\*&---------------------------------------------------------------------\*

\*& Report z\_lego\_code\_for\_batch\_pgm.

\*&

\*&---------------------------------------------------------------------\*

\*&

\*&

\*&---------------------------------------------------------------------\*

REPORT z\_lego\_code\_for\_batch\_pgm.

DATA : lt\_collectors TYPE TABLE OF zlego\_dt\_col\_tab,

ls\_collector TYPE zlego\_dt\_col\_tab,

lv\_json TYPE string,

lv\_response TYPE string,

lv\_url\_final TYPE string.

DATA : lo\_request TYPE REF TO if\_rest\_entity,

lo\_http\_client TYPE REF TO if\_http\_client,

lo\_rest\_client TYPE REF TO cl\_rest\_http\_client,

lo\_json TYPE REF TO cl\_clb\_parse\_json,

lo\_response TYPE REF TO if\_rest\_entity,

lo\_sql TYPE REF TO cx\_sy\_open\_sql\_db.

DATA : ls\_rest\_status TYPE i,

lo\_response\_entity TYPE REF TO if\_rest\_entity,

lv\_string TYPE string,

lv\_msgty TYPE symsgty.

\*Get all registered collectors

SELECT \* FROM zlego\_dt\_col\_tab INTO CORRESPONDING FIELDS OF TABLE @lt\_collectors.

\*If no records or an error is returned we exit.

IF sy-subrc <> 0.

EXIT.

ENDIF.

\*Process the internal table with register collectors

LOOP AT lt\_collectors INTO ls\_collector.

\*Call the collector function

CALL FUNCTION ls\_collector-collector\_name

IMPORTING

iv\_json = lv\_json

iv\_response = lv\_response.

\*Create URL for API call

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

EXPORTING

url = ls\_collector-collector\_api\_endp

IMPORTING

client = lo\_http\_client

EXCEPTIONS

OTHERS = 1.

lo\_http\_client->propertytype\_logon\_popup = lo\_http\_client->co\_disabled.

lo\_http\_client->request->set\_method( if\_http\_request=>co\_request\_method\_post ).

CREATE OBJECT lo\_rest\_client

EXPORTING

io\_http\_client = lo\_http\_client.

lo\_http\_client->request->set\_version( if\_http\_request=>co\_protocol\_version\_1\_0 ).

IF lo\_http\_client IS BOUND AND lo\_rest\_client IS BOUND.

cl\_http\_utility=>set\_request\_uri( request = lo\_http\_client->request uri = lv\_url\_final ).

\*--Set payload/body to JSON

lo\_request = lo\_rest\_client->if\_rest\_client~create\_request\_entity( ).

lo\_request->set\_content\_type( iv\_media\_type = if\_rest\_media\_type=>gc\_appl\_json ).

\*---Populate the request message

lo\_request->set\_string\_data( lv\_json ).

\*- http post method call

lo\_rest\_client->if\_rest\_resource~post( lo\_request ).

\*--Get the valid response fields

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

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

ls\_rest\_status = lo\_rest\_client->if\_rest\_client~get\_status( ).

\*--If there is an error write out an error log entry.

IF ls\_rest\_status <> 299.

"Write out a suitable error to the log.

ENDIF.

ENDIF.

ENDLOOP.

ENDFUNCTION.