the user experience of existing software.

When results are achieved with SAP Screen Personas, or SAP Fiori, the next level is to have customers build up skills on their own or empower their organization to have a user experience strategy.

The ultimate level is for the customer to become innovative. A goal for all customers is to be more innovative. They can achieve that by designing new products and looking for new services.

All of this is powered by design thinking.

In summary, the design service that SAP offers is to provide advice about the strategy and the benefits of a solid user experience.

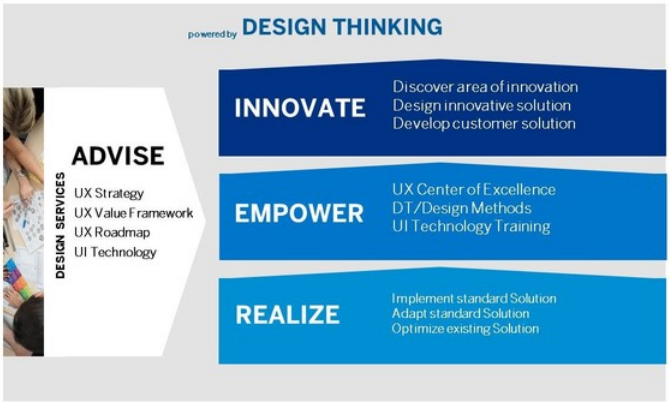


Figure 18: Overview of SAP's UX Design Services

LESSON SUMMARY

You should now be able to:

- Describe the SAP UX strategy

Unit 2

Learning Assessment

1. What impact does UX have on monetary values?

   Choose the correct answers.

   - [ ] **A** Increases user satisfaction
   - [ ] **B** Improves productivity and data quality
   - [ ] **C** Strengthens relationships with customers
   - [ ] **D** Saves training costs
   - [ ] **E** Decreases change requests and user errors

2. What are the key aspects of SAP's 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 and open standards
   - [ ] **D** User centered
   - [ ] **E** Re-imagines processes

Unit 2

Learning Assessment - Answers

1. What impact does UX have on monetary values?

   Choose the correct answers.

	A Increases user satisfaction
X	**B** Improves productivity and data quality
	C Strengthens relationships with customers
X	**D** Saves training costs
X	**E** Decreases change requests and user errors

   You are correct! Good UX improves productivity and data quality, it saves training costs and decreases the amount of change requests and user errors.

2. What are the key aspects of SAP's UX strategy?

   Choose the correct answers.

X	**A** Design strategy
	B New / Renew / Enablement
X	**C** New / Renew / Empower
X	**D** Architecture and Technology
	E SAP Screen Personas

   You are correct! The three key aspects of the SAP's UX strategy are to employ a design strategy, using architecture and technology, for new, renew and empower development of the client interface.

3. What impact does SAP Fiori have on business?

   Choose the correct answers.

- [X] **A** Digitalization
- [X] **B** Simplification
- [ ] **C** Web and open standards
- [ ] **D** User centered
- [X] **E** Re-imagines processes

You are correct! SAP Fiori leads us to re-imagine the processes, provides simplification of applications, and supports digitalization in the company.

UNIT 3 SAP Fiori UX

Lesson 1

Explaining the SAP Fiori UX 19

UNIT OBJECTIVES

- Explain the SAP Fiori UX

Unit 3
Lesson 1

Explaining the SAP Fiori UX

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Explain the SAP Fiori UX

Introduction to SAP Fiori UX

What is SAP Fiori UX?

Figure 19: What is SAP Fiori UX?

SAP Fiori UX is a paradigm shift that provides coherence across functional areas.

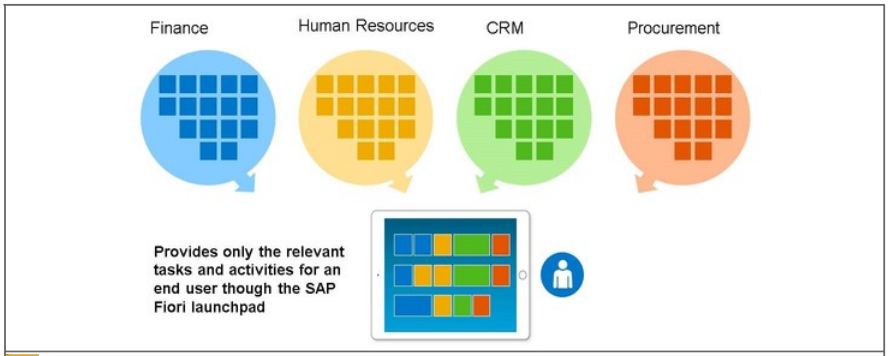


Figure 20: Coherence Across Functional Areas

It can be designed to suit specific business environments, specific roles, and the way people work.

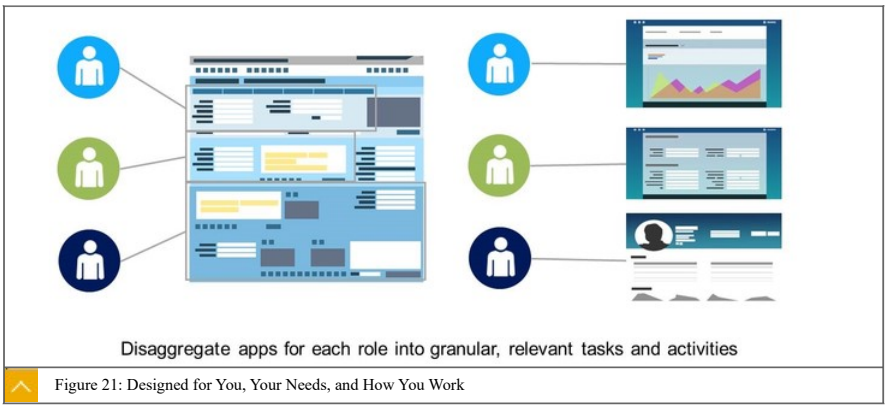


Figure 21: Designed for You, Your Needs, and How You Work

It provides a good user experience with a responsive design.

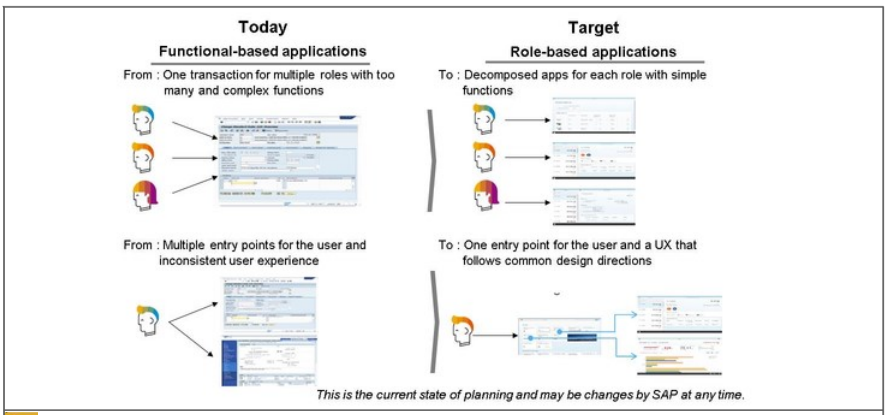


Figure 22: Intuitive User Experience and a Responsive Design

It upholds five basic principles.

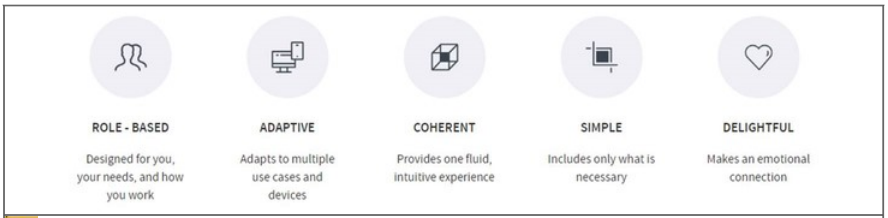


Figure 23: SAP Fiori Principles

SAP Fiori Object Browser (AT3)

There are three categories of apps currently developed in SAP Fiori.

These three categories help developers to guide their architecture and development effort.

- Archetype I: These scenarios are typically transactional and represent views on and interactions with existing business processes and solutions.
- Archetype II: Purely analytical apps. Typical representatives are smart business (KPI Cockpits) but also other analytical, predictive, and planning applications.
- Archetype III: Search applications.

In addition, it supports a mix of applications from different archetypes in a way that users can navigate from an "Insight" App to an "Action" App, which means from an archetype II analytical application or archetype III fact sheet to a transactional archetype I application.

There are three main app types.

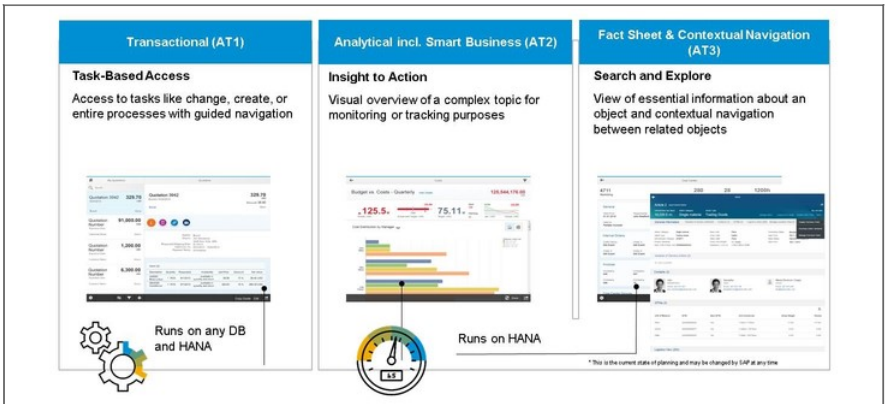


Figure 24: App Types

The SAP FIori launchpad contains three main areas, the workspace, notification and the me area.

Figure 25: SAP Fiori Launchpad at a Glance

It makes the information you want accessible.

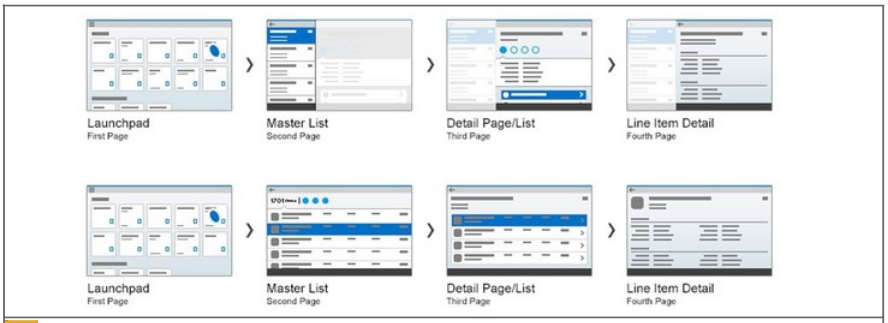


Figure 26: Simple, Delightful, and Intuitive User Experience

Technology has been developed to allow the development of an application from good design principles.

Figure 27: From Technology to Design

LESSON SUMMARY
You should now be able to:

- Explain the SAP Fiori UX

Unit 3

Learning Assessment

1. What are the key principles of SAP Fiori?

   Choose the correct answers.

   - [ ] **A** Role-based
   - [ ] **B** Adaptive
   - [ ] **C** Complex
   - [ ] **D** Coherent
   - [ ] **E** Simple

2. What application type is implemented to analyze a huge amount of data with diagrams?

   Choose the correct answers.

   - [ ] **A** Transactional
   - [ ] **B** Analytical
   - [ ] **C** Fact sheet

3. What names have the view ports of the SAP Fiori launchpad?

   Choose the correct answers.

   - [ ] **A** Me area
   - [ ] **B** Desktop
   - [ ] **C** Workspace
   - [ ] **D** Notification area

4. What is the preferred technology to implement SAP Fiori applications?

   Choose the correct answer.

- ☐ **A** Web Dynpro ABAP
- ☐ **B** SAPUI5
- ☐ **C** Web Dynpro Java
- ☐ **D** Dynpro

Unit 3

Learning Assessment - Answers

1. What are the key principles of SAP Fiori?

   Choose the correct answers.

   - [X] **A** Role-based
   - [X] **B** Adaptive
   - [ ] **C** Complex
   - [X] **D** Coherent
   - [X] **E** Simple

   You are correct! SAP Fiori applications are role-based, adaptive, coherent and simple.

2. What application type is implemented to analyze a huge amount of data with diagrams?

   Choose the correct answers.

   - [ ] **A** Transactional
   - [X] **B** Analytical
   - [ ] **C** Fact sheet

   You are not correct! Analytical Apps are implemented to display a huge amount of data.

3. What names have the view ports of the SAP Fiori launchpad?

   Choose the correct answers.

   - [X] **A** Me area
   - [ ] **B** Desktop
   - [X] **C** Workspace
   - [X] **D** Notification area

   You are correct! The three view ports of the SAP Fiori launchpad are the me area, the workspace and the notification area.

4. What is the preferred technology to implement SAP Fiori applications?

   Choose the correct answer.

- [ ] **A** Web Dynpro ABAP
- [X] **B** SAPUI5
- [ ] **C** Web Dynpro Java
- [ ] **D** Dynpro

You are correct! The preferred technology to implement SAP Fiori applications is SAPUI5.

UNIT 4 User Experience Design

UNIT OBJECTIVES

- Explain design thinking
- Explain decomposition and re-composition
- Work with SAP BUILD

Figure 32: Bringing the Best of Two Worlds Together

What is involved?

Solving "challenges" and problems by generating ideas is equivalent to being innovative. Actually this means, that design thinking is an innovation approach.

In this regard, design thinking seeks to identify the optimal solution in the intersection between people's desires, technological feasibility, and business viability.

The core focus of design thinking lies in the "desirability" section. Other techniques can complement design thinking to cover viability and feasibility.

A consequence of the strong focus on people's desire, are higher investments in the early stages of product and solution developments.

The problem space has to be deeply explored, before solution ideas can be generated.

This ensures, that the problem is really understood and solution ideas are validated before faulty designs are implemented.

Finally, this avoids expensive redesigns at later stages.

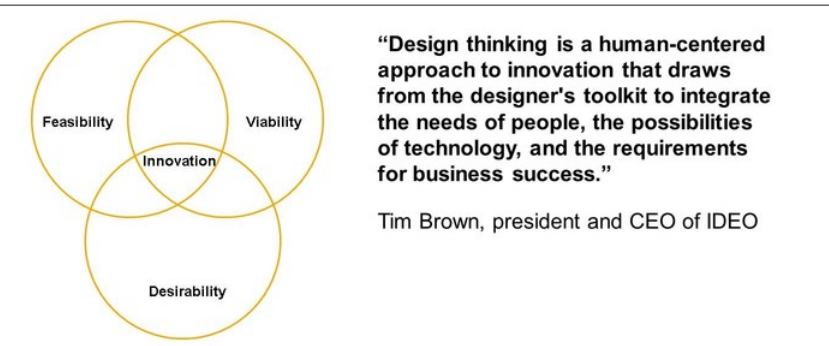


Figure 33: Design Thinking is a Human-centered Iterative Innovation Process

There are a number of phases to go through in the design process.

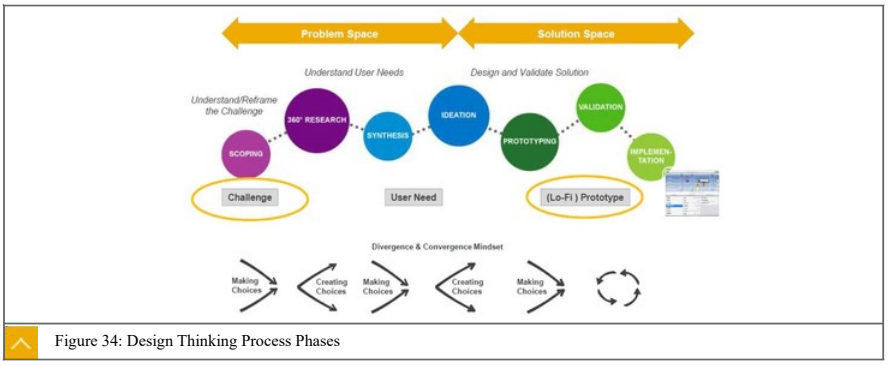


Figure 34: Design Thinking Process Phases

Some guidelines can be adopted at the brainstorming stage.

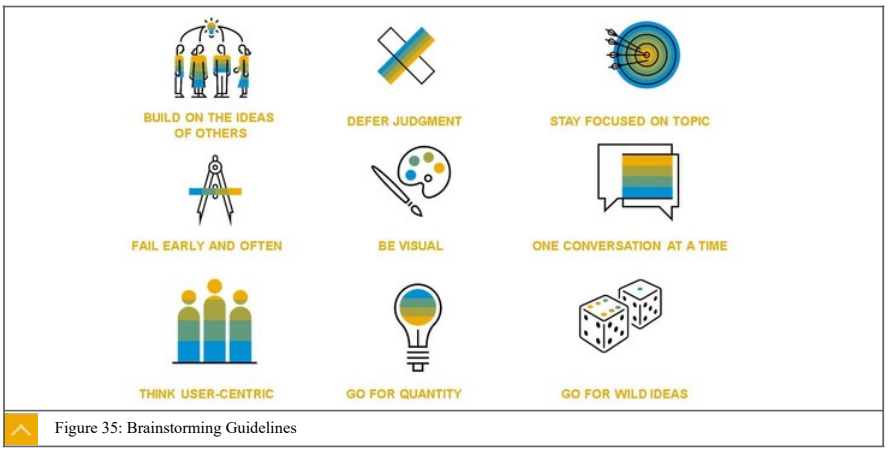


Figure 35: Brainstorming Guidelines

LESSON SUMMARY
You should now be able to:

- Explain design thinking

Unit 4
Lesson 2

Explaining Decomposition and Re-composition

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Explain decomposition and re-composition

Decomposition and Re-composition

There is a distinction between transactions and apps.

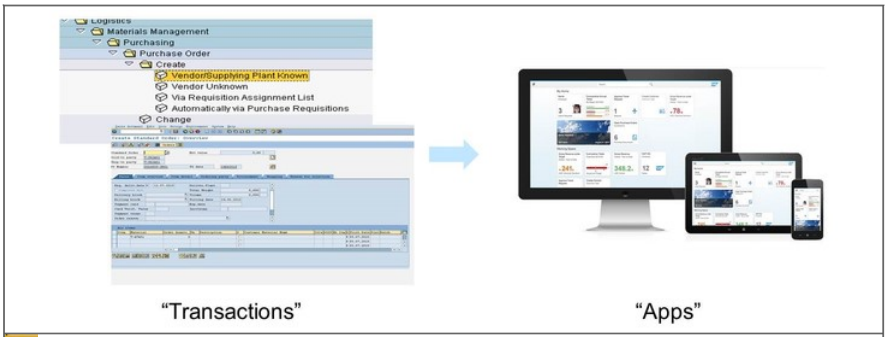


Figure 36: Composition – Transactions versus Apps

Transactions are designed from a systems perspective.

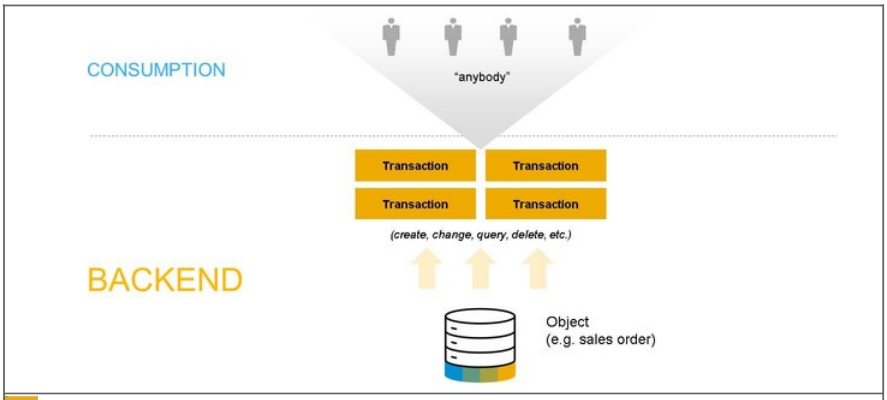


Figure 37: Designed from a System Perspective

This presents a vast complex system to the user where only a proportion is relevant.

Figure 38: From a User Perspective – Complexity

It is not efficient for the user to spend time trying to find what they are looking for, and then getting lost in the process.

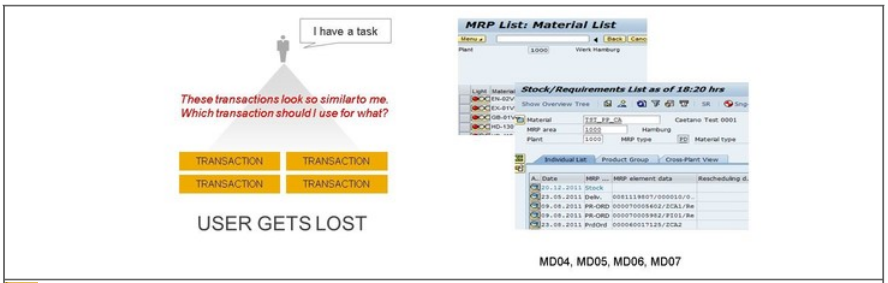


Figure 39: From a User Perspective – Complexity (Continued)

Implementing apps for role-based tasks creates a more efficient system.

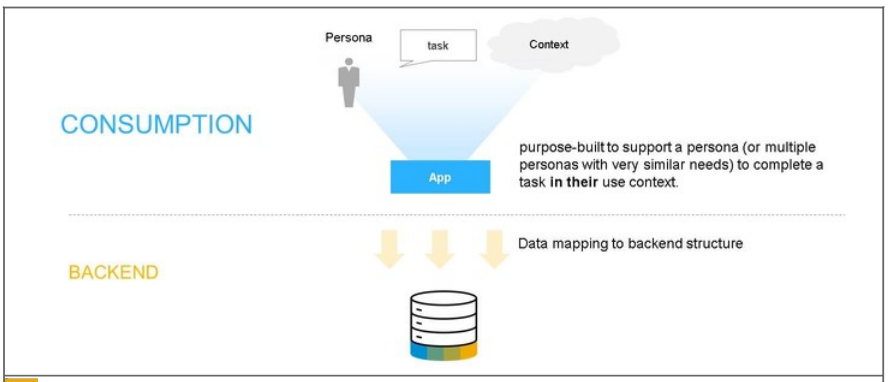


Figure 40: User and Task-centric Focus

The systems and processes are broken down and recomposed to present only what is relevant.

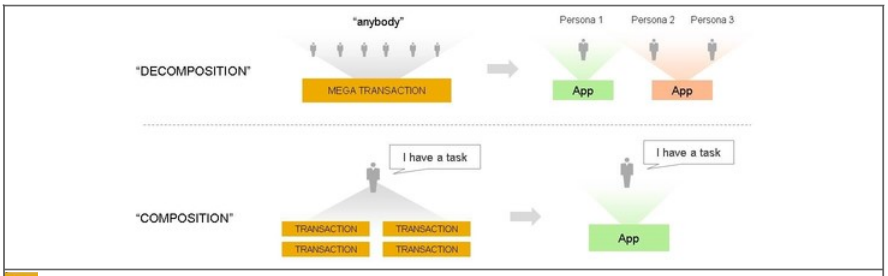


Figure 41: Decomposition and Composition

LESSON SUMMARY

You should now be able to:

- Explain decomposition and re-composition

Unit 4
Lesson 3

Working with SAP BUILD

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Work with SAP BUILD

Introduction to SAP BUILD

SAP Build is an online tool to create prototypes. Features include the ability to:

- Create interactive prototypes from low to high fidelity
- Create interactive prototypes collaboratively
- Leverage SAP Fiori user interface controls and add real sample data
- Handover a prototype to the SAP Web IDE with starter code

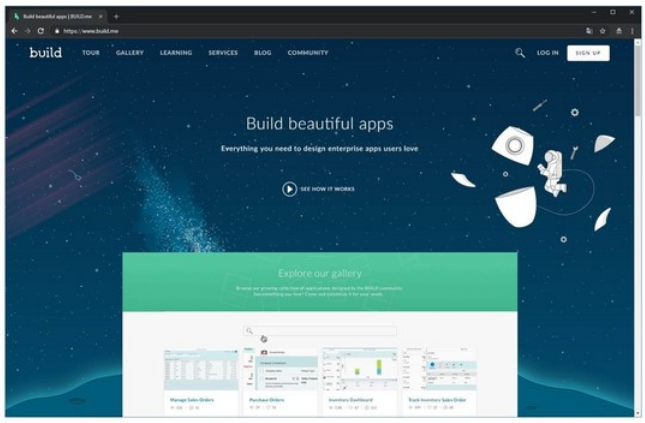


Figure 42: Using Build for User Interface Design

SAP Build provides a workspace, tools and guidance for user interface design.

SAP Build can be found at https://www.build.me

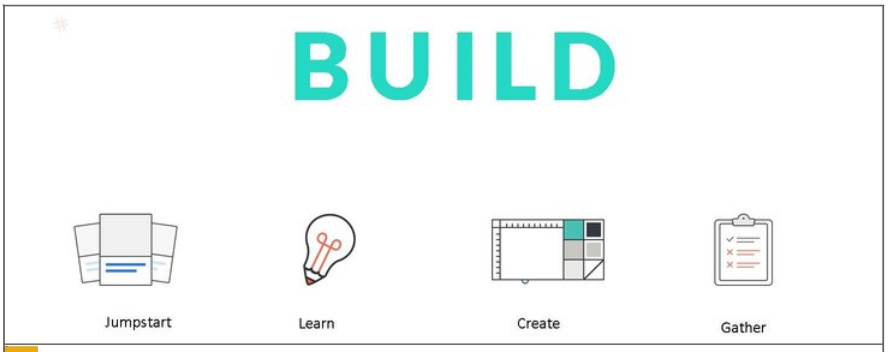


Figure 43: Use SAP Build for User Interface Design

It is preloaded with prototypes and controls that allow for fast and efficient interface development.

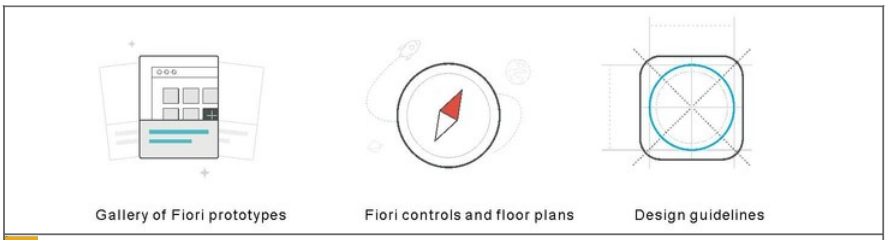


Figure 44: Build Provides Everything Needed to Design Delightful SAP Apps

An array of projects can be created and accessed from within the workspace.

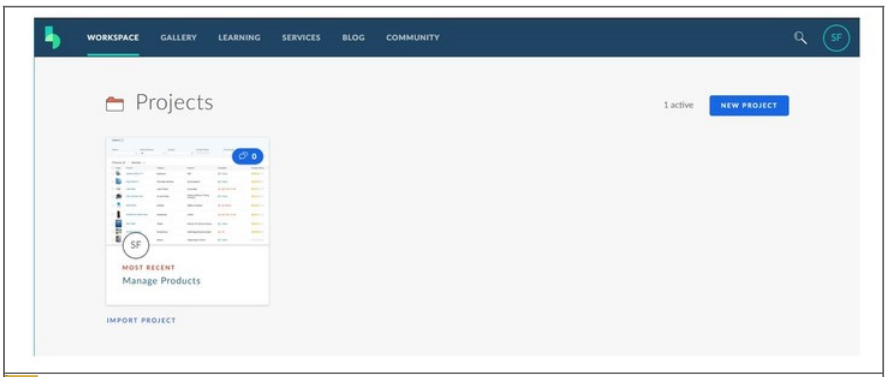


Figure 45: Create a Project

Each new project can incorporate a variety of predefined artefacts such as buttons and tiles.

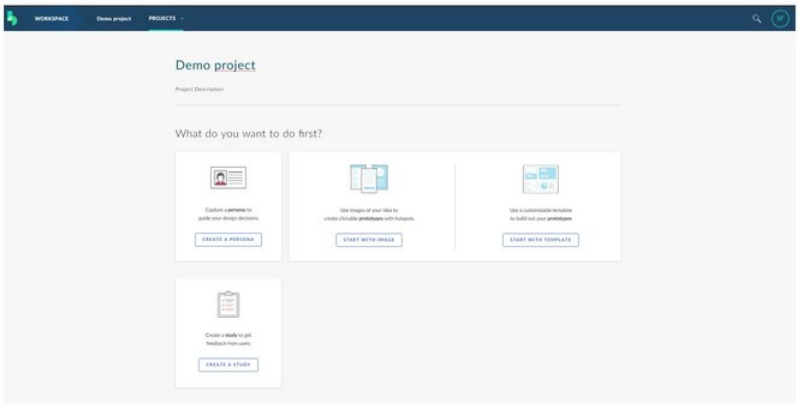


Figure 46: New Projects May Consist of Various Artefacts

SAP Build can be used for Low-Fidelity (LoFi) and High-Fidelity (HiFi) prototyping.

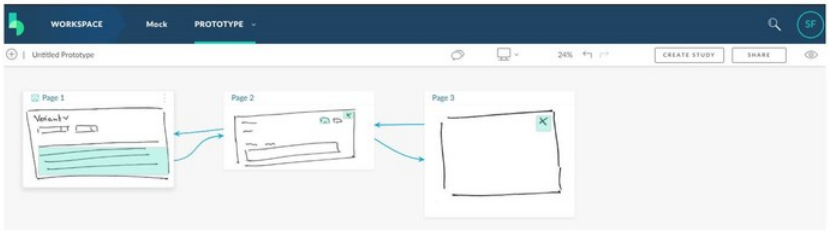


Figure 47: Implementing LoFi Prototypes

Templates are provided to use for new projects and assist with fast development.

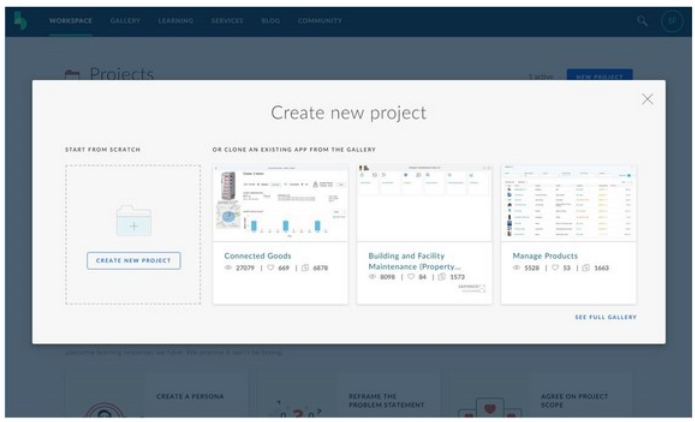


Figure 48: Create a Project Using Templates

There is a gallery of templates from which to choose.

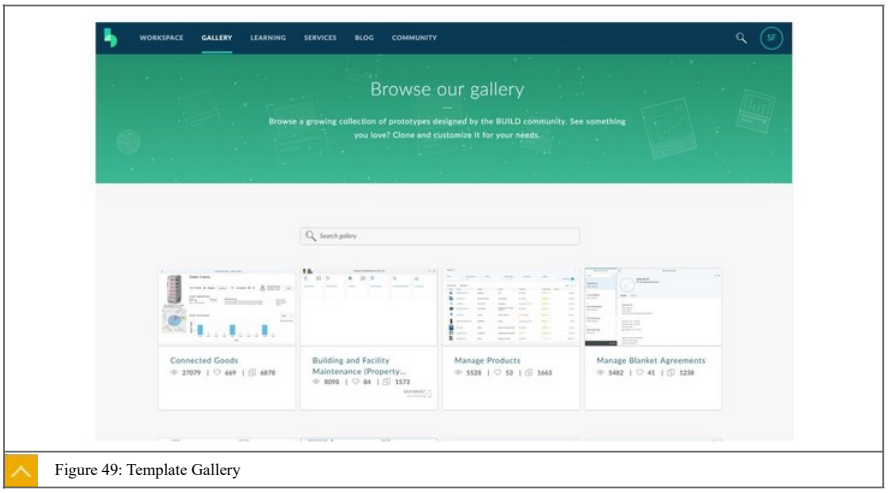


Figure 49: Template Gallery

The template displays a project overview.

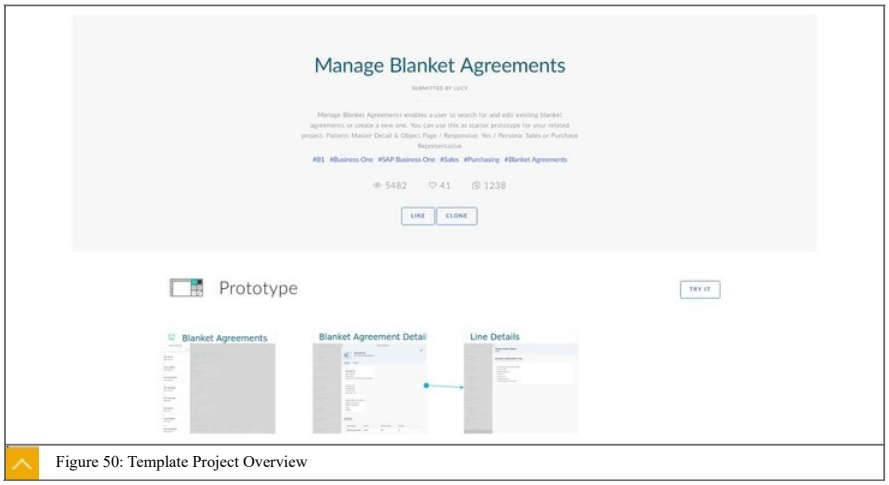


Figure 50: Template Project Overview

The project can be demonstrated to see how it displays and functions.

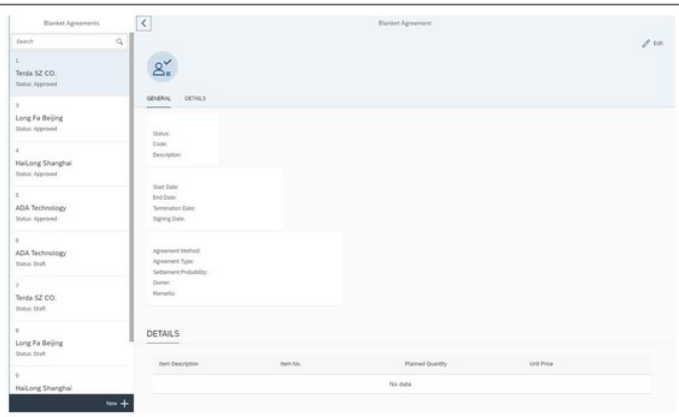


Figure 51: Template Project Demo

A project can be cloned to create a new project.

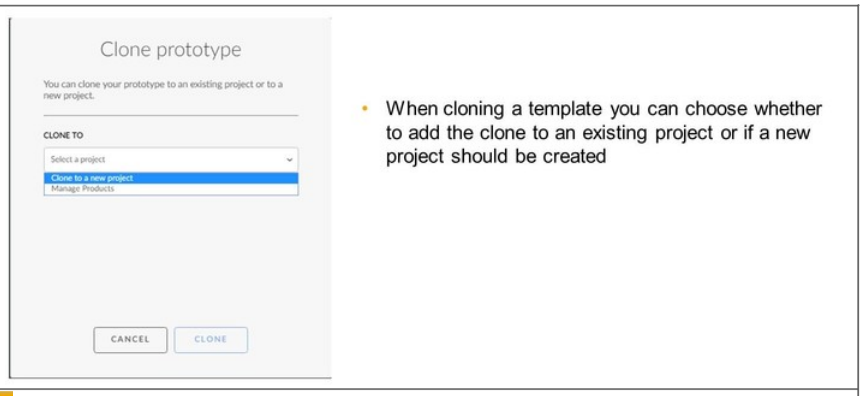


Figure 52: Clone a Template Project

Alternatively, create a variation on a project that already exists..

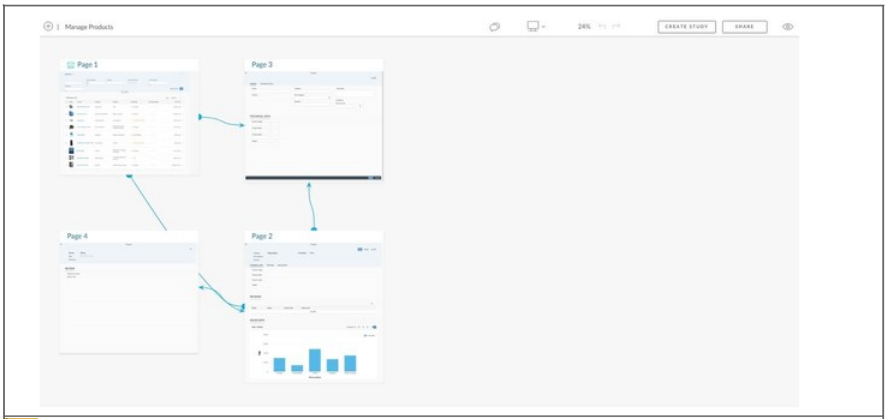


Figure 53: Work with a Clone Template Project

The templates can be extended with different floor plans to suit the business scenario.

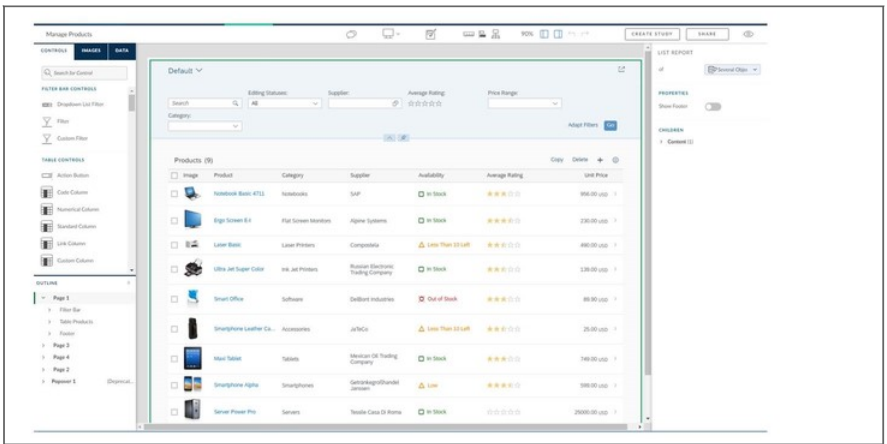


Figure 54: Adapting a Cloned Project

Using the data editor of SAP Build, you can model the Entity Data Model (EDM) for your application, and also add mock data.

The data editor also provides functionality to create small and simple formulae to calculate the content of a field.

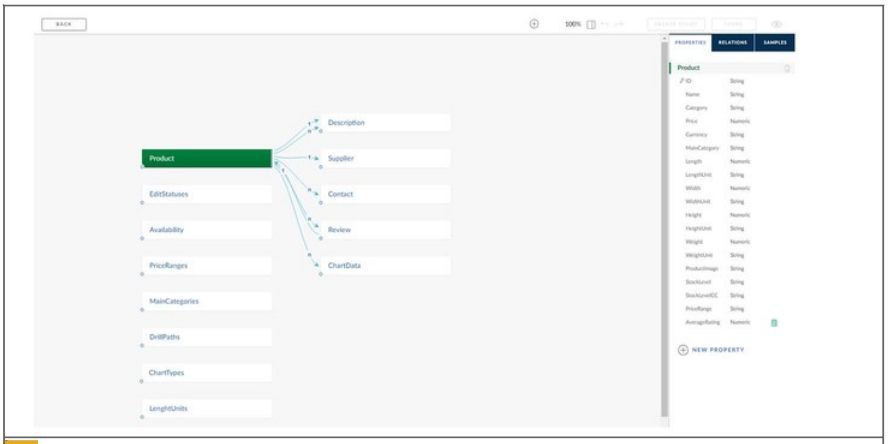


Figure 55: Creating and Working with Simulation Data

When running the prototype in preview mode, you can test the look and feel on different form factors.

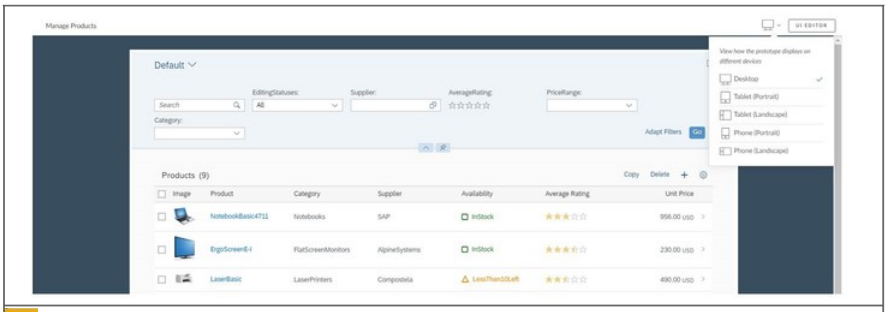


Figure 56: Preview Your Work Inside SAP Build

Using the share function, you can download, your project as a SAP Web IDE project. After the download you can easily import the project into the SAP Web IDE.

When importing the project, the `metadata.xml` file based on your simulation data and the mock data already exist.

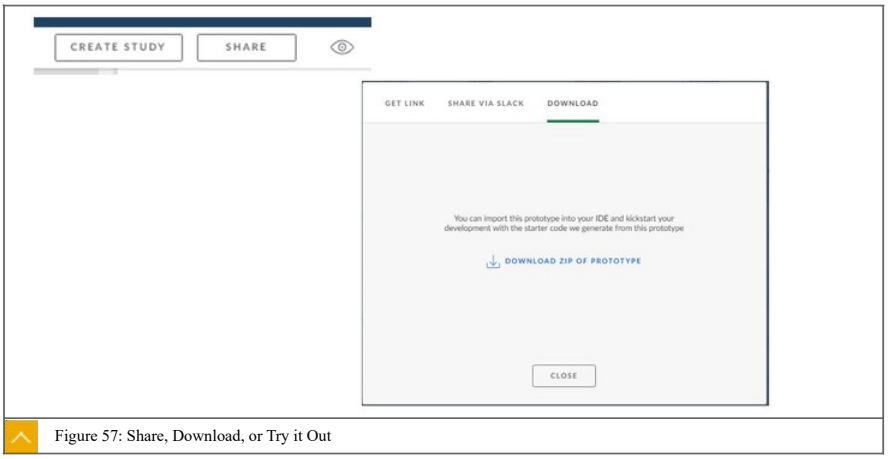


Figure 57: Share, Download, or Try it Out

LESSON SUMMARY
You should now be able to:

- Work with SAP BUILD

Unit 4

Learning Assessment

1. What three key aspects are important when talking about innovation?

   Choose the correct answers.

   - [ ] **A** Feasibility
   - [ ] **B** Scalability
   - [ ] **C** Viability
   - [ ] **D** Desirability

2. What are the main spaces in the design thinking process?

   Choose the correct answers.

   - [ ] **A** Problem Space
   - [ ] **B** Ideation Space
   - [ ] **C** Solution Space
   - [ ] **D** Test Space

3. What are common design thinking brainstorming rules?

   Choose the correct answers.

   - [ ] **A** Be visual
   - [ ] **B** Discuss and judge every idea
   - [ ] **C** Go for quality
   - [ ] **D** Fail early and often

4. What do you focus on when it comes to decomposition and re-composition?

   Choose the correct answer.

   - ☐ **A** The building of mega transactions to support all needs of all people
   - ☐ **B** The person's tasks that should be supported by the application
   - ☐ **C** The expensive of the implementation
   - ☐ **D** The data and functions that are already in the system

5. What is the purpose of SAP Build?

   Choose the correct answer.

   - ☐ **A** Provides tooling to support the designer during various phases of design thinking.
   - ☐ **B** Provides tooling to support the developer during the development of back-end services.
   - ☐ **C** Provides tooling to support the developer during the development of SAPUI5 applications.

Unit 4

Learning Assessment - Answers

1. What three key aspects are important when talking about innovation?

   Choose the correct answers.

   - [X] **A** Feasibility
   - [ ] **B** Scalability
   - [X] **C** Viability
   - [X] **D** Desirability

   You are correct. Design thinking is a human-centered approach to innovation that draws from the designer's toolkit to integrate the needs of people (Desirability), the possibilities of technology (Feasibility), and the requirements for business success (Viability).

2. What are the main spaces in the design thinking process?

   Choose the correct answers.

   - [X] **A** Problem Space
   - [ ] **B** Ideation Space
   - [X] **C** Solution Space
   - [ ] **D** Test Space

   You are correct! The design thinking process is divided in two spaces. The Problem Space and the Solution Space.

3. What are common design thinking brainstorming rules?

   Choose the correct answers.

   - [X] **A** Be visual
   - [ ] **B** Discuss and judge every idea
   - [ ] **C** Go for quality
   - [X] **D** Fail early and often

   You are correct! Design Thinking is a very visual approach where you use pen and paper. A common goal is to fail early and often.

4. What do you focus on when it comes to decomposition and re-composition?

   Choose the correct answer.

   - [ ] **A** The building of mega transactions to support all needs of all people
   - [X] **B** The person's tasks that should be supported by the application
   - [ ] **C** The expensive of the implementation
   - [ ] **D** The data and functions that are already in the system

   You are correct! An important aspect is to focus on the tasks to be undertaken.

5. What is the purpose of SAP Build?

   Choose the correct answer.

   - [X] **A** Provides tooling to support the designer during various phases of design thinking.
   - [ ] **B** Provides tooling to support the developer during the development of back-end services.
   - [ ] **C** Provides tooling to support the developer during the development of SAPUI5 applications.

   You are correct! SAP Build supports the designer during various phases of design thinking.

UNIT 5

SAP Fiori Design Guidelines

UNIT OBJECTIVES

- Understand SAP Fiori design guidelines
- Understand app types

Unit 5
Lesson 1

Understanding SAP Fiori Design Guidelines

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand SAP Fiori design guidelines

Foundation

The SAP Fiori Design Guidelines can be found at https://experience.sap.com/fiori-design/

Figure 58: SAP Fiori Design Guidelines Welcome Screen

The SAP Fiori Design Guidelines for iOS can be found at https://experience.sap.com/fiori-design-ios/ .

The SAP Fiori Design Guidelines reference the common Human Interface Guidelines of apple. These guidelines can be found at:

https://developer.apple.com/ios/human-interface-guidelines/overview/themes/ (or just click on the Designing for iOS option)

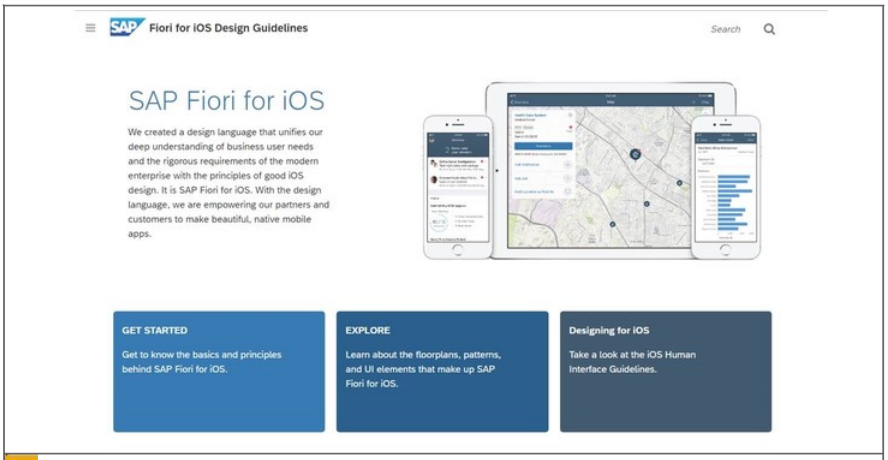


Figure 59: SAP Fiori for iOS

The SAP Fiori Design Guidelines for Android can be found at https://experience.sap.com/fiori-design-android/.

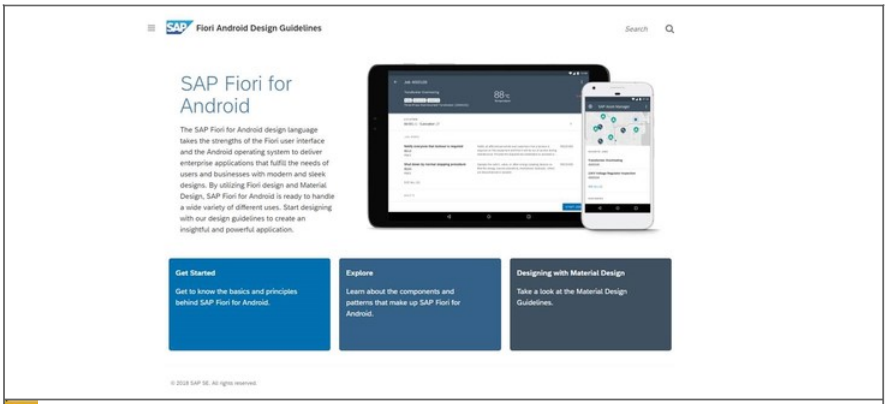


Figure 60: SAP Fiori for Android

SAP FIori Design guidelines have been developed for the web and can be found at https://experience.sap.com/fiori-design-web/.

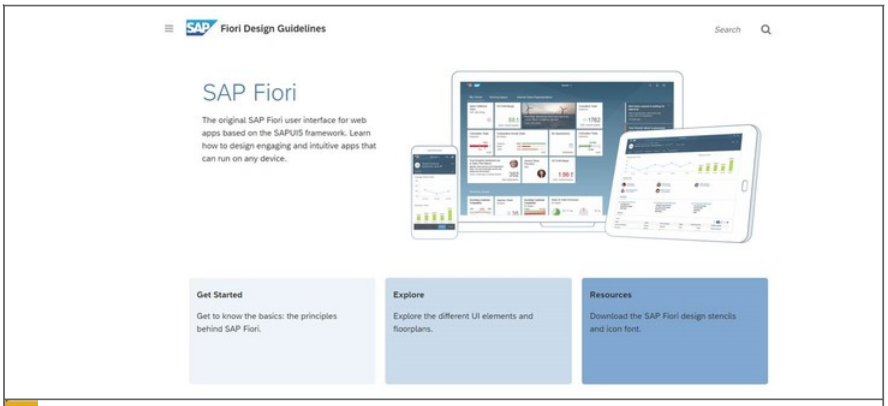


Figure 61: SAP Fiori Design Guidelines for the Web

Five golden principles apply when you implement a SAP Fiori application.

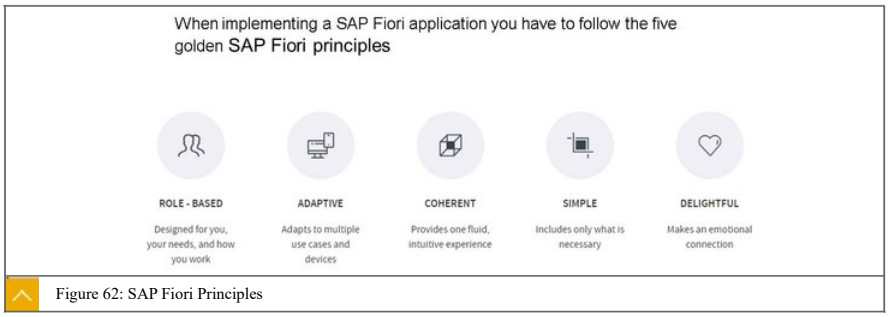


Figure 62: SAP Fiori Principles

These principles should be applied.

- SAP Fiori apps must run on all three major device types, desktop, tablet, and smartphone.
- When app functionality, information, or both cannot be offered equally on all device types, responsive methods should be used to automatically adjust the app.
- When app functionality and or information cannot be offered unaltered on different device types (due to use case, available screen real estate, or device capabilities), the app should be adjusted manually to device specific requirements.
- Adaptive and responsive methods can be mixed.
- If you are using desktop controls that are not recommended: sap.commons, allowed: sap.table, you currently must replace them with responsive controls to run on a mobile device.
- If you are using the ALV (analytical table), it should be replaced by the responsive table (sap.m) or another use case specific solution (for example, charts, texts or KPIs).

Responsive patterns and controls: SAPUI5 offers a variety of controls and patterns that automatically adjust to different form factors (table, object header, filter bar, and so on – plus the compact vs. cozy form factor). The big advantage of responsive design is the fact that apps can adjust to different device types without additional effort in coding and maintenance.

Adaptive design of apps: Ideally follow a mobile first design approach, where apps must manually define different designs for different form factors. This is more effort but allows more targeted support of device-specific use cases.

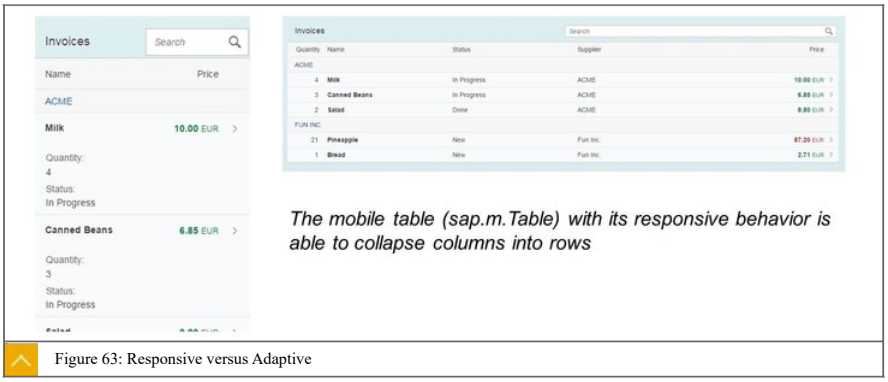


Figure 63: Responsive versus Adaptive

Designing for mobile first endures that apps are responsive across platforms.

- Mobile first means looking at the scenario as if you are designing first for a mobile application
- Mobile first means you have to deal with restrictions - ***Build up, don't tear down***
- Stay focused on what's important – ***Think ahead***
- Mobile first forces you to reconsider established solutions
- Working with restrictions will lead you to find smart ways to reduce, aggregate or group
- Mobile first has responsive or adaptive design in mind from the first moment of development

Figure 64: Mobile First Approach

A grid layout is used to provide a responsive interface.

- The responsive grid positions UI elements in a 12-column flow layout
- Can be configured to display variable number of columns
- Flexible layouts for various screen sizes are possible

Figure 65: Grid Layout (1 of 2)

A new measuring system in CSS 3 uses rem instead of pixels for responsive layouts.

- SAP Fiori uses a new measuring system using rem instead of px
- rem stands for root em
- A base size of 16 px defines 1 rem
- SAP Fiori UI elements are based on a 1 x 1 rem (16 x 16 px) flexible layout grid system
- For example if your font size is 18px the font size-value is 1.125rem → 18/16=1.125 (based on the assumption of 16 px standard font size and font size of root element 100%). More general ***PX / 16 = rem-value***
- ***Caveat-factor: rem is specified in CSS3***

Figure 66: Grid Layout (2 of 2)

In web design, it is common practice to restrict the page content to a certain width. This way the content does not stretch too wide and become unreadable. If a screen gets wider than this maximum width, blank areas to the left and right are shown. In many cases, these areas are used to display advertisements.

This design element is called letterboxing because it restricts the content to a certain width similar to the width of a letterbox format.

SAP Fiori also offers letterboxing. In earlier releases, letterboxing to 1024 px was automatically enforced to avoid distorting contents. Letterboxing became an optional feature in later releases. Today, SAPUI5 can easily adjust content to different screen sizes by using responsive layouts and controls. To take account of the increased average screen size in our customer's IT infrastructures, we have increased the letterboxing to 1280 px.

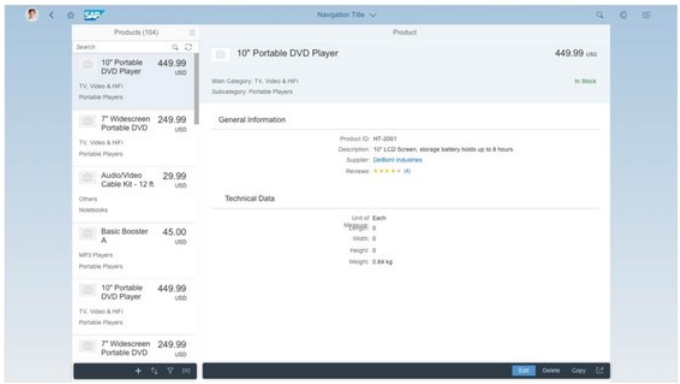


Figure 67: Letterboxing

To adapt to different interaction styles better, SAP Fiori apps need to support two different form factors:

- When running on touch devices, the app should display all controls with dimensions that are big enough for finger tips. This is called the cozy form factor.
- When running on mouse- and keyboard-enabled (non-touch) devices, the information density on the screen can be increased by reducing control dimensions. This is called the compact form factor. In the compact mode, the default control area of 3 rem is reduced to 2 rem.

The recommended form factors are cozy and compact.

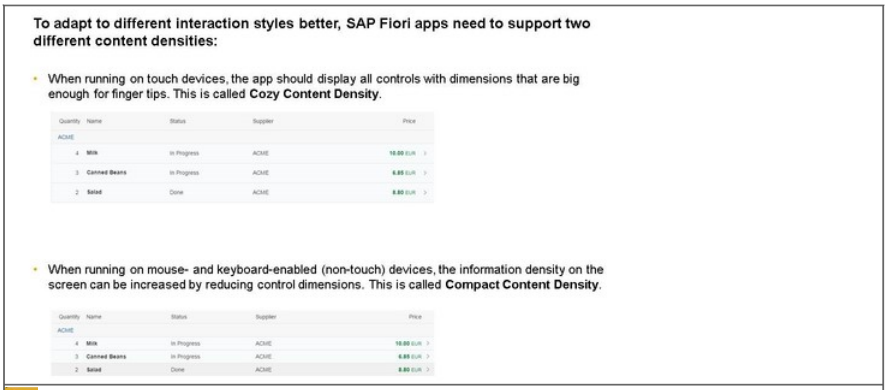


Figure 68: Content Density (Cozy and Compact)

The form factors are defined in the code.

```
getContentDensityClass : function() {
    if (this._sContentDensityClass === undefined) {
        // check whether FLP has already set the content density class; do nothing in this case
        if (jQuery(document.body).hasClass("sapUiSizeCozy") ||
                jQuery(document.body).hasClass("sapUiSizeCompact")) {
            this._sContentDensityClass = "";
        } else if (!Device.support.touch) { // apply "compact" mode if touch is not supported
            this._sContentDensityClass = "sapUiSizeCompact";
        } else {
            // "cozy" in case of touch support;
            // default for most sap.m controls, but needed for desktop-first
            // controls like sap.ui.table.Table
            this._sContentDensityClass = "sapUiSizeCozy";
        }
    }
    return this._sContentDensityClass;
}

            // apply content density mode to root view
            this.getView().addStyleClass(this.getOwnerComponent().getContentDensityClass());
        }
```

Figure 69: Working with the Factors

Iconography

- Unlike images of icons, the icon font is fully customizable with CSS. Therefore different themes can be applied.
- The SAPUI5 Icon Explorer provides access to more than 500 scalable pictograms.

Typography

- SAP Fiori uses a set of timeless, sans-serif system fonts.
- A font stack is used within the CSS that includes Arial and Helvetica in combination with the Roboto font for Android.
- System fonts offer the advantage of no loading time, and a consistent font is always available.

LESSON SUMMARY
You should now be able to:

- Understand SAP Fiori design guidelines

Unit 5
Lesson 2

Understanding App Types

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand app types

App Types

SAP Fiori has two different app types.

- Conceptual app types:

  A conceptual application type defines how the overall design of the application looks like, for example the approval app is a kind of conceptual application.

- Technical app types:

  A technical application type specifies the technical aspects of an application, for example the transactional app is a kind of technical app type.

The SAP Fiori reference library is found at: https://fioriappslibrary.hana.ondemand.com/sap/fix/externalViewer/.

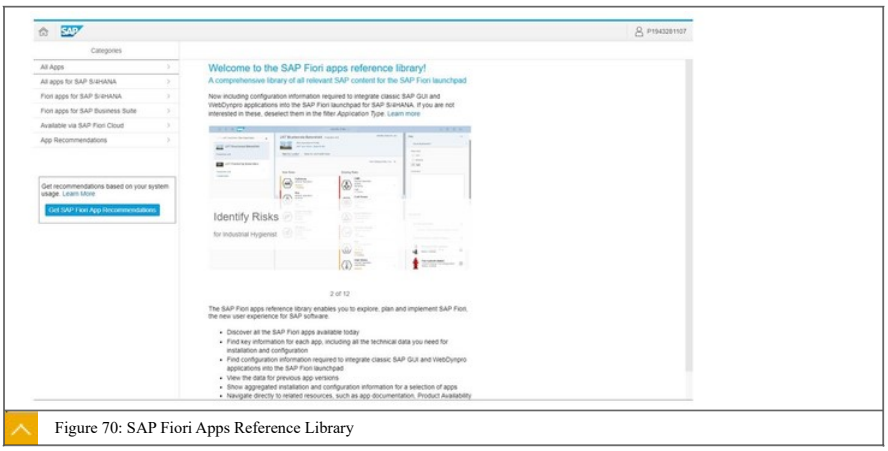


Figure 70: SAP Fiori Apps Reference Library

Transactional apps usually perform some processing task.

- These apps are used for performing transactional tasks, such as creating a leave request for an employee.

- They represent simplified views and interaction with existing business processes and solutions.

The SAP Fiori apps library delivers efficiency for development.

There is decoupling of the front-end and back-end systems.

- Life cycle decoupling of User Interface (UI) apps from the back-end server
- Especially for apps that must also run on any database
- Allows faster iterations for the UI apps
- Allows changes to a UI without the need for development privileges in the back-end server
- Add-on based delivery enables fast release cycles

There is a single point of UI maintenance like browser support or UI5 provisioning.

It has a central place for themes and branding.

It has a single place for configuration, personalization, and SAP Fiori shell services.

There is rule-based dispatching of requests in a multi-system landscape, for example, for approvals including aggregation.

It addresses security considerations.

- Similar to an application-level gateway (ALG) with protocol switch and whitelisting (excluding search)
- The admin for UI metadata does not need admin rights in the back-end server which holds sensitive data

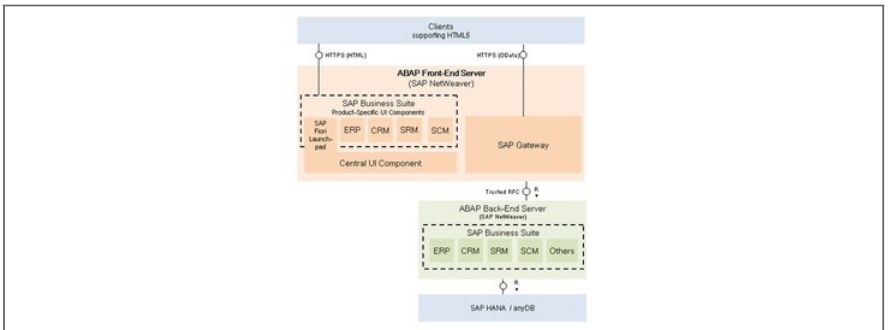


Figure 71: Transactional Apps Landscape

Contextual information about central objects used in business operations are displayed in these apps. Fact sheets are designed to be intuitive and harmonized. From a fact sheet area (tile), it is possible to drill down into its details. It is easy to navigate from one fact sheet to its related fact sheets such as navigate from a document to the related business partner or to the master data. From fact sheets, it is also possible to access related transactions in operational systems, access the back-end system to display document details or to edit the document from a document fact sheet.

Fact sheets only run on an SAP HANA database and require an ABAP stack.

- Display contextual information about central objects used in business operations
- Designed to be intuitive and harmonized
- Providing drill down into details
- Navigation from one fact sheet to related fact sheets
- Possible to access related transactions in operational systems
- Access the back-end system to display document details or possible to edit the document from a document fact sheet

An overview of the fact sheet landscape demonstrates these relationships.

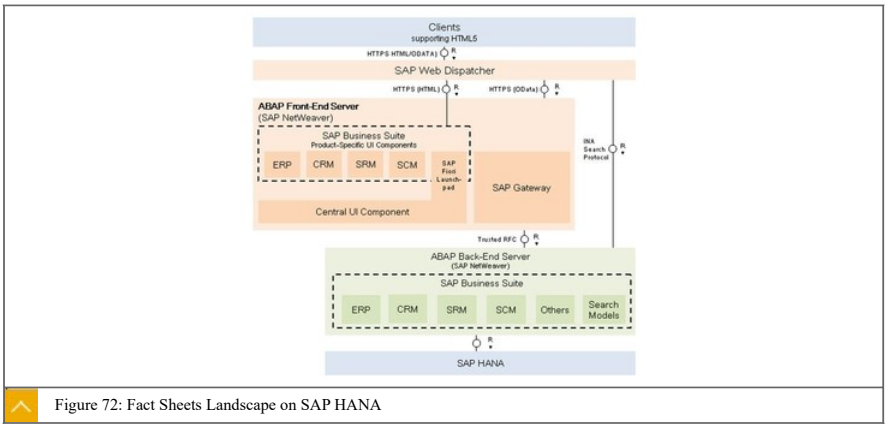


Figure 72: Fact Sheets Landscape on SAP HANA

It is possible to get a role-based insight into real-time operations of a business by collecting and displaying key figures directly in the browser, through the SAP Analytical apps. To do this, the SAP Analytical apps combine the data and analytical power of SAP HANA with the integration and interface components of SAP Business Suite. With SAP Analytical apps, the customer's company is able to closely monitor their most important KPIs in real time and react immediately on changes in market conditions or operations.

Get role -based insights into business by collecting and displaying key figures directly in the browser through SAP analytical apps.

Customer's company is able to closely monitor their most important KPIs in real time and react immediately on changes in market conditions or operations.

An overview of the analytical App landscape demonstrates these relationships on SAP HANA.

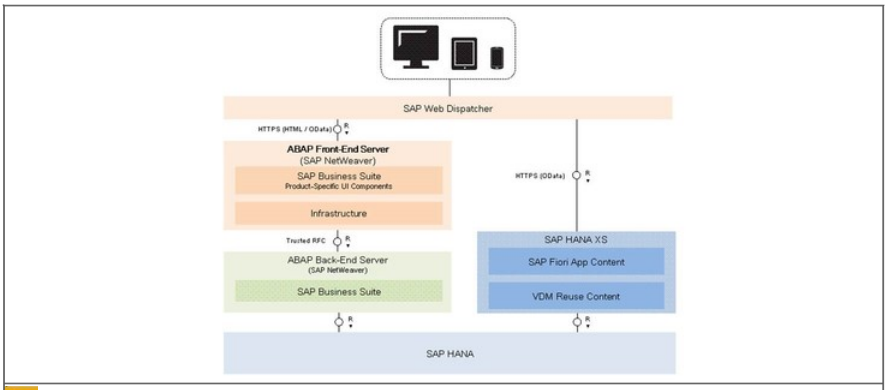


Figure 73: Analytical App Landscape on SAP HANA

An overview of the analytical landscape on S/4HANA demonstrates these new relationships.

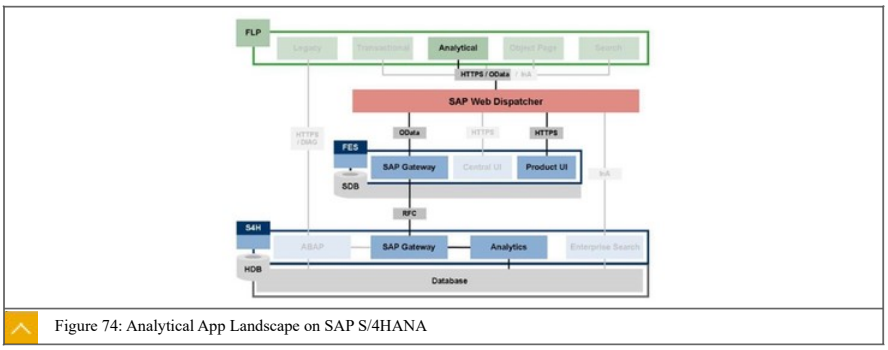


Figure 74: Analytical App Landscape on SAP S/4HANA

These new relationships provide services across the range of platforms required.

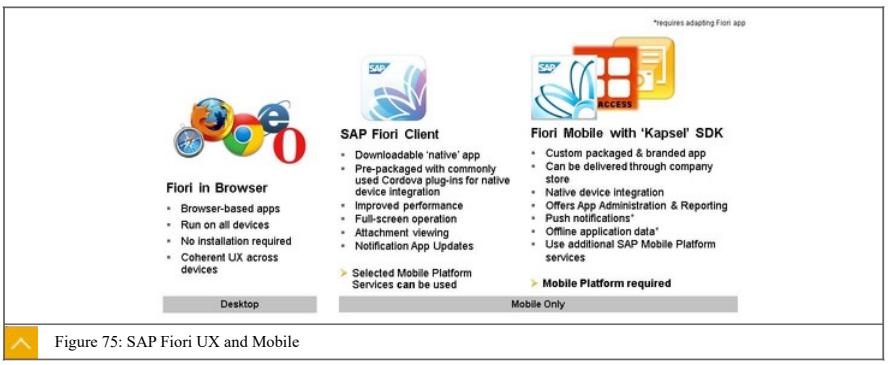


Figure 75: SAP Fiori UX and Mobile

LESSON SUMMARY
You should now be able to:

- Understand app types

Unit 5

Learning Assessment

1. The SAP Fiori Design Guidelines are available for various device types. For what device types are the design guidelines available?

   Choose the correct answers.

   - [ ] **A** SAP Fiori for Windows
   - [ ] **B** SAP Fiori for Android
   - [ ] **C** SAP Fiori for Web
   - [ ] **D** SAP Fiori for iOS
   - [ ] **E** SAP Fiori for Blackberry

2. What content density is used for touch-enabled devices?

   Choose the correct answer.

   - [ ] **A** Compact form factor
   - [ ] **B** Mobile form factor
   - [ ] **C** Cozy form factor
   - [ ] **D** Desktop form factor
   - [ ] **E** Small form factor

3. Which different app types can be found in SAP Fiori?

   Choose the correct answers.

   - [ ] **A** Conceptual
   - [ ] **B** Industrial
   - [ ] **C** Technical
   - [ ] **D** Infrastructural
   - [ ] **E** Mobile

4. Which technical application types are defined for SAP HANA?

   Choose the correct answers.

- [ ] **A** Transactional
- [ ] **B** Secure
- [ ] **C** Fact Sheets
- [ ] **D** Analytical
- [ ] **E** Drill-down

Unit 5

Learning Assessment - Answers

1. The SAP Fiori Design Guidelines are available for various device types. For what device types are the design guidelines available?

   Choose the correct answers.

   - [ ] **A** SAP Fiori for Windows
   - [X] **B** SAP Fiori for Android
   - [X] **C** SAP Fiori for Web
   - [X] **D** SAP Fiori for iOS
   - [ ] **E** SAP Fiori for Blackberry

   You are correct! There are currently three different SAP Fiori design guidelines available. The guidelines are available for Android, iOS, and for Web applications.

2. What content density is used for touch-enabled devices?

   Choose the correct answer.

   - [ ] **A** Compact form factor
   - [ ] **B** Mobile form factor
   - [X] **C** Cozy form factor
   - [ ] **D** Desktop form factor
   - [ ] **E** Small form factor

   You are correct! The form factor for touch-enabled devices is called the Cozy form factor.

3. Which different app types can be found in SAP Fiori?

   Choose the correct answers.

   - [X] **A** Conceptual
   - [ ] **B** Industrial
   - [X] **C** Technical
   - [ ] **D** Infrastructural
   - [ ] **E** Mobile

   You are correct! You can distinguish between conceptual and technical app types.

4. Which technical application types are defined for SAP HANA?

   Choose the correct answers.

   - [X] **A** Transactional
   - [ ] **B** Secure
   - [X] **C** Fact Sheets
   - [X] **D** Analytical
   - [ ] **E** Drill-down

   You are correct. For SAP on HANA there are transactional, analytical, and fact sheets apps defined.

UNIT 6

Development Basics SAP Web IDE

Lesson 1

UNIT OBJECTIVES

- Use the development tools of SAP Web IDE

Unit 6
Lesson 1

Using the Development Tools of SAP Web IDE

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Use the development tools of SAP Web IDE

SAP Web IDE and the SAP Cloud Platform

Access to the SAP Web IDE is available on the SAP Cloud Platform Cockpit.

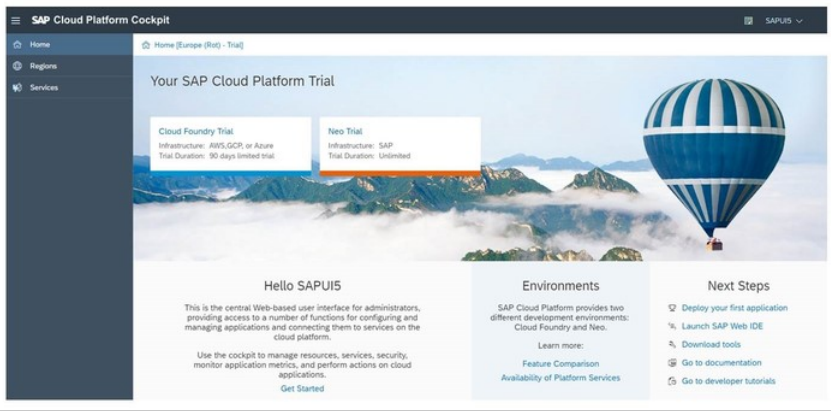


Figure 76: SAP Cloud Platform

Until December 2018, the SAP Cloud Platform provides two different versions of the SAP Web IDE. As published in 2019, only the SAP Web IDE Full-Stack will remain.

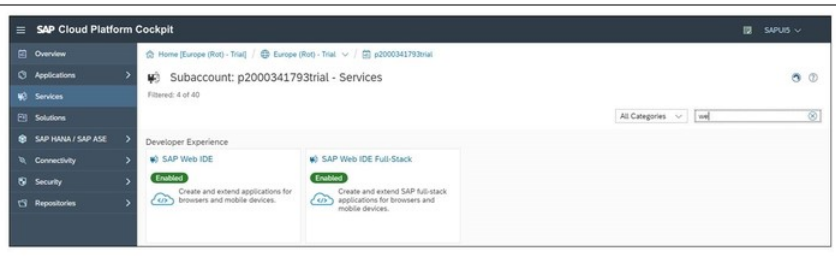


Figure 77: SAP Cloud Platform

SAP Web IDE is a web-based tool that enables you to create and extend end-user applications for browser and mobile devices.

The SAP Web IDE simplifies the end-to-end application life cycle: development, build, and deployment. Customer extensions for SAPUI5 and SAP Fiori applications can be built to fulfill end-user requirements and expectations more effectively. Features include:

- Browser-based development environment
- Standard web development tools like code editors, wizards, and WYSIWYG tooling that are optimized for building responsive HTML5 apps with SAPUI5.
- Application templates act as the foundation for highly efficient app development.
- Supports the end-to-end application lifecycle: UI design, development, testing, deployment, and customer extensions for responsive SAPUI5 apps.
- Everything is source-code based.

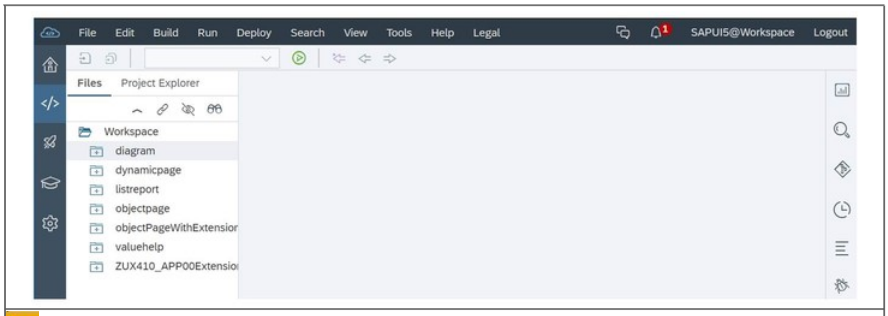


Figure 78: What is the SAP Web IDE?

It facilitates delivery across a range of platforms.

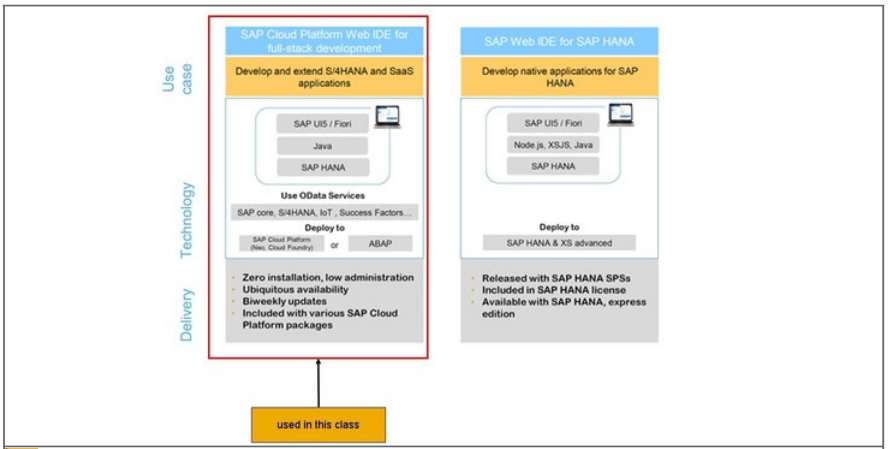


Figure 79: SAP Web IDE Delivery Options

In the Layout Editor, you can display the content of an XML view to see how it appears in your finished application.

The Layout Editor is only supported in Google Chrome and Internet Explorer 11 browsers.

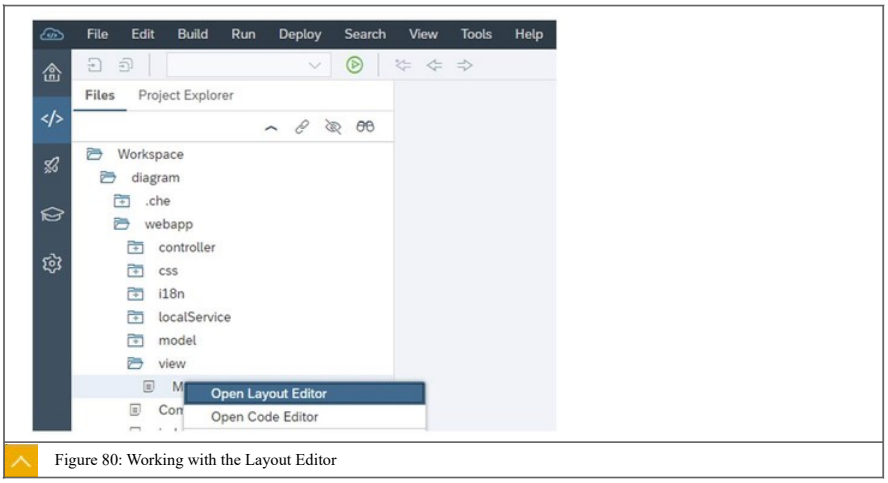


Figure 80: Working with the Layout Editor

The layout editor is composed of a canvas, a pane on the left that includes the controls and outline tabs, and a pane on the right that includes the events and properties panes.

The buttons on the layout editor toolbar allow you to perform the following tasks.

- Change the device format of the canvas to smartphone, tablet, or desktop view.
- Expand and collapse the panes to the right and left of the canvas.
- Undo and redo actions.

Controls Tab : You can expand or collapse each section by clicking the arrow on each section header. You can also search for controls by entering the control name in the search field at the top of the controls tab. The relevant sections expand to display the controls that match the search criteria. it is important to delete the search criteria if you want to expand other sections. You can drag and drop controls from the controls tab onto the canvas.

Outline Tab: Controls that are selected on the outline tab are automatically selected on the canvas and vice versa. You can use the outline tab to see the hierarchy of controls on the canvas. In addition, you can add and remove controls from the canvas using the outline tab.

Canvas : The canvas in the middle of the layout editor area provides a graphical display of the selected XML view. Click a control on the canvas to select it.

The **Events pane** allows you to assign an event handler from the controller to an event of the selected control. The icon next to each event opens the code editor to display the relevant controller of the XML view.

The **Properties pane** shows the properties of the control that is currently selected in the canvas, and allows you to modify its property values. The most commonly used properties for each control are displayed at the top of the list.

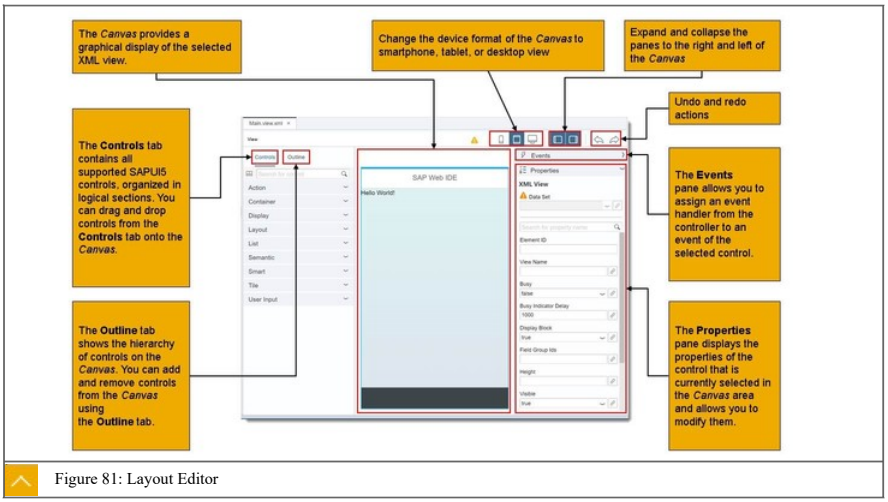


Figure 81: Layout Editor

The **Cloud Connector** combines an easy setup, with a clear configuration of the systems that are exposed to the SAP Cloud Platform. In addition, you can control the resources available for the cloud applications in those systems. Thus, you can benefit from your existing assets without exposing the whole internal landscape.

The **Cloud Connector** runs as on-premise agent in a secured network and acts as a reverse invoke proxy between the on-premise network and SAP Cloud Platform. Due to its reverse invoke support, you don't need to configure the on-premise firewall to allow external access from the cloud to internal systems.

The **Cloud Connector** provides fine-grained control over:

- On-premise systems and resources that shall be accessible by cloud applications
- Cloud applications that shall make use of the Cloud Connector

The **Cloud Connector** allows propagating identity of cloud users to on-premise systems in a secure way.

It can be used to connect an on-premise database, or business information tools to SAP HANA databases in the cloud. That means, it also supports the opposite connection direction (from the on-premise system to the cloud).

You can use the **Cloud Connector** in business critical enterprise scenarios. The tool takes care to automatically re-establish broken connections, provides audit logging of the inbound traffic and configuration changes, and can be run in a high-availability setup.

It supports additional protocols, apart from HTTP. For example, the RFC protocol supports native access to ABAP systems by invoking function modules.

The **Cloud Connector** must not be used with products other than SAP Cloud Platform or SAP S/4HANA Cloud.

The **Cloud Connector** serves as the link between on-demand applications in SAP Cloud Platform and existing on-premise systems.

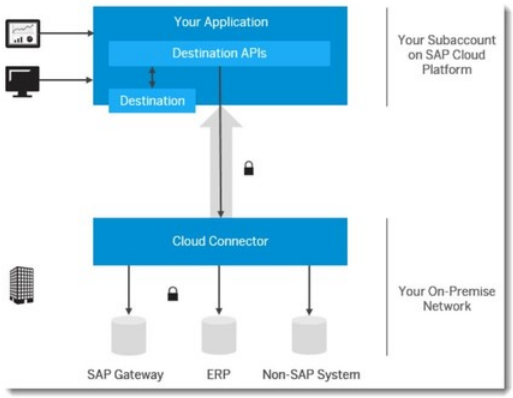


Figure 82: Connecting Cloud Applications to On-premise Systems

Connectivity Destinations are part of SAP Cloud Platform connectivity service.

The connectivity service resolves the destination at runtime based on the symbolic name provided. The result is an object that contains customer-specific configuration details, such as the URL of the remote system or service, and the authentication type.

Connectivity Destinations are used for the outbound communication of a cloud application to a remote system.

Destinations contain the connection details for the remote communication of an application.

Destinations are represented by symbolic names that are used by on-demand applications to refer to remote connections.

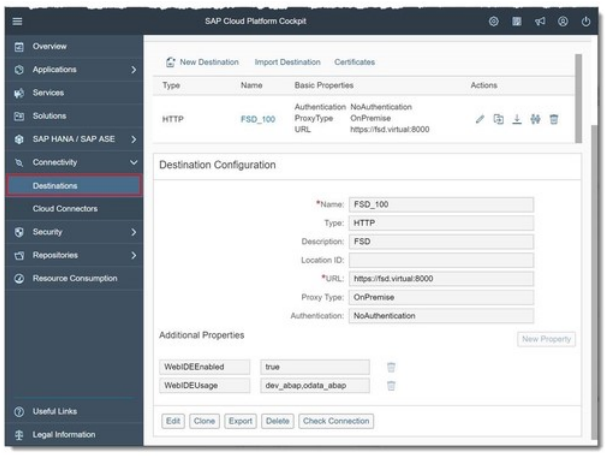


Figure 83: SAP Cloud Platform Cockpit – Destinations

LESSON SUMMARY
You should now be able to:

- Use the development tools of SAP Web IDE

Unit 6

Learning Assessment - Answers

1. Which of the following statements are correct with respect to the SAP Web IDE?

   Choose the correct answers.

	A SAP web IDE is a development environment for SAP ABAP.
X	**B** SAP web IDE is a browser-based development environment.
X	**C** SAP web IDE provides application templates as a foundation of highly efficient app development.

   You are correct! The SAP Web IDE is a browser-based development environment that provides application templates as a foundation for highly efficient app development.

2. What is true of the SAP Cloud Platform Web IDE for full-stack development.

   Choose the correct answers.

X	**A** It requires zero Installation.
	B It is released with SAP HANA SPS′s.
X	**C** There are twice weekly updates.
	D It is included with SAP HANA express edition.

   You are Correct! The SAP Cloud Web IDE full-stack development needs zero installation because it is hosted as a cloud service and can be updated twice weekly.

3. What are the main artefacts of an SAPUI5 application?

   Choose the correct answers.

	A CSS
X	**B** Views
X	**C** Controllers
X	**D** Models
	E controls

   You are correct! SAPUI5 is based on the Model-View-Controller paradigm, so views, controllers, and models are the main artefacts of a SAPUI5 application.

4. What are the tasks of the controller in the Model-View-Controller implementation?

   Choose the correct answers.

- [ ] **A** Updates using data binding
- [X] **B** Modifies the model
- [X] **C** Modifies the view via API
- [ ] **D** Holds the business data
- [ ] **E** Contains the UI implementation

You are correct! The tasks of the controller in the Model-View-Controller paradigm are modifying the model and modifying the view using the API.

UNIT 7

SAPUI5 Advanced Topics

UNIT OBJECTIVES

- Understand SAPUI5 at a Glance
- Understand SAPUI5: bootstrapping and MVC
- Understand SAPUI5: Routing and Navigation
- Understand SAPUI5: Margins, paddings, and CSS
- Understand SAPUI5: OData and the ODataModel
- Understand SAPUI5: Visualizing business data

Unit 7
Lesson 1

SAPUI5 at a Glance

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand SAPUI5 at a Glance

SAPUI5 at a Glance

SAPUI5 is a JavaScript library for modern web business applications.

- One consistent user experience for your apps
- Responsive across browsers and devices, smartphones, tablets, and desktops
- Feature rich UI controls for handling complex UI patterns
- Keyboard interaction support and accessibility features
- Full translation support
- Based on open standards like JavaScript , CSS3, and HTML5
- Powerful theme support based on CSS
- Using and including the popular JQuery library

When users access an SAPUI5 app, 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.

Depending on the environment in which SAPUI5 is used, the libraries or your applications can be stored, for example, on an SAP NetWeaver Application Server or an SAP Cloud Platform. Business data can be accessed, for example, using the OData model through an SAP Gateway.

Apps developed with SAPUI5 run in a browser on any device (mobile, tablet or desktop PC).

The preferred way to access business data for your application is using the OData model through an SAP Gateway.

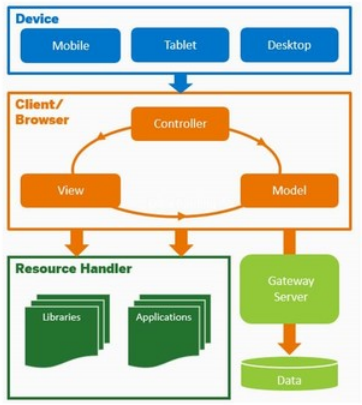


Figure 84: SAPUI5 Architecture

The Model View Controller (MVC) concept is used in SAPUI5 to separate the representation of information from user interaction. This separation facilitates development and the changing of parts independently.

The purpose of data binding in the user interface is to separate the definition of the user interface (view), the data visualized by the application (model), and the code for the business logic for processing the data (controller). The separation has the following advantages.

- It provides better readability, maintainability, and extensibility.
- It allows you to change the view without touching the underlying business logic.
- Several views of the same data can be defined.

Views and controllers often form a 1:1 relationship, but it is also possible to have controllers without a UI. These controllers are called application controllers. It is also possible to create views without controllers. From a technical perspective, a view is an SAPUI5 control and can have, or inherit, a SAPUI5 model.

Views and controllers represent reusable units, and distributed development is highly supported.

Model, view, and controller are assigned the following roles:

- The view is responsible for defining and rendering the UI.
- The model manages the application data.
- The controller reacts to view events and user interaction by modifying the view and model.

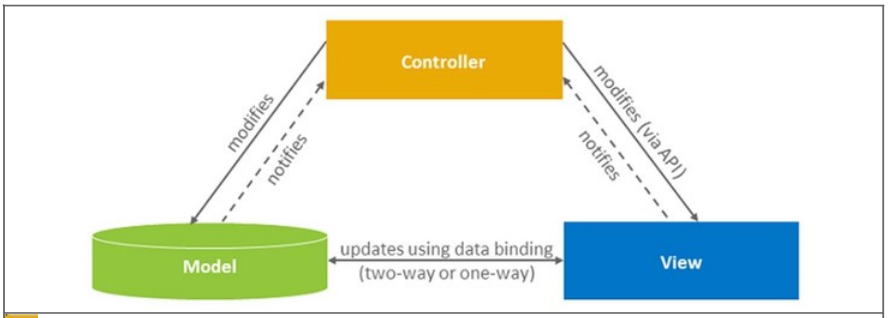


Figure 85: Model View Controller (MVC)

The key UI technologies and tools are those that directly support SAP′s "NEW – RENEW – EMPOWER" UX approach.

Floorplan Manager for Web Dynpro for ABAP

- Floorplan Manager is a framework for model-based declarative creation and adaptation of user interfaces. It is based entirely on Web Dynpro for ABAP.
- The framework provides coherent and guideline-conforming screen definition, with product standards support, including SAP Fiori 2.0 concepts.
- It is a key technology for stateful desktop apps in SAP Business Suite and SAP S/4HANA (SAP Fiori 2.0).

WebClient UI framework (not shown in the figure)

- The WebClient UI framework offers a modern user interface leveraging web 2.0 paradigms for SAP Customer Relationship Management.
- It is harmonized in look and feel, as well as interaction paradigms, with Floorplan Manager and SAP Web Dypro applications.

SAP Web Dynpro based front-end technologies – SAP graphical user interface (GUI) family

- SAP GUI is SAP′s universal client for accessing SAP Dynpro based applications ("transactions").
- The SAP GUI family consists of SAP GUI for Windows, SAP GUI for Java and SAP GUI for HTML. All three SAP GUI have unique attributes that make them especially suited for different user environments.
- It is the most widely adopted UI technology.
- The SAP GUI family is and will be maintained according to SAP′s support level agreements. It will continuously receive focused new investments.
- It is absolutely safe to continue using applications based on SAP GUI as well as developing your own applications or adding enhancements to existing applications.
- Following the SAP′s UX strategy, SAP GUI is not a key UI technology for SAP′s developers to build completely new applications. SAP recommends use of Web Dynpro ABAP / Floorplan Manager (WebClient UI framework for CRM applications) or SAPUI5. However, SAP continues to extend existing Dynpro-based applications with this technology.

SAP Screen Personas

- Facilitate beautification and streamlining of Dynpro-based screens.
- Facilitate simplification of complex Dynpro or Web Dynpro ABAP screens.
- Facilitate creation SAP Fiori designs for custom or standard transactions, including SAP S/4HANA.

Integration is key for a coherent user experience across multiple technologies and UI frameworks. It starts with offering the user one consolidated entry point via a shell and raises the need for consistent UI services.

The SAP Fiori launchpad is designed according to the simple and intuitive SAP Fiori user experience.

SAP Business Client

- SAP Business Client offers a single point of entry to SAP business applications for desktop users.
- It harmonizes access to SAP Fiori, FPM/WDA, and SAP GUI for Windows applications. It is the standard desktop client for the SAP Business Suite.
- It integrates the SAP Fiori user experience: From the SAP Fiori launchpad, SAP GUI transactions are launched using embedded SAP GUI for Microsoft Windows.

The SAP Portal offers solutions for on-premise and cloud-based UI integration.

SAP Cloud Platform Mobile Services

- SAP Cloud Platform mobile service for development and operations is an open, secure, full-spectrum platform that is an enterprise-grade on-demand mobile-app development solution.
- The SAP Mobile Platform Software Development Kit (SDK) simplifies the development of cross-platform apps with powerful enterprise capabilities. It includes offline data synchronization, cross-platform push notifications, connectivity for SAP and third party back-end systems. SDKs support native and hybrid mobile apps on iOS, Android, and Microsoft Windows.

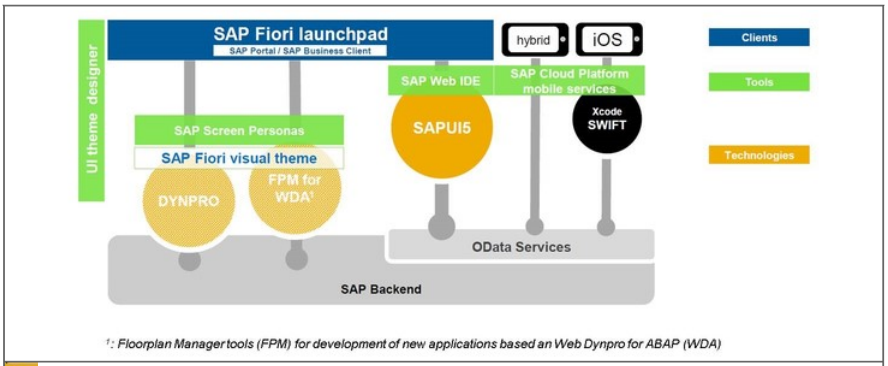


Figure 86: SAP Key UI Technologies and Tools to Deliver SAP Fiori User Experience – A Simplified Overview

LESSON SUMMARY
You should now be able to:

- Understand SAPUI5 at a Glance

Unit 7
Lesson 2

Understanding SAPUI5: Bootstrapping and MVC

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand SAPUI5: bootstrapping and MVC

SAP Fiori App Architecture

When implementing SAP Fiori apps the following basic principles apply:

- SAPUI5 Fiori apps are component-based and have a component controller.
- App descriptor contains application-specific configuration settings.
- SAPUI5 apps can have more than one model and can use different model types.
- Model instantiation is via app descriptor.
- SAPUI5 views are built with controls and can have a view controller to handle events.
- Reusable view content can be grouped in fragments and can be reused in views.
- Different view types can be used in one application.

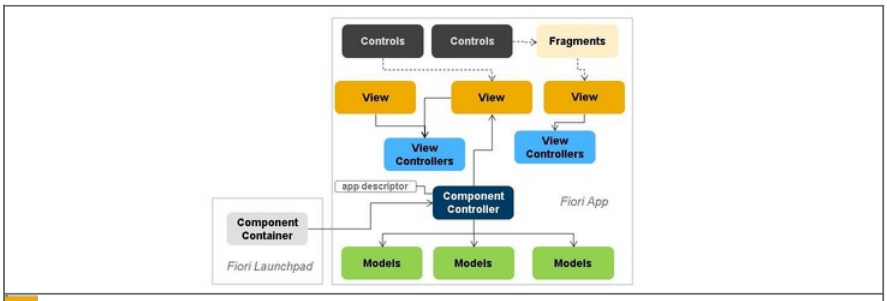


Figure 87: SAPUI5 Apps as Components

We create a `Component.js` file in the src folder that will hold our application setup.

The `Component.js` consists of two parts now: the new metadata section that simply defines a reference to the app descriptor and the previously introduced init function that is called when the component is initialized.

The content of the `manifest.json` file is a configuration object in JSON format that contains all global application settings and parameters. The manifest file is called the Descriptor for Applications, Components, and Libraries and is also often referred to as "app descriptor".

In previous versions of SAPUI5, the metadata section of the component contained the additional configuration settings for the application. These have been completely moved to the app descriptor in version 1.30 and should no longer be configured in the component.

The init function of the component is automatically invoked by SAPUI5 when the component is instantiated. Our component inherits from the base class sap.ui.core.UIComponent and it is obligatory to call the init function of the base class.

In the init function we instantiate our data model and the i18n model like we did before in the app controller. Be aware that the models are set on the component and not on the root view of the component. However, as nested controls automatically inherit the models from their parent controls, the models will be available on the view as well.

The component is named Component.js.

The app descriptor file is named manifest.json.

Together with all assets of the app, these files are located in the app folder (root folder of the app) – Do not move to any subfolder!

Component members:

- metadata: Simply define a reference to the app descriptor (manifest.json).
- init function: Called when the component is initialized.
- createContent function: You can optionally override the initialization of the root view (root view defined in the app descriptor).

```
sap.ui.define([
    "sap/ui/core/UIComponent",
    "sap/ui/Device",
    "sap/training/flight/model/models"
], function(UIComponent, Device, models) {
    "use strict";

    return UIComponent.extend("sap.training.flight.Component", {

        metadata: {
            manifest: "json"
        },

        /**
         * The component is initialized by UI5 automatically during the
         * startup of the app and calls the init method once.
         * @public
         * @override
         */
        init: function() {
            // call the base component's init function
            UIComponent.prototype.init.apply(this, arguments);


            // set the device model
            this.setModel(models.createDeviceModel(), "device");
            this.getRouter().initialize();
        }
    });

});
```

Figure 88: Create a New Component – Component.js

Aspects of the **Code** Document

- Descriptor for Applications, Components, and Libraries also known as "app descriptor"
- Contains the file called `manifest.json`
- Inspired from W3C manifest with own SAP namespaces

- Contains all global application settings and parameters, for example, the app id, name, version, the used data sources, and UI5 dependencies.
- Created with the application at development, delivered within the application and read-only after delivery.
- Existence is declared in component metadata.
- Created by the developer either manually or with the help of a wizard or template in SAP Web IDE.
- Part of SAP Fiori Guidelines

Language-Dependent Values in App Descriptor

- Text symbols are via handlebars like syntax: {{title}}
- When referring to text symbols from the properties file location under sap.app or the i18n attribute, the default is "i18n/i18n.properties".

Evaluation of the language dependence is done either on the server API or client based on the sap.app/i18n attribute, where the same language fallback applies as for UI5. UI5 core provides preprocessing for the language resolution.

Code Document

- Descriptor for Applications, Components, and Libraries also known an "app descriptor"
- File called manifest.json
- Contains all global application settings and parameters such as the app ID, name, version, used data sources and SAPUI5 dependencies
- Created with the application at development, delivered within the application and read-only after delivery
- Part of the Fiori Guidelines

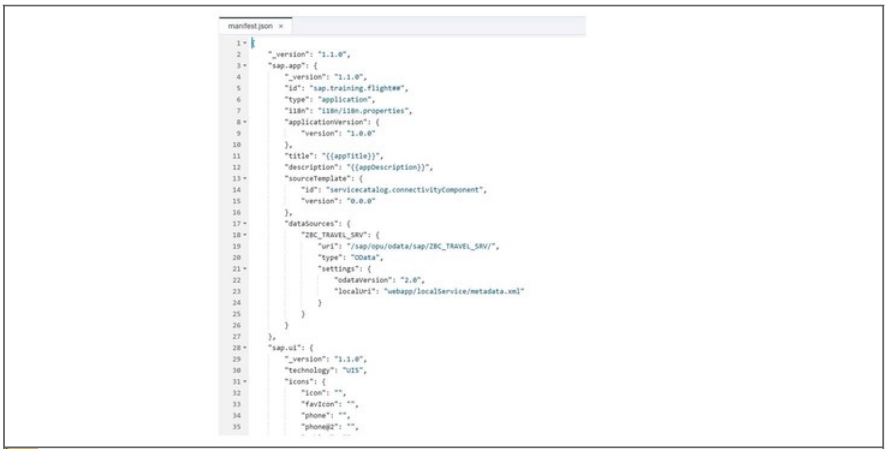


Figure 89: App Descriptor "code" Document

LESSON SUMMARY
You should now be able to:

- Understand SAPUI5: bootstrapping and MVC

Unit 7
Lesson 3

Understanding SAPUI5: Routing and Navigation

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand SAPUI5: Routing and Navigation

Routing and Navigating

Typical mobile applications are often composed of a number of pages, views, and screens between which the user can navigate. Two of these are now added to the App.

> Note:
> While sap.m.Page controls can be used as pages, and the aggregation is called "pages", other controls could be used as well.

Many mobile apps are composed of a few pages and the user can drill-down to detail pages and go back up again.

What resource or page was requested is determined at the front-end and it then displays in an appropriate way such as slide, show, fade, or flip.

SAPUI5 supports this pattern by providing the sap.m.App control which can handle pages and navigation between them. Internally, sap.m.App inherits the navigation capabilities from the sap.m.NavContainer control.

Typical mobile application structure:

- App with multiple screens
- Master-detail screens with split apps
- Drilldown navigation
- Edit and display
- Back and forward navigation
- Entries in browser History
- Deep links

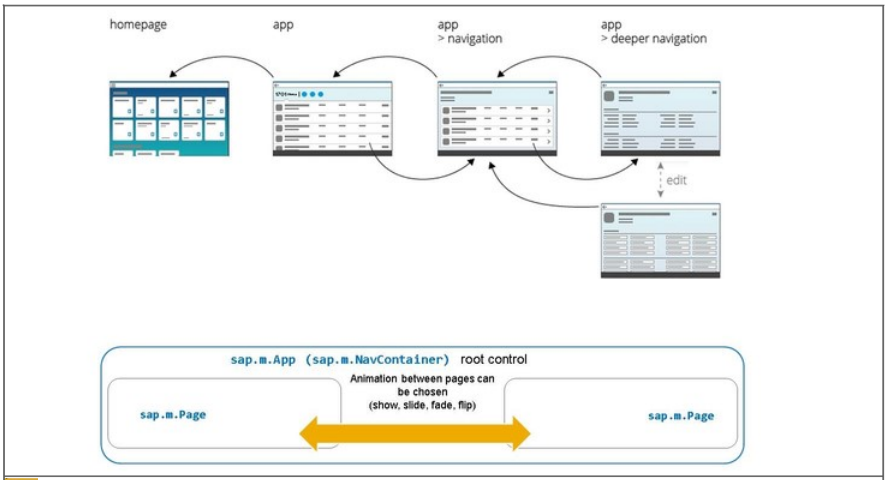


Figure 90: Typical SAPUI5 Mobile Usage

Modern web applications load data or changes in the document object model (DOM) without leaving the original page.

We can take advantage of the URL hash. By inserting a hash character into the URL, we can pick it up and use it to decide what page to display.

SAPUI5 Routing uses a hash-based navigation:

- The navigation is done by changing the hash and using a string as the name of the route.
- The browser does not reload the page.
- Navigation can be done with or without browser history handling.
- SAPUI5 routing supports bookmarks and or deep links.

We want to consider routing as a set of features that dispatch URLs to views of an app and wires the views together.

In contrast, navigation allows browsing through pages based on the routing configuration and allows parameters to be passed to targets.

The SAPUI5 core library provides an API for navigating with URL fragment changes. When an app is running within the SAP Fiori launchpad, the implementation will consider the shell-specific navigation parts and allow both inner-app and app-to-app navigation by similar means.

The navigation API provides the following functionality:

- Directly triggers navigation to a navigation target with parameters
- Registers navigation routes and listeners for fragment changes to restore a specific state

SAPUI5 navigation features:

- sap.m.routing.Router - SAPUI5 router class to load the pages and update the URL, (change and replace hash)

- Routing configuration in the app descriptor
- Navigation events
- Animation between pages, (show, slide, fade, flip)

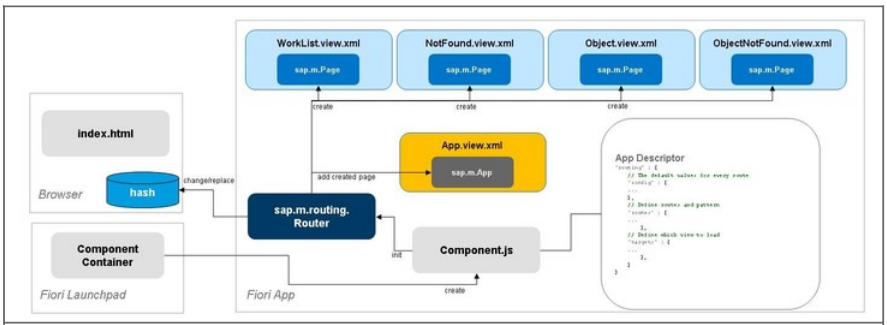


Figure 91: Inner-app Navigation

LESSON SUMMARY
You should now be able to:

- Understand SAPUI5: Routing and Navigation

Unit 7
Lesson 4

Understanding SAPUI5: Margins, Paddings, and CSS

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand SAPUI5: Margins, paddings, and CSS

Working with SAPUI5 Cascading Style Sheets (CSS) Classes

Consistency in the use of margins and padding provides a uniform format. It also facilitates fast update of documents in a system and responsive display across platforms.

As of SAPUI5 1.28, SAPUI5 libraries let you influence spacing in between controls and within containers by using predefined CSS margin/padding classes.

Related features in SAPUI5:

- List of predefined CSS classes to add full, single-sided, two-sided and responsive margins, or remove margins
- All predefined margin CSS classes support right-to-left (RTL) languages
- Predefined CSS classes to add (responsive) or remove content padding
- Benefits include consistent spacing around controls and within containers across layouts

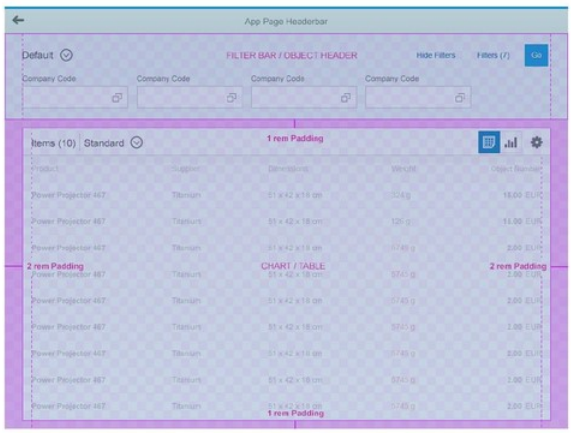


Figure 92: Margin and Padding CSS Class Features

Responsive padding and margins are predefined in SAPUI5.

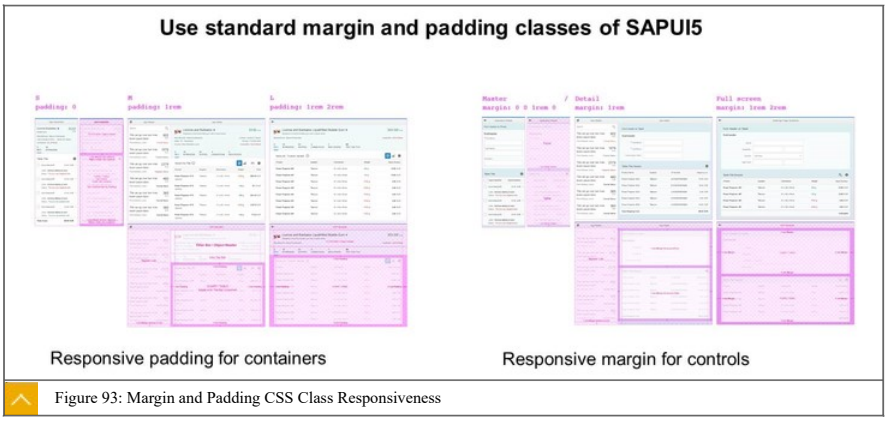


Figure 93: Margin and Padding CSS Class Responsiveness

For single-sided margins, choose a size and a direction.

Table 1: Margin Size

Size	Px
Tiny	8
Small	16
Medium	32
Large	48

Pick a direction named Begin, End, Top or Bottom, where Begin is left and End is right and the reverse in right to left mode.

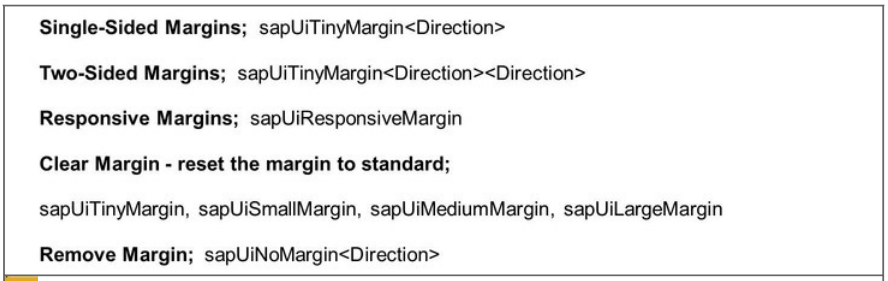


Figure 94: Pre-defined CSS Margin Classes

LESSON SUMMARY
You should now be able to:

- Understand SAPUI5: Margins, paddings, and CSS

Unit 7
Lesson 5

Understanding SAPUI5: OData and the ODataModel

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand SAPUI5: OData and the ODataModel

ODATA and the ODATA Model

To use SAPUI5, it's important to understand ODATA and the ODATA model.

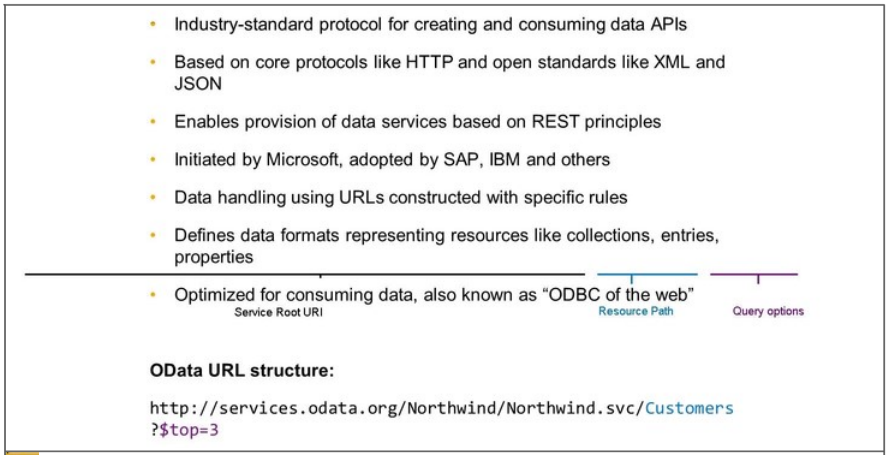
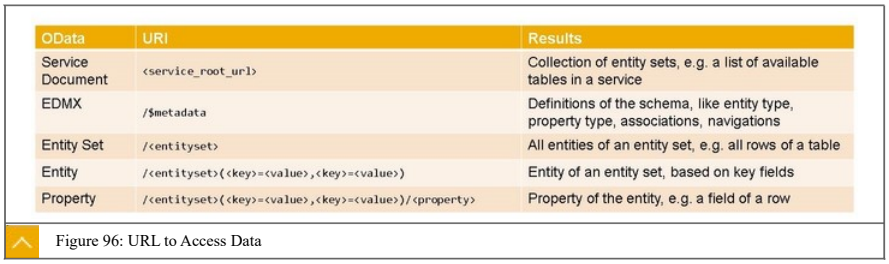


Figure 95: Open Data Protocol

URLs are used to navigate and target data.



OData	URI	Results
Service Document	<service_root_url>	Collection of entity sets, e.g. a list of available tables in a service
EDMX	/$metadata	Definitions of the schema, like entity type, property type, associations, navigations
Entity Set	/<entityset>	All entities of an entity set, e.g. all rows of a table
Entity	/<entityset>(<key>=<value>,<key>=<value>)	Entity of an entity set, based on key fields
Property	/<entityset>(<key>=<value>,<key>=<value>)/<property>	Property of the entity, e.g. a field of a row

Figure 96: URL to Access Data

ODATA model version 2 is implemented in SAPUI5.

- The OData model enables binding of control properties and aggregations to data from a remote server.
- The complete dataset is stored on the server and only the requested fields are transferred to the client.
- SAPUI5 supports multiple OData models, but the SAP Fiori Guidelines state you should use OData model version 2 `sap.ui.model.odata.v2.ODataModel`
- All SAPUI5 OData models are currently based on OData specification version 2.0 (with support of annotations from OData specification version 4.0).

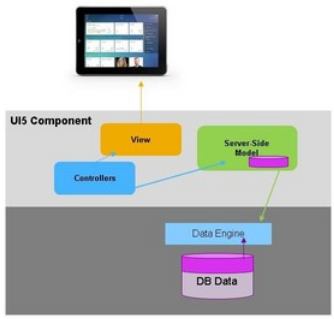


Figure 97: OData Model in SAPUI5

The Figure, ABAP programming model for SAP Fiori, illustrates the building blocks of the ABAP programming model for SAP Fiori.

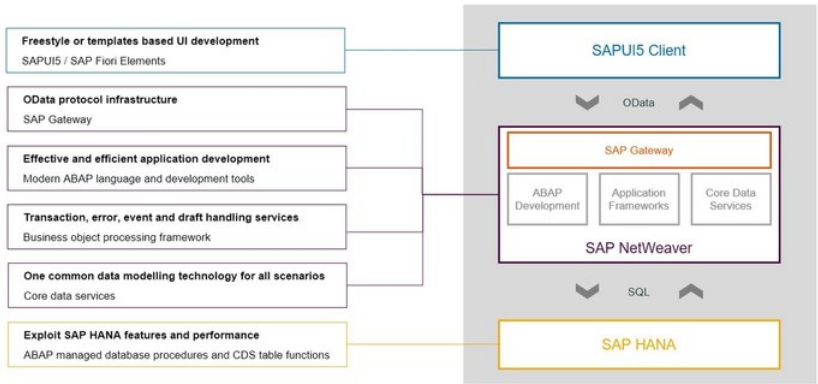


Figure 98: ABAP Programming Model for SAP Fiori

HTML5 applications can connect to one or multiple **REST services**.

The URL to the REST service is configured in a **destination**.

The mapping between an application URL and a destination is defined in the **application descriptor** called `neo-app.json`.

The application descriptor must be located in the **root folder of the repository**.

Destinations are configured using the **SAP HANA Cloud Cockpit**.

`neo-app.json`

```
"routes": [
  {
    "path": "/destinations/northwind",
    "target": {
              "type": "destination",
              "name": "northwind_service",
          "entryPath": "/destinations/northwind"
                    },
    "description": "Northwind"
  }
]
```

Figure 99: Back-end Routing in the Application Descriptor neo-app.json

The neo-app.json file contains all project settings for SAP Web IDE and is created in the root folder of your project. It is a JSON format file consisting of multiple configuration keys. The most important setting for you to configure is the path where the SAPUI5 runtime is located when starting the app.

You do this using the routes key and defining an array of resource objects. For running an SAPUI5 tutorial, you only need two entries - one that configures SAPUI5 to be available with the path /resources, and another one that configures the test resources needed for the SAP Fiori launchpad integration with the path /test-resources.

Create two configuration objects that contain a path, a target, and a description attribute with more configuration settings. The path and the entryPath values will point to the location on the server where the SAPUI5 resources will be stored.

SAP Web IDE reads these settings automatically when running the app.

It is quite common that a developer of an SAP Fiori application starts before the development of data service is finished to use real back-end data. Therefore, it is necessary to work as a UI-developer with local mock dataApplications can be connected to the real data server at a later stage of development. Such mock data can also be useful for testing and problem solving in cases where access to the data service is not available or requires effort to set up.

- Isolates your app from the back-end
- Simulates OData back-end calls with Sinon.JS
- Transparent to data binding
- Supports randomly generated data, (based on the service metadata)
- Supports mock data provided in JSON files, (use same module path as for metadata.xml)
- Each OData service needs its own mock server
- Can be extended with additional calls to mock function import calls

```
sap.ui.define([ "sap/ui/core/util/MockServer" ],
function (MockServer) {  "use strict";
    var oMockServer;
    return {
        _sServiceUrl : "/sap/opu/odata/sap/EHM_REF_APPS_PROD_MAN_SRV/", // service url to mock
        _sModulePath : "myproject.service", // module path containing metadata.xml

        init : function () {

            oMockServer = new MockServer({ rootUri: this._sServiceUrl });

            // configure mock server with a delay
            MockServer.config({
                autoRespond : true,
                autoRespondAfter : 1000
            });
            // simulate
            var sPath = jQuery.sap.getModulePath(this._sModulePath);
            oMockServer.simulate(sPath + "/metadata.xml",  {
                sMockdataBaseUrl : sPath,
                bGenerateMissingMockData : true
            });
            // Start
            oMockServer.start();
            jQuery.sap.log.info("Running the app with mock data");
        }
    };
});
```

Figure 100: Mock Server

To use the internal provided mock server of the SAP Web IDE you can configure a runtime configuration server on the SAP Web IDE to provide data and trial the functionality of your applications.

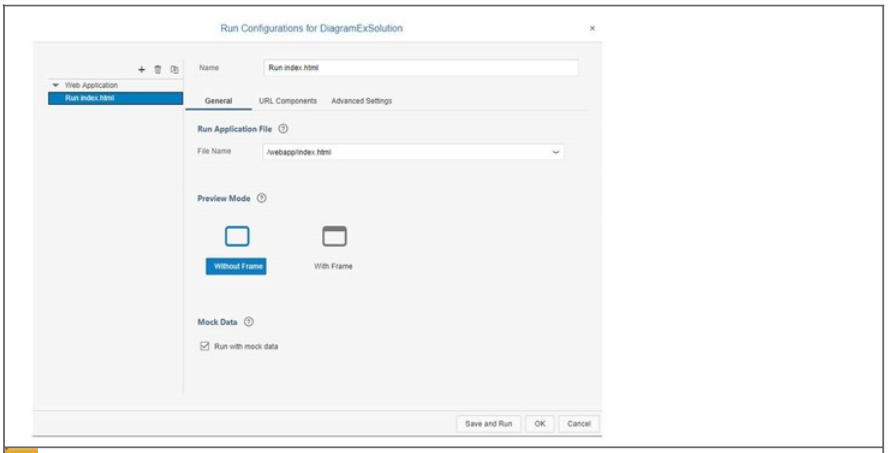


Figure 101: Working with a Mock Server Web IDE

LESSON SUMMARY

You should now be able to:

- Understand SAPUI5: OData and the ODataModel

Unit 7
Lesson 6

Understanding SAPUI5: Visualizing Business Data

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand SAPUI5: Visualizing business data

Visualizing Business Data

Charts are visualizing quantitative information to understand business aspects more easily.

Charts can draw attention to key information and patterns when they are well designed.

Namespaces are sap.viz and sap.suite.ui.commons.

Documentation can be found at: https://sapui5.netweaver.ondemand.com/docs/vizdocs/index.html .

Figure 102: Charts UI Controls

These controls are aggregated in the file SAP.suite.ui.commons.ChartContainer.

Constructor Detail

new sap.suite.ui.commons.ChartContainer(sId?, mSettings?)

Constructor for a new ChartContainer.

Accepts an object literal mSettings that defines initial property values, aggregated and associated objects as well as event handlers.

If the name of a setting is ambiguous (e.g. a property has the same name as an event), then the framework assumes property, aggregation, association, event in that order. To override this automatic resolution, one of the prefixes "aggregation:", "association:" or "event:" can be added to the name of the setting (such a prefixed name must be enclosed in single or double quotes).

The supported settings are:

- Properties
  - showPersonalization : boolean (default: false)
  - showFullScreen : boolean (default: false)
  - fullScreen : boolean (default: false)
  - showLegend : boolean (default: true)
  - title : string (default: '')
  - selectorGroupLabel : string
  - autoAdjustHeight : boolean (default: false)
  - showZoom : boolean (default: true)
  - showLegendButton : boolean (default: true)
- Aggregations
  - dimensionSelectors : sap.ui.core.Control[]
  - content **(default aggregation)** : sap.suite.ui.commons.ChartContainerContent[]
  - customIcons : sap.ui.core.Icon[]
- Associations
- Events
  - personalizationPress : fnListenerFunction or [fnListenerFunction, oListenerObject] or [oData, fnListenerFunction, oListenerObject]
  - contentChange : fnListenerFunction or [fnListenerFunction, oListenerObject] or [oData, fnListenerFunction, oListenerObject]
  - customZoomInPress : fnListenerFunction or [fnListenerFunction, oListenerObject] or [oData, fnListenerFunction, oListenerObject]
  - customZoomOutPress : fnListenerFunction or [fnListenerFunction, oListenerObject] or [oData, fnListenerFunction, oListenerObject]

Parameters:

(string) **sId?** id for the new control, generated automatically if no id is given

(object) **mSettings?** initial settings for the new control

Figure 103: Excerpt of the API-documentation of sap.suite.ui.commons.ChartContainer

```
<suite:ChartContainer autoAdjustHeight="true" contentChange="onContentChange" id="idChartContainer"
    showFullScreen="true" showLegend="true" showPersonalization="false" title="{i18n>diagramTitle}">
    <suite:dimensionSelectors>
        <Select id="idDimSelector" visible="true" change="onChange"></Select>
    </suite:dimensionSelectors>
    <suite:content>
        <suite:ChartContainerContent icon="sap-icon://line-chart" title="{i18n>lineChartTitle}">
            <suite:content>
                <viz:Popover id="idLineChartPopover"></viz:Popover>
                <viz:VizFrame id="idLineChartVizFrame" height="9.1rem" width="100%" uiConfig="{applicationSet:'fiori'}"></viz:VizFrame>
            </suite:content>
        </suite:ChartContainerContent>
        <suite:ChartContainerContent icon="sap-icon://vertical-bar-chart" title="{i18n>barChartTitle}">
            <suite:content>
                <viz:VizFrame id="idBarChartVizFrame" height="9.1rem" width="100%" uiConfig="{applicationSet:'fiori'}"></viz:VizFrame>
            </suite:content>
        </suite:ChartContainerContent>
        <suite:ChartContainerContent icon="sap-icon://table-chart" title="{i18n>tableTitle}">
            <suite:content>
                <Table id="idTable"></Table>
            </suite:content>
        </suite:ChartContainerContent>
    </suite:content>
</suite:ChartContainer>
```

Figure 104: Implementation of a ChartContainer Control

sap.viz.ui5.controls.VizFrame

VizFrame is a viz control that manages a visualization′s initialization, layout, feeding, customization and interactions.

Important aspects:

- DataSet – Reference to the entity set of the data model
- DimensionDefinition/MeasureDefinition – Defines the dimensions of the diagram
- Feed – Specifies the Axis

A FlattenedDataset contains data in a tabular format

The attribute data contains the path for the aggregation binding

```
_createDataSet: function () {
    var oDataset = new FlattenedDataset({
        data: {
            path: "SalesModel>/SalesFigures"
        }
    });
    return oDataset;
},
```

Figure 105: Implementation of the Helper Function _createDataSet

DimensionDefinition and MeasureDefinition are classes to specify the binding for the x and y axis

```
_createDataSetMap: function () {
    this.dataSetMap = {};
    this.dataSetMap.productDim = new DimensionDefinition({
        name: "Products",
        value: "{SalesModel>PRODUCT_NAME}"
    });
    this.dataSetMap.subregionDim = new DimensionDefinition({
        name: "Sub_Region_Name",
        value: "{SalesModel>SUB_REGION_NAME}"
    });
    this.dataSetMap.salesAmountMeasure = new MeasureDefinition({
        name: "SalesAmount",
        value: "{SalesModel>SALES_AMOUNT}"
    });

},
```

Figure 106: Implementation of the Helper Function _createDataSetMap

The FeedItem specifies what data of the dataset is shown inside the diagram.

```
_createFeedMap: function () {
    this.feedMap = {};
    this.feedMap.salesAmount = new FeedItem({
        "uid": "valueAxis",
        "type": "Measure",
        "values": ["SalesAmount"]
    });

    this.feedMap.products = new FeedItem({
        "uid": "categoryAxis",
        "type": "Dimension",
        "values": ["Products"]
    });

    this.feedMap.subregion = new FeedItem({
        "uid": "categoryAxis",
        "type": "Dimension",
        "values": ["Sub_Region_Name"]
    });

    this.feedMap.productsSubregion = new FeedItem({
        "uid": "categoryAxis",
        "type": "Dimension",
        "values": ["Products", "Sub_Region_Name"]
    });
},
```

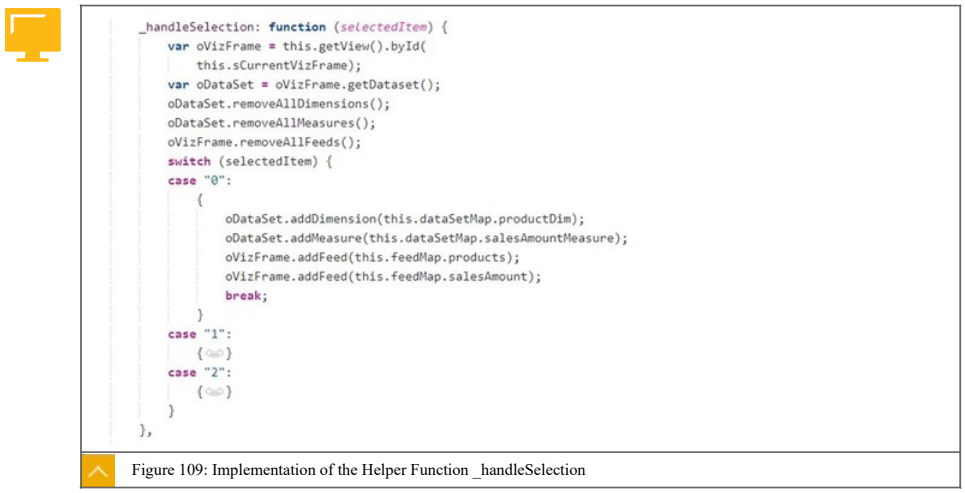
Figure 107: Implementation of the Helper Function _createFeedMap

The VizFrame-Object gets the binding through the setDataset-Method

```
_createLineDiagram: function () {
    var oVizFrame = this.getView().byId("idLineChartVizFrame");
    var oPop = this.getView().byId("idLineChartPopover");
    oVizFrame.setDataset(this._createDataSet());
    this._handleSelection(this.sCurrentSelectedDimension);
    oVizFrame.setVizProperties({
        plotArea: {
            dataLabel: {
                visible: true
            }
        },
        title: {
            visible: false
        }
    });
    oVizFrame.setVizType('line');
    oPop.connect(oVizFrame.getVizUid());
},
_createColumnChart: function () {
    var oVizFrame = this.getView().byId("idBarChartVizFrame");
    oVizFrame.setDataset(this._createDataSet());
    oVizFrame.setVizProperties({
        plotArea: {
            dataLabel: {
                visible: true
            }
        },
        title: {
            visible: false
        }
    });
    oVizFrame.setVizType('column');
},
```

Figure 108: Implementation of the Helper Function __createLineDiagram

It is possible to change the data basis for the diagram based on users selection.



```
_handleSelection: function (selectedItem) {
    var oVizFrame = this.getView().byId(
        this.sCurrentVizFrame);
    var oDataSet = oVizFrame.getDataset();
    oDataSet.removeAllDimensions();
    oDataSet.removeAllMeasures();
    oVizFrame.removeAllFeeds();
    switch (selectedItem) {
    case "0":
        {
            oDataSet.addDimension(this.dataSetMap.productDim);
            oDataSet.addMeasure(this.dataSetMap.salesAmountMeasure);
            oVizFrame.addFeed(this.feedMap.products);
            oVizFrame.addFeed(this.feedMap.salesAmount);
            break;
        }
    case "1":
        {}
    case "2":
        {}
    }
},
```

Figure 109: Implementation of the Helper Function _handleSelection

LESSON SUMMARY

You should now be able to:

- Understand SAPUI5: Visualizing business data

8. What is the purpose of the neo-app.json file located in the project root folder of your SAP Web IDE project?

   Choose the correct answer.

   - ☐ **A** It contains the OData service configuration parameters used at runtime.
   - ☐ **B** It contains the mapping between an application URL and a destination configured in the SAP Cloud Platform.
   - ☐ **C** It contains the configuration for the mock server.
   - ☐ **D** It contains the mapping configuration of the front-end and back-end server URLs.

9. What aggregations are defined by the sap.suite.ui.commons.ChartContainer?

   Choose the correct answers.

   - ☐ **A** content
   - ☐ **B** container
   - ☐ **C** dimensionSelector
   - ☐ **D** customicons
   - ☐ **E** legend

10. What main aspects does the developer have to configure for a sap.viz.ui5.controls.VizFrame control?

    Choose the correct answers.

    - ☐ **A** Dataset
    - ☐ **B** Datacontainer
    - ☐ **C** DimensionDefinition
    - ☐ **D** MeasureDefinition
    - ☐ **E** Feed

Unit 7

Learning Assessment - Answers

1. What are the main artefacts of an SAPUI5 application?

   Choose the correct answers.

	A CSS
X	**B** Views
X	**C** Controllers
X	**D** Models
	E controls

   You are correct! SAPUI5 is based on the Model-View-Controller paradigm, so views, controllers, and models are the main artefacts of a SAPUI5 application.

2. What are the tasks of the controller in the Model-View-Controller implementation?

   Choose the correct answers.

	A Updates using data binding
X	**B** Modifies the model
X	**C** Modifies the view via API
	D Holds the business data
	E Contains the UI implementation

   You are correct! The tasks of the controller in the Model-View-Controller paradigm are modifying the model and modifying the view using the API.

3. Which of the following statements are talking about the SAP Fiori app architecture?

   Choose the correct answers.

   - [X] **A** SAP Fiori apps are implemented using the component-based approach.
   - [ ] **B** SAP Fiori apps have a index.html file that is used for SAP Fiori launchpad integration.
   - [X] **C** SAP Fiori apps need to have an app descriptor that describes the applications metadata.
   - [ ] **D** SAP Fiori apps should use the SAPUI5 core as a place to hold app data models.

   You are correct! SAP Fiori apps are implemented using the component-based approach. Each component should be described by a set of metadata and have an application descriptor.

4. SAPUI5 provides two different router implementations. What router class is used in SAP Fiori apps as a router or as a router base class, when you want to implement your own router?

   Choose the correct answer.

   - [X] **A** sap.m.routing.Router
   - [ ] **B** sap.ui.core.routing.Router
   - [ ] **C** sap.m.Router
   - [ ] **D** sap.routing.Router

   You are correct! If you need standard routing, you will use the sap.m.routing.Router implementation of SAPUI5. This class is also used as a base class when you add special, application-specific routing features.

5. What are the main configuration objects of the routing configuration?

   Choose the correct answers.

   - [X] **A** config
   - [ ] **B** routing
   - [X] **C** routes
   - [X] **D** targets

   You are correct! The routing configuration consists of three main configuration objects: the config, the routes, and the targets.

6. What is the main benefit to using predefined CSS margin/padding classes of SAPUI5?

   Choose the correct answer.

   - [X] **A** Consistent spacing
   - [ ] **B** Consistent borders
   - [ ] **C** Consistent width
   - [ ] **D** Consistent height

   You are correct! Using the predefined CSS margin/padding classes of SAPUI5 will provide consistent spacing around controls and containers across layouts.

7. Which of the following artefacts are contained in the OData URL structure?

   Choose the correct answers.

   - [X] **A** Service Root URI
   - [ ] **B** Header data
   - [X] **C** Resource Path
   - [X] **D** Query Options

   You are correct! The OData URL contains the service root URL. It points to the service document of the OData service, followed by the resources path, and may contain optional query options.

8. What is the purpose of the neo-app.json file located in the project root folder of your SAP Web IDE project?

   Choose the correct answer.

   - [ ] **A** It contains the OData service configuration parameters used at runtime.
   - [X] **B** It contains the mapping between an application URL and a destination configured in the SAP Cloud Platform.
   - [ ] **C** It contains the configuration for the mock server.
   - [ ] **D** It contains the mapping configuration of the front-end and back-end server URLs.

   You are correct! The neo-app.json file contains the mapping between an application URL and a destination configured in the SAP Cloud Platform.

9. What aggregations are defined by the sap.suite.ui.commons.ChartContainer?

   Choose the correct answers.

   - [X] **A** content
   - [ ] **B** container
   - [X] **C** dimensionSelector
   - [X] **D** customicons
   - [ ] **E** legend

   You are correct! The sap.suite.ui.commons.ChartContainer defines three aggregations: The content aggregation as default aggregation, the dimensionSelector and the customIcons aggregation.

10. What main aspects does the developer have to configure for a sap.viz.ui5.controls.VizFrame control?

    Choose the correct answers.

    - [X] **A** Dataset
    - [ ] **B** Datacontainer
    - [X] **C** DimensionDefinition
    - [X] **D** MeasureDefinition
    - [X] **E** Feed

    You are correct! A VizFrame control needs to know where the data is coming from, this is done by implementing a DataSet referencing the data source. A DimensionDefinition and a MeasureDefinition are also needed to define which data fields of the data source are used. A feed is used to declare which data source should be used as an axis for the diagram.

UNIT 8

Golden Rules of SAPUI5 Development

Lesson 1

Knowing the Golden Rules of SAPUI5 Development 106

UNIT OBJECTIVES

- Know the golden rules of SAPUI5 development

Unit 8
Lesson 1

Knowing the Golden Rules of SAPUI5 Development

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Know the golden rules of SAPUI5 development

Golden Rules of SAPUI5 Development

Follow these golden rules when developing for SAP Fiori SAPUI5.

1. SAP Fiori apps must have an approved UX design (GR-01)
2. SAP Fiori UIs are built with SAPUI5 (GR-02)
3. SAP Fiori UIs and OData services must be defined in different software components (GR-05)
4. SAP Fiori apps are based on OData services (GR-03)
5. One SAP Fiori app - one OData service (GR-04)
6. Every SAP Fiori app is defined by a set of metadata (GR-06)
7. No custom CSS is allowed for SAP Fiori apps (GR-07)
8. Every SAP Fiori app must run as a web app (GR-08)
9. Every SAP Fiori app must run in the SAP Fiori Launchpad (GR-09)
10. Every SAP Fiori app must run on mobile devices along the native app paradigm (GR-10)

The Design-led Development process is valid for all UI scenarios.

Ensure there is a common and holistic understanding of which end-user tasks are to be solved and which app should be built before starting.

Design has to comply with the Fiori Design Guidelines (or your own).

Designs have to only use existing controls and patterns.

In general, new SAP Fiori applications should be built using SAPUI5 UI technology.

The recommended approach for standard SAP S/4HANA applications is to make use of Smart Templates (metadata driven UIs).

For pixel perfect applications that do not follow a pattern-like approach, use UI5 freestyle.

Smart Controls are strongly recommended if the back-end support annotates their use.

For pixel perfect applications that only have minor deviations from Smart Template applications, a combination is recommended.

Technologies like SAP Web Dynpro ABAP, SAP Floorplan Manager and SAP Web GUI are still supported. Use these for existing apps that are not built from scratch.

There is a one-to-one relationship between the OData service and the app.

The individual entity-relationship model is tailored to the data needs of the SAP Fiori app.

All, and only, the data needed by the application are delivered by one OData-service.

The above restriction simplifies the life-cycle of an SAP Fiori app and its data access.

An OData service is required for each app but there are other considerations.

- An app must be able to use several OData services, indirectly view reusable app parts, or direct data binding at run-time derived from metadata.
- SAP Fiori apps also rely on the central UI2 services.
- Message long texts are provided by one central reuse service.
- Value helps may be implemented as a separate service.

Two different deployment options are possible in the system landscape for SAP Fiori Apps.

- Local deployment: The UI, the gateway, and the business back end are deployed on the business back-end system.
- Remote deployment: The UI and the gateway are deployed on a different system from the business back-end system.

To support remote deployment, it is mandatory to package all UI relevant artefact in a separate software component.

The OData service development package should be treated as an add-on.

SAP Fiori apps define their set of metadata in an app descriptor (manifest.json) file.

App descriptor makes it easy to orchestrate and automate the development delivery..

Descriptor is mandatory for apps shipped on innovation stack (UI5 >= 1.30) app.

Custom CSS is not allowed for SAP Fiori apps.

Note:
The HTML/CSS generated by UI5 is not part of the public API and may change in patch and minor releases.

Overriding UI5 with CSS in SAP apps bears the risk of breaking the application on updating UI5.

Use the standard CSS classes of SAPUI5, such as for margins, paddings or visibility on various devices.

Make use of the UI theme designer.

SAP Fiori apps must build in platform independence.

They must run from out of the box without platform specific adjustments.

It is usually a bug in the app when an app does not run on a particular platform.

Avoid:

- Absolute URLs generated in the front- or back-end
- Hard dependencies between apps
- Use of private UI5 properties or functions
- Calling UI2 services directly instead of using the APIs
- Use of platform-specific functionality without an availability check

Every SAP Fiori app must run on mobile devices in the native app paradigm.

SAP Fiori apps can run on a mobile device within the browser, SAP Fiori Client and as self-contained native-packaged apps.

If an app wants to benefit from native features, it must run inside Fiori Client or be deployed as a Packaged App.

Running the app inside the SAP Fiori Client is the most generic way apps can run on mobile devices and still have access to native features.

SAP Fiori apps can be pre-packaged as self contained native apps. All static web resources are bundled into a Cordova container.

SAP Fiori apps must be implemented as self-contained UI5 Components.

- noShellIndex.html: local, isolated test without shell services
- FioriSandbox.html: local, integrated test with mock shell services
- FioriLaunchpad.html: productive usage

LESSON SUMMARY
You should now be able to:

- Know the golden rules of SAPUI5 development

Unit 8

Learning Assessment

1. What is the idea behind the golden rule of having one common UX?

   Choose the correct answer.

   - [ ] **A** Having a common UX reduces training costs and increases the quality of data.
   - [ ] **B** Having a common UX reduces the overall total cost of ownership.
   - [ ] **C** Having a common UX reduces development costs.

2. Why is it advisable to not use a custom CSS file in your SAP UI5 applications?

   Choose the correct answer.

   - [ ] **A** SAPUI5 has problems with custom CSS.
   - [ ] **B** Having one central design created with the theme designer is very efficient. When something needs to be changed at the client interface, it is easier to handle.
   - [ ] **C** Time spent designing modified CSS is expensive.

3. Which of the following should you avoid with regard to the golden rule “Every SAP Fiori app must run as a web app”?

   Choose the correct answers.

   - [ ] **A** Absolute URLs generated in front or back-end.
   - [ ] **B** Using SAPUI5
   - [ ] **C** Hard dependencies between apps
   - [ ] **D** Calling UI2 services directly instead of using APIs
   - [ ] **E** Use of platform-specific functionality without availability check

Unit 8

Learning Assessment - Answers

1. What is the idea behind the golden rule of having one common UX?

   Choose the correct answer.

   - [X] **A** Having a common UX reduces training costs and increases the quality of data.
   - [ ] **B** Having a common UX reduces the overall total cost of ownership.
   - [ ] **C** Having a common UX reduces development costs.

   You are correct! Having a common UX reduces training costs and increases the quality of data.

2. Why is it advisable to not use a custom CSS file in your SAP UI5 applications?

   Choose the correct answer.

   - [ ] **A** SAPUI5 has problems with custom CSS.
   - [X] **B** Having one central design created with the theme designer is very efficient. When something needs to be changed at the client interface, it is easier to handle.
   - [ ] **C** Time spent designing modified CSS is expensive.

   You are correct! Having a central design created with the SAP Theme Designer is much more efficient for possible future changes. Adapting one central theme that is applied to various applications is cheaper and more efficient.

3. Which of the following should you avoid with regard to the golden rule "Every SAP Fiori app must run as a web app"?

   Choose the correct answers.

   - [X] **A** Absolute URLs generated in front or back-end.
   - [ ] **B** Using SAPUI5
   - [X] **C** Hard dependencies between apps
   - [X] **D** Calling UI2 services directly instead of using APIs
   - [X] **E** Use of platform-specific functionality without availability check

   You are correct! You should avoid use of platform-specific functionality without an availability check. Avoid absolute URLs generated in the front or back-end. Avoid calling UI2 services directly instead of using APIs.

UNIT 9 SAP Fiori Launchpad

UNIT OBJECTIVES

- Understand the SAP Fiori launchpad
- Understand the technical perspective of SAP Fiori launchpad

Unit 9
Lesson 1

Understanding the SAP Fiori Launchpad

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the SAP Fiori launchpad

SAP Fiori Launchpad

General Concepts of the SAP Fiori launchpad.

The SAP Fiori launchpad home page is the first page that users see after they have logged in. It is the main entry point to SAP Fiori apps on both, mobile or desktop devices.

The launchpad home page presents tiles that allow the launch of apps, and may show additional information. The page can be personalized. Tiles can be added or removed and bundled in groups.

The home page is provided by the SAP Fiori launchpad. Apps use this home page and do not design their own.

Apps can use the following services offered by the SAP Fiori launchpad:

- Settings (apps only): Each app can provide app-specific settings to the launchpad.
- User Preferences: This service provides details about the user currently logged in to the app. In addition, it offers theme selection.
- Contact Support: You can offer a Contact Support option as a channel for user incidents. Note, that this option is only available if the customer configures the support setup.
- Give Feedback: This service allows users to give feedback on the app. Note, that this option is only available if the customer configures the feedback setup.
- About (apps only): This option is automatically available for all apps. It provides technical details about the app.

Apps can use the following services offered by the SAP Fiori launchpad:

- Log In / SSO / Log Out: All aspects of logging in and out are handled by the launchpad. If single-sign-on (SSO) is used, no user password is required. If SSO is not used, the launchpad provides a login screen.
- Save as Tile: This service allows users to save a snapshot of the app as a tile on the launchpad. The tile bookmarks the current state of the app.
- Navigation: SAP Fiori Launchpad handles all navigation between apps.

More detailed information is available by navigating to the SAP Help Portal, and searching for keywords: User interface technology 7.51. Select required SP level and navigate through the desired sections.

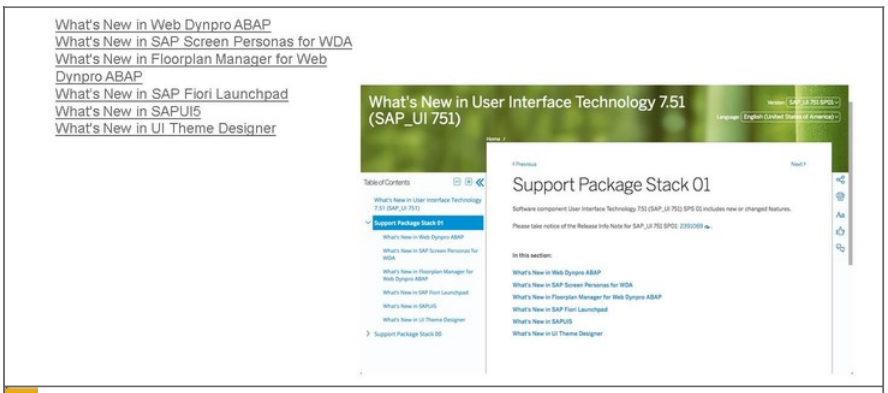


Figure 110: How to Find Detailed Information

SAP Fiori 2.0 is the next significant step in our evolution of user experience for business applications: an award-winning new design concept along with a delightful new visual theme, called Belize.

Key capabilities include:

- New delightful visual theme: Belize
- SAP Fiori launchpad extended with a viewport concept
- Notifications – with connection to SAP Business Workflow and My Inbox
- Me Area: direct navigation to recent apps and business objects, to settings, personalization and app finder.
- Improved navigation – through the Me area and by navigation to previously opened apps in the drop down of the new merged header.
- Merged header: only one header bar, giving more space for each app
- New SAP Fiori elements: overview page, list report and object page.

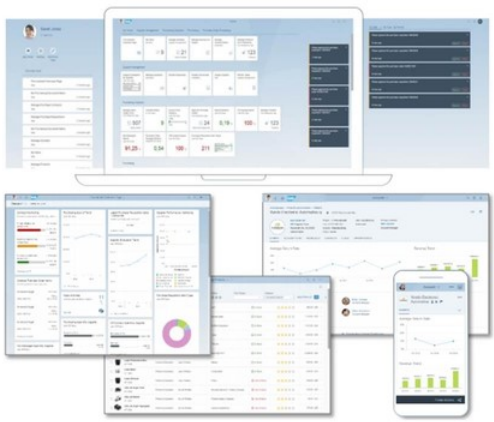


Figure 111: SAP Fiori 2.0

Whats new in SAP Fiori 2.0?

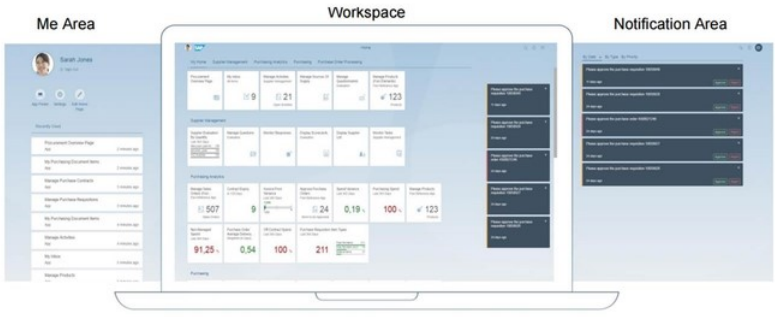


Figure 112: What's New?

You can find details on how to configure the notification center at:

https://blogs.sap.com/2017/02/13/leading-s4hana-ux-notification-center-part-1-activation/

https://blogs.sap.com/2017/02/14/leading-s4hana-ux-notification-center-part-2-providing-notifications/

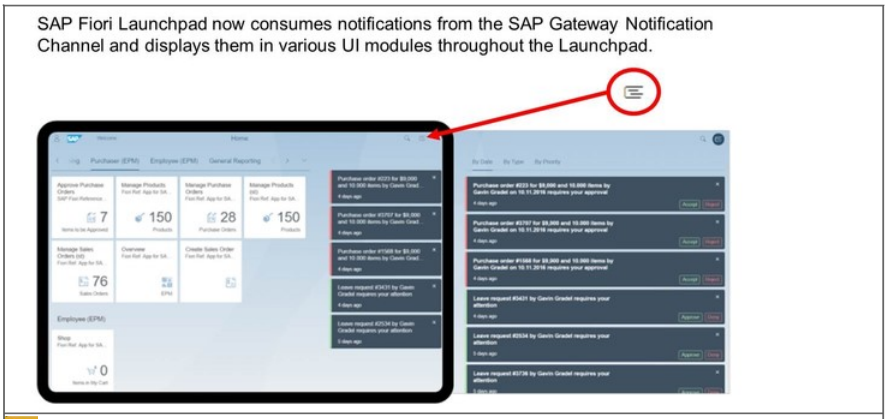


Figure 113: Business Notifications in the SAP Fiori Launchpad

The launchpad groups related apps on tab bars to display one group at a time on the homepage.

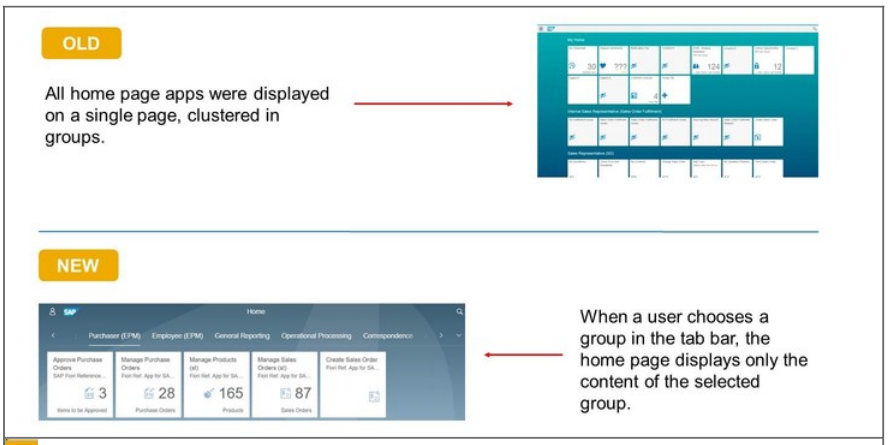


Figure 114: Tab Bar – Displaying Apps One Group at a Time on the Home Page

The launchpad now offers contextual help.

Lets users get started quickly and stay up-to-date easily

User assistance is part of the attractive, simple, and enjoyable user experience

- Instant: exactly when the user needs it
- Context-sensitive: exactly what is needed
- Seamless: within the target application
- Productivity: interactive user guidance

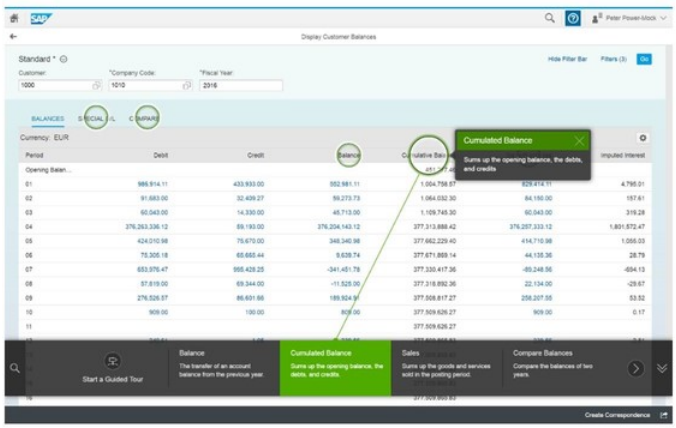


Figure 115: SAP Fiori Launchpad with Contextual Help

A tile is a container that represents an app on the SAP Fiori launchpad home page. All apps have at least one tile, except for fact sheets, (though users can also save fact sheets as tiles if they want). Tiles are only used for launching apps and presenting them on the launchpad.

Tiles can contain an icon, a title, an informative text, numbers, and charts.

The number of visible tiles on the launchpad home page depends on the screen resolution.

Tiles can be grouped

There is a variety of types of tiles:

- KPI Tile
- Comparison Chart Tile
- Mini-Charts like Bullet Chart, Trend Chart, Column Chart
- Basic Launch Tile
- Monitoring Tile
- SAP Jam Tile
- Feed Tile

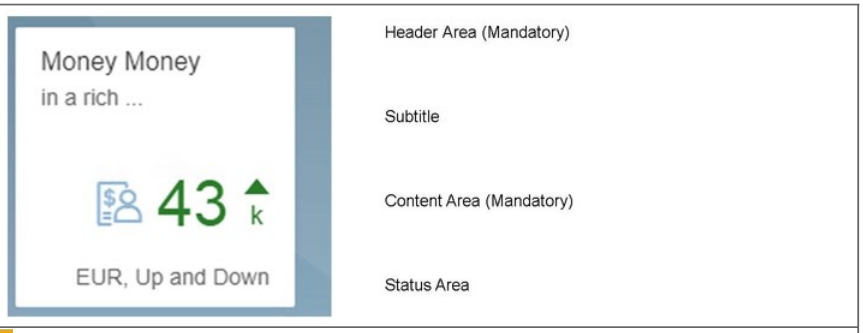


Figure 116: Tiles Layout

An app finder is available to search for a specific app.

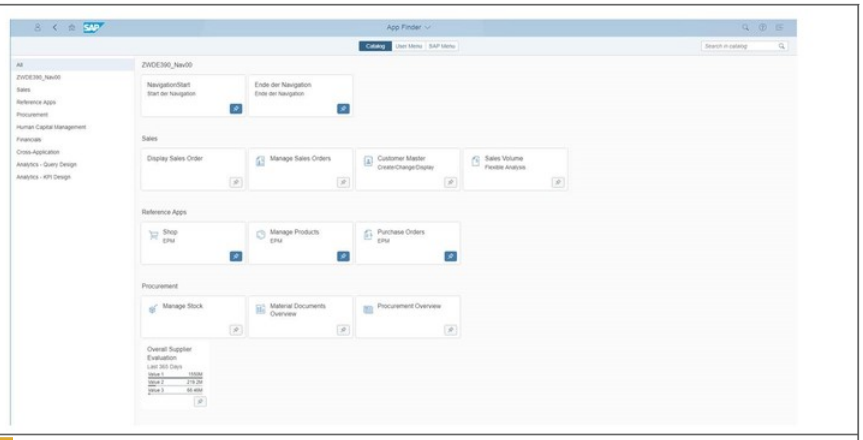


Figure 117: The App Finder

LESSON SUMMARY
You should now be able to:

- Understand the SAP Fiori launchpad

Unit 9
Lesson 2

Understanding the Technical Perspective of SAP Fiori Launchpad

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the technical perspective of SAP Fiori launchpad

Technical Perspective of SAP Fiori Launchpad

The SAP Fiori Launchpad - Technical Perspective

The user can access the tile catalog directly from the launchpad home page. They find all tiles they are allowed to use. The tiles are grouped into catalogs. A search field and a group selection help is available, to assist in finding the tile that is actually required.

The tile catalog has two functions:

- Tiles that are used most often can be added to the Home page
- Tiles, that are used more seldom, can be accessed directly from the catalog, without adding them to the Home page

The tile catalog is provided by the SAP Fiori launchpad.

Note:
Apps use this tile catalog and do not design their own.

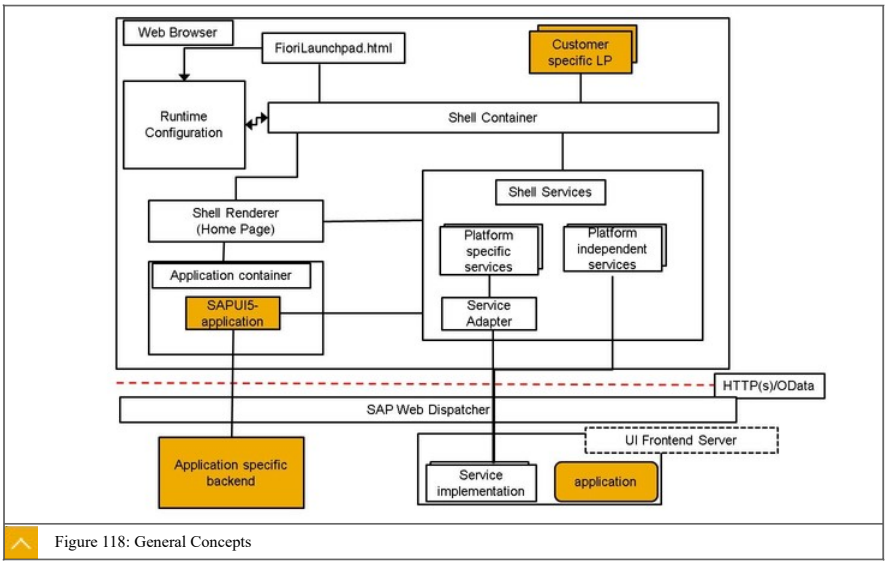


Figure 118: General Concepts

LESSON SUMMARY

You should now be able to:

- Understand the technical perspective of SAP Fiori launchpad

Unit 9
Lesson 2

Understanding the Technical Perspective of SAP Fiori Launchpad

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the technical perspective of SAP Fiori launchpad

Technical Perspective of SAP Fiori Launchpad

The SAP Fiori Launchpad - Technical Perspective

The user can access the tile catalog directly from the launchpad home page. They find all tiles they are allowed to use. The tiles are grouped into catalogs. A search field and a group selection help is available, to assist in finding the tile that is actually required.

The tile catalog has two functions:

- Tiles that are used most often can be added to the Home page
- Tiles, that are used more seldom, can be accessed directly from the catalog, without adding them to the Home page

The tile catalog is provided by the SAP Fiori launchpad.

Note:
Apps use this tile catalog and do not design their own.

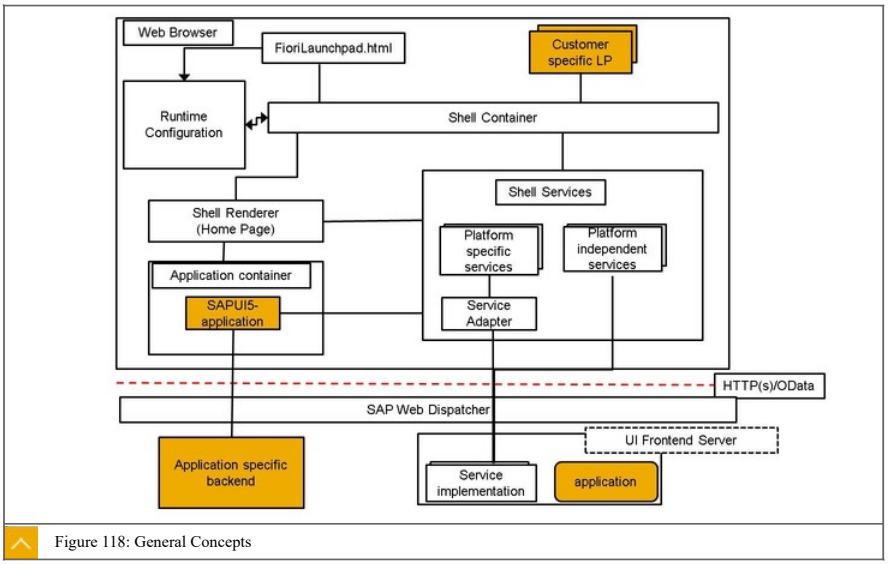


Figure 118: General Concepts

LESSON SUMMARY

You should now be able to:

- Understand the technical perspective of SAP Fiori launchpad

Unit 9

Learning Assessment - Answers

1. Which factors are key to the user experience with user assistance?

   Choose the correct answers.

   - [X] **A** Instant
   - [X] **B** Context-sensitive
   - [ ] **C** confusing
   - [X] **D** Seamless
   - [ ] **E** Error prone

   You are correct! User assistance must be instant, context-sensitive, and seamless

2. What areas are mandatory in the tile layout?

   Choose the correct answers.

   - [ ] **A** Subtitle
   - [X] **B** Header Area
   - [X] **C** Content Area
   - [ ] **D** Status Area
   - [ ] **E** KPI Area

   You are correct! The header area and the content area are mandatory for a tile.

3. Which of the following components are part of the SAP Fiori launchpad?

   Choose the correct answers.

   - [X] **A** Shell services
   - [ ] **B** UI2 services
   - [X] **C** Shell container
   - [ ] **D** Runtime container
   - [X] **E** Runtime configuration

   You are correct! The SAP Fiori launchpad consists of shell services, a shell-container and a runtime-configuration.

4. Which types of services are known or supported by the SAP Fiori launchpad?

   Choose the correct answers.

   - [X] **A** Platform-specific services
   - [ ] **B** UI5 services
   - [X] **C** Platform-independent services
   - [ ] **D** UI2 Services

   You are correct! The SAP Fiori Launchpad supports platform-specific and platform-independent services.

UNIT 10

SAP Fiori Launchpad Configuration

Lesson 1

Understanding the SAP Fiori Launchpad Configuration 125

UNIT OBJECTIVES

- Understand the SAP Fiori launchpad configuration

Unit 10
Lesson 1

Understanding the SAP Fiori Launchpad Configuration

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the SAP Fiori launchpad configuration

SAP Fiori Launchpad Configuration

The configuration of the SAP Fiori launchpad is illustrated in the figure, Configuration Steps.

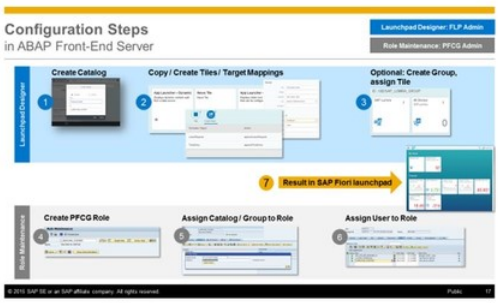


Figure 119: Configuration Steps

It is important to understand how the objects on the SAP Fiori launchpad relate to each other.

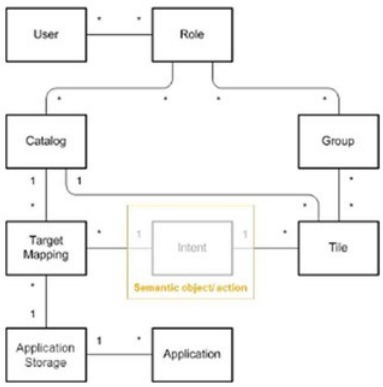


Figure 120: Understanding SAP Fiori Launchpad Object Relationships

Configure the tile by creating a target mapping, a semantic object, and a resulting action.

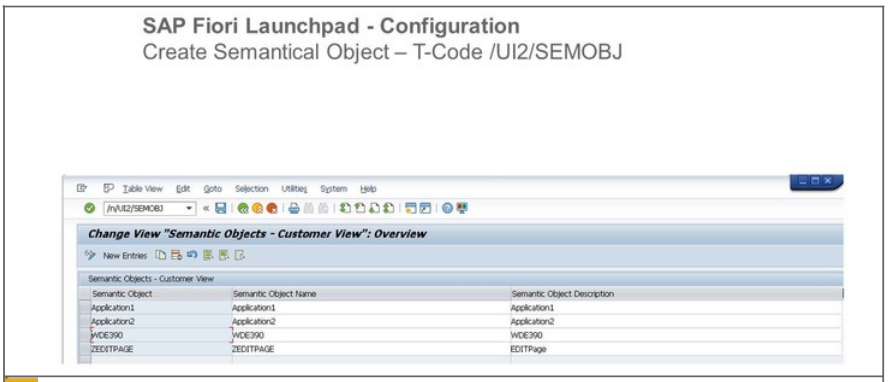


Figure 121: Create Semantic Object

A tile in the SAP Fiori launchpad does not point directly to an SAP Fiori app

A tile points to a target mapping in conjunction with a semantic object and an action.

When you create, update, or delete a catalogue or a tile, all these actions have to be captured.

If you launch the SAP Fiori launchpad designer in a customization scope, these actions are captured under the customizing request.

To start a configuration in SAP Fiori Launchpad Designer you need to create a transport request and assign it to a system (SE01)

- Customizing (CUST)

  This layer can be used for testing or other purposes.

- Workbench (CONF)

  All content that you want to deliver to customers must be created in the CONF layer.

Running in customization scope:

https://server:port/sap/bc/ui5_ui5/sap/arsrvc_upd_admin/main.html?scope=CUST

You can use transaction /UI2/FLPD_CUST to start the SAP Launchpad Designer in customizing mode.

Running in configuration scope:

https://server:port/sap/bc/ui5_ui5/sap/arsrvc_upd_admin/main.html?scope=CONF

You can also use transaction /UI2/FLPD_CONF to start the SAP Launchpad Designer in configuration mode.

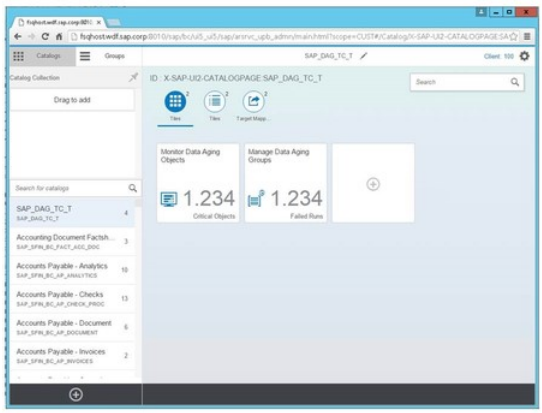


Figure 122: Log on to the Launchpad Designer

In the Launchpad Designer you need to assign whatever changes you make to one of the requests you created in the Gateway system.

Choose the icon to open the assignment dialog.

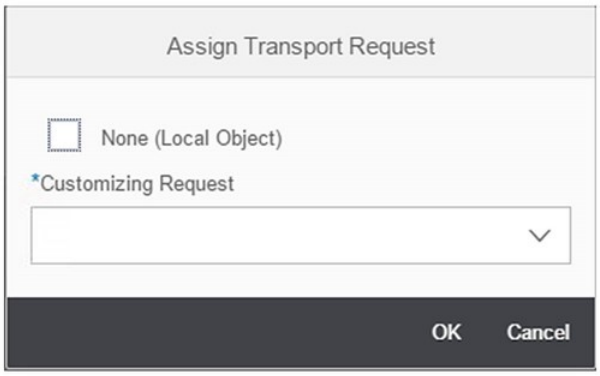


Figure 123: Assign a Transport Request

To create a new Fiori catalogue;

1. Choose the  at the bottom of the screen.
2. Enter the name and ID of your catalog.
3. Choose save.

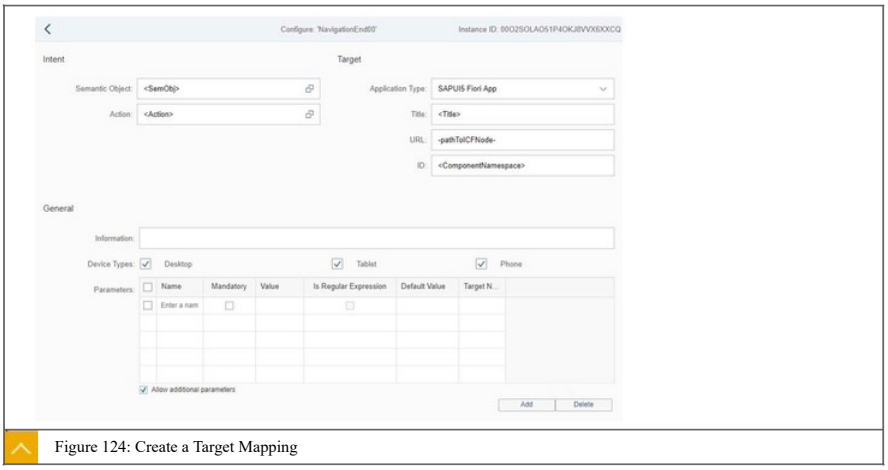


Figure 124: Create a Target Mapping

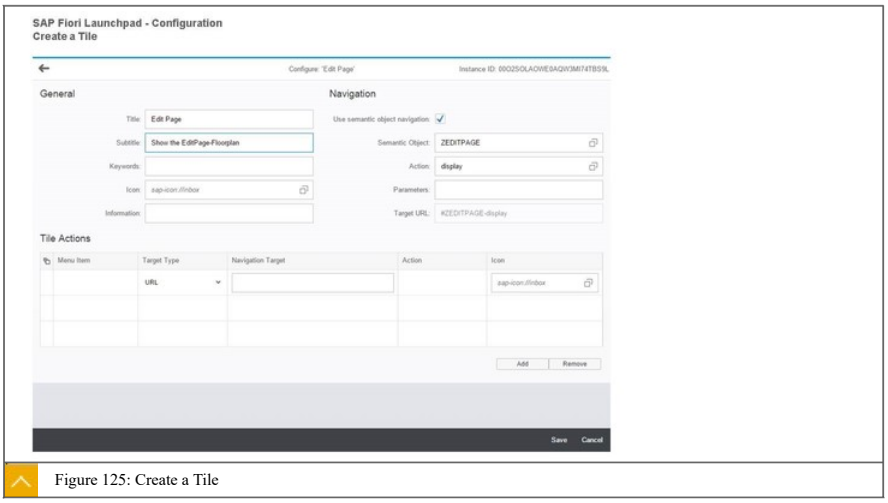


Figure 125: Create a Tile

Run transaction PFCG and create a new role.

Add a new role of type **SAP Fiori Tile Catalog** .

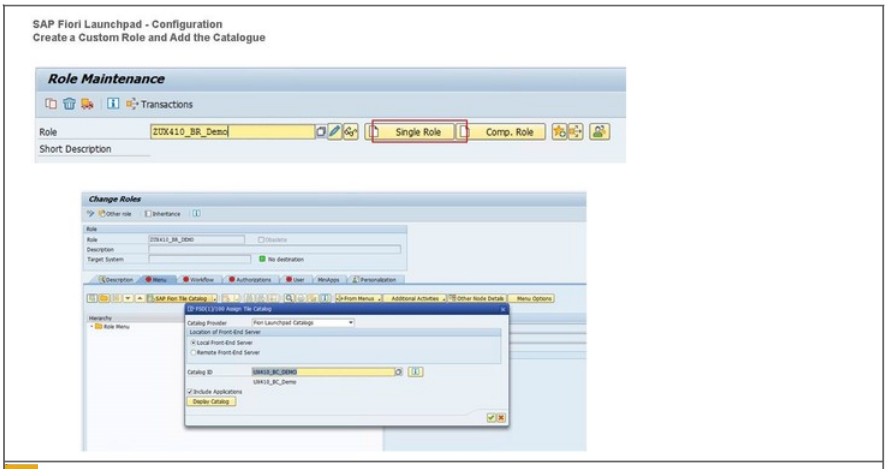


Figure 126: Create a Custom Role and Add the Catalogue

Assign a group or user to the role.

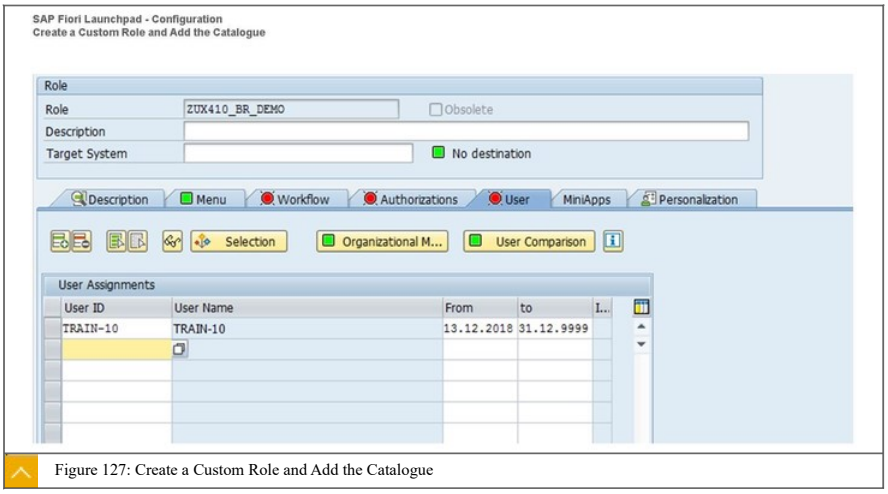


Figure 127: Create a Custom Role and Add the Catalogue

Add the application to the launchpad.

Figure 128: Add the Application to the Launchpad

Select or create a new group

Figure 129: Select or Create a New Group

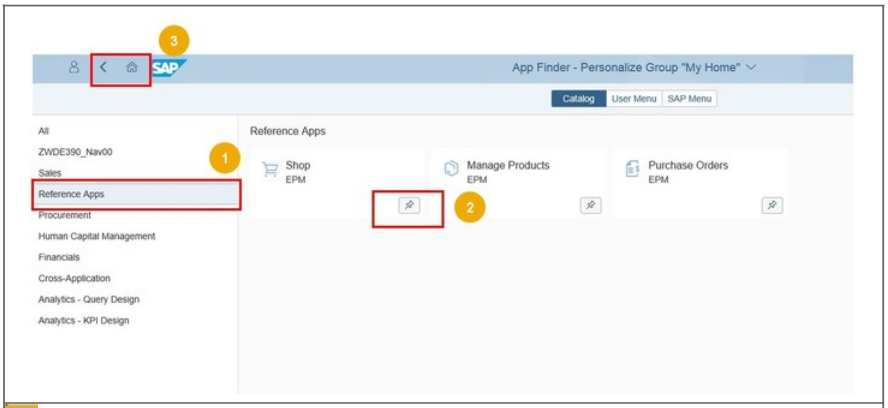


Figure 130: Select a Catalog and Add an App

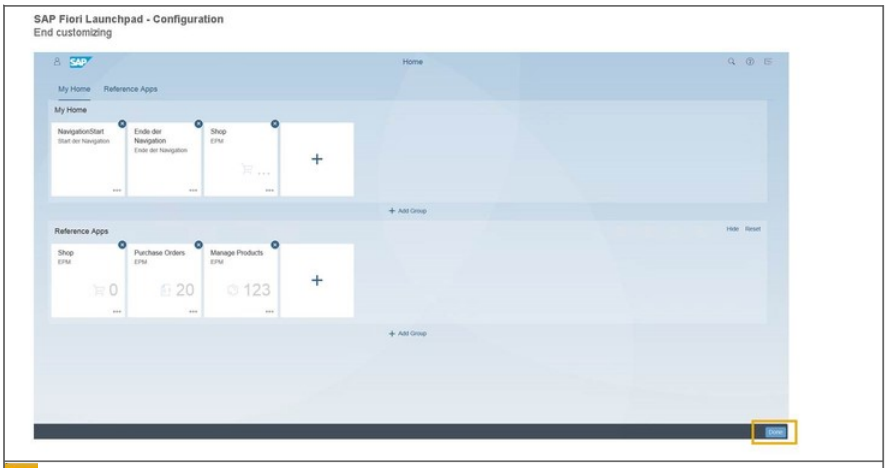


Figure 131: End Customizing

An application can be started from the launchpad tile.

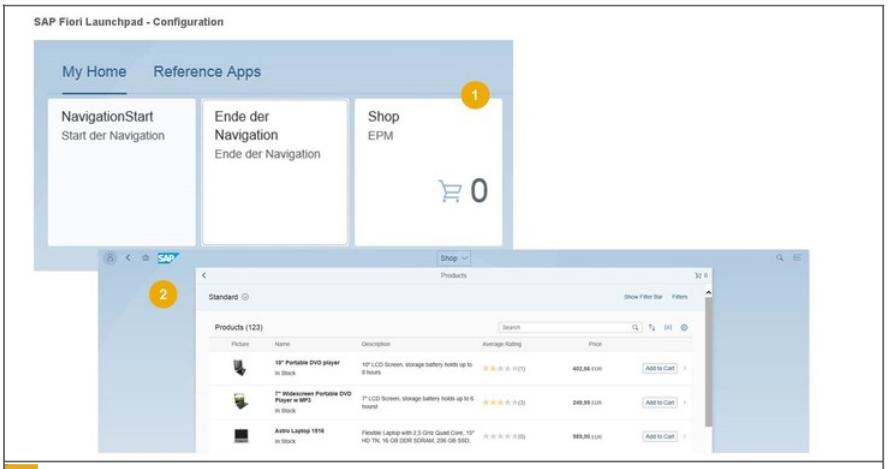


Figure 132: Start the Application from the Launchpad

To create dynamic tiles, it is important to understand the anatomy of a tile.

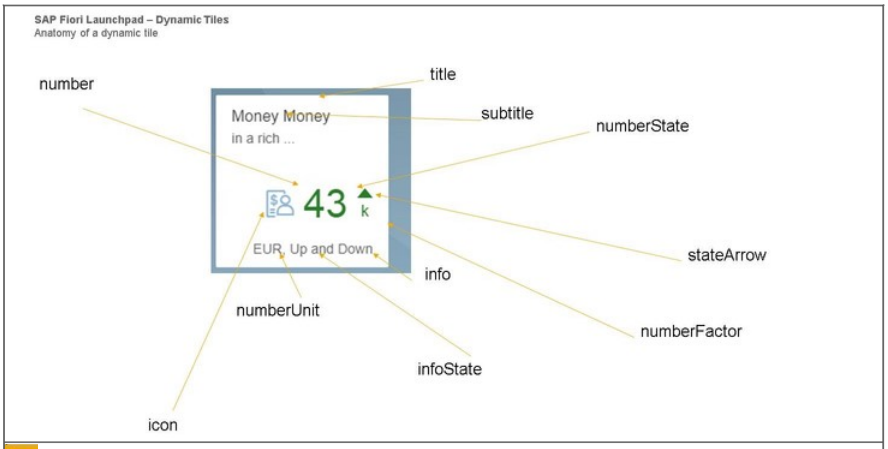


Figure 133: Anatomy of a Dynamic Tile

With this in mind, it is possible to build a service that provides structured information that can be shown on the tile dynamically.

Note:
When using dynamic tiles, you need to keep in mind that requests to get tile data create network traffic. The load created, in some cases, may be a heavy load on the system. This might lead to weak performance.

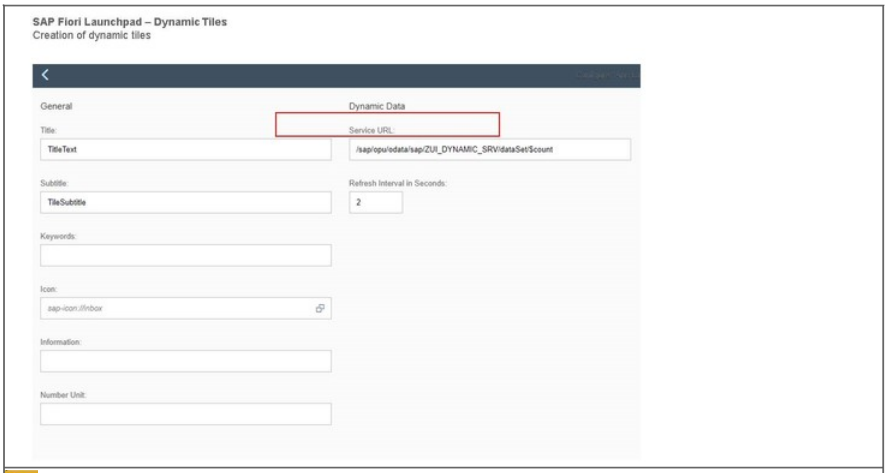


Figure 134: Creation of Dynamic Tiles (1 of 4)

It is also possible to implement a complete dynamic tile.

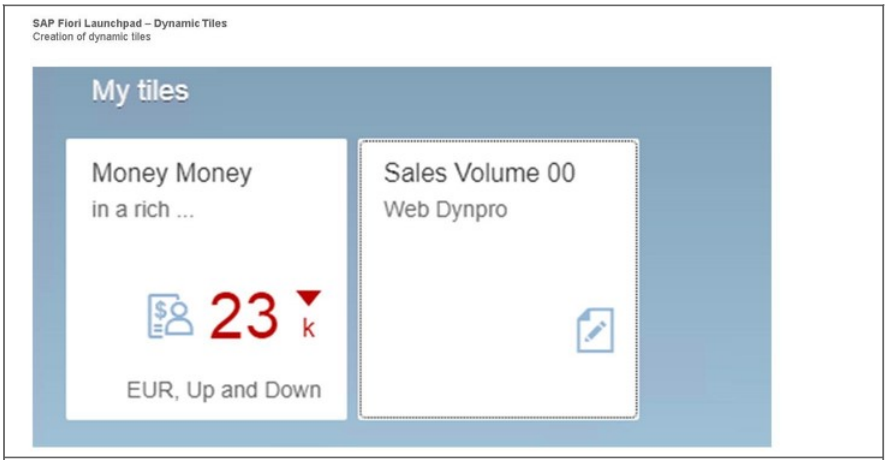


Figure 135: Creation of Dynamic Tiles (2 of 4)

Implement an OData service in the back-end system

Register the OData service on the front-end server.

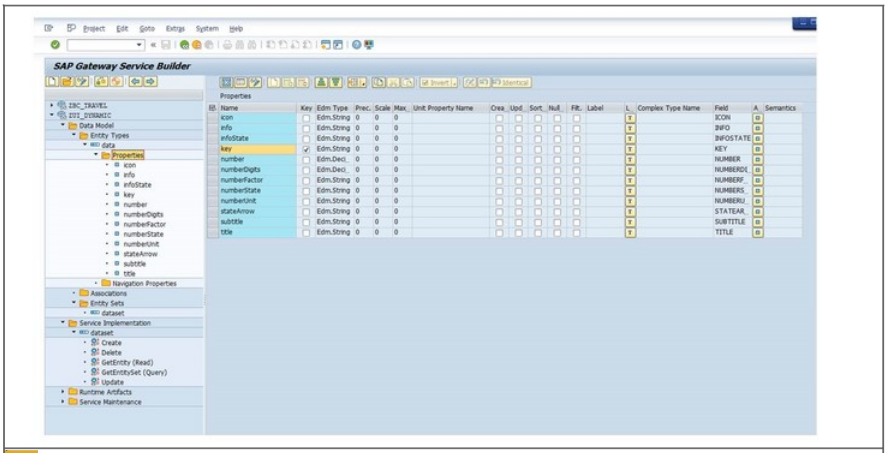


Figure 136: Creation of Dynamic Tiles (3 of 4)

For the second step, configure the dynamic tile. Make sure the result of the service invocation is in the JSON format.

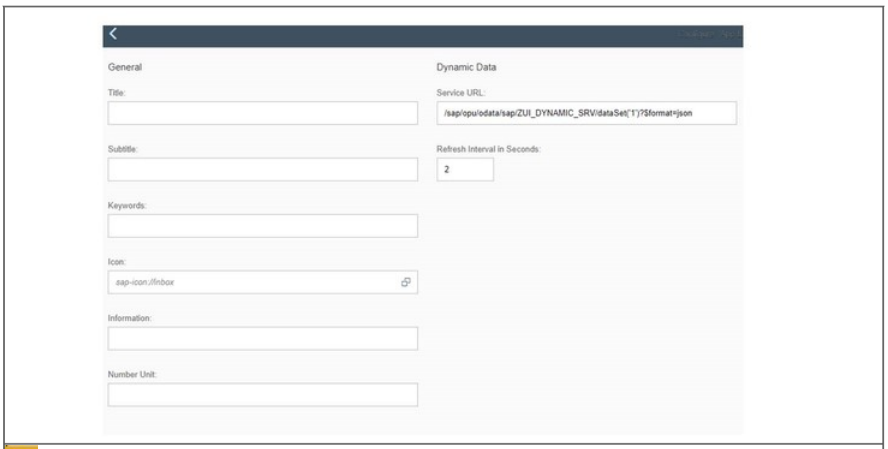


Figure 137: Creation of Dynamic Tiles (4 of 4)

LESSON SUMMARY
You should now be able to:

- Understand the SAP Fiori launchpad configuration

Unit 10

Learning Assessment

1. Which of the following make up the configuration of a tile for launching an SAP Fiori app of type SAPUI5 in the SAP Fiori launchpad designer?

   Choose the correct answers.

   - [ ] **A** Semantic object
   - [ ] **B** Launchpad creation via LPD_CUST
   - [ ] **C** Target Mapping
   - [ ] **D** Tile configuration
   - [ ] **E** Tile implementation

2. What transaction is used to create an semantic object for customer configuration?

   Choose the correct answer.

   - [ ] **A** /UI5/LPD_CUST
   - [ ] **B** /UI2/SEMOBJ
   - [ ] **C** /UI5/SEMOBJ
   - [ ] **D** pfcg
   - [ ] **E** su01

3. What application in the SAP Fiori launchpad helps the user to find applications that are available to the user?

   Choose the correct answer.

   - [ ] **A** Tile Finder
   - [ ] **B** Application Finder
   - [ ] **C** Fiori Designer
   - [ ] **D** Fiori Appsearch

Unit 10

Learning Assessment - Answers

1. Which of the following make up the configuration of a tile for launching an SAP Fiori app of type SAPUI5 in the SAP Fiori launchpad designer?

   Choose the correct answers.

   - [X] **A** Semantic object
   - [ ] **B** Launchpad creation via LPD_CUST
   - [X] **C** Target Mapping
   - [X] **D** Tile configuration
   - [ ] **E** Tile implementation

   You are correct! To create a tile, you need to implement a semantic object, a target mapping, and the tile configuration.

2. What transaction is used to create an semantic object for customer configuration?

   Choose the correct answer.

   - [ ] **A** /UI5/LPD_CUST
   - [X] **B** /UI2/SEMOBJ
   - [ ] **C** /UI5/SEMOBJ
   - [ ] **D** pfcg
   - [ ] **E** su01

   You are not correct. Use the transaction /UI2/SEMOBJ to create a semantic object.

3. What application in the SAP Fiori launchpad helps the user to find applications that are available to the user?

   Choose the correct answer.

   - [ ] **A** Tile Finder
   - [X] **B** Application Finder
   - [ ] **C** Fiori Designer
   - [ ] **D** Fiori Appsearch

   You are correct! The Application Finder helps the user to find and search for apps that are assigned to the user.

Unit 10

Learning Assessment - Answers

1. Which of the following make up the configuration of a tile for launching an SAP Fiori app of type SAPUI5 in the SAP Fiori launchpad designer?

   Choose the correct answers.

   - [X] **A** Semantic object
   - [ ] **B** Launchpad creation via LPD_CUST
   - [X] **C** Target Mapping
   - [X] **D** Tile configuration
   - [ ] **E** Tile implementation

   You are correct! To create a tile, you need to implement a semantic object, a target mapping, and the tile configuration.

2. What transaction is used to create an semantic object for customer configuration?

   Choose the correct answer.

   - [ ] **A** /UI5/LPD_CUST
   - [X] **B** /UI2/SEMOBJ
   - [ ] **C** /UI5/SEMOBJ
   - [ ] **D** pfcg
   - [ ] **E** su01

   You are not correct. Use the transaction /UI2/SEMOBJ to create a semantic object.

3. What application in the SAP Fiori launchpad helps the user to find applications that are available to the user?

   Choose the correct answer.

   - [ ] **A** Tile Finder
   - [X] **B** Application Finder
   - [ ] **C** Fiori Designer
   - [ ] **D** Fiori Appsearch

   You are correct! The Application Finder helps the user to find and search for apps that are assigned to the user.

UNIT 11

SAP Fiori Layout Decision Guidelines

UNIT OBJECTIVES

- Differentiate between application framework, page layout, and floorplans
- Create a dynamic page app
- Understand the flexible-column layout
- Understand the full screen layout
- Understand the split-screen layout
- Understand the master-detail layout

Unit 11
Lesson 1

Differentiating between Application Framework, Page Layout, and Floorplans

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Differentiate between application framework, page layout, and floorplans

Layout Decisions

As of the SAPUI5 version 1.44, SAP has decided to mark the full-screen layout deprecated. From SAPUI5 version 1.48, the master-detail layout deprecated. While the layout is deprecated, the SAPUI5 SDK UI controls remain valid.

Application Framework

The user interface for each business process is referred to as an app. All the user's apps are presented in the form of tiles on the workspace. This forms the heart of the SAP Fiori launchpad. An app in the SAP Fiori launchpad has to follow one of the following layouts:

- Dynamic Page
- Flexible Column Layout

Page Layouts

The page layout specifies the overall layout off the app and the way the information is displayed.

The application framework is based on a page layout, so they go hand in hand.

Template Choice

When deciding which template to use for your app, it is helpful to answer questions on the nature of the tasks and what the user wants to achieve when working in the app. The outcome will provide guidance on what type of template to use.

Table 2: Template choice

Question	Template
Is viewing, inspecting, or editing details on one or several elements from a list of elements an important user requirement?	If so, use the master-detail pattern, for example a tracking app.
Is inspecting the status of one or more objects important?	If so, use the master-detail pattern, for example a tracking app.

Question	Template
Are the objects you are inspecting so complex that you require charts to illustrate a point quickly?	If so, you should use a full screen app with charts.

Page layout and floorplans

The page layout is the basis (container) for the different floorplans described by the SAP Fiori design language.

A developer creates applications based on floorplans.

A floorplan is a UI design pattern that solves repeated UI problems in a consistent way.

A floorplan is displayed within the overall page layout.

LESSON SUMMARY
You should now be able to:

- Differentiate between application framework, page layout, and floorplans

Unit 11
Lesson 2

Creating a Dynamic Page App

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Create a dynamic page app

Dynamic Page

The dynamic page layout, which is a successor to the full-screen layout, provides the following features:.

- Fully responsive
- Additional page header functionality
- Page header use by all floorplans

With SAPUI5 release 1.54, you can also combine the dynamic page with the following floorplans:

- Initial page
- Object page

The following floorplans are allowed:

- Analytical list page
- List report
- Overview page
- Wizard
- Worklist
- Initial page
- Object page

The dynamic page itself is a skeleton structure filled by content elements.

Content elements can be built in the following three ways:

- Automatically
- Semantic pages
- Manually

Use the dynamic page layout if you are building a freestyle application that uses the dynamic page header and footer toolbar features of SAP Fiori 2.0 (versions 1.40 and higher)

Do not use the dynamic page layout if you are planning any of the following scenarios:

- Use SAP Fiori elements, such as the list report, analytical list page, overview page, or object page. These elements already incorporate the dynamic page layout.
- Implement an initial page or object page floorplan. These floorplans already incorporate snapping header and footer toolbar features. The behavior is similar to the dynamic page, but the technical foundation is different.
- Display a small amount of information. In this case, use a dialog instead. If you cannot avoid using the dynamic page layout, use letterboxing to mitigate the issue.

The dynamic page layout consists of the following areas:

- The header title combined with the subtitle or summary, key information, and global actions (always visible)
- The header content (expandable and collapsible area) (optional)
- The header features to actively collapse or expand and pin or unpin the header
- The content of the page, with a footer toolbar and finalizing actions (optional)
- Depending on your use case, you can either use one of the predefined floor plans, or create your own layout.

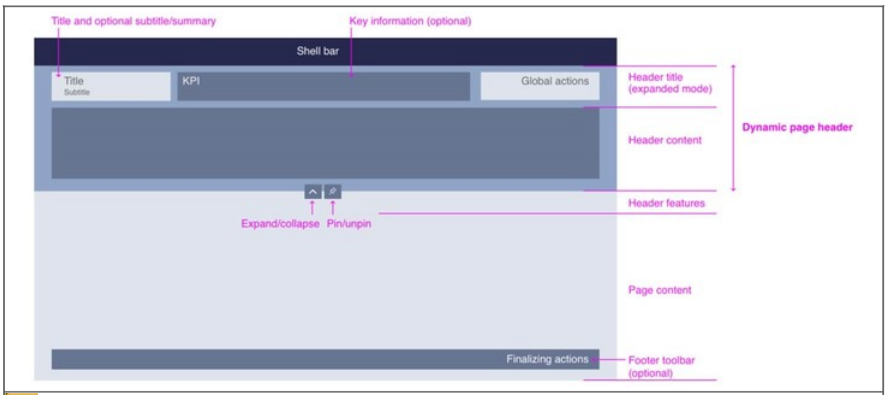


Figure 138: Dynamic Page Header area

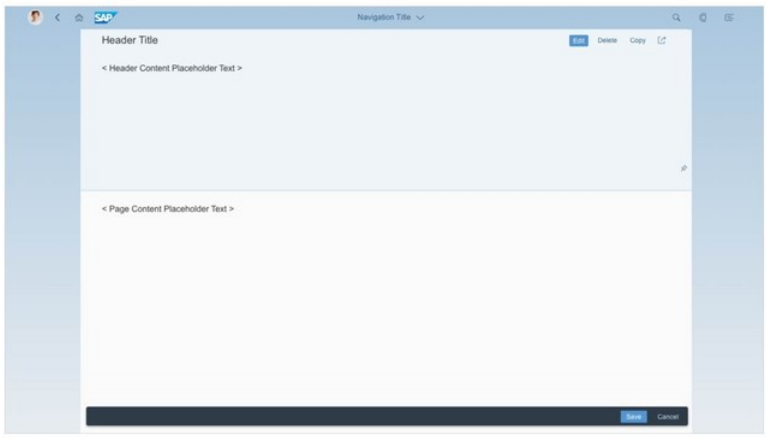


Figure 139: Header area expanded

LESSON SUMMARY
You should now be able to:

- Create a dynamic page app

Unit 11
Lesson 3

Understanding the Flexible Column Layout

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the flexible-column layout

Flexible Column Layout

The flexible-column layout is a layout control that displays multiple floorplans on a single page and displays up to three columns. This allows faster and more fluid navigation between floorplans than the usual page-by-page navigation. Users can expand the columns to find what they want to focus on. They can switch between different layouts, and view the rightmost column in full screen mode.

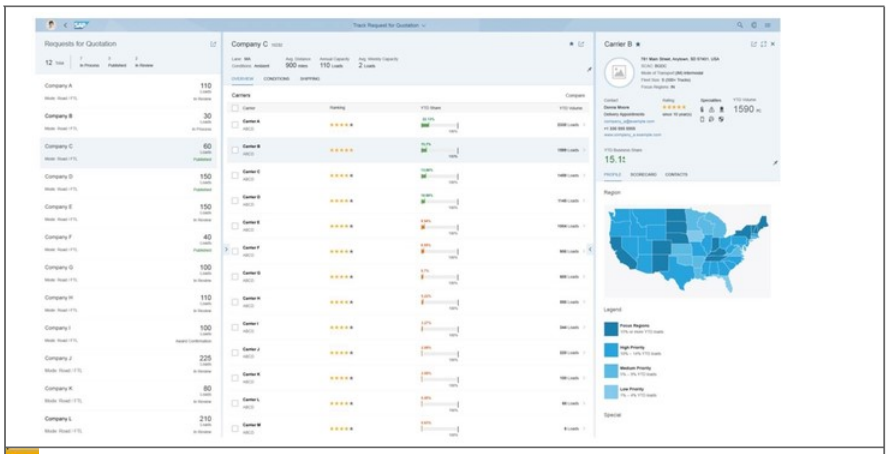


Figure 140: Flexible Column Layout

Use the flexible-column layout if you want to create a master-detail or master-detail-detail scenario in which the user can drill down or navigate.

Do not use the flexible-column layout if you are planning any of the following scenarios:

- You want to build a workbench or tools layout. The flexible-column layout is not meant to provide a main column with additional side columns on the left and right or either. If you want to display additional content to enrich the main content and to help users better perform their tasks, use the dynamic side content instead.
- You want to create a dashboard with context-independent pages.

- You want to open multiple instances of the same object type. Use the multi-instance-handling floorplan instead.
- You want to split a single object into multiple columns, or display only a small amount of information.
- You want to embed the SAP Fiori launchpad or overview page into one of the columns.

To give users a better overview, the flexible-column layout offers different layouts with one, two, or three columns. Depending on the screen size, up to five different base layouts are available. The overall look depends also on the reading direction.

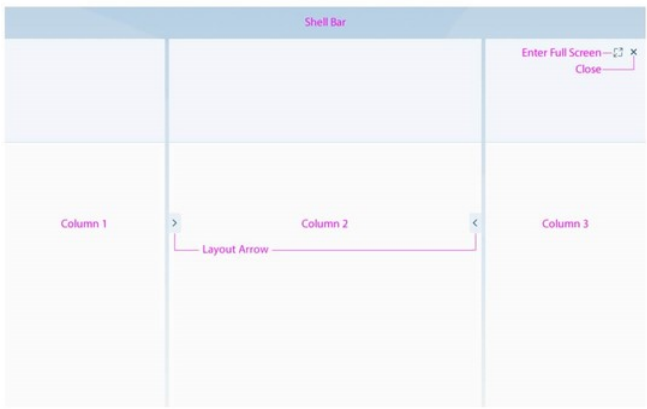


Figure 141: Flexible Column Layout

An example of a flexible column layout is shown.

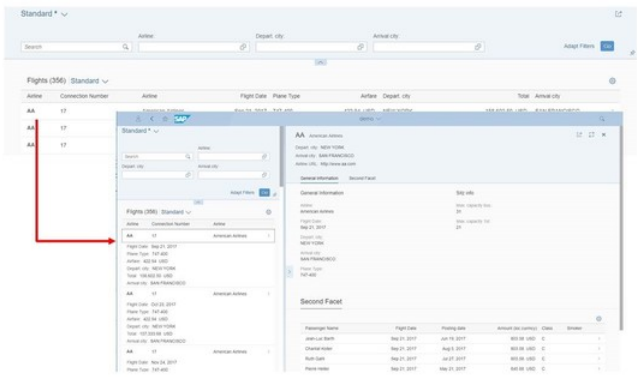


Figure 142: Flexible Column Layout in Action

LESSON SUMMARY
You should now be able to:

- Understand the flexible-column layout

Unit 11
Lesson 4

Understanding the Full Screen Layout

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the full screen layout

Full Screen Layout

Since SAPUI5 version 1.44, the implementation of the full-screen layout described by the SAP Fiori design language is deprecated.

If you have applications that implement the layout, it is recommended to update the implementation and use dynamic page instead. Change the implementation to align to the newest SAP Fiori design guidelines to have a consistent user experience.

The full-screen template affords maximum flexibility while still providing the look and feel of SAP Fiori. As the full screen template does not present a pre-defined hierarchy of content, it provides a higher degree of flexibility. UI elements inside the template can be arranged individually based on app use cases.

> Note:
> As of SAPUI5 version 1.44, this layout is deprecated and should, if possible, be replaced by the dynamic page layout.

The full-screen layout lets you exploit the full width of the screen. Use this layout for apps that need to display large amounts of data, large visualizations, or wide tables.

Avoid switching between full-screen and split-screen layouts within an app. If your app has some screens that require the full width, and others that do not, decide on one layout and stick to it.

The list report floorplan is the common floorplan for full screen layout.

```
<mvc:View controllerName="com.sap.training.worklist.controller.Worklist" xm
    xmlns:semantic="sap.m.semantic" xmlns="sap.m">
    <semantic:FullscreenPage id="page" navButtonPress="onNavBack"
                             showNavButton="true" title="{i18n>worklistViewTitle}">
        <semantic:content>
            <IconTabBar
                id="iconTabBar"
                select="onQuickFilter"
                expandable="false"
                applyContentPadding="false">
```

Figure 143: sap.m.semantic.FullscreenPage

Like all layouts, the full screen layout is embedded in the shell bar of the SAP Fiori launchpad. From the header, users have access to launchpad services. These include the home page, search, settings, and help. Apps are embedded in the shell and have little influence over its features.

The uppermost app element is the app header. This includes features such as the back navigation, the app title, and one optional action.

The long scrollable page below the app header contains the app content. You can either use one of the predefined floorplans, or create your own layout.

You can also add an application-specific footer-toolbar at the bottom of the screen.

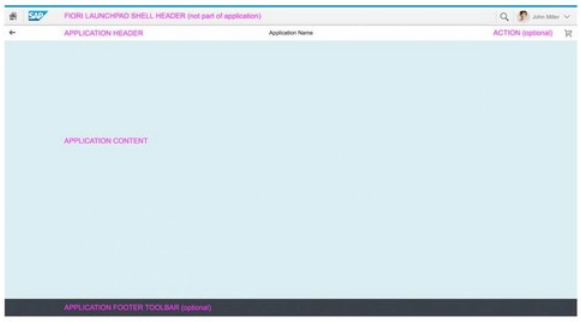


Figure 144: Structure of Full-screen Layout

LESSON SUMMARY
You should now be able to:

- Understand the full screen layout

Unit 11
Lesson 5

Understanding the Split Screen Layout

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the split-screen layout

Split Screen Layout

As of the version SAPUI5 1.48, the implementation of the split-app and master-detail using the sap.m.SplittApp control and related concepts are deprecated.

If you implemented applications using this layout, you must plan to reimplement the application to align with the newest guidelines. This is necessary to have one consistent user experience.

Split-screen layout is optimized for displaying and processing a list of items.

The split-screen layout is divided into the following two separate areas.

Master list area: The master list displays the items available to the user. The user can navigate between the items, perform a basic search, and organize the list using sort, filter, and grouping functions. Currently, only the master list is allowed in this area. For more details, see the Master List pattern.

Details area: The details area displays the details for single or multiple items that are selected in the master list. The details area can contain one of the following floorplans:

- Object view floorplan
- Flat object view floorplan
- Edit page floorplan
- Create page floorplan

Note:
As of wave 1.48, this layout is deprecated, and if possible, should be replaced by the dynamic page layout.

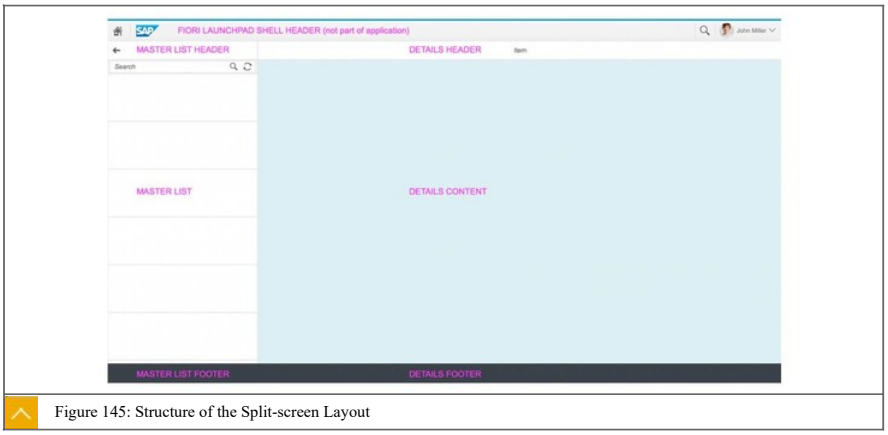


Figure 145: Structure of the Split-screen Layout

LESSON SUMMARY
You should now be able to:

- Understand the split-screen layout

Unit 11
Lesson 6

Understanding Master-Detail

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the master-detail layout

Master-Detail Layout

The master detail template displays a list on the left-hand side of the screen. When a list item is selected, the details are displayed on the right-hand side of the screen. This template is used in several apps. It offers a smart layout and is responsive to different form factors and orientation.

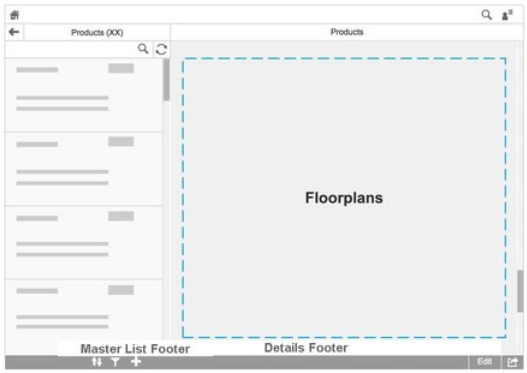


Figure 146: Master-Detail

```
<mvc:View
    controllerName="com.sap.training.master.controller.Master"
    xmlns:mvc="sap.ui.core.mvc"
    xmlns:core="sap.ui.core"
    xmlns="sap.m"
    xmlns:semantic="sap.m.semantic">

  <semantic:MasterPage
      id="page"
      title="{masterView>/title}"
      navButtonPress="onNavBack"
      showNavButton="true">
    <semantic:subHeader>
      <Bar id="headerBar">
        <contentMiddle>
          <SearchField
              id="searchField"
              showRefreshButton="{= !${device>/support/touch} }"
              tooltip="{i18n>masterSearchTooltip}"
              width="100%"
              search="onSearch">
          </SearchField>
        </contentMiddle>
      </Bar>
    </semantic:subHeader>

    <semantic:content>
```

Figure 147: sap.m.masterpage

LESSON SUMMARY
You should now be able to:

- Understand the master-detail layout

7. Since when is the full-screen layout deprecated?

   Choose the correct answer.

- ☐ **A** 1.42
- ☐ **B** 1.44
- ☐ **C** 1.56
- ☐ **D** 1.58
- ☐ **E** 1.60

8. Since when is the Split-Screen-layout deprecated?

   Choose the correct answer.

- ☐ **A** 1.42
- ☐ **B** 1.44
- ☐ **C** 1.46
- ☐ **D** 1.48
- ☐ **E** 1.50

Unit 11

Learning Assessment - Answers

1. Which of the following layouts, are described in version 1.48 of the Guidelines?

   Choose the correct answers.

   - [X] **A** Dynamic Page
   - [ ] **B** Full-Screen
   - [ ] **C** Split-Screen
   - [X] **D** Flexible Column Layout
   - [ ] **E** MobileDeviceLayout

   You are correct! As of SAPUI5 1.48, the guidelines define the Dynamic Page and the Flexible Column Layout.

2. What aspects should be considered in deciding which layout template to use?

   Choose the correct answers.

   - [X] **A** When deciding which template to use for your app, you need to consider what the user wants to achieve when working in the app.
   - [ ] **B** When deciding which template to use for your app, you need to analyse what end device the app is executed on.
   - [X] **C** It is helpful to answer questions on the nature of the tasks.
   - [ ] **D** When deciding which template to use for your app, you need to consider what browser type is used.
   - [ ] **E** You need to analyze the capabilities of the users.

   You are correct! When deciding which template to use for your app, you need to consider what the user wants to achieve when working in the app, and analyze the nature of the tasks.

3. Which types of application can use the dynamic page layout?

   Choose the correct answer.

   - [ ] **A** The layout is deprecated, which means that you should not use the layout anymore.
   - [X] **B** You can use the dynamic page layout for any kind of application.
   - [ ] **C** The layout is only used for master-detail scenarios.
   - [ ] **D** The dynamic page layout is only used for analytical applications.

   You are correct! You can use the dynamic page layout for any type of application.

4. When should you not use the dynamic page layout?

   Choose the correct answers.

   - [X] **A** You are planning to use SAP Fiori elements, such as the list report, analytical list page, overview page, or object page, which already incorporates the dynamic page layout.
   - [ ] **B** You are building a freestyle application that uses the dynamic page header and footer toolbar features of SAP Fiori 2.0 (versions 1.40 and higher).
   - [X] **C** You want to implement an initial page or object page floor plan. These floor plans already incorporate snapping header and footer toolbar features. The behavior is comparable to the dynamic page behavior, but the technical foundation is different.
   - [ ] **D** You are building an analytical application that is using the sap.viz API.
   - [X] **E** You only need to display a small amount of information. In this case, use a dialog instead. If you cannot avoid using the dynamic page layout, use letterboxing to mitigate the issue.

   You are correct! You should not use the dynamic page layout when you are planning to use SAP Fiori elements, such as the list report, analytical list page, overview page, or object page. These already incorporate the dynamic page layout. Do not use, when you want to implement an initial page or object page floor plan. These floor plans already incorporate snapping header and footer toolbar features. The behavior is comparable to the dynamic page, but the technical foundation is different. Do not use, when you only need to display a small amount of information. In this case, use a dialog instead. If you cannot avoid using the dynamic page layout, use letterboxing to mitigate the issue.

5. When do you use the flexible-column layout?

   Choose the correct answer.

   - [ ] **A** You want to open multiple instances of the same object type.
   - [X] **B** You want to create a master-detail or master-detail-detail scenario in which the user can drill down or navigate.
   - [ ] **C** You want to split a single object into multiple columns, or display only a small amount of information.
   - [ ] **D** You want to embed the SAP Fiori launchpad, or overview page into one of the columns.

   You are correct! You should implement the flexible-column-layout to create a master-detail or master-detail-detail scenario in which the user can drill down or navigate.

6. How many columns are supported by the flexible-column layout?

   Choose the correct answer.

   - [ ] **A** 1
   - [ ] **B** 2
   - [X] **C** 3
   - [ ] **D** 4
   - [ ] **E** 5

   You are correct! Up to three columns are available to show different floorplans when using the flexible-column layout.

7. Since when is the full-screen layout deprecated?

   Choose the correct answer.

   - [ ] **A** 1.42
   - [X] **B** 1.44
   - [ ] **C** 1.56
   - [ ] **D** 1.58
   - [ ] **E** 1.60

   You are correct! The full-screen layout is deprecated since SAPUI5 1.44

8. Since when is the Split-Screen-layout deprecated?

   Choose the correct answer.

- [ ] **A** 1.42
- [ ] **B** 1.44
- [ ] **C** 1.46
- [x] **D** 1.48
- [ ] **E** 1.50

You are correct! The Split-Screen-layout is deprecated since the SAPUI5 version 1.48.

UNIT 12 SAP Fiori Design Guidelines

UNIT OBJECTIVES

- Understand floorplans as defined in the SAP Fiori guidelines
- Understand list report, as defined in the SAP Fiori guidelines

- Understand object page
- Understand worklist, as defined in the SAP Fiori guidelines
- Understand object view, as defined in the SAP Fiori guidelines
- Understand wizard floorplan, as defined in the SAP Fiori Guidelines
- Understand the overview page, as defined in the SAP Fiori Guidelines
- Understand draft handling, as defined in the SAP Fiori guidelines
- Understand SAP Fiori locking

Unit 12
Lesson 1

Understanding Floorplans as Defined in the SAP Fiori Guidelines

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand floorplans as defined in the SAP Fiori guidelines

Design Guidelines for Floorplans

A floorplan is a UI layout that provides a consistent user experience for a specific task or operation.

The consistent use of floorplans provides a high recognition value to the user.

SAP identified various common user tasks of daily work. UI patterns were identified to be implemented and create a consistent UI over application boundaries.

The Initial Page .

Problem

- You want to provide an entry point where the user can search and navigate to a found object from the search result.

Solution

- The initial page provides a single page as an interaction starting point. The user can start with the selection of an object and navigate to the object required.

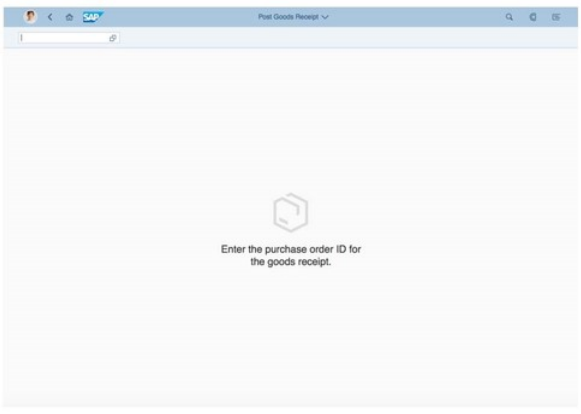


Figure 148: Initial Page

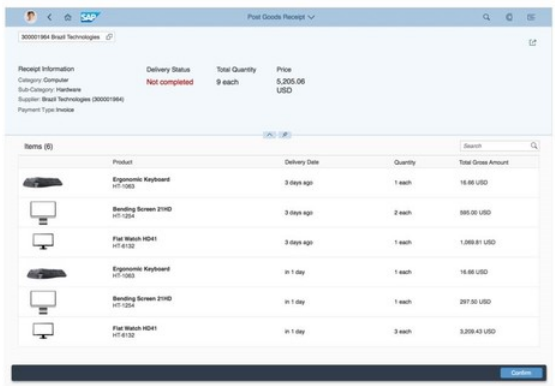


Figure 149: Initial Page

The initial page floorplan is used when the user needs to navigate to a single object.

An interaction or starting point, is a single input field that directs the user to the object they are looking for in a few steps.

An input field might be value help, or a search as you type field.

Do not use this floorplan when you want to show more than one object. In this case, the list report floorplan is a more appropriate solution.

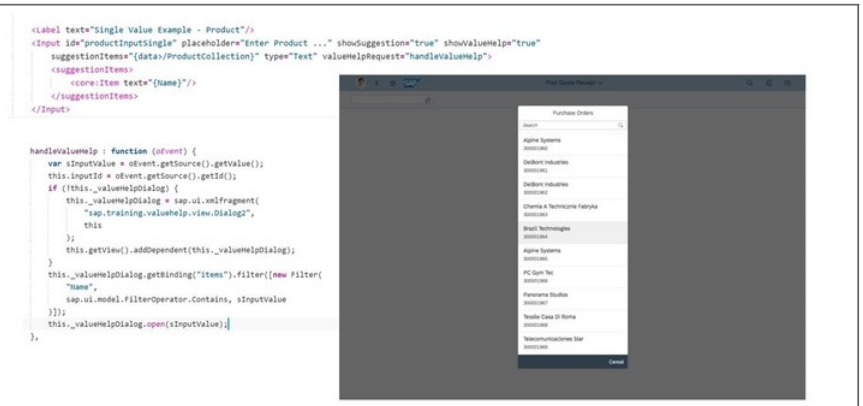


Figure 150: Value Help

Figure 151: Value Help – Tabular (1 of 2)

Figure 152: Value Help – Tabular (2 of 2)

LESSON SUMMARY
You should now be able to:

- Understand floorplans as defined in the SAP Fiori guidelines

Unit 12
Lesson 2

Understanding List Report, as Defined in the SAP Fiori Guidelines

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand list report, as defined in the SAP Fiori guidelines

List Report

Decisions on which floorplans to use depend on what problem you are trying to solve.

Problem

You want to display a large collection of items and the user should be able to take some actions.

Solution

The List Report Floorplan can show a large collection of items and provide some actions to the user.

List report allows users to work with a large list of items and take action.

- It provides functions for filtering large lists.
- It provides different ways of displaying the items.
- It is only applicable when using full screen layout

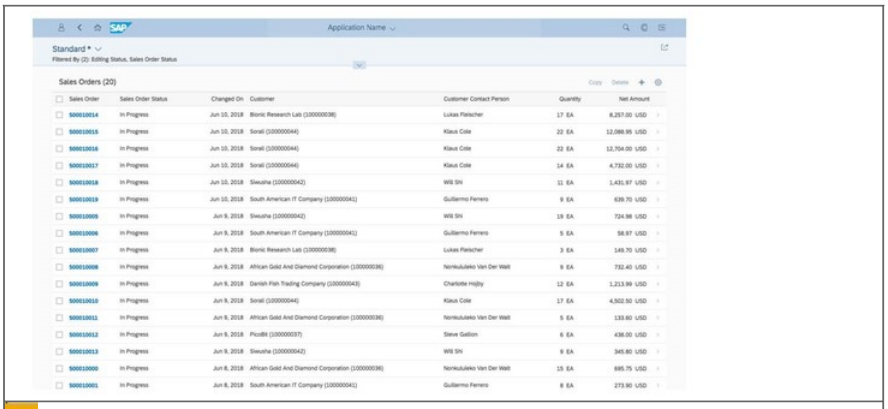


Figure 153: List Report

Use the list report floorplan if:

- Users need to find and act on relevant items within a large set of items by searching, filtering, sorting, and grouping.
- Users need to display the whole dataset using different visualizations, (for example, as a table or a chart), without requiring interactions between these visualizations. An example user case might be reporting.
- Users need to work with multiple views of the same content, for example on items that are "Open", "In Process", or "Completed". Views can be switched using tabs, segmented buttons, or a select control.
- Drill-down is rarely ever used, or is only available by navigation to another page, and not free or flexible to drill down within the page itself.
- Users work on different kinds of items.

Do not use the list report floorplan if:

- Users need to see or edit one item with all its details. Use the object page floorplan instead.
- Users need to find one specific item, and the item, or an identifying data point, is known to the user, (such as a barcode). Use the initial page floorplan instead.
- Users need to work through a comparably small set of items, one by one. Use the worklist floorplan instead.
- Users need to extract knowledge or insights from data, either to better understand the current situation, or to identify the root cause for a certain value. Use the analytical list page instead.
- Charts are not only used for visualization. Users need to switch between integrated chart and table views (hybrid view). Use the analytical list page instead.
- Users need to see the impact of their action on a KPI. Use the analytical list page instead.
- Users need to see, not only the result, but also the impact of their filter settings directly in a chart representation. Use the analytical list page instead.

Figure 154: List Report – General Structure

These functions are offered by the filter bar control (sap.ui.comp.smartfilterbar.SmartFilterBar), which is mandatory for any list report floorplan.

When you use the filter bar as part of the list report floorplan, you do not need to make any specific modifications. For more information, see the guideline for the filter bar itself.

The variant management functions allow users to define and manage predefined filter sets. They can set a default variant, and specify how the variant is executed. If **Execute** on **Select** is active, the variant is executed as soon as the user selects it (live update). In this case, the content area is populated automatically. If execute on select is not active, the user can modify the query, but has to execute the search manually to display the data (delayed update).

In the filter area, users specify the data that is displayed in the content area:

- Use different filter criteria to reduce the number of items displayed.
- Select predefined filter sets (Variants) to support different user cases.
- Filter with the filter bar expanded or collapsed.

Figure 155: The Filter Area

The Content Area

- The content area has a grid table and analytical table on the desktop and tablet.
- It has a responsive table on the smart phone.
- The content area has a tree table on the desktop and tablet.
- The content area uses a smart table as a wrapper.

Simple Content

In most cases, the content consists of just a table toolbar and a table. If required, provide an option to switch between the table and a corresponding chart view.

Multiple Views

For more complex scenarios, provide multiple views of the same content. Multiple views involve one or more of the following:

- Showing the same table, but with different columns
- Showing the same table in different pre-filtered states

  These states are usually based on a status column, for example, items that are Open, In Process, or Closed. Make sure the corresponding filter is not offered on the filter bar.
- Differentiating between the items displayed in the content in some other fundamental way

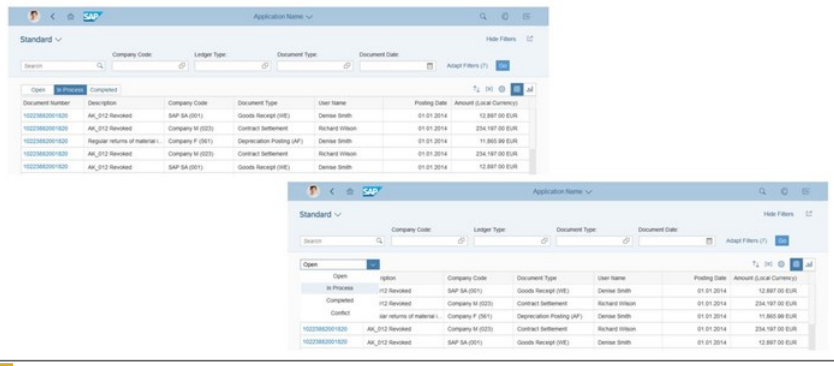


Figure 156: The Content Area

Table Actions

Add, remove, edit, trigger object function, change status of one or multiple

Line Item Actions

Start-stop batch job, approve item, assign item

Global Actions

Actions that affects all or selected items on the page

Actions for app

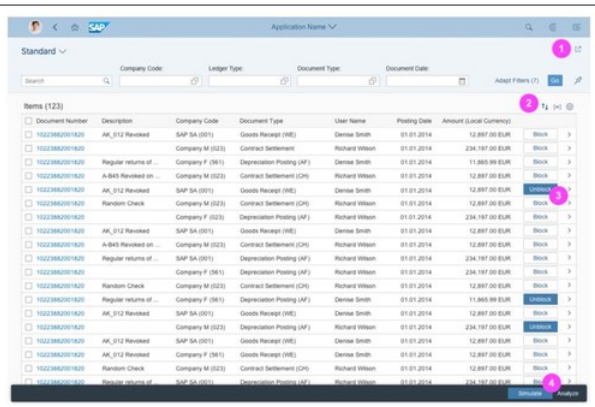


Figure 157: Actions

Variant Management

- Is optional
- If used, applied to the whole page
- Used to save and restore all settings
- Allows users to choose whether a variant should be executed automatically or not

- If now variant management is needed, you have to show a title that describes the current view state

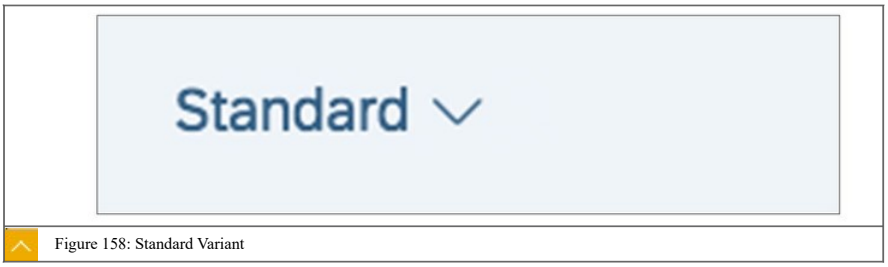


Figure 158: Standard Variant

LESSON SUMMARY
You should now be able to:

- Understand list report, as defined in the SAP Fiori guidelines

Unit 12
Lesson 3

Understanding the Object Page

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand object page

Object Page

Different problems require different solutions.

Problem

You want to display all the information of a simple or complex object with different facets in a responsive way.

Solution

The object page floorplan defines a floorplan that fits into full-screen and master-detail applications. It defines a stable UX when displaying complex objects with different facets.

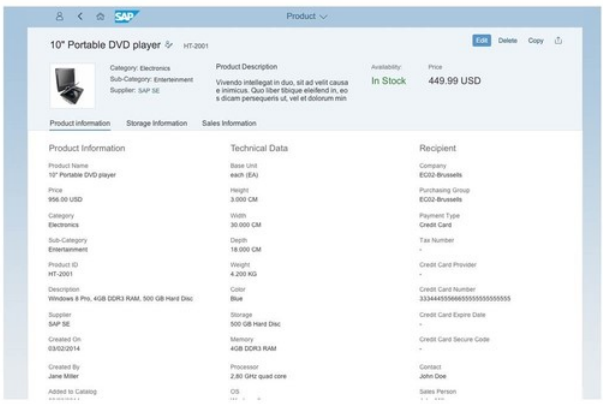


Figure 159: Structure of the Object Page Floorplan

Implementation Options

Use the pre-built SAP Fiori element. This implementation uses OData annotations. It allows you to speed up development if the supported feature set match your requirements. SAP Fiori elements do provide specific features that can be easily used.

Implement the floorplan using the respective SAPUI5 controls.

Figure 160: First Structure

The header consists of the following elements:

1. Title, (mandatory)
2. Subtitle, (optional)
3. Global actions, (optional)
4. Object marker, (optional, placed in the key information container of the dynamic header)
5. Header content, (optional)
6. Breadcrumbs, (optional)
7. Visual indicator, (mandatory if the header can be collapsed and expanded)

When the object page is used in the flexible column layout, it can also contain navigation actions.

Note:
To display images and placeholders in the header, use the avatar control. The subtitle is now below the title. (For the former object page header it was next to the title.

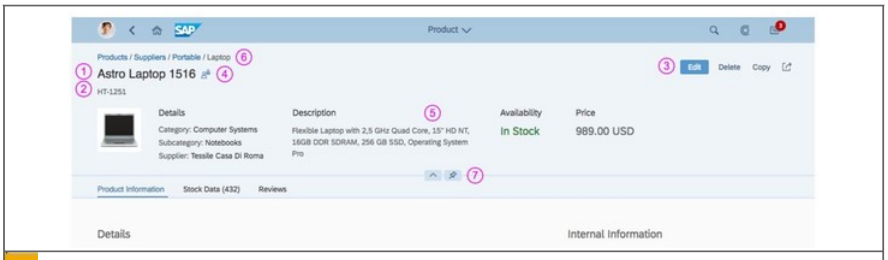


Figure 161: Second Structure

Use the object page floorplan when:

- users need to see, edit, or create an item with all its details.
- users need to get an overview of an object and interact with different parts of the object.

Don't use the object page floorplan if:

- users want to edit several items at the same time, use list report floorplan.
- users need to find relevant items without knowing the exact item details, use list report floorplan.
- users need to find one specific item, where the item or an identifying data point is known to the user, use initial page floorplan.
- users need to be guided through a series of steps when a new object is created, use wizard floorplan.

LESSON SUMMARY
You should now be able to:

- Understand object page

Unit 12
Lesson 4

Understanding Worklist, as Defined in the SAP Fiori Guidelines

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Understand worklist, as defined in the SAP Fiori guidelines

Worklist

Review when to work with the worklist floorplan.

Problem:

You want to display a collection of items and process them or delegate them to someone else.

Solution:

The worklist floorplan defines a format that allows you to show a collection of items, display details of each item and process or delegate each of these items.

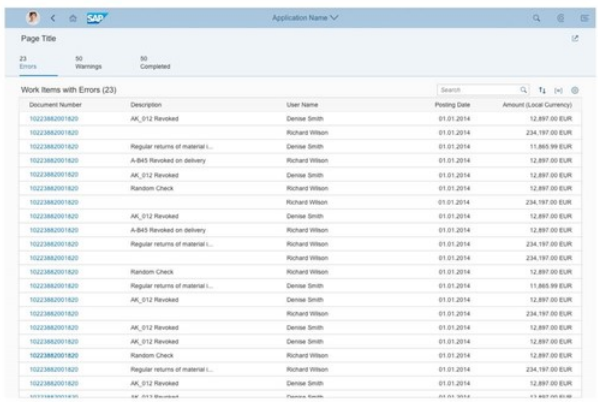


Figure 162: Worklist Implementation

Use the worklist floorplan if:

- You want to give the user a direct entry point for taking action on work items.
- The user needs to decide which work item to process first.
- Your app uses the full screen layout.

Do not use the worklist floorplan if:

- The items you are showing are not work items
- Your app uses the split-app layout
- You want to show a large items list with different data visualizations, (use list report instead)

Use of the Worklist floorplan

Use the worklist floorplan if:

- The user has numerous potential work items and needs to decide which ones to process first.
- You want to give the user a direct entry point for taking action on work items.
- Users need to work with multiple views of the same content, (for example, items that are "Open", "In Process", or "Completed".
- Users may switch between views using the tab bar.

Do not use the worklist floorplan if:

- The items you are showing are not work items.
- You want to show large item lists, or combine different data visualizations (charts or tables). In this case, use the list report floorplan instead.
- Users need to find and act on relevant items from within a large set of items by searching, filtering, sorting, and grouping. Use the list report floorplan instead.
-

The worklist floorplan is based on the dynamic page. In addition to the SAP Fiori launchpad shell bar, the dynamic page contains the following areas:

The **header title** : Use this to display a title or the variant for the whole page, KPI information (if relevant), and a header toolbar with global actions, such as Share.

The **content area** : Use this to display:

- An icon tab bar in the content area
- One table (per tab)
- One or multiple tables. You can use any kind of table and list. To ensure that the application runs on all devices, we recommend using the responsive table.

The **footer toolbar:** If required, use a footer toolbar to display the messaging button and finalizing actions.

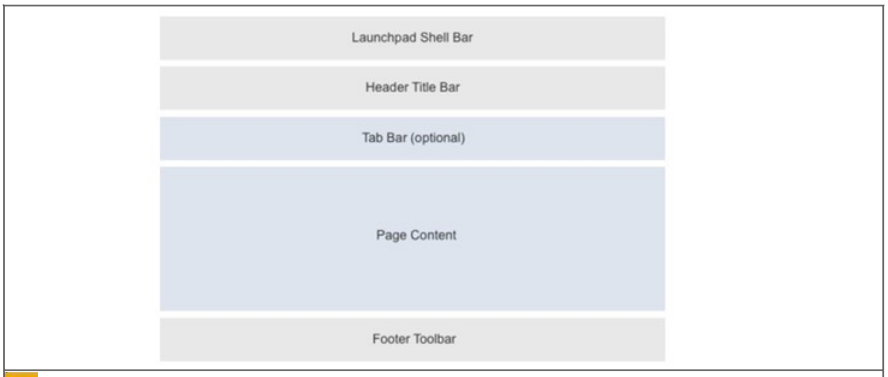


Figure 163: Structure of a Worklist

Worklist floorplan Types

Simple Worklist: Used when working down a list of items.

Category Worklist: Use the icon tab bar to provide a categorized access to a specific list of items.

KPI Worklist: Allows the user to track a KPI while processing the worklist. The KPI can be displayed in the subheader of the application.

LESSON SUMMARY
You should now be able to:

- Understand worklist, as defined in the SAP Fiori guidelines

Unit 12
Lesson 5

Understanding Object View, as Defined in the SAP Fiori Guidelines

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand object view, as defined in the SAP Fiori guidelines

Object View

> Note:
> The object view floorplan has become deprecated with the introduction of the object page floorplan. If you are using this floorplan you need to update your implementation and use object page floorplan instead.

Problem:

You want to display all the information of a simple or complex object with different facets in a responsive way

Solution:

The object view floorplan defines a floorplan that fits into full screen an master-detail applications. It defines a stable UX when displaying complex objects with different facets.

Use the object view floorplan when:

- You have objects with up to 5-7 facets.
- You want to display complex object with minimal navigation.
- You want to combine a long table with other object facets, but keep the table visible.
- You are building a typical approval app.

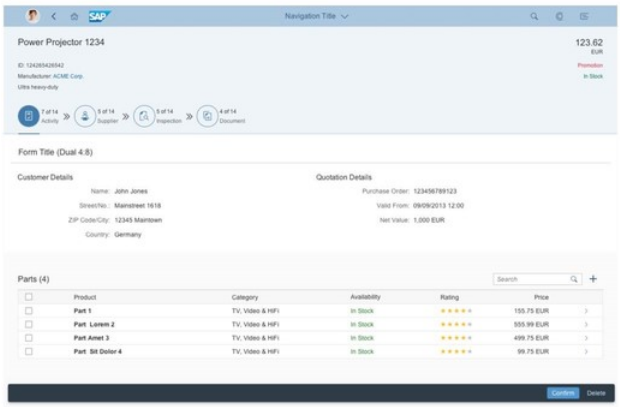


Figure 164: Object View

The Object View implementation structure consists of the following aspects:

- Object header
- Tabs (optional)
- Content area (lists, table, tree table charts)
- Footer toolbar

Figure 165: Structure of the Object View

Common guidelines of object view

- Only use the two icon tab bar types icon tabs and text only in the object view. Do not use the other types.
- Do not switch the actions in the footer toolbar when the user switches tabs. The footer toolbar is global and constant. If you have tab-specific actions, place them in the tab content area.

- You can hide empty tabs, but only if there is no way of adding content to the tab. For example, you still need to show an empty Attachments tab to enable users to upload attachments.
- Try to persist the state of selected tabs during a session so the user finds a similar view while navigating between instances.

LESSON SUMMARY
You should now be able to:

- Understand object view, as defined in the SAP Fiori guidelines

Unit 12
Lesson 6

Understanding Wizard Floorplan, as Defined in the SAP Fiori Guidelines

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand wizard floorplan, as defined in the SAP Fiori Guidelines

Wizard Floorplan

Problem

The user has to process a long and unfamiliar task.

Solution

Divide the task into different sections. Guide the user step by step through the data editing.

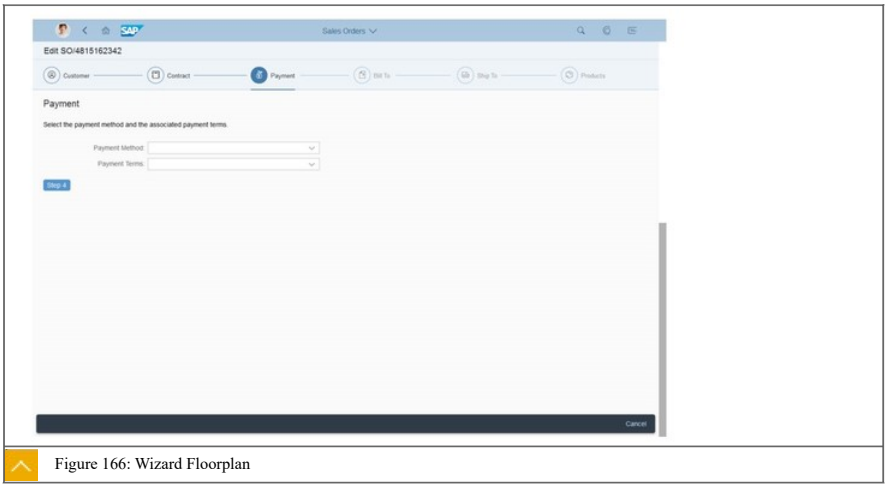


Figure 166: Wizard Floorplan

When to use the wizard floorplans

- The wizard aims to help users by dividing large or complex tasks into segments. Use the wizard if the user has to accomplish a long task (such as filling out a long questionnaire) or a task that is unfamiliar to the user.
- The flow should consist of a minimum of 3 and a maximum of 8 steps.

- The wizard can be used for both create and edit scenarios. If your application contains both, consider using the same method for both scenarios – either the wizard, or another create or edit screen, (edit flow or object page).

When not to use the wizard floorplans

- If you have a task with only 2 steps or a format that the user is familiar with, for example, it is part of their daily routine, do not use the wizard as it only adds unnecessary clicks to the process.
- If your process needs more than 8 steps, the wizard will not support those steps as the process is too long and can be confusing for the user. In this case, you should consider restructuring the task.
- Consider if the classic edit screens (edit flow or object page), are more suitable for your use case.

Structure of the wizard floorplan walk-through screen.

1. After calling the wizard, the first step of the floorplan appears.
2. When all necessary fields are completed, a button labelled Step # appears.
3. When completing the last step, a button labelled Review appears.
4. The footer of the wizard holds the cancel button with standard SAP Fiori cancel behaviour.
5. It is possible to have a save or draft button when the form is very long

Displayed is the structure of the wizard floorplan summary page. The review button takes the user to the screen displaying the data.

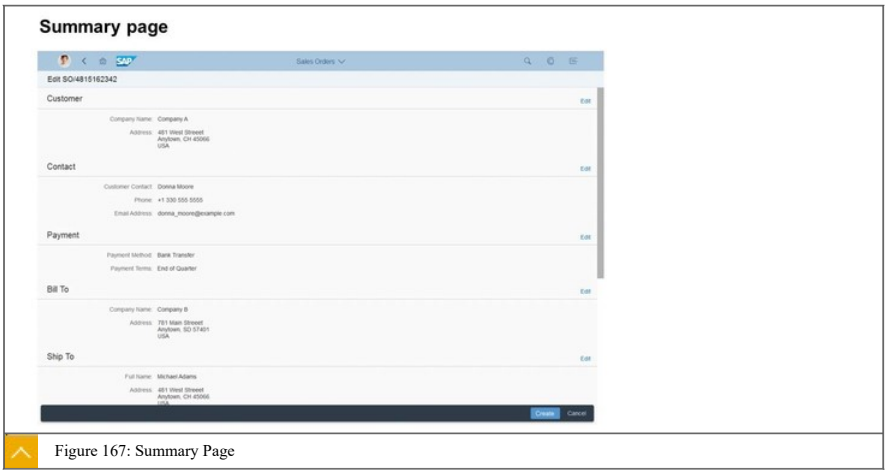


Figure 167: Summary Page

LESSON SUMMARY
You should now be able to:

- Understand wizard floorplan, as defined in the SAP Fiori Guidelines

Unit 12
Lesson 7

Understanding the Overview Page, as Defined in the SAP Fiori Guidelines

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the overview page, as defined in the SAP Fiori Guidelines

Overview Page

Highlighted are the differences between an overview page and an object page.

Overview Page	Object Page
Role-based	Object-based
"Street-level" view	"Details" view
Heterogeneous information	Homogeneous information

Figure 168: Overview Page Versus Object Page

The floorplan layout presents a mix of card types relevant to individual roles.

- Analytical Card
- List Card
- Bar Chart List Card
- Link List Card
- Table Card
- Stack Card
- Quick View Card

Figure 169: Card Types

Make a conscious decision on the number of cards: Show only cards that really support the specific role context or task.

Accentuate the most important information: Semantic colors in texts, charts, and images attract more attention.

Offer a well-balanced mixture of card types: Diversity makes it easy to recognize, select, and read information.

Define a deliberate card order: Users assume that cards at the top of the page are more important than others.

Group similar topics: Users assume that related cards will be shown next to each other.

Choose easy-to-read and actionable texts: If the user needs to take action, use the active voice (for example, "Reorder Soon" when stocks are running low).

Be semantically consistent: Users expect crucial terms like “urgent” or “out of stock” to be highlighted with semantic colors.

- Make a conscious decision on the number of cards.
- Accentuate the most important information.
- Offer a well-balanced mixture of card types.
- Define a deliberate card order.
- Group similar topics.
- Choose easy-to-read and actionable texts.
- Be semantically consistent.

Figure 170: Best Practice

LESSON SUMMARY
You should now be able to:

- Understand the overview page, as defined in the SAP Fiori Guidelines

Unit 12
Lesson 8

Understanding Draft Handling, as Defined in the SAP Fiori Guidelines

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand draft handling, as defined in the SAP Fiori guidelines

Draft Handling

We review draft handling in the development environment.

Table 3: Stateful Apps and Locking

Scenario	Result
Stateful apps work with an in-memory copy of data	The life of the copy is tied to the UI session, the UI owns pessimistic locks.
When editing data in a browser, the data is stateless	If the browser is closed or loses its connection to the back-end server, all data modifications are lost.
In a stateful app the in-memory copy is replaced with a "draft" document that is stored in a database	The draft document is the owner of pessimistic locks.
The draft concept retains all data even if the browser is closed	No data is lost.

Basic draft types include:

- New drafts – for initial data entry, where no active documents yet exist.
- Edit drafts – for editing an existing active document.

Behavioural draft types include:

- Exclusive drafts – limit visibility to a specific user.
- Shared drafts – may become visible to multiple users, but modified by only one user at a time.
- Collaborative drafts – visible and modifiable by multiple users simultaneously.

Each behavioural type has both a new and edit version; for example, there is a `Shared New` and a `Shared Edit` draft type.

The Manage Products app is the only one that contains drafting code; specifically the product edit screen

When the user logs back into the app, they are notified of the unsaved draft and are given the option to resume working on the draft, or discard the draft.

Figure 171: Drafting In Sample Apps

LESSON SUMMARY
You should now be able to:

- Understand draft handling, as defined in the SAP Fiori guidelines

Unit 12
Lesson 9

Understanding SAP Fiori Locking

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand SAP Fiori locking

SAP Fiori Locking

The concept of locking ensures that business objects can be changed by only one person at a time, ("exclusive editing").

If a user is already working on an object, it is locked so other users cannot edit at the same time.

The object remains locked until the user has finished editing.

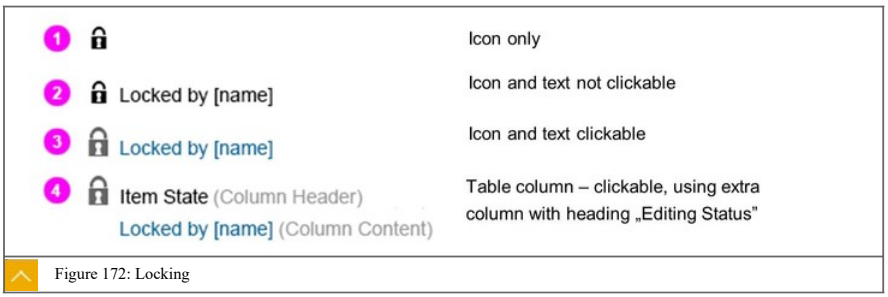


Figure 172: Locking

Optimistic Locking Control – OData and ETags

OData uses HTTP ETags for optimistic concurrency control. A few special considerations apply for ETags:

- When retrieving an entry the server returns an ETag.
- When retrieving a single entry, the ETag returned as a HTTP response header named ETag.
- The server can choose to include the ETag to the body.
- When getting several entries in a feed, the ETag value is included as metadata in the entry itself.
- For POST, PUT and merge operations, the server computes a new ETag and returns it in a response header.

When sending a PUT, MERGE or DELETE request, clients must indicate an ETag in the If-Match HTTP request header.

- If it is acceptable to overwrite any version of the entry in the server, then the value "*" may be used.
- If a given entry has an ETag and a client attempts to modify or delete the entry without an If-Match header, the request should fail with a HTTP 412 response code.

OData servers will often use weak ETags as a way of indicating that two resources may be semantically equivalent but a particular request may see a different representation of it.

The following interfaces provide methods for ETag handling:

- `/IWBEP/IF_MGW_APPL_SRV_RUNTIME method GET_IS_CONDITIONAL_IMPLEMENTED`
- `/IWBEP/IF_MGW_REQ_ENTITY_U method GET_CONDITIONAL_INFO`
- `/IWBEP/IF_MGW_REQ_ENTITY_D method GET_CONDITIONAL_INFO`

The figure gives a review of the syntax of the update of the OData model.

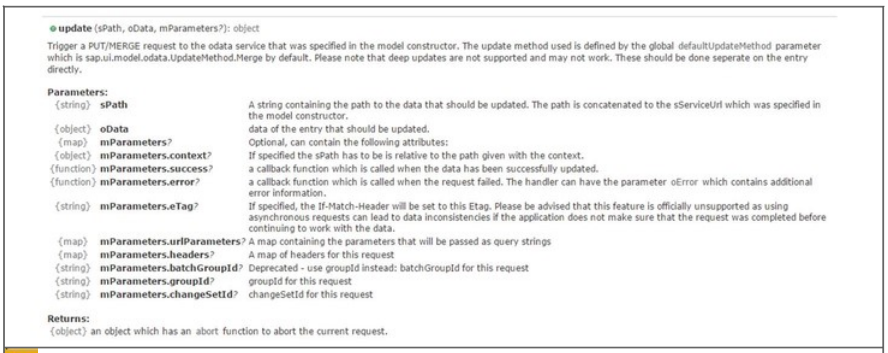


Figure 173: Optimistic Locking Control

Notice the last parameter is an ETag.

ETags allow us to perform optimistic locking control for our apps.

LESSON SUMMARY
You should now be able to:

- Understand SAP Fiori locking

7. What problem is solved by the wizard floorplan?

   Choose the correct answer.

   - ☐ **A** You want to display all the information of a simple or complex object, with different facets, in a responsive way.
   - ☐ **B** You want to display a collection of items and process them or delegate them to someone else.
   - ☐ **C** The user must process a long and unfamiliar task.
   - ☐ **D** You want to display a large collection of items, and the user can take some action.

8. What is the nature of the overview page floorplan?

   Choose the correct answer.

   - ☐ **A** You want to display all the information of a simple or complex object with different facets, in a responsive way.
   - ☐ **B** You want to display a collection of items, and process them or delegate them to someone else.
   - ☐ **C** You want to show all information that is needed by the user on a single page, based on a user specific domain or role.
   - ☐ **D** You want to display a large collection of items, and the user can take some action.

9. When do you not use the overview page floorplan?

   Choose the correct answers.

   - ☐ **A** A high-level or birds-eye view of an application content is sufficient.
   - ☐ **B** You want to display a collection of items and process them or delegate them to someone else.
   - ☐ **C** You want to show information about one object only. In this case, use the object page instead.
   - ☐ **D** You just represent one application and less than three cards.

Unit 12

Learning Assessment - Answers

1. What is a SAP Fiori floorplan?

   Choose the correct answer.

	A A floorplan is a layout control of SAPUI5.
X	**B** A floorplan is a UI pattern that provides consistent UX.
	C A floorplan defines the overall layout of an application.
	D A floorplan is a layout of the SAP Web-Dynpro layouts for SAP Fiori apps.

   You are correct! A floorplan is a UI layout that provides a consistent user experience for a specific task or operation.

2. When do you have to implement the SAP Fiori floorplan initial page?

   Choose the correct answer.

	A The Initial page floorplan is used when the user needs to select an item from a list.
X	**B** The Initial page floorplan is used when the user needs to navigate to a single object.
	C The Initial page floorplan is used to show a single object.
	D The Initial page floorplan is used when the user needs to process a complex task.

   You are correct! The Initial page floorplan is used when the user needs to navigate to a single object.

3. What SAPUI5 control is used as a starting point to implement the initial page?

   Choose the correct answer.

	A sap.m.Select
X	**B** sap.m.Input
	C sap.m.Table
	D sap.m.List

   You are correct! The sap.m.Input UI-control is used to implement an initial page floorplan.

4. What problem is solved by the list report floorplan?

   Choose the correct answer.

   - [ ] **A** You want to display all the information of a simple or complex object with different facets in a responsive way.
   - [ ] **B** You want to display a collection of items and process them or delegate them to someone else.
   - [ ] **C** The user needs to process a long and unfamiliar task.
   - [x] **D** You want to display a large collection of items and the user can take some action.

   You are correct! The list report floorplan is used when you want to display a large collection of items and the user can take some action.

5. What problem is solved by the object page floorplan?

   Choose the correct answer.

   - [x] **A** You want to display all the information of a simple or complex object with different facets in a responsive way
   - [ ] **B** You want to display a collection of items and process them or delegate them to someone else.
   - [ ] **C** The user must process a long and unfamiliar task.
   - [ ] **D** You want to display a large collection of items and the user can take some action.

   You are correct! The object page floorplan is used when you want to display all the information of simple or complex object with different facets in a responsive way.

6. What problem is solved by the worklist Report floorplan?

   Choose the correct answer.

   - [ ] **A** You want to display all the information of a simple or complex object with different facets in a responsive way.
   - [x] **B** You want to display a collection of items and process them or delegate them to someone else.
   - [ ] **C** The user must process a long and unfamiliar task.
   - [ ] **D** You want to display a large collection of items and the user can take some action.

   You are correct! The work list report floorplan is used when you want to display a collection of items and process them or delegate them to someone else.

7. What problem is solved by the wizard floorplan?

Choose the correct answer.

- [ ] **A** You want to display all the information of a simple or complex object, with different facets, in a responsive way.
- [ ] **B** You want to display a collection of items and process them or delegate them to someone else.
- [X] **C** The user must process a long and unfamiliar task.
- [ ] **D** You want to display a large collection of items, and the user can take some action.

You are correct! The wizard floorplan is used when the user must process a long and unfamiliar task.

8. What is the nature of the overview page floorplan?

Choose the correct answer.

- [ ] **A** You want to display all the information of a simple or complex object with different facets, in a responsive way.
- [ ] **B** You want to display a collection of items, and process them or delegate them to someone else.
- [X] **C** You want to show all information that is needed by the user on a single page, based on a user specific domain or role.
- [ ] **D** You want to display a large collection of items, and the user can take some action.

You are correct! The overview page (OVP) is a data-driven SAP Fiori app type and floorplan that provides all the information a user needs in a single page, based on the user's specific domain or role.

9. When do you not use the overview page floorplan?

Choose the correct answers.

- [X] **A** A high-level or birds-eye view of an application content is sufficient.
- [ ] **B** You want to display a collection of items and process them or delegate them to someone else.
- [X] **C** You want to show information about one object only. In this case, use the object page instead.
- [X] **D** You just represent one application and less than three cards.

You are correct! Do not use the overview page if a high-level or birds-eye view of application content is sufficient. When you just want the user to launch an application, use the SAP Fiori launchpad home page instead. When you want to show information about one object only, use the object page instead. When you just want to represent one application and less than three cards, use the object page instead.

UNIT 13

SAP Fiori Extension Concept

UNIT OBJECTIVES

- Explain extension points in SAPUI5
- Implement view extension, modifications, and replacement
- Implement controller extensions and hook methods
- Extend translations with customer properties

Unit 13
Lesson 1

Explaining Extension Points in SAPUI5

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Explain extension points in SAPUI5

SAP Fiori Extension Points

The SAP Fiori extension concept identifies areas of an app that can be extended. It is based on the configuration of components. The standard app remains unchanged and considered modification free. A customization can survive an application upgrade.

Several aspects can be customized:

- Views can be extended or replaced by custom ones.
- Controllers can be extended.
- I18n resources are available to extend apps.
- Additional views can be added.
- New navigation paths can be added.
- Navigation routes can be customized.

A number of aspects of the interface have been identified where extensions can be implemented.

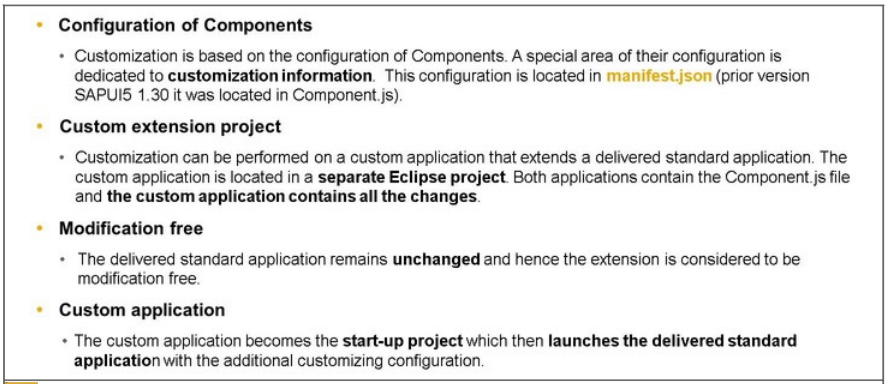


Figure 174: UI Extensibility

The figure, Model-View-Controller and Extension Possibilities, illustrates this relationship.

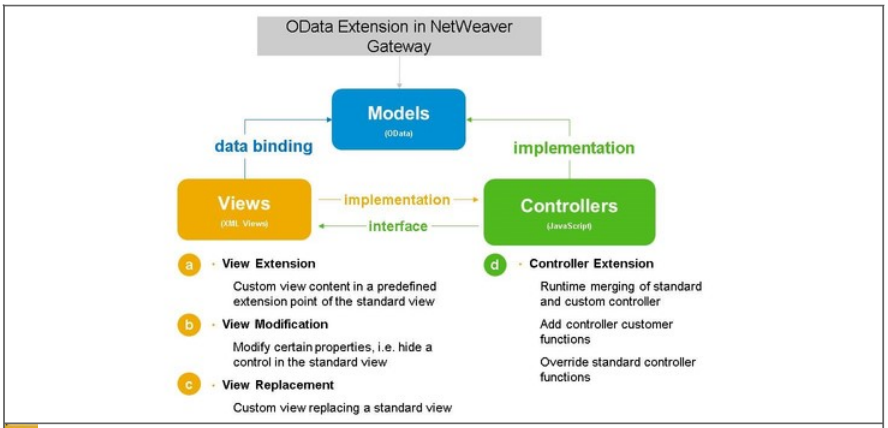


Figure 175: Model-View-Controller (MVC) and Extension Possibilities

The extension points provided can be checked in the SAP Fiori library.

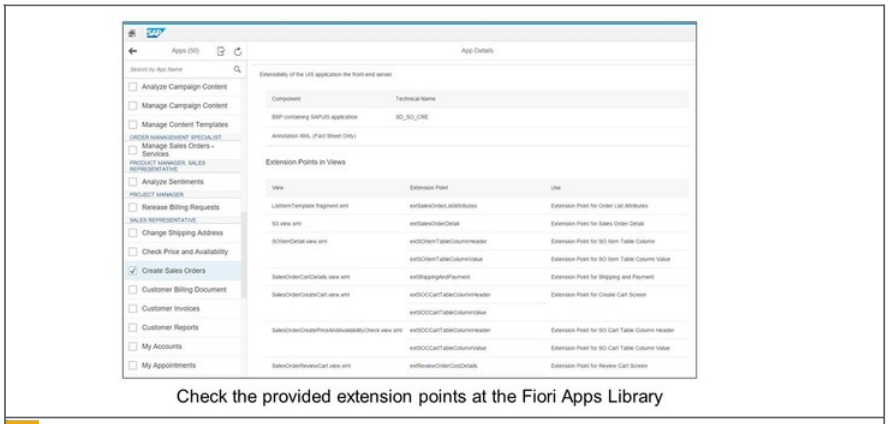


Figure 176: Find Out What Extensibility is Provided by the App

Configuration of Components

Customization is based on the configuration of components. A special area of their configuration is dedicated to customization information. This configuration is located in a JavaScript file named `Component.js` (or Configuration.js).

Custom extension project

Customization can be performed on a custom application that extends a delivered standard application. The custom application is located in a separate project. Both applications contain the `Component.js` , (or Configuration.js) file, and the custom application contains all the changes.

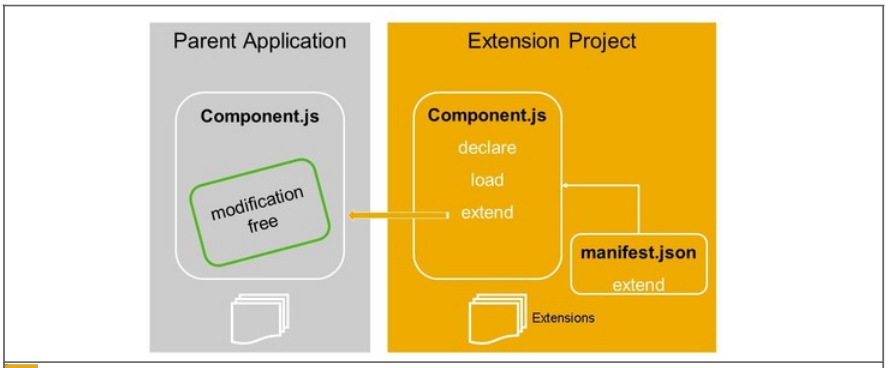


Figure 177: UI Extension Project

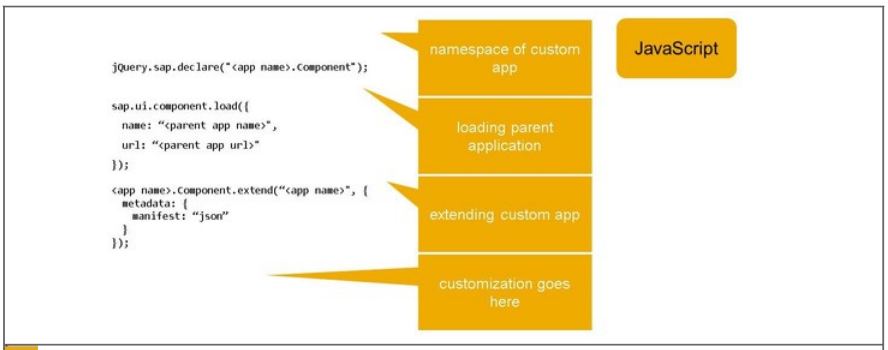


Figure 178: Component.js

Modification free

As the delivered standard application remains unchanged, the extension is considered to be modification free.

Custom application

The custom application becomes the start-up project, which then launches the delivered standard application with the additional customizing configuration.

For more details, see: https://sapui5.netweaver.ondemand.com/sdk/#docs/guide/Customization.html

For each provided extension possibility, there is a separate wizard that generates the necessary files.

A extension project may contain different extensions for one SAP Fiori application.

In these situations, follow the procedure to create a wizard.

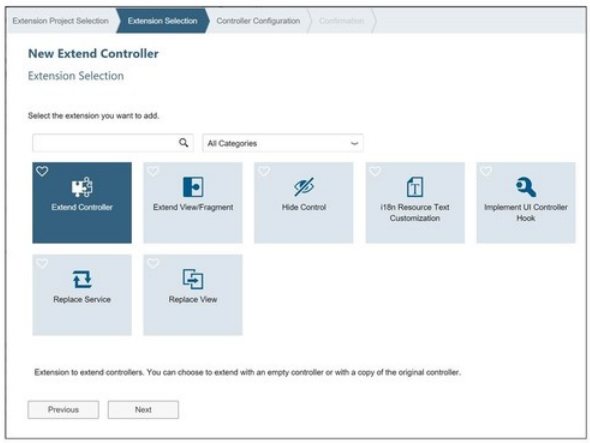


Figure 179: Extension Options in the SAP Web IDE

These are the methods that build upon the standard application `Component.js` file, allowing the original to be extended by the extension project.

The `viewModifications` used to hide or show different properties are limited.

- Extension configuration exists within manifest.json of the extended application by utilizing the following **properties**:
  - **sap.ui.viewExtensions** is used to provide custom view content in a specified extension point withn the standard delivered application.
  - **sap.ui.viewModifications** is used for overriding certain properties of controls in the standard delivered application.
  - **sap.ui.viewReplacements** is used for replacing a view in a delivered standard application with a custom view.
  - **sap.ui.controllerExtensions** is used for replacing a controller in a delivered standard application with a custom controller.

Figure 180: manifest.json Extension Configuration

The code of the manifest.json file is shown in the figure manifest.json.

```
"extends": {
    "component": "sap.training.flight",
    "extensions": {
        "sap.ui.viewExtensions": {
            "sap.training.flight.view.Bookings": {
                "bookingsextensionpoint": {
                    "className": "sap.ui.core.Fragment",
                    "fragmentName": "sap.training.flight.ZFLIGHTSYSTEM00Extension.view.Bookings_bookingsextensionpointCustom",
                    "type": "XML"
                }
            }
        }
    }
},
```

Figure 181: manifest.json

The figure, Component.js Example provides the code contained in the component.js file..

Figure 182: Component.js Example

Create a wizard
Create a wizard at an extension point

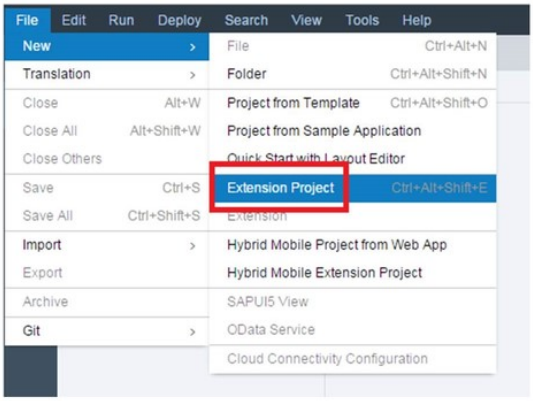
1. 

Figure 183: Create an Extension Project

2. Select the source of the SAP Fiori application.

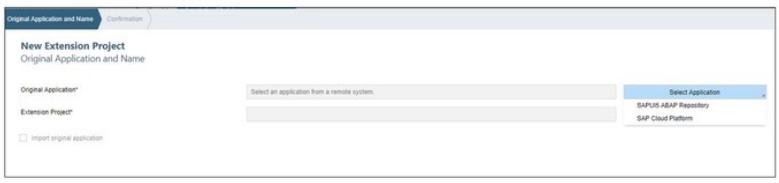


Figure 184: Repository Selection

3. Select the system.

4. Search for the application to extend.
5. Select the application for extension.

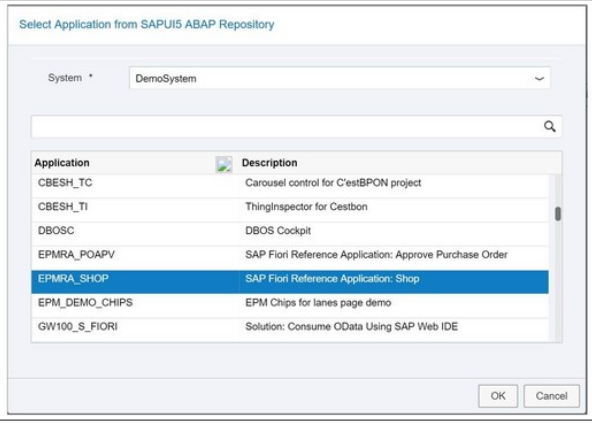


Figure 185: Select the Application to Extend

6. Set the name of the extension project

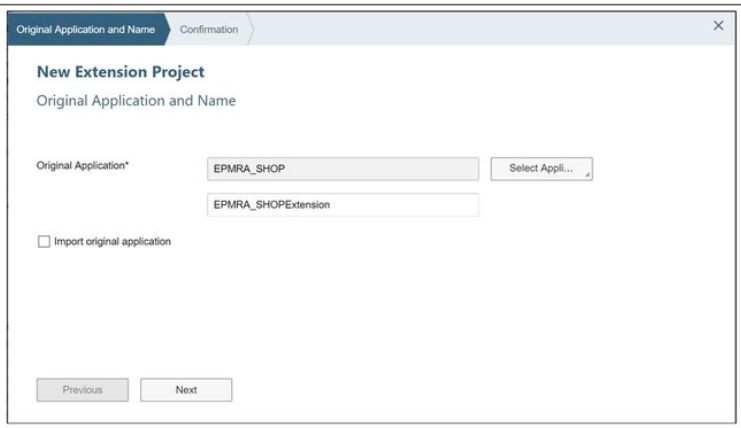


Figure 186: Name of the Extension Project

7. Uncheck the Import original application in extensibility pane if checked.
8. To finalize the creation of the extension project choose Finish .

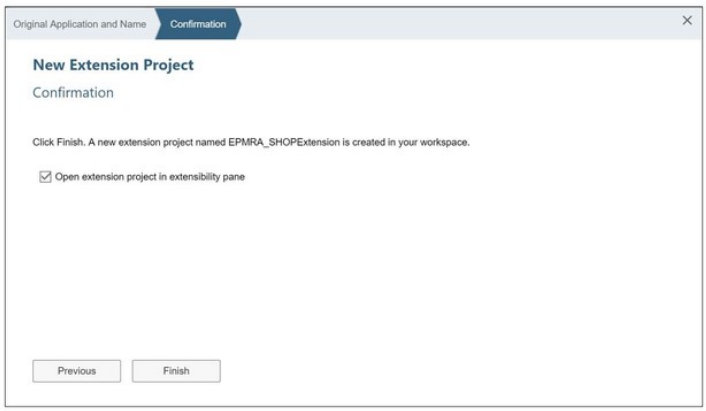


Figure 187: Create an Extension Project – Confirmation

9. View the displayed extensibility pane and select the points you want to extend.

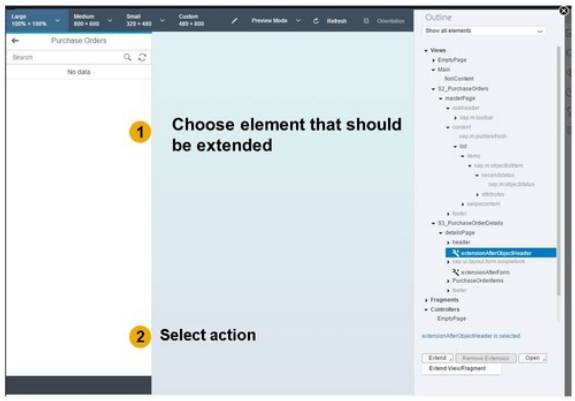


Figure 188: The Extensibility Pane

10. Having selected an extension action, navigate into the extension code or add some more extensions to the extension project.

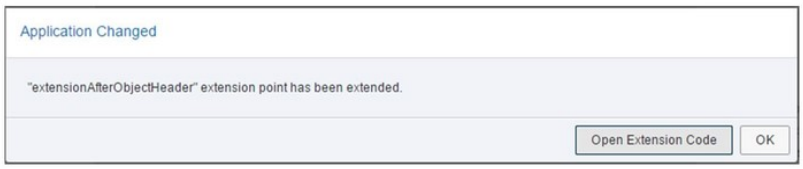


Figure 189: Open Extension Code

1. For each provided extension possibility, a separate wizard generates the necessary files.

   A extension project may contain different extensions for one SAP Fiori application.

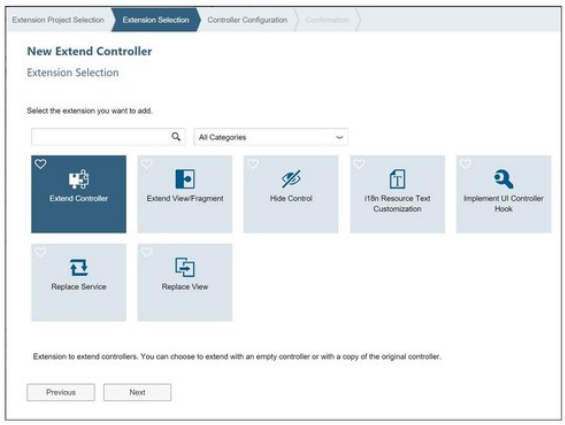


Figure 190: Extension Options in the SAP Web IDE

LESSON SUMMARY

You should now be able to:

- Explain extension points in SAPUI5

Unit 13
Lesson 2

Implementing View Extension, Modification, and Replacement

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Implement view extension, modifications, and replacement

View Extension, Modification, and Replacement

- SAPUI5 uses extension points to extend standard views with custom content.
- The assignment of a custom view to an extension point is done in component customizing.
- Extension points are inserted in a standard view to indicate where in the view, custom content can be inserted.
- In XML views, the `<ExtensionPoint>` tag is used, to be replaced by the controls provided by the customer.
- The tag should be placed in the view where a control would also be placed. Document the types of controls that are suitable.

View Extension Caveats

- If the extension point is removed, renamed, or in an area or container that is invisible under certain conditions, the view extension will no longer be applied.
- If the view name is changed, or the extension point is moved to a different view, the view extension will no longer be applied.
- If the controls around the extension point have changed, or the extension point has been moved to a different environment inside the same view, view extensions may look strange, have a broken layout or display, or not fit the new environment.
- If the updated application requires the extension to be of a certain control type, the view extension may break the application.
- If custom code relies on the presence of the extensions, the view extension may break the application.

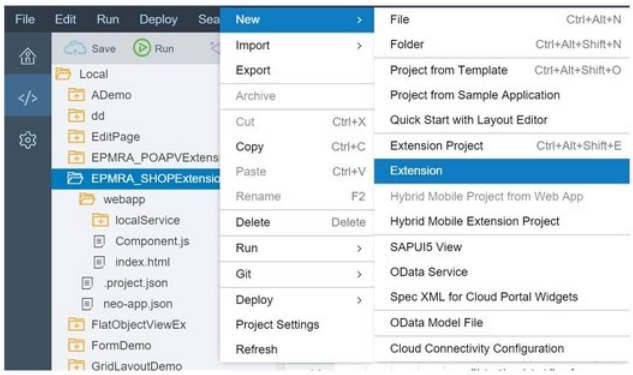


Figure 191: Add an Extension to an Existing Extension Project

To extend an existing view or fragment, choose Extend View/Fragment from the list of available extensions.

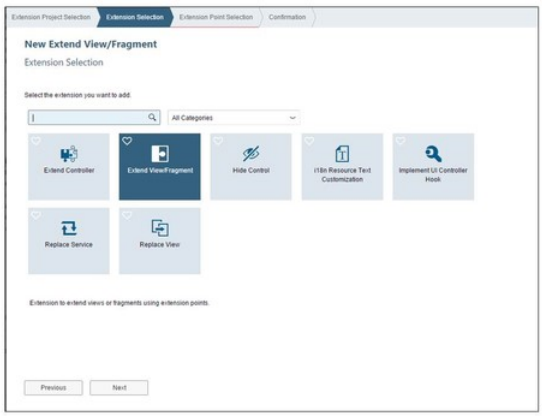


Figure 192: View or Fragment Extensions in the SAP Web IDE

View Modification

- View modification is currently restricted to the visible property of controls.
- The controls must have the visible property, and the control ID must be defined in the view.
- The view name and control ID together uniquely determine the control in the standard application.
- View modification is available for XML views, JS views, and HTML views.

Select the element you want to hide.

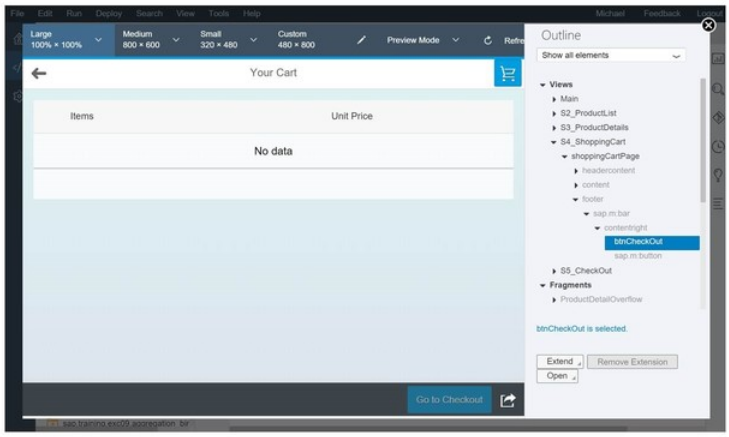


Figure 193: View Modification Within the Extensibility Pane

Choose Extend and choose to hide the control.

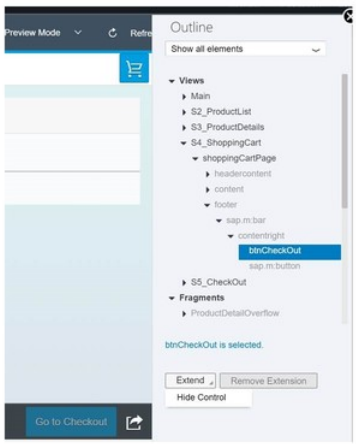


Figure 194: View Modifications

The button check out control is hidden.

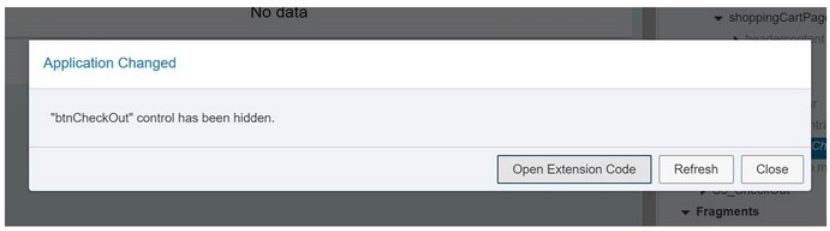


Figure 195: Working with the Extension Pane (1 of 2)

Work within the extension pane to extend your app.

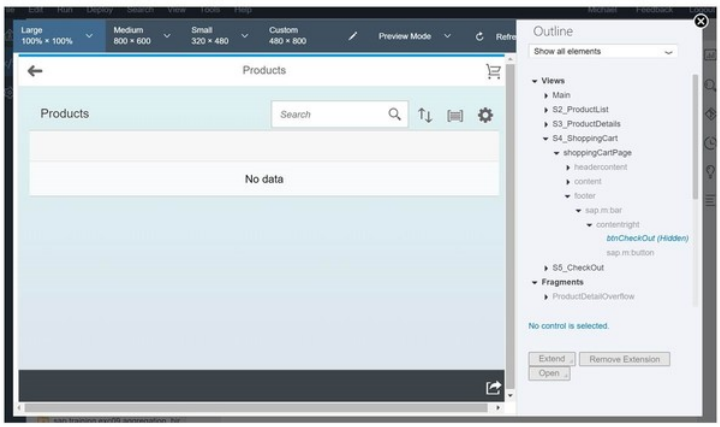


Figure 196: Working with the Extension Pane (2 of 2)

The modification can be seen in the source code.

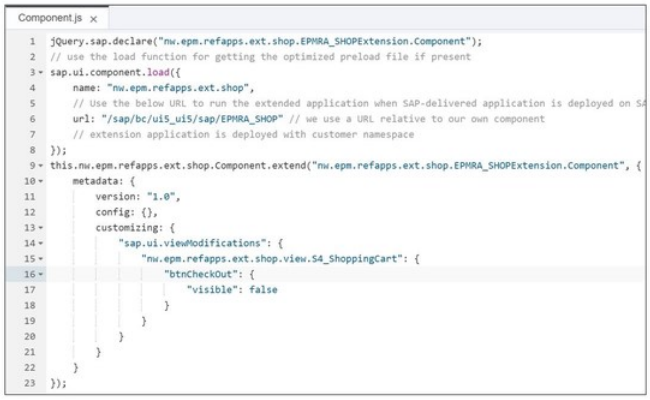


Figure 197: Result of the View Modification

View modification code indicates code that has been customized.

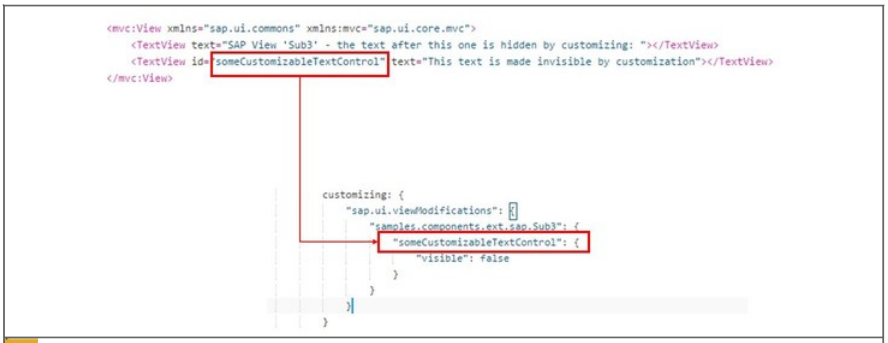


Figure 198: View Modification Example (1 of 2)

Figure 199: View Modification Example – manifest.json

View Modification Caveats

- If the original control is no longer used, has a different or no ID, a view modification will no longer be applied.
- If the original view name is changed or the view is no longer used, an invalid view modification will no longer be applied.
- However, a view modification is simply ignored and will never lead to a crash

Add an extension of type `Implement UI controller` and hook to your existing project.

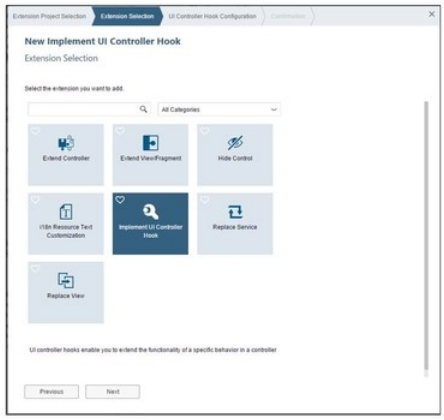


Figure 200: Controller Hook Methods in the SAP Web IDE

View Replacement

Replace delivered standard application views to adapt the application to customer needs.

Extension points are not sufficient for view replacement.

Replacement of the complete view is required.

```
customizing: {
    "sap.ui.viewReplacements": {
        "samples.components.ext.sap.Sub1": {
            viewName: "samples.components.ext.customer.CustomSub1",
            type: "XML"
        }
    },
```

Figure 201: View Replacement

View Replacement Caveats

- If the original view name is changed, or the view is no longer used, view replacements will no longer be applied.
- As long as no other custom code relies on the view to be present, view replacements should not cause a crash.

LESSON SUMMARY
You should now be able to:

- Implement view extension, modifications, and replacement

Unit 13
Lesson 3

Implementing Controller Extensions and Hook Methods

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Implement controller extensions and hook methods

Controller Extension and Hook Methods

To extend a controller, a new custom controller is provided by SAP. Methods to override standard methods with the same name. Exceptions are for the controller life cycle methods that are called in addition to the original controller method implementations. When overriding a controller method, any functionality that was previously provided by the SAP controller in this method, is no longer available. Likewise, any future changes made to the SAP controller method implementation will not be reflected in the custom controller.

Controller Extension

Extend the controller of a standard SAP Fiori application with customer specific functionality.

Methods others then the standard hook methods of the controller, are replaced by implementing a method with the same name in the custom controller.

Standard hook methods are not replaced with the corresponding methods in the custom controller. The methods are processed in the following way:

- Custom implementations of `onInit` and `onAfterRendering` are called after the corresponding method of the base controller are processed.
- Custom implementations of `onExit` and `onBeforeRendering` are called before the corresponding method of the base controller are processed.

Providing Hooks in the Standard Controller

Hooks are extension points in the controller code.

Hooks make the extension concept for controller code much more stable.

Instead of override, call the specific method from the custom controller.

Attention:

- A Hook works only for one extension layer.
- The last extension overrides any other hook implementation.

- If multi-layer extensions are needed it is recommended that middle-layer extensions provide and document their own hook function.

Providing Hooks in the standard controller

1. In the application, identify a strategic location within the controller code where customers may want to plug in and execute their customized code.
2. In the application, define a new function name which is reserved for the extension. Document the function and any arguments the function may receive or return.
3. Add code lines in the application, (see code snippet in figure for example), to check whether the function has been implemented and if so, to call the function. We also recommend implementing sanity checks for return values.
4. The customer can then configure a controller extension, implementing exactly this one function.
5. SAPUI5 runtime merges the new controller extension into the standard controller. If customizing is enabled, the new function can be executed.

Note:
This only works for one extension layer, as the most specific, or last extension overrides any other hook implementations. To allow multi-layer extensions, we recommend that middle-layer extensions provide and document their own hook functions.

This also requires flat, non-inherited controllers defined with the `sap.ui.controller(...)` function used as extension controller, and not with typed controllers.

By receiving the data object `oSomeData` from the server, the application enables you to access and modify the data object. The extension function name is `onDataReceived` and gets a reference to the data object as an argument.

Code of the standard controller:

```
// ...data object oSomeData has been received, possibly from an Ajax response...
if (this.onDataReceived) {          // check whether any extension has implemented the hook...
    this.onDataReceived(oSomeData); // ...and call it
}
// ...continue working with the (now possibly modified) data...
```

Code of the custom controller:

```
sap.ui.controller("customer.xy.Sub2ControllerExtension", {
  onDataReceived: function(oData){ // oSomeData will be passed in
    if (oData && oData.status === "important") {
      oData.message = oData.message + "!!!"; // modify some part of the data object, adding exclamation marks to a message text
    }
  } // no need to return anything as in this example the original object is modified
});
```

Figure 202: Example of Working with Controller Hooks

Controller Replacement

A standard controller can be replaced by specifying a new controller name in a replacement view, and implementing this controller.

A view that replaces a standard view can use the standard controller of the replaced view.

A view that replaces a standard view can use the standard controller, and extend the standard controller.

A view that replaces a standard view can use a totally new controller.

Select Extend Controller to add an controller extension.

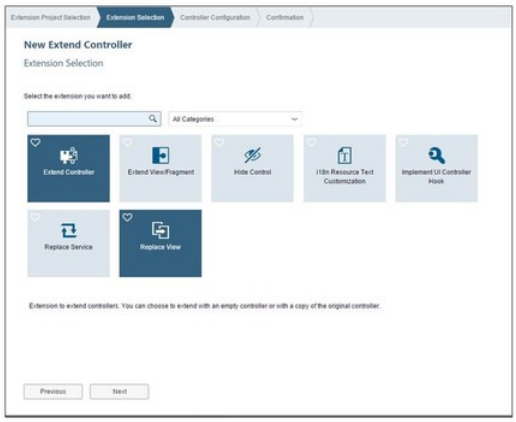


Figure 203: Controller Extension in the SAP Web IDE

When you want to create a copy or a new controller, select whether you want to replace the controller with an empty controller implementation, or with a copy of the original controller.

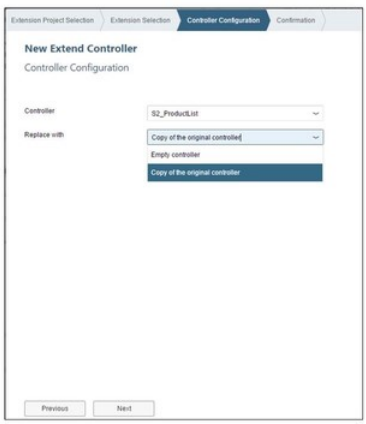


Figure 204: Create a Copy or a New Empty Controller

Before the new controller is generated you need to confirm the creation. Be aware of the possible loss of functionality when overriding controller methods.

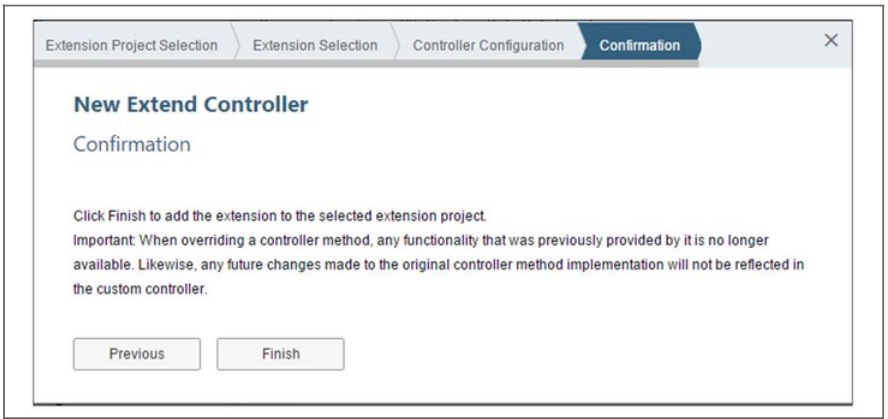


Figure 205: Confirm the Controller Enhancement or Replacement

Sample implementations are provided in the figures (component.js) and (copy of the existing controller).

Figure 206: Component.js

Figure 207: Copy of an Existing Controller

Controller Extensions Caveats

- If the extension code accesses parts of the original application which are changed, for example, removed, have a different type, or a different ID, controller extensions can cause a crash.
- If the extension code makes assumptions about the application which are no longer valid after an update, controller extensions can lead to a crash.
- If original code is overwritten which is required for the application to run properly, controller extensions can lead to a crash.
- If the controller name is changed, controller extensions are no longer applied

I18n Resource Text Customization .

- It is possible to enhance the `ResourceModel` with custom resource bundles
- The `ResourceModel` first tries to resolve the i18n text from the customized bundle.
- If text does not exist in the customized bundle, the `ResourceModel` tries to look up in the standard bundle.

```
var oModel = new sap.ui.model.resource.ResourceModel({bundleUrl:"./testdata/messages.properties"});
oModel.enhance({bundleUrl:"./testdata/messages_custom.properties"});
```

Figure 208: I18n Resource Text Customization

Use the i18n resources from the Fiori Lirary to extend the apps.

In the resulting file you are able to add new properties.

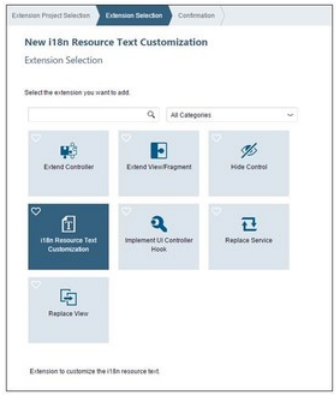


Figure 209: Customize I18n Texts

You can change the default translation of your SAP Fiori application.

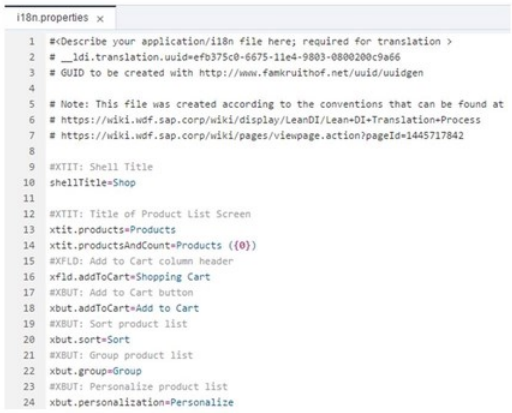


Figure 210: Change and Extend Code

LESSON SUMMARY
You should now be able to:

- Implement controller extensions and hook methods

Unit 13
Lesson 4

Extending Translations with Customer Properties

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Extend translations with customer properties

LESSON SUMMARY
You should now be able to:

- Extend translations with customer properties

Unit 13

Learning Assessment - Answers

1. What UI control can be used to define a hook to extend an application?

   Choose the correct answer.

   - [ ] **A** sap.m.Extend
   - [ ] **B** sap.ui.core.Extend
   - [X] **C** sap.ui.core.ExtensionPoint
   - [ ] **D** sap.comp.ExtendControl

   You are correct! The UI control sap.ui.core.ExtensionPoint specifies a point that provides a hook to extend an application.

2. What ways are there, of extending or modifying a SAPUI5 view?

   Choose the correct answers.

   - [X] **A** View extension
   - [ ] **B** View enhancement
   - [X] **C** View replacement
   - [ ] **D** View modification

   You are correct! You can extend or modify existing views in an application using view extensions, view replacements, and view modifications.

3. What concept is provided by SAP to make it possible to extend standard SAPUI5 function in the basic application?

   Choose the correct answer.

   - [ ] **A** Base methods
   - [X] **B** Hook methods
   - [ ] **C** View controllers
   - [ ] **D** Extension controller

   You are correct! You can extend and modify existing views in an application using view extensions, view replacements, and view modifications.

4. What happens if the text is not available in the customized bundle of your extension?

   Choose the correct answer.

- [ ] **A** The key will display in the application in capital letters.
- [X] **B** SAPUI5 tries to find the text in the resource bundle of the standard application.
- [ ] **C** An exception is raised.
- [ ] **D** No text is displayed.

You are correct! SAPUI5 will try to find the text in the resource bundle of the standard application.

UNIT 14 SAP Fiori Elements

UNIT OBJECTIVES

- Understand SAP Fiori elements
- Implement annotations and vocabularies
- Implement list report with SAP Fiori elements
- Implement search and filter capabilities with SAP Fiori elements
- Implement object page with SAP Fiori elements
- Display dependent entities as SAP Fiori elements

Unit 14
Lesson 1

Understanding SAP Fiori Elements

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand SAP Fiori elements

Introduction to SAP Fiori Elements

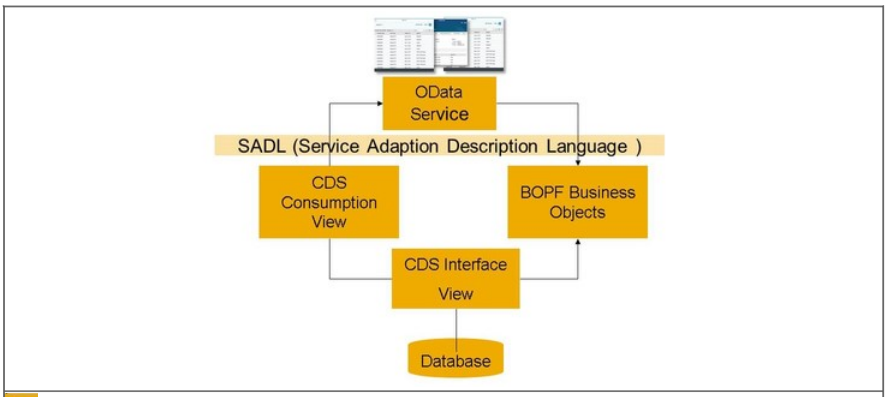


Figure 211: SAP S/4HANA Programming Model

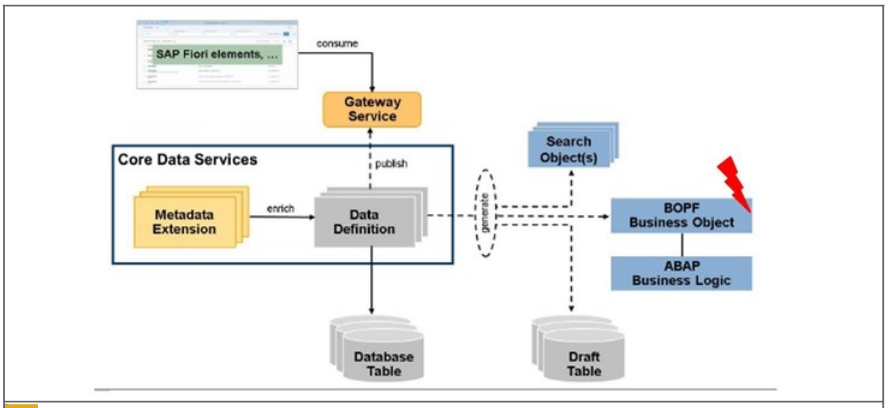


Figure 212: SAP S4/HANA Programming Model

The aim is to develop in an efficient manner.

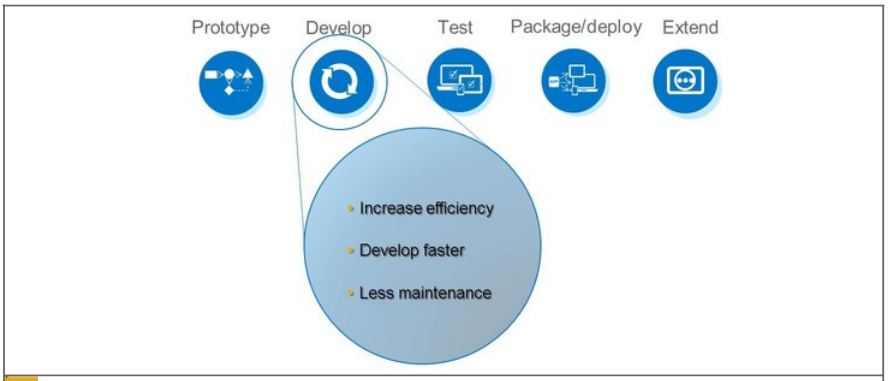


Figure 213: Every Developer's Goal

The diagrams provide an overview to the development environment.

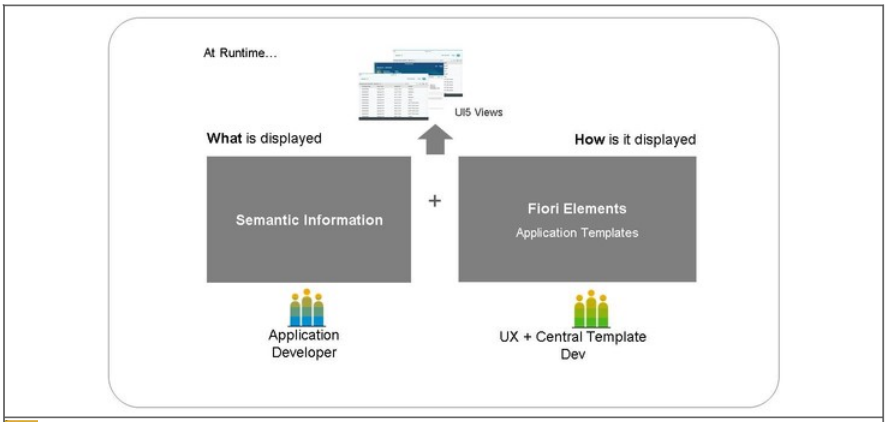


Figure 214: How They Work – A Bird's-eye (10,000 ft) View

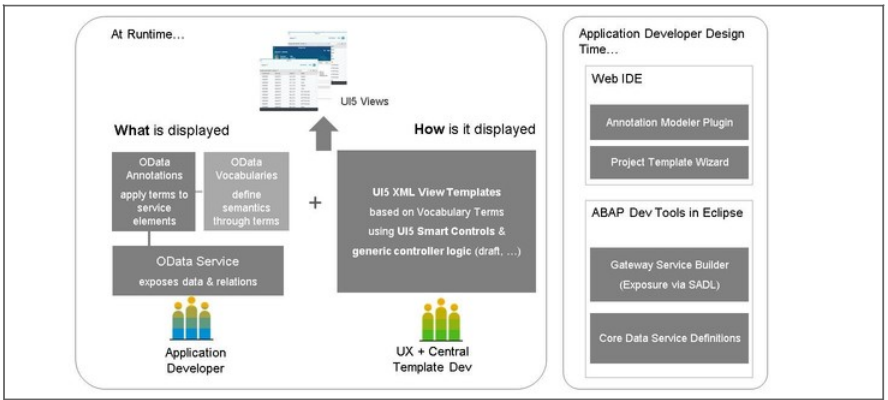


Figure 215: How They Work

Features of SAP Fiori Elements include:

- Reuse of functionality
- Common look and feel
- Edit mode control
- Message handling
- Support of SAPUI5 flexibility services
- Handling of draft documents and draft saving
- Status colors and icons to indicate criticality

SAP Fiori Elements – Update 2018

More than 50% of new Fiori apps at SAP are built using Fiori Elements, bringing a tremendous increase to the efficiency and maintainability of the development process. Understand the principles of metadata-driven UI development, learn about the available Fiori Elements list report, object page, overview page and the brand new analytical list page. Keep up to date with new features for analytics / charts, actions, navigation and many UX improvements. After this session, you should be ready to start developing and taking full advantage of our SAP Fiori Elements portfolio.

Figure 216: SAP Fiori Elements – Statement

List report elements are controlled by variant management.

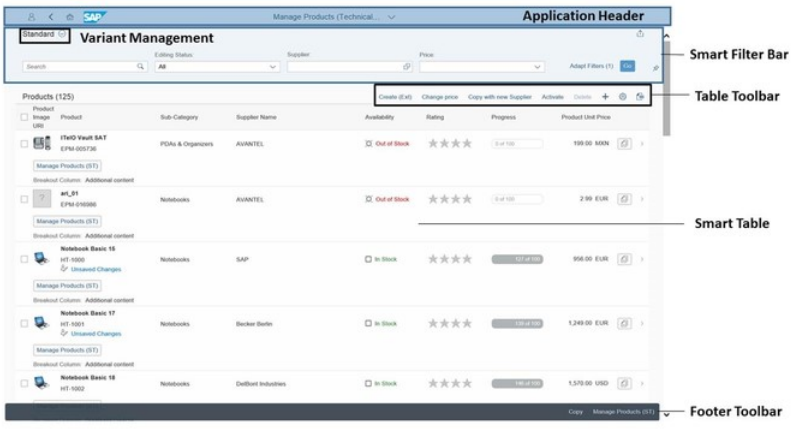


Figure 217: List Report Elements

Elements can be selected for various floorplan layouts.

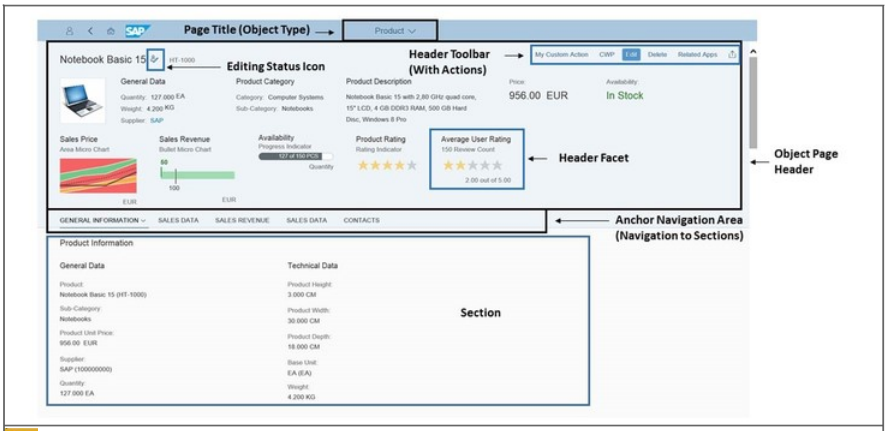


Figure 218: Object Page Elements

A different selection of elements may better suit the analytical report layout.

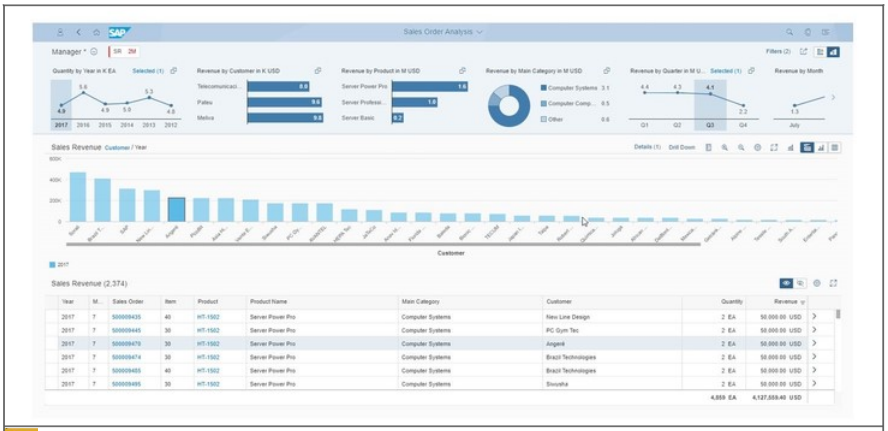


Figure 219: Analytical List Report

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 perform actions such as hiding cards, and showing or hiding the filter bar.
- 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. For more information about the Smart Filter Bar, see Smart Filter Bar.
- 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. Different card types display different data in different ways: 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.

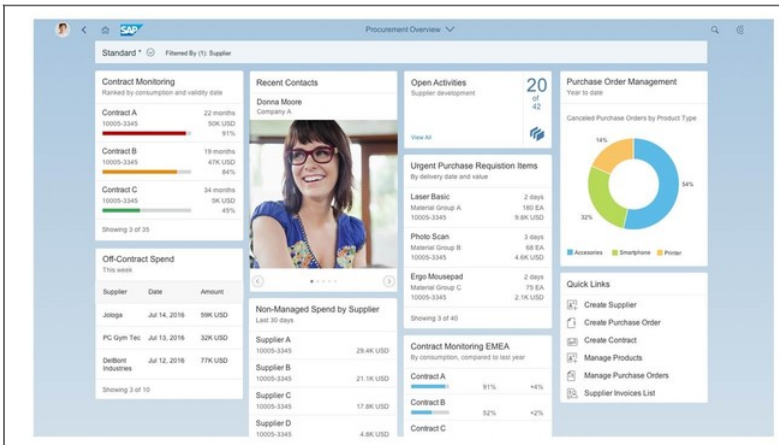


Figure 220: Overview Pages

LESSON SUMMARY
You should now be able to:

- Understand SAP Fiori elements

Unit 14
Lesson 2

Implementing SAP Fiori Elements

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Implement annotations and vocabularies

Annotations and Vocabularies

Annotations and vocabularies are mandatory for SAPUI development.

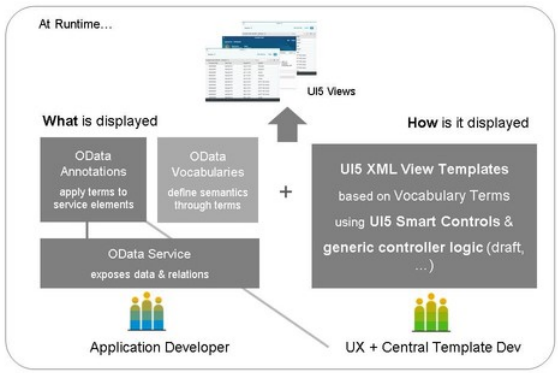


Figure 221: Mandatory Annotations

Annotations are specific to each domain.

Domain	Usage
UI Annotations	UI-relevant annotations used by SAP Fiori Elements
Consumption Annotations	Annotations for the consumption layer
Object Model Annotations	Annotations for structured and transactional aspects of the data model
Semantics Annotations	Sematic annotations for fields of the result set

Figure 222: Domain-specific Annotations

Further information on the use of annotations is available at: http://help.sap.com .

1. Search for _ABAP Programming Model for SAP Fiori. The result is based on the kernel version.
2. Select menu item Reference

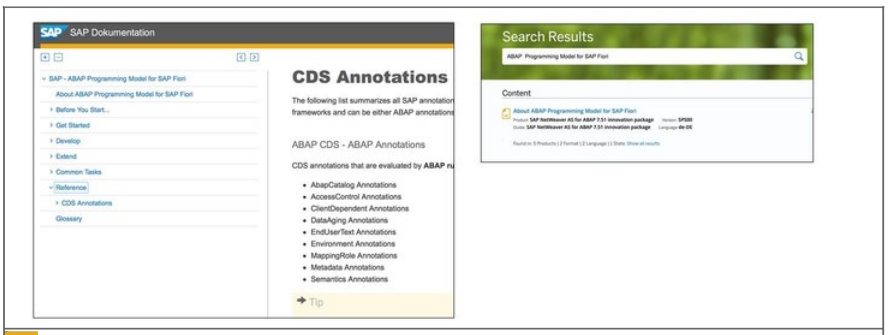


Figure 223: Finding Help on Annotations

Core Data System (CDS) view contains the logic for reading the data and semantic information from Data Dictionary (DDIC).

CDS annotation adds additional information on view or field level.

CDS core annotations contains information for all clients.

Domain specific information is only for specific clients.

Implementation depends on case use.

```
@AbapCatalog.sqlViewName: 'ZDemoSalesor'
@AbapCatalog.compiler.compareFilter: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'Consumtionview Salesorder'

@OData.publish: true

define view ZDemo_C_SalesOrder
  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 224: Annotations In Core Data System View

The annotation is embedded in the code of OData V2.

In Place Annotations in OData V2

```
<EntityType Name="Z_C_U2L1_Demo1Type" sap:label="Sales Order Report" sap:content-version="1">
    <Key>
        <PropertyRef Name="SalesOrderUUID"/>
    </Key>
    <Property Name="SalesOrderUUID" Type="Edm.Guid" Nullable="false" sap:label="Sales Order UUID"/>
    <Property Name="LifeCycleStatus" Type="Edm.String" MaxLength="1" sap:display-format="UpperCase" sap:label="Li
    <Property Name="OverallStatus" Type="Edm.String" MaxLength="1" sap:display-format="UpperCase" sap:label="Over
    <Property Name="SalesOrderID" Type="Edm.String" MaxLength="10" sap:display-format="UpperCase" sap:label="Sale
    <Property Name="CustomerName" Type="Edm.String" MaxLength="80" sap:label="Company"/>
    <Property Name="NetAmount" Type="Edm.Decimal" Precision="16" Scale="3" sap:unit="Currency" sap:label="Net Amo
    <Property Name="GrossAmount" Type="Edm.Decimal" Precision="16" Scale="3" sap:unit="Currency" sap:label="Gross
    <Property Name="TaxAmount" Type="Edm.Decimal" Precision="17" Scale="3"/>
    <Property Name="Currency" Type="Edm.String" MaxLength="5" sap:label="Currency Code" sap:value-list="standard"
    <Property Name="BillingStatus" Type="Edm.String" MaxLength="1" sap:display-format="UpperCase" sap:label="Conf
    <Property Name="DeliverStatus" Type="Edm.String" MaxLength="1" sap:display-format="UpperCase" sap:label="Orde
</EntityType>
```

Figure 225: Annotations for an OData Service – In Place

For example, place annotation in OData V4.

```
<Annotations Target="sap.com.cds_z_c_u2l1_demo1.Z_C_U2L1_Demo1Type">
    <Annotation Term="UI.HeaderInfo">
        <Record>
            <PropertyValue Property="TypeName" String=""/>
            <PropertyValue Property="TypeNamePlural" 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 226: Annotations for an OData Service – Out of Place

The code references vocabulary from the annotation file.

Referencing a vocabulary from annotation file

Sample of a vocabular

Figure 227: Vocabulary

The figure illustrates annotations at runtime.

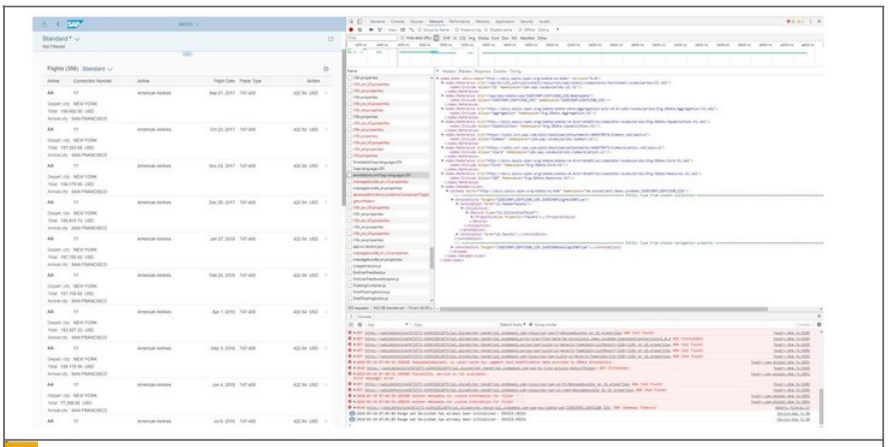


Figure 228: Annotations at Runtime

LESSON SUMMARY
You should now be able to:

- Implement annotations and vocabularies

Unit 14
Lesson 3

Implementing List Report using SAP Fiori Elements

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Implement list report with SAP Fiori elements

List Report with SAP Fiori Elements

Mandatory annotation for a list report floorplan with SAP Fiori elements is standardized.

```
@OData.publish: true

@UI.headerInfo.typeNamePlural: 'Flights'
define view ZMENWDE390_Demo1 as select from sflight as sf
    association to scarr as sc on sf.carrid = sc.carrid
```

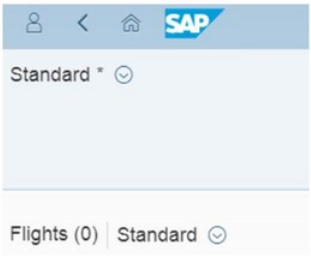


Figure 229: Mandatory Annotations

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

The data type for @UI.lineItem is a collection of `DataField`, but in most cases only one @UI.lineItem exists 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 field.

You can also add a 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.lable, or convert the data type to a data element by using function `cast`.

Figure 230: UI.lineItem Annotation

For the amount and quantity fields, indicate where to find its currency or unit fields.

Fields selected from DDIC, or other CDS entities may already have this information. If it is defined correctly, the currency or unit will be displayed with the amount or 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 the currency or unit field
- @semantics.amount.currencycode or @semantics.quantity.unitofmeasure to annotate amount or quantity field and say which field has the information for currencycode or unitofmeasure

You should always annotate semantic information for calculated fields you declare in your CDS view.

Figure 231: Add Semantic Information for Amount or Quantity Fields

Controls used in list report are designed for all clients.

Clients have various screen widths. The fields displayed in a smart phone should be less then a desktop browser.

Use @UI.lineitem.importance to determine in which platforms the fields should display.

- #HIGH: Default value, displays in all clients.

- #MEDIUM: Only display in a desktop browser or tablet.
- #LOW: Only display in a desktop browser.

```
@UI.hidden: true
sf.paymentsum,
```

Figure 232: Hide Fields

```
@UI.lineItem.position: 50
@UI.lineItem.importance: #HIGH
sf.planetype,
@UI.lineItem.position: 60
@UI.lineItem.importance: #MEDIUM
sf.price,
@Semantics.currencyCode: true
sf.currency,
@UI.lineItem.position:70
@UI.lineItem.importance: #LOW
@Semantics.amount.currencyCode: 'currency'
sf.price * sf.seatsocc as Total,
```

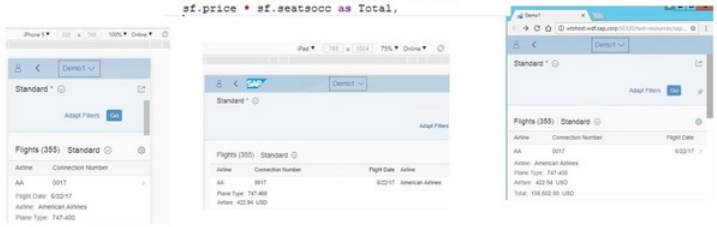


Figure 233: Importance of Columns

@UI.hidden will hide a field in the result for UI rendering.

A field annotation with @Consumption.hidden is not exposed to the client.

LESSON SUMMARY
You should now be able to:

- Implement list report with SAP Fiori elements

Unit 14
Lesson 4

Implementing Search and Filter Capabilities with SAP Fiori Elements

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Implement search and filter capabilities with SAP Fiori elements

Search and Filter Implementation

There is specific UI annotations for search and filter floorplan layouts to represent semantic views of business data.

The vocabulary used is completely independent of UI technologies. There is a distinction between a search field and a selection field.

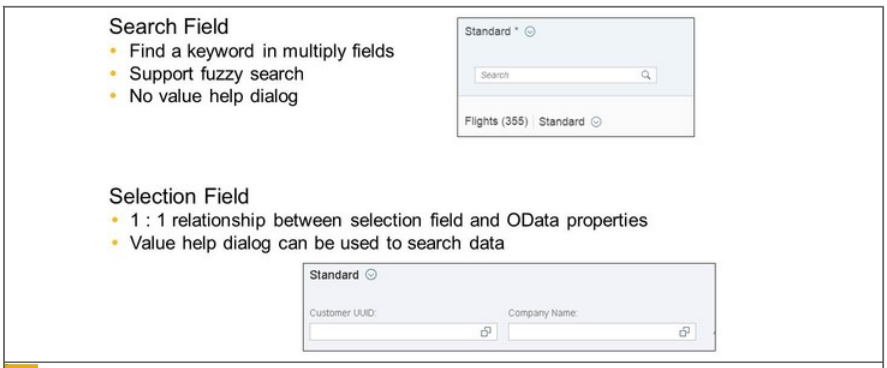


Figure 234: Search Fields and Selection Fields

For fields without fuzzy search function. the user must input the exact keyword of the data, or use wildcard characters (* and ?) to match the data.

For a field with fuzzy search function, even when the user has a typo in their key word, similar results will appear. In this example, the key word is "akitalia", but "Alitalia" appears in the result.

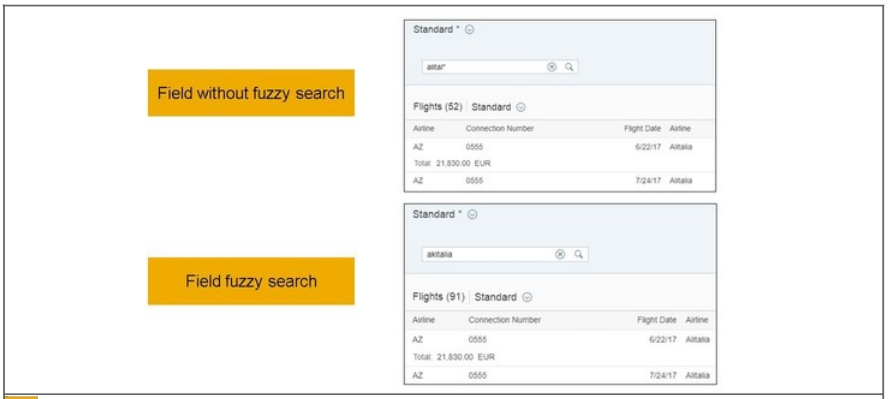


Figure 235: A Search Field From the End-User Perspective

To add a search field for a list report, the developer must first unable it on CDS view level by adding @Search.searchable:true annotation. At least one field should be set as a 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, with fewer records when values increase or decrease. Generally a suggested value in common use is 0.8.

Annotations for search fields only affects metadata of the 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 to SAPUI5, so that it will not appear in OData annotation.

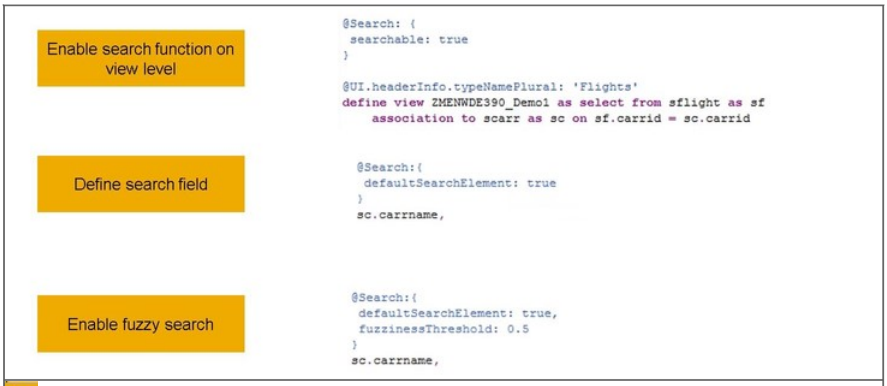


Figure 236: Search Field Annotations

Use @UI.selectionField.position annotation to declare a field as a selection field.

The order of selection fields is determined by value of this annotation.

Figure 237: Selection Fields

Add filters to allow the selection field to produce results.

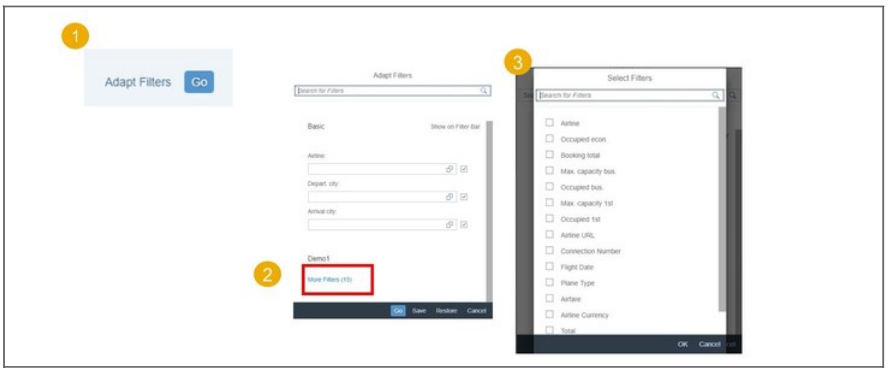


Figure 238: Adapt Filters for Selection Fields

Variants can also be added to aid the filter process.

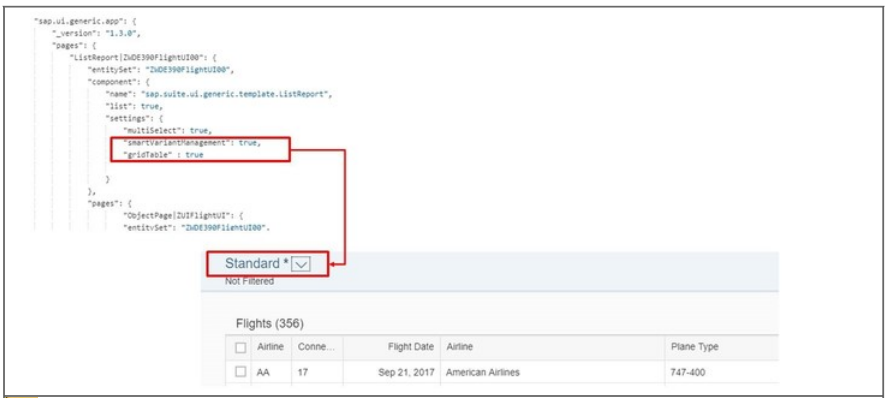


Figure 239: Selection Fields and Variants

These variants are embedded in the code.

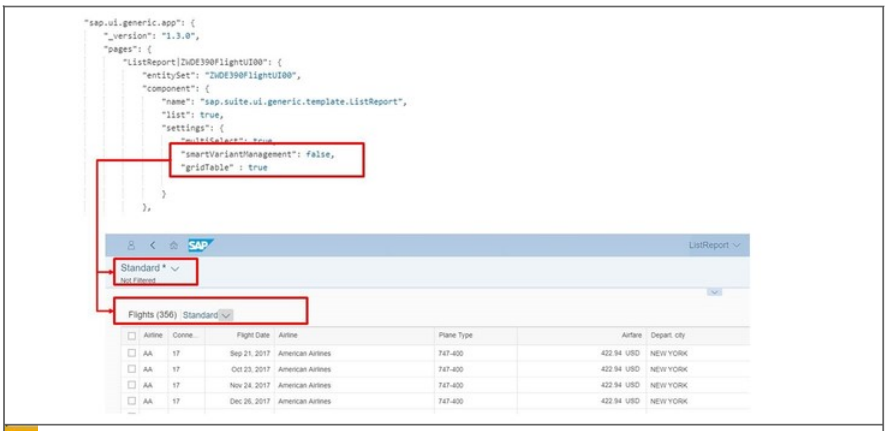


Figure 240: Table Variants

You can choose to hide the option of allowing variants within the code.

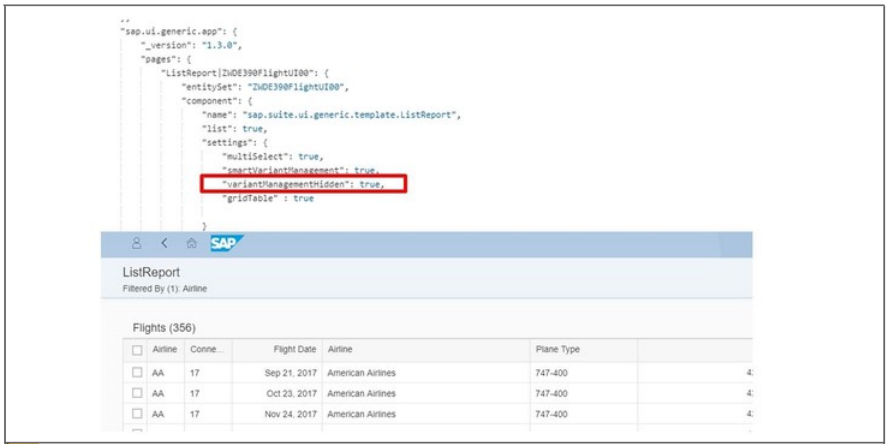


Figure 241: Hide Variant Management

The code to create a variant presentation is shown in the figure Implementing a Presentation Variant with the resulting display.

```
@UI.presentationVariant: [{
    sortOrder: [
       { by: 'carrname' },
       { by: 'carrid', direction: #DESC } ],
    groupBy: [ 'carrname', 'carrid' ],
    visualizations: [{ type: #AS_LINEITEM}],
    requestAtLeast: [ 'carrname']
    }]
```

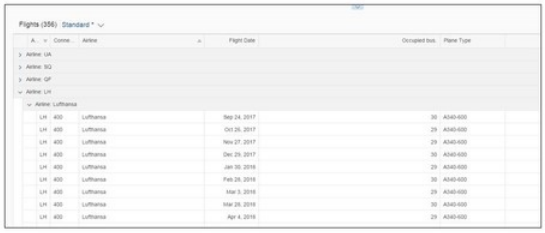


Figure 242: Implementing a Presentation Variant

It is possible to annotate a field as @DefaultAggregation, and assign the enumeration #SUM. This appears for each group of the presentation variant, and the overall sum appears in the projection list, at the end of the line of the annotated field.

```
@DefaultAggregation: #SUM
sf.seatsocc,
```

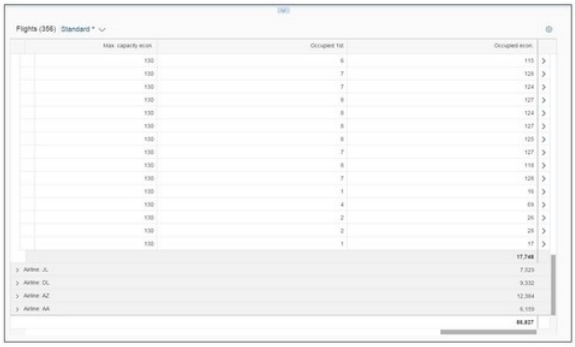


Figure 243: Building Sums

The property `criticality` will highlight status information in color.

```
case sc.carrname
 when  'American Airlines' then 2 // yellow
 when 'Alitalia' then 3 // green
 when 'Lufthansa' then 0 // grey
 when 'Emirates' then 1 // red
 else 0
end as Criticality,
```

```
@UI: {
lineItem:[{
    position:30,
    criticality:'Criticality'
    }
]}
@Search:{
 defaultSearchElement: true,
 fuzzinessThreshold: 0.5
}
@UI.selectionField.position:10
@UI.identification.position:10
sc.carrname,
```

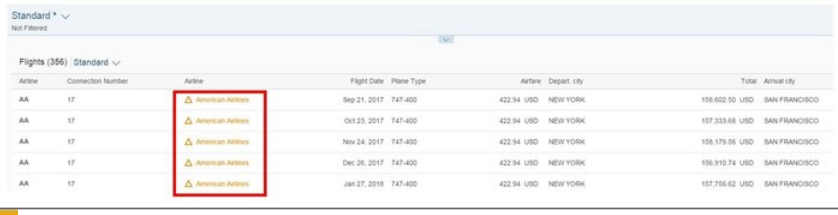


Figure 244: Show Status Information

A fuzzy search will display relevant flight details.

```
  @UI.lineItem: [ {
  position: 50,
  label: 'Show flight details',
  type: #FOR_INTENT_BASED_NAVIGATION,
  semanticObjectAction: 'toappnavsample2'                    -- Action
} ]
//@Consumption.semanticObject: 'Flightdetails'
```

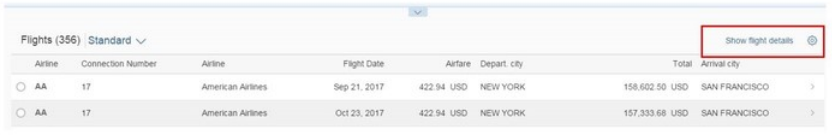


Figure 245: Result

LESSON SUMMARY
You should now be able to:

- Implement search and filter capabilities with SAP Fiori elements

Unit 14
Lesson 5

Implementing Object Page with SAP Fiori Elements

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Implement object page with SAP Fiori elements

Object Page with SAP Fiori Elements

There are specific annotations to use for the object page floorplan.

- @UI.headerinfo.typeName
  - Indeicate the type name of the business object
- @UI.headerinfo.title.value
  - Point 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 'General Information' section of the page

Figure 246: Annotations for Basic Object Page

Each facet relates to a specific section of the page.

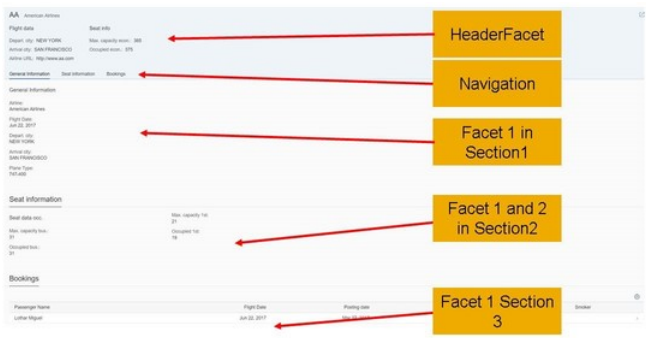


Figure 247: Arrangement of Sections and Facets

Header facets are pieces of information placed on the header part of an object page.

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

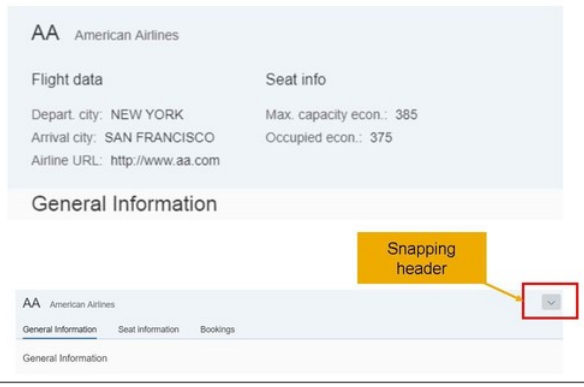


Figure 248: Header Facets

Field groups are defined within the code.

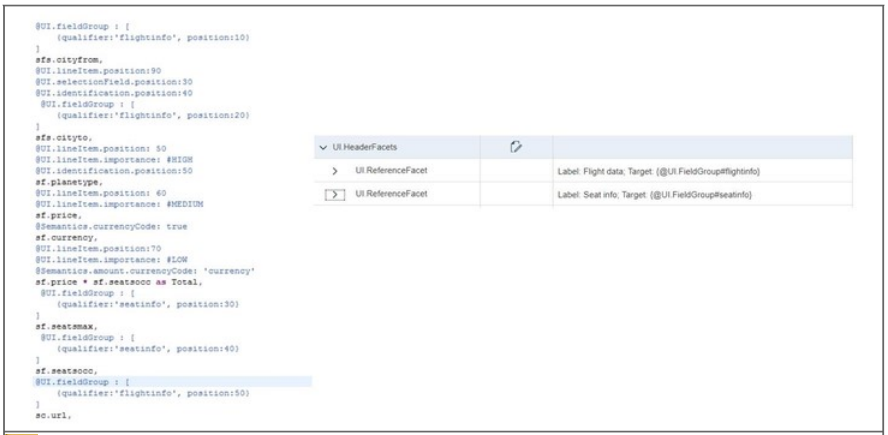


Figure 249: Define Field Groups

Use SAP UI reference vocabulary.

Figure 250: SAP UI Vocabulary

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

For Field Group Header Facet, add:

- 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.

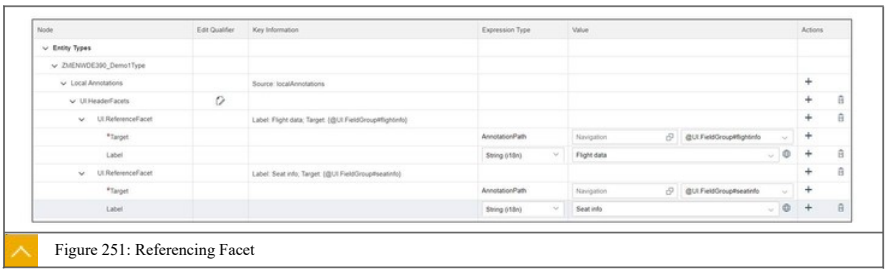


Figure 251: Referencing Facet

A common approach is to define an `UI.fieldGroup` in CDS view, as a field group is data intensive information. However, define `UI.HeaderFacet` annotation as local annotation, as facets are more appropriate with UI layer. Also `@UI.facet` annotation in CDS view is only supported by AS ABAP 7.52.

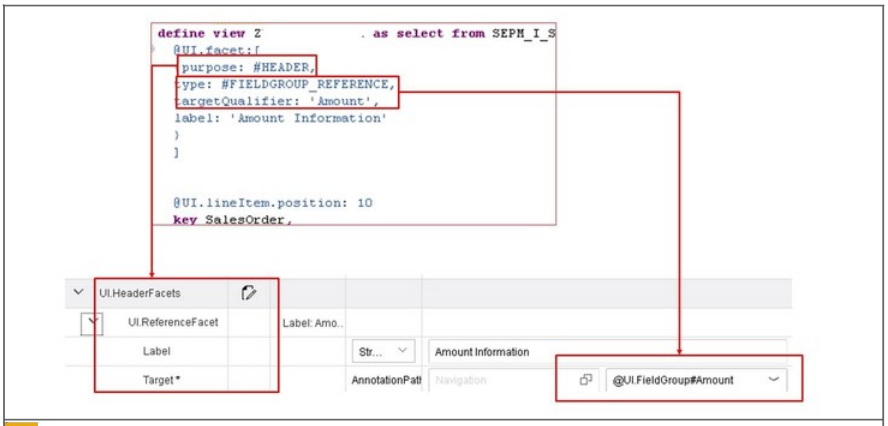


Figure 252: Define Field Group Header Facet

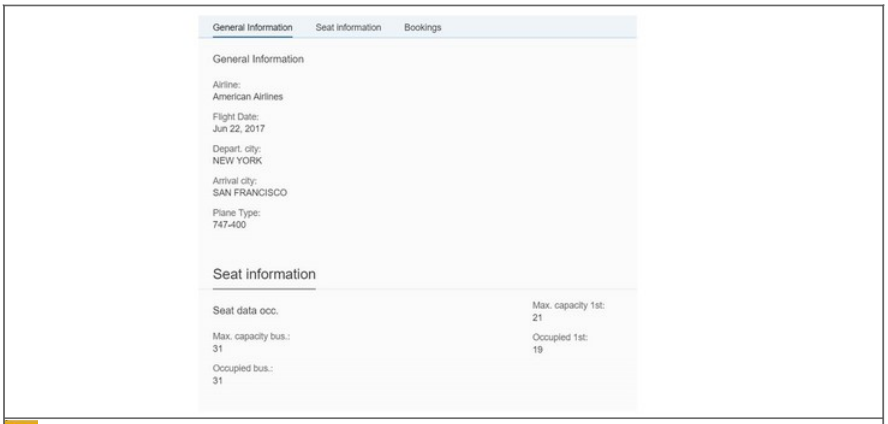


Figure 253: Display Sections With Facets (1 of 3)

Figure 254: Display Sections with Facets (2 of 3)

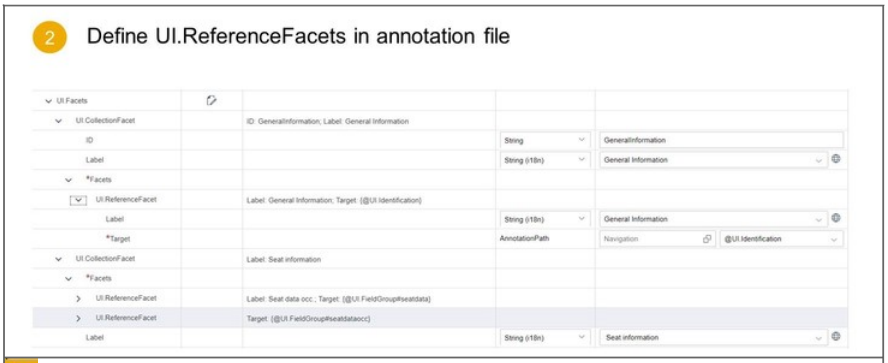


Figure 255: Display Sections with Facets (3 of 3)

LESSON SUMMARY

You should now be able to:

- Implement object page with SAP Fiori elements

Unit 14
Lesson 6

Displaying Dependent Entities as SAP Fiori Elements

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Display dependent entities as SAP Fiori elements

Display Dependent Entities as SAP Fiori Elements

When entities are displayed on the screen, sub-entities can be introduced using facets.

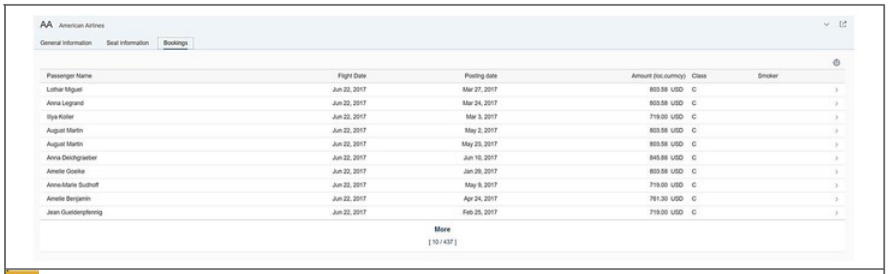


Figure 256: Display Sub-Entity Data

Steps for including these facets are shown in the following figures.

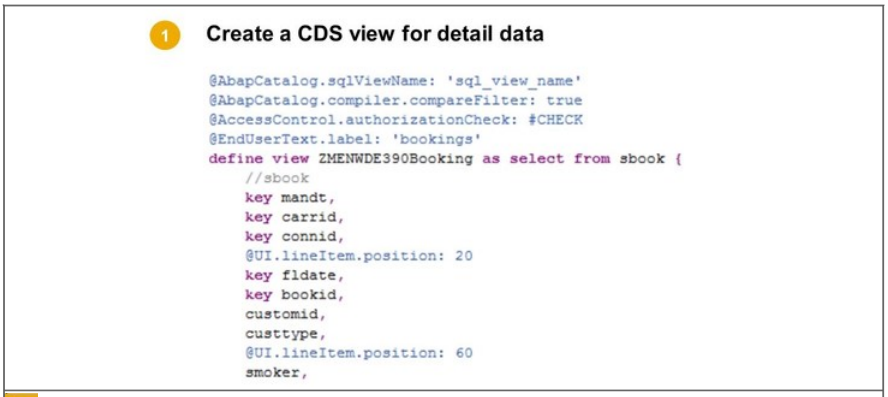


Figure 257: Steps for Facet Display of Sub-Entity Data (1 of 2)

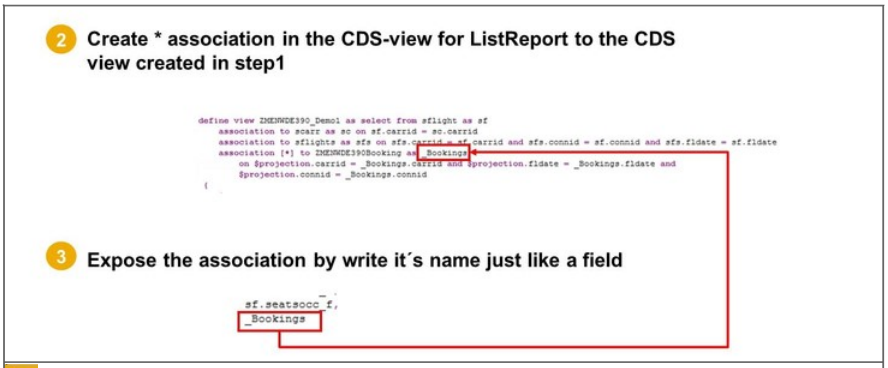


Figure 258: Steps for Facet Display of Sub-Entity Data (2 of 2)

For local annotation, create `UI.ReferenceFacet` as a direct child of `UI.Facets`.

For the reference facet, set the target by reference to the association and annotation.

When exposing CDS view as an OData service, SAP added "to" before the CDS association name, as OData association name. The annotation will be added automatically when creating a new project based on the list report template. Otherwise you have to add the annotation manually when updating an existing project.

From ABAP 7.52 the manual adaption of the annotation file is not necessary, there you can add the following to your CDS view:

```
@UI.facet {
type: #LINEITEM_REFERENCE,
targetElement: '_Bookings',
label: 'Bookings'
}
```

Local annotation can be introduced in the layout to orientate the user.

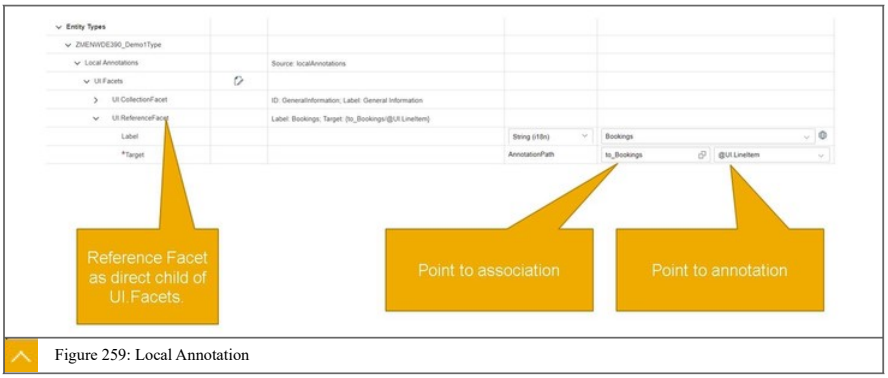


Figure 259: Local Annotation

This is updated in the manifest.json file.

Figure 260: Update manifest.json

Default UI.Facet Generated by the Web IDE

CDS views do not contain any `UI.Facet` annotation. A default `UI.Facet` local annotation is created automatically when a Fiori elements application is created by the wizard in the Web IDE.

The default facet contains a collection facet, and a reference facet, to refer to `UI.identification`.

IF `UI.Facet` annotation is added in CDS view after the SAPUI5 application was created, the developer should delete the default local annotation manually, otherwise it will override `UI.Facet` in CDS view.

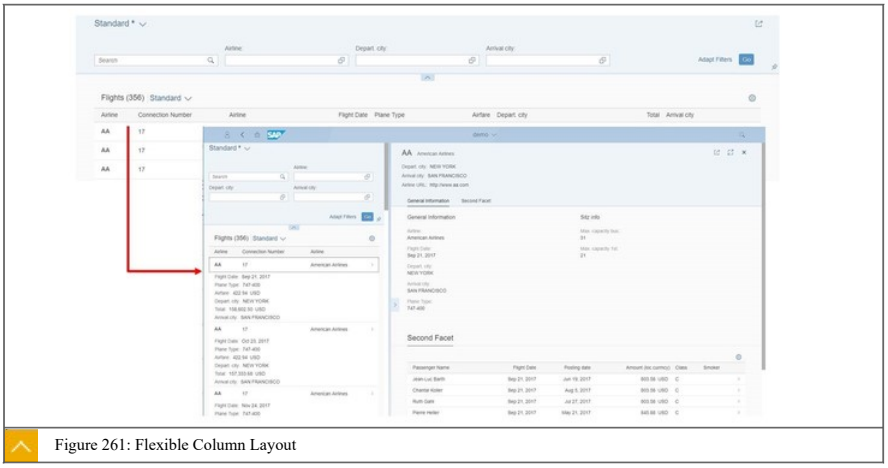


Figure 261: Flexible Column Layout

Switching to a Flexible Column Layout

Options for a 2 column layout – master-detail

- TwoColumnsBeginExpand – first column (master-detail) wider

- TwoColumnsMidExpand – second column – detail wider

Options for a 3 column layout –master-detail

- ThreeColumnsBeginExpanded – first column wider
- ThreeColumnsMidExpanded – second column wider
- ThreeColumnsEndExpanded – third column wider

```
"sap.ui.generic.app": {
    "_version": "1.3.0",
    "settings" : {
        "flexibleColumnLayout" : {
            "defaultTwoColumnLayoutType": "TwoColumnsMidExpanded",
            "defaultThreeColumnLayoutType": "ThreeColumnsEndExpanded"
        }
    },
```

Figure 262: Switching to a Flexible Column Layout

LESSON SUMMARY
You should now be able to:

- Display dependent entities as SAP Fiori elements

8. What annotation is used to show fields of the projection list in the general information section?

   Choose the correct answer.

   - ☐ **A** @UI.headerInfo.field
   - ☐ **B** @UI.lineItem.ident
   - ☐ **C** @UI.identification
   - ☐ **D** @UI.ident

9. What attribute is assigned to the @UI.fieldGroup to add an identifier?

   Choose the correct answer.

   - ☐ **A** identify
   - ☐ **B** quantify
   - ☐ **C** qualifier
   - ☐ **D** identifier

Unit 14

Learning Assessment - Answers

1. What is the goal of every developer?

   Choose the correct answers.

   - [X] **A** Increase efficiency
   - [X] **B** Develop faster
   - [ ] **C** Increase development effort
   - [X] **D** Less maintenance

   You are correct! It is the goal of every developer to increase efficiency, develop faster, and have less maintenance.

2. What possible values can be used for the annotation @UI.LineItem.importance?

   Choose the correct answers.

   - [X] **A** LOW
   - [ ] **B** VERY_HIGH
   - [ ] **C** HIGH
   - [X] **D** MEDIUM

   You are correct! To influence the responsive behavior of the SAP list report, you can use the enumeration values LOW, HIGH or MEDIUM.

3. What annotation is used to define the heading of list report?

   Choose the correct answer.

   - [X] **A** @UI.headerInfo.typeNamePlural
   - [ ] **B** @UI.LineItem.headerTitle
   - [ ] **C** @UI.headerInfo.typeName
   - [ ] **D** @UI.header.title

   You are correct! To define the heading of list report, you must use the annotation @UI.headerInfo.typeNamePlural.

4. What annotation is used to declare that a field of the projection list contains currency?

   Choose the correct answer.

   - [X] **A** @Semantics.currencyCode
   - [ ] **B** @UI.LineItem.currencyCode
   - [ ] **C** @UI.currencyCode
   - [ ] **D** @Semantics.currency

   You are correct! To declare a field of the projection list contains currency, use @Semantics.currencyCode.

5. What are the main characteristics of a search field in SAP Fiori element-based applications?

   Choose the correct answers.

   - [X] **A** No value help dialog
   - [X] **B** Supports fuzzy search
   - [ ] **C** Provides multiple search fields
   - [X] **D** Finds a keyword in multiply fields

   You are correct! The main characteristics of a search field in SAP Fiori element-based applications is that a search field does not provide value help but can find keywords in multiply fields by using one search field. If SAP HANA is used as an underlying database, fuzzy search is supported.

6. What annotations are required to declare that an ABAP Core Data System supports search capabilities?

   Choose the correct answer.

   - [ ] **A** @Search.possible
   - [X] **B** @Search.searchable
   - [ ] **C** @Search.allowed

   You are correct! To declare that an ABAP Core Data System provide search capabilities, the @Search.searchable annotation is used. The developer must assign the value true to the field.

7. What property is used in the manifest file, to configure how the selection and presentation variants display using a single variant management area?

   Choose the correct answer.

   - [ ] **A** variantManagement
   - [X] **B** smartVariantManagement
   - [ ] **C** allowSmartVariantManagement

   You are correct! The property smartVariantManagement is used in the manifest.json file to configure whether the presentation variant and the selection variant are combined.

8. What annotation is used to show fields of the projection list in the general information section?

   Choose the correct answer.

   - [ ] **A** @UI.headerInfo.field
   - [ ] **B** @UI.lineItem.ident
   - [X] **C** @UI.identification
   - [ ] **D** @UI.ident

   You are correct! Adding the annotation @UI.identification to a field of the projection list will show the field in the general information section of the object page.

9. What attribute is assigned to the @UI.fieldGroup to add an identifier?

   Choose the correct answer.

   - [ ] **A** identify
   - [ ] **B** quantify
   - [X] **C** qualifier
   - [ ] **D** identifier

   You are not correct. Add the property qualifier to the fieldgroup, and assign a unique identifier.

UNIT 15

Lean Development Infrastructure

Lesson 1

UNIT OBJECTIVES

- Understand the lean development infrastructure

Unit 15
Lesson 1

Understanding the Lean Development Infrastructure

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the lean development infrastructure

Lean Development

Lean development strategies are employed for development of SAPUI5 applications.

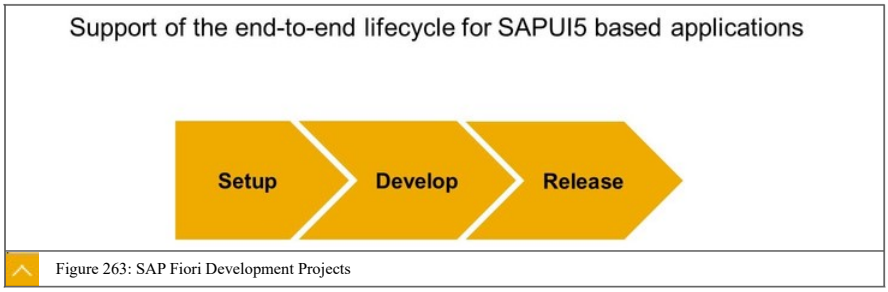


Figure 263: SAP Fiori Development Projects

Continuous Integration (CI) is a practice that involves developers checking-in their code several times a day to a central code repository.

Each time code is checked in, builds and test-runs are triggered automatically. This ensures that problems are detected early in the development process.

This practice requires a continuous integration server with a version-control system that tracks changes in the codebase.

At the beginning of the development process, each application developer on the project team has to set up their personal development environment.

Currently, SAP Web IDE, (formerly called WATT resp. River RDE), and the local Eclipse can be used.

It is recommended that SAP Web IDE be used as the development environment, even if you can also use Eclipse. In general it is possible to use both development environments in parallel. You can use local Eclipse in situations where it is not possible or difficult to work online. SAP Fiori templates are no longer supported in Eclipse. Therefore, when you want to create a new application, you must use SAP Web IDE for the initial application content creation.

A Gerrit configuration of the development environment is needed to check in and out source changes undertaken in the personal development environment to the central Git repository, (called "Fetch" and "Push and Commit"). Each developer, based on project membership, has the authority to use the central Git repository, and to do a code review with the Gerrit tool. Only a reviewed and approved code change will be merged into the Git repository. Additionally, a Fiori voter resp. Gerrit Voter Job running on the Build Server (Jenkins Job) of the development infrastructure, automatically performs a Maven test-build as a technical code check after each push and commit.

The upload to the translation system where localization-relevant code elements are translated to different languages are automatically triggered by the development infrastructure. They are then downloaded back into the development and test landscape. Built applications are stored and managed in the Nexus repository. From here the upload is done to distribute applications to the ABAP standalone UI test systems.

Development tests can be carried out in the local Tomcat with mock data. Back-end data can also be used through OData services. Therefore, each developer is able to test UI changes without the use of back-end services, or with back-end data, before publishing changes to other developers in the central Git. If also required, a central Tomcat server can be installed to centrally test the SAP Fiori UI before dispatch to the ABAP UI test systems. As it involves a manual upload, this option is rarely used.

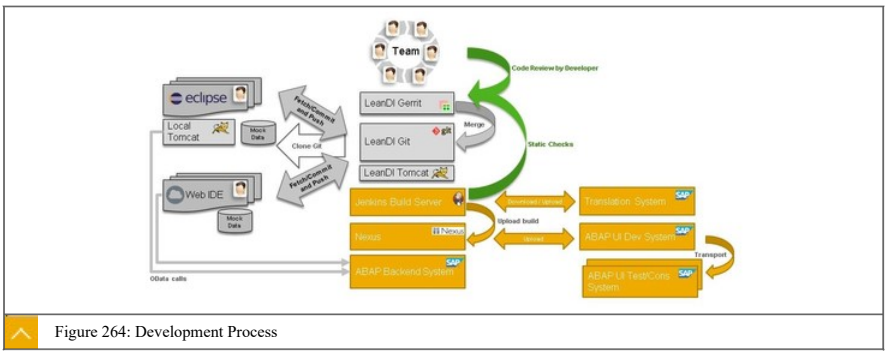


Figure 264: Development Process

Lean Development Infrastructure (Lean DI) is an SAP internal offering of services and tools that help to develop high-quality software. With Lean DI you gain the following benefits:

- Use of standard, established build tools like Git or Gerrit, Maven, Jenkins, and Sonar for building your products. All the features you want to deliver as a part of a product are built using Maven. Lean DI ensures that you can use Maven and these tools in a product standard compliant way.
- A development Infrastructure providing metadata about projects, or components or pieces of the software, in the project portal using an API. Having the meta-data allows you to automate processes. It also allows you to build better tools.
- Improvement in code quality, right where it is produced. This is performed by use of tooling for code review workflows. Git supports both easy branching, as well as easier merging and rebasing. This enables tools like Gerrit to automate code review workflows. The major advantage is that code reviews do not block developers from continuing with code development.

- Support for SAP specific processes such as translation, fortify scans, and code signing.
- Support for componentization. Once you have a feature in place, you can deliver it as a part of many products. Wagons (which are the different features of a component), can be hooked up together into multiple train compositions, (these are the different products). The goal is to build a wagon only once, and then re-use it in different trains. In terms of a product, this means you do not need to copy the source code, or perform double maintenance. To enable this type of componentization logic, you declare dependencies and pull new versions explicitly. This behavior is supported by Maven and OSGi.

Goals to implement a Lean DI strategy include:

- A lean-development infrastructure for Java, JavaScript, Android, Eclipse Tools, Web apps, for projects built with Maven, but not for C/C++, and iOS.
- Standardize build using established open-source tools
- A low barrier to entry
- With regard to processes, to either automate or provide self-service
- A committer and contributor collaboration model
- A project portal with a search engine
- To allow for faster innovation through independently released and serial versions of components
- Compliance with SAP product standards
- For lean development to be extensible: by bringing your own build plugins

Lean DI is a collection of components that software development teams use to develop quality software. The various components are useful at different stages.

- Stage 1:

  Project Portal - the portal collects, links and searches through projects meta-data.

- Stage 2:

  Git or Gerrit Server - support for automated code-review workflows

  JIRA - supports creation of JIRA components to track bugs or to maintain the backlog for your project.

- Stage 3:

  Eclipse - Integrated Development Environment (IDE) for Java development

  EGit - client tools, that allow you to work with Git source repository

- Stage 4:

  Maven is an open source build tool. Maven allows teams to develop their own build plug-ins, and hence gain more flexibility.

  Nexus - a remote Maven repository. Nexus allows you to share and reference serial versions of components.

Community Jenkins - a build box for projects that do not run their own build box. Jenkins provides continuous integration services for software development. It is a server-based system running in a servlet container such as Apache Tomcat.

- Stage 5:

  To make high-quality product, use Community Jenkins to monitor the executions of repeated jobs and additional tools like Fortify Security Code Scan or Checkmarx

  Sonar - a code metrics monitor for components

- Stage 6

  Once you are ready with the development, you can continue to release a version of your project. See (Releasing Maven Projects).

- Stage 7

  Conceptual information, services and tools for supporting globalization, signing code and artifacts, deploying into ABAP landscape for example.

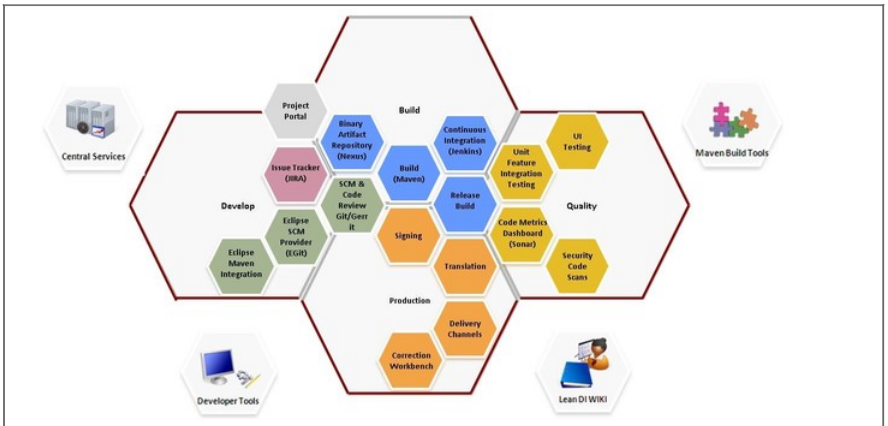


Figure 265: Lean DI Overview – Components

Git is a free and open-source, distributed version-controlled system designed to handle everything from small to very large projects with speed and efficiency. Every Git clone is a fully-fledged repository with complete history and full revision tracking capabilities. It is not dependent on network access or a central server. Branching and merging are fast and easy to carry out. The central Git or Gerrit repository contains and controls the project source code. It is a distributed version control system, keeping track of the developed software revisions of source code, with a corresponding hub in the Gerrit code review tool for the project.

Jenkins is an application which executes and monitors repeated jobs. It has extensive support for continuous build, test, and integration of software projects. Due to its flexibility, it is very well suited to all kinds of development processes.

Nexus - a remote Maven repository. Nexus allows you to share and reference serial versions of components. For more information, see Nexus and Nexus Landscape.

Release Build

The release server **ldirelease** is an offering of Lean DI that uses Jenkins, but not as a continuous integration server. The release server is only used to produce software releases in an SAP standard compliant way.

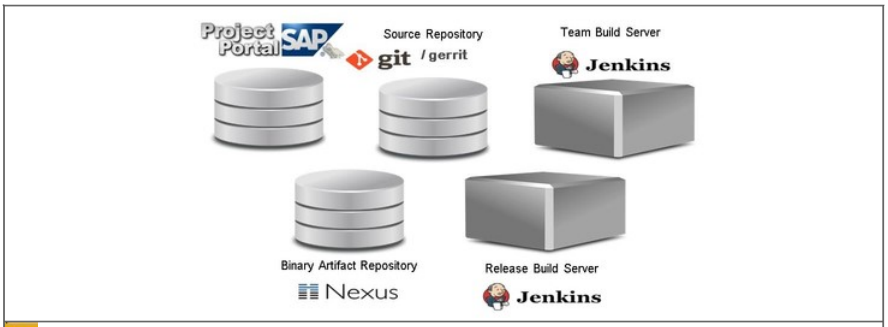


Figure 266: Lean DI Overview – Tools

The Lean DI overview diagram shows development workspace tools using the SAP Web IDE environment.

SAP Web IDE is available in the SAP HANA Cloud platform.

While source code is stored in a file system by way of using Orion, it is possible to use Git to manage app projects.

Orion is an open-source platform for cloud-based development. In SAP Web IDE, Orion is mainly used for Git integration.

SAPUI5 apps can be deployed in SAP HANA Cloud platform or SAP Gateway.

In the case of SAP HANA Cloud platform, the app is retrieved from SAP HANA Cloud platform while the data is retrieved from the on-premises SAP Business Suite systems. This is done through an on-premise SAP Gateway system that is connected to SAP HANA Cloud platform by the SAP HANA Cloud connector. Retrieving data from two different web pages is usually not supported by modern browsers that support the same origin policy. The problem is solved by using the HTML5 App Dispatcher that acts like a reverse proxy. From a consumer perspective, the OData service and the HTML5 source code now seem to stem from the same location.

Another option is to deploy the SAPUI5 app on the SAP Gateway server on premises. The SAP Gateway server is also called the front-end server in such a scenario. The same origin policy is fulfilled because the OData service and the SAPUI5 source code reside on the same system.

Integration with on-premises SAP Gateway systems is available through the SAP HANA Cloud connector.

HTML5 application development information can be found at https://help.hana.ondemand.com/help/frameset.htm.

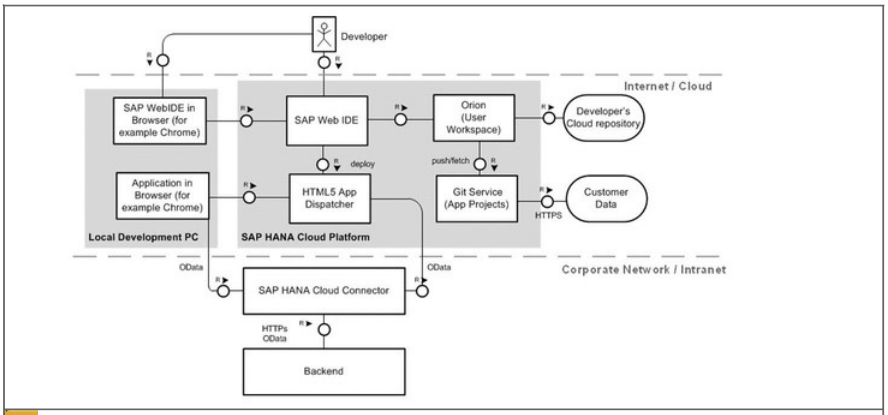


Figure 267: SAP Web IDE on SAP HANA Cloud Platform

LESSON SUMMARY
You should now be able to:

- Understand the lean development infrastructure

Unit 15

Learning Assessment

1. What is the purpose of a build server?

   Choose the correct answer.

   - ☐ **A** Used for code review
   - ☐ **B** Provides software that allows continuous integration.
   - ☐ **C** Used for version control
   - ☐ **D** Manages components, builds artifacts, and releases candidates in one central location

2. What goals are achieved by a Lean Development Infrastructure?

   Choose the correct answers.

   - ☐ **A** Allows for faster innovation through independently released and serial versions of components
   - ☐ **B** Either automates or provides self-service response
   - ☐ **C** Used to increase the development effort
   - ☐ **D** Used for ABAP development with Eclipse Oxygen

Unit 15

Learning Assessment - Answers

1. What is the purpose of a build server?

   Choose the correct answer.

☐	**A** Used for code review
X	**B** Provides software that allows continuous integration.
☐	**C** Used for version control
☐	**D** Manages components, builds artifacts, and releases candidates in one central location

   You are correct! A build server is software that allows continuous integration.

2. What goals are achieved by a Lean Development Infrastructure?

   Choose the correct answers.

X	**A** Allows for faster innovation through independently released and serial versions of components
X	**B** Either automates or provides self-service response
☐	**C** Used to increase the development effort
☐	**D** Used for ABAP development with Eclipse Oxygen

   You are correct! Using a Lean Development Infrastructure allows fast innovation through independently released and serial versions of components, and provides either automated or self-service response.

UNIT 16 Hybrid Application Toolkit

UNIT OBJECTIVES

- Mobilize SAP Fiori
- Understand the Kapsel SDK

Unit 16
Lesson 1

Mobilizing SAP Fiori

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Mobilize SAP Fiori

The Hybrid Application Toolkit

Mobilization of SAP Fiori .

SAP Fiori has been rolled out to desktops. The intent is to enable this functionality to be available on mobile devices. The roll-out of SAP Fiori will offer different consumption options to enable access on a mobile device. To accommodate the various means of consuming data from servers, the SAP Fiori interface can be accessed in a number of ways.

Three options will be available to consume SAP Fiori on mobile devices:

- SAP Fiori in browser
- SAP Fiori Client app or custom SAP Fiori Client
- SAP Fiori Mobile with SAP Kapsel SDK

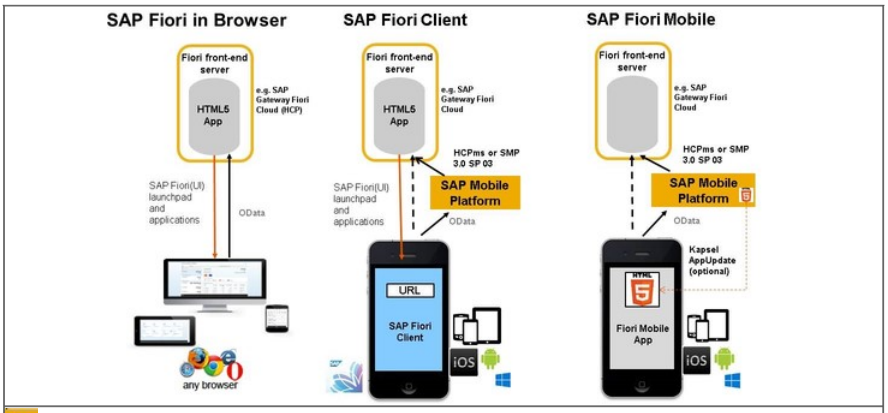


Figure 268: Three Options to Mobilizing SAP Fiori Apps

SAP Fiori apps run on a browser and work seamlessly on smart phones, tablets, and desktops.

Applications have been tested on the following operating systems:

- Apple OS

- Android
- Windows phone

Mobile OS (Phone + Tablet)

OS	OS Type	OS Version	SAP Fiori Client	Apple Safari	Microsoft Browser	Google Chrome
Apple [1] OS	Phone + Tablet	8 9	latest version	latest version	-	-
Android [2]	Phone + Tablet	4.4 - 5	latest version	-	-	latest version
Windows [3] Phone	Phone	10	-	-	Edge	-

Figure 269: SAP Fiori Web

Hybrid OS (Touch enabled desktop)

OS	OS Type	OS Version	Microsoft Browser	Google Chrome	Mozilla Firefox
Microsoft Windows	Touch/ Metro	8.1 10	IE 11 Edge [2]	latest version	-
	Desktop	8.1 10	IE 11 Edge [2]	latest version	latest RRC [3] latest ESR

[2] Windows 10 only
[3] RRC (Rapid Release Cycle); ESR (Extended Support Release)

Figure 270: SAP Fiori Web

A good start has been made, but the challenges remain:

- Start-up performance
- Attachment handling
- Native device capabilities integration
- Working without data connection (offline)
- Advanced security
- Life cycle management, support and administration

SAP Fiori Client, (SAP Fiori Web Plus), has better caching and performance than the other two options. Its features include:

- Native mobile app runtime for SAP Fiori
- Designed around Apache Cordova architecture

- Uses the same SAP Fiori web application
- Consistent performance produces a better user experience
- Implements enhanced cache management features in the SAP Fiori Client

- Support for Android, iOS and Windows devices
- Connection to SAP Fiori, using direct connection to the front-end server or through SAP Mobile Platform Server or SAP HANA Cloud Platform mobile services
- Application passcode for security
- Basic authentication, user certificate authentication and Security Assertion Markup Language (SAML) authentication.
- Settings screen
- Integrates device APIs such as camera, barcode scanner, and geo-location

Custom SAP Fiori Client for iOS and Android can be built using SAP Mobile Platform SDK.

Custom SAP Fiori Client incorporates Cordova and SAP Kapsel plugins.

Custom SAP Fiori Client allows customers and partners to:

- Build their own SAP Fiori Client
- Incorporate custom branding
- Add additional plugins to the application
- Tailor the applications behavior
- Wrap the application with Mobile Secure (Mocana)
- Deploy the application through enterprise app stores and Mobile Device Management tools

Through this, branding can be modified, custom authentication schemes or additional SAP, partner, or open-source plugins can be added.

Communication Security

To ensure integrity and confidentiality, all communication between the app on-device and target servers should be strongly encrypted.

SAP Fiori Client must use HTTPS protocol for all communications. The contacted server proves its identity by presenting a transport layer

Certification when connecting through HTTPS is done by the app acting as the HTTPS client, validating the identity of the server.

Using HTTPS mobile apps ensure that server certificate verification comprises checks for the following:

- Certificate expiration
- Signature validity
- Matching hostname and certificate subject name

- Certificate issued by a trusted certificate authority
- Basic constraints and key usage
- Certificate revocation

User Authentication and Single Sign-On

Different user authentication and single sign-on (SSO) mechanisms are supported.

One-Time Password (OTP) and SAP Single Sign-On (SSO), supports OTP based authentication using SAML IdP-initiated authentication and SAP Authenticator app.

SAML 2.0: SAML Assertions are a modern standard for web-based and cross-domain SSO. An Identity provider is required to issue SAML Assertions.

X.509 Client Certificates: In using SAP Fiori Client mobile app from the public app stores, client certificates must be provisioned with SAP Mobile Secure or SAP Afaria. In building custom SAP Fiori Client, third party mobile device management solutions can be used to provide certificates.

SAP Logon Tickets: Logon tickets are an SAP proprietary mechanism. It offers authentication and SSO in the form of a digitally signed cookie.

User Id and Password: Initial authentication can be based on user credentials. These are form-based and probably the least secure. Encryption of the communication path is essential.

Secure Storage of Data on Device

Local storage of data is a common vulnerability of mobile apps. Several measures are available to address this vulnerability:

- Application Passcode: This allows users to set an application passcode. When using SAP Mobile Platform Server or SAP HANA Cloud Platform mobile services, the passcode policy can be configured by the administrator.
- Logon Plugin Data Vault: Data vault is a reusable component for storing sensitive data securely on a device. Data vault is protected with the application passcode.
- Encrypted Storage: Encrypted storage plugin is available, allowing data to be stored securely on the device.
- Attachment handling: Attachment-viewer plugin enables secure and seamless handling any attachments opened.
- Privacy Screen: Since iOS and Android devices have app switchers that display screenshots of the apps, it could be a possible privacy risk for apps that display sensitive information. The privacy screen plugin hides application content in the app switcher.

Access to Native Device Capabilities

Some SAP Fiori apps take advantage of native device capabilities such as the camera, contacts, calendar, and geo-location. For security reasons it is vital to prevent unauthorized access to these capabilities.

No Bridge for Non-SAP Fiori Content: No Bridge is a security enhancement that prevents access to all non-essential plugins in SAP Fiori Client. It is used when the WebView URL has a different host than the SAP Fiori URL.

Feature Restriction Policy: A client policy that can be defined on the SAP Mobile Platform server or SAP HANA Cloud Platform mobile services, enabling the administrator to control the native device features available for use in SAP Fiori apps.

Cordova Whitelist: A plugin that allows configuration of a location from which a page can load its dynamic resources, locations it can communicate with, and the intents it can use.

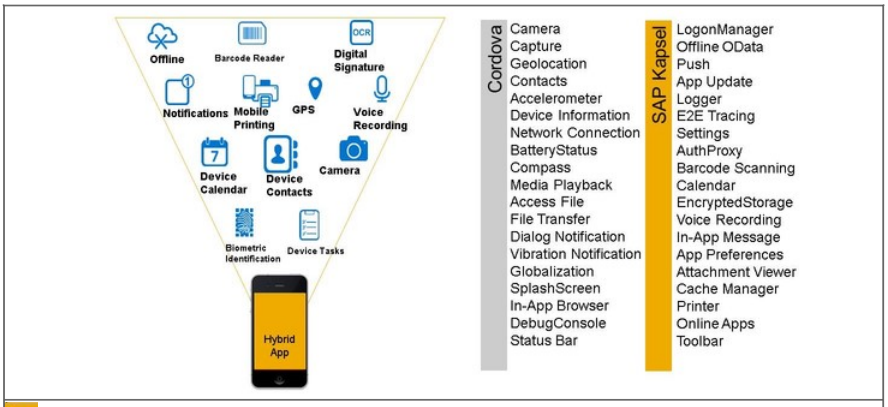


Figure 271: SAP Fiori Mobile with Cordova and SAP Kapsel Plugin and APIs

The SAP Mobile SDK for running SAP Fiori and other browser-based applications comes in a container named Kapsel SDK. The Kapsel SDK has a set of SAP specific plugins for Apache Cordova. It fits cleanly within existing Cordova development environments and processes. It is not a forked version of Cordova.

The Kapsel SDK includes plugins to make use of native mobile device functionalities and enterprise readiness:

- Secure on-boarding with encrypted storage
- Attachment handling and printing
- Asset updates and feature support

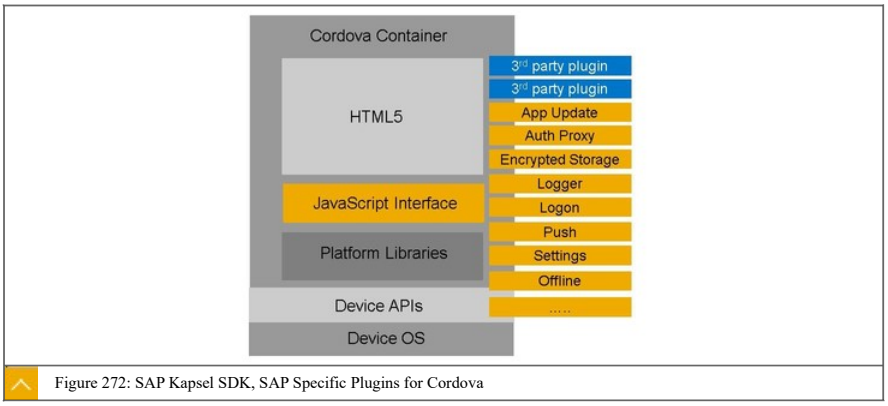


Figure 272: SAP Kapsel SDK, SAP Specific Plugins for Cordova

LESSON SUMMARY
You should now be able to:

- Mobilize SAP Fiori

Unit 16
Lesson 2

Understanding the Kapsel SDK

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the Kapsel SDK

The Kapsel SDK

An add-on toolkit for SAP Web IDE is available for Cloud deployments. It enables developers to create hybrid apps using Apache Cordova as well as the SAP Mobile Platform SDK for hybrid apps.

It comprises of three components:

- SAP Web IDE plugin
- Hybrid App Toolkit connector
- SAP Hybrid App Toolkit companion

SAP Web IDE Hybrid App Toolkit Plugin

SAP Web IDE Hybrid App Toolkit is available as a plugin on the SAP Web IDE

It adds hybrid app development features to the SAP Web IDE such as:

- Automatic code completion and documentation for Cordova and SAP Kapsel plugin
- APIs
- Hybrid project templates and code snippets
- Device configuration
- Previews on Hybrid App Toolkit Companion and browsers
- Deploys to local SAP HANA Cloud and SAP Mobile Platform environments
- Deploys and runs app on target device

Hybrid App Toolkit Connector

- The connector is included with the download add-on package for Hybrid App Toolkit.
- The toolkit can be downloaded from the SAP Store.
- The local server process enables SAP Web IDE to connect to the local systems Cordova development environment.
- It allows developers to create and manage a local Cordova project.

SAP Hybrid App Toolkit Companion

- The companion is included with the download add-on package for Hybrid App Toolkit.
- It can be downloaded from SAP Store and is available for download from Google Play store and Apple app store.
- It includes native mobile application-enabling live preview of a web application created with SAP Web IDE.

Benefits of using Hybrid App Toolkit Companion

- The companion contains all Cordova and Kapsel Plugins supported by the Hybrid App Toolkit.
- It allows preview of the web application apart from packaging in the application.
- It contains the Kapsel barcode scanner plugin, enabling scanning of the application barcode displayed in SAP Web IDE, to load the application.

In the scenario, the workstation connects to the Web through a proxy server, ensuring that all development tools are configured to correctly use the proxy server. Development tools such as Git, npm, or emulators can be used.

Network Configurations : SAP recommends the same network configuration be used to stall all tools.

When installing prerequisite software, validate the following environment and connectivity prerequisites:

- Optional Step: Use Hybrid App Toolkit, which utilizes functionality provided by SAP.
- Install the Software Mobile Platform Hybrid SDK (Kapsel).
- Download and install v5.4.1 of Node.js from https://nodejs.org/en/blog/release/v5.4.1/

Post installation, configure http and https proxy using the following commands:

- npm config set proxy <proxy server:port>
- npm config set https-proxy <proxy server:port>

Download and install the latest Java Development toolkit (JDK). It is required to generate certificates during installation. This allows a secure connection to be set up between the browser and the local tooling environment.

General Prerequisites:

- Node.js version 0.10.38
- Git client command line
- Cordova version 5.1.1
- SAP Kapsel SDK version 3.0 SP 10 (or later)

Prerequisites on Windows for Android development

- Android tools required
- JDK 1.7 or later

- Apache Ant 1.8 or later
- Android SDK 5.1.1 (API 22)
- Android AVD using API level 19

Prerequisites on Mac machine for iOS & Android development

- iOS tools required
- Xcode and command line tools
- iOS sim tool
- iOS deploy tool
- iOS signing profile
- Android tools required.

The Hybrid App Toolkit Companion app is a mobile application that runs on a mobile device or device emulator

Enables a live preview of a web application created with SAP Web IDE

Cordova based mobile app

Contains Cordova, SAP Mobile (Kapsel) and SAP UI5 libraries

Automatically built during HAT installation for iOS and Android

A QR code can be used to load corresponding application onto the mobile device.

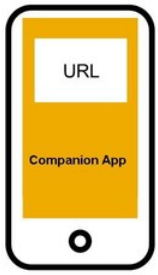
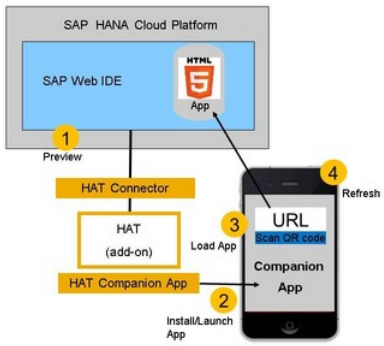


Figure 273: Hybrid App Toolkit Companion App



1. In SAP Web IDE: preview on simulator or device
2. Install and launch the companion app
3. Load Application from SAP HANA Cloud Platform by URL to the application
4. Possibility to refresh the application after modifications have been made to the application in SAP Web IDE

Figure 274: Hybrid App Toolkit Add-on – Preview

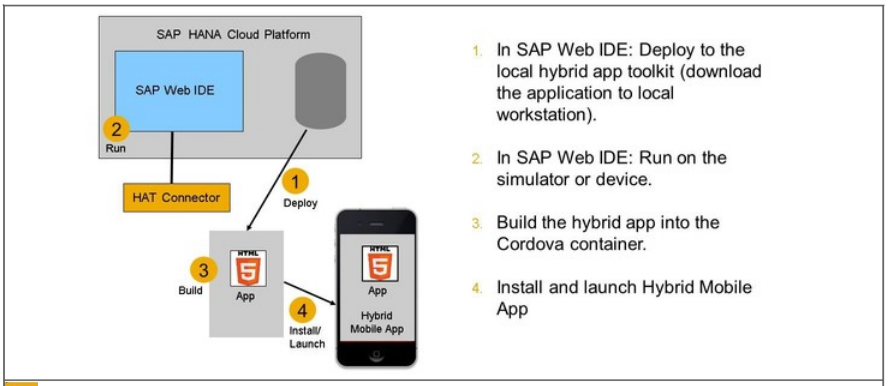


Figure 275: Hybrid App Toolkit Add-on – Running

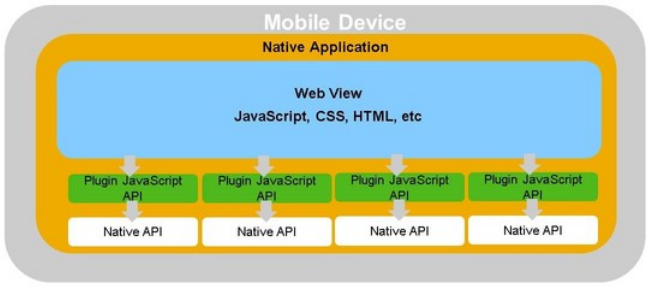


Figure 276: Cordova High Level Architecture

Cordova Standard Plugins

- Events; pause, resume, online, offline, backbutton, menubutton
- Notification; alert, confirm, prompt, beep, vibrate
- Capture; capture media files using devices media capture applications
- Storage; (DB device storage interface)
- Geolocation, file, media; (auto recording on playback)
- Contacts

Apache Cordova Benefits

- Apache Cordova has a large community of contributors and developers.
- It supports most devices.
- Web Developers can focus on the application and do not need to understand native code.
- Apache Cordova has better performance at app start-up and page loading.
- The has a `Config.xml` approach to configure web app and turn on or off plugins.
- It uses the network whitelisting approach to handle most web security threats.

- SAP specific plug-ins for Cordova
- For a Cordova based web application to be able to interact with SMP3/HCPms. SAP has provided a set of plugins that are collectively known as “Kapsel”

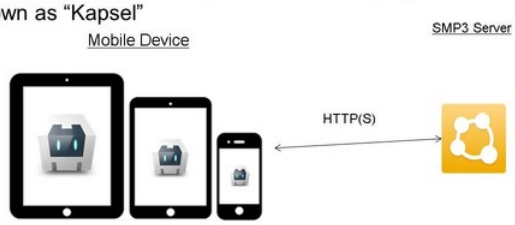


Figure 277: SAP Mobile Platform SDK – SAP Kapsel

1. AppUpdate
2. AuthProxy
3. EncryptedStorage
4. Logon
5. Logger
6. Push
7. OData
8. E2E
9. Barcode Scanner
10. Online App
11. App Preferences

Figure 278: SAP Kapsel Plugins

LESSON SUMMARY
You should now be able to:

- Understand the Kapsel SDK

Unit 16

Learning Assessment

1. What options are known to consume SAP Fiori apps on mobile clients without using the SDK for iOS or Android?

   Choose the correct answers.

   - [ ] **A** SAP Fiori in a browser
   - [ ] **B** SAP Fiori Client app or custom SAP Fiori Client
   - [ ] **C** SAP Fiori GUI Client
   - [ ] **D** SAP Fiori Mobile with Kapsel SDK

2. Which of the following user authentication and Single Sign-On mechanisms are supported by the SAP Fiori Client?

   Choose the correct answers.

   - [ ] **A** One-Time Password
   - [ ] **B** SAML
   - [ ] **C** Iris scan
   - [ ] **D** X.509 client certificates

3. What is Apache Cordova?

   Choose the correct answer.

   - [ ] **A** Apache Cordova is an open-source mobile development framework.
   - [ ] **B** Apache Cordova is the SAP product to run SAP Fiori applications on mobile devices.
   - [ ] **C** Apache Cordova is a cloud service to access data from mobile devices.
   - [ ] **D** Apache Cordova is the API provided by SAP to access mobile device features.

4. Which of the following components make up the SAP Web IDE Hybrid App Toolkit?

   Choose the correct answers.

   - ☐ **A** SAP Web IDE plugin
   - ☐ **B** Hybrid App Toolkit connector
   - ☐ **C** SAP Cloud Platform service
   - ☐ **D** SAP Hybrid App Toolkit companion

5. What features are available in the SAP Web IDE when using the SAP Web IDE plugin from the Hybrid App Toolkit?

   Choose the correct answers.

   - ☐ **A** Code completion for Cordova and Kapsel plugin APIs
   - ☐ **B** Device Configuration
   - ☐ **C** Deploy and run an app on a target device
   - ☐ **D** Access to iOS-only features

6. What is SAP Kapsel?

   Choose the correct answers.

   - ☐ **A** SAP specific plug-in for Cordova
   - ☐ **B** SAP specific plug-in for Apple iOS
   - ☐ **C** SAP specific plug-in for Android
   - ☐ **D** SAP Kapsel provides the API to interact from Cordova based web applications to SMP3 and software cloud platforms.

Unit 16

Learning Assessment - Answers

1. What options are known to consume SAP Fiori apps on mobile clients without using the SDK for iOS or Android?

   Choose the correct answers.

   - [X] **A** SAP Fiori in a browser
   - [X] **B** SAP Fiori Client app or custom SAP Fiori Client
   - [ ] **C** SAP Fiori GUI Client
   - [X] **D** SAP Fiori Mobile with Kapsel SDK

   You are correct! There are currently three options besides the use of native frameworks like the iOS SDK or the Android SDK. You can run the application inside a standard web-browser, using the SAP Fiori Client, a customer specific SAP Fiori Client, or using the SAP Kapsel

2. Which of the following user authentication and Single Sign-On mechanisms are supported by the SAP Fiori Client?

   Choose the correct answers.

   - [X] **A** One-Time Password
   - [X] **B** SAML
   - [ ] **C** Iris scan
   - [X] **D** X.509 client certificates

   You are correct! One-Time Password, SAML and X.509-certificates are supported user authentication and Single Sign-On mechanisms among others.

3. What is Apache Cordova?

   Choose the correct answer.

   - [X] **A** Apache Cordova is an open-source mobile development framework.
   - [ ] **B** Apache Cordova is the SAP product to run SAP Fiori applications on mobile devices.
   - [ ] **C** Apache Cordova is a cloud service to access data from mobile devices.
   - [ ] **D** Apache Cordova is the API provided by SAP to access mobile device features.

   You are correct! Apache Cordova is an open-source mobile development framework. It allows you to use standard web technologies, such as HTML5, CSS3, and JavaScript for cross-platform development.

4. Which of the following components make up the SAP Web IDE Hybrid App Toolkit?

   Choose the correct answers.

   - [X] **A** SAP Web IDE plugin
   - [X] **B** Hybrid App Toolkit connector
   - [ ] **C** SAP Cloud Platform service
   - [X] **D** SAP Hybrid App Toolkit companion

   You are correct! The SAP Web IDE Hybrid App Toolkit consists of three components. The SAP Web IDE plugin, the Hybrid App Toolkit connector, and the SAP Hybrid App Toolkit companion.

5. What features are available in the SAP Web IDE when using the SAP Web IDE plugin from the Hybrid App Toolkit?

   Choose the correct answers.

   - [X] **A** Code completion for Cordova and Kapsel plugin APIs
   - [X] **B** Device Configuration
   - [X] **C** Deploy and run an app on a target device
   - [ ] **D** Access to iOS-only features

   You are not correct! The SAP Web IDE plugin allows the deployment and running of an app on a target device and allows device configuration. It adds automatic code completion and documentation for Cordova and Kapsel plugin APIs, such as hybrid project templates, code snippets, and previews on Hybrid App Toolkit Companion and browsers. It deploys to local Cloud and SAP Mobile Platform environments.

6. What is SAP Kapsel?

Choose the correct answers.

X	**A** SAP specific plug-in for Cordova
	B SAP specific plug-in for Apple iOS
	C SAP specific plug-in for Android
X	**D** SAP Kapsel provides the API to interact from Cordova based web applications to SMP3 and software cloud platforms.

You are correct! SAP Kapsel is a set of plug-ins provided by SAP to interact from Cordova based web applications to SAP Mobile Platform 3 and software cloud platforms.