ORACLE

Extending Oracle E-Business Suite Release 12.2 Using Oracle APEX

Oracle APEX and E-Business Suite Technical paper

April, 2025, Version 5.2

ORACLE

Purpose Statement

This document provides an overview of extending Oracle E-Business Suite Release 12.2 using Oracle APEX.

Disclaimer

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 remain at the sole discretion of Oracle.

ORACLE

Table of Contents

Executive Overview

This paper outlines how to extend Oracle E-Business Suite (EBS) functionality utilizing Oracle APEX. This paper is a collaboration between both product teams to ensure that any extensions developed following the strategy outlined in this paper are fully supported. Recommended architecture and security considerations are discussed in detail.

Introduction

Oracle E-Business Suite (EBS) delivers a wide range of functionality to handle core areas of your business processing needs. However, there are situations where you want to extend your information systems beyond the range of Oracle E-Business Suite. Many times, these necessary extensions are meant to handle unique industry conventions, specific customer requirements, or perhaps to offer some other competitive edge. Sometimes these change requests are simple enough, but other times more extensive customizations are needed. In these scenarios, Oracle APEX provides an easy way to create supplemental applications that are easily integrated with your Oracle E-Business Suite and its data.

Oracle APEX is a low-code application development platform for the Oracle database. Oracle APEX combines the qualities of a personal database (productivity, ease of use, and flexibility) with the qualities of an enterprise database (security, integrity, performance, scalability, availability, and built for the web). The browser-based interface, declarative programming framework, and simple wizards make Oracle APEX easy to learn and enable you to quickly build robust applications.

Oracle APEX offers you a quick and highly productive way to extend your Oracle E-Business Suite environment with almost no impact to your existing implementation. By moving custom behaviors out of Oracle E-Business Suite and into Oracle APEX, you can make Oracle E-Business Suite patching and upgrading much simpler.

You should be able to easily combine your Oracle APEX applications with your Oracle E-Business Suite installation using the architecture and techniques described in this white paper.

What's New

This paper is dedicated to extending Oracle E-Business Suite Release 12.2.x and more sophisticated components are used to show case the use of Oracle APEX Faceted Search, Cards Region, and Maps Region with EBS.

- Version 5.1 of this technical paper leverages the new features from the Oracle E-Business Suite APEX extension that simplify the integration between the two products.
- Version 5.2 of this technical paper addresses the updated FWK bundle and updates the timeout URLs and parameters.

Prerequisites

The prerequisites for the solution found in this paper must be applied in the order listed as follows:

1. Oracle E-Business Suite Release 12.2.7 or later
2. Full-use Oracle Database[1] Release 19c or later
3. Oracle APEX 24.1[2] or later
    - Oracle recommends installing the latest available version of Oracle APEX.
    - Screenshots in this paper utilize APEX 24.1, which may vary for other releases.
4. Oracle REST Data Services (ORDS) 24.3 or later
    - Oracle recommends installing the latest available version of ORDS.
5. Oracle WebLogic Server (WLS)[3] 12c or later
    - You can also utilize Apache Tomcat or ORDS Standalone (Jetty) server, instead of Oracle WebLogic Server.
6. Oracle E-Business Suite instances prior to Release 12.2.14 require the following patches:
    - R12.AD.C.Delta.16
    - R12.TXK.C.Delta.16
    - R12.ATG_PF.C.Delta.13
7. Oracle Application Framework (FWK) Release 12.2.14 Bundle 1 Patch 37423078:R12.FWK.C
8. Oracle E-Business Suite One-Off Patch 37751883 to register APEX_PUBLIC_USER with the application.

> ✓ **Note: The minimum version of products in the listing of prerequisites are those supported at the time of writing. Please review the relevant Lifetime Support Policy documents for each product to ensure you install supported software and know when support ends.**

---

[1] The limited-use Oracle Database license agreement included with an Oracle E-Business Suite license does not allow for the creation of an additional schema as outlined in this paper below. Refer to the following license agreement: https://www.oracle.com/contracts/docs/application_licensing_table_070571.pdf. Therefore, to be fully compliant, full-use Oracle Database Enterprise Edition and Oracle Internet Application Server Enterprise Edition licenses are required.

[2] Read the Application Express (APEX) Database and Web Server Certification Reference in My Oracle Support.

[3] A separate Oracle WebLogic Server license is required. You cannot utilize the Oracle WebLogic Server installed by Oracle E-Business Suite Release 12.2, as this release does not support Java 11 or 17 required by ORDS. It is not supported to update the release of Oracle WebLogic Server utilized by Oracle E-Business Suite.

Concept Overview

Oracle APEX offers you the ability to create reports, transaction screens, and dashboards, based on live Oracle E-Business Suite (EBS) data. To allow you to create Oracle APEX applications that reference EBS data, the desired data is exposed through database views in the APPS schema. READ access to those views is then granted to an existing or new Oracle E-Business Suite custom schema (for this document, XX_APEX). Because direct INSERT, UPDATE, and DELETE operations on Oracle E-Business Suite data is unsupported, applications that manipulate Oracle E-Business Suite data will use the public, documented and supported, Oracle E-Business Suite APIs. References to these APIs should be coded in custom PL/SQL packages in the APPS schema which you will use in the Oracle APEX applications.

The extensions built in Oracle APEX will appear seamless to your end users because you can use the same authentication (who can login) and authorization (who can see what) within your Oracle APEX applications that are used within your Oracle E-Business Suite installation. Your new applications can either be stand-alone or fully integrated. Stand-alone applications are accessed directly but use the same login credentials used for Oracle E-Business Suite access. Fully integrated applications are registered within the Oracle E-Business Suite, so they are available from within the Oracle E-Business Suite menus. Oracle APEX contains a pre-configured authentication scheme for HTTP Header Variable for use with Oracle Access Manager (OAM).

If your Oracle E-Business Suite instance uses OAM for authentication, setting up authentication will be very simple. If you are using custom authentication for Oracle E-Business Suite, you can still achieve seamless integration, but you will need to code a function within the APPS schema that validates the user, and then add an authentication scheme to your Oracle APEX applications that calls that new function.

This document discusses the Oracle APEX architecture and provides detailed instructions for creating all the components necessary to extend Oracle E-Business Suite. In support of the instructions, scripts and sample packaged applications are available for download to ensure that any developer can quickly replicate the examples.

Oracle APEX Architecture

Oracle APEX resides completely within the Oracle database in its own schema and can be installed on any version of Oracle Database from version 19c and later. Runtime, development, and deployment do not require client software as access is 100% browser-based using a web listener communicating with the Oracle database. The application definitions are stored as metadata within the Oracle APEX schema which is accessed to perform page rendering and processing.

The middle tier, also known as the application tier, requires Oracle REST Data Services (ORDS). ORDS is a JAVA EE based solution that is certified with Oracle WebLogic Server, ORDS Standalone (Jetty server), and Apache Tomcat.

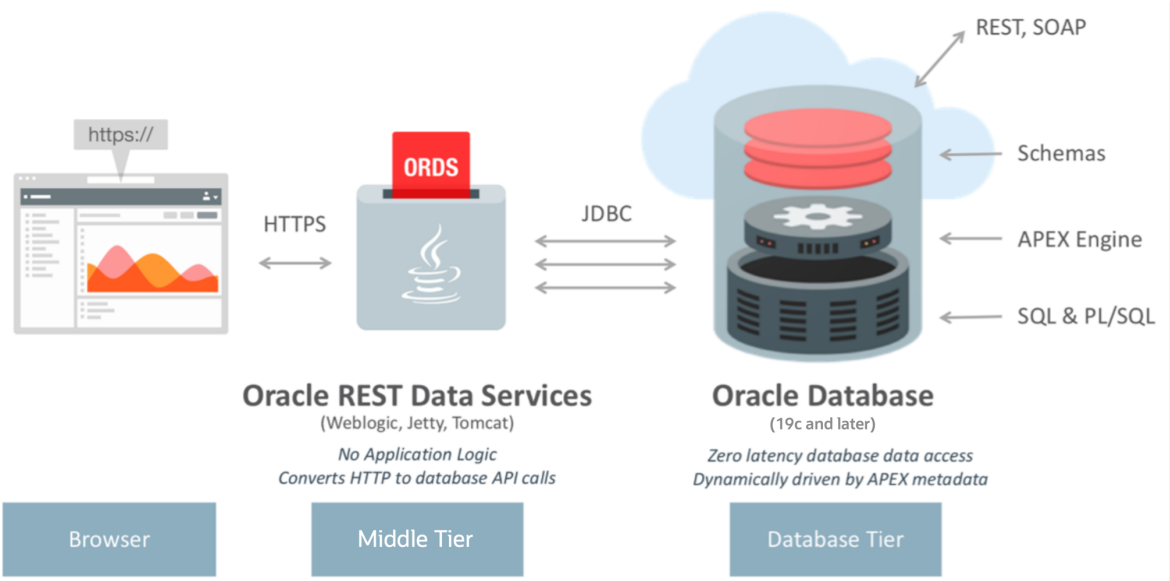

Figure 1. Oracle APEX architecture

> ✓ **Note: You cannot utilize the Oracle WebLogic Server installed by Oracle E-Business Suite Release 12.2 as this release does not support Java 11 or 17 required by ORDS. It is not supported to update the release of Oracle WebLogic Server utilized by Oracle E-Business Suite.**

Recommended Middle Tier Deployment Architecture

The recommended configuration for the middle tier (also known as the application tier) requires two separate application servers—one for the Oracle REST Data Services to be installed within Oracle WebLogic Server on a separate application server from the one configured for Oracle E-Business Suite (see Figure 2 and Figure 3). Installing separate application servers allows for each to be managed independently.

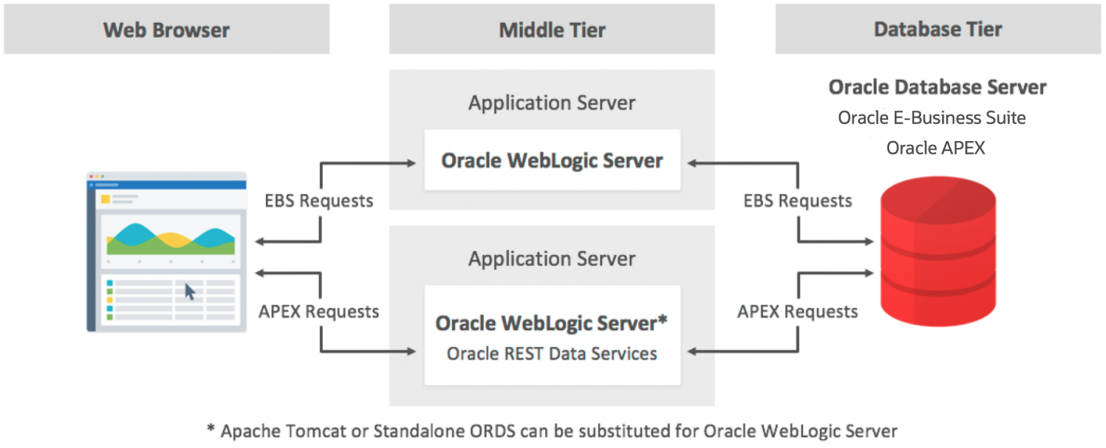

Figure 2. Recommended middle tier configuration – EBS Release 12.2

> ✓ **Note: You can also install Oracle REST Data Services into ORDS Standalone (Jetty server) or Apache Tomcat instead of Oracle WebLogic Server and still be fully supported and in compliance. It is not certified to install Oracle REST Data Services into the Oracle WebLogic Servers utilized by Oracle E-Business Suite Release 12.2.**

The Oracle E-Business Suite authorization scheme for APEX used in this paper requires the same host name and domain for both Oracle REST Data Services and Oracle E-Business Suite, as shown in the following examples:

- Oracle E-Business Suite URL: `https://<EBS_hostname>:<port>/OA_HTML/AppsLogin`
- Oracle REST Data Services URL: `https://<EBS_hostname>:<port>/ords`

We strongly recommend configuring TLS/SSL on your environment.

Recommended Database Architecture

Oracle E-Business Suite embedded mode requires configuring Oracle APEX within the Oracle Database server where Oracle E-Business Suite is installed (see Figure 3). A separate, minimally privileged, Oracle E-Business Suite custom schema (XX_APEX in this document) must be defined as the basis for building Oracle APEX applications.

Defining an Oracle APEX workspace directly against the owning Oracle E-Business Suite schema (APPS) is not supported, as this is a breach of security protocol. This recommendation is because the Oracle E-Business Suite APPS schema has full access to all the Oracle E-Business Suite database objects. If you were to associate the APPS schema with an Oracle APEX workspace, then the Oracle APEX applications in that workspace would also have full access to all the underlying Oracle E-Business Suite database objects. These applications could then perform insert, update, and delete statements directly on the APPS database objects and potentially invalidate the APPS schema, as critical business rules, enforced by the public APIs, are bypassed. Therefore, not only is this a breach of security protocol, but it is also potentially catastrophic to your Oracle E-Business Suite environment and is not supported.

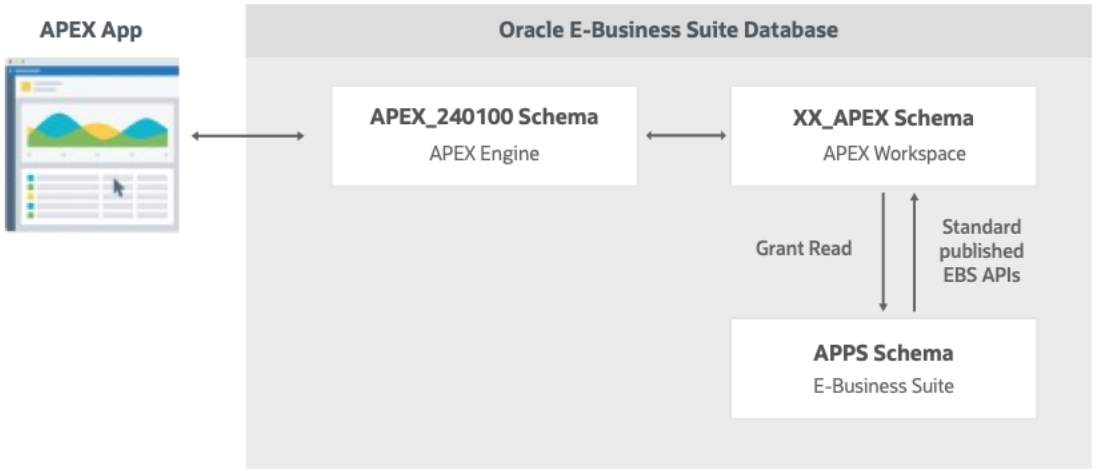

Figure 3. Recommended database configuration

> ✓ **Note: APEX_240100 represents the schema name for APEX 24.1. If you have later version of Oracle APEX, the schema name will change accordingly.**

Installing Oracle APEX

For a fresh install where APEX extensions have not been built within your Oracle E-Business Suite environment, download the latest version of Oracle APEX (currently APEX 24.1) from the Oracle Technology Network at http://apex.oracle.com/download and follow instructions found in Installing and Configuring Oracle APEX and Oracle REST Data Services in the *Oracle APEX Installation Guide*.

You must also download the latest version of Oracle REST Data Services (currently 24.3) from Oracle Technology Network at http://www.oracle.com/technetwork/developer-tools/rest-data-services/overview/index.html and follow the installation instructions for the Oracle WebLogic Server.

After completing the installation steps for both Oracle APEX (including enabling Network Services in Oracle Database) and Oracle REST Data Services, you must create a workspace for your development and then create applications within that workspace. For your production instance, it is highly recommended to run Oracle APEX in runtime-only mode to further harden security. For more information, see Converting Between Runtime and Full Development Environments in the *Oracle APEX Installation Guide*.

Oracle APEX Schemas and Workspaces

Workspaces are logical containers within Oracle APEX that provide functional security for Oracle APEX development. Workspaces can be associated with one or more custom database schemas. Each custom schema associated with an Oracle APEX workspace can be used to parse SQL and PL/SQL requests.

Oracle E-Business Suite uses a schema called APPS, which has full access to the complete Oracle E-Business Suite data model. As stated earlier, associating an Oracle APEX workspace directly with the APPS schema is not supported. Instead, create at least one separate schema in the Oracle E-Business Suite database for developing your Oracle APEX applications. In this document, a custom schema XX_APEX is created. You can also use your current custom schema as your parsing schema when developing APEX applications.

As per Oracle E-Business Suite development standards, any custom object created in the database should be prefixed with XX, YY, or ZZ. XX is the most popular prefix used and is the prefix used in this paper; hence, the custom schema is called XX_APEX.

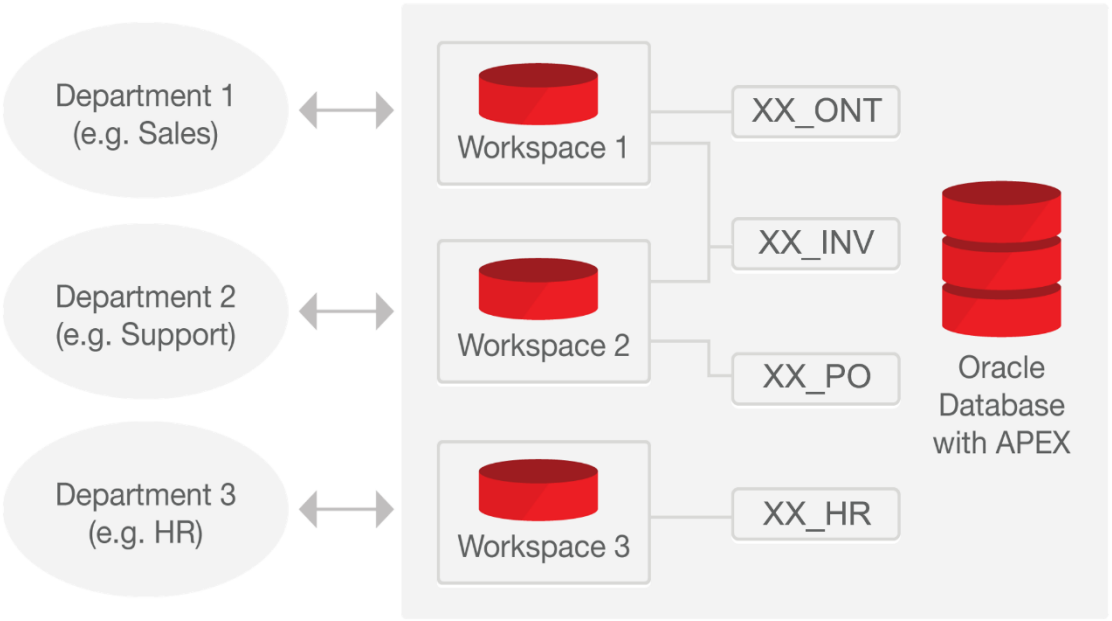

Figure 4. Example of Oracle APEX workspaces

Defining the Oracle APEX Workspace and Schema

Based on the recommended database architecture, you should create a workspace and a custom schema on the database server where Oracle E-Business Suite is installed. If you already have a custom schema defined, you can also associate that schema with an Oracle APEX workspace. For this paper you will use EBS_APEX for the workspace and XX_APEX for the custom schema.

Creating the Custom Schema XX_APEX

The following instructions show you how to create the least privileged custom schema XX_APEX:

1. Connect to your Oracle E-Business Suite database server.
2. Start SQL*Plus and connect as SYS specifying the SYSDBA role.
3. Create the XX_APEX schema:

```
create user XX_APEX identified by <<xx_apex>>
default tablespace SYSAUX;
```

> ✓ **Note: XX_APEX does not have privileges to create objects in the database. If you plan to add your own custom database objects or import any sample APEX applications to the workspace associated with this schema, you will need to grant the privileges necessary to create the supporting objects required by that application.**

Registering XX_APEX with Oracle E-Business Suite

Register the XX_APEX schema with Oracle E-Business Suite by performing the following steps:

1. Source the run edition file system environment file by running the appropriate command for your platform.

   UNIX:

   ```
   $ . <RUN_BASE>/EBSapps/appl/APPS<CONTEXT_NAME>.env
   ```

   Windows:

   ```
   C:\> <RUN_BASE>\EBSapps\appl\envshell<CONTEXT_NAME>.cmd
   ```

2. Run the `ADAPEXSCHREG.sql` script passing the APEX schema you want to register. Use the appropriate command for your platform.

   UNIX:

   ```
   $ sqlplus apps/<APPS password> @$AD_TOP/patch/115/sql/ADAPEXSCHREG.sql XX_APEX
   ```

   Windows:

   ```
   C:\> sqlplus apps/<APPS password> @%AD_TOP%\patch/115/sql\ADAPEXSCHREG.sql XX_APEX
   ```

3. Enter the following values when prompted: EBS_SYSTEM schema password, APPS schema username, and APPS schema password.

Creating the EBS_APEX Workspace

Create your APEX workspace EBS_APEX by performing the following steps:

1. Log in to Oracle APEX Administration (`https://<EBS_hostname>:<port>/ords/apex_admin`).
2. In the **Username** field, enter `ADMIN` and enter the password defined during installation. Click **Log in**.
3. Navigate to **Manage Workspaces** and click **Create Workspace**.
4. Enter the **Workspace name** EBS_APEX and click **Next**.
5. In the **Re-use Existing Schema** field, select Yes.
6. In the **Schema Name** field, enter `XX_APEX`. Click **Next**.
7. Enter the following values and then click **Next**:
   a. **Administrator User name**: `ADMIN`
   b. **Administrator Password**: `<APEX_Password>`
   c. **E-mail**
8. Click **Create Workspace** and then click **Done**.

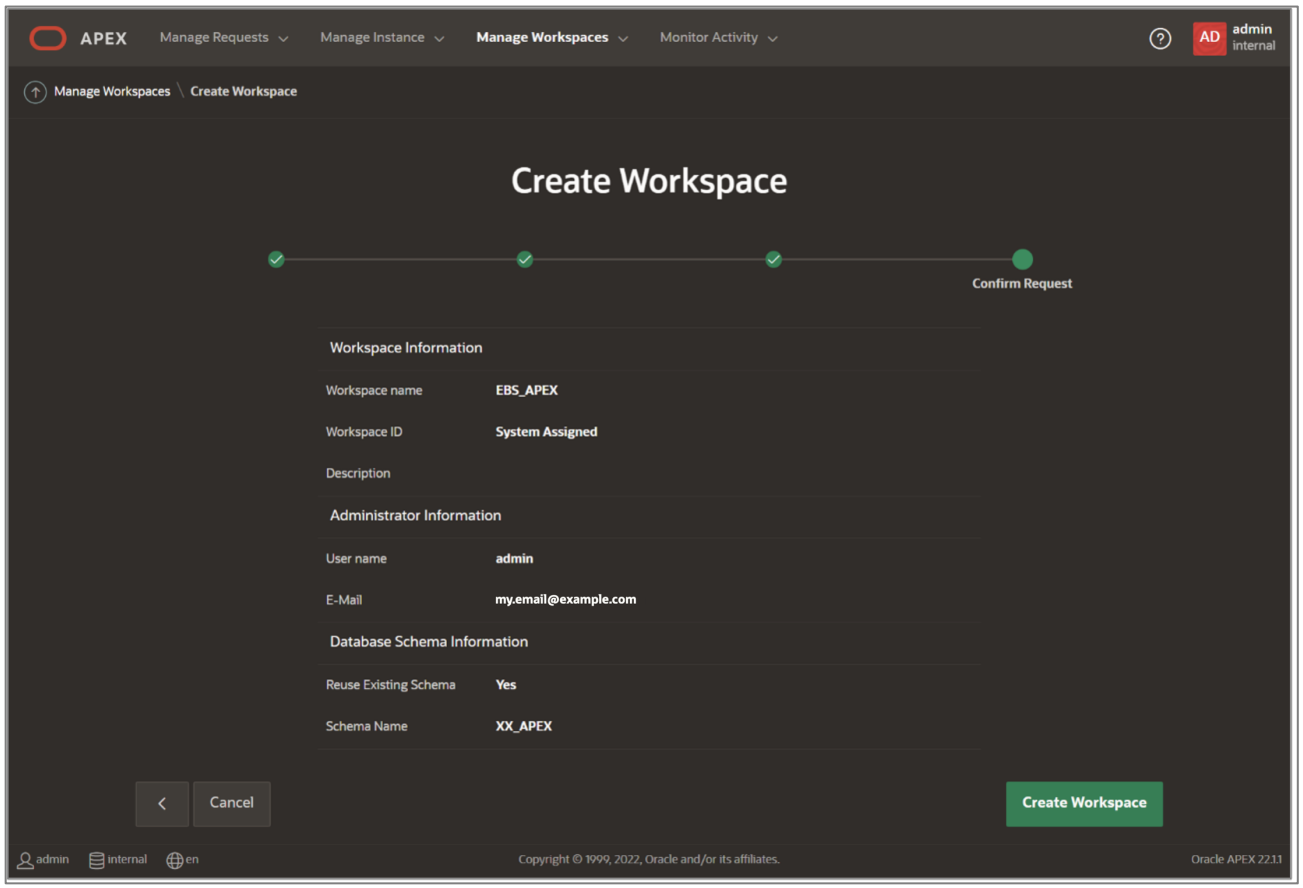

Figure 5. Oracle APEX workspace provisioning

Configuring Developers Within APEX

Now that the workspace has been created, development will be performed within the Application Builder. To log in to the Application Builder and create developer accounts, use the following steps:

1. Log in to Oracle APEX Application Builder (`https://<EBS_hostname>:<port>/ords/apex`).
2. In the **Workspace** field enter `EBS_APEX`, enter the following values and then click **Sign In**:
   a. **Administrator Username**: `ADMIN`
   b. **Administrator Password:** Enter the APEX password entered when creating the workspace.
3. Once logged in, you must reset the password. Enter the following values:
   a. **New Password**: `<Your_New_APEX_Password>`
   b. **Confirm New Password**: `<Your_New_APEX_Password>`

   Click **Change Password**.
4. Navigate to **Administration**, then select **Manage Users and Groups**.
5. Depending on your requirements, you can create one or more developers using the **Create User** or **Create Multiple Users** buttons (top right).

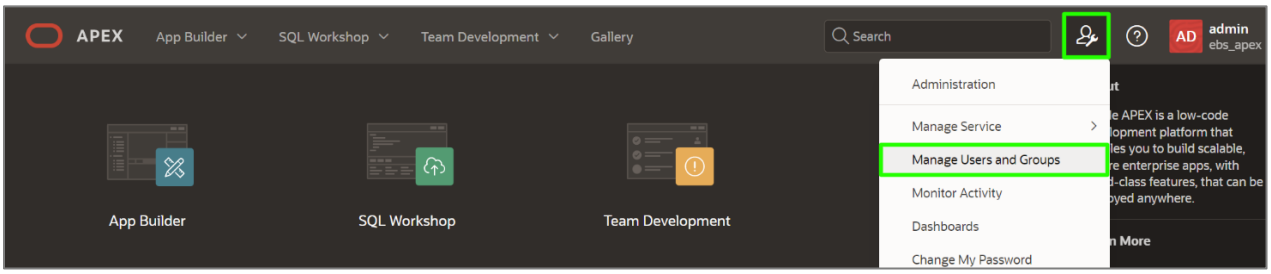

Figure 6. Navigation to Administration, Manage Users and Groups

Accessing Oracle E-Business Suite Data

Oracle APEX provides numerous wizards to rapidly build application components on existing Oracle tables or views. It is advisable to define additional database views for the Oracle Applications objects you wish to access. For security purposes, it is recommended that Oracle E-Business Suite data is accessed only through these views. The views can incorporate security, joins, etc., and prevent Oracle APEX applications from making any unauthorized changes to the underlying data.

The following is sample code for creating such a view in the APPS schema:

```
CREATE OR REPLACE VIEW xx_apex_ebs_user
    (  user_id
     , user_name
     , start_date
     , end_date
     , description
     , email_address
    , user_guid
    , person_party_id
    , CONSTRAINT APEX_EBS_USER_pk
          PRIMARY KEY (user_id)
          RELY DISABLE NOVALIDATE
    )
AS
    SELECT user_id
    ,      user_name
    ,      start_date
    ,      end_date
    ,      description
    ,      email_address
    ,      user_guid          /* Used for Single-Sign On */
    ,      person_party_id    /* FK to party information */
    FROM fnd_user;
```

This functionality is incorporated into the `apex_ebs_setup.sql` script referenced later in this paper.

It is then necessary to grant rights to the new custom schema (XX_APEX) from the APPS schema:

```
GRANT READ ON xx_apex_ebs_user TO XX_APEX;

GRANT READ on fnd_responsibility_vl TO XX_APEX;
```

Updating Oracle E-Business Suite Tables

Although it may be tempting to directly insert, update, and delete records on Oracle E-Business Suite tables, this is unsupported and must be avoided. Keep in mind that direct updates to Oracle E-Business Suite tables are not supported unless explicitly documented. Direct updates will bypass validation, security, and business logic which could lead to data corruption or unexpected system behavior. Oracle recommends that all Oracle E-Business Suite updates are made through calls to public APIs. The Oracle E-Business Suite Integration Repository provides a searchable list of public APIs for the system.

The following is sample code for creating a package body within the APPS schema:

```
CREATE OR REPLACE PACKAGE BODY xx_apex_sample_apis AS

PROCEDURE apex_update_email (  p_username        in varchar2
                             , p_owner           in varchar2
                             , p_email_address   in varchar2
                            )
IS
BEGIN
    wf_event.setdispatchmode('async');
    fnd_user_pkg.updateuser
    (  x_user_name     => p_username
     , x_owner         => p_owner
     , x_email_address => p_email_address);
END apex_update_email;

END xx_apex_sample_apis;
/
```

This functionality is incorporated into the `apex_ebs_setup.sql` script referenced later in this paper.

When utilizing the Oracle APEX wizards on the new views to define applications, DML automatic row processes will be defined referencing the view specified. These processes can be deleted and alternate processes written which call the appropriate APIs.

Downloading and Installing the Sample Code

The code samples in the previous section are available for download from GitHub. To run these scripts, perform the following steps:

1. Download the setup script file from GitHub – https://www.oracle.com/a/otn/docs/apex_ebs_setup.sql.
2. Save the file as `apex_ebs_setup.sql` and upload it to your Oracle E-Business Suite database server.
3. Connect to your Oracle E-Business Suite database server.
4. Start SQL*Plus and connect as the `APPS` user.
5. Run the script, providing the appropriate criteria:
   ```
   @apex_ebs_setup.sql
   ```
6. Enter the APEX Schema Name when prompted.

Integrating with Oracle E-Business Suite

To connect from Oracle E-Business Suite to your new Oracle APEX applications, a secure mechanism is required to hand control from Oracle E-Business Suite to Oracle APEX. The steps to do so are described in the following sections:

1. **Developing the Sample APEX Application**
    a. Creating an Oracle APEX Application
    b. Updating the Generated Processes and Page Items
    c. Defining Oracle APEX Authentication and Authorization
    d. Applying the Authentication and Authorization Schemes to the Pages of Your Application
    e. Enabling Deep Linking and Setting the Timeout URL
    f. Enabling Embedded Mode on Your APEX Application
2. **Registering Your Custom Sample APEX Application with Oracle E-Business Suite**
    a. Defining Oracle E-Business Suite Profile and Form Functions
    b. Defining Oracle E-Business Suite Menus and Responsibilities

Developing the Sample APEX Application

Developers define applications within Oracle APEX as a collection of pages. Pages generally have one or more regions, buttons, items, computations, processes, branches, dynamic actions, tabs, lists, and breadcrumbs. Each application within an Oracle APEX instance has a unique application ID. For users to access an Oracle APEX application directly they enter a URL of the form:
`https://<EBS_hostname>:<port>/ords/f?p=<Application_Id>:<Home_Page>.`

To proceed with the specific steps to create a custom Oracle APEX, continue with the following steps:

a. Creating an Oracle APEX Application
b. Updating the Generated Processes and Page Items
c. Defining Oracle APEX Authentication and Authorization
d. Applying the Authentication and Authorization Schemes to the Pages of Your Application
e. Enabling Deep Linking and Setting the Timeout URL
f. Enabling Embedded Mode on Your APEX Application

As an alternative to these steps, if you want to deploy a working application in which all steps from this document have been implement, you can import and use the steps in Using the Sample Packaged Application to Automate Development of the Sample APEX Application.

Creating an Oracle APEX Application

To create an Oracle APEX application, perform the following steps:

1. Navigate to the Oracle APEX Application Builder.
2. Click **Create**, then **click Use Create App Wizard**.

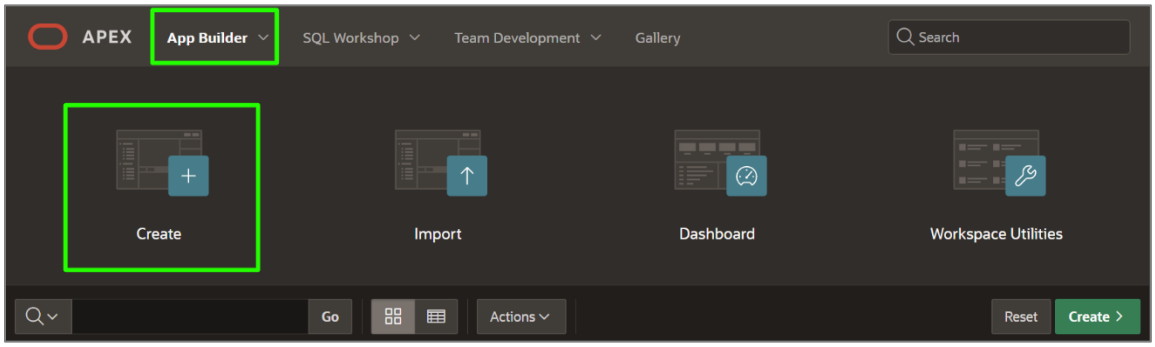

Figure 7. Creating an Oracle APEX application

> ✓ **Note: All screenshots and steps provided used Oracle APEX 24.1. If you are using other APEX versions, the screenshots and steps may vary.**

3. In the Create an Application wizard page, for **Name** enter `APEX EBS Demo`.
4. Click **Appearance**.

5. Select the **Theme Style** "Redwood Light" and then click **Save Changes**.

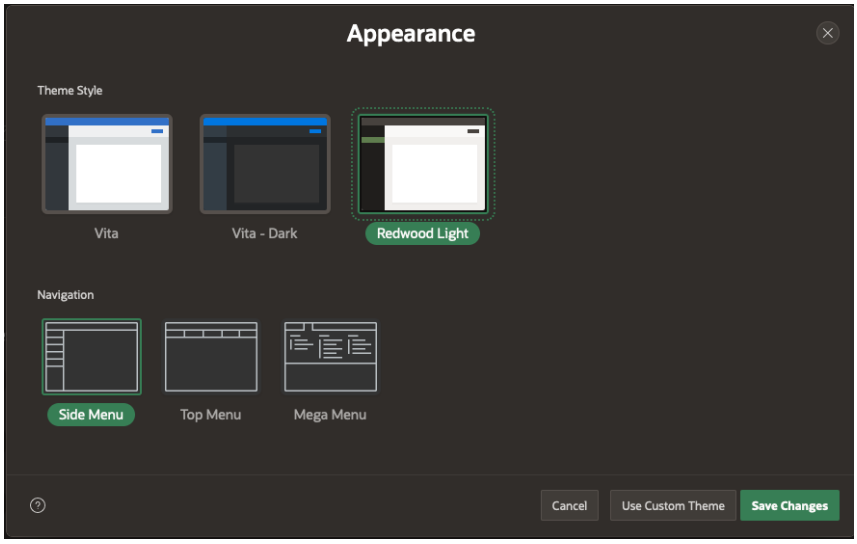

Figure 8. Defining application appearance in the Create Application wizard

6. Click **Create Application**.

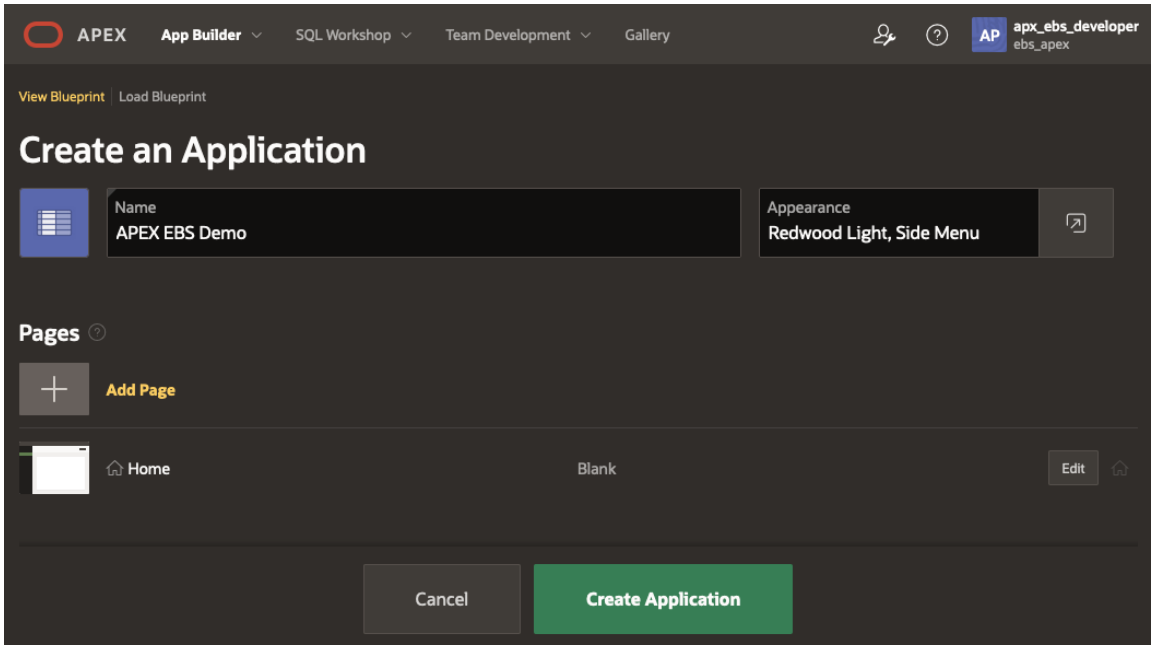

Figure 9. Creating an application in Create Application wizard

7. On the following screen **Click Create Page**, then select **Form**.

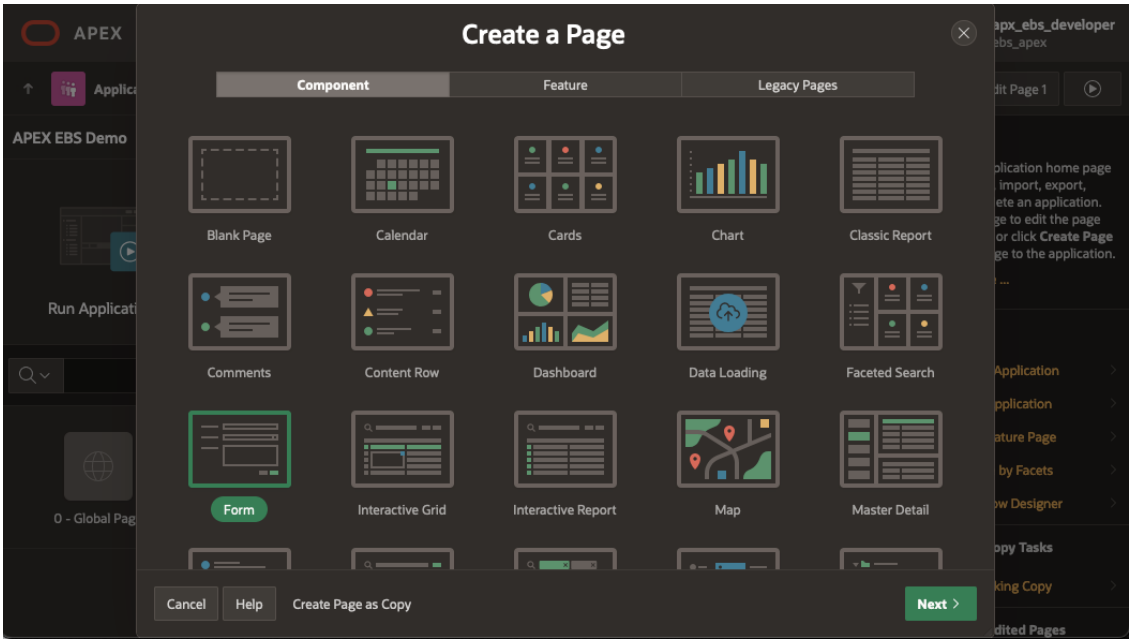

Figure 10. Creating a Form page in Application Details

8. In the Create From dialog, for Page Number enter `2`.
9. For **Name**, enter `Update Email - Without Responsibility`.
10. For **Table/View Owner**, select APPS from the drop-down list.

11. For **Table/View Name**, select XX_APEX_EBS_USER.
12. Disable by switching **Use Navigation** to off and click **Next**.

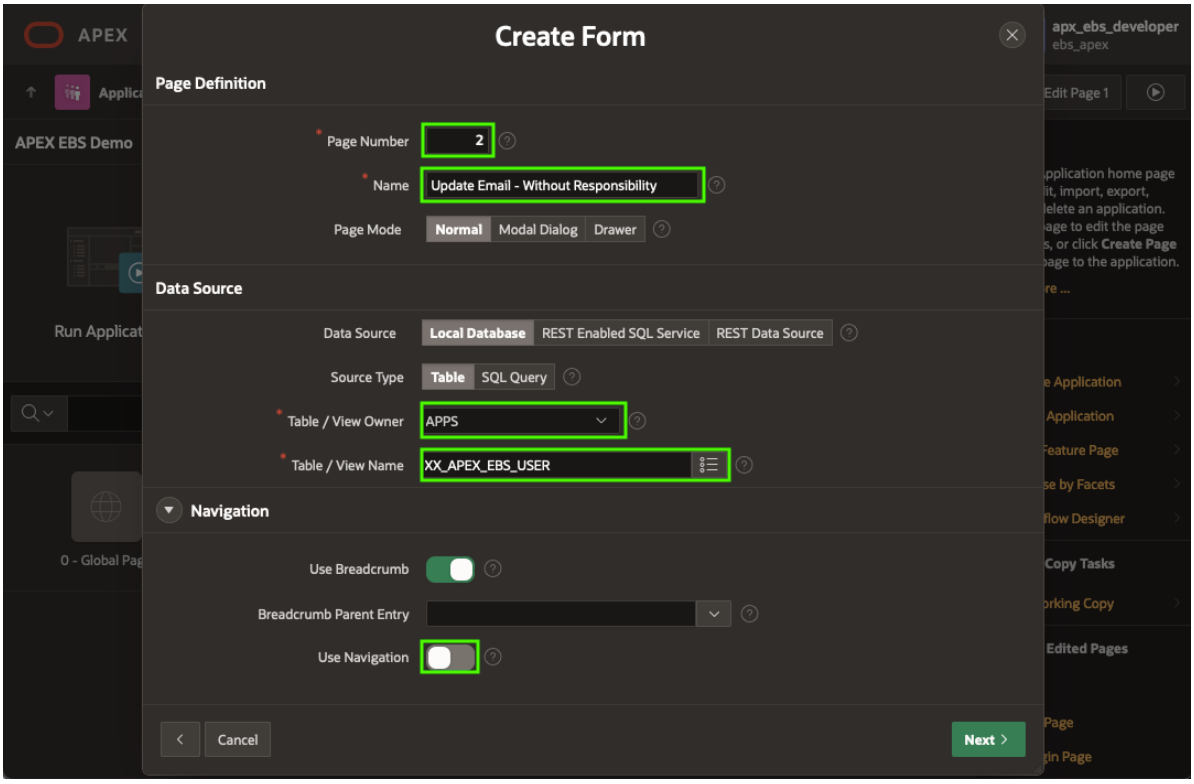

Figure 11. Creating a Form page in Application Details – Page Definition, Data Source, Navigation

13. Choose USER_NAME (Varchar2) from the drop-down list for **Primary Key Column 1**. Click **Create Page**.

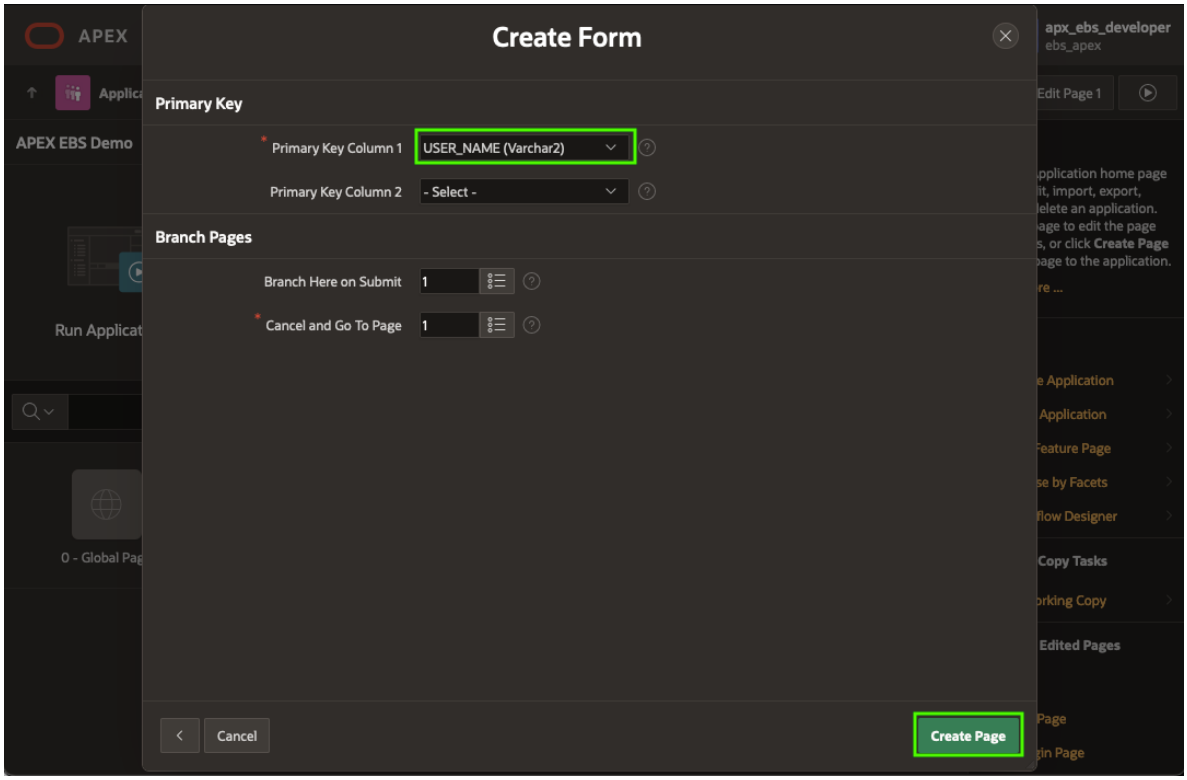

Figure 12. Creating a Form page in Application Details – Primary Key Column 1

14. In the right side property panel under **Appearance**, set **Page Template** to “Minimal (No Navigation) using the drop-down list. Then close the Page Details Definition and go back to Application Page by clicking **Save** and then select the Application breadcrumb link.

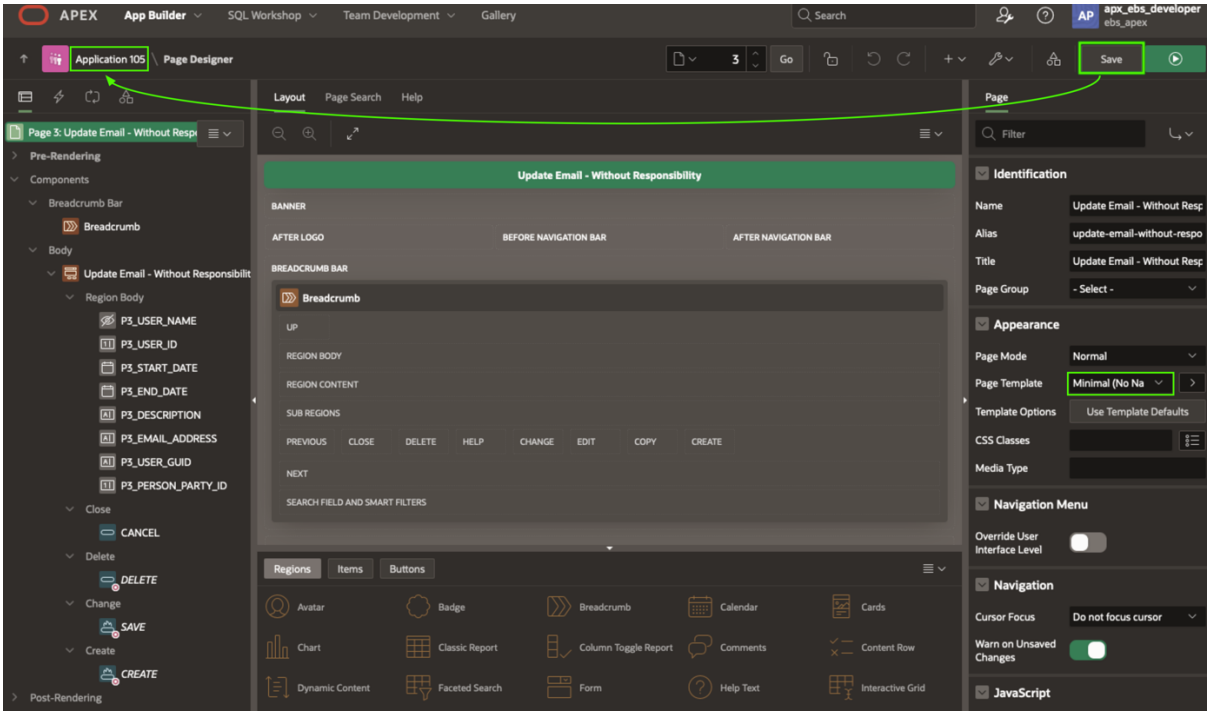

Figure 13. Save and navigate to Application home

15. Add another page by repeating steps 7 to 13. In step 9, name the new page `Update Email – Using Responsibility`.

You have now created an application with two form pages based on the XX_APEX_EBS_USER view. Next, you will need to update these pages to work correctly with the view and for the second page to correctly utilize the Oracle E-Business Suite (EBS) responsibility.

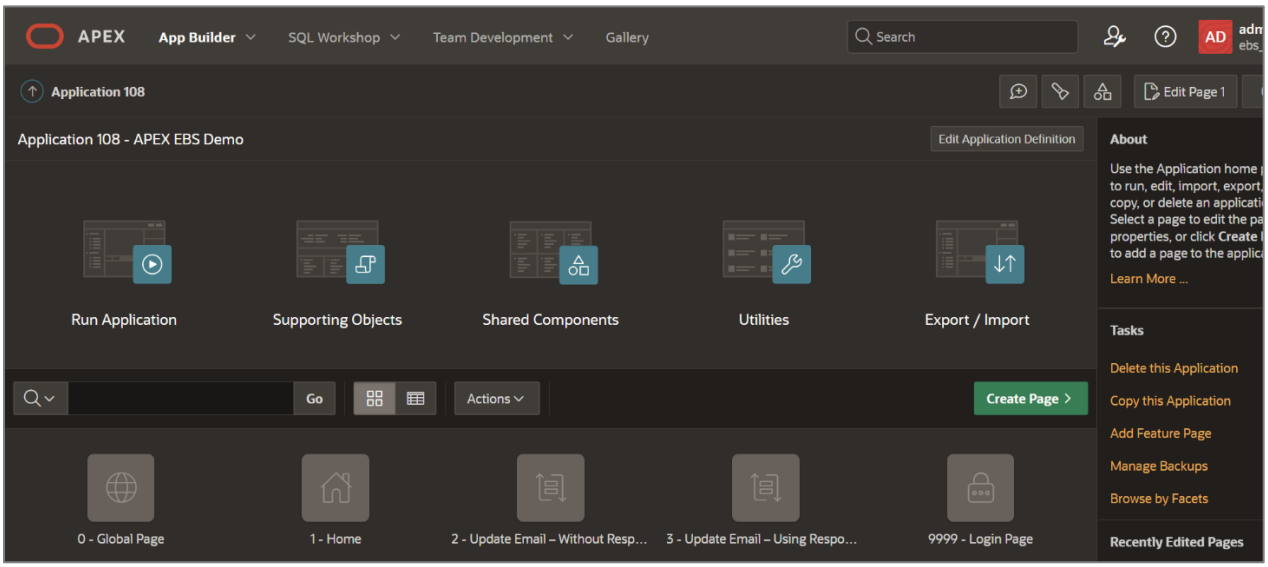

Figure 14. New application displaying pages generated

Updating the Generated Processes and Page Items

Updating the Generated Process to Use the Oracle APEX User Name

By default, the Create Page Wizard created a Fetch Row from XX_APEX_EBS_USER process to retrieve the data from the associated table or view. However, you need to update the generated process to use the Oracle APEX user name. Use the following steps to do so:

1. From the Application home page, click **2 – Update Email – Without Responsibility**, which will bring up Page Designer and display Page 2.
2. In the left pane, within the Rendering tab, expand the **Update Email – Without Responsibility** node.
3. Select **P2_USER_NAME**.
4. In the right pane within the Property Editor, find the Primary Key attribute and make sure it is enabled.
5. Right-click the P2_USER_NAME and select **Create Computation**.
6. In the right pane within the Property Editor, set the Point attribute to “Before Header”.
7. Set the Type to “Item” and set the Item Name to “APP_USER”.
8. At the top of the page, click **Save**.

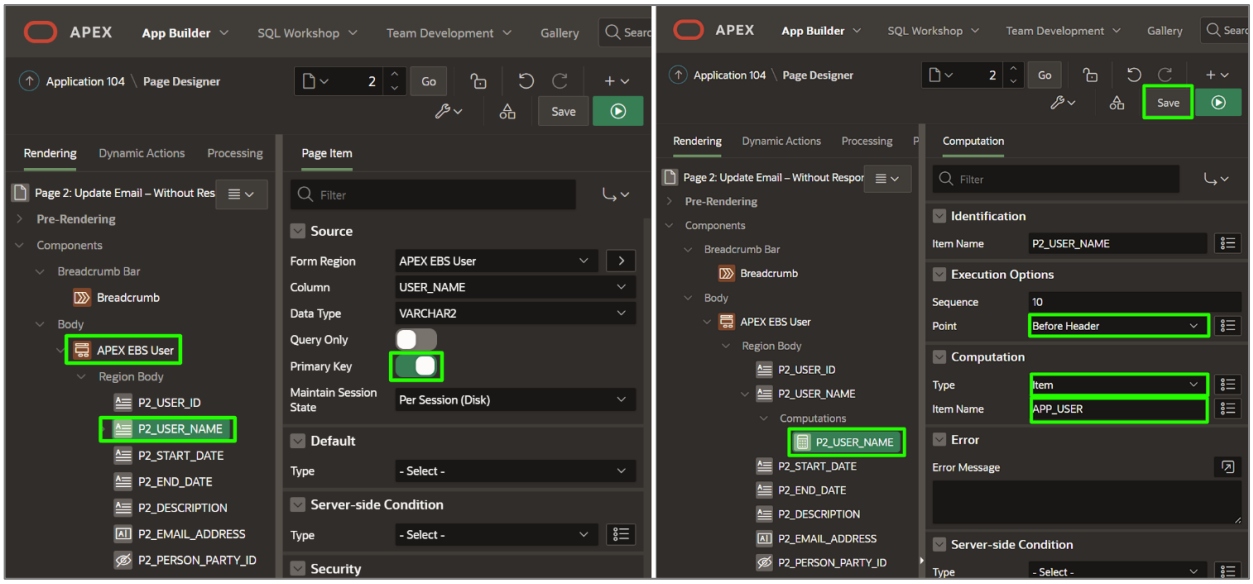

Figure 15. Updating the pre-rendering process

Updating Items to Display Only

The items on the page were generated as text items by default. For this example, we only want the user to be able to update the Email Address, so we need to update the page and change item properties:

1. From the Application Builder, within Page Designer, ensure you are on Page 2.
2. In the left pane within the **Rendering** tab, select the **Update Email – Without Responsibility** region.
3. In the right pane within the **Property Editor**, update the **Name** and **Title** attributes to `APEX EBS User`.

4. In the left pane within the **Rendering** tab, under the APEX EBS User region, click P2_USER_NAME.
5. Hold down the Shift key and click **P2_DESCRIPTION**.
6. Hold down the Control (Ctrl) key and click **P2_USER_GUID** then click **P2_PERSON_PARTY_ID**.
7. In the right pane within the Property Editor, find the **Type** attribute within the Identification section.
8. Use the drop-down list to update the **Type** to “Display Only”.
9. Click **Save**.

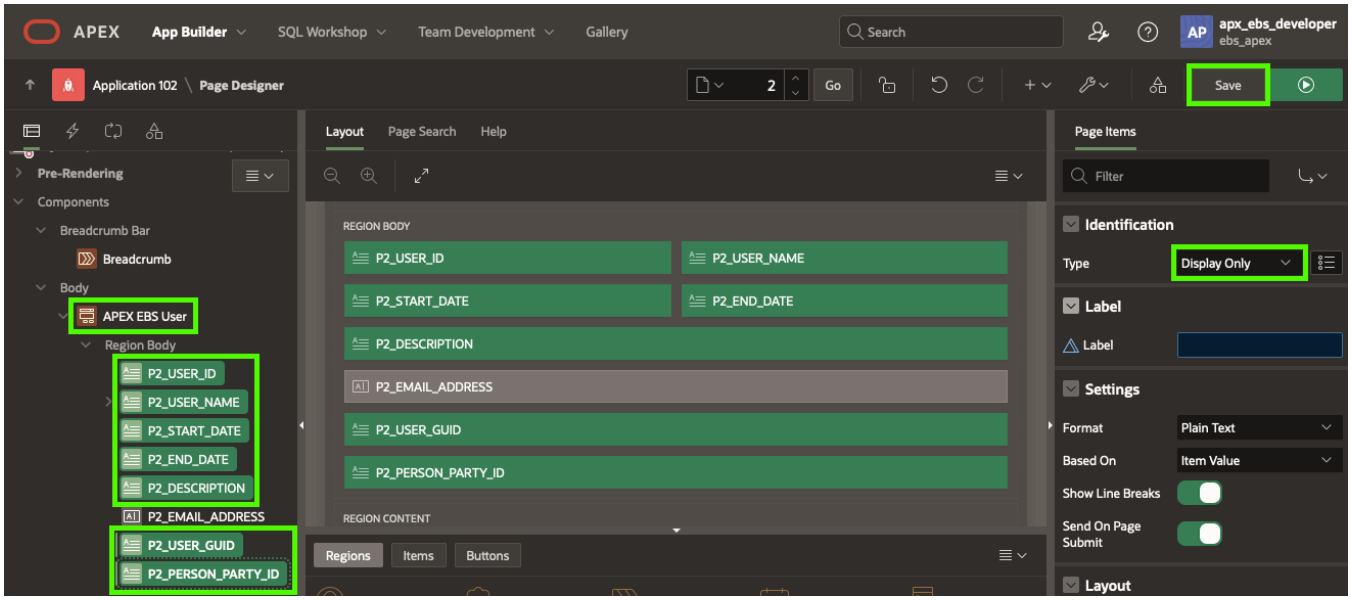

Figure 16. Updating items to Display Only

Deleting the Current Process and Creating the Update Email Process

The wizard also created a process called Process form Update Email – Without Responsibility which is an Automatic Row Processing (DML) process to perform inserts, updates, and deletes on the specified table or view. However, you need to replace the process with a call to the apex_update_email procedure defined earlier:

1. From the Application Builder within Page Designer, ensure you are on Page 2.
2. In the left pane, click the **Processing** tab.
3. Right-click the “Process form Update Email – Without Responsibility” process and select **Delete**.

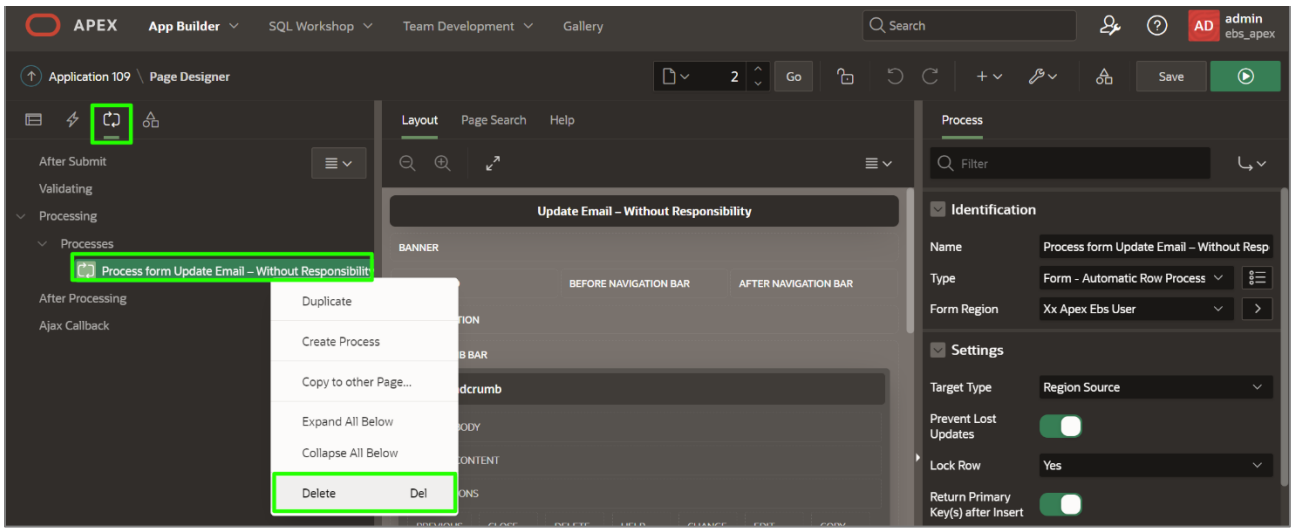

Figure 17. Deleting the generated process

4. Right-click the Processing region and click **Create Process**.
5. In the right pane within the Property Editor, enter the following:
    - Identification > Name – enter Update Email
    - Source > PL/SQL Code - enter:

```
apps.xx_apex_sample_apis.apex_update_email
  (  p_username      => :APP_USER
   , p_owner         => :APP_USER
   , p_email_address => :P2_EMAIL_ADDRESS
  );
```

    - Success Message > Success Message – enter Email updated successfully.
    - Error > Error Message – enter Email not updated.
    - Server-side Condition > When Button Pressed – select SAVE.
6. Under After Processing, Branches, right-click Go To Page 1 node and click **Delete**.
7. In the left pane, within the **Rendering** tab, find the **Delete** button, right-click it and select **Delete**. Do the same thing with the **Create** button.
8. Click **Save**.

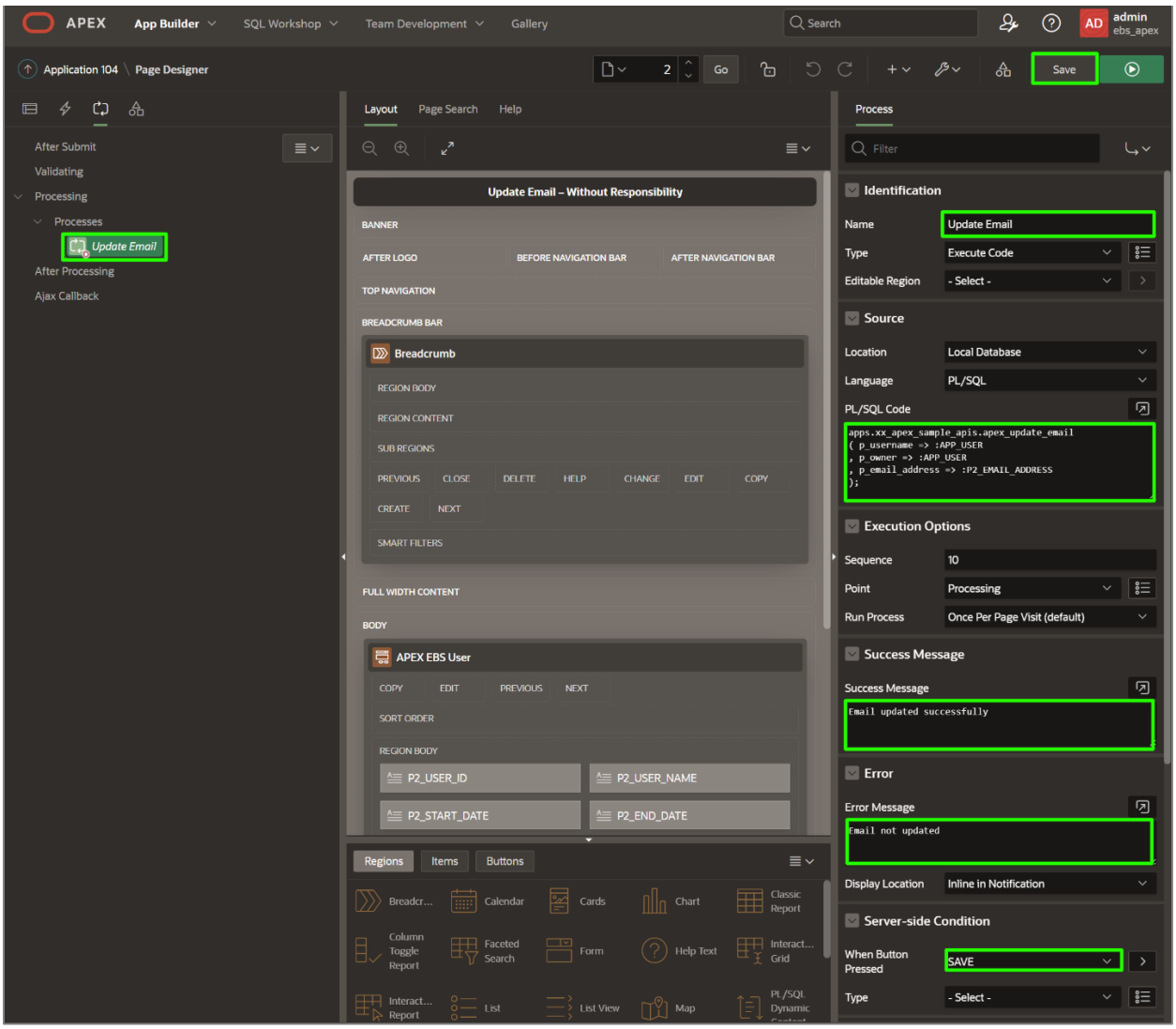

Figure 18. Adding a new process

You must perform the same operations described in Updating the Generated Process to Use the Oracle APEX User Name on Page 3 as you performed on Page 2,

To show the Oracle E-Business Suite responsibility that is going to be passed to Page 3, you must add a new region, two items, and a process.

Creating a Region and Page Items

1. From the Application Builder within Page Designer, ensure you are on Page 3.
2. In the left pane, click the **Rendering** tab.
3. Right-click Body and select **Create Region**.

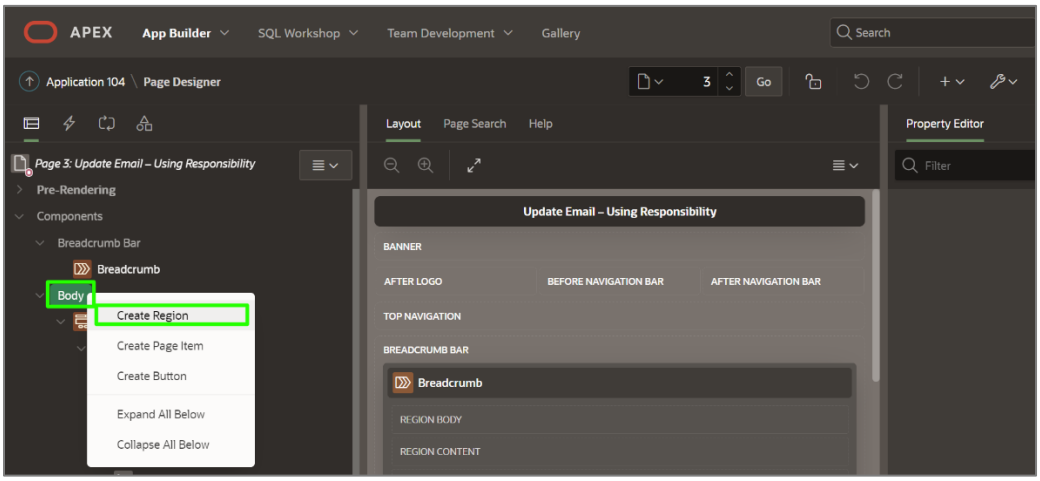

Figure 19. Creating a region on Page 3

4. In the right pane within the Property Editor, enter the following: Identification > Name – enter Responsibility Details.
5. In the left pane within the **Rendering** tab, right-click the Responsibility Details region, and select **Create Page Item**.
6. In the right pane within the Property Editor, enter the following:
    - In the Identification pane, in the **Name** field, enter `P3_RESPONSIBILITY_NAME`.
    - In the Identification pane, from the **Type** drop-down list, select “Display Only”.
    - In the Source pane, from the **Type** drop-down list, select “Null”.

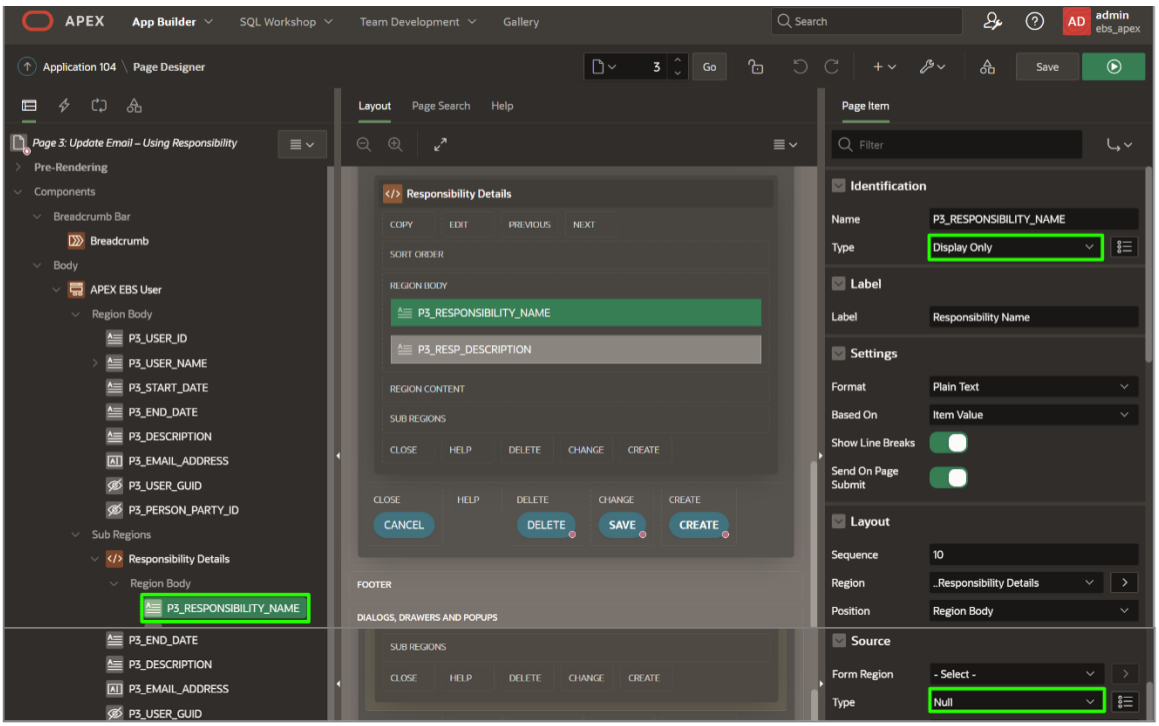

Figure 20. Creating a page item

7. In the left pane within the Rendering tab, right-click the P3_RESPONSIBILITY_NAME item, and select **Duplicate**.
8. In the right pane within the Property Editor, enter the following:
    - In the Identification pane, in the **Name** field, enter `P3_RESP_DESCRIPTION`.
    - In the Label pane, in the **Label** field, enter `Description`.

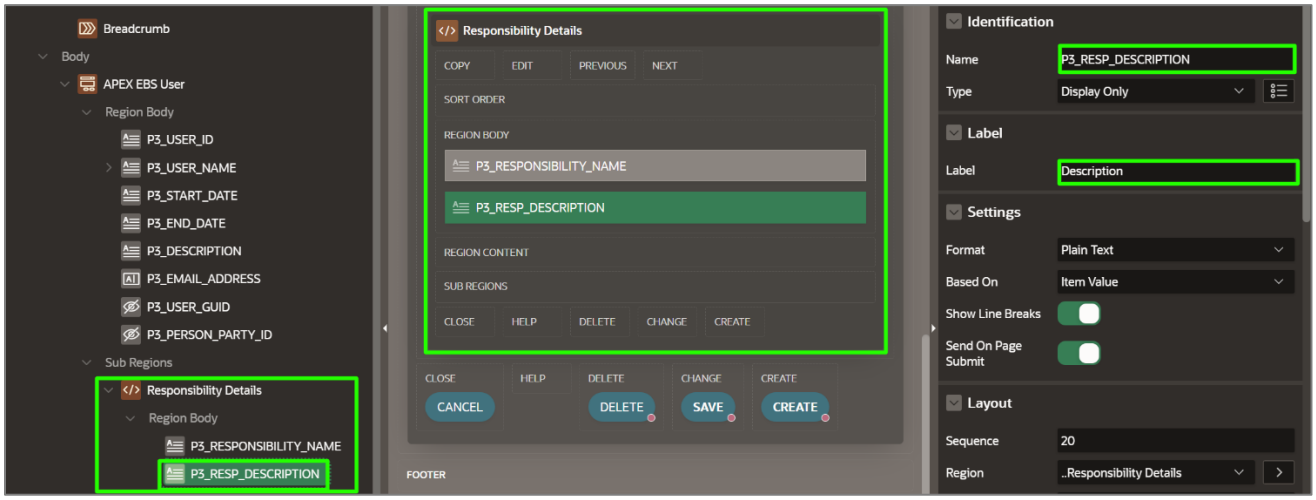

Figure 21. New region and items created on Page 3

Creating a Process to Retrieve Responsibility Details

To create the process required to retrieve the responsibility details use the following:

1. In the left pane within the **Rendering** tab, expand Pre-Rendering.
2. Right-click **Processes** and select **Create Process**.
3. In the right pane within the Property Editor, enter the following:
    - Under Identification, enter `Fetch Responsibility` in the **Name** field.
    - Under Source, enter the following in the **PL/SQL Code** text field:

```
begin
  for c1 in (select responsibility_name
             ,      description
             from apps.fnd_responsibility_vl
             where application_id    = :EBS_APP_ID
             and   responsibility_id = :EBS_RESP_ID
            ) loop
    :P3_RESPONSIBILITY_NAME := c1.responsibility_name;
    :P3_RESP_DESCRIPTION    := c1.description;
  end loop;
exception
  when others then
    null;
end;
```

4. Click **Save**.

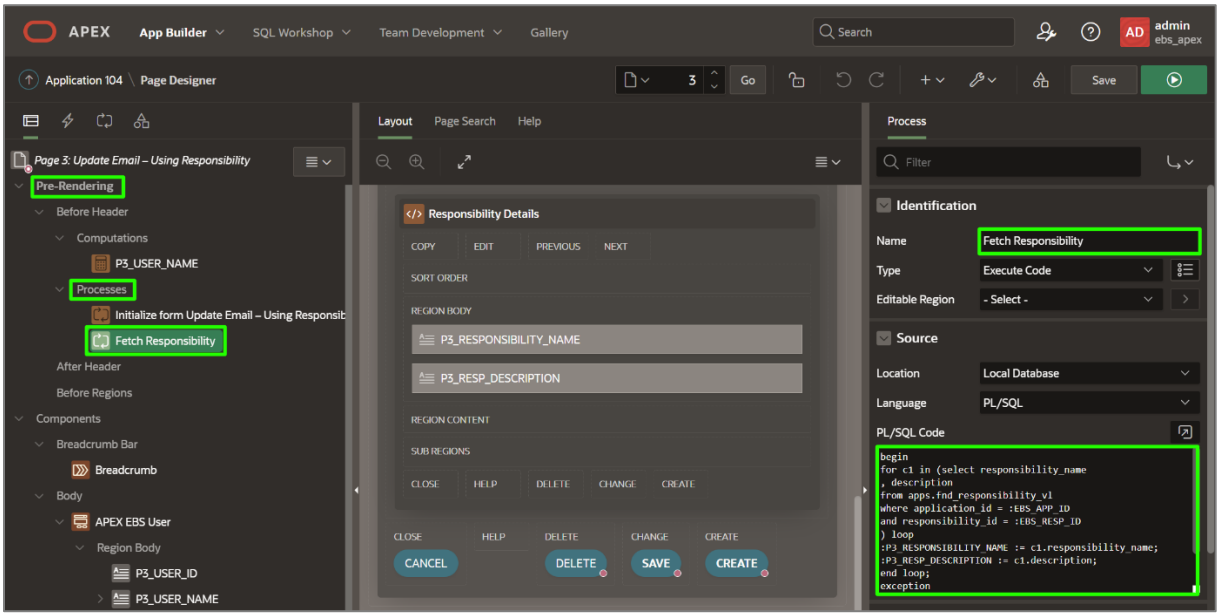

Figure 22. Create process to populate responsibility

Populating Additional Application Items for Page 3

You need to create three Oracle APEX Application Items to hold the three Oracle E-Business Suite parameters—Responsibility Id, Application Id, and Security Group Id—that will be passed by the Oracle E-Business Suite function. Follow these steps to add Application Items into Oracle APEX:

1. Log in to Oracle APEX Application Builder (`https://<EBS_hostname>:<port>/ords/apex`).
2. From the Application Builder for the APEX EBS Demo application, go to **Shared Components**.
3. Click **Application Items**.
4. Click **Create** and enter the following values:
   - **Name**: `EBS_RESP_ID`
   - **Session State Protection**: Unrestricted

   Click **Create Application Item**.
5. Click **Create** and enter the following values:
   - **Name**: `EBS_APP_ID`
   - **Session State Protection**: Unrestricted

   Click **Create Application Item**.
6. Click **Create** and enter the following values:
   - **Name**: `EBS_SEC_GROUP`
   - **Session State Protection**: Unrestricted

   Click **Create Application Item**.

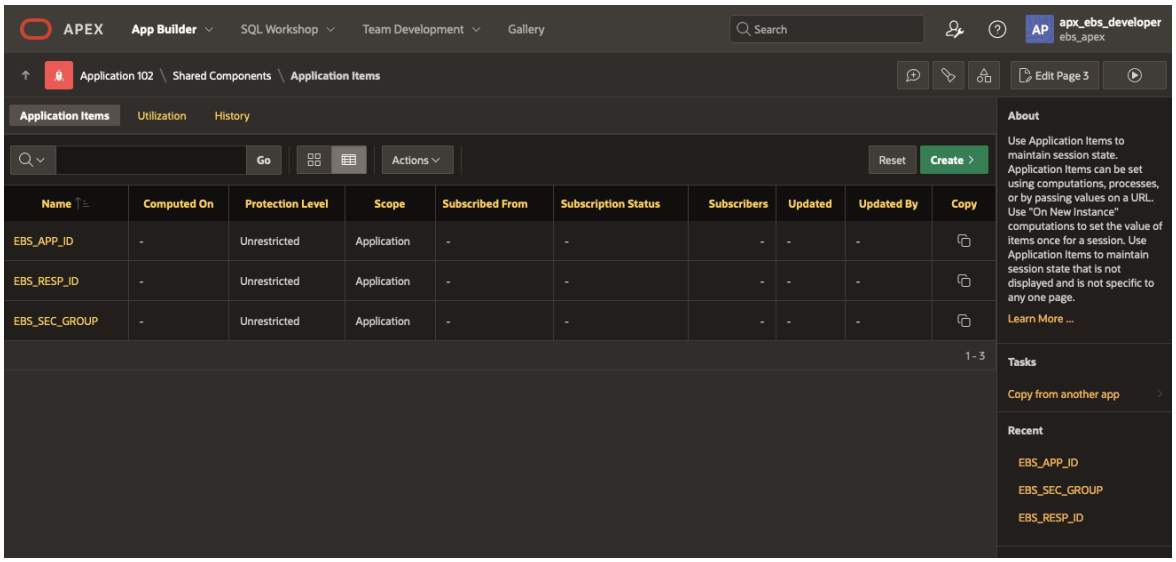

Figure 23. Summary of Application Items under Shared Components

Defining Oracle APEX Authentication and Authorization

Oracle APEX provides “out-of-the-box” mechanisms to handle both authentication (Is the user a valid Oracle E-Business Suite user?) and authorization (What privileges or functions does the user have within the application while using this responsibility?).

Authentication schemes check the user’s user name and password credentials before the user is allowed to access the application. Authorization schemes, on the other hand, control display and user access to pages, regions, items, buttons, and processes within an application.

Defining Oracle APEX Authentications

There are several pre-configured authentication schemes defined within Oracle APEX that can be utilized, such as the HTTP Header variable for use with Oracle Access Manager (OAM) or custom authentication schemes (See: Establishing User Identity Through Authentication in the *Oracle APEX App Builder User's Guide*.

This paper describes an Oracle APEX custom authentication delivered in Oracle E-Business Suite that leverages a pre-authenticated EBS session. If the attempt to access the page cannot locate the pre-authenticated session, it will re-direct the user to the EBS login mechanism.

Defining the APEX Authentication for your Sample Application

Follow these steps to define your sample application custom authentication scheme:

1. Log in to Oracle APEX Application Builder (`http://<EBS_hostname>:<port>/ords/apex`).
2. From the Application Builder, select the APEX EBS Demo application, and navigate to **Shared Components**.
3. Click **Authentication Schemes**, click **Create**, and select **Based on a pre-configured scheme from the gallery**. Then, click **Next**.
4. For **Name**, enter `FND_APEX_AUTHENTICATION` and select “Custom” in **Schema Type** drop-down list.
5. Under **Settings**, enter the following in the **Sentry Function Name** field: `apps.fnd_apex.authentication`.
6. Click **Create Authentication Scheme**.

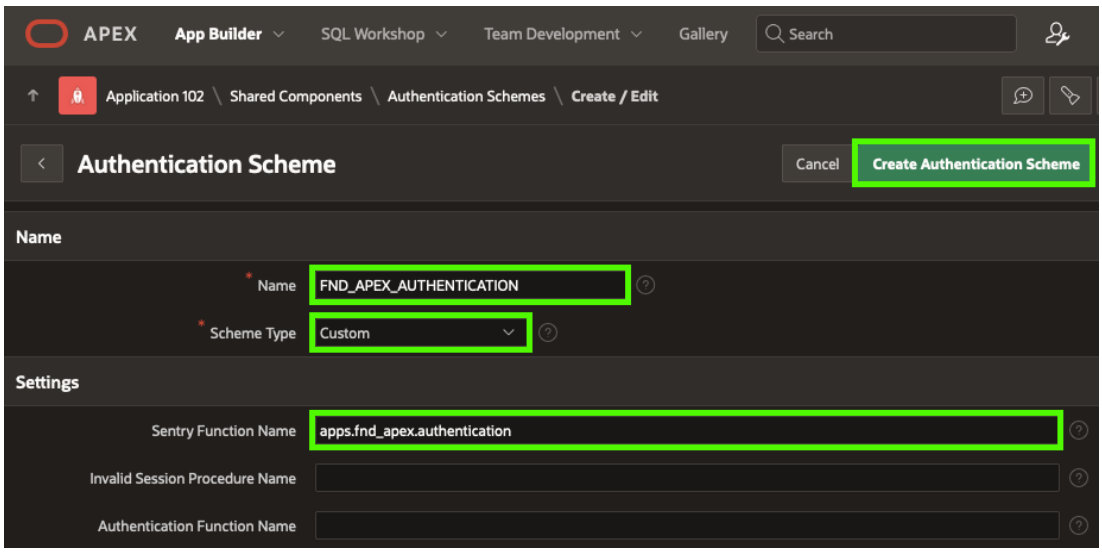

Figure 24. Create Authorization Scheme

> ✓ **Note: This API leverages the authentication performed by EBS and re-builds the EBS Applications context to be used when invoking any PL/SQL artifacts and queries.**

Configuring the Sample Application Authentication

The following instructions set the authentication scheme that will be used by your application.

1. In the Oracle APEX Application Builder, return to the APEX EBS Demo application level by clicking the “Application 102” breadcrumb.
2. Click **Edit Application Definition**.
3. Select the **Security** tab. Under **Authentication**, set the Authentication Scheme.
4. Set the Authentication Scheme to the scheme created in the previous section, Defining the APEX Authentication for your Sample Application.
5. Click **Apply Changes** to set authentication scheme.

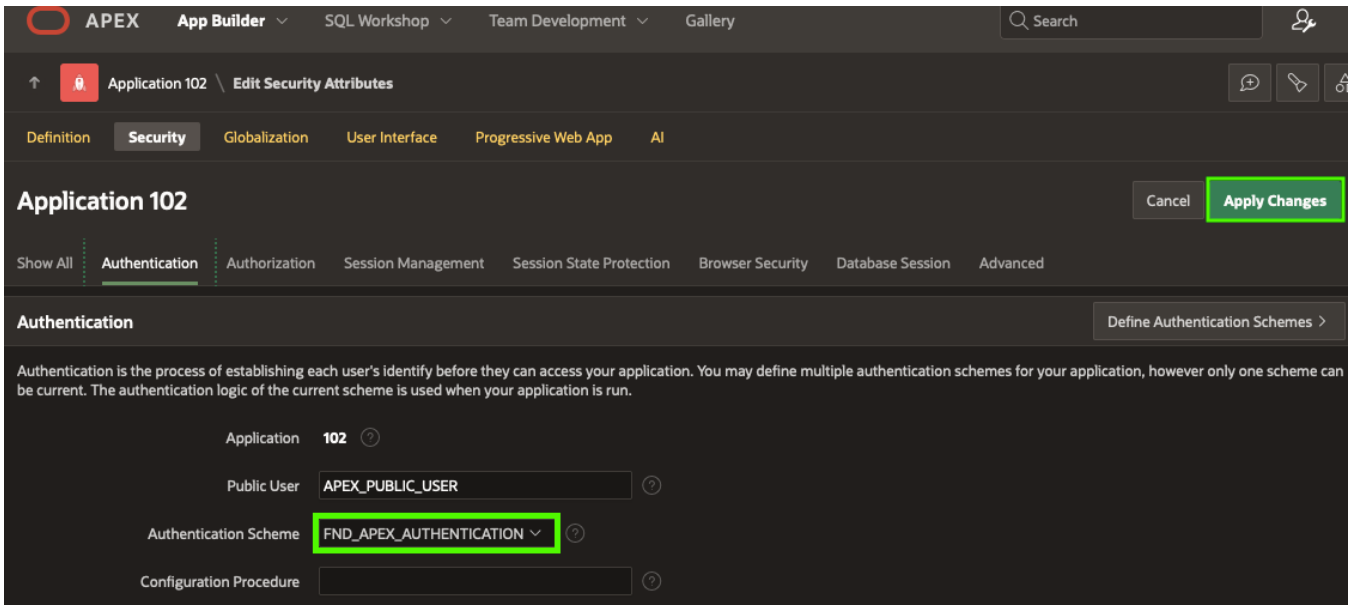

Figure 25. Configuring the sample application authentication scheme.

Defining the Sample Application Login Page

Since Oracle E-Business Suite will perform the authentication, you must set the sample application log in page to Oracle E-Business Suite by using the following steps:

1. Continuing in the Oracle APEX Application Builder, under “Application 102” and click **Edit Application Definition**.
2. Select the **User Interface** tab and click **Attributes**.
3. In the **Login URL** field, enter the following: `http://<EBS_hostname>:<port>/OA_HTML/AppsLogin`
4. Click **Apply Changes**

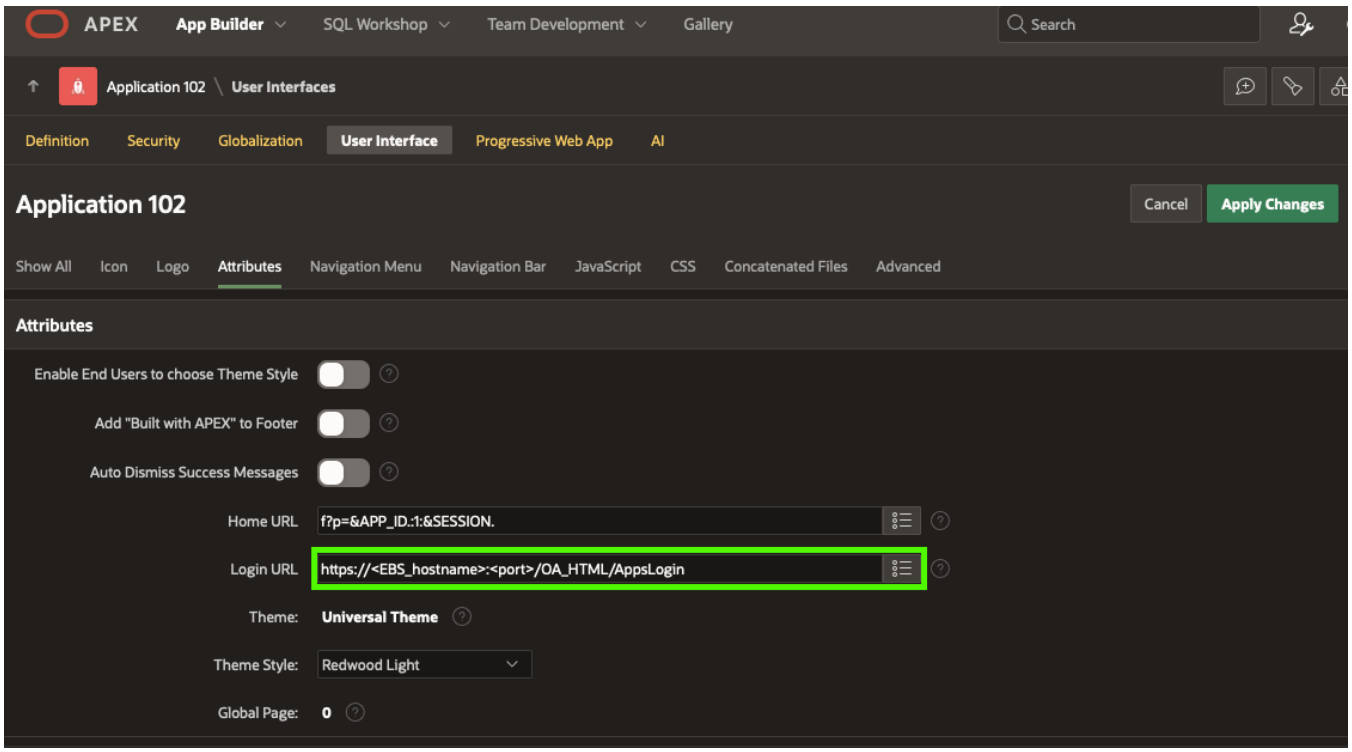

Figure 26. Defining the sample application log in page

Defining Oracle APEX Authorizations

Oracle APEX provides the ability to define authorization schemes that can be used on every component within an application (for example pages, regions, buttons, items, validations, etc.) to restrict access (See: Providing Security Through Authorization in the *Oracle APEX App Builder User's Guide*).

Oracle E-Business Suite includes the concept of responsibilities, which based on the underlying menus, determines what capabilities and data users are allowed to access. The recommended practice is to create Oracle APEX authorization schemes that mimic the necessary Oracle E-Business Suite responsibilities to define cohesive access plans.

It is strongly recommended that authorizations are applied to all non-public Oracle APEX pages (and those limited by Oracle E-Business Suite responsibilities) rather than just to menu items. Oracle recommends defining an authorization scheme at the application level referencing the Oracle E-Business Suite form function which called the APEX application. Session state protection should also be defined for the application to prevent users from accessing pages by manipulating the URL (See: Understanding Developer Security Best Practices in the *Oracle APEX App Builder User's Guide*).

Creating an Authorization Scheme for the Sample Application

Create an authorization scheme by performing the following steps:

1. Log in to Oracle APEX Application Builder (`http://<EBS_hostname>:<port>/ords/apex`).
2. From the Application Builder, select the APEX EBS Demo application, and navigate to **Shared Components**.
3. Select **Authorization Schemes** and click **Create**.

4. Select **Create Authorization Scheme: From Scratch** and click **Next**.
5. Enter the name of the authorization scheme, `FND_APEX_AUTHORIZATION`, in the **Name** field.
6. Select the **Scheme Type** "PL/SQL Function Returning Boolean" from the drop-down list.
7. With in the **PL/SQL Function** body text box, enter the following:

```
begin
return apps.fnd_apex.authorization;
end;
```

8. Provide an error message in case of authentication or authorization failure after the execution of the above PL/SQL. "Access Denied – You are not authorized to access the specified function based on your credentials."
9. Under **Validate authorization scheme**, select the "Once per page view" option.
10. Save the authorization scheme changes by clicking **Create Authorization Scheme**.

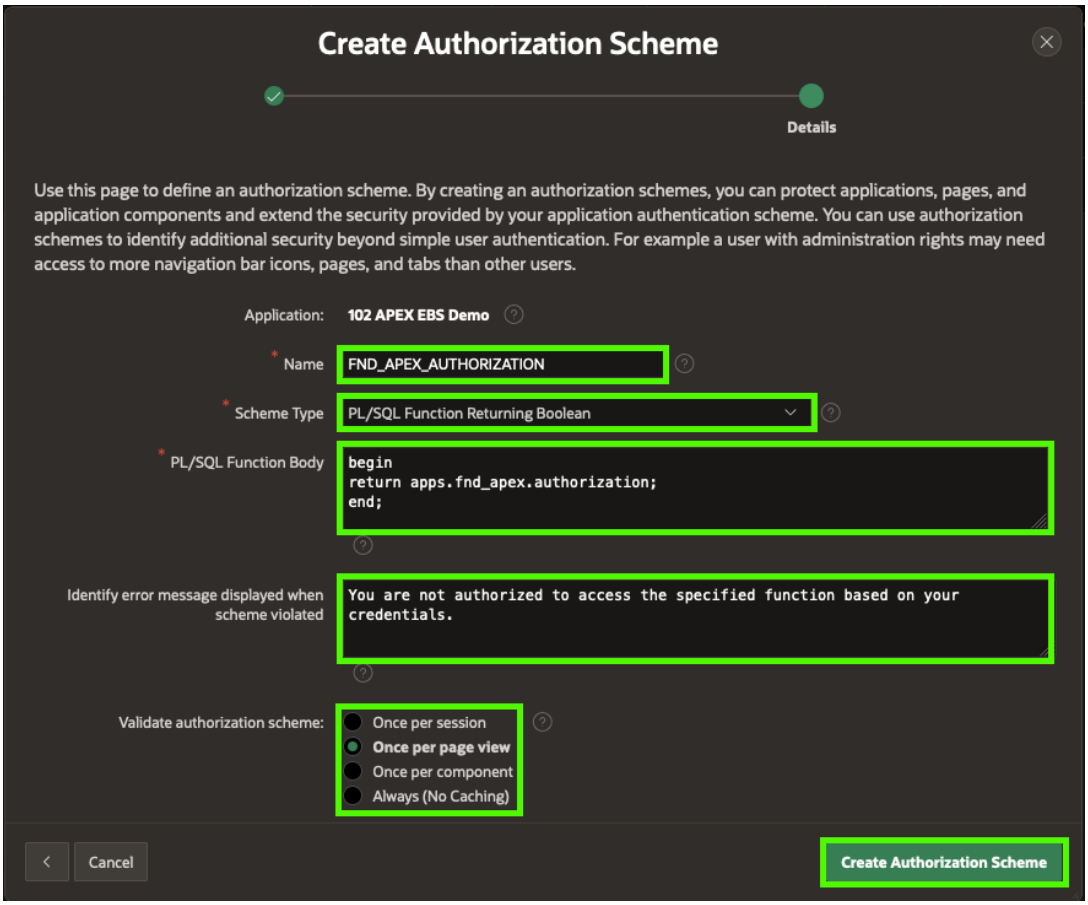

Figure 27. Create Authorization Scheme

Applying the Authentication and Authorization Schemes to the Pages of Your Application

In the respective pages, select the created authorization scheme from the **Authorization Scheme** drop-down list. For **Authentication**, select “Page Requires Authentication”.

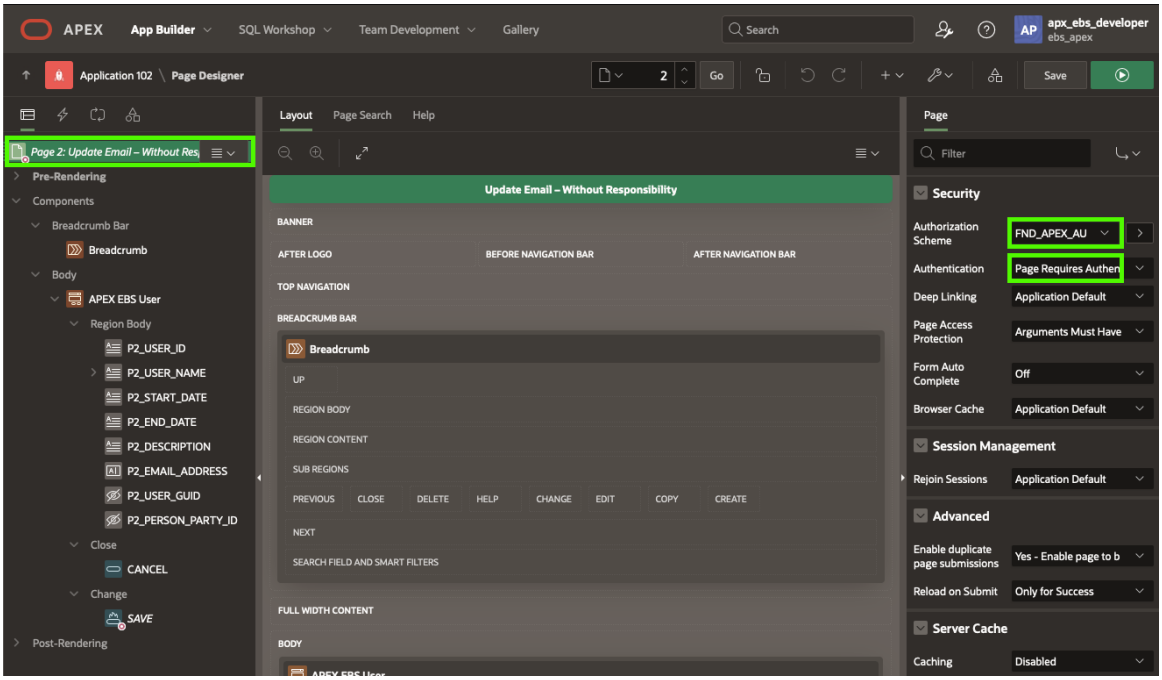

Figure 28. Define Authorization Scheme and disregard Authentication at Page level

For Page 3, make sure Page Access Protection is set to **Unrestricted**.

Make sure you click the **Save** button on the Page Designer after setting there parameters.

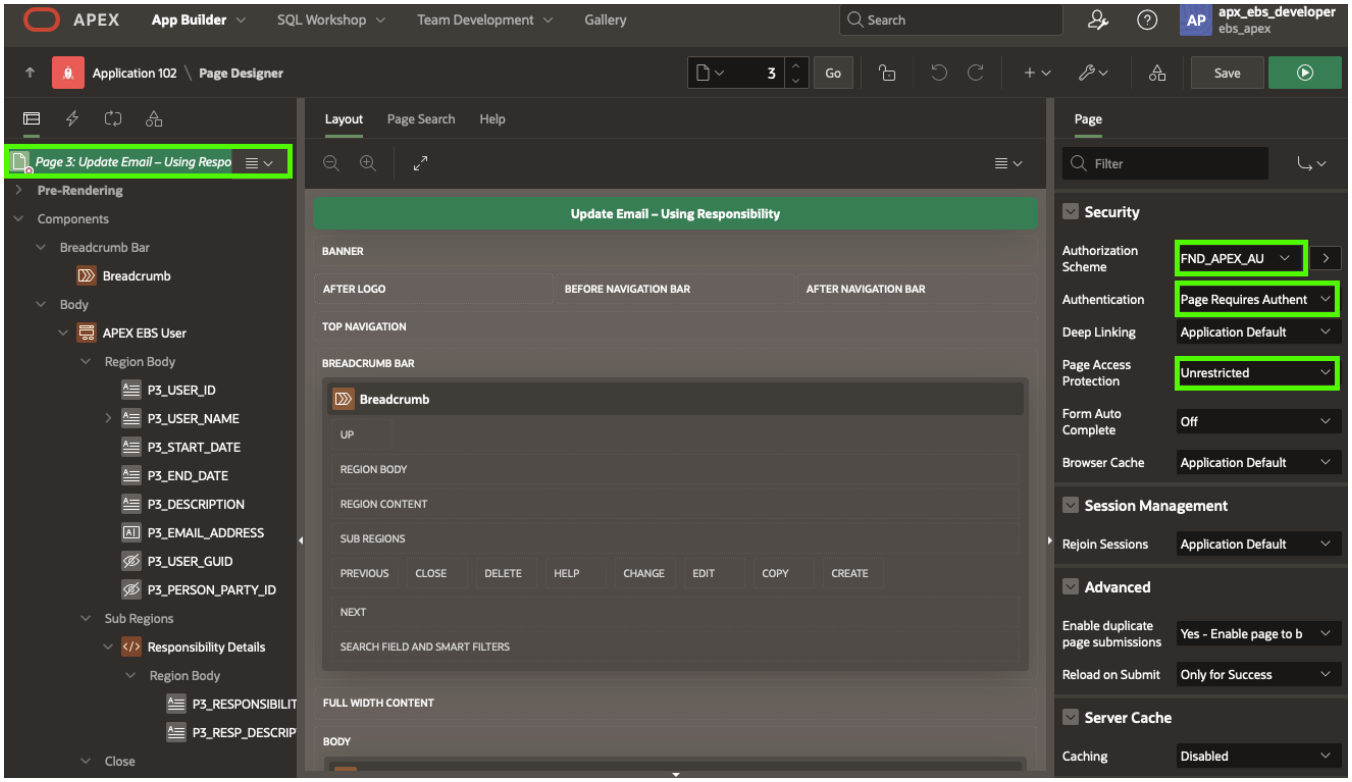

Figure 29. Page 3 Authorization Scheme and Page Access Protection

Enabling Deep Linking and Setting the Timeout URL

Follow these instructions to allow deep linking to the application when called from Oracle E-Business Suite:

1. From the Application Builder for the APEX EBS Demo application, click **Shared Components**.
2. In Security (middle pane), click **Security Attributes**.
3. Under Session Management, in the **Deep Linking** drop-down list select Enabled.
4. In the **Session Timeout URL** field, enter `https://<EBS_hostname>:<port>/OA_HTML/OAParentReload.jsp`.
5. In the **Maximum Session Idle Time in Seconds** field, enter the value found for the ICX: Session Timeout profile option in your Oracle E-Business Suite environment. For more information, see ICX: Session Timeout in *Oracle E-Business Suite Security Guide*.
6. In the **Session Idle Timeout URL** field, enter `https://<EBS_hostname>:<port>/OA_HTML/OAParentReload.jsp`.
7. Click **Apply Changes**.

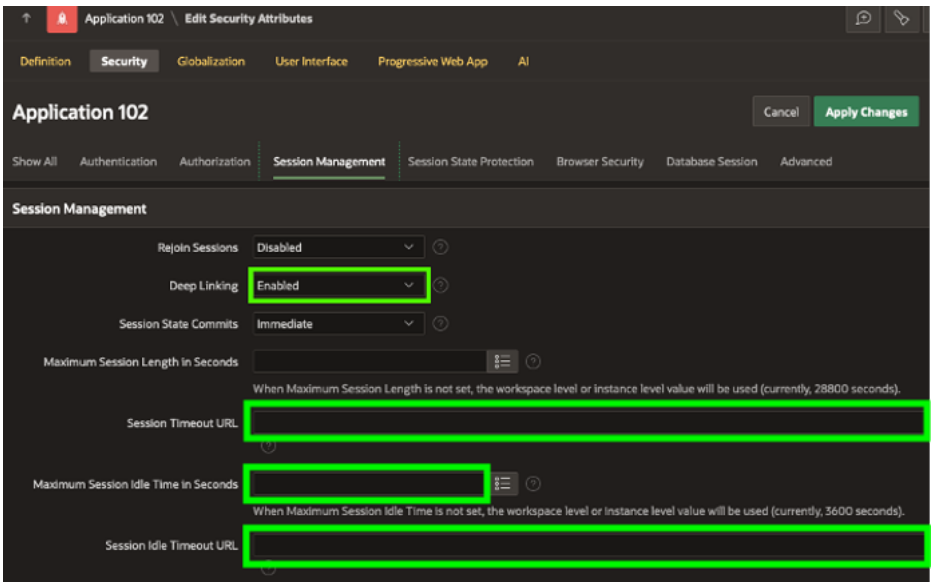

Figure 30. Deep Linking enablement

Enabling Embedded Mode on Your APEX Application

Set the Oracle E-Business Suite URL as CSP Frame ancestors under HTTP Response Headers in the APEX Application security attributes by performing the following steps:

1. Navigate to **Application**, **Shared Components**, **Security Attributes**, **Security**, and then select **Browser Security**.
2. For **Embed in Frames**, select Allow from same origin from the drop-down list.
3. For HTTP Response Headers, enter the following:

```
Content-Security-Policy: frame-ancestors https://<EBS_hostname>:<port>;
```

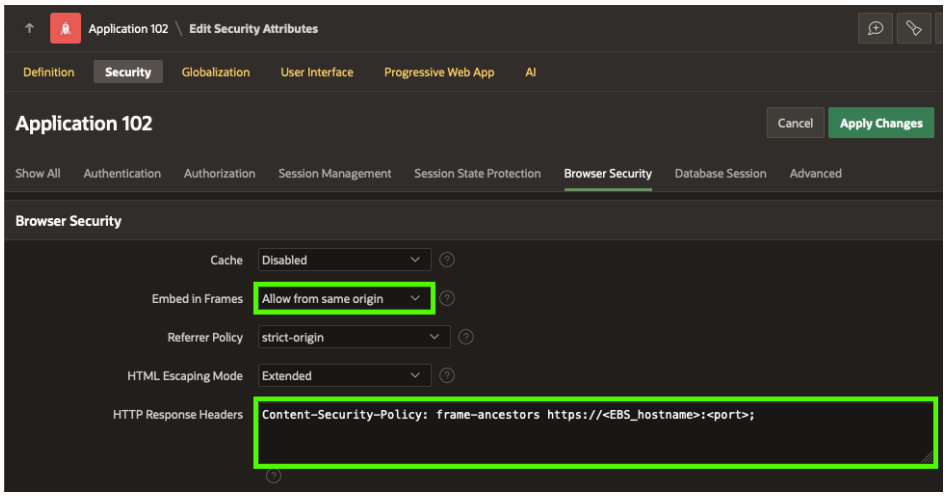

Figure 31. Allowing APEX pages to be embedded in frames

Using the Sample Packaged Application to Automate Development of the Sample APEX Application

To quickly configure Oracle APEX within your Oracle E-Business Suite environment, a sample Oracle APEX application is available which includes working examples for updating the user's email address based on the steps performed previously in this paper.

To download from Oracle Technology Network (OTN) and install the sample application, perform the following steps:

1. Download the Oracle APEX sample application script file from OTN – https://www.oracle.com/a/otn/docs/APEX_EBS_Extension.sql.
2. Save the file as `APEX_EBS_Extension.sql` on your local desktop.
3. Log in to Oracle APEX Application Builder (`https://<EBS_hostname>:<port>/ords`).
4. From the Application Builder, click **Import**.
5. Select **Import File APEX_EBS_Extension.sql**.
6. Click **Next**, then click **Install**.
7. Update the sample application Login Page, Timeout URLs and HTTP Response Headers with your EBS URL by performing the steps described in the following sections:
   a. Defining the Sample Application Login Page
   b. Enabling Deep Linking and Setting the Timeout URL
   c. Enabling Embedded Mode on Your APEX Application
8. Click **Run Application**.

> ✓ **Note: This application will produce errors at runtime unless the `apex_ebs_setup.sql` script has been run in the APPS schema as outlined earlier in this paper.**

Registering Your Custom Sample APEX Application with Oracle E-Business Suite

To register your custom sample APEX application with EBS, perform the following steps:

1. Defining Oracle E-Business Suite Profile and Form Functions
2. Defining Oracle E-Business Suite Function Menus and Responsibilities

As an alternative to these steps, you can utilize the script `apex_ebs_env.sql` to automatically integrate your custom APEX application with Oracle E-Business Suite. To do so, follow the steps in Using the Environment Script to Automate Oracle APEX Application Configuration with Oracle E-Business Suite.

Defining Oracle E-Business Suite Profile and Form Functions

This configuration requires setting an Oracle E-Business Suite profile option (FND: APEX URL) to contain the Oracle REST Data Services host name, and defining Oracle E-Business Suite functions that point to the actual APEX pages.

Setting the FND:APEX URL Profile Option

Update the FND: APEX URL profile option with the correct setting at the site level using the following steps:

1. Log in to Oracle E-Business Suite as the `SYSADMIN` user (`https://<EBS_hostname>:<port>/OA_HTML/AppsLogin`).
2. Under the Functional Administrator responsibility, navigate to **Core Services**, then click **Profiles**.
3. Search for `Name %APEX%` and click **Find**.
4. Select the profile option name FND: APEX URL.
5. Under the Site tab, enter `https://<EBS_hostname>:<port>/ords` in the **Value** column.
6. Click **Update** to save the profile value.

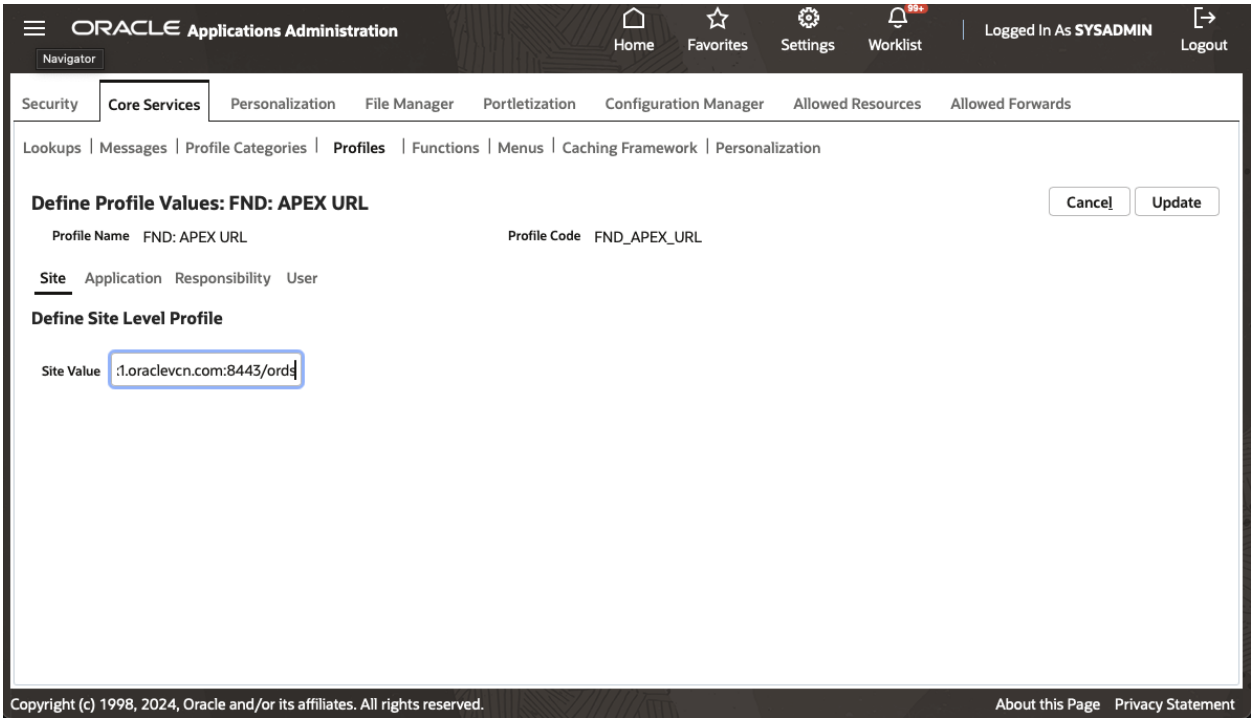

Figure 32. Defining the Oracle E-Business Suite Profile option value

> ✓ **All steps needed to perform within Oracle E-Business Suite can automatically be performed by running the *apex_ebs_env.sql* script outlined later in this paper.**

Defining the Oracle E-Business Suite Functions

Perform the following steps to define the Oracle E-Business Suite functions:

1. Under the Functional Administrator responsibility, navigate to Core Services, then click Functions.
2. Click the "Create Function" link.
3. On the Create Function: Define Function page:
    a. **Name**: Specify a name for the function, “Update Email (Without Responsibility)”
    b. **Code**: Specify a code for the function, XX_APEX_UPDATE_EMAIL_NR. The code is the internal name for the function. Once the function has been saved, the code cannot be updated.
    c. **Type**: Select "APEX".
    d. **Maintenance Mode Support**: Select "None".
    e. **Context Dependence**: Select "Responsibility".
    f. Click **Continue**.

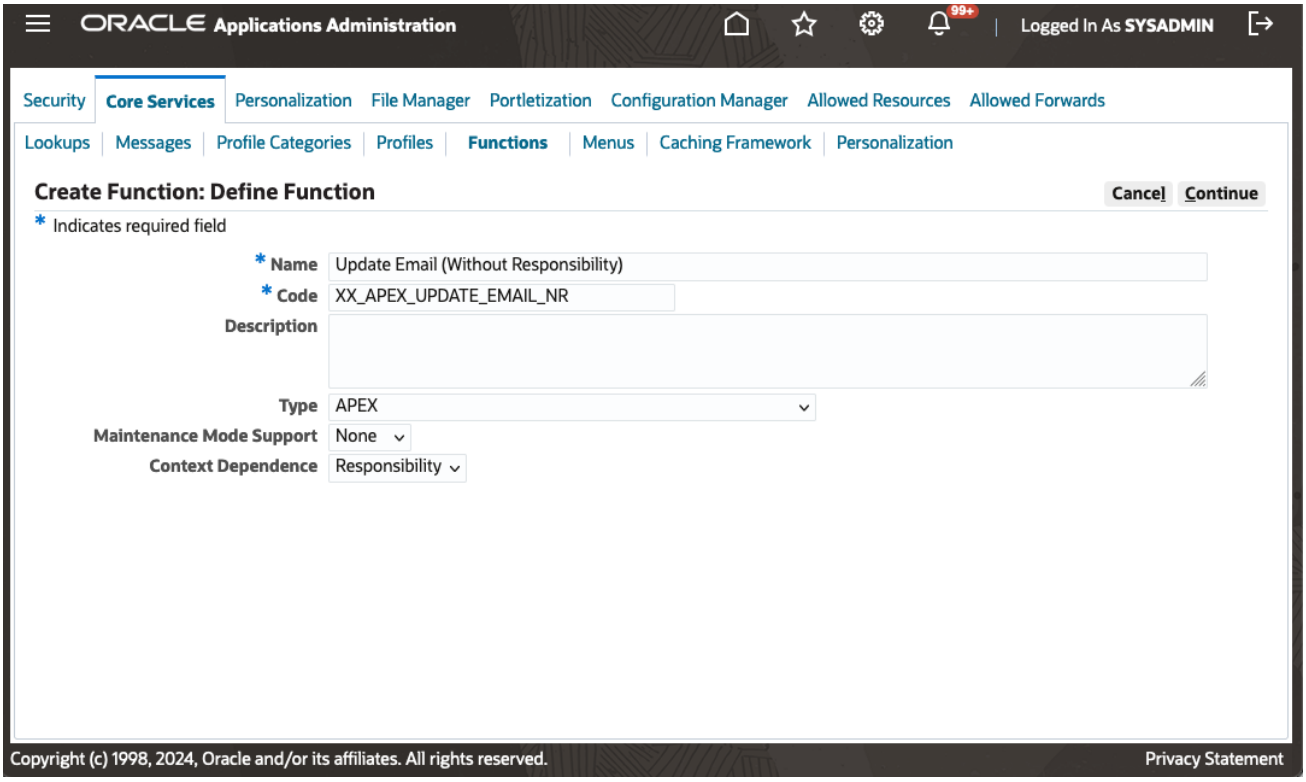

Figure 33. Create Function: Define Function Page

4. On the Create Function: Details page:
    a. **Layout**: Select your layout - either "Embedded" or "Full Window".
    b. **Workspace**: Select the workspace. For this procedure, select EBS_APEX.
    c. **Application**: Select the APEX application APEX EBS Demo.
    d. **Page**: Update Email – Without Responsibility.

e. For additional parameters, click + to add and define them in the following table. Enter a **Name** and **Value** pair.

f. Click **Submit**.

For more information, see: Functions, *Oracle E-Business Suite Security Guide*.

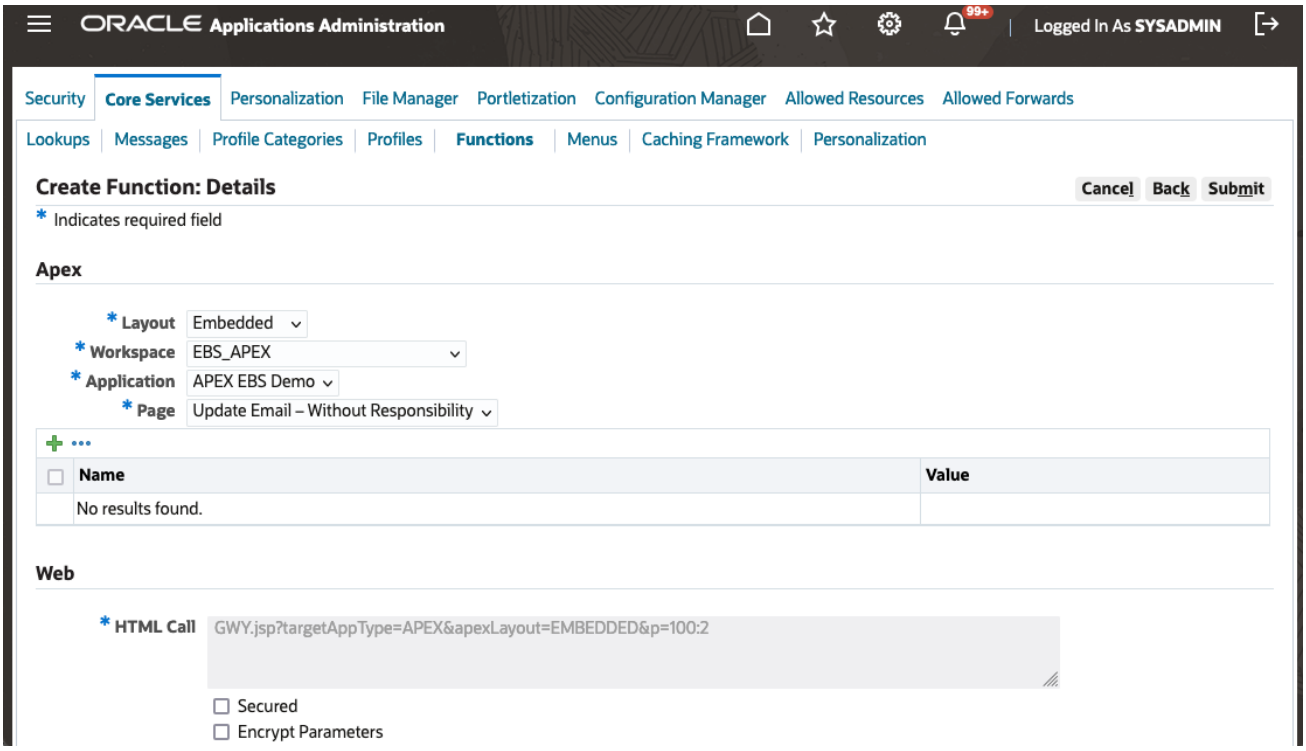

Figure 34. Create Function: Details Page

For the second page (Update Email with Responsibility), create a function with the following details:

1. On the Define Function page, enter the following:
   - **Function**: XX_APEX_UPDATE_EMAIL_WR
   - **User Function Name**: Update Email (With Responsibility)
   - **Type**: APEX
2. On the Function Details page:
   a. Select the appropriate **Layout**, **Workspace**, **Application**, and **Page**.
   b. Add the following parameters by clicking the green + symbol:
   - **Name**: EBS_RESP_ID; **Value**: [RESPONSIBILITY_ID]
   - **Name**: EBS_APP_ID; **Value**: [RESP_APPL_ID]
   - **Name**: EBS_SEC_GROUP; **Value**: [SECURITY_GROUP_ID]

Note: Parameter values in squared brackets "[ ]" are calculated dynamically at runtime, their value will be set based on the applications context of the authenticated user at runtime. You could also use constant values.

3. Click **Submit** to save the functions.

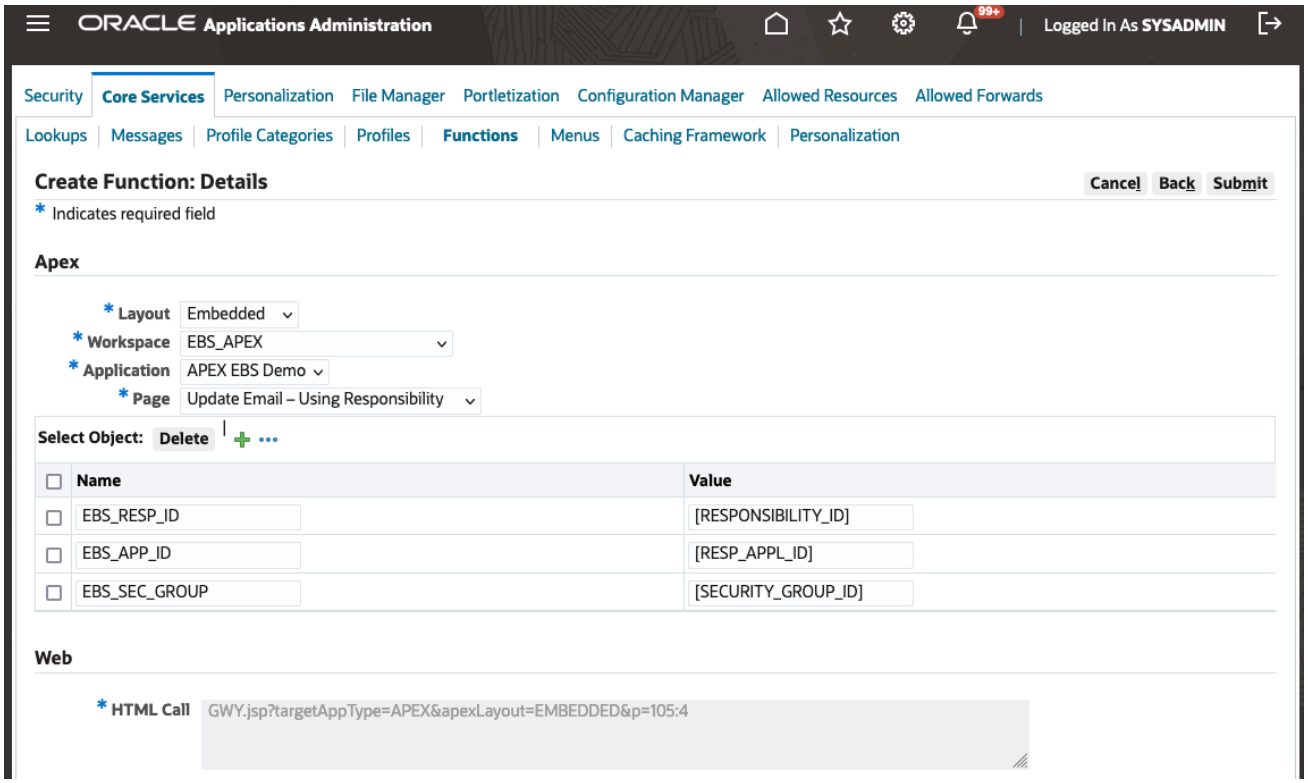

Figure 35. Create Function with Additional Parameters: Details Page

Defining Oracle E-Business Suite Menus and Responsibilities

The next step is to define menu options and responsibilities for your Oracle APEX applications within the Oracle E-Business Suite environment. Once the function is defined you may attach it to appropriate Menus and Responsibilities. Users having access to such responsibilities will be able to see the link in the Oracle E-Business Suite home page. Once they click the link the current browser will launch the targeted APEX page in Oracle APEX.

Adding an Oracle E-Business Suite Menu

Perform the following steps to add an Oracle E-Business Suite menu:

1. Under the Functional Administrator responsibility, navigate to **Core Services**, then click **Menus**.
2. Create a new menu:
   - **Name**: APEX Demo Functions
   - **Code**: XX_APEX_MENU
   - **Description**: Menu for Oracle APEX Demonstration app.
   - **Menu Type**: Home Page
3. Create menu component:
   - **Prompt**: Update Email
   - **Function**: Update User Email (Without Responsibility)
     {Form function XX_APEX_DEMO_1}
4. Create menu component:
   - **Prompt**: Update Email (Using Responsibility)
   - **Function**: Update User Email (Using Responsibility)
     {Form function XX_APEX_DEMO_2}
5. Click **Apply** to save the menu.

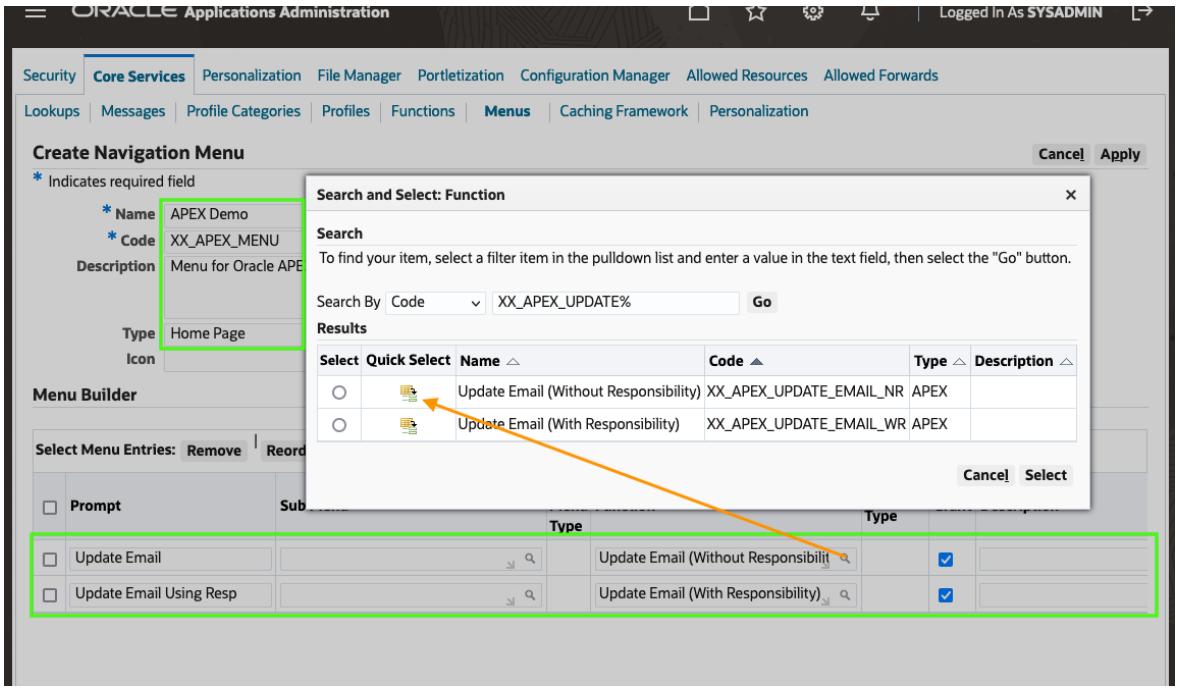

Figure 36. Creating Oracle E-Business Suite Menu

Adding an Oracle E-Business Suite Responsibility

Perform the following steps to add an Oracle E-Business Suite responsibility:

1. Navigate to the **User Management** responsibility, **Responsibility**, and then select the **Create Responsibility** option.
2. Create a new responsibility:
   - **Responsibility Name**: XX_APEX_RESP
   - **Menu**: XX_APEX Demo
   - **Application**: Application Object Library
   - **Responsibility Key**: XX_APEX_RESP
   - **Description**: Responsibility for APEX Demonstration
   - **Available From**: Oracle Self Service Web Applications
   - **Data Group**: Standard
   - **Application**: Application Object Library
3. Click **Create** to save the responsibility.

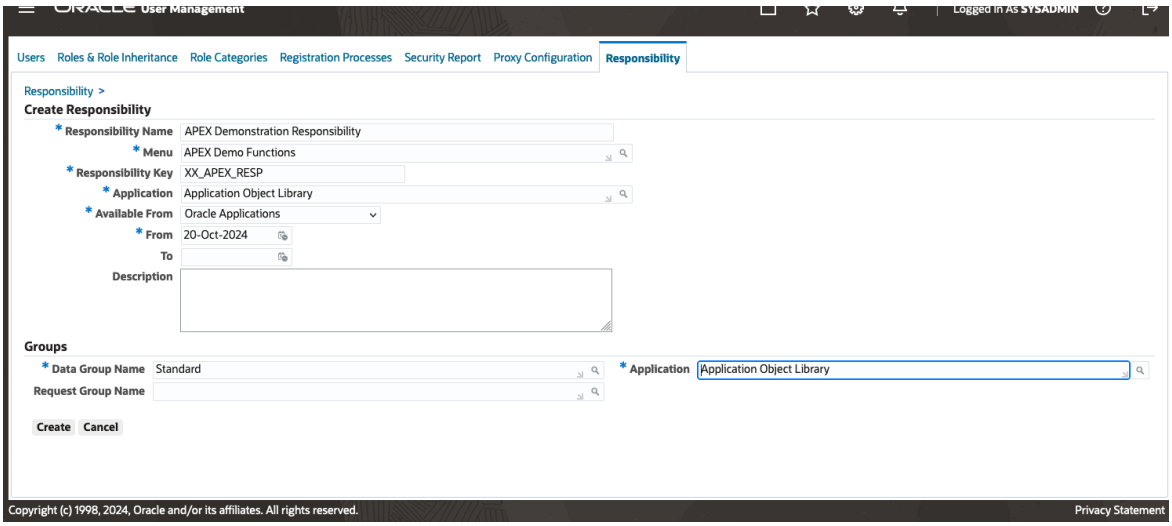

Figure 37. Creating an Oracle E-Business Suite responsibility

Adding a User

Perform the following steps to add a user:

1. Navigate to the User Management responsibility, select **Users**, select **User Account for Register**, and then click **Go**.
2. Create a new user account by entering the listed values in the following fields on the Create User Account page:
   - **User Name**: APEX
   - **Active From**: Leave this value as-is (it should be the current date)
   - **Password**: Enter a password for the APEX user.

     Note: The **Password** field will only be visible if the profile option Manual Password Reset Enabled is set to Yes.
3. Click **Submit**.

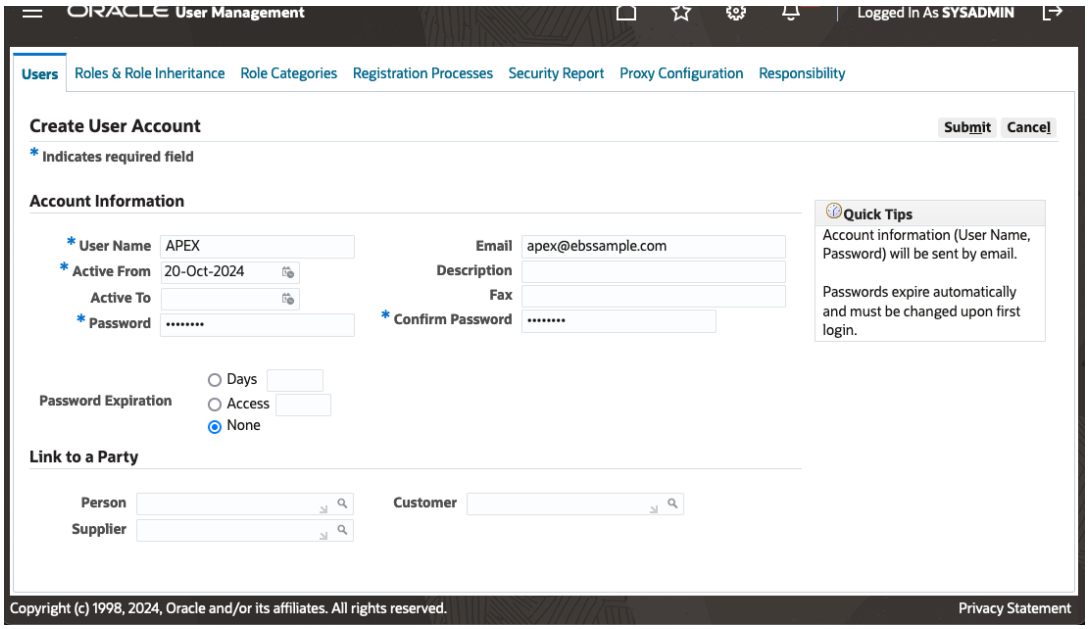

Figure 38. Creating an Oracle E-Business Suite user

4. Click **Assign Roles**.
5. Under Roles, click **Assign Roles**.
6. Search for `XX_APEX_RESP`.
7. Supply a justification.
8. Click **Save**.

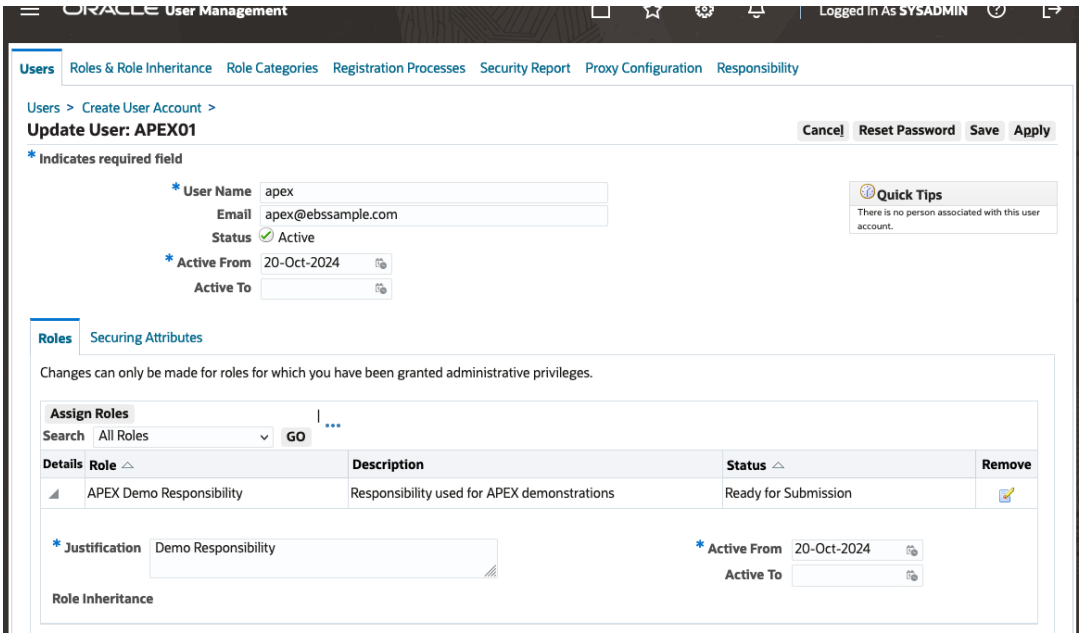

Figure 39. Assigning a Responsibility to an APEX user

Perform the following steps to complete the configuration:

1. Navigate to **System Administrator** responsibility, **Concurrent**, then **Requests**. Click **Submit a New Request**.
2. For **Name**, select Compile Security Menu. Click **Submit**.
3. Click **Find** and review the request to ensure it has completed successfully.
4. Navigate to the Functional Administrator responsibility, **Core Services**, and then **Caching Framework**.
5. Click **Global Configuration**, click **Clear All Cache**, and then click **Yes**.

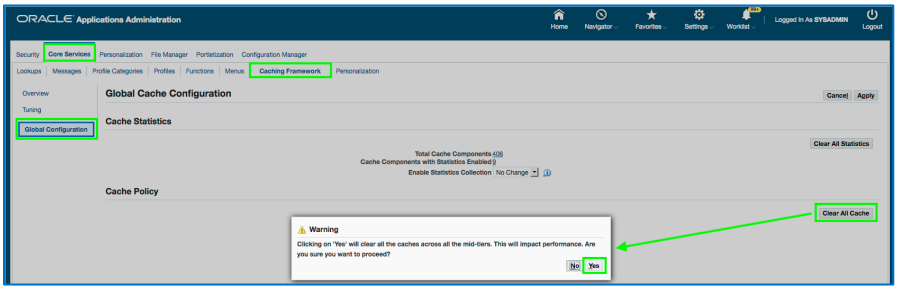

Figure 40. Clearing the cache

6. Log out of Oracle E-Business Suite.
7. Log in to Oracle E-Business Suite as the APEX user (`https://<EBS_hostname>:<port>/OA_HTML/AppsLogin`).

Using the Environment Script to Automate Oracle APEX Application Configuration with Oracle E-Business Suite

You can automate integration by utilizing the script `apex_ebs_env.sql` to quickly integrate your custom APEX application with Oracle E-Business Suite. This script configures the FND profile option, adds the functions, a menu, and a responsibility and then creates a user and associates the responsibility with the given user.

To run this script, perform the following steps:

1. Download the environment script file from Oracle Technology Network (OTN) – https://www.oracle.com/a/otn/docs/apex_ebs_env.sql.
2. Save the file as `apex_ebs_env.sql` and upload to your Oracle E-Business Suite database server.
3. Connect to your Oracle E-Business Suite database server.
4. Start SQL*Plus and connect as `APPS`.
5. Run the script, providing the appropriate criteria:
   `@apex_ebs_env.sql`
6. Enter the following values when prompted:
   - APEX_URL (Profile value)
   - APEX Application Id
   - APEX Page Number Without Responsibility [2]
   - APEX Page Number Using Responsibility [3]
   - EBS user name to be associated with Menu
   - EBS user name password
   - EBS user name email address

Running Your Custom Oracle APEX Application from E-Business Suite

Now that you have successfully built your custom sample Oracle APEX application, you can run Oracle E-Business Suite and select your new menu links to access your Oracle APEX application. Use the following steps to test your application:

1. Log in to Oracle E-Business Suite using `APEX` / `<APEX_Password>` (`https://<EBS_hostname>:<port>/OA_HTML/AppsLogin`).
2. Change the default password as prompted after you log in.
3. Navigate to the APEX Responsibility and click the **Update Email** function.

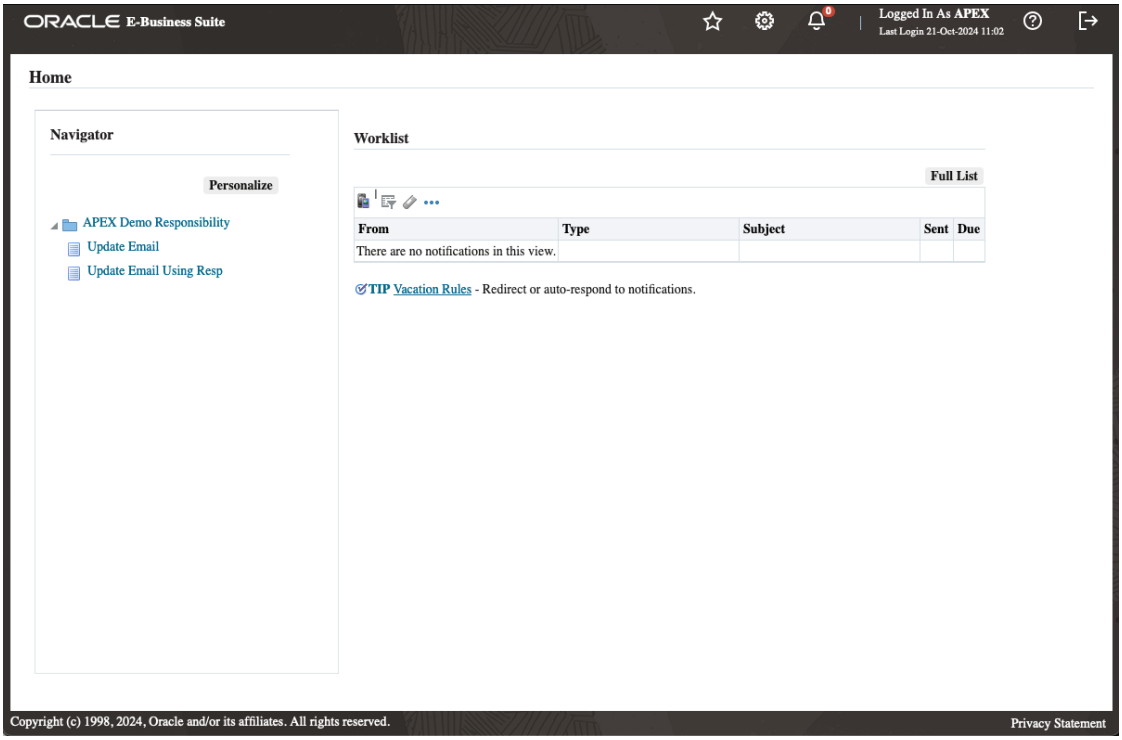

Figure 41. Oracle E-Business Suite home page

4. Update the Email Address and click **Apply Changes**. This will return you to the application home page.

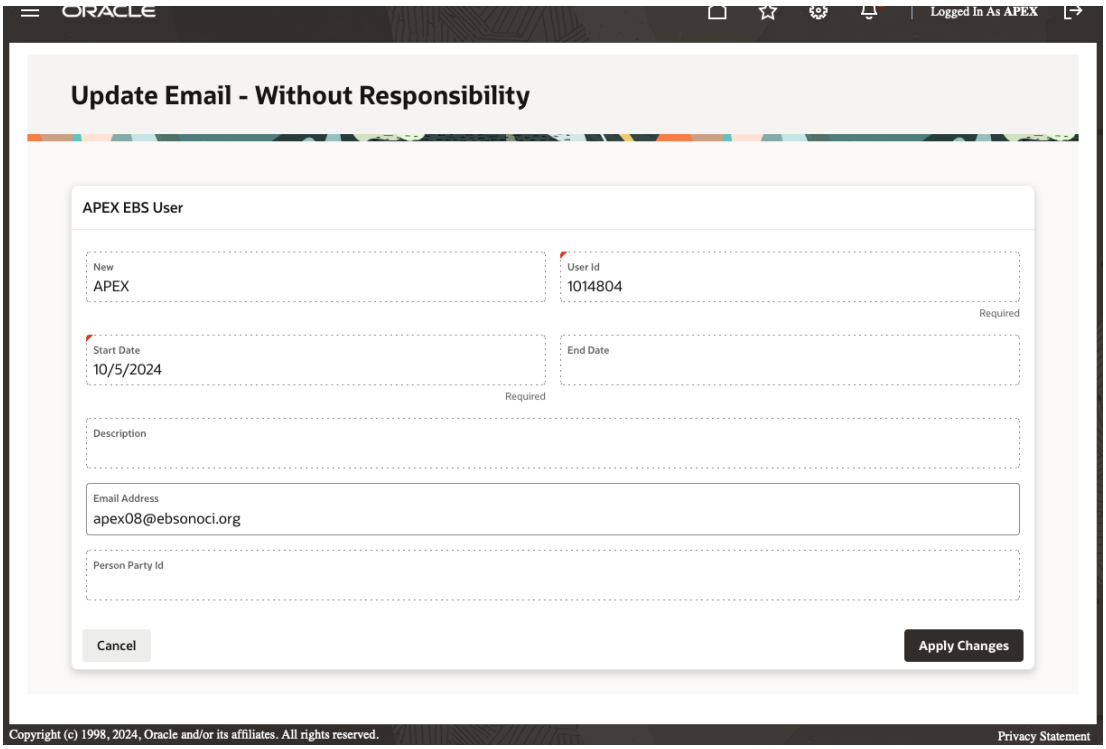

Figure 42. Oracle APEX application page for updating user email

5. Go back to your Oracle E-Business Suite home page, by clicking the **Home** button on the top of the screen.
6. Click **Update Email (Using Responsibility)**. Update the Email Address and click **Apply Changes**.
   This will redirect you to the Oracle APEX application and set your Oracle E-Business Suite responsibility. The responsibility region will display your current responsibility.

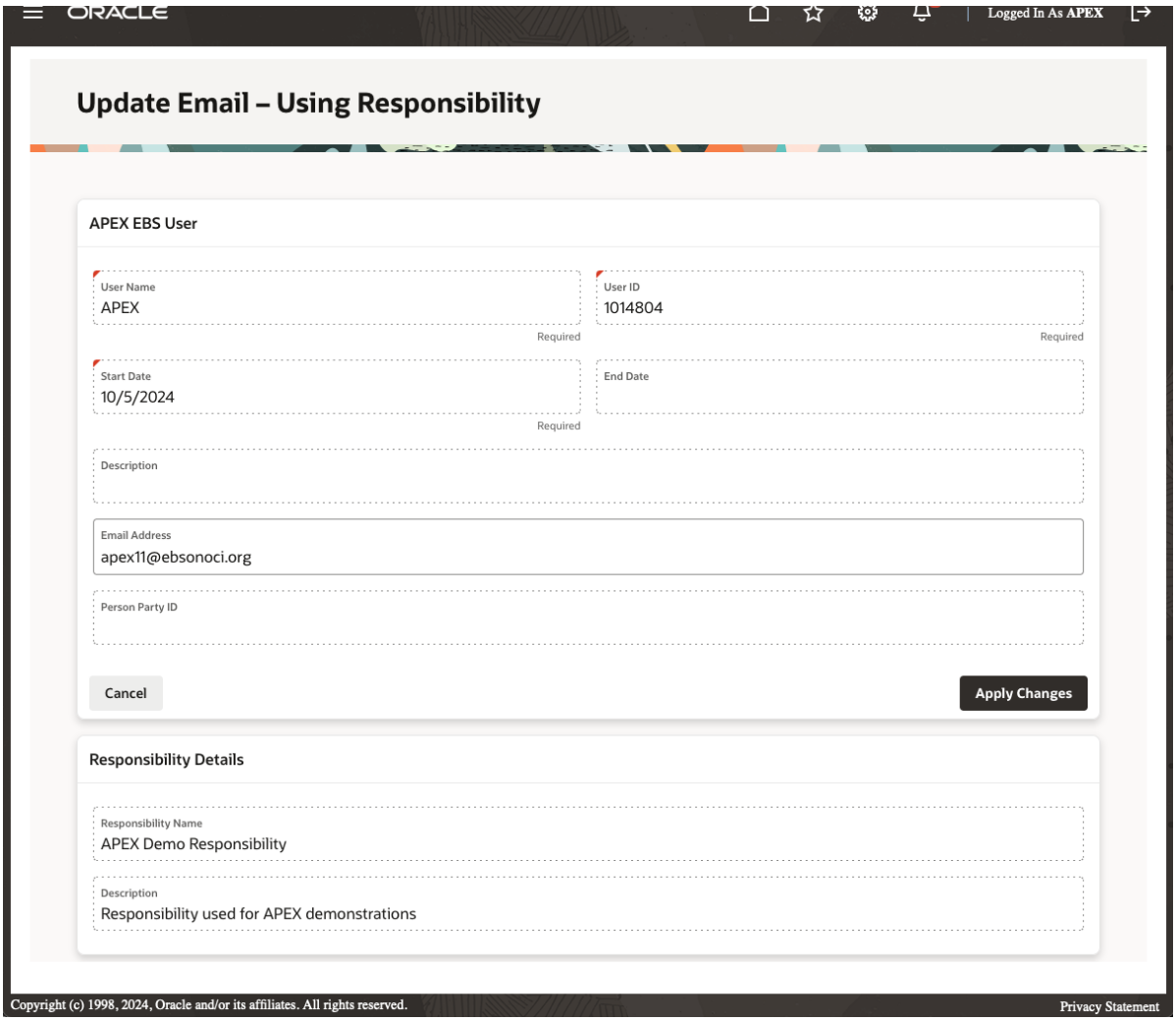

Figure 43. Oracle APEX application page showing responsibility

Developing and Deploying Your APEX Application in a Registered APEX Workspace

The following guidelines and standards are based upon Oracle E-Business Suite and Oracle APEX parameters and configurations.

Guidelines

Use the following guidelines for deploying and deploying your APEX application in a registered APEX workspace.

- Always use public APIs to interact with the application.
    - For reference, see the Oracle E-Business Suite Technology Blog article Integration Repository - Catalog of Oracle E-Business Suite APIs.
- Always access the EBS data model using views or synonyms in the APPS schema.
    - For reference, see the Oracle E-Business Suite Technology Blog article Update eTRM for EBS 12.2 Now Available.
- Add a custom data model in your own schema and register it with EBS.

Standards

Use the following standards when developing and deploying your APEX application:

- Follow Online Patching development standards to create and maintain your own DB objects
    - For reference, see My Oracle Support Knowledge Document 1577661.1, *Developing and Deploying Customizations in Oracle E-Business Suite Release 12.2*.
- Follow EBS System Schema migration project standards.

Procedure

Deployment instructions are available in My Oracle Support Knowledge Document 1577661.1, *Developing and Deploying Customizations in Oracle E-Business Suite Release 12.2*.

Upgrading Your APEX Extensions

Now that you have your initial APEX extension to Oracle E-Business Suite deployed, it is important to understand how to enhance your APEX applications within this integrated environment.

To fix bugs, modify existing functionality, enhance an application, or implement a new application, you will want to import an APEX application into your QA/Test and then Production environments. This application import may also require database object upgrades.

When there are no modifications or new database objects required, enhanced APEX applications can simply be imported into the relevant environments. This can be performed without impacting the Oracle E-Business Suite environment and while users are utilizing the extensions. The user will pick up the latest application definition when they next access the imported application. If they were already in that application, then they will pick up the new application definition when they next request or submit a page.

If an updated or new APEX application requires only new database objects, such as a new table, view, or database package, then these new database objects should be implemented into the environment before the APEX application is imported. For Oracle E-Business Suite Release 12.2 environments, Oracle recommends creating these new database objects outside of the E-Business Suite Online Patching Cycle. Once the new database objects are available then the APEX application can be imported. Again, this can be performed while users are utilizing the extensions.

However, if an updated or new APEX application requires modification or updates to existing database objects or existing data then this should be performed while users are prevented from accessing the extensions. For Oracle E-Business Suite Release 12.2 environments, Oracle recommends implementing these changes within an E-Business Suite Online Patching Cycle.

For more information on how to deploy such applications, see My Oracle Support Knowledge Document 1577661.1, *Developing and Deploying Customizations in Oracle E-Business Suite Release 12.2*.

For example, if you wanted to enhance the application developed earlier in this paper to allow users to update “Known As” as well as their email address, then you would need to modify existing database objects before importing the enhanced APEX application. You would need to update the XX_APEX_EBS_USER view to include the known_as column from the `FND_USERS` table, and update the apex_update_email function within the xx_apex_sample_apis package to accept an extra parameter and update the API call.

In an Oracle E-Business Suite Release 12.2 environment, the Oracle Database feature Edition-Based Redefinition (EBR) is employed within your database for Oracle E-Business Suite Release 12.2 Online Patching. You should utilize this capability to deploy changes to database objects within the schema or schemas associated with Oracle APEX workspaces, such as XX_APEX. The previous Oracle recommendation allows the database administrators to implement all modifications to database objects within the patch edition, so that users are not impacted, except for minimal downtime when the Oracle WebLogic Server is shutdown while the APEX application is imported. In fact, it is critical to ensure that you do not implement such database object changes in the run edition, as any changes will be overwritten once the E-Business Suite Online Patch Cycle is completed.

Given that the Oracle APEX Engine (for APEX Release 24.1, that schema is APEX_240100) is not editioned, when you import a new APEX application definition, it will be “live” immediately to all users. This is irrespective of if Oracle E-Business Suite is currently in Online Patching Cycle or not. If the new application relies on updated database objects and those updates have been performed in the patch edition, then your applications will not see those updated database objects as they will be referencing the run edition. Therefore, it is very important to coordinate when required updates to underlying database objects and when APEX applications are performed, to minimize disruption to users.

Additional Oracle E-Business Suite Security Considerations

Oracle E-Business Suite Security Features

Starting with Oracle E-Business Suite (EBS) Release 12.2.6, new security features are in place to reduce the attack surface of EBS and making it harder to use redirects. When using this feature, you may find unexpected access errors when the profile options and whitelists are not set properly.

For example, you may see an error such as "An invalid redirect has been blocked" when redirecting from EBS to APEX.

The issue is usually related to tightening security on the application server. There are specifically 3 new profile options that need your attention:

- Security: Allowed Resources
- FND: Security Resource Logging
- Allow Unrestricted Redirects

You must create an allowlist of JSPs and allowed redirects, depending on your organization security policy.

The following sections of *Oracle E-Business Suite Security Guide*, Release 12.2 (E22952) provide information on how to set an environment according to an organization requirement:

- Oracle Application Tier Security
- Allowed Resources
- Allowed Redirects

Known Issue During EBS Release 12.2 Online Patching

Some EBS Applications database administrators (DBAs) performing the online patching process on EBS Release 12.2 may encounter an incorrect privileges warning on the DUAL table when APEX is installed on EBS.

```
Validating system setup.
    Node registry is valid.
    [WARNING]    SYS.DUAL table has incorrect privileges
                 Shut down the application tier services and then run the following commands in the order shown
                    sqlplus apps/<APPS_Password> @ $AD_TOP/patch/115/sql/ADFIXUSER.sql
                    sqlplus apps/<APPS_Password> @ $AD_TOP/sql/adutlrcmp.sql
```

Figure 44. Example of output

See the following supporting My Oracle Support knowledge documents to resolve this issue:

- Document 2729531.1, *FLASHBACK Error on SYS.DUAL When Patching E-Business Suite 12.2 and APEX*
- Document 2312862.1, *Warning SYS.DUAL Table Has Incorrect Privileges*

Extras

With little bit more effort, you can utilize more powerful APEX components in your extension. The Sample Application contains more pages than shown in the previous section. The `apex_ebs_setup.sql` script does create a couple more views. Those views can be used to create an Employee Directory and Customer Locations based on data available in your EBS instance.

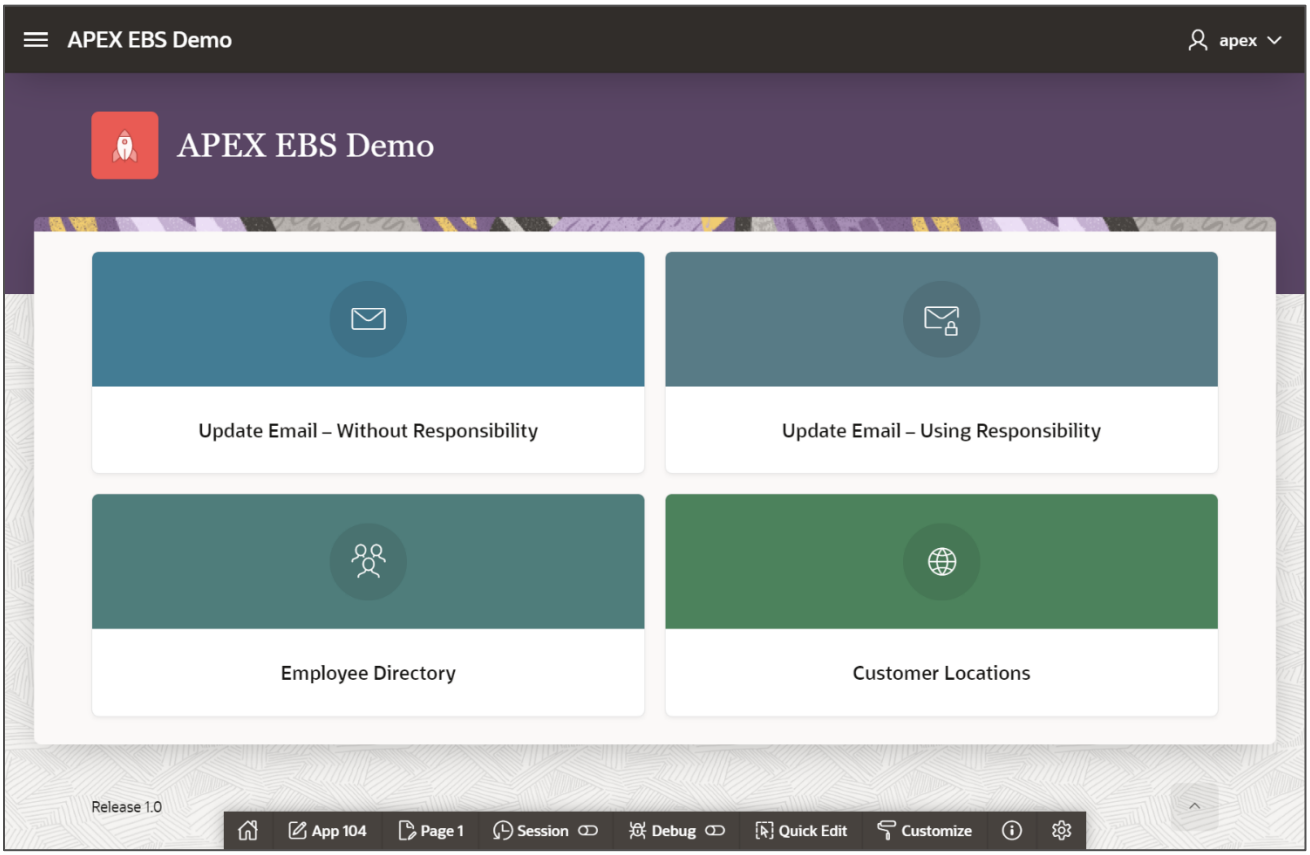

Figure 45. APEX EBS Demo page

Combining the power of Oracle APEX Faceted Search and the Cards region, in minutes you can effortlessly have a searchable employee directory.

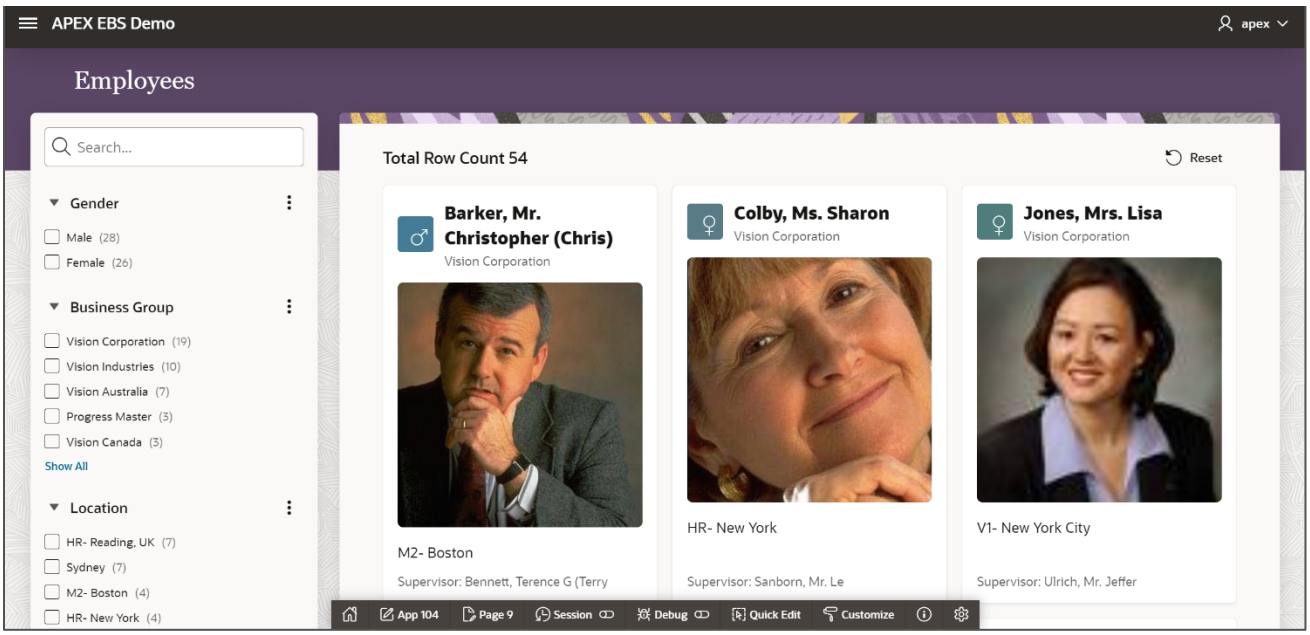

Figure 46. Employees Page in the APEX EBS Demo

The same technique can be used to create a Maps region integrated with Faceted Search. This allows you to create a map with locations for your customers and find them on the map interactively.

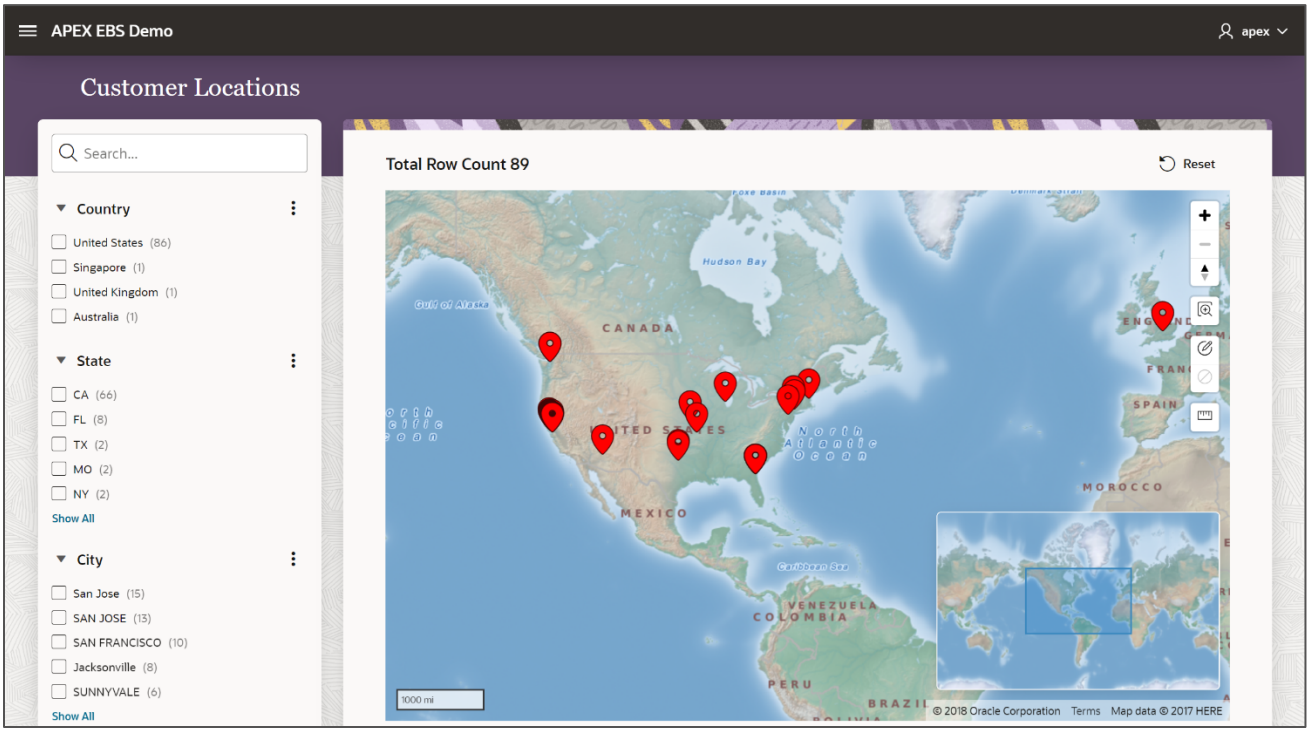

Figure 47. Customer Locations Page in the APEX EBS Demo

Conclusion

Oracle APEX is an excellent complement to Oracle E-Business Suite. As a development framework, Oracle APEX allows cross-functional teams to easily collaborate and quickly deliver solid custom extensions to Oracle E-Business Suite. The Oracle APEX footprint on an Oracle E-Business Suite environment is negligible but allows deeply integrated extensions using only standard customizations within the Oracle E-Business Suite implementation.

Use of a separate Oracle E-Business Suite custom schema for Oracle APEX with only the minimum required privileges for Oracle E-Business Suite objects ensures that Oracle APEX applications only have access to the interfaces and data required for their operation. This minimizes the security risk to the Oracle E-Business Suite inherent in making any extensions. By integrating with Oracle E-Business Suite Function Security (Authorization), Oracle APEX extensions utilize a single point of security administration. When a responsibility is added or removed from a user within Oracle E-Business Suite, the user's access to the associated functionality in Oracle APEX applications is affected likewise.

Further, the advantage of utilizing Oracle APEX over building extensions from scratch is that Oracle APEX provides a hardened declarative framework to minimize security vulnerabilities. Oracle APEX release procedures include extensive security testing to minimize the threat of security breaches from cross site scripting and SQL injection.

Oracle APEX gives you the power to quickly create forms, reports and complete applications rapidly, with little or no programming. With the proper deployment architecture, as outlined in this paper, you can use Oracle APEX to extend the capabilities of your Oracle E-Business Suite, using the same data for both sets of applications, sharing user authentication, and calling applications and components seamlessly.

Acknowledgments

This paper is a collaboration between the Oracle APEX and Oracle E-Business Suite teams. The Oracle E-Business Suite architects provided significant input to ensure the solutions provided meets the Oracle E-Business Suite best practices and presents fully-supported configurations. Further, those responsible for architecting the E-Business Suite Online Patching Cycle ensured the Oracle APEX extensions work correctly with Oracle E-Business Suite Release 12.2 and provided technical assistance in defining a script to automate the Oracle E-Business Suite environment configuration included with this paper.

ORACLE

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle