Source code for extending Oracle Sales Cloud using APEX and JWT

Here is an example of how to set the headers before executing *make_rest_request*:

```
apex_web_service.g_request_headers(1).name := 'Content-Type';
apex_web_service.g_request_headers(1).value :=
'application/vnd.oracle.adf.resourceitem + json';
apex_web_service.g_request_headers(2).name := 'Authorization';
apex_web_service.g_request_headers(2).value := jwt;
apex_web_service.g_request_headers(3).name := 'Accept-Encoding';
apex_web_service.g_request_headers(3).value := 'gzip';
```

The source code of the functions:

function custom_authenticate

```
Create or replace function custom_authenticate (p_username in
varchar2,  p_password in varchar2) return boolean is
begin
     apex_debug.error('custom_authenticate accessed')
     return APEX_UTIL.IS_LOGIN_PASSWORD_VALID(  p_username, p_password);
End custom_authenticate ;
/
```

function jwt_check_sentry

```
create or replace function jwt_check_sentry
return boolean
is
v_jwt varchar2(4000);
v_username varchar2(4000);
l_url_params APEX_APPLICATION_GLOBAL.VC_ARR2;
l_param_names APEX_APPLICATION_GLOBAL.VC_ARR2;
l_param_values APEX_APPLICATION_GLOBAL.VC_ARR2;
v_jwt_index number;
l_session NUMBER;

begin
--apex_debug.error('SCRIPT_NAME: %s -- %s -- %s ',OWA_UTIL.get_cgi_env
('SCRIPT_NAME'),APEX_UTIL.HOST_URL('SCRIPT'), OWA_UTIL.get_cgi_env
('QUERY_STRING'));
--extracting jwt from URL
--apex_debug.error('current user name:
```

```
%s',APEX_CUSTOM_AUTH.GET_USERNAME);
-- don't do anything if user is already logged on

   IF APEX_APPLICATION.G_USER <> 'nobody'
   THEN
       RETURN TRUE;
   END IF;

apex_debug.error(OWA_UTIL.get_cgi_env ('QUERY_STRING'));

--checking if there are JWT in the url
if instr(OWA_UTIL.get_cgi_env ('QUERY_STRING'),'JWT') > 0 then

--getting the url params (all)
 l_url_params := APEX_UTIL.STRING_TO_TABLE(OWA_UTIL.get_cgi_env
('QUERY_STRING'));

--getting 7th params (items name)
 l_param_names :=  APEX_UTIL.STRING_TO_TABLE(l_url_params(7),',');

--getting 8th params (item values)
 l_param_values := APEX_UTIL.STRING_TO_TABLE(l_url_params(8),',');

--calculating which parameter index is JWT
 for i in 1 ..l_param_names.count loop
   if l_param_names(i) = 'JWT' then
     v_jwt_index := i;
     exit;
   end if;
 end loop;

 v_jwt := l_param_values(v_jwt_index);

 --getting the username from jwt
 v_username := GET_USERNAME_FROM_JWT(v_jwt);
 apex_debug.error('Username extracted from jwt: %s',v_username);
 if v_username is not null then
   apex_debug.error('Setting the user %s to the session',v_username);

   -- is there already a session?
   l_session := APEX_CUSTOM_AUTH.GET_SESSION_ID_FROM_COOKIE;
   IF l_session IS NOT NULL
       THEN
           -- test if the session is still valid and get a new session
id, if not valid
           IF NOT APEX_CUSTOM_AUTH.IS_SESSION_VALID
           THEN
               l_session := APEX_CUSTOM_AUTH.GET_NEXT_SESSION_ID;
           END IF;
       ELSE
```

```
            -- no session in cookie found, get a new session id
            l_session := APEX_CUSTOM_AUTH.GET_NEXT_SESSION_ID;
   END IF;
   APEX_CUSTOM_AUTH.DEFINE_USER_SESSION (v_username, l_session);
   return true;
 end if;
end if;

if v_jwt is null or APEX_CUSTOM_AUTH.GET_USERNAME is null then
    apex_debug.error('No jwt provided and no user logged in');
    return false;
end if;
```

```
end jwt_check_sentry;

/
```

A utility function designed to extract the username from the JWT token:

function "GET_USERNAME_FROM_JWT"

```
create or replace function "GET_USERNAME_FROM_JWT"
(jwt in VARCHAR2)
return varchar2
is
l_jwt varchar2(4000);
l_start number;
l_end number;
l_middle varchar2(4000);
l_json varchar2(4000);
l_output varchar2(4000);
l_jwt_user varchar2(4000);
l_return varchar2(4000);

begin
   if jwt is null then
      return null;
   end if;

   l_jwt := jwt;
   --find the location of the two period markers
   l_start := INSTR( l_jwt, '.' );
   l_end := INSTR( l_jwt, '.', l_start+1);

   --take the middle substring
   l_middle := SUBSTR( l_jwt, l_start+1 , (l_end - l_start)-1);

   --need to base64 decode the middle string to get a json string
   l_json := utl_raw.cast_to_varchar2(UTL_ENCODE.BASE64_DECODE(
utl_raw.cast_to_raw( l_middle)));

   --use the APEX parser and then find the username at the 'prn' node
   apex_json.parse(l_json);
   l_jwt_user := initcap(apex_json.get_varchar2(p_path => 'prn'));

   return l_jwt_user;
End;

/
```