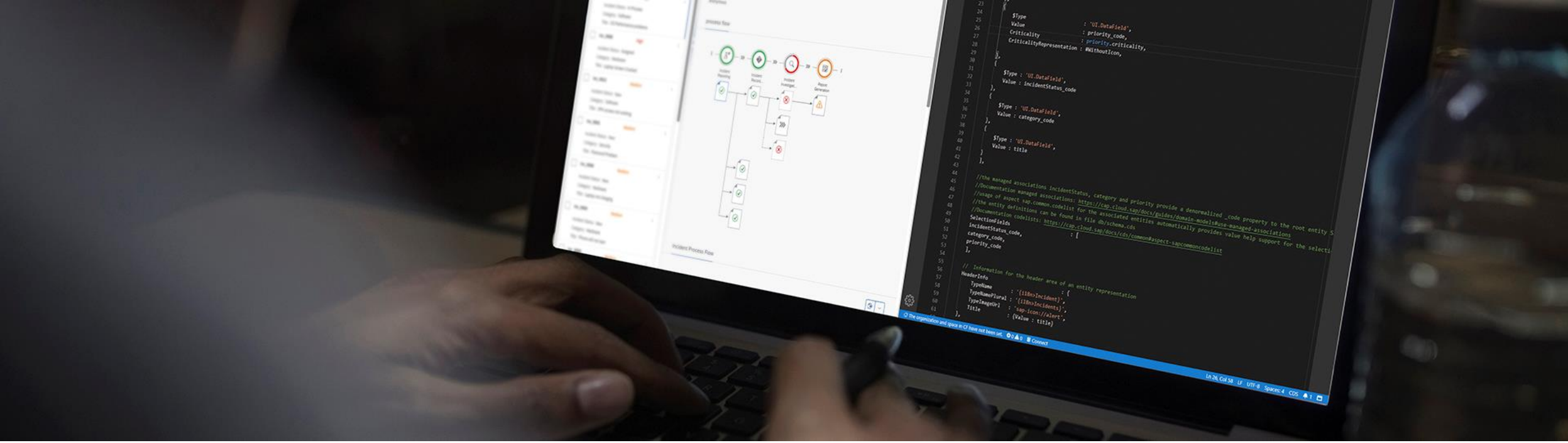


Week 1: Introducing SAP Fiori Elements

Unit 1: Painting the Big Picture

Introducing SAP Fiori Elements

- Painting the Big Picture — you are here
- Understanding the Architecture
- Introducing OData Services and Annotations
- Preparing Your ABAP Development Environment
- Creating an OData Service with ABAP RESTful Application Programming Model
- Creating a List Report

Building Your First SAP Fiori Elements App

- Creating an Object Page
- Accelerating SAP Fiori App Development with SAP Fiori Tools
- Preparing Your Front-end Development Environment
- Generating the List Report –– Object Page App
- Configuring the App
- Adding Custom Controls and Logic

Using More SAP Fiori Elements Capabilities

- Creating an Analytical List Page
- Enhancing the Analytical List Page
- Creating an Overview Page
- Deploying Your SAP Fiori Elements App
- Building an SAP Fiori Elements App with an External OData Service
- Using XML Annotation LSP for Defining Advanced UI Features

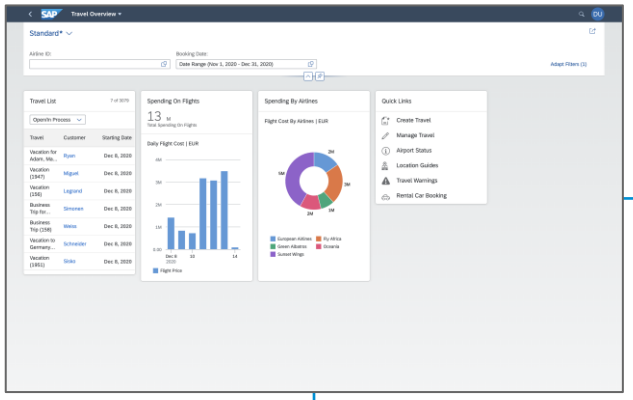
Extending a Standard SAP Fiori App

- Understanding SAPUI5 Flexibility
- Adapting an App as a Key User
- Adapting the UI as a Developer – Basic Extensions
- Adapting the UI as a Developer – Advanced Extensions
- Productizing the Extended App

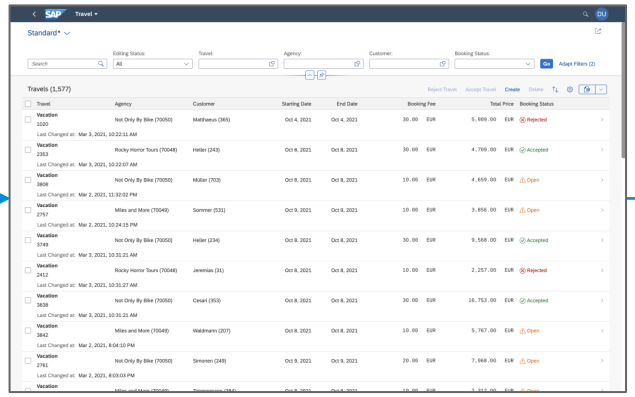
Painting the big picture

We will build apps for a travel desk clerk

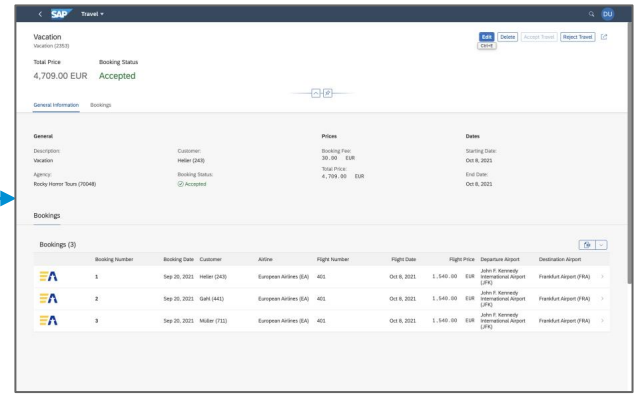
Travel overview



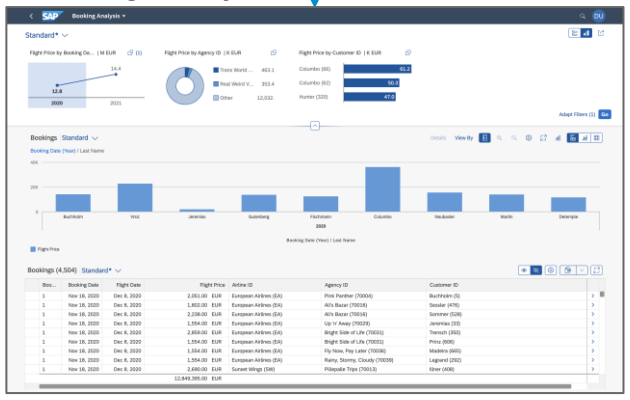
Travel list



Travel details

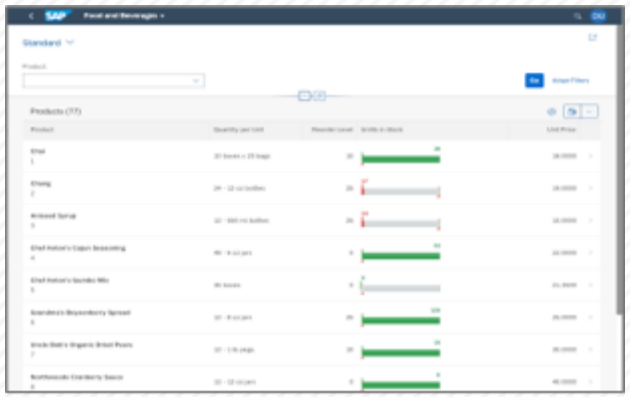


Booking analysis



Travel desk clerk Jay

Food and beverages



A consistent user experience is important to SAP's customers

"The current developments in user experience at SAP share one common vision: **a consistent user experience**… putting the user front and center."

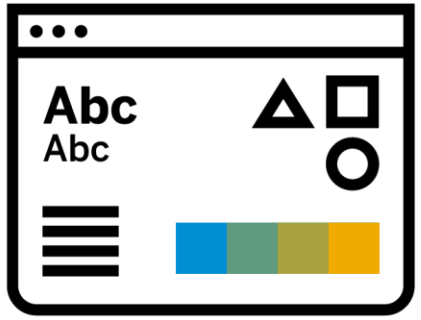
Thomas Saueressig
Member of the Executive Board of SAP SE
for SAP Product Engineering

Users care about UX consistency, so developers should too

There are measurable business benefits (monetary and human value) in good UX.
Consistency is at the heart of a great user experience – and a key design principle of SAP Fiori.

UX consistency

- Increases **adoption** of your app
- Improves business **user productivity**
- Decreases **training time**
- Improves **data quality**
- Enhances user **satisfaction**



SAPUI5 powers our web applications and their great UX

SAPUI5 is an HTML framework tailored to building modern, cross-platform, enterprise-grade web applications.

SAPUI5 is the reference UI technology for the SAP Fiori user experience.

SAPUI5

- Enables the SAP Fiori design evolution for a **consistent UX** across SAP solutions (and beyond)
- Provides built-in support for **SAP's product standards** such as security, accessibility, and internationalization
- Allows a single, **responsive** app implementation for all browsers, platforms, and devices
- Can **extend** SAP's standard applications and **customize** UIs in a **code-free way**

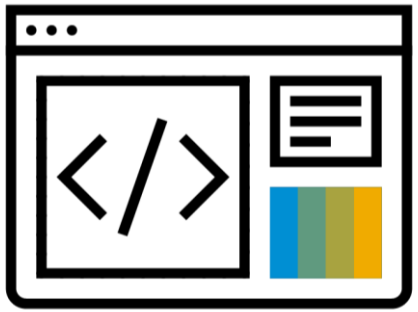
SAP Fiori elements lets you scale SAPUI5 app development

SAP Fiori elements provides standard floorplans and uses metadata to streamline and accelerate developing SAPUI5 applications.

SAP S/4HANA development teams use SAP Fiori elements to build most new apps.

SAP Fiori elements

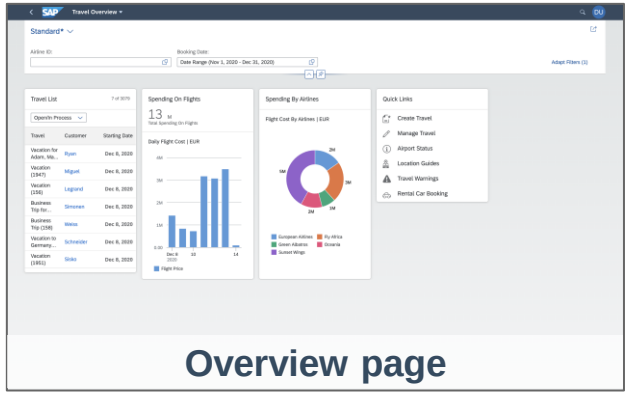
- Delivers **enterprise-ready** SAPUI5 apps based on stable, optimized, out-of-the-box UI code
- Boosts **developer efficiency** by scaling development and maintenance in a cost-efficient way
- Provides a centrally managed, **consistent UX** complying with the latest SAP Fiori design
- Ensures **upgrade stability** and forward compatibility
- Allows **extensibility** of standard floorplans with custom controls and logic



Standard floorplans drive developer productivity and UX consistency

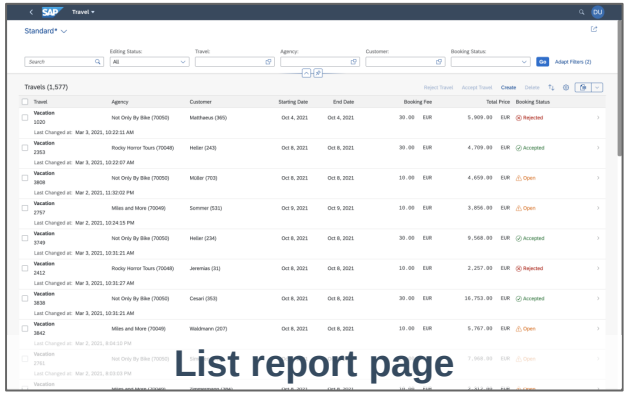
Most enterprise use cases involve providing an **overview** of business-relevant data, working with **lists** of business objects, and managing these business **objects**.

Provide a role-based **overview**



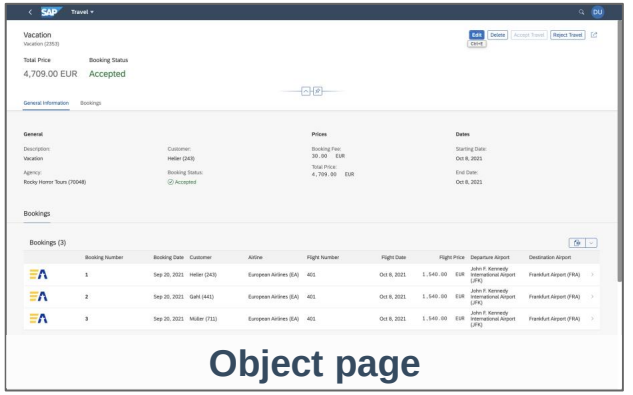
Overview page

Work on a **list** of business objects

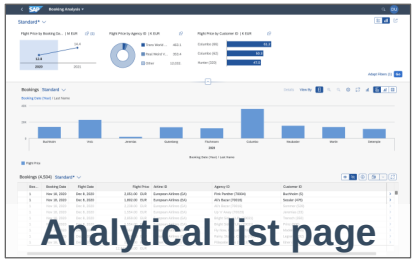


List report page

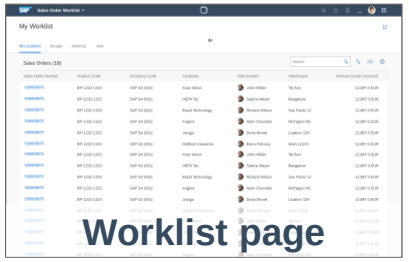
Manage individual business **objects**



Object page



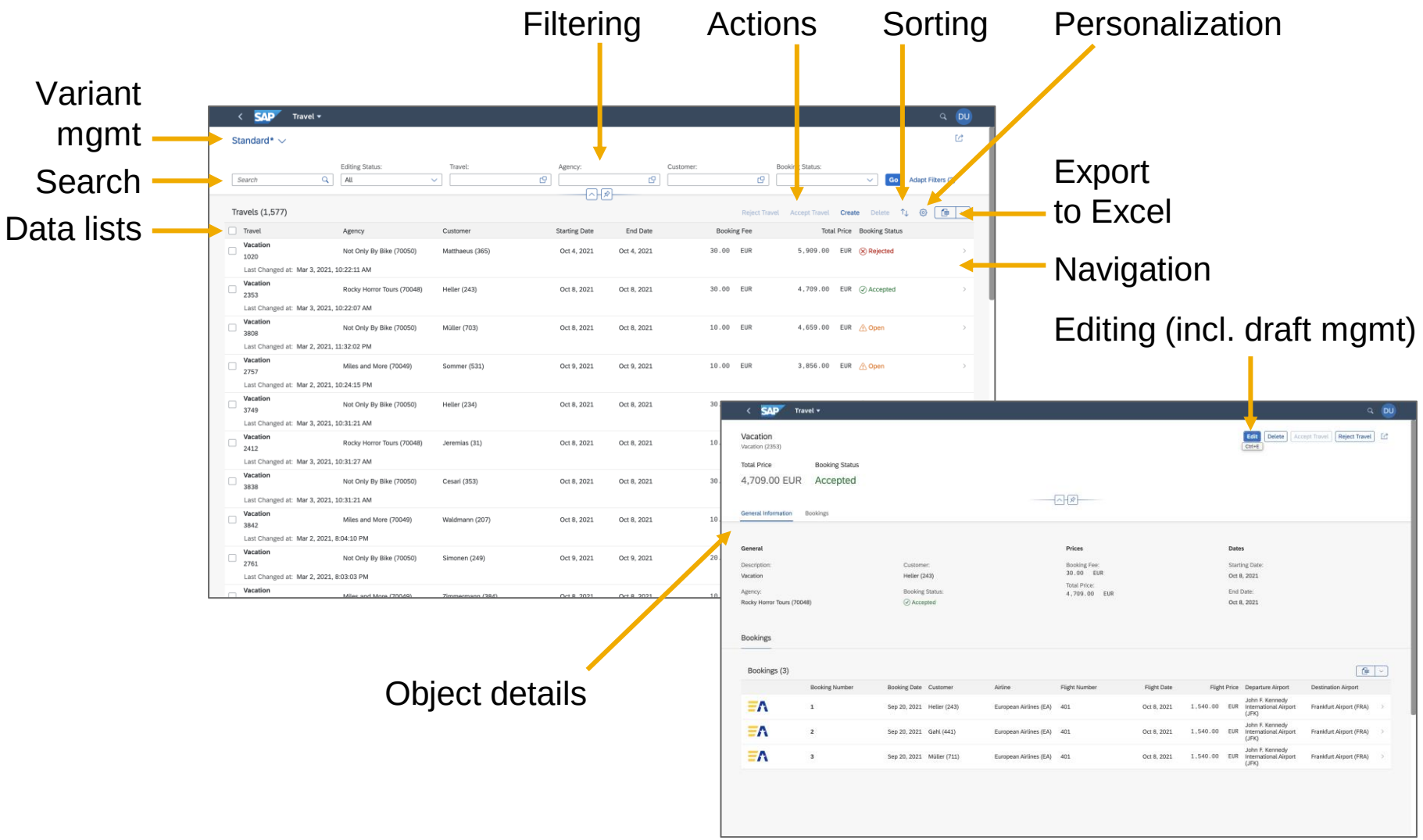
Analytical list page



Worklist page

Painting the big picture

SAP Fiori elements provides enterprise-ready apps out of the box



SAP Fiori elements / SAPUI5 enterprise readiness features:

- Accessibility
- Internationalization
- Mobile compatibility
- Responsiveness
- Performance optimizations
- Integration
- Security checks
- Lifecycle stability
- Test automation support
- …and more

Painting the big picture

SAP Fiori elements prioritizes efficiency over flexibility

SAPUI5 freestyle		SAP Fiori elements
Flexibility	**Overall approach**	Efficiency
Freestyle designs	Design requirements	SAP floorplans
Web development	Development knowledge	Annotations
Own coding, own UI logic	App ownership / maintenance	Own annotations, SAP’s UI logic
Higher	**Total cost of development and maintenance**	Lower

Wrap-up

IN THIS UNIT, YOU LEARNED

- Why UX consistency matters
- How SAPUI5 and SAP Fiori elements drive developer productivity, enforce UX consistency, and deliver enterprise-ready apps
- When to use SAP Fiori elements vs. SAPUI5 freestyle

NEXT UNIT

- Week 1 – Unit 2
  Understanding the architecture

Thank you.

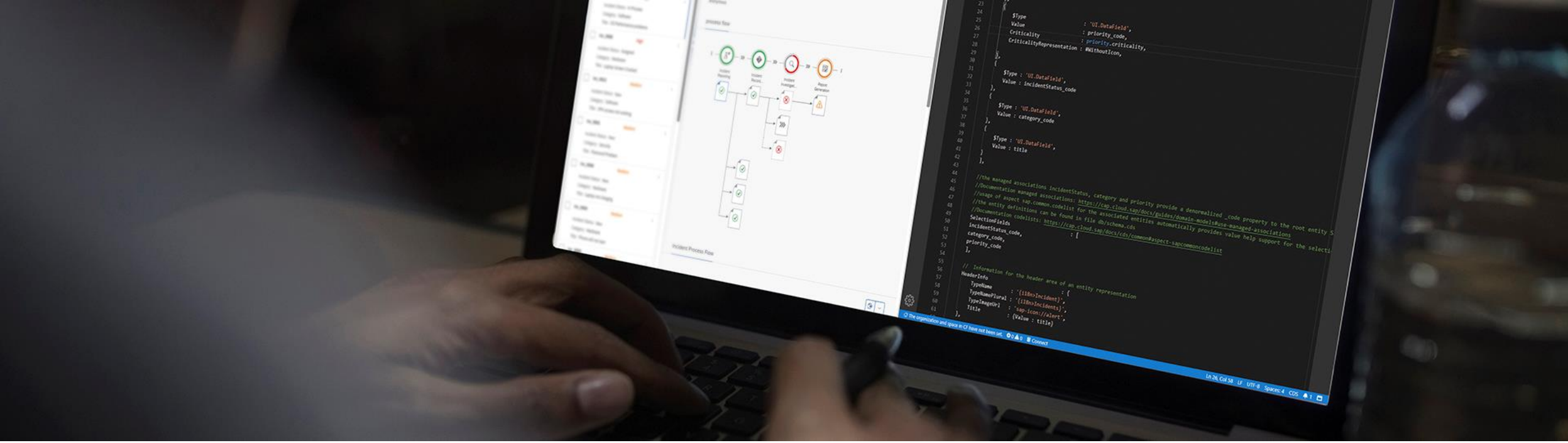
Contact information:

open@sap.com

Follow all of SAP

www.sap.com/contactsap

THE BEST RUN SAP®



Week 1: Introducing SAP Fiori Elements

Unit 2: Understanding the Architecture

Introducing SAP Fiori Elements

- Painting the Big Picture
- Understanding the Architecture — you are here
- Introducing OData Services and Annotations
- Preparing Your ABAP Development Environment
- Creating an OData Service with ABAP RESTful Application Programming Model
- Creating a List Report

Building Your First SAP Fiori Elements App

- Creating an Object Page
- Accelerating SAP Fiori App Development with SAP Fiori Tools
- Preparing Your Front-end Development Environment
- Generating the List Report – Object Page App
- Configuring the App
- Adding Custom Controls and Logic

Using More SAP Fiori Elements Capabilities

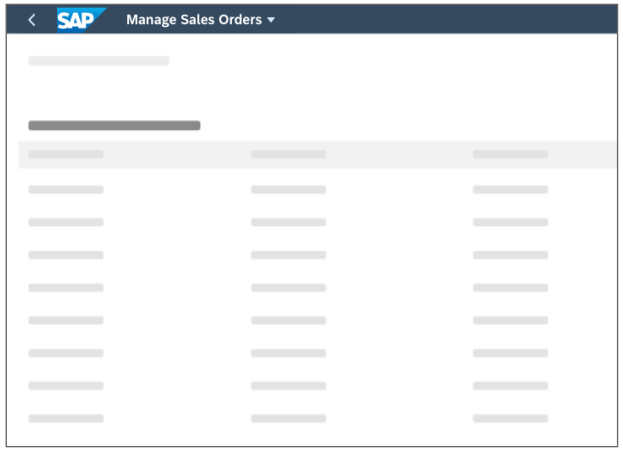
- Creating an Analytical List Page
- Enhancing the Analytical List Page
- Creating an Overview Page
- Deploying Your SAP Fiori Elements App
- Building an SAP Fiori Elements App with an External OData Service
- Using XML Annotation LSP for Defining Advanced UI Features

Extending a Standard SAP Fiori App

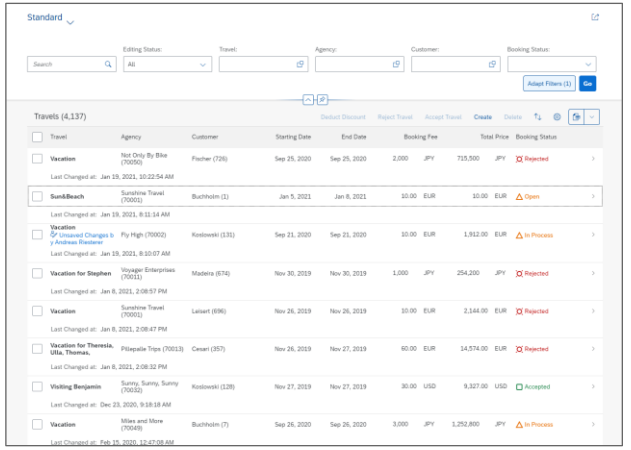
- Understanding SAPUI5 Flexibility
- Adapting an App as a Key User
- Adapting the UI as a Developer – Basic Extensions
- Adapting the UI as a Developer – Advanced Extensions
- Productizing the Extended App

Understanding the architecture

SAP Fiori elements takes you from floorplans to an enterprise-ready application

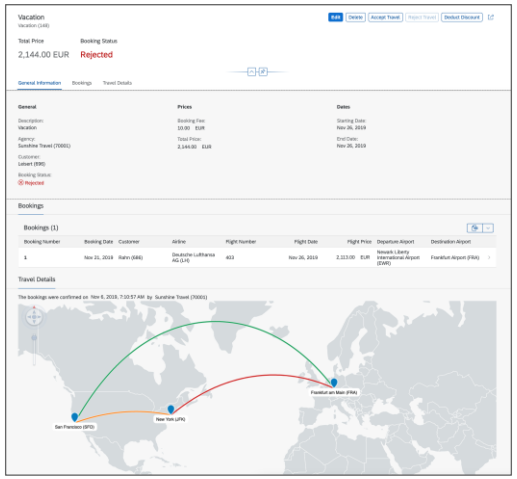


Floorplans



OData service

Annotations



Configuration (optional)

Custom code (optional)

Provided by:
SAP Fiori elements framework

Provided by:
Application developers

The floorplans and application controller provide the basis for your application

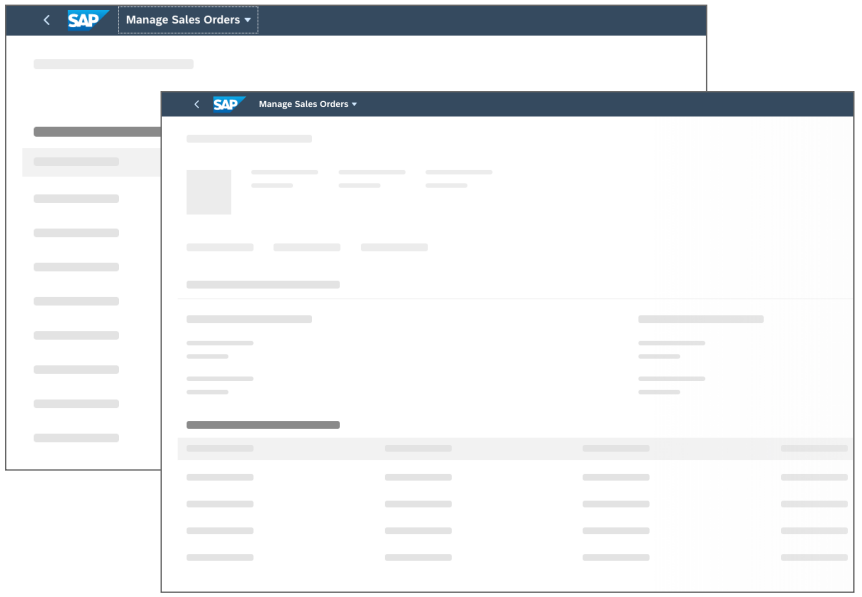
Floorplans

OData service

Annotations

Configuration

Custom code

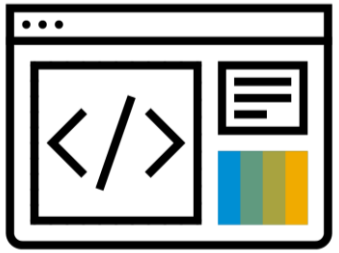


SAP Fiori elements floorplans

- Implemented as XML templates
- Incorporate current UX design

SAP Fiori elements application controller

- Connects the pages and controls
- Provides key features (e.g. draft)



Understanding the architecture

OData connects your app to the back end

Floorplans

OData service

Annotations

Configuration

Custom code

+ SAP draft support (optional)

OData V2 || V4

Meta description of data model
CRUD and actions
Query language
Capabilites and restrictions

SAP S/4HANA

SAP BW

SAP BTP

Other OData provider

OData annotations define how the app looks and behaves

Floorplans

OData service

Annotations

Configuration

Custom code

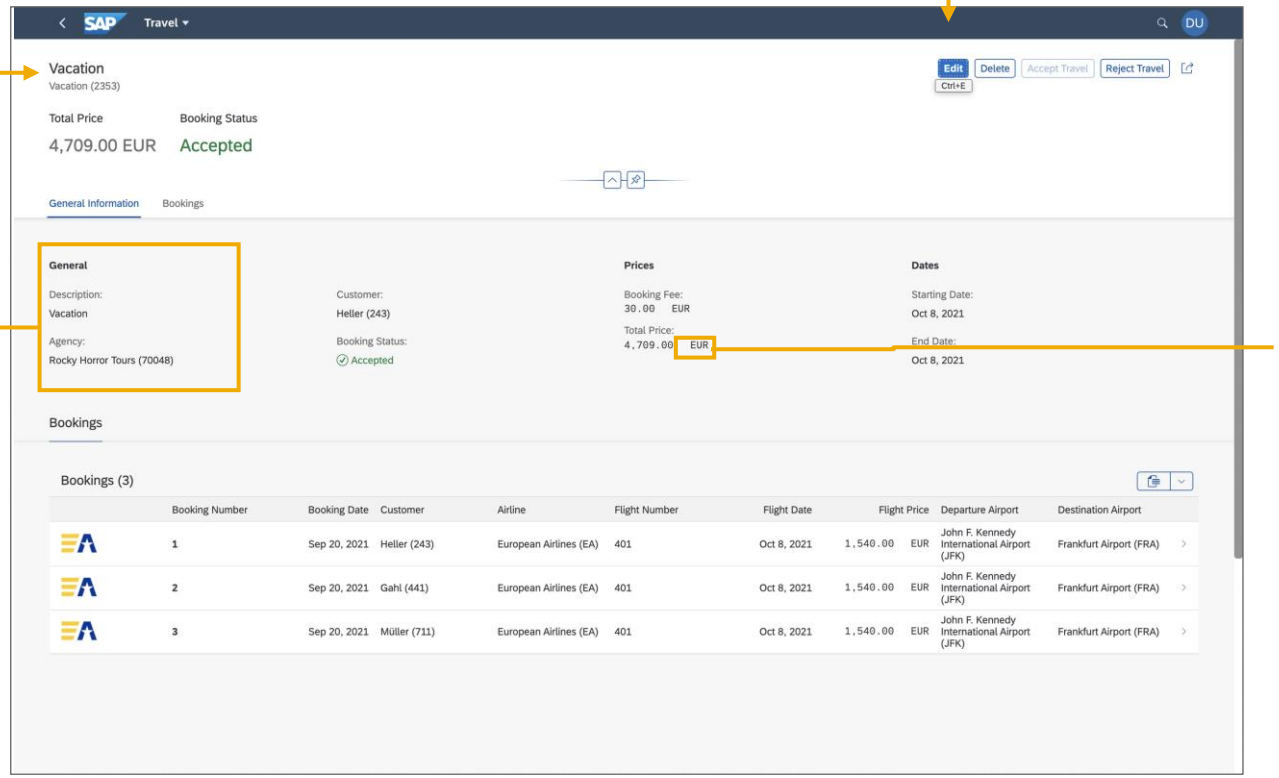
Consumption hints

Which fields allow a human to identify the entity?

Which fields are clickable?

Semantics

“These fields shall be displayed together in a form”



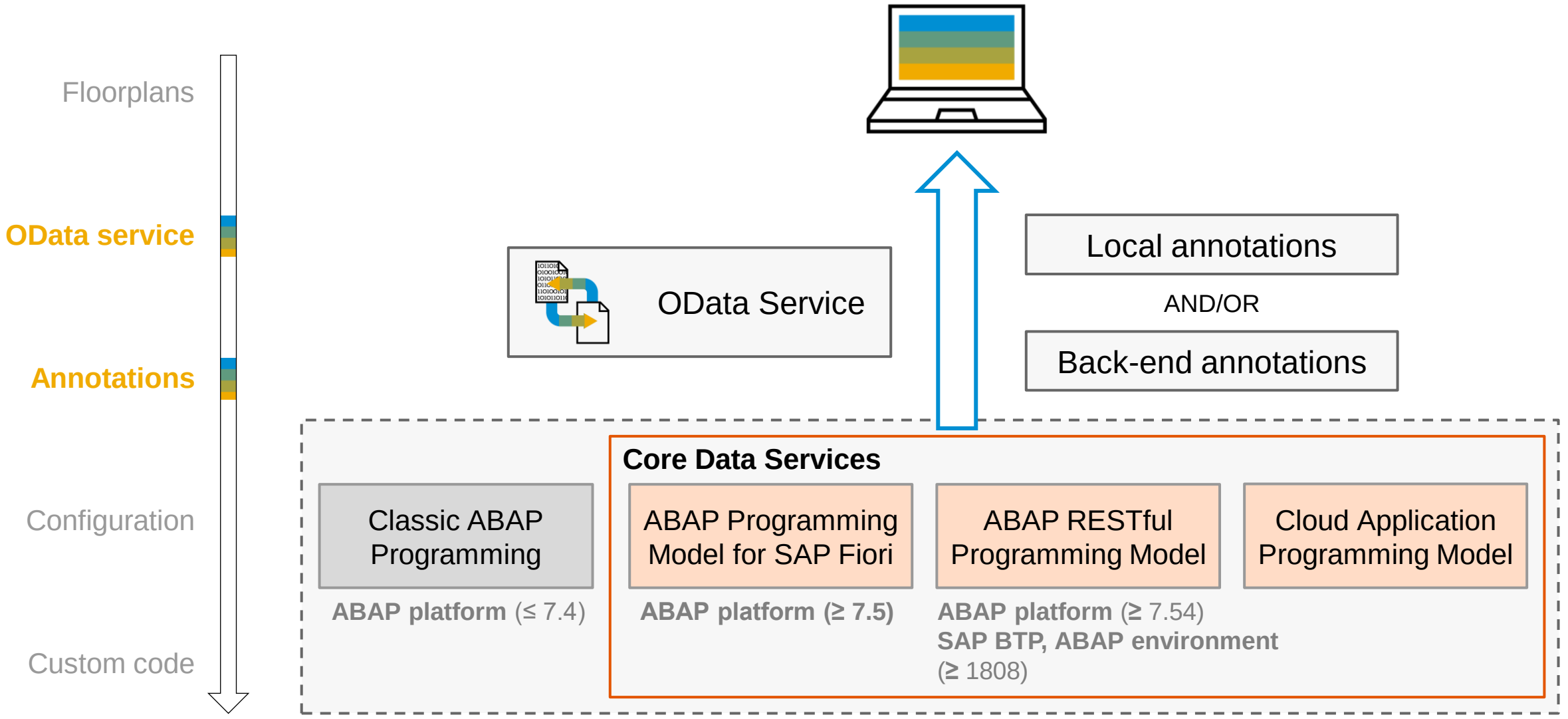
Behavior

“This entity can be edited”

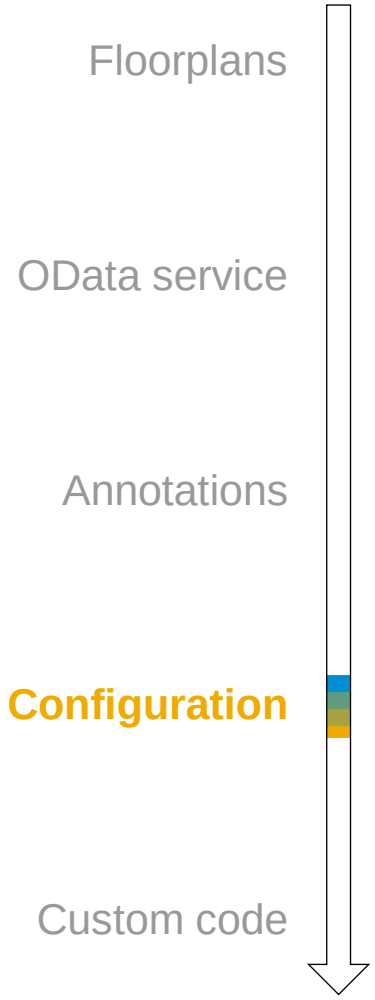
Semantics

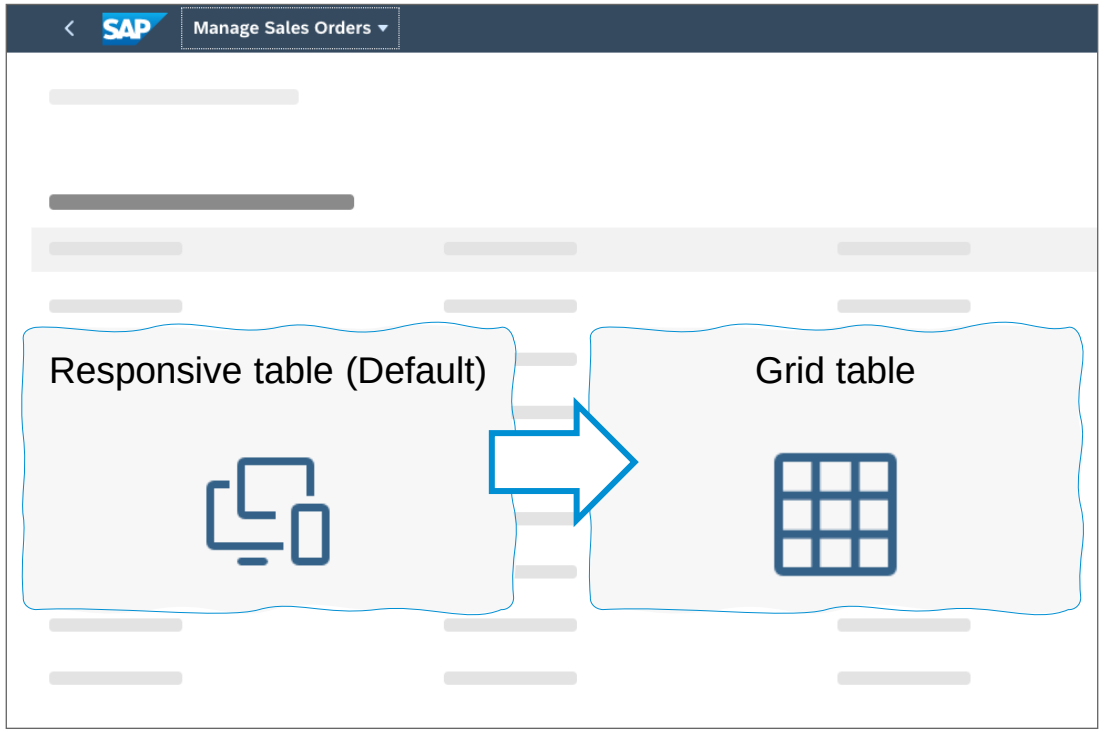
“This is the currency for the amount”

Use the power of SAP's back ends to create OData services



Configure the behavior and layout of your app

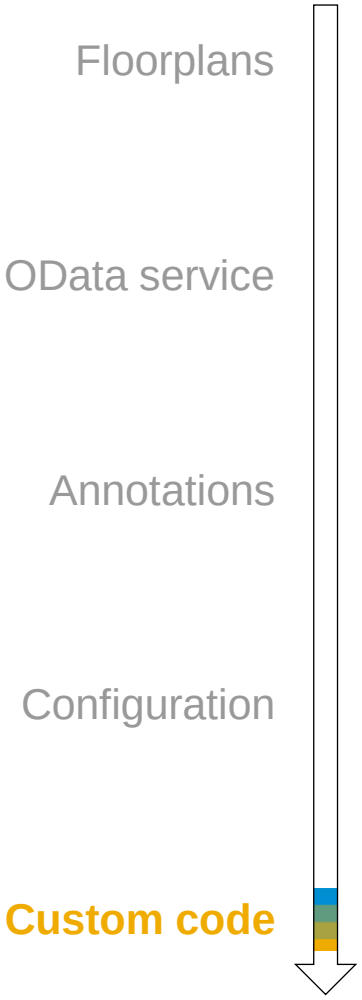


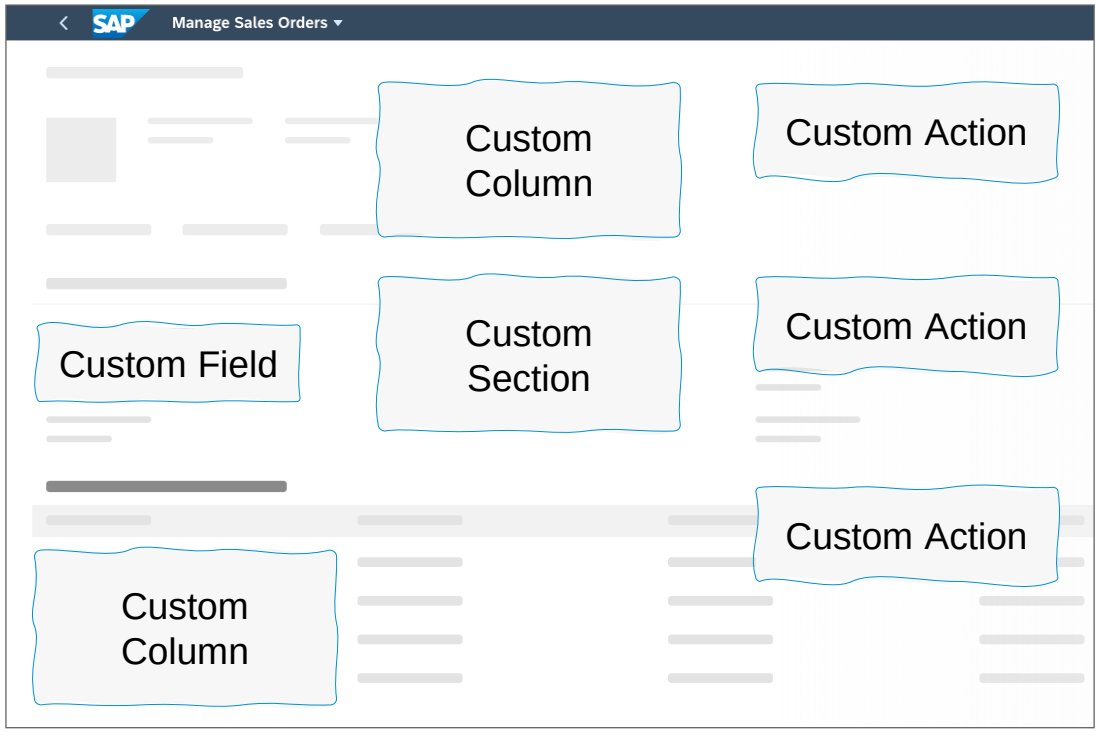


SAP Fiori elements provides

- Default configuration settings based on latest UX guidelines
- An easy way to change it using SAP Fiori tools
- Layered configuration using SAPUI5 flexibility

Custom code complements your app beyond standard SAP Fiori elements functionality





Extend your app with

- Custom controls
- Custom logic using the SAP Fiori elements ExtensionAPI

The closer you stay to the standard SAP Fiori elements floorplans, the greater the savings in development and maintenance costs

Wrap-up

IN THIS UNIT, YOU LEARNED

- About how SAP Fiori elements floorplans provide the basis for your application
- How OData services and annotations define your application
- How you can configure the behavior and layout of your app
- How you can add custom controls and logic to your SAP Fiori elements apps

NEXT UNIT

- Week 1 – Unit 3
  Introducing OData services and annotations

Thank you.

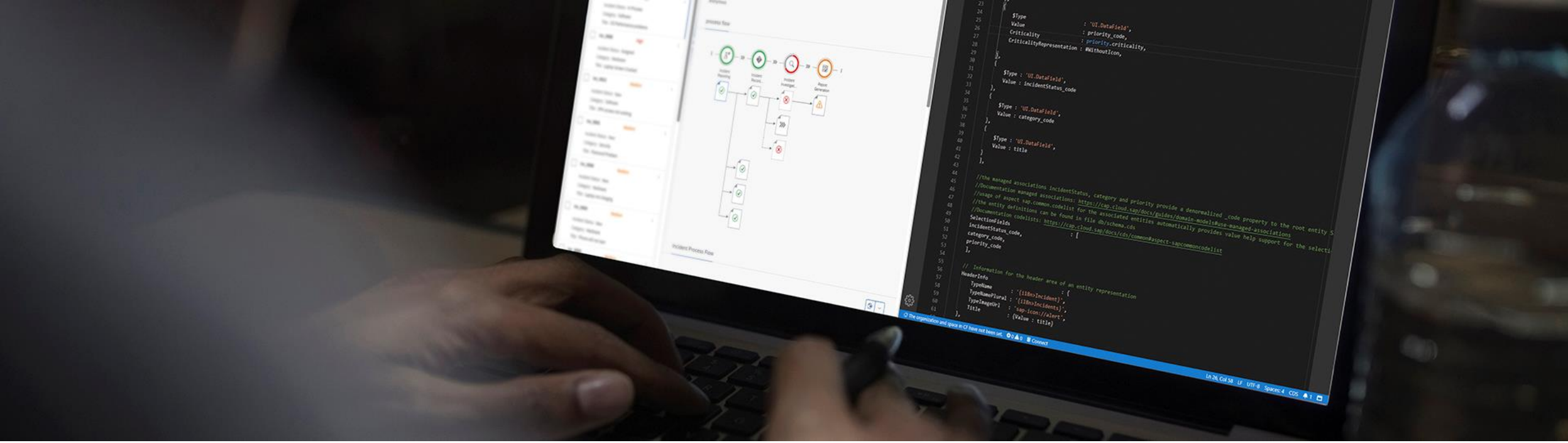
Contact information:

open@sap.com

Follow all of SAP

www.sap.com/contactsap

THE BEST RUN SAP®



Week 1: Introducing SAP Fiori Elements

Unit 3: Introducing OData Services and Annotations

Introducing SAP Fiori Elements

- Painting the Big Picture
- Understanding the Architecture
- Introducing OData Services and Annotations — you are here
- Preparing Your ABAP Development Environment
- Creating an OData Service with ABAP RESTful Application Programming Model
- Creating a List Report

Building Your First SAP Fiori Elements App

- Creating an Object Page
- Accelerating SAP Fiori App Development with SAP Fiori Tools
- Preparing Your Front-end Development Environment
- Generating the List Report – Object Page App
- Configuring the App
- Extending the App with Custom Elements

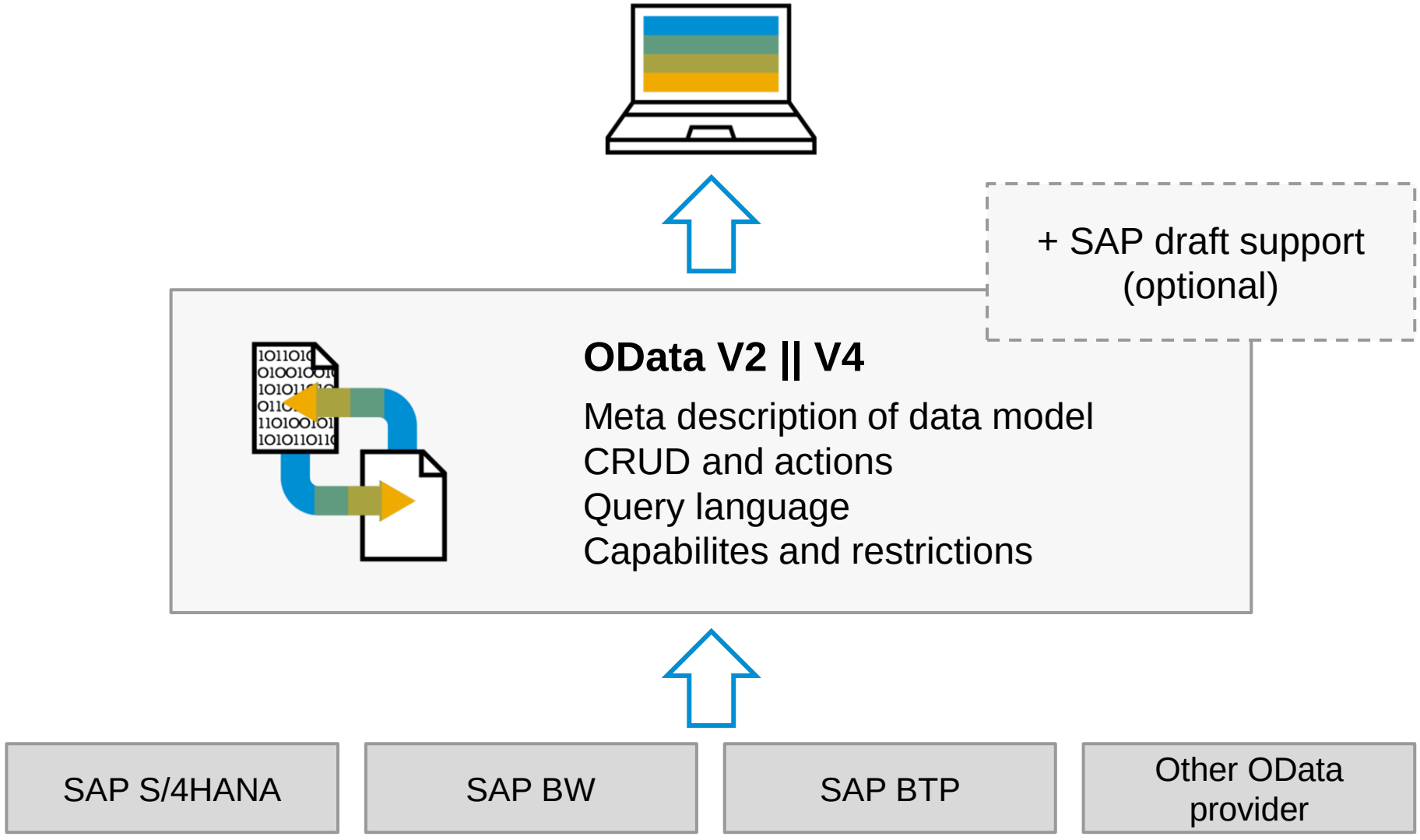
Using More SAP Fiori Elements Capabilities

- Creating an Analytical List Page
- Enhancing the Analytical List Page
- Creating an Overview Page
- Deploying Your SAP Fiori Elements App
- Building an SAP Fiori Elements App with an External OData Service
- Using XML Annotation LSP for Defining Advanced UI Features

Extending a Standard SAP Fiori App

- Understanding SAPUI5 Flexibility
- Adapting an App as a Key User
- Adapting the UI as a Developer – Basic Extensions
- Adapting the UI as a Developer – Advanced Extensions
- Productizing the Extended App

OData connects your application to the back end



OData metadata is a standardized and machine-readable description of the data model

OData contains a meta description of your data model:

- An **entity** can for example be a ‘Person’ with the **property** ‘FirstName’
- The property can be further **qualified**, for example by providing a **datatype** or **information** like ‘nullable’
- Entities can have **relationships**.
  A 1:n relationship could be, for example, ‘1 person can be related to n trips’
- The meta description can contain complex types, actions, singletons, and so on

Example: OData.org trip service metadata

```
<?xml version="1.0" encoding="utf-8"?>
<edmx:Edmx Version="4.0" xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">
  <edmx:DataServices>
    <Schema Namespace="Trippin" xmlns="http://docs.oasis-open.org/odata/ns/edm">
      <EntityType Name="Person">
        <Key>
          <PropertyRef Name="UserName" />
        </Key>
        <Property Name="UserName" Type="Edm.String" Nullable="false" />
        <Property Name="FirstName" Type="Edm.String" Nullable="false" />
        <Property Name="LastName" Type="Edm.String" MaxLength="26" />
        <Property Name="MiddleName" Type="Edm.String" />
        <Property Name="Gender" Type="Trippin.PersonGender" Nullable="false" />
        <NavigationProperty Name="Friends" Type="Collection(Trippin.Person)" />
        <NavigationProperty Name="BestFriend" Type="Trippin.Person" />
        <NavigationProperty Name="Trips" Type="Collection(Trippin.Trip)" />
      </EntityType>
      <EntityType Name="Airline">
        <Key>
          <PropertyRef Name="AirlineCode" />
        </Key>
        ...
```

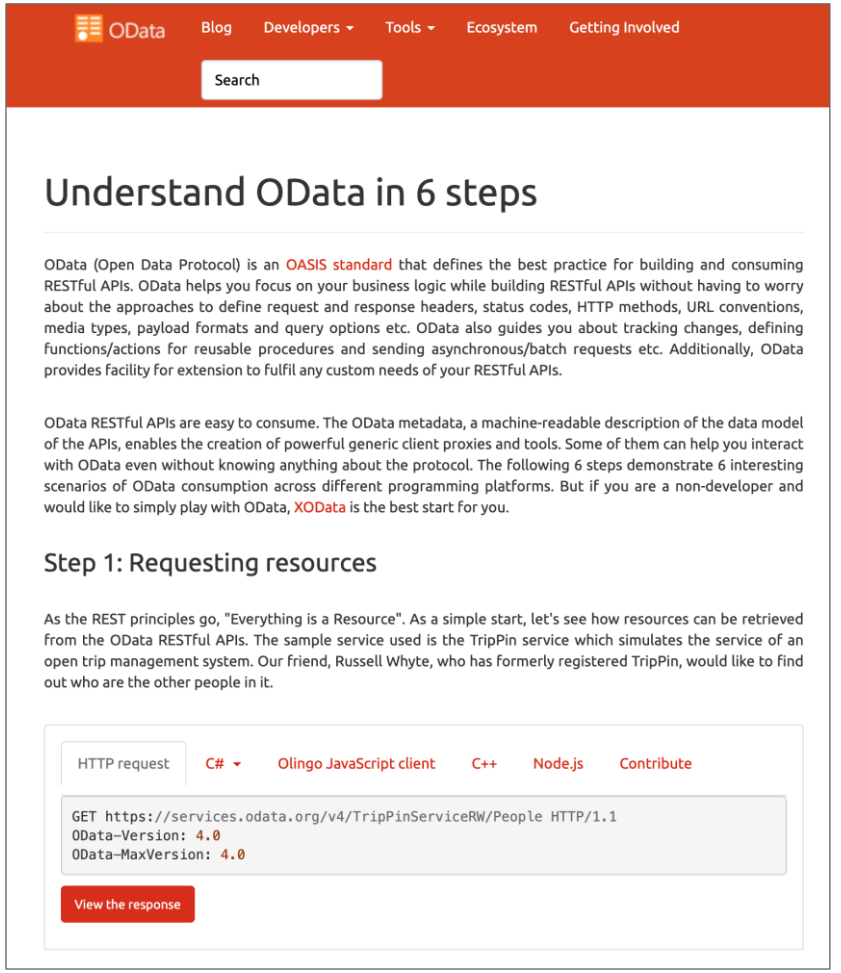
OData lets you request and modify resources and invoke actions

Requesting resources

- Get **list of people**
  `GET https://.../TripPinServiceRW/People`
- Get the **data of an individual person**
  `GET https://.../TripPinServiceRW/People('russellwhyte')`
- **Complex queries** like 'Get the first 2 persons in the system who have registered at least one trip that costs more than 3000, and only display their first name and last name.':
  `GET https://.../TripPinServiceRW/People?$top=2 &`
  `$select=FirstName, LastName &`
  `$filter=Trips/any(d:d/Budget gt 3000)`

Creating a resource

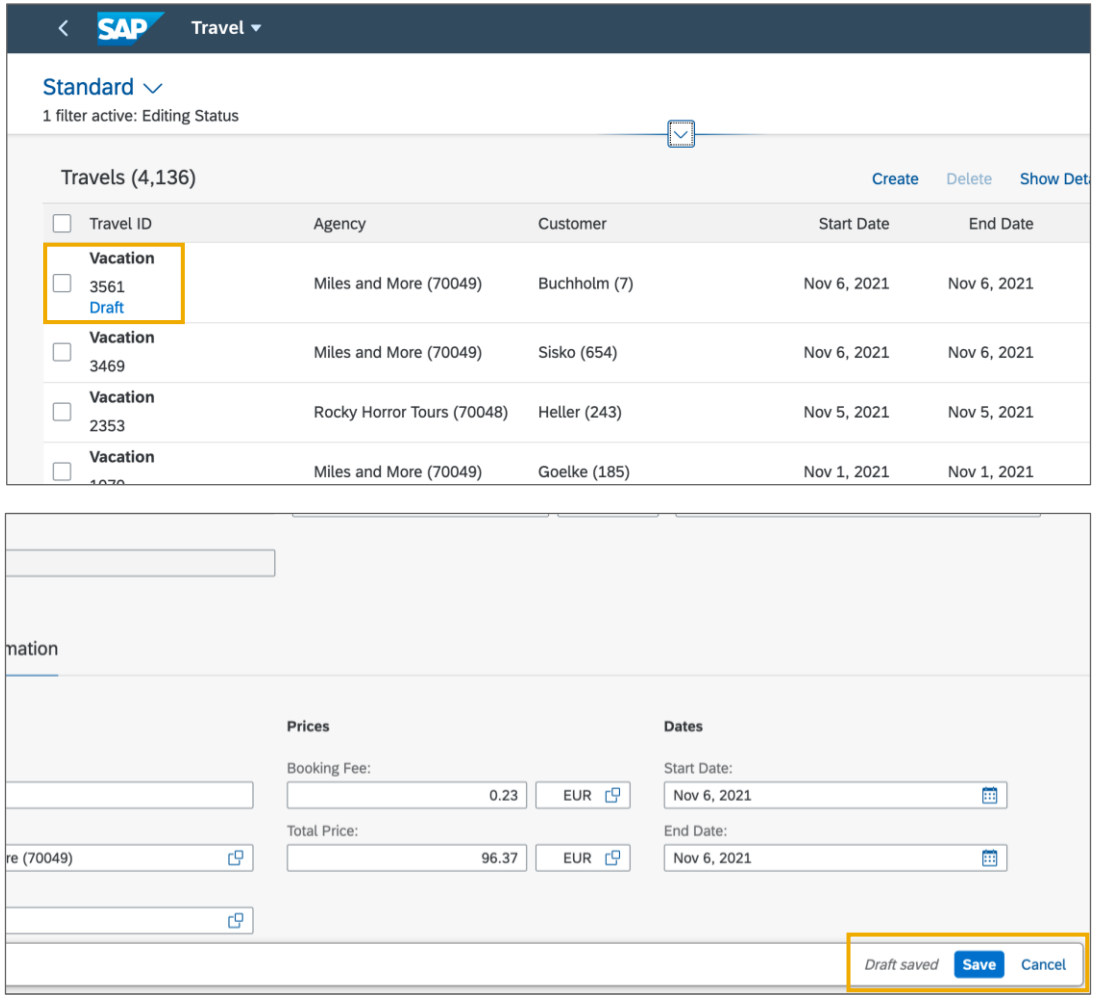
- `POST https://...TripPinServiceRW/People` and provide data like first name, last name etc. in the payload



https://www.odata.org/getting-started/understand-odata-in-6-steps/

SAP defined a way to implement a draft pattern using OData

- Using the draft pattern is optional, but makes sense for creating and updating business objects to keep unsaved changes if an editing activity is interrupted, allowing users to resume editing later.
- A draft is an interim version of a business entity that has not been explicitly saved as an active version to the original database tables.
- In SAP Fiori elements, drafts are saved automatically in the background whenever users add or change information within a business entity while in edit state (autosave). This ongoing communication with the back end also enables dynamic interaction patterns.
- Draft handling should be implemented using a framework like ABAP RESTful Application Programming Model, ABAP Programming Model for SAP Fiori, or SAP Cloud Application Programming Model.



OData annotations define how the app looks and behaves

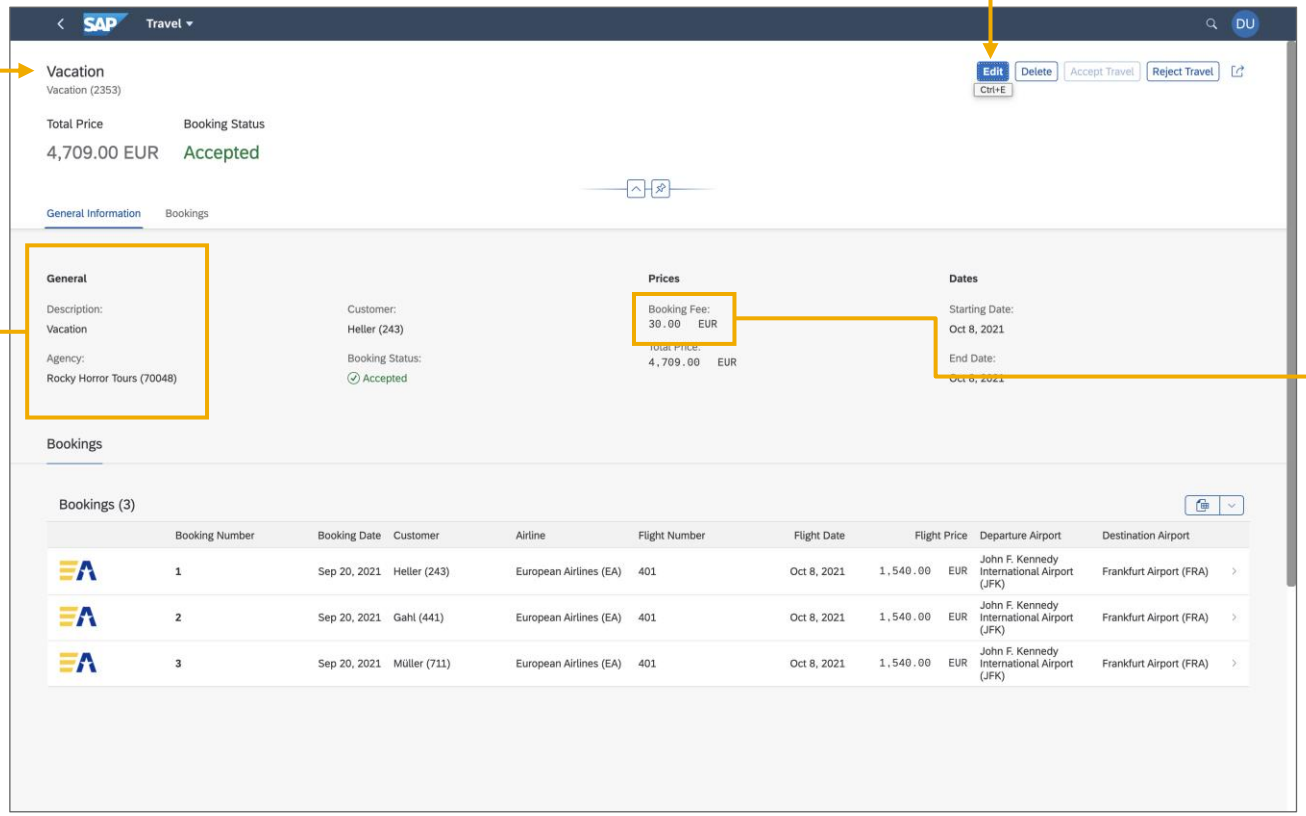
Consumption hints

Which fields allow a human to identify the entity?

Which fields are clickable?

Semantics

“These fields shall be displayed together in a form”



Behavior

“This entity can be edited”

Semantics

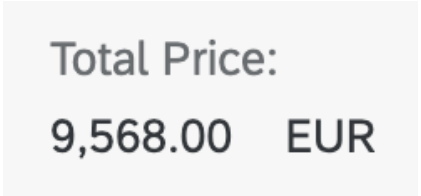
“This is the currency for the amount”

Annotations provide semantics and consumption hints

Annotations are additional metadata and provide abstract definitions of data semantics. These abstract definitions are stored in vocabularies from OData and SAP. SAP Fiori elements understand the annotations and generates the UI based on them.

Example:
Specify that the field “TotalPrice” has a currency assigned to it. In the measures vocabulary of OData there is a term ISOCurrency defined for that:

```
<Annotations Target="SAP__self.TravelType/TotalPrice">
  <Annotation Term="SAP__measures.ISOCurrency" Path="CurrencyCode"/>
</Annotations>
```

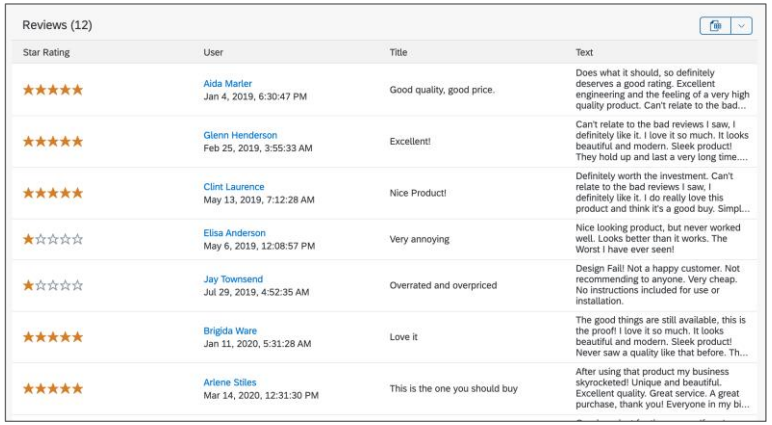


SAP Fiori elements can then know that whenever the “TotalPrice” is shown on the UI, the field “CurrencyCode” comprises its currency and should be shown next to it.

Annotations can also define standard UI elements like table, form, or chart

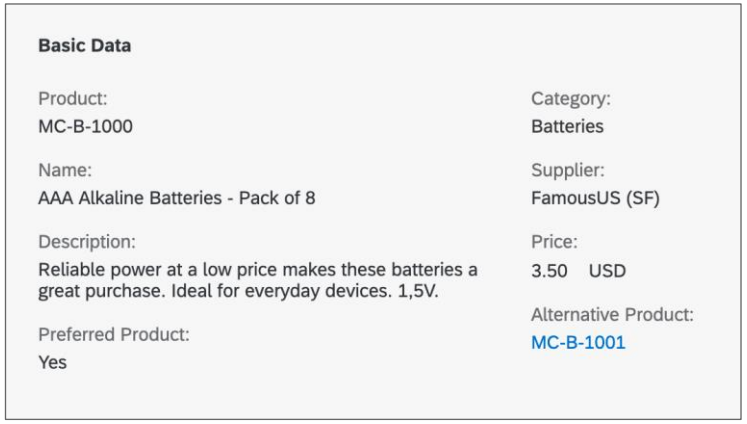
UI.LineItem

Annotations define columns, actions, etc.



UI.FieldGroup

Annotations define the group title and form fields



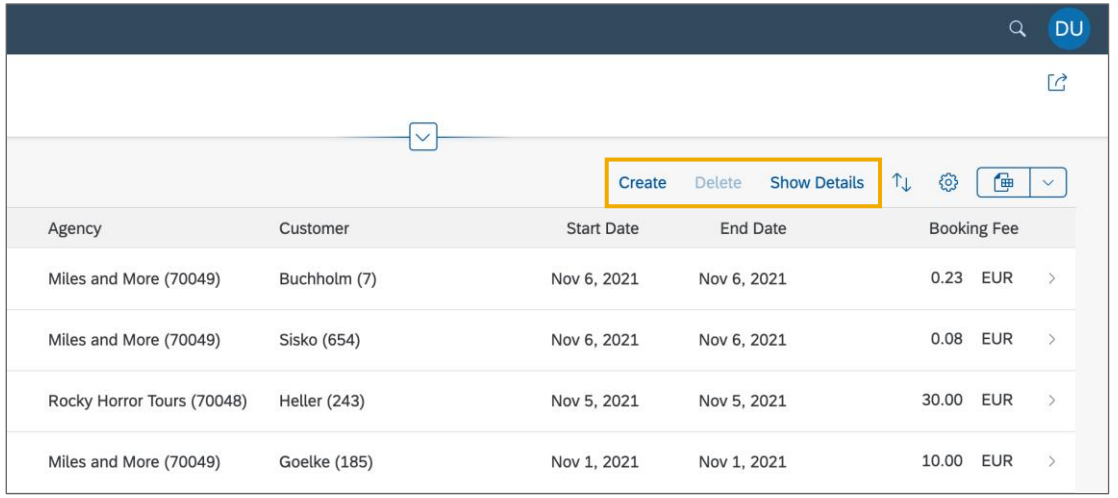
UI.Chart

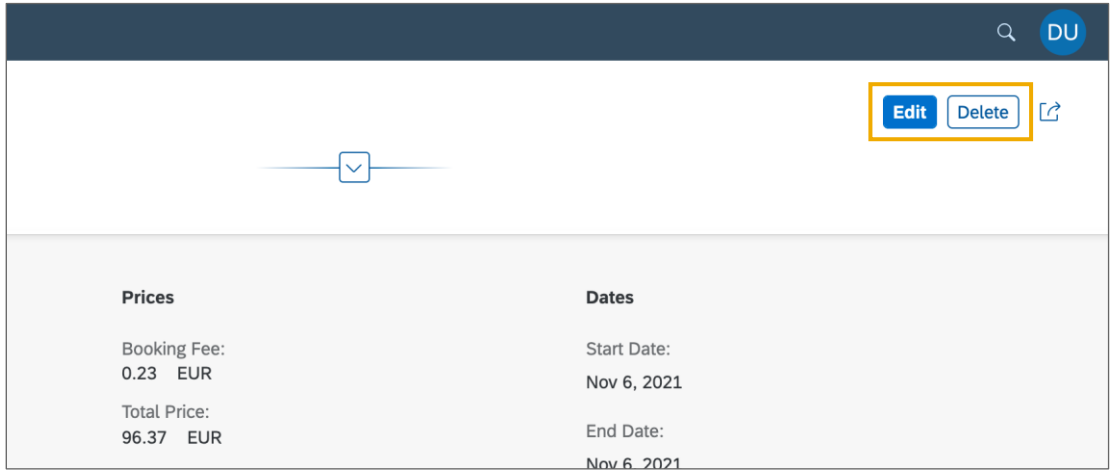
Annotations define measures, dimensions, chart type, etc.

Annotations also describe behavioral aspects of the application

Examples of behavioral aspects are:

- An instance of a business entity can be edited and deleted. Based on this information, the corresponding buttons can be shown/hidden on the UI.
- An action can be executed on an instance of a business entity. Only if the action is applicable is a corresponding button enabled/displayed.
- When specific fields get changed on the UI it is required to request additional data from the back end. For example, when a price on an item table gets changed, the UI needs to request the new total price of the header entity from the back end.





SAP Fiori elements apps can be built using ABAP CDS, CAP CDS, and XML annotations

- All types of annotations enrich the OData service with semantic information that **allows the generation of metadata-driven UIs**
- Annotations can be defined either locally (XML annotations) or in the backend (ABAP CDS, CAP CDS annotations)
- Annotations in CDS views get translated to XML annotations by the corresponding back ends
- The type of annotation (ABAP CDS, CAP CDS, and XML) does not impact performance or functionality
- XML annotations can be used with any back end, for example, SAP back ends leveraging previous ABAP programming models, SAP BW, or non-SAP back ends

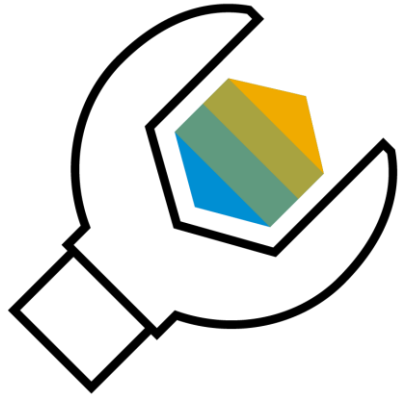
```
@UI.hidden: true
TravelUUID;

@UI: {
  lineItem: [ { position: 10, importance: #HIGH } ],
  selectionField: [ { position: 10 } ]
}
TravelID;

@UI: {
  lineItem:       [ { position: 20, importance: #HIGH } ],
  identification: [ { position: 30 } ],
  selectionField: [ { position: 20 } ]
}
AgencyID;

@UI: {
  lineItem:       [ { position: 30, importance: #HIGH } ],
  identification: [ { position: 40 } ],
  selectionField: [ { position: 30 } ]
}
CustomerID;
```

Annotations in ABAP CDS

Your preferred tooling affects your choice of annotation type

SAP Fiori tools
(available in **SAP Business Application Studio** or **Visual Studio Code**)

- Makes it easy and convenient to define UI semantics
- Reduces effort and the required skill level through step-by-step guides and code completion, for example
- Both **XML** and **CAP CDS** annotations are supported by SAP Fiori tools



ABAP Development Tools in Eclipse

- Used to maintain **ABAP CDS** annotations for UI semantics and other domains as well (for example, search, analytics, or ObjectModel)
- SAP Fiori tools still needed for application generation

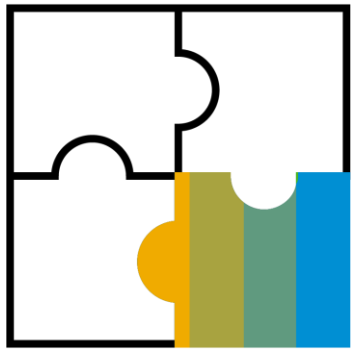
Lifecycle management and feature availability influence your choice of annotation type

Where do you want to manage the lifecycle?

- XML annotations follow the lifecycle of your app (local project)
- CDS annotations follow the lifecycle of your OData service

Do you require recently released features?

- Due to different release cycles, some new annotations might be available later in ABAP CDS
- For on-premise SAP systems, upgrade cycles can further impact the availability of new features for ABAP CDS annotations



Wrap-up

IN THIS UNIT, YOU LEARNED

- What OData is
- The role of drafts
- How annotations affect your app
- That you can use different types of annotations

NEXT UNIT

- Week 1 – Unit 4
  Preparing your ABAP development environment

Thank you.

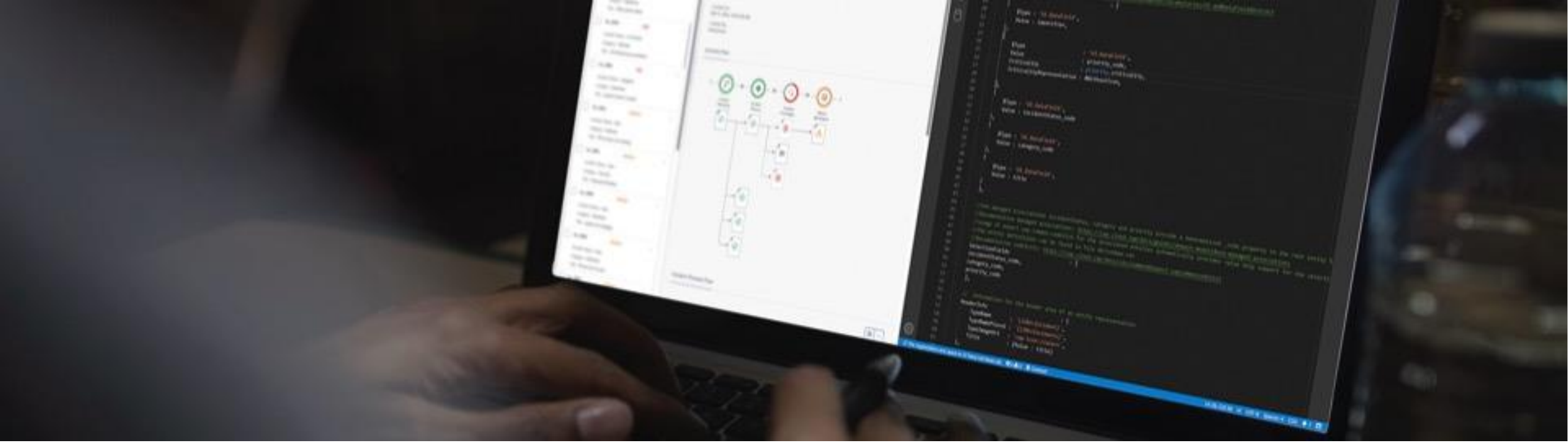
Contact information:

open@sap.com

Follow all of SAP

www.sap.com/contactsap

THE BEST RUN SAP®



Week 1: Introducing SAP Fiori Elements

Unit 4: Preparing Your ABAP Development Environment

Introducing SAP Fiori Elements

- Painting the Big Picture ✓
- Understanding the Architecture ✓
- Introducing OData Services and Annotations ✓
- Preparing Your ABAP Development Environment — you are here
- Creating an OData Service with ABAP RESTful Application Programming Model
- Creating a List Report

Building Your First SAP Fiori Elements App

- Creating an Object Page
- Accelerating SAP Fiori App Development with SAP Fiori Tools
- Preparing Your Front-end Development Environment
- Generating the List Report – Object Page App
- Configuring the App
- Extending the App with Custom Elements

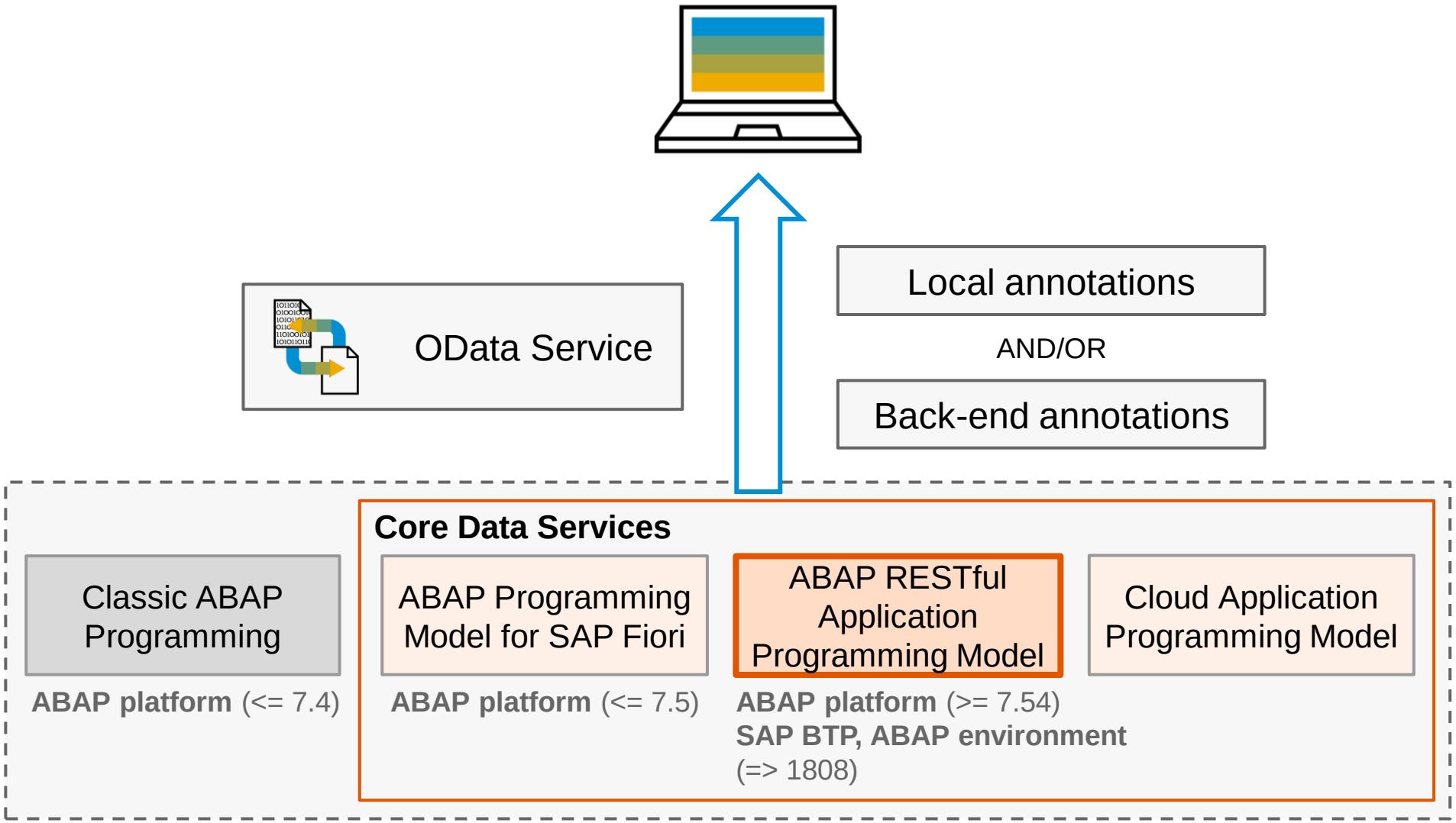
Using More SAP Fiori Elements Capabilities

- Creating an Analytical List Page
- Enhancing the Analytical List Page
- Creating an Overview Page
- Deploying Your SAP Fiori Elements App
- Building an SAP Fiori Elements App with an External OData Service
- Using XML Annotation LSP for Defining Advanced UI Features

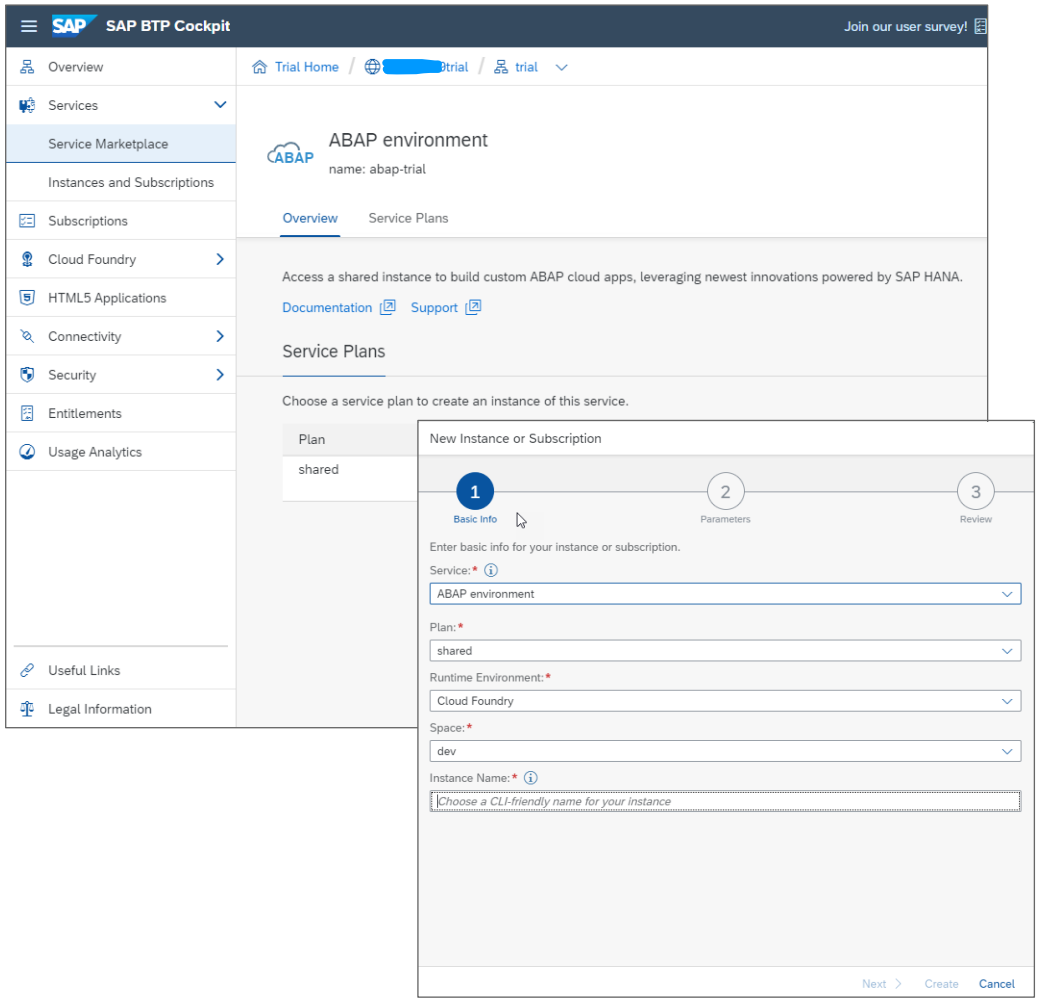
Extending a Standard SAP Fiori App

- Understanding SAPUI5 Flexibility
- Adapting an App as a Key User
- Adapting the UI as a Developer – Basic Extensions
- Adapting the UI as a Developer – Advanced Extensions
- Productizing the Extended App

In this course we will use the ABAP environment on SAP Business Technology Platform



Register for an SAP BTP, ABAP environment trial account



Access to trial system

- Free of charge
- Via SAP Business Technology Platform trial access https://www.sap.com/cmp/td/sap-cloud-platform-trial.html
- Trial account expires after 365 days

Rules of the game

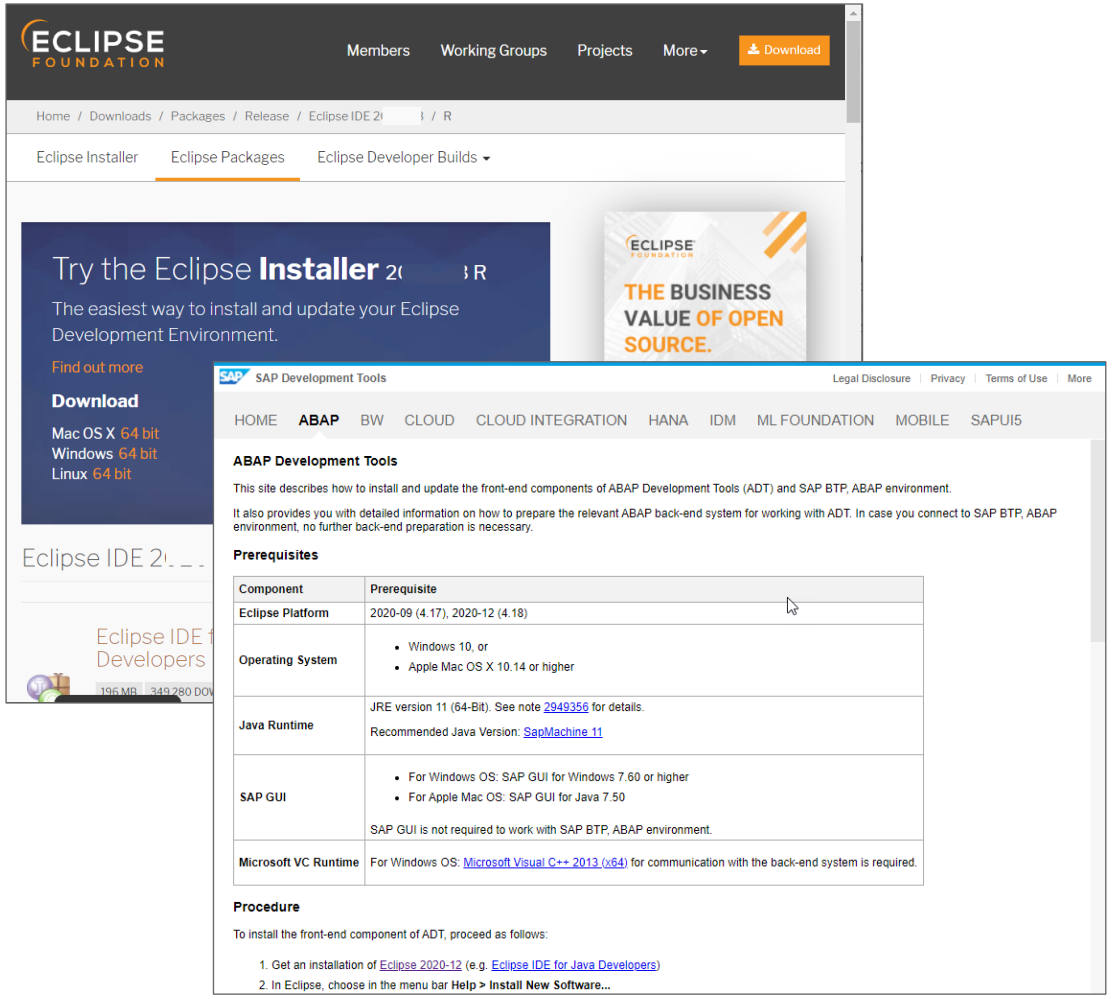
Shared Trial Offering

For educational purposes only
No service-level agreement with regard to platform availability

Trial users share the same back-end system

No content separation between different users

Install and connect ABAP Development Tools to the ABAP environment trial system



Access to ABAP tools

- Instructions about the local installation of Eclipse and ADT available on the central update site for all Eclipse-based SAP development tools
- https://tools.hana.ondemand.com/#abap

Wrap-up

IN THIS UNIT, YOU LEARNED

- How to set up your ABAP development environment for this course

NEXT UNIT

- Week 1 – Unit 5
  Creating an OData Service using ABAP RESTful Application Programming Model (RAP)

Thank you.

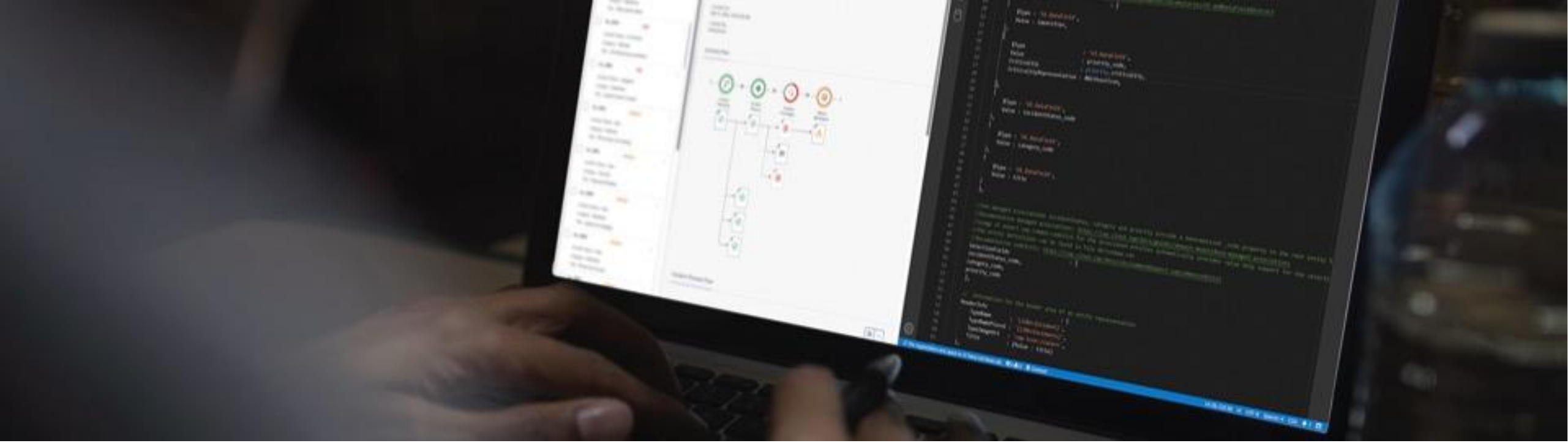
Contact information:

open@sap.com

Follow all of SAP

www.sap.com/contactsap

THE BEST RUN SAP®



Week 1: Introducing SAP Fiori elements

Unit 5: Creating an OData Service with ABAP RESTful Application Programming Model (RAP)

Introducing SAP Fiori Elements

- Painting the Big Picture
- Understanding the Architecture
- Introducing OData Services and Annotations
- Preparing Your ABAP Development Environment
- Creating an OData Service with ABAP RESTful Application Programming Model — you are here
- Creating a List Report

Building Your First SAP Fiori Elements App

- Creating an Object Page
- Accelerating SAP Fiori App Development with SAP Fiori Tools
- Preparing Your Front-end Development Environment
- Generating the List Report – Object Page App
- Configuring the App
- Extending the App with Custom Elements

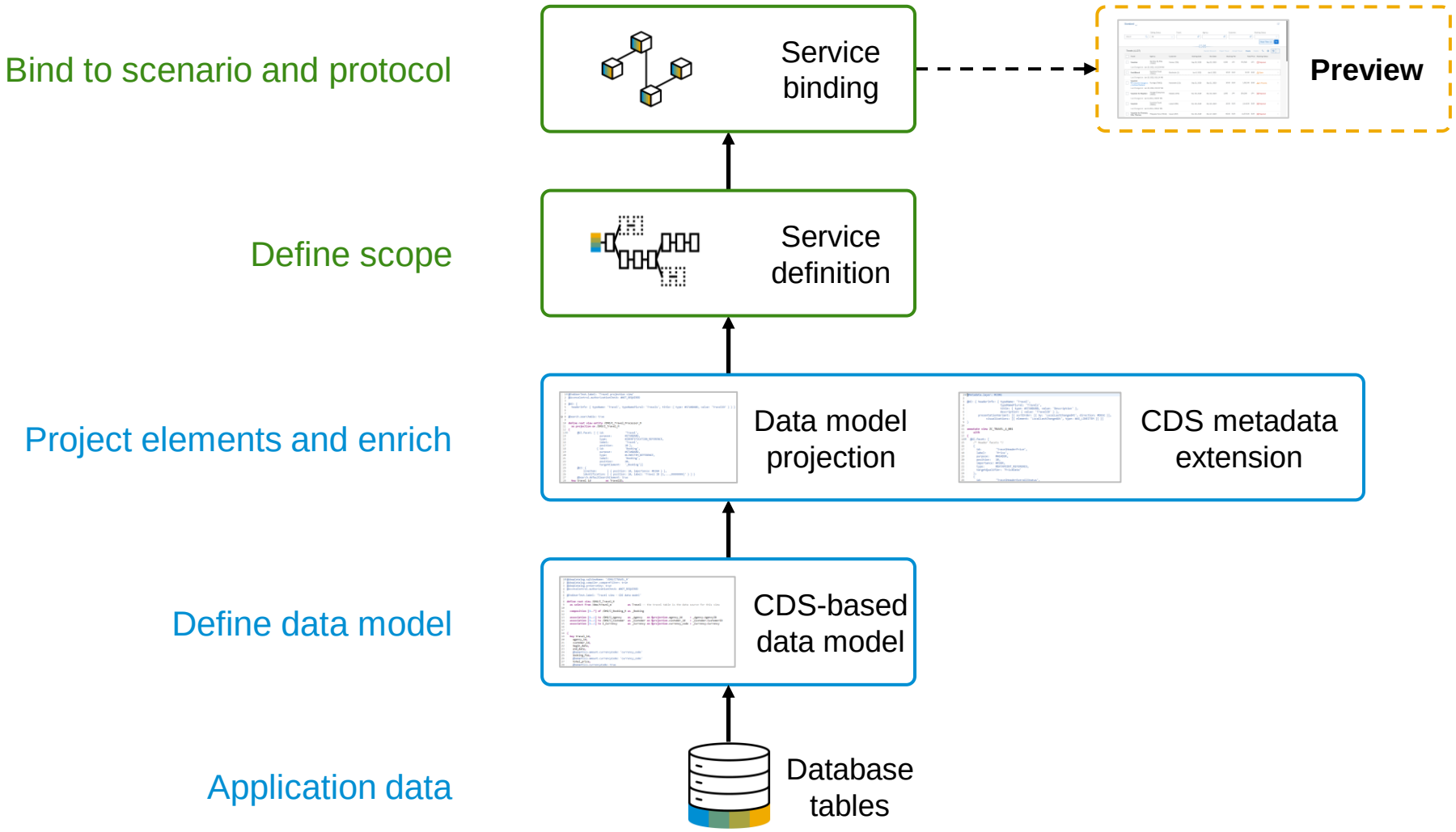
Using More SAP Fiori Elements Capabilities

- Creating an Analytical List Page
- Enhancing the Analytical List Page
- Creating an Overview Page
- Deploying Your SAP Fiori Elements App
- Building an SAP Fiori Elements App with an External OData Service
- Using XML Annotation LSP for Defining Advanced UI Features

Extending a Standard SAP Fiori App

- Understanding SAPUI5 Flexibility
- Adapting an App as a Key User
- Adapting the UI as a Developer – Basic Extensions
- Adapting the UI as a Developer – Advanced Extensions
- Productizing the Extended App

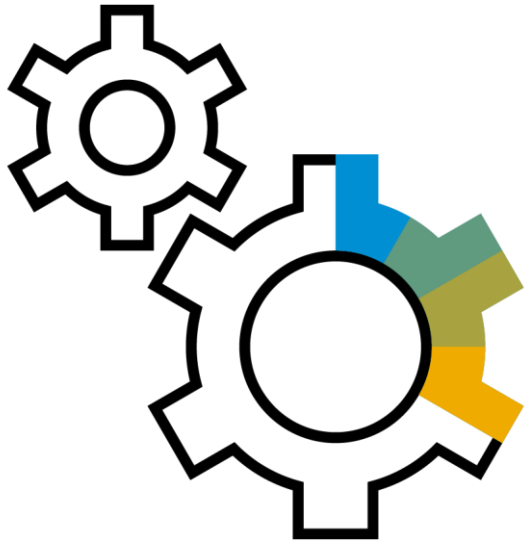
Our travel apps need back-end logic implemented by several development artifacts



A generator will create the back-end artifacts and the OData services

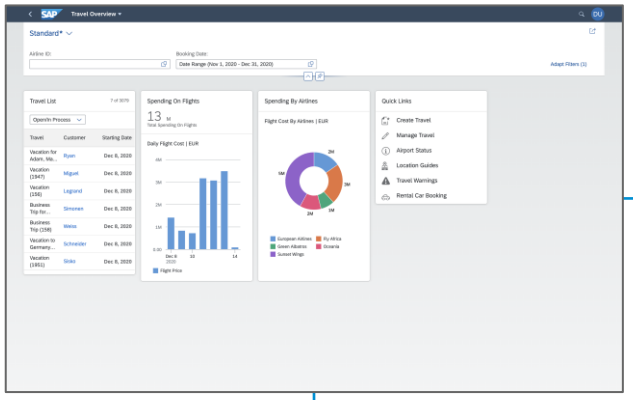
As this openSAP course focuses on SAP Fiori elements and not the creation of the OData service, a generator was developed for the course which creates all the required RAP development artifacts for you. You can then use this OData service and enhance its metadata extensions.

However, if you're interested in how to build such an OData service from scratch using RAP, check out the openSAP course Building Apps with the ABAP RESTful Application Programming Model.

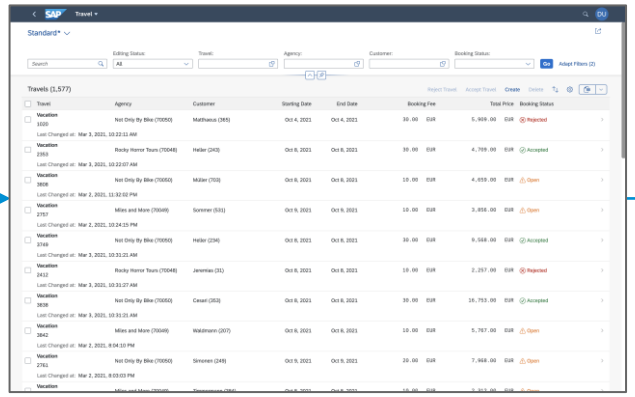


We will build the OData services for all the travel-related apps in this unit

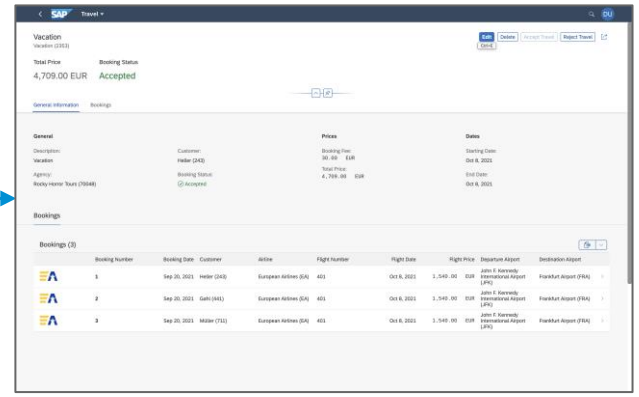
Travel overview



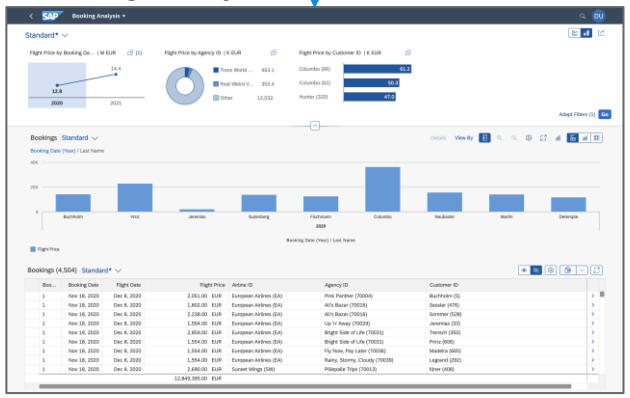
Travel list



Travel details

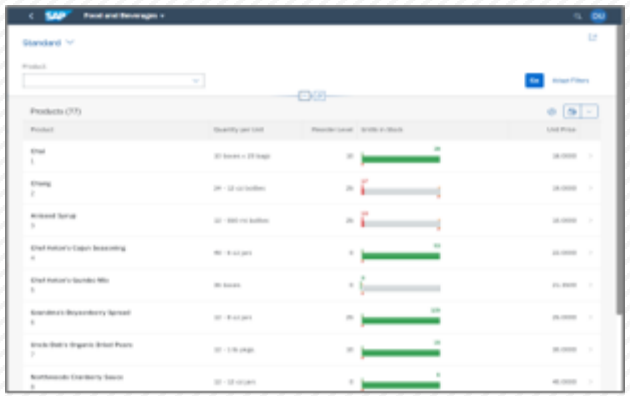


Booking analysis



Travel desk clerk Jay

Food and beverages



Wrap-up

IN THIS UNIT, YOU LEARNED

- How to generate the back-end artifacts and the OData services you need to build the travel-related apps

NEXT UNIT

- Week 1 – Unit 6
  Creating a list report

Thank you.

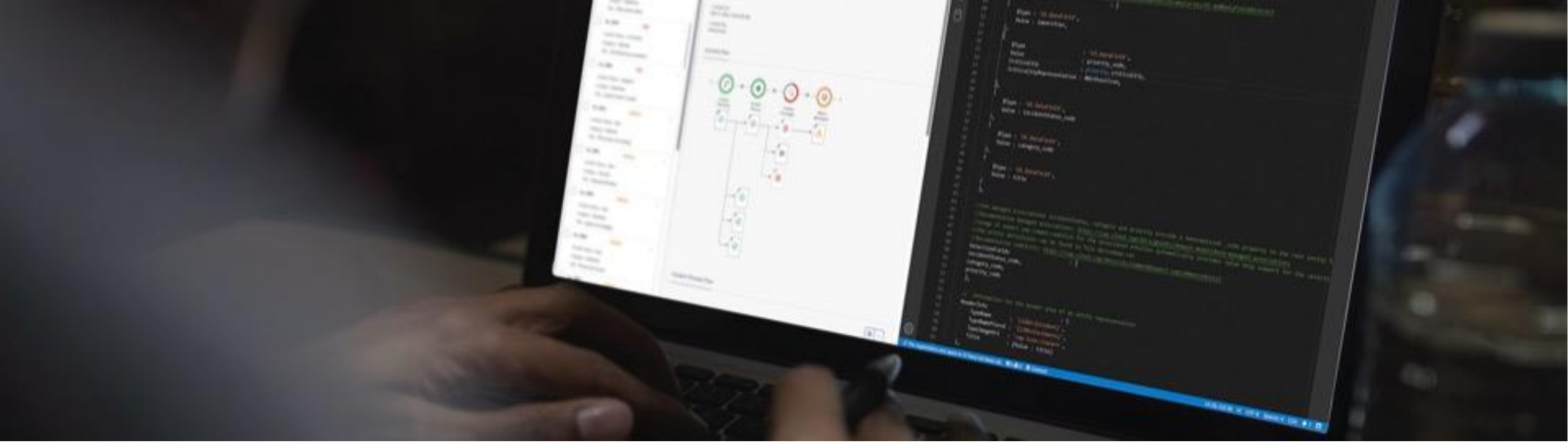
Contact information:

open@sap.com

Follow all of SAP

www.sap.com/contactsap

THE BEST RUN SAP®



Week 1: Introducing SAP Fiori Elements

Unit 6: Creating a List Report

Introducing SAP Fiori Elements

- Painting the Big Picture ✓
- Understanding the Architecture ✓
- Introducing OData Services and Annotations ✓
- Preparing Your ABAP Development Environment ✓
- Creating an OData Service with ABAP RESTful Application Programming Model ✓
- Creating a List Report — you are here

Building Your First SAP Fiori Elements App

- Creating an Object Page
- Accelerating SAP Fiori App Development with SAP Fiori Tools
- Preparing Your Front-end Development Environment
- Generating the List Report – Object Page App
- Configuring the App
- Extending the App with Custom Elements

Using More SAP Fiori Elements Capabilities

- Creating an Analytical List Page
- Enhancing the Analytical List Page
- Creating an Overview Page
- Deploying Your SAP Fiori Elements App
- Building an SAP Fiori Elements App with an External OData Service
- Using XML Annotation LSP for Defining Advanced UI Features

Extending a Standard SAP Fiori App

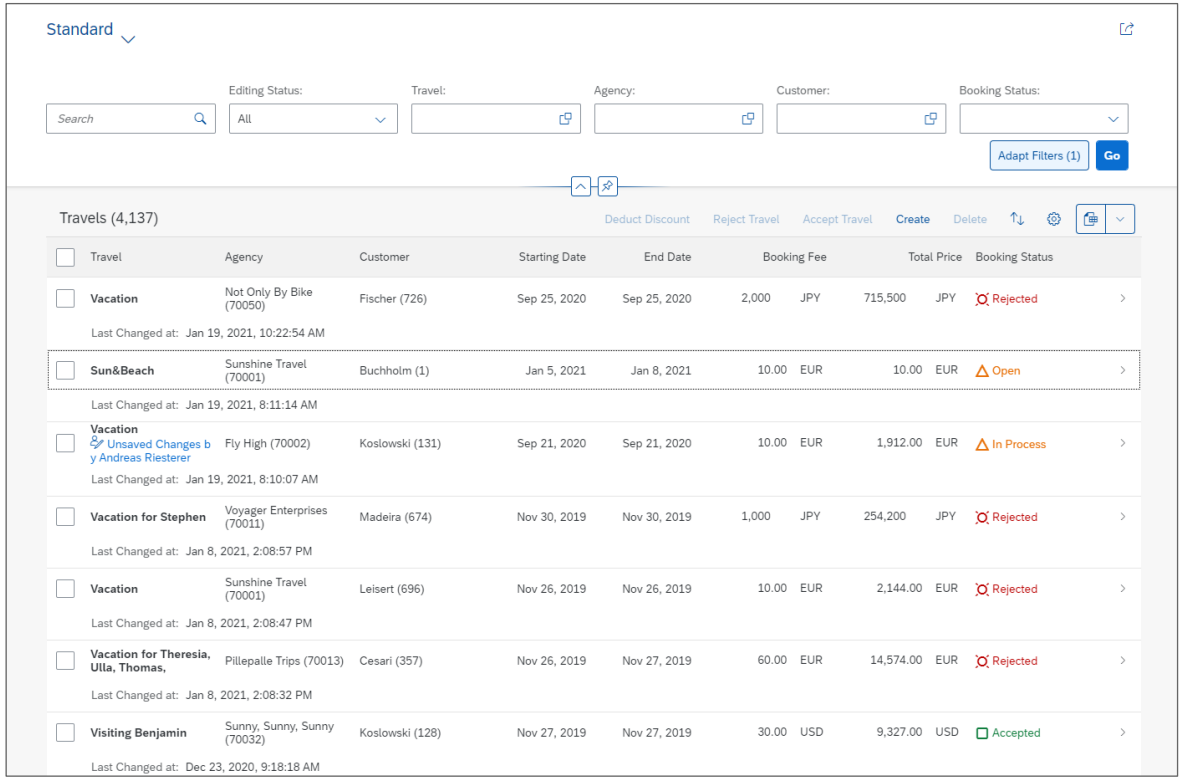
- Understanding SAPUI5 Flexibility
- Adapting an App as a Key User
- Adapting the UI as a Developer – Basic Extensions
- Adapting the UI as a Developer – Advanced Extensions
- Productizing the Extended App

A list report allows users to view and work with a large set of items

The **list report** offers powerful features for finding and acting on relevant data sets. It is often used as an entry point for navigating to the item details, which are usually shown on an object page.

With the list report, users can

- Search, filter, sort, and group large sets of items
- Work with different views on the same set of data, for example, on items with different statuses like ‘open’ or ‘in progress’
- Select one or more items in the list for further action



The list report has several key functions

Filter bar

- Allows users to filter the table based on various selection fields

Variant management

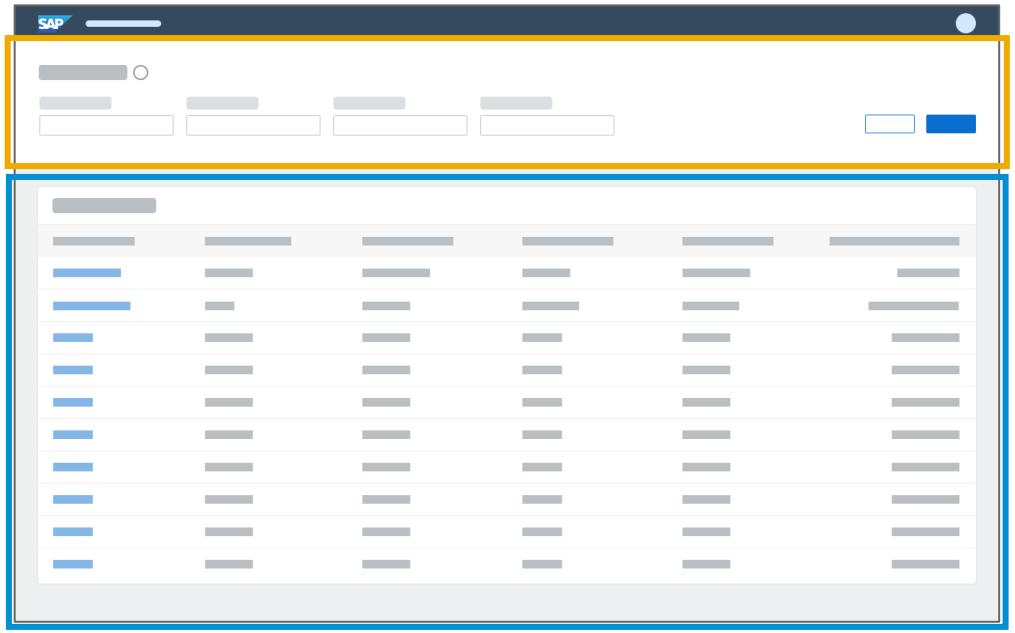
- Allows users to store custom views, such as filter settings, sorting, or table columns to be displayed

Visibility

- Is expanded initially and collapses when the user scrolls in the table
- Can be pinned to stay expanded during scrolling

Table toolbar

- Contains the table title, personalization and export options, as well as the standard actions create, edit, delete
- Can also contain app-specific actions



Check the infographic in our documentation

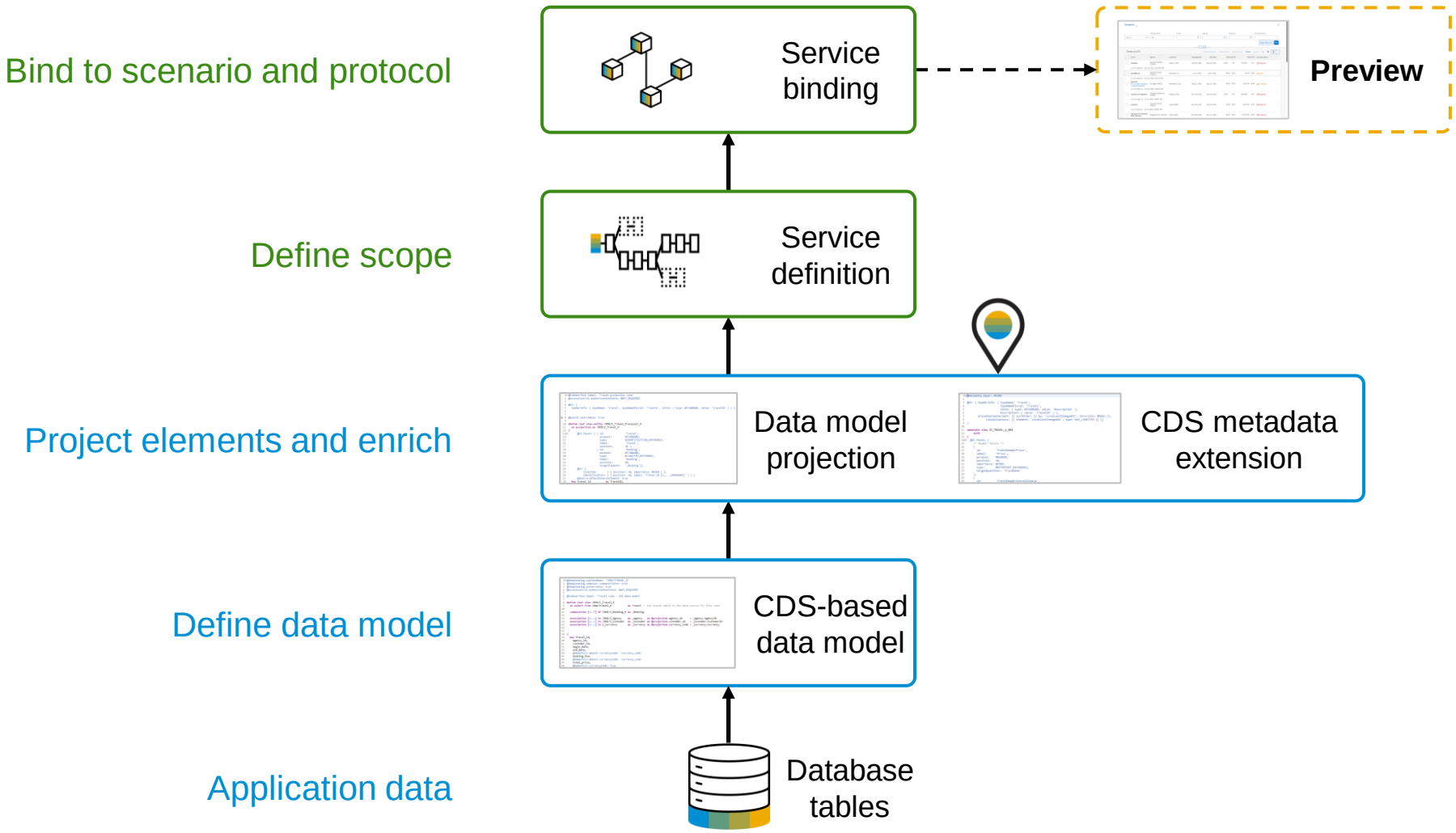
Personalization options

- Show and hide table columns, specify sorting and grouping

Table

- Multiple table types are supported, such as responsive table or grid table

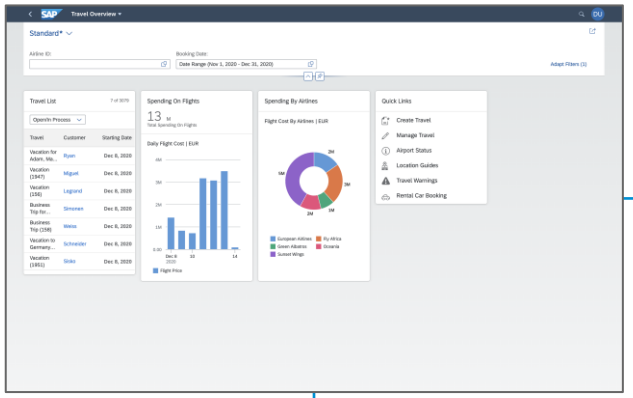
We will create UI-related ABAP CDS annotations primarily in the CDS metadata extension



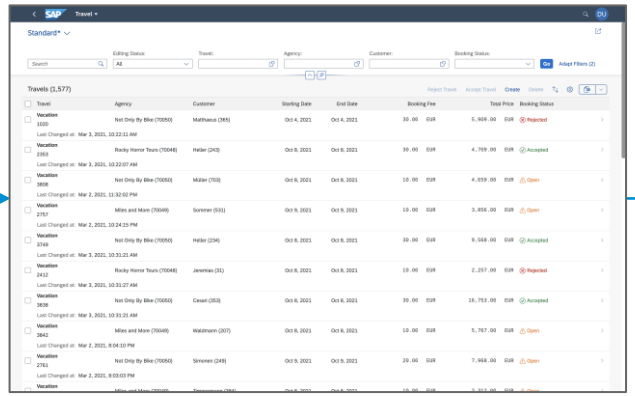
Creating a list report

We will build the travel list page in this unit

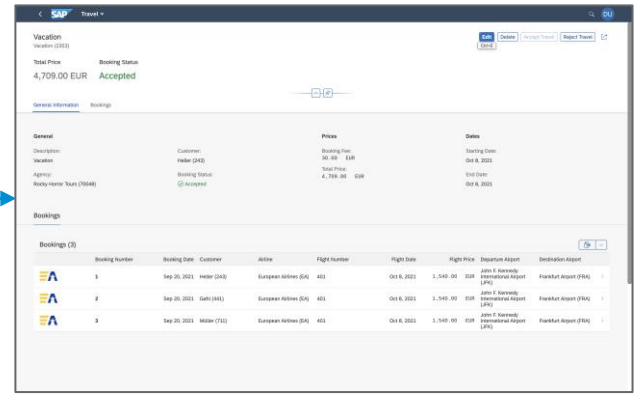
Travel overview



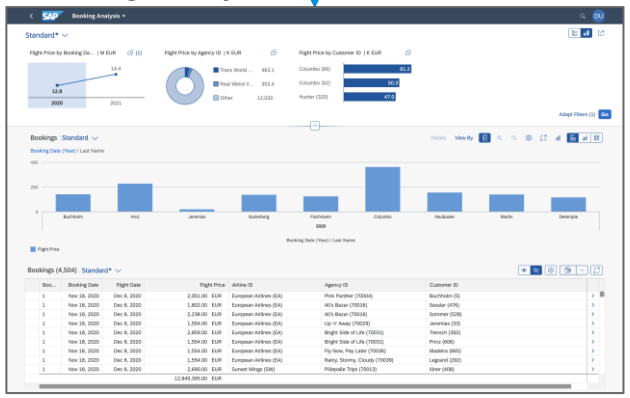
Travel list



Travel details

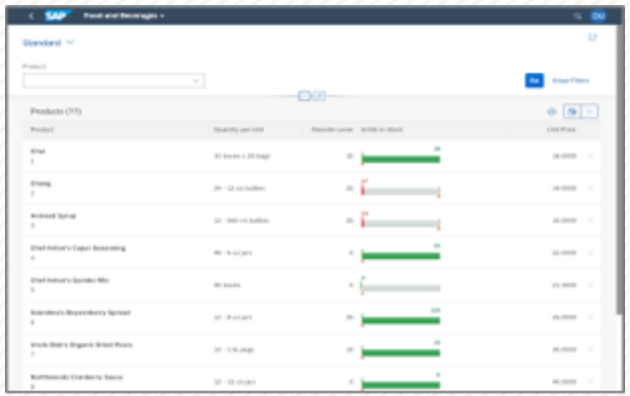


Booking analysis



Travel desk clerk Jay

Food and beverages



Wrap-up

IN THIS UNIT, YOU LEARNED

- When to use a list report floorplan
- About the components in a list report
- How to create a list report using ABAP CDS annotation

NEXT UNIT

- Week 2 – Unit 1
  Creating an object page

Thank you.

Contact information:

open@sap.com

Follow all of SAP

www.sap.com/contactsap

THE BEST RUN SAP®