

# Before:

The screenshot shows the SAP ABAP IDE with the 'GET\_SAS\_TOKEN' method selected. The code is as follows:

```
25
26
27 call method cl_abap_hmac=>calculate_hmac_for_raw
28   exporting
29     if_algorithm = 'sha-256'
30     if_key       = decoded
31     if_data      = body_xstring
32     if_length    = 0
33   importing
34     ef_hmac64string = sign.
35   clear: lv_sas_key, decoded.
36 END-OF-DEFINITION.
37 encrypt_key.
38 new_expiry = lv_expiry_time.
39 CONDENSE new_expiry.
40 IF NOT sign IS INITIAL.
41   DATA wa_policy TYPE zadf_ehub_policy.
42   SELECT SINGLE * FROM zadf_ehub_policy INTO wa_policy WHERE interface_id EQ gv_interface_id.
43   IF sy-subrc EQ 0.
44     sign = escape( val = sign format = format ).
45     CONCATENATE 'SharedAccessSignature sig=' sign '%se=' new_expiry '%skn=' wa_policy '%sr=' encoded_base_address INTO final_token.
46     rv_sas_token = final_token.
47   ENDIF.
48 ELSE.
49   RAISE EXCEPTION TYPE zcx_adf_service
50     EXPORTING
51       textid      = zcx_adf_service=>sas_key_not_generated
52       interface_id = gv_interface_id.
53 ENDIF.
54 ENDMETHOD.
```

A red box highlights the 'wa\_policy' variable in the concatenation statement on line 45. A callout bubble points to it with the text: "Incorrect code for skn value".

The screenshot shows the 'Fields' tab in the SAP ABAP IDE. The 'RV\_SAS\_TOKEN' field is selected, showing its data type as 'CSTRING(172)' and absolute type as 'TYPE=STRING'. Below the field details, the 'SharedAccessSignature' field is highlighted with a red box. A callout bubble points to it with the text: "Incorrect skn value".

After:

Class Builder Class ZCL\_ADF\_SERVICE\_SERVICEBUS Change

Method GET\_SAS\_TOKEN

Corrected code for skn value

```
25: call method cl_abap_hmac=>calculate_hmac_for_raw
26: exporting
27:   if_algorithm = 'sha-256'
28:   if_key       = decoded
29:   if_data      = body_xstring
30:   if_length    = 0
31: importing
32:   ef_hmacb64string = sign.
33: clear: lv_sas_key, decoded.
34: END-OF-DEFINITION.
35: encrypt_key.
36: new_expiry = lv_expiry_time.
37: CONDENSE new_expiry.
38: IF NOT sign IS INITIAL.
39:   DATA wa_policy TYPE zadf_ehub_policy.
40:   SELECT SINGLE * FROM zadf_ehub_policy INTO wa_policy WHERE interface_id EQ gv_interface_id.
41:   IF sy-subrc EQ 0.
42:     sign = escape( val = sign format = format ).
43:     * CONCATENATE 'SharedAccessSignature sig=' sign 'sse=' new_expiry '&skn=' wa_policy '&srt=' encoded_base_address INTO final_token.
44:     CONCATENATE 'SharedAccessSignature sig=' sign 'sse=' new_expiry '&skn=' wa_policy_policy '&srt=' encoded_base_address INTO final_token.
45:     lv_sas_token = final_token.
46:   ENDIF.
47: ELSE.
48:   RAISE EXCEPTION TYPE zcx_adf_service
49:   EXPORTING
50:     textid      = zcx_adf_service=>sas_key_not_generated
51:     interface_id = gv_interface_id.
52: ENDIF.
53: ENDMETHOD.
```

Fields Detail Displ.

Field RV\_SAS\_TOKEN

Data Type CString(159)

Absolute Type \TYPE=STRING

View Past Display

SharedAccessSignature sig= sse=1554739410&skn=sap-smithes&srt= ,servicebus.windows.net