

# Hands-on Lab: Fusion Development with ABAP Cloud in SAP Build



In partnership with **SAP TechEd**

Anderson Silva, EPI-USE  
Pedro Vieira, EPI-USE  
João Rocha, EPI-USE  
March 2026

# Agenda

## Exercise 1

- *Start your development by creating an ABAP project in the SAP Build Lobby*

## Exercise 2

- *Create a Shopping Cart business object, projection and service with the Graphical Modeler in SAP Build Code*

## Exercise 3

- *Implement transactional behavior in ADT, create validations and determinations*

## Exercise 4

- *Create, preview and adapt a SAP Fiori elements application*

# Get started



In partnership with **SAP TechEd**

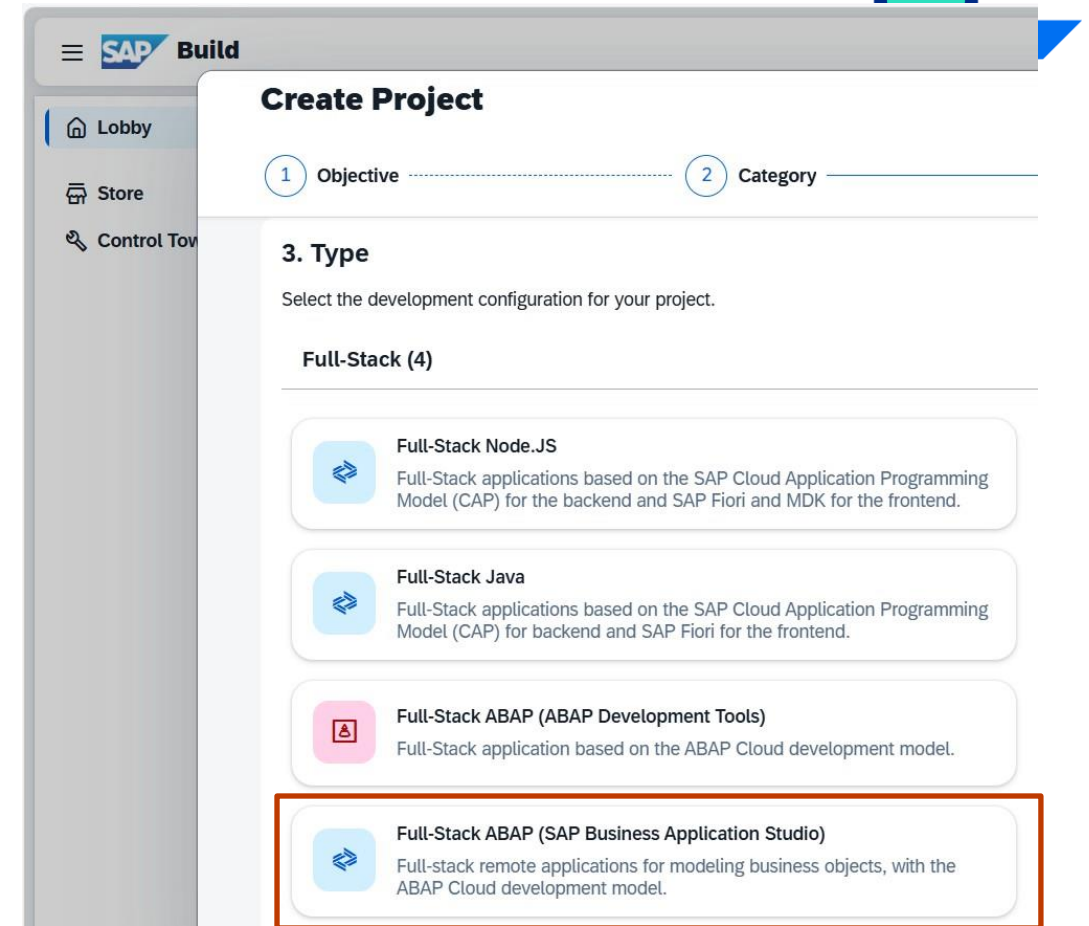
# Introduction



In partnership with **SAP TechEd**

# What you will do in this hands-on workshop – Practice fusion development of ABAP Cloud

1. SAP Build Lobby - Create a Full-Stack ABAP project (SAP Business Application Studio)



# What you will do in this hands-on workshop – Practice fusion development of ABAP Cloud

1. SAP Build Lobby - Create a Full-Stack ABAP project (SAP Business Application Studio)
2. SAP Business Application Studio – Graphical Modeler



The screenshot displays the SAP Business Application Studio Graphical Modeler interface. It shows two data models side-by-side:

- Cart Business Object:** A table with 6 rows and 2 columns. The first column lists attributes, and the second column lists their corresponding ABAP data types.
- Item Entity:** A table with 10 rows and 3 columns. The first column lists attributes, the second column lists their corresponding ABAP data types, and the third column lists their corresponding SAP data types.

The interface includes a search bar at the top, a list of attributes on the left, and a table of attributes on the right. The status bar at the bottom indicates 'Layout: German'.

Cart Business Object	
OrderUid	abap.raw(16)
Ab Currency	abap.cuky
Ab Notes	abap.char(100)
Ab OrderID	abap.numc(8)
RequestDeliveryDate	abap.datt
12 TotalPrice	abap.curr(11,2)

Item Entity		
ItemUid	abap.raw(16)	
10 01 ParentUid	abap.raw(16)	
Ab ItemID	abap.numc(8)	
Ab OrderedItem	abap.char(40)	
Ab Quantity	abap.numc(4)	
12 ItemPrice	abap.curr(11,2)	
Ab Currency	abap.cuky	
12 ItemUnitPrice	abap.curr(11,2)	

# What you will do in this hands-on workshop – Practice fusion development of ABAP Cloud

1. SAP Build Lobby - Create a Full-Stack ABAP project (SAP Business Application Studio)
2. SAP Business Application Studio – Graphical Modeler
3. ABAP Development Tools - Implementation of Business Logic



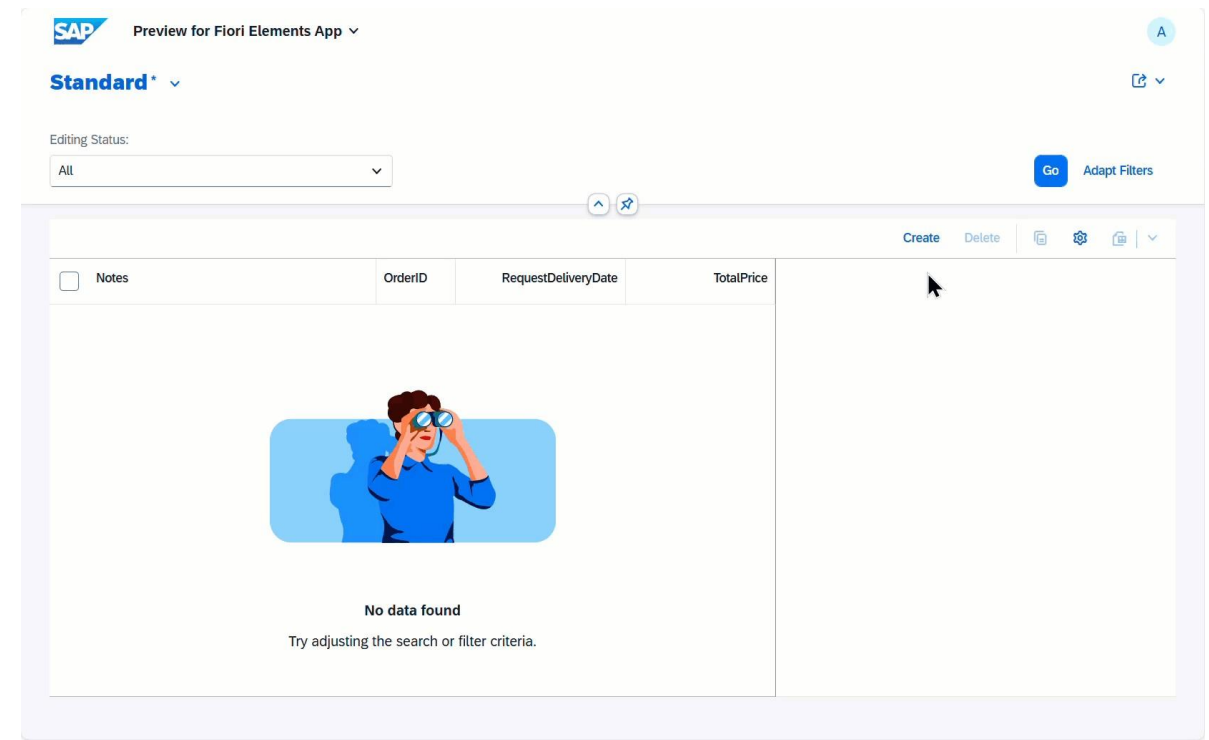
```
35  
36 validation validateRequestDeliveryDate on save { create; field RequestDeliveryDate; }  
37  
38 draft action Activate optimized;  
39 draft action Discard;  
40 draft action Edit;  
41 draft action Resume;  
42 //draft determine action Prepare;  
43 draft determine action Prepare  
44 {  
45     validation validateRequestDeliveryDate;  
46 }
```

```
METHOD validateRequestDeliveryDate.  
  
ENDMETHOD.
```

Predict RAP Business Logic

# What you will do in this hands-on workshop – Practice fusion development of ABAP Cloud

1. SAP Build Lobby - Create a Full-Stack ABAP project (SAP Business Application Studio)
2. SAP Business Application Studio – Graphical Modeler
3. ABAP Development Tools - Implementation of Business Logic
4. SAP Business Application Studio - Create, preview and adapt a SAP Fiori elements application



# **Exercise 1: Start your development by creating an ABAP project in the SAP Build Lobby**



In partnership with **SAP TechEd**

# Exercise 1: Create a Full-Stack ABAP (SAP Business Application Studio) Project

Create a Full-Stack ABAP (SAP Business Application Studio) Project

Connect to backend system

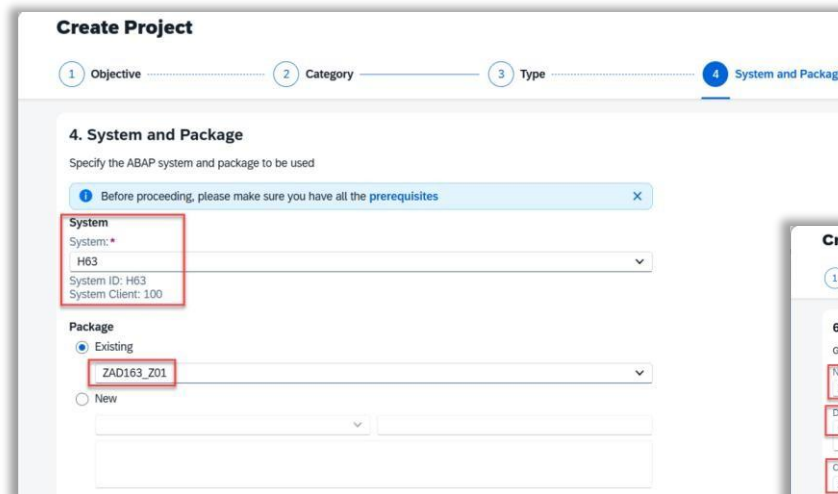
**H63**

Use a pre-created package

**ZAD163-A##**

Create and start Dev Space  
in SAP Business Application Studio

 **Full-Stack ABAP (SAP Business Application Studio)**  
Full-stack remote applications for modeling business objects, with the ABAP Cloud development model.



**Create Project**

1 Objective 2 Category 3 Type 4 System and Package

**4. System and Package**

Specify the ABAP system and package to be used

Before proceeding, please make sure you have all the prerequisites

System

System: \*  
H63

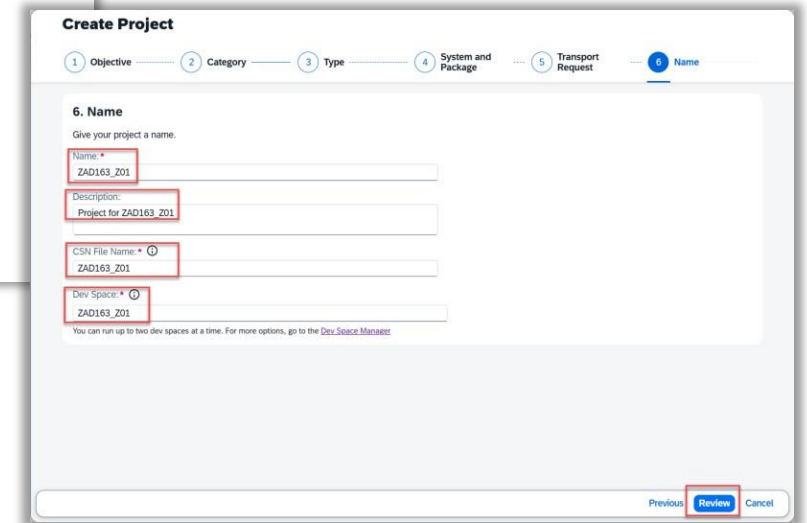
System ID: H63  
System Client: 100

Package

Existing

ZAD163\_Z01

New



**Create Project**

1 Objective 2 Category 3 Type 4 System and Package 5 Transport Request 6 Name

**6. Name**

Give your project a name.

Name: \*  
ZAD163\_Z01

Description:  
Project for ZAD163\_Z01

CSN File Name: \*  
ZAD163\_Z01

Dev Space: \*  
ZAD163\_Z01

You can run up to two dev spaces at a time. For more options, go to the [Dev Space Manager](#).

Previous Review Cancel

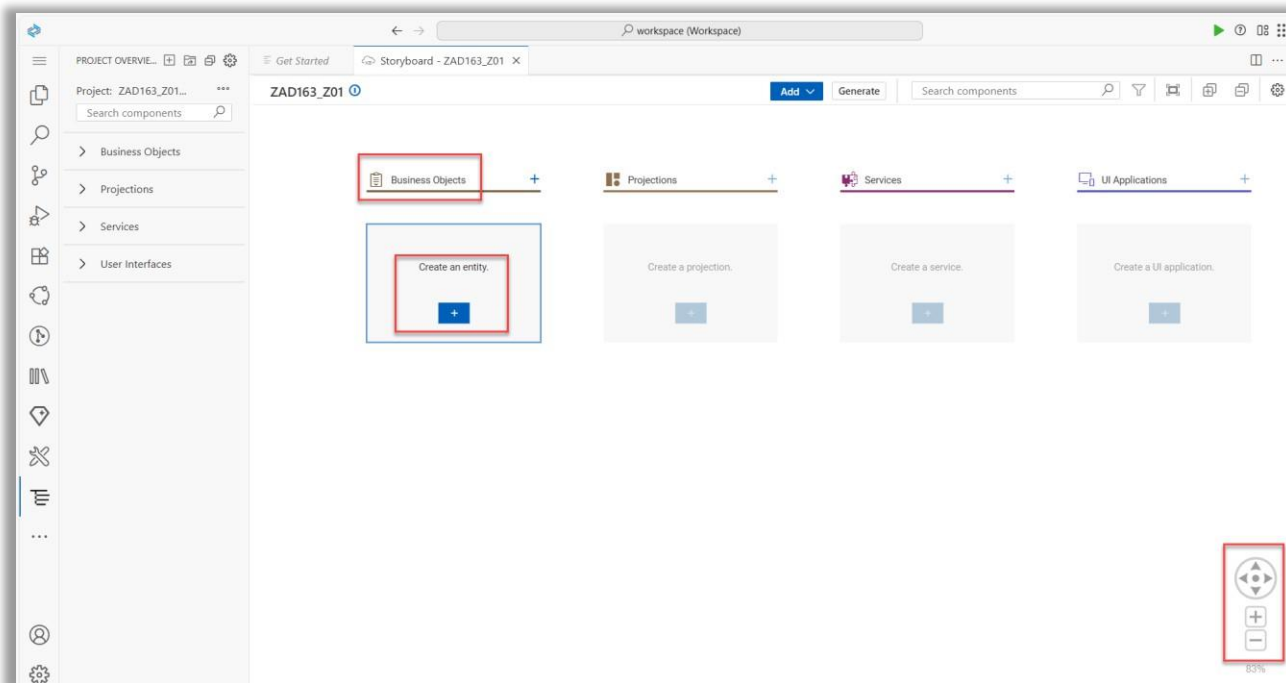
**Exercise 2 :  
Create a Shopping  
Cart business  
object,  
projection and  
service with the  
Graphical Modeler  
in SAP Build Code**



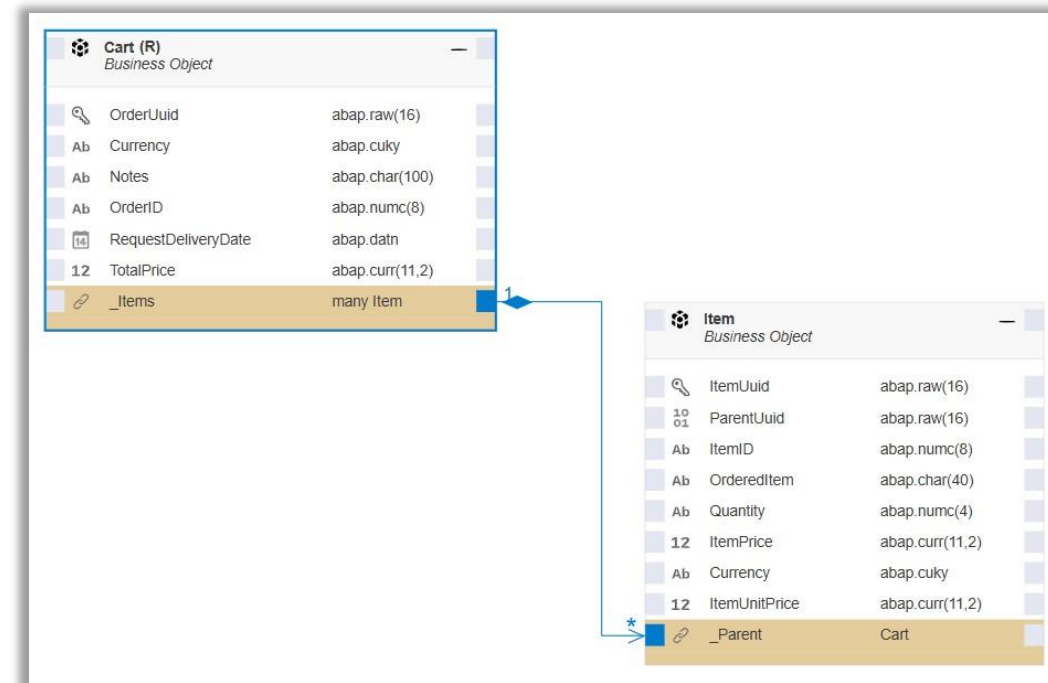
In partnership with **SAP TechEd**

# Exercise 2: Model the ShoppingCart Business Object

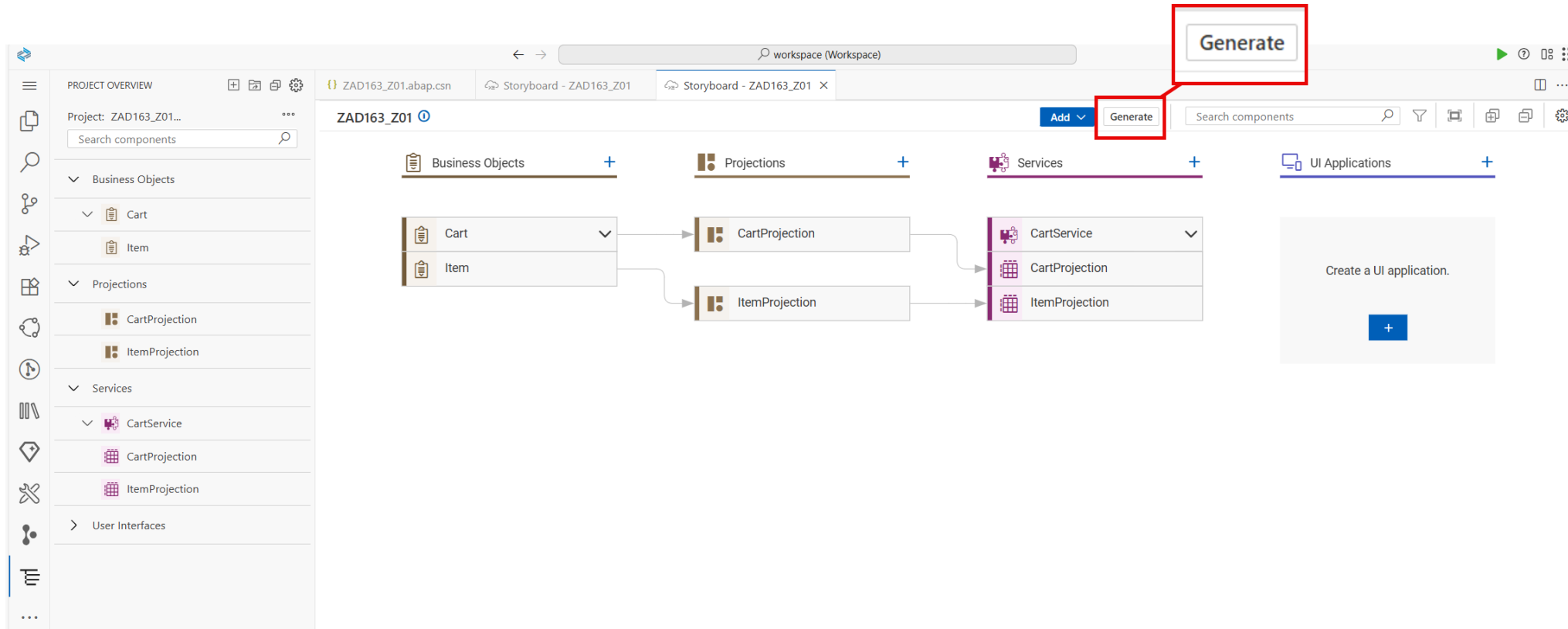
Storyboard



Graphical Modeller



# Exercise 2: Add Projection and Service Layer, Generate ABAP artefacts



# **Exercise 3: Implement transactional behavior in ADT, create validations and determinations**



In partnership with **SAP TechEd**

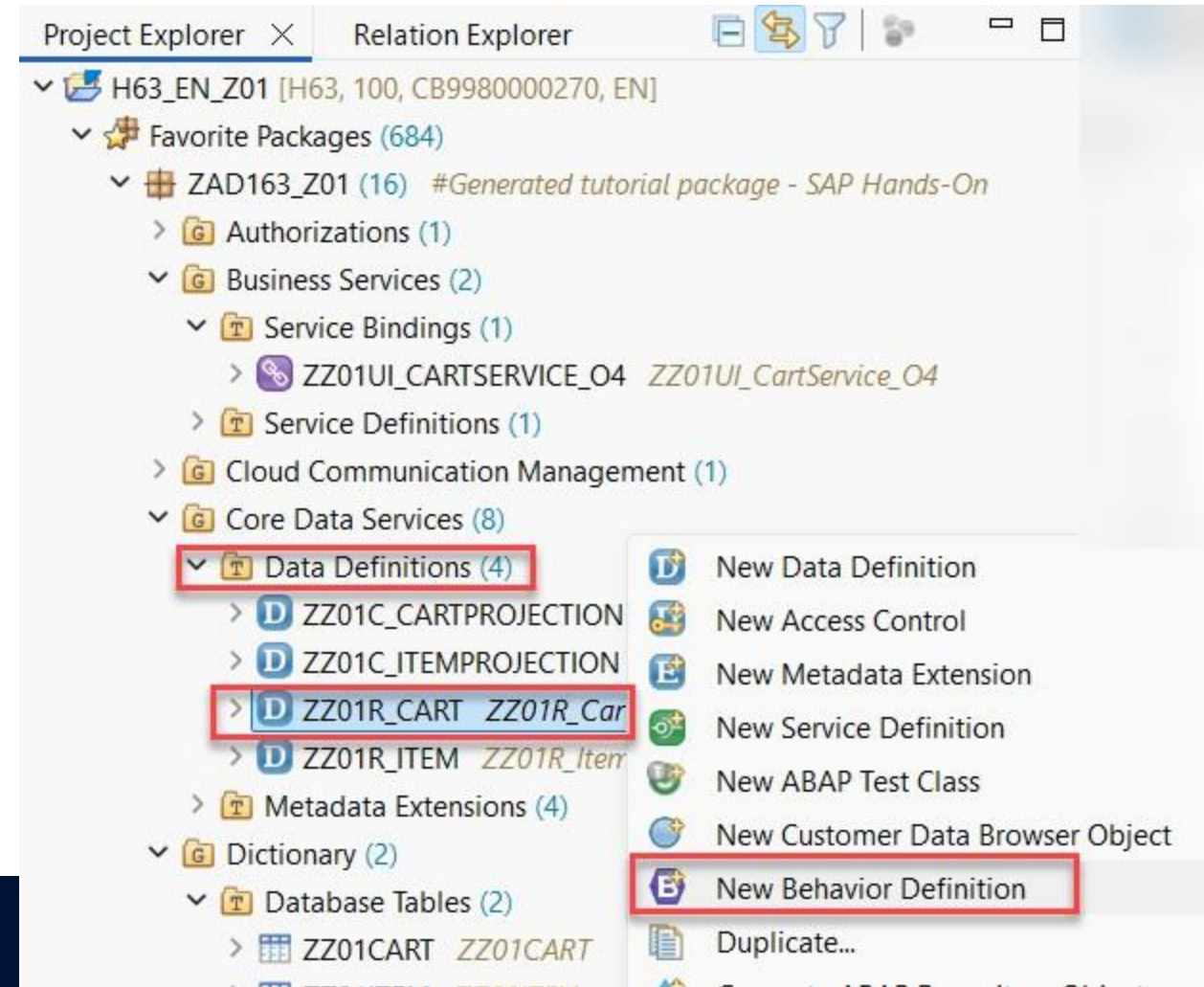
# Exercise 3: Create ABAP cloud project in ADT

The screenshot displays the SAP ADT workspace interface. On the left, the Explorer pane shows the project structure for 'ZAD163\_Z01', with 'Service Bindings' expanded and 'ZZ01UI\_CARTSERVICE\_O4.abap.srvb' selected. The main editor area shows an error message: 'Unable to open the "/H63/ZAD163\_Z01/Service Bindings/ZZ01UI\_CARTSERVICE\_O4.abap.srvb" file.' Below the error is a link 'Open this object in ADT'. Two dialog boxes are overlaid. The 'Open ADT Link' dialog has a 'Choose a Project:' field and a 'New...' button. The 'New Project' dialog shows 'Select a wizard' with 'Create an ABAP cloud project' selected. Under 'Wizards', 'ABAP Cloud Project' is highlighted. The 'Next >' button is visible at the bottom right.

1 – Create ABAP cloud project in ADT

# Exercise 3: Add transactional behavior

```
1 @EndUserText.label : 'ZZ01CART'
2 @AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
3 @AbapCatalog.tableCategory : #TRANSPARENT
4 @AbapCatalog.deliveryClass : #A
5 @AbapCatalog.dataMaintenance : #ALLOWED
6 define table zz01cart {
7
8   key client      : abap.clnt not null;
9   key order_uuid  : abap.raw(16) not null;
10  currency        : abap.cuky;
11  notes           : abap.char(100);
12  order_id        : abap.numc(8);
13  request_delivery_date : abap.datn;
14  @Semantics.amount.currencyCode : 'zz01cart.currency'
15  total_price     : abap.curr(11,2);
16  local_created_by : abp_creation_user;
17  local_created_at : abp_creation_tstmpl;
18  local_last_changed_by : abp_locinst_lastchange_user;
19  local_last_changed_at : abp_locinst_lastchange_tstmpl;
20  last_changed_at   : abp_lastchange_tstmpl;
21
22 }
```



# Exercise 3: Implement determinations and validations using Joule

```
35  
36 validation validateRequestDeliveryDate on save { create; field RequestDeliveryDate; }
```


```
37  
38 draft action Activate optimized;  
39 draft action Discard;  
40 draft action Edit;  
41 draft action Resume;
```

```
42 //draft determine action Prepare;  
43 draft determine action Prepare  
44 {  
45     validation validateRequestDeliveryDate;  
46 }
```

```
64  
65 association _Items { create; with draft; }  
66  
67 }
```

```
METHOD validateRequestDeliveryDate.
```

```
ENDMETHOD.
```

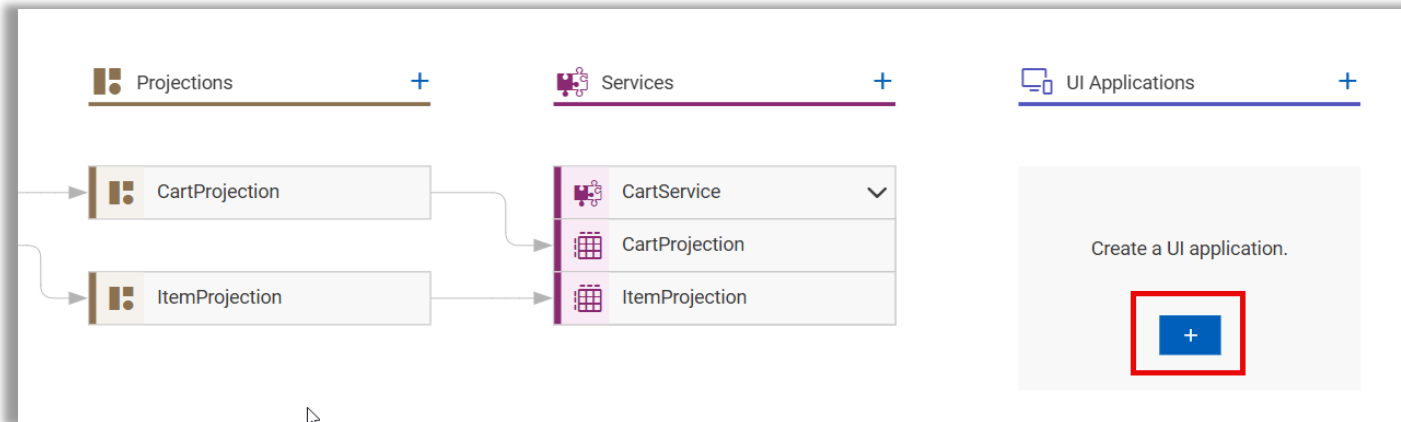
 Predict RAP Business Logic

# **Exercise 4: Create, preview and adapt a SAP Fiori elements application**



In partnership with **SAP TechEd**

# Exercise 4: Create, preview and adapt a SAP Fiori elements application



The screenshot shows the SAP Fiori generator configuration screen. The left sidebar contains the following sections:

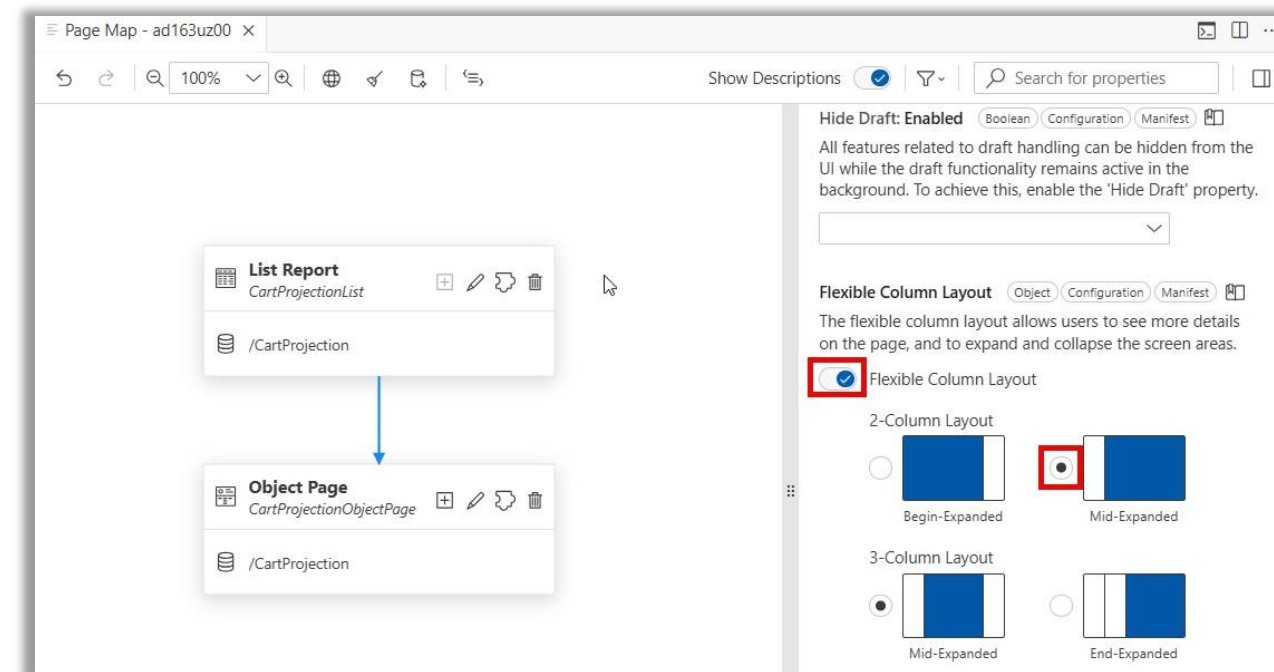
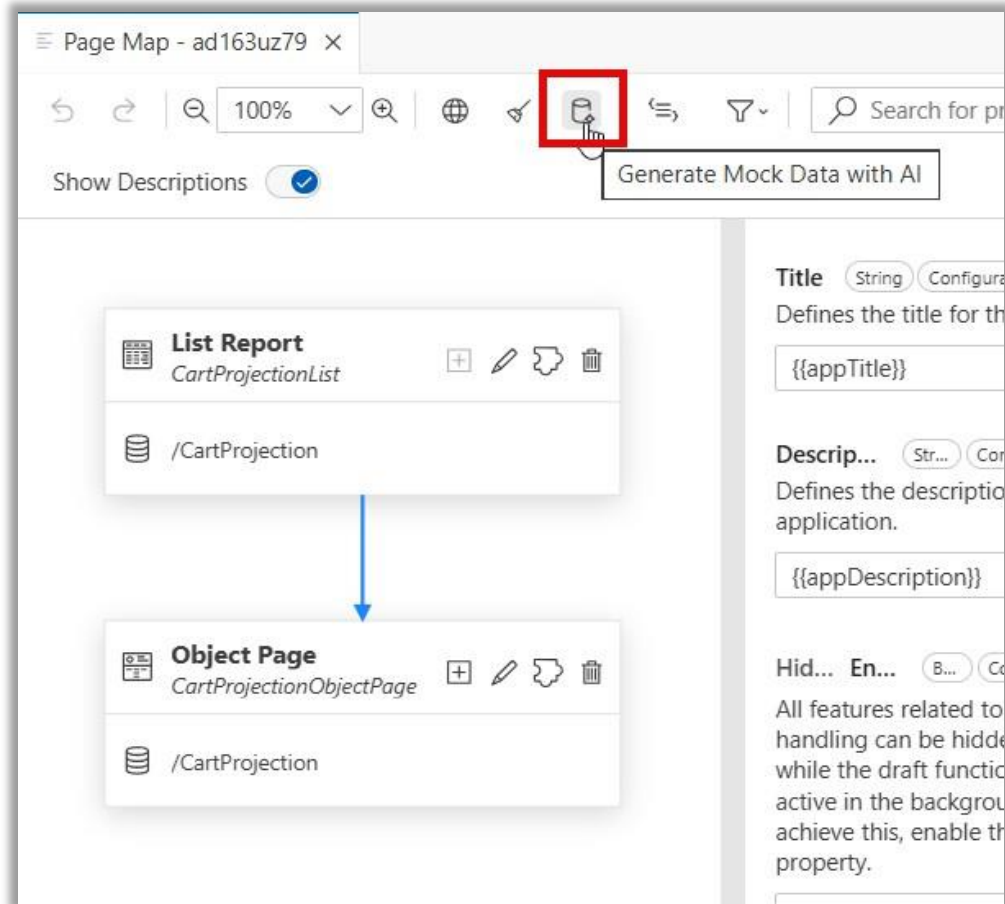
- Template Selection:** Template: List Report Page
- Data Source and Service Selection:** Data Source: Connect to a System; System: H63 (BTP); Service: ZZ00UI\_CARTSERVICE\_O4 > ZZ00UI\_CARTSERVICE\_O4 (001) - OData V4
- Entity Selection:** Main Entity: CartProjection; Navigation Entity: None; Generate Annotations: Yes; Table Type: Responsive
- Project Attributes:** Module Name: ad163uz00; Application Title: Shopping Cart App AD163UZ00; Application Namespace: /home/user/projects; Description: A Fiori application.

The right sidebar contains the following sections:

- Project Attributes:** Configure the main project attributes.
- Module Name:** ad163uz00 (highlighted with a red rectangle)
- Application Title:** Shopping Cart App AD163UZ00
- Application Namespace:** /home/user/projects
- Description:** A Fiori application.
- Project Folder Path:** /home/user/projects
- Minimum SAPUI5 Version:** 1.136.8 (Source system version)
- Enable TypeScript:** Yes (selected), No
- Add Deployment Configuration:** Yes (selected), No (highlighted with a red rectangle)
- Add SAP Fiori Launchpad Configuration:** Yes (selected), No
- Use Virtual Endpoints for Local Preview:** Yes (selected), No
- Configure Advanced Options:** Yes (selected), No

At the bottom, there are two buttons: "< Back" and "Finish" (highlighted with a red rectangle).

# Exercise 4: Create, preview and adapt a SAP Fiori elements application



**Let's start with  
the Hands-on Lab**



In partnership with **SAP TechEd**

# Thank you!



In partnership with **SAP TechEd**