Exploring the details of APEX sessions

APEX World 2018, Mar 22, 2018
About me

Menno Hoogendijk
✓ Fulltime APEX developer
✓ Working with Oracle since 2008
✓ Tries to be a fullstack developer

Speaking at:
✓ APEX World 2018
✓ APEX Alpe Adria 2018
✓ APEX Connect 2018
✓ Kscope18

@mennooo
mennooo
www.menn.ooo
Powerful JavaScript skills for all APEX developers

About me

- Over 160 colleagues work in the Netherlands
- Over 40 colleagues work in the Caribbean
Today’s menu

Introduction

Creating a session

The login process

Session state

Smaller session features

The logout process
HTTP Protocol

HTTP request messages:
GET www.google.com HTTP/1.1
User-Agent: Mozilla/5.0
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

HTTP response messages:
HTTP/1.1 200 OK
Content-Length: 88
Content-Type: text/html
Connection: Closed
<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
HTTP is stateless

Web server

User 1

User 2

User 3

User 4
What is a session?

“A session establishes **stateful** behavior across pages for each user”
How does APEX create a session?
You’re browsing as a Guest

Pages you view in this window won’t appear in the browser history and they won’t leave other traces, like cookies, on the computer after you close all open Guest windows. Any files you download will be preserved, however.

LEARN MORE
You’re seeing this as a test.

Pages you view in this window don’t leave any cookies, on the off chance you close all open Chrome windows, however.

Recording network activity...

Perform a request or hit F5 to record the reload.
The HTTP response status code 302 Found is a common way of performing URL redirection.
What happened in APEX?

1. Check if the session is valid
2. Create a new session for user “nobody”
3. Redirect to the “session not valid URL”
1. Check if the session is valid

How depends on your **Authentication Scheme**

If a **sentry** function exists, it will use that one.

If a **sentry** function does not exist, it will use the internal one.
This is a custom Authentication Scheme

```sql
function sentry
return boolean
is
begin
  return true;
end sentry;
```
HTTPS only?
2. Create a new session for user “nobody”

Even before login, a new session is created.

Tip: use apex_dictionary view

```sql
select *
  from apex_dictionary
where apex_view_name like 'SESSION%' 
  and column_id = 0;
```

Tip: grant role to schema to see all data

```sql
grant apex_administrator_role to <SCHEMA>;
```
3. Redirect to the “session not valid URL”

Only when `sentry` returns false
Login URL: ?p=&APP_ID:LOGIN_DESKTOP:&SESSION.
The login process
Authentication Process

- apex_authentication.login
- Pre Authentication
- Authentication
- Post Authentication
APEX Sessions

- P101_USERNAME
- P101_PASSWORD

PL/SQL Code:
```
apex_authentication.login(
    p_username => :P101_USERNAME,
    p_password => :P101_PASSWORD
);
```
function authenticate (  
    p_username in varchar2,  
    p_password in varchar2  
)  
return boolean
Tip: FSP_AFTER_LOGIN_URL in Post Authentication

Purpose:
Do not go to predefined home page, but redirect to custom URL

:FSP_AFTER_LOGIN_URL := apex_page.get_url(p_page => 2)
Plugin your APEX widgets

User Interface Details

Attributes

- Auto Detect: Yes/No
- Default: Yes/No
- Enable End Users to choose Theme Style: Yes/No

Home URL: ?p=\&APP_ID::1\SESSION.
Login URL: ?p=\&APP_ID::LOGIN_DESKTOP\SESSION.

Theme: Universal Theme
Theme Style: Vita
Global Page:

About

Use the user interface details to define the specific settings for the selected user interface type. Attributes enable you to select the home and login pages that users are redirected to when they access this user interface. You can also set the current theme style and global page.

Using the Navigation Menu you can select the list and position used to render the navigation menu as well as setting the navigation menu list template.

Navigation Bar settings enable you to select list and list templates. Selecting classic implementation uses tabs instead of a list.

The JavaScript, Cascading Style Sheet and Concatenated Files attributes can be used to select alternate content delivery networks, load additional CSS files and to optimize the way JavaScript and CSS files are delivered.
Update session for user “nobody” to “ADMIN”
Session State – Page Rendering
There are different kinds of session state

1. Persisted Session State
2. In Memory Session State

Difference: is or is not stored in WWV_FLOW_DATA table
In Memory Session State

1. Automatic Row Fetch
2. Page item default value
3. Page item source value
Demo
Persisted Session State

1. Computation
2. PL/SQL Process

PL/SQL Code

```
:P4_ENAME := 'Scott';
```
Persisted Session State - When does the commit take place?

1. If item value has changed using apex_util.set_session_state -> immediately

2. If item value has changed -> end of block

3. If no item value has changed -> end of page rendering
Demo
Session State – Page Processing
What’s new in 5.1

1. Always via JSON (using XMLHttpRequest)

2. Reload on submit
Always submit via JSON

Processed via APEX_APPLICATION.ACCEPT

Original JSON in APEX_JSON variables
XMLHttpRequest page submits and the 32k limitation

People often mix up three limitations on 32K.
• 32K was the max size of a report row
• 32K is the max size of an APEX item
• mod_plsql limits the size of a single parameter that can be passed to a procedure to 32K. (ORDS does not have this limitation)

Doing asynchronous page submits in 5.1 only solves the limitation for mod_plsql because XMLHttpRequest supports chunked uploads.
Solutions for CLOBs (POST requests)

1. Use an editable Interactive Grid

2. `apex.ajax.clob`

```javascript
var ajaxClob = new apex.ajax.clob()
ajaxClob._set('very long text..')

select clob001
    from apex_collections
    where collection_name = 'CLOB_CONTENT';
```
Reload on submit

1. Always

2. Only for success
When **reload on submit** is set to **always**, APEX will process the page synchronous.
When **reload on submit** is set to **only for success**, APEX will process the page asynchronous.

The request returns a URL in JSON format

```javascript
apex.navigation.redirect( responseData.redirectURL );
```
Tip:
Do not use **Reload on Submit** set to **Only for Success** in combination with **Enable Duplicate Page Submissions** set to **No**

This will result in an error when first submit is not successful
Demo
Rejoin Sessions
Rejoin sessions

When is it useful?
• User is already working in application
• Opens a link to the same application in another tab
  □ Via link in email
  □ Via bookmark
  □ Other..
Application 106

Session Management

- Rejoin Sessions: Enabled for All Sessions
- Deep Linking: Enabled for All Sessions

Maximum Session Length in Seconds: Enabled for All Sessions

Session Timeout URL

Maximum Session Idle Time in Seconds: Enabled for All Sessions

Session Idle Timeout URL

History

Created: 4 hours ago - ADMIN
Rejoin sessions needs to be enabled in the Instance Administration
Demo
Session cloning
APEX session isolation between multiple browser tabs

When is it useful?
- When you depend on page/application items that are not part of page submission
- For example: an application item that holds a certain context value
  - Tab 1: Context is customer A
  - Tab 2: Context is customer B
- Security risks are not fully guaranteed yet, therefore option is disabled by default
Step 1: enable feature

begin
    apex_instance_admin.set_parameter(
        p_parameter => 'CLONE_SESSION_ENABLED',
        p_value => 'Y'
    );
end;
Step 2: add navigation bar list entry for this URL

f?p=&APP_ID.:&APP_PAGE_ID.:&APP_SESSION.:APEX_CLONE_SESSION
Demo
The logout process
Logout URL: &LOGOUT_URL.

apex_authentication.logout?p_app_id=106&p_session_id=6311950320799

Session is purged and no longer in apex_workspace_sessions view
<table>
<thead>
<tr>
<th>Sequence</th>
<th>Name</th>
<th>Parent Entry</th>
<th>Target</th>
<th>Conditional</th>
<th>Updated</th>
<th>Level</th>
<th>Authorization Scheme</th>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Log Out</td>
<td></td>
<td>&amp;LOGOUT_URL</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Session Not Valid**

Verify Function Name

- **Go To:**
  - Login Page
  - Built-In Login Page
  - URL

Authentication Scheme

Authentication is the process of establishing each user's identity before they can access your application. It may require a user to enter a user name and password or could involve the use of digital certificates or a secure key.

Use this page to create or edit Authentication Schemes and set the one which your application is using. The current Authentication Scheme determines how your application identifies and verifies the end user. Depending on the selected Authentication Scheme Type, you can use various settings and program hooks to fine-tune your application's authentication.

- Return to page
Session State Management

**Recent Sessions**
Review the recent sessions instance-wide, including username and workspace.

**Purge Sessions**
Purge session state information based on the age of the session.

**Session State Statistics**
Statistics on the number of sessions and cached entries.
You can purge all instance sessions

Purge Sessions

Maximum Sessions to Purge: 1000
Only Purge Sessions Older Than: 1 day

[Buttons: Cancel, Purge Sessions]
Thank you