

# Oracle APEX Social Login

Marc Sewtz

June 5th, 2019



## Step Up to Modern Cloud Development

# Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

# Contents

- 1 ➤ Authentication in Oracle APEX**
- 2 ➤ OAuth 2.0**
- 3 ➤ Using Google Authentication**
- 4 ➤ Using Facebook Authentication**
- 5 ➤ Using LinkedIn Authentication**
- 6 ➤ Authorization in Oracle APEX**

- 1** ➤ **Authentication in Oracle APEX**
- 2 ➤ OAuth 2.0
- 3 ➤ Using Google Authentication
- 4 ➤ Using Facebook Authentication
- 5 ➤ Using LinkedIn Authentication
- 6 ➤ Authorization in Oracle APEX

# APEX Authentication

## Establishing User Identity Through Authentication

- Authentication establishes the identity of each user who accesses your application.
- Most authentication processes require that a user provide some type of credentials such as a user name and password.

# APEX Authentication

## Establishing User Identity Through Authentication

- When creating an application, you can choose to:
  - **Not require authentication:** Oracle Application Express does not check any user credentials. All pages of your application are accessible to all users.
  - **Select a built-in authentication scheme:** Create an authentication method based on available preconfigured authentication schemes.
  - **Create custom authentication scheme:** Provided complete control over the authentication interface. To implement this approach, you must provide a PL/SQL function the APEX engine executes before processing each page request. This function's Boolean return value determines whether the Application Express engine processes the page normally or displays a failure page.

# APEX Authentication

## Preconfigured Authentication Schemes

- **Application Express Accounts:** User accounts that are created within and managed in the Oracle Application Express user repository. When you use this method, your application is authenticated against these accounts.
- **Custom Authentication:** Creating a Custom Authentication scheme from scratch to have complete control over your authentication interface.
- **Database Accounts:** Database Account Credentials authentication utilizes database schema accounts to authenticate users.
- **HTTP Header Variable:** Authenticate users externally by storing the username in a HTTP Header variable set by the web server.
- **LDAP Directory Verification:** Authenticate a user and password with an authentication request to a LDAP server.

# APEX Authentication

## Preconfigured Authentication Schemes

- **No Authentication (using DAD):** Adopts the current database user. This approach can be used in combination with a `mod_plsql` Database Access Descriptor (DAD) configuration that uses basic authentication to set the database session user.
- **Open Door Credentials:** Enable anyone to access your application using a built-in login page that captures a user name.
- **Oracle Single Sign-On :** Delegates authentication to the Oracle Application Server Single Sign-On (SSO) Server. To use this authentication scheme, your site must have been registered as a partner application with the SSO server.
- **Social Sign-In:** Supports authentication with Google, Facebook, and other social network that supports OpenID Connect or OAuth2 standards.

# APEX Authentication

## Social Log In

- Supports authentication with Google, Facebook, and other social network that supports OpenID Connect or OAuth2 standards.
- Social Login for application was introduced in APEX 18.1
- Social Login for the Builder / APEX IDE is being added in APEX 19.1
- Social Login uses Web Credentials, which are stored on the Workspace level
- Web Credentials are not included in the export of an application
- When installing applications that use Social Login, you will be prompted to provide the Client ID (username) and Client Secret (password)

# APEX Authentication

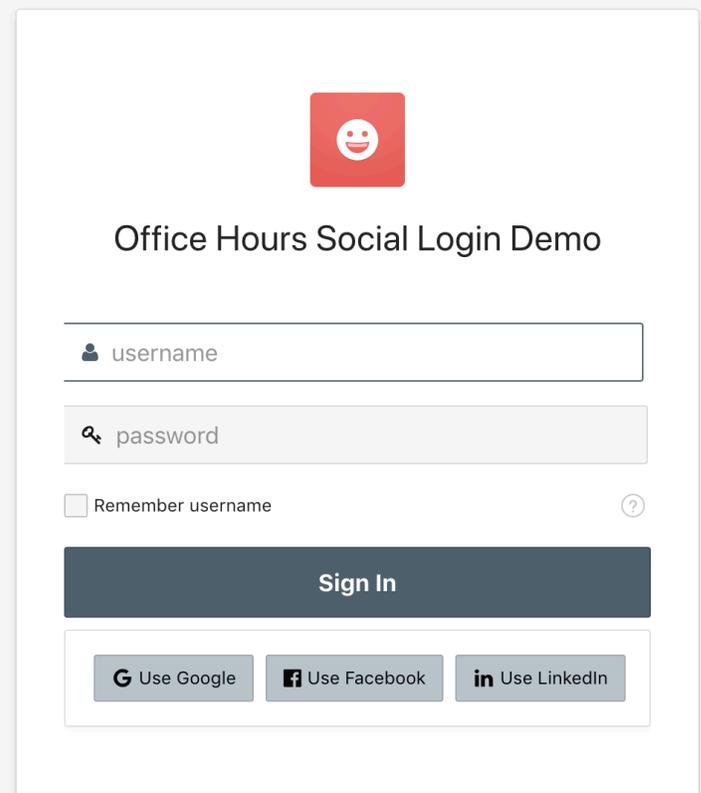
## Deep Linking

- Applications that use authentication schemes support deep linking.
- Deep linking refers to the ability to link to an APEX page out of context, e.g. from a hyperlink in an email or workflow notification.
- It can be enabled in “***Session Management***” in the application’s security attributes.
- When you link to a page out of context and the application requires the user be authenticated, the user is taken to the login page.
- After credentials verification, APEX automatically displays the page that was referenced in the original link.

# APEX Authentication

## Using multiple authentication schemes in one app

- Some applications may require more than one authentication scheme
- For example, if you wish to allow your users to login with a social network account of their choice (e.g. Google, Facebook, LinkedIn, etc.).
- To enable multiple authentication schemes:
  - Enable “Switch in Session” (see “login processing” in your authentication scheme)
  - If enabled, the current session's authentication scheme can be changed by passing *APEX\_AUTHENTICATION=scheme name* in a URL's request parameter.



The screenshot shows a login interface for 'Office Hours Social Login Demo'. At the top center is a red square icon with a white smiley face. Below the icon is the title 'Office Hours Social Login Demo'. The form contains a 'username' input field with a user icon, a 'password' input field with a magnifying glass icon, and a 'Remember username' checkbox with a question mark icon. A dark grey 'Sign In' button is positioned below the input fields. At the bottom, there are three buttons: 'Use Google' with the Google logo, 'Use Facebook' with the Facebook logo, and 'Use LinkedIn' with the LinkedIn logo.

- 1 Authentication in Oracle APEX
- 2 OAuth 2.0**
- 3 Using Google Authentication
- 4 Using Facebook Authentication
- 5 Using LinkedIn Authentication
- 6 Authorization in Oracle APEX

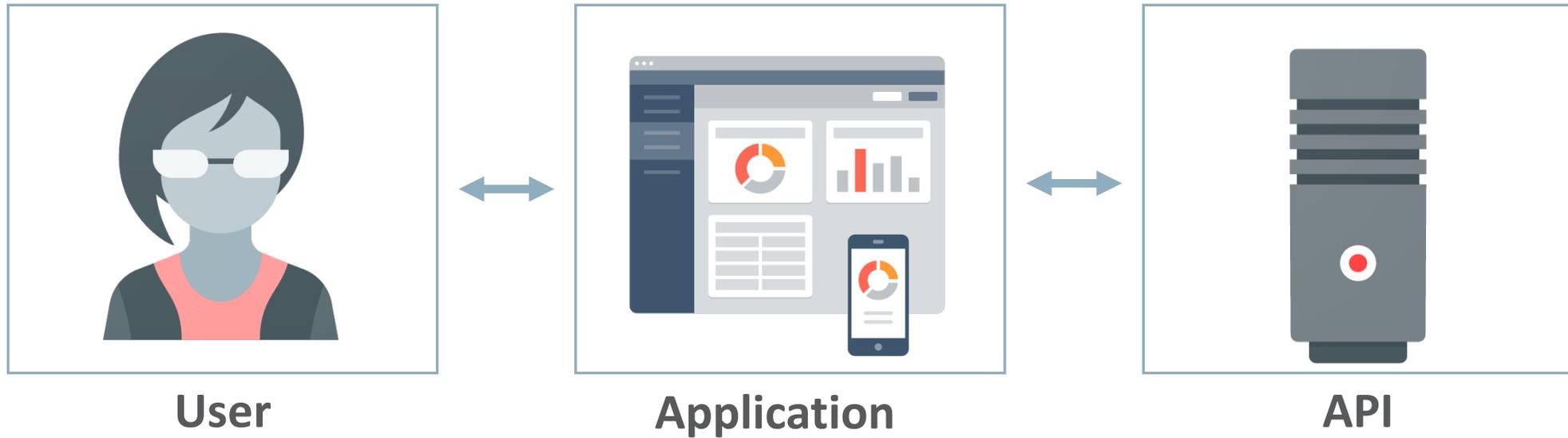
# OAuth 2.0

## Overview

- OAuth 2.0 is an authorization framework that enables applications to obtain limited access to user accounts on an HTTP service, such as Facebook, Google, LinkedIn, Azure, Okta, etc.
- It works by delegating user authentication to the service that hosts the user account, and authorizing third-party applications to access the user account.
- OAuth 2 provides authorization flows for web applications, desktop applications and mobile devices.

# OAuth 2.0

## OAuth 2.0 Roles



# OAuth 2.0

## OAuth 2.0 Roles

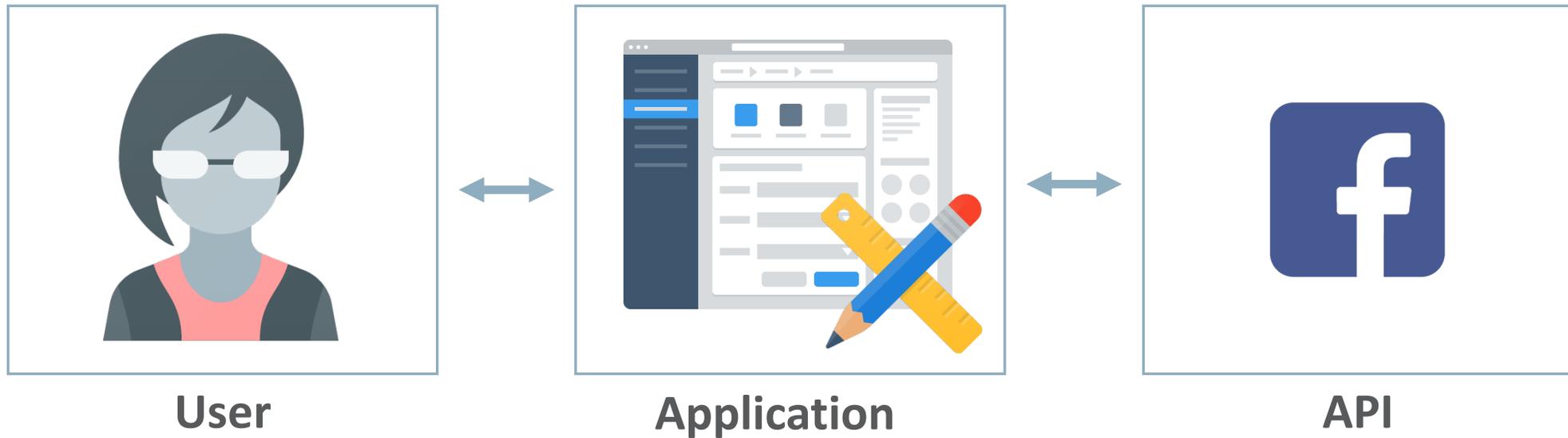
The image shows a screenshot of the OpenTable website with a sign-in modal form overlaid. The modal form is titled "Please sign in" and contains the following elements:

- Input field for "Email"
- Input field for "Password"
- A link for "Forgot Password?"
- A red "Sign In" button
- A section titled "Don't want to complete the form?" with two options: "Continue with Facebook" and "Continue with Google"
- A link for "New to OpenTable? Create an account"

The background of the website shows the OpenTable logo, navigation links for "Home", "United States", "New York / Tri-State Area", and "Brooklyn", and a search bar with a date of "Mar 3, 2019" and a time of "7:00 PM". There are also buttons for "Sign up" and "Sign in" in the top right corner.

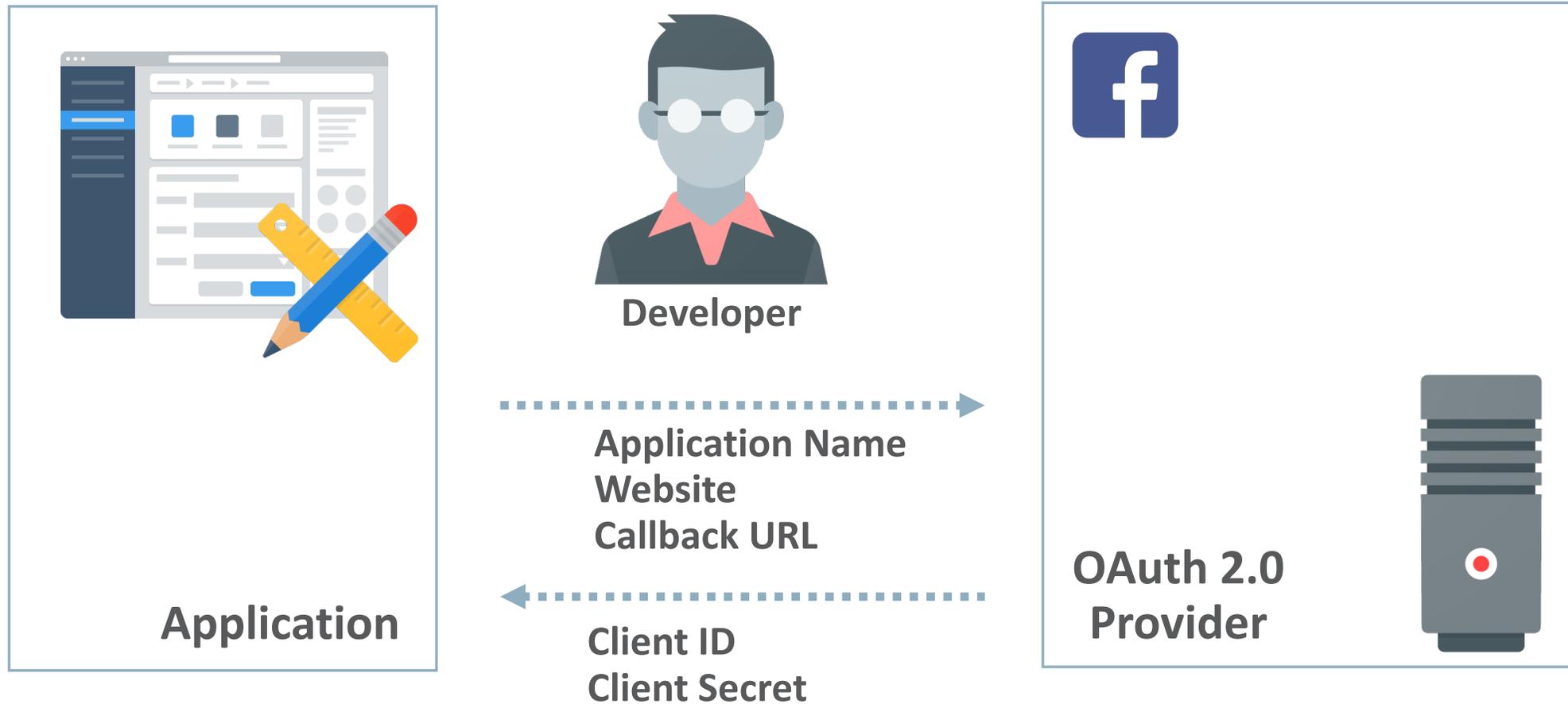
# OAuth 2.0

## OAuth 2.0 Roles



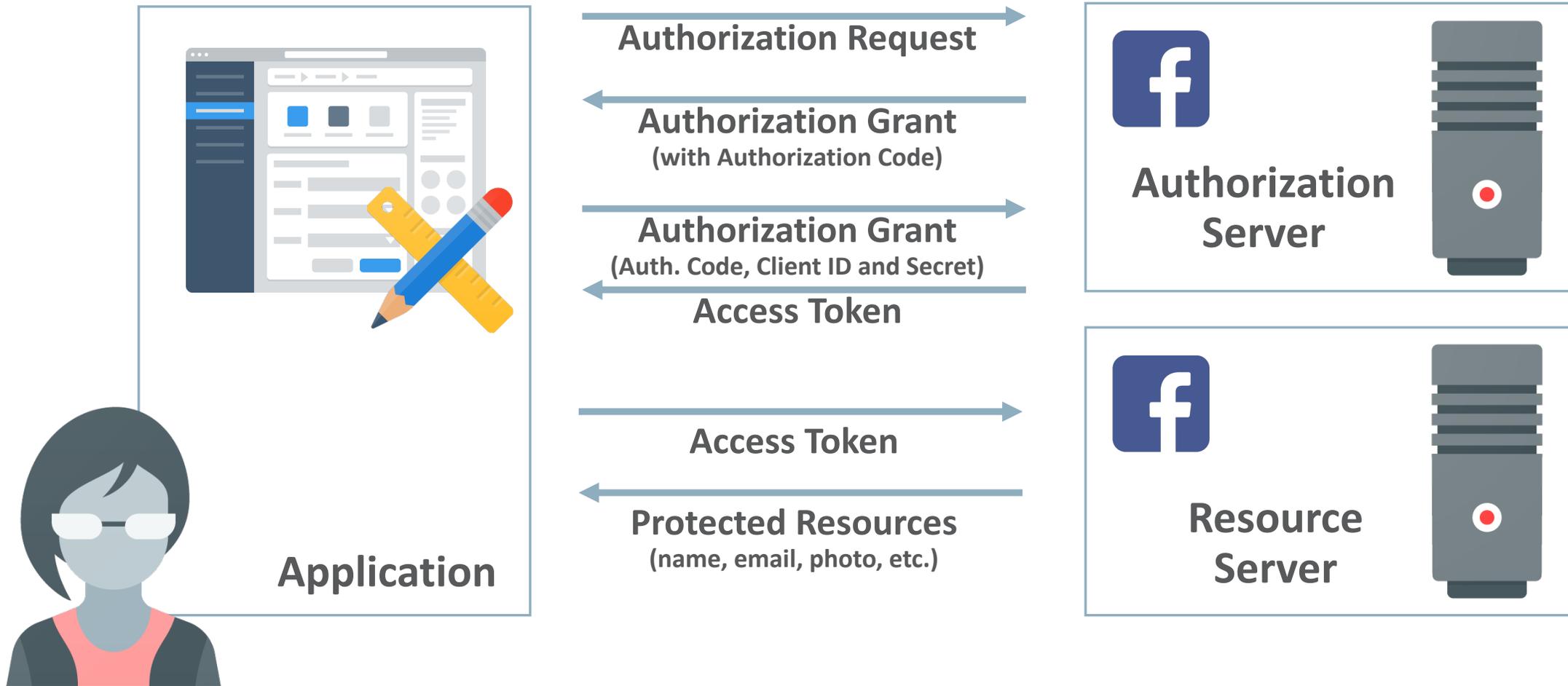
# OAuth 2.0

## Register Application with OAuth 2.0 provider



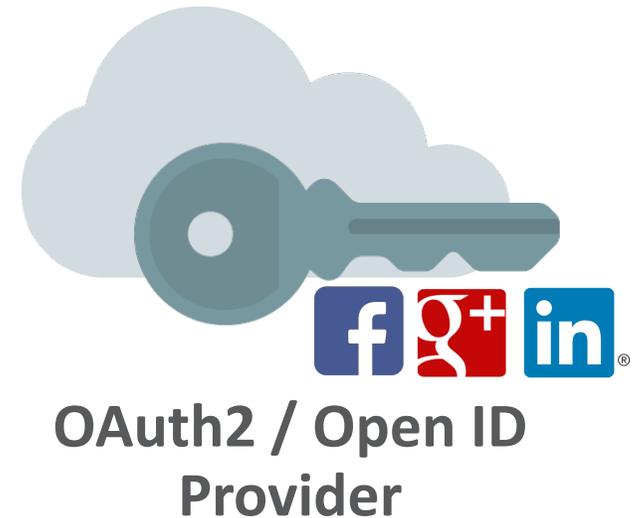
# OAuth 2.0

## OAuth 2.0 Flow



# Social Login Authentication Flow

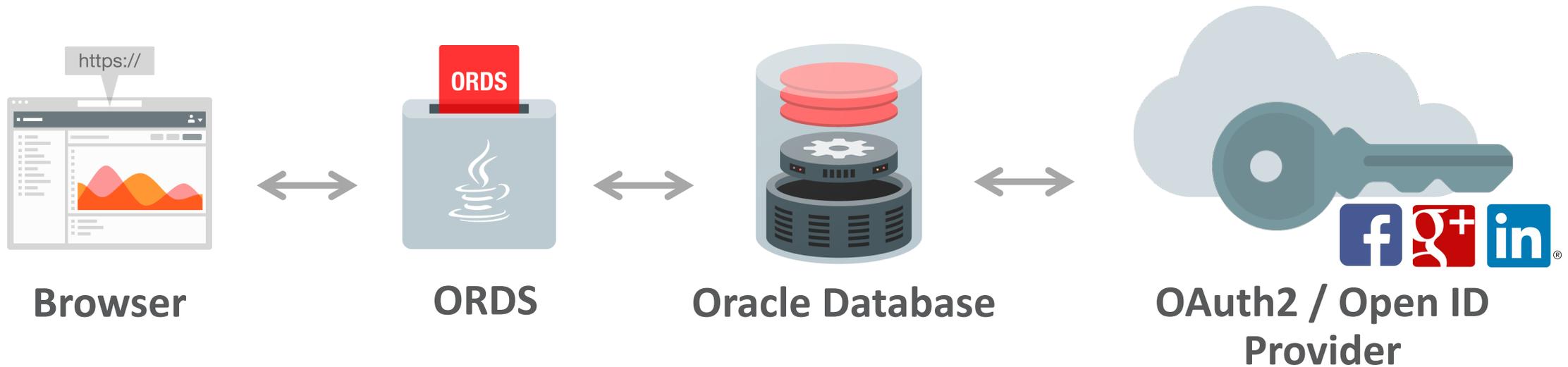
## Register Application with OAuth 2.0 provider



- APEX app is registered as OAuth2 client with the provider (Google, Facebook, etc.).
- Provider generates Client ID (username) and client secret (password), which are stored as “Web Credentials” on APEX workspace level.

# Social Login Authentication Flow

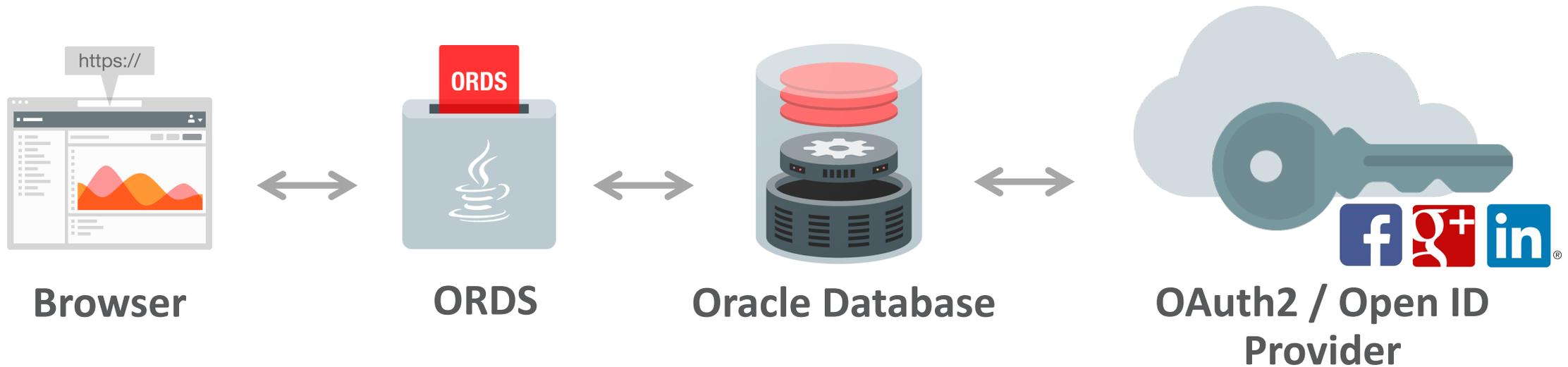
## Communication Flow in APEX



- The user opens APEX app in browser:  
e.g. <https://apexea.oracle.com/ords/f?p=111:1>

# Social Login Authentication Flow

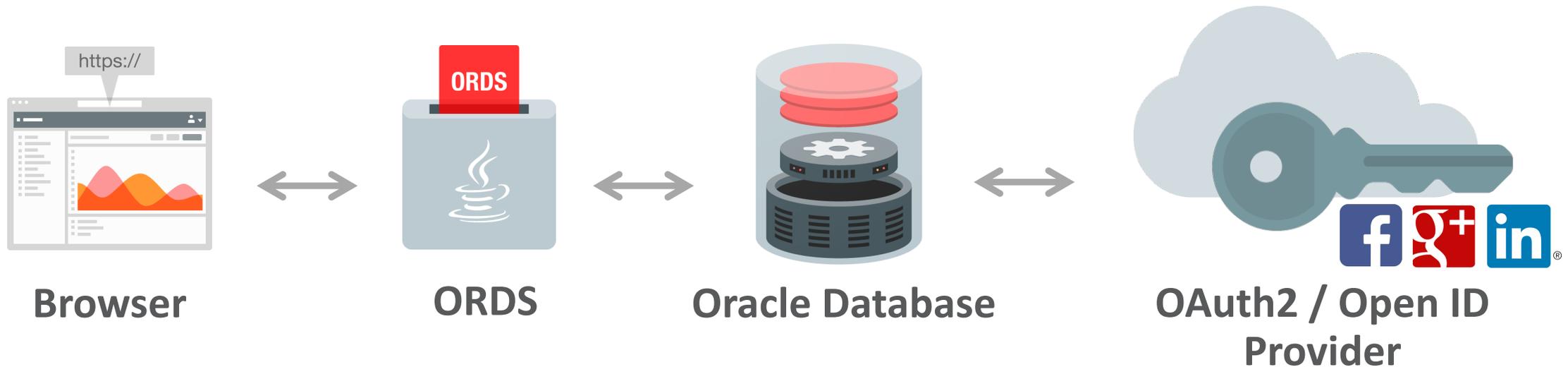
## Communication Flow in APEX



- APEX creates a session for “nobody” and checks if the page requires authentication.
- If authentication is required, details of the current authentication scheme are retrieved from the metadata.

# Social Login Authentication Flow

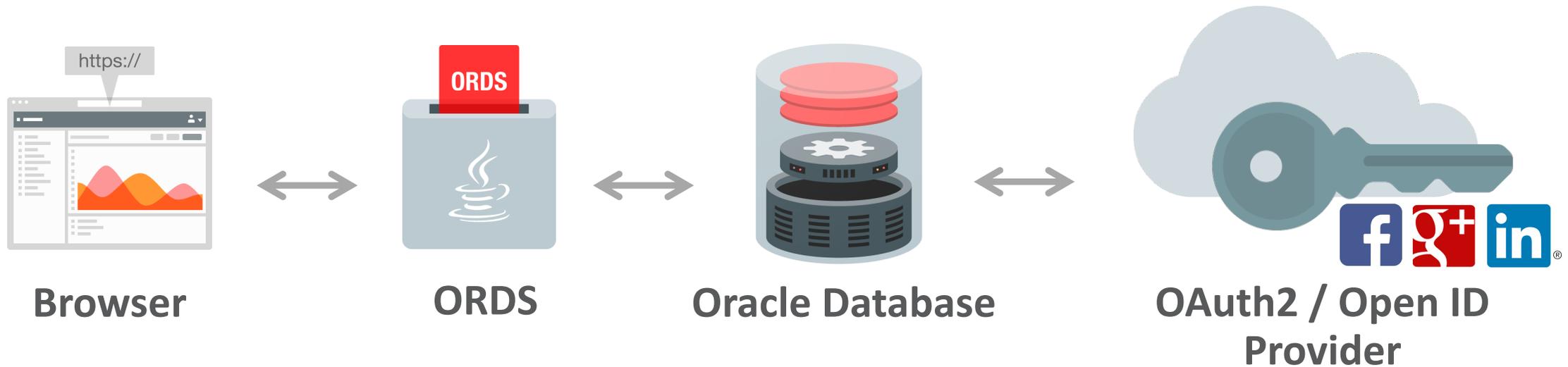
## Communication Flow in APEX



- The user is redirected to the authorization endpoint URL.
  - When using generic OpenID, the discovery URL is used to retrieve the ***endpoint/token/user info URL*** with its parameters.
  - When using generic OAuth2 provider, these URLs have to be entered in the authentication scheme definition in APEX.
  - When using Facebook or Google, the URLs are pre-define in the authentication scheme.

# Social Login Authentication Flow

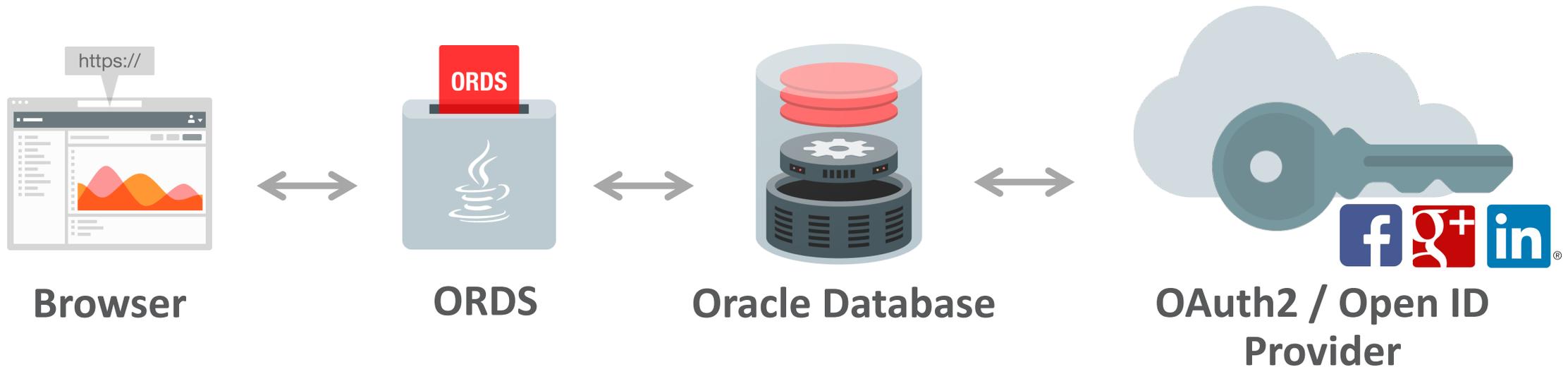
## Communication Flow in APEX



- The user logs in through the providers login page, using **username** and **password**.
- The user may also need to **explicitly provide consent** to access to resources like email and photo.

# Social Login Authentication Flow

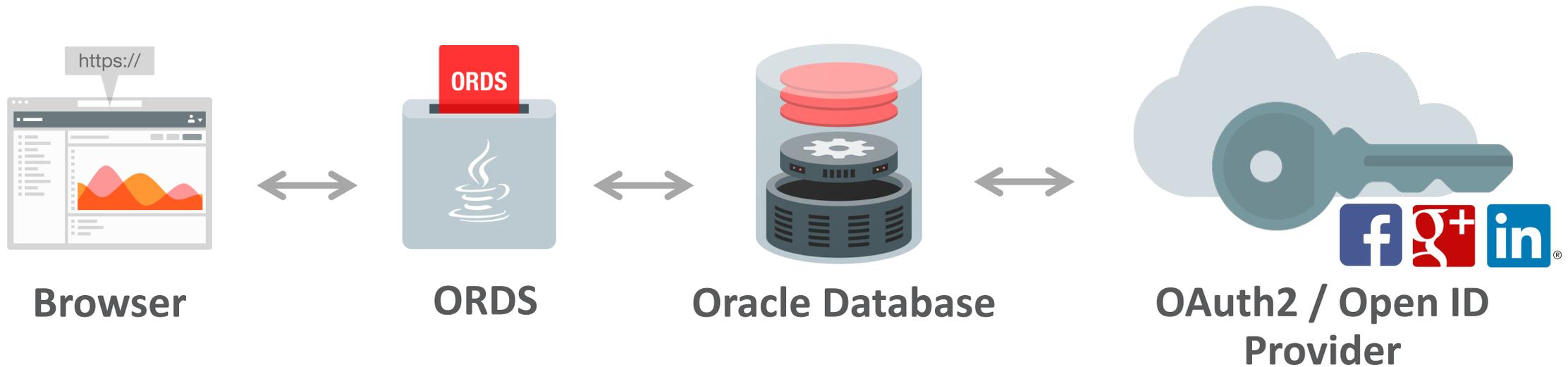
## Communication Flow in APEX



- After successful authentication, the providers returns an **authorization code**.
- Browser is redirected, along with the authorization code, to the ***APEX callback URL***:  
[https://apex.oracle.com/ords/apex\\_authentication.callback](https://apex.oracle.com/ords/apex_authentication.callback)

# Social Login Authentication Flow

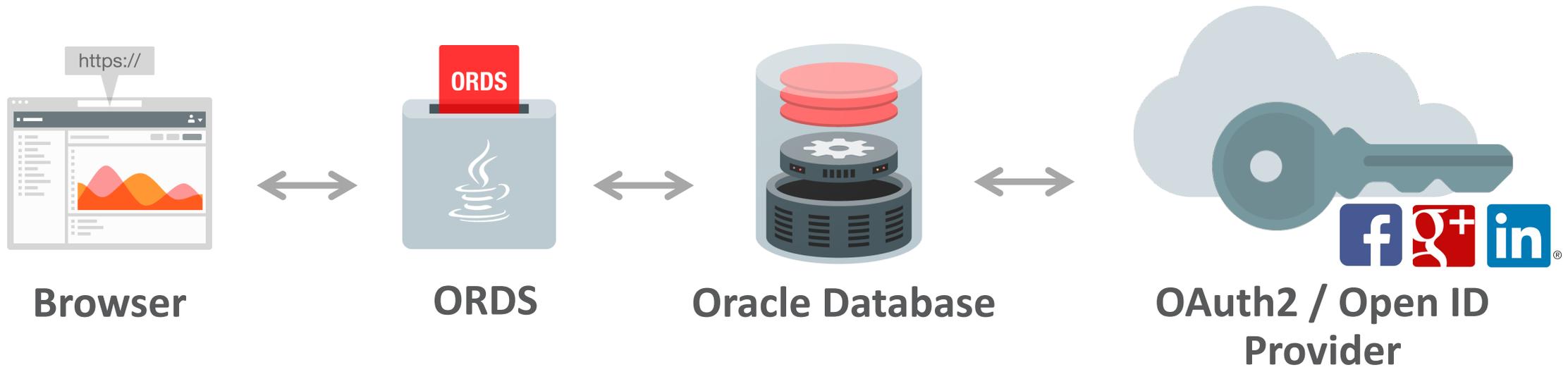
## Communication Flow in APEX



- APEX requests an access token from the ***token endpoint URL***, passing the authorization code, client ID and client secret.
- Provider returns an access token (plus refresh token and optional ID token).
- All tokens are stored in APEX.

# Social Login Authentication Flow

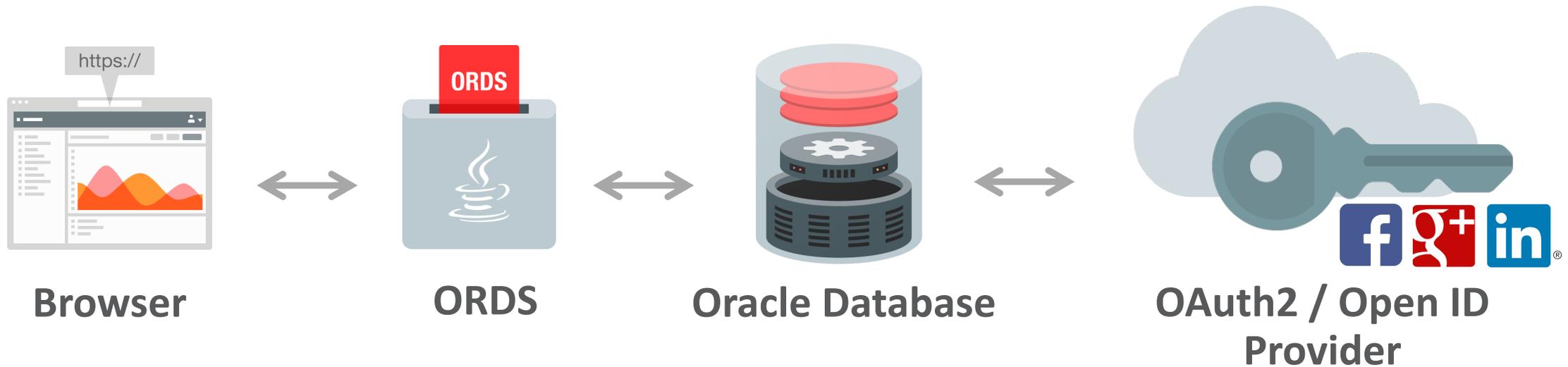
## Communication Flow in APEX



- Based on the “scope” attribute defined in the authentication scheme, the user info endpoint URL is used to get additional information (e.g. photo) about the user.
- Information is returned as JSON.
- All attributes (incl. ID- & access token) are stored in `APEX_JSON.G_VALUES`.

# Social Login Authentication Flow

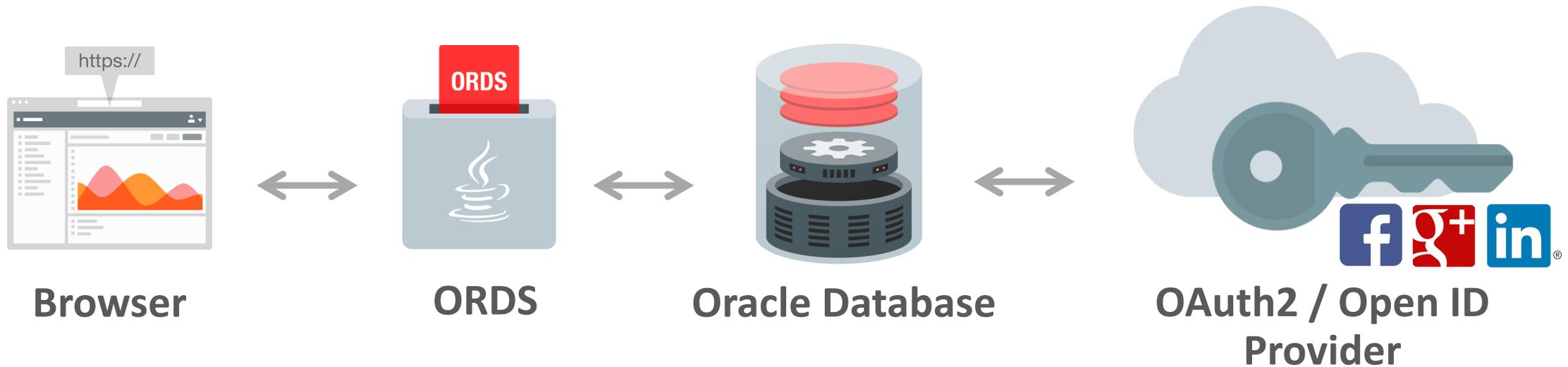
## Communication Flow in APEX



- APEX retrieves the username from the tokens (access/ID).
- APEX authenticates user in the session and redirects to the originally requested page.

# Social Login Authentication Flow

## Communication Flow in APEX



- All subsequent browser requests for APEX pages include a session cookie, which APEX uses to identify the user.
- While the APEX session is valid, requested pages are loaded.
- The access token is not checked for expiration.
- The refresh token has no usage for the authentication process.

# Demo

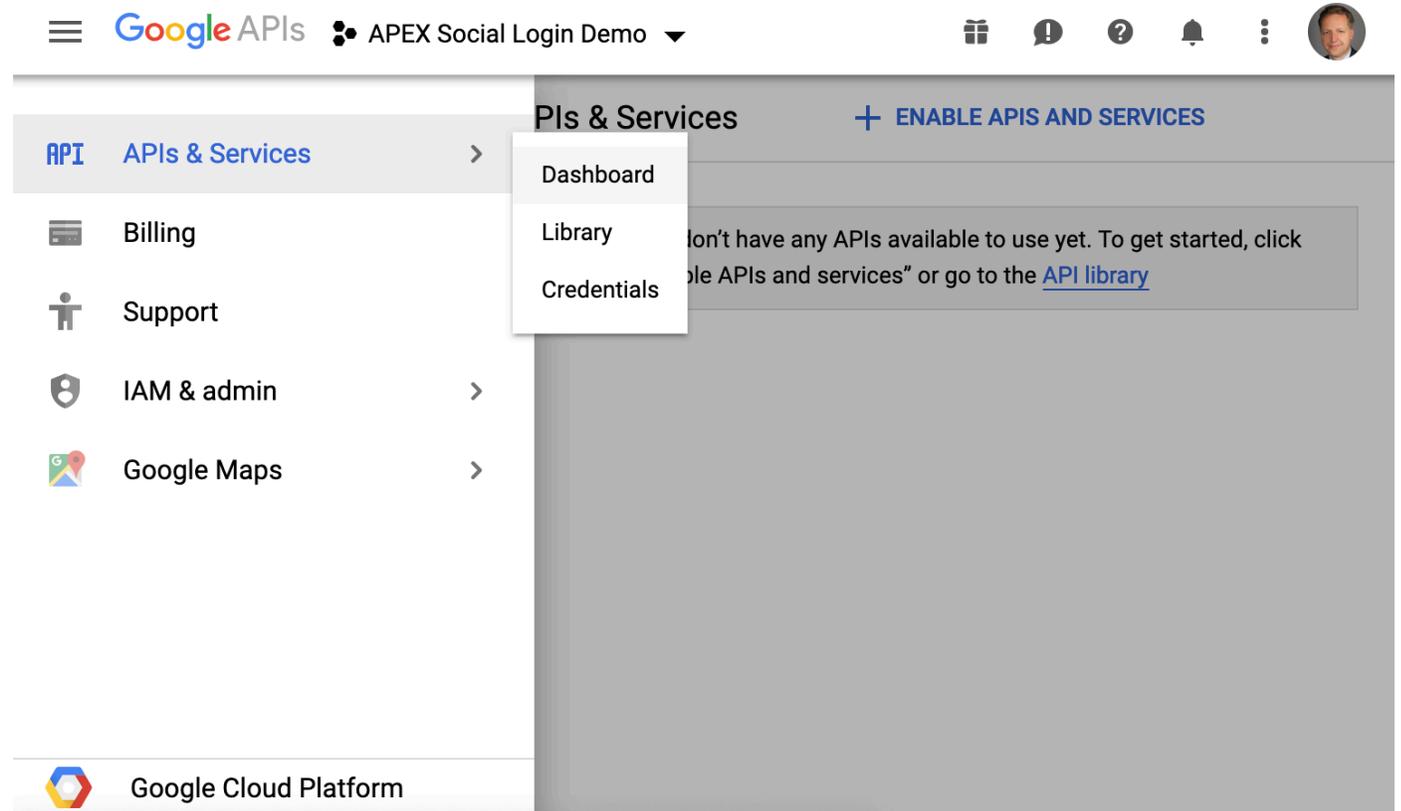
[http://bit.ly/apex\\_social\\_login\\_demo](http://bit.ly/apex_social_login_demo)

- 1 Authentication in Oracle APEX
- 2 OAuth 2.0
- 3 Using Google Authentication**
- 4 Using Facebook Authentication
- 5 Using LinkedIn Authentication
- 6 Authorization in Oracle APEX

# Using Google Authentication

## Configure Google APIs “Project” to use Google Authentication

- Google APIs Developer Console: <https://console.developers.google.com>
- Create a “Project” from the drop-down on top of the page
- Select “Credentials” in menu on the left



# Using Google Authentication

## Client ID

- Create OAuth 2.0 client IDs
- Requires name of OAuth Client ID and Authorized redirect URIs:  
[https://apexea.oracle.com/pls/apex/apex\\_authentication.callback](https://apexea.oracle.com/pls/apex/apex_authentication.callback)
- Provides Client ID and Client Secret, which are used when creating Web Credentials in APEX

The screenshot shows the Google APIs console interface for configuring a Client ID for a Web application. The page title is "Client ID for Web application" and it includes action buttons for "DOWNLOAD JSON", "RESET SECRET", and "DELETE".

Client ID	[REDACTED]
Client secret	[REDACTED]
Creation date	Feb 12, 2019, 1:30:39 PM

**Name** ?  
apex\_social\_login\_demo

**Restrictions**  
Enter JavaScript origins, redirect URIs, or both [Learn More](#)  
Origins and redirect domains must be added to the list of Authorized Domains in the [OAuth consent settings](#).

**Authorized JavaScript origins**  
For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (https://\*.example.com) or a path (https://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

https://www.example.com  
Type in the domain and press Enter to add it

**Authorized redirect URIs**  
For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

https://apexea.oracle.com/pls/apex/apex\_authentication.callback

https://www.example.com  
Type in the domain and press Enter to add it

**Save** **Cancel**

# Using Google Authentication

## Create Web Credentials in APEX

- OAuth2 Client ID and Secret are stored in APEX Web Credentials
- Web Credentials are managed on the Workspace level

ORACLE App Builder SQL Workshop Team Development App Gallery

Workspace Utilities \ Web Credentials \ Create/Edit

Web Credentials Cancel Delete Apply Changes

Attributes

\* Name  ?

Static Identifier  ?

Authentication Type  ?

OAuth Scope  ?

Client ID or Username  ?

Client Secret or Password  ?

Verify Client Secret or Password  ?

Prompt On Install  Yes ?

Comments

Web Credentials

Store authentication credentials for external REST services or REST Enabled SQL services.

The Client Secret will be stored encrypted, can only be used by Application Express and not be retrieved in clear text.

Also, credential information will not be included in application export files. After importing an application into the target workspace, prompts will be displayed to re-enter the credentials.

# Using Google Authentication

## Google Authentication Scheme

- In your APEX app, create “Social Sign-In” authentication scheme
- Select your previously created Web Credentials
- Select “Google” as Authentication Provider
- Define Scope, Username Attribute and Additional User Attributes

The screenshot shows the Oracle APEX interface for configuring an authentication scheme. The breadcrumb trail is: Application 20000 > Shared Components > Authentication Schemes > Create / Edit. The page title is "Authentication Scheme".

**Name**

\* Name:  ?

\* Scheme Type:  ?

**Subscription**

Reference Master Authentication Scheme From:  ?  Refresh

**This is the "master" copy of this authentication scheme.**

There are no subscribers to this authentication scheme.

**Settings**

Credential Store:  ?

Authentication Provider:  ?

\* Scope:  ?

Authentication URI Parameters:  ?

\* Username Attribute:  ?

Convert Username To Upper Case:  ?

Additional User Attributes:  ?

# Demo

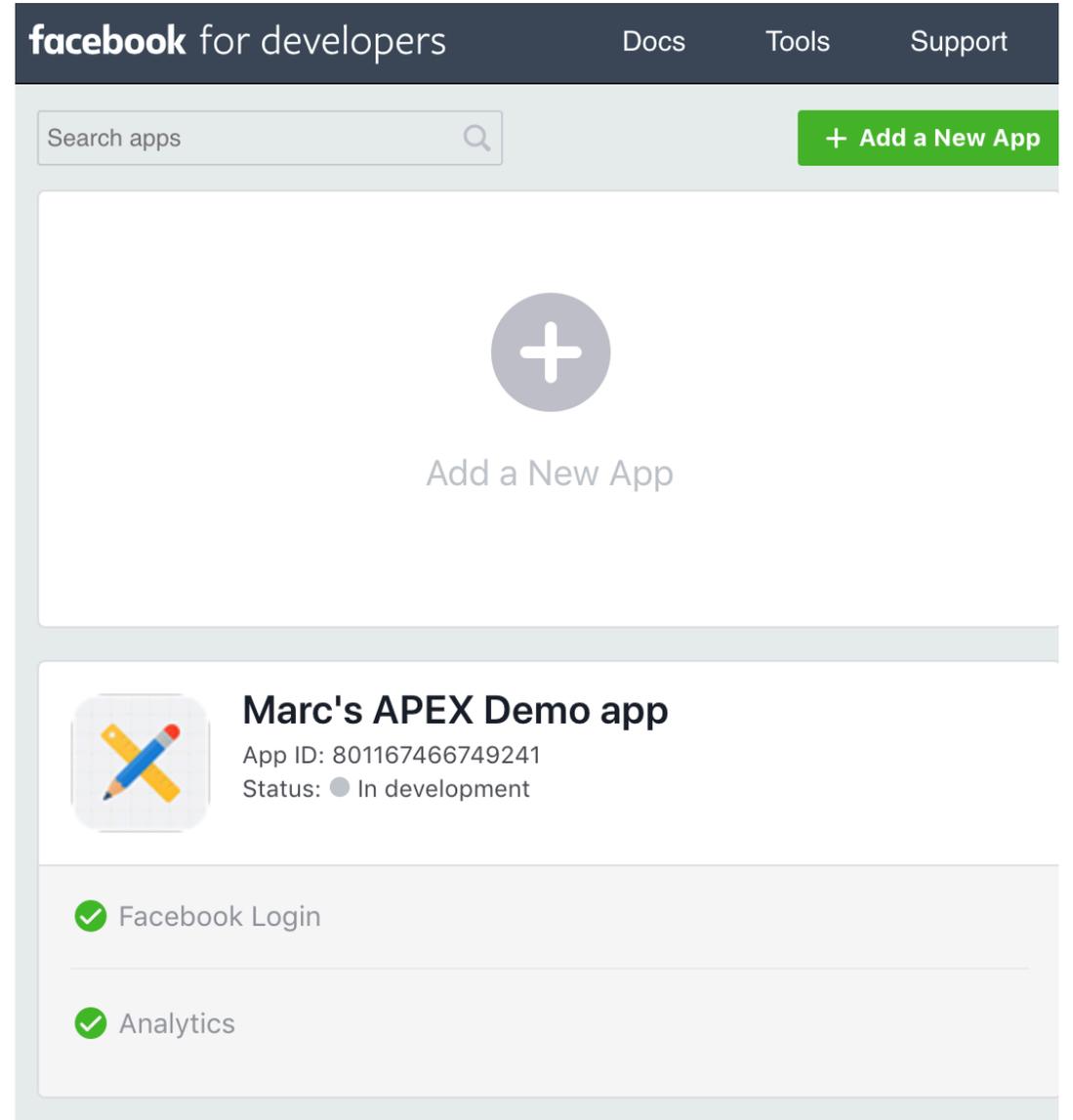
## Google Authentication

- 1 Authentication in Oracle APEX
- 2 OAuth 2.0
- 3 Using Google Authentication
- 4 Using Facebook Authentication**
- 5 Using LinkedIn Authentication
- 6 Authorization in Oracle APEX

# Using Facebook Authentication

## Create App to use Facebook Authentication

- Go to Facebook for Developers:  
<https://developers.facebook.com/apps>
- Select “Add a new App”

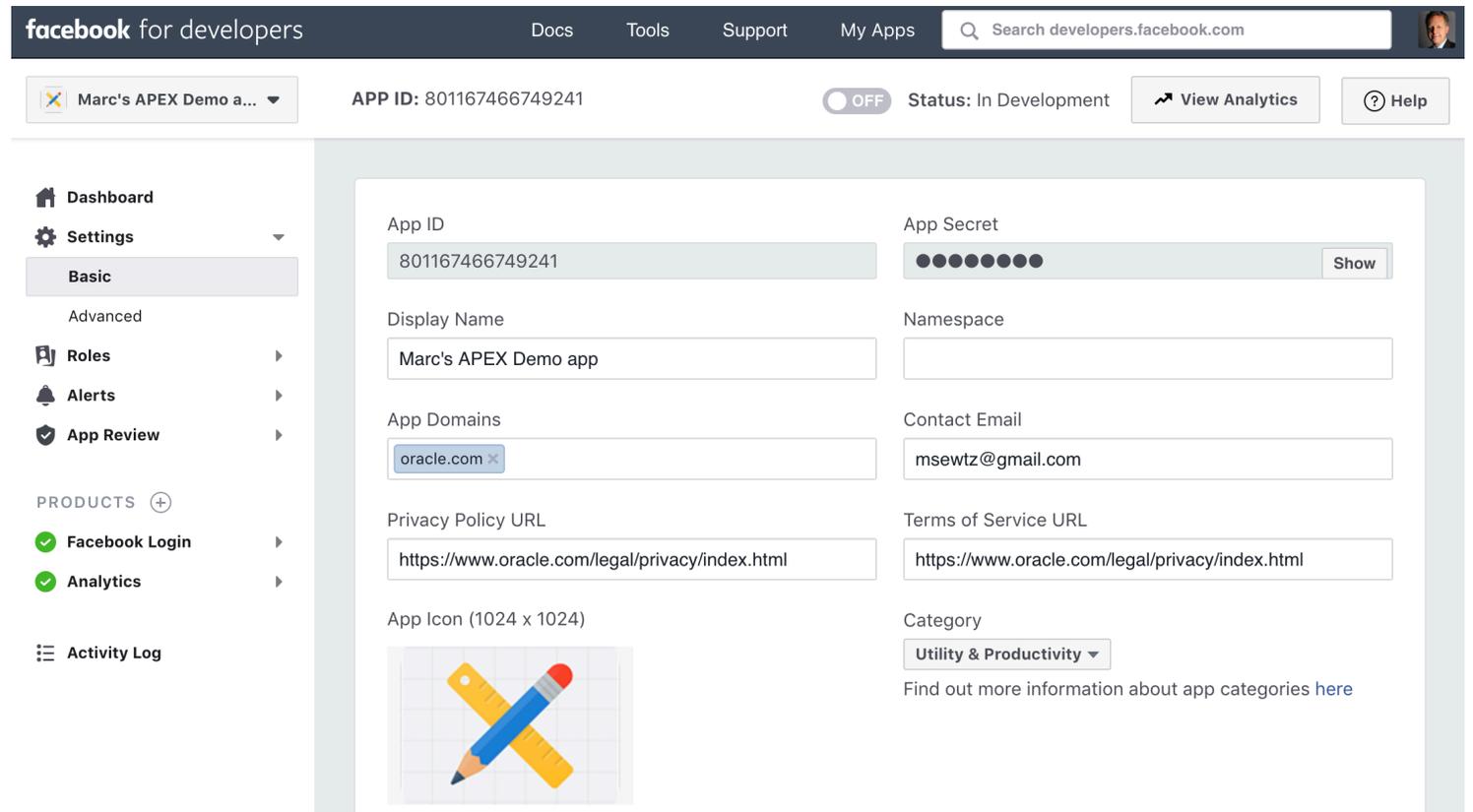


The screenshot shows the 'facebook for developers' interface. At the top, there are navigation links for 'Docs', 'Tools', and 'Support'. Below this is a search bar labeled 'Search apps' and a green button labeled '+ Add a New App'. The main content area features a large grey circle with a white plus sign and the text 'Add a New App'. Below this, there is a card for 'Marc's APEX Demo app' with an icon of two crossed pencils. The card displays 'App ID: 801167466749241' and 'Status: ● In development'. At the bottom of the card, there are two green checkmarks indicating that 'Facebook Login' and 'Analytics' are enabled.

# Using Facebook Authentication

## App Definition

- Define App name, icon, domains, contact email, privacy and terms of service URLs
- Copy App ID and App Secret, those are used in APEX Web Credentials for Client ID and Client Secret



The screenshot shows the Facebook for developers app configuration interface. The top navigation bar includes 'facebook for developers', 'Docs', 'Tools', 'Support', 'My Apps', and a search bar. The main header displays the app name 'Marc's APEX Demo a...', the 'APP ID: 801167466749241', a status toggle set to 'OFF', 'Status: In Development', and buttons for 'View Analytics' and 'Help'.

The left sidebar contains navigation options: Dashboard, Settings (Basic and Advanced), Roles, Alerts, App Review, PRODUCTS (Facebook Login and Analytics), and Activity Log.

The main content area is divided into two columns of configuration fields:

- App ID:** 801167466749241
- App Secret:** Masked with dots, with a 'Show' button.
- Display Name:** Marc's APEX Demo app
- Namespace:** (Empty field)
- App Domains:** oracle.com
- Contact Email:** msewtz@gmail.com
- Privacy Policy URL:** https://www.oracle.com/legal/privacy/index.html
- Terms of Service URL:** https://www.oracle.com/legal/privacy/index.html
- App Icon (1024 x 1024):** A placeholder image showing a blue pencil and a yellow ruler.
- Category:** Utility & Productivity

A link at the bottom right of the configuration area reads: 'Find out more information about app categories here'.

# Using Facebook Authentication

## App Definition

- New to also define “Facebook Login” Settings (see products)
- This is where the APEX redirect URL is stored:  
[https://apexea.oracle.com/pls/apex/apex\\_authentication.callback](https://apexea.oracle.com/pls/apex/apex_authentication.callback)

The screenshot shows the Facebook for developers dashboard for an application named "Marc's APEX Demo a...". The top navigation bar includes "facebook for developers", "Docs", "Tools", "Support", "My Apps", and a search bar. The application ID is 801167466749241, and the status is "In Development". A blue banner at the top right contains the text "Visit our reference on recent Facebook Login updates." with a close button.

The left sidebar contains navigation options: Dashboard, Settings, Roles, Alerts, App Review, PRODUCTS (+), Facebook Login, Settings (selected), Quickstart, Analytics, and Activity Log.

The main content area is titled "Client OAuth Settings" and contains several toggle switches and text fields:

- Client OAuth Login**: Enabled (Yes). Description: "Enables the standard OAuth client token flow. Secure your application and prevent abuse by locking down which token redirect URIs are allowed with the options below. Disable globally if not used. [?]"
- Web OAuth Login**: Enabled (Yes). Description: "Enables web-based Client OAuth Login. [?]"
- Enforce HTTPS**: Enabled (Yes). Description: "Enforce the use of HTTPS for Redirect URIs and the JavaScript SDK. Strongly recommended. [?]"
- Force Web OAuth Reauthentication**: Disabled (No). Description: "When on, prompts people to enter their Facebook password in order to log in on the web. [?]"
- Embedded Browser OAuth Login**: Disabled (No). Description: "Enable webview Redirect URIs for Client OAuth Login. [?]"
- Use Strict Mode for Redirect URIs**: Enabled (Yes). Description: "Only allow redirects that use the Facebook SDK or that exactly match the Valid OAuth Redirect URIs. Strongly recommended. [?]"

At the bottom, there is a section for "Valid OAuth Redirect URIs" with a text input field containing the URL: `https://apexea.oracle.com/pls/apex/apex_authentication.callback`.

# Using Facebook Authentication

## Create Web Credentials in APEX

- Facebook App ID and App Secret are stored as Client ID and Client Secret in APEX Web Credentials
- Web Credentials are managed on the Workspace level

ORACLE® App Builder SQL Workshop Team Development App Gallery

Workspace Utilities \ Web Credentials \ Create/Edit

Web Credentials Cancel Delete Apply Changes Clear Tokens

Attributes

\* Name  ?

Static Identifier  ?

Authentication Type  ?

OAuth Scope  ?

Client ID or Username  ?

Client Secret or Password  ?

Verify Client Secret or Password  ?

Prompt On Install **Yes** ?

Comments

# Using Facebook Authentication

## Facebook Authentication Scheme

- In your APEX app, create “Social Sign-In” Auth. scheme
- Select your previously created Web Credentials
- Select “Facebook” as Authentication Provider
- Define Scope, Username Attribute and Additional User Attributes

Authentication Scheme Cancel Delete Make Current Scheme Apply Changes

Show All **Name** Subscripi... Settings Source Session N... Login Proc... Post-Logo... Session S... Comments

Name

\* Name  ?

\* Scheme Type  ?

Subscription

Reference Master Authentication Scheme From  ?  Refresh

**This is the "master" copy of this authentication scheme.**

There are no subscribers to this authentication scheme.

Settings

Credential Store  ?

Authentication Provider  ?

\* Scope  ?

Authentication URI Parameters  ?

\* Username Attribute  ?

Convert Username To Upper Case  ?

Additional User Attributes  ?

# Demo

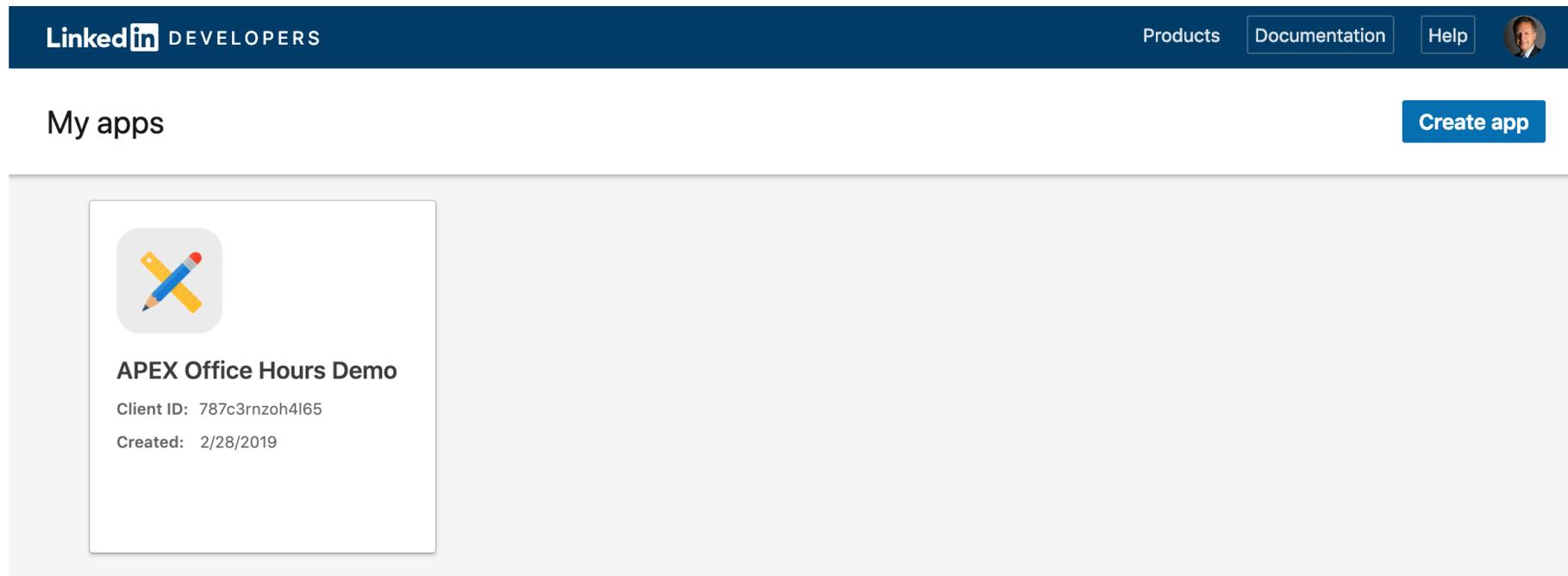
## Facebook Authentication

- 1 Authentication in Oracle APEX
- 2 OAuth 2.0
- 3 Using Google Authentication
- 4 Using Facebook Authentication
- 5 Using LinkedIn Authentication**
- 6 Authorization in Oracle APEX

# Using LinkedIn Authentication

## Create App to use Facebook Authentication

- Go to LinkedIn for Developers: <https://www.linkedin.com/developers/apps>
- Select “Create App”

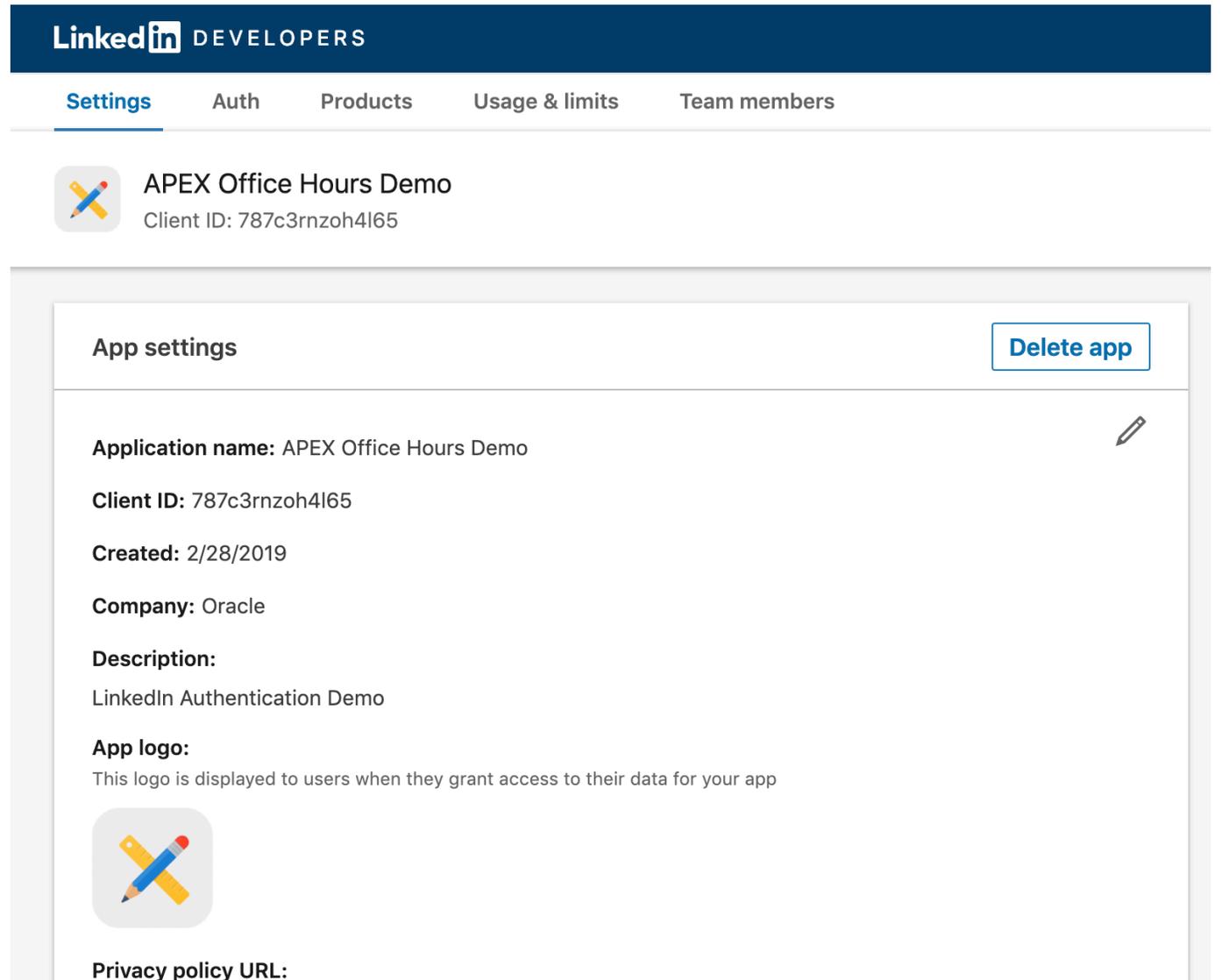


The screenshot displays the LinkedIn Developers interface. At the top, a dark blue navigation bar contains the LinkedIn logo and the word 'DEVELOPERS' on the left, and 'Products', 'Documentation', and 'Help' on the right. Below this, the main content area is titled 'My apps' and features a prominent blue 'Create app' button. A single application card is visible, titled 'APEX Office Hours Demo'. The card includes a logo of two crossed pencils and provides the following details: Client ID: 787c3rnzoh4l65 and Created: 2/28/2019.

# Using LinkedIn Authentication

## App Settings

- Define Application Name, Logo, Description, Business Email, Privacy Policy and Terms



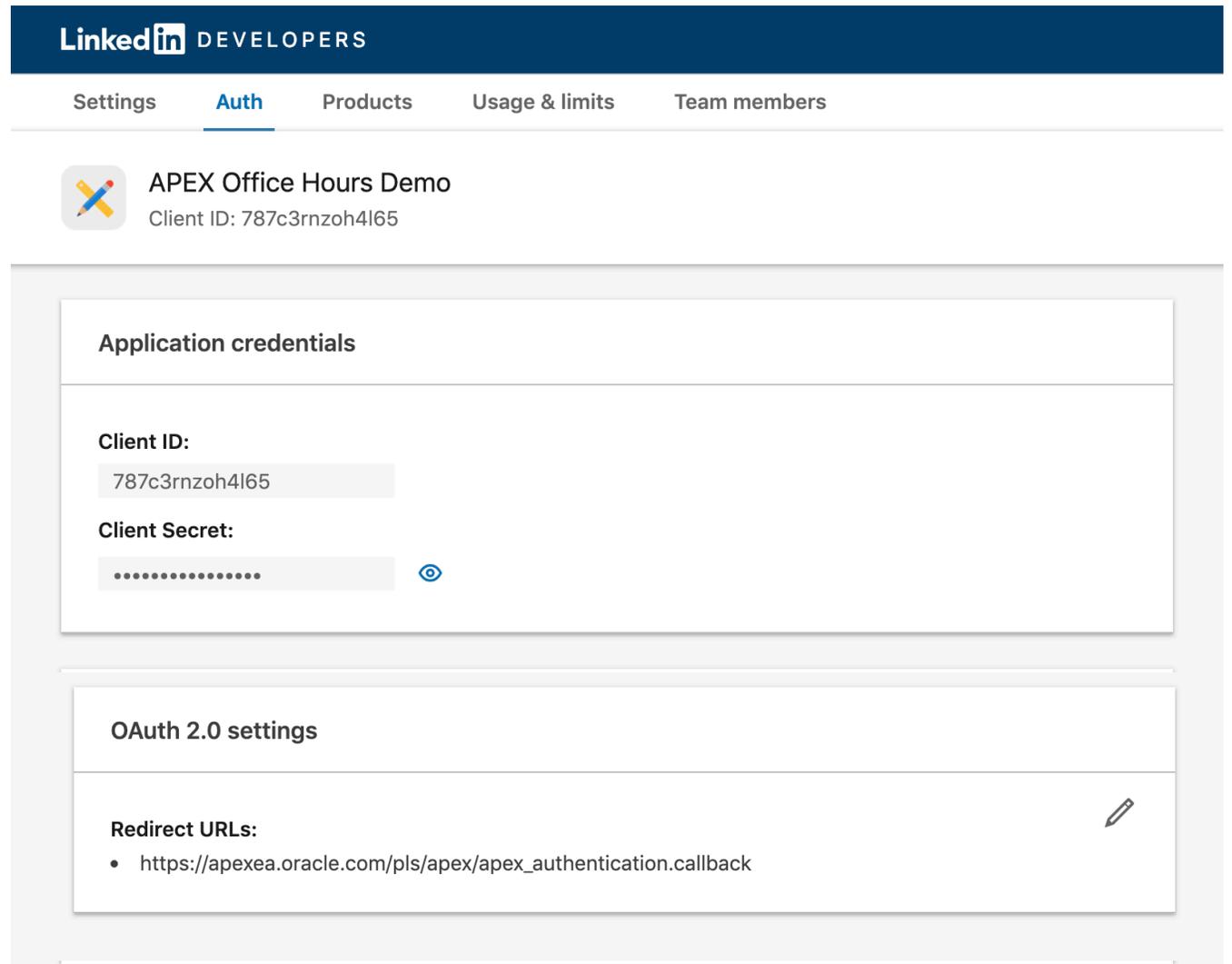
The screenshot shows the LinkedIn Developers interface for an application named 'APEX Office Hours Demo'. The top navigation bar includes 'Settings', 'Auth', 'Products', 'Usage & limits', and 'Team members'. The application details are as follows:

- App settings** (with a 'Delete app' button)
- Application name:** APEX Office Hours Demo (with an edit icon)
- Client ID:** 787c3rnzoh4l65
- Created:** 2/28/2019
- Company:** Oracle
- Description:** LinkedIn Authentication Demo
- App logo:** This logo is displayed to users when they grant access to their data for your app (with a placeholder image of two crossed pencils)
- Privacy policy URL:**

# Using LinkedIn Authentication

## Auth

- Copy Client ID and Client Secret to be used for Web Credentials in APEX
- Enter Redirect URL at bottom of "Auth" settings: [https://apexea.oracle.com/pls/apex/apex\\_authentication.callback](https://apexea.oracle.com/pls/apex/apex_authentication.callback)



The screenshot displays the LinkedIn Developers console for an application named "APEX Office Hours Demo". The interface includes a navigation bar with "Settings", "Auth", "Products", "Usage & limits", and "Team members". The "Auth" section is active, showing the application's Client ID as 787c3rnzoh4l65 and a masked Client Secret. Below this, the "OAuth 2.0 settings" section shows a single Redirect URL: https://apexea.oracle.com/pls/apex/apex\_authentication.callback.

# Using LinkedIn Authentication

## Create Web Credentials in APEX

- LinkedIn Client ID and Client Secret are stored in APEX Web Credentials
- Web Credentials are managed on the Workspace level

The screenshot shows the Oracle APEX interface for creating or editing web credentials. The breadcrumb trail is: Workspace Utilities > Web Credentials > Create/Edit. The page title is "Web Credentials". There are four buttons: "Cancel", "Delete", "Apply Changes", and "Clear Tokens".

The "Attributes" section contains the following fields:

- Name:** LinkedIn (Demo) (required, indicated by an asterisk)
- Static Identifier:** LinkedIn
- Authentication Type:** OAuth2 Client Credentials Flow (dropdown menu)
- OAuth Scope:** r\_fullprofile
- Client ID or Username:** 787c3rnzoh4l65
- Client Secret or Password:** (empty text field)
- Verify Client Secret or Password:** (empty text field)
- Prompt On Install:** Yes
- Comments:** https://www.linkedin.com/developer/apps

# Using LinkedIn Authentication

## Facebook Authentication Scheme

- In your APEX app, create “Social Sign-In” Auth. scheme
- Select your previously created Web Credentials
- Select “Facebook” as Authentication Provider
- Define Scope, Username Attribute and Additional User Attributes

The screenshot shows the 'Authentication Scheme' configuration page in Oracle APEX. The page is titled 'Authentication Scheme' and has a 'Cancel' button in the top right corner. Below the title is a navigation bar with tabs: 'Show All', 'Name', 'Subscription', 'Settings', 'Source', 'Session Not Valid', and 'Login Processing'. The 'Name' tab is selected, and the 'Name' field contains 'LinkedIn'. The 'Scheme Type' is set to 'Social Sign-In'. Below this is the 'Subscription' section, which includes a 'Reference Master Authentication Scheme From' dropdown and a 'Refresh' checkbox. A message states: 'This is the "master" copy of this authentication scheme. There are no subscribers to this authentication scheme.' The 'Settings' section contains several fields: 'Credential Store' (LinkedIn (Demo)), 'Authentication Provider' (Generic OAuth2 Provider), 'Authorization Endpoint URL' (https://www.linkedin.com/oauth/v2/authorization), 'Token Endpoint URL' (https://www.linkedin.com/oauth/v2/accessToken), 'User Info Endpoint URL' (https://api.linkedin.com/v1/people/~:(#USER\_ATTRIBUTES)?format=json), 'Scope' (r\_basicprofile,r\_emailaddress), 'Authentication URI Parameters' (empty), 'Username Attribute' (emailAddress), 'Convert Username To Upper Case' (No), and 'Additional User Attributes' (id,formattedName,location).

# Demo

## LinkedIn Authentication

- 1 Authentication in Oracle APEX
- 2 OAuth 2.0
- 3 Using Google Authentication
- 4 Using Facebook Authentication
- 5 Using LinkedIn Authentication
- 6 **Authorization in Oracle APEX**

# APEX Authorization

## Providing Security Through Authorization

- Extend the security of your application by creating an authorization scheme.
- While conditions control the rendering and processing of specific page controls or components, **authorization schemes control user access to specific controls or components.**
- You can specify an authorization scheme for an entire application, page, or specific control such as a region, item, or button.
- For example, you could use an authorization scheme to selectively determine which tabs, regions, or navigation bars a user sees.
- When you define an authorization scheme, you give it a unique name. Once defined, you can attach it to any component or control in your application.

# Access Control

## Controlling Access to Applications, Pages, and Page Components

- Adding the **Access Control feature** to an application, creates multiple pages and the following components:
  - Adds an Access Control region to the Administration page you specify.
  - Creates the access roles: *Administrator*, *Contributor*, and *Reader*.
  - Creates the authorization schemes: *Administration Rights*, *Contribution Rights*, and *Reader Rights*.
  - Creates the build option: *Feature: Access Control*.
  - Creates the Application Setting: *ACCESS\_CONTROL\_SCOPE*.

# Access Control

## Controlling Access to Applications, Pages, and Page Components

- Configure Access Control by running the app and accessing the Access Control region on the Administration page.
- Click **Users** to add new users, change a user's role, or disable access control by locking an account.
- Click **Access Control** to specify the behavior when authenticated users access the application.

The screenshot displays the 'Administration' page. At the top, the title 'Administration' is visible. Below it, the 'Access Control' section is highlighted with a blue border. This section includes a table with the following data:

Access Control		Add
All authenticated users can access this application		
Administrator		1
Contributor		0
Reader		0

Below the table, there are two navigation options:

- Users** (with a user icon): Change access control settings and disable access control
- Access Control** (with a key icon): Set level of access for authenticated users of this application

# Demo

## Authorization

# Q&A

# Integrated Cloud

## Applications & Platform Services