日日是Oracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2021年11月10日水曜日

RS256のJWTをDBMS_CRTYPTO.SIGNを使って生成する

以前にRS256を使ったJWTを生成するという記事を書いたのですが、最近Autonomous Databaseの19cでRSA暗号がサポートされていることに気がつきました。そうであればJavaのRSAの実装を使う必要はありません。

以前の記事のコードでJavaを使っている部分をDBMS_CRYTPO.SIGNを使うように書き直して動作を確認しました。

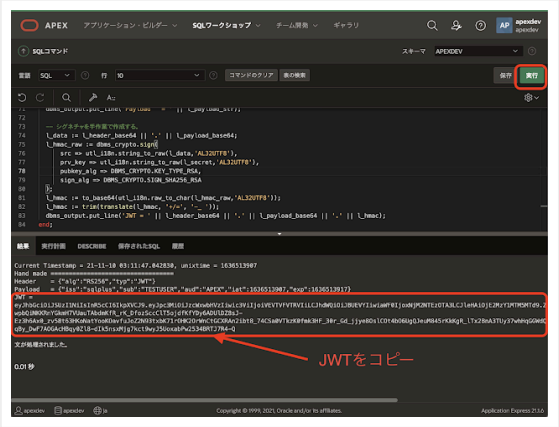
置き換えたコードは以下です。

```
declare
 l_now       timestamp;
 l_secret    varchar2(32767) := 'MIIEogIBA* PKCS#1形式の秘密鍵 *RsvCjBJo=';
 l_username  varchar2(32) := 'TESTUSER';
 l_jwt       varchar2(32767);
 l_jwt_token apex_jwt.t_token;
 l_jwt_t     apex_t_varchar2;

 l_header_json json_object_t;
 l_header_str  varchar2(200);
 l_header_base64 varchar2(400);
 l_payload_json json_object_t;
 l_payload_str  varchar2(200);
 l_payload_base64 varchar2(800);
 l_token varchar2(1000);
 l_hmac varchar2(1000);

 -- DBMS_CRYPTO.SIGNへの置き換えのために追加。
 l_data varchar2(400); -- header.payload
 l_hmac_raw raw(2000);

 -- Unix時間の取得
 function unixtime(p_timestamp in timestamp)
 return pls_integer
 is
   l_date date;
   l_epoc number;
 begin
   l_date := sys_extract_utc(p_timestamp);
   l_epoc := l_date - date'1970-01-01';
   return l_epoc * 24 * 60 * 60;
 end unixtime;

 -- Base64のデコード
 function from_base64(t in varchar2) return varchar2 is
 begin
   return utl_raw.cast_to_varchar2(utl_encode.base64_decode(utl_raw.cast_to_raw(t)));
 end from_base64;
```

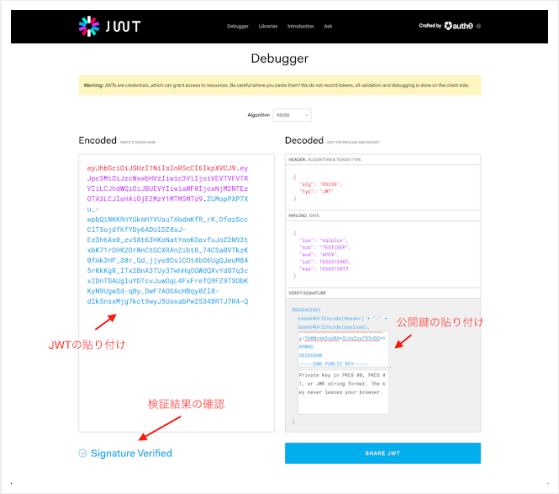
```
  -- Base64のエンコード
  function to_base64(t in varchar2) return varchar2 is
    l_base64 varchar2(32767);
  begin
    l_base64 := utl_raw.cast_to_varchar2(utl_encode.base64_encode(utl_raw.cast_to_raw(t)));
    l_base64 := replace(l_base64, chr(13)||chr(10), '');
    return l_base64;
  end to_base64;
begin
  -- 共通で使用する現在時刻
  l_now := current_timestamp;
  dbms_output.put_line('Current Timestamp = ' || l_now || ', unixtime = ' || unixtime(l_now));

  -- ヘッダーを手作業で生成する。
  dbms_output.put_line('Hand made ==================================');
  l_header_json := json_object_t();
  l_header_json.put('alg','RS256');
  l_header_json.put('typ','JWT');
  l_header_str := l_header_json.to_string();
  l_header_base64 := to_base64(l_header_str);
  dbms_output.put_line('Header    = ' || l_header_str);

  -- ペイロードを手作業で作成する。
  l_payload_json := json_object_t();
  l_payload_json.put('iss','sqlplus');
  l_payload_json.put('sub',l_username);
  l_payload_json.put('aud','APEX');
  l_payload_json.put('iat',unixtime(l_now));
  l_payload_json.put('exp',unixtime(l_now)+10);
  l_payload_str := l_payload_json.to_string();
  l_payload_base64 := to_base64(l_payload_str);
  dbms_output.put_line('Payload   = ' || l_payload_str);

  -- シグネチャを手作業で作成する。
  l_data := l_header_base64 || '.' || l_payload_base64;
  l_hmac_raw := dbms_crypto.sign(
     src => utl_i18n.string_to_raw(l_data,'AL32UTF8'),
     prv_key => utl_i18n.string_to_raw(l_secret,'AL32UTF8'),
     pubkey_alg => DBMS_CRYPTO.KEY_TYPE_RSA,
     sign_alg => DBMS_CRYPTO.SIGN_SHA256_RSA
  );
  l_hmac := to_base64(utl_i18n.raw_to_char(l_hmac_raw,'AL32UTF8'));
  l_hmac := trim(translate(l_hmac, '+/=', '-_ '));
  dbms_output.put_line('JWT = ' || l_header_base64 || '.' || l_payload_base64 || '.' || l_hmac);
end;
```

APEXの**SQLコマンド**より実行してみました。事前に、管理ユーザーのADMINにてパッケージDBMS_CRYPTOの実行権限をワークスペース・スキーマに与えておきます。

grant execute on dbms_crypto to <ワークスペース・スキーマ>;



出力されたJWTをjwt.ioで検証します。



Oracle APEX 21.2より認証スキームにSAMLが追加されました。DBMS_CRYTPOにRSA暗号のサポートが追加されていることを前提としているSAMLがサポートされたということは、JWTを扱うパッケージAPEX_JWTの署名アルゴリズムにRS256なども追加されるかもしれません。

以上になります。

Oracle APEXのアプリケーション作成の参考になれば幸いです。

完

Yuji N. 時刻: 12:31

共有

‹ ホーム ›

ウェブ バージョンを表示

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。
こちらの記事につきましては、免責事項の参照をお願いいたします。

詳細プロフィールを表示