

# 日々是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。

2022年6月14日 火曜日

## SAML認証で使ったデバッグ手順

SAML認証のデバッグで使った、いくつかの手順を紹介します。

### 公開鍵証明書の確認

SAMLの認証スキームの編集画面より、証明を登録したときに**証明書が無効です**と表示されることがあります。



証明書の貼り付け方に問題があるのか（コピペミスなど）、または証明書自体に問題があるのか、このメッセージだけでは特定が難しいです。openssl x509 -textを実行することで、証明書の内容が確認できている場合は、さらに難しいです。

### openssl x509 -in 公開鍵証明書ファイル -text -noout

マニュアルには記載されていませんが、Oracle APEXで扱う公開鍵証明書を検証するファンクションがあります。**WWV\_FLOW\_PUBLIC\_KEY\_UTIL.IS\_CERTIFICATE**です。公開鍵証明書を文字列として受け取り、証明書として正しい場合はTRUE、そうでなければFALSEを返します。

このファンクションを使った検証スクリプトの雛形です。

```
set serveroutput on
declare
  l_cert  varchar2(32767);
begin
  l_cert := q'~
-----BEGIN CERTIFICATE-----
```

```
MIIClzCCAX8CBgGBWpvrHTANBgqhkiG9w0BAQsFADAPMQ0wCwYDVQQDDARhcGV4
```

検証する証明書を貼り付ける。

```
F04IqzTVHoCfuqtrcHVkuG1BCyMEtU0Aafuhkpz4CRMsIEgws2CgSbUJ8S9dNtjk
wABHyWhq64wG17BfEq/utwgkBKdqKcOM9StviMayg9bMsfe2BBfxa00ZrA==
```

```
-----END CERTIFICATE-----
```

```
~';
    if
        wwv_flow_public_key_util.is_certificate(l_cert)
    then
        dbms_output.put_line('Certificate is valid.');
```

```
    else
```

```
        dbms_output.put_line('Certificate is NOT valid.');
```

```
    end if;
```

```
end;
```

```
/
```

verify-cert.sql hosted with ❤ by GitHub

[view raw](#)

opensslではパースできるのに、上記でFALSEとなる（Certificate is NOT valid.が印刷される）場合は、Oracle APEXが提供しているファンクションを使って、証明書がパースできるかどうか確認します。WWV\_FLOW\_PUBLIC\_KEY\_UTIL.PARSE\_CERTIFICATEを使います。

このファンクションを使った検証スクリプトの雛形です。

```
set serveroutput on
declare
    l_cert      varchar2(32767);
    l_cert_info wwv_flow_public_key_util.t_cert_info;
begin
    l_cert := q'~
-----BEGIN CERTIFICATE-----
MIICuzCCAa0gAwIBAgIUDBhT6l9jzhzdyuwqKAJLnwi/31YwDQYJKoZIhvcNAQEL
公開鍵証明書を貼り付ける。
r31APDa1oRH5DwBB8euBn9UNWY4Hs+MmbQu2FmJy0hZMkuE9FLE0/YYwRmTIhWA4
Q6z0qVXrS7e637VQAX3op+Rq0RlyUvIOMAtpsZzAbg==
-----END CERTIFICATE-----
~';
    l_cert_info := wwv_flow_public_key_util.parse_certificate(l_cert,true);
    dbms_output.put_line('cert_sig_algorithm = ' ||
        case l_cert_info.cert_sig_algorithm
        when 0 then 'c_sha1_rsa'
        when 1 then 'c_sha256_rsa'
        when 2 then 'c_sha386_rsa'
        when 3 then 'c_sha512_rsa'
        end
    );
    dbms_output.put_line('key_algorithm = ' || l_cert_info.key_algorithm);
```

```
dbms_output.put_line('key_length = ' || l_cert_info.key_length);  
end;  
/
```

parse-cert.sql hosted with ❤ by GitHub

[view raw](#)

スクリプトの実行にあたっては、事前にAPEXのスキーマをCURRENT\_SCHEMAに設定します。

APEX 22.1の場合は、APEX\_220100がAPEXがインストールされているスキーマになります。

```
alter session set current_schema = APEX_220100;
```

残念ながらAutonomous Databaseでは、APEXのスキーマは保護されています。そのため、内部で使用するパッケージに含まれるプロシージャやファンクションを呼び出すことはできません。オンプレの環境を用意する必要があります。

正常に実行されると、以下のように表示されます。

```
SQL> alter session set current_schema = apex_220100;
```

```
Session altered.
```

```
SQL> @verify-cert  
Certificate is valid.
```

```
PL/SQL procedure successfully completed.
```

```
SQL> @parse-cert  
cert_sig_algorithm = c_sha256_rsa  
key_algorithm = RSA  
key_length = 2048
```

```
PL/SQL procedure successfully completed.
```

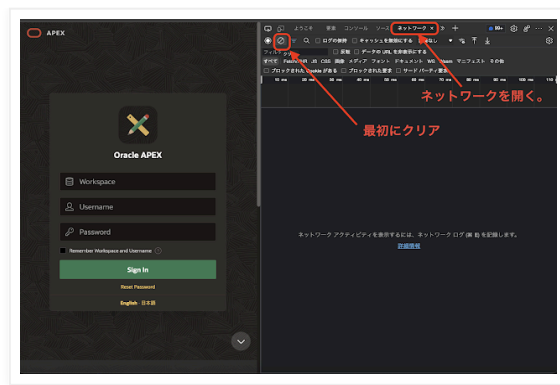
```
SQL>
```

## HARファイルの確認

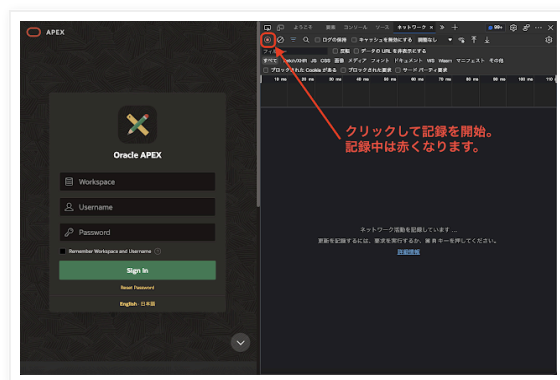
ORDSへ送信されるOriginヘッダーの内容は、HARをエクスポートして確認しています。

ChromeまたはEdgeを例に取ります。ブラウザの**開発者ツール**を実行し、**ネットワーク**を開きます。

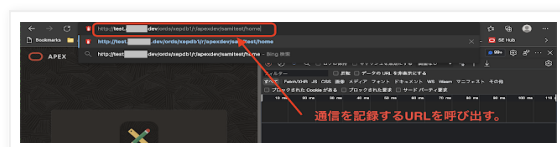
最初に、それまでの記録をクリアします。



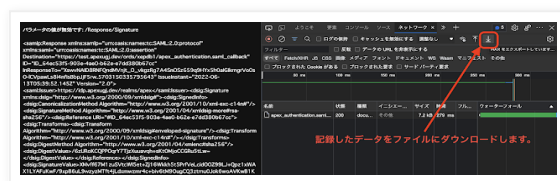
レコード・ボタンをクリックして、ネットワーク・ログの記録を開始します。記録中は、ボタンが赤くなります。



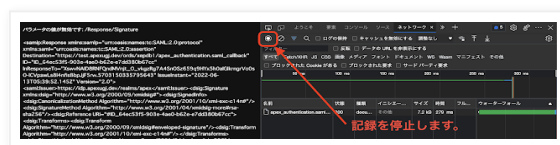
ネットワーク・ログを記録するURLにアクセスします。



記録したデータを、ファイルにダウンロードします。



レコード・ボタンをクリックし、レコーディングを停止します。



ダウンロードしたファイルの内容を確認します。

HTTPヘッダーなどの情報を確認することができます。



ブラウザごとに操作に違いがあります。使用しているブラウザでのHARの取得方法を確認しましょう。

## d0、d1、d2スクリプトの使用

SAMLコールバックのデバッグ・メッセージを取得するには、インスタンス全体でデバッグを有効にする必要があります。

このためにAPEXのダウンロード・メディアのapex/utilities/debug以下に配置されているd0.sqlを使用します。

APEXのデータベースに、sysまたはsystemで接続し、CURRENT\_SCHEMAをAPEXのスキーマに設定します。続いてd0.sqlを実行します。デバッグ・レベルが9、つまり完全トレースになります。APEX全体への影響が大きいため、本番環境での実施には特別な注意が必要です。

```
SQL> alter session set current_schema = apex_220100;
```

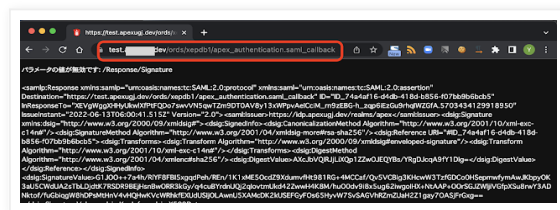
```
Session altered.
```

```
SQL> @d0
```

```
Changed debug level from "" to "9"
```

```
SQL>
```

SAMLのサインインを実施します。



再度d0.sqlを実行し、デバッグ・メッセージの取得を停止します。

```
SQL> alter session set current_schema = apex_220100;
```

```
Session altered.
```

```
SQL> @d0
Changed debug level from "" to "9"
SQL> @d0
Changed debug level from "9" to ""
SQL>
```

**d1.sql**を実行し、直近のAPEXへのアクセスを一覧します。

PAGE_VIEW_ID SESSION_ID	STARTED	SECS	LVL	COUNT	PATH_INFO WORKSPACE	USER	APP:PAGE
@d2 8003 0	06:00:00	0.03 Unknown		16	DBMS_SCHEDULER/ORACLE_APEX_M AIL_QUEUE		
@d2 8004 0	06:00:00	0.01 Unknown		5	DBMS_SCHEDULER/ORACLE_APEX_W S_NOTIFICATIONS		
@d2 8005 0	06:00:02	0.01 Unknown		9	DBMS_SCHEDULER/ORACLE_APEX_P URGE_SESSIONS		
@d2 8006 16849648434733	06:00:10	0.20 INTERNAL	WRN	220	show nobody	4500:1000	0-

PAGE_VIEW_ID SESSION_ID	STARTED	SECS	LVL	COUNT	PATH_INFO WORKSPACE	USER	APP:PAGE
@d2 8007 16849648434733	06:00:11	0.72 INTERNAL		1610	show nobody	4550:1	0-
@d2 8618 9657959965886	06:00:24	0.18 APEXDEV	WRN	315	show nobody	102:1	0-
@d2 8619 9657959965886	06:00:43	0.36 APEXDEV	WRN	508	ajax plugin nobody	102:1	0-

30 rows selected.

```
SQL>
```

アプリケーションIDやページIDなどを参照して、確認すべきページ・ビューIDを特定します。SAMLコールバックのPATH\_INFOはajax pluginになります。

上記の例では@d2 8619より、SAMLコールバックのデバッグ・メッセージを確認することができます。

```
SQL> @d2 8619
```

デバッグ・メッセージが手元のファイルにダウンロードされ、エディタが開きます。



詳細なデバッグ・メッセージを参照できます。通常は、このデータをファイルに保存してオラクルのサポートに提出することになるでしょう。

## 画面へのエラー詳細表示

ORDSのプロパティ **debug.printDebugToScreen** を **true** にすると、ブラウザの画面にエラーの詳細が表示されます。マニュアルの [こちら](#) に説明があります。

debug.printDebugToScreen が false（デフォルト）のときに、ORDS でエラーが発生した画面です。



ORDS 22.1にて、上記のプロパティをtrueにする方法です。ordsコマンドのパスおよび構成ファイルが含まれるディレクトリは、インストール毎に異なります。

```
/usr/local/bin/ords --config /etc/ords/config config set debug.printDebugToScreen true
```

```
[oracle@apex ~]$ /usr/local/bin/ords --config /etc/ords/config config set  
debug.printDebugToScreen true
```

```
ORDS: Release 22.1 Production on Tue Jun 14 04:39:29 2022
```

```
Copyright (c) 2010, 2022, Oracle.
```

```
Configuration:  
  /etc/ords/config/
```

```
The global setting named: debug.printDebugToScreen was set to: true  
[oracle@apex ~]$
```

変更を有効にするには、ORDS（またはTomcat）を再起動する必要があります。

ORDS 21.4.3以前では、以下のコマンドを実行します。

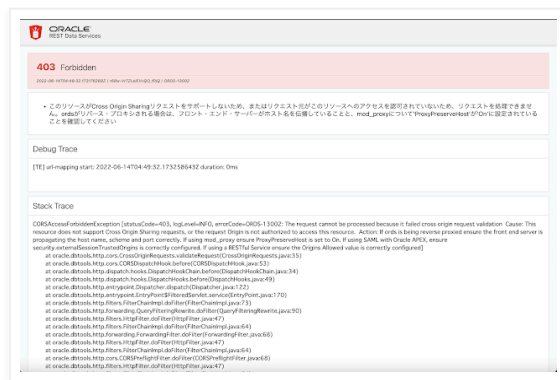
```
java -jar ords.war set-property debug.printDebugToScreen true
```

```
[oracle@ords ords]$ java -jar ords.war set-property debug.printDebugToScreen true  
2022-06-14T04:45:10.789Z INFO Modified:  
/opt/oracle/ords/conf/ords/defaults.xml, setting: debug.printDebugToScreen = true  
[oracle@ords ords]$
```

ORDS 22.1および21.4.3の双方で、上記のコマンドによって、設定ファイルに以下の行が書き込まれます。設定ファイルの名前はORDS 22.1ではsettings.xml、ORDS 21.xではdefaults.xmlになります。

```
<entry key="debug.printDebugToScreen">true</entry>
```

以上の変更より、エラーの詳細が画面に表示されます。



エラー発生時の詳細が画面に表示されるため、この状況での本番運用は望ましくありません。デバッグが終了したら、以下のコマンドにより設定を削除します。

```
/usr/local/bin/ords --config /etc/ords/config config delete debug.printDebugToScreen
```

```
[oracle@apex ~]$ /usr/local/bin/ords --config /etc/ords/config config delete debug.printDebugToScreen
```

ORDS: Release 22.1 Production on Tue Jun 14 05:00:18 2022

Copyright (c) 2010, 2022, Oracle.

Configuration:  
/etc/ords/config/

The global setting named: debug.printDebugToScreen was removed from the configuration  
[oracle@apex ~]\$

ORDS 21.xではエディタでdefaultxmlを開いて、該当の設定を削除します。

変更の反映には、ORDSの再起動が必要です。

SAML認証で使用した、デバッグ手順の紹介は以上になります。

完

Yuji N. 時刻: 14:06

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](#)