



ABAP Editor: Display Report ZBW_REST_API_OASIS

Report

ZBW_REST_API_OASIS

Active

Pattern

Text Elements

Pretty Printer

```

1   *->
2   *-> Report ZBW_REST_API_OASIS
3   *->
4   *->
5   *->
6 REPORT ZBW_REST_API_OASIS.

7

8 DATA: go_rapiadso TYPE REF TO zcl_oasis_apigee_extractor,
9      gv_target  TYPE rsinfolprov.
10
11 DATA: i_date type sy-datum.
12
13
14   * Selection Screen
15
16
17   ↳ SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-t01.
18   | PARAMETERS: p_date type sy-datum.
19
20   | SELECTION-SCREEN END OF BLOCK b1.
21
22
23   * Start of selection
24
25 START-OF-SELECTION.
26
27   * Convert the first day of month
28
29   i_date = p_date.
30   i_date+6(2) = '01'.
31
32   * Execute data extraction  I
33
34   create object go_rapiadso
35   EXPORTING
36     iv_date = i_date .
37   EXCEPTIONS
38   OTHERS = 1.

```



ABAP Editor: Display Report ZBW REST API OASIS

Report

16
17 □ SELECTION-SCREEN BEGIN OF BLOCK bl WITH FRAME TITLE TEXT-t01.
18 PARAMETERS: p_date type sy-datum.
19
20 SELECTION-SCREEN END OF BLOCK bl.
21
22 [****
23 * Start of selection
24 *
25 START-OF-SELECTION.
26
27 * Convert the first day of month
28
29 i_date = p_date.
30 i_date+6(2) = '01'.
31
32 * Execute data extraction
33
34 create object go_rapiadso
35 EXPORTING
36 iv_date = i_date
37 EXCEPTIONS
38 OTHERS = 1.
39
40 [IF sy-subrc EQ 0.
41
42 CALL METHOD go_rapiadso->execute().
43
44 IF sy-subrc <> 0.
45 MESSAGE TEXT-e01 TYPE 'E'.
46
47 ENDIF.
48



Cass Ede Goto Utilities Environment System User



Class Builder: Display Class ZCL OASIS APIGEE EXTRACTOR



Class/Interface

Implemented / Active

■ Local Test Classes

一四

Properties	Interfaces	Friends	Attributes	Methods
------------	------------	---------	------------	---------

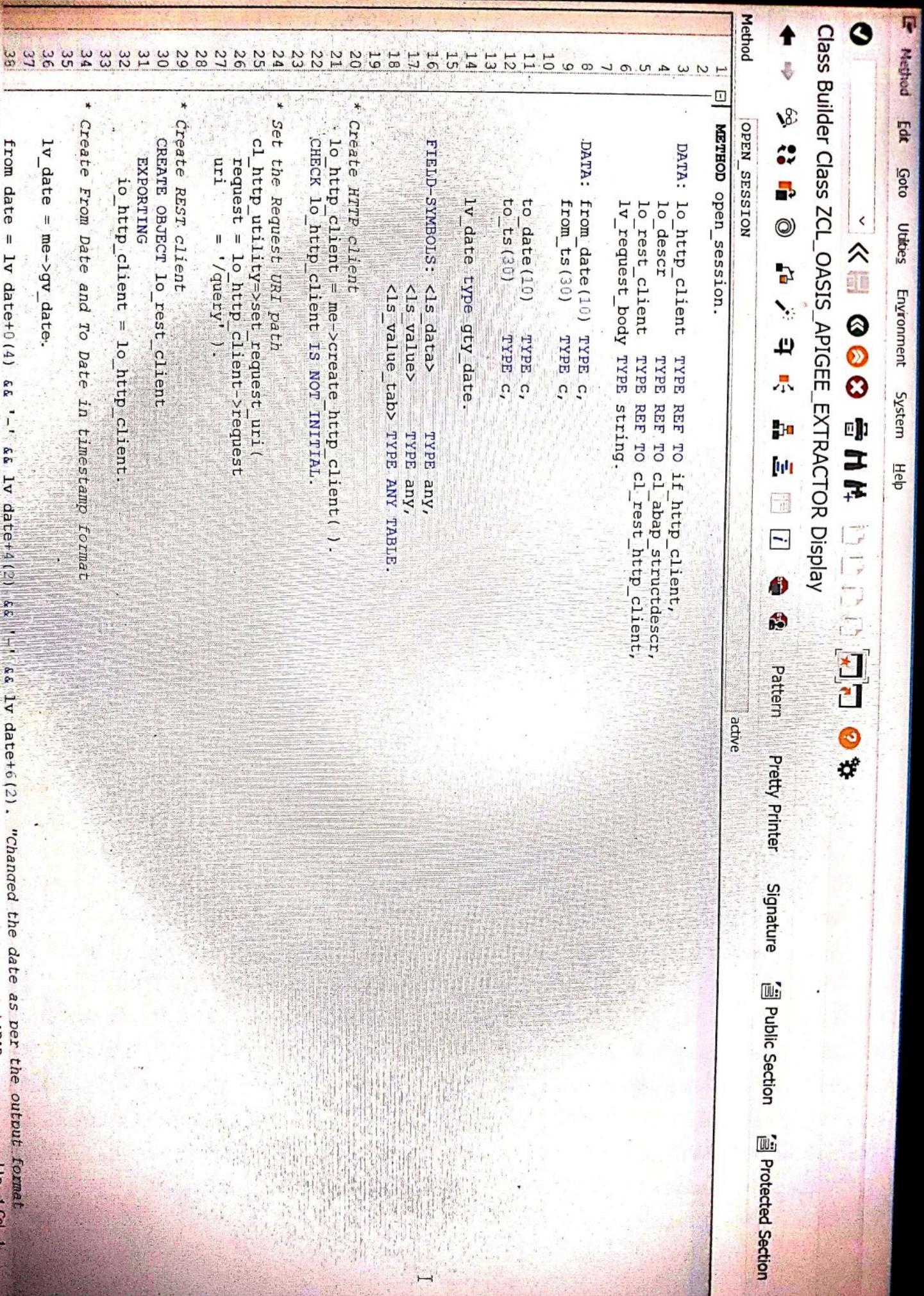
REFERENCES / REVIEW

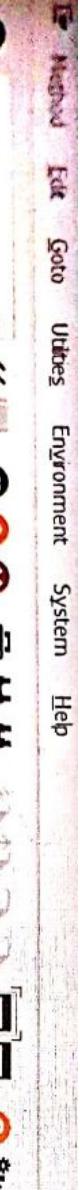
Local Definitions/Implementations

■ Local Test Classes

一四

Method	Exceptions		Sourcecode			
	Level	Visibility	M..	Description	Filter	
CONSTRUCTOR	Instance	Method Public	Tc	CONSTRUCTOR		
GET_DETAILS	Instance	Method Public		Get Account Details (Can be used to test connection)		
EXECUTE	Instance	Method Public		Main Method to Load the Apigee Dataset in R5		
OPEN_SESSION	Instance	Method Protected		Open the OASIS Query Session		
CLOSE_SESSION	Instance	Method Protected		Close the OASIS Query Session		
FETCH_DATA	Instance	Method Protected		Fetch Data		
LOAD_DATA	Instance	Method Protected		Load Data to STAPIAPIG ADSO		
CREATE_HTTP_CLIENT	Instance	Method Protected		Create HTTP Client for OASIS DAL API Call		
CONVERT_TIMESTAMP	Static	Method Private		Convert Timestamp Format		





Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display

Pattern Pretty Printer Signature Public Section Protected Section Private

OPEN_SESSION active

```
io_http_client = lo_http_client.
```

* Create From Date and To Date in timestamp format

lv date = me->gv date.

from date = lv date+0 (

```
from_ts = from_date && 'T00:00:00.000+00:00'.  
"Create the timestamp as per the OASIS format
```

`to_date = lv_date+0(4) && '-' && lv_date+4(2) && '-' && lv_date+6(2).` "Changed the date as per the output format

卷之三

* **TODO:** Create a structure then serialize to JSON
lv request body = `{"application-identifier":"OASIS IRIS APP" && sysid && "

```
&& "to-date": `` && to_ts && `` , "from-date": `` && from_ts && `` ,  
&& "data-format": "PARSED", "source-name": "oas-prd-apiw-api-call-and-upload-to-iris-monthly"}]
```

* Get parvovirus antibody (IgM or IgG)

```
DATA(lo_request) = lo_rest_client->if_rest_client~create_request_entity( ).
```

```
lv_request->set_content_type( lv_media_type ->_content_type );  
lv_request->set_string_data( lv_request_body ).
```

* POST

```
lo_rest_client->if_rest_resource~post( lo_request ).
```

* Collect response

```
DATA(1o_data) = /ui2/c1_json=>generate( json = lv_response ).
```

* Get Token ID - The actual property in OASIS is TOKEN-ID. For some reasons SAP interprets it as TOKEN_ID /ui2/cl data access=>create(ir data = lo data iv component = 'token id')->value(IMPORTING ev data = me->gv_token_id).

```
* print response (for testing purposes)
* cl_demo_output->display_json( json = lv_response ).
```

ENDMETHOD.

Scope: ME | HOB open_session

Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display

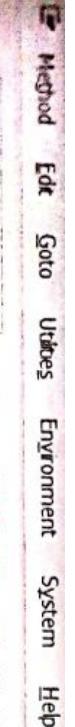
Method FETCH_DATA active

Pattern Pretty Printer Signature Public Section Protected Section Private

```

37
38 * Set Payload or body ( JSON or XML )
39 DATA(lo_request) = lo_rest_client->if_rest_client~create_request_entity( ).
40 lo_request->set_content_type(iv_media_type = if_rest_media_type=>gc_appl_json).
41 lo_request->set_string_data( lv_request_body ).
42
43
44 .lv_endloop = abap_false.
45
46 DO.
47
48 * POST
49
50 * rest_client->if_rest_resource~put( lo_request ). .
51
52 * get the total response, then get the object reference of the 'records' property
53 DATA(lo_response) = /ui2/cl_json->generate( json = lo_http_client->response->get_cdata() ) .
54
55 lo_rest_client->create( ir_data = lo_response iv_component = 'records_number' )->value( IMPORTING ev_data = lv_recordcount ) .
56 IF SY-index = 1.
57 WAIT UP TO 10 SECONDS.
58 CONTINUE.
59 ELSE.
60 EXIT.
61 ENDIF.
62
63
64 DATA(lo_data) = /ui2/cl_data_access->create( ir_data = lo_response iv_component = 'records' )->ref( ) .
65
66 CHECK lo_data IS BOUND.
67
68 * assign data reference of 'records' to a field symbol
69 ASSIGN lo_data->* TO <lt_records>.
70 CHECK <lt_records> IS ASSIGNED.
71
72 * get OASIS-SAP field mapping
73 SELECT * FROM ztbl_apigee_fmap
74 INTO TABLE @DATA(lt_fieldmap).

```



Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display

FETCH DATA

active

 Public Section  Protected Section  Private

'Private

```
73     SELECT * FROM ztbl_apigee_fmap
74     INTO TABLE @DATA(lt_fieldmap).
75
```

LOOP AT <lt_records> ASSIGNING FIELD-SYMBOL(<ls_record>).

* convert the json-string back to a data reference
* because method /ui2/cl_data-access=>create doesn't work with strings.
ASSIGN <ls_record>->* TO <ls_data>.
DATA(lo_record) = /ui2/cl_json=>generate(json = <ls_data>).
CHECK lo_record IS NOT INITIAL.

• dynamically populate the internal table using the field mapping
CLEAR: IS data.

LOOP AT lt_fieldmap ASSIGNING FIELD-SYMBOL(<ls_fieldmap>).

```
ASSIGN COMPONENT <ls_fieldmap>-sap_field OF STRUCTURE ls_data TO FIELD-SYMBOL(<ls_value_ref>).
CHECK sy-subrc EQ 0.
```

`lv_comppname = <cls_fieldmap>-oasis_field.`

CASE IV - comphane.

```
/ui2/cl_data_access->create( ir_data = lo_record iv_component = lv_comppname )->value( IMPORTING ev_data = lv_timestamp ).
```

```
zcl_oasis_apigee_extractor=>_convert_timestamp(
```

```
iv_timestamp = lv_timestamp  
IMPORTING
```

```
rv_timestamp = <ls_value_ref>  
).
```

WHEN OTHERS

```
/ui2/cl_data_access=>create( ir_data = lo_record iv_component = lv_comppname )->value( IMPORTING ev_data = <ls_value_ref> ).
```

Scope: \METHOD fetch_data



Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display



```

Method
FETCH_DATA active
Method
90 ASSIGN COMPONENT <ls_fieldmap>-sap_field OF STRUCTURE ls_data TO FIELD-SYMBOL(<ls_value_ref>).
91 CHECK sy-subrc EQ 0.
92 lv_comppname = <ls_fieldmap>-oasis_field.
93 CASE lv_comppname.
94 WHEN 'SOURCE-TIMESTAMP'.
95   /ui2/cl_data_access->create( ir_data = lo_record iv_component = lv_comppname )->value( IMPORTING ev_data = lv_timestamp ).
96   zcl_oasis_apigee_extractor->convert_timestamp(
97     EXPORTING
98       iv_timestamp = lv_timestamp
99       IMPORTING
100      rv_timestamp = <ls_value_ref>
101    ).
102
103
104
105
106 WHEN OTHERS.
107   /ui2/cl_data_access->create( ir_data = lo_record iv_component = lv_comppname )->value( IMPORTING ev_data = lv_timestamp ).
108 ENDCASE.
109
110
111
112 IF ls_data IS NOT INITIAL.
113 APPEND ls_data TO lt_data.
114 ENDIF.
115
116 ENDDO.
117
118 ENDDO.
119
120 gt_data[] = lt_data[].
121
122 * cl_demo_output->display_data(
123   value = gv_data ).
124
125
126 ENDMETHOD.

```



Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display



Method

LOAD_DATA

active

```

1  □ method LOAD_DATA.
2
3   CONSTANTS:  lc_adsomm TYPE rsroadsmm VALUE 'STAPIGOAS'.
4
5   DATA:  lv_debug TYPE rs_bool.
6
7   * CHECK gt_data_sprdacnti IS NOT INITIAL.
8
9   * Load data to ADSO SPRDACNTI
10  CALL FUNCTION 'RSDSO_DU_WRITE_API'
11    EXPORTING
12      i_adsomm          = lc_adsomm
13      i_allow_new_sids = abap_false
14      i_insert_only    = abap_false
15      it_data           = me->GT_DATA
16    EXCEPTIONS
17      write_failed       = 1
18      datastore_not_found = 2
19      OTHERS             = 3.
20
21  IF sy-subrc <> 0.
22
23  * Implement suitable error handling here
24
25  CASE sy-subrc.
26
27    WHEN 1.
28      MESSAGE 'Write Failed' TYPE 'S' DISPLAY LIKE 'E'.
29
30    WHEN 2.
31      MESSAGE 'Datastore Not Found' TYPE 'S' DISPLAY LIKE 'E'.
32
33    WHEN 3.
34      MESSAGE 'Others' TYPE 'S' DISPLAY LIKE 'E'.
35
36  ENDCASE.
37
38  endmethod.

```

Scope: \METHOD LOAD_DATA



Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display

↑
↓
←
→
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
Pattern
Pretty Printer

Protected Secti

Pretty Printer Signature

Signature Public Section

Method

CLOSE SESSION

active

```

3
4
5
6
7
8
9 FIELD-SYMBOLS: <ls_data>          TYPE any,
10 <ls_value>           TYPE any,
11 <ls_value_tab>        TYPE ANY TABLE.
12
13 * Create HTTP client
14 lo_http_client = me->create_http_client( ).  

15 CHECK lo_http_client IS NOT INITIAL.
16
17 * Set the Request URI path
18 cl_http_utility->set_request_uri(
19   request = lo_http_client->request
20   uri      = '/close' ).  

21
22 * Create REST client
23 CREATE OBJECT lo_rest_client
24 EXPORTING
25
26   lv_request_body = '{"token-id": "'
27
28   && gv_token_id
29   && '"}'.
30
31 * Set Payload or body ( JSON or XML )
32 DATA(lo_request) = lo_rest_client->if_rest_client~create_
33   lo_request->set_content_type(iv_media_type = if_rest_
34   lo_request->set_string_data(lv_request_body ) .
35
36
37 * POST
38
39 lo_rest_client->if_rest_resource~put( lo_request ).
```

Scope: \METHOD dose_session

Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display



Method

CLOSE_SESSION

active

	Pattern	Pretty Printer	Signature	Public Section
--	---------	----------------	-----------	----------------

```

10      <!ls_value>      TYPE any,
11      <cls_value_tab>  TYPE ANY TABLE.
12
13      * Create HTTP client
14      lo_http_client = me->create_http_client( ).
15      CHECK lo_http_client IS NOT INITIAL.
16
17      * Set the Request URI path
18      c1_http_utility->set_request_uri(
19          request = lo_http_client->request
20          uri      = '/close' ).
```

* Create REST client

```

22
23      CREATE OBJECT lo_rest_client
24      EXPORTING
25          io_http_client = lo_http_client.
```

lv_request_body = `{"token-id": ``

```

26
27          && gv_token_id
28          && `}`.
```

* Set Payload or body (JSON or XML)

```

30
31      DATA(lo_request) = lo_rest_client->if_rest_client->create_request_entity( );
32      lo_request->set_content_type( lv_media_type = if_rest_media_type->gc_appl_json ).
```

lo_request->set_string_data(lv_request_body).

```

35
36      * POST
37      lo_rest_client->if_rest_resource->put( lo_request ).
```

* Collect response

```

39
40      DATA(lo_response) = lo_rest_client->if_rest_client->get_response_entity( ).
```

DATA(lv_status) = lo_response->get_header_field('status_code').

```

42      DATA(lv_response) = lo_response->get_string_data( ).
```

ENDMETHOD.



10

1

二十一

Environment

Help

Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display

Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display

1

Method CREATE_HTTP_CLIENT

active

10

10

10

10

10

11

1

Scanned with CamScanner



Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display



Method CONVERT_TIMESTAMP

active

```

1 | METHOD_CONVERT_TIMESTAMP.
2 |
3 | DATA: lv_tz TYPE ttzz-tzone,
4 |       lv_date TYPE sy-datum,
5 |       lv_time TYPE sy-uzeit.
6 |
7 |
8 | IF iv_timestamp CS sy-abcd OR iv_timestamp CS '-'.
9 |   rv_timestamp = iv_timestamp.
10| ELSE.
11| TRY .
12|   cl_pco_utility->convert_java_timestamp_to_abap(
13|     EXPORTING
14|       iv_timestamp = iv_timestamp
15|     IMPORTING
16|       ev_date      = lv_date
17|       ev_time      = lv_time
18|     ).
19|
20|   rv_timestamp = lv_date+0(4) && '-' && lv_date+4(2) && '-' && lv_date+6(2) ss '00:00:00.000Z'.
21|
22| CATCH cx_root.
23| RETURN.
24|
25| ENDTRY.
26|
27| ENDMETHOD.
```

Method Edit Goto Utilities Environment System Help



Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display



active

Pattern

Pretty Printer

Method CONSTRUCTOR

```
1 1 method CONSTRUCTOR.  
2  
3 me->gv_date = iv_date.  
4  
5 endmethod.
```



Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display

GET_DETAILS

Method

DATA: lo_http_client TYPE REF TO if_http_client,
lo_descr TYPE REF TO cl_abap_structdescr.

FIELD-SYMBOLS: <ls_data> TYPE any,
<ls_value> TYPE any,

<ls_value_tab> TYPE any TABLE.

* Create HTTP client

10 lo_http_client = me->create_http_client().
11 CHECK lo_http_client IS NOT INITIAL.

* Set the Request URI path

14 c1_http_utility->set_request_uri(
15 request = lo_http_client->request
16 uri = '/my-details').

* Set the Request Header fields

19 lo_http_client->request->set_method(if_http_request->co_request_method_get).
20 .
21 .
22 lo_http_client->request->if_http_entity~set_formfield_encoding(formfield_encoding = c1_http_request->if_http_entity~co_encoding_raw).
23 lo_http_client->request->if_http_entity~set_content_type(content_type = if_rest_media_type->gc_appl_json).

24 .

25 lo_http_client->send(
EXCEPTIONS

26 http_communication_failure = 1
27 http_invalid_state = 2
28 http_processing_failed = 3
29 http_invalid_timeout = 4
30 OTHERS = 5
31 .
32).
33 .
34 .

35 lo_http_client->receive(
36 EXCEPTIONS
37 http_communication_failure = 1
http_invalid_state = 2

Scope: METHOD GET_DETAILS

active

Public Section

Protected Section

Private Section



Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display



Pattern

Pretty Printer

Signature

Public Section

Protected

Method

GET_DETAILS

```
36
37     EXCEPTIONS
38         http_communication_failure = 1
39         http_invalid_state        = 2
40         http_processing_failed    = 3
41         OTHERS                   = 4
42
43     CHECK sy-subrc EQ 0.
44
45     * get response
46     DATA(lo_data) = /ui2/c1_json->generate( json = lo_http_client->response->get_cdata( ) ).
47
48     * assign data reference to a field symbol
49     ASSIGN lo_data->* TO <ls_data>.
50     CHECK <ls_data> IS ASSIGNED.
51
52     * dynamically populate the output table
53
54     TRY :
55         lo_descr ?= c1_abap_typedescr->describe_by_name( 'ZSTR_OASIS_MYDETAILS' ).  

56         CATCH cx_root.
57
58     RETURN.
59
60     LOOP AT lo_descr->components[] ASSIGNING FIELD-SYMBOL(<ls_component>).
61     ASSIGN COMPONENT <ls_component>-name OF STRUCTURE <ls_data> TO FIELD-SYMBOL(<ls_value_ref>).
62     CHECK sy-subrc EQ 0.
63
64     CASE <ls_component>-name.
65     WHEN 'ROLES'.
66         ASSIGN <ls_value_ref>->* TO <ls_value_tab>.
67         WHEN OTHERS.
68             ASSIGN <ls_value_ref>->* TO <ls_value>.
69         ENDCASE.
70
71     ASSIGN COMPONENT <ls_component>-name OF STRUCTURE es_details TO FIELD-SYMBOL(<ls_target_field>).
72     IF sy-subrc EQ 0.
73         IF <ls_value tab> IS ASSIGNED.
```

Scope: \METHOD GET_DETAILS

1 ABAP

1 In 1 G



Class Builder Class ZCL_OASIS_APIGEE_EXTRACTOR Display

Method GET_DETAILS

active

```
57
58      LOOP AT lo_descr->components[] ASSIGNING FIELD-SYMBOL(<ls_component>).
59
60      ASSIGN COMPONENT <ls_component>-name OF STRUCTURE <ls_data> TO FIELD-SYMBOL(<ls_value_ref>).
61      CHECK sy-subrc EQ 0.
62
63      CASE <ls_component>-name.
64          WHEN 'ROLES'.
65              ASSIGN <ls_value_ref>->* TO <ls_value_tab>.
66
67          WHEN OTHERS.
68              ASSIGN <ls_value_ref>->* TO <ls_value>.
69          ENDCASE.
70
71          ASSIGN COMPONENT <ls_component>-name OF STRUCTURE es_details TO FIELD-SYMBOL(<ls_target_field>).
72          IF sy-subrc EQ 0.
73              IF <ls_value_tab> IS ASSIGNED.
74                  LOOP AT <ls_value_tab> ASSIGNING FIELD-SYMBOL(<ls_value_row>).
75                      ASSIGN <ls_value_row>->* TO FIELD-SYMBOL(<lv_row_val>).
76                      CHECK sy-subrc EQ 0.
77                      APPEND <lv_row_val> TO es_details-roles.
78                  ENDLOOP.
79
80              ELSE.
81                  <ls_target_field> = <ls_value>.
82              ENDIF.
83
84          UNASSIGN: <ls_value>, <ls_value_tab>.
85
86      ENDDO.
87
endmethod.
```