

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**

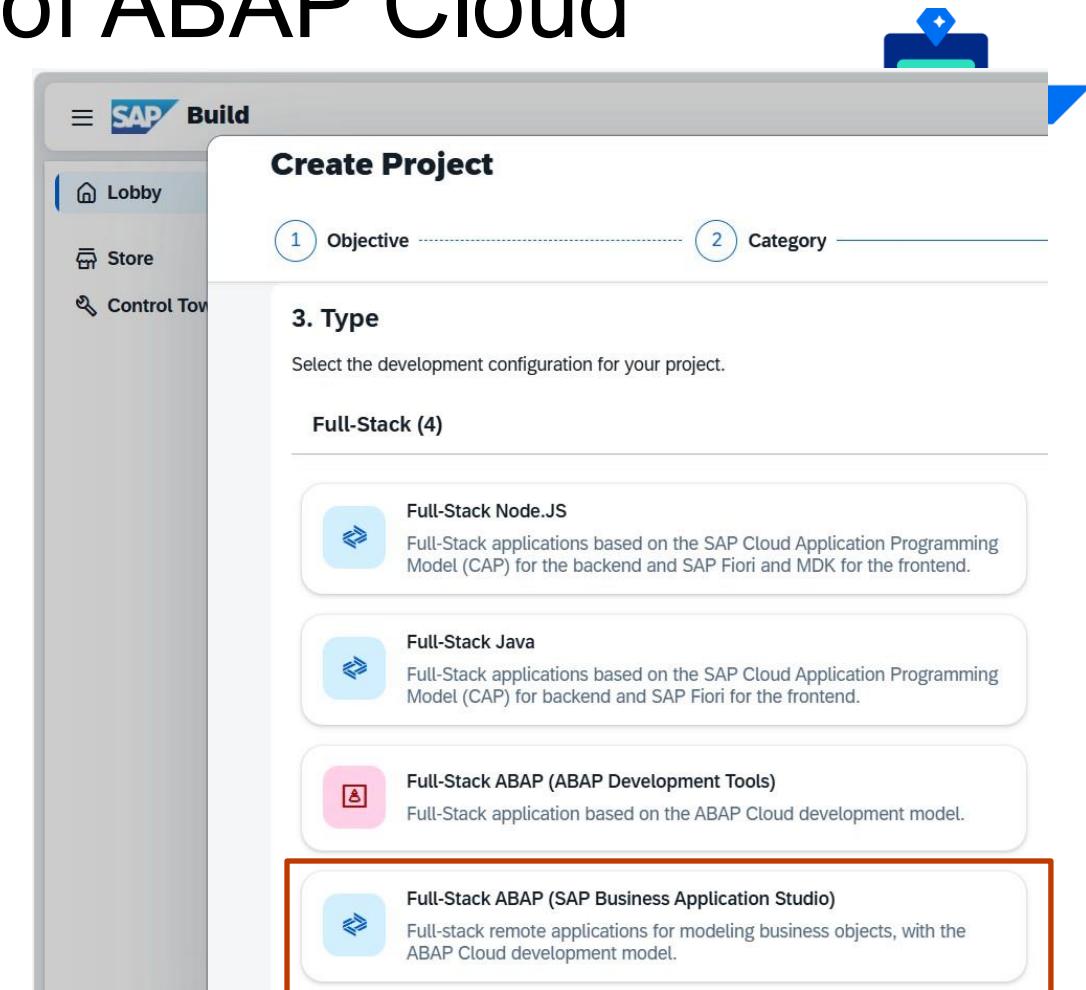
Introduction



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 shows the SAP Business Application Studio Graphical Modeler interface. Two entity definitions are displayed side-by-side:

Cart Business Object

| OrderUuid | abap.raw(16) | |
|---------------------|-----------------|--|
| Currency | abap.cuky | |
| Notes | abap.char(100) | |
| OrderID | abap.numc(8) | |
| RequestDeliveryDate | abap.datn | |
| TotalPrice | abap.curr(11,2) | |

Item Entity

| ItemUuid | abap.raw(16) | |
|---------------|-----------------|--|
| ParentUuid | abap.raw(16) | |
| ItemID | abap.numc(8) | |
| OrderedItem | abap.char(40) | |
| Quantity | abap.numc(4) | |
| ItemPrice | abap.curr(11,2) | |
| Currency | abap.cuky | |
| ItemUnitPrice | abap.curr(11,2) | |

The interface includes a toolbar with icons for zooming, search, and other functions, and a status bar at the bottom right indicating "Layout: German".

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 }
```

A screenshot of an ABAP code editor. The code shown is validation logic for a 'validateRequestDeliveryDate' method. Lines 36 through 46 are highlighted with a red box. The code includes several 'draft' actions and a block of code starting with 'draft determine action Prepare'. The code editor has a light gray background with syntax highlighting for keywords and comments.

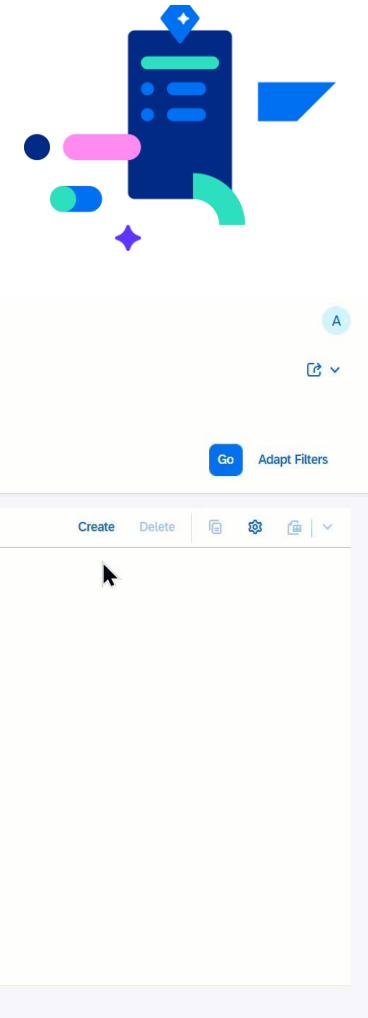
```
METHOD validateRequestDeliveryDate.
ENDMETHOD.
```

Predict RAP Business Logic

A screenshot of the SAP Business Application Studio interface. A tooltip labeled 'Predict RAP Business Logic' is displayed over a portion of the code. The code itself shows a single METHOD definition for 'validateRequestDeliveryDate'.

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

The image shows the SAP Business Application Studio 'Create Project' wizard with two visible steps:

- Step 4: System and Package**:
 - System: H63
 - Package: ZAD163_Z01
- Step 6: Name**:
 - Name: ZAD163_201
 - Description: Project for ZAD163_201
 - CSN File Name: ZAD163_201
 - Dev Space: ZAD163_201

Both the 'System and Package' and 'Name' fields are highlighted with red boxes.

Exercise 2 :

Create a Shopping

Cart business

object,

projection and

service with the

Graphical Modeler

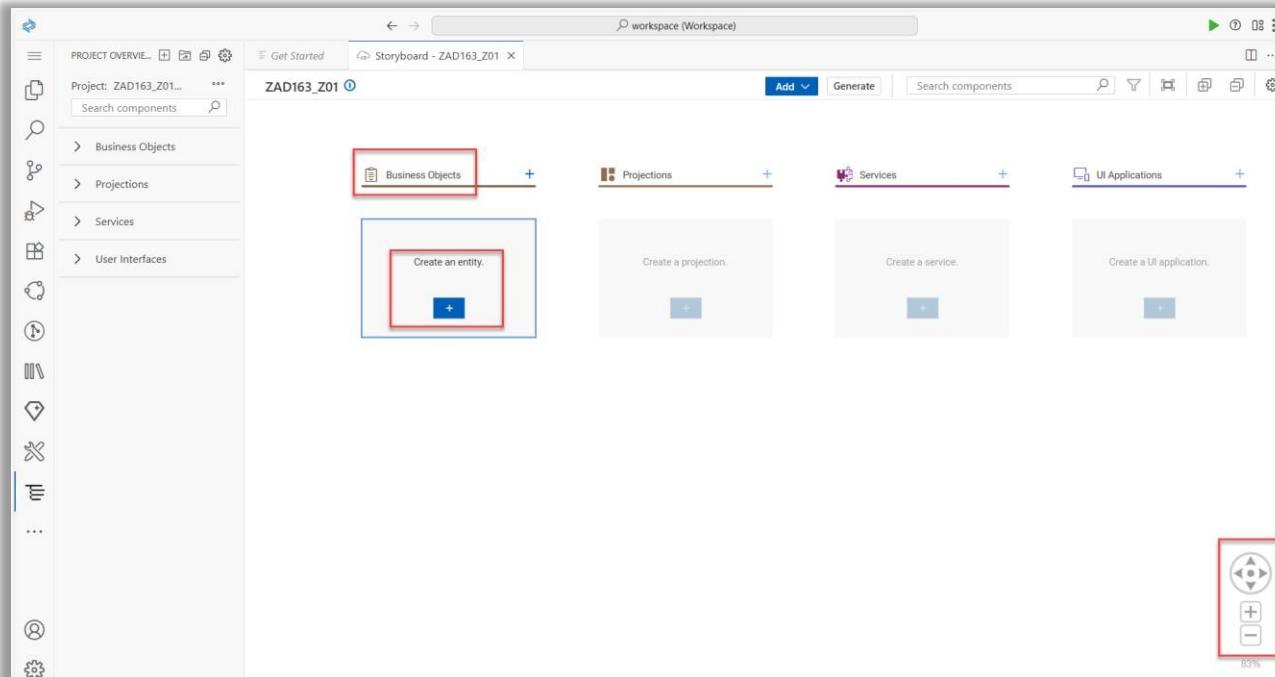
in SAP Build Code



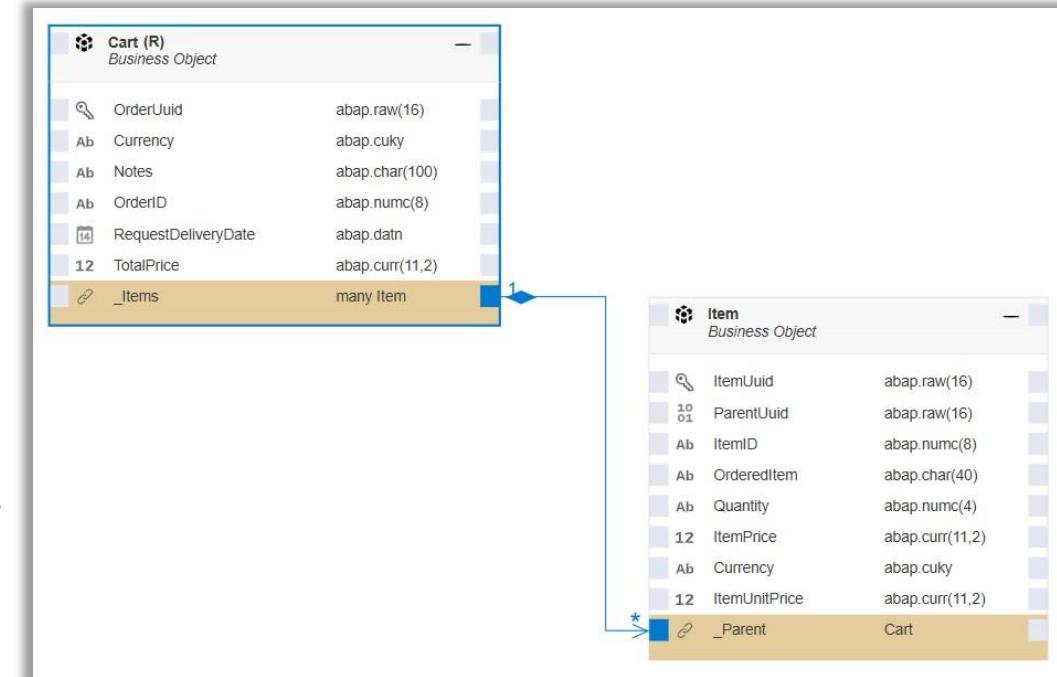
In partnership with **SAP TechEd**

Exercise 2: Model the ShoppingCart Business Object

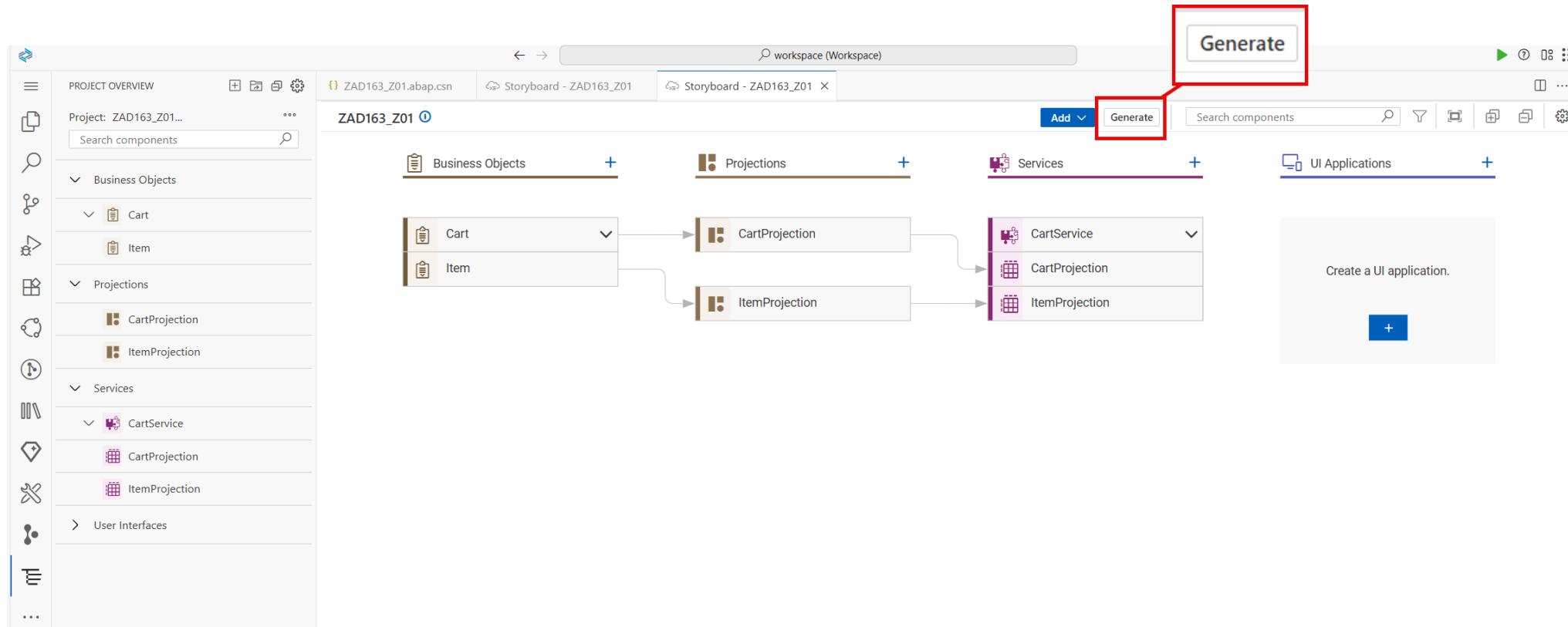
Storyboard



Graphical Modeller



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

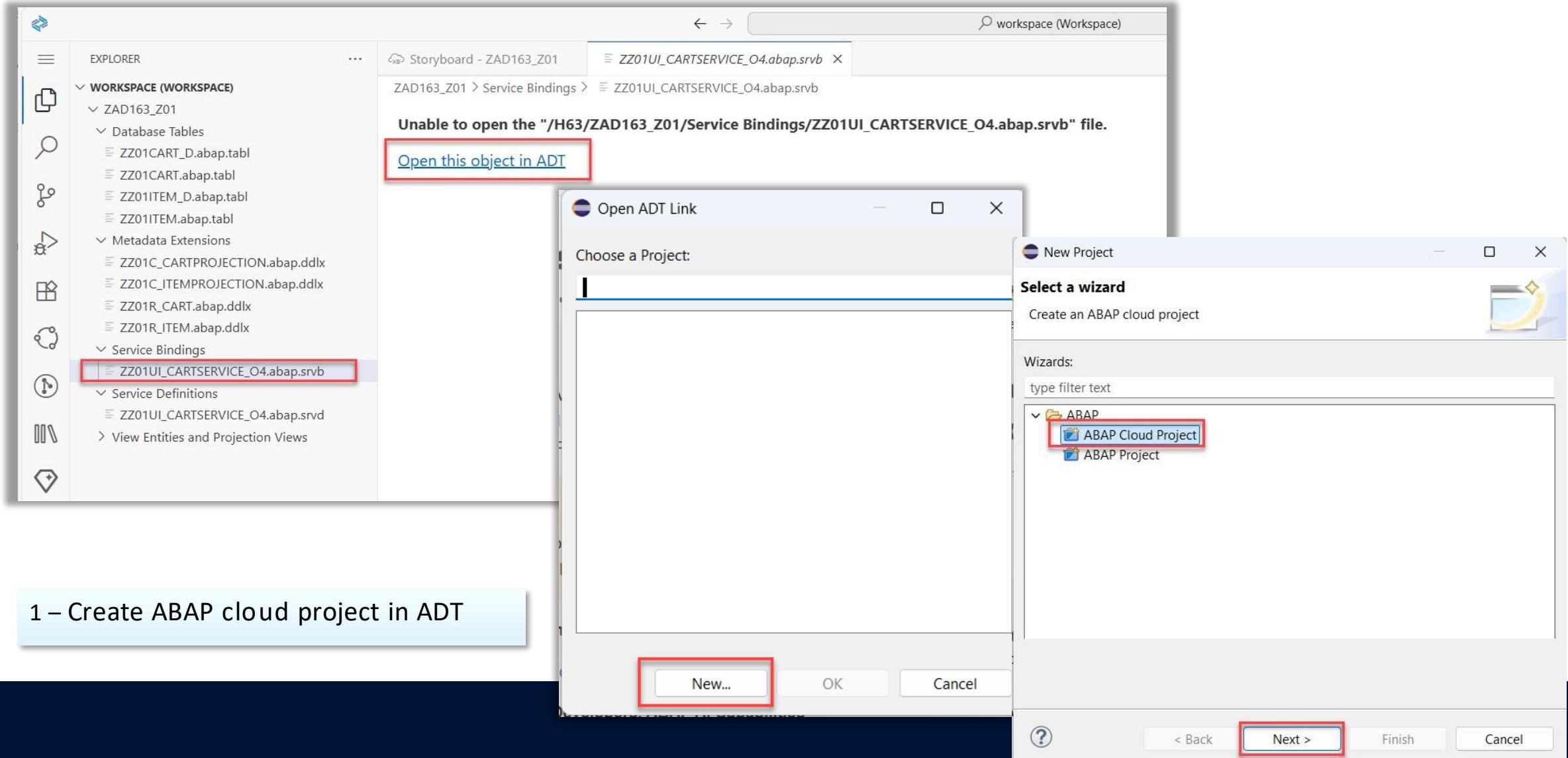


Exercise 3: Implement transactional behavior in ADT, create validations and determinations



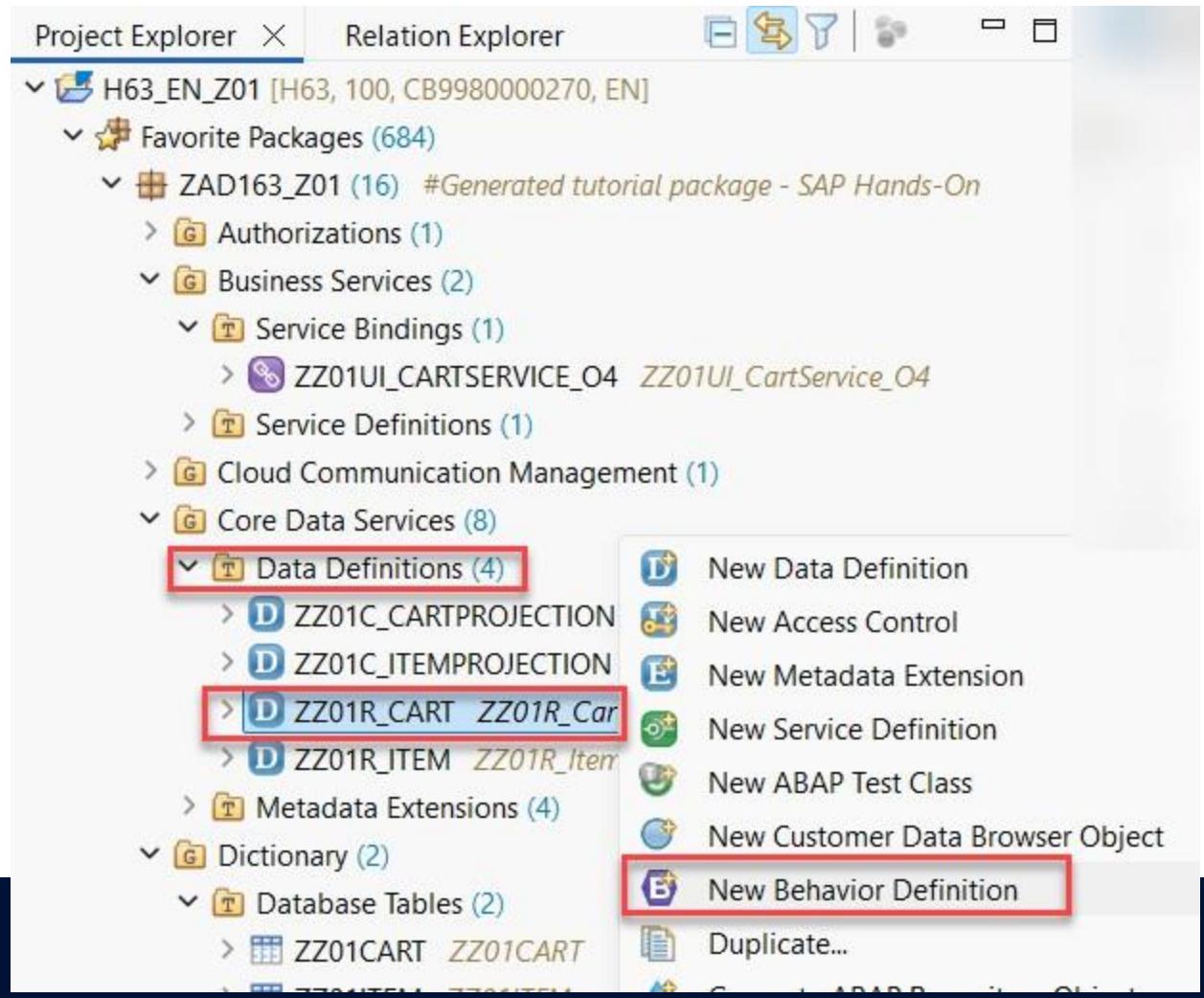
In partnership with **SAP TechEd**

Exercise 3: Create ABAP cloud project in ADT



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.cln1 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_tstmp;
18  local_last_changed_by : abp_locinst_lastchange_user;
19  local_last_changed_at : abp_locinst_lastchange_tstmp;
20  last_changed_at    : abp_lastchange_tstmp;
21
22 }
```



Exercise 3: Implement determinations and validations using Joule

The screenshot shows a code editor interface for SAP Joule. The code is written in a simplified RAP Business Logic syntax.

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

A red box highlights the validation block at lines 36-37. Another red box highlights the draft determine action at lines 42-45. The code editor has a wavy grey border at the bottom.

Below the code editor, a modal dialog is open:

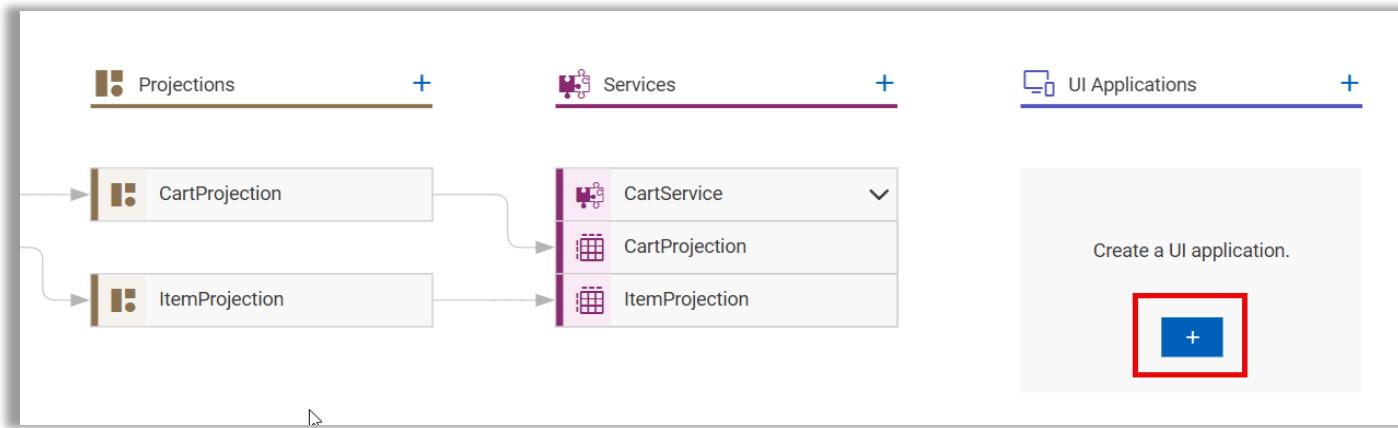
- METHOD** validateRequestDeliveryDate.
- ENDMETHOD.**
- Predict RAP Business Logic** button.

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



SAP Fiori generator ⓘ

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:
Description: A Fiori application.
Project Folder Path: /home/user/projects
Minimum SAPUI5 Version: 1.136.8 (Source system version)
Enable TypeScript: No
Add Deployment Configuration: No
Add SAP Fiori Launchpad Configuration: No
Use Virtual Endpoints for Local Preview: Yes
Configure Advanced Options: No

[Back](#) [Finish](#)

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

The screenshot shows the SAP Fiori Elements Page Map interface. At the top, there's a toolbar with various icons and a search bar. A red box highlights the 'Generate Mock Data with AI' button, which has a hand cursor icon over it. Below the toolbar, there's a 'Show Descriptions' toggle switch that is turned on. The main area displays two components: a 'List Report' and an 'Object Page'. Each component has edit, copy, and delete icons. To the right of each component, there are detailed configuration descriptions. A blue arrow points from the 'List Report' section down to the 'Object Page' section.

The screenshot shows the SAP Fiori Elements Page Map interface. On the right side, there are several configuration sections. One section is titled 'Hide Draft: Enabled' with a description explaining that it hides features related to draft handling while keeping the draft functionality active. Another section is titled 'Flexible Column Layout' with a description explaining that it allows users to see more details and expand/collapse screen areas. A blue box highlights the 'Flexible Column Layout' checkbox, which is checked. Below this, there are examples for '2-Column Layout' and '3-Column Layout', each with 'Begin-Expanded' and 'Mid-Expanded' options. A blue arrow points from the 'Object Page' section down to the 'Flexible Column Layout' section.

Let's start with
the Hands-on Lab



In partnership with SAP TechEd

Thank you!



In partnership with **SAP TechEd**