FUNCTION z\_rnm\_call\_aspentech\_rest\_api .  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  IMPORTING  
\*"     REFERENCE(IT\_DATA) TYPE  DATA  
\*"     REFERENCE(IV\_URL) TYPE  STRING  
\*"  EXPORTING  
\*"     REFERENCE(EX\_RESULT) TYPE  STRING  
\*"  EXCEPTIONS  
\*"      INTERNALERROR  
\*"      POSTINGERROR  
\*"      HTTP\_COMMUNICATION\_FAILURE  
\*"      HTTP\_INVALID\_STATE  
\*"      HTTP\_PROCESSING\_FAILED  
\*"      HTTP\_INVALID\_TIMEOUT  
\*"----------------------------------------------------------------------  
\*&---------------------------------------------------------------------\*  
\*& TITLE           : Z\_RNM\_CALL\_ASPENTECH\_REST\_API  
\*& CD              : 8048060: Interface data from SAP to Aspentech  
\*& TECHNICAL AUTHOR: PEOL\_046 (Pranjal Ghimire)  
\*& FUNCTIONAL      : Shridhar/Niranjan Shisode  
\*&---------------------------------------------------------------------\*  
\*& PURPOSE: THIS FM IS USED TO SEND DATA TO ASPENTECH USING REST API  
\*& RELEASE: 1.0  
\*&---------------------------------------------------------------------\*  
  TYPES : BEGIN OF lty\_response,  
           status       TYPE string,  
           description  TYPE string,  
          END OF lty\_response.  
  
\*          lt\_response\_t TYPE STANDARD TABLE OF lty\_response.  
  
  DATA:   lt\_response\_tt TYPE TABLE OF lty\_response.  
  
  TYPES: BEGIN OF lty\_sendstockdetailsres,  
           send\_stock\_details\_res LIKE lt\_response\_tt ,  
         END OF lty\_sendstockdetailsres,  
  
        BEGIN OF lty\_resp,  
          data TYPE lty\_sendstockdetailsres,  
        END OF lty\_resp.  
  
  DATA: snd\_json        TYPE zui\_cl\_json=>json,  
        lw\_result       TYPE lty\_response,  
        lw\_resp         TYPE lty\_resp,  
        lo\_http\_client  TYPE REF TO if\_http\_client,  
        lw\_response     TYPE string.  
  CONSTANTS : lc\_contentype TYPE string  VALUE 'Content-Type', "#EC NOTEXT  
               lc\_value      TYPE  string VALUE 'application/json', "#EC NOTEXT  
               lc\_req\_mthd   TYPE string VALUE '~request\_method', "#EC NOTEXT  
               lc\_req\_token  TYPE string VALUE 'x-access-token', "#EC NOTEXT  
               lc\_req\_val    TYPE string VALUE 'POST'.      "#EC NOTEXT  
  
  IF it\_data IS NOT INITIAL.  
    " serialize internal table into JSON, skipping initial fields and converting ABAP field names into camelCase  
    zui\_cl\_json=>serialize(  
        EXPORTING  
              data            =  it\_data   " Data to serialize  
              compress        = abap\_false ""C\_BOOL-FALSE    " Skip empty element  
              pretty\_name     =  zui\_cl\_json=>pretty\_mode-camel\_case ""PRETTY\_MODE-NONE    " Pretty Print property names  
        RECEIVING  
              r\_json          =  snd\_json   " JSON string  
              ).  
  ENDIF.  
  
  REPLACE FIRST OCCURRENCE OF  '[' IN snd\_json WITH space.  
  REPLACE  '"content":[{"' IN snd\_json WITH '"content":{"'.  
  REPLACE  '}]}]}]' IN snd\_json WITH '}]}}'.  
  REPLACE  '"}]}]' IN snd\_json WITH '"}}'.  
  
  "To create http call  
  cl\_http\_client=>create\_by\_url(  
    EXPORTING  
      url                =  iv\_url  " URL  
    IMPORTING  
      client             =  lo\_http\_client    " HTTP Client Abstraction  
    EXCEPTIONS  
      argument\_not\_found = 1  
      plugin\_not\_active  = 2  
      internal\_error     = 3  
      OTHERS             = 4 ) .  
  IF sy-subrc = 0.  
    lo\_http\_client->request->set\_header\_field(  
              EXPORTING  
                name  =   lc\_contentype  " Name of the header field  
                value =   lc\_value ).  " HTTP header field value  
  
    lo\_http\_client->request->set\_header\_field(  
             EXPORTING  
               name  =   lc\_req\_mthd  " ~request\_method header field  
               value =   lc\_req\_val   " HTTP header field value  
           ).  
\* lo\_http\_client->request->set\_header\_field(  
\*             EXPORTING  
\*               name  =   lc\_req\_token  " ~rx-access-token  
\*               value =   lv\_token   " HTTP header x-access-token  
\*           ).  
    lo\_http\_client->request->set\_cdata(  
                 EXPORTING  
                   data               =  snd\_json   " Binary data  
               ).  
    lo\_http\_client->request->set\_version( if\_http\_request=>co\_protocol\_version\_1\_1 ).  
  
    lo\_http\_client->send(  
                  EXCEPTIONS  
                    http\_communication\_failure = 1  
                    http\_invalid\_state         = 2  
                    http\_processing\_failed     = 3  
                    http\_invalid\_timeout       = 4  
                    OTHERS                     = 5  
                ).  
  
    CASE sy-subrc.  
      WHEN 0.  
      WHEN 1.  
        RAISE  http\_communication\_failure .  
      WHEN 2.  
        RAISE http\_invalid\_state.  
      WHEN 3.  
        RAISE http\_invalid\_state.  
      WHEN 4.  
        RAISE http\_processing\_failed.  
      WHEN 5.  
        RAISE http\_invalid\_timeout.  
      WHEN OTHERS.  
        RAISE http\_invalid\_state.  
    ENDCASE.  
  
    lo\_http\_client->receive(  
              EXCEPTIONS  
                http\_communication\_failure = 1  
                http\_invalid\_state         = 2  
                http\_processing\_failed     = 3  
                OTHERS                     = 4  
            ).  
    CASE sy-subrc.  
      WHEN 0.  
      WHEN 1.  
        RAISE  http\_communication\_failure.  
      WHEN 2.  
        RAISE http\_invalid\_state.  
      WHEN 3.  
        RAISE http\_processing\_failed.  
    ENDCASE.  
  
    lw\_response = lo\_http\_client->response->get\_cdata( ).  
    IF lw\_response IS NOT INITIAL.  
      zui\_cl\_json=>deserialize(  
        EXPORTING  
          json         = lw\_response  
          pretty\_name  = zui\_cl\_json=>pretty\_mode-camel\_case  
        CHANGING  
          data         = lw\_resp  
      ).  
      READ TABLE lw\_resp-data-send\_stock\_details\_res INTO lw\_result INDEX 1.  
      IF sy-subrc = 0.  
        ex\_result = lw\_result-description.  
      ENDIF.  
    ENDIF.  
  ENDIF.  
ENDFUNCTION.