Odata SEGW projects for Github API's

Monday, April 17, 2023

5:56 PM

OData services Examples

Tuesday, April 18, 2023

12:49 AM

When ever change in entity types - make sure load the metadata again

Odata get expanded entity set example:

<https://blogs.sap.com/2020/01/08/get-expanded-entity-set-get-expanded-entity-sap-odata/>

Example:

***URI :*** /sap/opu/odata/sap/ZDEEPSTRUCTURES\_SRV/HeaderSet?$expand=HeaderToItemNav

*<*[*https://blogs.sap.com/2020/01/08/get-expanded-entity-set-get-expanded-entity-sap-odata/*](https://blogs.sap.com/2020/01/08/get-expanded-entity-set-get-expanded-entity-sap-odata/)*>*

T46 :

/sap/opu/odata/SAP/Z\_CCS\_GITHUB\_FILES\_SRV/GitrepoSet?$expand=GitrepoToTree

Below API is getting called for the odata service : Z\_CCS\_GITHUB\_FILES\_SRV

API to fetch all the folders and files inside folders for a given repository

<https://api.github.com/repos/Devsouravpanda/Sample/git/trees/main?recursive=1>

Odata Filter option example:

<http://www.techippo.com/2016/08/filter-query-option-sap-odata.html>

T46:

/sap/opu/odata/SAP/Z\_CCS\_GITHUB\_FILES\_SRV/GitcommitSet?$filter=Date eq '20230912'

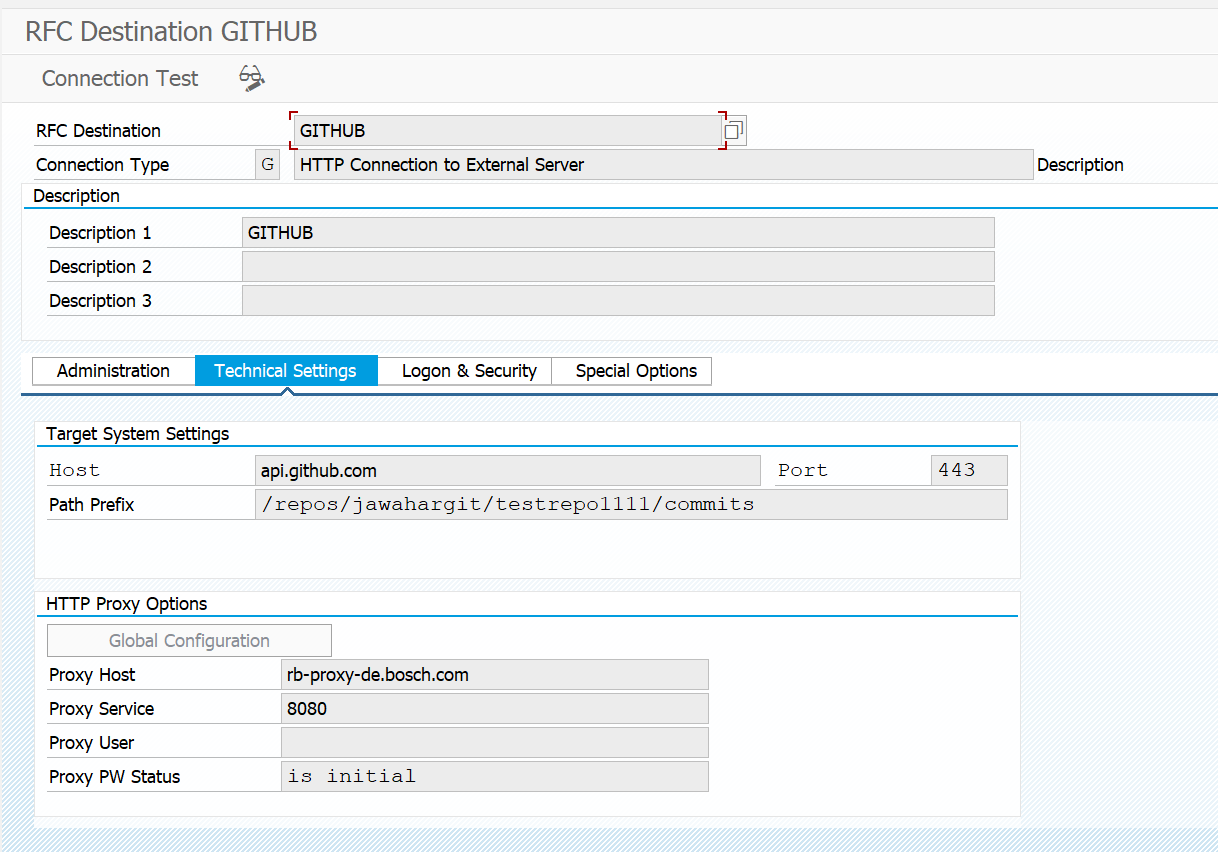
To get roles for a given userid from system

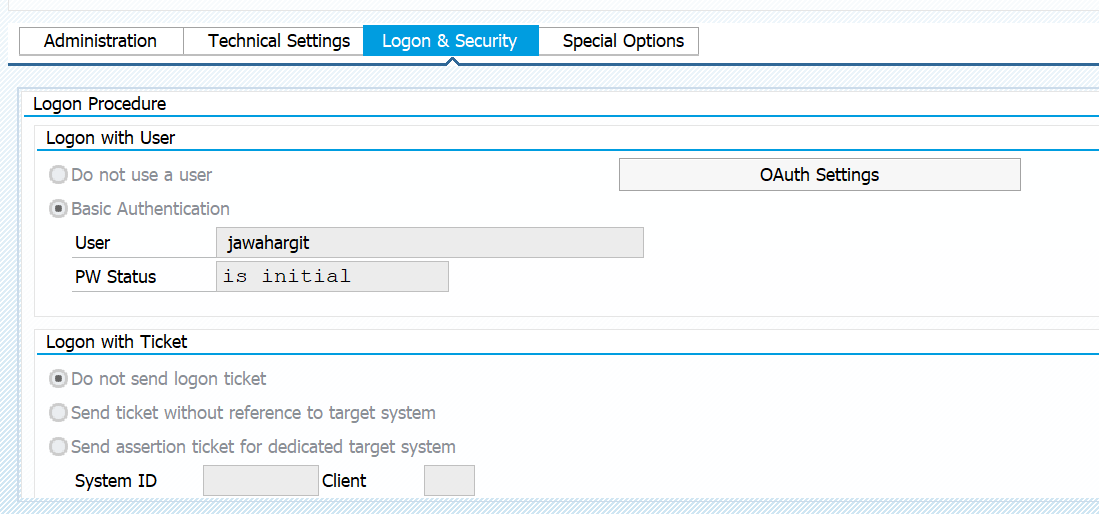
/sap/opu/odata/SAP/Z\_CCS\_ROLES\_SRV/RolesSet?$filter=User eq 'TRNGAB98'

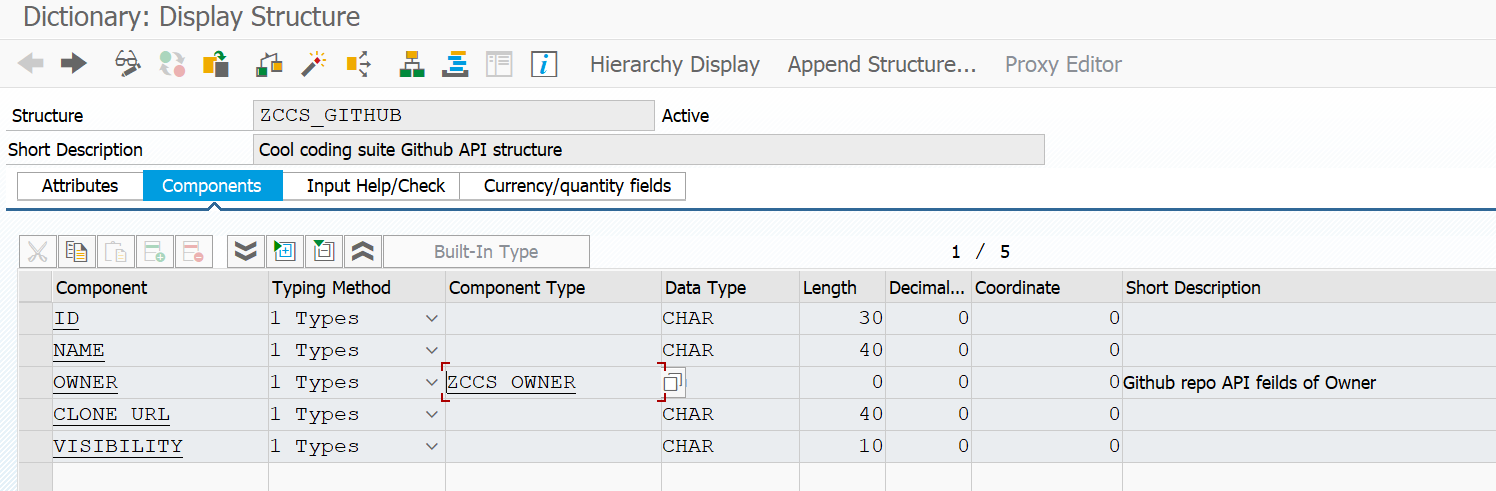
 T46/022 - SEGW : Project name : Z\_CCS\_GITHUB\_REPO

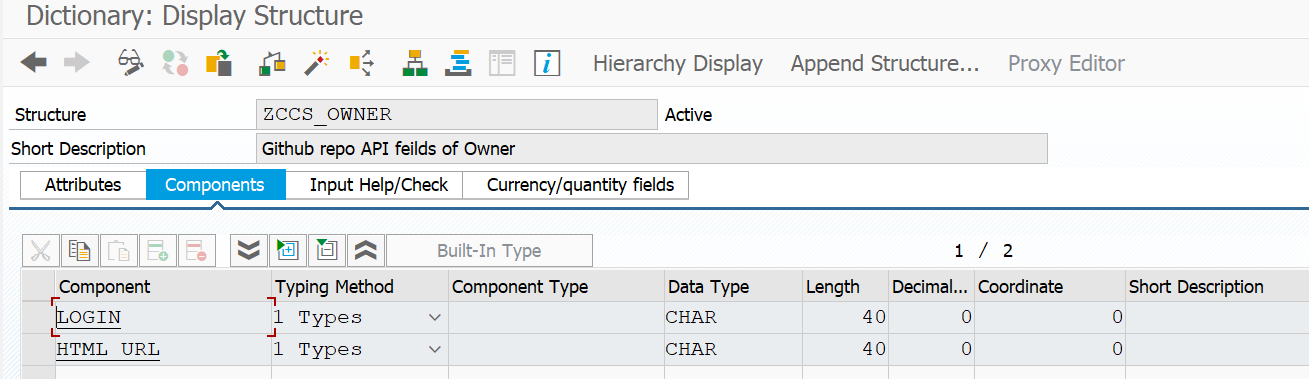
API used : <https://api.github.com/users/jawahargit/repos>

Below structures created in SE11 to match the API json response .

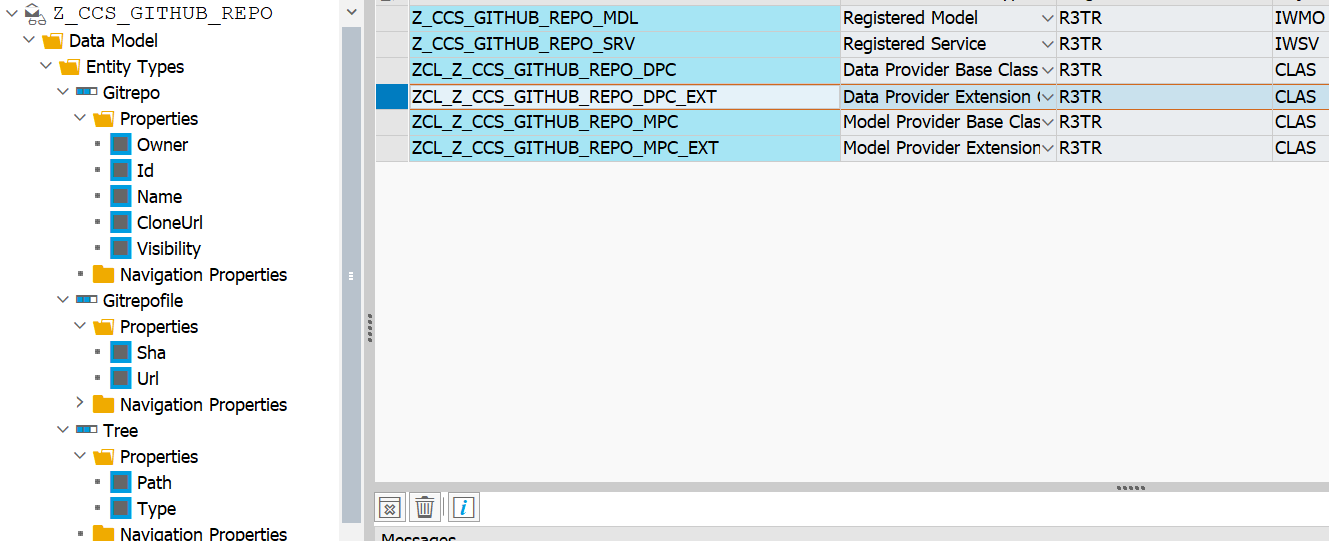








Below enttiy type Gitrepo is created by importing the structure - ZCCS\_GITHUB



METHOD gitreposet\_get\_entityset.

*\*\*TRY.*

*\*CALL METHOD SUPER->GITREPOSET\_GET\_ENTITYSET*

*\*  EXPORTING*

*\*    IV\_ENTITY\_NAME           =*

*\*    IV\_ENTITY\_SET\_NAME       =*

*\*    IV\_SOURCE\_NAME           =*

*\*    IT\_FILTER\_SELECT\_OPTIONS =*

*\*    IS\_PAGING                =*

*\*    IT\_KEY\_TAB               =*

*\*    IT\_NAVIGATION\_PATH       =*

*\*    IT\_ORDER                 =*

*\*    IV\_FILTER\_STRING         =*

*\*    IV\_SEARCH\_STRING         =*

*\*\*    io\_tech\_request\_context  =*

*\*\*  IMPORTING*

*\*\*    et\_entityset             =*

*\*\*    es\_response\_context      =*

*\*    .*

*\*\*  CATCH /iwbep/cx\_mgw\_busi\_exception.*

*\*\*  CATCH /iwbep/cx\_mgw\_tech\_exception.*

*\*\*ENDTRY.*

*\* Declarations*

    DATA lo\_client         TYPE REF TO if\_http\_client.

    DATA lo\_conv           TYPE REF TO cl\_abap\_conv\_in\_ce.

    DATA lo\_rest\_client    TYPE REF TO cl\_rest\_http\_client.

    DATA lo\_response       TYPE REF TO if\_rest\_entity.

    DATA lv\_result\_url     TYPE string.

    DATA lv\_url            TYPE string.

    DATA lv\_json           TYPE string.

    DATA lv\_response       TYPE string.

    DATA lv\_httpcode       TYPE i.

    DATA lv\_reason         TYPE string.

    DATA lv\_message        TYPE string.

    DATA lv\_index          TYPE i.

    DATA lt\_gitrepo      TYPE STANDARD TABLE OF ZCCS\_GITHUB .

    DATA response          TYPE string.

    CALL METHOD cl\_http\_client=>create\_by\_destination

      EXPORTING

        destination              = 'GITHUB'

      IMPORTING

        client                   = lo\_client

      EXCEPTIONS

        argument\_not\_found       = 1

        destination\_not\_found    = 2

        destination\_no\_authority = 3

        plugin\_not\_active        = 4

        internal\_error           = 5

        OTHERS                   = 6.

    IF sy-subrc <> 0.

    ENDIF.

    IF lo\_client IS BOUND.

*\* Send the HTTP request*

      CALL METHOD lo\_client->send

        EXCEPTIONS

          http\_communication\_failure = 1

          http\_invalid\_state         = 2

          http\_processing\_failed     = 3

          http\_invalid\_timeout       = 4

          OTHERS                     = 5.

      IF sy-subrc <> 0.

      ENDIF.

*\* HTTP call receive*

      CALL METHOD lo\_client->receive

        EXCEPTIONS

          http\_communication\_failure = 1

          http\_invalid\_state         = 2

          http\_processing\_failed     = 3

          OTHERS                     = 4.

      IF sy-subrc <> 0.

      ENDIF.

*\*   get status of the response*

      CALL METHOD lo\_client->response->get\_status

        IMPORTING

          code   = lv\_httpcode

          reason = lv\_reason.

*\* Instantiate REST client*

      CREATE OBJECT lo\_rest\_client

        EXPORTING

          io\_http\_client = lo\_client.

*\* Get Response data*

      lo\_response = lo\_rest\_client->if\_rest\_client~get\_response\_entity( ).

*\* Get string data*

      lv\_json = lo\_response->get\_string\_data( ).

*\* Convert string to Abap structures (*

      CALL METHOD /ui2/cl\_json=>deserialize

        EXPORTING

          json        = lv\_json

          pretty\_name = /ui2/cl\_json=>pretty\_mode-camel\_case

        CHANGING

          data        = lt\_gitrepo.

      IF sy-subrc = 0.

        DATA wa\_entityset LIKE LINE OF et\_entityset.

        LOOP AT lt\_gitrepo ASSIGNING FIELD-SYMBOL(<lfs\_gitrepo>).

          wa\_entityset-id = <lfs\_gitrepo>-id.

          wa\_entityset-name = <lfs\_gitrepo>-name. "repo name

          wa\_entityset-owner-login = <lfs\_gitrepo>-owner-login.

          wa\_entityset-owner-html\_url = <lfs\_gitrepo>-owner-html\_url.

          wa\_entityset-clone\_url = <lfs\_gitrepo>-clone\_url.

          wa\_entityset-visibility = <lfs\_gitrepo>-visibility.

          APPEND wa\_entityset TO et\_entityset.

          CLEAR: wa\_entityset.

        ENDLOOP.

      ENDIF.

    ENDIF.

  ENDMETHOD.

 T46/022 - SEGW : Project name : Z\_CCS\_GITHUB\_FILES

API used -

API to fetch all the folders and files inside folders for a given repository

<https://api.github.com/repos/Devsouravpanda/Sample/git/trees/main?recursive=1>

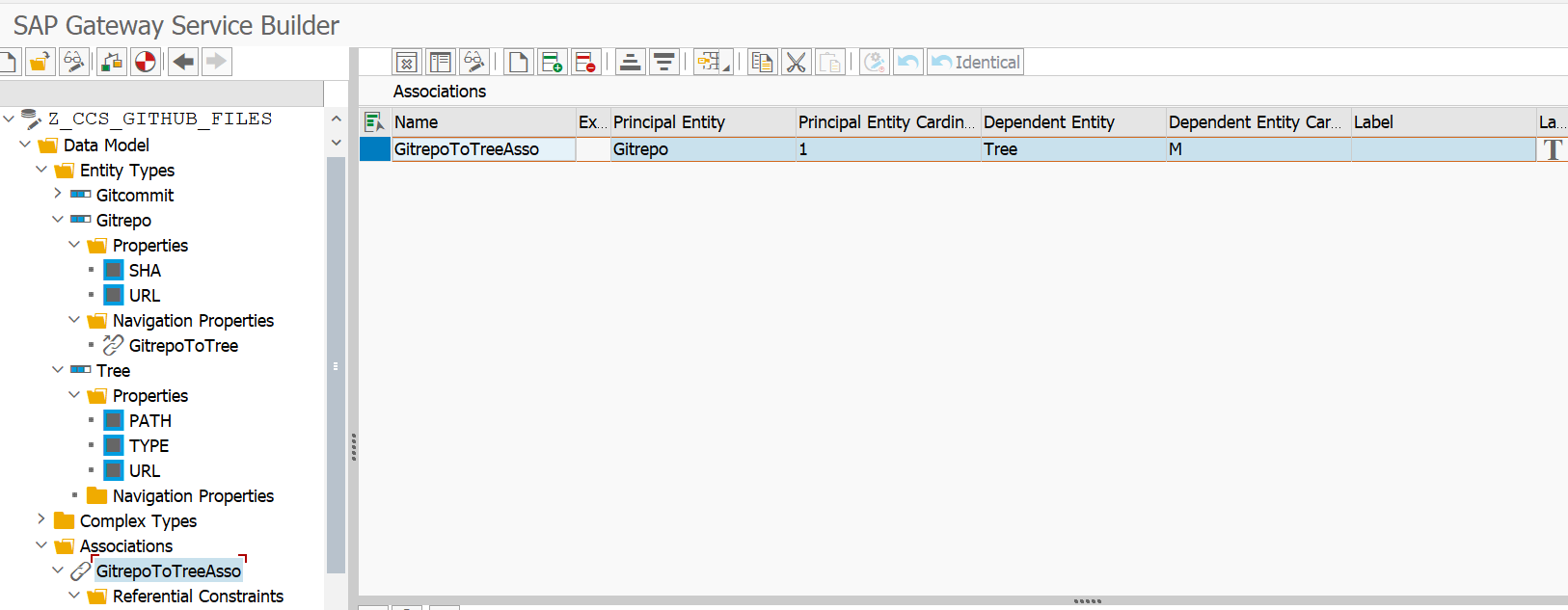
As we can see the Json repsonse is nested table ( see [ ] in reponse

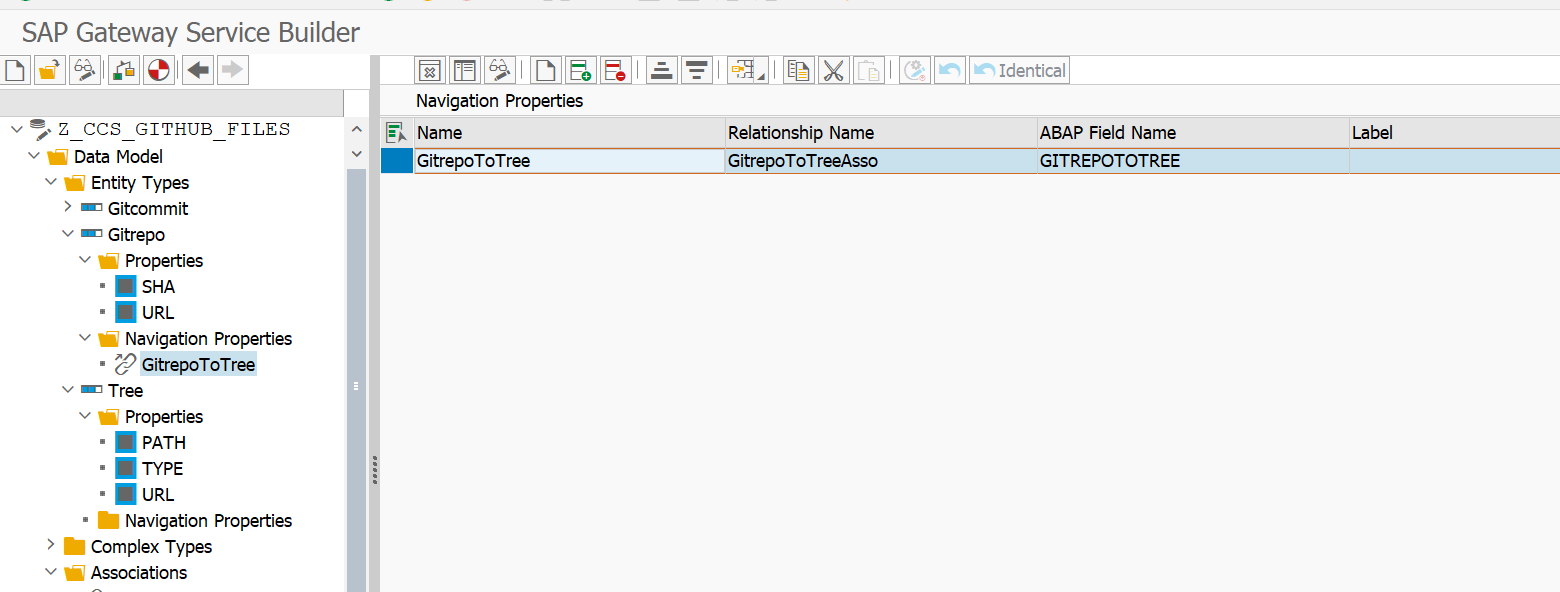
Always consider [ and ] at end of json string in ABAP

And any [ ] in between those are nested internal tables

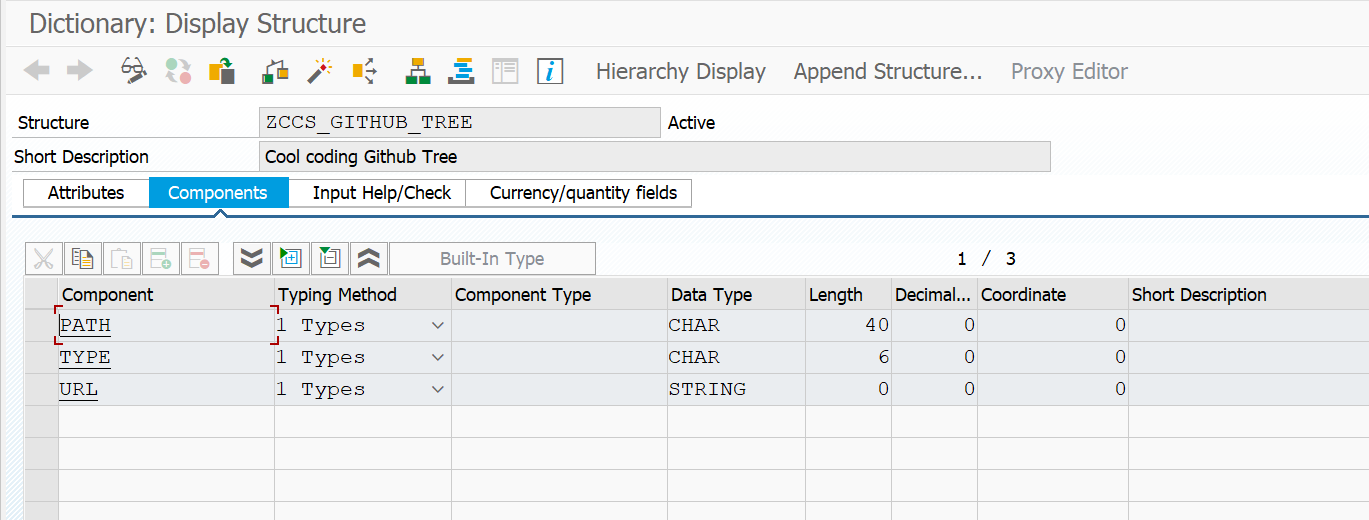
Thats why we go 1 -> many and use get expanded entity ( header -> item )

Association , Navigation concepts comes.





Below structure created in se11 for Tree entity type ( imported the ddic structure)



  METHOD /iwbep/if\_mgw\_appl\_srv\_runtime~get\_expanded\_entityset.

*\*\*TRY.*

*\*CALL METHOD SUPER->/IWBEP/IF\_MGW\_APPL\_SRV\_RUNTIME~GET\_EXPANDED\_ENTITYSET*

*\*\*  EXPORTING*

*\*\*    iv\_entity\_name           =*

*\*\*    iv\_entity\_set\_name       =*

*\*\*    iv\_source\_name           =*

*\*\*    it\_filter\_select\_options =*

*\*\*    it\_order                 =*

*\*\*    is\_paging                =*

*\*\*    it\_navigation\_path       =*

*\*\*    it\_key\_tab               =*

*\*\*    iv\_filter\_string         =*

*\*\*    iv\_search\_string         =*

*\*\*    io\_expand                =*

*\*\*    io\_tech\_request\_context  =*

*\*\*  IMPORTING*

*\*\*    er\_entityset             =*

*\*\*    et\_expanded\_clauses      =*

*\*\*    et\_expanded\_tech\_clauses =*

*\*\*    es\_response\_context      =*

*\*    .*

*\*\*  CATCH /iwbep/cx\_mgw\_busi\_exception.*

*\*\*  CATCH /iwbep/cx\_mgw\_tech\_exception.*

*\*\*ENDTRY.*

    TYPES: tt\_github\_tree TYPE STANDARD TABLE OF zccs\_github\_tree WITH DEFAULT KEY.

    TYPES: BEGIN OF ty\_gitrepo,

             sha           TYPE char40,

             url           TYPE char40,

             GitrepoToTree TYPE tt\_github\_tree, *"GitrepoToTree is Navigation name*

           END OF ty\_gitrepo.

    TYPES: tt\_gitrepo TYPE STANDARD TABLE OF ty\_gitrepo WITH DEFAULT KEY.

*"TYPES : tt\_github\_deep TYPE STANDARD TABLE OF zccs\_github\_deep WITH DEFAULT KEY.*

*"DATA : it\_out\_tab  TYPE tt\_GITHUB\_DEEP.*

*"wa\_out\_tab  LIKE LINE OF it\_out\_tab.*

*\* Declarations*

    DATA lo\_client         TYPE REF TO if\_http\_client.

    DATA lo\_conv           TYPE REF TO cl\_abap\_conv\_in\_ce.

    DATA lo\_rest\_client    TYPE REF TO cl\_rest\_http\_client.

    DATA lo\_response       TYPE REF TO if\_rest\_entity.

    DATA lv\_result\_url     TYPE string.

    DATA lv\_url            TYPE string.

    DATA lv\_json           TYPE string.

    DATA lv\_response       TYPE string.

    DATA lv\_httpcode       TYPE i.

    DATA lv\_reason         TYPE string.

    DATA lv\_message        TYPE string.

    DATA lv\_index          TYPE i.

    DATA lt\_gitrepo        TYPE  tt\_gitrepo.

    DATA lwa\_final  TYPE ty\_gitrepo.

    DATA lt\_final TYPE tt\_gitrepo .

    DATA lwa\_gitrepo TYPE ty\_gitrepo.

    DATA response          TYPE string.

    CASE iv\_entity\_set\_name.

      WHEN 'GitrepoSet'.

        CALL METHOD cl\_http\_client=>create\_by\_destination

          EXPORTING

            destination              = 'GITHUB'

          IMPORTING

            client                   = lo\_client

          EXCEPTIONS

            argument\_not\_found       = 1

            destination\_not\_found    = 2

            destination\_no\_authority = 3

            plugin\_not\_active        = 4

            internal\_error           = 5

            OTHERS                   = 6.

        IF sy-subrc <> 0.

        ENDIF.

        IF lo\_client IS BOUND.

*\* Send the HTTP request*

          CALL METHOD lo\_client->send

            EXCEPTIONS

              http\_communication\_failure = 1

              http\_invalid\_state         = 2

              http\_processing\_failed     = 3

              http\_invalid\_timeout       = 4

              OTHERS                     = 5.

          IF sy-subrc <> 0.

          ENDIF.

*\* HTTP call receive*

          CALL METHOD lo\_client->receive

            EXCEPTIONS

              http\_communication\_failure = 1

              http\_invalid\_state         = 2

              http\_processing\_failed     = 3

              OTHERS                     = 4.

          IF sy-subrc <> 0.

          ENDIF.

          lv\_url = lo\_client->path\_prefix.

          REPLACE ALL OCCURRENCES OF 'repos/' IN lv\_url  WITH ' '.

          CONDENSE lv\_url.

          REPLACE ALL OCCURRENCES OF 'git/trees/main?recursive=1' IN lv\_url  WITH ' '.

          CONDENSE lv\_url.

          lv\_url = '<https://github.com>' && lv\_url.

*\*   get status of the response*

          CALL METHOD lo\_client->response->get\_status

            IMPORTING

              code   = lv\_httpcode

              reason = lv\_reason.

*\* Instantiate REST client*

          CREATE OBJECT lo\_rest\_client

            EXPORTING

              io\_http\_client = lo\_client.

*\* Get Response data*

          lo\_response = lo\_rest\_client->if\_rest\_client~get\_response\_entity( ).

*\* Get string data*

          lv\_json = lo\_response->get\_string\_data( ).

          REPLACE ALL OCCURRENCES OF '"tree"' IN lv\_json WITH '"GitrepoToTree"'.

*"REPLACE ALL OCCURRENCES OF '"Gitrep"' IN lv\_json WITH '"tree"'.*

          lv\_json = '[' && lv\_json && ']' .

*\* Convert string to Abap structures (*

          CALL METHOD /ui2/cl\_json=>deserialize

            EXPORTING

              json        = lv\_json

              pretty\_name = /ui2/cl\_json=>pretty\_mode-camel\_case

            CHANGING

              data        = lt\_gitrepo.

          LOOP AT lt\_gitrepo ASSIGNING FIELD-SYMBOL(<lfs\_gitrepo>).

*\* PROCESS the second level internal table.*

            LOOP AT <lfs\_gitrepo>-GitrepoToTree

              ASSIGNING FIELD-SYMBOL(<lfs\_gittree>).

              IF sy-subrc = 0.

                if <lfs\_gittree>-type ne 'blob'.

                  <lfs\_gittree>-type = 'tree'.

                  endif.

                <lfs\_gittree>-url = lv\_url && <lfs\_gittree>-type && '/main/' && <lfs\_gittree>-path .

              ENDIF.

            ENDLOOP.

          ENDLOOP.

*"MOVE-CORRESPONDING lt\_gitrepo-tree to GitrepoToTree.*

*"MOVE-CORRESPONDING lt\_gitrepo-tree TO er\_entityset-GitrepoToTree.*

          copy\_data\_to\_ref(

        EXPORTING

         is\_data = lt\_gitrepo

        CHANGING

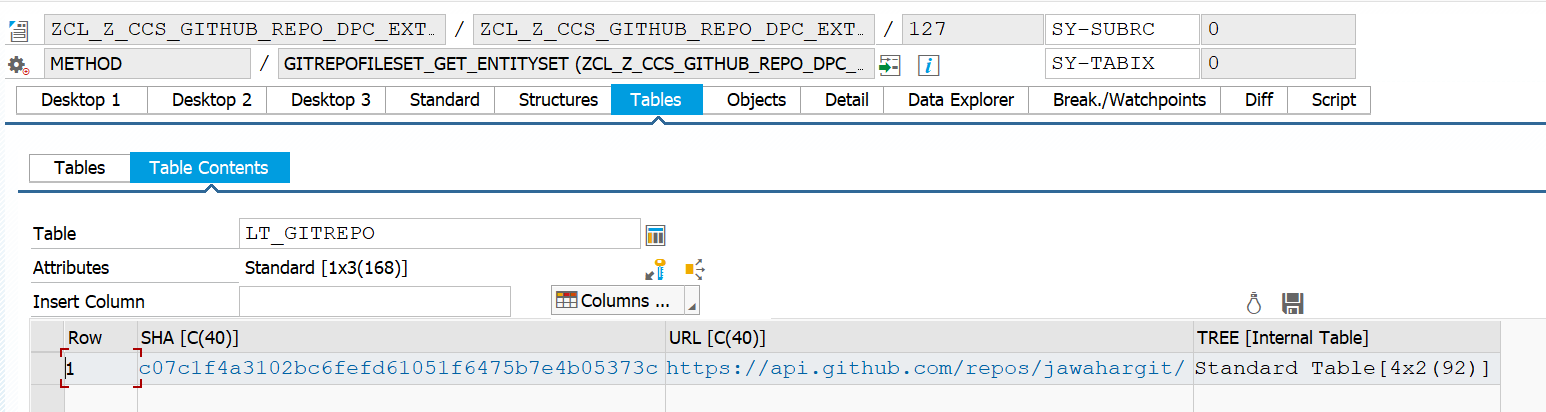
         cr\_data = er\_entityset ).

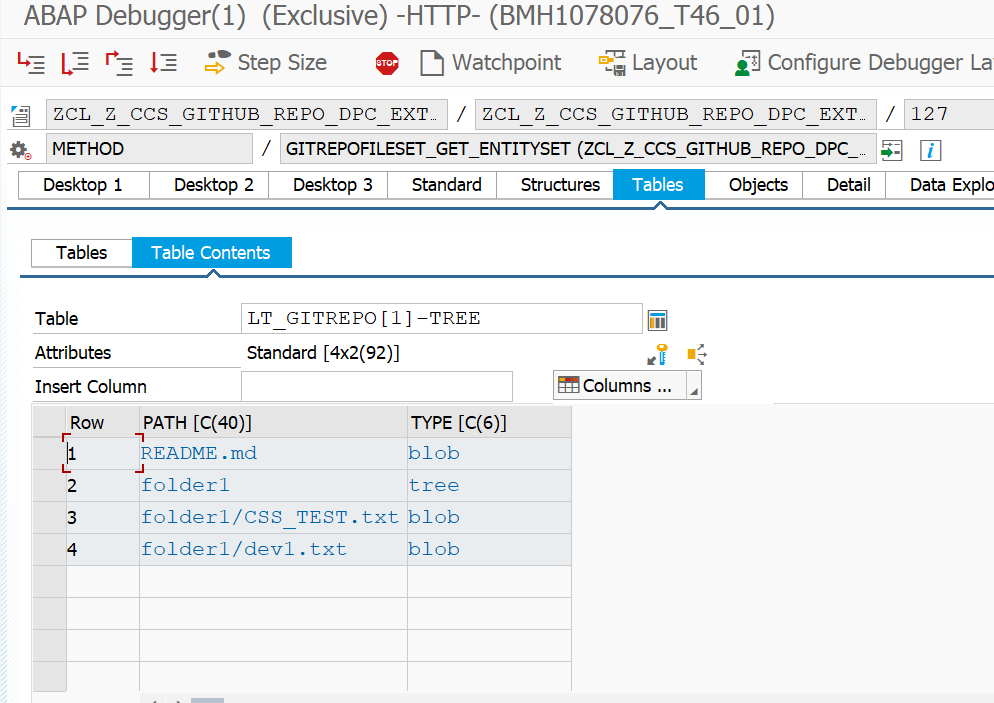
        ENDIF.

    ENDCASE.

  ENDMETHOD.

Debug mode screenshots:



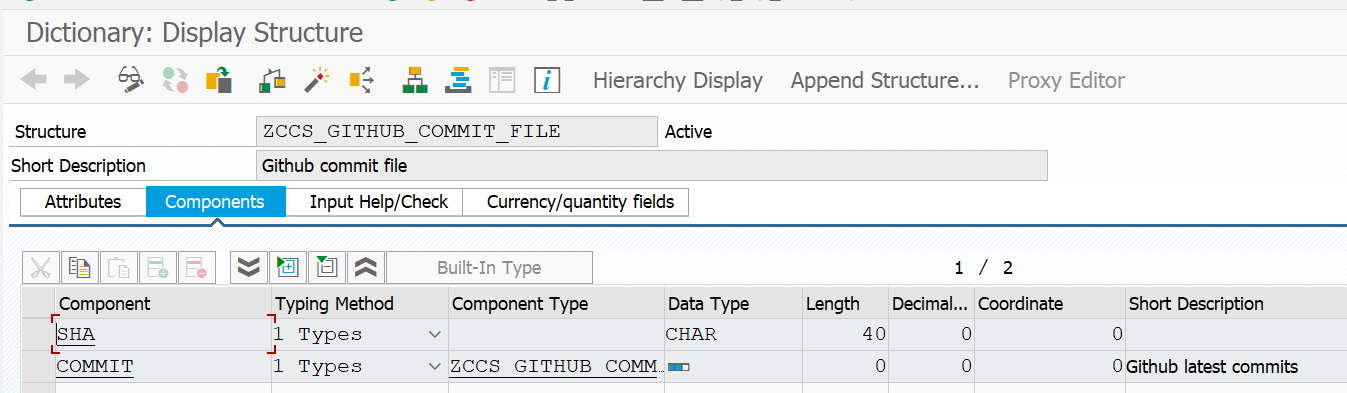


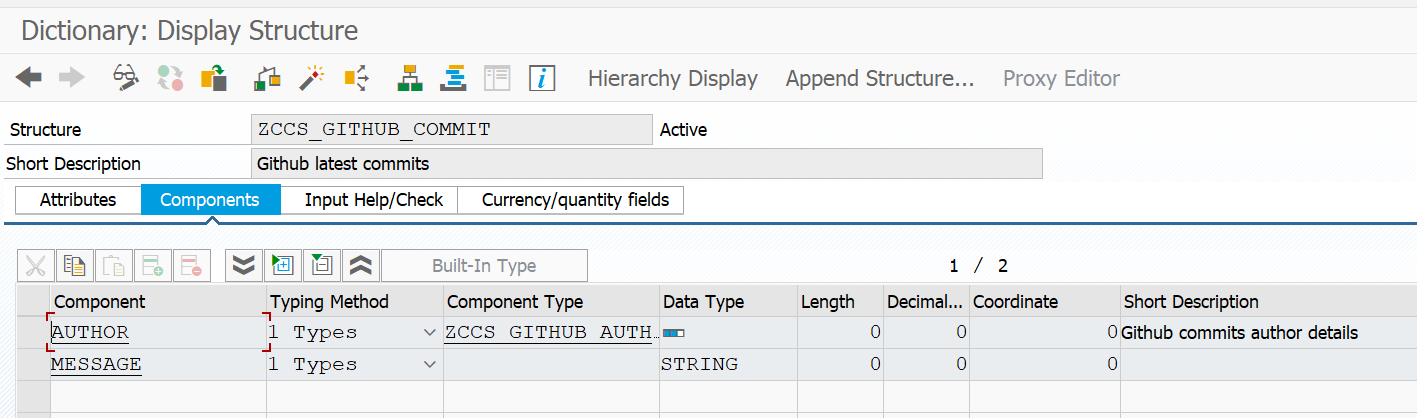
T46/022 - SEGW : Project name : Z\_CCS\_GITHUB\_FILES

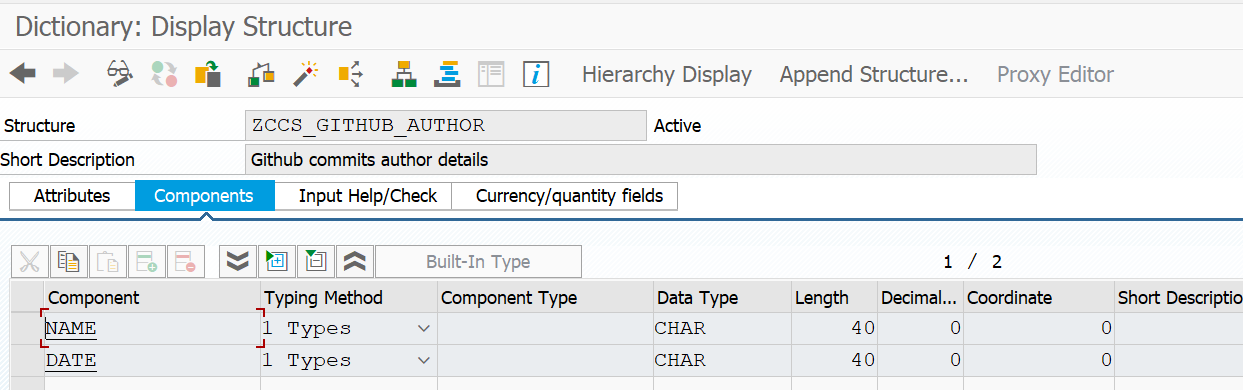
API to fetch latest commit date for a given file/ folder

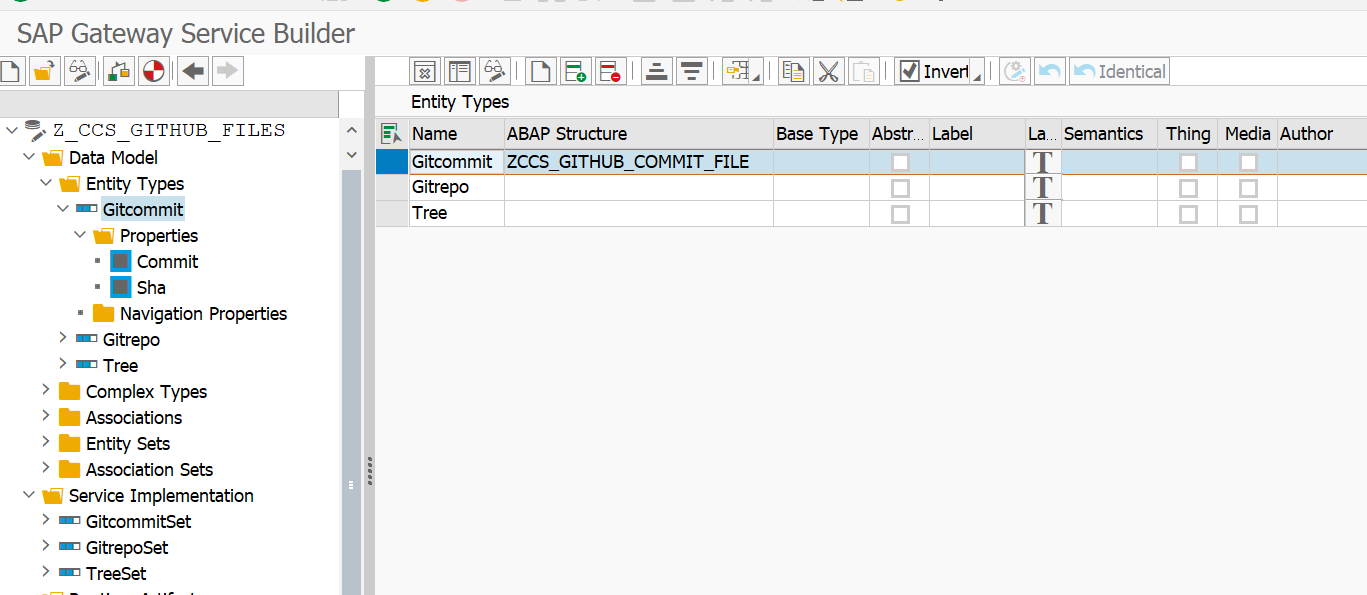
API used : <https://api.github.com/repos/jawahargit/testrepo1111/commits?path=Python>

Structures created in se11









  METHOD gitcommitset\_get\_entityset.

*\*\*TRY.*

*\*CALL METHOD SUPER->GITCOMMITSET\_GET\_ENTITYSET*

*\*  EXPORTING*

*\*    IV\_ENTITY\_NAME           =*

*\*    IV\_ENTITY\_SET\_NAME       =*

*\*    IV\_SOURCE\_NAME           =*

*\*    IT\_FILTER\_SELECT\_OPTIONS =*

*\*    IS\_PAGING                =*

*\*    IT\_KEY\_TAB               =*

*\*    IT\_NAVIGATION\_PATH       =*

*\*    IT\_ORDER                 =*

*\*    IV\_FILTER\_STRING         =*

*\*    IV\_SEARCH\_STRING         =*

*\*\*    io\_tech\_request\_context  =*

*\*\*  IMPORTING*

*\*\*    et\_entityset             =*

*\*\*    es\_response\_context      =*

*\*    .*

*\*\*  CATCH /iwbep/cx\_mgw\_busi\_exception.*

*\*\*  CATCH /iwbep/cx\_mgw\_tech\_exception.*

*\*\*ENDTRY.*

*\* Declarations*

    DATA lo\_client         TYPE REF TO if\_http\_client.

    DATA lo\_conv           TYPE REF TO cl\_abap\_conv\_in\_ce.

    DATA lo\_rest\_client    TYPE REF TO cl\_rest\_http\_client.

    DATA lo\_response       TYPE REF TO if\_rest\_entity.

    DATA lv\_result\_url     TYPE string.

    DATA lv\_url            TYPE string.

    DATA lv\_json           TYPE string.

    DATA lv\_response       TYPE string.

    DATA lv\_httpcode       TYPE i.

    DATA lv\_reason         TYPE string.

    DATA lv\_message        TYPE string.

    DATA lv\_index          TYPE i.

    DATA lt\_git\_commit\_file      TYPE STANDARD TABLE OF zccs\_github\_commit\_file .

    DATA response          TYPE string.

    CALL METHOD cl\_http\_client=>create\_by\_destination

      EXPORTING

        destination              = 'GITHUB'

      IMPORTING

        client                   = lo\_client

      EXCEPTIONS

        argument\_not\_found       = 1

        destination\_not\_found    = 2

        destination\_no\_authority = 3

        plugin\_not\_active        = 4

        internal\_error           = 5

        OTHERS                   = 6.

    IF sy-subrc <> 0.

    ENDIF.

    IF lo\_client IS BOUND.

*\* Send the HTTP request*

      CALL METHOD lo\_client->send

        EXCEPTIONS

          http\_communication\_failure = 1

          http\_invalid\_state         = 2

          http\_processing\_failed     = 3

          http\_invalid\_timeout       = 4

          OTHERS                     = 5.

      IF sy-subrc <> 0.

      ENDIF.

*\* HTTP call receive*

      CALL METHOD lo\_client->receive

        EXCEPTIONS

          http\_communication\_failure = 1

          http\_invalid\_state         = 2

          http\_processing\_failed     = 3

          OTHERS                     = 4.

      IF sy-subrc <> 0.

      ENDIF.

*\*   get status of the response*

      CALL METHOD lo\_client->response->get\_status

        IMPORTING

          code   = lv\_httpcode

          reason = lv\_reason.

*\* Instantiate REST client*

      CREATE OBJECT lo\_rest\_client

        EXPORTING

          io\_http\_client = lo\_client.

*\* Get Response data*

      lo\_response = lo\_rest\_client->if\_rest\_client~get\_response\_entity( ).

*\* Get string data*

      lv\_json = lo\_response->get\_string\_data( ).

*\* Convert string to Abap structures (*

      CALL METHOD /ui2/cl\_json=>deserialize

        EXPORTING

          json        = lv\_json

          pretty\_name = /ui2/cl\_json=>pretty\_mode-camel\_case

        CHANGING

          data        = lt\_git\_commit\_file.

      IF sy-subrc = 0.

        DATA wa\_entityset LIKE LINE OF et\_entityset.

        LOOP AT lt\_git\_commit\_file ASSIGNING FIELD-SYMBOL(<lfs\_gitrepo>).

          wa\_entityset-sha = <lfs\_gitrepo>-sha.

          wa\_entityset-commit-author-name = <lfs\_gitrepo>-commit-author-name.

          wa\_entityset-commit-author-date = <lfs\_gitrepo>-commit-author-date.

          REPLACE ALL OCCURRENCES OF 'Create' IN <lfs\_gitrepo>-commit-message

          WITH space.

          CONDENSE <lfs\_gitrepo>-commit-message NO-GAPS.

          wa\_entityset-commit-message = <lfs\_gitrepo>-commit-message.

          APPEND wa\_entityset TO et\_entityset.

          CLEAR: wa\_entityset.

        ENDLOOP.

      ENDIF.

    ENDIF.

  ENDMETHOD.