Technical Articles

Paul McFarling

December 29, 2019 | 8 minute read

SAPUI5 and OData UploadCollection Example

💬 4 👍 16 👁 10,382

Follow

👍 Like

RSS Feed

Overview

In this post you will see an end to end example of implementing file upload/download with the UploadCollection control. This post is meant to help you get basic upload/download functionality working.

I have studied several blog posts, questions as well as personal experience to produce this post. I couldn't find a "one stop" example so I decided to put this together help the community get started.

Assumptions:

- You have an on premise SAP Netweaver Gateway system
- You have access to WebIDE and a connection to your Gateway System

- This was developed on an S4/1909 system using primarily ADT*.
- The front end was targeted for SAPUI5 Version 1.73*

**

**I have implemented a similar solution on an R/3 ABAP 7.50 System using SAPUI5 version 1.52 so it should work with very little to no changes on earlier versions.*

Basic Steps:

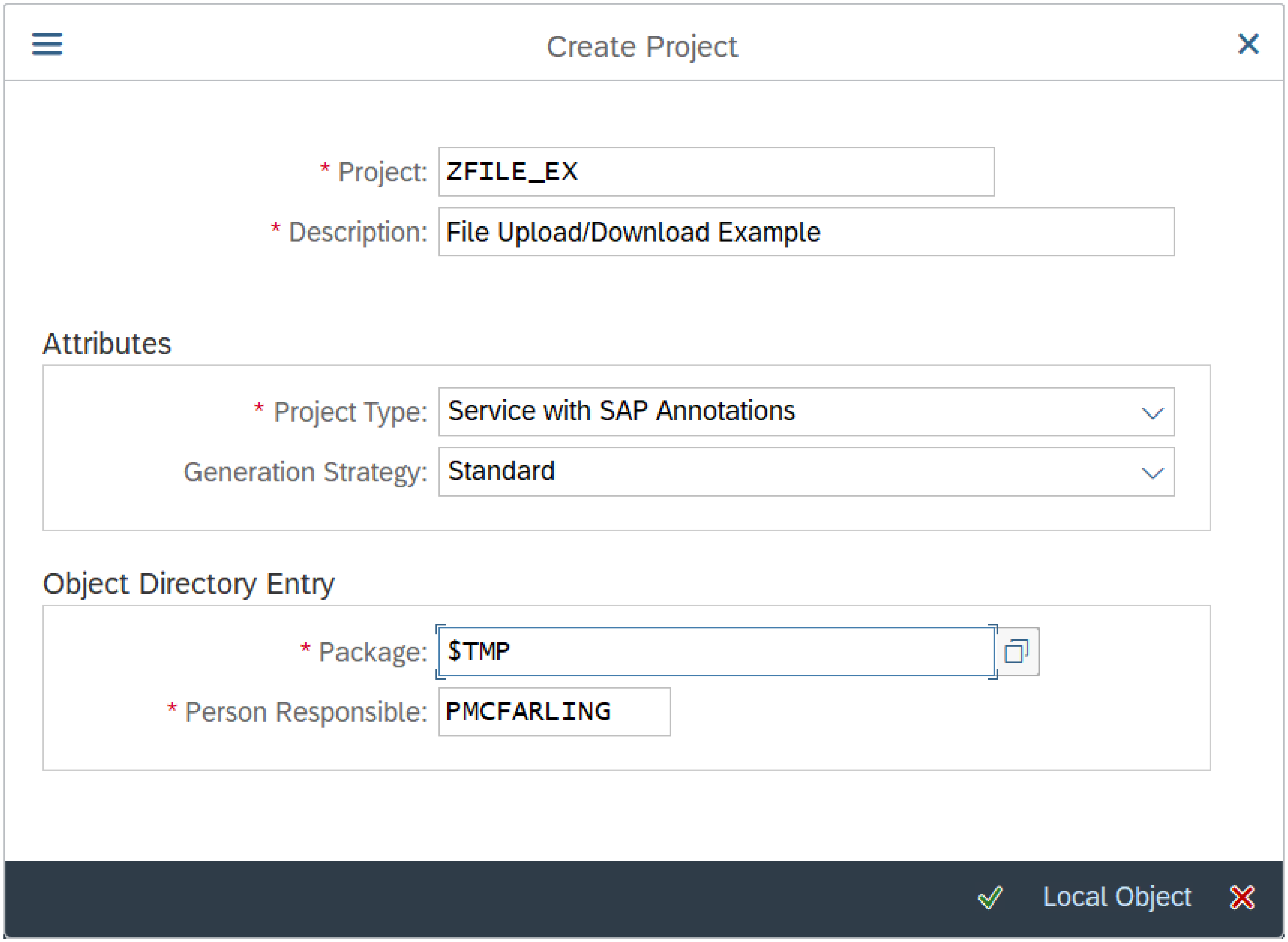
- Define a Z-Table to hold files data and attributes
- Use the gateway service builder (T-Code SEGW) to create the Entity Type/Entity Sets
- Redefine methods in DPC and MPC classes.
- Expose the Service on the Gateway
- Create a new SAPUI5 Application using WebIDE on SAP Cloud Platform
- Configure neo-app.json and manifest.json
- Add the UploadCollection to the View.
- Implement Supporting controller code

Create Z Table and OData Service:

1. Create a ztable to store the file and file attributes. This could easily be adapted to store the file in a content server or on the file system. Note:I have used a UUID as the Primary Key should this table be used in a draft enabled BOPF in the future.

```
@EndUserText.label : 'Files'
@AbapCatalog.enhancementCategory : #NOT_EXTENSIBLE
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #LIMITED
define table zfile {
  key client : abap.clnt not null;
  key id     : uuid not null;
  filename   : char255;
  sydate     : dats;
  sytime     : tims;
  value      : xstringval;
  mimetype   : char100;

}
```

2. Using T-Code SEGW on your business system, Create a new Project



* Project: ZFILE_EX

* Description: File Upload/Download Example

Attributes

* Project Type: Service with SAP Annotations

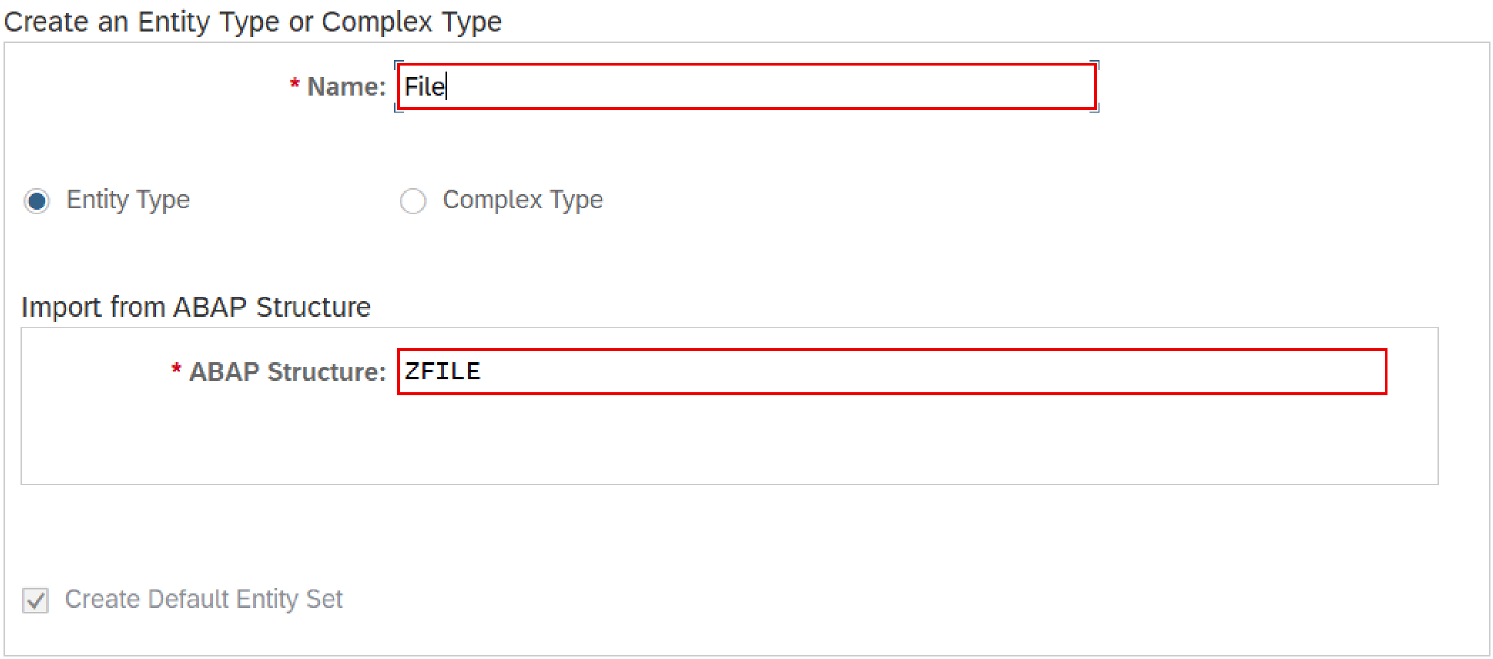
Generation Strategy: Standard

Object Directory Entry

* Package: $TMP

* Person Responsible: PMCFARLING

3. Enter a Name and give the Z-Table name from step 1. Make sure “Create Default Entity Set” is checked. Then, press Next



Create an Entity Type or Complex Type

* Name: File

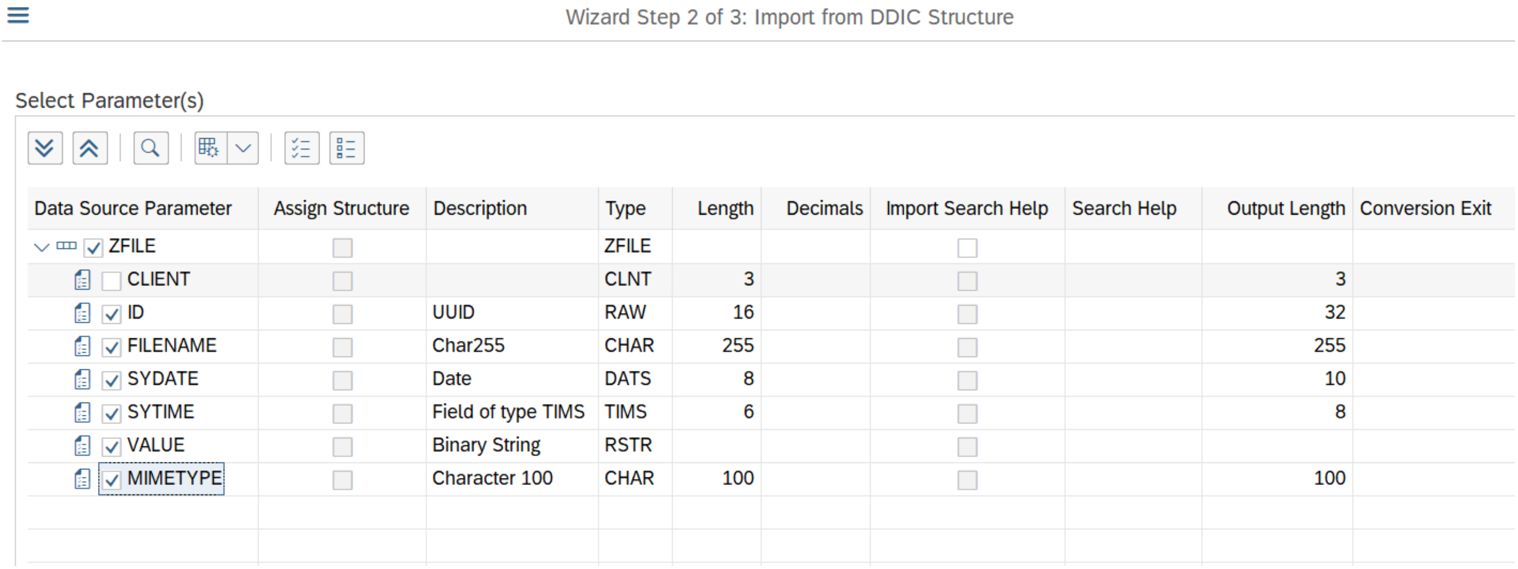
● Entity Type ○ Complex Type

Import from ABAP Structure

* ABAP Structure: ZFILE

☑ Create Default Entity Set

4. Select all the fields, and press next. (Selecting CLIENT is not necessary)

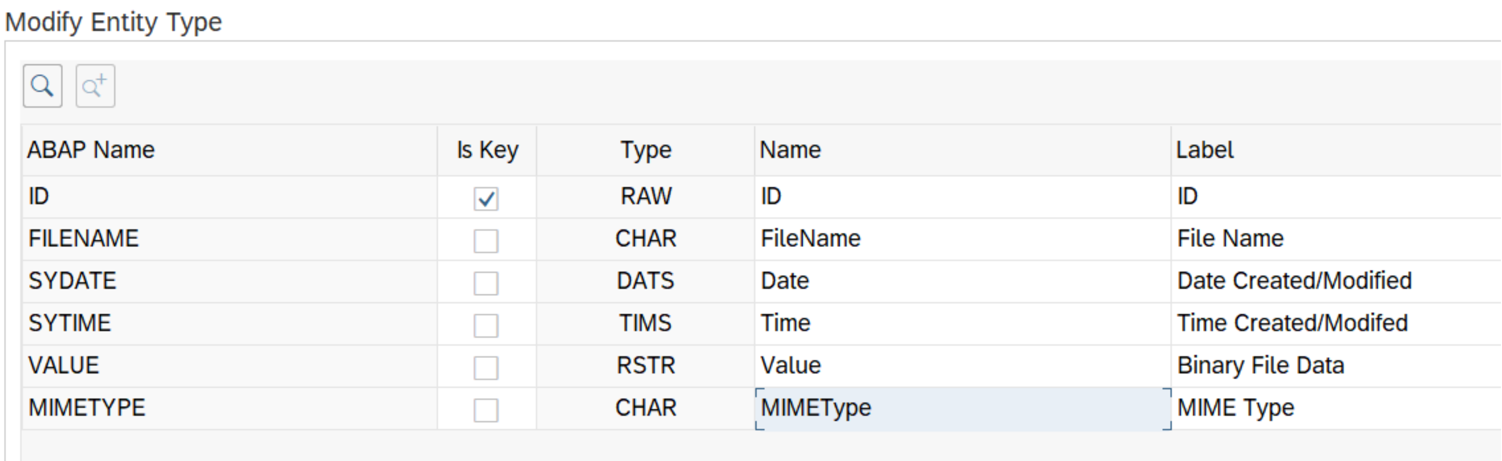


Wizard Step 2 of 3: Import from DDIC Structure

Select Parameter(s)

Data Source Parameter	Assign Structure	Description	Type	Length	Decimals	Import Search Help	Search Help	Output Length	Conversion Exit
☑ ZFILE	□		ZFILE			□			
□ CLIENT	□		CLNT	3		□		3	
☑ ID	□	UUID	RAW	16		□		32	
☑ FILENAME	□	Char255	CHAR	255		□		255	
☑ SYDATE	□	Date	DATS	8		□		10	
☑ SYTIME	□	Field of type TIMS	TIMS	6		□		8	
☑ VALUE	□	Binary String	RSTR			□			
☑ MIMETYPE	□	Character 100	CHAR	100		□		100	

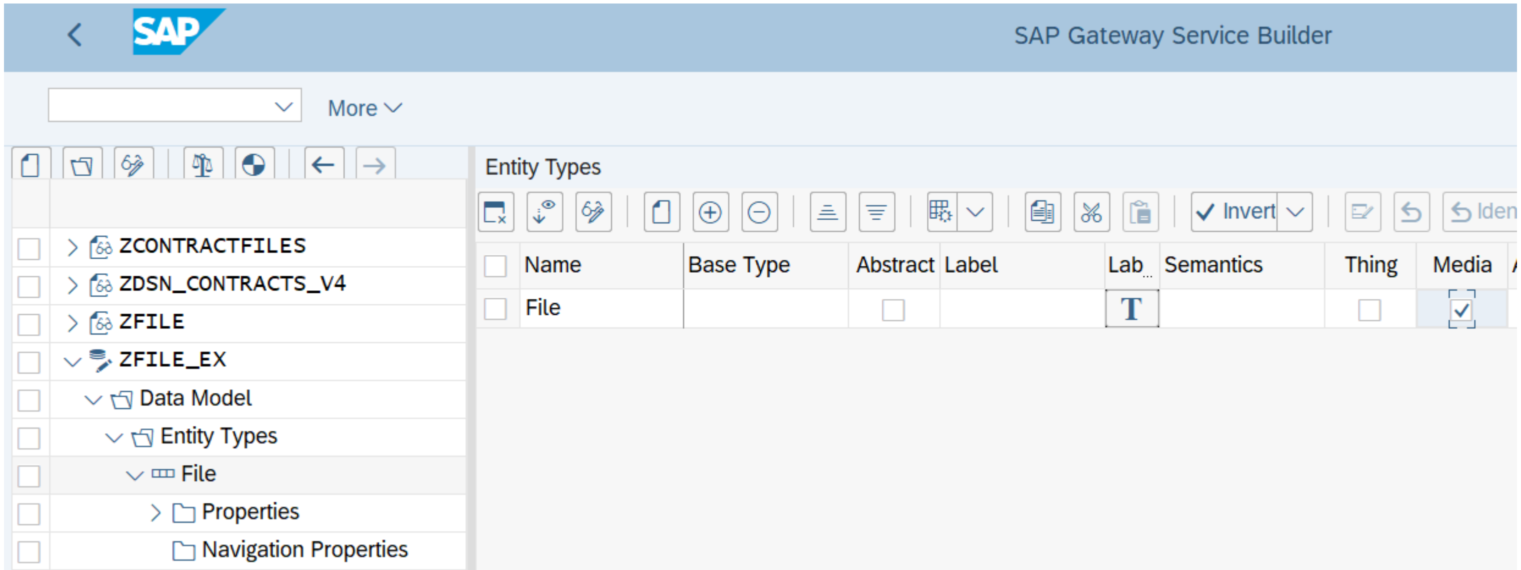
5. Select “ID” as the Key and give a meaningful name and label to the fields. Then, Press Finish.



Modify Entity Type

ABAP Name	Is Key	Type	Name	Label
ID	☑	RAW	ID	ID
FILENAME	□	CHAR	FileName	File Name
SYDATE	□	DATS	Date	Date Created/Modified
SYTIME	□	TIMS	Time	Time Created/Modifed
VALUE	□	RSTR	Value	Binary File Data
MIMETYPE	□	CHAR	MIMEType	MIME Type

6. Select the entity type you just created (File) from the tree hierarchy on the left. Check the check box in the “Media” column.

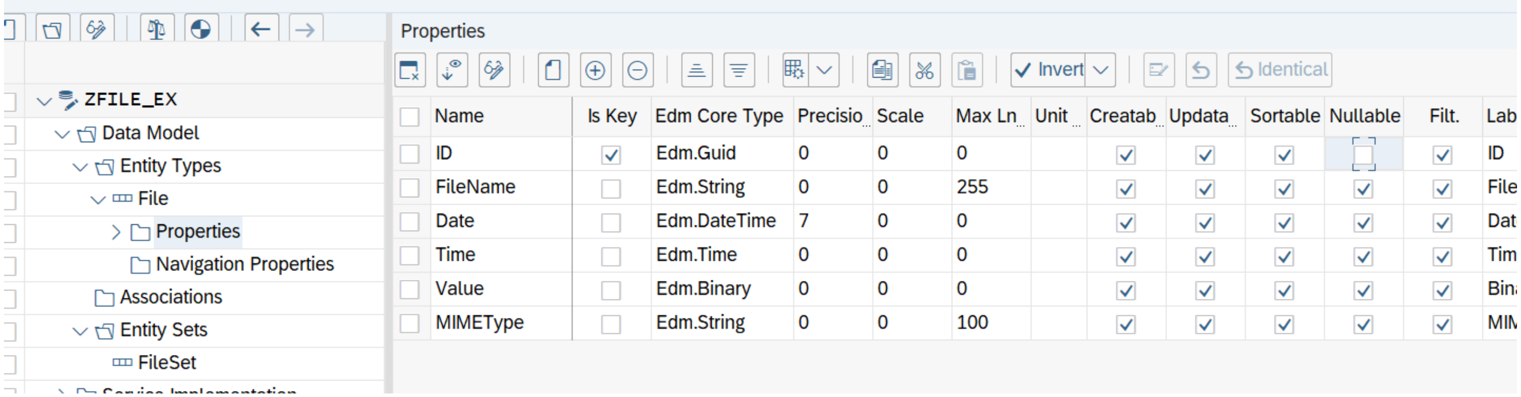


- □ ZCONTRACTFILES
- □ ZDSN_CONTRACTS_V4
- □ ZFILE
- □ ZFILE_EX
  - □ Data Model
    - □ Entity Types
      - □ File
        - □ Properties
        - □ Navigation Properties

Entity Types

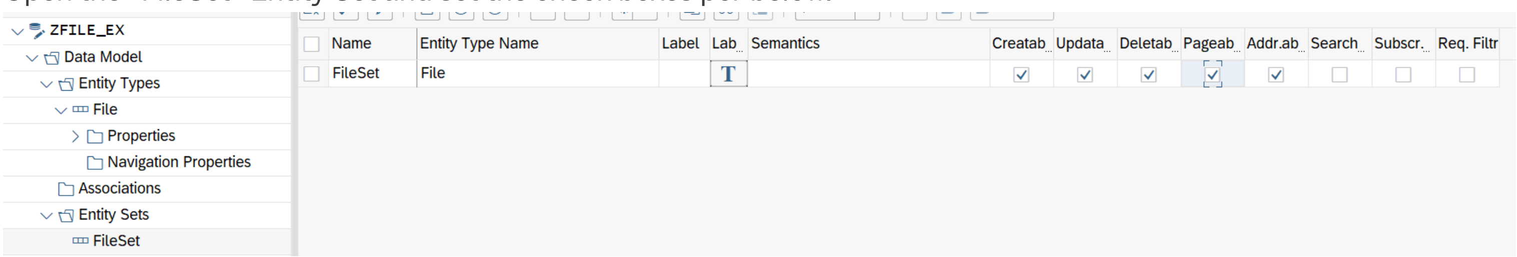
	Name	Base Type	Abstract	Label	Lab...	Semantics	Thing	Media
□	File		□		T		□	☑

7. Open the Entity Type Properties and set the check boxes per below



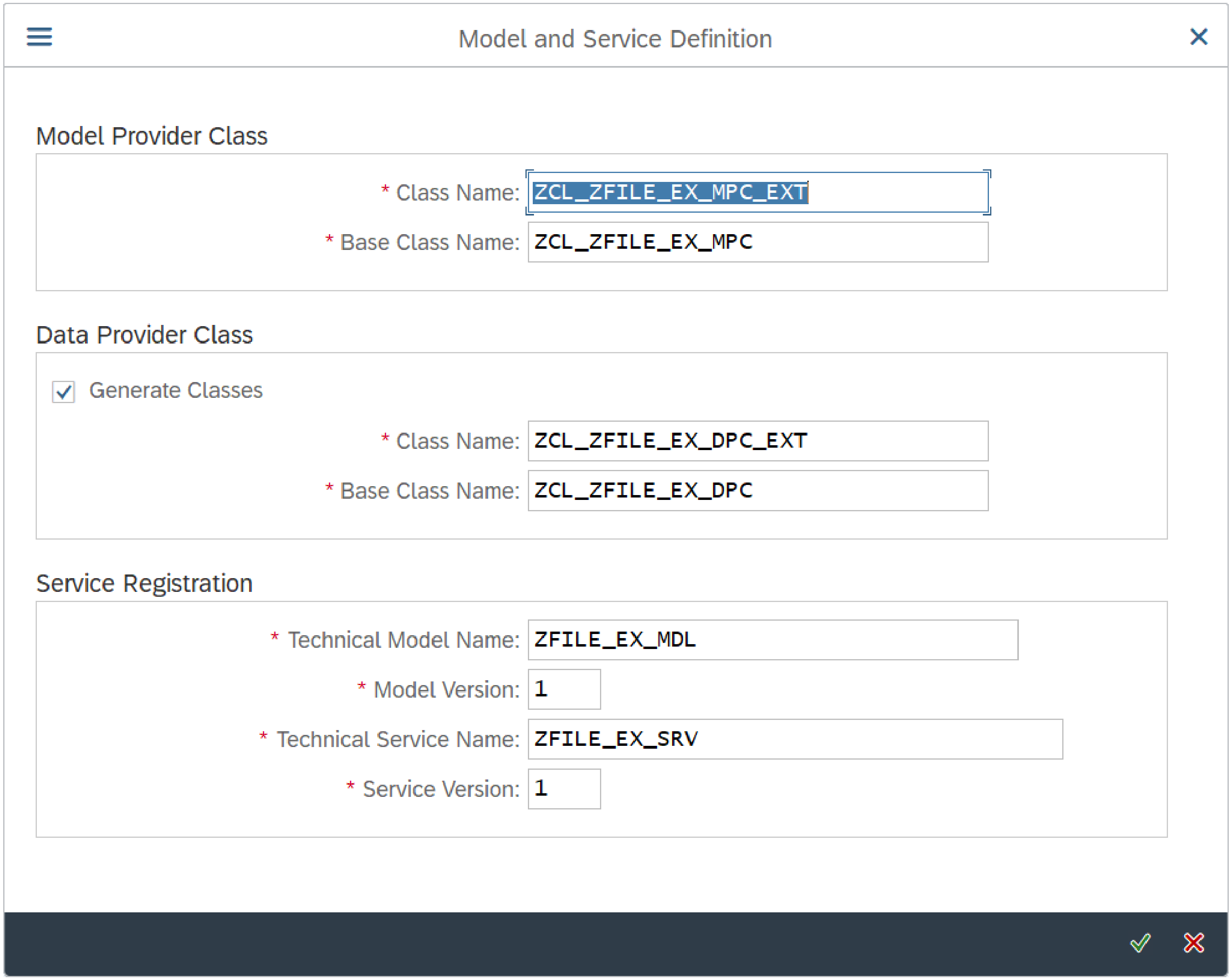
☐	Name	Is Key	Edm Core Type	Precisio	Scale	Max Ln	Unit	Creatab	Updata	Sortable	Nullable	Filt.	Lab
☐	ID	☑	Edm.Guid	0	0	0		☑	☑	☑	☐	☑	ID
☐	FileName	☐	Edm.String	0	0	255		☑	☑	☑	☑	☑	File
☐	Date	☐	Edm.DateTime	7	0	0		☑	☑	☑	☑	☑	Dat
☐	Time	☐	Edm.Time	0	0	0		☑	☑	☑	☑	☑	Tim
☐	Value	☐	Edm.Binary	0	0	0		☑	☑	☑	☑	☑	Bin
☐	MIMEType	☐	Edm.String	0	0	100		☑	☑	☑	☑	☑	MIM

8. Open the “FileSet” Entity Set and set the check boxes per below.



☐	Name	Entity Type Name	Label	Lab	Semantics	Creatab	Updata	Deletab	Pageab	Addr.ab	Search	Subscr.	Req. Filtr
☐	FileSet	File		T		☑	☑	☑	☑	☑	☐	☐	☐

9. Press “Generate Runtime Objects” Icon. Take the defaults in the screen that pops up then, press the check box.



Model and Service Definition

Model Provider Class

* Class Name:	ZCL_ZFILE_EX_MPC_EXT
* Base Class Name:	ZCL_ZFILE_EX_MPC

Data Provider Class

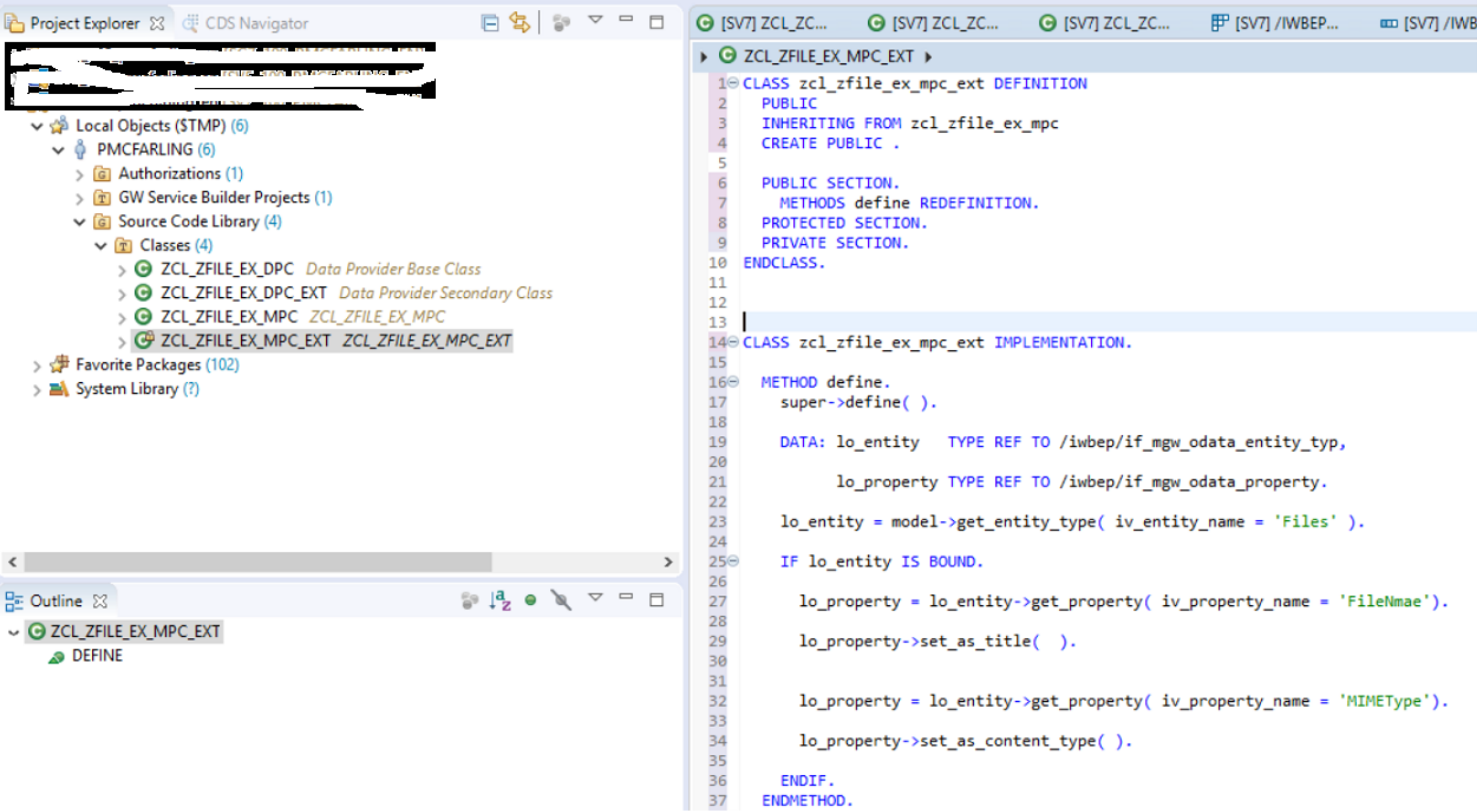
☑ Generate Classes

* Class Name:	ZCL_ZFILE_EX_DPC_EXT
* Base Class Name:	ZCL_ZFILE_EX_DPC

Service Registration

* Technical Model Name:	ZFILE_EX_MDL
* Model Version:	1
* Technical Service Name:	ZFILE_EX_SRV
* Service Version:	1

10. in ADT (or SE24) navigate to the generated Model Provider Class i.e. the *MPC_EXT class. Redefine the "DEFINE" Method. with the code below. Hint: Pressing <CTRL>+<SPACE> in ADT will allow you to select the DEFINE method to redefine.



```
CLASS zcl_zfile_ex_mpc_ext DEFINITION
  PUBLIC
  INHERITING FROM zcl_zfile_ex_mpc
  CREATE PUBLIC .

  PUBLIC SECTION.
    METHODS define REDEFINITION.
  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.



CLASS zcl_zfile_ex_mpc_ext IMPLEMENTATION.

  METHOD define.
    super->define( ).

    DATA: lo_entity   TYPE REF TO /iwbep/if_mgw_odata_entity_typ,

          lo_property TYPE REF TO /iwbep/if_mgw_odata_property.

    lo_entity = model->get_entity_type( iv_entity_name = 'Files' ).

    IF lo_entity IS BOUND.

      lo_property = lo_entity->get_property( iv_property_name = 'FileNmae').

      lo_property->set_as_title(  ).


      lo_property = lo_entity->get_property( iv_property_name = 'MIMEType').

      lo_property->set_as_content_type( ).

    ENDIF.
  ENDMETHOD.
```

```
METHOD define.
  super->define( ).

  DATA: lo_entity   TYPE REF TO /iwbep/if_mgw_odata_entity_typ,

        lo_property TYPE REF TO /iwbep/if_mgw_odata_property.

  lo_entity = model->get_entity_type( iv_entity_name = 'File' ).

  IF lo_entity IS BOUND.

    lo_property = lo_entity->get_property( iv_property_name = 'FileName').

    lo_property->set_as_title(  ).


    lo_property = lo_entity->get_property( iv_property_name = 'MIMEType').

    lo_property->set_as_content_type( ).

  ENDIF.
ENDMETHOD.
```

11. Now open the generated Data Provider Class i.e. *_DPC_EXT. There are several methods here to redefine.

    i. Redefine FILESET_GET_ENTITY with the below code

```
method fileset_get_entity.
      DATA lw_pk TYPE zfile.
   " DO. ENDDO.
    READ TABLE it_key_tab INTO DATA(lw_key_tab) WITH KEY name = 'ID'.
    IF sy-subrc = 0.
      SELECT single id, filename, sydate, sytime, mimetype from zfile into CORRESPONDING FIE
    ENDIF.
    er_entity = lw_pk.
endmethod.

```

    ii. Redefine FILESET_GET_ENTITYSET with the below code.

```
  METHOD fileset_get_entityset.
    DATA:
      it_final   TYPE STANDARD TABLE OF zfile,
      lt_filters TYPE                   /iwbep/t_mgw_select_option,
      ls_filter  TYPE                   /iwbep/s_mgw_select_option,
      ls_so      TYPE                   /iwbep/s_cod_select_option,
      p_name     TYPE c LENGTH 15.

    SELECT
```

```
      id
      filename
      sydate
      sytime
      "value
      mimetype

    FROM zfile
    INTO CORRESPONDING FIELDS OF TABLE et_entityset.
  ENDMETHOD.
```

iii. Redefine method /IWBEP/IF_MGW_APPL_SRV_RUNTIME~CREATE_STREAM. Note that there is a call to FILESET_GET_ENTITY. This is required or the upload will succeed but will return an HTTP 500 error of “Service provider did not return any business data”

```
METHOD /iwbep/if_mgw_appl_srv_runtime~create_stream.
    DATA: lw_file TYPE zfile.

    lw_file-id = cl_system_uuid=>create_uuid_x16_static( ).
    lw_file-filename = iv_slug.
    lw_file-value    = is_media_resource-value.
    lw_file-mimetype = is_media_resource-mime_type.
    lw_file-sydate  = sy-datum.
    lw_file-sytime  = sy-uzeit.

    INSERT INTO zfile VALUES lw_file.

    TRY.
        DATA : ls_key_tab TYPE /iwbep/s_mgw_name_value_pair,

               lt_key_tab TYPE /iwbep/t_mgw_name_value_pair.

        DATA: ls_filemetadata TYPE zfile.
        DATA: lw_key_tab LIKE LINE OF it_key_tab.
        " data: lt_key_tab TYPE /iwbep/t_mgw_name_value_pair.

        lw_key_tab-name = 'ID'.
        lw_key_tab-value = lw_file-id.
        APPEND lw_key_tab TO lt_key_tab.


        CALL METHOD fileset_get_entity
          EXPORTING
            iv_entity_name     = iv_entity_name
            iv_entity_set_name = iv_entity_set_name
            iv_source_name     = iv_source_name
            it_key_tab         = lt_key_tab
*           io_request_object  =
*           io_tech_request_context =
            it_navigation_path = it_navigation_path
          IMPORTING
            er_entity          = ls_filemetadata.
*         es_response_context     =

        copy_data_to_ref(

        EXPORTING

        is_data = ls_filemetadata

        CHANGING

        cr_data = er_entity ).

        "do. enddo.
      CATCH /iwbep/cx_mgw_busi_exception.
      CATCH /iwbep/cx_mgw_tech_exception.
    ENDTRY.
  ENDMETHOD.
```

iv. Redefine /IWBEP/IF_MGW_APPL_SRV_RUNTIME~GET_STREAM.

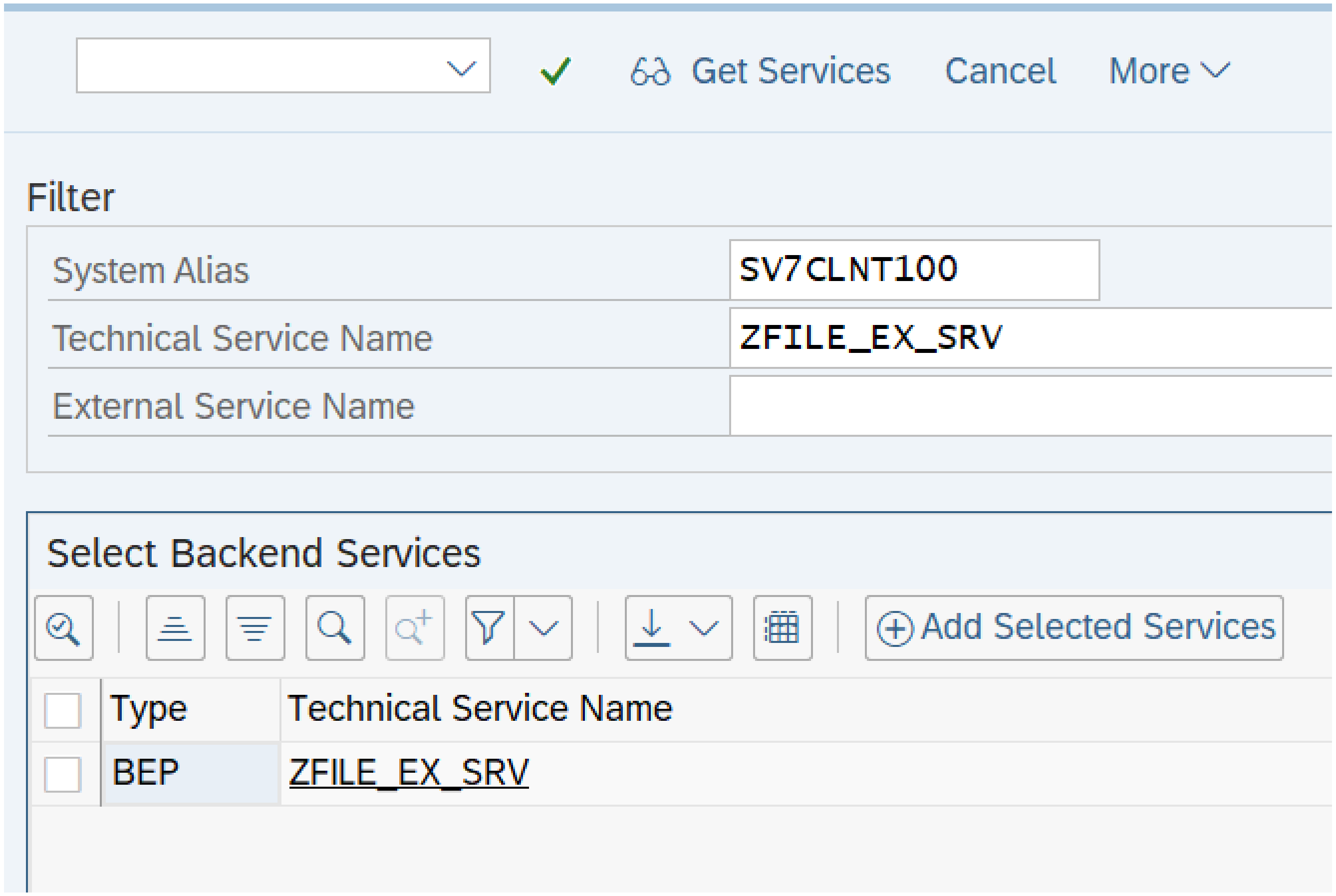
```
method /IWBEP/IF_MGW_APPL_SRV_RUNTIME~GET_STREAM.
  DATA : ls_stream TYPE ty_s_media_resource,

         ls_upld   TYPE zfile.

  DATA: lv_id TYPE zfile-id.

  FIELD-SYMBOLS:<fs_key> TYPE /iwbep/s_mgw_name_value_pair.
  DATA ls_header TYPE ihttpnvp.
  ls_header-name = 'Content-Disposition'.



  READ TABLE it_key_tab ASSIGNING <fs_key> with key name = 'ID'.
  lv_id = <fs_key>-value.
  SELECT SINGLE * FROM zfile INTO ls_upld WHERE id = lv_id.
  IF ls_upld IS NOT INITIAL.
    ls_stream-value = ls_upld-value.
    ls_stream-mime_type = ls_upld-mimetype.
    ls_header-value = 'outline; filename="' && ls_upld-filename && '";'.
    set_header( is_header = ls_header ).



    copy_data_to_ref( EXPORTING is_data = ls_stream   CHANGING  cr_data = er_stream ).
  ENDIF.
endmethod.
```

v. Additional methods from interface /IWBEP/IF_MGW_APPL_SRV_RUNTIME can be redefined. Such as UPDATE_STREAM etc.. I will limit this post to just File creation and retrieval.

Add the new service to the Gateway System:

1. On the gateway system run T-Code /n/iwfnd/maint_service

2. Click “Add Service”
3. Select the System Alias for the system where the OData service was created.
4. Enter the Odata Service name in “Technical Service Name”
5. Click “Get Services”



Filter

System Alias	SV7CLNT100
Technical Service Name	ZFILE_EX_SRV
External Service Name	

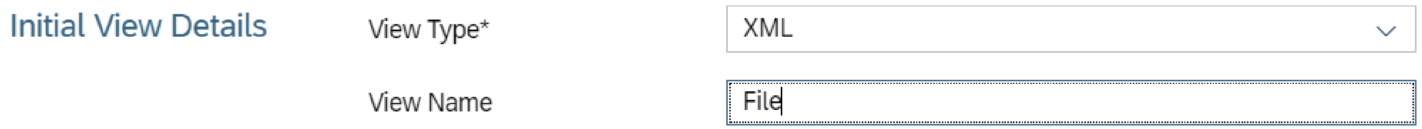
Select Backend Services

□	Type	Technical Service Name
□	BEP	ZFILE_EX_SRV

6. Select the Service and click “Add Selected Services”
7. In the pop up, enter a package assignment and accept the defaults and press the check box.
8. Press back, you should see the service listed it. Select it and chose “Gateway Service Client’
    i. You can test out the service here, there is ample documentation on the Gateway Service Client so I will not go into it here
9. At this point the service is now ready to be consumed in a UI5 App.

Creating the SAPUI5 application via WebIDE on SAP Cloud Platform

1. Login to your SAP Cloud account and navigate to WebIDE
2. Click on File->New->Project From Template
3. Select SAPUI5 Application
4. Give the Project a name “FileUpload” and a namespace i.e “test” and press Next
5. Keep the default XML View type and give the new view a name (or accept the default) Then press “Finish”



Initial View Details

View Type* XML

View Name File

6. Expand the project tree to FileUpload->webapp->View and select your view “File.view.xml”
7. Insert the following code between the <content> and </content> tags to add the UploadCollection to the view.

```
<UploadCollection
  maximumFilenameLength="55"
  maximumFileSize="1000"
  fileDeleted="onFileDeleted"
  filenameLengthExceed="onFilenameLengthExceed"
  fileRenamed="onFileRenamed"
  fileSizeExceed="onFileSizeExceed"
  id="UploadCollection"
  change="onChange"
  mode="SingleSelectMaster"
  beforeUploadStarts="onBeforeUploadStarts"
  items="{path: '/FileSet'}" multiple="true"
  uploadUrl="/sap/opu/odata/sap/ZFILE_EX_SRV/FileSet"
  uploadComplete="onUploadComplete" noDataText="No files found."
  noDataDescription="Drop files to upload, or use the &quot;+&quot; button.">
  <items>
    <UploadCollectionItem
      documentId="{ID}"
      fileName="{FileName}"
      url="/sap/opu/odata/sap/ZFILE_EX_SRV/FileSet(guid'{ID}')/$value"
      mimeType="{MIMEType}"
      enableEdit="false"
      enableDelete="false"
      visibleDelete="false"
      visibleEdit="false">
    </UploadCollectionItem>
  </items>
</UploadCollection>
```

8. In the controller folder open File.controller.js

   i. Add the following code to the init method

```
onInit: function () {
   var oUploadCollection = this.getView().byId('UploadCollection');
   oUploadCollection.setUploadUrl("/sap/opu/odata/sap/ZFILE_EX_SRV/FileSet");
   var oModel = new sap.ui.model.odata.ODataModel("/sap/opu/odata/sap/ZFILE_EX_SRV",false);
   this.getView().setModel(oModel);
}
```

   ii. Create the onBeforeUploadStarts method

```
onBeforeUploadStarts: function (oEvent) {

            var oCustomerHeaderSlug = new sap.m.UploadCollectionParameter({
                name: "slug",
                value: oEvent.getParameter("fileName")
            });
            oEvent.getParameters().addHeaderParameter(oCustomerHeaderSlug);

             var oModel = this.getView().getModel();

            oModel.refreshSecurityToken();
            var oHeaders = oModel.oHeaders;

            var sToken = oHeaders['x-csrf-token'];

            var oCustomerHeaderToken = new sap.m.UploadCollectionParameter({

                name: "x-csrf-token",

                value: sToken

             });
            oEvent.getParameters().addHeaderParameter(oCustomerHeaderToken);

        }
```

iii. Create the onUploadComplete method

```
        onUploadComplete: function(oEvent){
                this.getView().getModel().refresh();
        }
```

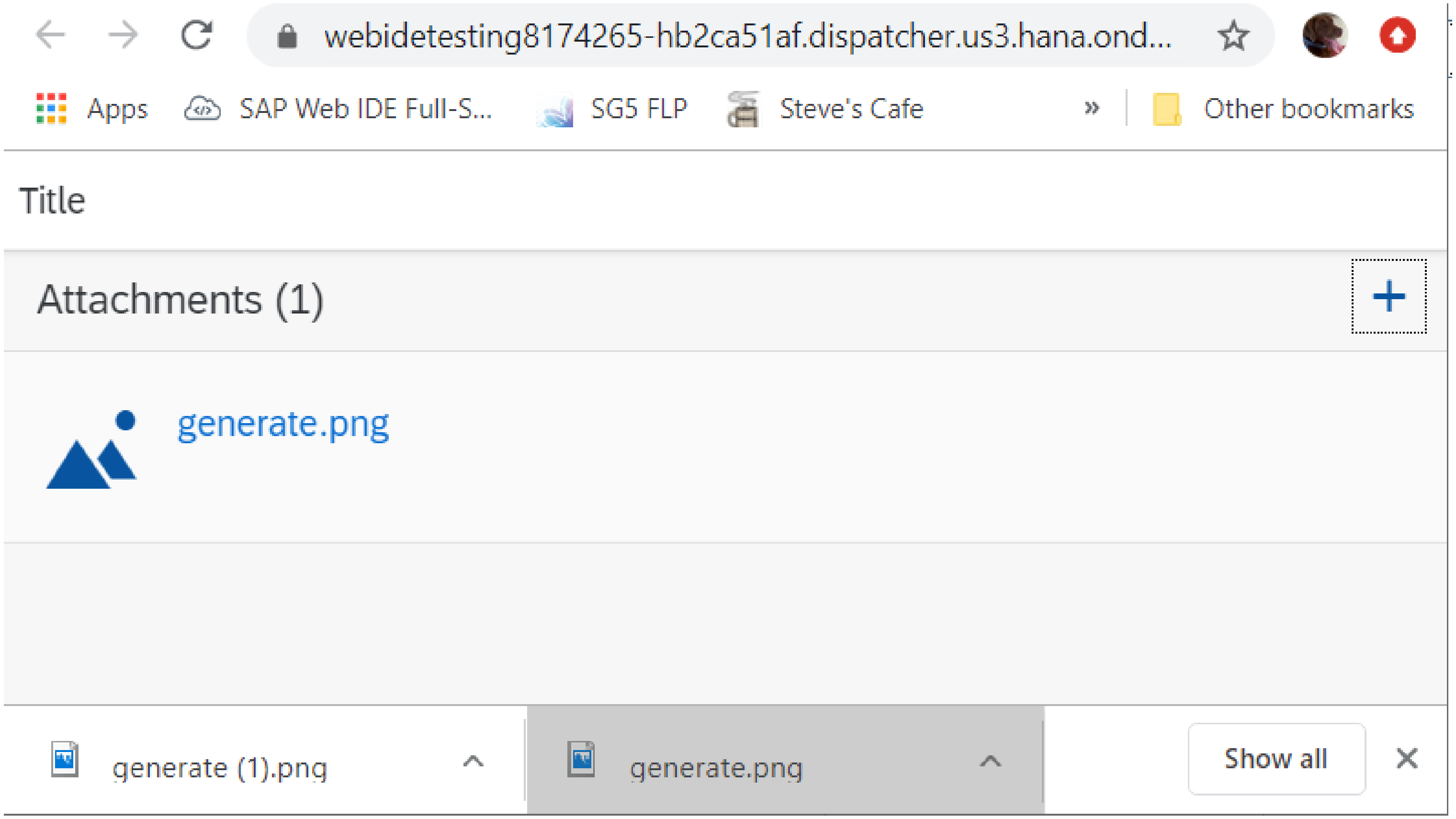
9. For the WebIDE environment you must add a route entry in neo-app.json. This should reference a configured destination on your cloud account to your gateway system. Setting up the Cloud Connector and destinations is outside the scope of this blog. Below is an example entry needed in the “routes” array in the neo-app.json file

```
    {
      "path": "/sap/opu/odata",
      "target": {
        "type": "destination",
        "name": "SG7_OnPremise",
        "entryPath": "/sap/opu/odata"
      },
      "description": "SG7_OnPremise"
    }
```

10. Now setup a dataSources attribute of “sap.app” in manifest.json. The only the “dataSources” entry should be added. the rest of the code is given for context. (This step isn’t 100% necessary for this example)

```
"_version": "1.12.0",
"sap.app": {
    "id": "dsn.File",
    "type": "application",
    "i18n": "i18n/i18n.properties",
    "applicationVersion": {
        "version": "1.0.0"
    },
    "title": "{{appTitle}}",
    "description": "{{appDescription}}",
    "sourceTemplate": {
        "id": "ui5template.basicSAPUI5ApplicationProject",
        "version": "1.40.12"
    },
    "dataSources": {
        "mainService": {
            "uri": "/sap/opu/odata/sap/ZFILE_EXT_SRV/",
            "type": "OData",
            "settings": {
                "annotations": [
                    "ZFILE_EXT_ANNO_MDL"
                ],
                "localUri": "localService/metadata.xml"
            }
        },
        "ZFILE_EXT_ANNO_MDL": {
            "uri": "/sap/opu/odata/IWFND/CATALOGSERVICE;v=2/Annotations(TechnicalName='ZFILE_
            "type": "ODataAnnotation",
            "settings": {
                "localUri": "localService/ZFILE_EXT_ANNO_MDL.xml"
            }
        }
    }
},
```

11. Run the project via index.html and you should be able to upload and download files



Conclusion

This blog is intended to give a bare minimum to get you up and running uploading and downloading files. There are numerous improvements that could be made. I welcome all suggestions for improvement or questions. Thanks for reading.

Alert Moderator

Assigned tags

SAPUI5

NW ABAP Gateway (OData)

SAP Fiori

SAP Gateway

SAP Web IDE

CREATE_STREAM

GET_STREAM

View more...

Similar Blog Posts

New E2E Tutorial: Gain Hands-on Developer Experience with SAPUI5 and SAP NetWeaver Gateway

By Bertram Ganz Dec 10, 2012

Navigation and Routing Simple Example SAPUI5

By Sk Manirujjaman Feb 20, 2016

SAPUI5 Application Consuming OData service with SAP WEB IDE

By rajeesh o Apr 13, 2016

Related Questions

How to send uploadcollection attached documents to sap using odata in sapui5 project?

By DURAI VENKATESH Jan 05, 2017

Uploadcollection Unable to load existing documents from Odata

By Srinivasu Y Feb 27, 2020

Unable to set UploadUrl of uploadcollection in code

By Lesedi Diamond Damane Aug 30, 2016

4 Comments

You must be Logged on to comment or reply to a post.

Mahesh Kumar Palavalli

December 30, 2019 at 4:59 am

Good End-End example Paul !!!. Couple of small suggestions.

1. You can use odata v2 model instead of odata model as it is deprecated.
2. As I can see that you are using upload collection, you can go with the attachment servicewhich will make the development a lot easier (Assuming that you are working in s4hana system, it should be available) for older ones your's is a good approach.

Thanks,

Mahesh

Like 1 | Share

Paul McFarling | Blog Post Author

December 30, 2019 at 5:18 pm

Thanks! I was unaware of the attachment service. I do intend to use this with a draft enabled list report so it appears the attachment service might work better.

Like 1 | Share

pavan kumar Mallipeddi

May 18, 2020 at 10:35 am

Hi Paul McFarling

How can we use the MIME type and Value in back end to attach the documents to Email.

Thanks,

Pavan

Like 0 | Share

Danil Badykshanov

August 25, 2021 at 3:41 pm

I found a little bug.

It is better to use the following code, otherwise there will be an error

```
var oCustomerHeaderSlug = new sap.m.UploadCollectionParameter({
  name: "slug",
  value: encodeURIComponent(oEvent.getParameter("fileName"))
});
```

Like 0 | Share