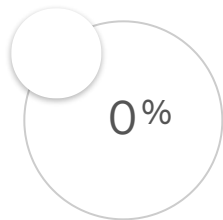
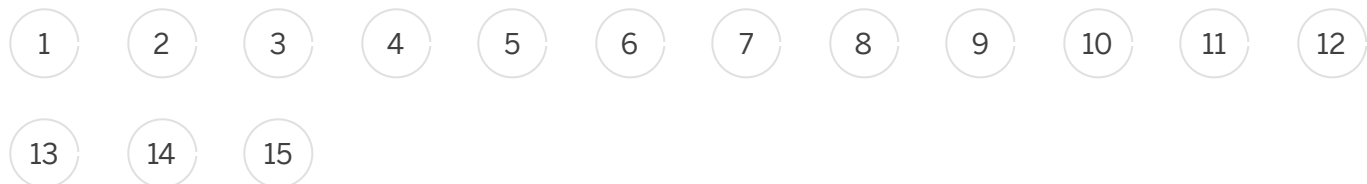


Group /
Tutorial



Create an ABAP Core Data Services (CDS) View in SAP Cloud Platform, ABAP Environment



Details

Code Snippets



Create an ABAP Core Data Services (CDS) View in SAP Cloud Platform, ABAP Environment



Julie Plummer January 18, 2021

Created by



July 9, 2020

Beginner 45

 [ABAP Development, Tutorial,](#)

Series that include this tutorial:

[GROUP: Create and Enhance a CDS View in SAP Cloud Platform, ABAP Environment](#)

- 1 Step 1: Create package
- 2 Step 2: Create CDS View
- 3 Step 3: Define CDS View
- 4 Step 4: Display in Data Preview
- 5 Step 5: Create a service definition [</>](#)
- 6 Step 6: Create service binding

min. [SAP BTP, ABAP environment, Beginner](#)

Create a CDS View, display it in Fiori Elements preview, and enhance its appearances using built-in annotations

You will learn

- ✓ How to create a CDS view with parameters
- ✓ How to display your CDS view in a Fiori Elements preview
- ✓ How to add selection fields to Fiori Elements preview
- ✓ How to extract the metadata of your CDS view
- ✓ How to add semantic annotations
- ✓ How to add a search function
- ✓ How to add selection fields

 [Provide Feedback](#)

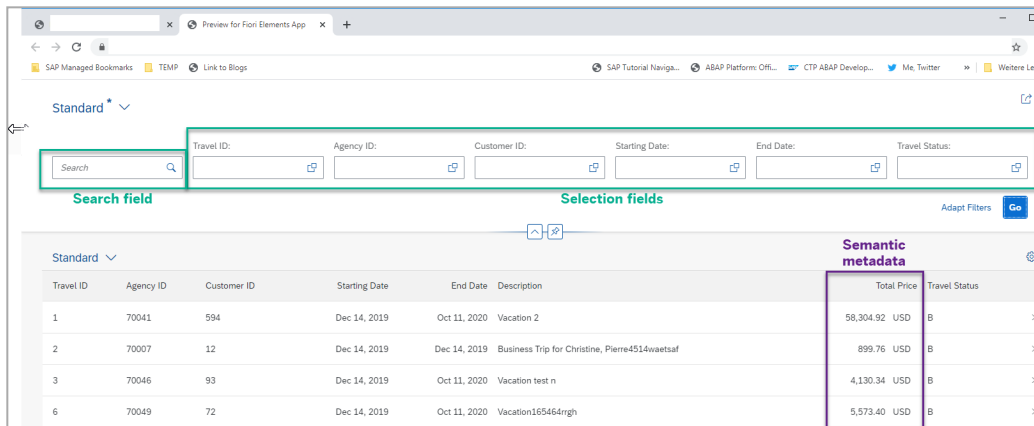
- 7 [Step 7: Activate service binding](#)
- 8 [Step 8: Display Fiori Elements Preview](#)
- 9 [Step 9: Add annotation for automatic display](#) [</>](#)
- 10 [Step 10: Extract UI metadata](#)
- 11 [Step 11: Add semantic metadata](#) [</>](#)
- 12 [Step 12: Add search field](#) [</>](#)
- 13 [Step 13: Add selection fields](#) [</>](#)
- 14 [Step 14: Check your code](#) [</>](#)
- 15 [Step 15: Test yourself](#)
- [^ Back to Top](#)

 [Provide Feedback](#)

Prerequisites

- You have done one of the following:
 - **Tutorial:** [Create an SAP Cloud Platform ABAP Environment Trial User](#)
 - You have a licensed version of SAP Cloud Platform, ABAP Environment
- You have installed [ABAP Development Tools](#), latest version
- You have downloaded the ABAP Flight Reference Scenario. To pull this reference scenario from Github, see [Downloading the ABAP Flight Reference Scenario](#)

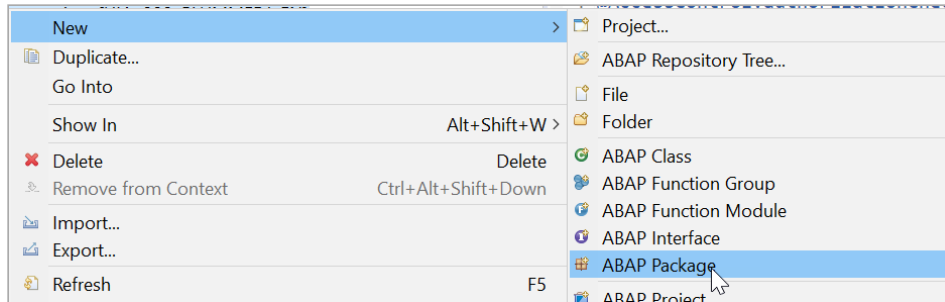
You can then use some of these features in productive development to make your applications more powerful and more user-friendly. By the end of this tutorial, your application should look like this.



Throughout this tutorial, object names may include a suffix or group number, such as **xxx**. Always replace this with your own group number or initials.

1 Step 1: Create package

1. Create a new package for this tutorial, by choosing **New > ABAP Package**.



2. Enter a name **Package Z_ENHANCE_CDS_XXX** and description **Enhance CDS Tutorial 2020**, then follow the wizard.

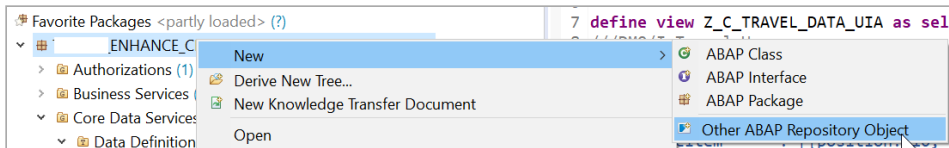
A screenshot of the 'New ABAP Package' wizard. The wizard is titled 'New ABAP Package' and has a subtitle 'Create an ABAP package'. The 'Name' field is filled with 'Z_ENHANCE_CDS_XXX' and the 'Description' field is filled with 'Enhance CDS Tutorial 2020'. The 'Original Language' is set to 'EN'. The 'Package Type' is set to 'Development'. The 'Next >' button is highlighted with a red box. Other buttons visible include '< Back', 'Finish', and 'Cancel'.

Done

Log on to answer question

Step 2: Create CDS View

1. In your package, create a CDS view. Select the package, then choose **New > Other** from the context menu, then choose **Data Definition**.

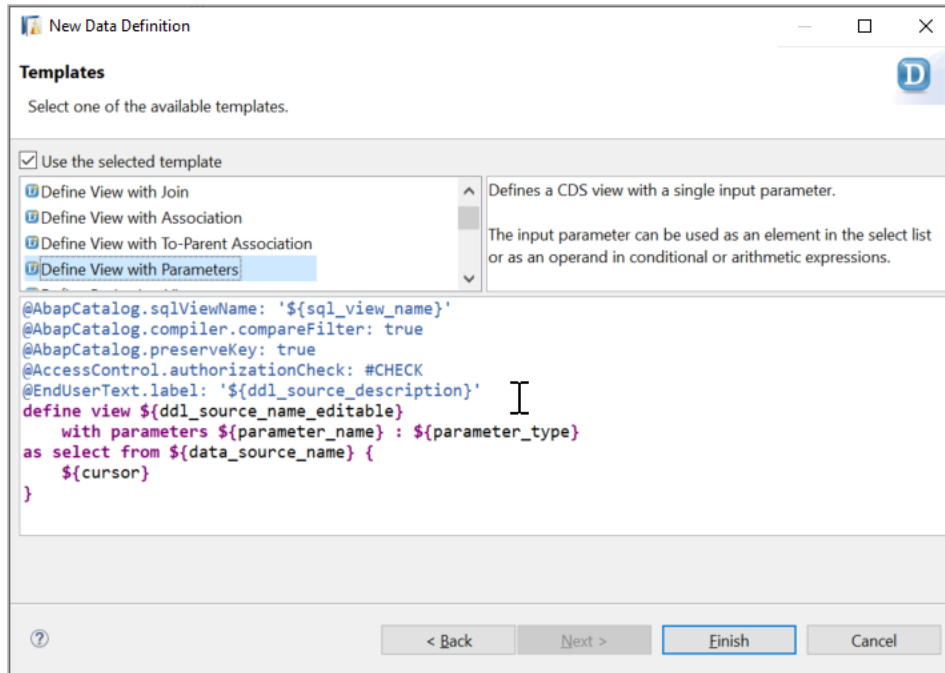


2. Add a name, `Z_C_TRAVEL_DATA_XXX`, and description `Consumption` view from `/DMO/I_TRAVEL_U`.

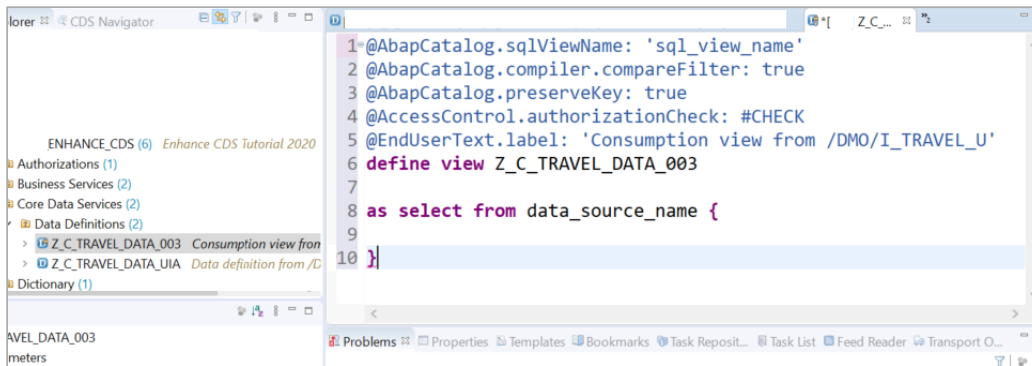
Your CDS view is a consumption view. It is based on the business object (BO) view, `/DMO/I_TRAVEL_U`, which view provides a given data model independent of the consumption layer. It contains all core information required by applications running on top of it.

A consumption view is a CDS view defined on top of a BO view and is used:

- to expose the fields fitting to a given consumption use case
 - to enrich the data model with metadata using annotations (e.g. for UI, search, and OData)
3. Choose or create a transport request, then choose **Next**. Do not choose **Finish**.
 4. Finally, choose **Use template** then choose **Define view**. Then choose **Finish**.



Your CDS view appears in a new editor.



Done

Log on to answer question

Step 3: Define CDS View

1. Add the following:

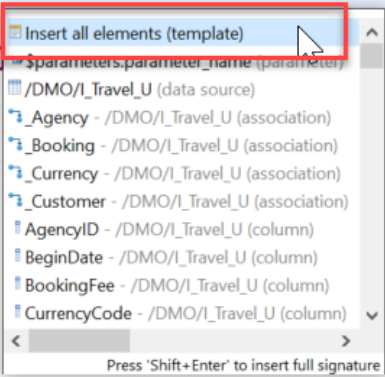
- `sql_view_name` = `ZCTRAVEL_XXX`
- `data_source_name` = `/DMO/I_Travel_U`. You can use **Auto-Complete** `Ctrl+Space`

2. Insert all the elements from `/DMO/I_TRAVEL_U` by placing your cursor inside the `as select from` statement (curly brackets) and again choosing **Auto-Complete** `Ctrl+Space`.

```

1 @AbapCatalog.sqlViewName: 'ZCTRAVEL_003'
2 @AbapCatalog.compiler.compareFilter: true
3 @AbapCatalog.preserveKey: true
4 @AccessControl.authorizationCheck: #CHECK
5 @EndUserText.label: 'Consumption view from /DMO/I_TRAVEL_U'
6 define view Z_C_TRAVEL_DATA_003
7   with parameters parameter_name : parameter_type
8   as select from /DMO/I_Travel_U {
9
10
11
12 }

```



Press 'Shift+Enter' to insert full signature

3. Comment out the statement `with parameters parameter_name : parameter_type` for now, so that the error disappears.
4. Format, save, and activate your code by choosing `Shift+F1`, `Ctrl+S`, `Ctrl+3`.

 Done

Log on to answer question

4 Step 4: Display in Data Preview

1. Click anywhere in the editor and choose **Open With > Data Preview** from the context menu.



2. The Data Preview is displayed in a new tab. You can investigate the data, by filtering, specifying the number of rows, and so on.

The values in `LastChangedAt` are not user-friendly, but you solve that by providing a Fiori elements preview in the next step.

TravelID	AgencyID	CustomerID	BeginDate	EndDate	BookingFee	TotalPrice	Cu...	Memo	Status	LastChangedAt
00000001	070011	00000001	2019-12-14	2020-10-11	40.00	58304.92	USD	Vacation 2	B	20200330095241.3632...
00000002	0700	00000002	2019-12-14	2020-10-11	20.00	899.76	USD	Business Tri...	B	20200416123713.6379...
00000003	0700	00000003	2020-10-11	2020-10-11	80.20	4130.34	USD	Vacation test...	B	20200325143206.0944...
00000006	0700	00000006	2020-10-11	2020-10-11	120.00	5573.40	USD	Vacation165...	B	20200416152655.2250...
00000007	0700	00000007	2020-10-11	2020-10-11	120.00	5691.00	USD	Vacation	P	20191128065706.0000...
00000008	0700	00000008	2019-12-16	2020-10-11	60.00	2777.00	USD	Vacation for ...	B	20200317074056.4680...
00000009	0700	00000009	2020-10-11	2020-10-11	120.00	5792.00	USD	Visiting Anna	B	20191122222506.0000...
00000010	070016	000697	2019-12-14	2019-12-16	20.00	957.26	USD	Business Tri...	B	20200323071623.7813...
00000011	070005	000582	2019-12-14	2019-12-14	20.00	959.00	USD	Vacation	B	20200317074400.6109...
00000012	070028	000583	2020-03-03	2020-03-04	90.00	4293.80	USD	Sightseeing L...	B	20200303163902.6525...

Done

Log on to answer question

5

Step 5: Create a service definition

You will now expose the CDS view as a **business service**. This will allow you to preview your changes in Fiori elements preview.

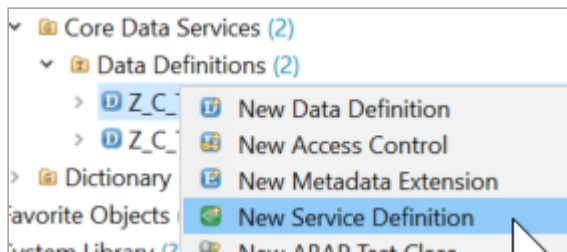
A **business service** consists of a **service definition** and a **service binding**.

You use a **service definition** to define which data is to be exposed (with the required granularity) as a Business Service.

You then use the **service binding** to bind a service definition to a client-server communication protocol such as OData. This allows you to provide several bindings for the same definition, e.g. to expose the service to a UI, and to an **A2X** provider.

For more information, see:

- SAP Help Portal: [Creating a Service Definition](#)
 - SAP Help Portal: [Creating a Service Binding](#)
1. First, create the service definition, by selecting your CDS view and choosing **New > Service Definition** from the context menu.



2. Choose a name and description:

- **Z_EXPOSE_TRAVEL_XXX**
- **Service exposes Travel data**

New Service Definition

Service Definition
Create a service definition

Project: * Browse...

Package: * TEST_JP_ENHANCE_CDS Browse...

☐ Add to favorite packages

Name: * Z_EXPOSE_TRAVEL_003

Description: * Service exposes Travel Data

Original Language: EN

Source Type: Service Definition

? < Back Next > Finish Cancel

3. Choose the transport request; choose **Next**.
4. Use the selected template; choose **Finish**. The name of your custom entity is inserted automatically.

```
D _ Z_C_TRAVEL_DATA_003 * _ Z_EXPOSE_TRAVEL_003
1 @EndUserText.label: 'Service exposes Travel Data'
2 define service Z_EXPOSE_TRAVEL_XXX {
3   expose Z_C_TRAVEL_DATA_XXX;
4 }
```

5. To make the service definition more readable, add an alias to the **expose** statement:

CDS

[Copy](#)

```
1 | expose Z_C_TRAVELS_XXX as Travel;
2 |
```

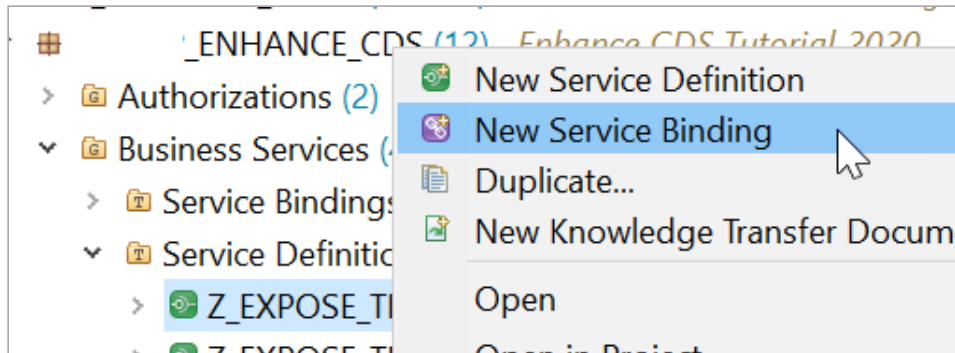
6. Format, save, and activate (**Shift+F1, Ctrl+S, Ctrl+F3**) the service definition.

 Done

Log on to answer question

6 Step 6: Create service binding

1. Select your service definition, then choose **Service Binding** from the context menu, then choose **Next**.



2. Choose:

- Name = **Z_BIND_TRAVEL_XXX**
- Description = **Service binding for Travel data**
- Binding Type = **ODATA V2 (UI...)**
- Service Definition = **Z_EXPOSE_TRAVEL_XXX**

Service Binding

Create Service Binding

Project: *

Browse...

Package: *

TEST_JP_ENHANCE_CDS

Browse...

☐ Add to favorite packages

Name: *

Z_BIND_TRAVEL_003

Description: *

Service binding for Travel data

Original Language:

EN

Binding Type: *

ODATA V2 - UI

Service Definition: *

Z_EXPOSE_TRAVEL_003

Browse...

?

< Back

Next >

Finish

Cancel

3. Choose the transport request; choose **Finish**.

The service binding automatically references the service definition and thus the exposed custom entity.

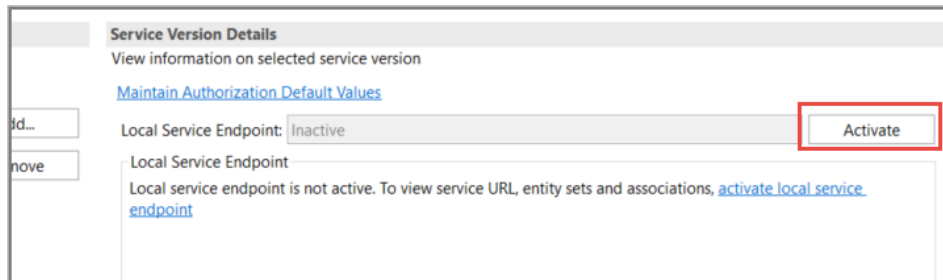
 Done

Log on to answer question

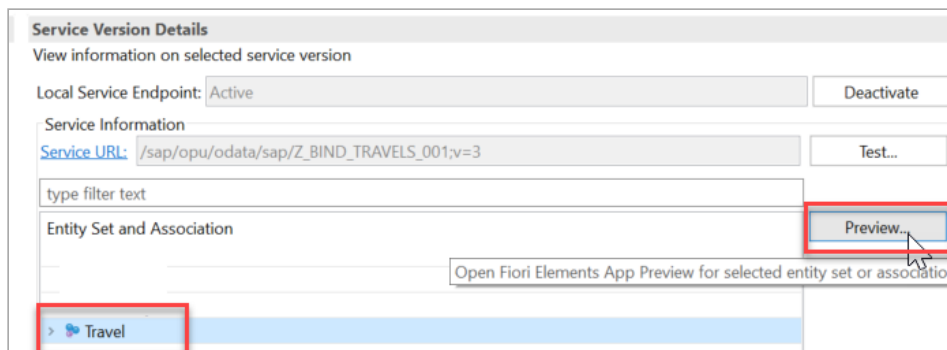
7

Step 7: Activate service binding

1. In the editor that appears, choose **Activate**.



2. You can now see the Service URL and Entity Set.



3. To open the Service Document (`XML`) in your browser, choose **Service URL**.

8

Step 8: Display Fiori Elements Preview


1. Select the entity set and choose **Preview**.

The screenshot shows the 'Service Version Details' dialog box. It contains the following elements:

- Service Version Details** header.
- View information on selected service version** subtitle.
- Local Service Endpoint:** A text field containing 'Active' and a 'Deactivate' button.
- Service Information** section:

 - Service URL:** A text field containing '/sap/opu/odata/sap/Z_BIND_TRAVEL_003' and a 'Test...' button.
 - A 'type filter text' input field.




- Entity Set and Association** table:


Entity Set and Association
 Z_C_TRAVEL_DATA_003



- Preview...** button.

Red circles with numbers 1 and 2 are overlaid on the image. Circle 1 points to the 'Z_C_TRAVEL_DATA_003' entity set in the table. Circle 2 points to the 'Preview...' button.

2. Log in using your ABAP Environment user and password; the Fiori Elements preview appears.
3. By default, no columns are selected. To see the data, choose **Settings**, then choose **Select All**.

Standard*   

Not Filtered 

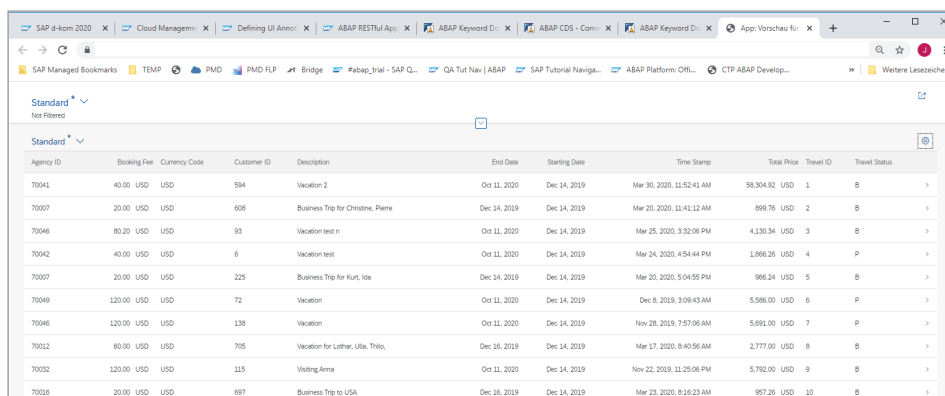
Standard  

No data found. Try adjusting the filter settings.

View Settings

Columns	Sort
Search	
<input checked="" type="checkbox"/> Select All (11/11)	
<input checked="" type="checkbox"/> Agency ID	

4. Display the data by choosing **Go**.



Agency ID	Booking Fee	Currency Code	Customer ID	Description	End Date	Starting Date	Time Stamp	Total Price	Travel ID	Travel Status
70041	40.00	USD	594	Vacation 2	Oct 11, 2020	Dec 14, 2019	Mar 30, 2020, 11:52:41 AM	58,304.92	USD 1	B
70007	20.00	USD	608	Business Trip for Christine, Pierre	Dec 14, 2019	Dec 14, 2019	Mar 20, 2020, 11:41:12 AM	899.76	USD 2	B
70046	80.20	USD	93	Vacation test n	Oct 11, 2020	Dec 14, 2019	Mar 25, 2020, 3:32:06 PM	4,130.34	USD 3	B
70042	40.00	USD	6	Vacation test	Oct 11, 2020	Dec 14, 2019	Mar 24, 2020, 4:54:44 PM	1,866.26	USD 4	P
70007	20.00	USD	225	Business Trip for Kurt, Iida	Dec 14, 2019	Dec 14, 2019	Mar 20, 2020, 5:04:55 PM	995.24	USD 5	B
70049	120.00	USD	72	Vacation	Oct 11, 2020	Dec 14, 2019	Dec 8, 2019, 3:09:43 AM	5,586.00	USD 6	P
70046	120.00	USD	138	Vacation	Oct 11, 2020	Dec 14, 2019	Nov 26, 2019, 7:57:06 AM	5,661.00	USD 7	P
70012	60.00	USD	705	Vacation for Lothar, Ulla, Thilo,	Dec 16, 2019	Dec 14, 2019	Mar 17, 2020, 8:40:56 AM	2,777.00	USD 8	B
70002	120.00	USD	115	Visiting Anna	Oct 11, 2020	Dec 14, 2019	Nov 22, 2019, 11:25:06 PM	5,792.00	USD 9	B
70026	20.00	USD	897	Business Trip to USA	Dec 16, 2019	Dec 14, 2019	Mar 23, 2020, 8:18:23 AM	957.26	USD 10	B

 Done

Log on to answer question

Step 9: Add annotation for automatic display

1. It would be nice if at least some fields were displayed immediately for the user. To do this, simply add the following annotation to the relevant fields in `Z_C_TRAVEL_DATA_XXX`. The start of the CDS view will then look like this.

`BookingFee` is not automatically displayed. The numbers for each field are relative to the other fields and are responsive - they do not refer to a specific pixel position or similar. For larger entities, you can specify *HIGH*, **MEDIUM**, or *LOW*, so that less important fields are automatically hidden on a smaller screen, such as a mobile phone.

CDS

[Copy](#)

```

1  @UI          : {
2  lineItem     : [{position: 10, importance: #HIGH}]
3  }
4  @UI          : {
5  lineItem     : [{position: 10, importance: #HIGH}]
6  }
7  key TravelID;
8
9  @UI          : {
10 lineItem     : [{position: 15, importance: #HIG

```

```

11     }
12     AgencyID,
13
14     @UI      : {
15         lineItem      : [{position: 20, importance: #HIGH}]
16     }
17     CustomerID,
18
19     @UI      : {
20         lineItem      : [{position: 30, importance: #HIGH}]
21     }
22     BeginDate,
23
24     @UI      : {
25         lineItem      : [{position: 40, importance: #HIGH}]
26     }
27     EndDate,
28
29     BookingFee,
30
31     @UI      : {
32         lineItem      : [{position: 50, importance: #HIGH}]
33     }
34     TotalPrice,
35

```

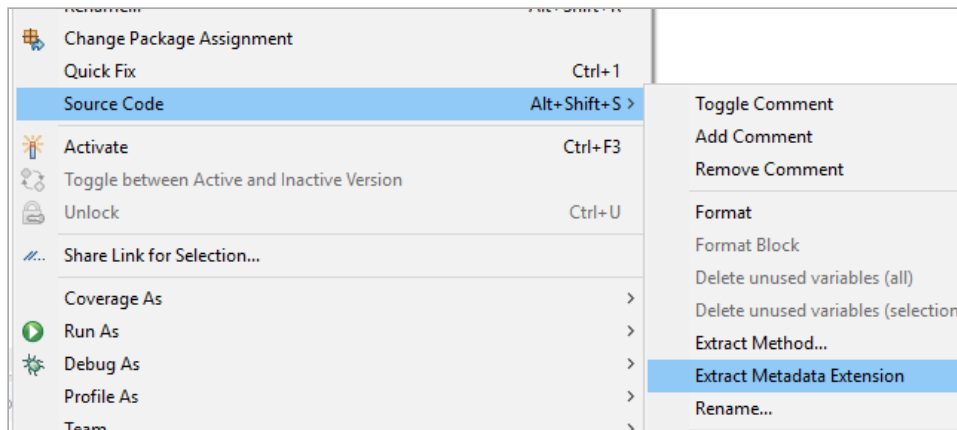
2. If you now refresh your Fiori Elements preview, you will notice that you do not have to choose the fields; you simply have to choose **Go**.

 Done

10 Step 10: Extract UI metadata

At present, you only have minimal annotations. As you add more, your CDS view will start to get cluttered. So you should extract your UI annotations to a separate object, a **metadata extensions** object, as follows:

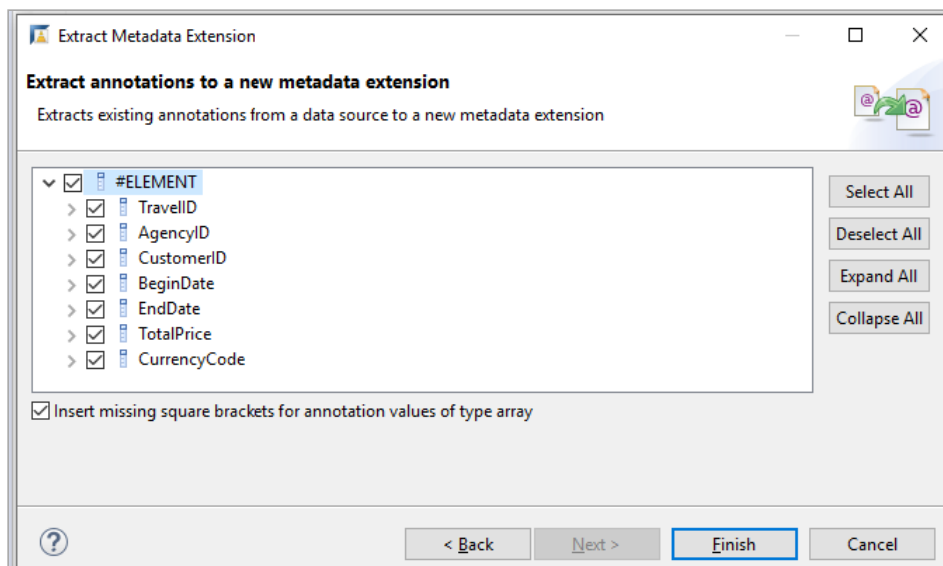
1. First add the annotation `@Metadata.allowExtensions: true` to your CDS view.
2. Then, click anywhere in the editor, then choose **Source Code > Extract Metadata Extension** from the context menu.



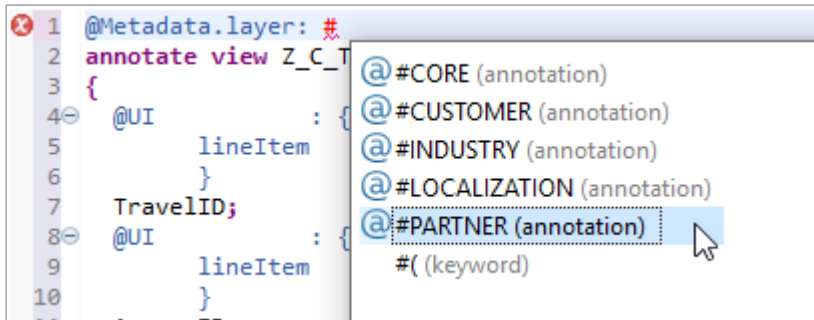
3. Enter a name and description for your metadata extension object, clearly similar to your CDS view name, and choose **Next**:

- `Z_TRAVEL_METADATA_XXX`
- `Metadata for Z_C_TRAVEL_DATA_XXX`

4. Accept the transport request, choose **Next**, select all elements, then choose **Finish**.



5. You will get an error, because you have not yet assigned the metadata extension to a layer. Since you are in sandbox mode, enter the value `#CORE` using auto-complete (**Ctrl+Space**).



Layers allow customers or partners, for example, to enhance the metadata without modifying the CDS entity. You can also add industry- or country-specific enhancements.

The metadata extensions are evaluated in a specific order. For more information, see [Annotation Propagation](#).

6. Format, save, and activate (**Shift+F1, Ctrl+S, Ctrl+3**).

 Done

Log on to answer question

11

Step 11: Add semantic metadata

If you define currency amounts and currency codes semantically, then the system will apply specific rules to handle these fields appropriately. For example, in this tutorial, if you define `TotalPrice` as a currency amount, and define `CurrencyCode` as a currency code field, then the system will add the appropriate currency to the `TotalPrice` column automatically. There is no need to display `CurrencyCode` as a separate column.

1. To do this, add the following two annotations to your CDS view:

CDS

[Copy](#)

```
1 | @Semantics.amount.currencyCode: 'CurrencyCode'  
2 | TotalPrice,  
3 |  
4 | @Semantics.currencyCode  
5 | CurrencyCode,  
6 |
```

2. Format, save, and activate (`Shift+F1, Ctrl+S, Ctrl+3`).
3. If you refresh the Fiori Elements preview, the **Total Price** column now looks like this.

Total Price		
58,304.92	USD	>
899.76	USD	>
4,130.34	USD	>
5,573.40	USD	>

 Done

Log on to answer question

12 Step 12: Add search field

You will now add a fuzzy search capability.

1. First, add the search annotation to your CDS view:

CDS

[Copy](#)

```
1 | @Search.searchable: true
```

2. Then add the following two annotations to the field you want to search, in this case `Memo` :

CDS

Copy

```
1 | @Search.defaultSearchElement: true
2 | @Search.fuzzinessThreshold: 0.90
3 |
```

3. For convenience, add the following annotation to the metadata extension object, so that the **Memo** field appears by default in the preview, then format, save, and activate (`Shift+F1, Ctrl+S, Ctrl+3`):

CDS

Copy

```
1 | @UI          : {
2 |     lineItem  : [{position: 60, importance: #HIGH}]
3 | }
4 | Memo;
5 |
```

4. Refresh the Fiori elements preview in your browser.
5. There is a new **Search** input field.

Standard* ✓

Standard ✓

Travel ID Agency ID Customer ID

- Enter the search text **Miami**. The app only displays trips to Miami (to date, eleven trips).

Standard* ✓

Miami ✕ 🔍

Adapt Filters (1) Go

Standard ✓

Travel ID	Agency ID	Customer ID	Starting Date	End Date	Description	Total Price
12	70028	583	Mar 3, 2020	Mar 4, 2020	Sightseeing in Miami, Florida	4,293.80 USD >
15	70028	466	Dec 14, 2019	Oct 9, 2020	Sightseeing in Miami, Florida	3,001.00 USD >
40	70026	357	Oct 9, 2020	Oct 11, 2020	Sightseeing in Miami, Florida	1,324.00 USD >
216	70027	370	Dec 15, 2019	Dec 15, 2019	Sightseeing in Miami, Florida	448.00 USD >
226	70026	442	Dec 15, 2019	Dec 15, 2019	Sightseeing in Miami, Florida	175.00 USD >

- Optional: You can test the fuzziness threshold by changing the value to 0.70. After you save and activate, the app will now show trips to Miami and trips involving Matthias. (You may need to empty the cache.)

226	70026	442	Dec 15, 2019	Dec 15, 2019	Sightseeing in Miami, Florida
228	70028	380	Dec 15, 2019	Dec 15, 2019	Sightseeing in Miami, Florida
230	70026	455	Dec 15, 2019	Dec 15, 2019	Sightseeing in Miami, Florida
233	70027	288	Dec 15, 2019	Dec 15, 2019	Sightseeing in Miami, Florida
234	70028	268	Dec 15, 2019	Dec 15, 2019	Sightseeing in Miami, Florida
275	70029	486	Dec 15, 2019	Dec 15, 2019	Sightseeing in Miami, Florida
528	70030	610	Oct 10, 2020	Oct 10, 2020	Sightseeing in Miami, Florida
1186	70010	365	Dec 13, 2019	Dec 13, 2019	Business Trip for Juan, Amelie, Matthias
1345	70006	90	Dec 14, 2019	Dec 14, 2019	Business Trip for Adam, Matthias, Jean-Luc
1356	70006	660	Dec 14, 2019	Dec 14, 2019	Business Trip for James, Matthias

 Done

Log on to answer question

13 Step 13: Add selection fields

As well as search fields, you can filter the list using an input field. In the next tutorial, you will provide input value help for these fields.

1. Add the `selectionField` annotation to the field `TravelID` in your metadata extension file, so that the whole UI annotation looks like this:

CDS

[Copy](#)

```

1 | @UI          : {
2 |     lineItem   : [{position: 10, importance: #HIGH}]
3 |     selectionField: [{position: 10 }]

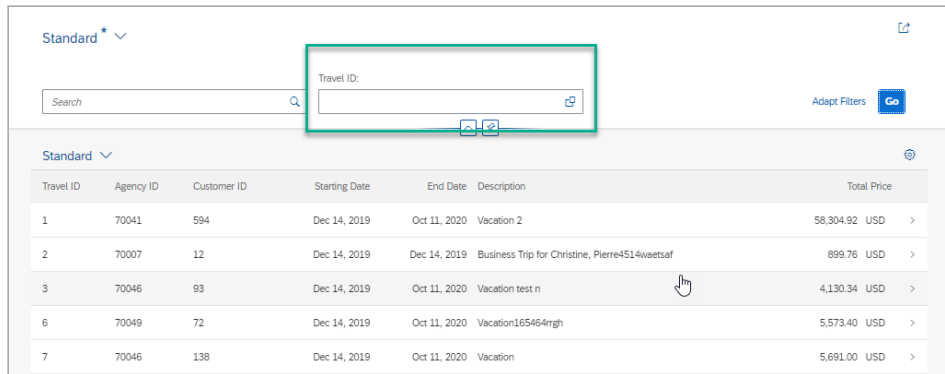
```

```

4     }
5     TravelID;

```

2. Format, save, and activate (**Shift+F1, Ctrl+S, Ctrl+3**). The Fiori elements preview should now look like this:



Travel ID	Agency ID	Customer ID	Starting Date	End Date	Description	Total Price
1	70041	594	Dec 14, 2019	Oct 11, 2020	Vacation 2	58,304.92 USD >
2	70007	12	Dec 14, 2019	Dec 14, 2019	Business Trip for Christine, Pierre4514waetsaf	899.76 USD >
3	70046	93	Dec 14, 2019	Oct 11, 2020	Vacation test n	4,130.34 USD >
6	70049	72	Dec 14, 2019	Oct 11, 2020	Vacation165464ngh	5,573.40 USD >
7	70046	138	Dec 14, 2019	Oct 11, 2020	Vacation	5,691.00 USD >

3. Add other fields as input fields by adding the following to the metadata extensions file, so that the file looks like this:

CDS

Copy

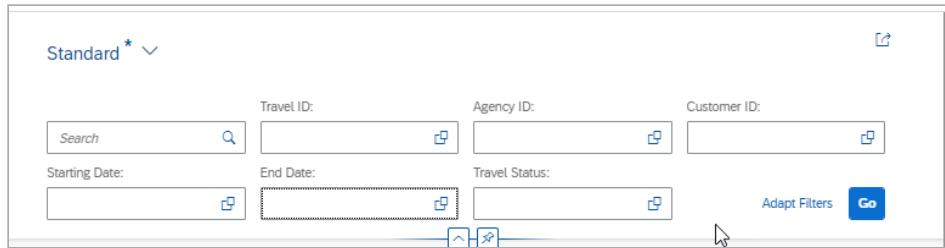
```

1  @UI      : {
2      lineItem      : [{position: 15, importance: #HIG
3      selectionField: [{position: 15 }]
4      }
5  AgencyID;
6
7  @UI      : {
8      lineItem      : [{position: 20, importance: #HIG
9      selectionField: [{position: 20 }]
10     }
11 CustomerID;

```

```
12
13 @UI      : {
14     listItem      : [{position: 10, importance: #HIGH}
15     selectionField: [{position: 10 }]
16     }
17 TravelID;
18
19 @UI      : {
20     listItem      : [{position: 30, importance: #HIGH}
21     selectionField: [{position: 30 }]
22     }
23 BeginDate;
24
25 @UI      : {
26     listItem      : [{position: 40, importance: #HIGH}
27     selectionField: [{position: 40 }]
28     }
29 EndDate;
30
31 @UI      : {
32     listItem      : [{position: 50, importance: #HIGH}
33     }
34 TotalPrice;
35
36 @UI      : {
37     listItem      : [{position: 50, importance: #HIGH}
38     }
39 Memo;
40
41 @UI      : {
42     listItem      : [{position: 60, importance: #HIGH}
43     selectionField: [{position: 60 }]
44     }
45 Status;
```

Your app should now look like this:



 Done

Log on to answer question

14

Step 14: Check your code

Your CDS entity code should look like this:

CDS

[Copy](#)

```
1 @AbapCatalog.sqlViewName: 'ZCTRAVEL_XXX'
2 @AbapCatalog.compiler.compareFilter: true
3 @AbapCatalog.preserveKey: true
4 @AccessControl.authorizationCheck: #NOT_REQUIRED
5 @EndUserText.label: 'Consumption view from /DMO/I_TRAVEL_U'
6 @Metadata.allowExtensions: true
7 @Search.searchable: true
8
```

```
9  define view Z_C_TRAVEL_DATA_XXX
10    as select from /DMO/I_Travel_U
11    {
12
13        ///DMO/I_Travel_U
14
15        key TravelID,
16        AgencyID,
17        CustomerID,
18        BeginDate,
19        EndDate,
20        BookingFee,
21
22        @Semantics.amount.currencyCode: 'CurrencyCode'
23        TotalPrice,
24
25        @Semantics.currencyCode
26        CurrencyCode,
27
28        @Search.defaultSearchElement: true
29        @Search.fuzzinessThreshold: 0.90
30        Memo,
31        Status,
32        LastChangedAt,
33
34        /* Associations */
35        ///DMO/I_Travel_U
36        _Agency,
37        _Booking,
38        _Currency,
39        _Customer
40
41    }
42
```


Your MDE code should look like this:

CDS

[Copy](#)

```
1  @Metadata.layer: #CORE
2  annotate view Z_C_TRAVEL_DATA_XXX with
3  {
4
5  @UI          : {
6      lineItem      : [{position: 15, importance: #HIGH}],
7      selectionField: [{position: 15 }]
8  }
9  AgencyID;
10
11 @UI          : {
12     lineItem      : [{position: 20, importance: #HIGH}],
13     selectionField: [{position: 20 }]
14 }
15 CustomerID;
16
17 @UI          : {
18     lineItem      : [{position: 10, importance: #HIGH}],
19     selectionField: [{position: 10 }]
20 }
21 TravelID;
22
23 @UI          : {
24     lineItem      : [{position: 30, importance: #HIGH}],
25     selectionField: [{position: 30 }]
26 }
27 BeginDate;
28
29 @UI          : {
```

```
30     lineItem      : [{position: 40, importance: #HIGH}],
31     selectionField: [{position: 40 }]
32   }
33   EndDate;
34
35   @UI           : {
36     lineItem      : [{position: 50, importance: #HIGH}]
37   }
38   TotalPrice;
39
40   @UI           : {
41     lineItem      : [{position: 50, importance: #HIGH}]
42   }
43   Memo;
44
45   @UI           : {
46     lineItem      : [{position: 60, importance: #HIGH}],
47     selectionField: [{position: 60 }]
48   }
49   Status;
50
51 }
52
```

 Done


Log on to answer question

Step 15: Test yourself

Where do you specify that your OData service is an OData version 2 service for UI?

- ☐ Data definition
- ☐ Metadata extension
- ☐ Service definition
- ☐ Service binding

 **Submit Answer**

Log on to answer question 

Next Steps

This tutorial is part of these groups and missions:



Group

Beginner

 1 hr. 50 min.

2 tutorials

Create and Enhance a CDS View in SAP Cloud Platform, ABAP Environment

Create and enhance a CDS view using annotations and expressions, to make your applications more powerful and more user-friendly.

 [ABAP Development](#)

Developer Products

[ABAP Platform](#)

[SAP Business Application Studio](#)

[SAP Business Technology Platform](#)

[SAP Conversational AI](#)

[SAP Data Intelligence](#)

[SAP HANA](#)

[All Products](#)

Trials & Downloads

[ABAP Development Tools](#)

[Mobile Development Kit Client](#)

[SAP Business Application Studio](#)

[SAP Data Intelligence Trial](#)

[SAP HANA Cloud Trial](#)

[All Trials & Downloads](#)

Site Information

[Privacy](#)

[Terms of Use](#)

[Legal Disclosure](#)

[Copyright](#)

[Trademark](#)

[Newsletter](#)

[Sitemap](#)

[Text View](#)

Find us on