**OData - Media File Handling**

**Structure Creation**

Create a Table for the ODATA Entity to save the file udaipur by following the below.

Go to the TCODE SE11->Database Tables->Click on Create Button

A screenshot of a computer

Description automatically generated

Click Enter now.

A screenshot of a computer

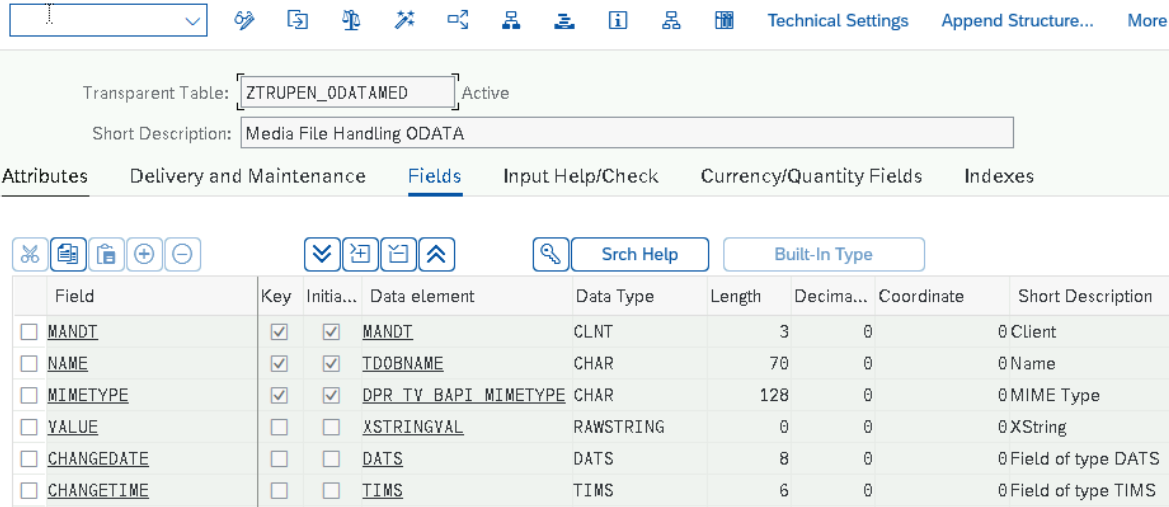
Description automatically generated

Maintain the Delivery Class as A and Table view editing as Display/Maintenance Allowed

A screenshot of a computer

Description automatically generated

Add the Below Fields in to the Table and click on save and activate the table.



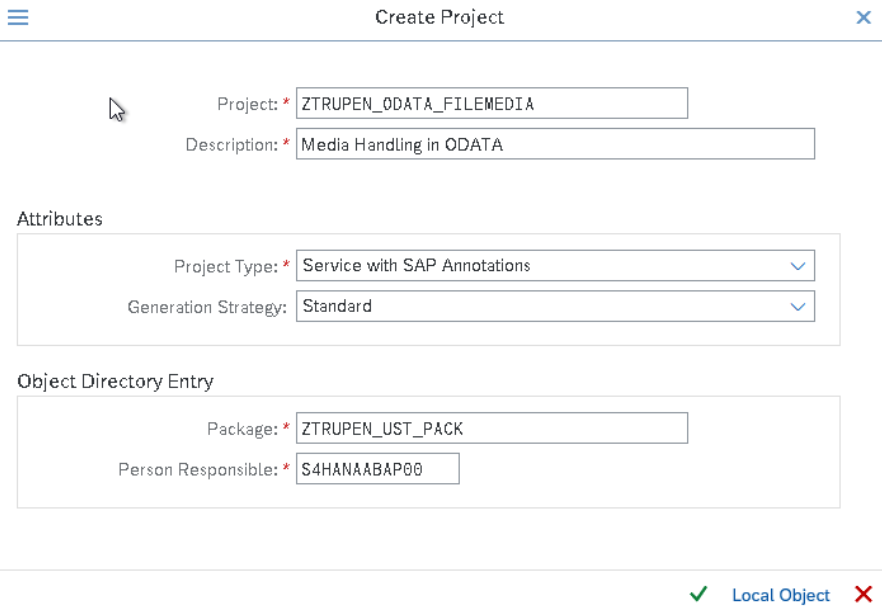
Create the Gateway Client through the Transaction SEGW.

Click on Create Project Button

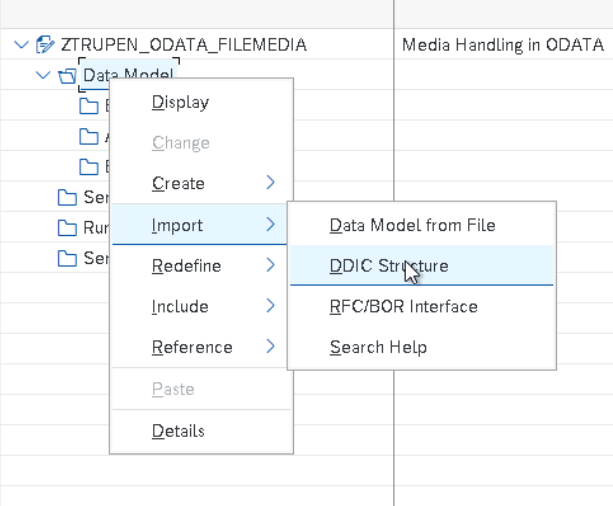
A screenshot of a computer

Description automatically generated

Provide the Project Name, Description and the Package Name. Click on Enter to Create the Project.



Place the Cursor on the Data Model and Right Click->Click on Import->DDIC Structure to create the Entity type and Entity sets.



Now Provide the Name for the Entity Type, Mention the Newly Created Table ZTRUPEN\_ODATAMED to the ABAP Structure and Check the Create Default Entity Set and Click on Next Button.

A screenshot of a computer

Description automatically generated

Select the Required fields for the Entity Structure from the list of fields selected from the Table and click in Next.

A screenshot of a computer

Description automatically generated

Choose the Key Fields for the Entity type and Click on Finish button.

A screenshot of a computer

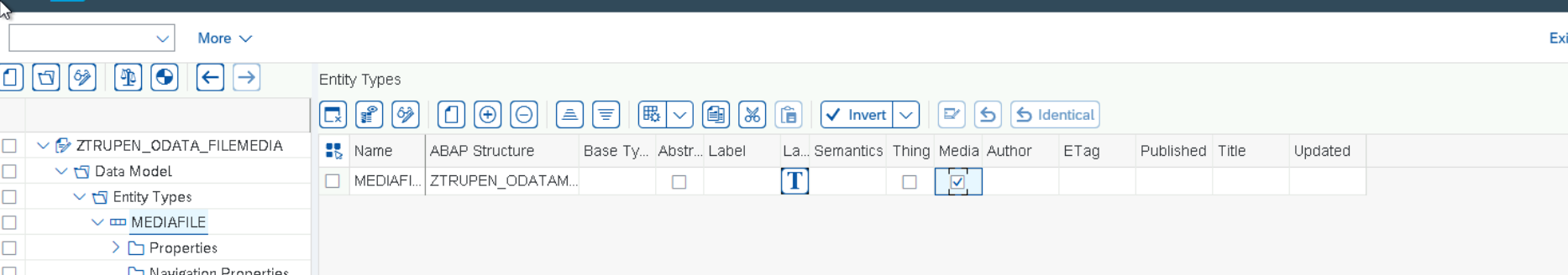
Description automatically generated

Now the Below Screen will appear and we can see the list of fields created as a properties for the Etity types and the Corresponding Sets also created.

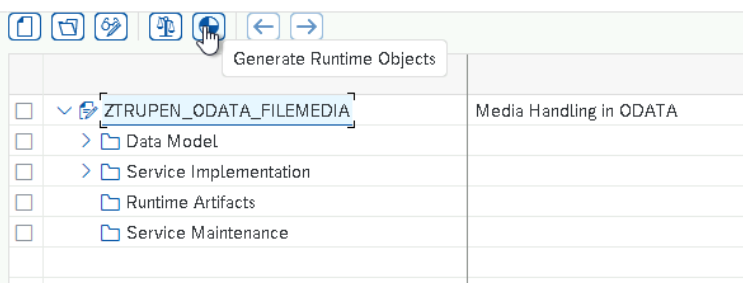
A screenshot of a computer

Description automatically generated

Now Doble Click on newly created Entity types MEDIAFILE and check the Media field to indicate that this entity is used for handling the Image.



Now Click on Generate Run time Objects to generate the Classes for the ODATA Object.



Enter the TR Number for the Service Builder and Click on Enter.A screenshot of a computer program

Description automatically generated

System proposes the Class name, Model Name and Service name here by default and we can modify it to the Customer Namespace if required. Now Click on Enter to complete the Runtime Object Generation.

A screenshot of a computer

Description automatically generated

Now Enter the Package Name and Transport to save the generated Run time Objects.

A screenshot of a computer

Description automatically generated

Now we could see the Below Screen which shows the Messages along with the Class names, Model name and service name generated for the ODATA Service.

A screenshot of a computer

Description automatically generated

**MODEL Provider Class Enhancement**

Now Double click on Runtime Objects and Double click on Model provider extension class ends with \_MPC\_EXT to redefine the Define Method as we need to Enhance the Property for the Entity.

A screenshot of a computer

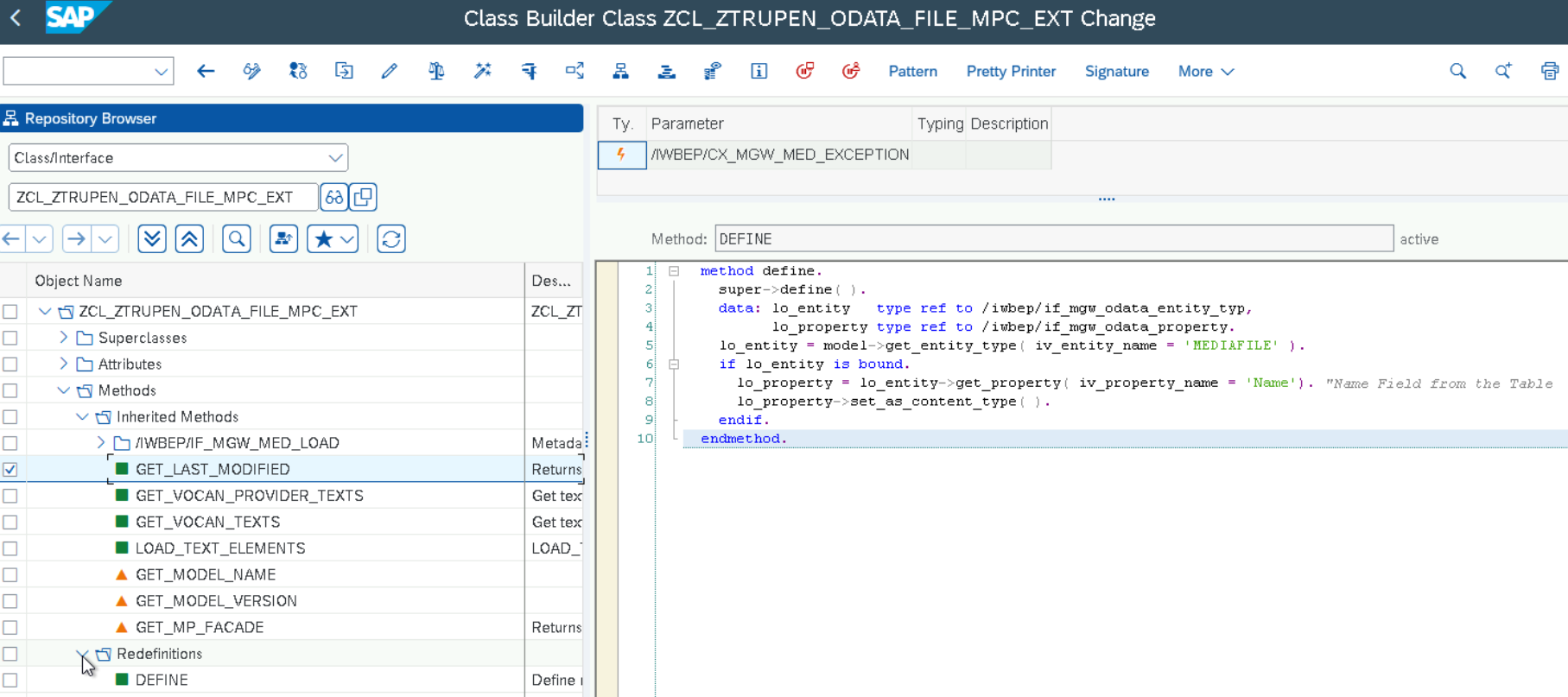
Description automatically generated

Now the Below Screen will be appeared and Expand the Methods -> Inherited Methods -> /IWBEP/IF\_MGW\_MED\_LOAD -> Right Click on the Method DEFINE->Click on Redefine.

A screenshot of a computer

Description automatically generated

Now Implement the Define Method with the Below Code and activate. Go back to Gateway service builder.



**Code Snippet:**

 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 = 'MEDIAFILE' ).  
    if lo\_entity is bound.  
      lo\_property = lo\_entity->get\_property( iv\_property\_name = 'Name'). "Name Field from the Table  
      lo\_property->set\_as\_content\_type( ).  
    endif.  
  endmethod.

**Data Provider Class Enhancement**

Enhance the Data provide class to implement the Stream methods for handling Media Files

Go to Gateway Service Builder –>Double Click on Runtime Artifacts->Double click on the Data Provider class extension Ends with \_DPC\_EXT.

A screenshot of a computer

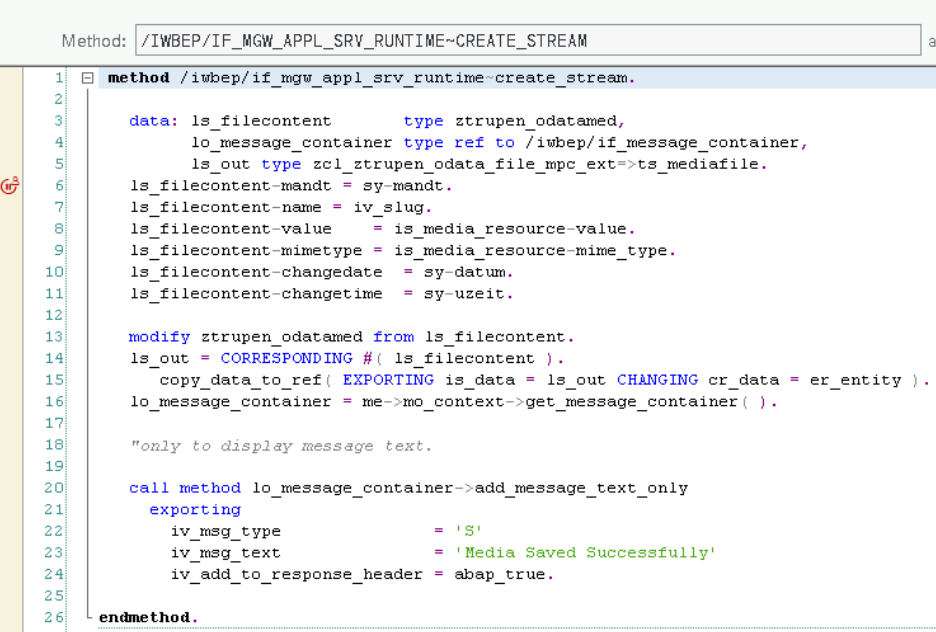
Description automatically generated

Now the Below Screen Will appear and Expand the Methods -> Inherited Methods -> /IWBEP/IF\_MGW\_APPL\_SRV\_RUNTIME -> Right Click on CREATE\_STREAM -> click on Redefine.

A screenshot of a computer

Description automatically generated

Now we can redefine the code and write our own logic to save the Image content to the Table.



Similarly Redefine the Methods UPDATE\_STREAM to modify the Image, DELETE\_STREAM to delete the Image and GET\_STREAM to retrieve the Image from the inherited methods/IWBEP/IF\_MGW\_APPL\_SRV\_RUNTIME and Refine the Method MEDIA\_FILESET\_GET\_ENTITYSET from the inherited method /IWBEP/IF\_MGW\_SOST\_SRV\_RUNTIME.

**Code Snippets**

CREATE\_STREAM

method /iwbep/if\_mgw\_appl\_srv\_runtime~create\_stream.

data: ls\_filecontent type ztrupen\_odatamed,

lo\_message\_container type ref to /iwbep/if\_message\_container,

ls\_out type zcl\_ztrupen\_odata\_file\_mpc\_ext=>ts\_mediafile.

ls\_filecontent-mandt = sy-mandt.

ls\_filecontent-name = iv\_slug.

ls\_filecontent-value = is\_media\_resource-value.

ls\_filecontent-mimetype = is\_media\_resource-mime\_type.

ls\_filecontent-changedate = sy-datum.

ls\_filecontent-changetime = sy-uzeit.

modify ztrupen\_odatamed from ls\_filecontent.

ls\_out = CORRESPONDING #( ls\_filecontent ).

copy\_data\_to\_ref( EXPORTING is\_data = ls\_out CHANGING cr\_data = er\_entity ).

lo\_message\_container = me->mo\_context->get\_message\_container( ).

"only to display message text.

call method lo\_message\_container->add\_message\_text\_only

exporting

iv\_msg\_type = 'S'

iv\_msg\_text = 'Media Saved Successfully'

iv\_add\_to\_response\_header = abap\_true.

endmethod.

**DELETE\_STREAM**

method /iwbep/if\_mgw\_appl\_srv\_runtime~delete\_stream.

data: ls\_filecontent type ztrupen\_odatamed,

lv\_filename type char30,

lo\_message\_container type ref to /iwbep/if\_message\_container.

field-symbols:<fs\_key> type /iwbep/s\_mgw\_name\_value\_pair.

read table it\_key\_tab assigning <fs\_key> WITH KEY name = 'Name'.

if <fs\_key> is assigned.

lv\_filename = <fs\_key>-value.

endif.

select single \* from ztrupen\_odatamed into ls\_filecontent where name = lv\_filename.

if sy-subrc eq 0.

delete ztrupen\_odatamed from ls\_filecontent.

endif.

lo\_message\_container = me->mo\_context->get\_message\_container( ).

"only to display message text.

call method lo\_message\_container->add\_message\_text\_only

exporting

iv\_msg\_type = 'S'

iv\_msg\_text = 'Image Deleted Successfully'

iv\_add\_to\_response\_header = abap\_true.

endmethod.

**UPDATE\_STREAM**

method /IWBEP/IF\_MGW\_APPL\_SRV\_RUNTIME~UPDATE\_STREAM.

data: ls\_filecontent type ztrupen\_odatamed,

lv\_filename type char30,

lo\_message\_container type ref to /iwbep/if\_message\_container.

field-symbols:<fs\_key> type /iwbep/s\_mgw\_name\_value\_pair.

read table it\_key\_tab assigning <fs\_key> WITH KEY name = 'Name'.

if <fs\_key> is assigned.

lv\_filename = <fs\_key>-value.

endif.

ls\_filecontent-mandt = sy-mandt.

ls\_filecontent-name = lv\_filename.

ls\_filecontent-value = is\_media\_resource-value.

ls\_filecontent-mimetype = is\_media\_resource-mime\_type.

ls\_filecontent-changedate = sy-datum.

ls\_filecontent-changetime = sy-uzeit.

modify ztrupen\_odatamed from ls\_filecontent.

lo\_message\_container = me->mo\_context->get\_message\_container( ).

"only to display message text.

call method lo\_message\_container->add\_message\_text\_only

exporting

iv\_msg\_type = 'S'

iv\_msg\_text = 'Image Changed Successfully'

iv\_add\_to\_response\_header = abap\_true.

endmethod.

**GET\_STREAM**

method /iwbep/if\_mgw\_appl\_srv\_runtime~get\_stream.

data : ls\_stream type ty\_s\_media\_resource,

ls\_filecontent type ztrupen\_odatamed,

lv\_filename type char30.

field-symbols:<fs\_key> type /iwbep/s\_mgw\_name\_value\_pair.

read table it\_key\_tab assigning <fs\_key> WITH KEY name = 'Name'.

if <fs\_key> is assigned.

lv\_filename = <fs\_key>-value.

endif.

select single \* from ztrupen\_odatamed into ls\_filecontent where name = lv\_filename.

if sy-subrc eq 0 and ls\_filecontent is not initial.

ls\_stream-value = ls\_filecontent-value.

ls\_stream-mime\_type = ls\_filecontent-mimetype.

copy\_data\_to\_ref( exporting is\_data = ls\_stream changing cr\_data = er\_stream ).

endif.

endmethod.

**MEDIA\_FILESET\_GET\_ENTITYSET**

method MEDIAFILESET\_GET\_ENTITYSET.

SELECT \* FROM ztrupen\_odatamed INTO CORRESPONDING FIELDS OF TABLE et\_entityset.

endmethod.

**SERVICE Activation**

Service has to be activated using the Transaction Code /IWFND/MAINT\_SERVICE.

GO to TCODE /IWFND/MAINT\_SERVICE -> Click on Add Service Button.

A screenshot of a computer

Description automatically generated

Enter the system Alias as Local and filter the technical service name created during the Runtime Generation. Now Select the Service in the Check box and Click on Add Selected Service.

A screenshot of a computer

Description automatically generated

System proposes the Technical service name and technical model Name by default and enter the Package Name and Click on Enter.

A screenshot of a computer

Description automatically generated

Enter the Transport for the Model Metadata and Click on Enter.

A screenshot of a computer

Description automatically generated

Enter the Transport for the ICF Service and Click on Enter

A screenshot of a computer

Description automatically generated

Now the Service is Activated and we need to configure the SICF Service by Expanding the Drop Down in ICF Node -> Click on Configure SICF .

A screenshot of a computer

Description automatically generated

Now the Below screen will appear after clicking the Configure SICF button and double click on the Service now to check the ICF Node.A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

We can see the Service details now and Go in Change Mode and Click on the GUI Configuration Button in the Service Data section to add the parameter for disabling the CSRF Token Validation. (This step depends upon the system configuration and the CSRF Token Validation is enabled at the system level, we are disabling the CSRF token validation at the Service level to test the Gateway service from Gateway Client)

A screenshot of a computer

Description automatically generated

Click on Execute SAP Gateway Client button to test the Service

A screenshot of a service catalog

Description automatically generated

**TESTING**

**Post –** Upload Image

Choose the Entity set MEDIAFILESet on the URI

**URI :** http://172.35.0.26:8000/sap/opu/odata/sap/ZTRUPEN\_ODATA\_FILEMEDIA\_SRV/MEDIAFILESet

Now we need to add the HTTP Parameter SLUG and its value as name of file in the Header.

Note: SLUG is the header parameter is used to pass the key field along with media type in ODATA services for attachments.

Click on Add Header

A screenshot of a computer

Description automatically generated

Input the HTTP Header Parameters now.

A screenshot of a computer

Description automatically generated

Now Click on the Add file button to add the image in to the HTTP Body.

A screenshot of a computer

Description automatically generated

Check the Check Box and Click on Allow.

A screenshot of a computer

Description automatically generated

Now the Image is uploaded and the Content-Type Image/JPEG is added automatically to the HTTP header based on the Image Uploaded and Click on Execute Button to post the image.

A screenshot of a computer

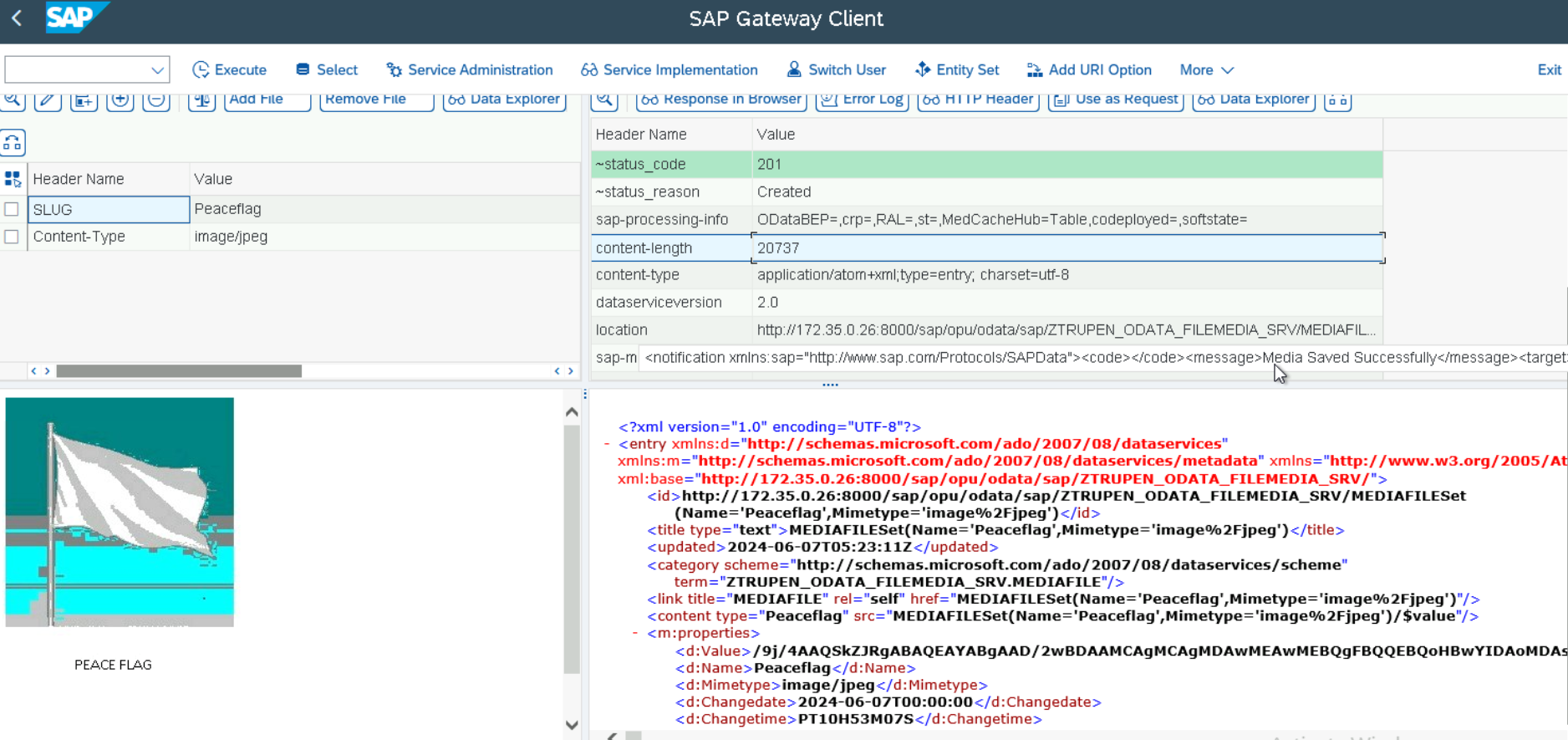
Description automatically generated

Enter the Log on Credentials to execute the URI in the Gateway Client.

A screenshot of a login box

Description automatically generated

**Output**



Z Table entries are added no IN SE11



**Update:PUT**

**URI:** http://172.35.0.26:8000/sap/opu/odata/sap/ZTRUPEN\_ODATA\_FILEMEDIA\_SRV/MEDIAFILESet(Name='Peaceflag',Mimetype='image%2Fjpeg')

Now Click on Add File again and choose different image.

Click on Execute button to post the image now.

A screenshot of a computer

Description automatically generated

Enter the Log on Credentials to execute the URI in the Gateway Client.

A screenshot of a login box

Description automatically generated

Output

A screenshot of a computer

Description automatically generated

**GET - retrieve the image**

**URI:** http://172.35.0.26:8000/sap/opu/odata/sap/ZTRUPEN\_ODATA\_FILEMEDIA\_SRV/MEDIAFILESet(Name='Peaceflag',Mimetype='image%2Fjpeg')/$value

Click on Execute button to retrieve the image now.

**A screenshot of a computer

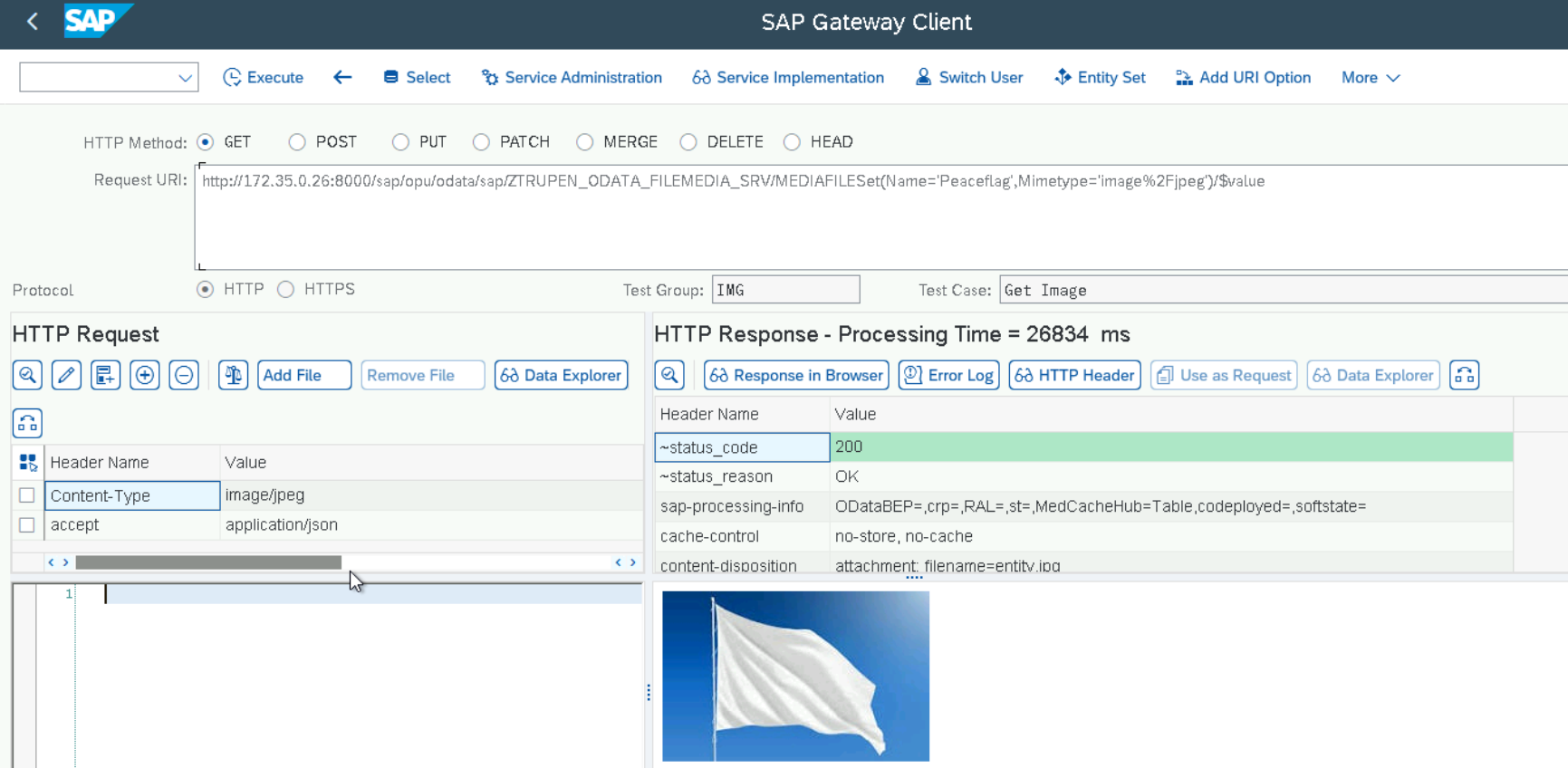
Description automatically generated**

Enter the Log on Credentials to execute the URI in the Gateway Client.

A screenshot of a login box

Description automatically generated

**Output**

****

**DELETE – Delete the image**

**URI:** <http://172.35.0.26:8000/sap/opu/odata/sap/ZTRUPEN_ODATA_FILEMEDIA_SRV/MEDIAFILESet(Name='Peaceflag',Mimetype='image%2Fjpeg')>

**C**lick on Execute button

**A screenshot of a computer

Description automatically generated**

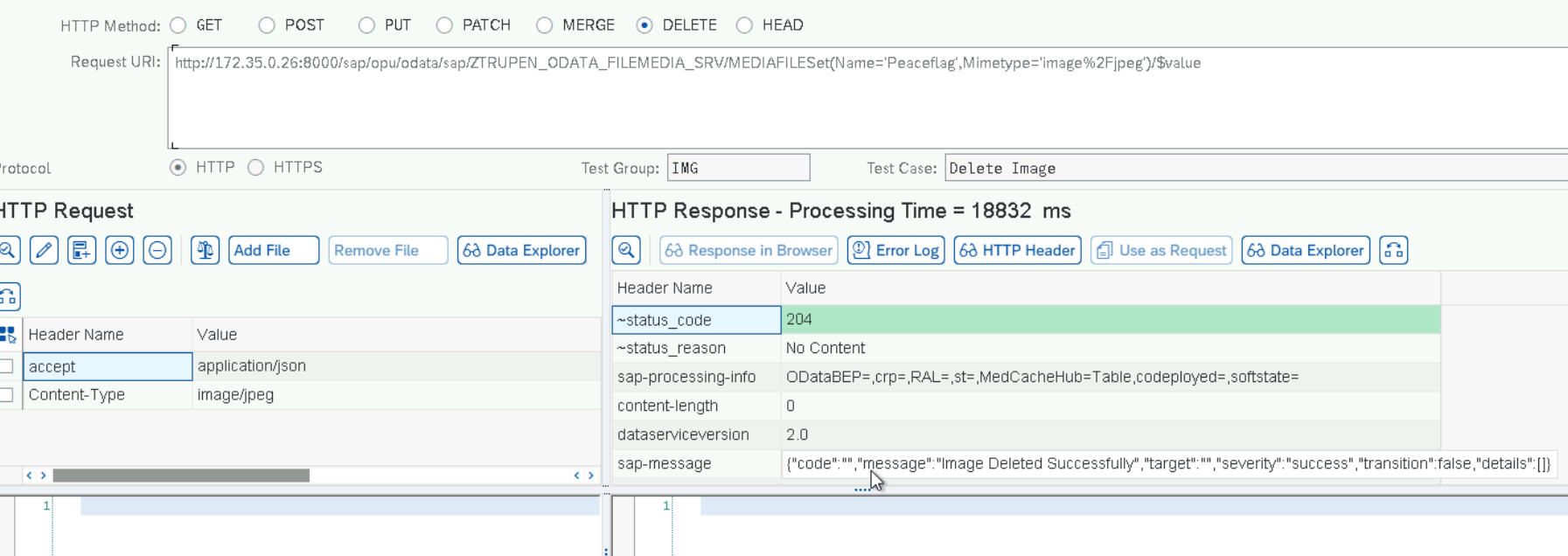
Enter the Log on Credentials to execute the URI in the Gateway Client.

A screenshot of a login box

Description automatically generated

**OUTPUT**

Image Deleted successfully.

****