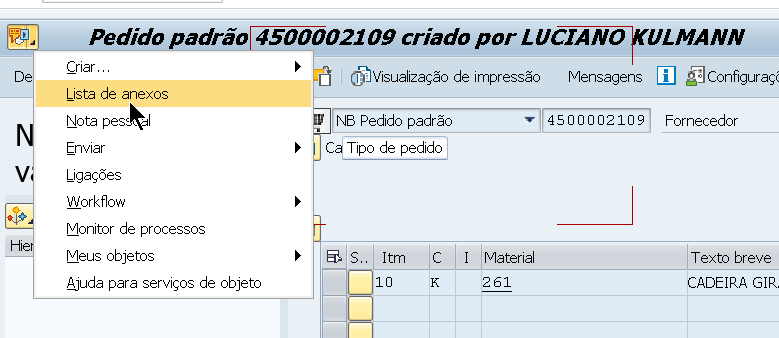
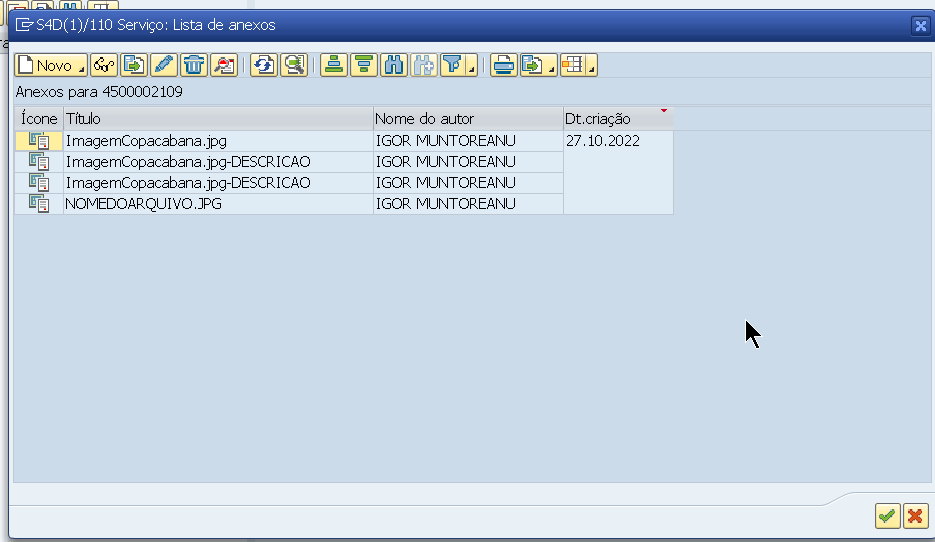
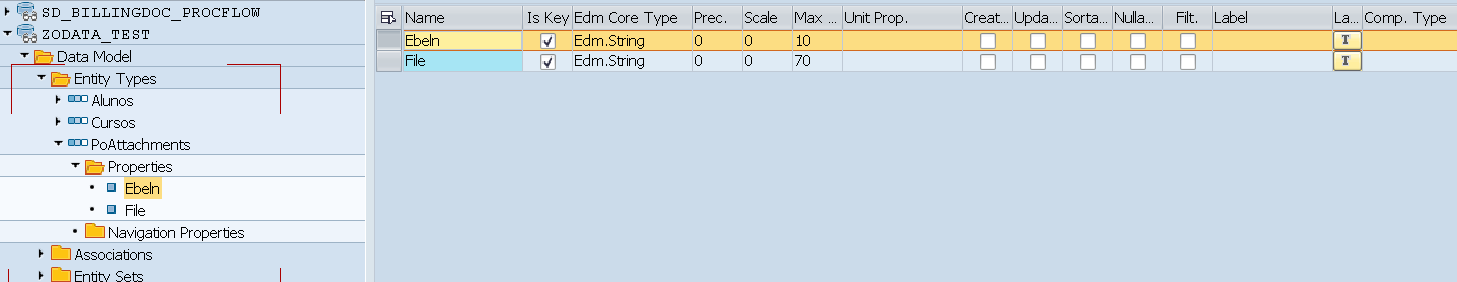
ODATA Media Manipulation:

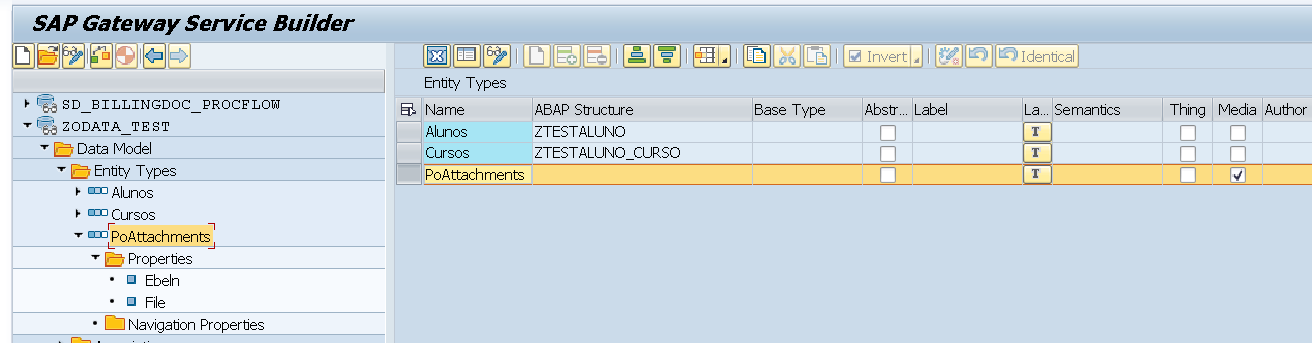
Use-Case:

Read/Add Files to PO in BOS:

SEGW:





MPC\_EXT:

|  |
| --- |
| METHOD define.    super->define( ).    DATA(lo\_entity) = model->get\_entity\_type( iv\_entity\_name = 'PoAttachments' ).    IF lo\_entity IS BOUND.      DATA(lo\_property) = lo\_entity->get\_property( iv\_property\_name = 'File').      lo\_property->set\_as\_content\_type( ).    ENDIF.   ENDMETHOD. |

DPC\_EXT:

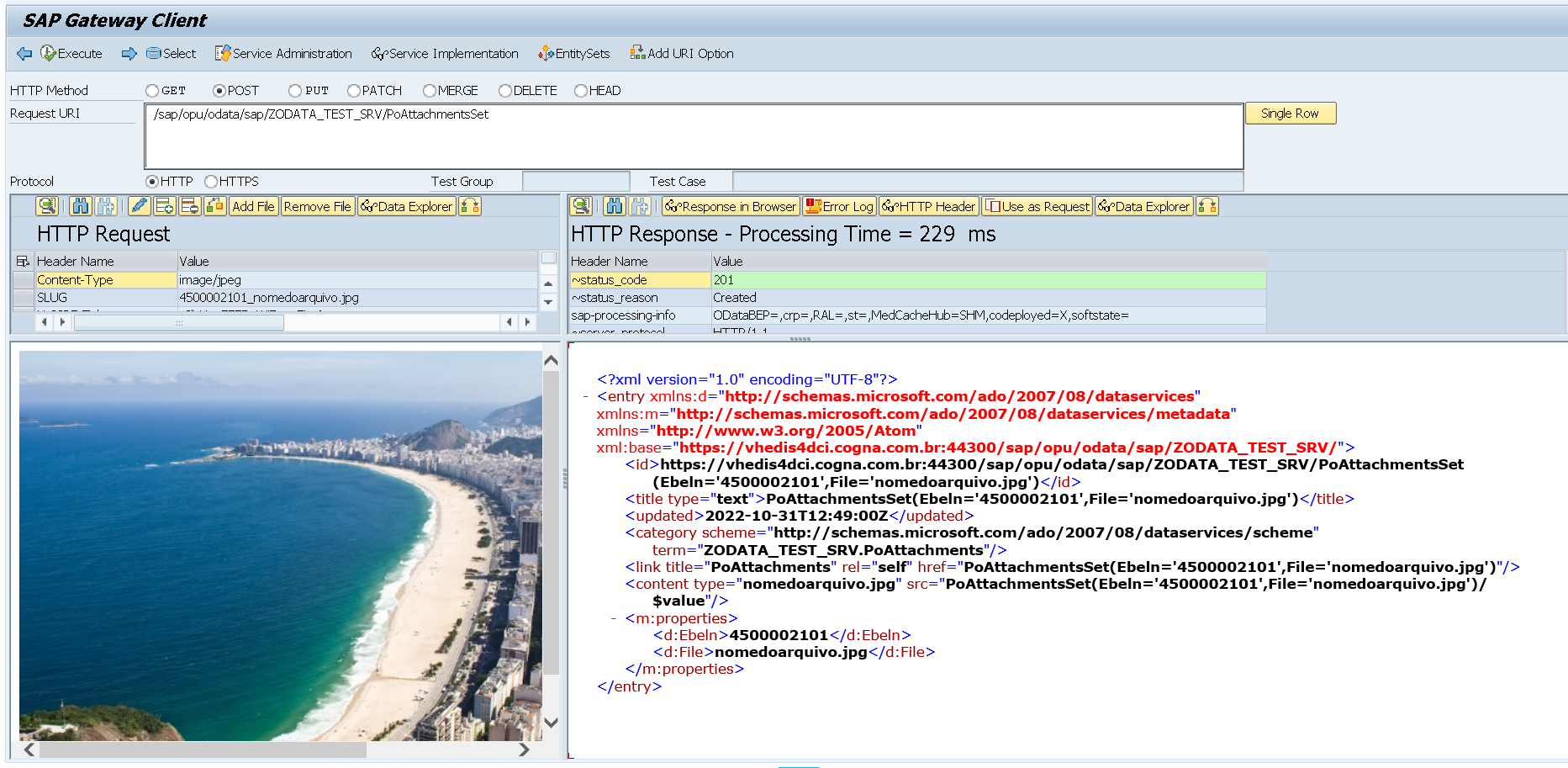
|  |
| --- |
| METHOD /iwbep/if\_mgw\_appl\_srv\_runtime~get\_stream.    DATA: lt\_object\_header   TYPE TABLE OF solisti1,         lt\_object\_content  TYPE TABLE OF  solisti1,         lt\_object\_para     TYPE TABLE OF soparai1,         lt\_object\_parb     TYPE TABLE OF soparbi1,         lt\_attachment\_list TYPE TABLE OF soattlsti1,         lt\_receiver\_list   TYPE TABLE OF soreclsti1,         lt\_contents\_hex    TYPE TABLE OF  solix.    DATA: ls\_sofolenti1 TYPE sofolenti1,         lv\_mime\_type  TYPE mimetypes-type,         lv\_extension  TYPE w3fileext.    DATA: lv\_data TYPE xstring. \*   DATA : ls\_stream TYPE ty\_s\_media\_resource,          lv\_ebeln  TYPE ekko-ebeln,          lv\_file   TYPE c LENGTH 70.     READ TABLE it\_key\_tab ASSIGNING FIELD-SYMBOL(<fs\_key>) WITH KEY name = 'Ebeln'.   CHECK sy-subrc = 0.   lv\_ebeln = <fs\_key>-value.    READ TABLE it\_key\_tab ASSIGNING <fs\_key> WITH KEY name = 'File'.   CHECK sy-subrc = 0.   lv\_file = <fs\_key>-value. \*\*\* Passo 1: Le arquivo com  base no ID do mesmo   CALL FUNCTION 'SO\_DOCUMENT\_READ\_API1'     EXPORTING       document\_id                = CONV so\_entryid( lv\_file ) \*     FILTER                     = 'X '     IMPORTING       document\_data              = ls\_sofolenti1     TABLES       object\_header              = lt\_object\_header       object\_content             = lt\_object\_content       contents\_hex               = lt\_contents\_hex     EXCEPTIONS       document\_id\_not\_exist      = 1       operation\_no\_authorization = 2       x\_error                    = 3       OTHERS                     = 4.   IF sy-subrc = 0. \*\*\* Passo 2: Transforma tabela de binários em XSTRING     CALL FUNCTION 'SCMS\_BINARY\_TO\_XSTRING'       EXPORTING         input\_length = CONV i( ls\_sofolenti1-doc\_size )       IMPORTING         buffer       = lv\_data       TABLES         binary\_tab   = lt\_contents\_hex       EXCEPTIONS         failed       = 1         OTHERS       = 2.     IF sy-subrc = 0.     ENDIF.  \*\*\* Passo 3: Coleta MIMETYPE para exibicao correta do arquivo     lv\_extension = ls\_sofolenti1-obj\_type.     CALL FUNCTION 'SDOK\_MIMETYPE\_GET'       EXPORTING         extension = lv\_extension       IMPORTING         mimetype  = lv\_mime\_type.       ls\_stream-value = lv\_data.     ls\_stream-mime\_type = lv\_mime\_type. \*\*\* Passo 4: Exporta resposta do servico     copy\_data\_to\_ref( EXPORTING is\_data = ls\_stream   CHANGING  cr\_data = er\_stream ).    ENDIF.  ENDMETHOD. |

|  |
| --- |
| METHOD /iwbep/if\_mgw\_appl\_srv\_runtime~create\_stream.    DATA: lv\_folder         TYPE soodk,         lv\_file           TYPE xstring,         lv\_size           TYPE i,         lv\_doctype        TYPE soodk-objtp,         lt\_object\_content TYPE TABLE OF  solisti1,         ls\_docinfo        TYPE sofolenti1,         ls\_docdata        TYPE sodocchgi1,         lv\_ebeln          TYPE ekko-ebeln,         lv\_filename       TYPE c LENGTH 70,         lo\_data           TYPE REF TO data.    DATA: ls\_object\_a TYPE sibflporb,         ls\_object\_b TYPE sibflporb.  \*\*\* Passo 1: Separa Pedido do nome do arquivo   SPLIT iv\_slug AT '\_' INTO lv\_ebeln lv\_filename. \*\*\* Passo 2: Coleta Pasta central onde arquivo é salvo.   CALL FUNCTION 'SO\_FOLDER\_ROOT\_ID\_GET'     EXPORTING       region    = 'B'     IMPORTING       folder\_id = lv\_folder. \*\*\* Passo 3: Transforma o arquivo de XString em uma tabela de Binarios   CALL FUNCTION 'SCMS\_XSTRING\_TO\_BINARY'     EXPORTING       buffer          = is\_media\_resource-value       append\_to\_table = 'X'     IMPORTING       output\_length   = lv\_size     TABLES       binary\_tab      = lt\_object\_content. \*\*\* Passo 4: Insere Arquivo no sistema   ls\_docdata-obj\_name  = lv\_filename.   ls\_docdata-obj\_descr = lv\_filename.   ls\_docdata-doc\_size  = lv\_size.    lv\_doctype = 'BIN'. "Binário.    CALL FUNCTION 'SO\_DOCUMENT\_INSERT\_API1'     EXPORTING       folder\_id                  = lv\_folder       document\_data              = ls\_docdata       document\_type              = lv\_doctype     IMPORTING       document\_info              = ls\_docinfo     TABLES       object\_content             = lt\_object\_content     EXCEPTIONS       folder\_not\_exist           = 1       document\_type\_not\_exist    = 2       operation\_no\_authorization = 3       parameter\_error            = 4       x\_error                    = 5       enqueue\_error              = 6       OTHERS                     = 7.   IF sy-subrc = 0. \*\*\* Passo 5: Cria relacao entre o arquivo e o Pedido de Compras     ls\_object\_a-instid = lv\_ebeln.     ls\_object\_a-typeid = 'BUS2012'.     ls\_object\_a-catid  = 'BO'.      ls\_object\_b-instid = ls\_docinfo-doc\_id.     ls\_object\_b-typeid = 'MESSAGE'.     ls\_object\_b-catid = 'BO'.      TRY.         CALL METHOD cl\_binary\_relation=>create\_link           EXPORTING             is\_object\_a = ls\_object\_a             is\_object\_b = ls\_object\_b             ip\_reltype  = 'ATTA'.     ENDTRY.      COMMIT WORK AND WAIT.  \*\*\* Passo 6: Prepara resposta do servico     CREATE DATA lo\_data TYPE zcl\_zodata\_test\_mpc=>ts\_poattachments.     ASSIGN lo\_data->\* TO FIELD-SYMBOL(<fs\_odata>).     CHECK <fs\_odata> IS ASSIGNED.     ASSIGN COMPONENT 'EBELN' OF STRUCTURE <fs\_odata> TO FIELD-SYMBOL(<fs\_data>).     IF <fs\_data> IS ASSIGNED.       <fs\_data> = lv\_ebeln.       UNASSIGN <fs\_data>.     ENDIF.      ASSIGN COMPONENT 'FILE' OF STRUCTURE <fs\_odata> TO <fs\_data>.     IF <fs\_data> IS ASSIGNED.       <fs\_data> = lv\_filename.       UNASSIGN <fs\_data>.     ENDIF.      er\_entity = lo\_data.    ENDIF.  ENDMETHOD. |

|  |
| --- |
| METHOD poattachmentsset\_get\_entityset.    DATA: lr\_ebeln     TYPE RANGE OF ekko-ebeln,         ls\_entityset TYPE LINE OF zcl\_zodata\_test\_igor\_mpc=>tt\_poattachments.    IF it\_key\_tab IS NOT INITIAL.     lr\_ebeln = VALUE #( FOR i = 1 UNTIL i = lines( it\_key\_tab ) + 1 ( sign = 'I' option = 'EQ' low = it\_key\_tab[ i ]-value ) ).   ENDIF. \*\*\* Passo 1: Le todos os Pedidos que contém anexos   SELECT \*   FROM srgbtbrel   INTO TABLE @DATA(lt\_srgbtbrel)   WHERE instid\_a IN @lr\_ebeln     AND reltype   = 'ATTA'     AND typeid\_a = 'BUS2012'.   CHECK sy-subrc = 0.    LOOP AT lt\_srgbtbrel ASSIGNING FIELD-SYMBOL(<fs\_srgbtbrel>).     ls\_entityset-ebeln = <fs\_srgbtbrel>-instid\_a.     ls\_entityset-file = <fs\_srgbtbrel>-instid\_b.     APPEND ls\_entityset TO et\_entityset. CLEAR ls\_entityset.   ENDLOOP.   ENDMETHOD. |

|  |
| --- |
| METHOD poattachmentsset\_get\_entity.   DATA(lv\_ebeln) = VALUE #( it\_key\_tab[ name = 'Ebeln' ]-value OPTIONAL ).   DATA(lv\_file) = VALUE #( it\_key\_tab[ name = 'File' ]-value OPTIONAL ).    SELECT \* UP TO 1 ROWS   FROM srgbtbrel   INTO @DATA(ls\_srgbtbrel)   WHERE instid\_a = @lv\_ebeln     AND instid\_b = @lv\_file     AND reltype = 'ATTA'     AND typeid\_a = 'BUS2012'.     CHECK sy-subrc = 0.   ENDSELECT.    er\_entity-ebeln = ls\_srgbtbrel-instid\_a.   er\_entity-file = ls\_srgbtbrel-instid\_b. ENDMETHOD. |

URI:  
POST: /sap/opu/odata/sap/ZODATA\_TEST\_SRV/PoAttachmentsSet

  
  
GET:

