



Hardware and Software
Engineered to Work Together

Oracle Application Express Workshop II

Activity Guide
D79657GC20
Edition 2.0 | August 2015 | D92341

Learn more from Oracle University at oracle.com/education/

Authors

Salome Clement
Dimpi Sarmah

Technical Contributors and Reviewers

Bryan Roberts
Wayne Abbott
Iloon Ellen
Marc Swetz
Patrick Wolf
Sathish K Kumar
Shakeeb Rahman
Nancy Greenberg
Ashley.chen
Klaus.Husermann
Marcie Young
Andrew Rothstein
Anthony Rayner
Chaitanya Koratamaddi
David Peake
Denise Simpson
Krishnanjani Chitta
Yi Lu
Lakshmi Narapareddi
Fabrizio Siracusa
Hanson Hutabarat
Hilary Farrell
Ashwin Agarwal
Anjani Pothula

Editor

Nikita Abraham

Graphic Designers

Seema Bopaiah
Kavya Bellur
Maheshwari Krishnamurthy

Publishers

Syed Ali
Giri Venugopal
Jayanthy Keshavamurthy
Pavithran Adka

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Table of Contents

Course Practice Environment: Security Credentials	1
Course Practice Environment: Security Credentials.....	2
Practices for Lesson 1: Course Overview	1-1
Practices for Lesson 1.....	1-2
Practices for Lesson 2: Introduction and Review	2-1
Practices for Lesson 2: Overview.....	2-2
Practice 2-1: Accessing the Practice Environment.....	2-3
Practice 2-2: Reviewing the GlobalMart Management Tool	2-4
Practice 2-3: Reviewing the Project Tracking System.....	2-5
Solution 2-1: Accessing the Practice Environment.....	2-6
Solution 2-2: Reviewing the GlobalMart Management Tool	2-9
Solution 2-3: Reviewing the Project Tracking System.....	2-15
Practices for Lesson 3: Using Oracle APEX Collections.....	3-1
Practices for Lesson 3: Overview.....	3-2
Practice 3-1: Creating and Updating a Collection.....	3-3
Practice 3-2: Accessing a Collection	3-4
Solution 3-1: Creating and Updating a Collection.....	3-5
Solution 3-2: Accessing a Collection	3-13
Practices for Lesson 4: Creating Dynamic Actions	4-1
Practice 4-1: Creating Advanced Dynamic Actions	4-2
Solution 4-1: Creating Advanced Dynamic Actions	4-3
Practices for Lesson 5: Using Plug-ins in an Application.....	5-1
Practices for Lesson 5: Overview.....	5-2
Practice 5-1: Using a Region Plug-in	5-3
Practice 5-2: Using a Dynamic Action Plug-in.....	5-4
Practice 5-3: Using a Process Plug-in.....	5-5
Solution 5-1: Using a Region Plug-in	5-6
Solution 5-2: Using a Dynamic Action Plug-in.....	5-11
Solution 5-3: Using a Process Plug-in	5-22
Practices for Lesson 6: Incorporating Interactivity Using JavaScript and jQuery	6-1
Practices for Lesson 6: Overview.....	6-2
Guided Practice 6-1: Using jQuery in an APEX application	6-3
Guided Practice 6-2: Using JavaScript in an APEX application	6-13
Practices for Lesson 7: Generating and Using Table APIs	7-1
Practices for Lesson 7: Overview.....	7-2
Practice 7-1: Generating and Using Table APIs.....	7-3
Solution 7-1: Generating and Using Table APIs.....	7-4
Practices for Lesson 8: Creating and Using RESTful Web Services.....	8-1
Practices for Lesson 8: Overview.....	8-2
Practice 8-1: Creating and Using a RESTful Web Service	8-3
Practice 8-2: Creating and Consuming a RESTful Web Service	8-4
Solution 8-1: Creating and Using a RESTful Web Service	8-5
Solution 8-2: Creating and Consuming a RESTful Web Service	8-34

Practices for Lesson 9: Using Templates and Themes	9-1
Practices for Lesson 9: Overview	9-2
Practice 9-1: Copying and Customizing a Button Template	9-3
Practice 9-2: Creating a Custom Theme	9-4
Practice 9-3: Creating a Master Application	9-5
Solution 9-1: Copying and Customizing a Button Template	9-6
Solution 9-2: Creating a Custom Theme	9-27
Solution 9-3: Creating a Master Application	9-33
Practices for Lesson 10: Developing a New Theme for Your Application Using Theme Roller.....	10-1
Practices for Lesson 10: Overview.....	10-2
Practice 10-1: Developing a New Theme for the GMT Application	10-3
Solution 10-1: Developing a New Theme for the GMT Application	10-4
Practices for Lesson 11: Securing an Application.....	11-1
Practice 11-1: Preventing SQL Injection Attacks.....	11-2
Solution 11-1: Preventing SQL Injection Attacks.....	11-3
Practices for Lesson 12: Deploying an Application.....	12-1
Practice 12-1: Identifying and Creating Supporting Objects.....	12-2
Solution 12-1: Identifying and Creating Supporting Objects.....	12-3
Practices for Lesson 13: Optimizing Your APEX Application.....	13-1
Practices for Lesson 13: Overview.....	13-2
Practice 13-1: Monitoring Activity and Improving Performance of the GMT Application.....	13-3
Practices for Lesson 14: Globalization and Translation.....	14-1
Practices for Lesson 14: Overview.....	14-2
Practice 14-1: Translating the GMT Application	14-3
Practice 14-2: Specifying the Primary Language for the Translated GMT Application	14-4
Practice 14-3: Translating Messages Used in PL/SQL	14-5
Solution 14-1: Translating the GMT Application	14-6
Solution 14-2: Specifying the Primary Language for the Translated GMT Application	14-22
Solution 14-3: Translating Messages Used in PL/SQL	14-26

Course Practice Environment: Security Credentials

Course Practice Environment: Security Credentials

For OS usernames and passwords, see the following:

- If you are attending a classroom-based or live virtual class, ask your instructor or LVC producer for OS credential information.
- If you are using a self-study format, refer to the communication that you received from Oracle University for this course.

For product-specific credentials used in this course, see the following table:

Product-Specific Credentials		
Product/Application	Username	Password
Workspace Administrator User	apex_admin	apex
Workspace Developer User	apex	apex
Workspace Developer User	brad.knight	apex
Workspace Administrator User	susie.parker	apex
Workspace End User	john.bell	apex

Note: The credentials mentioned above are for the users of the **APEX** Workspace.

Practices for Lesson 1: Course Overview

Chapter 1

Practices for Lesson 1

There are no practices for Lesson 1.

Practices for Lesson 2: Introduction and Review

Chapter 2

Practices for Lesson 2: Overview

Practices Overview

In these practices, you access the course practice environment and identify the location of lab files and scripts. You review a Project Tracking System (PTS) application that is used to demonstrate the APEX features taught in this course. You also review the GlobalMart Management Tool (GMT) application that you build/modify to test your understanding of the concepts taught in the course.

Note: These applications were created in the demonstrations and practices for the *Oracle Application Express: Workshop I* course. *Oracle Application Express: Workshop I* is a prerequisite for this *Oracle Application Express: Workshop II* course. If you have not attended the *Oracle Application Express: Workshop I* course, you should be able to create the features available in these applications so far.

Practice 2-1: Accessing the Practice Environment

Overview

In this practice, you log on to the machine provided to you by your instructor and identify the files required to complete the rest of the practices in this course.

Tasks

- a. Log in to the practice environment assigned to you.
- b. Open a file browser and navigate to /home/oracle/labs.
- c. Review the folders available in the labs directory.

Practice 2-2: Reviewing the GlobalMart Management Tool

Overview

In this practice, you review the database objects and application pages for the GlobalMart Management Tool (GMT) application. This application is used in the rest of the practices in this course.

Tasks

- a. Open a web browser and connect to your APEX instance using the following URL:
`http://<IP address of machine assigned to you>:8080/apex`
- b. Log in to Oracle Application Express. Refer to the password document for the login credentials.
- c. From the Application Builder home page, run the GMT application.
- d. Perform the following actions in the GMT application:
 - View the list of customers.
 - View the list of products.
 - View details of any one of the products.
 - View the list of orders.
- e. Using the Developer toolbar, navigate to the GMT application home page.

Practice 2-3: Reviewing the Project Tracking System

Overview

In this practice, you review the database objects and application pages for the Project Tracking System (PTS) application. This application is used in the demonstrations in this course.

Tasks

- a. From the Application Builder home page, run the PTS application.
- b. Review the pages in the PTS Application.
- c. Using the Developer toolbar, navigate to the PTS application home page.
- d. Review the database objects in the PTS schema.

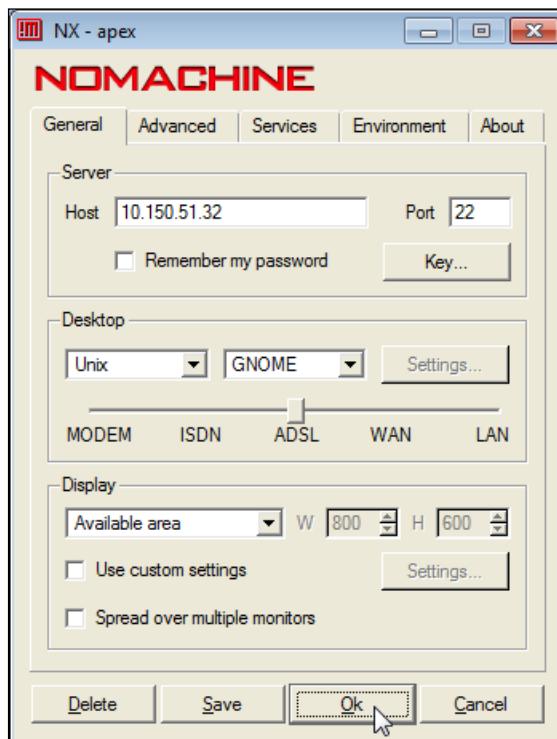
Solution 2-1: Accessing the Practice Environment

Overview

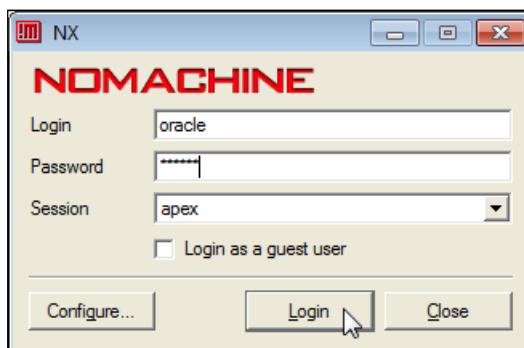
In this solution, the steps to log on to the machine provided to you by your instructor and identify the files required to complete the rest of the practices in this course are provided.

Steps

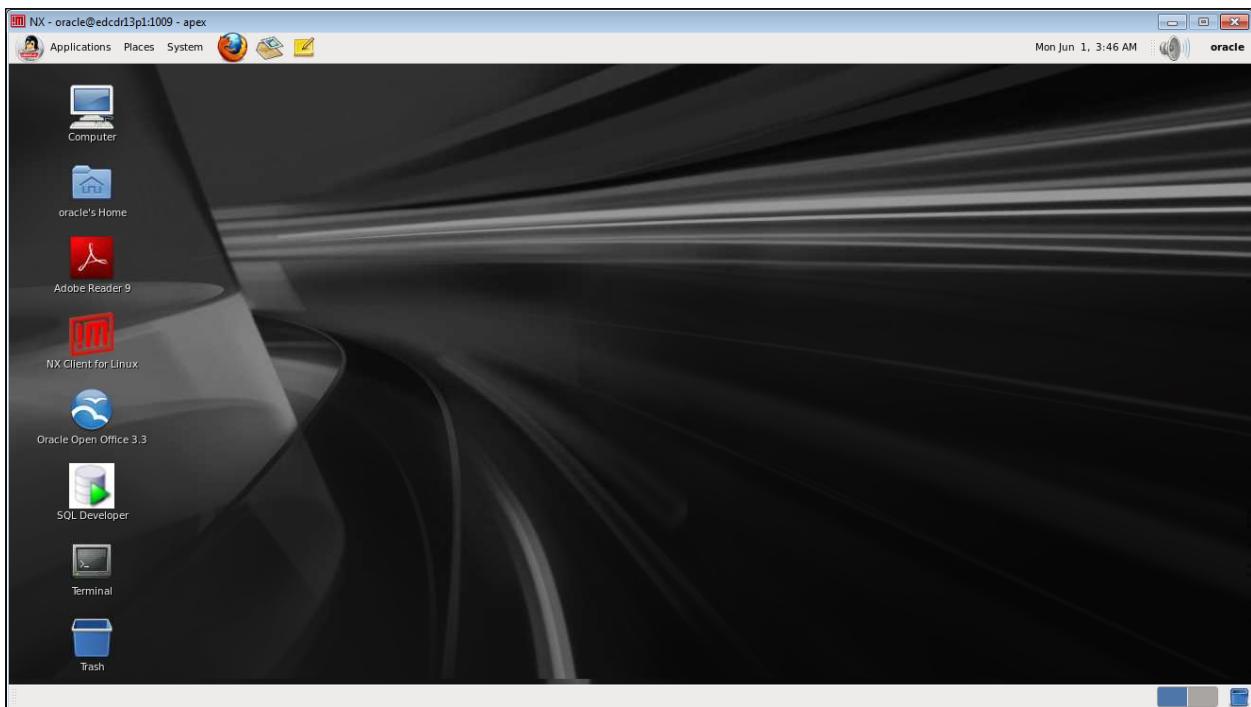
- a. Log in to the practice environment assigned to you.
 1. Start NX Client. Enter **apex** for Session and click **Configure**. Enter the IP address of the machine assigned to you. Set the port to **22** and click **Ok**.



2. Enter the username and password and click **Login**. Refer to the password document for the credentials.



3. The desktop of the machine assigned to you is displayed.

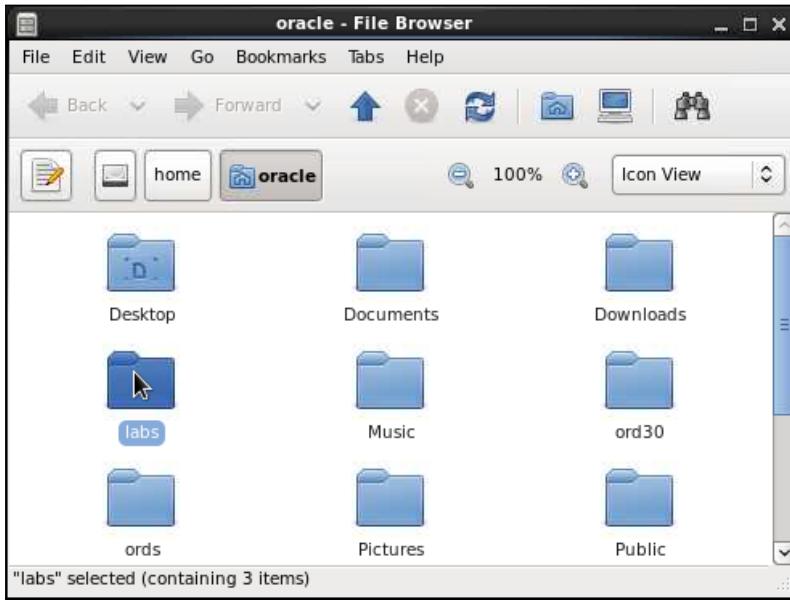


- b. Open a file browser and navigate to /home/oracle/labs.

1. In the Linux machine, from the top menu bar, select **Applications > System Tools > File Browser**.



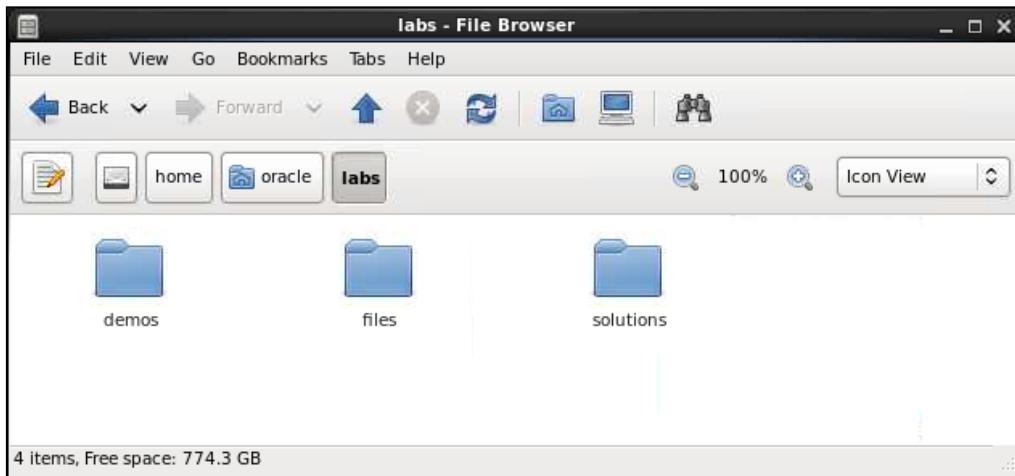
2. In the file browser, navigate to /home/oracle/labs.



- c. Review the folders available in the labs directory.

1. The labs directory has the following folders:

- **files:** Contains files you require to complete the practices.
- **demos:** Contains files that the instructor uses for demonstrations.
- **solutions:** Contains files with the completed tasks for each practice.



Solution 2-2: Reviewing the GlobalMart Management Tool

Overview

In this solution, the steps to review the database objects and application pages for the GlobalMart Management Tool (GMT) application are provided. This application is used in the rest of the practices in this course.

Steps

- Open a web browser and connect to your APEX instance using the following URL:

`http://<IP address of the machine assigned to you>:8080/apex`

- Click the Firefox icon in the top toolbar.

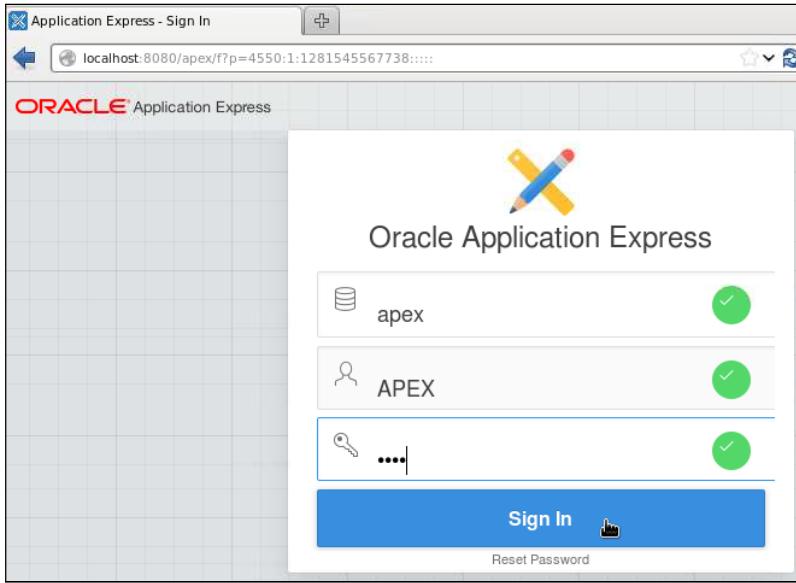


- Enter the APEX URL in the address bar and click **Go**.

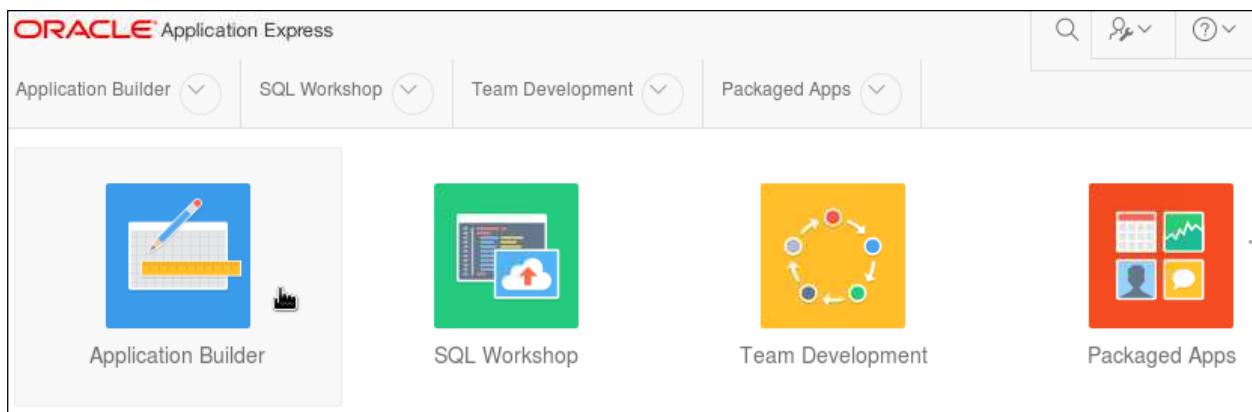


- Log in to Oracle Application Express. Refer to the password document for the login credentials.

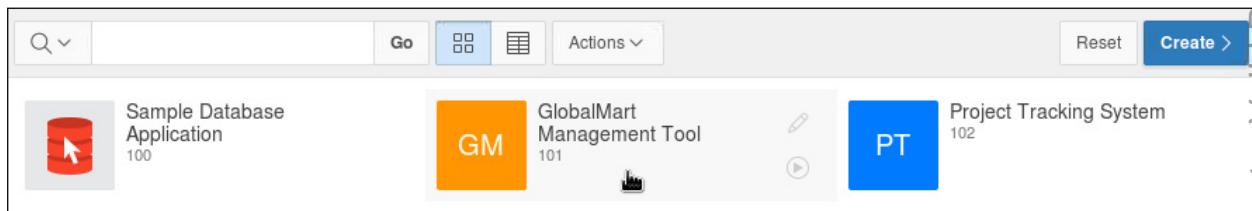
- In the APEX login page, enter workspace name, username, and password, and click **Sign In**.



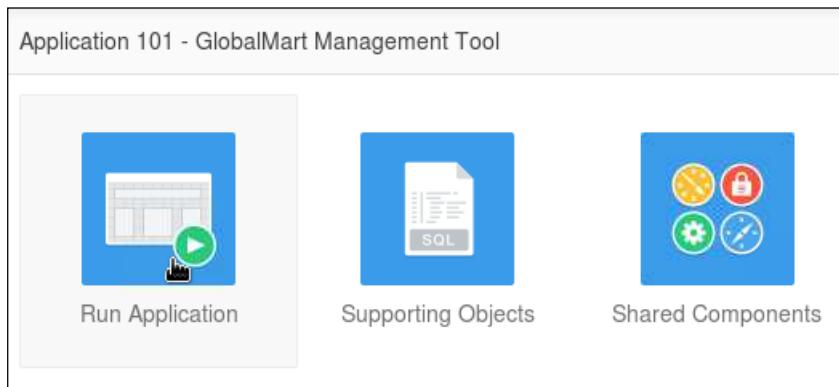
- c. From the Application Builder home page, run the GMT application.
- From the APEX workspace homepage, click **Application Builder**.



- Click the **GlobalMart Management Tool** application.



- Click the **Run Application** icon.



- Enter username/password and click **Log In**.

A screenshot of a Log In dialog box. It has two input fields: "Username" containing "apex" and "Password" containing "****". Below the password field is a "Log In" button with a cursor pointing at it.

5. The GMT homepage is displayed.

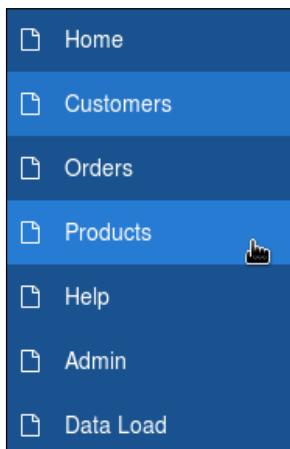
- d. Perform the following actions in the GMT application.

- View the list of customers.
 - View the list of products.
 - View details of any one of the products.
 - View the list of orders.
1. To view the list of customers, click the **Manage Customers** drop-down menu, and select **View Customers**.

2. A report showing all the customers is displayed.

	City	Name	Account mgr	Cust Email
	Bangalore	Scott Jordan	Bates	SCOTT.JORDAN@WILLET.COM
	Chennai	Shammi Pacino	Cambrault	Shammi.Pacino@BITTERN.COM
	Cochin	Sharmila Kazan	Cambrault	Sharmila.Kazan@BRANT.COM
	Cochin	Sharmila Fonda	Cambrault	Sharmila.Fonda@BUFFLEHEAD.COM

3. To view the list of products, click **Products** from the left side menu.



4. A report showing all the products is displayed.

Products			
Products List			
View Details	Product Name	List Price	Discounted Price
	LCD Monitor 11/PM	259	208
	Chemicals - RCP	80	66
	PS 220V /UK	89	76
	Cable RS232 10/AM	6	5
	Cable SCSI 10/FW/ADS	8	6
	PS 110V /US	86	70
	SDRAM - 128 MB	299	248
	TD 12GB/DAT	134	111

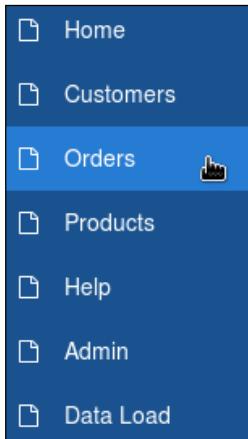
5. Click the View Details icon next to any one of the products.



6. The selected product's details are displayed.

Product Details	
Product Name	Cable SCSI 10/FW/ADS
Product Description	10ft SCSI2 F/W Adapt to DSxx0 Cable
Supplier Id	102095
Product Status	orderable
List Price	8
Discounted Price	6
Catalog Url	http://www.supp-102095.com/cat/hw/p1737.html
Back to Product List	

7. To view the list of orders, click **Orders** from the left side menu.



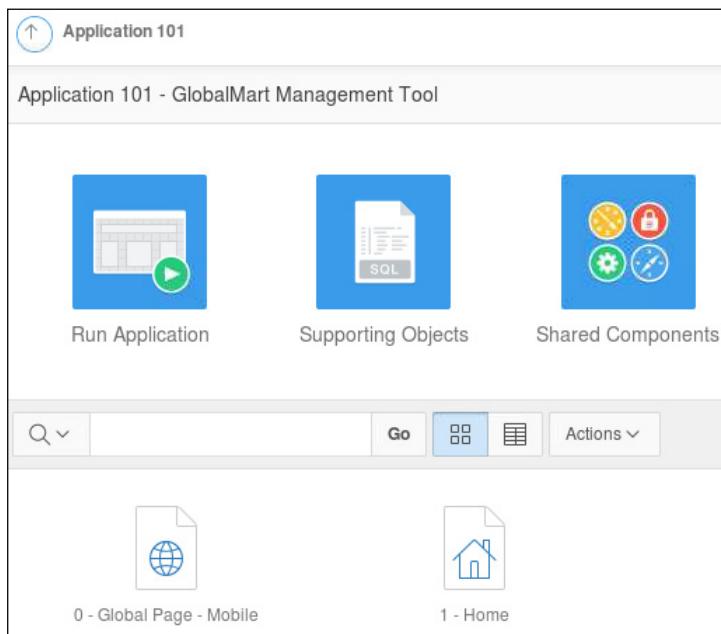
8. A report showing all the orders is displayed. Close the GlobalMart Management Tool application.

List of Orders						
List of Orders						
Order Id	Order Date	Order Mode	Customer Id	Order Status	Order Total	Sales Rep
1	09-APR-15 02.34.12.000000 PM	direct	101	0	78279.6	153
2	13-JUL-15 03.41.54.000000 PM	direct	102	1	42283.2	154
3	26-MAY-15 04.49.34.000000 PM	direct	103	1	6653.4	154
4	07-MAR-16 05.18.23.000000 PM	direct	104	0	46257	155

- e. Using the Developer toolbar, navigate to the GMT application home page.
1. Click **Application 101** from the Developer toolbar.



2. The GMT application home page is displayed.



Solution 2-3: Reviewing the Project Tracking System

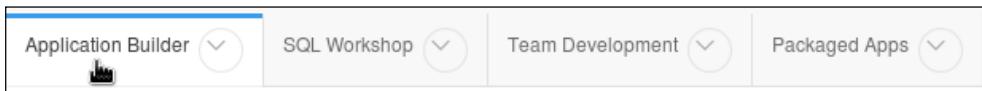
Overview

In this solution, steps to review the database objects and application pages for the Project Tracking System (PTS) application are provided. This application is used in the demonstrations in this course.

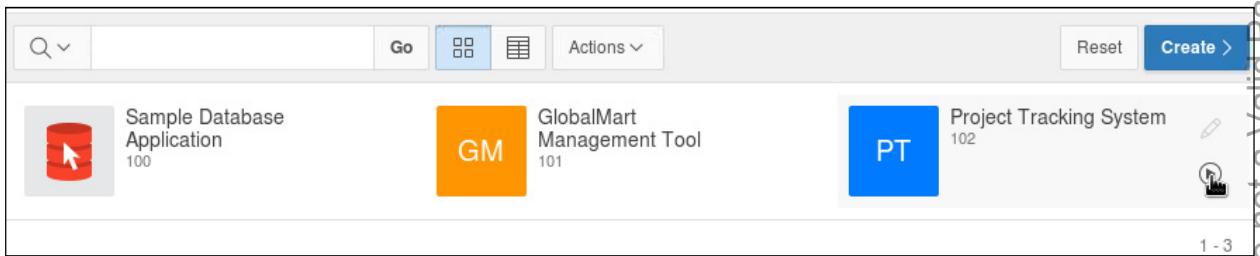
Steps

- From the Application Builder home page, run the PTS application.

- Click **Application Builder** from the top menu bar.



- Click the Run icon displayed in the Project Tracking System application icon.



- Enter username/password and click **Log In**.

A screenshot of the Oracle Application Builder Log In page. It has two input fields: "Username" containing "apex" and "Password" containing "****". Below the password field is a small placeholder icon. At the bottom is a blue "Log In" button with a cursor icon pointing at it. The background is white with some light gray horizontal lines.

4. The PTS application home page is displayed.

The screenshot shows the PTS application's home page. At the top, there is a blue header bar with the text "PROJECT TRACKING SYSTEM". On the right side of the header are links for "Feedback", "Log Out", and "Help". Below the header is a left sidebar with a dark blue background containing several menu items: "Home", "Create Employees", "Modify Employee Details", "Manage Projects", "Manage Project Action Items", "Project Documents", "Project Charts", and "Salary and Service Chart". Under "Home", there is a sub-menu with "Admin" and "Load Multiple Projects". To the right of the sidebar, the main content area has a white background. It displays the text "Last login 10-JUN-15 11.37.44.340882 PM" and "Home" in large bold letters. Below that is a section titled "Browse the Web" with the text "Want to search anything?". It includes a link to "www.bing.com". At the bottom of the main content area are four buttons: "View Projects", "View Status", "View Employees", and "View Action Items". A "Projects Timeline" link is also visible at the bottom.

- b. Review the pages in the PTS application.

1. Click the various links on the home page or left side menu and view the pages displayed. For example, click the View Projects drop-down menu and select Projects Master Report.

The screenshot shows a close-up of the "View Projects" button from the previous screenshot. The button is highlighted with a gray background and contains the text "View Projects ▾". A mouse cursor is hovering over the button. Above the button, there are four other buttons: "View Status", "View Employees", and "View Action Items", each with a small icon and a dropdown arrow. Below the "View Projects" button is a blue rectangular button labeled "Projects Master Report".

2. The Projects Master Report page is displayed. Close the Project Tracking System application.

Project Id	Project Name	Project Type	Project Description	Project Status	Project Planned Start Date	Project Start Date	Project Planned End Date
612	MFG Petrol Industry	304	Engineering Design Capabilities in the Petrol Industry	101	19-JUN-15	19-JUN-15	01-JUL-15

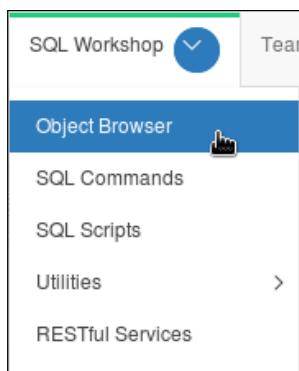
- c. Using the Developer toolbar, navigate to the PTS application home page.

1. Click **Application 102** from the Developer toolbar.

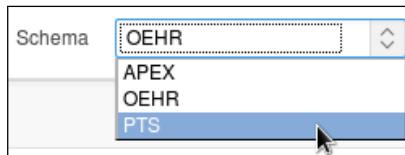


2. The PTS application home page is displayed.

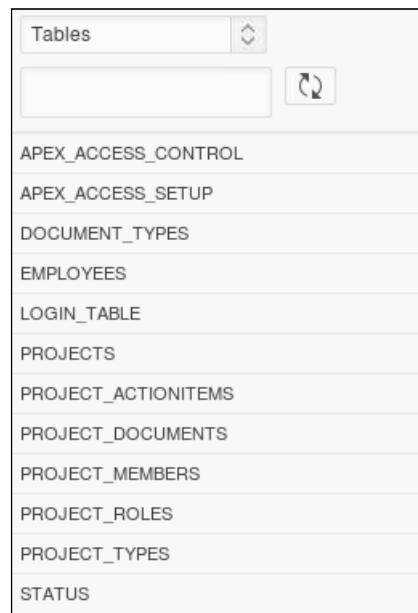
- d. Review the database objects in the PTS schema.
1. Click the **SQL Workshop** drop-down menu and select **Object Browser**.



2. Select **PTS** from the Schema list.



3. The tables in the PTS schema are displayed.



Practices for Lesson 3: Using Oracle APEX Collections

Chapter 3

Practices for Lesson 3: Overview

Practices Overview

In these practices, you use the collections feature of Oracle APEX to add functionality that allows users to place an order for multiple products in the GlobalMart Management Tool.

Practice 3-1: Creating and Updating a Collection

Overview

In this practice, you create a process that will **create a collection** to store products that a customer wants to purchase. You also create the relevant processes to **update the collection** as and when new products are selected for purchase.

Assumptions

None

Tasks

- a. Run the GMT application. Click Products from the navigational menu. The Products page displays a list of all the products available for purchase. Click the View Details icon to display additional information regarding that product. On this product details page, create a button called Add to Cart.
- b. Create the relevant processes to ensure that when the Add to Cart button is clicked, product details (product_id, product_name, min_price, quantity (default value of 1)) should be added to a collection called UserCart.
- c. Run the GMT application and test the pages to ensure that the collection is created and updated properly.

Practice 3-2: Accessing a Collection

Overview

In this practice, you **access a collection** and display its content in a report. You also update wizard pages created to place an order successfully.

Assumptions

You have performed Practice 3-1 or imported the solution application sol_03_01.

Tasks

- a. Create a new blank page called View Cart.
- b. Create a navigational link called View Cart. When this link is clicked, all the products currently added to the cart should be displayed as a report in the View Cart page.
- c. Create a report region called Cart Details on the View Cart page that displays the product details stored in the UserCart collection.
- d. Create a button called Place Order on the View Cart page. When this button is clicked, you should be redirected to a Place Order wizard. Wizard pages to place an order are already created for you. The wizard has the following five pages: Identify Customer, Review Order, Confirm Address, Payment Details, and Place Order.
- e. Ensure that you are on the Identify Customer page in the running GMT application. Select a customer ID and click Next. Edit this page and create the processes required to display the purchase details. Run the application and confirm that the order details are displayed correctly.
- f. Click Next in the Place Order wizard (assuming that you are currently on the Review Order page of the wizard). The code to display the customers' address as mentioned in the database is already completed for you. Click Next. Click Next on the Payment Details page. You are taken to the Place Order page. The code to display the final order details is already completed for you. You should be able to see the products to purchase and the shipping address.
- g. Now, edit the Place Order page so that when the Place Order button is clicked, the order details are stored in the Orders and Order_Items tables. The user should be redirected to the Order Confirmation page that is already created for you.

Solution 3-1: Creating and Updating a Collection

Overview

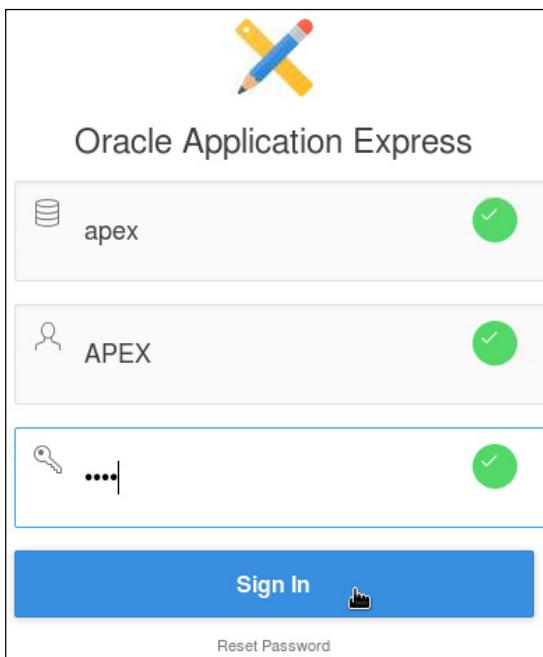
In this solution, the steps to create a process that will **create a collection** to store products that a customer wants to purchase are provided. The steps to create the relevant processes to **update the collection** as and when new products are selected for purchase are also provided.

Assumptions

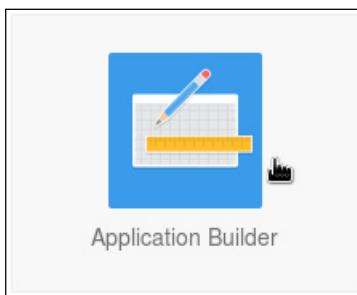
None.

Steps

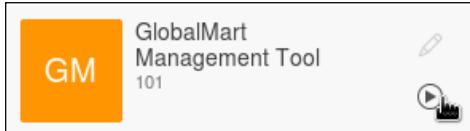
- a. Run the GMT application. Click Products from the navigational menu. The Products page displays a list of all the products available for purchase. Click the View Details icon to display additional information regarding that product. On this product details page, create a button called Add to Cart.
1. Log in to the Oracle APEX workspace, if not already logged in.



2. Click **Application Builder**.



3. Click the Run icon for the GMT application.



4. If prompted, enter the login credentials and click **Log In**.

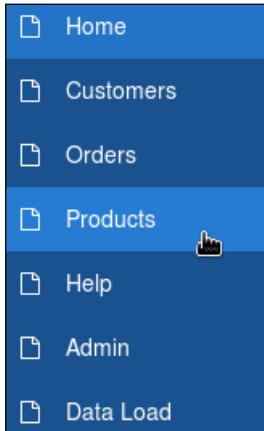
Log In

Username apex

Password ****

Log In

5. Click **Products** from the navigational menu.



6. In the Products List report, click the icon in the View Details column next to any of the products.

View Details	Product Name	List Price	Discounted Price
	LCD Monitor 11/PM	259	208
	Chemicals - RCP	80	66

7. The Product Details page is displayed. You need to create a button on this page. Click **Edit Page 32** from the Developer toolbar.

8. Right-click the **Region Buttons** node in the Product Details region and select **Create Button**.

9. Name the button **AddCart** and give it the Label **Add to Cart**.

10. Select the button position as **Create** and select **Yes** for Hot.

▼ Layout

Sequence 20

Region Product Details

Button Position Create

▼ Appearance

Button Template Text

Hot Yes No

11. Confirm that **Submit Page** is set for Action.

▼ Behavior

Action Submit Page

Execute Validations Yes No

12. Save the page.



13. Run the page.



14. The Add to Cart button should be seen on the page.

Product Details

Product Name	Chemicals - RCP
Product Description	Cleaning Chemicals - 3500 roller clean pads
Supplier Id	103094
Product Status	orderable
List Price	80
Discounted Price	66
Catalog Url	http://www.supp-103094.com/cat/off/p1729.html

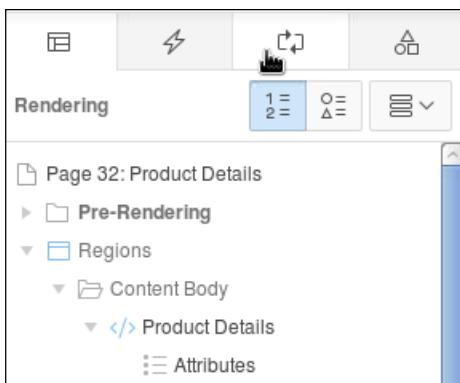
Back to Product List Add to Cart

15. Click the **Edit Page 32** link in the Developer toolbar.

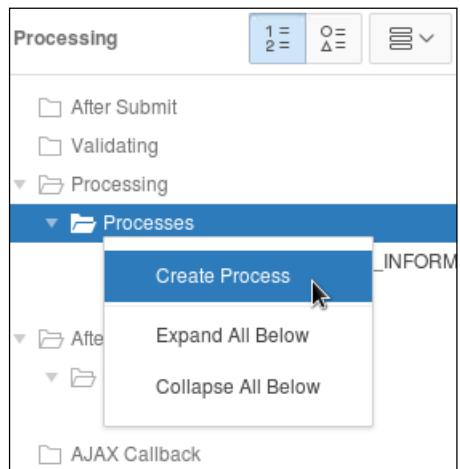


- b. Create the relevant processes to ensure that when the Add to Cart button is clicked, product details (product_id, product_name, min_price, quantity (default value of 1)) should be added to a collection called UserCart.

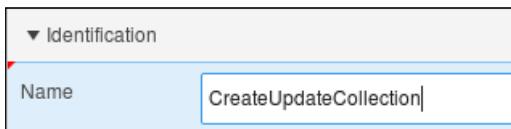
1. Ensure that you are in the Page Designer view for the Product Details page. To create a process, click the Processing icon.



2. Under the Processing node, right-click **Processes**, and select **Create Process**.



3. Name the process **CreateUpdateCollection**.



4. Ensure the Type is **PL/SQL Code** and enter the following code in the PL/SQL Code box. (Code available in lab_03_01.txt file under ~/labs/files.)

```
BEGIN
IF NOT APEX_COLLECTION.COLLECTION_EXISTS('USERCART') THEN
APEX_COLLECTION.CREATE_COLLECTION('USERCART');
END IF;

FOR x IN(SELECT * FROM OEHR.PRODUCT_INFORMATION WHERE PRODUCT_ID
= :P32_PRODUCT_ID) LOOP
APEX_COLLECTION.ADD_MEMBER
(P_COLLECTION_NAME => 'USERCART',
p_c001 => x.product_id,
p_c002 => x.product_name,
p_c003 => x.min_price,
```

```
p_n001 => '1') ;
END LOOP;
END;
```

Type PL/SQL Code

▼ Source

```
PL/SQL Code
```

```
BEGIN
IF NOT APEX_COLLECTION.COLLECTION_EXISTS('USERCART') THEN
APEX_COLLECTION.CREATE_COLLECTION('USERCART');
END IF;

FOR x IN(SELECT * FROM OEHR.PRODUCT_INFORMATION WHERE PRODUCT_ID = :P32_PRODUCT_ID) LOOP
APEX_COLLECTION.ADD_MEMBER
(P_COLLECTION_NAME => 'USERCART',
p_c001 => x.product_id,
p_c002 => x.product_name,
p_c003 => x.min_price,
p_n001 => '1');
END LOOP;
END;
```

5. Set the Sequence to **10** (or first in the list of processes).

▼ Execution Options

Sequence	10
Point	Processing
Tabular Form	- Select -
Run Process	Once Per Page Visit (default)

6. Enter success and error messages.

▼ Success Message

Success Message

```
Product added to cart.
```

▼ Error

Error Message

```
Could not add product to cart. Please try again.
```

7. Select **AddCart** for When Button Pressed under Condition.

▼ Condition

When Button Pressed	AddCart
------------------------	---------

8. Save the page.

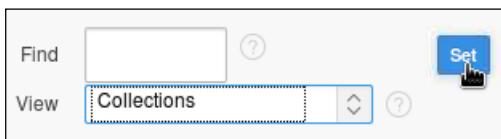
- c. Run the GMT application and test the pages to ensure that the collection is created and updated properly.
- Run the Products page. You can enter the page number (11) in the text field and click Go. Then, click the Run button.



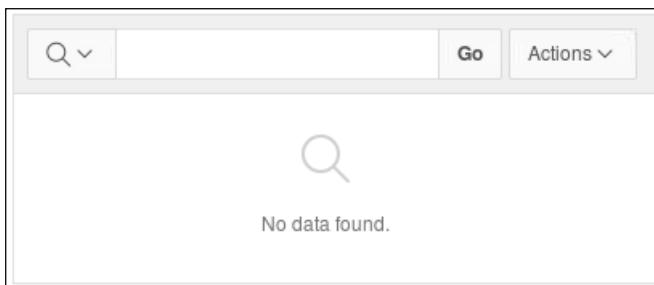
- Click Session from the Developer toolbar.



- Select Collections for View and click Set.



- Note that there are no collections in the session state for the GMT application.



- Switch to the running GMT application. Click the View icon next to any of the products.

View Details	Product Name	List Price	Discounted Price
	LCD Monitor 11/PM	259	208
	Chemicals - RCP	80	66
	PS 220V /UK	89	76
	Cable RS232 10/AM	6	5

6. On the Product Details page, click the **Add to Cart** button.

The screenshot shows a 'Product Details' page for a product named 'PS 220V /UK'. The page includes fields for Product Name, Product Description, Supplier Id, Product Status, List Price, Discounted Price, and Catalog Url. At the bottom are 'Back to Product List' and 'Add to Cart' buttons.

Product Name	PS 220V /UK
Product Description	220V Power supply type - United Kingdom
Supplier Id	102080
Product Status	orderable
List Price	89
Discounted Price	76
Catalog Url	http://www.supp-102080.com/cat/hw/p1733.html

Back to Product List **Add to Cart**

7. Repeat steps 2 and 3 a couple of times by selecting different products. Then, switch to the Session window and click Set.



8. You should be able to see the collection and the data stored in the collection. Close the session window.

The screenshot shows a table grid with columns for Collection Name, Sequence, and various data fields (C001 through N001). The data rows represent four items added to the 'USERCART' collection.

Collection Name	Sequence	C001	C002	C003	C004	C005	C006	C007	C008	C009	C010	N001
USERCART	1	1733	PS 220V /UK	76	-	-	-	-	-	-	-	1
USERCART	2	1734	Cable RS232 10/AM	5	-	-	-	-	-	-	-	1
USERCART	3	1726	LCD Monitor 11/PM	208	-	-	-	-	-	-	-	1
USERCART	4	1739	SDRAM -128 MB	248	-	-	-	-	-	-	-	1

Solution 3-2: Accessing a Collection

Overview

In this solution, the steps to **access a collection** and display its content in a report are given. The steps to update wizard pages created to place an order successfully are also provided.

Assumptions

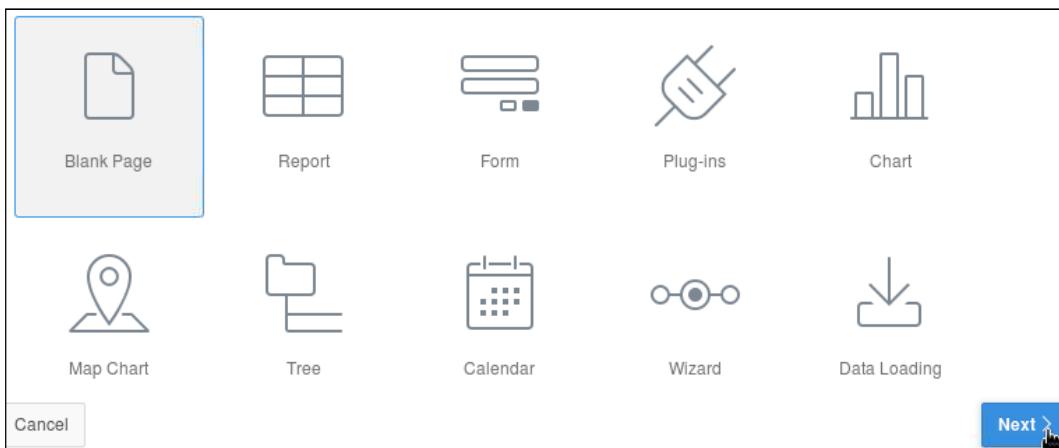
You have performed Practice 3-1 or imported the solution application sol_03_01.

Steps

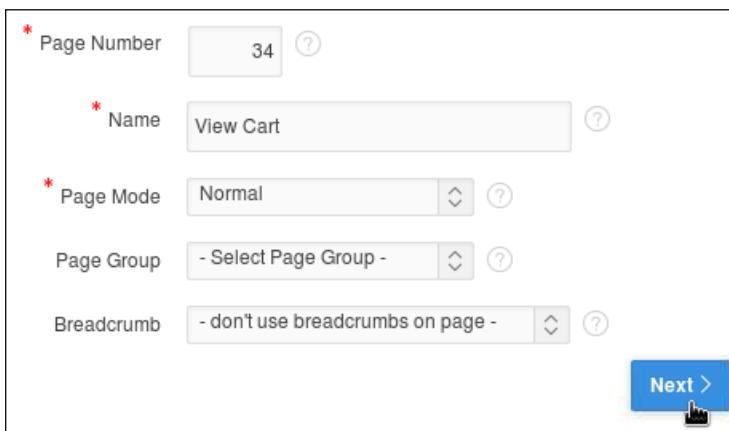
- a. Create a new blank page called View Cart.
 1. Ensure that you are in the GMT application development home page and click **Create Page**.



2. Select **Blank Page** and click **Next**.



3. Enter **View Cart** for Name and click **Next**.



4. Click Next.

Navigation Preference Do not associate this page with a navigation menu entry [?](#)
 Create a new navigation menu entry
 Identify an existing navigation menu entry for this page

Next >

5. Click Finish.

Application	101
Page	34
Page Name	View Cart
Page Title	View Cart

< Cancel **Finish**

- b. Create a navigational link called View Cart. When this link is clicked, all the products currently added to the cart should be displayed as a report on the View Cart page.

1. Navigate to the Shared Components page.



2. Under Navigation, click **Navigation Bar List**.

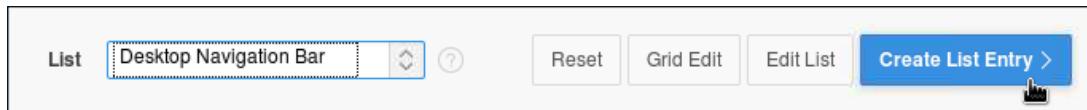
Navigation

- Lists
- Navigation Menu
- Tabs
- Breadcrumbs
- Classic Navigation Bar Entries
- Navigation Bar List**

3. Click **Desktop Navigation Bar**.

Name	Type	Entries
Desktop Navigation Bar	Static	5

4. Click **Create List Entry**.



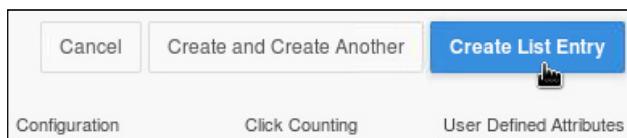
5. Enter **View Cart** for List Entry Label.

 A configuration page for a list entry. The "List" dropdown is set to "Desktop Navigation Bar". The "Parent List Entry" dropdown shows "- No Parent List Item -". The "Sequence" input field contains "20". The "Image/Class" and "Attributes" fields are empty. The "Alt Attribute" field is also empty. The "List Entry Label" field is highlighted with a blue border and contains the value "View Cart".

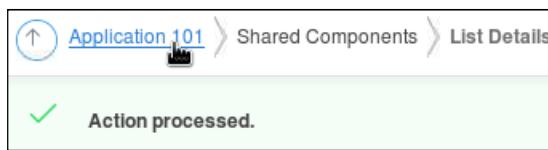
6. Select the View Cart page for Page under Target.

 A dialog box for target configuration. The "Target type" dropdown is set to "Page in this Application". The "Page" input field contains "34". Two checkboxes are present: "reset pagination for this page" (unchecked) and "Printer Friendly" (unchecked).

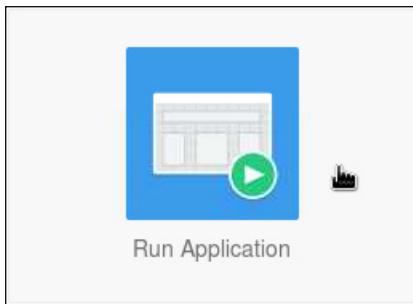
7. Click **Create List Entry**.



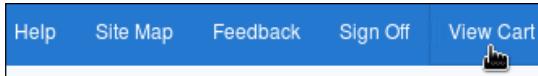
8. Use the breadcrumb to navigate to the application home page.



- Run the application.

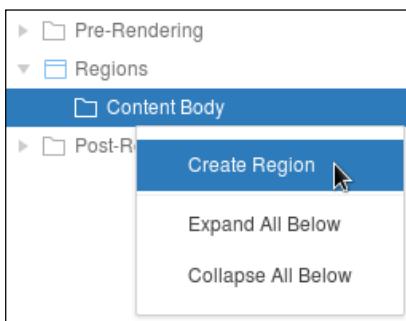


- Click the **View Cart** link from the Navigation Bar.

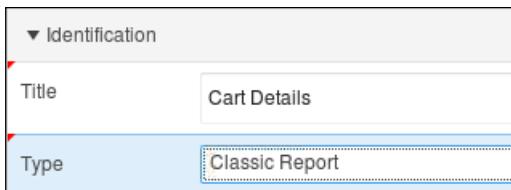


- The View Cart page should be displayed. Here, the page is empty. Use the Developer toolbar to navigate to the Page Designer view.

- Create a report region called **Cart Details** on the View Cart page that displays the product details stored in the **UserCart** collection.
- Ensure that you are in the Page Designer view for the View Cart page. Right-click **Content Body** and click **Create Region**.

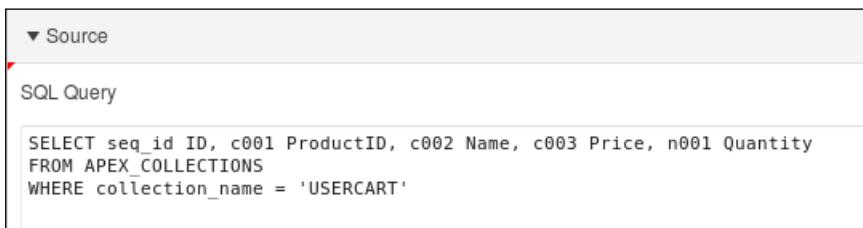


- Enter **Cart Details** for Title and select **Classic Report** for Type.



- Enter the following query in the SQL Query field under Source. (Code also available in `lab_03_02.txt` file under `~/labs/files`.)

```
SELECT seq_id ID, c001 ProductID, c002 Name, c003 Price, n001  
Quantity FROM APEX_collections  
WHERE collection_name = 'USERCART'
```



4. **Save** the page.



5. **Run** the page.



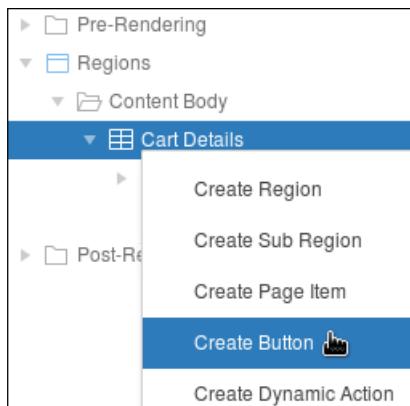
6. You should see the products added to cart.

Cart Details				
Id	Productid	Name	Price	Quantity
1	1733	PS 220V /UK	76	1
2	1734	Cable RS232 10/AM	5	1
3	1726	LCD Monitor 11/PM	208	1
4	1739	SDRAM - 128 MB	248	1

1 - 4

- d. Create a button called Place Order on the View Cart page. When this button is clicked, you should be redirected to a Place Order wizard. Wizard pages to place an order are already created for you. The wizard has the following five pages: Identify Customer, Review Order, Confirm Address, Payment Details, and Place Order.

1. Click Edit Page in the Developer toolbar.
2. Right-click the **Cart Details** region and select **Create Button**.



3. Enter **PlaceOrder** for Button Name and **Place Order** for Label.

Identification	
Button Name	PlaceOrder
Label	Place Order

4. Select **Top of Region** for Button Position.

▼ Layout

Sequence 10

Region Cart Details

Button Position Top of Region

Horizontal Alignment Right

5. Select **Yes** for Hot.

▼ Appearance

Button Template Text

Hot Yes No

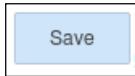
6. Select **Redirect to Page in this Application** for Action and select the **Identify Customer** page for Target.

▼ Behavior

Action Redirect to Page in this Application

Target Page 35

7. Save the page.



8. Run the page.



9. Click the **Place Order** button.

Cart Details

Id	Productid	Name	Price	Quantity
1	1733	PS 220V /UK	76	1
2	1734	Cable RS232 10/AM	5	1
3	1726	LCD Monitor 11/PM	208	1
4	1739	SDRAM - 128 MB	248	1

Place Order

1 - 4

10. You should be taken to the Identify Customer page.

The screenshot shows a modal dialog titled "Identify Customer". Inside, there is a label "Select a Customer ID" next to a dropdown menu containing the value "101". A "Next" button is located in the top right corner of the dialog. At the bottom left is a "Cancel" button, and at the bottom right is a link "release 1.0 Set Screen Reader Mode On".

- e. Ensure that you are on the Identify Customer page in the running GMT application. Select a customer ID and click Next. Edit this page and create the processes required to display the purchase details. Run the application and confirm that the order details are displayed correctly.
1. In the running GMT application, the Identify Customer page is open. Select any one of the customers from the list. (In an ideal application, you use some kind of an authentication scheme to identify the customer.)

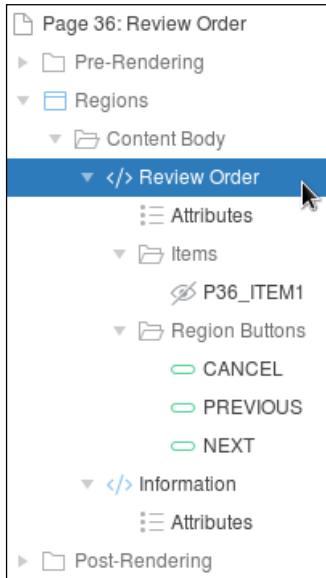
This screenshot is identical to the one above, but the dropdown menu now contains the value "105", indicating a selection change.

2. Click **Next**. The Review Order page is displayed. You need to display the collection details here.

The screenshot shows a modal dialog titled "Review Order". It features a "Previous" button on the left and a "Next" button on the right. A "Cancel" button is positioned at the bottom left. At the bottom right is a link "release 1.0 Set Screen Reader Mode On".

3. Click **Edit Page** in the Developer toolbar.

4. Select the **Review Order** region.



5. Select **Classic Report** for Type.



6. In the Source field, enter the following SQL Query: (code also available in lab_03_02.txt file under ~/labs/files)

```
SELECT seq_id ID, c001 ProductID, c002 Name, c003 Price, n001  
Quantity FROM APEX_collections  
WHERE collection_name = 'USERCART'
```



7. Save and Run the page.



8. The collection details should be displayed.

Review Order					
		Previous		Next	
Id	Productid	Name	Price	Quantity	
1	1733	PS 220V /UK	76	1	
2	1734	Cable RS232 10/AM	5	1	
3	1726	LCD Monitor 11/PM	208	1	
4	1739	SDRAM - 128 MB	248	1	

1 - 4

[Cancel](#)

release 1.0 Set Screen Reader Mode On

- f. Click Next in the Place Order wizard (assuming that you are currently on the Review Order page of the wizard). The code to display the customers' address as mentioned in the database is already completed for you. Click Next. Click Next in the Payment Details page. You are taken to the Place Order page. The code to display the final order details is already completed for you. You should be able to see the products to purchase and the shipping address.

Review Order

Previous Next 

Id ↑	Productid	Name	Price	Quantity
1	1733	PS 220V /UK	76	1
2	1734	Cable RS232 10/AM	5	1
3	1726	LCD Monitor 11/PM	208	1
4	1739	SDRAM - 128 MB	248	1

1 - 4

Cancel

release 1.0 Set Screen Reader Mode On

Confirm Address

Previous Next 

Street address ↑	Postal code	City	State province
4019 W 3Rd St	47404	Bloomington	IN

1 - 1

Cancel

release 1.0 Set Screen Reader Mode On

Payment Details

Previous Next 

Select your payment option **Cash on Delivery**

Cancel

release 1.0 Set Screen Reader Mode On

Place Order

Previous

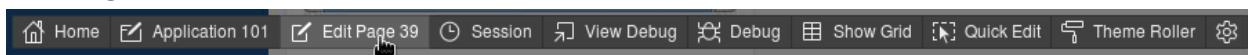
Products Purchased

Id	Name	Price	Quantity
1726	LCD Monitor 11/PM	208	1
1733	PS 220V /UK	76	1
1734	Cable RS232 10/AM	5	1
1739	SDRAM - 128 MB	248	1

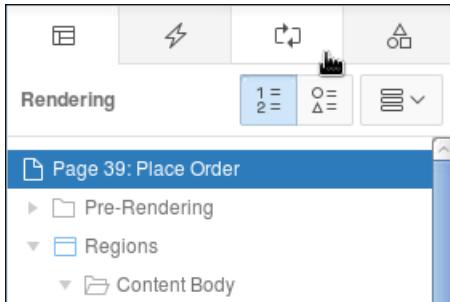
1 - 4

Shipping Address

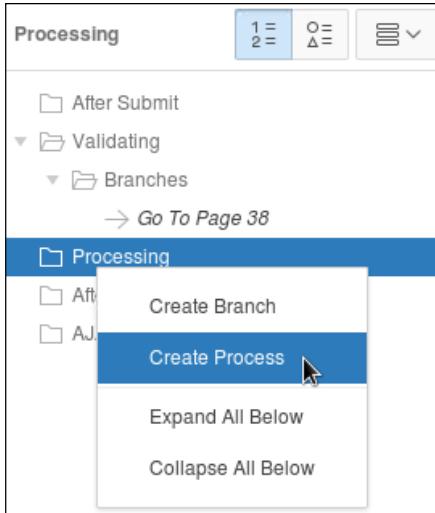
- g. Now, edit the Place Order page so that when the Place Order button is clicked, the order details are stored in the Orders and Order_Items tables. The user should be redirected to the Order Confirmation page that is already created for you.
- In the running GMT application, ensure that you are on the Place Order page. Click **Edit Page** from the Developer toolbar.



2. Click the Processing icon.



3. Right-click the Processing node and select Create Process.



4. Enter **InsertToDatabase** for Name and ensure the Type is **PL/SQL Code**.

The screenshot shows the Identification panel with two fields: 'Name' containing 'InsertToDatabase' and 'Type' containing 'PL/SQL Code'.

5. Enter the following code under Source:

```
declare
total NUMBER;
order_id number;
begin
select sum(c003) into total from apex_collections
where collection_name = 'USERCART';
insert into OEHR.ORDERS(order_date, order_mode, customer_id,
order_total) values
(sysdate, 'online', :P35_CUSTOMER, total);
select order_id into order_id from OEHR.ORDERS
where order_date = sysdate and order_mode = 'online' and
customer_id = :P35_CUSTOMER and order_total = total;
```

```

for c1 in (select c001, c003, n001 from apex_collections where
collection_name = 'USERCART') loop
insert into OEHR.ORDER_ITEMS(order_id, line_item_id, product_id,
unit_price, quantity) values
(order_id,5,c1.c001,c1.c003,c1.n001);
end loop;
end;

```

PL/SQL Code

```

declare
total NUMBER;
order_id number;
begin

select sum(c003) into total from apex_collections
where collection_name = 'USERCART';

insert into OEHR.ORDERS(order_date, order_mode, customer_id, order_total) values
(sysdate, 'online', :P35_CUSTOMER, total);

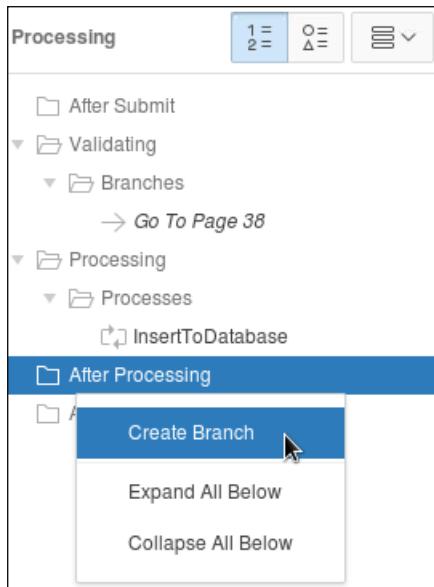
select order_id into order_id from OEHR.ORDERS
where order_date = sysdate and order_mode = 'online' and customer_id = :P35_CUSTOMER and
order_total = total;

for c1 in (select c001, c003, n001 from apex_collections where collection_name =
'USERCART') loop
insert into OEHR.ORDER_ITEMS(order_id, line_item_id, product_id, unit_price, quantity)
values
(order_id,5,c1.c001,c1.c003,c1.n001);

end loop;
end;

```

6. Right-click After Processing and select Create Branch.



7. For Target, select the Order Confirmation page.

▼ Behavior

Type Page or URL (Redirect)

Target Page 40

8. Save the page.



9. Run the page.



10. Click Place Order.

Shipping Address

Street address	Postal code	City	State province
4019 W 3Rd St	47404	Bloomington	IN

1 - 1

Cancel Place Order

11. You should be taken to the Confirmation page.

Order Confirmation

Your order has been placed successfully.

release 1.0 Set Screen Reader Mode On

Practices for Lesson 4: Creating Dynamic Actions

Chapter 4

Practice 4-1: Creating Advanced Dynamic Actions

Overview

In this practice, you modify the Add to Cart and Place Order functionalities created in the GlobalMart Management Tool. You use dynamic actions to deliver a better user experience.

Assumptions

You have completed Practice 3 or imported the solution application sol_03_02.

Tasks

- a. Review the View Cart page. This page displays a report of all the products added by a user to their cart. Create a new column in this report called Delete and display a red cross icon image. When this icon is clicked, a dynamic action should be fired that deletes this row from the UserCart collection and refreshes the report too. Users should be made to confirm the deletion and a deleted notification should be given.
- b. Modify the Review Order page to allow the user to change the quantity of the product ordered. When the quantity is changed, a dynamic action should be fired that updated the UserCart collection and refreshed the report too.
Hint: You can add two link columns to the report: one with a plus sign icon and another with a minus sign icon. When these links are clicked, the Quantity column should be updated accordingly.
- c. In the Place Order wizard, there is a page that displays the shipping address of the customer. On that page, create a button called Change Address. When this button is clicked, a modal window should be displayed that has a form region, which allows the user to update their address in the database. In the modal window, create a button called Update. The form should only allow the addresses to be updated. When this button is clicked, the database should be updated, the modal window closed, and the address in the wizard refreshed to show the new address.

Solution 4-1: Creating Advanced Dynamic Actions

Overview

In this solution, the steps to modify the Add to Cart and Place Order functionalities, created in Practice 3, using dynamic actions to deliver a better user experience are given.

Assumptions

You have completed Practice 3 or imported the solution application sol_03_02.

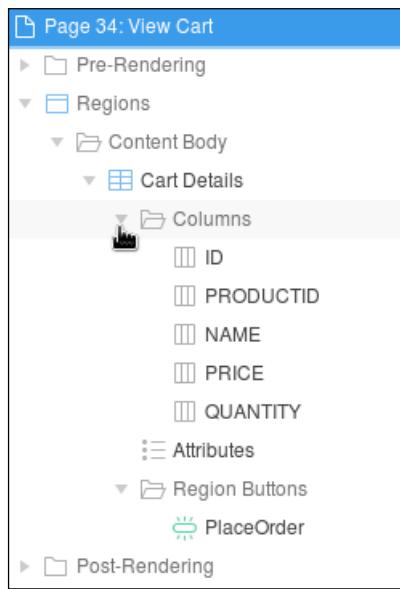
Steps

- a. Review the View Cart page. This page displays a report of all the products added by a user to their cart. Create a new column in this report called Delete and display a red cross icon image. When this icon is clicked, a dynamic action should be fired that deletes this row from the UserCart collection and refreshes the report too. Users should be made to confirm the deletion and a deleted notification should be given.
1. Run the GMT application. Click View Cart. If this page is empty, add a couple of products to cart, and click View Cart again. You need to edit this page to allow products to be deleted from the list.

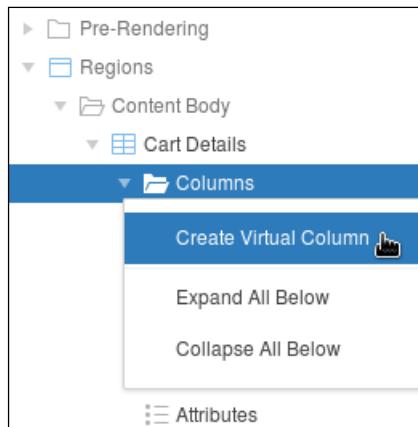
Cart Details				
				Place Order
Id	Productid	Name	Price	Quantity
1	1733	PS 220V /UK	76	1
2	1734	Cable RS232 10/AM	5	1
3	1726	LCD Monitor 11/PM	208	1
4	1739	SDRAM - 128 MB	248	1

2. Click Edit Page on the Developer toolbar.

3. Expand the **Columns** node in the Cart Details region.



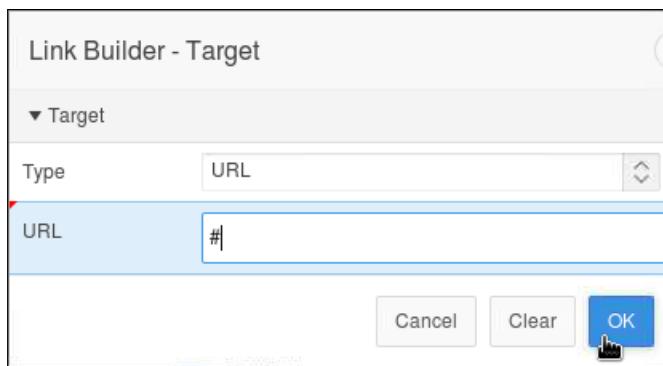
4. You want to add a new column to show the delete red cross icon. Right-click the **Columns** node and select **Create Virtual Column**.



5. Enter **Delete** for Heading.

Column Name	DERIVED\$01
Type	Link
Heading	<input type="text" value="Delete"/>
Alignment	center

6. Click **No Link Defined** under Target, select **URL** for Target, and enter **#** in the URL field. Click **OK**.



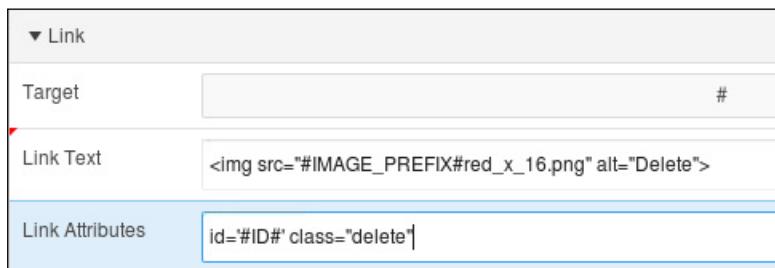
7. Enter the following code in Link Text. The code is also available in the lab_04_01.txt file under ~/labs/files.

```

```

Enter the following in Link Attributes:

```
id="#ID#" class="delete"
```

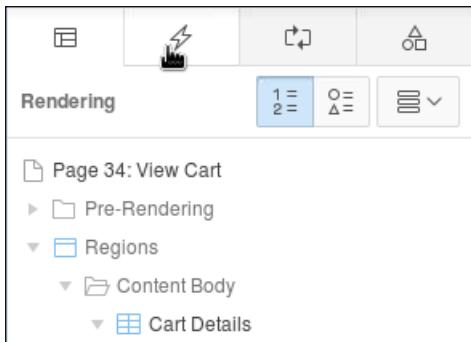


8. Click **Save** and run the page. You should see a new column Delete in the report and a red cross icon displayed next to each product.

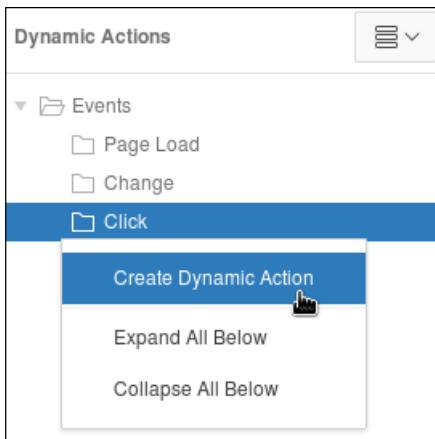
Id	Productid	Name	Price	Quantity	Delete
1	1733	PS 220V /UK	76	1	
2	1734	Cable RS232 10/AM	5	1	
3	1726	LCD Monitor 11/PM	208	1	
4	1739	SDRAM - 128 MB	248	1	

9. Now you will create the dynamic action that should be fired when this red cross icon is clicked. Click **Edit Page**.

10. Click the **Dynamic Actions** tab.



11. Right-click the **Click** node and select **Create Dynamic Action**.



12. Enter **DeleteFromCart** for Name and ensure **Click** is selected for Event.

The screenshot shows the 'Identification' section of the dynamic action creation dialog. The 'Name' field is set to 'DeleteFromCart'. The 'Execution Options' section shows 'Sequence' set to '10'. The 'When' section shows 'Event' set to 'Click'.

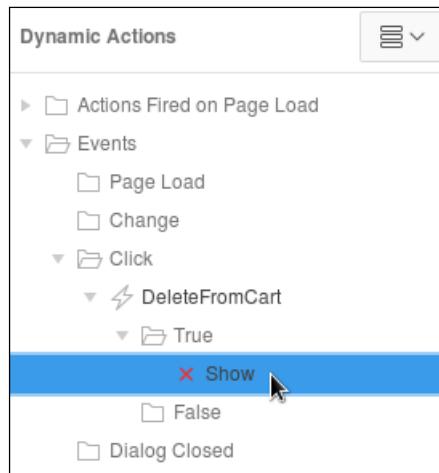
13. Select **jQuery Selector** for Selection Type and enter **a.delete** for jQuerySelector.

The screenshot shows the 'When' section of the dynamic action creation dialog. The 'Event' field is set to 'Click'. The 'Selection Type' field is set to 'jQuery Selector'. The 'jQuery Selector' field contains 'a.delete'.

14. Select **Dynamic** for Event Scope.



15. You will now create the actions that should be performed when a delete icon is clicked. Select the default **Show** action listed under the True node.

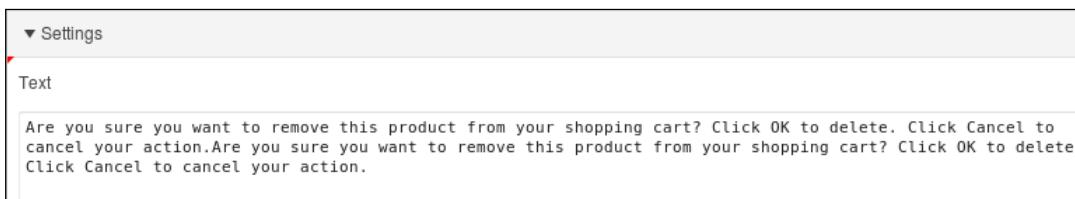


16. Select **Confirm** for Action.



17. Enter the following in the Text area: (The code is also available in the lab_04_01.txt file under ~/labs/files.)

Are you sure you want to remove this product from your shopping cart? Click OK to delete. Click Cancel to cancel your action.



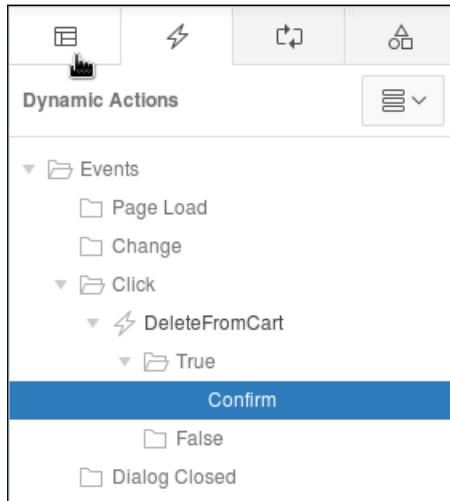
18. **Save** and **run** the page. Click the delete icon for one of the products.

Id	Productid	Name	Price	Quantity	Delete
1	1733	PS 220V /UK	76	1	
2	1734	Cable RS232 10/AM	5	1	
3	1726	LCD Monitor 11/PM	208	1	
4	1739	SDRAM - 128 MB	248	1	

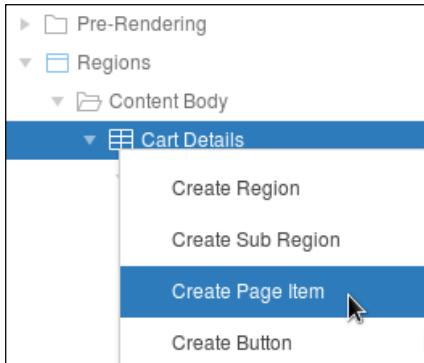
19. You should see the confirmation dialog box. Click **OK**.



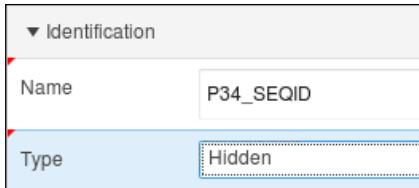
20. You can define other actions to be executed after the **OK** button is clicked. Click **Edit Page**.
21. When the Delete icon is clicked and a user confirms the action, you need to get the seq_id of the row clicked. You can use a javascript expression to get the ID of the triggering element. However, you cannot directly use a java expression in the PL/SQL code to delete a member in the collection. So, you create a hidden item in this page, and create a dynamic action to set the value of this item with the ID of the triggering element. Click the rendering tab.



22. Right-click Cart Details and select Create Page Item.



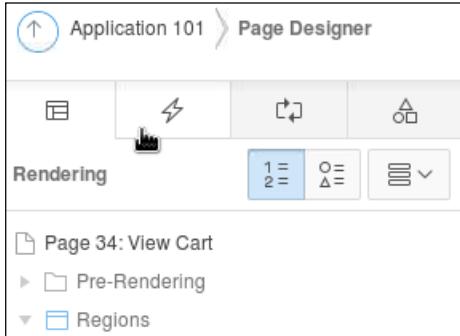
23. In the Name field, replace NEW with SEQID and select Hidden for Type.



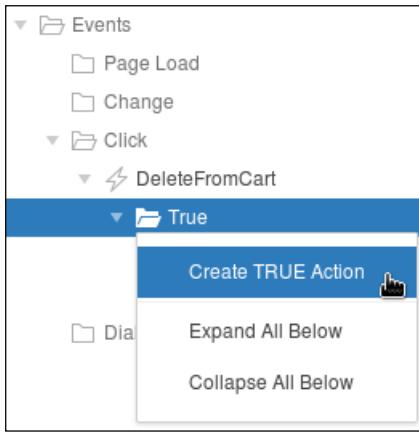
24. Click Save.



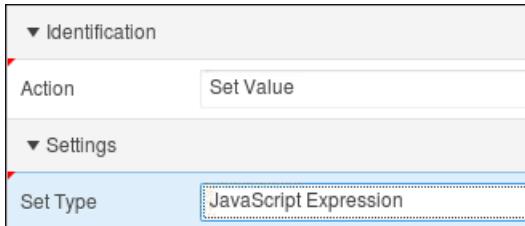
25. Click the Dynamic Actions tab.



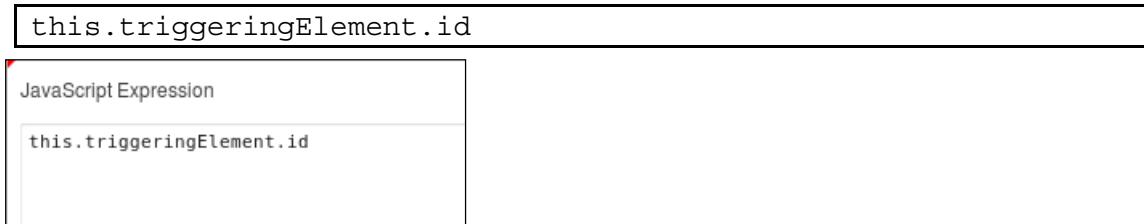
26. Right-click True and select Create TRUE Action.



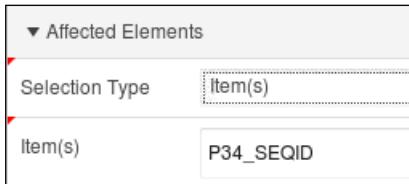
27. Select **Set Value** for Action and **JavaScript Expression** for Set Type.



28. Enter the following code in the JavaScript Expression field. The code is also available in the lab_04_01.txt file under ~/labs/files.



29. Under Affected Elements, select **Item(s)** for Selection Type. Select the **SEQID** item for Item(s).



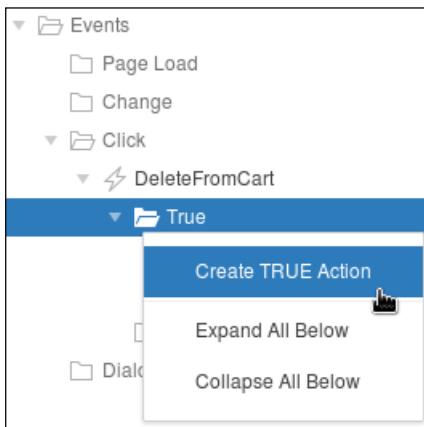
30. Select **No** for Fire On Page Load under Execution Options.



31. Click **Save**.



32. You can now create a dynamic action to execute the PL/SQL code to remove the selected product from the UserCart collection. Right-click **True** and select **Create TRUE Action**.



33. Select **Execute PL/SQL Code** for Action and enter the following code in the PL/SQL Code field. (The code is also available in the lab_04_01.txt file under ~/labs/files). For Page Items to Submit, select the **SEQID** item.

```
APEX_COLLECTION.DELETE_MEMBER(
    p_collection_name => 'USERCART',
    p_seq => :P34_SEQID);
```

Identification

Action: Execute PL/SQL Code

Settings

PL/SQL Code:

```
APEX_COLLECTION.DELETE_MEMBER(
    p_collection_name => 'USERCART',
    p_seq => :P34_SEQID);
```

Page Items to Submit: P34_SEQID

34. Click **Save**.



35. The product will be deleted from the Cart, but the page will still show the old report on the original collection. You want to create a dynamic action to refresh the report so that the deleted product is removed from this list. Right-click **True** and select **Create TRUE Action**.

Events

- Page Load
- Change
- Click
- DeleteFromCart
- True

Create TRUE Action

Expand All Below

Collapse All Below

36. Select **Refresh** for Action, **Region** for Selection Type, and **Cart Details** for Region.

Identification

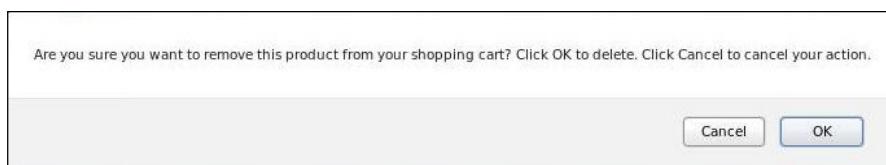
Action: Refresh

Affected Elements

Selection Type: Region

Region: Cart Details

37. Click **Save** and run the page. Click a Delete icon. The confirmation message should be displayed. Click **OK**.

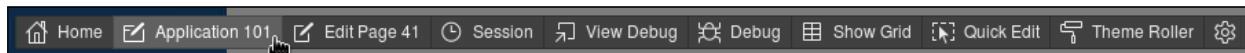


38. The product should be removed from the report.

Id	Productid	Name	Price	Quantity	Delete
1	1733	PS 220V /UK	76	1	
2	1734	Cable RS232 10/AM	5	1	
4	1739	SDRAM - 128 MB	248	1	

1 - 3

39. Click **Application 101** link from the Developer toolbar.



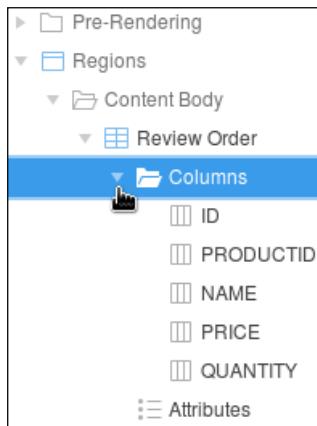
- b. Modify the Review Order page to allow the user to change the quantity of the product ordered. When the quantity is changed, a dynamic action should be fired that updated the UserCart collection and refreshed the report too.

Hint: You can add two link columns to the report: one with a plus sign and another with a minus sign. When these links are clicked, the Quantity column should be updated accordingly,

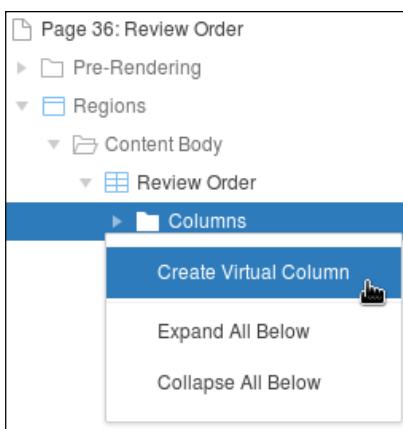
1. In the GMT application development mode, find and click **Review Order** page.



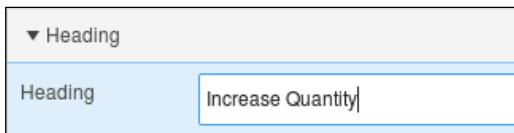
2. Select **Columns** in the Review Order region.



3. You want to add a new column to increase quantity. Right-click **Columns** node and select **Create Virtual Column**.



4. Enter **Increase Quantity** for Heading



5. Select **Center** for Column Alignment.



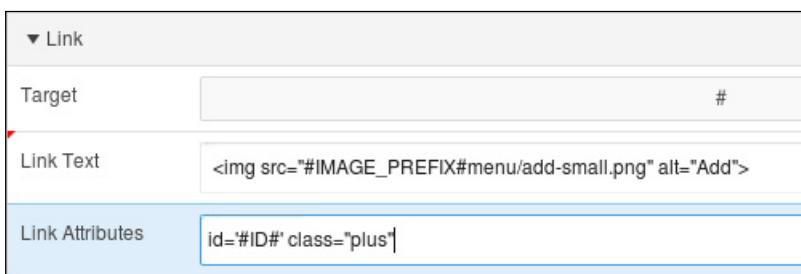
6. Select **URL** for Target and enter # in the URL field. Enter the following in Link Text. The code is also available in the `lab_04_01.txt` file under `~/labs/files`.

```

```

Enter the following in Link Attributes:

```
id="#ID#" class="plus"
```



- Click **Save** and run the page. You should see a new column Increase Quantity in the report and '+' sign image displayed next to each product.
- Note:** If you do not see any products listed on the page, view some products and add them to your cart, and then reach this page by placing an order.

Review Order

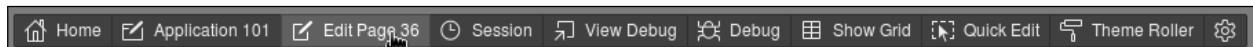
Id	Productid	Name	Price	Quantity	Increase Quantity
1	1729	Chemicals - RCP	66	1	
2	1737	Cable SCSI 10/FW/ADS	6	1	
3	1740	TD 12GB/DAT	111	1	

1 - 3

Cancel

release 1.0 Set Screen Reader Mode On

- Now you create the dynamic action that should be fired when this '+' sign is clicked. Click **Edit Page**.



- When the plus sign is clicked, you need to get the seq_id of the row clicked. So, you need a hidden item on this page. Then, you need to create a dynamic action to set the value of this item with the ID of the triggering element. There will already be a hidden item that you can use. Select that item.

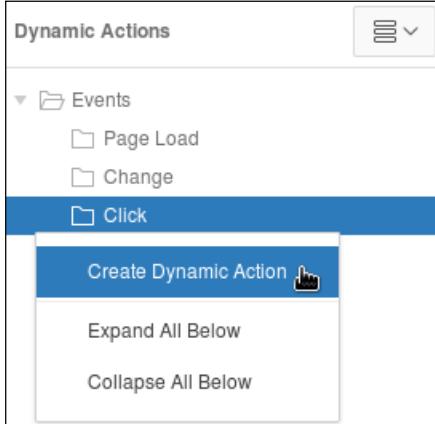
10. In the Name field, replace ITEM1 with **SEQID** and click **Save**.

▼ Identification

Name P36_SEQID

Type Hidden

11. You can now create the dynamic actions. Navigate to the Dynamic Actions tab. Click the **Click** node and select **Create Dynamic Action**.



12. Enter **Increase Quantity** for Name and ensure Click is selected for Event. Select **jQuery Selector** for Selection Type and enter **a.plus** for jQuery Selector. Select **Dynamic** for Event Scope.

▼ Identification

Name Increase Quantity

▼ Execution Options

Sequence 10

▼ When

Event Click

Selection Type jQuery Selector

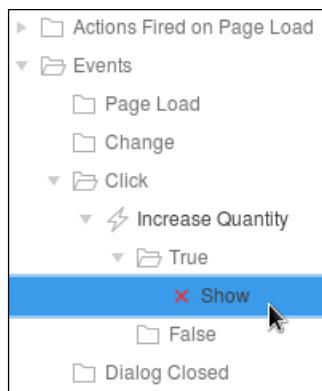
jQuery Selector a.plus

Condition - Select -

▼ Advanced

Event Scope Dynamic

13. You will now create the actions that should be performed when a plus sign is clicked. Select the default **Show** action listed under the True node.



14. Select **Set Value** for Action and **JavaScript Expression** for Set Type. Enter the following code in the JavaScript Expression field. The code is also available in the lab_04_01.txt file under ~/labs/files.

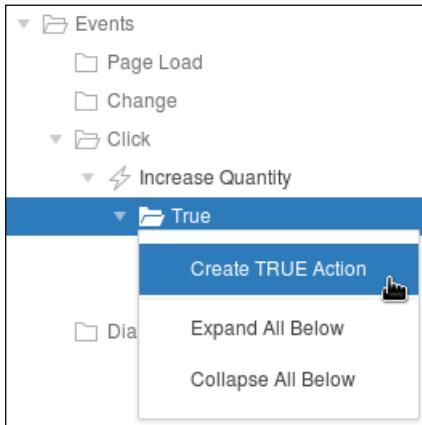
```
this.triggeringElement.id
```

The screenshot shows the 'Identification' tab of the Dynamic Action configuration. The 'Action' dropdown is set to 'Set Value'. The 'Set Type' dropdown is set to 'JavaScript Expression'. The 'JavaScript Expression' input field contains the code 'this.triggeringElement.id'.

15. Under Affected Elements, select **Item(s)** for Selection Type and **SEQID** item for Item(s). Select **No** for Fire On Page Load under Execution Options. Click **Save**.

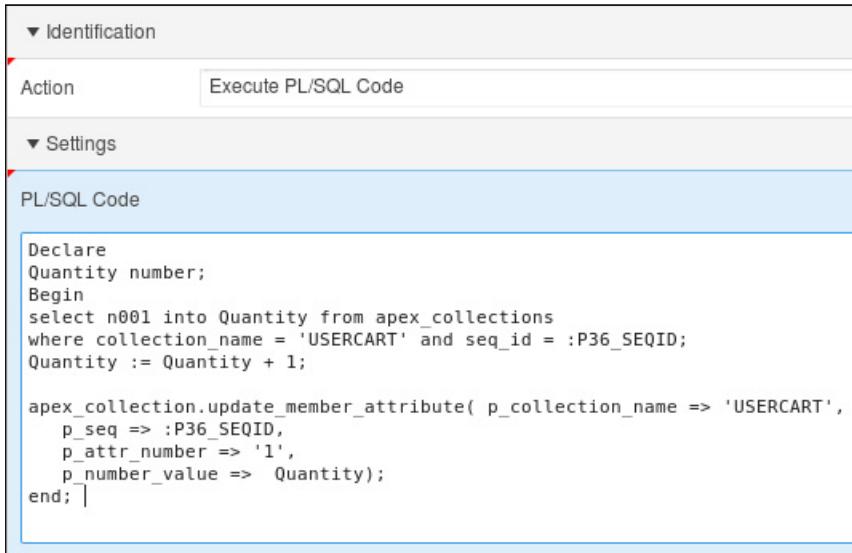
The screenshot shows the 'Affected Elements' tab of the Dynamic Action configuration. The 'Selection Type' dropdown is set to 'Item(s)'. The 'Item(s)' input field contains 'P36_SEQID'. Under the 'Execution Options' section, the 'Fire On Page Load' dropdown has 'Yes' and 'No' options. The 'No' option is selected and highlighted with a blue background.

16. You now create a dynamic action to execute the PL/SQL code to increase the quantity of the selected product in the UserCart collection. Right-click **True** and select **Create TRUE Action**.



17. Select **Execute PL/SQL Code** for Action and enter the following code in the PL/SQL Code field. The code is also available in the lab_04_01.txt file under ~/labs/files.

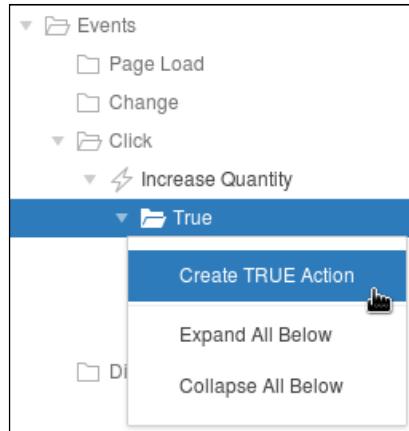
```
Declare
Quantity number;
Begin
select n001 into Quantity from apex_collections
where collection_name = 'USERCART' and seq_id = :P36_SEQID;
Quantity := Quantity + 1;
apex_collection.update_member_attribute( p_collection_name =>
'USERCART',
p_seq => :P36_SEQID,
p_attr_number => '1',
p_number_value =>  Quantity);
end;
```



18. For Page Items to Submit, select the SEQID item. Click **Save**.

Page Items to Submit	P36_SEQID
----------------------	-----------

19. The quantity will be updated in the UserCart, but the page will still show the old report on the original collection. You want to create a dynamic action to refresh the report so that the updated quantity is shown in this list. Right-click **True** and select **Create TRUE Action**.



20. Select **Refresh** for Action, **Region** for Selection Type, and **Review Order** for Region.

Identification	
Action	Refresh
Affected Elements	
Selection Type	Region
Region	Review Order

21. Click **Save** and run the page. Click the plus sign next to any of the products.

Id	Productid	Name	Price	Quantity	Increase Quantity
1	1729	Chemicals - RCP	66	1	
2	1737	Cable SCSI 10/FW/ADS	6	1	
3	1740	TD 12GB/DAT	111	1	

22. You should be able to see the quantity increase for that product.

Id	Productid	Name	Price	Quantity	Increase Quantity
1	1729	Chemicals - RCP	66	2	
2	1737	Cable SCSI 10/FW/ADS	6	1	
3	1740	TD 12GB/DAT	111	1	

23. Repeat steps 3-20 (skip 9 and 10) to create another column in the report that will decrease the quantity. Use the following information:

- Name the column Decrease Quantity and display the minus sign (-) image (step 4 and 6)
- In the Link attributes, name the class ‘minus’ (step 6)
- Use Decrease Quantity for Name and a.minus in jQuery selector (step 12)
- In the PL/SQL code, replace ‘Quantity +1’ with ‘Quantity -1’ (step 17)

Tip: While creating the dynamic action for decreasing the quantity, you can right-click the Increase Quantity dynamic action and select Duplicate to duplicate the Increase Quantity dynamic action. You can now make the required changes.

24. Save and run the page. Click the minus sign next to any of the products.

Id	Productid	Name	Price	Quantity	Increase Quantity	Decrease Quantity
1	1729	Chemicals - RCP	66	2		
2	1737	Cable SCSI 10/FW/ADS	6	1		
3	1740	TD 12GB/DAT	111	1		

25. You should be able to see the Quantity decrease for that product.

Note: You will notice that the quantity can also become a negative number. In a real application, you should handle this in the code so that the user cannot decrease the quantity if it is 1.

Id	Productid	Name	Price	Quantity	Increase Quantity	Decrease Quantity
1	1729	Chemicals - RCP	66	1		
2	1737	Cable SCSI 10/FW/ADS	6	1		
3	1740	TD 12GB/DAT	111	1		

26. Click **Application 101** link from the Developer toolbar.



- c. In the Place Order wizard, there is a page that displays the shipping address of the customer. On that page, create a button called Change Address. When this button is clicked, a modal window should be displayed that has a form region, which allows the user to update their address in the database. In the modal window, create a button called Update. The form should only allow addresses to be updated. When this button is clicked, the database should be updated, the modal window closed, and the address in the wizard should refresh to show the new address.

1. Click **Create Page** in the GMT development application home page.



2. Select **Form** and click **Next**.



3. Select Form on a Table or View and click Next.



4. Select CUSTOMERS (table) and click Next.



5. Enter or select the following and click **Next**:

Page Name: **Update Address**
Page Mode: **Modal Dialog**
Region Title: **Update Address**

* Page Number: 41

* Page Name: Update Address

* Page Mode: Modal Dialog

Page Group: - Select Page Group -

* Region Title: Update Address

Region Template: Standard

Breadcrumb: - do not add breadcrumb region to page -

6. Click **Next**.

Navigation Preference

- Do not associate this page with a navigation menu entry
- Create a new navigation menu entry
- Identify an existing navigation menu entry for this page

7. Select **Select Primary Key Column(s)** for Primary Key Type and click **Next**.

* Primary Key Type: Managed by Database (ROWID) [?](#)
 Select Primary Key Column(s)

* Primary Key Column 1: CUSTOMER_ID (Number) [?](#)

Primary Key Column 2: - Select Primary Key 2 - [?](#)

8. Click **Next**.

Primary Key Column 1: CUSTOMER_ID [?](#)

* Source Type: Existing trigger [?](#)

- Custom PL/SQL function
- Existing sequence

9. Select the Street_Address, Postal_Code, City, State_Province, and Country_ID columns to include, and click Next.

10. Select No for Show Create Button and Show Delete Button, and click Next.

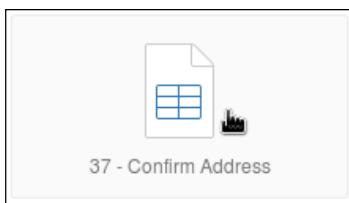
Cancel Button Label	Cancel
Show Create Button	No
Show Save Button	Yes
Save Button Label	Apply Changes
Show Delete Button	No

11. Click **Create** to create the page.

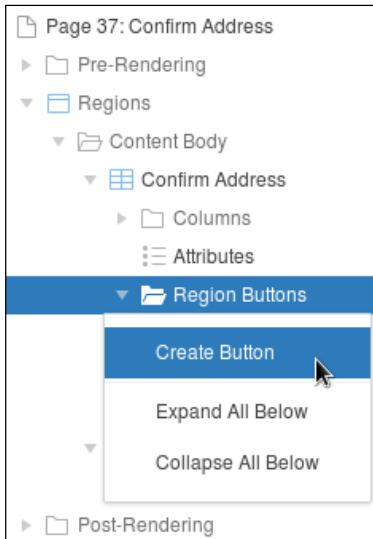
Application	101
Page	41
Page Name	Update Address
Region Title	Update Address
Region Template	Standard
Table / View Owner	OEHR
Table / View Name	CUSTOMERS
Primary Key Column 1	CUSTOMER_ID
Primary Key Column 2	

Create

12. Navigate to the GMT application development home page and click **Confirm Address** page.



13. Right-click Region Buttons and select Create Button.



14. Enter/Select the following in the Button page properties:

Button Name: **CHANGE**

Label: **Change Address**

Button Position: **Create**

Hot: **Yes**

A screenshot of the Oracle Application Express Button page properties dialog. The "Identification" section shows "Button Name" set to "CHANGE" and "Label" set to "Change Address". The "Layout" section shows "Sequence" set to 40 and "Region" set to "Confirm Address". The "Button Position" is set to "Create". The "Appearance" section shows "Button Template" set to "Text". The "Hot" section at the bottom has "Yes" selected, while "No" is unselected.

Identification	
Button Name	CHANGE
Label	Change Address
Layout	
Sequence	40
Region	Confirm Address
Button Position	Create
Appearance	
Button Template	Text
Hot	<input checked="" type="radio"/> Yes <input type="radio"/> No

15. Set the target link and specify the customer ID item from the Identify Customer page.
Action: **Redirect to Page in this Application**
Target: Update Address page

Link Builder - Target

Type Page in this application

Page 41

▼ Set Items

Name	Value
P41_CUSTOMER_ID	&P35_CUSTOMER_ID

16. Click **Save** and run the Confirm Address page. (If no details are displayed, you will need to run the Identify Customer page and reach the Confirm Address page.) Click **Change Address**.

Confirm Address

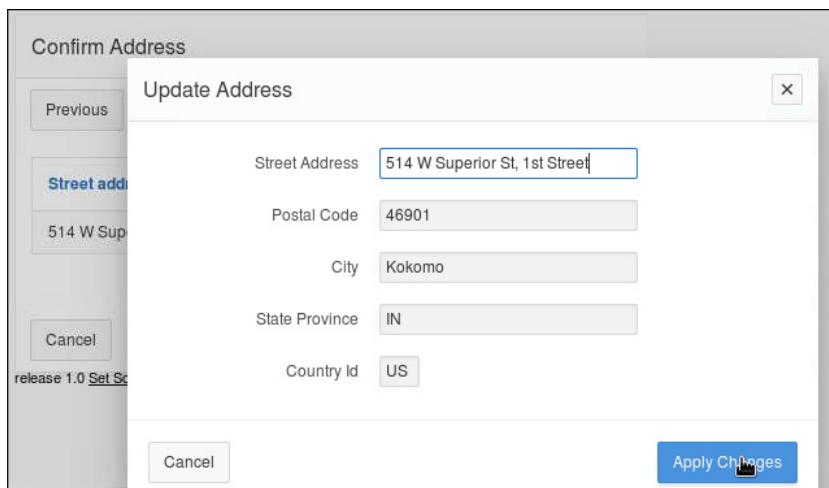
Previous Next

Street address	Postal code	City	State province
514 W Superior St	46901	Kokomo	IN

1 - 1

Cancel Change Address

17. Make a change in the address and click **Apply Changes**. The database is updated and the dialog box is closed.



18. The Confirm Address page shows the old address only. Refresh the page.

Street address	Postal code	City	State province
514 W Superior St	46901	Kokomo	IN

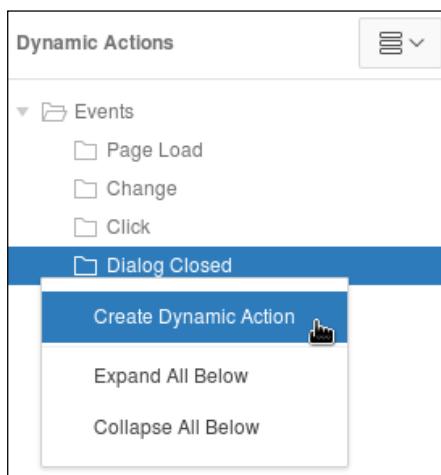
1 - 1

19. Note that the address is updated.

Street address	Postal code	City	State province
514 W Superior St, 1st Street	46901	Kokomo	IN

1 - 1

20. You want to create a dynamic action to update the page automatically when the dialog box is closed. Click **Edit Page**.
21. Click the Dynamic Actions tab. Right-click **Dialog Closed** and select **Create Dynamic Action**.



22. Enter or select the following:

Name: **Refresh Report**

Event: **Dialog Closed**

Selection Type: **Button**

Button: **Change**

The screenshot shows the 'Identification' section of a dynamic action configuration. It includes fields for Name (Refresh Report), Sequence (10), Event (Dialog Closed), Selection Type (Button), and a Button labeled 'CHANGE'.

23. Click the default action created under True and enter or select the following:

Action: **Refresh**

Selection Type: **Region**

Region: **Confirm Address**

The screenshot shows the 'Identification' section of a dynamic action configuration. It includes fields for Action (Refresh), Selection Type (Region), and a Region labeled 'Confirm Address'.

24. Click **Save** and run the page. Click **Change Address**.

The screenshot shows a 'Confirm Address' dialog box. It has a 'Previous' button on the left and a 'Next' button on the right. Below these are four input fields: 'Street address' containing '514 W Superior St, 1st Street', 'Postal code' containing '46901', 'City' containing 'Kokomo', and 'State province' containing 'IN'. At the bottom left is a 'Cancel' button, and at the bottom right is a blue 'Change Address' button with a cursor icon pointing to it. A small note at the bottom left says 'release 1.0 Set Screen Reader Mode On'.

25. Make a change in the address and click **Apply Changes**.

The screenshot shows a 'Confirm Address' dialog box with the title 'Update Address'. On the left, there's a sidebar with buttons for 'Previous', 'Street' (which is selected), '514 W Sup', 'Cancel', and 'release 1.0 Set Sc'. The main area has fields for 'Street Address' (514 W Superior St, 5th Street), 'Postal Code' (46901), 'City' (Kokomo), 'State Province' (IN), and 'Country Id' (US). At the bottom are 'Cancel' and 'Apply Changes' buttons, with 'Apply Changes' being highlighted.

26. Note that the report is now refreshed and the new address listed.

Street address	Postal code	City	State province
514 W Superior St, 5th Street	46901	Kokomo	IN

Practices for Lesson 5: Using Plug-ins in an Application

Chapter 5

Practices for Lesson 5: Overview

Practices Overview

In these practices, you import and use region, dynamic action, and process plug-ins in the GlobalMart Management Tool.

Practice 5-1: Using a Region Plug-in

Overview

In this practice, you import and use a region type plug-in to display a summary of orders on the home page.

Assumptions

You have completed all previous practices or imported the solution application sol_04_01.

Tasks

- a. Import the `region_type_plugin_com_oracle_apex_badge_list` file into the GMT application from the `~/labs/files/plugins` directory.
- b. On the GMT application home page, create a new region called Orders By Mode Summary using the badge list region plug-in. Set the attributes for the region as per your preference.
Note: You can review how this plug-in is used in the Sample Database Application.
- c. Run the page and confirm that the region plug-in is displayed properly.

Practice 5-2: Using a Dynamic Action Plug-in

Overview

In this practice, you import and use dynamic action type plug-ins to display a region using fade-in and fade-out effects.

Assumptions

You have completed all previous practices or imported the solution application sol_05_01.

Tasks

- a. Import the plug-ins from the fadein and fadeout zip files, located in the `~/labs/files/plugins` directory, into the GMT application. Review the instructions in the `readme.txt` files for both the plug-ins.
- b. In the GMT application, edit the Product Details page. Create an item called `View_Image` and set its type as Select List. The Select List should display two values: Yes and No, which return y and n respectively.
- c. Create a new region on the page called `View Product Image` and for the purposes of this lab, enter some static text like "Display Product Image here" in that region.
- d. Using the imported plug-ins, create a dynamic action that shows the `View Product Image` region when the value of `View Image` item is 'Yes' using the fade-in effect. Similarly, create a dynamic action that hides the `View Product Image` region when the value of `View Image` item is 'No' using the fade-out effect.
- e. Run the page and confirm that the dynamic action plug-ins work successfully.

Practice 5-3: Using a Process Plug-in

Overview

In this practice, you import and use a process plug-in to generate a PDF containing all the report regions in a page.

Assumptions

You have completed all previous practices or imported the solution application sol_05_02.

Tasks

- a. Unzip the `process_type_plugin_nl_amis_scheffer_report2pdf.zip` file in the `~/labs/files/plugins` directory. Review the `readme.txt` file. As per the instructions, import and run the `as_pdf2.sql` file.
- b. Import the process plug-in into the GMT application.
- c. In the GMT application, click the Reports page. Run this page. View the various reports displayed on this page. Using the imported plug-in, create a process that will run when the Download PDF button is clicked.
- d. Run the page and confirm that the process plug-in runs successfully and a PDF containing all the reports on the page is displayed.

Solution 5-1: Using a Region Plug-in

Overview

In this solution, the steps to import and use a region type plug-in to display a summary of orders on the home page are provided.

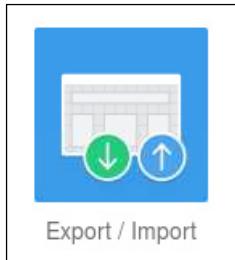
Assumptions

You have completed all previous practices or imported the solution application sol_04_01.

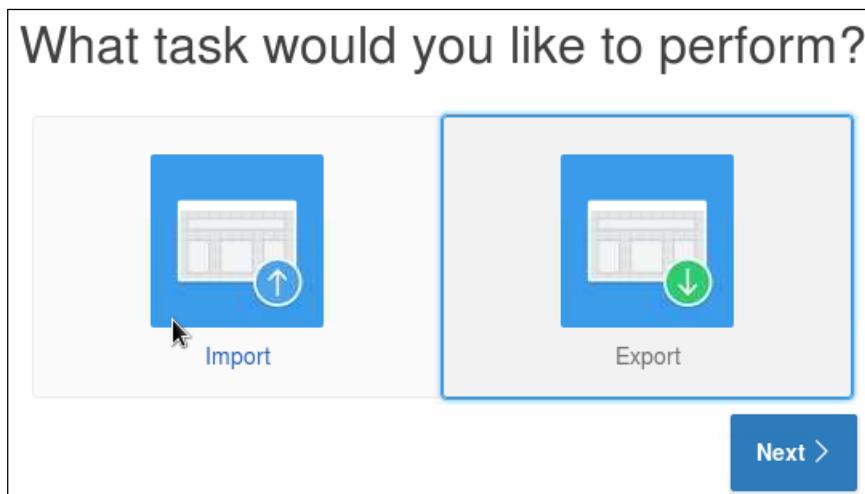
Steps

- a. Import the `region_type_plugin_com_oracle_apex_badge_list` file in to the GMT application from the `~/labs/files/plugins` directory.

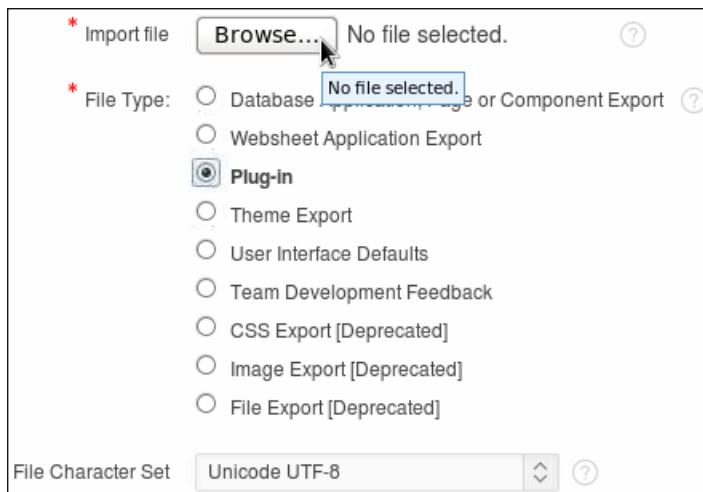
1. From the GMT application development home page, click the **Export / Import** icon.



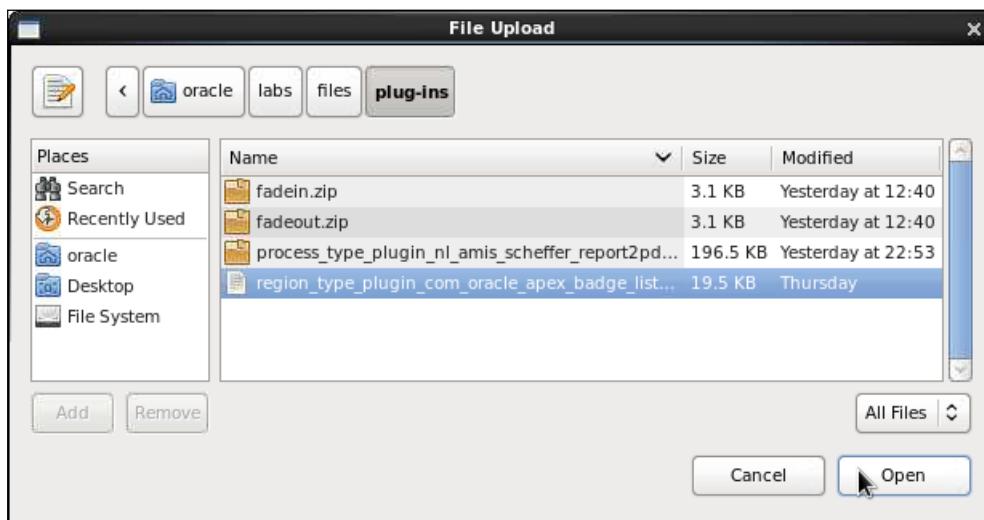
2. Select **Import** and click **Next**.



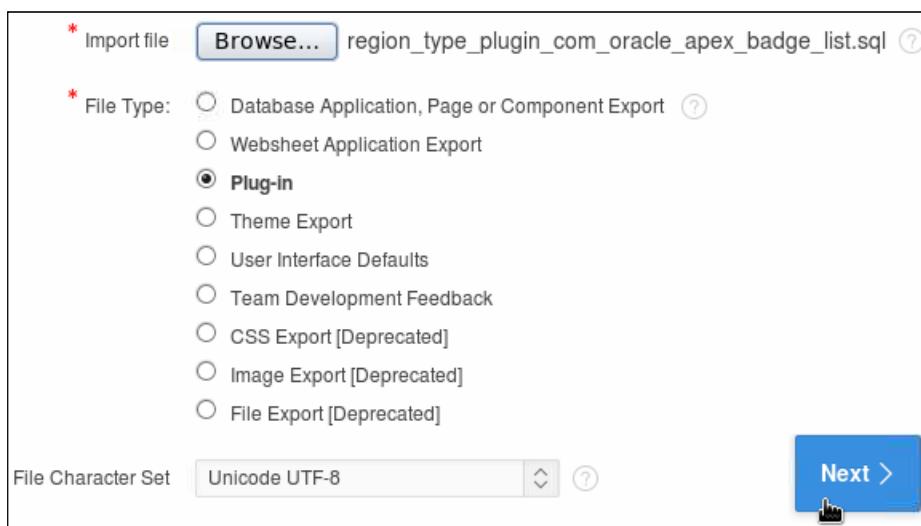
3. Select **Plug-in** for File Type and click **Browse...** .



4. Navigate to ~/labs/files/plugins, select **region_type_plugin_com_oracle_apex_badge_list** and click **Open**.



5. Click **Next**.



6. Click Next.

The export file has been imported successfully.

If you wish to install now, click the **Next** button. You can also install this file at a later time by navigating to the Export Repository.

› Tasks

Next > 

7. Click Install Plug-in.

When you install a plug-in into the current application, the new plug-in will overwrite an existing plug-in having the same plug-in name. If the installation succeeds, the installation of the plug-in becomes permanent. If any errors are encountered, the actions are rolled back, resulting in no permanent changes.

Export File Version:	2013.01.01 
Name:	Badge List 
Internal Name:	COM.ORACLE.APEX.BADGE_LIST 
Install Into Application:	101 GlobalMart Management Tool  
Action:	New plug-in will be created in application 101. 

Install Plug-in 

8. The plug-in is successfully installed in the plug-in repository. Use the breadcrumb link to return to the GMT application home page.

 [Application 101](#)  Shared Components > Plug-ins

 Action processed.

Name	Type	Updated	Version
Badge List	Region		5.0.2
Notification	Dynamic Action		1.3
Star Rating	Item		1.0

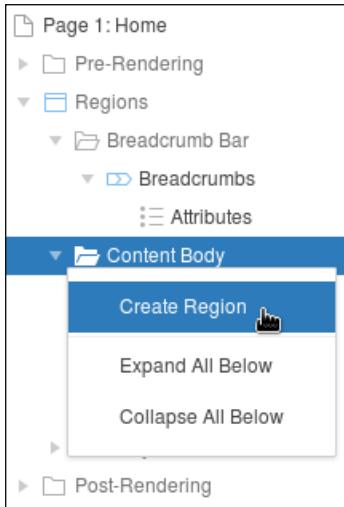
- b. On the GMT application home page, create a new region called Orders By Mode Summary using the badge list region plug-in. Set the attributes for the region as per your preference.

Note: You can review how this plug-in is used in the Sample Database Application.

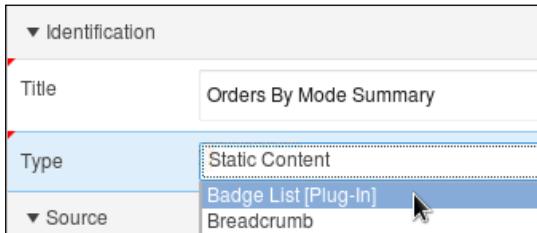
1. From the GMT home page, click the home page icon.



- Right-click **Content Body** node and select **Create Region**.



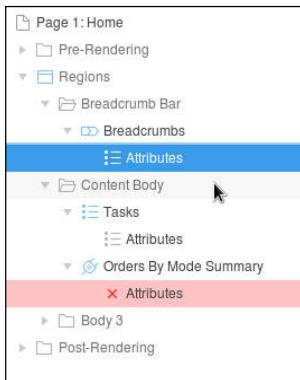
- Enter **Orders By Mode Summary** and select **Badge List [Plug-In]** from the Type select list.



- Enter the following code in the SQL Query field. Code is available in `lab_05_01.txt` file under `~/labs/files`.

```
select label, value from
(select 'Online' label, count(*) value from oehr.orders where
order_mode='online'
union all
select 'Direct' label, count(*) value from oehr.orders where
order_mode='direct')
```

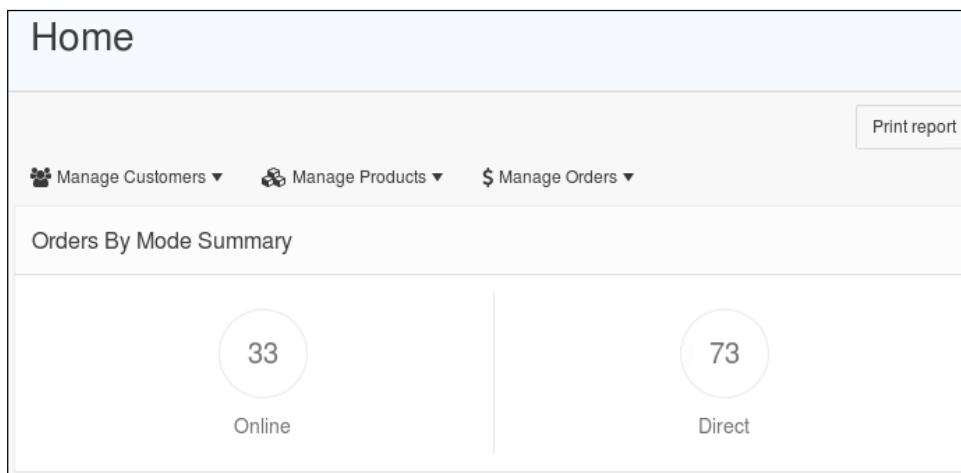
- Click the **Attributes** link.



6. Set the fields as required. Select **LABEL** for the label column and **VALUE** for the value column. For Layout, specify how many columns should be used to display the data. You can choose if you want the region to display the values in circles or rectangles. You can choose to have the region colored or not. **Save** the page.

▼ Settings	
Label Column	LABEL
Value Column	VALUE
Percent Column	- Select -
Link Target	No Link Defined
Layout	2 columns
Chart Type	Circular
Chart Size	Large
Color	No

- c. Run the page and confirm that the region plug-in is displayed properly.
 1. **Run** the page. The region created using the region plug-in is displayed.



2. Return to the application home page using the Developer toolbar.

Solution 5-2: Using a Dynamic Action Plug-in

Overview

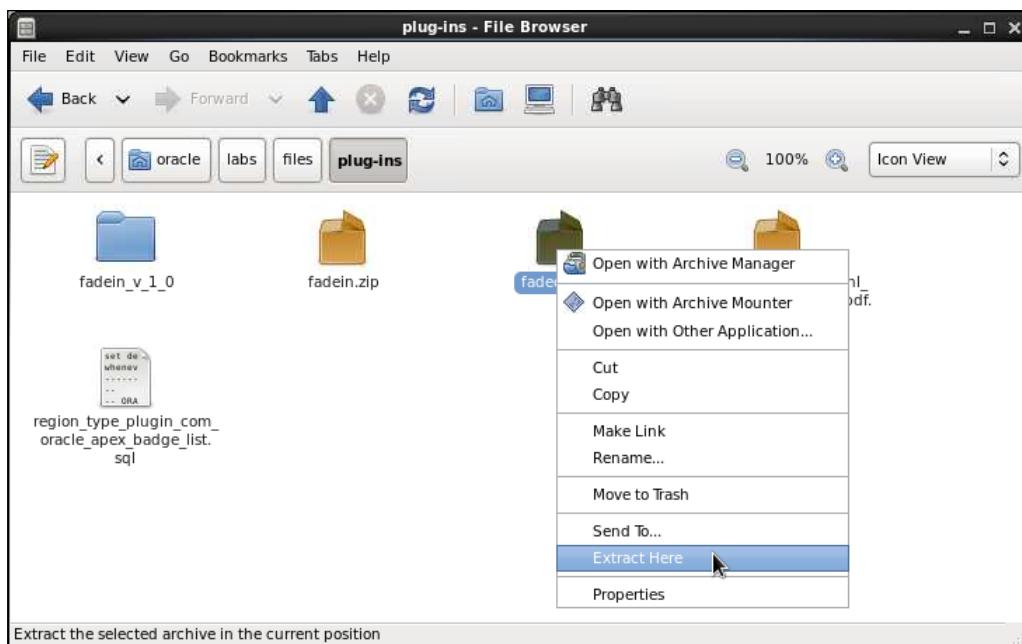
In this solution, steps to import and use dynamic action type plug-ins to display a region using fade-in and fade-out effects are provided.

Assumptions

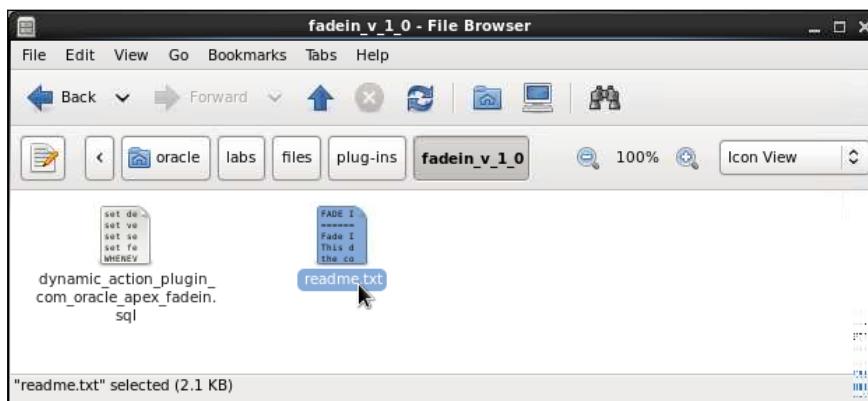
You have completed all previous practices or imported the solution application sol_05_01.

Steps

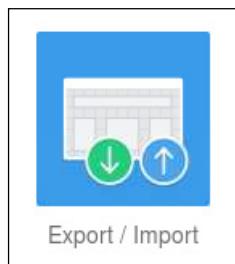
- a. Import the plug-ins from the fadein and fadeout zip files, located in the `~/labs/files/plugins` directory, into the GMT application. Review the instructions in the `readme.txt` files for both the plug-ins.
 - 1 Open a file browser and navigate to the `~/labs/files/plugins` directory. Unzip the fadein and fadeout zip files.



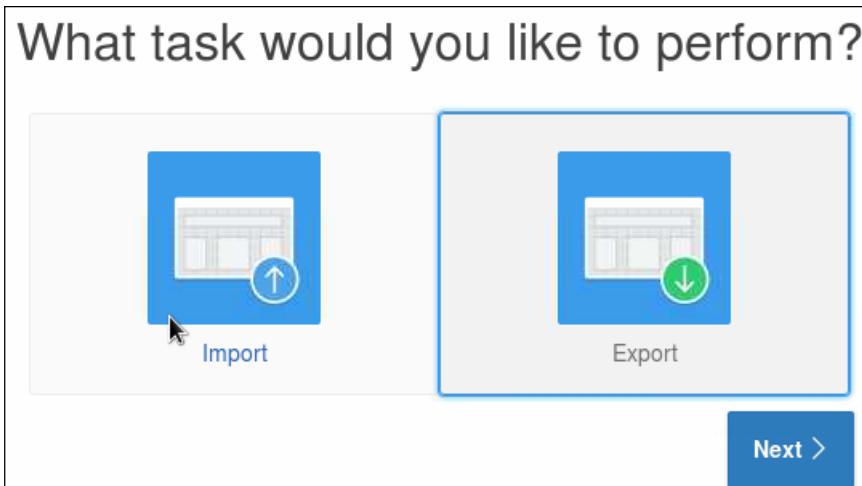
2. Each of the extracted folders contains the plugin and a `readme.txt` file. Open the `readme.txt` files to view more information about each of these plug-ins.



3. Switch to the GMT application home page in APEX and click the **Export/Import** icon.



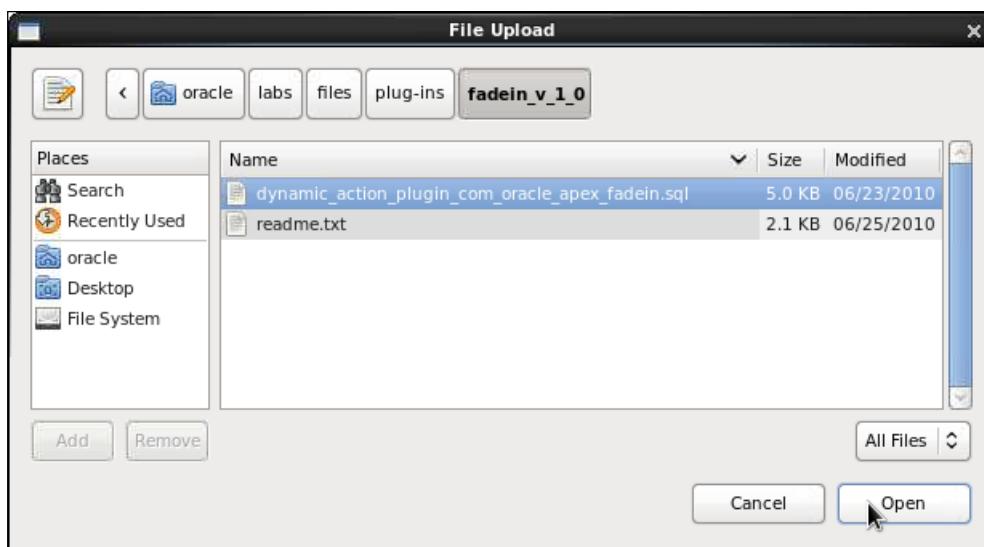
4. Select **Import** and click **Next**.



5. Select **Plug-in** for File Type and click **Browse...** .

A screenshot of the "File Type" configuration dialog. It includes fields for "Import file" (with a "Browse..." button), "File Character Set" (set to "Unicode UTF-8"), and a list of file types. The "File Type" section has a radio button for "Plug-in" which is selected. Other options include "Database Application, Page or Component Export", "Websheet Application Export", "Theme Export", "User Interface Defaults", "Team Development Feedback", "CSS Export [Deprecated]", "Image Export [Deprecated]", and "File Export [Deprecated]".

6. Navigate to `~/labs/files/plugins/fadein_v_1_0`, select `dynamic_action_plugin_com_oracle_apex_fadein`, and click **Open**.



7. Click **Next**.

* Import file dynamic_action_plugin_com_oracle_apex_fadein.sql [?](#)

* File Type: Database Application, Page or Component Export [?](#)
 Websheet Application Export
 Plug-in
 Theme Export
 User Interface Defaults
 Team Development Feedback
 CSS Export [Deprecated]
 Image Export [Deprecated]
 File Export [Deprecated]

File Character Set [?](#)

Next >

8. Click **Next**.

The export file has been imported successfully.

If you wish to install now, click the **Next** button. You can also install this file at a later time by navigating to the Export Repository.

> Tasks

Next >

9. Click **Install Plug-in**.

When you install a plug-in into the current application, the new plug-in will overwrite an existing plug-in having the same plug-in name. If the installation succeeds, the installation of the plug-in becomes permanent. If any errors are encountered, the actions are rolled back, resulting in no permanent changes.

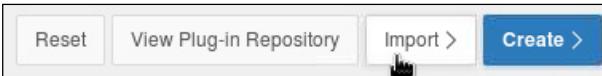
The screenshot shows the 'Install Plug-in' dialog box. It includes fields for 'Export File Version' (2010.05.13), 'Name' (Fade In), 'Internal Name' (COM.ORACLE.APEX.FADEIN), 'Install Into Application' (101 GlobalMart Management Tool), and an 'Action' message stating 'New plug-in will be created in application 101.' A prominent blue 'Install Plug-in' button is at the bottom right.

10. The plug-in is successfully installed in the plug-ins repository.

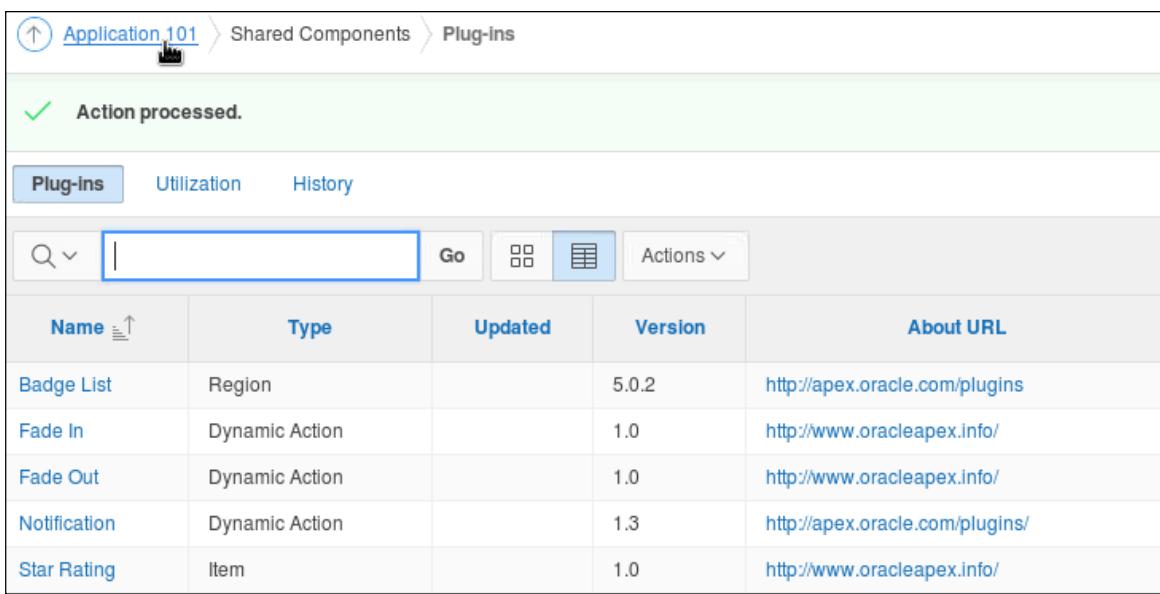
The screenshot shows the 'Plug-ins' page in the Oracle APEX interface. The breadcrumb navigation shows 'Application 101 > Shared Components > Plug-ins'. A green checkmark icon indicates 'Plug-in installed'. The table lists five entries:

Name	Type	Updated	Version	About URL
Badge List	Region		5.0.2	http://apex.oracle.com/plugins
Fade In	Dynamic Action		1.0	http://www.oracleapex.info/
Notification	Dynamic Action		1.3	http://apex.oracle.com/plugins/
Star Rating	Item		1.0	http://www.oracleapex.info/

11. Click **Import** and repeat the steps to import the fade-out plug-in.



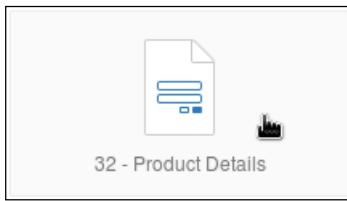
12. The fade-out plug-in will also be listed in the repository. Use the breadcrumb link to return to the GMT application home page.



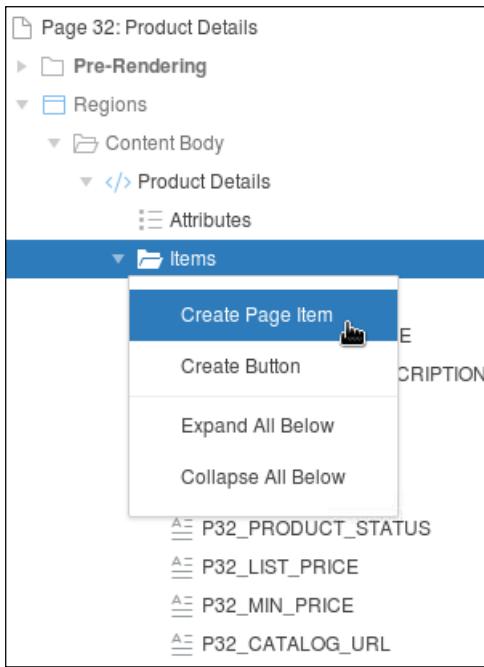
The screenshot shows the Oracle Application 101 Shared Components Plug-ins page. At the top, there is a breadcrumb trail: Application 101 > Shared Components > Plug-ins. A green checkmark icon indicates "Action processed". Below the header, there are tabs for "Plug-ins", "Utilization", and "History", with "Plug-ins" selected. There is a search bar with a magnifying glass icon and a "Go" button. To the right of the search bar are two grid icons: a small grid and a large grid, followed by an "Actions" dropdown menu. The main area is a table listing five plug-ins:

Name	Type	Updated	Version	About URL
Badge List	Region		5.0.2	http://apex.oracle.com/plugins
Fade In	Dynamic Action		1.0	http://www.oracleapex.info/
Fade Out	Dynamic Action		1.0	http://www.oracleapex.info/
Notification	Dynamic Action		1.3	http://apex.oracle.com/plugins/
Star Rating	Item		1.0	http://www.oracleapex.info/

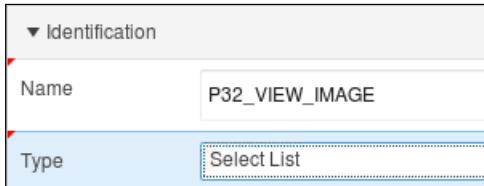
- b. In the GMT application, edit the Product Details page. Create an item called View_Image and set its type as Select List. The Select List should display two values: Yes and No, which return y and n respectively.
- On the GMT application home page, click the **Product Details** page.



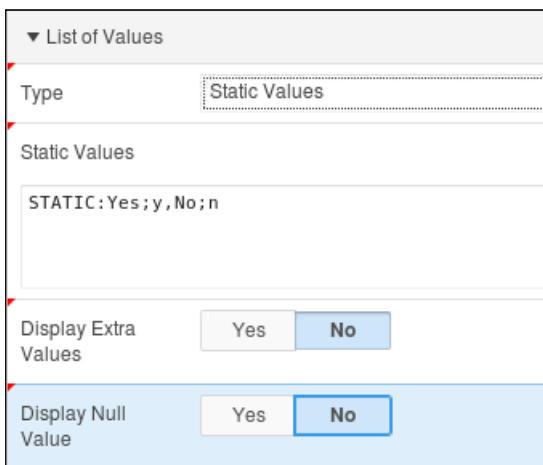
2. Right-click **Items** and select **Create Page Item**.



3. Name the Item **P<n>_VIEW_IMAGE** and select the Type as **Select List**.



4. Under List of Values, select **Static Values** for Type and enter **STATIC:Yes;y,No;n** in the Static Values field. Select **No** for Display Extra Values and Display Null Value.



5. Under Source, select Type as Item and select P<n>_VIEW_IMAGE for Item.

▼ Source	
Type	Item
Item	P32_VIEW_IMAGE
Used	Only when current value in session state is null

6. Select Static Value for Default and enter No in the Static Value field.

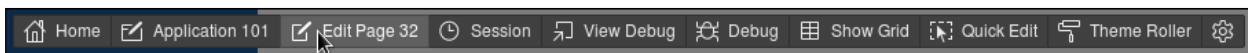
▼ Default	
Type	Static Value
Static Value	No

7. Save and run the page. The created item should be displayed.

Note: If you do not see any values for the rest of the items, it is fine.

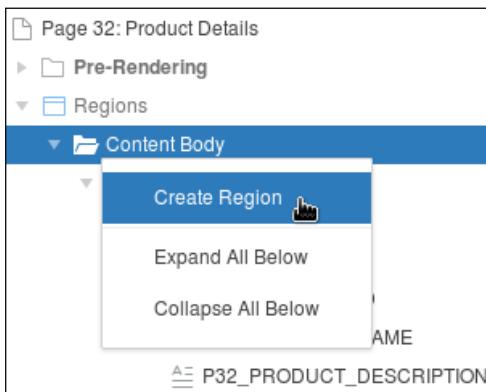
Product Details	
Product Name	
Product Description	
Supplier Id	
Product Status	
List Price	
Discounted Price	
Catalog Url	
View image	No
	No
	Yes

8. Click Edit Page.



- c. Create a new region on the page called View Product Image and for the purposes of this lab, enter some static text like “Display Product Image here” in that region.

1. Right-click **Content Body** and select **Create Region**.



2. Enter **View Product Image** for Name and enter “**Display Product Image here.**” in the Source field.

Identification	
Title	<input type="text" value="View Product Image"/>
Type	<input type="text" value="Static Content"/>
Source	
Text	<input type="text" value="Display Product Image here."/>

3. Select **No** for Start New Row so that this region is displayed next to the existing region.

Start New Row	<input type="button" value="Yes"/>	<input checked="" type="button" value="No"/>
Column	<input type="text" value="Automatic"/>	
New Column	<input checked="" type="button" value="Yes"/>	<input type="button" value="No"/>
Column Span	<input type="text" value="Automatic"/>	

4. **Save** and run the page. The created region should be displayed.

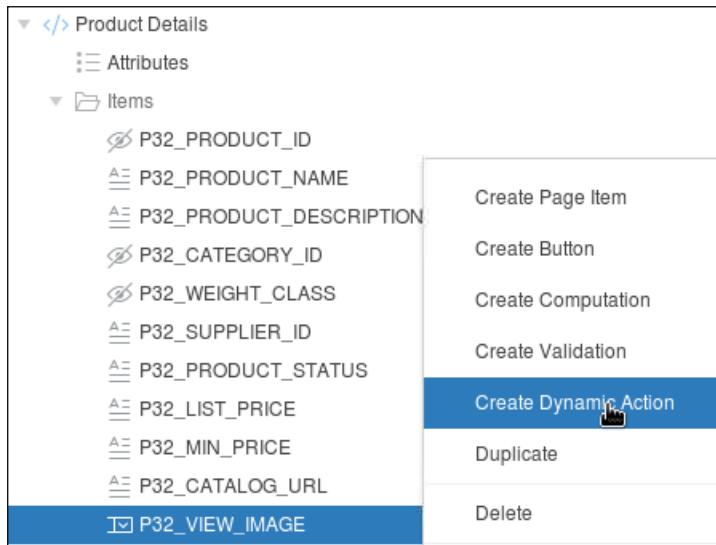
Product Details	View Product Image
Product Name	<input type="text" value="Display Product Image here."/>
Product Description	

5. Click **Edit Page**.



- d. Using the imported plug-ins, create a dynamic action that shows the View Product Image region when the value of View Image item is 'Yes' using the fade-in effect. Similarly, create a dynamic action that hides the View Product Image region when the value of View Image item is 'No' using the fade-out effect.

1. Right-click View Image item and select Create Dynamic Action.



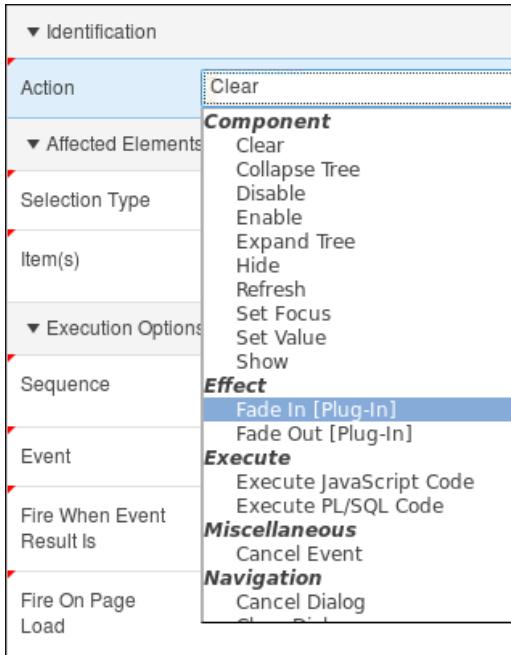
2. Enter **Show Region** for Name. Select **equal to** for Condition and enter **y** for Value.

A configuration dialog for a 'Show Region'. The 'Identification' section has 'Name' set to 'Show Region'. In the 'Execution Options' section, 'Sequence' is set to 10. Under the 'When' section, 'Event' is set to 'Change', 'Selection Type' is 'Item(s)', and 'Item(s)' is set to 'P32_VIEW_IMAGE'. The 'Condition' field is set to 'equal to' and the 'Value' field is set to 'y'.

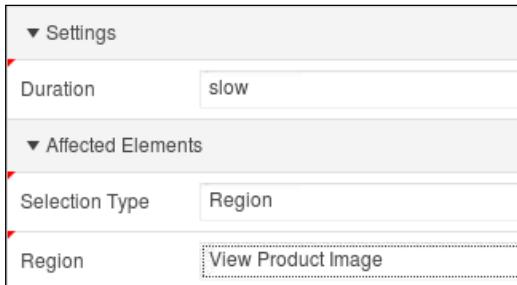
3. Click the **Show** link under True node.



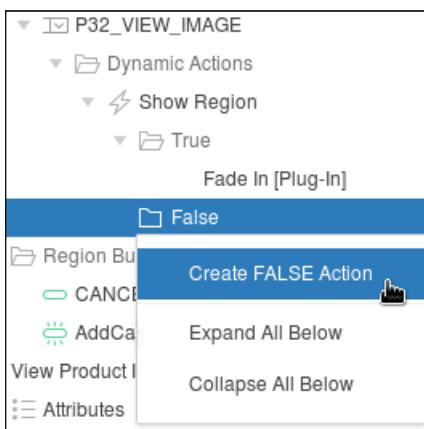
4. Select the **Fade In [Plug-In]** for Action.



5. Under Settings, you can change the Duration if you want. Under Affected Elements, select **Region** for Selection Type and **View Product Image** for Region.



6. Right-click the **False** node and select **Create FALSE Action**.

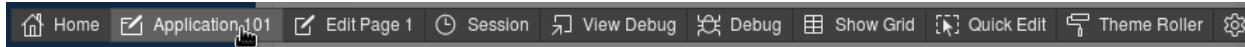


7. Select **Fade Out [Plug-In]** for Action. Under Settings, you can change the Duration if you want. Under Affected Elements, select **Region** for Selection Type and **View Product Image** for Region.

The screenshot shows a configuration dialog with the following settings:

Action	Fade Out [Plug-In]
Duration	slow
Selection Type	Region
Region	View Product Image

- e. Run the page and confirm that the dynamic action plug-ins work successfully.
1. **Save** and run the page. Select **Yes** for View Image. You should be able to see the fade-in effect. Select **No** for View Image. The region should fade out.
Note: On the class machines, the effect might not be very clear.
 2. Return to the application home page using the Developer toolbar.



Solution 5-3: Using a Process Plug-in

Overview

In this solution, the steps to import and use a process plug-in to generate a PDF containing all the report regions in a page are provided.

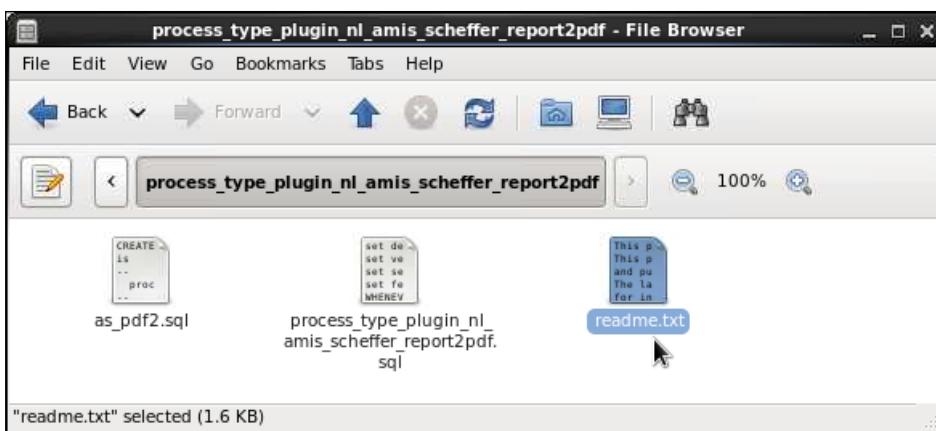
Assumptions

You have completed all previous practices or imported the solution application sol_05_02.

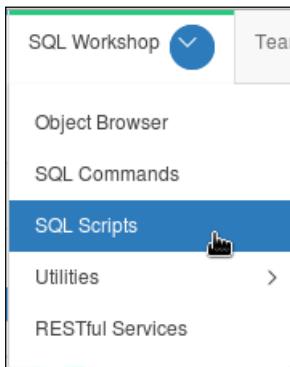
Steps

- a. Unzip the `process_type_plugin_nl_amis_scheffer_report2pdf.zip` file in the `~/labs/files/plugins` directory. Review the `readme.txt` file. As per the instructions, import and run the `as_pdf2.sql` file.

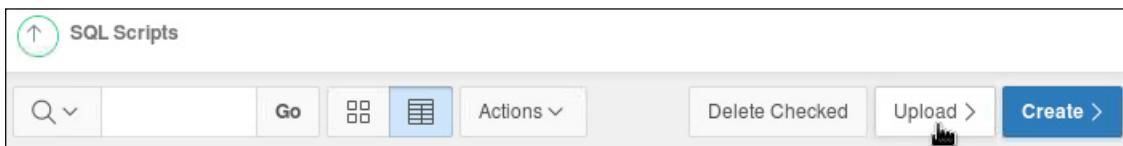
1. Unzip the plug-in zip file. Open and read the `readme.txt` file.



2. In Oracle APEX, click the **SQL Workshop** drop-down menu and select **SQL Scripts**.



3. Click **Upload**.



4. Click **Browse**, select the `as_pdf2.sql` file from the `~/labs/files/plugins/` directory, and click **Upload**.

* File as_pdf2.sql

Script Name

File Character Set

5. Click the Run icon for the uploaded script.

Bytes	Results	Run
89,148	0	

6. Click **Run Now**.

You have requested to run the following script. Please confirm your request.

Script Name	as_pdf2.sql
Created	on 06/03/2015 12:43:29 PM by APEX
Updated	on 06/03/2015 12:43:29 PM by APEX
Number of Statements	3
Script Size in Bytes	89,148

7. You can view the results of the SQL script.

Delete Checked		
Statements	Bytes	View Results
3 of 3	0	

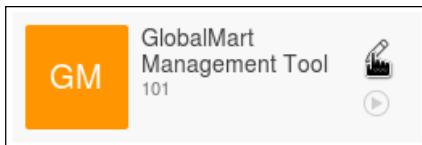
SQL Scripts > Results					
Script: as_pdf2.sql		Status: Complete			
View: <input type="radio"/> Detail <input checked="" type="radio"/> Summary		Rows: 15	Go		
Number	Elapsed	Statement		Feedback	Rows
1	0.20	CREATE OR REPLACE package as_pdf2 is -- procedure init;--		Package created.	0
2	0.35	CREATE OR REPLACE package body as_pdf2 is -- type tp_pls_t		Package Body created.	0
3	0.08	begin if dbms_xdb.gethttpport = 0 then execute immed		Statement processed.	1

[Download](#)

8. Click the Application Builder tab.

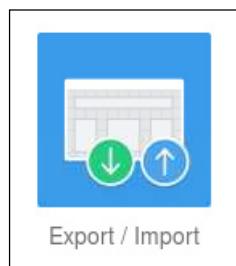


9. Click the Edit icon for the GMT application.



- b. Import the process plug-in into the GMT application.

1. From the GMT home page, click the Export/Import icon.



2. Follow the import wizard and import the plug-in. Once successfully imported, it will be shown in the plug-ins repository. Use the breadcrumb link to return to the GMT application home page.

The screenshot shows the 'Plug-ins' tab selected in the navigation bar. A success message 'Plug-in installed.' is displayed above the table. The table lists six imported plug-ins:

Name	Type	Updated	Version	About URL
Badge List	Region		5.0.2	http://apex.oracle.com/plugins
Fade In	Dynamic Action		1.0	http://www.oracleapex.info/
Fade Out	Dynamic Action		1.0	http://www.oracleapex.info/
Notification	Dynamic Action		1.3	http://apex.oracle.com/plugins/
Reports2PDF	Process		0.802	
Star Rating	Item		1.0	http://www.oracleapex.info/

- c. In the GMT application, click the Reports page. Run this page. View the various reports displayed on this page. Using the imported plug-in, create a process that will run when the Download PDF button is clicked.
1. From the GMT application home page, click the **Reports** page.



- Run the page. You will see three reports and one button on this page. You will now add a process to make the button functional. Click the Edit Page link on the Developer toolbar.

Customer Reports

[Download Reports as PDF](#)

High Credit Limit Customers			Total Orders per Customer			Total Purchases Per Customer		
Name	Credit limit	City	Customer	Orders	Total	Customer	Purchases	Orders
Ally Brando	5000	Bern	Sivaji Landis	5	160284.6	Markus Rampling	403119.7	5
Ally Fawcett	5000	Zurich	Mammutfi Pacino	5	71717.9	Ishwarya Roberts	371278.2	5
Ally Streep	5000	Baden-Daettwil	Elia Fawcett	5	88462.6	Goldie Slater	282694.3	1
Bo Ashby	5000	Bangalore	Ishwarya Roberts	5	371278.2	Christian Cage	265255.6	4
Bo Dickinson	5000	Bangalore	Gustav Steenburgen	5	185700.5	Meenakshi Mason	213399.7	4

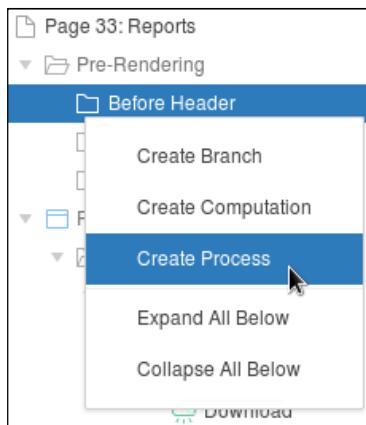
row(s) 1 - 5 of 33 [Next ▶](#)

row(s) 1 - 5 of 47 [Next ▶](#)

row(s) 1 - 5 of 47 [Next ▶](#)

release 1.0 Set Screen Reader Mode On

- Expand the Pre-Rendering node, right-click the Before Header node, and select Create Process.



- Enter DownloadPDF for Name and select the Reports2PDF [Plug-In] for Type.

▼ Identification

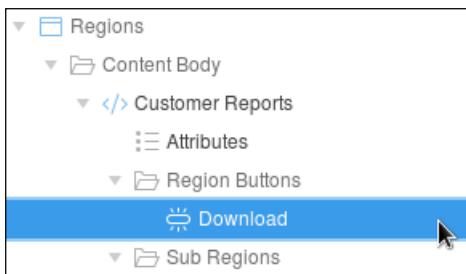
Name	DownloadPDF
Type	Reports2PDF [Plug-In]

- Under Condition, select Request = Value for Type and enter download for Value.

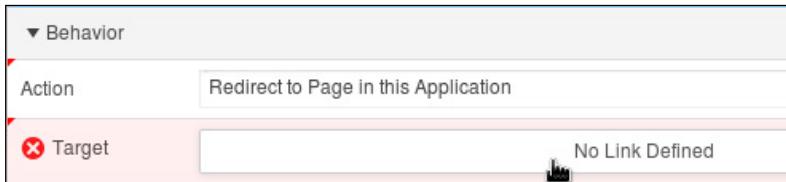
▼ Condition

When Button Pressed	- Select -
Type	Request = Value
Value	download

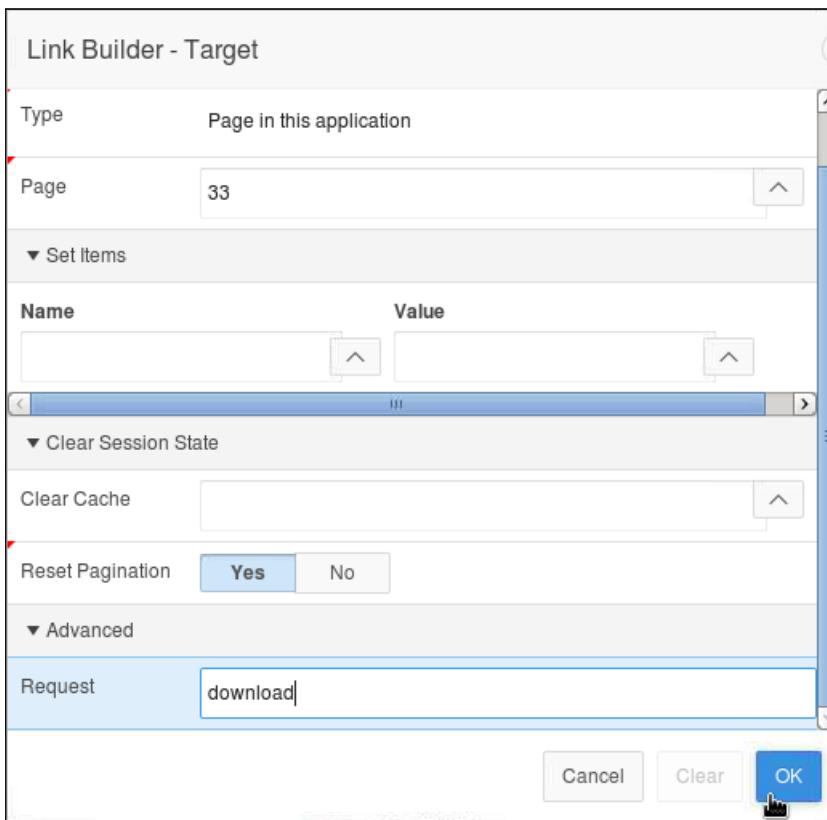
6. Click the **Download** button link.



7. For Action, select **Redirect to Page in this Application** and click the Target link.



8. Enter the page number of your current page and enter **download** for Request under the Advanced section. Click **OK**.



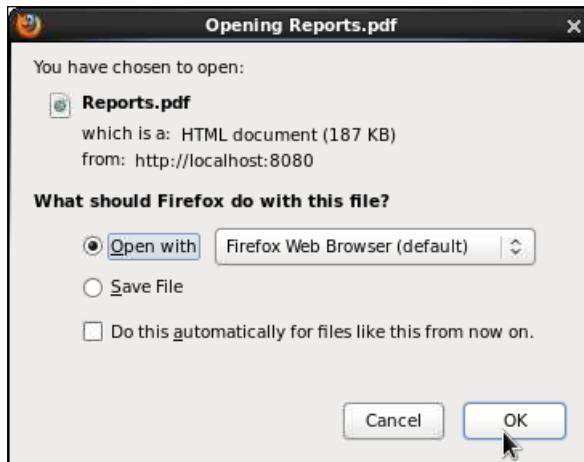
9. **Save** the page.

- d. Run the page and confirm that the process plug-in runs successfully and a PDF containing all the reports in the page is displayed.

1. **Run** the page. Click the Download button.



2. You will be prompted to open or save the PDF file. Choose an option and click **OK**.



3. The PDF is displayed.

Note: If the PDF does not open automatically, you can open it from the Downloads folder.

Name	Credit limit	City
Scott Jordan	5000	Bangalore
Ally Fawcett	5000	Zurich
Ally Brando	5000	Bern
Ally Streep	5000	Baden-Daettwil
Bo Hitchcock	5000	Racine
Bob McCarthy	5000	Buffalo
Dom McQueen	5000	Albany
Dom Hoskins	5000	Schenectady
Don Siegel	5000	Syracuse
Gvtz Bradford	5000	Pittsburgh
Holly Kurosawa	5000	Lancaster
Rob MacLaine	5000	Philadelphia
Don Barkin	5000	Philadelphia
Kurt Danson	5000	Baltimore
Kurt Heard	5000	Baltimore
Goetz Falk	5000	Mumbai

Practices for Lesson 6: Incorporating Interactivity Using JavaScript and jQuery

Chapter 6

Practices for Lesson 6: Overview

Practices Overview

In these practices, you follow the given steps to learn how technologies like jQuery and JavaScript can be used with Oracle APEX.

Guided Practice 6-1: Using jQuery in an APEX application

Overview

This is a guided practice. You can follow the steps given in this practice to understand how jQuery can be used in Oracle APEX.

Assumptions

You have completed all previous practices or imported the solution application sol_05_03.

Tasks Overview

In the GMT application, there is a dialog box that is opened when the Change Address button is clicked on the Confirm Address page. The default box that opens is of fixed size. When the dialog box is opened, a user can move the box around and place it at a desired location on the screen. However, when the box is closed and opened again, the dialog box opens at the center of the screen only. By default, the dialog box does not remember if the user moved the box to a different position on the screen and does not open the dialog box at that location.

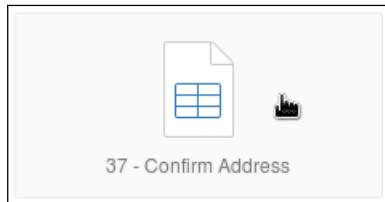
In this guided practice, you will use jQuery with dynamic actions to implement the following enhancements to the Update Address dialog box that opens when the Change Address button on the Confirm Address page is clicked:

- a. Animate the opening and closing of the modal dialog box.
- b. Modify dialog box attributes to allow dialog box resizing.
- c. Save the position and size of the dialog box when it is moved around the screen and/or re-sized.
- d. Restore saved position and/or size when opening the dialog box again.

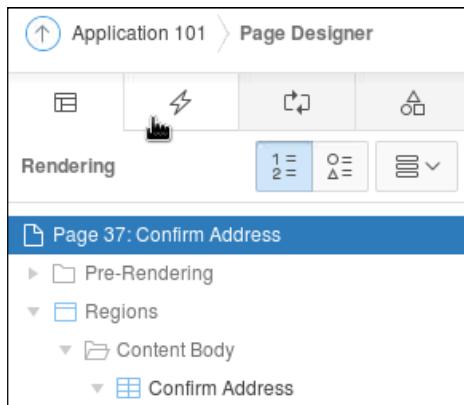
Perform this guided practice to accomplish each of the above tasks.

Steps

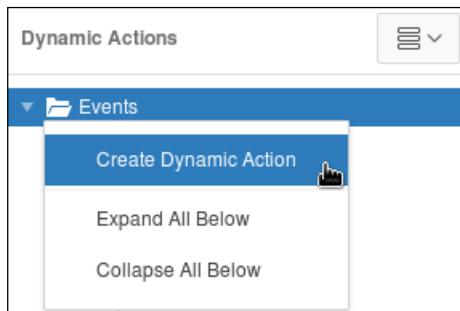
- a. Animate the opening and closing of the modal dialog box.
 1. Navigate to the GMT application and click the **Confirm Address** page.



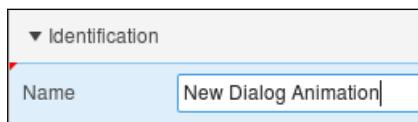
2. You need to create a dynamic action that fires the animation effect when a dialog box is opened. Click the Dynamic Actions tab.



3. Right-click Events and click Create Dynamic Action.



4. Enter New Dialog Animation for Name.



5. Since there is no predefined event to open a dialog box, select **Custom** for Event. In Custom Event, enter **dialogopen**. This is an existing browser event so you need not explicitly call it elsewhere on the page or application. For Selection Type, select **jQuerySelector**, and enter **body** in the jQuery Selector field.

The screenshot shows a configuration panel with the following settings:

- Event:** Custom
- Custom Event:** dialogopen
- Selection Type:** jQuery Selector
- jQuery Selector:** body

6. You now need to specify the animation code that should be executed when the dialogopen event is triggered. Select the default action listed under true and change the action to **Execute JavaScript Code**.

The screenshot shows a configuration panel with the following setting:

- Action:** Execute JavaScript Code

7. Enter the following code in the Code box:

```
// Sometimes there may be more than one dialog on the page, when  
// a dialog opens.  
// To make sure we get the right one, we should grab the last  
// one in the document (guaranteed to be the one we want!)  
var dialog$ = $(".ui-dialog").last();  
var modal$ = $(".ui-widget-overlay").last();  
// Change a property that has been declared to be animatable  
// (see the CSS) and the browser will take care of the rest  
// Just add a class that overrides the existing properties for  
// the animation to start.  
dialog$.addClass("animate-in-dialog");  
modal$.addClass("animate-in-modal");
```

8. Ensure that there is no selected value for Affected Elements.

The screenshot shows a configuration panel with the following setting:

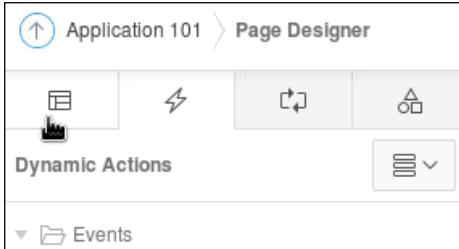
- Selection Type:** - Select -

9. Set Fire on Page Load to **No**.

The screenshot shows a configuration panel with the following setting:

- Fire On Page Load:** No

10. Click the Page Rendering tab.



11. Click the page name node and enter the following in the page CSS inline field:

```
body .ui-dialog, body .ui-widget-overlay {  
    opacity:0;  
    transition: transform 1s, opacity 1s, margin-top 1s;  
}  
.ui-dialog {  
    margin-top: -75px;  
}  
.ui-dialog.animate-in-dialog {  
    margin-top: 0px;  
    opacity: 1;  
    transform: rotateY(360deg) ;  
}  
.ui-widget-overlay.animate-in-modal {  
    opacity: .15;  
}  
  
.ui-dialog.animate-out-dialog {  
    transition: .5s all;  
}
```

12. Save the page and run it.



13. Click the **Change Address** button. Notice the animation effect when the dialog box opens.

Note 1: Due to network speed, you might not be able to see the flip animation clearly. The slight delay in rendering the dialog is an indication that the animation is being used. You can try the same code in apexcorp.oracle.com to view the animation clearly.

Note 2: There may be no content on the Confirm Address page as there is no session data yet. This is fine. Otherwise, you can click Previous two times to reach the Identify Customer page. Then, click Next two times to return to the Confirm Address page.

14. Close the dialog box. Notice that no animation is seen. (Dialog closes immediately without any delay).
15. You will now add the code to animate the dialog box when it is closed. Click the edit link.

16. Open the dynamic action created previously and enter the following code below the existing code:

```
// If you comment out all the code below this line, you'll see  
// that the animation for closing does not work.  
  
// The reason why is that jQueryUI dialogs will automatically  
// remove dialogs the moment you close them from your page.  
// To work around this, we need to "intercept" the event before  
// close, start the animation, and then  
// 1 second later, actually remove the dialog from the body.  
  
// This variable will stop the first close event, but once the  
// animation starts, it will allow the dialog  
// to be closed normally.  
var closeForReal = false;  
dialog$.on( "dialogbeforeclose", function( event, ui ) {  
    // Using "closeForReal" as a guard, we can determine whether or  
    // not the animation has started, and accordingly  
    // if the dialog should be removed from the DOM.  
    if ( closeForReal ) {  
        // If the animation has started, let the event proceed as  
        // normal.  
        return;  
    }  
    // Remove the css classes we had before so the animation on the  
    // dialog/modal goes in reverse  
    dialog$.removeClass("animate-in-dialog").addClass("animate-out-  
    dialog"); // We add this class just so dialog goes away a bit  
    // faster  
    modal$.removeClass("animate-in-modal");  
    // Indicate the animation has started and the guard is no longer  
    // needed  
    closeForReal = true;  
    // We can't just immediately close the dialog though! We need to  
    // wait 1 second (1000 milliseconds) until the  
    // animation is finished. Then and only then, can we call the  
    // jQueryUI widget's close directly.  
    setTimeout(function() {  
        // Note that the dialog reference is not attached to ui-dialog,  
        // it's actually buried in the dialog's child with the class  
        // "ui-dialog-content". See the jqueryUI api for more details:  
        // http://api.jqueryui.com/dialog/.  
        dialog$.find('.ui-dialog-content').dialog("close");  
    }, 1000);  
    // When you return "false" in an jQuery event, that's a signal  
    // to all the events or triggers that come after it to
```

```
// NOT continue. In this case, returning will stop the dialog  
from closing.  
return false;  
});
```

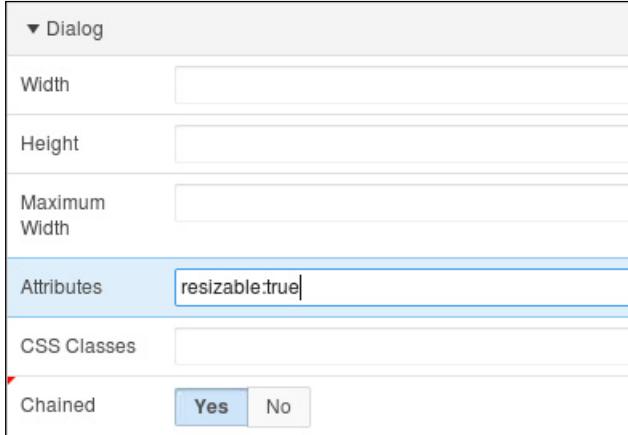
17. Save and run the page. Open the dialog box and now close the dialog. Notice the animation effect.

Note: Due to network speed, you might not be able to see the flip animation clearly. The slight delay in rendering the dialog is an indication that the animation is being used. You can try the same code in apexcorp.oracle.com to view the animation clearly.

- b. Modify dialog box attributes to allow dialog box resizing.

1. Ensure that you are on the **Confirm Address** web page and click the **Change Address** button to open the dialog box.
2. Try to resize the dialog box. You will not be able to resize the dialog box. You have to modify some settings to allow a dialog box to be resized. Click the edit page link from the Developer toolbar.
3. In page properties, under Dialog, enter the following code in the attributes field:

```
resizable:true
```



4. Save the Update Address page.



5. Run the Confirm Address page.



6. Click **Change Address** button to open the dialog box. Notice that you can now resize the dialog box.

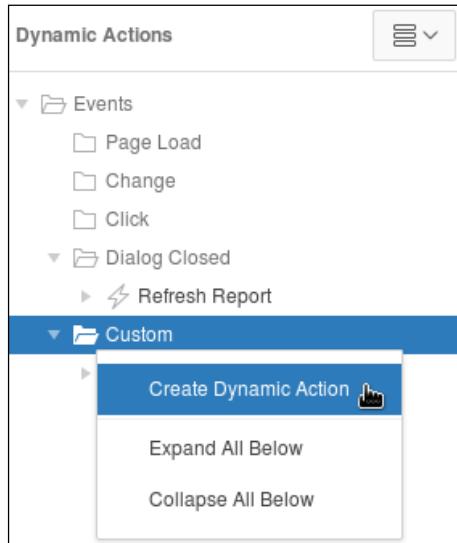
7. Close the dialog box.

- c. Save the position and size of the dialog box when it is moved around the screen and/or resized.

1. Ensure that you are on the Confirm Address page and click **Change Address** button to open the dialog box.
2. Now move the dialog box to a different location in the screen.
3. Close the dialog box and open it again. Notice that the dialog box opens in the default center of the screen position.

4. Similarly, open the dialog box and resize it.
5. Now close the dialog box and open it again. Notice that the dialog box is opened with the default size. Close the dialog box. You will now create dynamic action JavaScript code that will identify and store the position and size of a dialog box when it is changed by a user. Click Edit Page.
6. On the Dynamic Actions tab, right-click **Custom** and select **Create Dynamic Action**.

Note: Ensure that you are in the Confirm Address page.



7. Enter Persist Dialog Position and Size for Name.

▼ Identification	
Name	Persist Dialog Position and Size

8. Enter dialogresizestop dialogdragstop for Custom Event.

Event	Custom
Custom Event	dialogresizestop dialogdragstop

9. Select **jQuery Selector** for Selection Type and enter **body** in the jQuery Selector field.

Selection Type	jQuery Selector
jQuery Selector	body

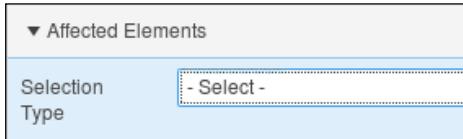
10. Select the default true action and change it to **Execute JavaScript Code**.

▼ Identification	
Action	Execute JavaScript Code

11. Enter the following code in the Code box:

```
// handle dialog move and resize
var key, value, store,
dialog$ = $(this.browserEvent.target), // the dialog being moved
or resized
// the dragstop (move) event doesn't include size so get it from
the options
size = this.data.size ||
{ width: dialog$.dialog("option", "width"),
height: dialog$.dialog("option", "height") },
pos = this.data.position;
// Only save the state of apex dialogs.
// Give your dialogs their own class if you want to target
specific dialogs
if (dialog$.parents(".ui-dialog--apex").length) {
// store the dialog state under a key that includes
// the app id, page id, and dialog page id
value = [size.width, size.height, pos.left, pos.top];
key = dialog$.find("iframe").attr("src");
key = "dlg_" + key.split(":")[1]; // get the page of the dialog
being opened
// save the dialog state in browser session storage
store = apex.storage.getScopedSessionStorage( {
prefix: "DialogSize", usePageId: true} );
store.setItem(key, value.join(":"));
}
```

12. Ensure that there is no selected value for Affected Elements.

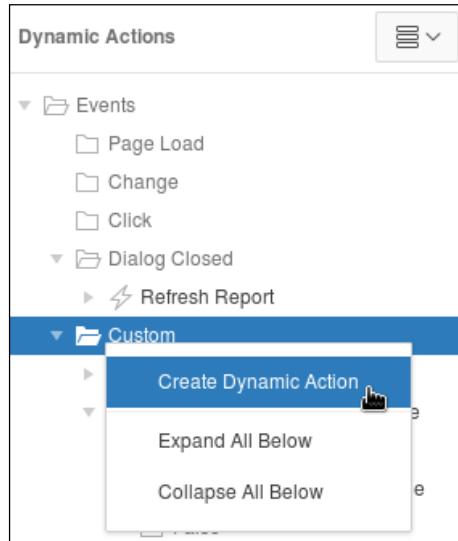


13. Set Fire on Page Load to No.



14. Save and run the page. Try dragging and resizing the dialog box. What do you observe? Notice that even though you change the position and size of the dialog box, once you close it and then open it, the dialog box still opens in the default position. This is because you have saved the custom position and size, but you have not used this saved data to create the dialog box. The dialog box is still created using the default values. You need to create another dynamic action that fires when the dialog box is created. Click the edit page link.

- d. Restore saved position and/or size when opening the dialog box again.
- You need to create another dynamic action that is fired each time a dialog box is opened that will execute some JavaScript code that will restore the dialog box size and position to the saved attributes, if any. On the Dynamic Actions tab, right-click **Custom** and select **Create Dynamic Actions**.



- Enter **Restore Dialog Size** for Name.

The 'Identification' section of a dynamic action configuration. The 'Name' field contains the value 'Restore Dialog Size'.

- Enter **dialogcreate** for Custom Event.

The 'When' section of a dynamic action configuration. The 'Event' field is set to 'Custom' and the 'Custom Event' field is set to 'dialogcreate'.

- Select **jQuery Selector** for Selection Type and enter **body** for jQuery Selector.

The 'Selection Type' section of a dynamic action configuration. The 'Selection Type' field is set to 'jQuery Selector' and the 'jQuery Selector' field is set to 'body'.

- Change the default true action to **Execute JavaScript Code**.

The 'Action' section of a dynamic action configuration. The 'Action' field is set to 'Execute JavaScript Code'.

6. Enter the following code in the Code box:

```
var key, value, store,  
dialog$ = $(this.browserEvent.target); // this is the dialog  
created  
// Only restore state for apex dialogs  
if (dialog$.parents(".ui-dialog--apex").length) {  
// the key includes the dialog page that just opened  
key = dialog$.find("iframe").attr("src");  
key = "dlg_" + key.split(":")[1];  
store = apex.storage.getScopedSessionStorage( {  
prefix: "DialogSize", usePageId: true} );  
value = store.getItem(key);  
if (value) {  
value = value.split(":");  
if (value.length === 4) {  
dialog$.dialog("option", "width", parseInt(value[0], 10));  
dialog$.dialog("option", "height", parseInt(value[1], 10));  
dialog$.dialog("option", "position",  
[parseInt(value[2], 10), parseInt(value[3], 10)]);  
}  
}  
}
```

7. Ensure that there is no selected value for Affected Elements.



8. Set Fire on Page Load to **No**.



9. Save and run the page.

10. Now the dialog box opens at the customized position and size.
11. Test the dialog box by dragging it across the screen and also resizing it. You will notice that after you close and open the dialog box, it opens with the last used position and size and not the default position and size.

Guided Practice 6-2: Using JavaScript in an APEX application

Overview

This is a guided practice. You can follow the steps given in this practice to understand how JavaScript can be written on the page property panel.

Assumptions

You have completed all previous practices or imported the solution application sol_06_01.

Tasks Overview

In the GMT application, you will use JavaScript code on the Products page to make the Products report scroll automatically.

Steps

1. In the GMT application, click the **Products** page.



2. Click the page name node and under JavaScript, enter the following code in the Functions and Global Variables field:

```
var ticker = function( body, direction, speed ) {
    var height = body.find("tbody").height() +
body.find("thead").height() - body.height(); // the scroll height
is the height of the table children, minus the visible height of the
region.
    body.parents(".t-Region-body").animate(
        {
            scrollTop: direction * height
        },
        height * speed, // the duration of the animation. the
simple equations means every pixel scrolled should take 20
milliseconds at least.
        "linear",
        function() {
            ticker(body, -1 * direction, speed); // make it scroll
in the opposite direction once finished.
        });
};

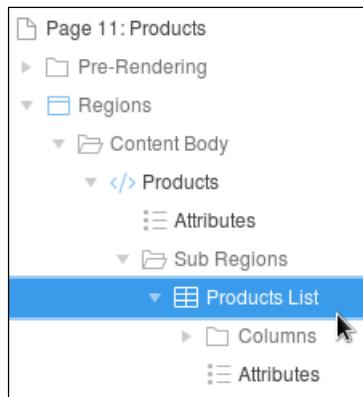
var startTicker = function ( regionId, height, speed) {
    if (!speed) {
        speed = 20;
    }
    var region$ = $("#" + regionId);
    var table$ = region$.find(".t-Report-report");
    region$.find(".t-Report-pagination").hide();
    table$.css({
```

```
        "display": "block"
    });
    table$.height(height);
    ticker(table$, 1, speed); // 1 means scroll down, -1 means
scroll up.
};
```

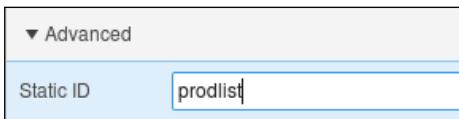
3. In the Execute when Page Loads field, enter the following code:

```
startTicker ( "prodlist", 300);
```

4. Click the **Products List** report node.



5. Under Advanced, enter **prodlist** for Static ID.



6. Save and run the page.

7. You should be able to see the report region scroll automatically.

Practices for Lesson 7: Generating and Using Table APIs

Chapter 7

Practices for Lesson 7: Overview

Practices Overview

In this practice, you generate methods on tables and add business rules to the API. You then create a form in an application that uses this API and add an error handling functionality to the GlobalMart Management Tool application.

Practice 7-1: Generating and Using Table APIs

Overview

In this practice, you generate methods on tables and add business rules to the API. You then create a form in the application that uses this API and add an error handling functionality in the GlobalMart Management Tool application.

Assumptions

You have completed Practice 6 or imported the catch-up application GMT_les_06.

Tasks

- a. Create a new package called `PRODUCTS_API` to generate procedures on the `PRODUCT_INFORMATION` table to perform DML operations in the GMT application.
- b. Add additional business rules to the `PRODUCTS_API` package.
- c. Create a form and report to a table in the GMT application based on the `PRODUCT_INFORMATION` table. The form uses the `PRODUCTS_API` for DML processes.
- d. Change the conditions for the DML processes in the **Maintain Products** page to Never.
- e. Change the **Maintain Products** form page's items source.
- f. Add an After Header process to run the GET procedure on the **Maintain Products** page.
- g. Add three On Submit-After Computations and Validations processes each for the Update, Insert, and Delete procedures on the **Maintain Products** page.
- h. Test these processes by editing, deleting, and inserting records in the form.
- i. View how errors are raised using the API.

Solution 7-1: Generating and Using Table APIs

Overview

In this practice, you generate methods on tables and add business rules to the API. You then create a form in the application that uses this API and add an error handling functionality in the GlobalMart Management Tool application.

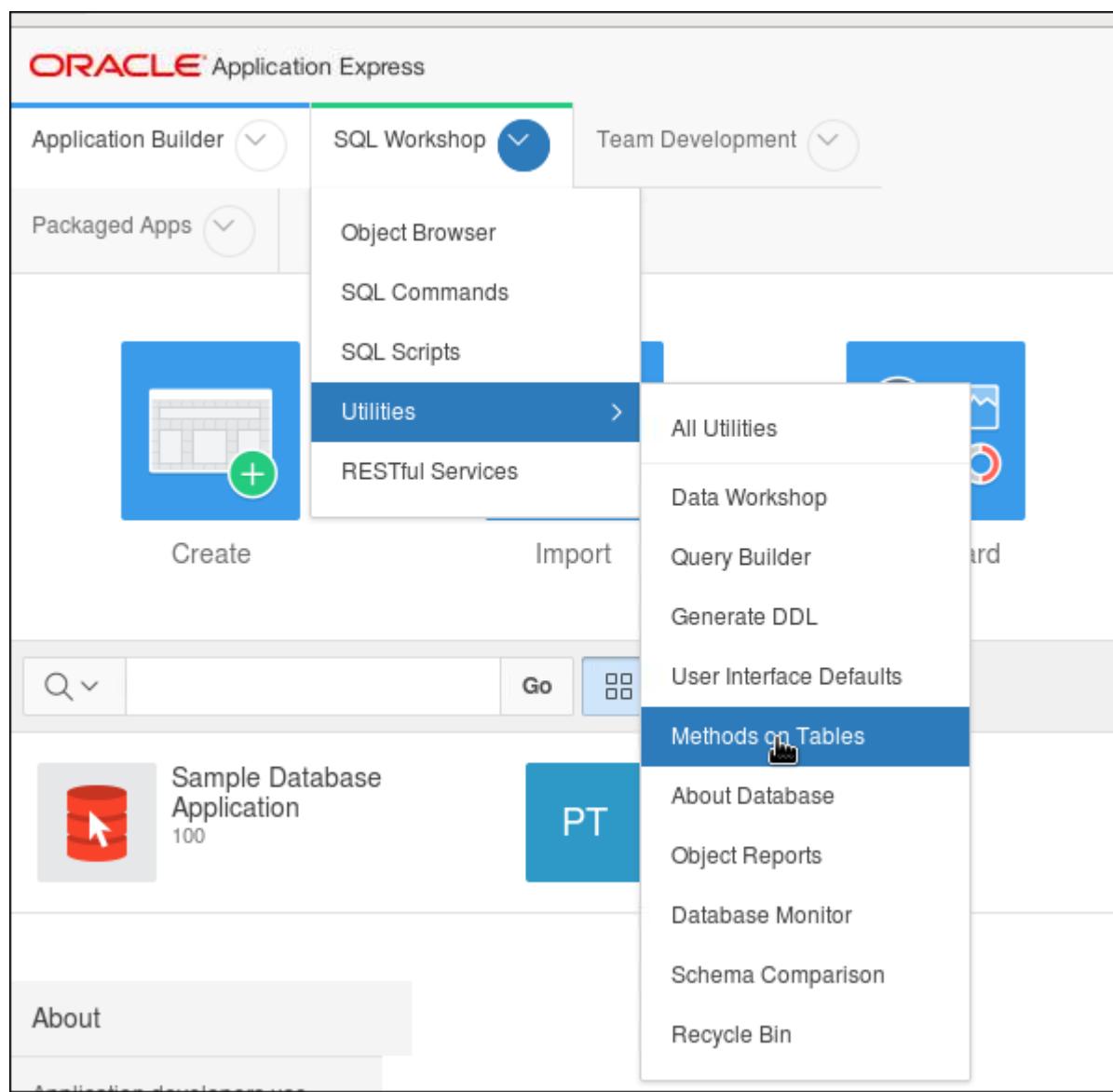
Assumptions

You have completed Practice 6 or imported the catch-up application GMT_les_06.

Steps

- a. Create a new package called `PRODUCTS_API` to generate procedures on the `PRODUCT_INFORMATION` table to perform DML operations in the GMT application.

1. Navigate to the Application Express home page and select **SQL Workshop**. Click **Utilities > Methods on Tables**.



2. Enter the Package Name as **PRODUCTS_API**. Click **Next**.

Create Package on Tables

Name

* Package Name [?](#)

Preserve Case [?](#)

< Cancel **Next >**

3. Select the table name as **PRODUCT_INFORMATION** and click **Next**.

Create Package on Tables

Schema: OEHR [?](#)

Package Name: PRODUCTS_API [?](#)

Table 1 [^](#) [?](#)

Table 2 [^](#) [?](#)

Table 3 [^](#) [?](#)

Table 4 [^](#) [?](#)

Table 5 [^](#) [?](#)

Table 6 [^](#) [?](#)

Table 7 [^](#) [?](#)

Table 8 [^](#) [?](#)

Table 9 [^](#) [?](#)

Table 10 [^](#) [?](#)

Specify the tables for which you wish to generate a PL/SQL package-based application program interface (API).

[<](#) [Cancel](#) [Next >](#)

4. Click **Create Package**.

Subprogram	Description
ins_product_information	use to insert data into PRODUCT_INFORMATION, primary key needs to be supplied
upd_product_information	use to update data in PRODUCT_INFORMATION identified by primary key(s)
del_product_information	use to delete data from PRODUCT_INFORMATION based on primary key(s)
get_product_information	use to retrieve data from PRODUCT_INFORMATION based on primary key(s)

- b. Add additional business rules to the **PRODUCTS_API** package.

1. Select the **PRODUCTS_API** and select **Body**.

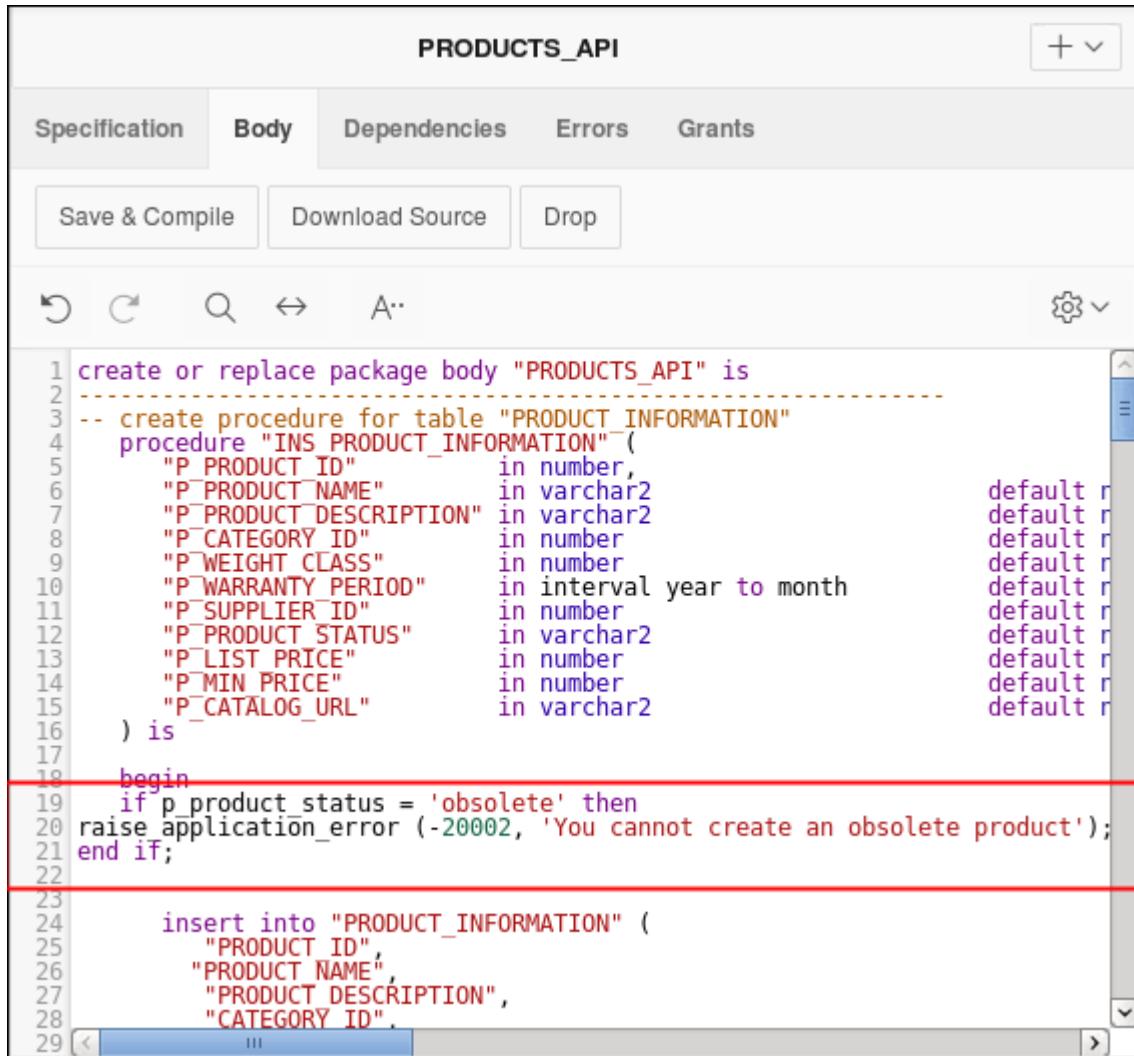
```

create or replace package "PRODUCTS_API" is
-- create procedure for table "PRODUCT_INFORMATION"
procedure "INS_PRODUCT_INFORMATION"( 
  "P_PRODUCT_ID"      in number,
  "P_PRODUCT_NAME"    in varchar2,
  "P_PRODUCT_DESCRIPTION" in varchar2,
  "P_CATEGORY_ID"     in number,
  "P_WEIGHT_CLASS"    in number,
  "P_WARRANTY_PERIOD" in interval year to month,
  "P_SUPPLIER_ID"     in number,
  "P_PRODUCT_STATUS"  in varchar2,
  "P_LIST_PRICE"      in number,
  "P_MIN_PRICE"       in number,
  "P_CATALOG_URL"     in varchar2
);
-- update procedure for table "PRODUCT_INFORMATION"
procedure "UPD_PRODUCT_INFORMATION"( 
  "P_PRODUCT_ID"      in number,
  "P_PRODUCT_NAME"    in varchar2,
  "P_PRODUCT_DESCRIPTION" in varchar2,
  "P_CATEGORY_ID"     in number,
  "P_WEIGHT_CLASS"    in number,
  "P_WARRANTY_PERIOD" in interval year to month
);

```

2. Add the following validation also located in the /home/oracle/labs/files/lab_07_01.txt file to the Create and Update procedures. This code checks if a product has the valid product status. Enter the given validation on line number 19 right after the first begin and before the insert into "PRODUCTS" (line in the Create procedure and on line number 94 just before the condition if ("P_MD5" is null) or ("L_MD5" = "P_MD5") then in the Update procedure.

```
if p_product_status = 'obsolete' then
raise_application_error (-20002, 'You cannot create an obsolete
product');
end if;
```



The screenshot shows the Oracle SQL Developer interface with the 'PRODUCTS_API' package body open. The code editor displays the package body with several sections: a create clause, a procedure for inserting into 'PRODUCT_INFORMATION', a begin block containing the validation logic, and an insert statement. The validation logic (lines 19-22) is highlighted with a red border, indicating it has been added as per the task requirements.

```
create or replace package body "PRODUCTS_API" is
-- create procedure for table "PRODUCT_INFORMATION"
procedure "INS_PRODUCT_INFORMATION"(

    "P_PRODUCT_ID"          in number,
    "P_PRODUCT_NAME"         in varchar2,
    "P_PRODUCT_DESCRIPTION"  in varchar2,
    "P_CATEGORY_ID"          in number,
    "P_WEIGHT_CLASS"         in number,
    "P_WARRANTY_PERIOD"      in interval year to month,
    "P_SUPPLIER_ID"          in number,
    "P_PRODUCT_STATUS"       in varchar2,
    "P_LIST_PRICE"           in number,
    "P_MIN_PRICE"            in number,
    "P_CATALOG_URL"          in varchar2
) is
begin
    if p_product_status = 'obsolete' then
        raise_application_error (-20002, 'You cannot create an obsolete product');
    end if;

    insert into "PRODUCT_INFORMATION" (
        "PRODUCT_ID",
        "PRODUCT_NAME",
        "PRODUCT_DESCRIPTION",
        "CATEGORY_ID",
        "WEIGHT_CLASS",
        "WARRANTY_PERIOD",
        "SUPPLIER_ID",
        "LIST_PRICE",
        "MIN_PRICE",
        "CATALOG_URL"
    );
end;
/
```

PRODUCTS_API

Specification Body Dependencies Errors Grants

Save & Compile Download Source Drop

undo redo search A...

```
81      c1."PRODUCT_NAME",
82      c1."PRODUCT_DESCRIPTION",
83      c1."CATEGORY_ID",
84      c1."WEIGHT_CLASS",
85      c1."WARRANTY_PERIOD",
86      c1."SUPPLIER_ID",
87      c1."PRODUCT_STATUS",
88      c1."LIST_PRICE",
89      c1."MIN_PRICE",
90      c1."CATALOG_URL"
91  );
92
93 end loop;
94
95 end if;
96   if p_product_status = 'obsolete' then
97     raise_application_error (-20002, 'You cannot create an obsole
98 end if;

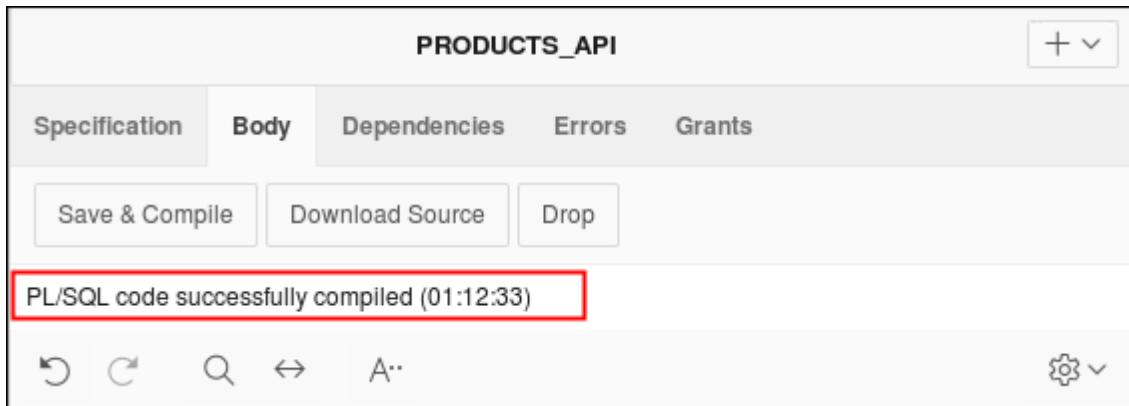
99
100 if ("P_MD5" is null) or ("L_MD5" = "P_MD5") then
101   update "PRODUCT_INFORMATION" set
102     "PRODUCT_ID"          = "P_PRODUCT_ID",
103     "PRODUCT_NAME"        = "P_PRODUCT_NAME",
104     "PRODUCT_DESCRIPTION" = "P_PRODUCT_DESCRIPTION",
105     "CATEGORY_ID"         = "P_CATEGORY_ID",
106     "WEIGHT_CLASS"        = "P_WEIGHT_CLASS",
107     "WARRANTY_PERIOD"     = "P_WARRANTY_PERIOD",
108     "SUPPLIER_ID"         = "P_SUPPLIER_ID",
109
```

3. Click **Save & Compile**.

PRODUCTS_API

Specification Body Dependencies Errors Grants

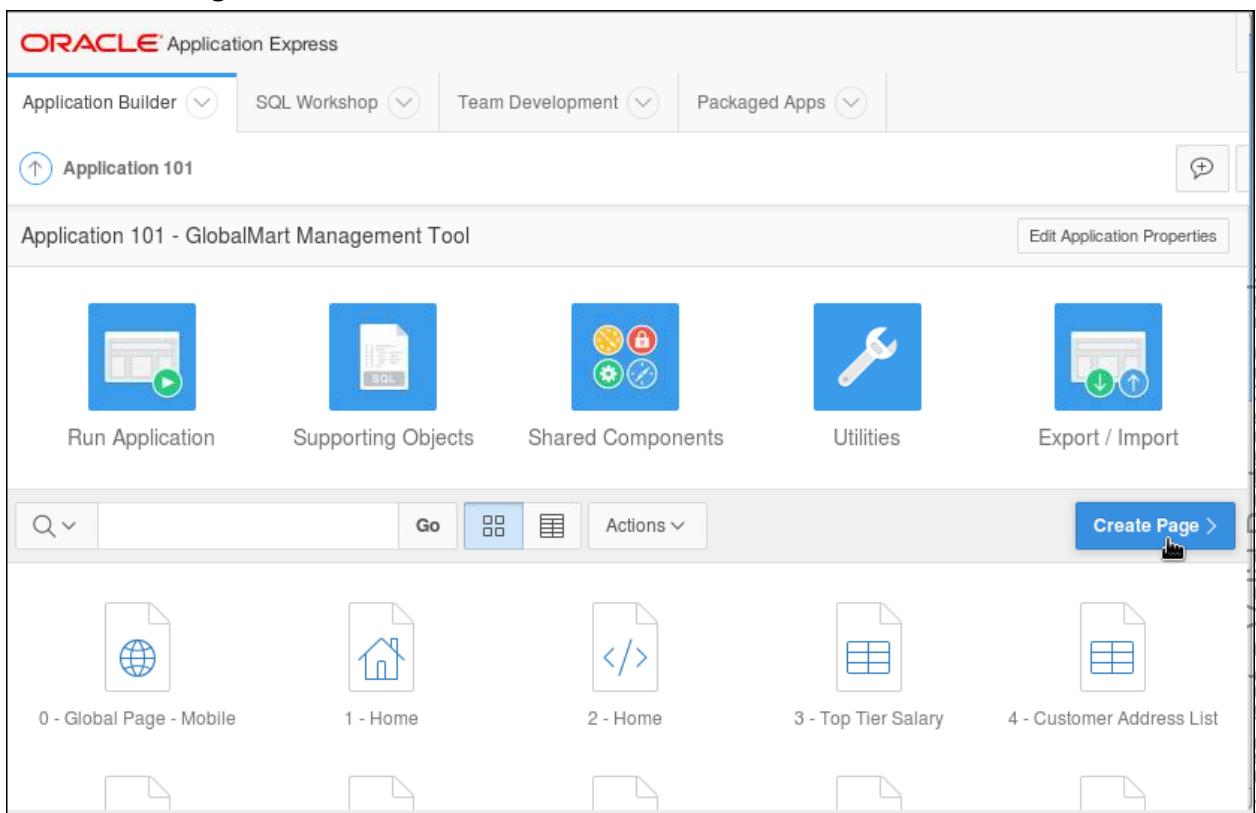
Save & Compile Download Source Drop



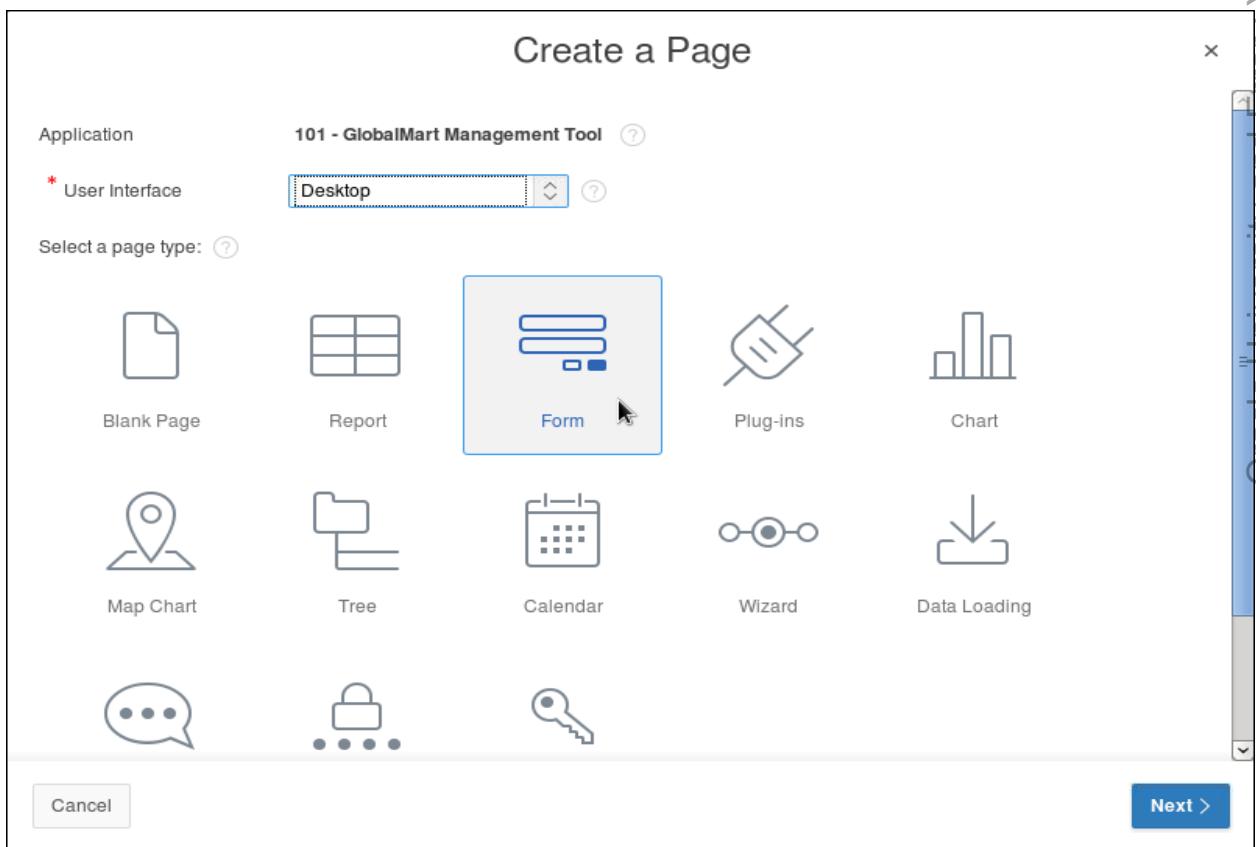
- c. Navigate to Application Builder and add a form and report to a table in the GMT application based on the `PRODUCT_INFORMATION` table. The form uses the `PRODUCTS_API` for DML processes.
1. Select **Application Builder** and click **GMT application**.

The screenshot shows the Oracle Application Express Application Builder interface. At the top, there are tabs for Application Builder, SQL Workshop, Team Development, and Packaged Apps. Below the tabs are four main icons: Create, Import, Dashboard, and Workspace Utilities. In the center, there's a search bar and a toolbar with Go, Actions, Reset, and Create buttons. Below the toolbar, there are several application tiles. One tile is highlighted in orange and labeled "GM GlobalM... Manage... Tool 101". Other visible tiles include "Sample Database Application 100" (grey), "PT PROJECT TRACKI... SYSTEM 102" (purple), "EH Error Handling 103" (orange), and "Error Handling 103" (orange). At the bottom of the screen, the URL "localhost:8080/apex/f?p=4000:1:10718818681053::NO:RP:FB_FLOW_ID,F4000_P1_FLOW,P0_FLOWPAGE,RECENT_PAGES:101,101,101" is displayed.

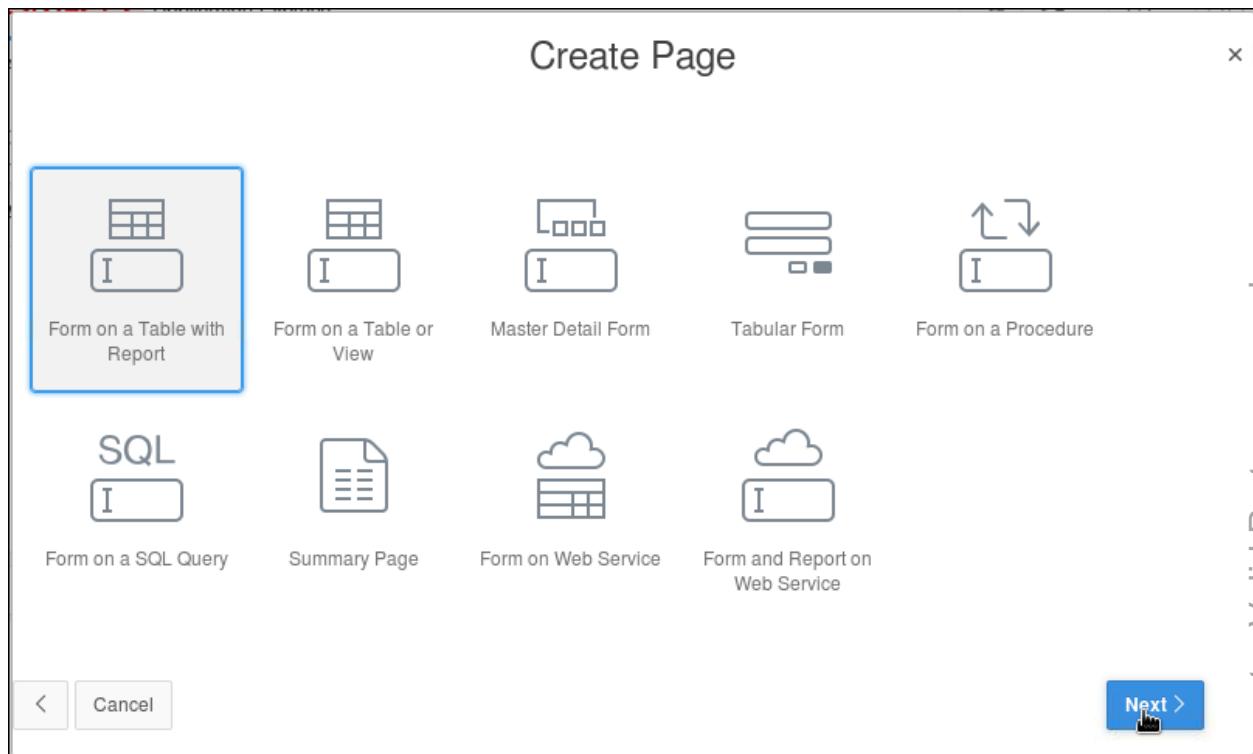
2. Click **Create Page**.



3. Click **Form**.



4. Select **Form on a Table with Report** and click **Next**.



5. Enter the Page Name and Region Title as **Product List**. Click **Next**.

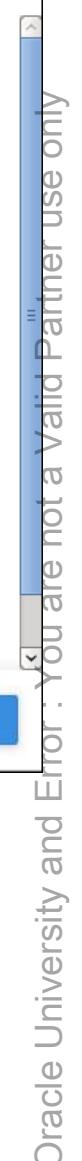
Create

Report Page

The Report page is used to select the rows to be edited. It also includes a button to create a new row.

Implementation	Interactive	?
* Page Number	42	?
* Page Name	Product List	?
* Page Mode	Normal	?
Page Group	- Select Page Group -	?
* Region Title	Product List	?
Region Template	Interactive Report	?
Report Template	template: 42. Standard	?
Pagination Size	15	?

< Cancel Next >



6. Select **PRODUCT_INFORMATION** as the Table name and click **Next**.

Create

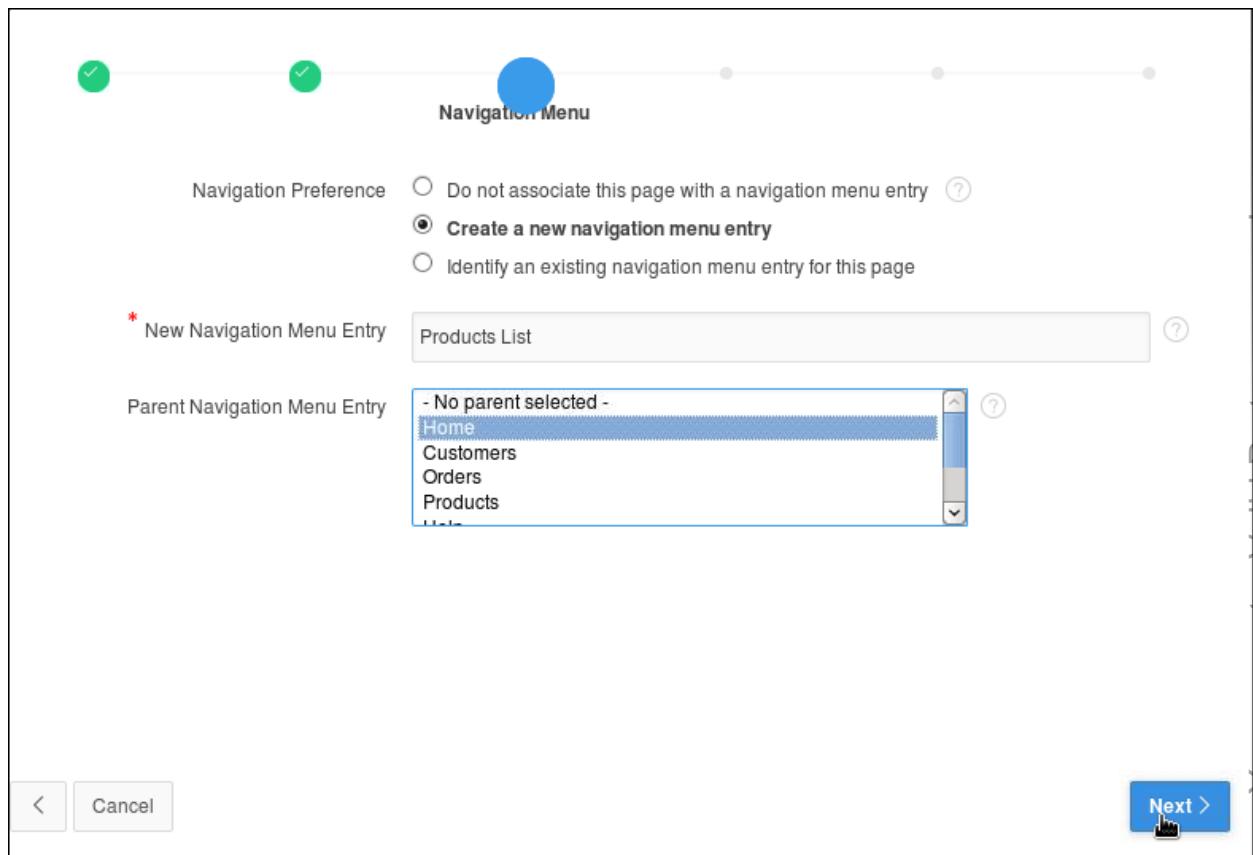
Data Source

This wizard builds a two page Report and form combination on a single table or view. The first page enables users to specify the row to be updated. The second page provides users with ability to update the selected table or view. Select the schema owner who owns the database table or view.

* Table / View Owner	OEHR	?
* Table / View Name	PRODUCT_INFORMATION (table)	?

< Cancel Next >

7. Select **Create a new navigation menu entry** for Navigation Preference. Enter **Products List** for New Navigation Menu Entry, select **Home** for Parent Navigation Menu Entry, and click **Next**.



8. Select all the columns and click **Next**.

Create Form and Report

Select the columns to include in the Report page. Define an optional WHERE clause to restrict the result set.

* Select Column(s)

Optional WHERE clause

Example SQL WHERE clause

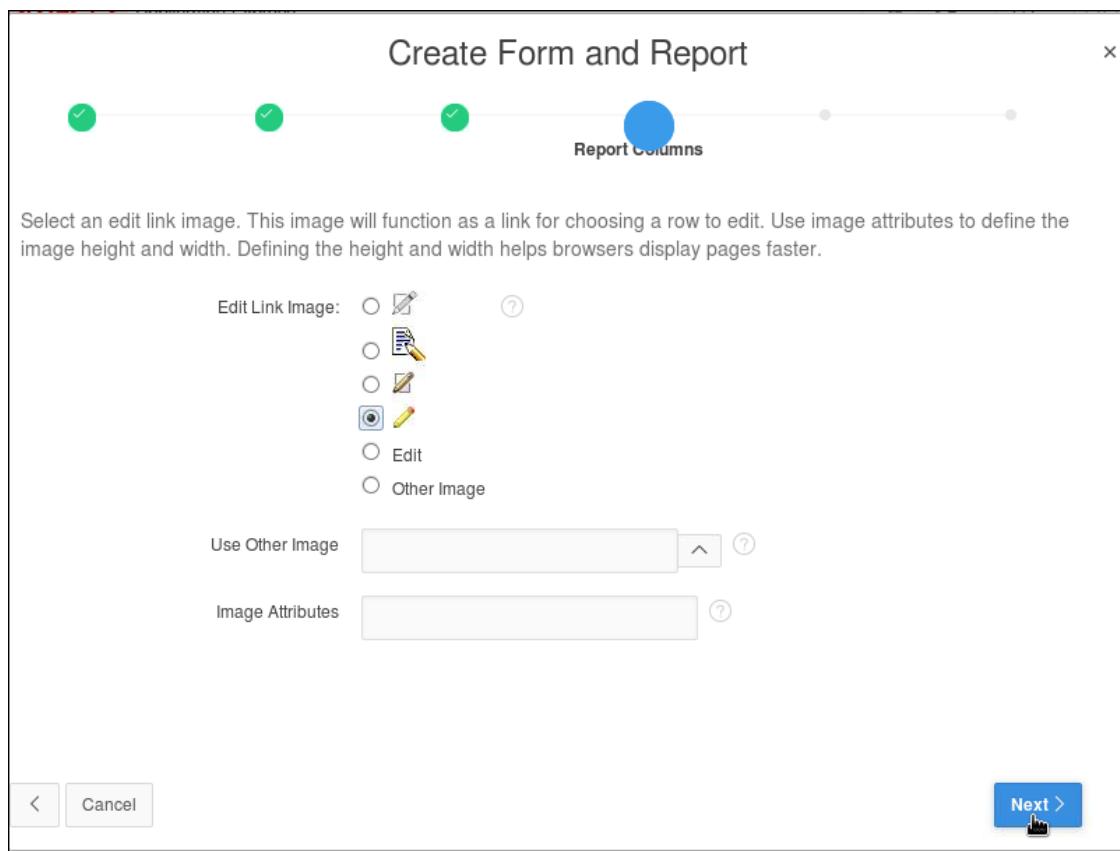
Report Columns

PRODUCT_NAME (Varchar2)
PRODUCT_DESCRIPTION (Varchar2)
CATEGORY_ID (Number)
WEIGHT_CLASS (Number)
WARRANTY_PERIOD (Interval Year(2))
SUPPLIER_ID (Number)
PRODUCT_STATUS (Varchar2)
LIST_PRICE (Number)
MIN_PRICE (Number)
CATALOG_URL (Varchar2)

Next >

Oracle University and Error : You are not a Valid Partner use only

9. Select the Edit link icon and click **Next**.



10. Enter the Page Name and Region Title as **Maintain Products** and click **Next**.

Create Form and Report

Form Page

Specify page and region information for the Form Page. The Form Page is used to insert, update, and delete rows from the selected table.

* Page Number: 43

* Page Mode: Normal

* Page Name: Maintain Products

* Region Title: Maintain Products

Region Template: Standard

< Cancel Next >

Oracle University and Error : You are not a Valid Partner use only

11. Select Primary Key Column(s) for Primary Key Type and click **Next**.

Create Form on Table

Forms perform insert, update and delete operations on table rows in the database. The rows are identified using either a Primary Key defined on the table, or the ROWID pseudo column, which uniquely identifies a row in a table. Select "Managed by Database" if you would like to use the ROWID. Otherwise select the Primary Key column(s) defined for your table. Forms support up to two columns in the Primary Key. For tables using Primary Keys with more than two columns, the ROWID option should be used.

* Primary Key Type: Managed by Database (ROWID) [?](#)
 Select Primary Key Column(s)

* Primary Key Column 1 PRODUCT_ID (Number) [?](#)

Primary Key Column 2 - Select Primary Key 2 - [?](#)

< Cancel Next >

12. Click **Next**.

Create Form and Report

Primary Key Column 1: **PRODUCT_ID** [?](#)

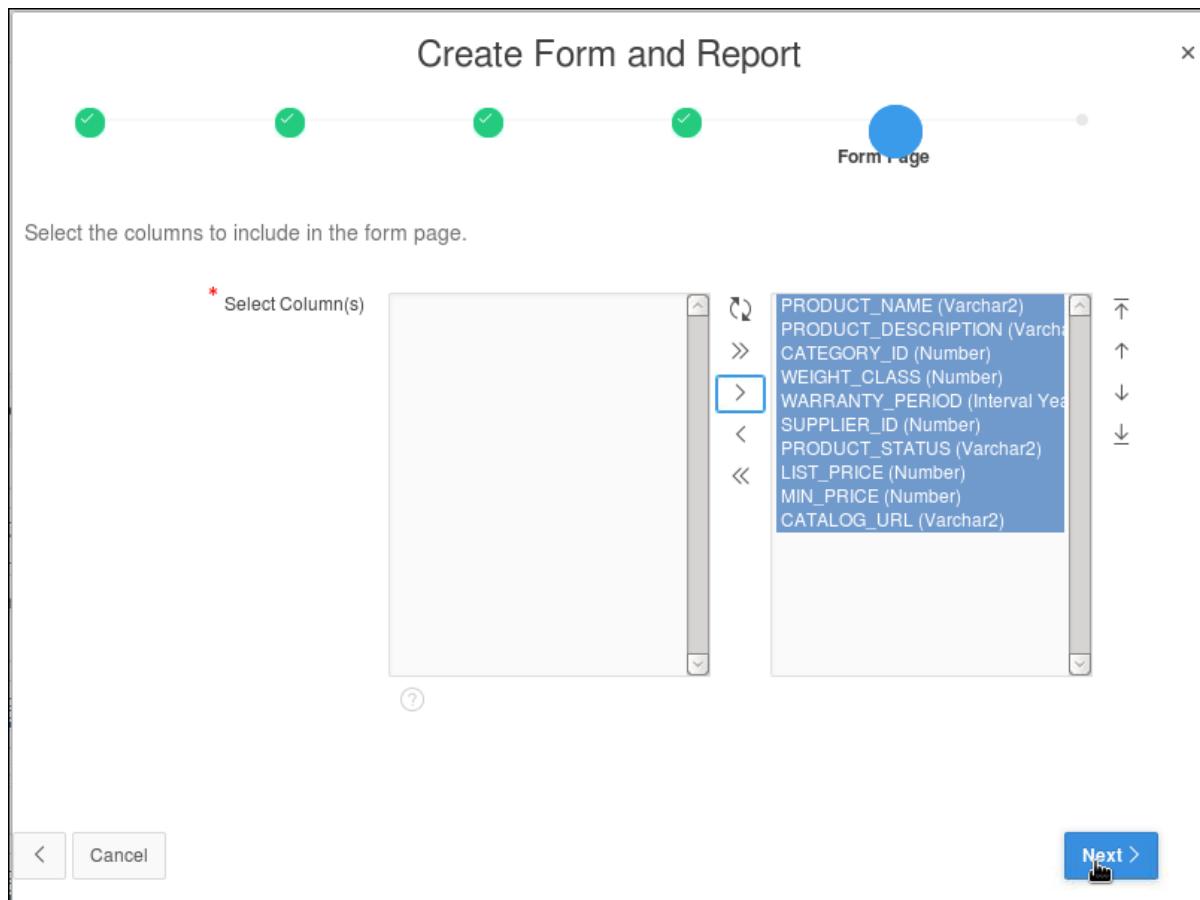
* Source for Primary Key Column 1: Existing trigger [?](#)
 Custom PL/SQL function
 Existing sequence

Existing Triggers

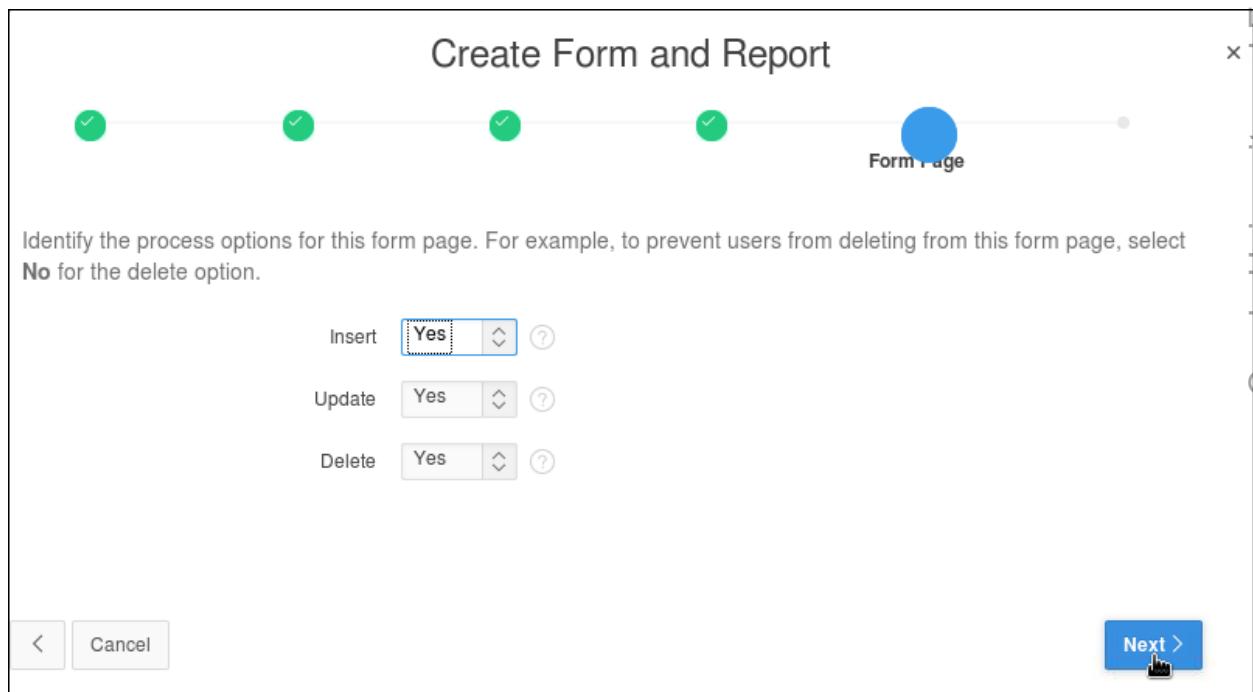
Custom PL/SQL function example

< Cancel [Next >](#)

13. Select all the columns and click **Next**.



14. Click **Next**.



15. Click **Create**.

Create Form and Report

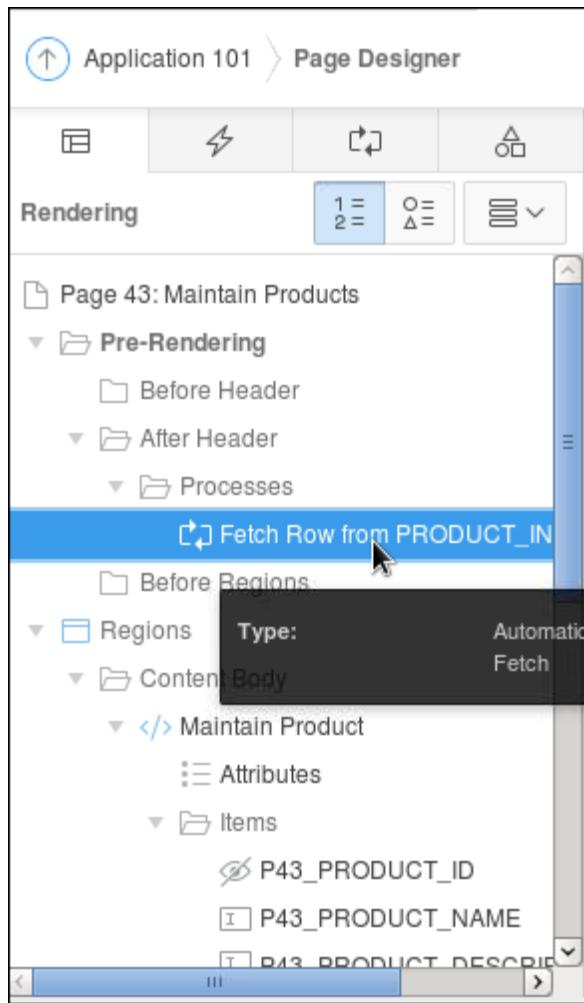
You have requested to create form and report pages with the following attributes. Please confirm your selections.

Application	101
Report Page	42
Report Page Name	Product List
Report Implementation	Interactive
Form Page	43
Form Page Name	Maintain Products
Table / View Owner	OEHR
Table / View Name	PRODUCT_INFORMATION
Primary Key Column 1	PRODUCT_ID

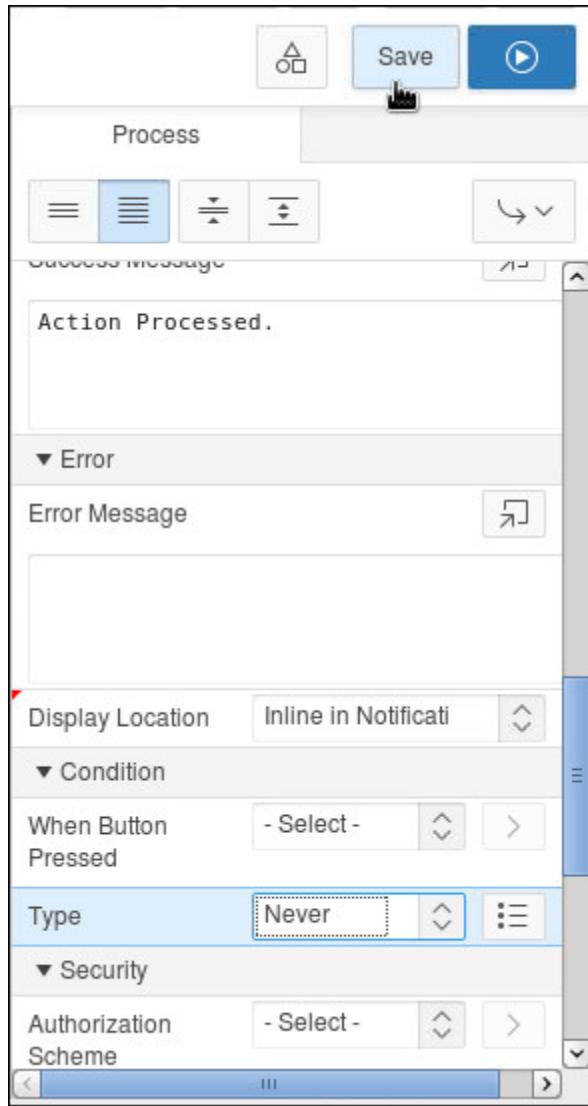
< Cancel Create

Oracle University and Error : You are not a Valid Partner user only

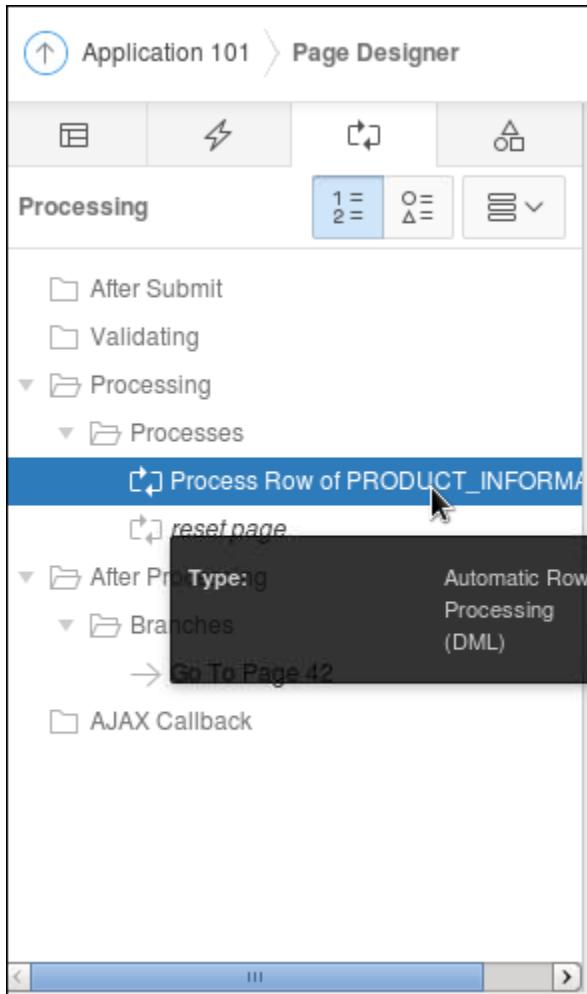
- d. Change the conditions for the DML processes on the **Maintain Products** page to **Never**.
1. Go to the **Maintain Products** page and click the **Fetch Row from PRODUCTS** process.



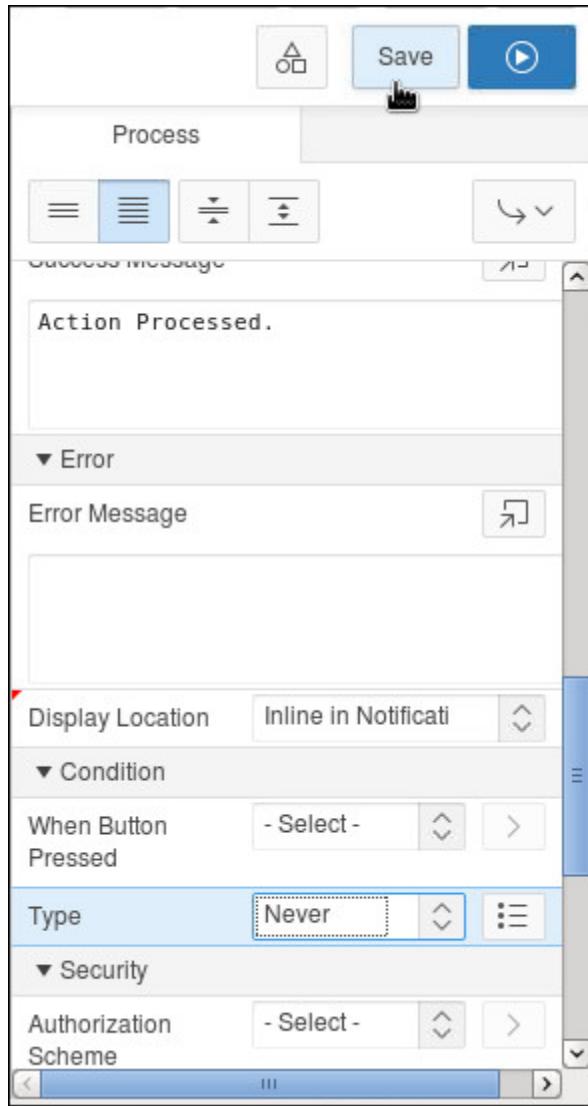
2. In the Property Editor section, select **Never** for Condition Type. Click **Save**.



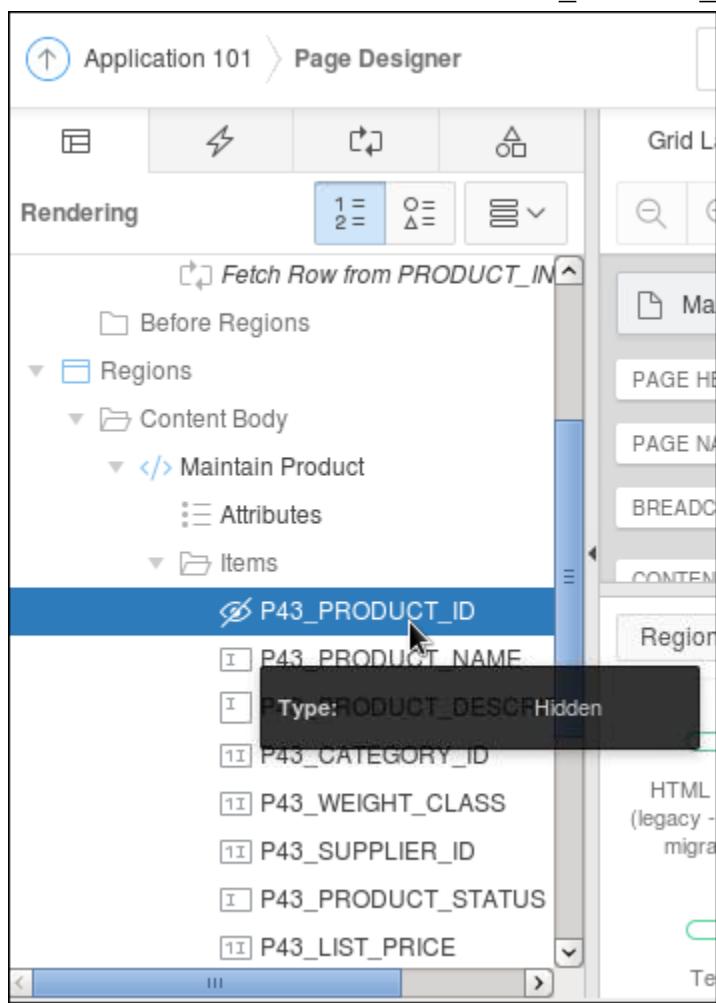
3. Similarly, click **Process Row of PRODUCTS** under **Page Processing**.



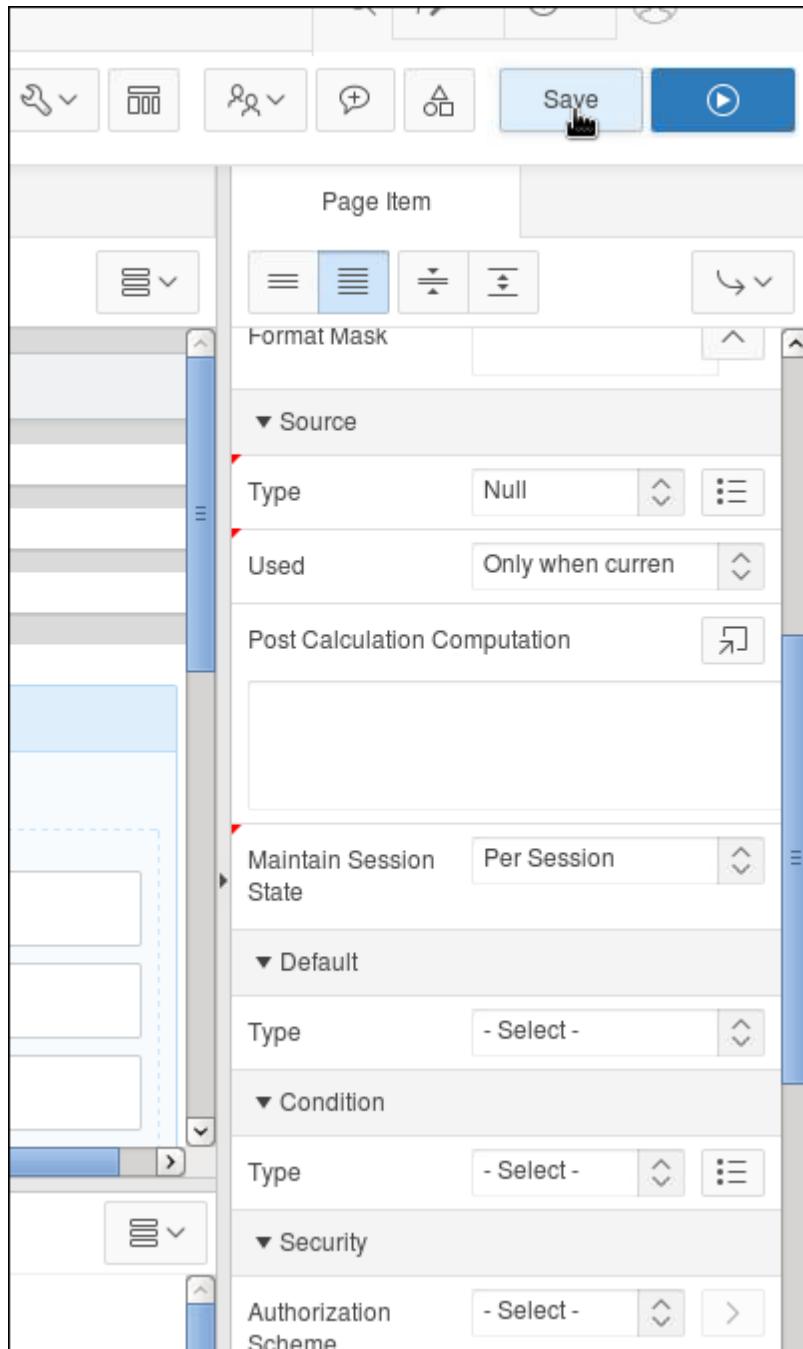
4. In the Property Editor section, select **Never** for Condition Type. Click **Save**.



- e. You now have to change the form page's items source.
1. In the **Maintain Products** form, click `P<n>_PRODUCT_ID`.

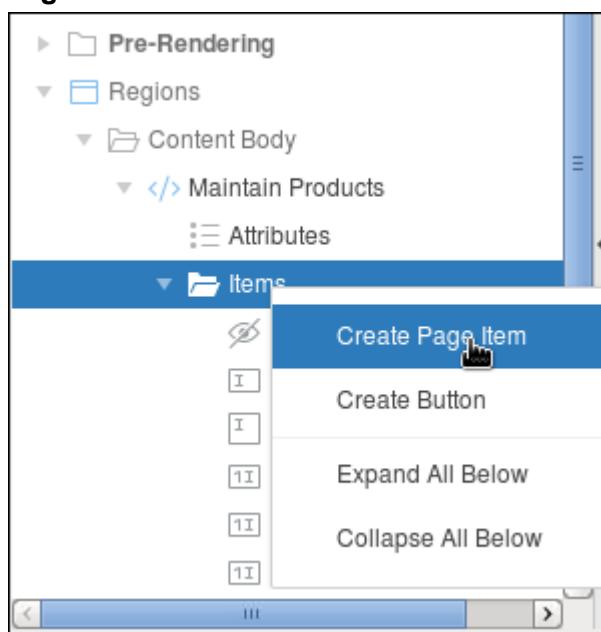


2. Change **Source Type** to Null. Click **Save**.

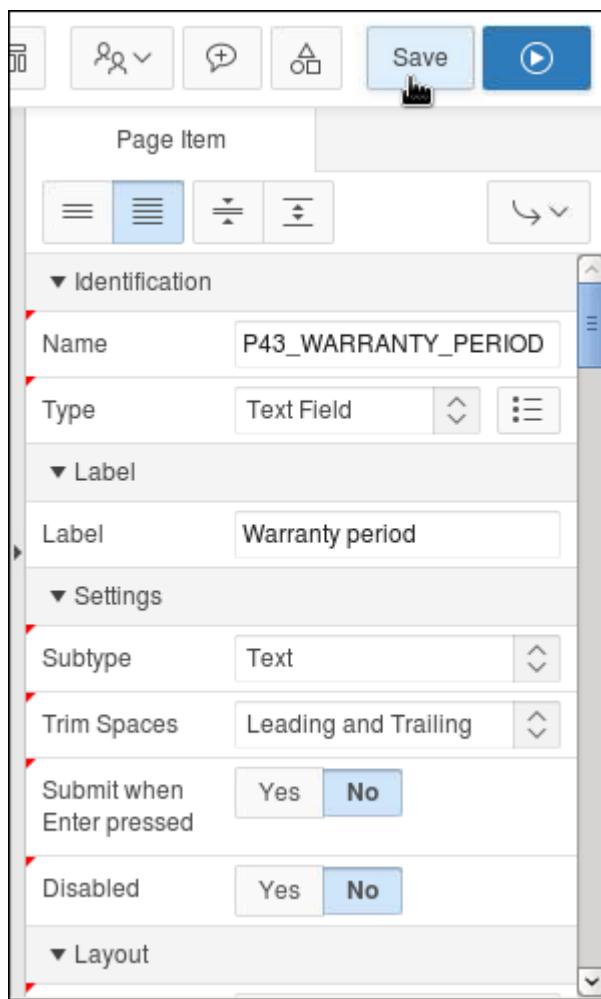


3. Repeat the previous step (Step b) for all the items on the Maintain Products form page.

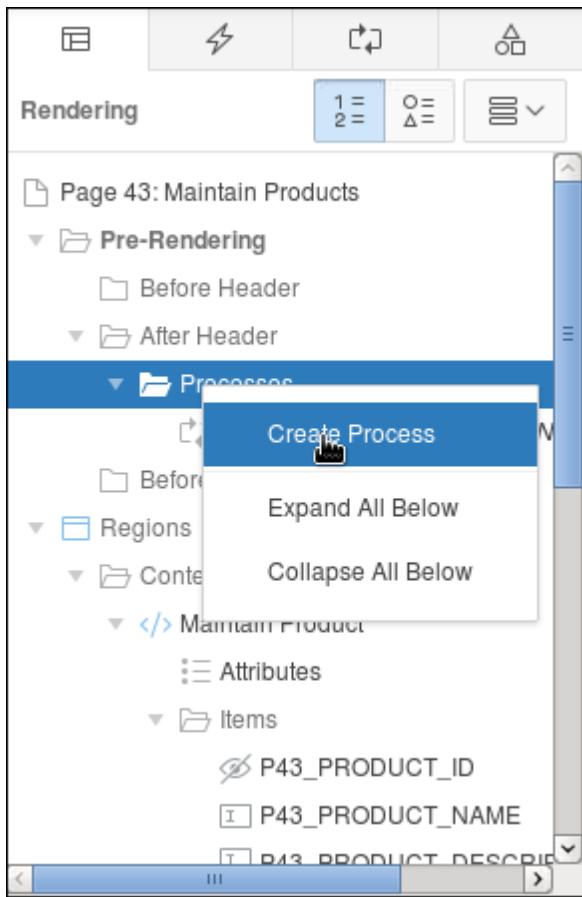
4. Add the `WARRANTY_PERIOD` item in the form. Right-click Items and select **Create Page Item**.



5. Enter **P<n>_WARRANTY_PERIOD** for name. Change **Source Type** to **Null**. Click **Save**.



- f. Add an After Header process to run the GET procedure.
- On the **Maintain Products** form page, right-click **Processes** under **After Header** and select **Create Process**.



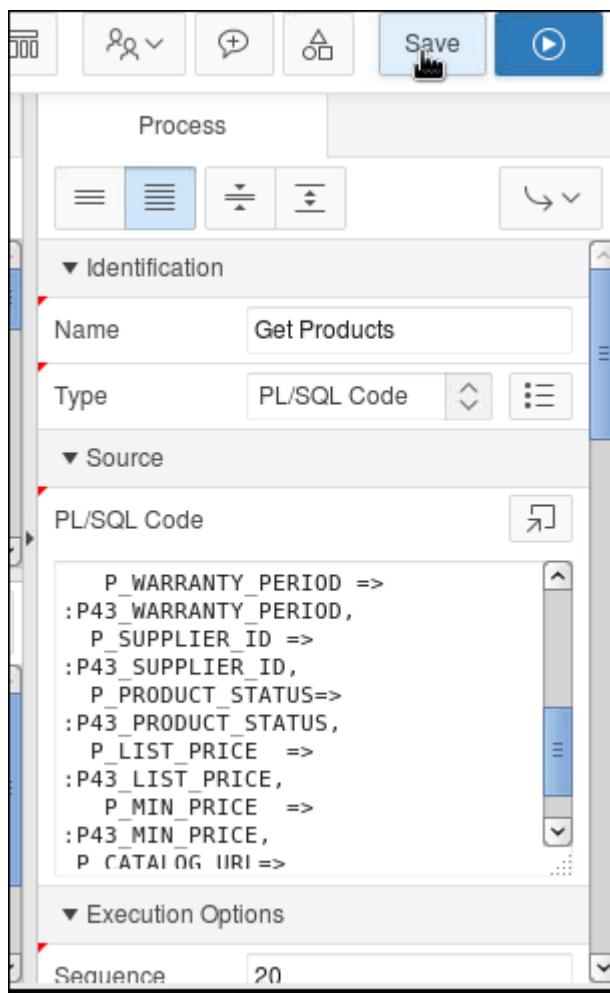
- In the Property Editor section, enter **Get Products** for **Name** and select **PL/SQL** for **Type**.
- Enter the following PL/SQL also located in the `/home/oracle/labs/files/lab_07_01.txt` file.

Note: Enter the `P<n>` in the code below based on your page number.

```
products_api.get_product_information (
    P_PRODUCT_ID => :P43_PRODUCT_ID,
    P_PRODUCT_NAME => :P43_PRODUCT_NAME,
    P_PRODUCT_DESCRIPTION=> :P43_PRODUCT_DESCRIPTION,
    P_CATEGORY_ID => :P43_CATEGORY_ID,
    P_WEIGHT_CLASS=> :P43_WEIGHT_CLASS,
    P_WARRANTY_PERIOD => :P43_WARRANTY_PERIOD,
    P_SUPPLIER_ID => :P43_SUPPLIER_ID,
    P_PRODUCT_STATUS=> :P43_PRODUCT_STATUS,
    P_LIST_PRICE => :P43_LIST_PRICE,
    P_MIN_PRICE => :P43_MIN_PRICE,
    P_CATALOG_URL=> :P43_CATALOG_URL);
```

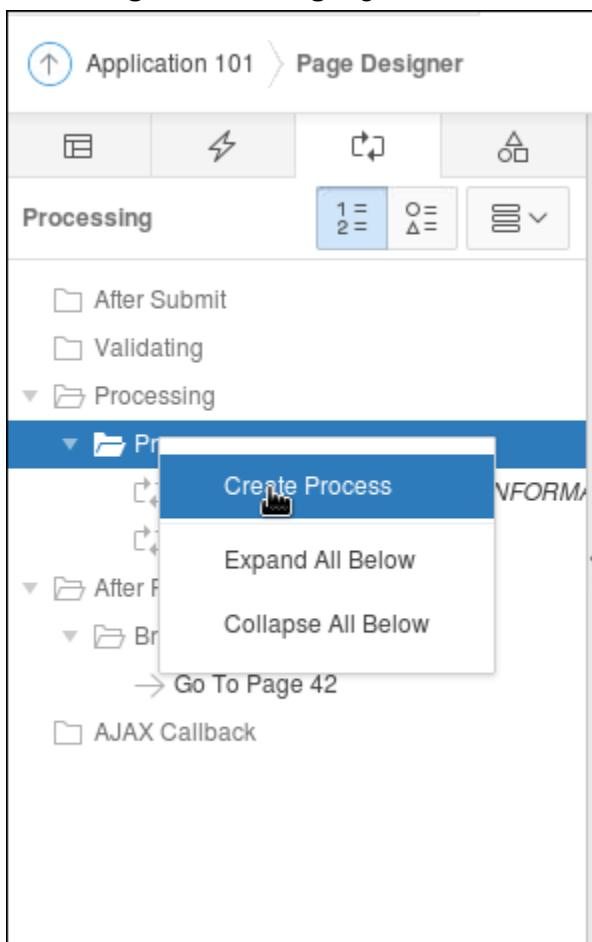
- Enter **Success** and **Error** messages for the process.

5. For Condition Type, select **Item is NOT NULL** and enter `P<n>_PRODUCT_ID` for **Item**. Click **Save**.



- g. You will next add two On Submit-After Computations and Validations processes each for the Update and Insert procedures.

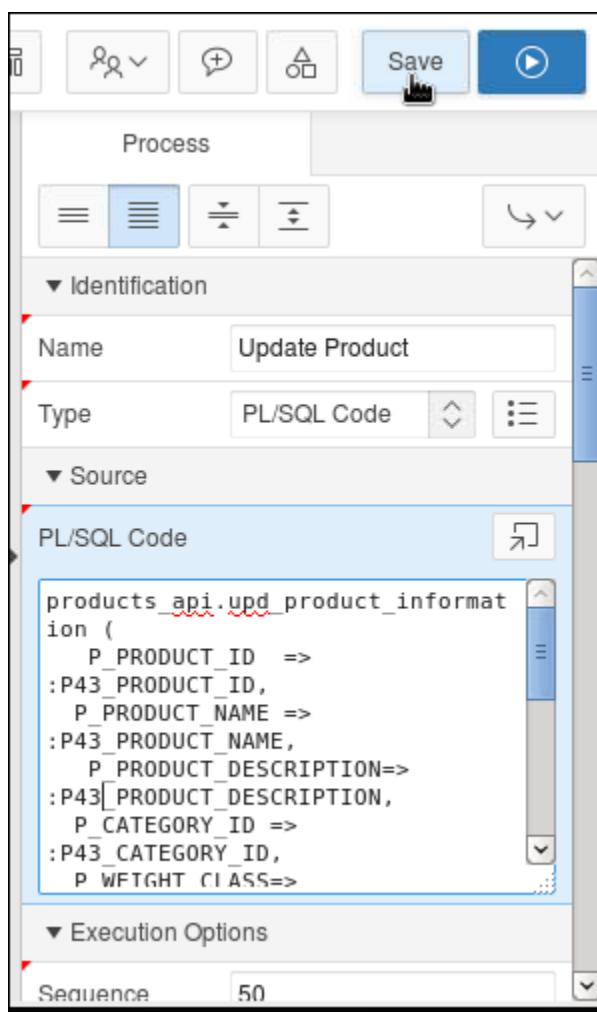
- Under **Page Processing**, right-click **Processes** and select **Create Process**.



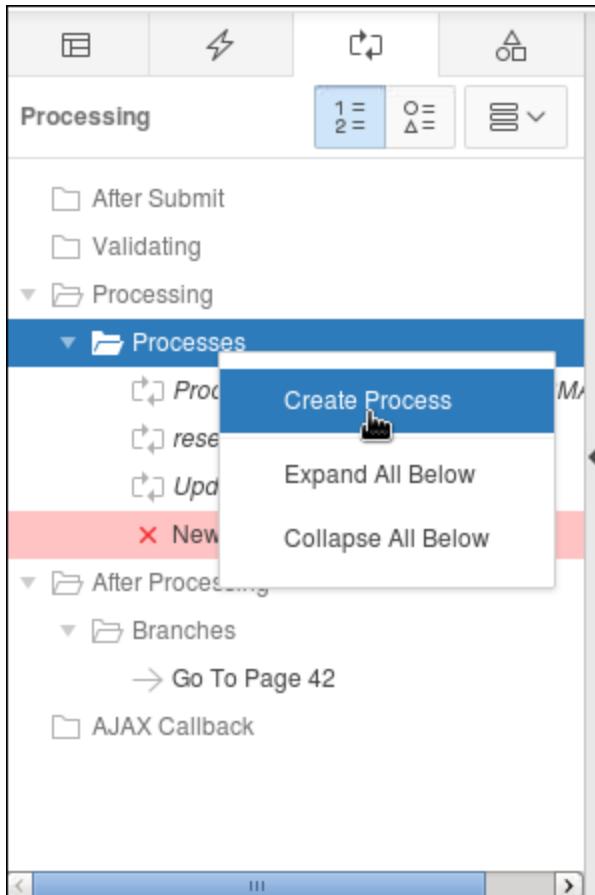
- In the Property Editor section, enter **Update Product** for **Name** and select **PL/SQL** for **Type**.
- Enter the following PL/SQL also located in the /home/oracle/labs/files/lab_07_01.txt file for the Update process.

```
products_api.upd_product_information (
    P_PRODUCT_ID => :P43_PRODUCT_ID,
    P_PRODUCT_NAME => :P43_PRODUCT_NAME,
    P_PRODUCT_DESCRIPTION=> :P43_PRODUCT_DESCRIPTION,
    P_CATEGORY_ID => :P43_CATEGORY_ID,
    P_WEIGHT_CLASS=> :P43_WEIGHT_CLASS,
    P_WARRANTY_PERIOD => :P43_WARRANTY_PERIOD,
    P_SUPPLIER_ID => :P43_SUPPLIER_ID,
    P_PRODUCT_STATUS=> :P43_PRODUCT_STATUS,
    P_LIST_PRICE => :P43_LIST_PRICE,
    P_MIN_PRICE => :P43_MIN_PRICE,
    P_CATALOG_URL=> :P43_CATALOG_URL);
```

4. Enter Success and Error messages.
5. Select **SAVE** for **When Button Pressed** and click **Save**.



6. Right-click **Processes** and select **Create Process**.

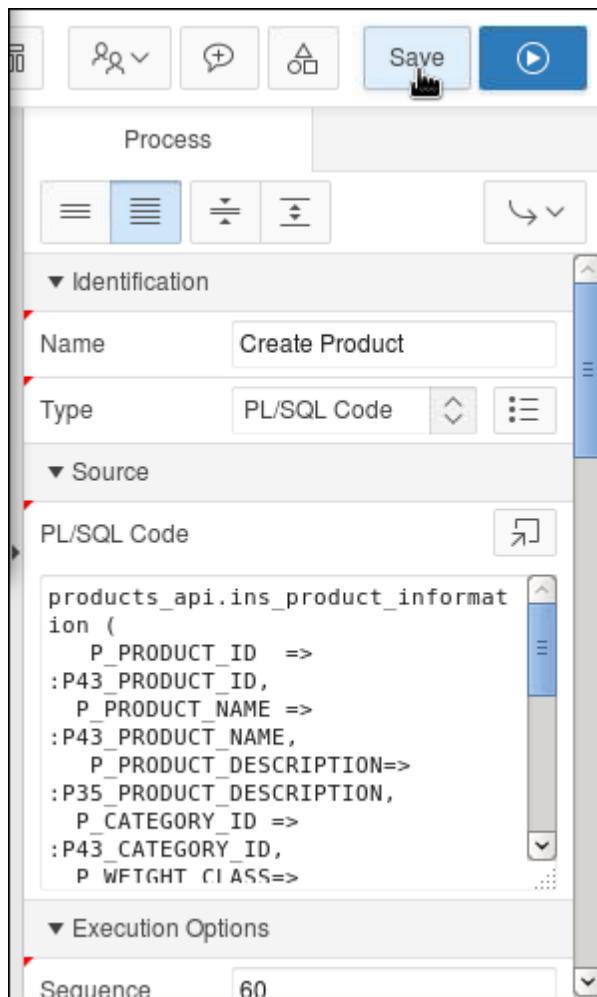


7. In the Property Editor section, enter **Create Product** for **Name** and select **PL/SQL** for **Type**.
8. Enter the following PL/SQL also located in the
`/home/oracle/labs/files/lab_07_01.txt` file for Insert process.

```
products_api.ins_product_information (
    P_PRODUCT_ID => :P43_PRODUCT_ID,
    P_PRODUCT_NAME => :P43_PRODUCT_NAME,
    P_PRODUCT_DESCRIPTION=> :P43_PRODUCT_DESCRIPTION,
    P_CATEGORY_ID => :P43_CATEGORY_ID,
    P_WEIGHT_CLASS=> :P43_WEIGHT_CLASS,
    P_WARRANTY_PERIOD => :P43_WARRANTY_PERIOD,
    P_SUPPLIER_ID => :P43_SUPPLIER_ID,
    P_PRODUCT_STATUS=> :P43_PRODUCT_STATUS,
    P_LIST_PRICE => :P43_LIST_PRICE,
    P_MIN_PRICE => :P43_MIN_PRICE,
    P_CATALOG_URL=> :P43_CATALOG_URL);
```

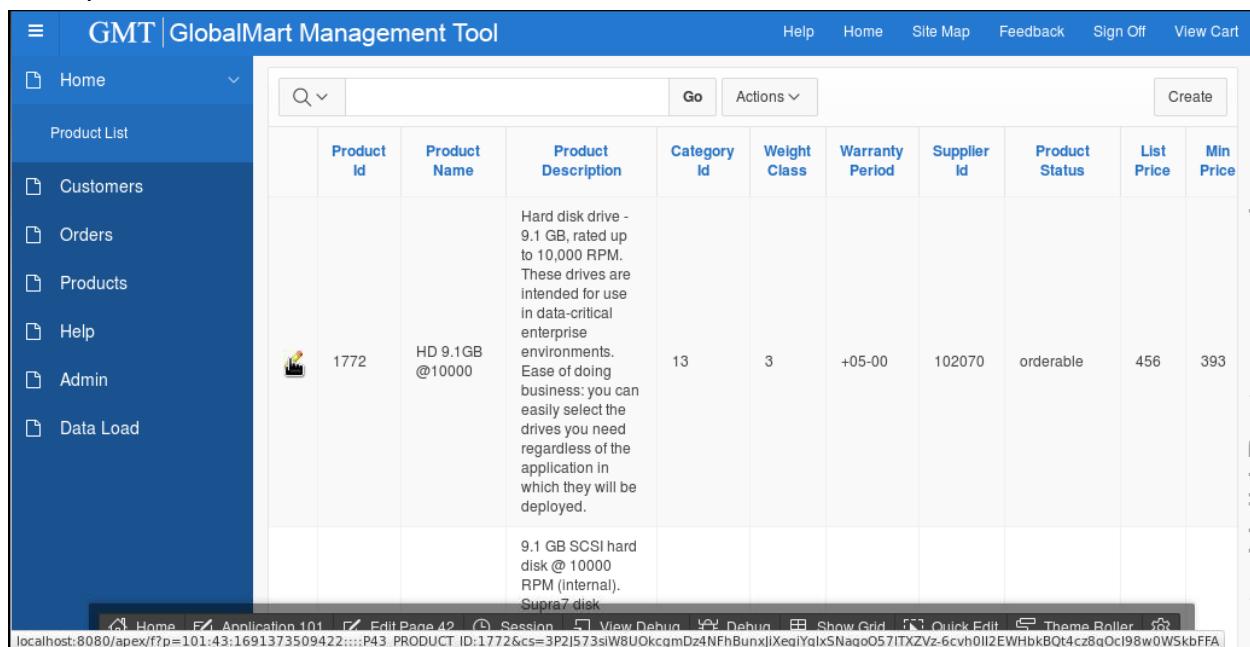
9. Enter Success and Error messages.

10. Select **CREATE** for **When Button Pressed** and click **Save**.



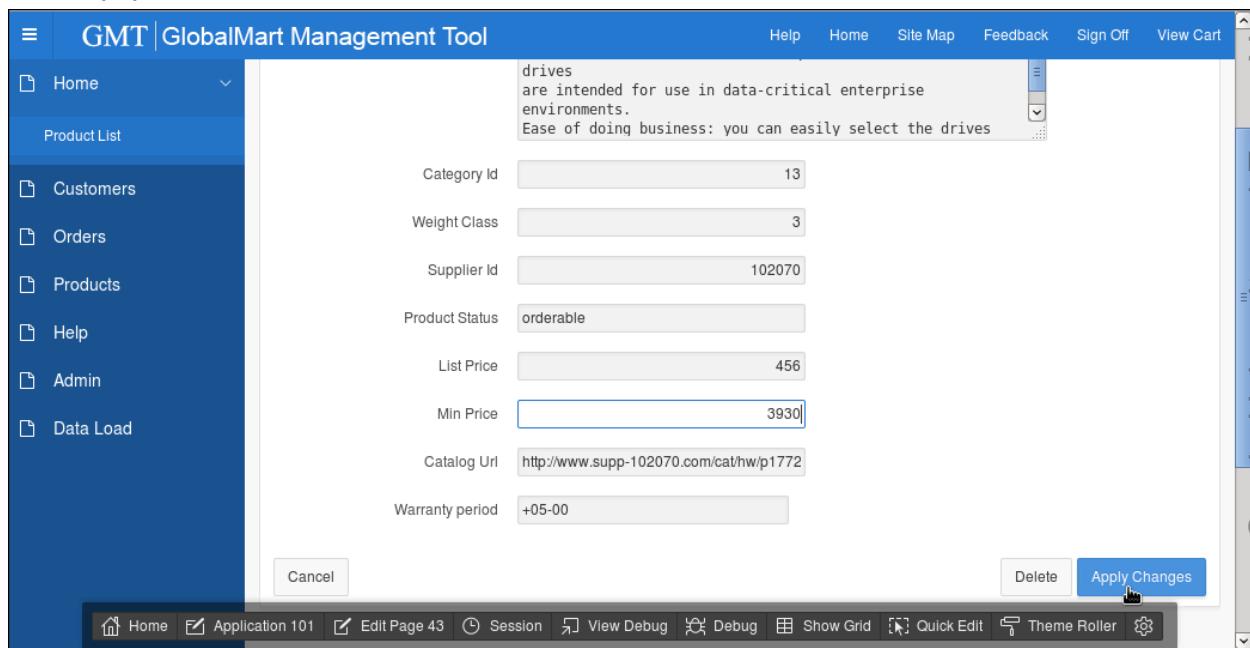
h. You can test these processes by editing and inserting records in the form.

1. Navigate to the **Product List** report page and run the page. Click the **Edit** icon on one of the products.



The screenshot shows the GlobalMart Management Tool interface. On the left is a sidebar with navigation links: Home, Product List (which is selected), Customers, Orders, Products, Help, Admin, and Data Load. The main area displays a table titled "Product List". The table has columns: Product Id, Product Name, Product Description, Category Id, Weight Class, Warranty Period, Supplier Id, Product Status, List Price, and Min Price. A single row is selected, showing Product Id 1772, Product Name "HD 9.1GB @10000", Product Description "Hard disk drive - 9.1 GB, rated up to 10,000 RPM. These drives are intended for use in data-critical enterprise environments. Ease of doing business: you can easily select the drives you need regardless of the application in which they will be deployed.", Category Id 13, Weight Class 3, Warranty Period "+05-00", Supplier Id 102070, Product Status "orderable", List Price 456, and Min Price 393. At the bottom of the table, there are various toolbar icons including Home, Application 101, Edit Page 42, Session, View Debug, Debug, Show Grid, Quick Edit, Theme Roller, and a URL bar showing "localhost:8080/apex/f?p=101:43:1691373509422:::P43_PRODUCT_ID:1772&cs=3P2J573sW8UOkcgmDz4NFhBurxjIXegiYqlxSNagoO57ITXZv2-6cvh0I2EWHbkBQl4cz8qOcl98w0W5kbFFA".

2. Edit the product record and click **Apply Changes**. Your changes will be saved. Similarly, you can add a new record.



The screenshot shows the GlobalMart Management Tool interface. The sidebar is identical to the previous screenshot. The main area shows an edit dialog for a product record. The dialog fields are: Category Id (13), Weight Class (3), Supplier Id (102070), Product Status (orderable), List Price (456), Min Price (3930), Catalog Url (http://www.supp-102070.com/cat/hw/p1772), and Warranty period (+05-00). At the bottom right of the dialog are "Delete" and "Apply Changes" buttons. The URL bar at the bottom of the screen is identical to the one in the previous screenshot.

The screenshot shows the GMT GlobalMart Management Tool interface. On the left is a navigation sidebar with links to Home, Customers, Orders, Products, Help, Admin, and Data Load. The main area is titled "Product List" and contains a table with columns: Product Id, Product Name, Product Description, Category Id, Weight Class, Warranty Period, Supplier Id, Product Status, List Price, and Min Price. Two rows of product data are visible. A green success message at the top right says "Successfully Updated." with a checkmark icon. The bottom of the screen has a toolbar with various icons.

Product Id	Product Name	Product Description	Category Id	Weight Class	Warranty Period	Supplier Id	Product Status	List Price	Min Price
1772	HD 9.1GB @10000	-	13	3	+05-00	102070	orderable	456	3930
2414	HD 9.1GB @10000 /I	9.1 GB SCSI hard disk @ 10000 RPM (internal). Supra7 disk drives are available in 10,000 RPM spindle speeds and capacities of 18GB and 9.1 GB. SCSI and RS-232 interfaces. 9.1 GB hard disk @ 7200 RPM. Universal option	13	3	+05-00	102098	orderable	454	399

- You can view how errors are raised using the API.
- Create a record on the **Product List** page.

This screenshot is identical to the one above, showing the Product List page. However, the status bar at the bottom of the screen displays a red error message: "An error occurred while saving the record. Please try again or contact your system administrator." This indicates that a creation attempt failed.

2. Enter the Product Status as **obsolete** and click **Create**. What do you observe?

The screenshot shows the GlobalMart Management Tool interface. On the left is a sidebar with navigation links: Home, Product List, Customers, Orders, Products, Help, Admin, and Data Load. The main area is a form for creating a product. The fields are as follows:

- Product Description: Testing a product
- Category Id: 12
- Weight Class: 3
- Supplier Id: 102094
- Product Status: obsolete
- List Price: 400
- Min Price: 300
- Catalog Url: www.test.com
- Warranty period: +02-00

At the bottom of the form is a toolbar with various developer tools: Home, Application 101, Edit Page 43, Session, View Debug, Debug, Show Grid, Quick Edit, Theme Roller, and a gear icon.

A modal dialog box is displayed, indicating an error. It contains the following text:

1 error has occurred
Unsuccessful.

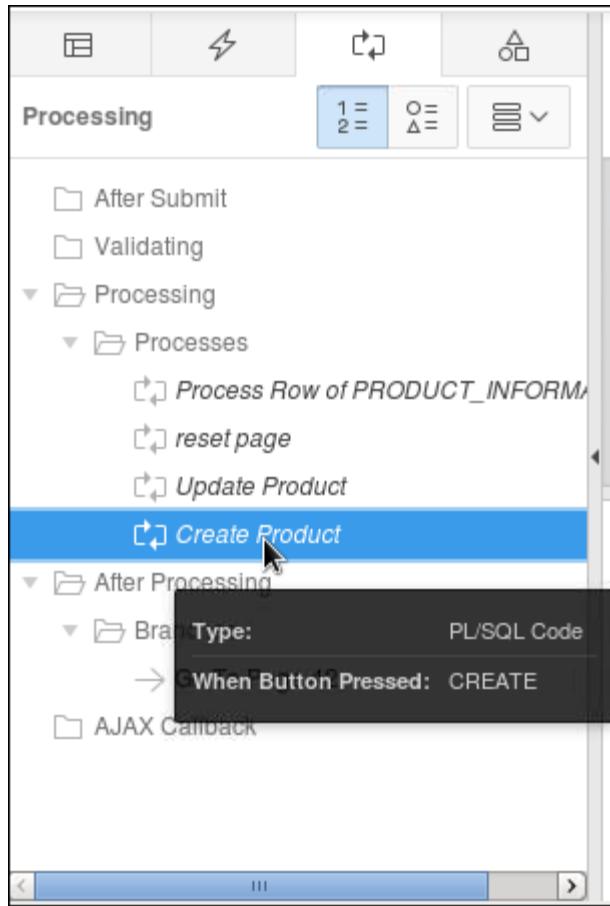
The modal also contains a 'Maintain Product' section with the same fields as the form, but they are not populated with the same values as the main form.

3. The validation checks if the status of the product is obsolete. If it is obsolete, then an error is raised. Enter orderable for Product Status and click **Create**. The record will be processed successfully.

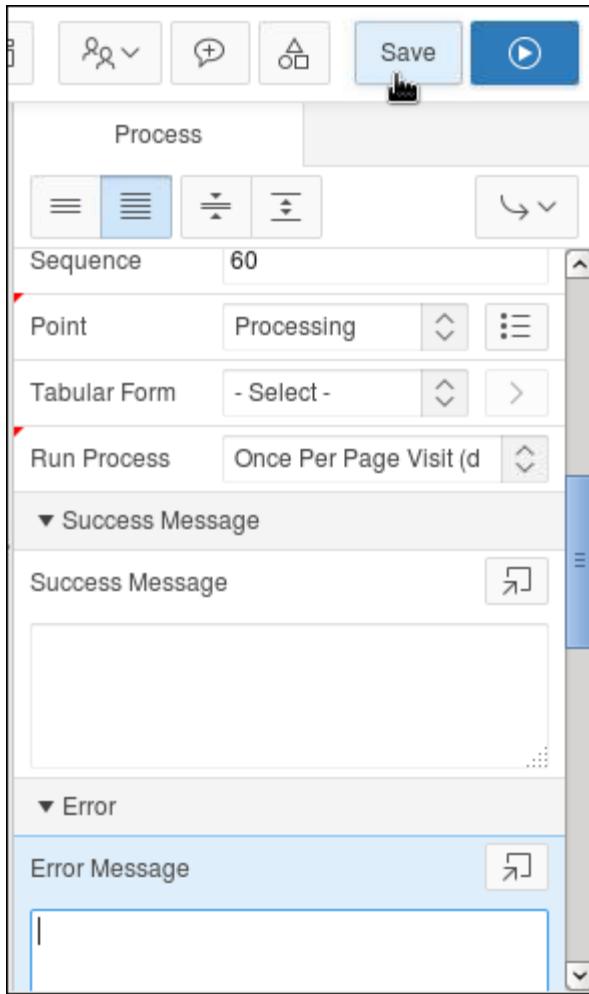
Note: The error message that is being displayed is the Error message that was specified at the time of creating the Create Products process. To check for the error message from the API, delete the error message specified in the process. The Error message from the API will be displayed.

4. Click **Edit Page<n>** on the Developer toolbar.

5. In the Page Processing section, select **Create Products**.



6. Delete the Success and Error messages and click **Save**.



7. Run the page and you will notice that the error message that was defined in the API is displayed.

The screenshot shows the GlobalMart Management Tool interface. The left sidebar has a navigation menu with 'Home' selected. The main content area is titled 'Product List' and contains the following form fields:

Category Id	12
Weight Class	3
Supplier Id	102094
Product Status	obsolete
List Price	400
Min Price	300
Catalog Url	www.test.com
Warranty period	+02-00

At the bottom of the main panel are 'Cancel' and 'Create' buttons. The status bar at the bottom of the screen includes links for Home, Application 101, Edit Page 43, Session, View Debug, Debug, Show Grid, Quick Edit, Theme Roller, and a gear icon.

The screenshot shows a web-based application titled "GMT | GlobalMart Management Tool". The left sidebar contains a navigation menu with links: Home, Product List, Customers, Orders, Products, Help, Admin, and Data Load. The main content area is titled "Maintain Product". It displays a form with the following fields and values:

- Product Name: Test Product
- Product Description: Testing a product
- Category Id: 12
- Weight Class: 3
- Supplier Id: 102094
- Product Status: obsolete

An error message box at the top right states: "1 error has occurred" followed by the Oracle error code: "ORA-20002: You cannot create an obsolete product".

Note: If you have not been able to complete the practice for this lesson, you can import the `sol_07_01.sql` exported application available in the `/home/oracle/labs/solutions` folder.

Practices for Lesson 8: Creating and Using RESTful Web Services

Chapter 8

Practices for Lesson 8: Overview

Practices Overview

In this practice, you create a RESTful Web Service and access the web service through an application.

Practice 8-1: Creating and Using a RESTful Web Service

Overview

In this practice, you create a RESTful Web Service. The RESTful Web Service wizard is a set of pages in the SQL Workshop area that helps you create a RESTful Web Service declaratively.

Assumptions

Tasks

- a. Create a RESTful Web Service that calls a specific SQL statement in your database.
- b. Create a new template to retrieve JSON resultset based on Query One Row with a bind variable.
- c. Create a new template for Retrieving JSON Resultset Based on a Feed.

Practice 8-2: Creating and Consuming a RESTful Web Service

Overview

In this practice, you consume the RESTful Web Service in APEX by creating a database application and a Web Service Reference in the application. You then create a form and report page that uses the web service.

Tasks

- a. Consume the RESTful Web Service in APEX by creating a database application and a Web Service Reference in the application. You then create a form and report page that uses the web service.
- b. Create a Web Service Reference in the RESTful Web Service application. You then create a form and report page that uses the web service.

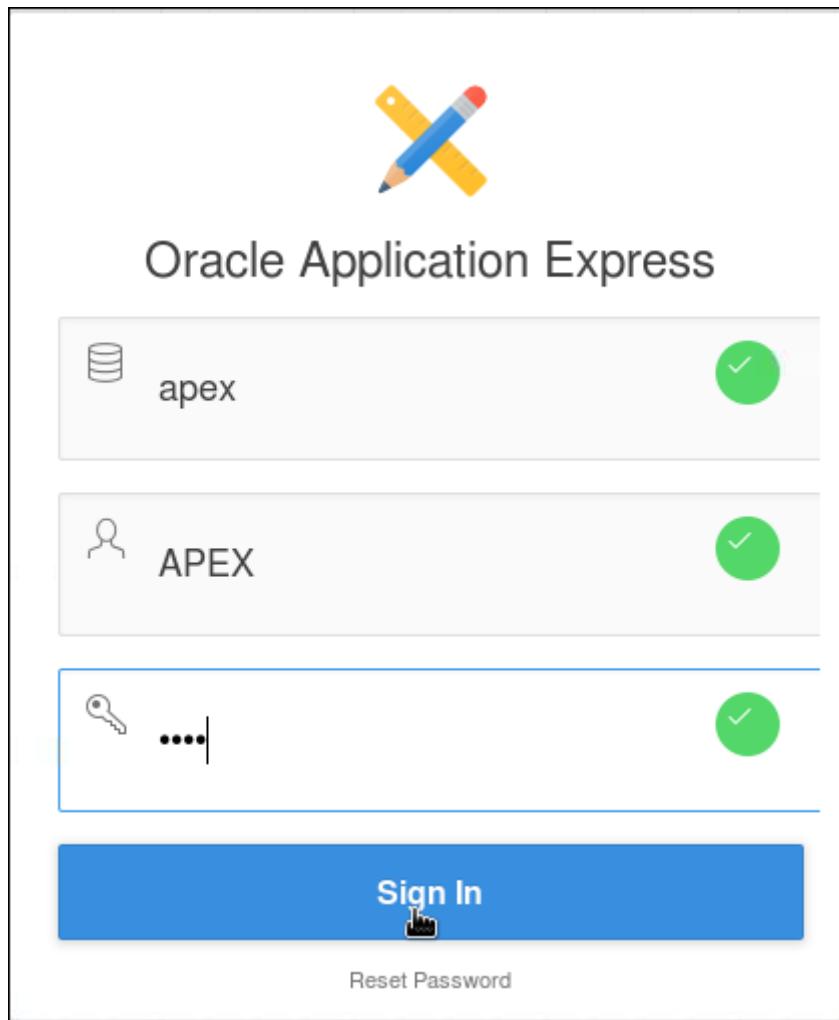
Solution 8-1: Creating and Using a RESTful Web Service

Overview

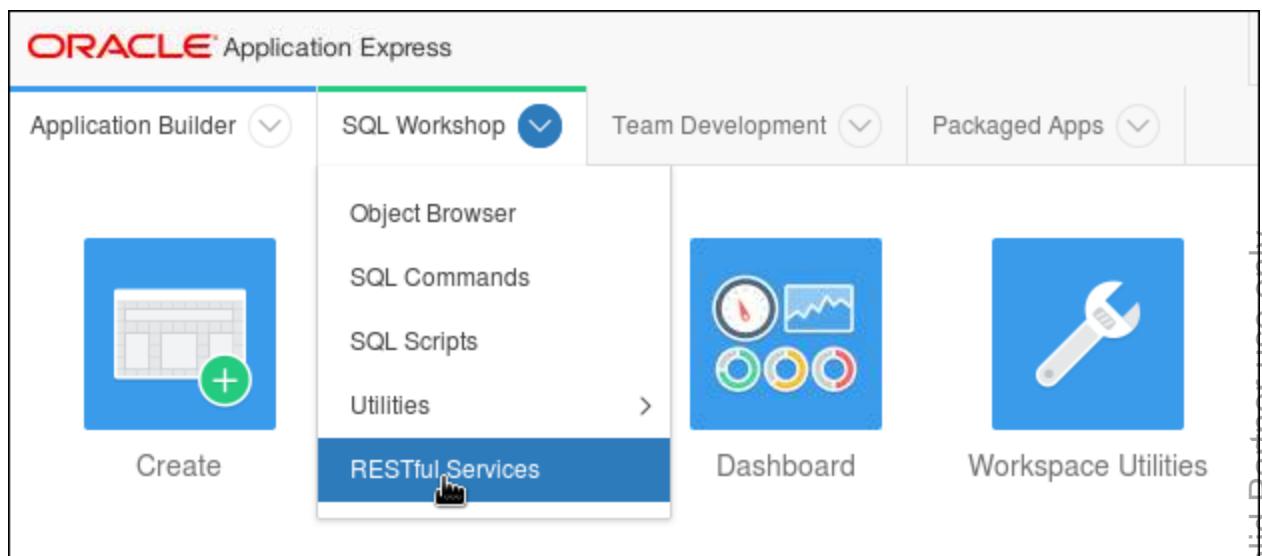
In this practice, you create a RESTful Web Service. The RESTful Web Service wizard is a set of pages in the SQL Workshop area that helps you to create a RESTful Web Service declaratively.

Steps

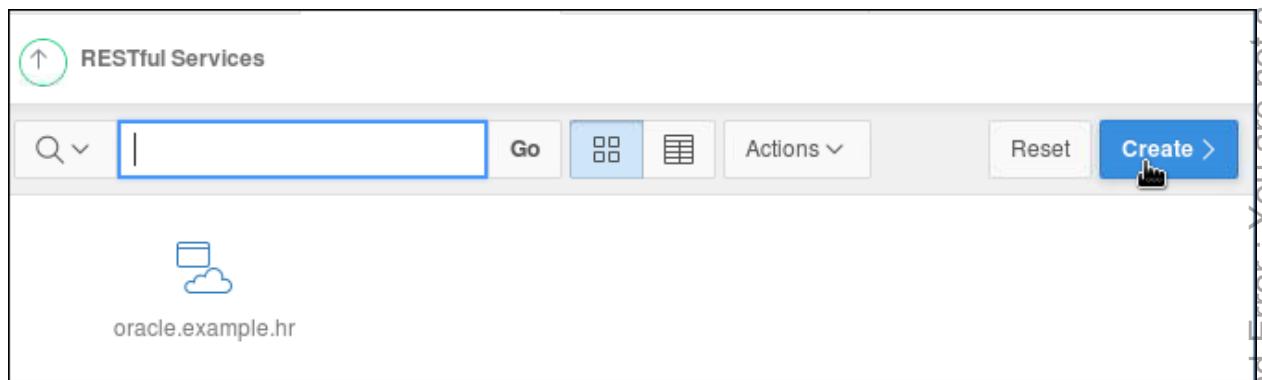
- Create a RESTful Web Service that calls a specific SQL statement in your database.
 - Start Oracle Application Express in the web browser.
 - Enter your login credentials. Refer the EDAG0000.doc for the credentials.



3. Navigate to your workspace home page and select the SQL Workshop tab. Select **RESTful Services**.



4. Click **Create**.



5. Enter **Employees** as the **Name** for the RESTful Service module. Enter the **URI template** as **oehr/** to identify your Uniform Resource Identifier. Select **GET** as the Method. Select **JSON** for **Format**. This identifies the HTTP method to be used for the Resource Handler. For Source, enter the following SQL Query. This is responsible for handling the selected HTTP method. Click **Create Module**.

```
select * from "OEHR".employees
```

RESTful Services > Create RESTful Service Module

Cancel **Create Module**

RESTful Services Module

* Name Employees

URI Prefix

Origins Allowed

Status Published

* Pagination Size 25

Required Privilege - Assign Privilege -

Resource Handler

Add a Resource Handler

Method GET

Source Type Query

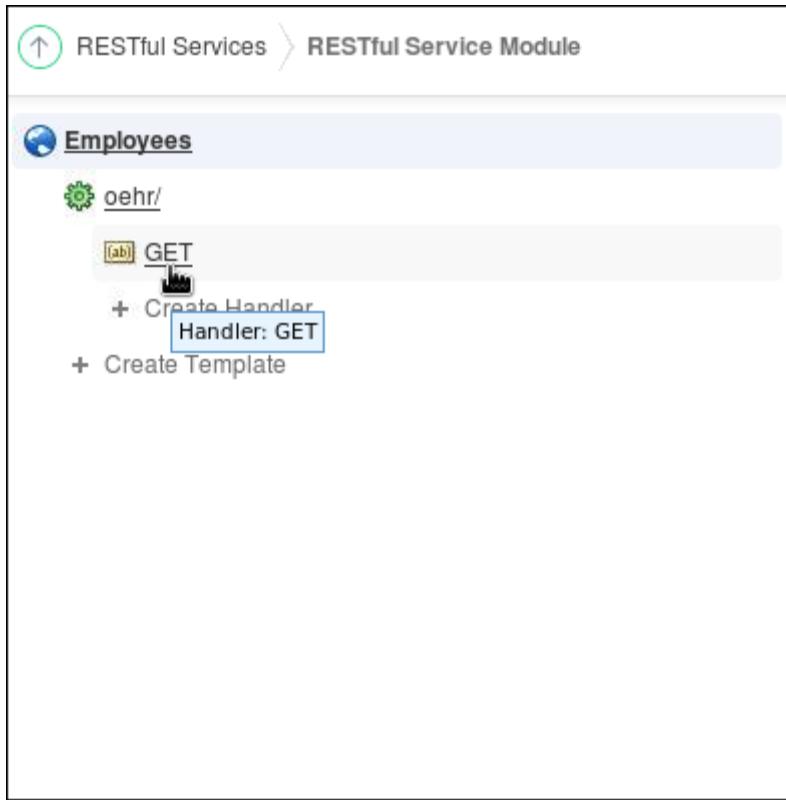
Format JSON

* Source select * from "OEHR".employees

Oracle University and Error : You are not a Valid Partner use only

This screenshot shows the Oracle RESTful Services Module creation interface. At the top, there's a breadcrumb navigation: RESTful Services > Create RESTful Service Module. On the right, there are 'Cancel' and 'Create Module' buttons. Below the header, the title 'RESTful Services Module' is displayed. The form contains several input fields: 'Name' (Employees), 'URI Prefix' (empty), 'Origins Allowed' (empty), 'Status' (Published), 'Pagination Size' (25), and 'Required Privilege' (empty). A 'Resource Handler' section is expanded, showing 'Add a Resource Handler' and a configuration for a GET method with a query source type and JSON format. The 'Source' field contains the SQL query 'select * from "OEHR".employees'. A vertical watermark on the right side of the interface reads 'Oracle University and Error : You are not a Valid Partner use only'.

6. Click the **GET** Method Resource Handler.



7. Change the **Require Secure Access** option from Yes to No. Click **Apply Changes**.

Resource Handler: GET

A resource handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method. Although multiple resource handlers can be defined for a resource template, only one resource handler per HTTP method is permitted.

RESTful Service Module: [?](#)

URI Template: oehr/ [?](#)

Method: GET [?](#)

Source Type: Query [?](#)

Format: JSON [?](#)

Requires Secure Access: No [?](#)

Pagination Size: [?](#)

Source

* Source [?](#)

```
1 select * from "OEHR".employees
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle University and Error : You are not a Valid Partner use only

8. Click **Test** to test the behavior of the RESTful Service Handler.

The screenshot shows the Oracle REST Data Services interface. At the top, there's a toolbar with icons for back, forward, search, and refresh. Below the toolbar is a source editor with the title "Source". It contains a code snippet: "1 select * from \"OEHR\".employees". To the right of the source editor is a gear icon. Below the source editor is a section titled "Example". Underneath "Example" is a "Test" tab. On the "Test" tab, there are two buttons: "Test" (with a play icon) and "Set Bind Variables >". A tooltip below the "Test" button says: "To test the behavior of the RESTful Service Handler, click the Test button. If the RESTful Service Handler uses". On the far right edge of the interface, there is a vertical bar with the text "Oracle University and Error : You are not a Valid Partner use only".

9. The JSON result is displayed. Note the URL, which will be used in Practice 8-2. In this case, it is <http://localhost:8080/apex/2210243963958320/oehr/>
It will differ according to your machine settings.

```
{  
  - next: {  
    $ref: http://localhost:8080/apex/2210243963958320/oehr/?page=1  
  },  
  - items: [  
    - {  
      employee_id: 100,  
      first_name: "Steven",  
      last_name: "King",  
      email: "SKING",  
      phone_number: "515.123.4567",  
      hire_date: "1987-06-17T00:00:00Z",  
      job_id: "SA_REP",  
      salary: 10000,  
      commission_pct: 0.05,  
      department_id: 90,  
      active_yn: "Y"  
    },  
    - {  
      employee_id: 101,  
      first_name: "Neena",  
      last_name: "Kochhar",  
      email: "NKOCHHAR",  
      phone_number: "515.123.4568",  
      hire_date: "1989-09-21T00:00:00Z",  
      job_id: "AD_VP",  
      salary: 17000,  
      manager_id: 100,  
      department_id: 90  
    },  
    - {  
      employee_id: 102,  
      first_name: "Lex",  
      last_name: "De Haan",  
      email: "LDEHAAN",  
      phone_number: "515.123.4569",  
      hire_date: "1993-01-13T00:00:00Z",  
      job_id: "AD_VP"  
    }  
  ]  
}
```

10. Click the Get Handler again and change the format to **CSV**. Click **Apply Changes**.

Resource Handler: GET

Cancel Delete **Apply Changes**

A resource handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method. Although multiple resource handlers can be defined for a resource template, only one resource handler per HTTP method is permitted.

RESTful Service Module: [?](#)

URI Template: **oehr/** [?](#)

Method: **GET** [?](#)

Source Type: **Query** [?](#)

Format: **CSV** [?](#)

Requires Secure Access: **No** [?](#)

Pagination Size: [?](#)

Source

* Source [?](#)

```
1 select * from "OEHR".employees
```

11. Click **Test**.

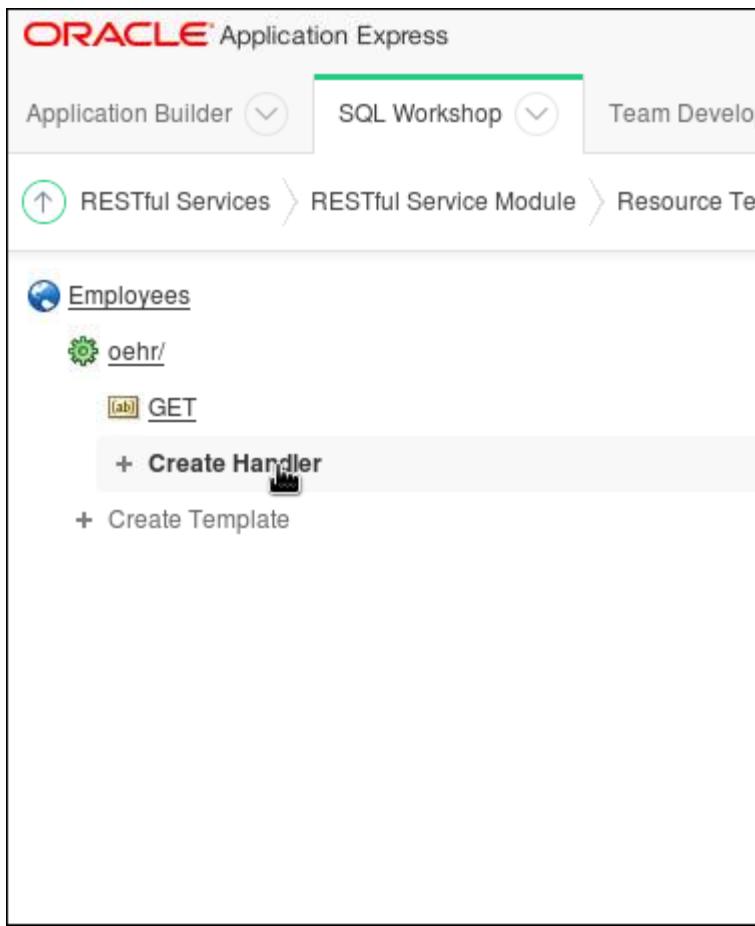
12. You are prompted to save the file which you can then view using a CSV editor. The CSV format resultset is displayed. Make a note of the URI which will be used later while creating a client service.

The screenshot shows a Firefox browser window with a title bar 'Opening'. The main content area displays a message: 'You have chosen to open:' followed by 'which is: CSV document' and 'from: http://localhost:8080'. Below this, a section titled 'What should Firefox do with this file?' contains three options: 'Open with' (selected, pointing to 'gedit (default)'), 'Save File', and 'Do this automatically for files like this from now on.' At the bottom are 'Cancel' and 'OK' buttons, with 'OK' being clicked. The large text area below the dialog box contains a CSV data dump with 128 rows, starting with:

```
100,Steven,King,SKING,515.123.4567,1987-06-17T00:00:00Z,SA REP,10000,0.05,,90,Y  
101,Neena,Kochhar,NKOCHHAR,515.123.4568,1989-09-21T00:00:00Z,AD_VP,17000,,100,90,  
102,Lex,De Haan,LDEHAAN,515.123.4569,1993-01-13T00:00:00Z,AD_VP,17000,,100,90,  
103,Alexander,Hunold,AHUNOLD,590.423.4567,1990-01-03T00:00:00Z,IT_PROG,9000,,102,60,  
104,Bruce,Ernst,BERNST,590.423.4568,1991-05-21T00:00:00Z,IT_PROG,6000,,103,60,  
105,David,Austin,DAUSTIN,590.423.4569,1997-06-25T00:00:00Z,IT_PROG,4800,,103,60,  
106,Valli,Pataballa,VPATABAL,590.423.4560,1998-02-05T00:00:00Z,IT_PROG,4800,,103,60,  
107,Diana,Lorentz,DLORENTZ,590.423.5567,1999-02-07T00:00:00Z,IT_PROG,4200,,103,60,  
108,Nancy,Greenberg,NGREENBE,515.124.4569,1994-08-17T00:00:00Z,FI_MGR,12000,,101,100,  
109,Daniel,Faviet,DFAVIET,515.124.4169,1994-08-16T00:00:00Z,FI_ACCOUNT,9000,,108,100,  
110,John,Chen,JCHEN,515.124.4269,1997-09-28T00:00:00Z,FI_ACCOUNT,8200,,108,100,  
111,Ismael,Sciarra,ISCIARRA,515.124.4369,1997-09-30T00:00:00Z,FI_ACCOUNT,7700,,108,100,  
112,Jose Manuel,Urman,JMURMAN,515.124.4469,1998-03-07T00:00:00Z,FI_ACCOUNT,7800,,108,100,  
113,Luis,Popp,LPOPP,515.124.4567,1999-12-07T00:00:00Z,FI_ACCOUNT,6900,,108,100,  
114,Den,Raphaely,DRAPHEAL,515.127.4561,1994-12-07T00:00:00Z,PU_MAN,11000,,100,30,  
115,Alexander,Khoo,AKHOO,515.127.4562,1995-05-18T00:00:00Z,PU_CLERK,3100,,114,30,  
116,Shelli,Baida,SBAIDA,515.127.4563,1997-12-24T00:00:00Z,PU_CLERK,2900,,114,30,  
117,Sigal,Tobias,STOBIAS,515.127.4564,1997-07-24T00:00:00Z,PU_CLERK,2800,,114,30,  
118,Guy,Himuro,GHIMURO,515.127.4565,1998-11-15T00:00:00Z,PU_CLERK,2600,,114,30,  
119,Karen,Colmenares,KCOLMENA,515.127.4566,1999-08-10T00:00:00Z,PU_CLERK,2500,,114,30,  
120,Matthew,Weiss,MWEISS,650.123.1234,1996-07-18T00:00:00Z,ST_MAN,8000,,100,50,  
121,Adam,Fripp,AFRIPP,650.123.2234,1997-04-10T00:00:00Z,ST_MAN,8200,,100,50,  
122,Payam,Kaufling,PKAUFLIN,650.123.3234,1995-05-01T00:00:00Z,ST_MAN,7900,,100,50,  
123,Shanta,Vollman,SVOLLMAN,650.123.4234,1997-10-10T00:00:00Z,ST_MAN,6500,,100,50,  
124,Kevin,Mourgos,KMOURGOS,650.123.5234,1999-11-16T00:00:00Z,ST_MAN,5800,,100,50,  
125,Julia,Nayer,JNAYER,650.124.1214,1997-07-16T00:00:00Z,ST_CLERK,3200,,120,50,  
126,Irene,Mikkilineni,IMIKKILI,650.124.1224,1998-09-28T00:00:00Z,ST_CLERK,2700,,120,50,  
127,James,Landry,JLANDRY,650.124.1334,1999-01-14T00:00:00Z,ST_CLERK,2400,,120,50,  
128,Steven,Markle,SMARKLE,650.124.1434,2000-03-08T00:00:00Z,ST_CLERK,2200,,120,50,
```

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 INS

13. Click **Create Handler** to create a Resource Handler for the POST method.



14. Select **POST** for **Method**, **Source Type** as **PL/SQL**, and **Mime Types Allowed** as **application/json**. Select **No** for **Requires Secure Access**.

Resource Handler:

A resource handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method. Although multiple resource handlers can be defined for a resource template, only one resource handler per HTTP method is permitted.

RESTful Service Module: [?](#)

URI Template: `oehr/` [?](#)

Method: `POST` [?](#)

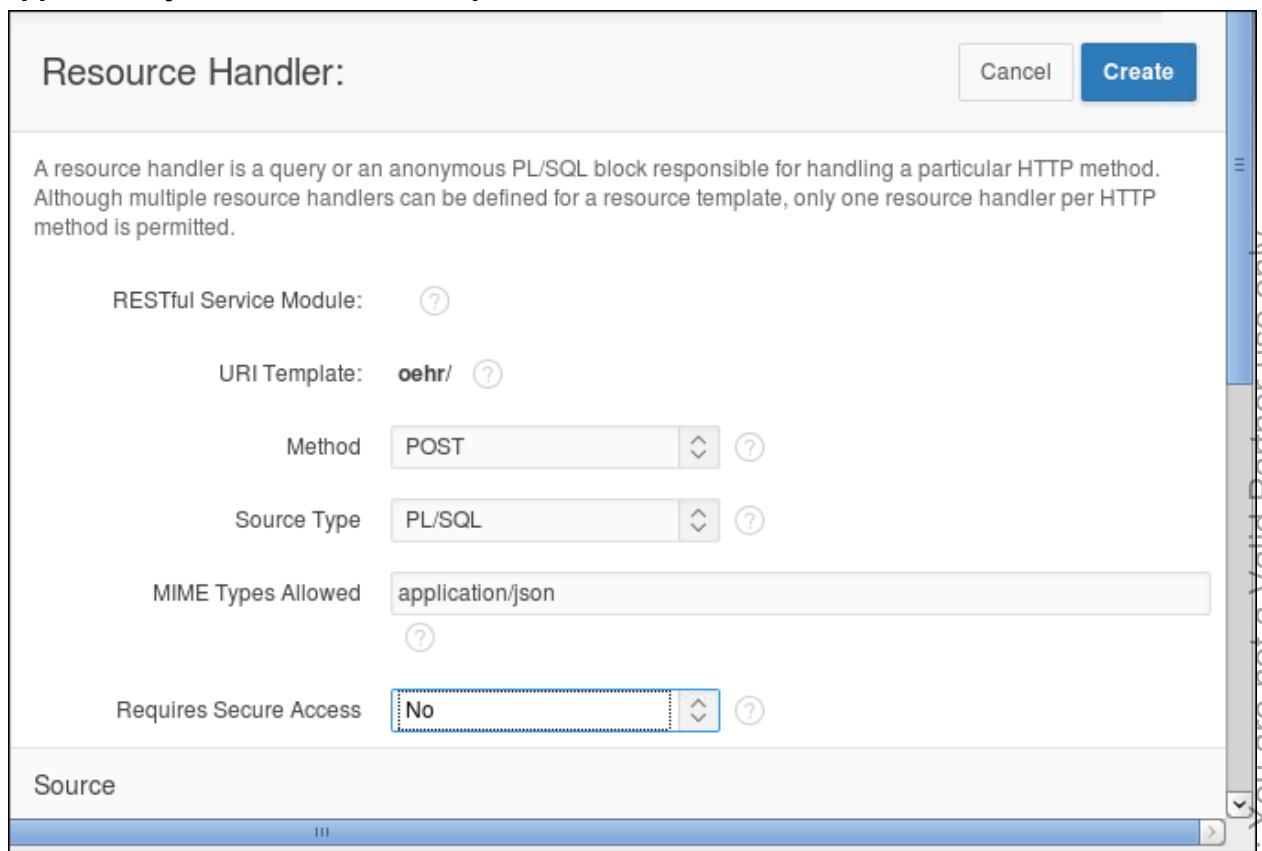
Source Type: `PL/SQL` [?](#)

MIME Types Allowed: `application/json` [?](#)

Requires Secure Access: `No` [?](#)

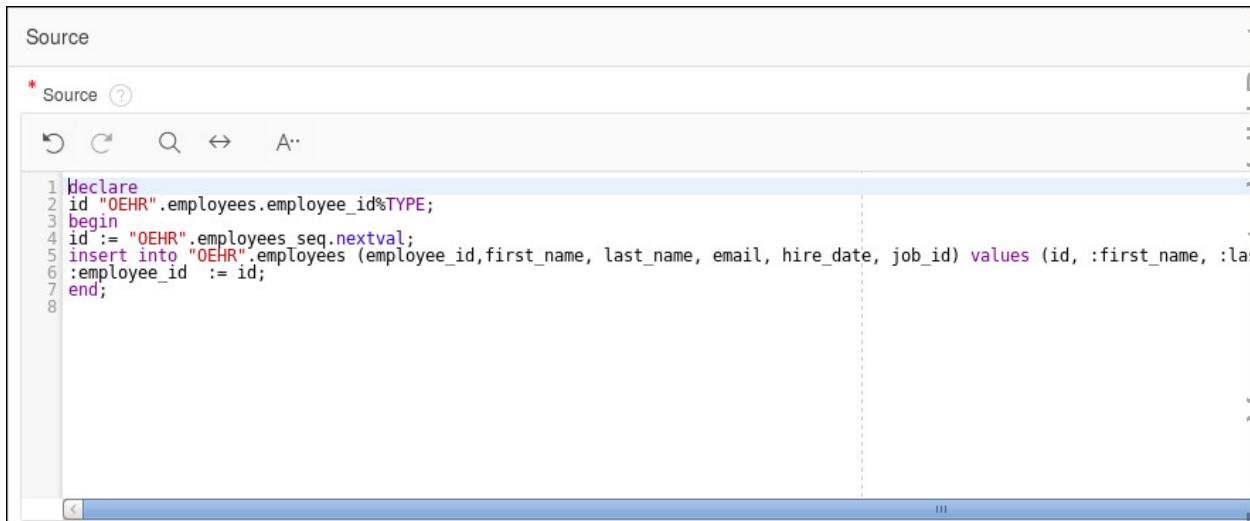
Source

Oracle University and Error : You are not a Valid Partner use only



15. In the **Source** section, enter the following PL/SQL code (also located in the `/home/oracle/labs/files/lab_08_01.txt` file) to create a row by inserting values into the `employees` table. Click **Create**.

```
declare
  id "OEHR".employees.employee_id%TYPE;
begin
  id := "OEHR".employees_seq.nextval;
  insert into "OEHR".employees (employee_id,first_name,
last_name, email, hire_date, job_id) values (id, :first_name,
:last_name, :email, to_date(:hire_date, 'DD-MM-YYYY'), :job_id);
  :employee_id := id;
end;
```



16. Scroll down the page and click **Create Parameter** to add an OUT parameter to the handler that will return the newly created employee ID.



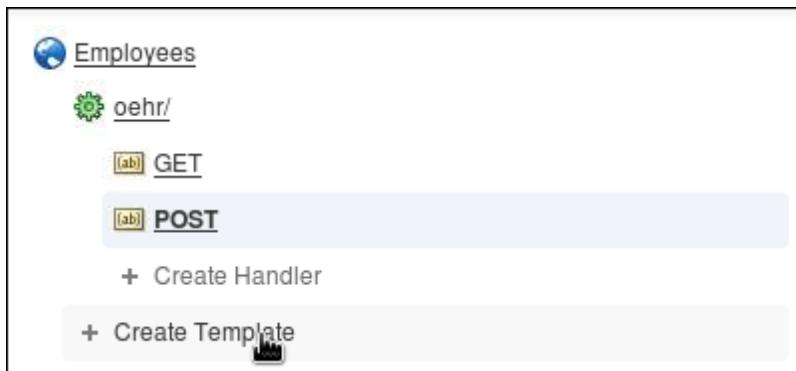
17. Enter **id** for **Name** and **id** for **Bind Variable Name**. Select **OUT** for **Access Method**, **HTTP Header** for **Source Type**, and **String** for **Parameter Type**. Click **Create**.

The screenshot shows the 'Resource Handler Parameter' configuration screen. At the top right are 'Cancel' and 'Create' buttons, with 'Create' being clicked. The main area contains the following fields:

- RESTful Service Module:** Employees
- URI Template:** oehr/
- Resource Handler:** POST
- Handler Source:** A PL/SQL block:

```
declare
id "OEHR".employees.employee_id%TYPE;
begin
id := "OEHR".employees_seq.nextval;
insert into "OEHR".employees (employee_id,first_name
hire_date,job_id) values (id,:first_name,:last_name,:e
to_date(:hire_date,'DD-MM-YYYY'),:job_id);
```
- * Name:** id
- Bind Variable Name:** id
- Access Method:** OUT

- b. Create a new template to retrieve JSON resultset based on Query One Row with a bind variable.
1. Create a new template by clicking **Create Template**.

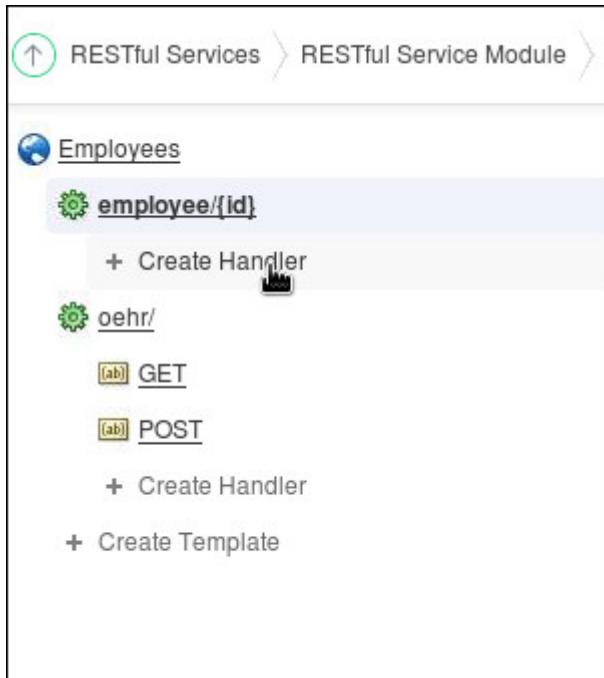


2. Enter employee/{id} for URI template. Click **Create**.

The dialog is titled "URI Template:" with "Cancel" and "Create" buttons. It has fields for "RESTful Service Module" (Employees), "URI Template" (employee/{id}), "Priority" (0), and Entity Tag (Secure HASH). A "Example" button is also present.

RESTful Service Module:	Employees
* URI Template	employee/{id}
* Priority	0
Entity Tag	Secure HASH

3. Click **Create Handler**.



4. Select **Query One Row** for Source Type and select **No** for Requires Secure Access. Enter the following SQL statement (also located in the /home/oracle/labs/files/lab_08_01.txt file) for Source and click **Create**.

```
select * from "OEHR".employees where employee_id=:id
```

Resource Handler:

A resource handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method. Although multiple resource handlers can be defined for a resource template, only one resource handler per HTTP method is permitted.

RESTful Service Module: [?](#)

URI Template: **employee/{id}** [?](#)

Method: **GET** [?](#)

Source Type: **Query One Row** [?](#)

Requires Secure Access: **No** [?](#)

Pagination Size: [?](#)

Source

* Source [?](#)

```
1 select * from "OEHR".employees where employee_id=:id
```

5. Click **Create Parameter**.

Parameters [Create Parameter >](#)

No Parameters defined

[apex_admin](#) [apex](#) [en](#) Application Express 5.0.0.00.31

6. Enter **id** for **Name** and **Bind Variable Name**. Select Source Type as **HTTP Header**. Click **Create**.

The screenshot shows a dialog box titled "Resource Handler Parameter". At the top right are "Cancel" and "Create" buttons, with "Create" being highlighted. The main area contains the following fields:

- RESTful Service: Employees (with a help icon)
- Module:
- URI Template: employee/{id} (with a help icon)
- Resource Handler: GET (with a help icon)
- Handler Source:
select * from "OEHR".employees where employee_id=:id (with a help icon)
- * Name: id (with a help icon)
- Bind Variable Name: id (with a help icon)
- Access Method: IN (with a help icon)

7. **Note:** The Source Type URI is not being displayed and hence you want to edit the Resource Handler Parameter that you just created. Select **id** from the Parameter list and select URI for Source Type. Click **Apply Changes**.

The screenshot shows the Oracle RESTful Service Handler configuration interface. At the top, there is a table titled "Parameters" with columns: Name, Bind Variable Name, Access Method, Source Type, and Parameter Type. A row is selected with the value "id" in the Bind Variable Name column.

Below the table, a modal dialog titled "Resource Handler Parameter" is open. It contains fields for RESTful Service Module ("Employees"), URI Template ("employee/{id}"), Resource Handler ("GET"), Handler Source (SQL query), and parameters like Name ("id") and Bind Variable Name ("id"). The "Source Type" field is currently set to "URI".

The "Apply Changes" button in the dialog is highlighted with a mouse cursor.

8. Click **Set Bind Variables**.

A modal dialog titled "Test" is shown. It contains a "Test" button and a "Set Bind Variables" button. Below the buttons, there is a note about testing the RESTful Service Handler.

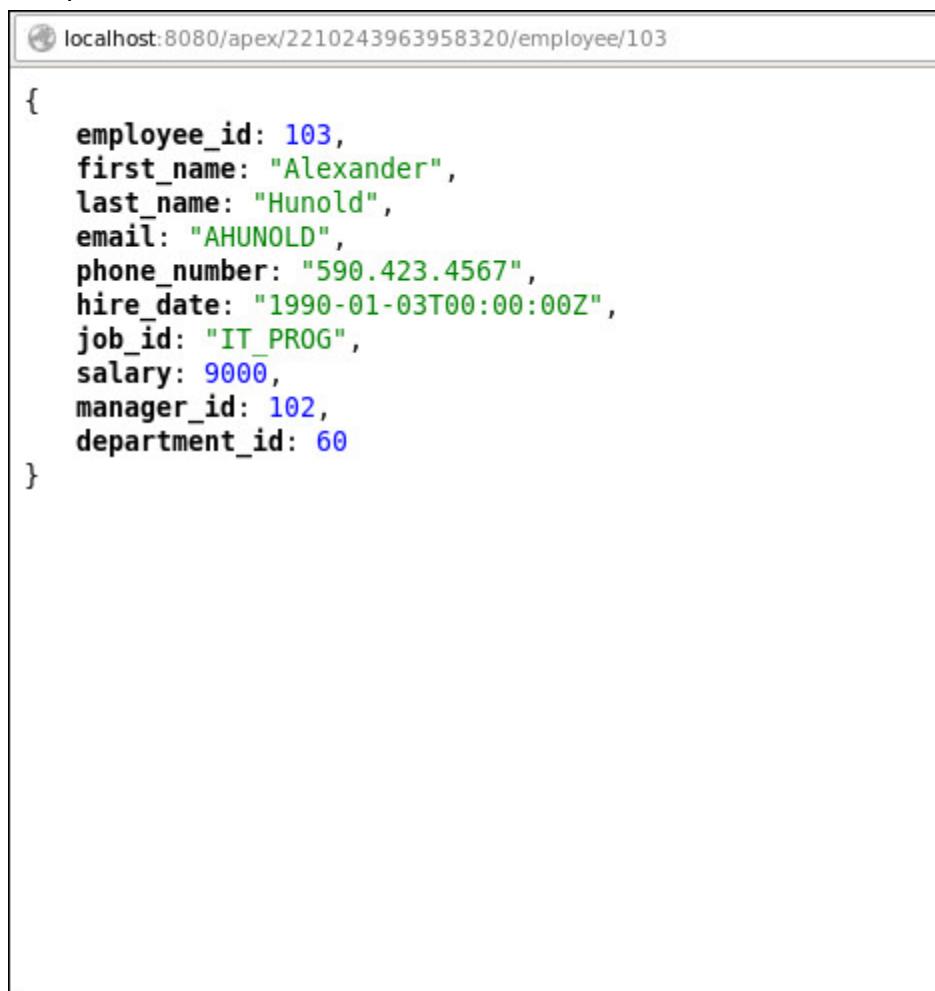
9. Enter 103 or any Employee ID for the Value of the :ID Bind Variable and click **Test**.

The screenshot shows a dialog box for testing a database query. At the top right are 'Cancel' and 'Test' buttons, with 'Test' being highlighted. Below the buttons is a section labeled 'Source' containing a code snippet. A question mark icon is in the top right corner of this section. The code is:

```
* Source select * from "OEHR".employees where employee_id=:id
```

Below the source code is a table titled 'Bind Variable' with one row. The table has two columns: 'Bind Variable' and 'Value'. The 'Bind Variable' column contains ':ID' and the 'Value' column contains '103'. The table has a footer row with the text '1 - 1'.

10. All the information for employee_id is displayed, which is the variable passed as the final portion of the URI. Close this window.



A screenshot of a web browser window. The address bar shows the URL: localhost:8080/apex/2210243963958320/employee/103. The main content area displays a JSON object representing an employee record:

```
{  
    employee_id: 103,  
    first_name: "Alexander",  
    last_name: "Hunold",  
    email: "AHUNOLD",  
    phone_number: "590.423.4567",  
    hire_date: "1990-01-03T00:00:00Z",  
    job_id: "IT_PROG",  
    salary: 9000,  
    manager_id: 102,  
    department_id: 60  
}
```

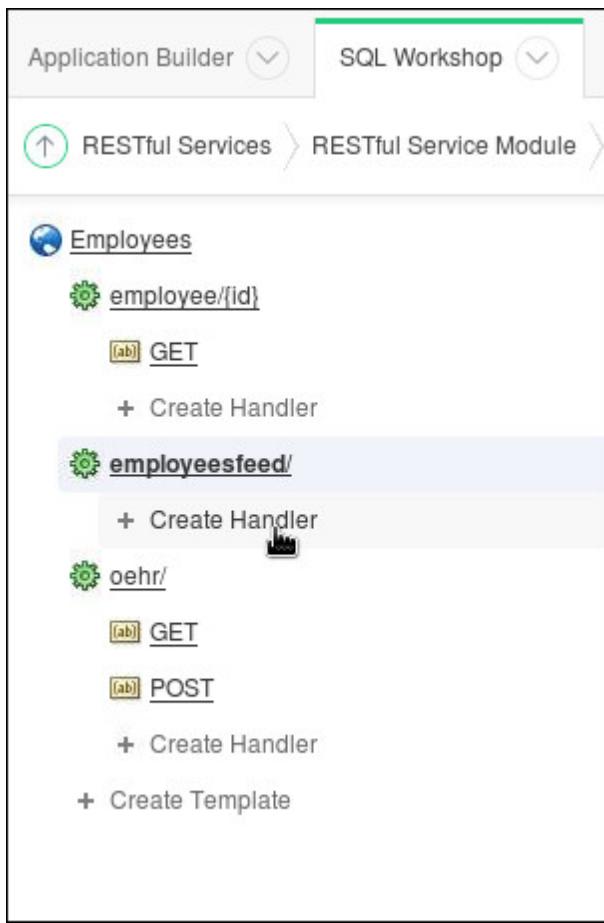
- c. Create a new template for Retrieving JSON Resultset Based on a Feed. Here, you create the `employeesfeed/` RESTful service, which selects the `employee_id` and the `first_name` values in the `employees` table and displays them as a feed.
1. Create a new template by clicking **Create Template**.

The screenshot shows the Oracle Application Express interface. The top navigation bar has tabs for Application Builder, SQL Workshop, and Team Development. Below that, a breadcrumb trail shows RESTful Services > RESTful Service Module > Resource Temp. On the left, there's a sidebar with a globe icon and the text "Employees". Under "Employees", there are several entries: "employee/{id}" with a "GET" method, "oehr/" with "GET" and "POST" methods, and a "+ Create Template" option. The "Create Template" option is highlighted with a mouse cursor.

2. Enter `employeesfeed/` for URI template. Click **Create**.

The screenshot shows a dialog box titled "URI Template:". It contains fields for "RESTful Service Module" set to "Employees", "URI Template" set to "employeesfeed/", "Priority" set to "0", and "Entity Tag" set to "Secure HASH". There are "Cancel" and "Create" buttons at the top right, with the "Create" button being clicked.

3. Click **Create Handler**.



4. Select **Feed** for Source Type and select **No** for Requires Secure Access. Enter the following SQL statement (also located in the /home/oracle/labs/files/lab_08_01.txt file) for **Source** and click **Create**.

```
select employee_id, first_name from "OEHR".employees  
order by employee_id, first_name
```

Resource Handler:

A resource handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method. Although multiple resource handlers can be defined for a resource template, only one resource handler per HTTP method is permitted.

RESTful Service Module: [?](#)

URI Template: **employeesfeed/** [?](#)

Method: **GET** [?](#)

Source Type: **Feed** [?](#)

Requires Secure Access: **No** [?](#)

Pagination Size: [?](#)

Source

* Source [?](#)

```
1 select employee_id, first_name from "OEHR".employees  
2 order by employee_id, first_name
```

5. Click **Create Template**.

The screenshot shows a list of RESTful service modules under the 'Employees' module. The 'employeesfeed/' entry is selected, and its details are shown below. The 'Create Template' button is highlighted with a mouse cursor.

- Employees
- employee/{id}
- [ab] GET
- + Create Handler
- employeesfeed/
- [ab] GET
- + Create Handler
- oehr/
- [ab] GET
- [ab] POST
- + Create Handler
- + Create Template

6. Enter employeesfeed/{id} for URI template. Click **Create**.

The dialog box for creating a new URI template. The 'RESTful Service Module' is set to 'Employees'. The 'URI Template' field contains 'employeesfeed/{id}'. The 'Priority' field is set to '0'. The 'Entity Tag' field is set to 'Secure HASH'. The 'Create' button is highlighted with a mouse cursor.

URI Template:

RESTful Service Module: Employees

* URI Template: employeesfeed/{id}

* Priority: 0

Entity Tag: Secure HASH

Example

Cancel Create

7. Click Create Handler.

The screenshot shows the Oracle RESTful Services interface. The top navigation bar includes a back arrow, the text 'RESTful Services', a right arrow, 'RESTful Service Module', another right arrow, and 'Resources'. Below this, under the heading 'Employees', there are several resource entries:

- employee/{id}**: Includes a 'GET' method and a '+ Create Handler' link.
- employeesfeed/**: Includes a 'GET' method and a '+ Create Handler' link.
- employeesfeed/{id}**: This entry is highlighted with a light blue background. It includes a '+ Create Handler' link and a small icon resembling a trash can or delete button.
- oehr/**: Includes a 'GET' method, a 'POST' method, and '+ Create Handler' and '+ Create Template' links.

8. Select **Feed** for Source Type and select **No** for Requires Secure Access. Enter the following SQL statement (also located in the /home/oracle/labs/files/lab_08_01.txt file) for Source and click **Create**.

```
select employee_id, first_name from "OEHR".employees  
where employee_id=:id
```

Resource Handler:

A resource handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method. Although multiple resource handlers can be defined for a resource template, only one resource handler per HTTP method is permitted.

RESTful Service Module: [?](#)

URI Template: **employeesfeed/{id}** [?](#)

Method: GET [?](#)

Source Type: Feed [?](#)

Requires Secure Access: No [?](#)

Pagination Size: [?](#)

Source

* Source [?](#)

```
1 select employee_id, first_name from "OEHR".employees  
2 where employee_id=:id
```

9. Click **GET** method of the `employeesfeed/` URI template type.

The screenshot shows the Oracle RESTful Services interface. The navigation bar at the top says "RESTful Services > RESTful Service Module > Resource Template". Below this, there's a tree view of resource types:

- Employees**:
 - `employee/{id}`:
 - [ab] **GET**
 - + Create Handler
 - `employeesfeed/`:
 - [ab] **GET** (with a cursor icon over it)
 - + Create Handler
 - `employeesfeed/{id}`:
 - [ab] **GET**
 - + Create Handler
 - `oehr/`:
 - [ab] **GET**
 - [ab] **POST**
 - + Create Handler
 - + Create Template

10. Click **Test**.

The screenshot shows the Oracle RESTful Service Handler interface. At the top, there's a 'Source' tab with a code editor containing the following SQL query:

```
1 select employee_id, first_name from "OEHR".employees
2 order by employee_id, first_name
```

Below the source code, there's an 'Example' section with a play icon. Underneath, there's a 'Test' section with two buttons: 'Test' (which has a hand cursor icon) and 'Set Bind Variables >'. A tooltip for the 'Test' button provides instructions: 'To test the behavior of the RESTful Service Handler, click the Test button. If the RESTful Service Handler uses parameters, click the Set Bind Variables button to set test values for the parameters. Before testing, ensure that you have saved all changes to this page by clicking Apply Changes. For better results, ensure you have a JSON Viewer installed in your browser.'

The data returned is displayed for all employees in the EMPLOYEES table. The URI contains the link to the individual record. This link is actually using the **employeesfeed/{id}** Resource Template.

```
{  
  - next: {  
    $ref: http://localhost:8080/apex/2210243963958320/employeesfeed/?page=1  
  },  
  - items: [  
    - {  
      - uri: {  
        $ref: http://localhost:8080/apex/2210243963958320/employeesfeed/100  
      },  
      employee_id: 100,  
      first_name: "Steven"  
    },  
    - {  
      - uri: {  
        $ref: http://localhost:8080/apex/2210243963958320/employeesfeed/101  
      },  
      employee_id: 101,  
      first_name: "Neena"  
    },  
    - {  
      - uri: {  
        $ref: http://localhost:8080/apex/2210243963958320/employeesfeed/102  
      },  
      employee_id: 102,  
      first_name: "Lex"  
    },  
    - {  
      - uri: {  
        $ref: http://localhost:8080/apex/2210243963958320/employeesfeed/103  
      },  
      employee_id: 103,  
      first_name: "Alexander"  
    },  
    - {  
      - uri: {  
        $ref: http://localhost:8080/apex/2210243963958320/employeesfeed/104  
      }  
    }  
  ]  
}
```

localhost:8080/apex/2210243963958320/employeesfeed/100

11. Click the URL. It will take you to the individual employee record.

```
{  
  - next: {  
    $ref: http://localhost:8080/apex/2210243963958320/employeesfeed/100?page=1  
  },  
  - items: [  
    - {  
      - uri: {  
        $ref: http://localhost:8080/apex/2210243963958320/employeesfeed/100  
      },  
      employee_id: 100,  
      first_name: "Steven"  
    }  
  ]  
}
```

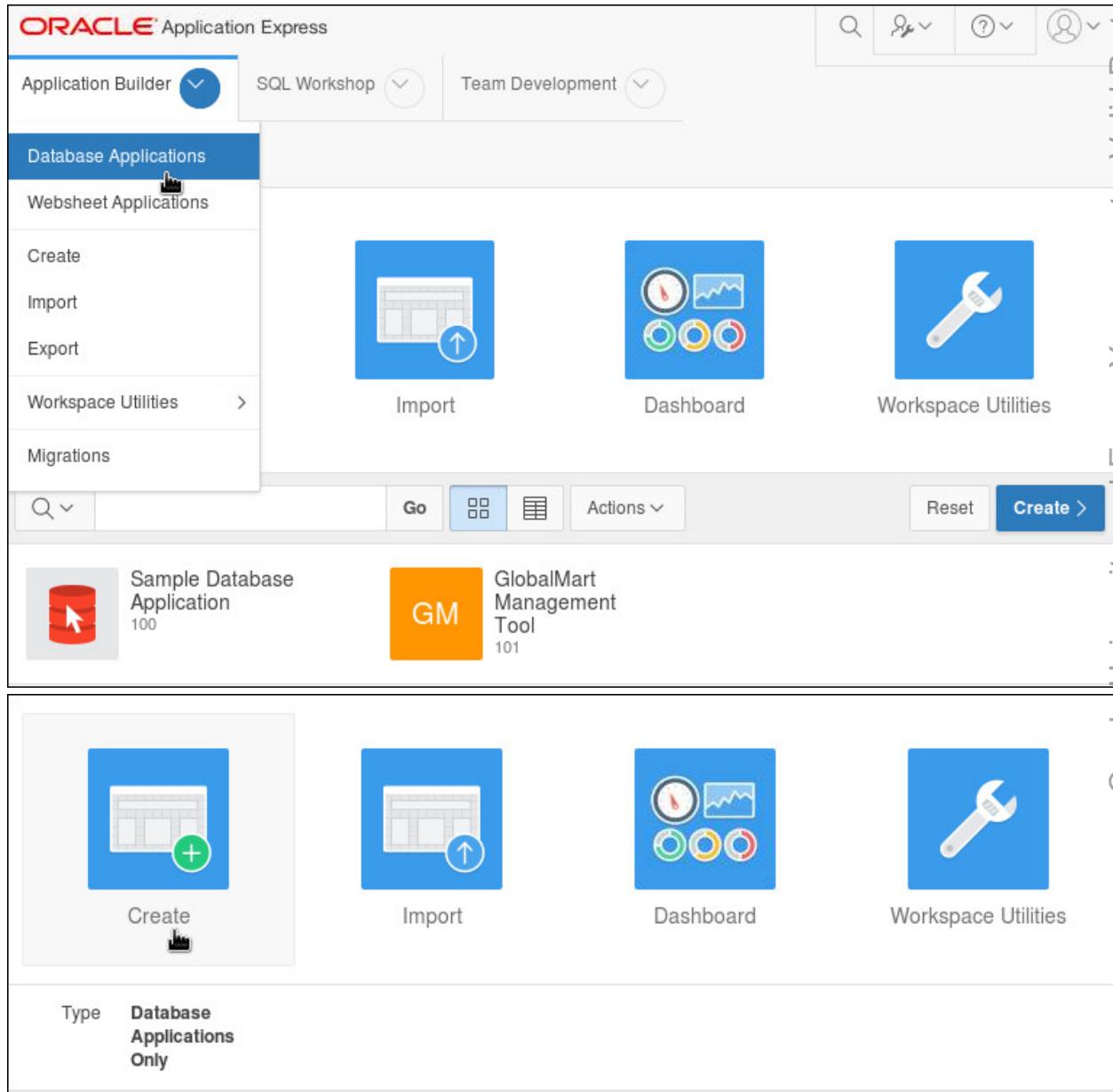
Solution 8-2: Creating and Consuming a RESTful Web Service

Overview

In this practice, you consume the RESTful Web Service in APEX by creating a database application and a Web Service Reference in the application. You then create a form and report page that uses the web service.

Steps

- Consume the RESTful Web Service in APEX by creating a database application and a Web Service Reference in the application. You then create a form and report page that uses the web service.
1. Navigate to your workspace home page and create a new application by selecting **Application Builder > Database Application > Create**.



2. Click Next.

Create an Application

What type of application would you like to create?

 Desktop  Mobile  Websheet  Packaged Application

[From a spreadsheet](#) · [Copy an existing application](#)

 About

 Cancel

 Next >

3. Enter the Name as **Security Services App** and click **Create Application**.

The screenshot shows the 'Create an Application' dialog box. At the top, it says 'Create an Application'. Below that, there is a progress bar with a blue circle at the first step, labeled 'Name'. The steps are: Name, User Interface, Schema, Name, Application, Theme, and Theme Style. The 'Name' step is active. The 'User Interface' dropdown is set to 'Desktop'. The 'Schema' dropdown is set to 'OEHR'. The 'Name' input field contains 'Security Services App'. The 'Application' dropdown is set to '105'. The 'Theme' dropdown is set to 'Universal Theme (42)'. The 'Theme Style' dropdown is set to 'Vita'. At the bottom left are 'Cancel' and 'Next >' buttons. At the bottom right are 'Create Application' and 'Next >' buttons. A vertical watermark on the right side reads 'Oracle University and Error : You are not a Valid Partner use only'.

4. Click **Create Application**.

Create an Application

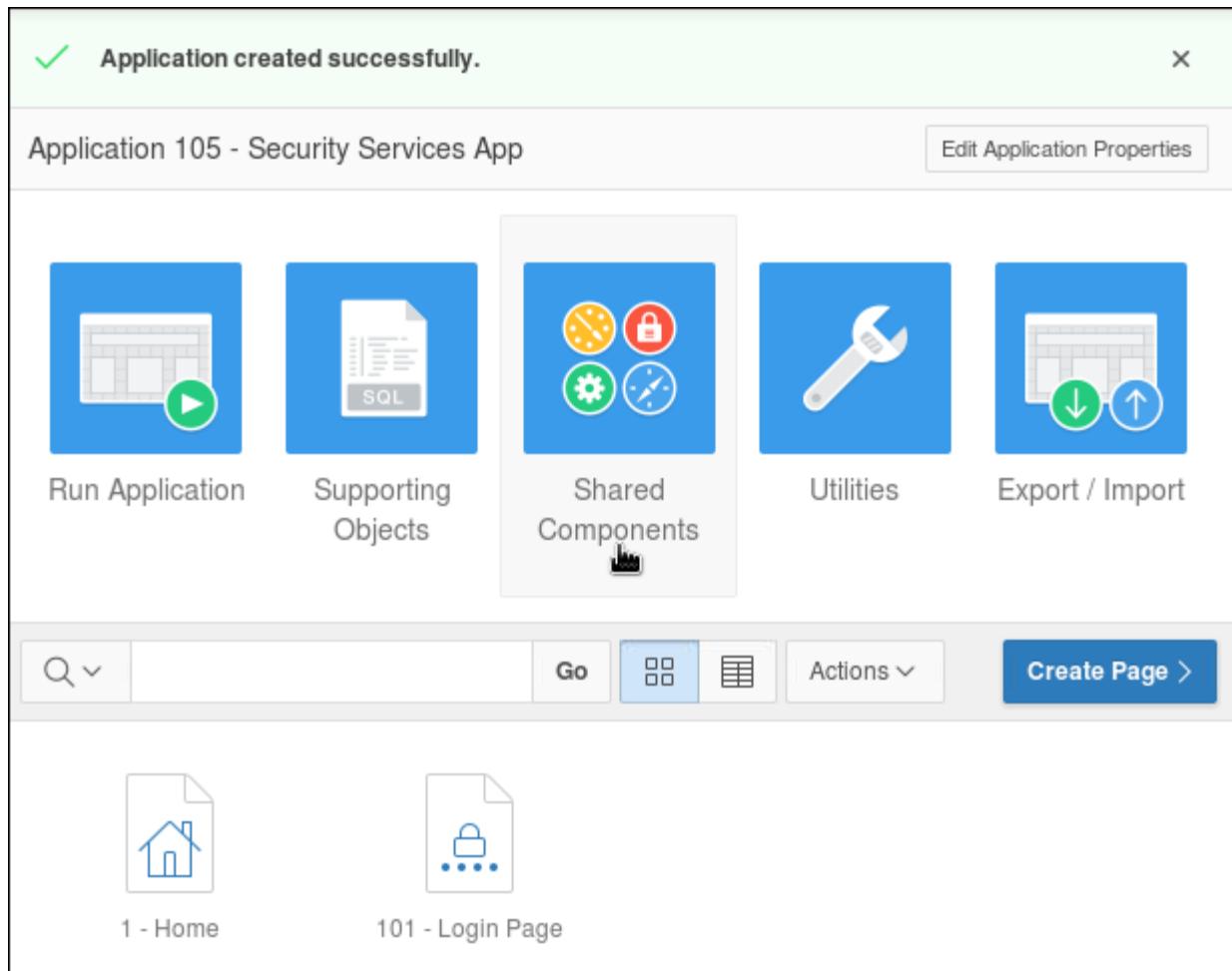
You have requested to create an application with the following attributes. Please confirm your selections.

Application	105
Name	Security Services App
Parsing Schema	OEHR
Default Language	en
Navigation	Navigation Menu List
Default Authentication Scheme	APEX
Theme Type	Standard
Theme	Universal Theme
Subscribe Theme	No

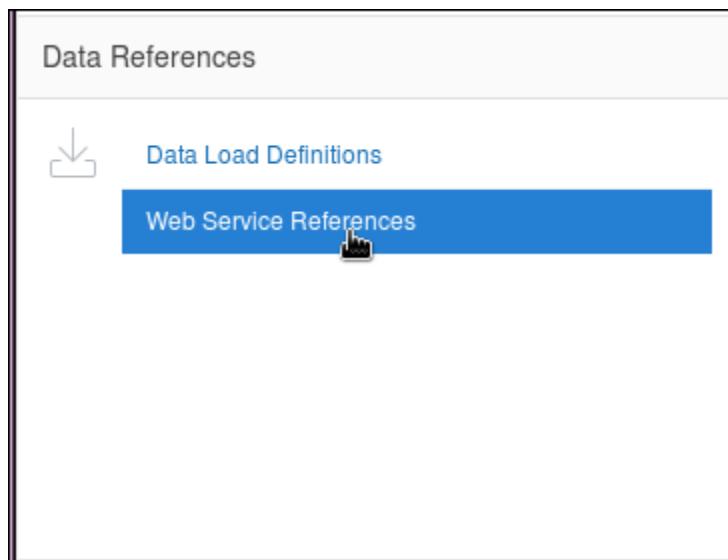
< Cancel

Create Application 

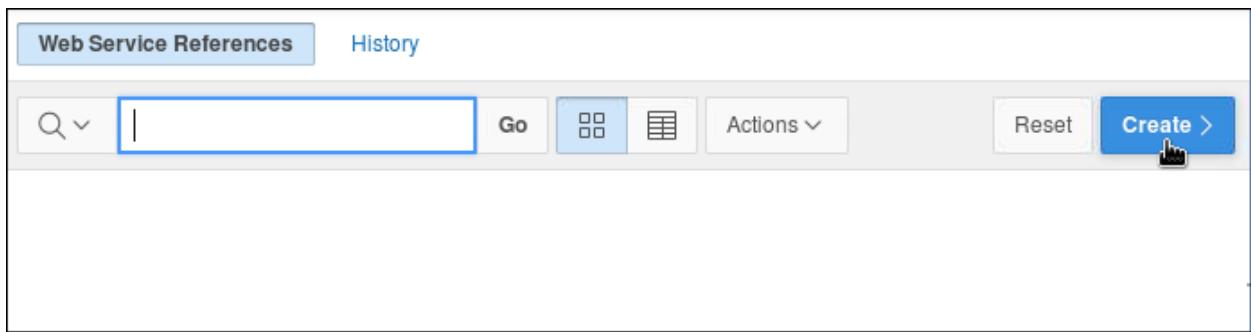
- b. Create a Web Service Reference in the Security Service application. You then create a form and report page that uses the web service.
- On the application home page, click **Shared Components**.



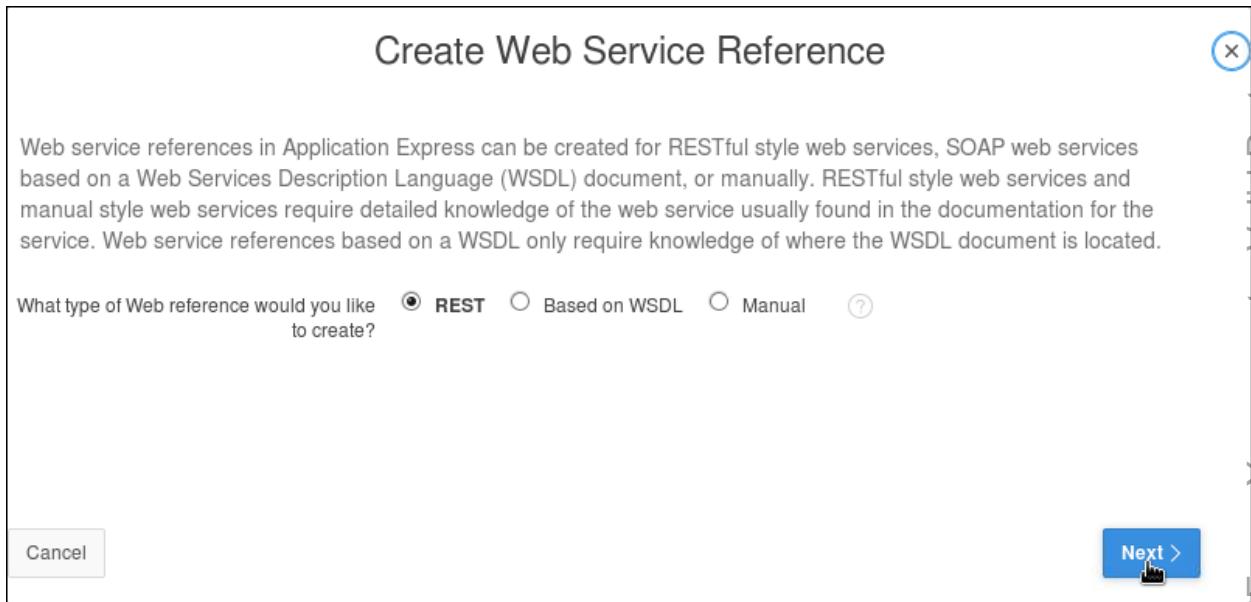
- Under **Data References**, click **Web Service References**.



3. Click **Create**.



4. Select **REST** for Web Reference and click **Next**.



5. Enter **Employees** for **Name** and enter the **URL** which you noted down in Practice 8-1 (step 9) for URL. In this case, enter <http://localhost:8080/apex/2210243963958320/oehr/>

Since the REST Web Service does not require a HTTP Header parameter, click the **Delete Header** icon. Click **Next**.

Create REST Web Reference

REST Details

RESTful Web services rely on a simple resource-oriented architecture. The resource is identified by the URL and the method is described by the HTTP method. Inputs to the service are sometimes contained in the URL itself or in the HTTP payload. Inputs can also be read from HTTP headers sent with the request.

Application: 105 Security Services App

* Name: Employees

* URL: http://localhost:8080/apex/2210243963958320/oehr/

Proxy:

HTTP Method: GET HEAD POST PUT DELETE

< Cancel Next >

Create REST Web Reference

REST Details

* URL: http://localhost:8080/apex/2210243963958320/oehr/

Proxy:

HTTP Method: GET HEAD POST PUT DELETE

Basic Authentication: Yes No

HTTP Request Headers

	Name

Add Header < Cancel Next >

Create REST Web Reference

REST Details

RESTful Web services rely on a simple resource-oriented architecture. The resource is identified by the URL and the method is described by the HTTP method. Inputs to the service are sometimes contained in the URL itself or in the HTTP payload. Inputs can also be read from HTTP headers sent with the request.

Application: 105 Security Services App

* Name: Employees

* URL: http://localhost:8080/apex/2210243963958320/oehr/

Proxy:

HTTP Method: GET HEAD POST PUT DELETE

< Cancel Next >

6. Delete the Input parameter by clicking **Delete Header** icon. Click **Next**.

Create REST Web Reference

REST Inputs

Choose the type of input parameters if the method chosen on the previous page is POST or PUT. Otherwise, the input type is assumed to be Name Value Pairs.

Add Parameter

Name	Type
<input type="text"/>	String

< Cancel Next >

7. Select **Text** for Output Format. Enter the names of the parameters like Employee ID and First Name by mapping them to the response. Click **Test**.

Create REST Web Reference

REST Outputs

list, then define output parameters as necessary. You can review the output from this web service by clicking the Test button.

Output Format: XML Text JSON [?](#)

Parameter Delimiter: [?](#)

New Record Delimiter: [?](#)

Add Parameter [Type](#)

	Name	Path	Type
X	Employee ID	1	String

< Cancel [Test](#) [Create](#)

Create REST Web Reference

REST Outputs

Parameter Delimiter: [?](#)

New Record Delimiter: [?](#)

Add Parameter

	Name	Path	Type
X	Employee ID	1	String
X	First Name	2	String

< Cancel [Test](#) [Create](#)

8. Click Test.

REST Web Service Details

Use this page to test the reference of a REST Web service.
Modify the URL to the service if it contains any references to session state. Enter a Proxy Override if you wish to use a specific one for the test you initiate on this page. Supply basic authentication credentials if it is required by the Web service. Enter values for required HTTP headers and Input Parameters and click Test. The response will appear in a scrollable region below the Input Parameters region.

Name **Employees** 

* URL 

Proxy Override 

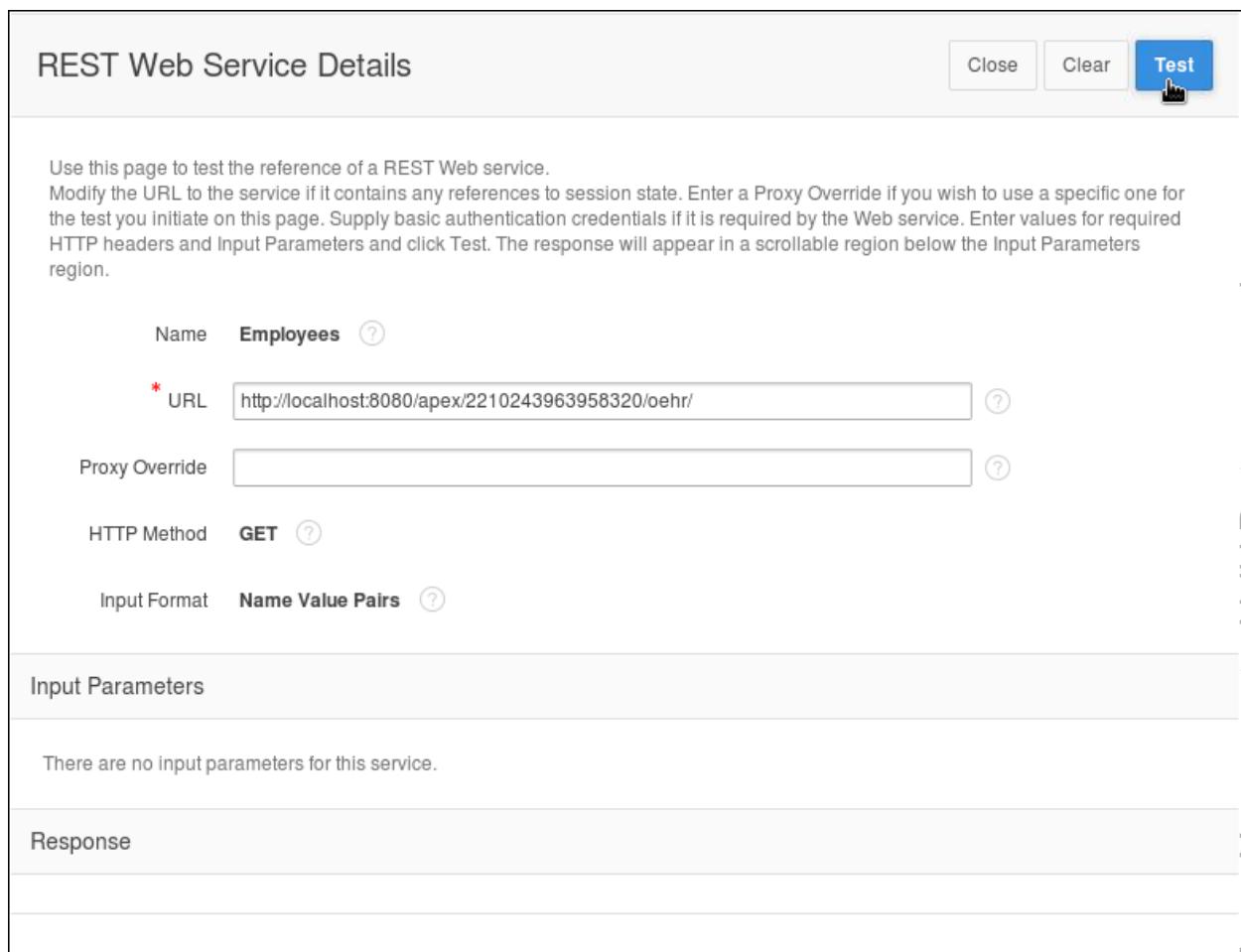
HTTP Method **GET** 

Input Format **Name Value Pairs** 

Input Parameters

There are no input parameters for this service.

Response



9. Scroll down to view the response. Click **Close**.

The screenshot shows a Mozilla Firefox window titled "Test REST Web Reference - Mozilla Firefox". The address bar displays "localhost:8080/apex/f?p=4000:975:8071837524455::NO:::region.". The main content area is a configuration interface for a REST service. It includes fields for "Name" (set to "Employees"), "URL" (set to "http://localhost:8080/apex/f?p=4000:975:8071837524455::NO:::region."), "Proxy Override", "HTTP Method" (set to "GET"), and "Input Format" (set to "Name Value Pairs"). Below these settings is a section titled "Input Parameters" which states "There are no input parameters for this service." Under the "Response" section, a large block of text lists employee data from the HR table:

```
100,Steven,King,SKING,515.123.4567,1987-06-17T00:00:00Z,SA_REP,10000,0.05,,90,Y  
101,Neena,Kochhar,NKOCHHAR,515.123.4568,1989-09-21T00:00:00Z,AD_VP,17000,,100,90,  
102,Lex,De Haan,LDEHAAN,515.123.4569,1993-01-13T00:00:00Z,AD_VP,17000,,100,90,  
103,Alexander,Hunold,AHUNOLD,590.423.4567,1990-01-03T00:00:00Z,IT_PROG,9000,,102,60,  
104,Bruce,Ernst,BERNST,590.423.4568,1991-05-21T00:00:00Z,IT_PROG,6000,,103,60,  
105,David,Austin,DAUSTIN,590.423.4569,1997-06-25T00:00:00Z,IT_PROG,4800,,103,60,  
106,Valli,Pataballa,VPATABAL,590.423.4560,1998-02-05T00:00:00Z,IT_PROG,4800,,103,60,  
107,Diana,Lorentz,DLORENTZ,590.423.5567,1999-02-07T00:00:00Z,IT_PROG,4200,,103,60,  
108,Nancy,Greenberg,NGREENBE,515.124.4569,1994-08-17T00:00:00Z,FI_MGR,12000,,101,100,  
109,Daniel,Faviet,DFAVIET,515.124.4169,1994-08-16T00:00:00Z,FI_ACCOUNT,9000,,108,100,  
110,John,Chen,JCHEN,515.124.4269,1997-09-28T00:00:00Z,FI_ACCOUNT,8200,,108,100,  
111,Ismael,Sciarra,ISCIARRA,515.124.4369,1997-09-30T00:00:00Z,FI_ACCOUNT,7700,,108,100,
```

10. Click **Create**.

Create REST Web Reference

REST Outputs

Parameter Delimiter : ,

New Record Delimiter : \n

Output Parameters

	Name	Path	Type
X	Employee ID	1	String
X	First Name	2	String

Add Parameter

< Cancel Test Create

11. Click Application<n> link.

Application 105 Shared Components Web Service References

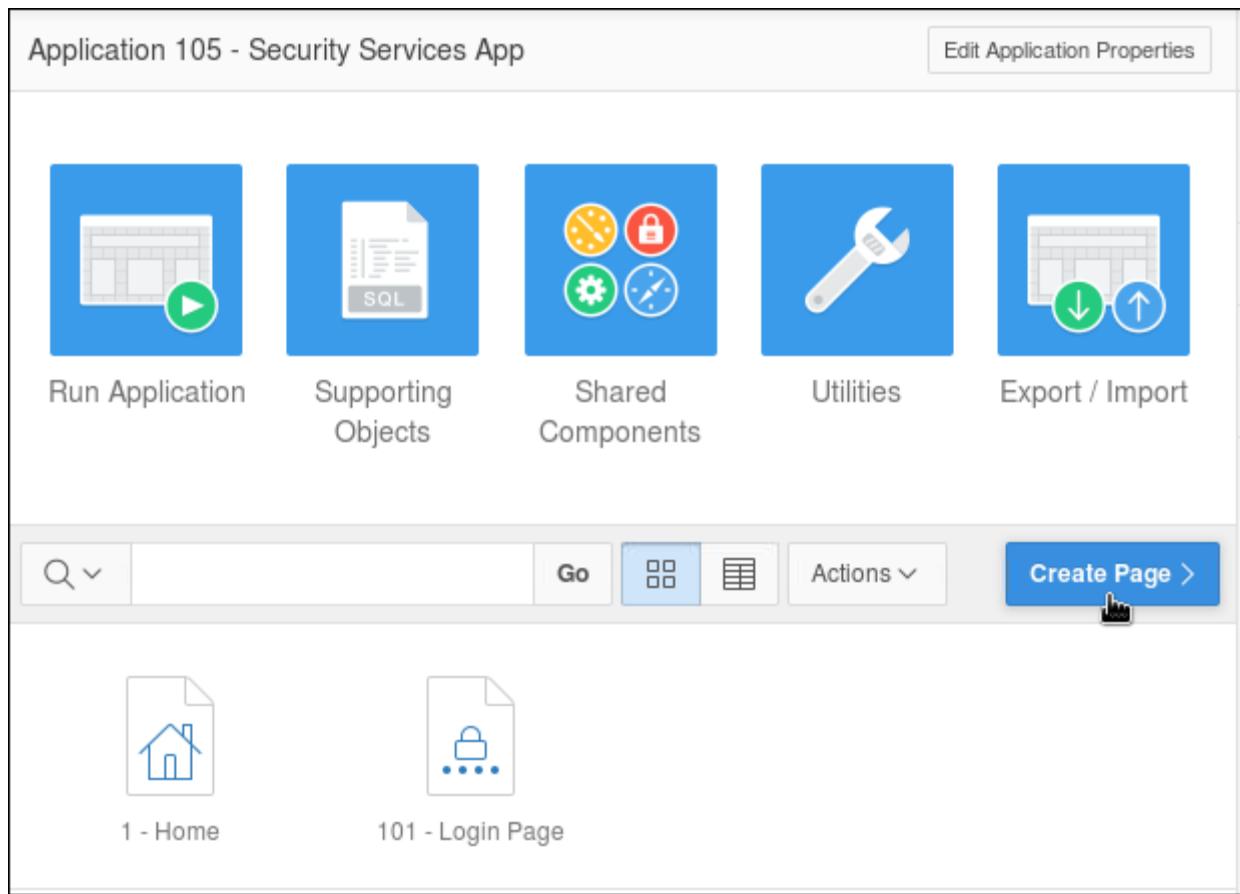
Web Service References History

Search: Employees

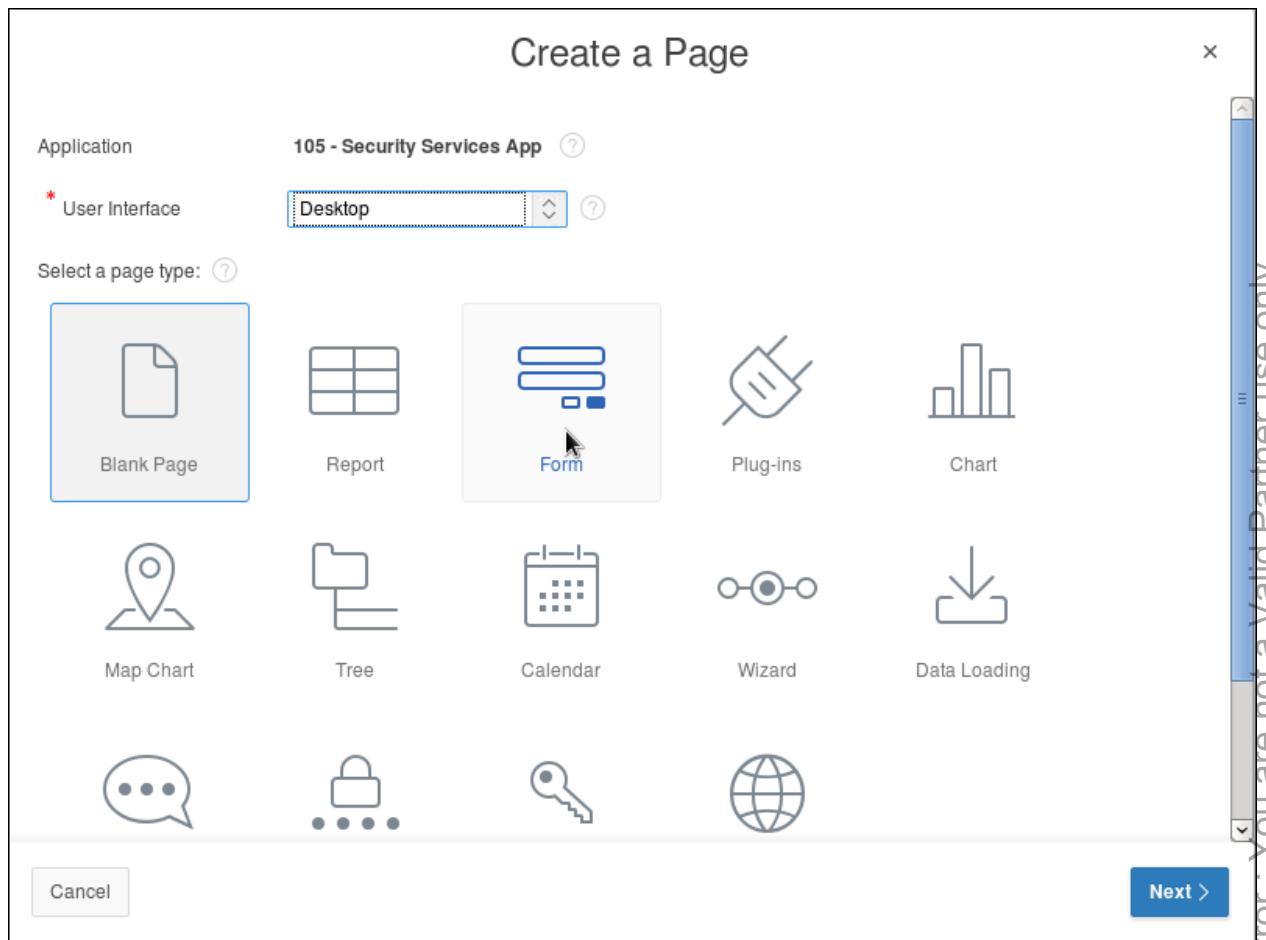
Actions: Go, Actions, Reset, Create

Employees

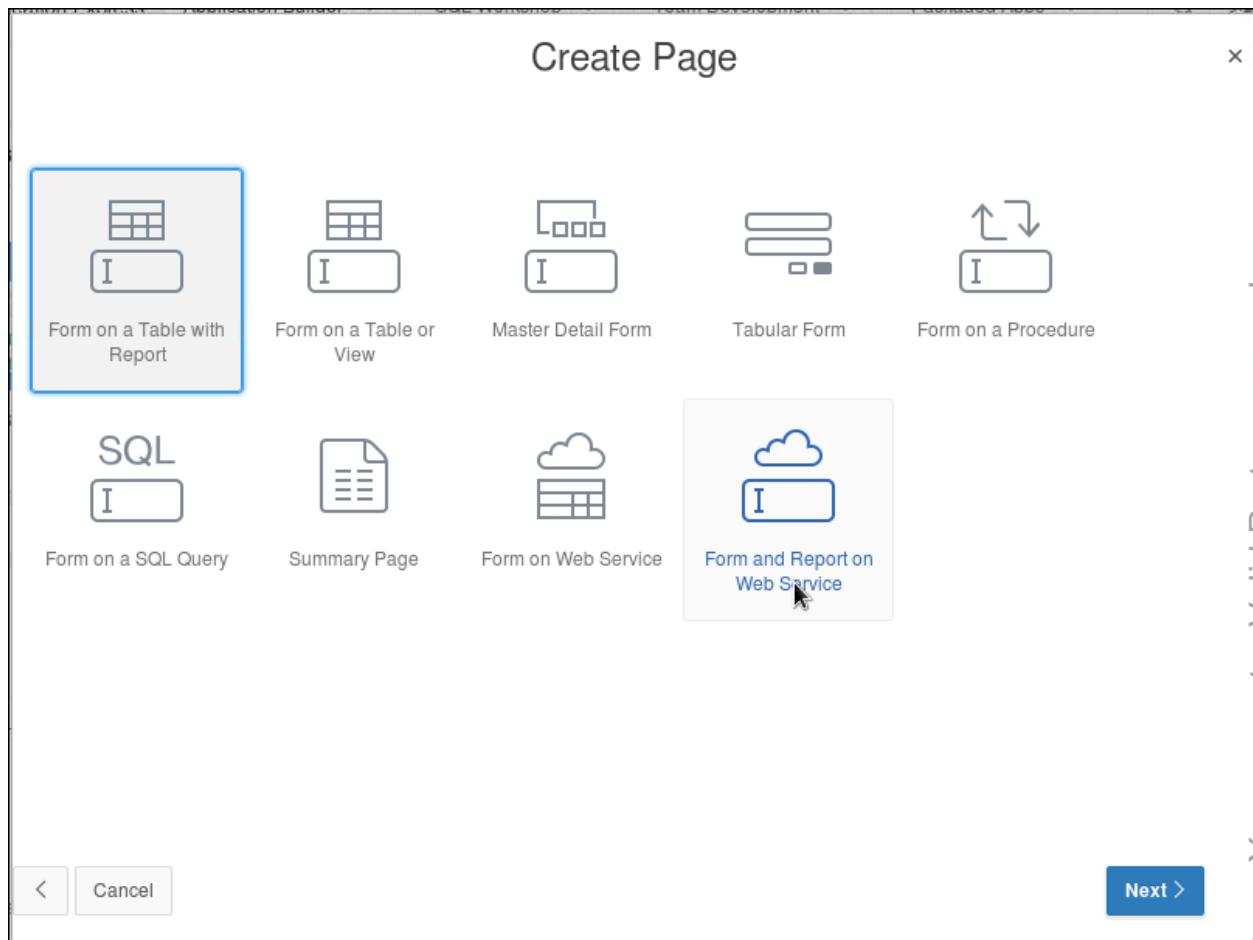
12. Click **Create Page**.



13. Click **Form**.



14. Click **Form and Report on Web Service**.



15. Select Web Service Reference as **Employees**. Click **Next**.

Create Form and Report on Web Service

Choose Service and Operation

Select the web service definition and an operation on which to build the form. If you do not have a web service reference defined, expand the Tasks region and create one.

* Web Service Reference Employees

Web Service Reference Type RESTful

* Operation doREST

Tasks

< Cancel

16. Accept the default values and click **Next**.

Create Form and Report on Web Service

Page and Region Attributes

Use this page to specify page and region information.

Web Service Reference: Employees [?](#)

Operation: doREST [?](#)

* Page 2 [?](#)

* Page Name Employees [?](#)

* Page Mode Normal [?](#)

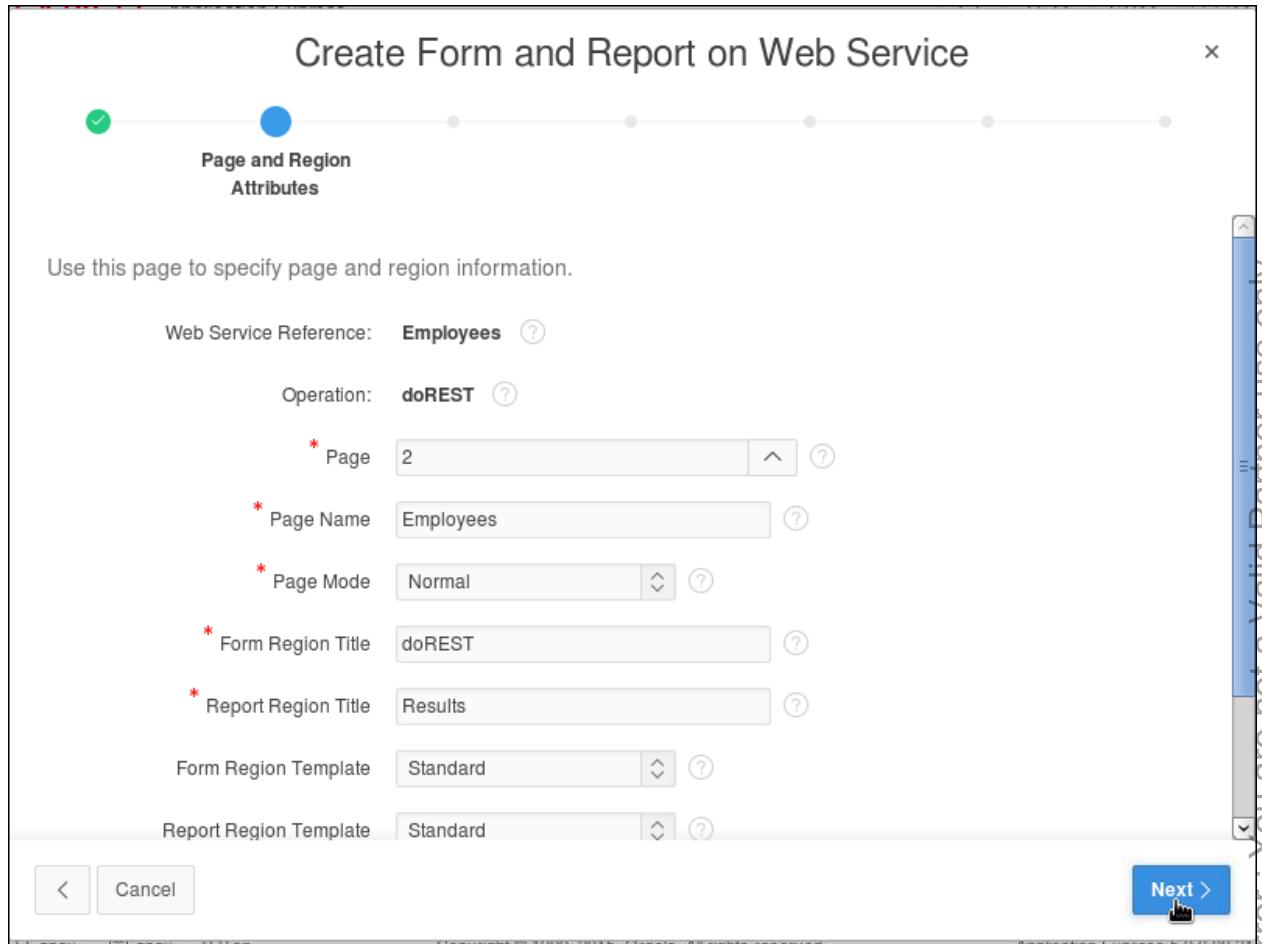
* Form Region Title doREST [?](#)

* Report Region Title Results [?](#)

Form Region Template Standard [?](#)

Report Region Template Standard [?](#)

< Cancel **Next >**



17. Click **Next**.

Create Form and Report on Web Service

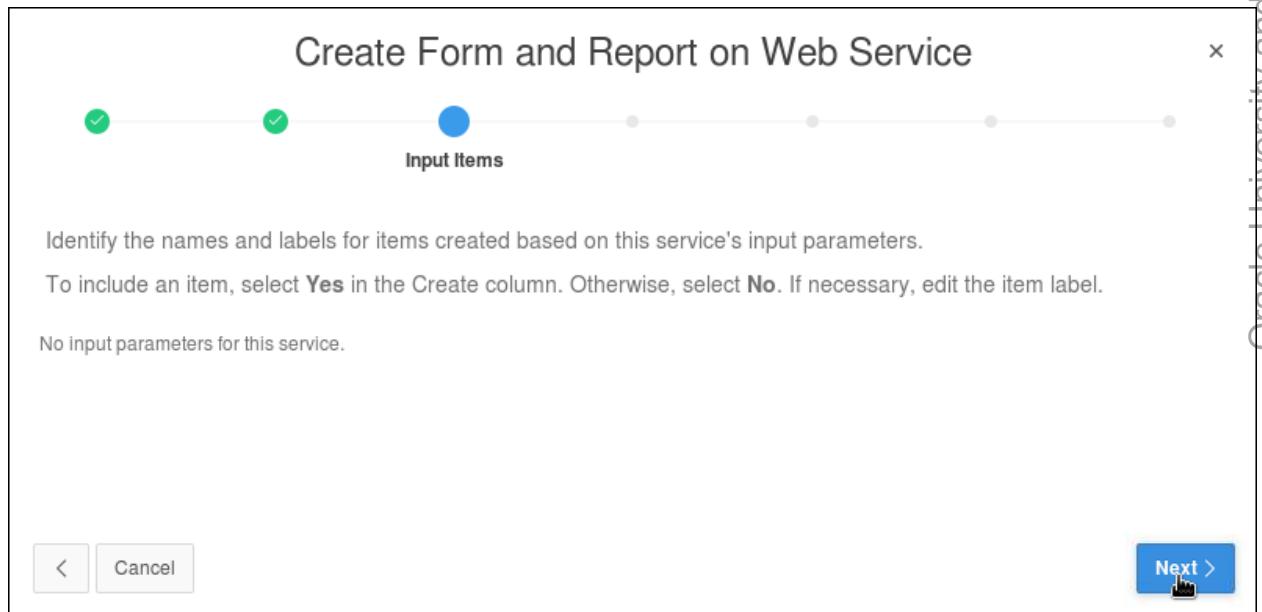
Input Items

Identify the names and labels for items created based on this service's input parameters.

To include an item, select **Yes** in the Create column. Otherwise, select **No**. If necessary, edit the item label.

No input parameters for this service.

< Cancel **Next >**



18. Select the check box next to **Name** to select all the parameters and click **Next**.

Create Form and Report on Web Service x

Report Parameters

Select the parameters to include in the report. To select or deselect all parameters, select the check box in the column header.

For JSON response you only need to supply the name of the collection to store the response.

* Store Results in Collection ?

	Name	Parameter Type
<input checked="" type="checkbox"/>	Employee ID	string
<input checked="" type="checkbox"/>	First Name	string

< Cancel Next >

19. Select **Create a new navigation menu entry** for the Navigation Preference. Enter **Employees Detail** for the New Navigation Menu entry. Click **Next**.

Navigation Preference

Create a new navigation menu entry

Do not associate this page with a navigation menu entry

Identify an existing navigation menu entry for this page

* New Navigation Menu Entry

Parent Navigation Menu Entry
- No parent selected -
Home

Cancel Next >

20. Click **Create**.

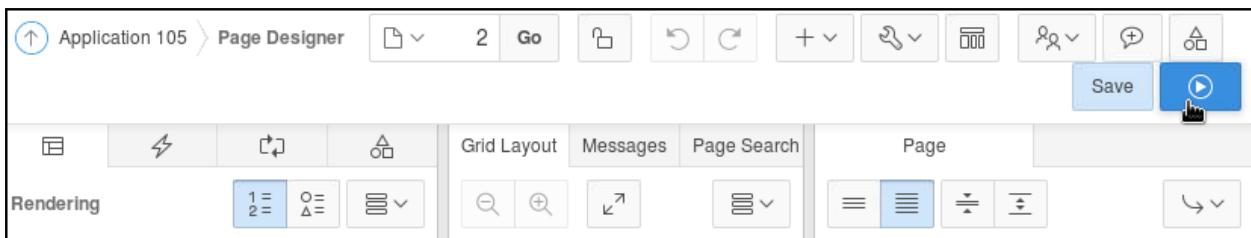
Create Form and Report on Web Service

You have requested to create a form and report on a Web service reference page with the following attributes.
Please confirm your selections.

Application	105
Page	2
Page Name	Employees
Form Region Title	doREST
Report Region Title	Results
Form Region Template	Standard
Report Region Template	Standard
Report Template	template: Standard

Confirm Create

21. Click **Save and Run Page**.



22. Enter your login credentials if prompted and click **Log In**.



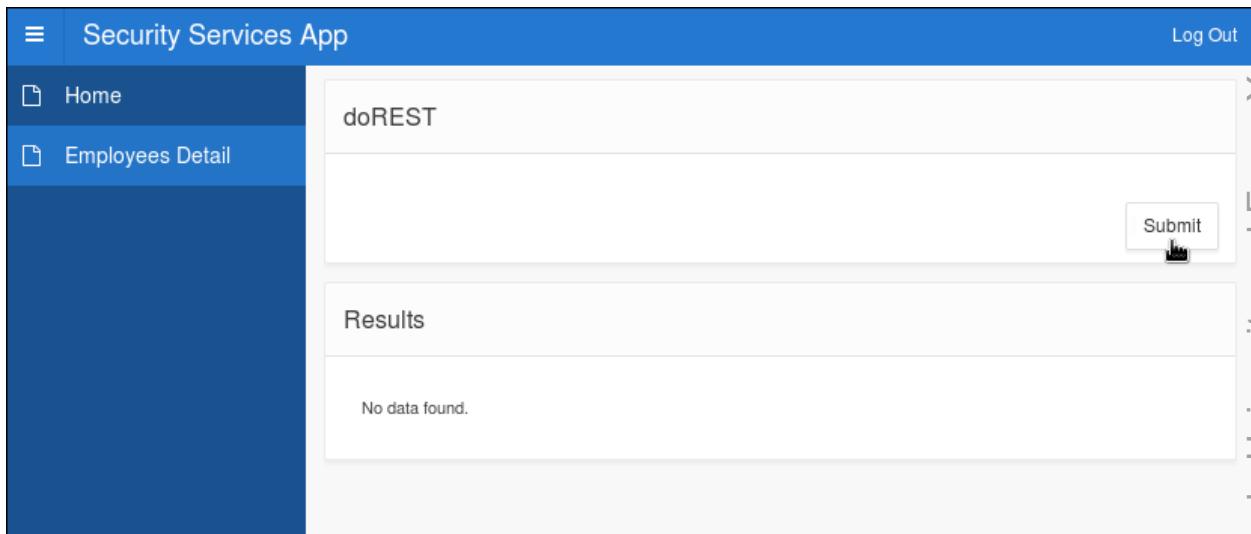
Log In

Username apex

Password ****

Log In

23. Click **Submit**.



Security Services App

Log Out

doREST

Results

No data found.

Submit

24. The resultset is displayed.

The screenshot shows a web application titled "Security Services App". On the left, a sidebar menu includes "Home" and "Employees Detail". The main content area has a header "doREST" and a "Submit" button. Below this is a section titled "Results" containing a table with two columns: "Employee Id" and "First Name". The table data is as follows:

Employee Id	First Name
100	Steven
101	Neena
102	Lex
103	Alexander
104	Bruce
105	David
106	Valli
107	Diana
108	Nancy

At the bottom of the page, there is a navigation bar with links: Home, Application 105, Edit Page 2, Session, View Debug, Row Grid, Quick Edit, Theme Roller, and a user icon. The "Theme Roller" link is highlighted.

Practices for Lesson 9: Using Templates and Themes

Chapter 9

Practices for Lesson 9: Overview

Practices Overview

In these practices, you copy and customize a Button Template in the GMT application. You create a custom theme and apply this theme on a master application.

Practice 9-1: Copying and Customizing a Button Template

Overview

In this practice, you copy and customize a Button Template in the GlobalMart Management Tool Application.

Assumptions

You should have imported the GMT application.

Tasks

- a. Create a classic report page called New Customer Information in the GMT Application with a Create button.
- b. Copy the Button Template and customize the template.
- c. Modify the Custom Button Template and add a class called redButton.
- d. Associate the copied template with your page by including the CSS styles to the page definition.

Practice 9-2: Creating a Custom Theme

Overview

In this practice, you create a custom theme by exporting, copying, and editing your theme. You add the custom theme as a Workspace Theme so that you can make it available to any application you create.

Assumptions

You should have completed Practice 9-1.

Tasks

- a. In the GMT application, go to Shared Components and export your theme.
- b. Copy the theme.
- c. Make the custom theme available to any application you create.

Practice 9-3: Creating a Master Application

Overview

In this practice, you create an application with the custom theme that all groups will subscribe to.

Assumptions

You should have completed Practice 9-1 and 9-2.

Tasks

- a. Create an application with login and home pages.
- b. Add the custom theme that you created in the previous practice to this application.
- c. Switch the theme to make the Custom Universal–Master Theme your current theme.

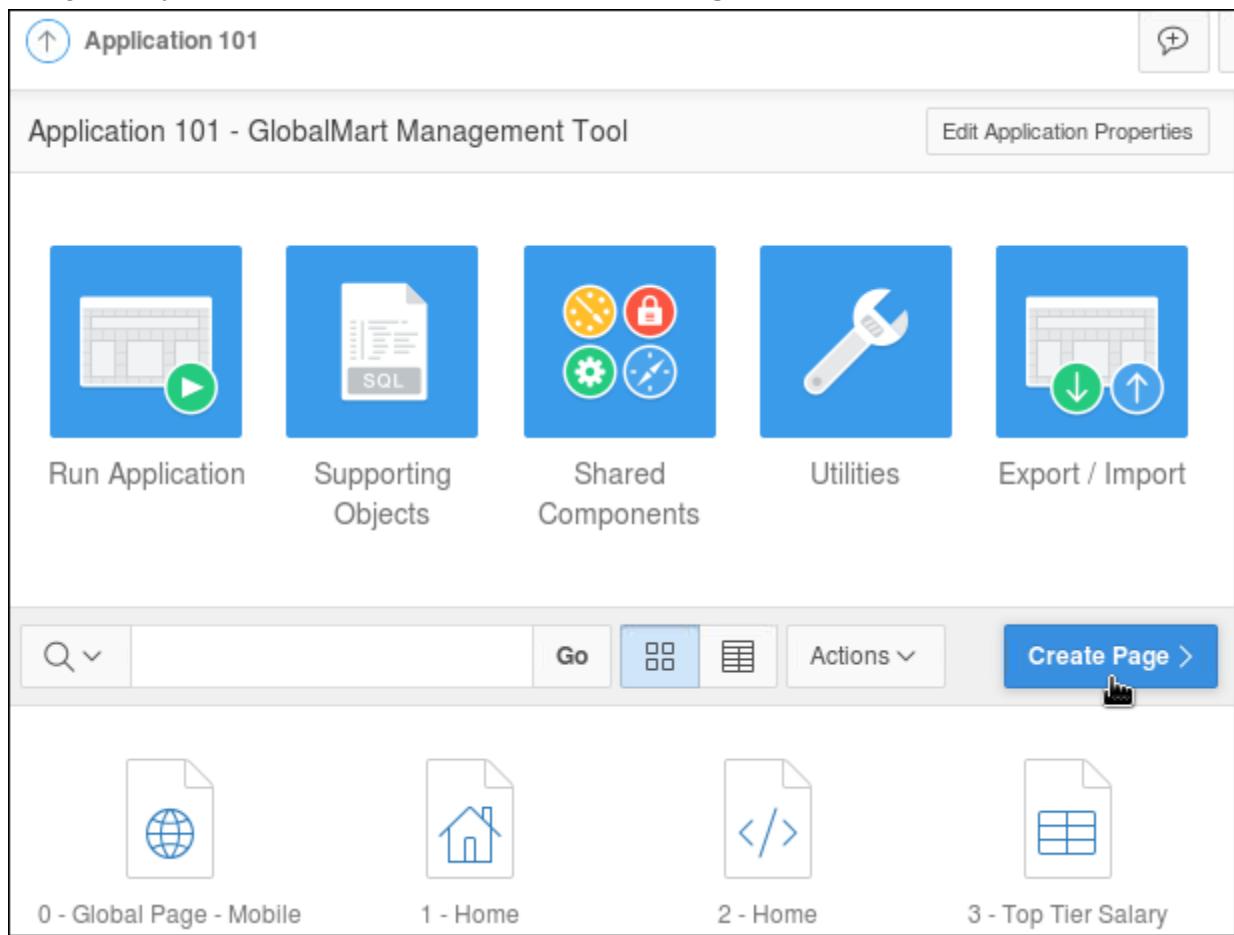
Solution 9-1: Copying and Customizing a Button Template

Overview

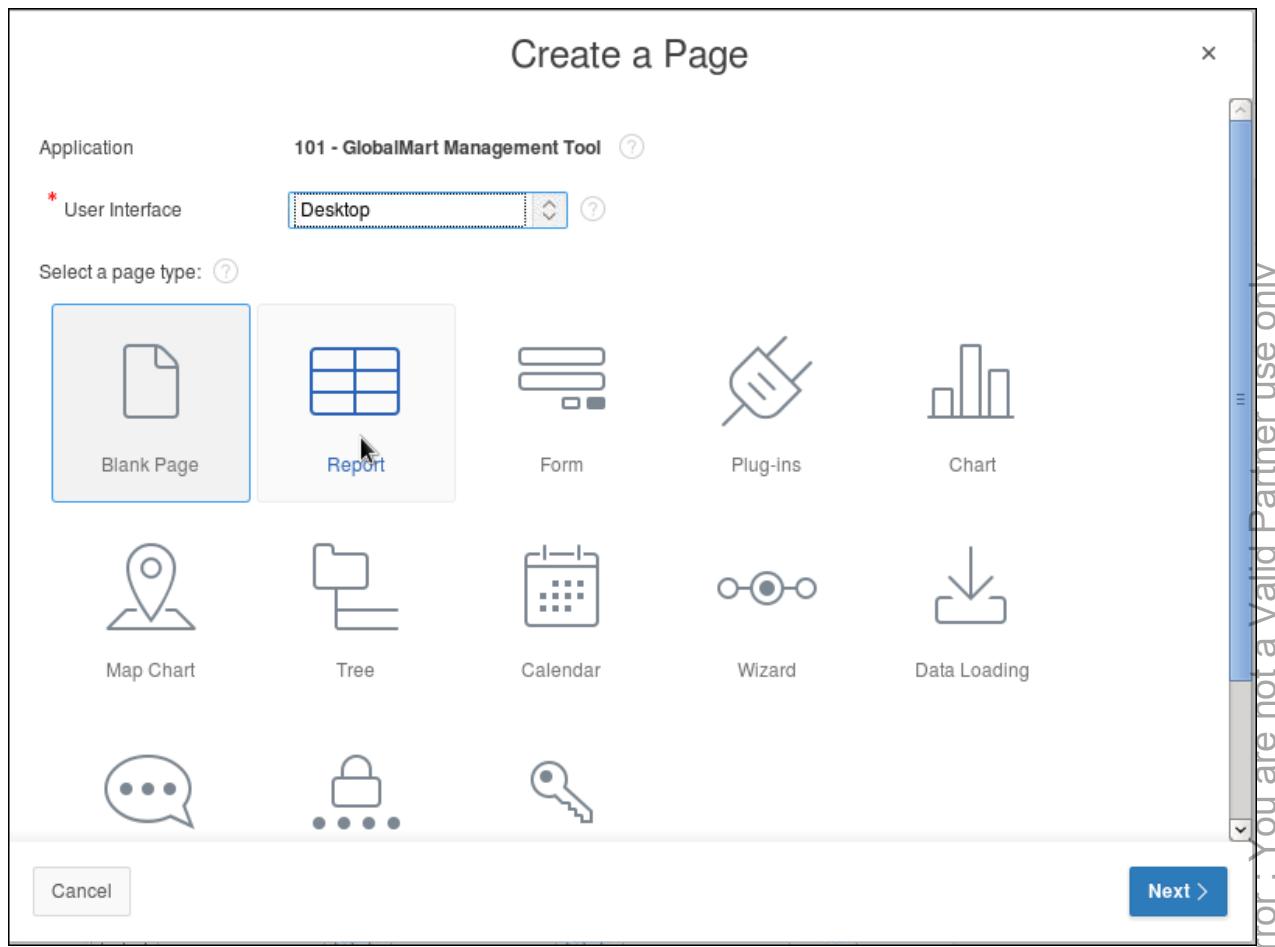
In this practice, you copy and customize a Button Template in the GlobalMart Management Tool Application.

Steps

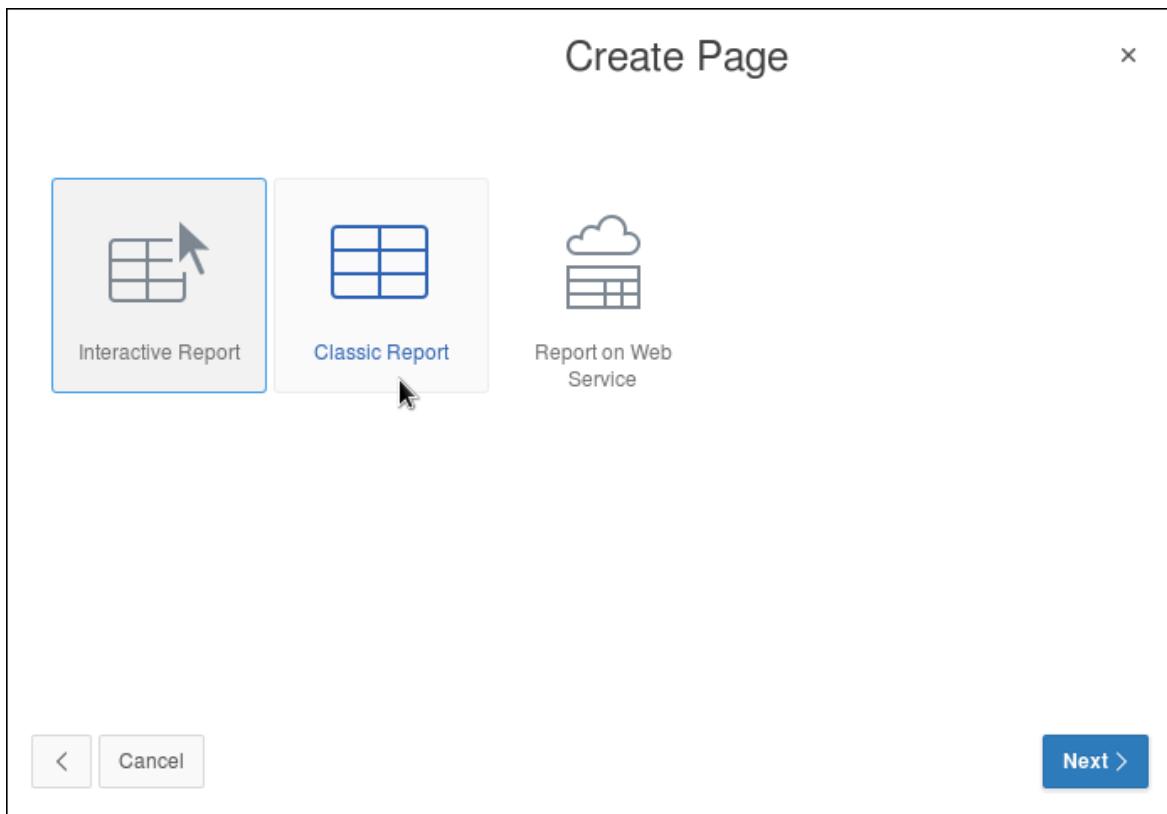
- Create a classic report page called New Customer Information in the GMT Application with a Create button.
- Navigate to your GMT Application and click **Create Page**.



2. Select Report.



3. Select Classic Report.



4. Enter **New Customer Information** for Page Name and Region Name, and click Next.

Create Classic Report

Page Attributes

* Page Number: 44

* Page Name: New Customer Information

* Page Mode: Normal

Page Group: - Select Page Group -

* Region Name: New Customer Information

Region Template: Standard

* Report Template: template: 42. Standard

Breadcrumb: - do not use breadcrumbs on page -

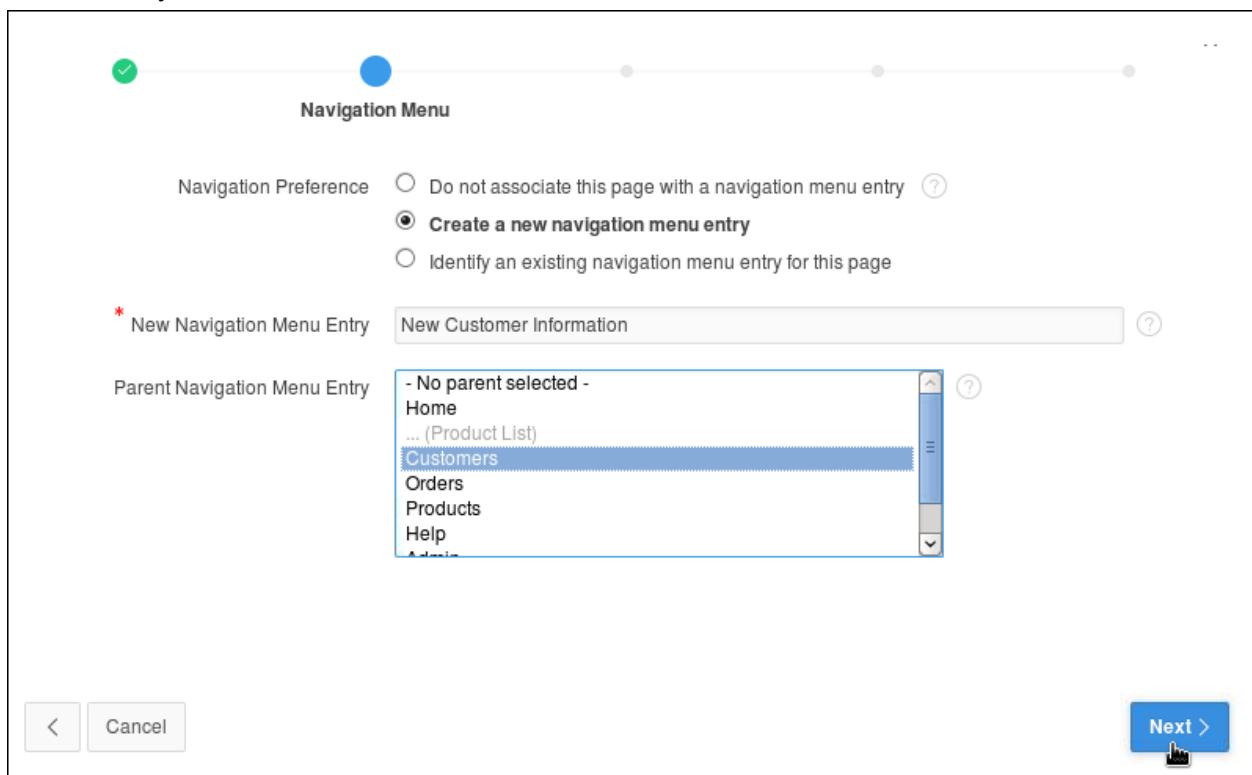
Tasks

Use first available free page

< Cancel Next >

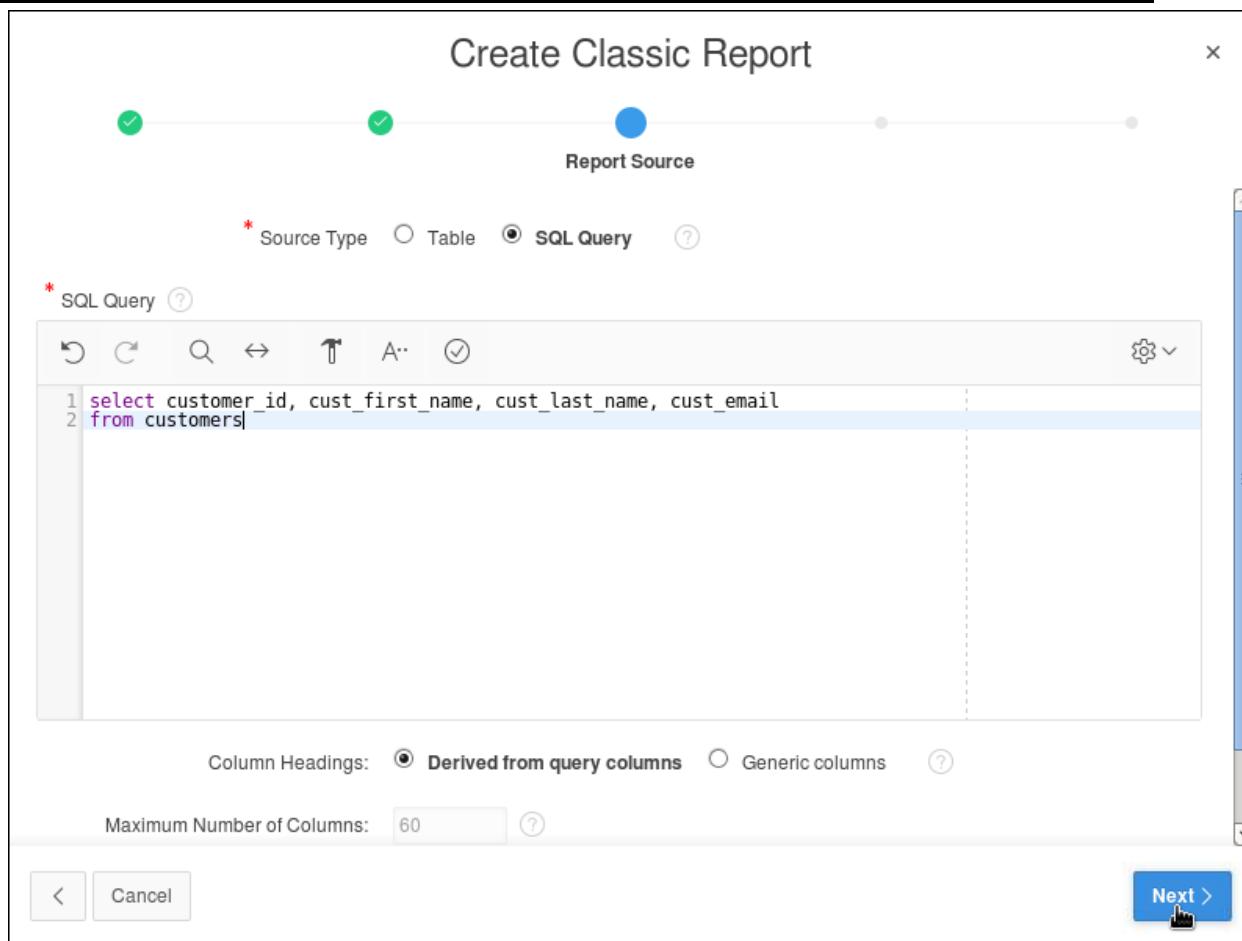
Oracle University and Error : You are not a Valid Partner use only

5. Select **Create a new navigation menu entry**, select **Customers** for Parent Navigation Menu Entry, and click **Next**.



6. Select **SQL Query** for Report Source and enter the following SQL SELECT statement (also located in the /home/oracle/labs/files/lab_09_01.txt file). Click **Next**.

```
select customer_id, cust_first_name, cust_last_name, cust_email  
from customers
```



7. Select **No** for CSV Output and click **Next**.

Create Classic Report

x

Report Attributes

Column Heading Sorting Yes

CSV Output No

Link Label Download

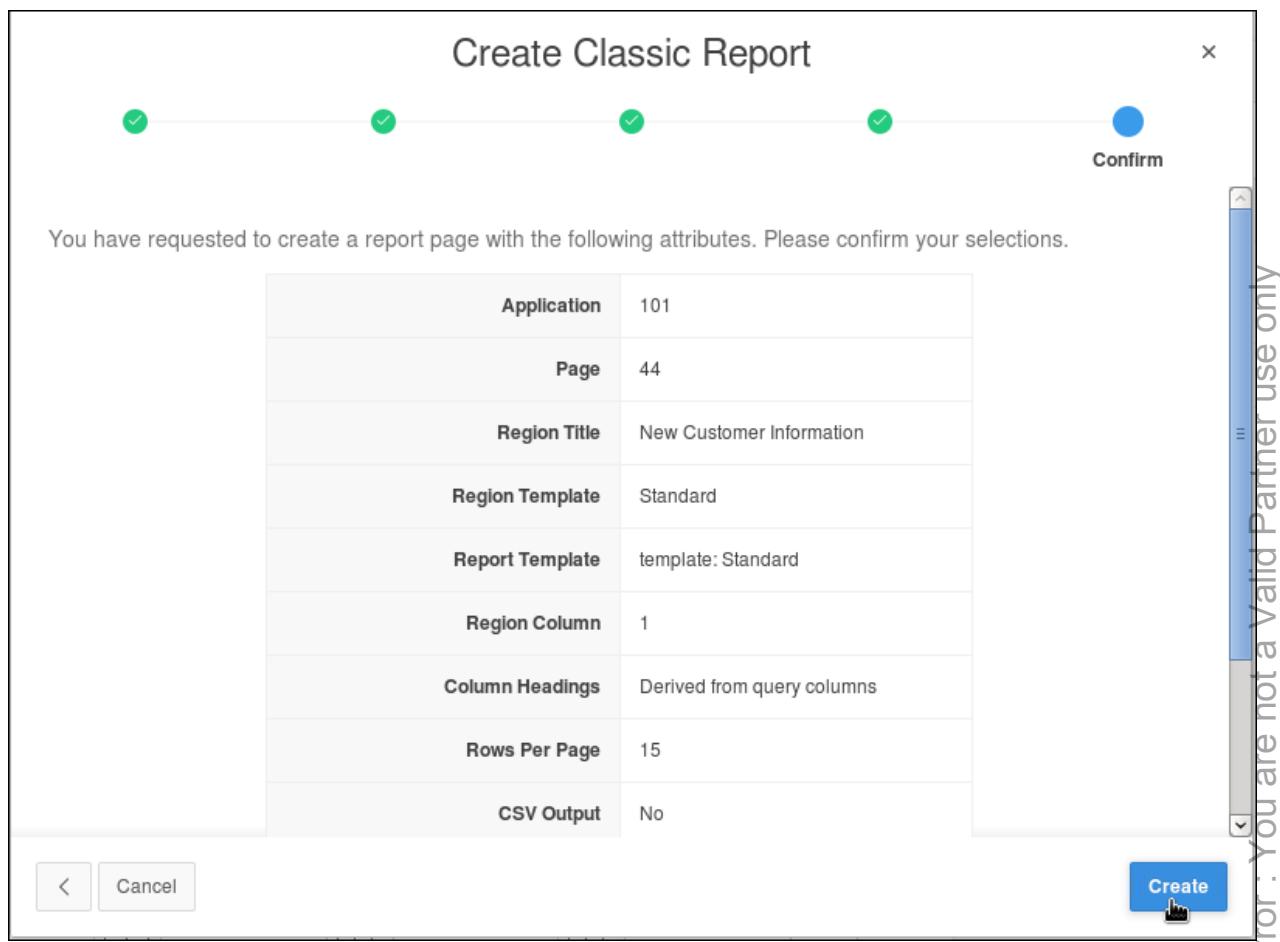
Report Printing No

Link Label

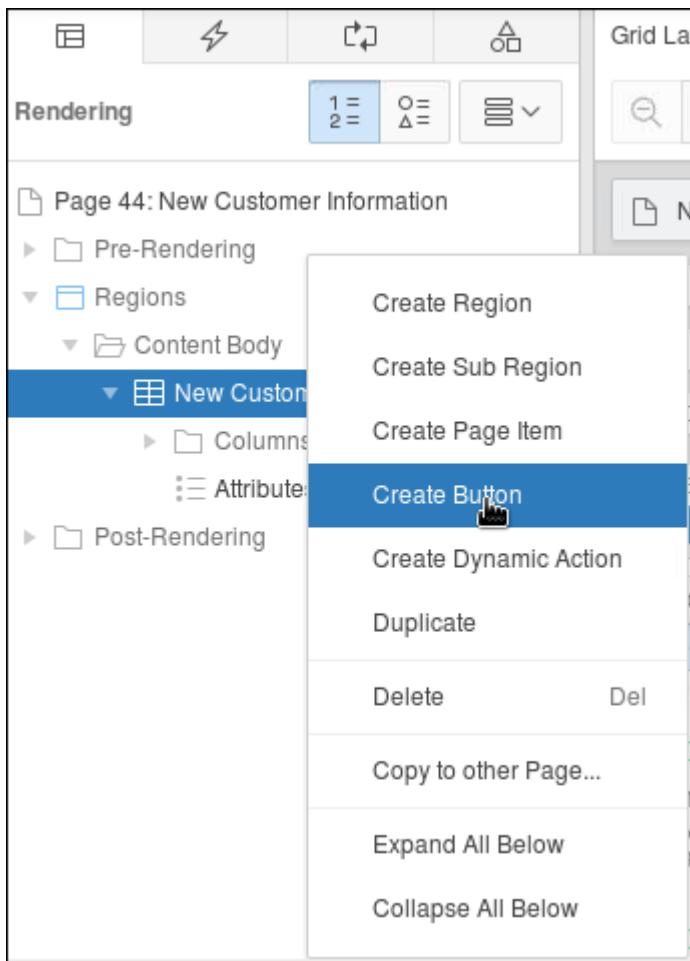
Output Format PDF

Enable Search No

< Cancel

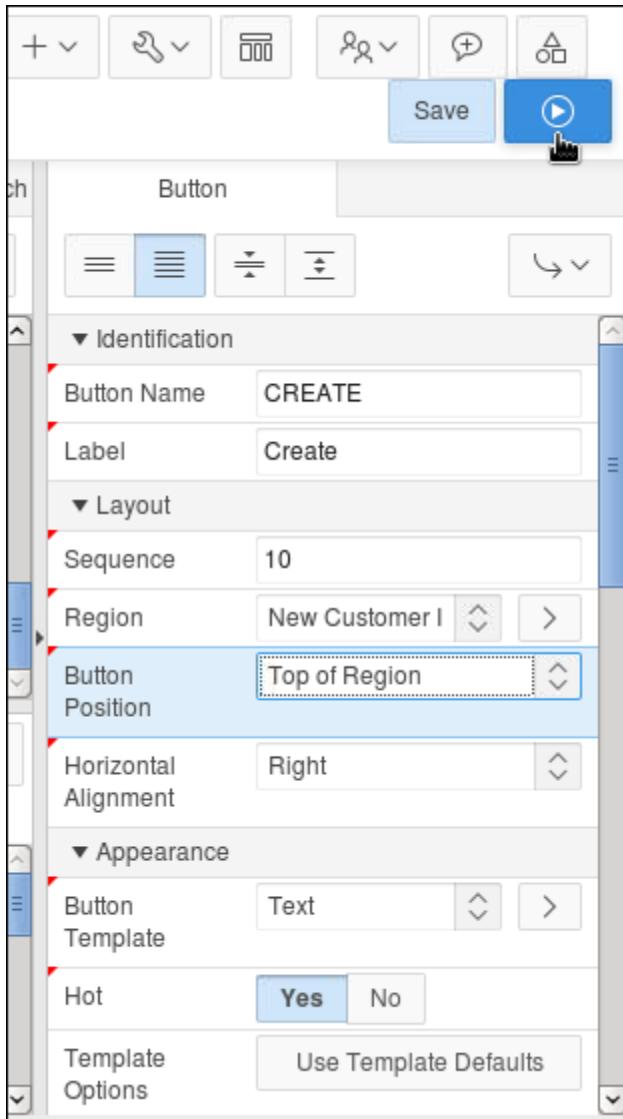
8. Click **Create**.

9. Right-click **New Customer Information** region and select **Create Button**.



10. On the Property Editor page, enter **CREATE** for **Button Name**. Select **Yes** for **Hot**.

11. Select **Top of Region** for Button position and click **Save and Run page**.



12. Enter Username and Password and click **Log In**.

The screenshot shows the Oracle APEX log in page. The 'Username' field contains 'apex' and the 'Password' field contains '****'. The 'Log In' button is highlighted with a cursor icon.

Log In

Username: apex

Password: ****

Log In

Notice that the Create button is blue in color.

New Customer Information

Customer Id	Cust First Name	Cust Last Name	Cust Email
101	Constantin	Welles	Constantin.Welles@ANHINGA.COM
102	Harrison	Pacino	Harrison.Pacino@ANI.COM
103	Manisha	Taylor	Manisha.Taylor@AUKLET.COM
104	Harrison	Sutherland	Harrison.Sutherland@GODWIT.COM
105	Matthias	MacGraw	Matthias.MacGraw@GOLDENYE.COM
106	Matthias	Hannah	Matthias.Hannah@GREBE.COM
107	Matthias	Cruise	Matthias.Cruise@GROSBEAK.COM
108	Meenakshi	Mason	Meenakshi.Mason@JACANA.COM
109	Christian	Cage	Christian.Cage@KINGLET.COM
110	Charlie	Sutherland	Charlie.Sutherland@LIMPKIN.COM
111	Charlie	Pacino	Charlie.Pacino@LONGSPUR.COM
112	Guillaume	Jackson	Guillaume.Jackson@MOORHEN.COM
113	Guillaume	Costner	Daniel.Costner@PARULA.COM

Home Application 101 Edit Page 44 Session View Debug Debug Show Grid Quick Edit Theme Roller

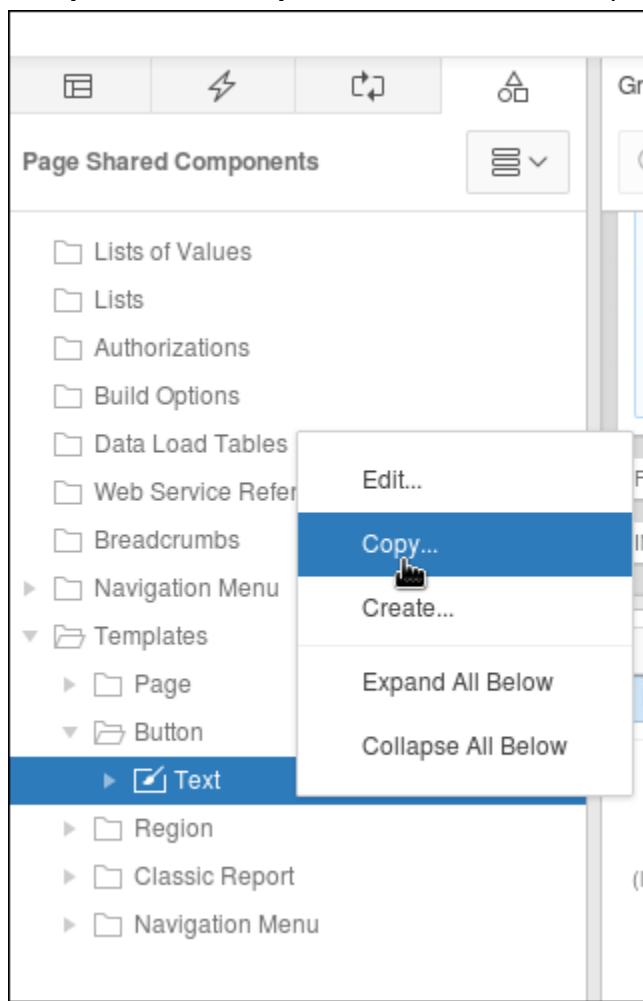
- b. You want to copy the Button Template and customize the template.
- Click **Edit Page <n>** on the Developer toolbar.

The screenshot shows the GlobalMart Management Tool interface. On the left is a sidebar with navigation links: Home, Customers (selected), New Customer Information, Orders, Products, Help, Admin, and Data Load. The main content area is titled "New Customer Information" and displays a table of customer data:

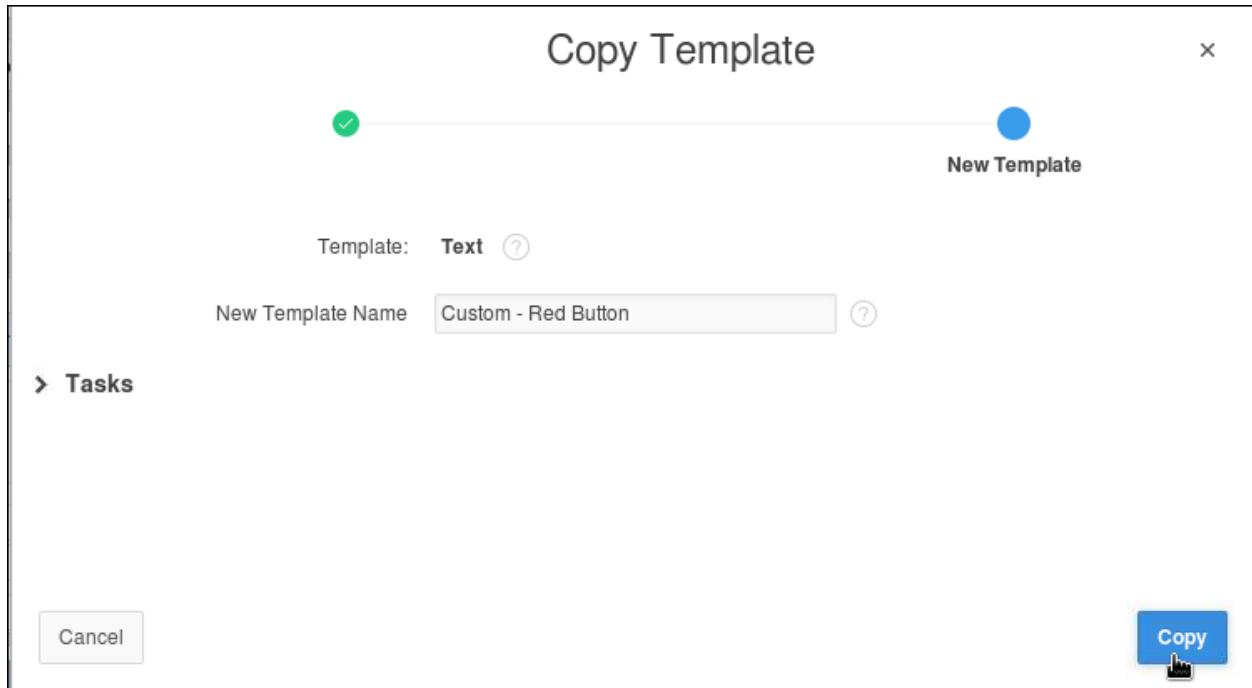
Customer Id	Cust First Name	Cust Last Name	Cust Email
101	Constantin	Welles	Constantin.Welles@ANHINGA.COM
102	Harrison	Pacino	Harrison.Pacino@ANI.COM
103	Manisha	Taylor	Manisha.Taylor@AUKLET.COM
104	Harrison	Sutherland	Harrison.Sutherland@GODWIT.COM
105	Matthias	MacGraw	Matthias.MacGraw@GOLDENEYE.COM
106	Matthias	Hannah	Matthias.Hannah@GREBE.COM
107	Matthias	Cruise	Matthias.Cruise@GROSBEAK.COM
108	Meenakshi	Mason	Meenakshi.Mason@JACANA.COM
109	Christian	Cage	Christian.Cage@KINGLET.COM
110	Charlie	Sutherland	Charlie.Sutherland@LIMPKIN.COM
111	Charlie	Pacino	Charlie.Pacino@LONGSPUR.COM

The developer toolbar at the bottom has several buttons: Home, Application 101, Edit Page 44 (highlighted with a red box), Session, View Debug, Debug, Show Grid, Quick Edit, Theme Roller, and a user icon. The "Edit Page 44" button is specifically highlighted.

2. On the page, in the **Shared Components > Templates** section, expand Button, right-click the **Text** button, and select **Copy**. Alternatively, you can also navigate to **Shared Components > Templates** and select the template you want to copy from the list.



3. Enter Custom–Red Button for New Template Name and click Copy.



- c. Modify the **Custom–Red Button** Template and add a class called `redButton`.
1. Click **Custom–Red Button** on the Templates page.

The screenshot shows the Oracle APEX 'Templates' page. At the top, there is a breadcrumb navigation: Application 101 > Shared Components > Templates. To the right of the breadcrumb are several icons: a speech bubble, a cursor, a triangle, a square, a checkmark, and a play button. Next to the icons is the number '44'. Below the header, a green success message says '✓ Button Template Copied.' Below the message, there are tabs for 'Templates', 'Subscription', 'Publish', 'Utilization', and 'History'. A search bar with a magnifying glass icon and a 'Go' button is followed by an 'Actions' dropdown menu. The main content area is a table with two columns: 'Type' and 'Name'. The 'Type' column includes 'Breadcrumb', 'Button', and 'HTML button (legacy - APEX 5 migration)'. The 'Name' column lists 'Breadcrumb', 'Breadcrumb Menu', 'Hierarchical Menu', 'Button', 'Button, Alternative 1 - Hot Button', 'Button, Alternative 2', 'Button, Alternative 3', and 'Custom - Red Button'. The row for 'Custom - Red Button' is highlighted with a red rectangular border and has a small hand cursor icon over the link text. The URL in the browser's address bar is 'localhost:8080/apex/f?p=4000:204:8006913046153::NO::F4000_P204_ID:189625380'.

Type	Name
Breadcrumb	Breadcrumb
Breadcrumb	Breadcrumb Menu
Breadcrumb	Hierarchical Menu
Button	Button
Button	Button, Alternative 1 - Hot Button
Button	Button, Alternative 2
Button	Button, Alternative 3
Button	<u>Custom - Red Button</u>
Button	HTML button (legacy - APEX 5 migration)

2. Enter the following tags of definition (also located in the /home/oracle/labs/files/lab_09_01.txt file) in the **Definition** section of the **Hot Template** area. Click **Apply Changes**.

```
<button value="#LABEL#" onclick="#JAVASCRIPT#" class="redButton"  
type="button" #BUTTON_ATTRIBUTES# id="#BUTTON_ID#">  
    <span width="100%">#LABEL#</span>  
</button>
```

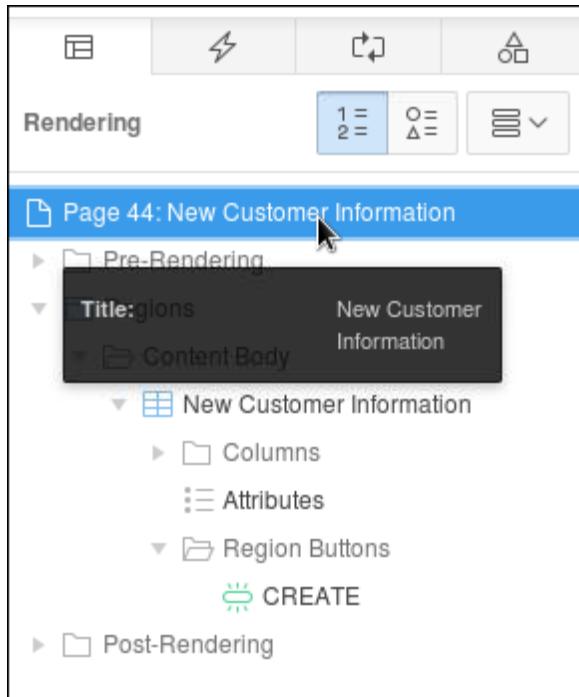
The screenshot shows the Oracle Application Express 'Button Templates' page. At the top, it says 'Button Templates: 1 of 12'. Below that, there's a 'Name' field containing 'Custom - Red Button'. To the right of the name are 'Cancel', 'Delete', and a large blue 'Apply Changes' button with a hand cursor icon over it. Below these buttons is a toolbar with tabs: 'Show All', 'Name', 'Subscription', 'Template O...', 'Definition' (which is highlighted in blue), 'Comments', and 'Substitution ...'. The 'Definition' tab shows the HTML code for the button. In the 'Hot Template' section below, the same code is displayed, with line numbers 1, 2, and 3 visible.

- d. Associate the copied template with your page by including the CSS styles to the page definition.

1. Navigate to the **New Customer Information** page.

The screenshot shows the Oracle Application Express navigation bar with 'Application 101' selected. Below the navigation bar, a green message box displays 'Action processed.' with a checkmark icon. The main content area has tabs: 'Templates' (which is highlighted in blue), 'Subscription', 'Publish', 'Utilization', and 'History'. At the bottom of the screen are search and action buttons.

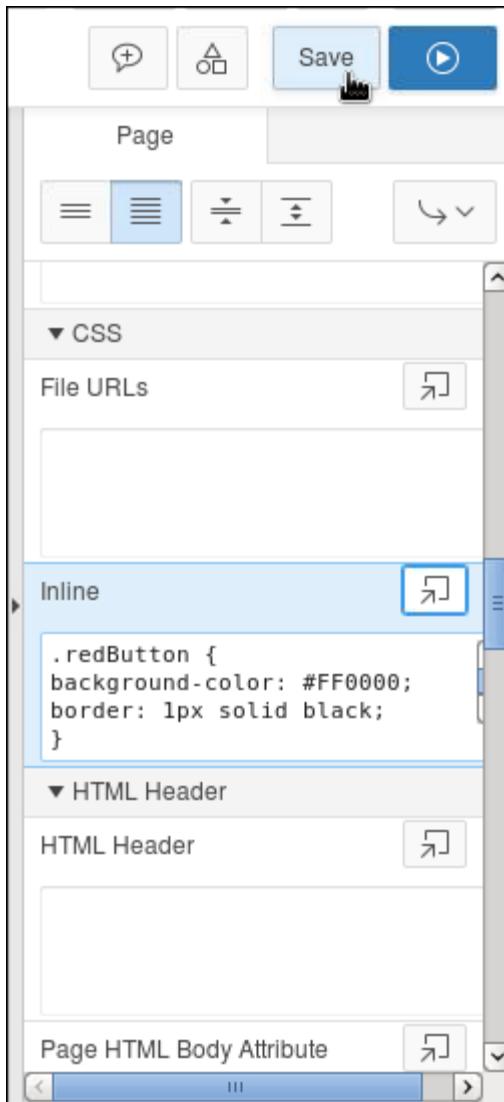
2. Click **New Customer Information** page definition.



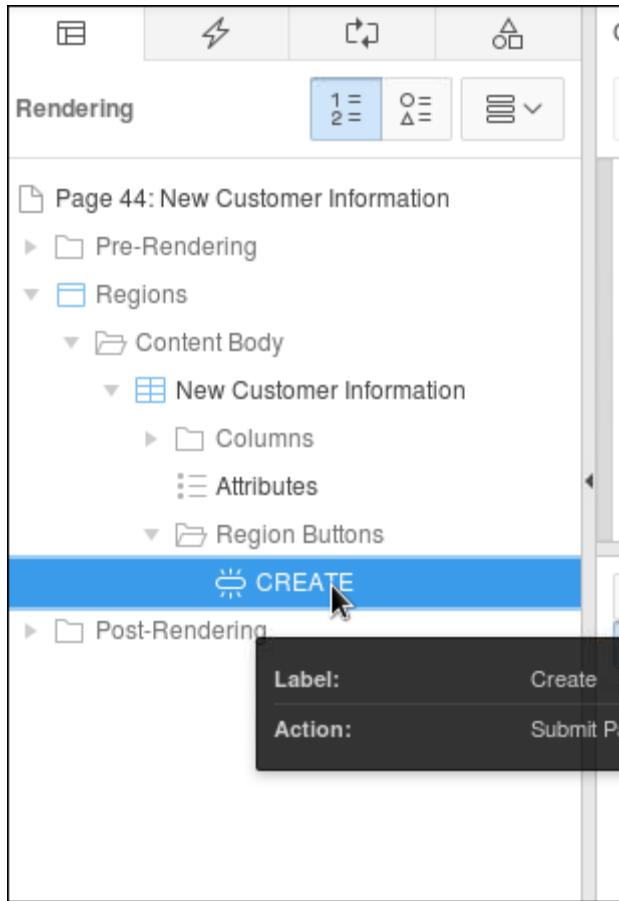
3. In the Property Editor section, navigate to CSS section.

4. In the Inline text field, enter the following CSS (also located in the /home/oracle/labs/files/lab_09_01.txt file). Click **Save**.

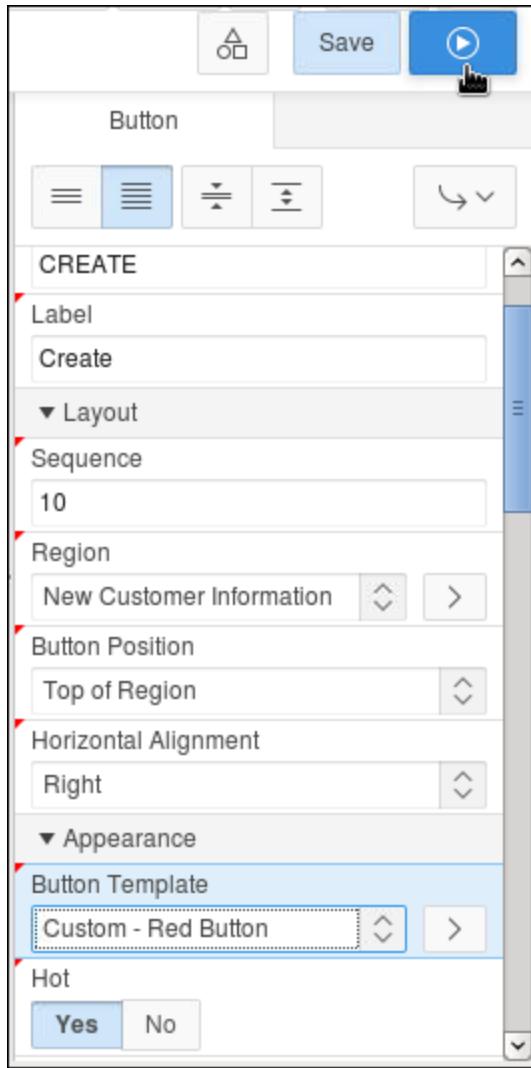
```
.redButton {  
background-color: #FF0000;  
border: 1px solid black;  
}
```



5. Click **CREATE** button in the **New Customer Information** region.



6. Select **Custom–Red Button** for **Button Template** and click **Save and Run Page**.



The color of the CREATE button has been changed.

The screenshot shows a web-based application titled "GMT | GlobalMart Management Tool". The left sidebar contains a navigation menu with items like Home, Customers (selected), New Customer Information, Orders, Products, Help, Admin, and Data Load. The main content area is titled "New Customer Information" and displays a table of customer data. The table has columns for Customer Id, Cust First Name, Cust Last Name, and Cust Email. The data consists of 12 rows, each containing a unique customer ID and names. A red "Create" button is located in the top right corner of the table header. At the bottom of the page, there is a toolbar with various icons and links, including Home, Application 101, Edit Page 44, Session, View Debug, Debug, Show Grid, Quick Edit, Theme Roller, and a user profile icon.

Customer Id	Cust First Name	Cust Last Name	Cust Email
101	Constantin	Welles	Constantin.Welles@ANHINGA.COM
102	Harrison	Pacino	Harrison.Pacino@ANI.COM
103	Manisha	Taylor	Manisha.Taylor@AUKLET.COM
104	Harrison	Sutherland	Harrison.Sutherland@GODWIT.COM
105	Matthias	MacGraw	Matthias.MacGraw@GOLDENYE.COM
106	Matthias	Hannah	Matthias.Hannah@GREBE.COM
107	Matthias	Cruise	Matthias.Cruise@GROSBEAK.COM
108	Meenakshi	Mason	Meenakshi.Mason@JACANA.COM
109	Christian	Cage	Christian.Cage@KINGLET.COM
110	Charlie	Sutherland	Charlie.Sutherland@LIMPKIN.COM
111	Charlie	Pacino	Charlie.Pacino@LONGSPUR.COM
112	Guillaume	Jackson	Guillaume.Jackson@MOORHEN.COM

Solution 9-2: Creating a Custom Theme

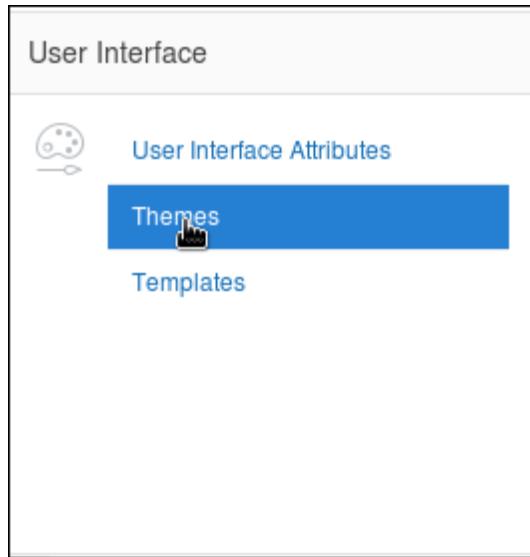
Overview

In this practice, you create a custom theme by exporting, copying, and editing your theme. You add the custom theme as a Workspace Theme so that you can make it available to any application that you create.

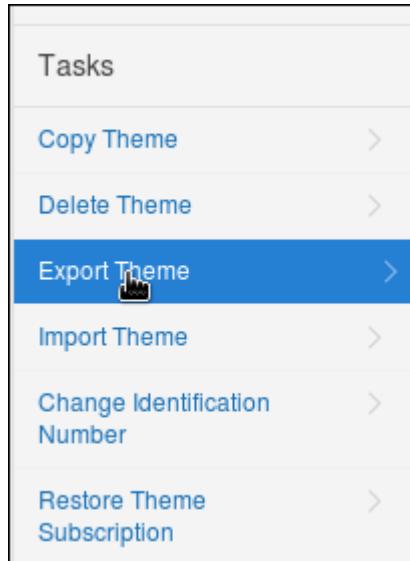
Steps

- a. You want to export your theme.

1. Navigate to the **Shared Components** page in your **GMT Application** and select Themes.



2. Select **Export Theme** from the **Tasks** list on the right side of the page.



3. Select the theme you want to export. In this case **42.UniverseTheme** theme has been exported. Click **Export**. Save the exported file.

The screenshot shows the 'Export' interface with the 'Themes' tab selected. The main title is 'Export Theme'. Below it, under 'Choose Application', the application '101 GlobalMart Management Tool' is selected. Under 'Export Theme', the theme '42. Universal Theme' is selected. The 'File Format' is set to 'UNIX' and the 'File Character Set' is 'Unicode UTF-8'. At the bottom right, there are 'Reset' and 'Export' buttons, with a cursor icon pointing to the 'Export' button.

Export

Export Applications Websheets Themes Plug-ins User Interface Defaults

Feedback

Export Theme

Choose Application

* Application 101 GlobalMart Management Tool

Export Theme

* Export Theme 42. Universal Theme

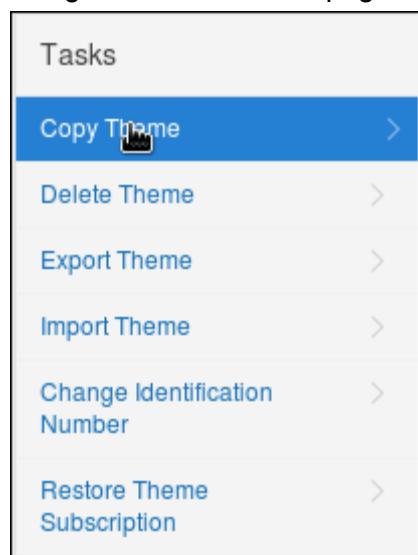
File Format UNIX

File Character Set Unicode UTF-8

Reset Export

- b. You want to copy a theme.

1. Navigate to the Themes page and click **Copy Theme** from the **Tasks** list.



2. Select the theme you want to copy in the **Copy from Theme** select list. Enter a **Theme Number** in the **Copy to this Theme ID** text field and click **Next**.

Copy Theme

Copy Theme

Each theme is identified by a numeric identification number (ID). Use this to make a copy of an existing theme and specify a new theme ID. Copying a theme is useful if you wish to export a theme with a different ID.

Application: 101 GlobalMart Management Tool ?

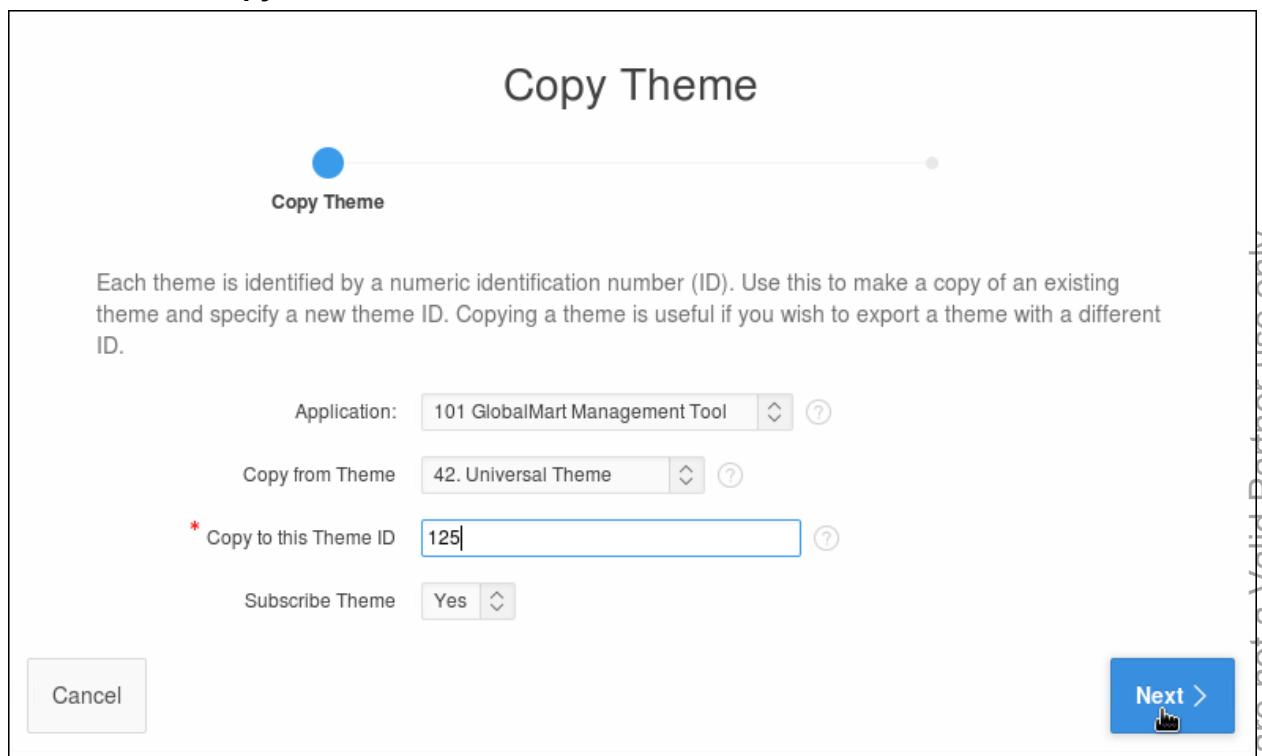
Copy from Theme: 42. Universal Theme ?

* Copy to this Theme ID: ?

Subscribe Theme: Yes ?

Cancel

Next >



3. Click **Copy Theme**.

Copy Theme

✓ Confirm

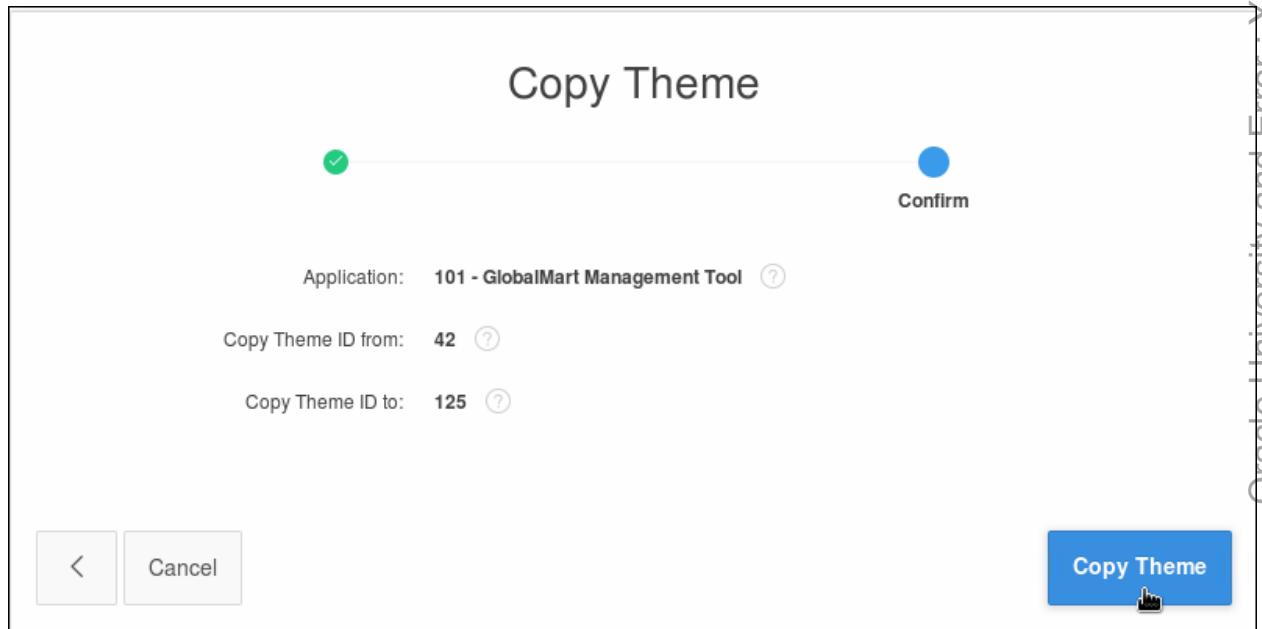
Application: 101 - GlobalMart Management Tool ?

Copy Theme ID from: 42 ?

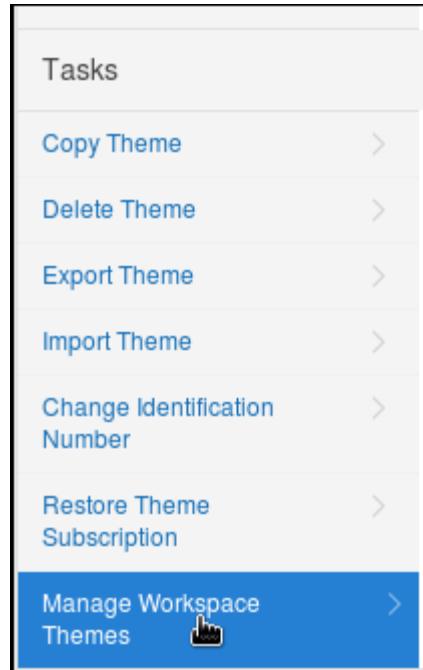
Copy Theme ID to: 125 ?

< Cancel

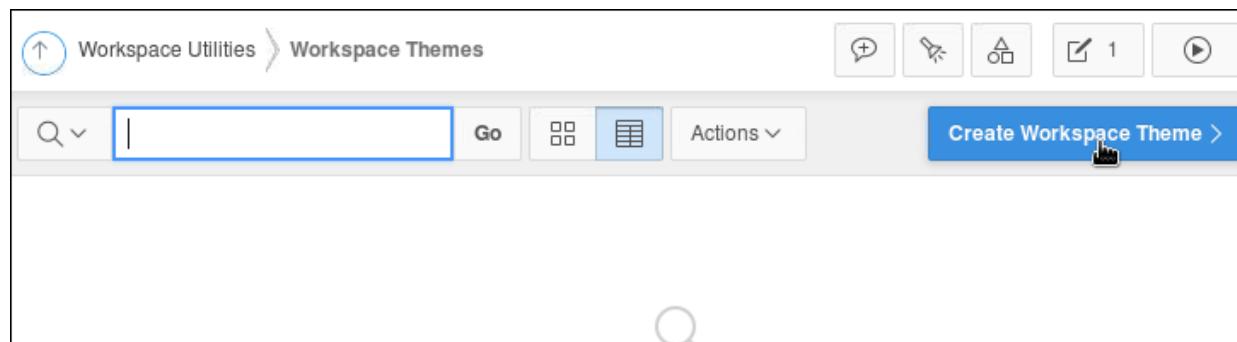
Copy Theme



- c. You want to make the custom theme available to any application you create.
 1. Log out of the workspace and log in to the workspace as an administrator. In this case log in as `apex_admin`.
 2. Navigate to the **GlobalMart Management Tool** and select **Shared Components**. Select **Themes** under **User Interface**.
 3. Click **Manage Workspace Themes** from the **Tasks** list.



4. Select **Create Workspace Theme**.



5. Select the Application and select the Application Theme to Copy – in this case **125. Universal Theme** and click **Next**.

Create Workspace Theme

Identify Theme

Select an application theme to create within the workspace theme repository.

* Application 101. GlobalMart Management Tool

* Application Theme to Copy 125. Universal Theme

Cancel Next >

6. Enter the **Theme Name** as Custom Universal – Master Theme. Click **Next**.

Create Workspace Theme

Identify Name

You can change the name and number for the new workspace theme to be created.

Selected Theme: 125. Universal Theme

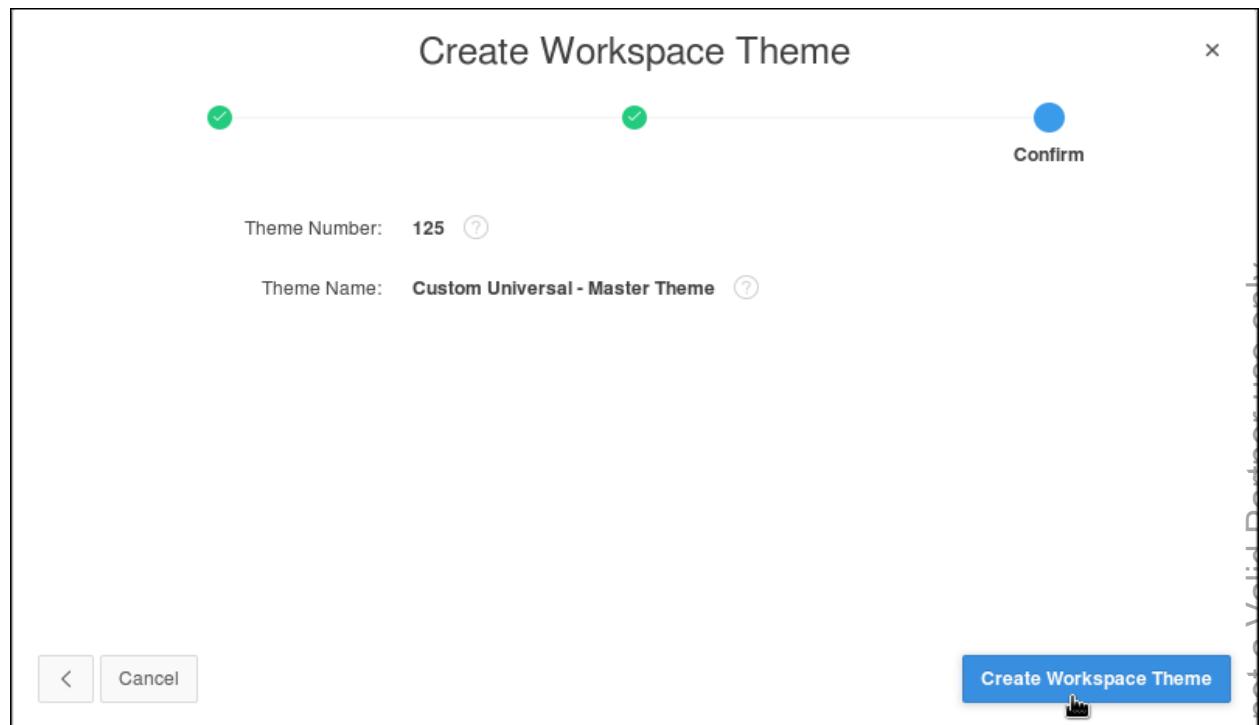
* Theme Number 125

* Theme Name Custom Universal - Master Theme

Description

< Cancel Next >

7. Click Create Workspace Theme.



Solution 9-3: Creating a Master Application

Overview

In this practice, you create an application with the custom theme that all groups will subscribe to.

Steps

- a. Create an application with login and home pages.
 1. Navigate to Application Builder home page and select Database Applications. Click Create.
 2. Select **Desktop** and click **Next**.
 3. Enter **Master Application** for **Name** and select **Midnight Blue (3)** for Theme. Click **Create Application**.

The screenshot shows the 'Create an Application' wizard. The first step, 'Name', is completed. The form fields are as follows:

- User Interface: Desktop
- Schema: APEX
- Name: Master Application
- Application: 107
- Theme: Midnight Blue (3)
- Theme Style: No Theme Style Selected

At the bottom, there are 'Cancel' and 'Create Application' buttons, with 'Create Application' being the active button.

4. Click **Create Application**.

Create an Application

You have requested to create an application with the following attributes. Please confirm your selections.

Application	107
Name	Master Application
Parsing Schema	APEX
Default Language	en
Navigation	One Level of Tabs
Default Authentication Scheme	APEX
Theme Type	Standard
Theme	Midnight Blue
Subscribe Theme	No

< Cancel Create Application

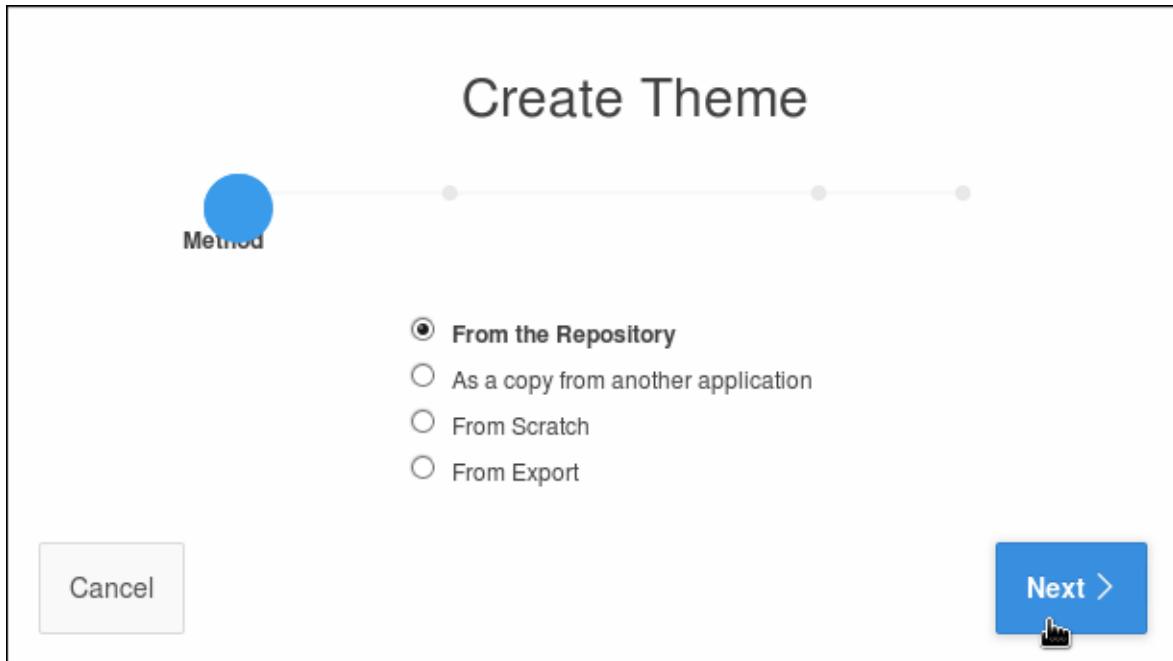
5. The application is created.

- b. You want to add the custom theme that you created in the previous practice to this application.
1. Click **Shared Components**.
 2. Select **Themes**.
 3. Click **Create**.

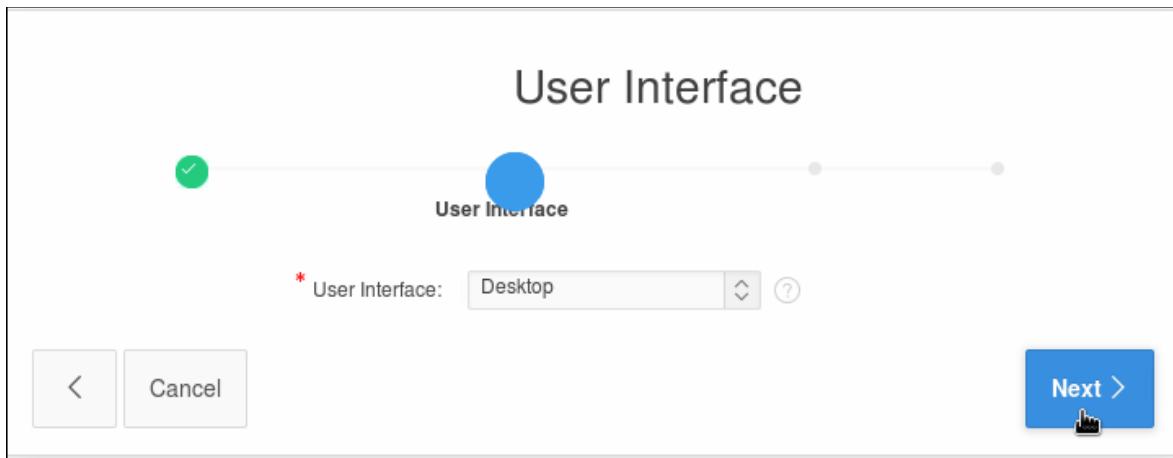
Themes Reports History

Number	Name	User Interface	Is Current	Subscribed From	Subscribers	Templates	Page Templates	Region Templates	Button Templates
3	Midnight Blue - 3 *	Desktop	✓	Theme Repository		80	15	24	

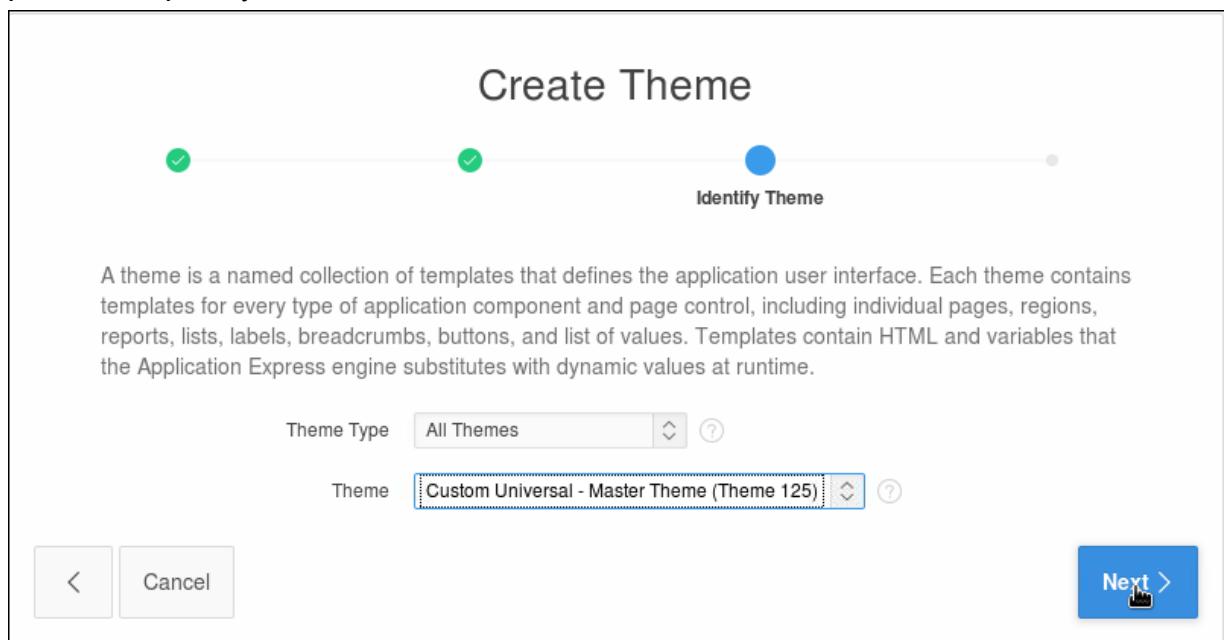
4. Select **From the Repository** and click **Next**.



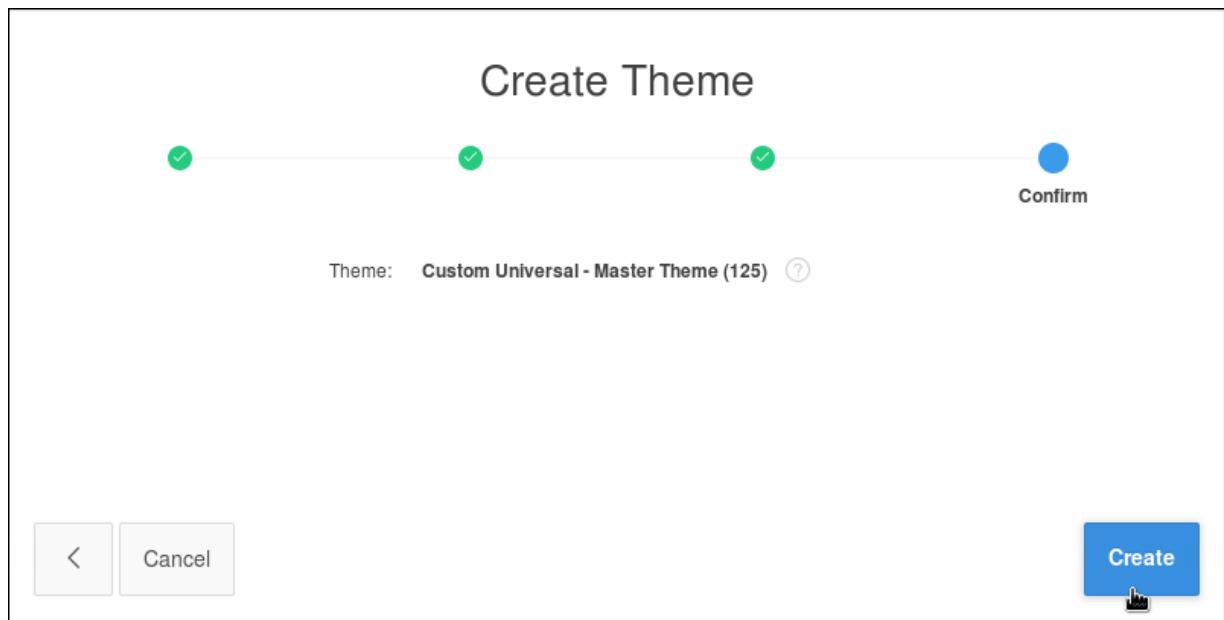
5. Select Desktop for **User Interface** and click **Next**.



6. Select **All Themes** for **Theme Type**. Select **Custom Universal – Master Theme (Theme 125)** that you created and click **Next**.

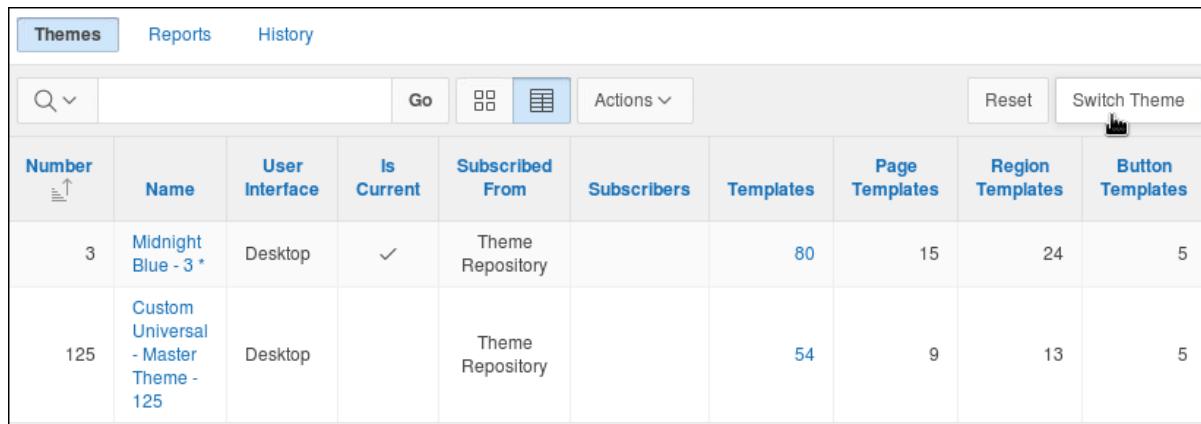


7. Click **Create**.



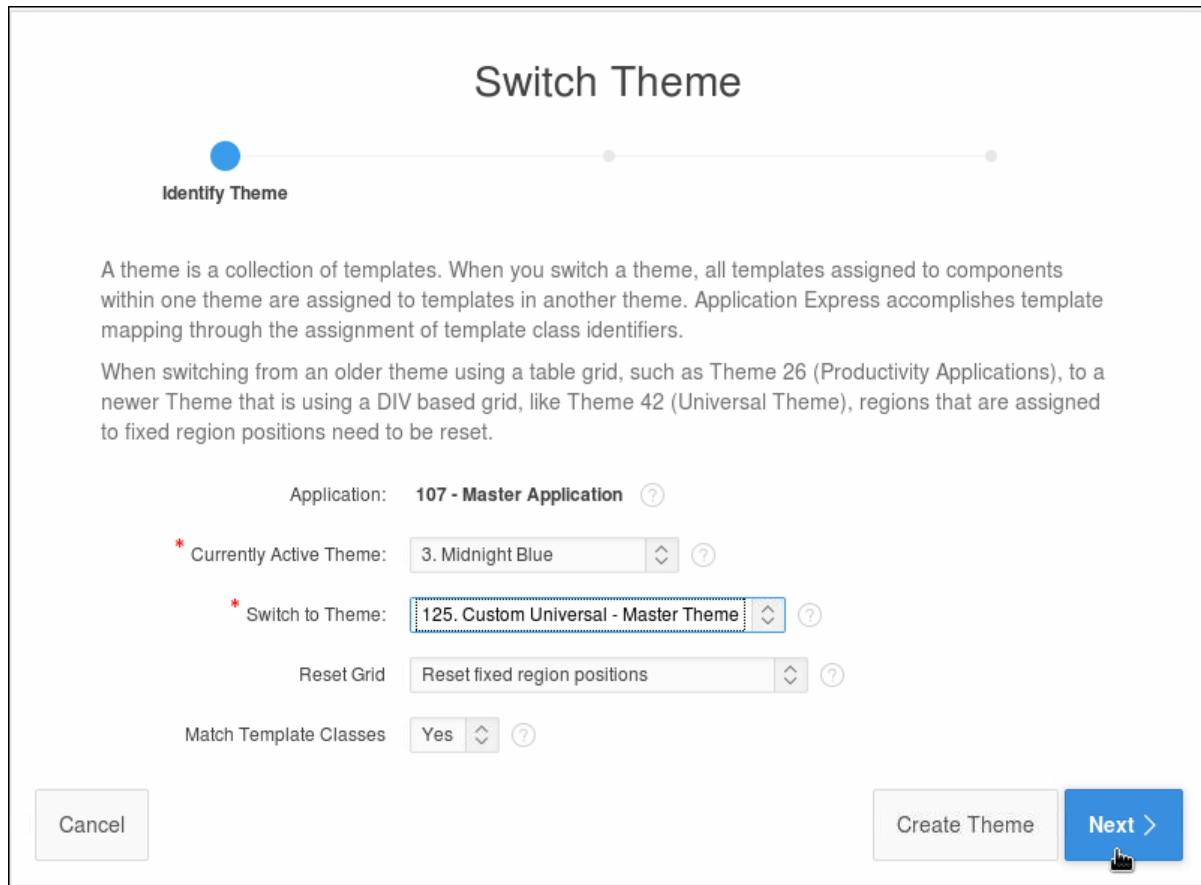
- c. You want to switch the theme to make the Custom Universal - Master Theme (125) your current theme.

- Click **Switch Theme**.



Number	Name	User Interface	Is Current	Subscribed From	Subscribers	Templates	Page Templates	Region Templates	Button Templates
3	Midnight Blue - 3 *	Desktop	✓	Theme Repository		80	15	24	5
125	Custom Universal - Master Theme - 125	Desktop		Theme Repository		54	9	13	5

- Select the **Currently Active Theme** as **3. Midnight Blue** and **Switch to Theme** as **125. Custom Universal- Master Theme**, and click **Next**.



Switch Theme

Identify Theme

A theme is a collection of templates. When you switch a theme, all templates assigned to components within one theme are assigned to templates in another theme. Application Express accomplishes template mapping through the assignment of template class identifiers.

When switching from an older theme using a table grid, such as Theme 26 (Productivity Applications), to a newer Theme that is using a DIV based grid, like Theme 42 (Universal Theme), regions that are assigned to fixed region positions need to be reset.

Application: 107 - Master Application

* Currently Active Theme: 3. Midnight Blue

* Switch to Theme: 125. Custom Universal - Master Theme

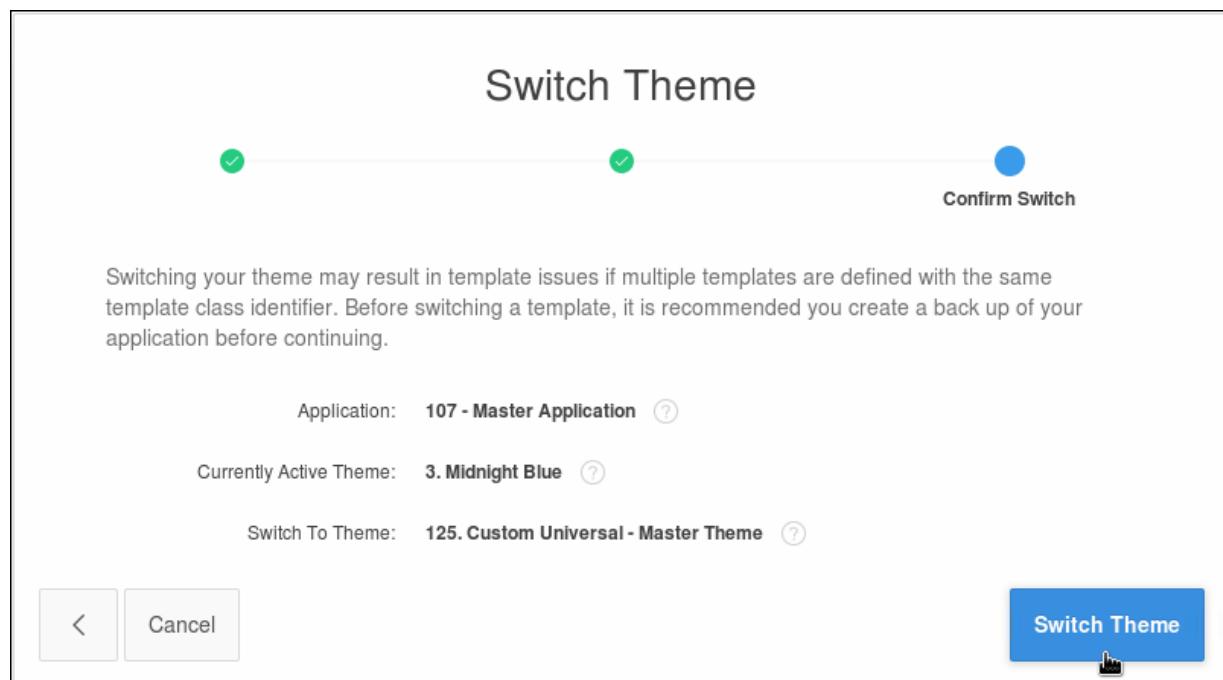
Reset Grid: Reset fixed region positions

Match Template Classes: Yes

Cancel Create Theme Next >

- View the **Theme Compatibility** window and click **Next**.

4. Click Switch Theme.



You now have a Master Application with a Custom Master theme which you can use as a base application for creating customized applications.

Note: If you have not been able to complete practice for this lesson, you can import the `sol_09_01.sql` exported application available in the `/home/oracle/labs/solutions` folder.

Practices for Lesson 10: Developing a New Theme for Your Application Using Theme Roller

Chapter 10

Practices for Lesson 10: Overview

Practices Overview

In these practices, you develop a new theme for the GMT Application using Theme Roller.

Practice 10-1: Developing a New Theme for the GMT Application

Overview

In this practice, you develop a new theme for the GMT application using Theme Roller.

Assumptions

You should have completed Practice 9.

Tasks

- a. Navigate to Application Builder, run the GMT application, and use Theme Roller to modify the look and feel of the application.

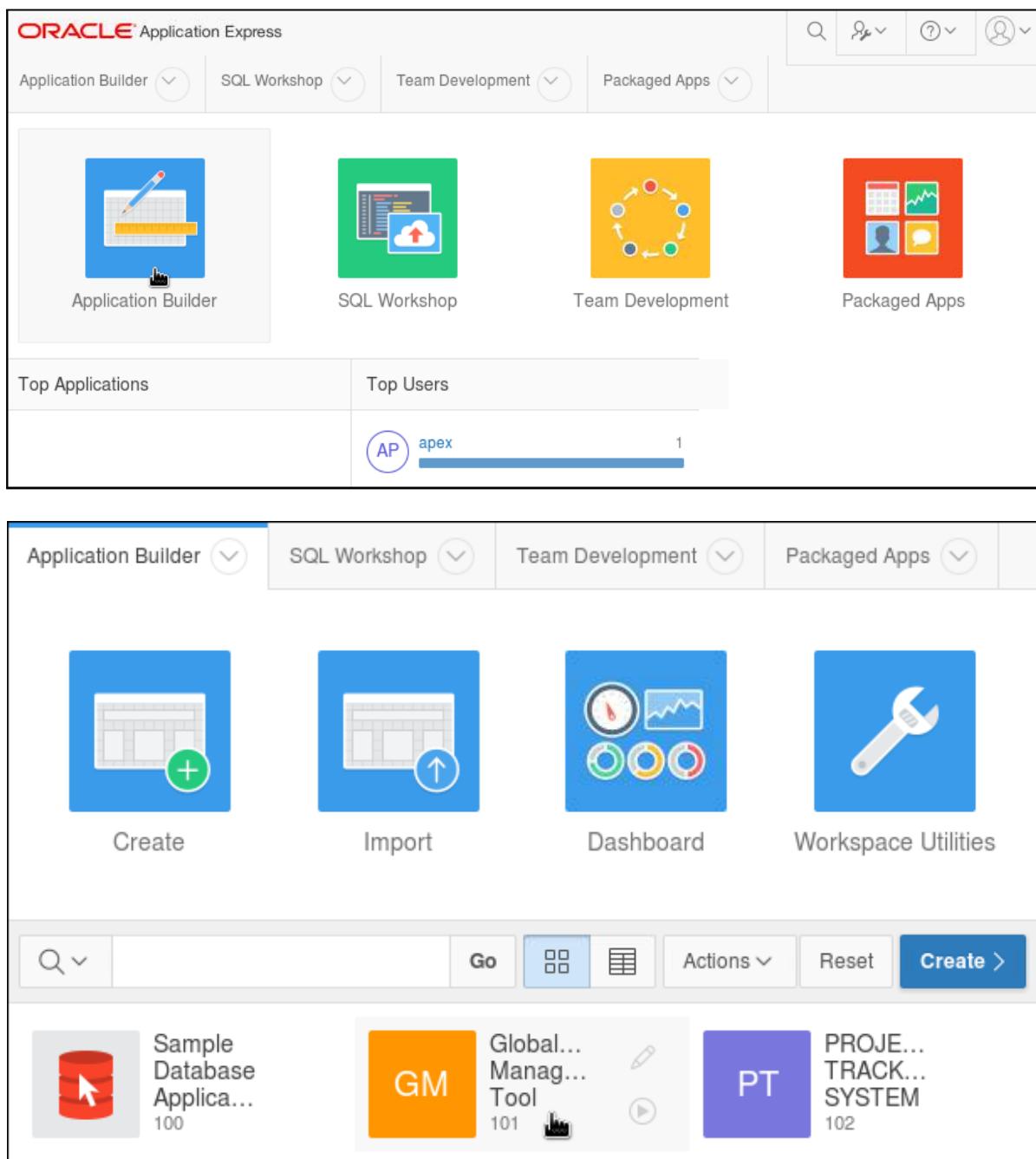
Solution 10-1: Developing a New Theme for the GMT Application

Overview

In this practice, you develop a new theme for the GMT application using Theme Roller.

Steps

- Navigate to Application Builder, run the GMT application, and use Theme Roller to modify the look and feel of the application.
- Run the GMT application.

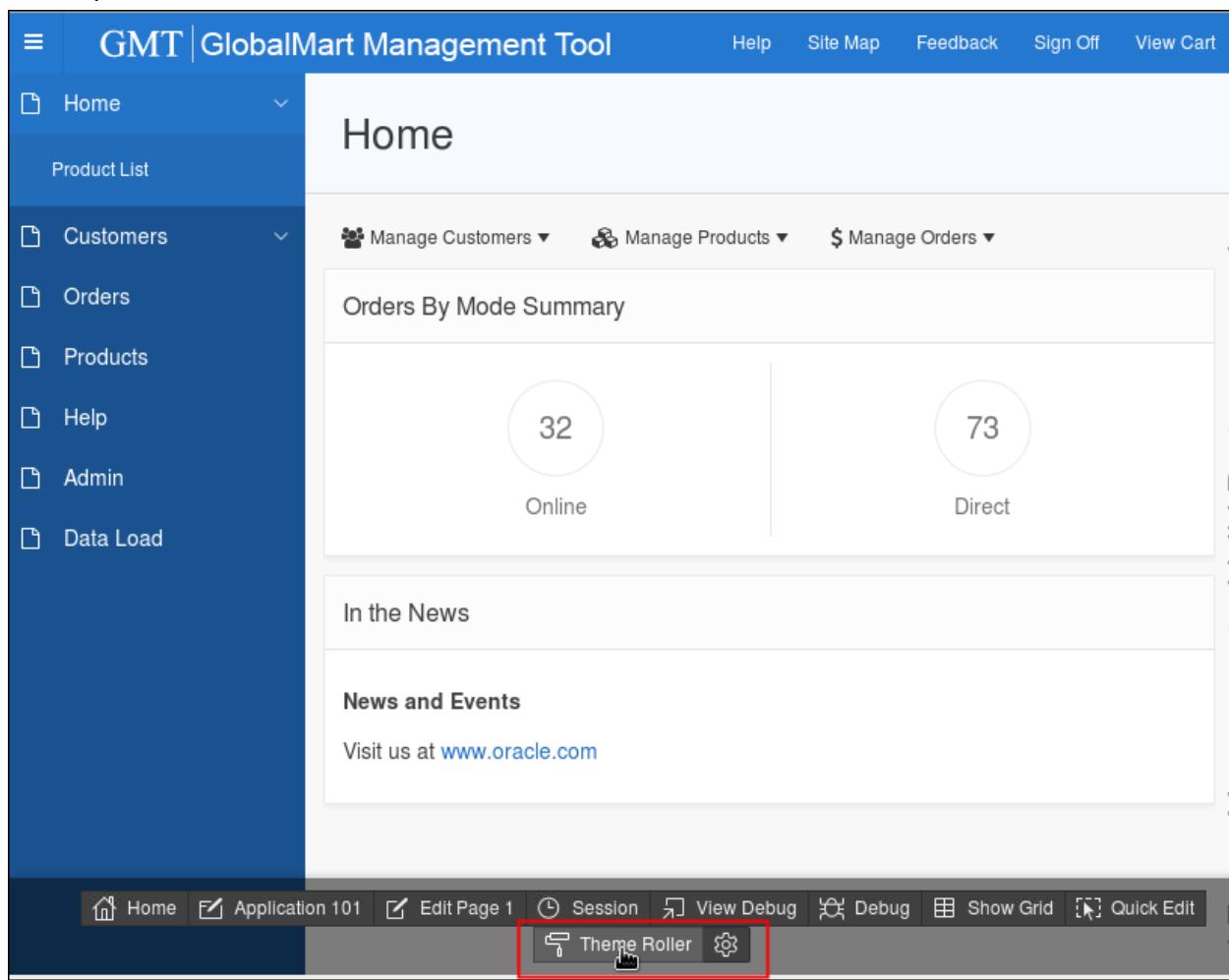


2. Enter Username and Password. Click **Log In**.

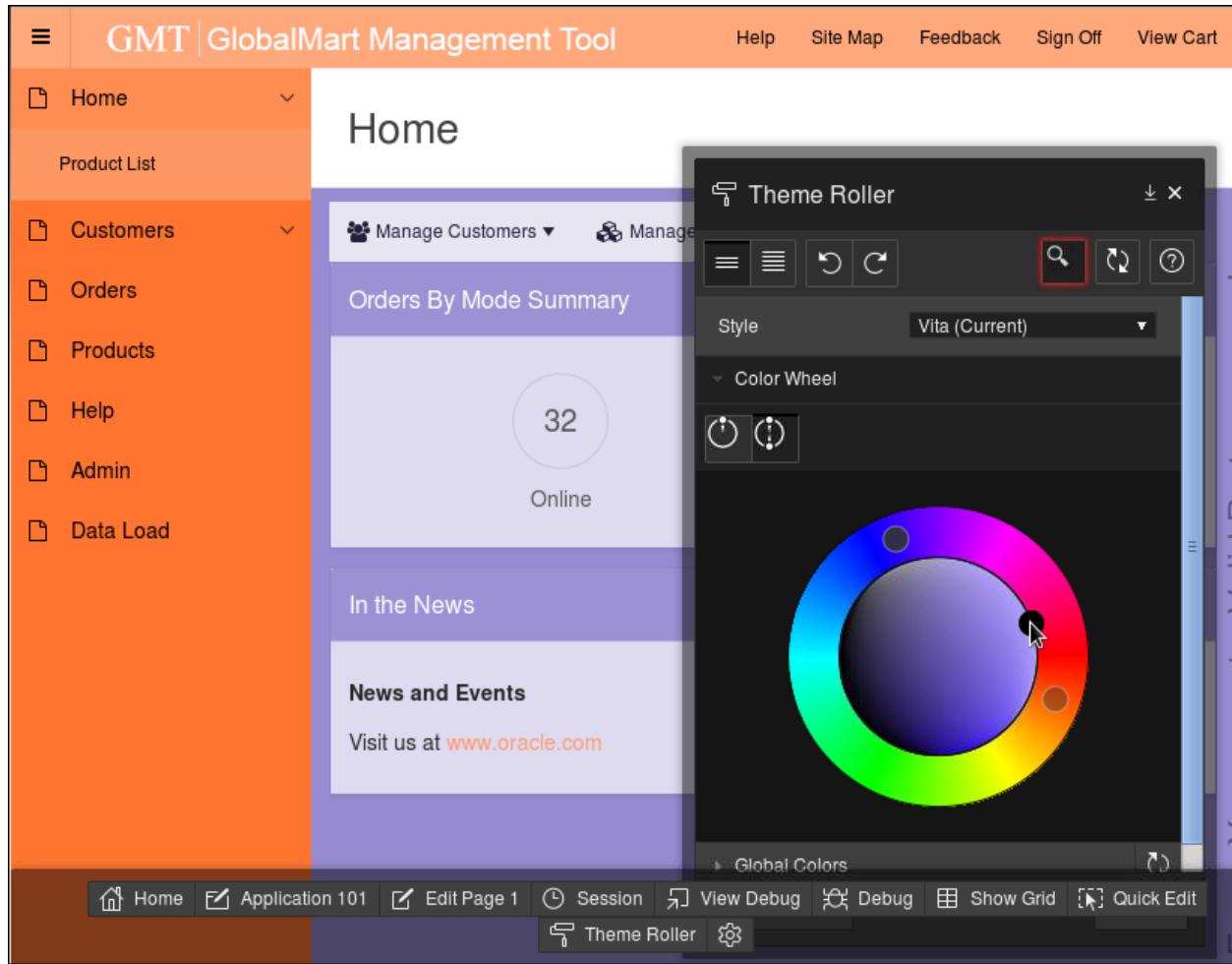
The image shows a login interface with the following elements:

- A title "Log In" at the top.
- A "Username" field containing "apex".
- A "Password" field containing "****".
- A blue "Log In" button with a hand cursor icon pointing to it.

3. You want to change the look and feel of the application. Click **Theme Roller** in the Developer toolbar.



4. Change the color of the application by moving the black dots in the Color Wheel.

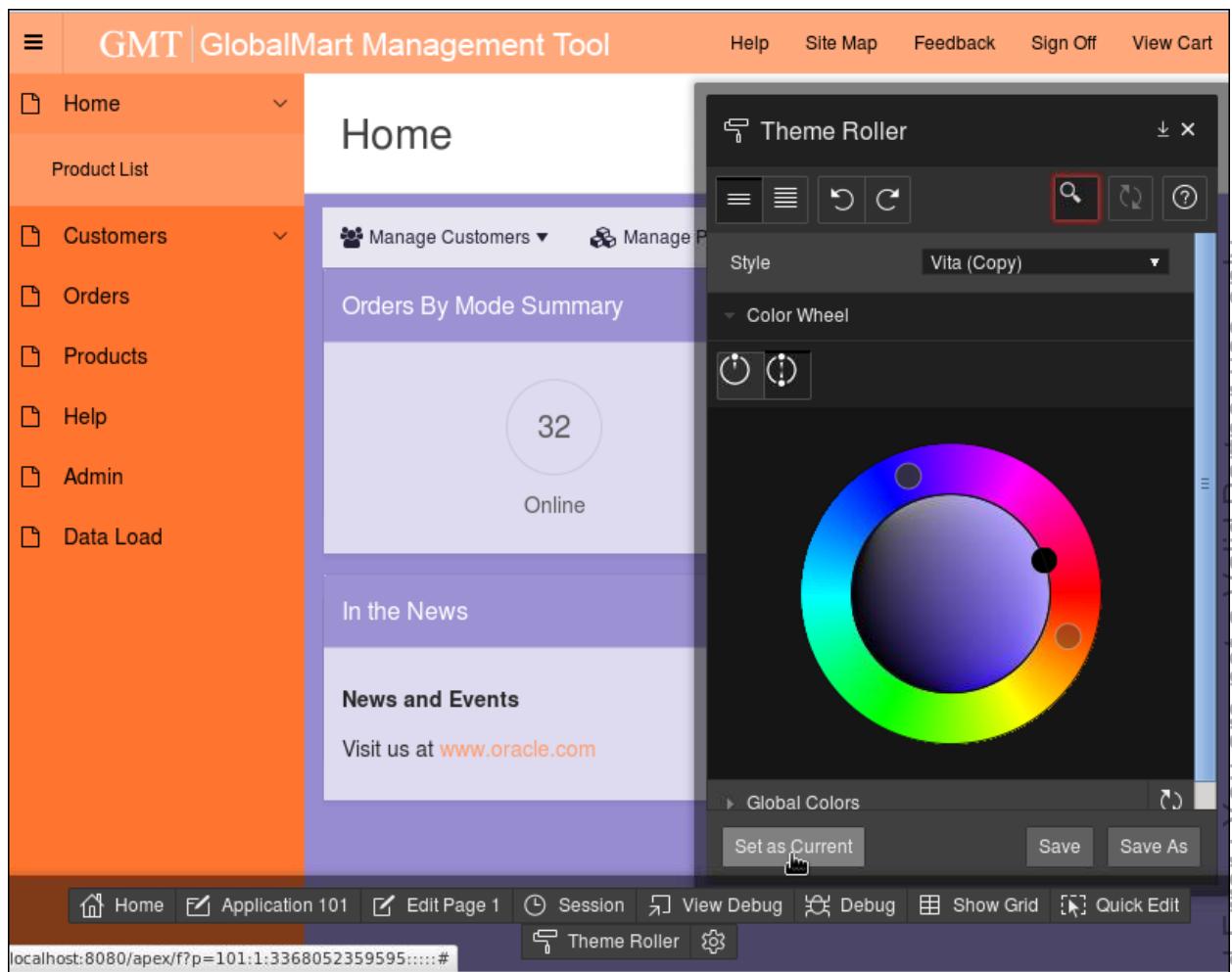


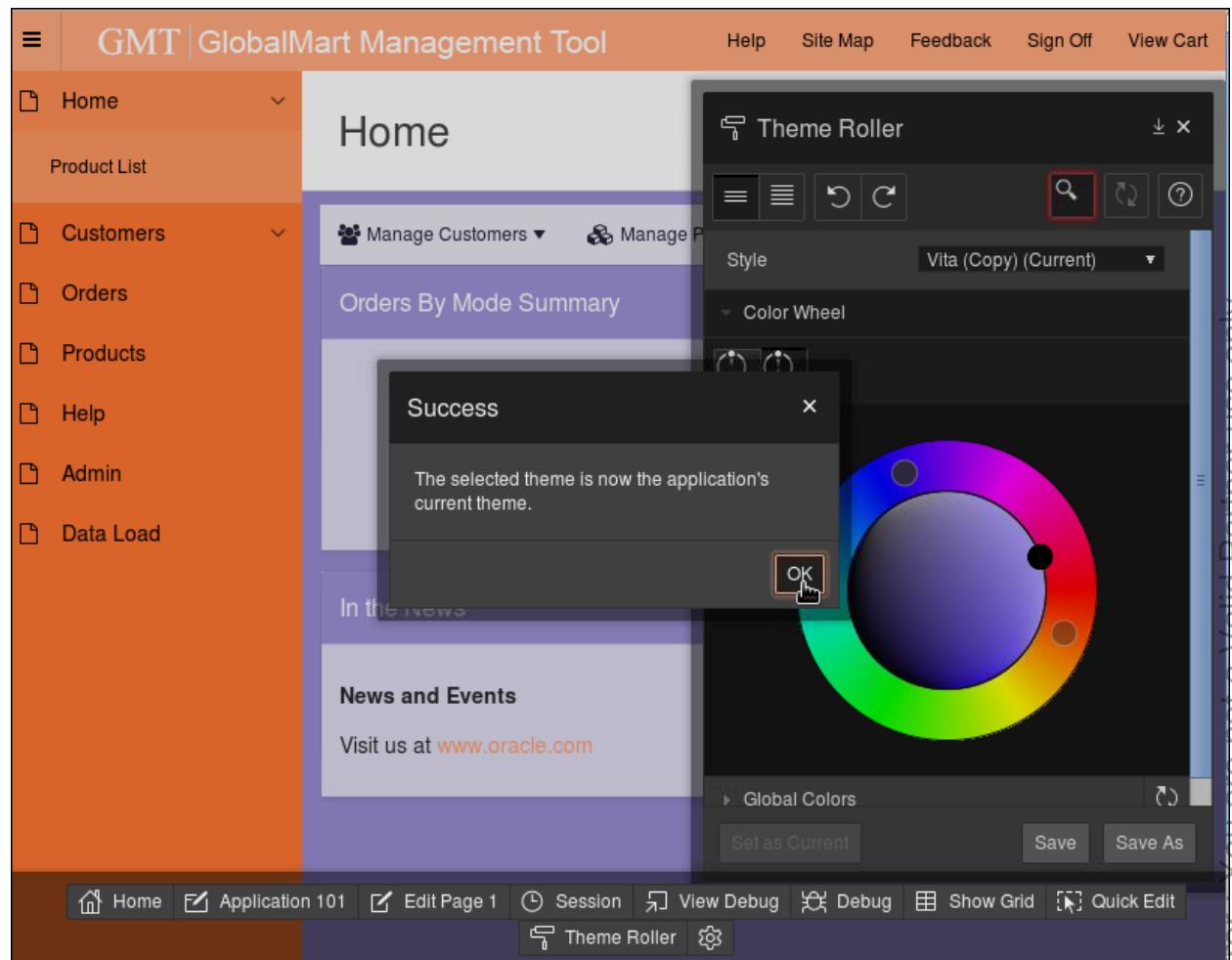
5. Click **Save As** to save your changes as a new theme.

The screenshot shows the GlobalMart Management Tool interface. On the left is a sidebar with links: Home, Product List, Customers, Orders, Products, Help, Admin, and Data Load. The main content area displays a dashboard with sections for Manage Customers, Manage Products, Orders By Mode Summary (showing 32 Online), In the News, and News and Events. A link to visit oracle.com is present. On the right is the Theme Roller panel, which includes a color wheel, style selection dropdown (set to 'Vita (Current)'), and various theme settings. A red box highlights the 'Save As' button in the Theme Roller toolbar. Below the main window is a status bar with navigation icons and the URL 'localhost:8080/apex/f?p=101:1:3368052359595::::#'. A separate 'Save As' dialog box is overlaid on the bottom left, prompting for a new style name. The 'Style Name' field contains 'Vita (Copy)' and the 'Save' button is highlighted with a red box.

Oracle University and Error : You are not a Valid Partner use only

6. Click **Set as Current** to set the saved theme as the new theme for the application.





Note: If you have not been able to complete the practice for this lesson, you can import the `sol_10_01.sql` exported application available in the `/home/oracle/labs/solutions` folder.

Practices for Lesson 11: Securing an Application

Chapter 11

Practice 11-1: Preventing SQL Injection Attacks

Overview

In this practice, you create a new page with a report region using a dynamic SQL Query. You test the page for SQL Injection and then fix the query to prevent SQL Injection.

Assumptions

None.

Tasks

Note: You might be logged in to the APEX workspace as the apex_admin user while performing the previous practice. Log out and log in again as the APEX user.

- a. Create a new blank page called Find Employee Details in the GMT application. Create a Static Content region called Locate Employee on this new page. Create a text item called P<n>_SEARCH. Create a button called Go that submits the page when clicked.
- b. On the Find Employee Details page, create a region called Employee Details of type PL/SQL Dynamic Content using the following code:

```
FOR c1 IN (SELECT first_name FROM oechr.employees  
WHERE employee_id = &P43_SEARCH.) LOOP  
  http.p('<br>Employee Name: ' || c1.first_name);  
END LOOP;
```

- c. Run the page. Enter an employee ID such as 103 in the search text field and click the Go button. What do you see in the report region?
- d. Enter “1 or 0=0” in the search text field and click the Go button. What do you see in the report region?
- e. Now, enter the following text in the search text field and click the Go button. What do you see in the report region?

```
103 union all select username from all_users
```
- f. Fix the security issue seen here and confirm that this page is safe from SQL Injection Attacks.

Solution 11-1: Preventing SQL Injection Attacks

Overview

In this solution, the steps to create a new page with a report region using a dynamic SQL Query are given. Also, the steps to test the page for SQL Injection and then fix the query to prevent SQL Injection are provided.

Assumptions

None.

Steps

- Note:** You might be logged in to the APEX workspace as the apex_admin user while performing the previous practice. Log out and log in again as the APEX user.
- Create a new blank page called Find Employee Details in the GMT application. Create a Static Content region called Locate Employee on this new page. Create a text item called P<n>_SEARCH and give it a default value of 103. Create a button called Go that submits the page when clicked.

1. Navigate to the GMT application development home page and click **Create Page**.



2. Select **Blank Page**. Complete the wizard by specifying the page name as **Find Employee Details** and selecting the default values for the rest of the options. Click **Finish** on the last page of the wizard to create the page.

Application	101
Page	45
Page Name	Find Employee Details
Page Title	Find Employee Details

Finish

A screenshot of the 'Create Page' wizard step 1 of 3. It shows a table with four rows: Application (101), Page (45), Page Name (Find Employee Details), and Page Title (Find Employee Details). At the bottom right of the table is a blue 'Finish' button.

3. In Page Designer view, right-click the **Content Body** node and select **Create Region**. Enter **Locate Employee** as the region name.

▼ Identification	
Title	Locate Employee
Type	Static Content

A screenshot of the 'Create Region' dialog box in the Page Designer. It has a title bar '▼ Identification'. Under 'Title', the text 'Locate Employee' is entered. Under 'Type', the text 'Static Content' is selected.

- Right-click the **Locate Employee** region and select **Create Page Item**. Name the item **P<n>_SEARCH**.

▼ Identification

Name	P45_SEARCH
Type	Text Field

- Under Default, select **Static Value** for Type and enter **103** in the value field.

▼ Default

Type	Static Value
Static Value	103

- Right-click the **Items** node and select **Create Button**. Enter **GO** for Button name and label.

▼ Identification

Button Name	GO
Label	Go

- Select **No** for Start New Row.

▼ Grid

Start New Row	Yes	No
Column	Automatic	
New Column	Yes	No
Column Span	Automatic	

- Save and run the page to ensure the items are displayed. Click the edit page link.

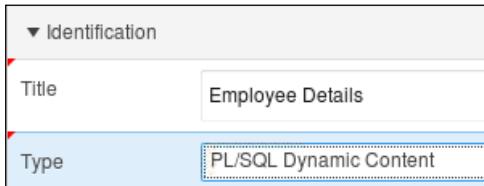
Locate Employee

Search	103
Go	

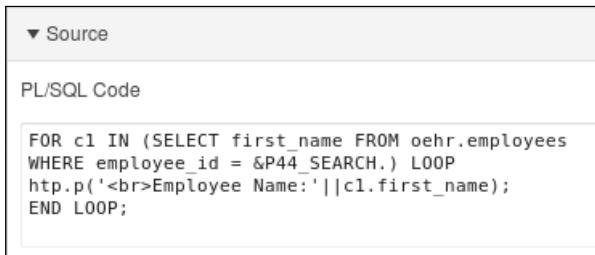
- b. On the Find Employee Details page, create a region called Employee Details of type PL/SQL Dynamic Content using the following code:

```
FOR c1 IN (SELECT first_name FROM oechr.employees
WHERE employee_id = &P44_SEARCH.) LOOP
htp.p('<br>Employee Name: '||c1.first_name);
END LOOP;
```

1. On the Find Employee Details page, right-click the **Content Body** node and select **Create Region**. Enter **Employee Details** for Title and select **PL/SQL Dynamic Content** for Type.



2. Enter the given code in the PL/SQL code (available in lab_11_01.txt) field and save the page.



- c. Run the page. What do you see in the report region?

1. Run the page. Log in if prompted. Since you have provided a default value for the search field, you should see the first name of the employee with employee_id 103 displayed.



- d. Enter “1 or 0=0” in the search text field and click the Go button. What do you see in the report region?

1. Enter “1 or 0=0” in the search field and click **Go**.

The screenshot shows a search interface with a purple header bar containing the text "Locate Employee". Below the header is a search form with a "Search" button, a text input field containing "1 or 0=0", and a "Go" button.

2. You will see all the employee names listed in the region.

The screenshot shows a report region with a purple header bar containing the text "Employee Details". The main body of the report lists 15 employee names:

- Employee Name: Ellen
- Employee Name: Sundar
- Employee Name: Mozhe
- Employee Name: David
- Employee Name: Hermann
- Employee Name: Shellie
- Employee Name: Amit
- Employee Name: Elizabeth
- Employee Name: Sarah
- Employee Name: David
- Employee Name: Laura
- Employee Name: Harrison
- Employee Name: Alexis
- Employee Name: Anthony
- Employee Name: Gerald
- Employee Name: Nanette
- Employee Name: John

- e. Now, enter the following text in the search text field and click the Go button. What do you see in the report region?

```
103 union all select username from all_users
```

1. Enter the given code in the search field and click **Go**.

Locate Employee

Search Go

2. You will see the data from the `all_users` table exposed in the report region.

Employee Details

Employee Name: Alexander
Employee Name: ORDS_PUBLIC_USER
Employee Name: ORDS_METADATA
Employee Name: PTS
Employee Name: OEHR
Employee Name: APEX
Employee Name: APEX_REST_PUBLIC_USER
Employee Name: APEX_LISTENER
Employee Name: APEX_050000
Employee Name: APEX_PUBLIC_USER
Employee Name: FLOWS_FILES
Employee Name: SCOTT
Employee Name: OE
Employee Name: PM
Employee Name: SH
Employee Name: IX

- f. Fix the security issue seen here and confirm that this page is safe from SQL Injection Attacks.

1. Enter an employee number such as **102** and click **Go**. Confirm that the employee name is displayed.

Locate Employee

Search Go

Employee Details

Employee Name: Lex

release 1.0 Set Screen Reader Mode On

2. Click the edit page link in the Developer toolbar. Click the **Employee Details** Dynamic PL/SQL Content region. Change the source code to use bind variable.

The screenshot shows a PL/SQL Code editor window. The code is as follows:

```
PL/SQL Code
FOR c1 IN (SELECT first_name FROM oechr.employees
WHERE employee_id = :P45_SEARCH LOOP
  htp.p('<br>Employee Name: '||c1.first_name);
END LOOP;
```

A red box highlights the line `:P45_SEARCH`, indicating it is a bind variable.

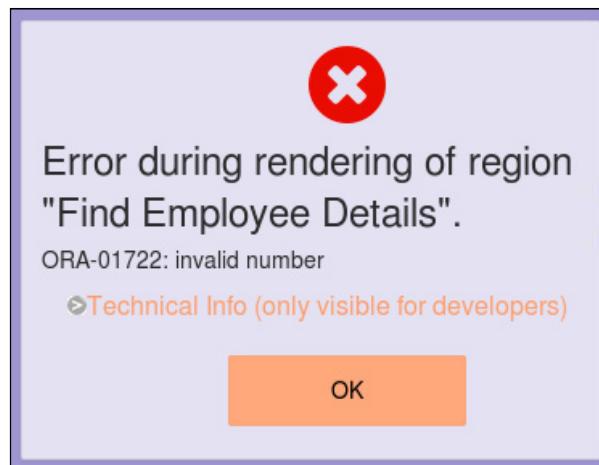
3. Save and run the page. You will see the employee name of employee 102 displayed.

The screenshot shows a page titled "Locate Employee". The search field contains "102". The "Employee Details" report section displays "Employee Name: Lex".

4. Enter the SQL Injection code and click Go.

The screenshot shows the same "Locate Employee" page. The search field now contains the SQL injection code `|102 union all select username|`. The "Go" button is highlighted with a cursor icon.

5. You will now see that the report region throws an error and SQL Injection is prevented.



Practices for Lesson 12: Deploying an Application

Chapter 12

Practice 12-1: Identifying and Creating Supporting Objects

Overview

In this practice, you prepare the GMT application for deployment. For this, you identify the supporting objects for the application and create the required installation and de-installation scripts.

Assumptions

None.

Tasks

- a. Identify the supporting objects for the GMT application.
- b. Add pre-installation scripts to check if user has required privileges and space.
- c. Add the access control objects and customers and employees tables to the prerequisite section.
- d. Add installation scripts to create the customers and employees tables. (In a real world situation you would want to add all the tables. Here, for the sake of the labs, creating only two is ok)
- e. Add the installation scripts to create the access control objects.
- f. Add the ~/labs/files/customers.sql file called insert_cust to the installation scripts. This will populate the customers table.
- g. Create an installation script from scratch called insert_emp and use the text from the ~/labs/files/emp.txt file to create the script.
- h. Add a custom installation message.
- i. Add de-installation script that will drop the customers and employees tables.

Solution 12-1: Identifying and Creating Supporting Objects

Overview

In this solution, you prepare the GMT application for deployment. The steps to identify the supporting objects for the application and create the required installation and de-installation scripts are provided.

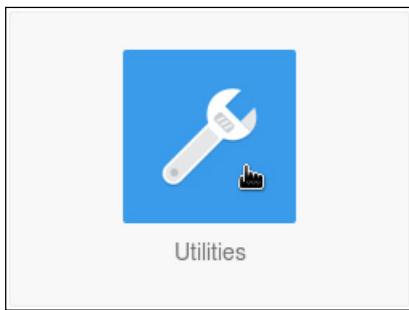
Assumptions

None.

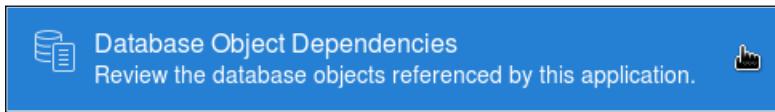
Tasks

- a. Identify the supporting objects for the GMT application.

1. From the GMT application development home page, click the **Utilities** icon.



2. Click **Database Object Dependencies**.



3. Click the **Compute Dependencies** button.



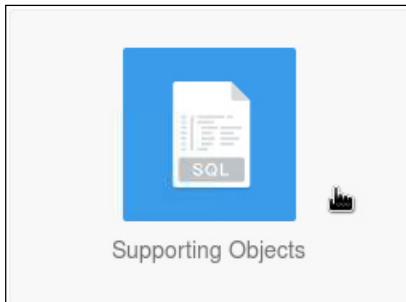
4. A list of all the database objects referenced by the application is displayed. The number of references is also mentioned.

Owner	Referenced Name	Referenced Type	Reference Count
APEX_050000	WWV_FLOW_GLOBAL	Package	2
OEHR	APEX_ACCESS_CONTROL	Table	4
	APEX_ACCESS_SETUP	Table	4
	AUDITS	Table	1
	AUTHORIZATION_CHECK	Function	1
	COUNTRIES	Table	2
	CUSTOMERS	Table	14
	CUSTOMERS_SEQ	Sequence	1
	DEPARTMENTS	Table	2
	EMPLOYEES	Table	9

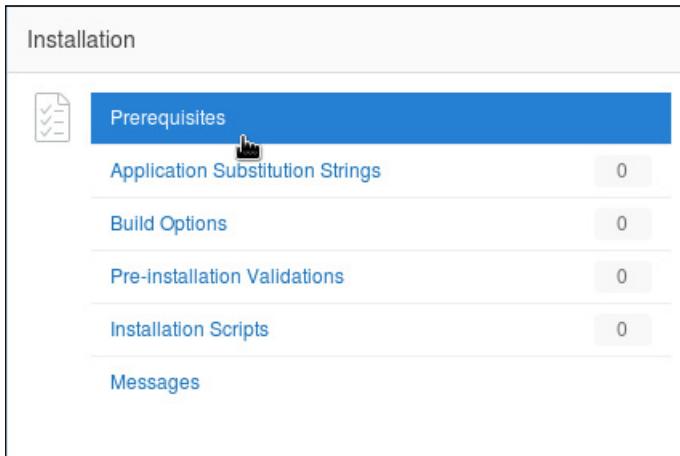
- b. Add pre-installation scripts to check if user has required privileges and space.
1. Navigate to the application development home page.



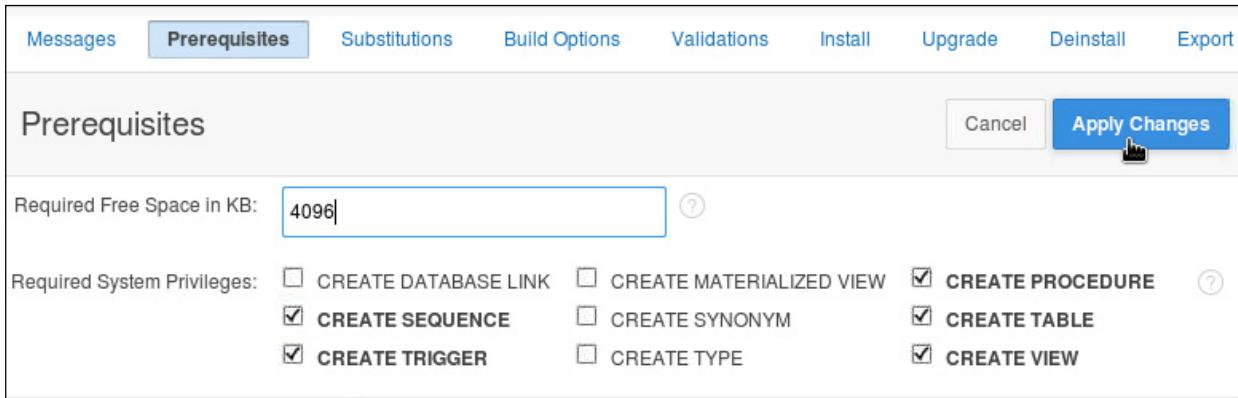
2. Click **Supporting Objects**.



3. Under Installation, click **Prerequisites**.

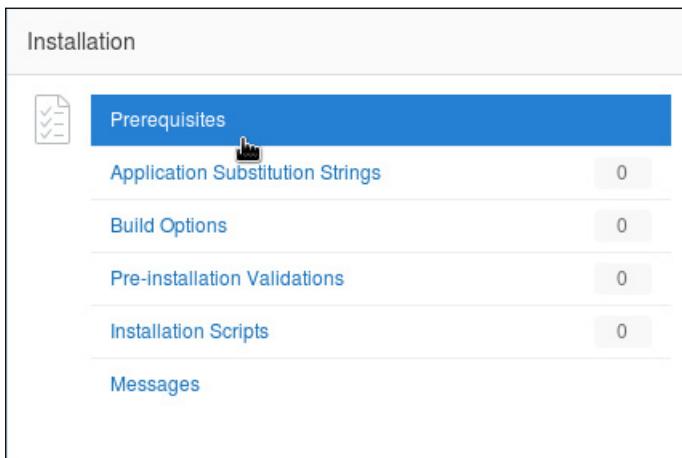


4. Enter the required free space and select the required privileges and click **Apply Changes**.

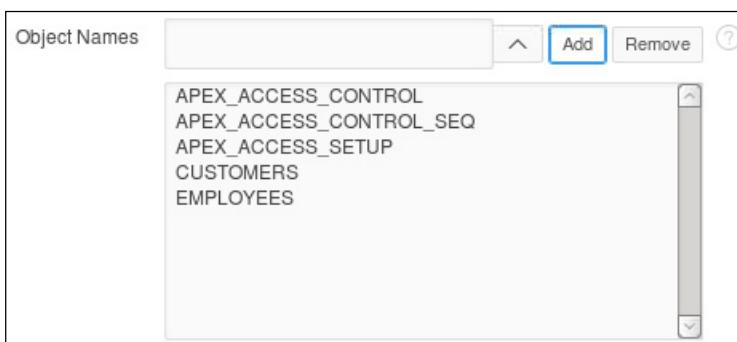


- c. Add the access control objects and customers and employees tables to the prerequisite section.

1. Click the **Prerequisites** link again.

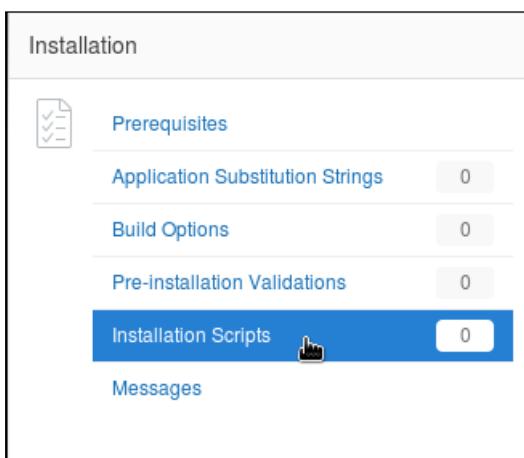


2. Under **Objects that will be Installed**, use the up arrow to locate the mentioned objects and click **Add**. Click **Apply Changes**.

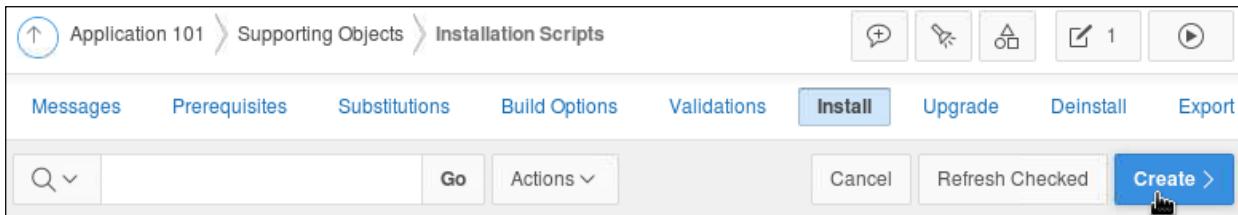


- d. Add installation scripts to create the customers and employees tables. (In a real world situation you would want to add all the tables. Here, for the sake of the labs, creating only two is ok)

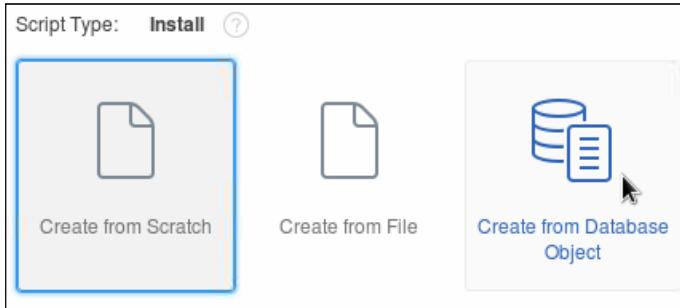
1. Click the **Installation Scripts** link.



2. Click **Create**.



3. Select **Create from Database Object**.



4. Enter **CreateTables** for Name, select **TABLE** for Object Type and click **Next**.

Script Type: **Install** [?](#)

Script Source Type: **From Database Object** [?](#)

* Name: **CreateTables** [?](#)

* Sequence: **10** [?](#)

Object Owner: **OEHR** [?](#)

* Object Type: **FUNCTION**
INDEX
PACKAGE
PROCEDURE
SEQUENCE
TABLE **TRIGGER**
VIEW [?](#)

Next >

5. Select the **Include DDL related to tables** checkbox. Select **CUSTOMERS** and **EMPLOYEES** tables and click **Next**.

Object Owner: **OEHR** [?](#)

Object Types: **TABLE** [?](#)

Table Options: **Include DDL related to tables** [?](#)

Filter: **Clear** [?](#)

* Object: APEX_ACCESS_CONTROL (TABLE)
APEX_ACCESS_SETUP (TABLE)
AUDITS (TABLE)
COUNTRIES (TABLE)
DEPARTMENTS (TABLE)
EMPLOYEES3 (TABLE)
INVENTORIES (TABLE)
JOBS (TABLE)
JOB_HISTORY (TABLE)
LOCATIONS (TABLE)
ORDERS (TABLE)
ORDER_ITEMS (TABLE)
PRODUCT_DESCRIPTIONS (TABLE)
PRODUCT_INFORMATION (TABLE)
PROMOTIONS (TABLE) [?](#)

Selected: CUSTOMERS (TABLE)
EMPLOYEES (TABLE) [?](#)

Next >

6. Review the DDL code and click **Create**.

The screenshot shows the Oracle Database Script Editor interface. At the top, the path is Application 101 > Supporting Objects > Installation Scripts > Script Editor. The script name is 'CreateTables'. A checkbox 'Include DDL related to tables' is checked. Below the table definitions, the DDL code for the 'EMPLOYEES' table is displayed:

```

CREATE TABLE "EMPLOYEES"
(
    "EMPLOYEE_ID" NUMBER(6,0),
    "FIRST_NAME" VARCHAR2(20),
    "LAST_NAME" VARCHAR2(25) CONSTRAINT "EMP_LAST_NAME_NN" NOT NULL ENABLE,
    "EMAIL" VARCHAR2(25) CONSTRAINT "EMP_EMAIL_NN" NOT NULL ENABLE,
    "PHONE NUMBER" VARCHAR2(20),
    "HIRE DATE" DATE CONSTRAINT "EMP_HIRE_DATE_NN" NOT NULL ENABLE,
    "JOB ID" VARCHAR2(10) CONSTRAINT "EMP_JOB_NN" NOT NULL ENABLE,
    "SALARY" NUMBER(8,2),
    "COMMISSION PCT" NUMBER(2,2),
    "MANAGER ID" NUMBER(6,0),
    "DEPARTMENT ID" NUMBER(4,0),
    "ACTIVE YN" VARCHAR2(1),
)

```

e. Add the installation scripts to create the access control objects.

1. Click **Create**.

The screenshot shows the Installation Scripts page. A green checkmark indicates 'Script saved'. The 'Create' button is highlighted with a cursor. The table 'CreateTables' has a status of '4 seconds ago' and was updated by 'apex'. The source is 'Database Object'.

Lock	Name	Sequence	Script	Status	Updated	Updated By	Source
	CreateTables	10	CREATE TABLE "EMPLOYEES" ("EMPLOYEE_ID" NUMBER(6,0), "FIRST_NAME" VARCHAR2(20), "LAST_NAME" VARCHAR2(25) CONSTRAINT "EMP_LAST_NAME_NN" NOT NULL ENABLE, "EMAIL" VARCHAR2(25) CONSTRAINT "EMP_EMAIL_NN" NOT NULL ENABLE, "PHONE NUMBER" VARCHAR2(20), "HIRE DATE" DATE CONSTRAINT "EMP_HIRE_DATE_NN" NOT NULL ENABLE, "JOB ID" VARCHAR2(10) CONSTRAINT "EMP_JOB_NN" NOT NULL ENABLE, "SALARY" NUMBER(8,2), "COMMISSION PCT" NUMBER(2,2), "MANAGER ID" NUMBER(6,0), "DEPARTMENT ID" NUMBER(4,0), "ACTIVE YN" VARCHAR2(1),)		4 seconds ago	apex	Database Object

2. Expand the **Tasks** link and click **Create Scripts for Access Control Tables**.

The screenshot shows the 'Tasks' section expanded. A blue button labeled 'Create Scripts for Access Control Tables' is visible.

3. Click the **Create Script**.

This generates installation scripts for the Access Control tables. There will be one script that creates the tables, if they do not already exist. A second script will contain insert statements to recreate the data, as it exists at this time, for the current application.

The deinstallation script is appended with commands to remove the Access Control data specific to the current application. Commands to drop the tables are not included because the tables could be used by other applications.

The following tables are included:

- APEX_ACCESS_SETUP
- APEX_ACCESS_CONTROL

Note: These scripts can be manually edited after generation.

Create Script

4. The scripts are created.

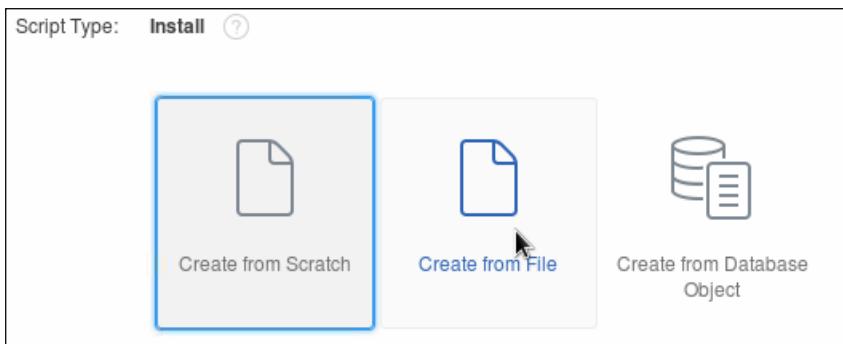
		acl_tab.sql	20	declare l_count number; begin select count(*) into l_count from user_objects where upper(object_name) = 'APEX_ACCESS_CONTROL_SEQ'; if l_count = 0 then execute immediate ('create sequence apex_a...')
		acl_data.sql	30	insert into APEX_ACCESS_SETUP (APPLICATION_MODE,APPLICATION_ID) values ('RESTRICTED',v('FB_FLOW_ID')); declare l_setup_id number; begin for c1 in (select id from apex_access_setup ...

- f. Add the `~/labs/files/customers.sql` file called **insert_cust** to the installation scripts. This will populate the customers table.

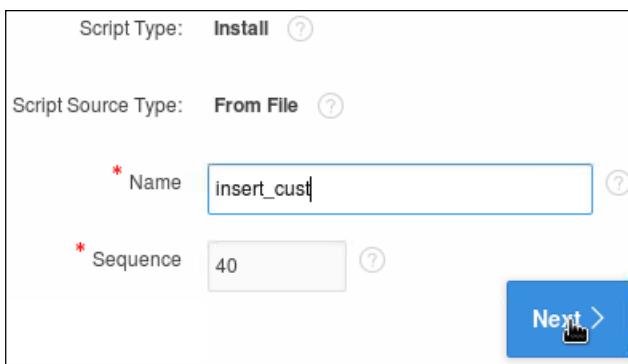
1. Click **Create**.

Actions									Cancel	Refresh Checked	Create >
Lock	Name	Sequence	Script			Status	Updated	Updated By	Source	<input type="checkbox"/>	
		CreateTables	10	CREATE TABLE "EMPLOYEES" ("EMPLOYEE_ID" NUMBER(6,0), "FIRST_NAME" VARCHAR2(20), "LAST_NAME" VARCHAR2(25) CONSTRAINT "EMP_LAST_NAME_NN" NOT NULL ENABLE, "EMAIL" VARCHAR2(25) CONSTRAINT ...		4 seconds ago	apex	Database Object	<input type="checkbox"/>		

2. Select **Create from File**.



3. Enter **insert_cust** for Name and click **Next**.

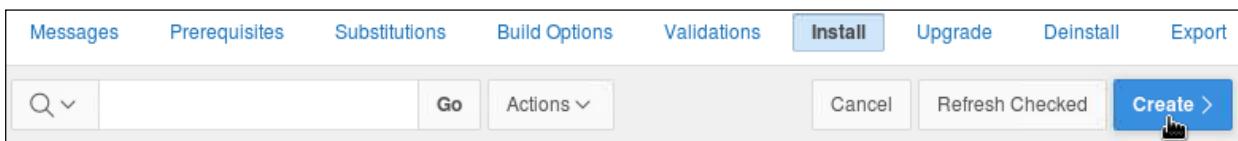


4. Click **Browse** and select the **customers.sql** file. Click **Create Script**.

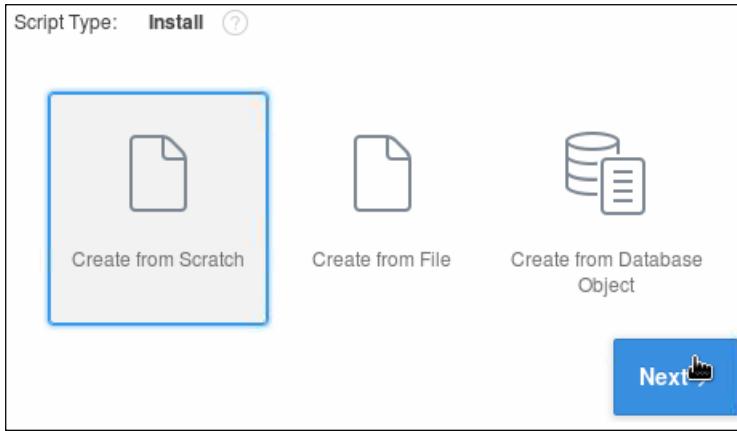


g. Create an installation script from scratch called **insert_emp** and use the text from the `~/labs/files/emp.txt` file to create the script.

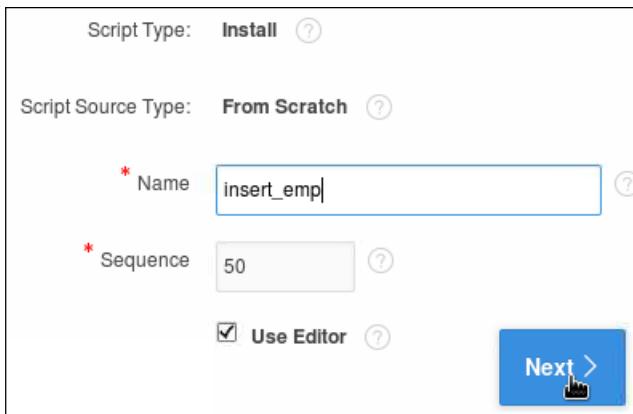
1. Click **Create**.



2. Ensure that **Create from Scratch** is selected and click **Next**.



3. Enter **insert_emp** for Name and click **Next**.



4. In a file browser, open the `~/labs/files/emp.txt` file. Copy the content and paste it into the script editor open in apex. Click **Create**.

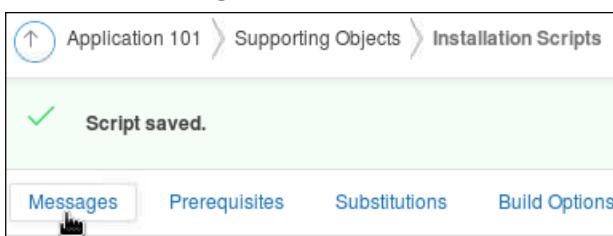
```

1 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
2 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
3 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
4 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
5 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
6 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
7 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
8 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
9 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
10 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
11 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
12 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
13 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
14 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)
15 Insert into OEHR.EMPLOYEES (EMPLOYEE_ID,FIRST_NAME, LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMM)

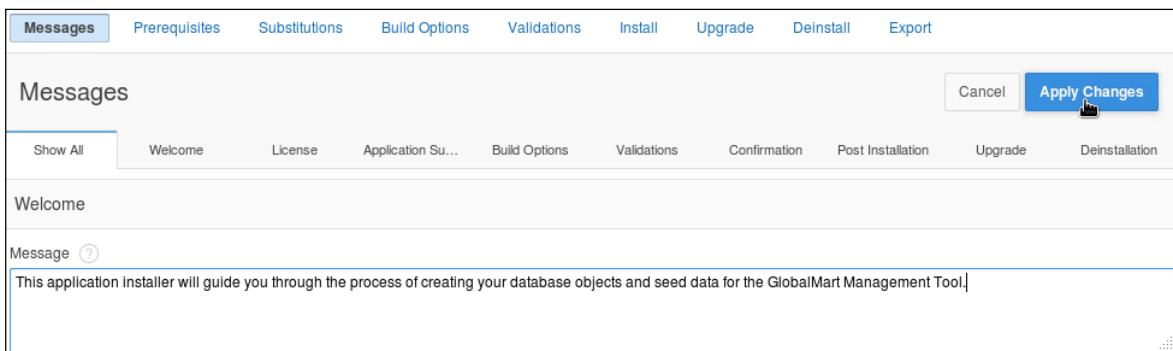
```

- h. Add a custom installation message.

- Click the **Messages** tab.

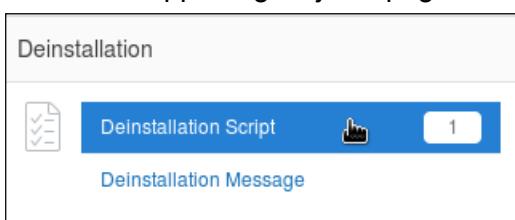


- Customize the message and click **Apply Changes**.

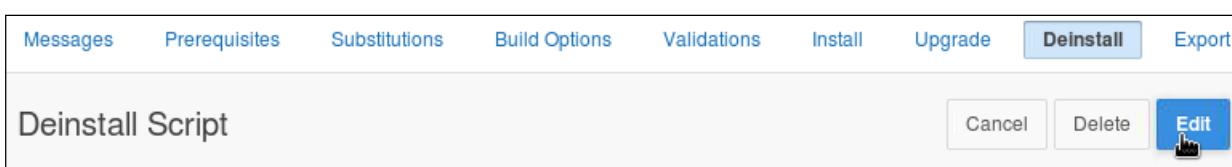


- i. Add de-installation script that will drop the customers and employees tables.

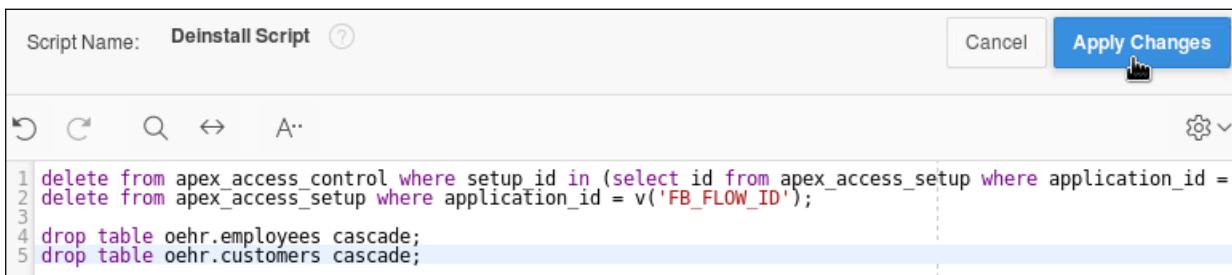
- From the Supporting Objects page, click the **Deinstallation Script**.



- Click **Edit**.



- Enter the drop table commands and click **Apply Changes**.



Practices for Lesson 13: Optimizing Your APEX Application

Chapter 13

Practices for Lesson 13: Overview

Practices Overview

In this practice, you review the activity of the GMT application, and monitor and improve its performance.

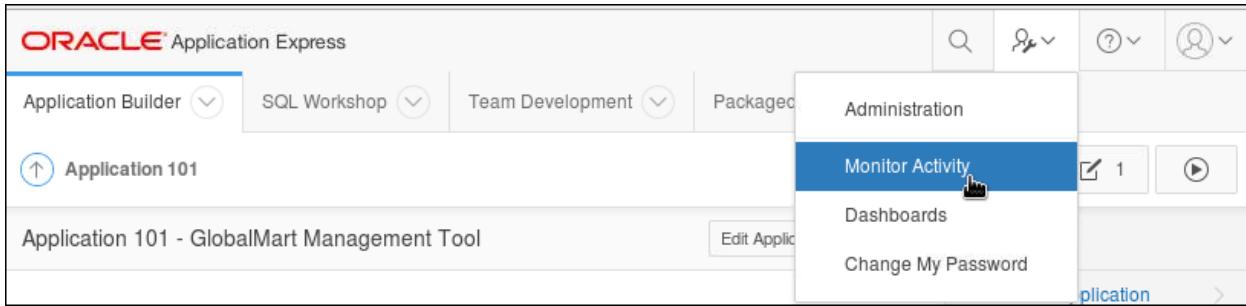
Practice 13-1: Monitoring Activity and Improving Performance of the GMT Application

Overview

In this practice, you review the activity of the GMT application, and monitor and improve its performance.

Steps

- Log in to your workspace and select Administration. Click **Monitor Activity** and review the Real Time Activity page.



ORACLE Application Express

Application Builder SQL Workshop Team Development

Packaged Apps

Monitor Activity

Real Time Activity Archived Activity

[View Dashboard](#)

Page Views	Developer Activity
By View	By Developer
By Application	By Developer Bar Chart
By Application and Page	By Day
By Day	By Application
By Hour	Application Changes, detailed
By Interactive Report	By Day, Monthly View

- b. How many page views were there yesterday?

Hint: Perform the following steps:

1. Click the **By Application and Page** link under Page Views.

The screenshot shows the Oracle Monitor Activity interface. At the top, there's a header with a circular icon containing an upward arrow and the text "Monitor Activity". Below the header, there are two tabs: "Real Time Activity" (which is selected and highlighted in blue) and "Archived Activity". A "View Dashboard" button is located below the tabs. The main content area is divided into two columns: "Page Views" and "Developer Activity". In the "Page Views" column, there are several links: "By View", "By Application", "By Application and Page" (which is highlighted with a blue box and has a cursor icon pointing at it), "By Day", "By Hour", and "By Interactive Report". In the "Developer Activity" column, there are also several links: "By Developer", "By Developer Bar Chart", "By Day", "By Application", "Application Changes, detailed", and "By Day, Monthly View".

2. Select Since 2 days.

The screenshot shows the Oracle Application Express Monitor Activity page. The 'Since' dropdown menu is open, showing various time intervals. The '2 days' option is selected and highlighted with a blue background. The main table displays page view statistics for different applications over the last two days. The columns include Application ID, Application Name, Distinct Users, Distinct Sessions, Page Events, Partial Page Views, Cached Regions, and Page Name.

Application ID	Application Name	Distinct Users	Distinct Sessions	Page Events	Partial Page Views	Cached Regions	Page Name
4000	Page Designer	2	9	174	0	0	Page Designer
4850	Resource Handler	1	4	119	0	0	Resource Handler
4000	Application Builder	2	13	69	6	0	Application Builder
4000	Application Builder	2	16	56	0	0	Application Builder
4000	Export Application	1	4	50	0	0	Export Application
4000	Edit REST Web Reference	1	3	46	0	0	Edit REST Web Reference
4500	zero	1	7	45	0	0	zero
4000	Global Page	1	4	38	0	0	Global Page
4000	Run Page	1	8	36	0	0	Run Page
4000	Web Service References	1	3	35	0	0	Web Service References
4850	Resource Template	1	4	33	0	0	Resource Template

- c. Which page was viewed the most?

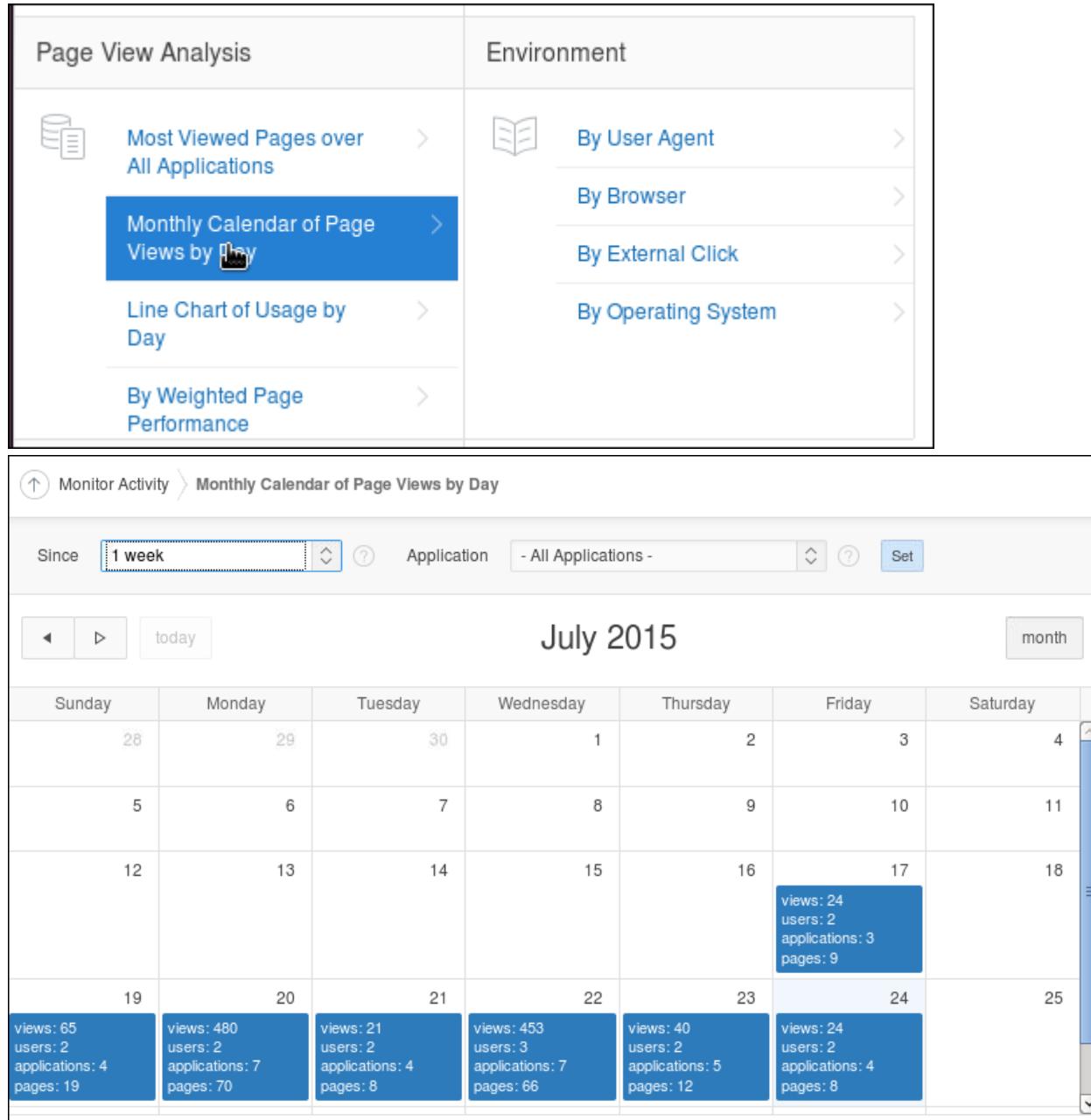
Hint: Click the Most Viewed Pages over All Applications link under Page View Analysis.

The screenshot shows the Oracle Analytics Cloud interface. The main navigation bar at the top has 'Analytics' selected. Below it, the 'Page View Analysis' section is highlighted. In this section, the 'Most Viewed Pages over All Applications' link is highlighted with a blue box and a cursor icon, indicating it is the target for the question. Other links in this section include 'Monthly Calendar of Page Views by Day'. To the right of this section are three other sections: 'Bar Chart of Active Sessions by Hour', 'Environment' (with links to 'By User Agent', 'By Browser', 'By External Click', and 'By Operating System'), and 'Login Attempts' (with links to 'Login Attempts', 'Login Attempts by Authentication Result', and 'Developer Login Summary').

- d. What were the total page views over the course of this class?

Hint: Perform the following steps:

1. Click **Monthly Calendar of Page Views by Day** under **Page View Analysis**.
2. Select **Since 1 week**.

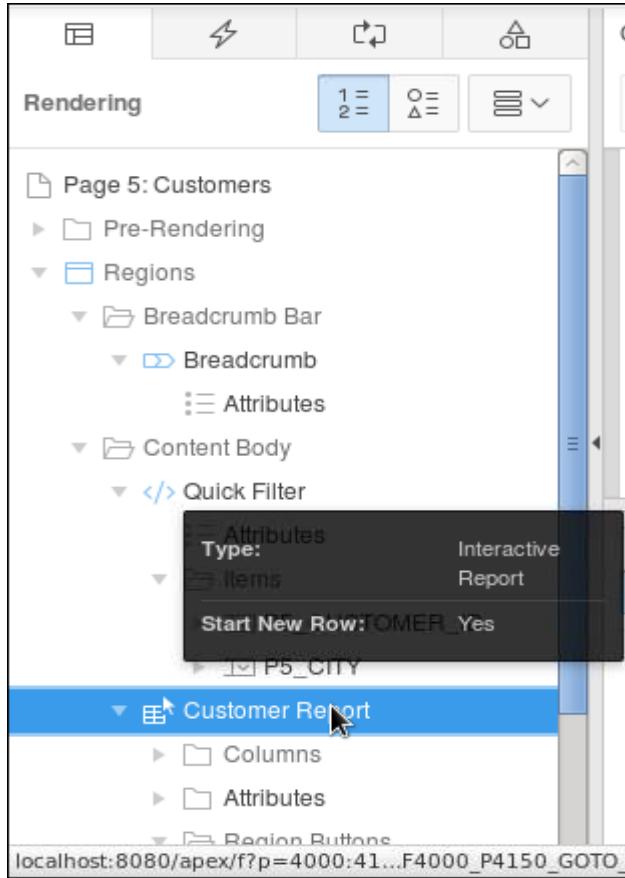


- e. Add "This page took #TIMING# CPU seconds" to the Region Footer Text on one of the report pages in an application.
1. Navigate to the GMT application.
 2. Navigate to the page definition of a report page.

The screenshot shows the Oracle Application Express (APEX) Page Catalog. At the top, there is a search bar, a 'Go' button, a grid icon, a list icon, an 'Actions' dropdown, and a 'Create Page >' button. Below the header, there are two rows of page definitions, each with an icon, a label, and a small description.

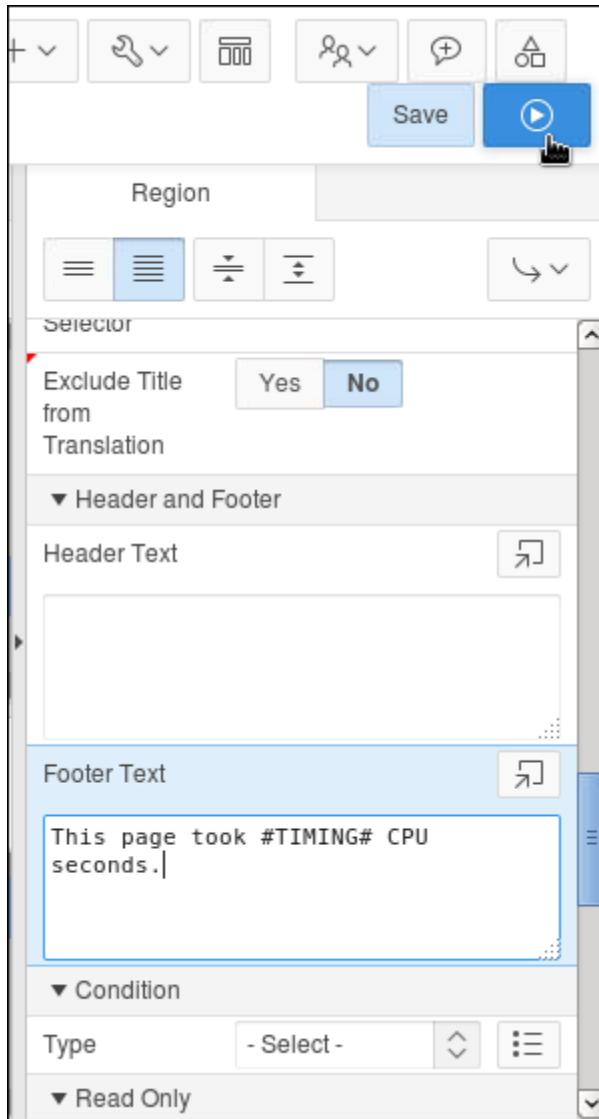
Page ID	Page Name	Description
0	0 - Global Page - Mobile	
1	1 - Home	
2	2 - Home	
3	3 - Top Tier Salary	
4	4 - Customer Address List	
5	5 - Customers	
6	6 - Employee Commission	
7	7 - Customer Details	
8	8 - List of Orders	

- Under Regions, click the page region name.



- Go to **Header and Footer** in the Property Editor section.
- Enter **This page took #TIMING# CPU seconds.** for **Footer Text**.

6. Run the page.



The screenshot shows the GMT | GlobalMart Management Tool interface. On the left is a sidebar with an orange header containing 'Home' and 'Customers' with dropdown menus, and sections for 'New Customer Information', 'Orders', 'Products', 'Help', 'Admin', and 'Data Load'. The main content area has a purple header 'Breadcrumb' with 'Home / Customers'. Below is a table with columns 'City', 'Name', 'Account mgr', and 'Cust Email'. The data includes:

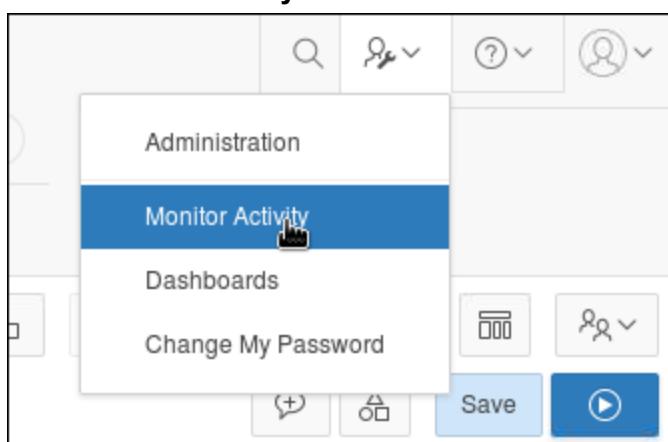
City	Name	Account mgr	Cust Email
Buffalo	Ellen Khan	Zlotkey	Ellen.Khan@VERDIN.COM
Rochester	Emmet Garcia	Zlotkey	Emmet.Garcia@VIREO.COM
Lockport	Fred Reynolds	Zlotkey	Fred.Reynolds@WATERTHRUSH.COM
Pittsburgh	Fred Lithgow	Zlotkey	Fred.Lithgow@WHIMBREL.COM
Philadelphia	George Adjani	Zlotkey	George.Adjani@WILLET.COM
Philadelphia	Irene Laughton	Zlotkey	Irene.Laughton@ANHINGA.COM

At the bottom of the page, a message is highlighted with a red box: 'This page took 0.22 CPU seconds.'

- f. How many login attempts were made since the beginning of the class and by what authentication method?

Hint: Perform the following steps:

1. Navigate to **Home > Administration**.
2. Click **Monitor Activity**.



3. Under Login Attempts, click **Login Attempts**.

- g. How will you view the Active Sessions for your applications?
Hint: Under Sessions, click the Active Sessions link.
- h. What additional tasks can you perform to improve the application's performance?
- i. Discuss your answers to the previous question with your neighbor.

Practices for Lesson 14: Globalization and Translation

Chapter 14

Practices for Lesson 14: Overview

Practices Overview

In these practices, you import the GMT application and translate it in to German language.

Practice 14-1: Translating the GMT Application

Overview

In this practice, you translate the GMT application developed in Application Builder by using Oracle Application Express 5.0

Assumptions

Tasks

- a. Map the primary language and target language of the GMT application.
- b. Seed and export text to a translation file.
- c. Translate the text and apply the translation file.
- d. Publish the translated application.

Practice 14-2: Specifying the Primary Language for the Translated GMT Application

Overview

In this practice, you specify the primary language for the translated GMT application.

Assumptions

You have completed Practice 14-1.

Tasks

- a. Specify the primary language for the translated GMT application.

Practice 14-3: Translating Messages Used in PL/SQL

Overview

If your application includes PL/SQL regions, PL/SQL processes, or calls PL/SQL package, procedures, or functions, you may need to translate generated HTML. First, you define each message on the Translatable Messages page. Second, you use the `APEX_LANG.MESSAGE` API to translate messages from PL/SQL stored procedures, triggers, or packaged procedures and functions. You create translatable messages on the Translate Messages page.

Assumptions

You have completed Practice 14-1 and 14-2.

Tasks

- a. Define each message on the Translatable Messages page. Use the `APEX_LANG.MESSAGE` API to translate messages from PL/SQL stored procedures, triggers, or packaged procedures and functions. Create translatable messages on the Translate Messages page.

Solution 14-1: Translating the GMT Application

Overview

In this practice, you translate the GMT application developed in Application Builder by using Oracle Application Express 5.0

Steps

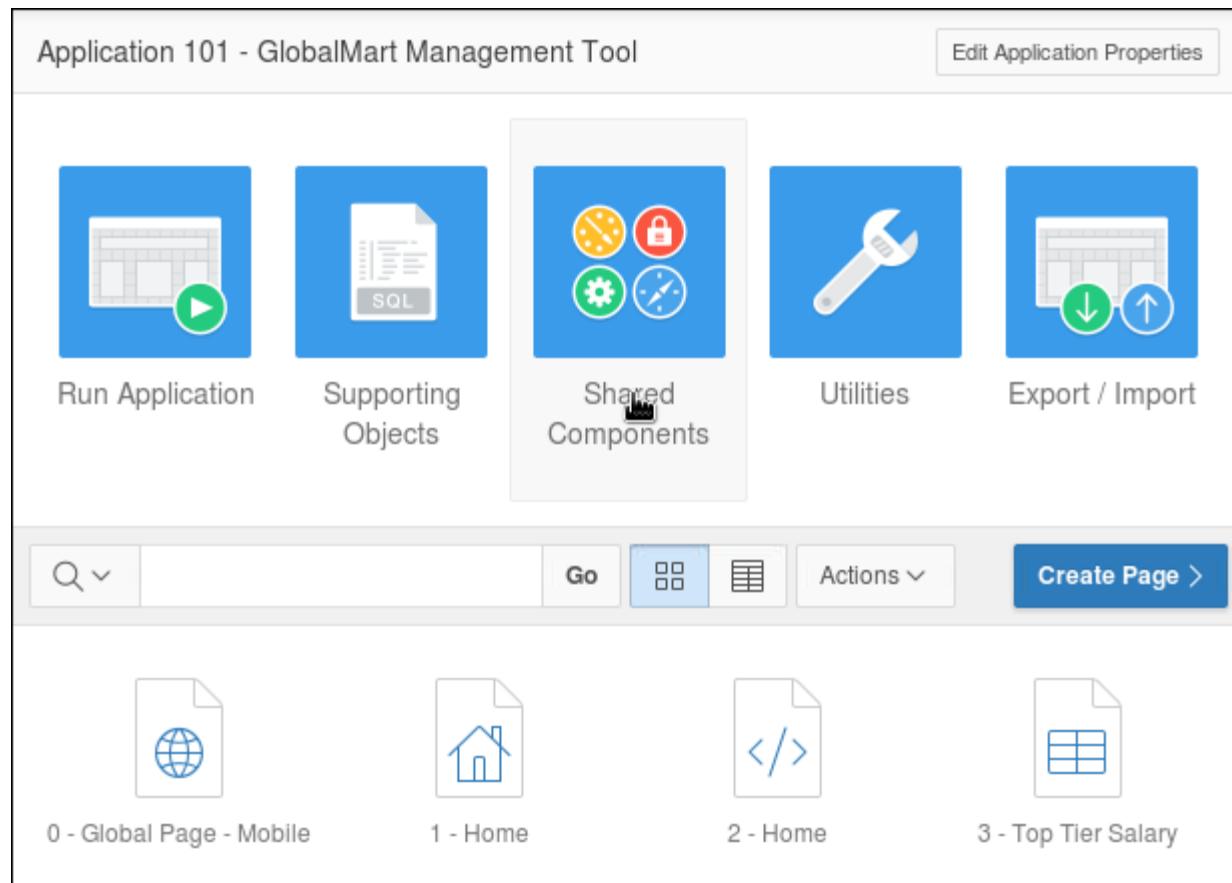
- Map the primary language and target language of the GMT application.

- On the GlobalMart application home page, click Application<n> on the Developer toolbar.

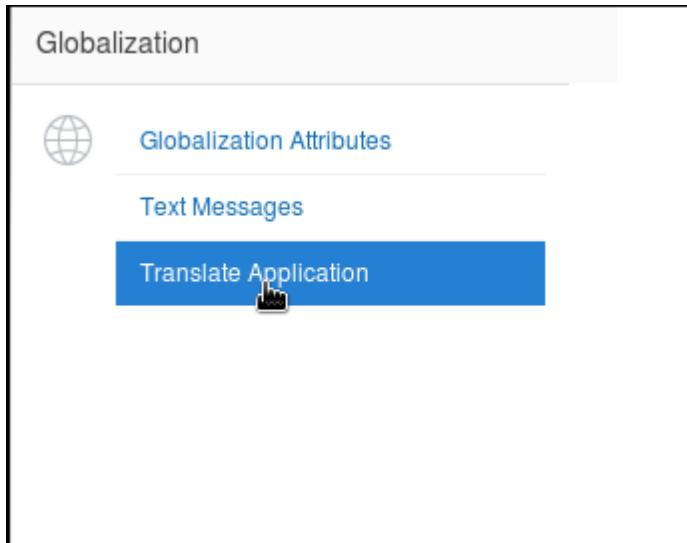
Note: Here <n> refers to the Application ID of your application.

The screenshot shows the GlobalMart Management Tool home page. The left sidebar contains a navigation menu with links: Home, Product List, Customers, Orders, Products, Help, Admin, and Data Load. The main content area features a summary of orders by mode, showing 32 Online and 73 Direct. Below this are sections for news and events, with a link to visit oracle.com. The bottom toolbar includes links for Home, Application 101, Edit Page 1, Session, View Debug, Debug, Show Grid, Quick Edit, Theme Roller, and a gear icon. The title bar reads "GMT | GlobalMart Management Tool".

2. Click **Shared Components**.



3. In the Globalization section, click **Translate Application**.



4. On the Translate Application page, click **Define application languages**.

The screenshot shows the Oracle Translate Application interface. At the top, the navigation path is "Application 101 > Shared Components > Translate". Below this, there are two main sections: "How to Translate" on the left and "Translations" on the right. The "How to Translate" section contains three items: "Define application languages" (with a speech bubble icon), "Seed translatable text" (with a download icon), and "Download XLIFF translation files" (with a download icon). The "Translations" section displays the message "No translations found." A URL "localhost:8080/apex/f?p=4000:4751:9293851071754::NO:::" is visible at the bottom of the page.

5. Click **Create >**.

The screenshot shows the "Language Mappings" page under "Shared Components > Translate". The top navigation bar includes "Application 101", "Go", "Actions", "Cancel", and a prominent blue "Create >" button with a hand cursor icon. The main content area is currently empty, indicated by a large circular placeholder icon.

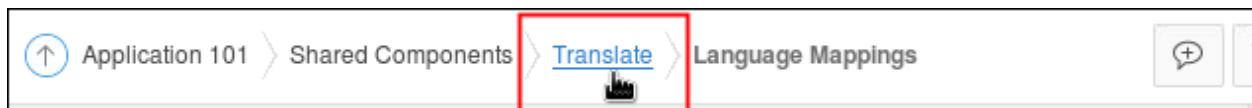
6. On the Application Language Mapping page, you have to enter a unique application ID for the translated application. Enter the unique application ID for Translation Application. Select **German (Germany) (de)** for Language. Click **Create**.

Create/Edit Application Language Mapping

To perform a translation, you create a unique application ID for the translated application. Use this page to map an existing application ID to a translation application ID. Note that when an application mapping is deleted, any corresponding translated applications are also deleted.

Primary Language Application	101 GlobalMart Management Tool	?
* Translation Application	60005	?
* Language	German (Germany) (de)	?
	Brazilian Portuguese, Chinese (China), Chinese (Taiwan), English, French, German, Italian, Japanese, Korean, Spanish	?
Image Directory	?	
Comments	?	
Application Language Mappings		
Cancel	Create	

7. Click **Translate**.

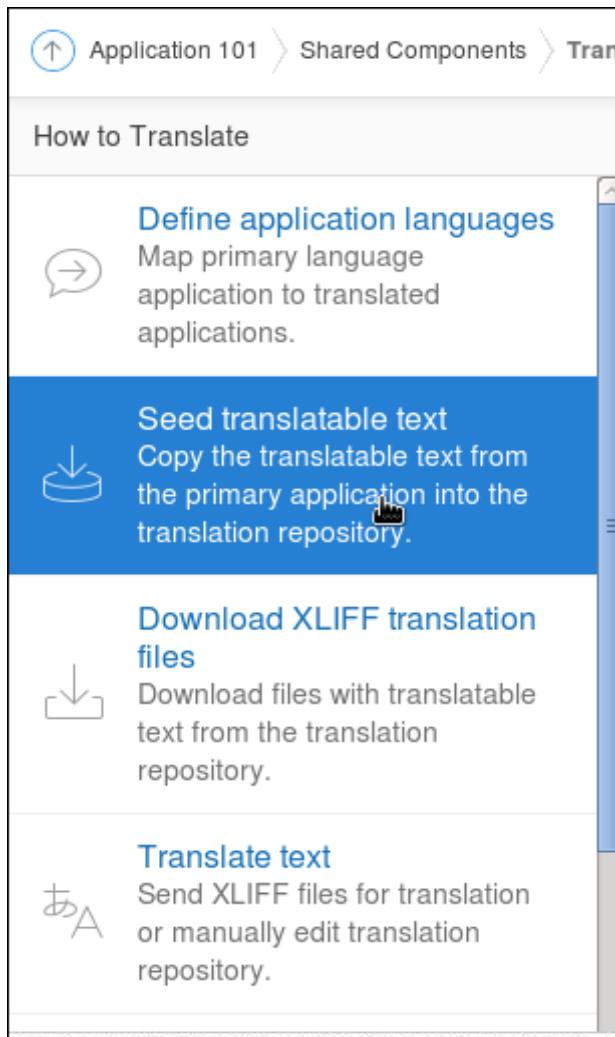


Action processed.

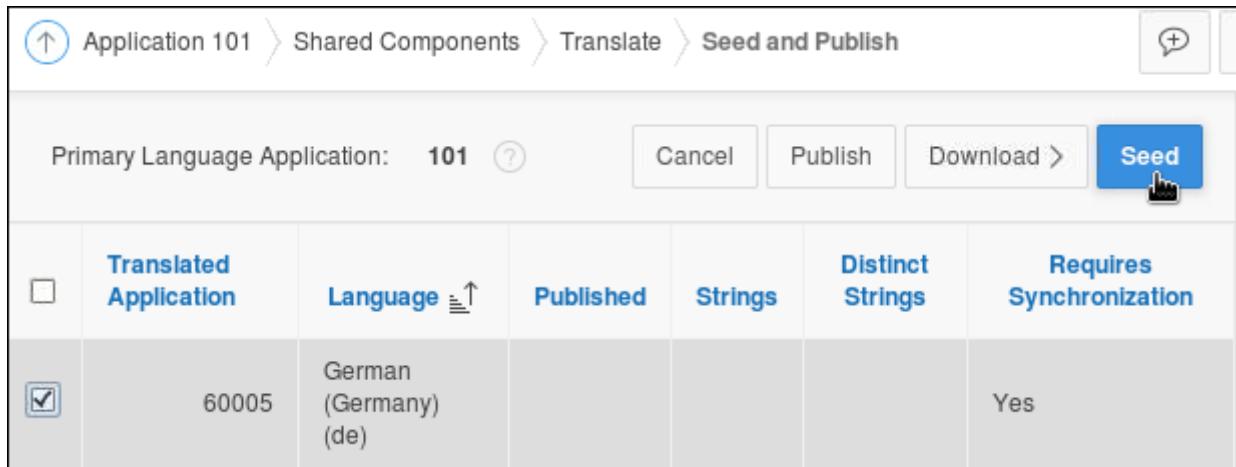
Translated Application	Language	Application Exists	Published	Translation Comment	Created	Updated
60005	German (Germany) (de)	No	-	-	Now	Now

b. Seed and export text to a translation file.

- On the Translate Application page, click **Seed translatable text** to translationrepository. Seeding the translation copies translatable text into the translation text repository. Once the translatable text is seeded, you can translate the text strings.



- Click the check box against the translated application and click **Seed**.



3. Click **Translate**.

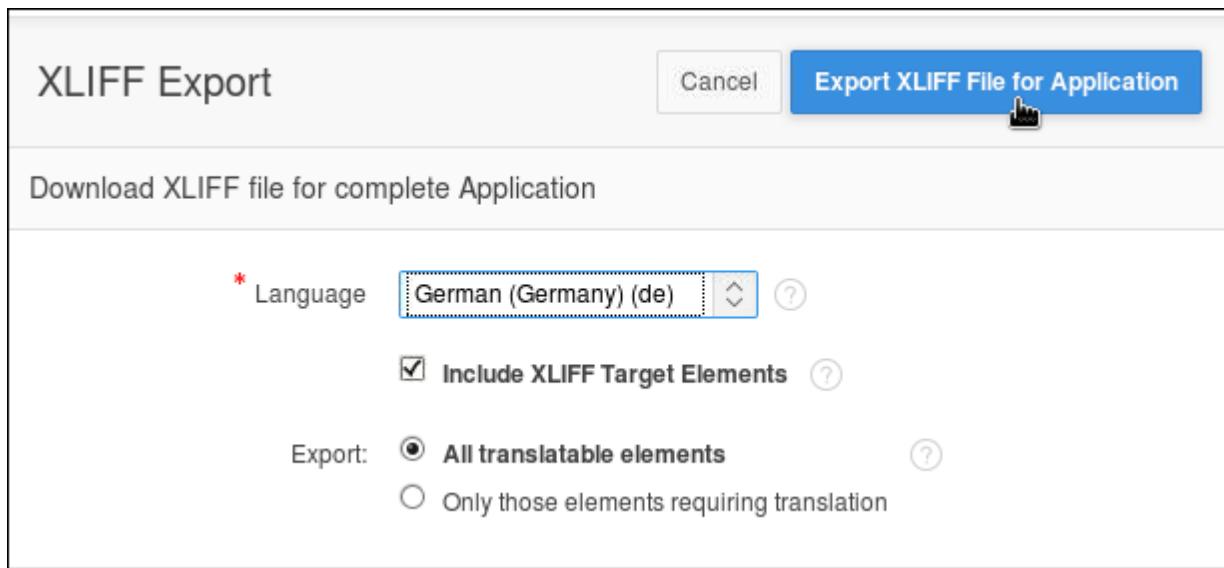
The screenshot shows the Oracle Translate application interface. At the top, there is a breadcrumb navigation: Application 101 > Shared Components > Translate > Seed and Publish. A green checkmark message says "Successfully seeded translatable text." Below this, there is a table with one row. The columns are: Translated Application (checkbox), Language (dropdown set to German (Germany) (de)), Published (checkbox), Strings (15,320), Distinct Strings (1,141), and Requires Synchronization (Yes). At the bottom right of the table is a blue "Seed" button.

<input type="checkbox"/> Translated Application	Language ↗	<input type="checkbox"/> Published	Strings	Distinct Strings	Requires Synchronization
<input type="checkbox"/> 60005	German (Germany) (de)	<input type="checkbox"/>	15,320	1,141	Yes

4. Next, on the Translate Application page, click **Download XLIFF translation files**.

The screenshot shows the Oracle Translate application interface. On the left, there is a sidebar with four items: "How to Translate", "Define application languages", "Seed translatable text", and "Download XLIFF translation files". The "Download XLIFF translation files" item is highlighted with a blue background and has a blue downward arrow icon. To its right, there is a detailed description: "Download files with translatable text from the translation repository." At the bottom of the sidebar, there is another item: "Translate text" with a gear and document icon, and a description: "Send XLIFF files for translation or manually edit translation repository."

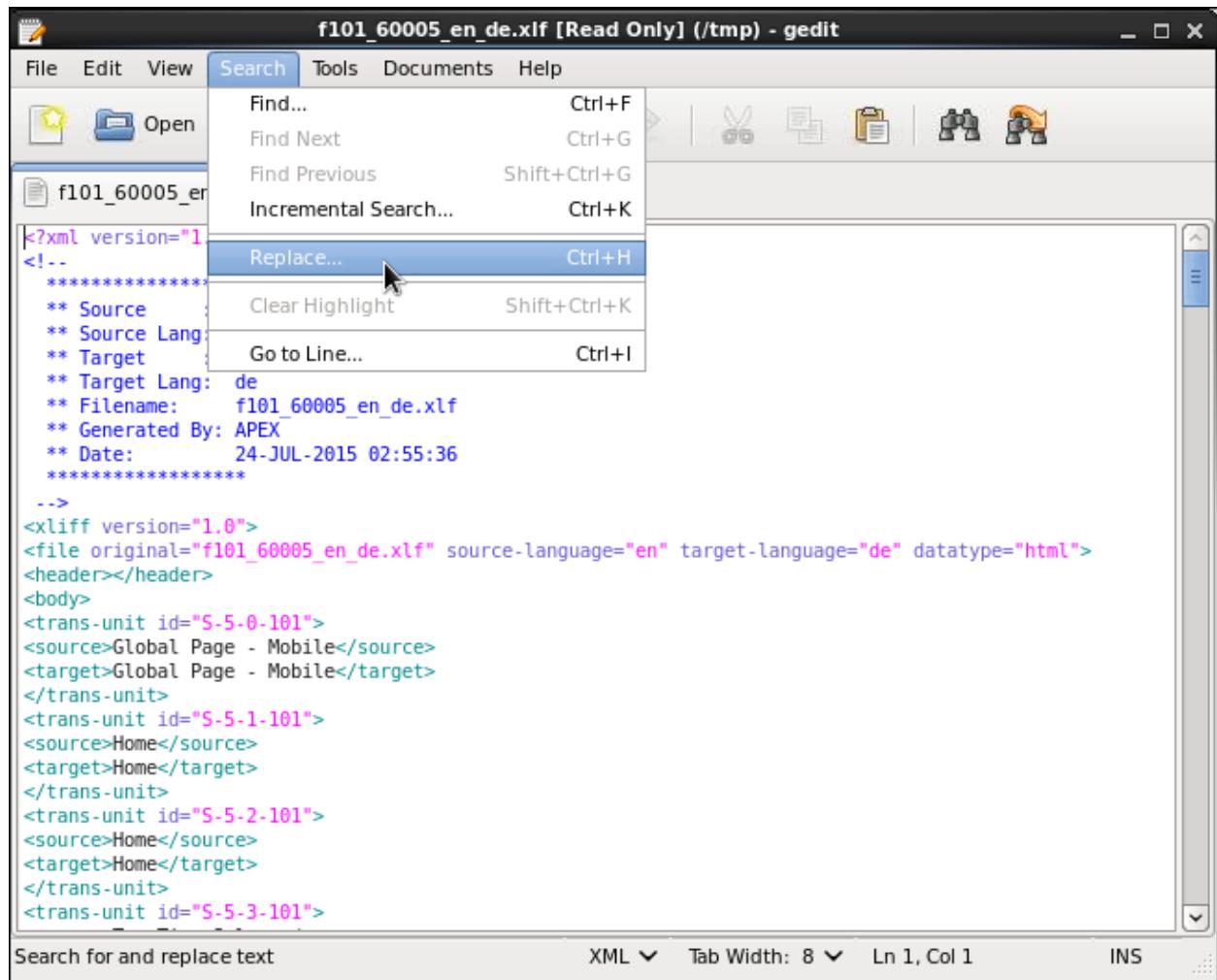
5. On the XLIFF Export page, under the Download XLIFF File for complete Application section, select Language as **German (Germany) (de)** and click **Export XLIFF File for Application**.

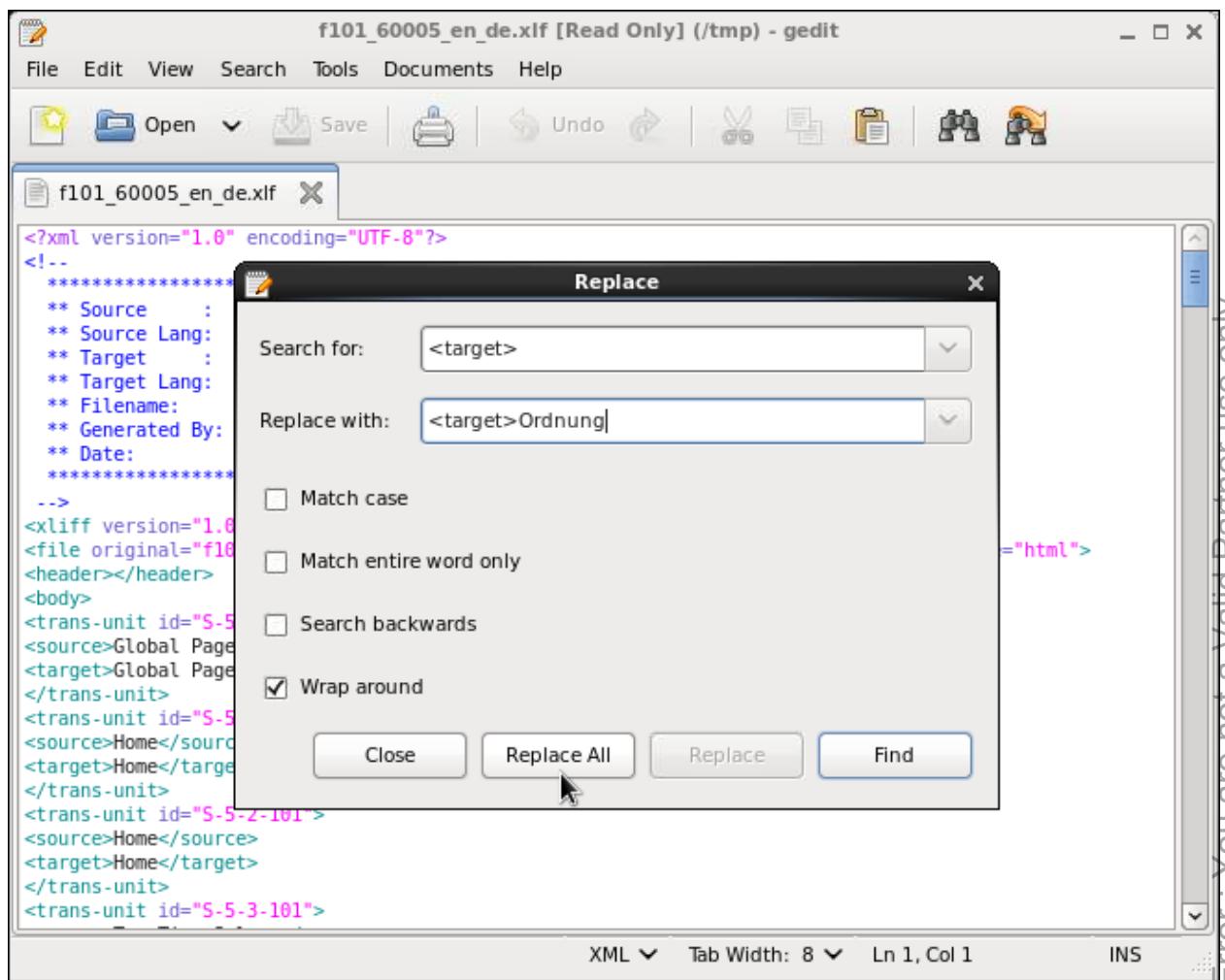


6. Open the XLIFF file that was downloaded.

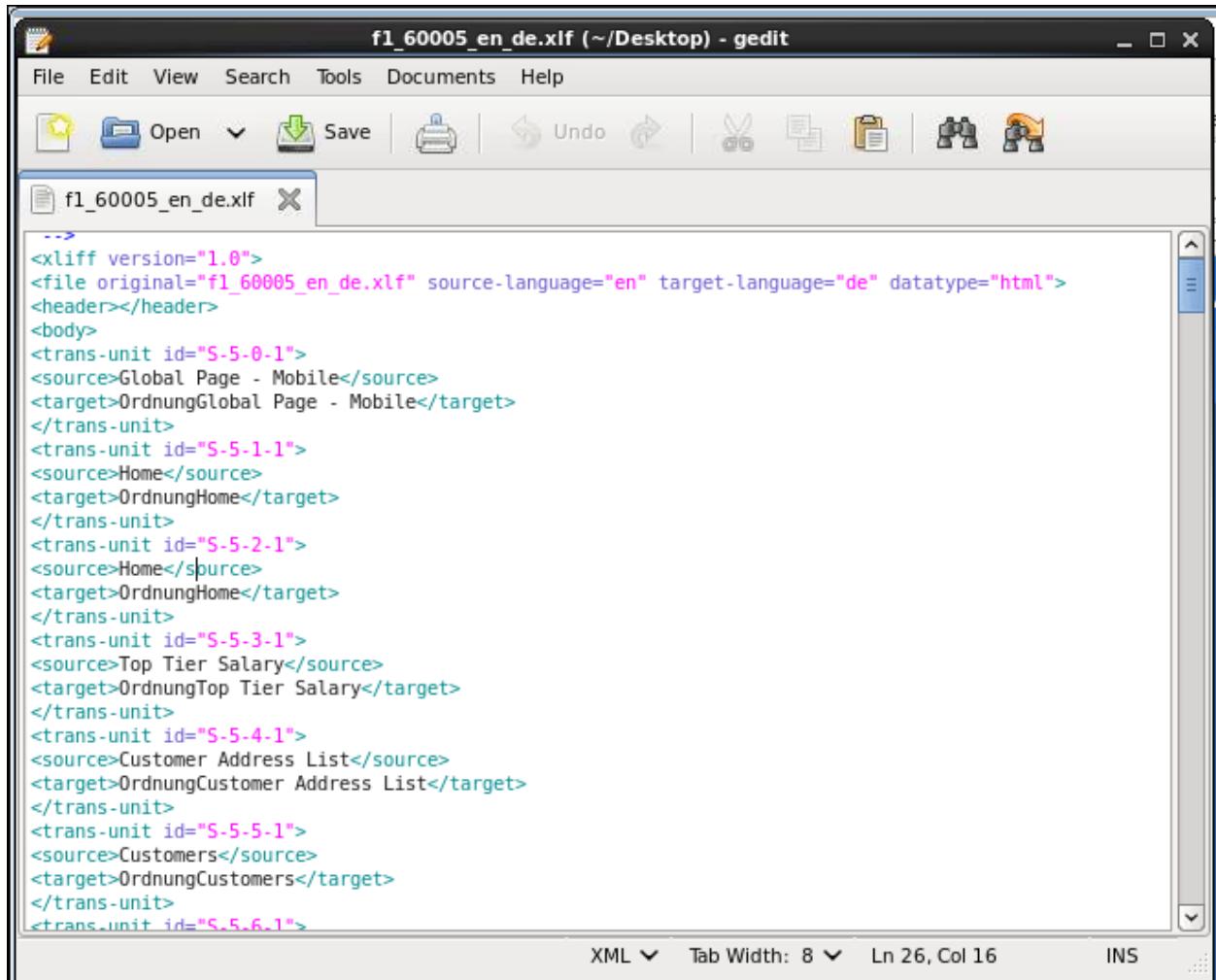
Note: You can navigate to the /home/oracle/Desktop folder and look for the file with the .xlf extension.

Open the file using a gedit tool and translate the elements by replacing the target with an equivalent German phrase. Click **Search** and select **Replace**. In this tutorial, you will replace all the <target> text strings with <target>Ordnung as prefix to the existing English phrases. Click **Replace All**.



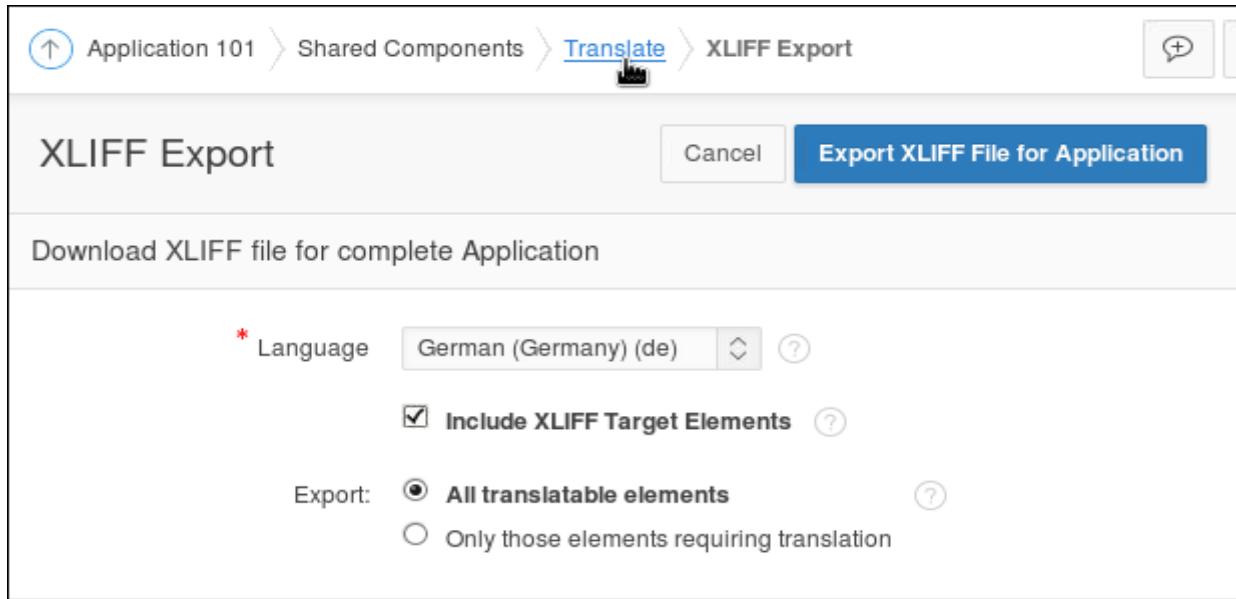


7. Save and close the file.



```
<!-->
<xliff version="1.0">
<file original="f1_60005_en_de.xlf" source-language="en" target-language="de" datatype="html">
<header></header>
<body>
<trans-unit id="S-5-0-1">
<source>Global Page - Mobile</source>
<target>OrdnungGlobal Page - Mobile</target>
</trans-unit>
<trans-unit id="S-5-1-1">
<source>Home</source>
<target>OrdnungHome</target>
</trans-unit>
<trans-unit id="S-5-2-1">
<source>Home</source>
<target>OrdnungHome</target>
</trans-unit>
<trans-unit id="S-5-3-1">
<source>Top Tier Salary</source>
<target>OrdnungTop Tier Salary</target>
</trans-unit>
<trans-unit id="S-5-4-1">
<source>Customer Address List</source>
<target>OrdnungCustomer Address List</target>
</trans-unit>
<trans-unit id="S-5-5-1">
<source>Customers</source>
<target>OrdnungCustomers</target>
</trans-unit>
<trans-unit id="S-5-6-1">
```

8. Click **Translate**.



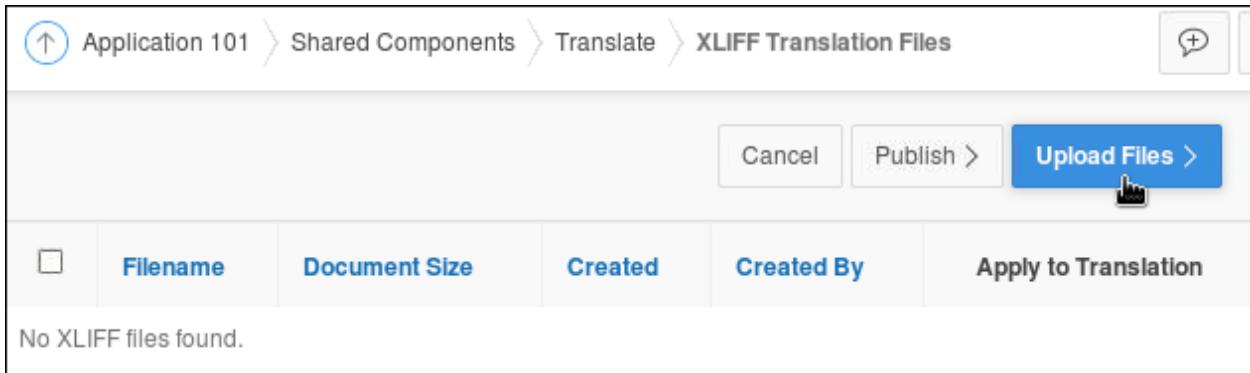
- c. Translate the text and apply the translation file.
- Once the XLIFF file is populated with the translations, the XLIFF file is uploaded back into the translation repository. To do this, on the Translate Application page, click **Apply XLIFF translation files** to translation repository.

The screenshot shows a web-based application interface for translating text. At the top, there's a navigation bar with icons for back, forward, and search, followed by the text "Application 101 > Shared Components > Trans". Below this is a section titled "How to Translate" with a sub-section "translation repository". There are four main steps listed:

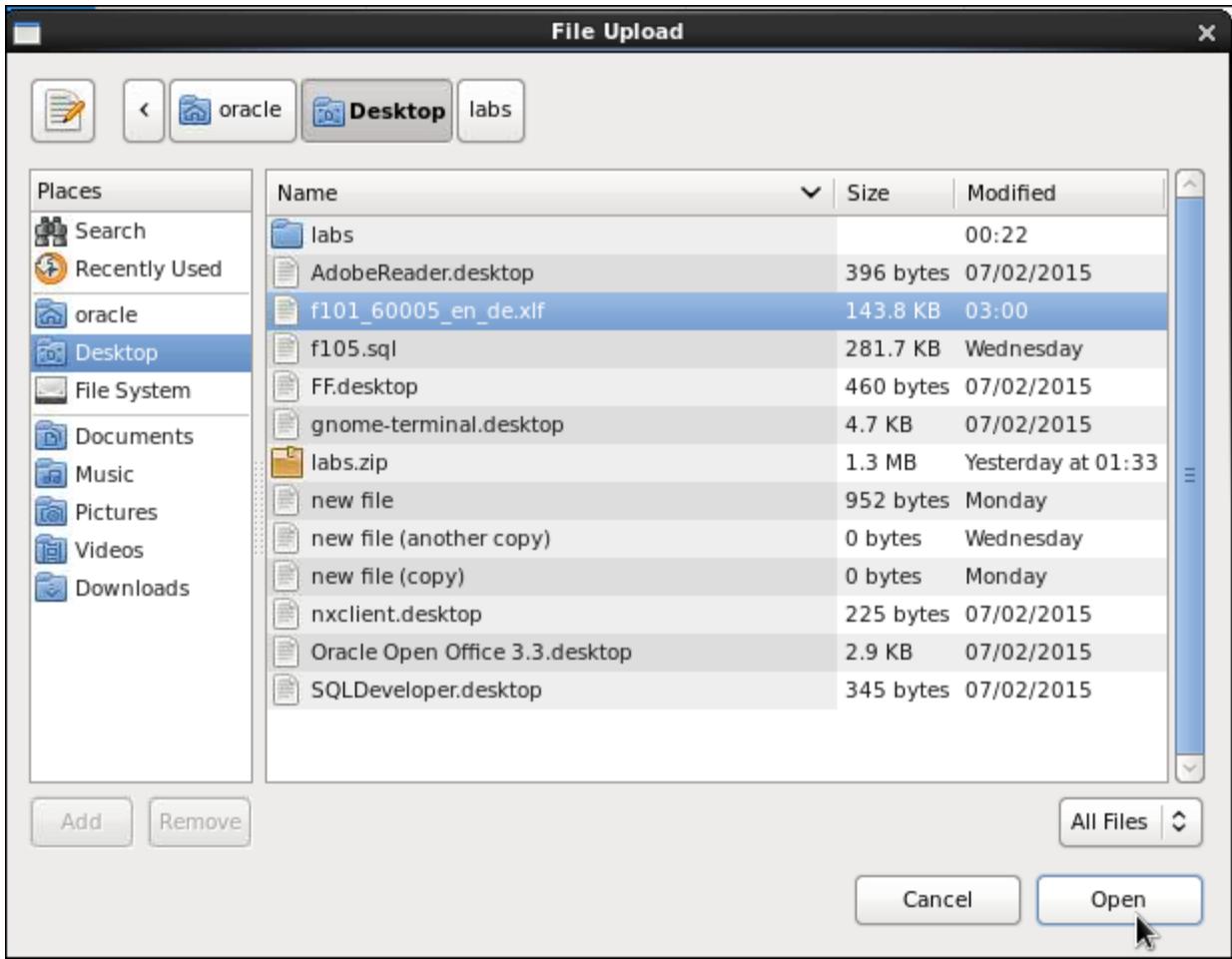
- Download XLIFF translation files**: An icon of a downward arrow. Description: Download files with translatable text from the translation repository.
- Translate text**: An icon of a document with a pencil. Description: Send XLIFF files for translation or manually edit translation repository.
- Apply XLIFF translation files**: An icon of an upward arrow. Description: Upload XLIFF files with translated text and apply translations to the translation repository. This step is highlighted with a blue background.
- Publish translated applications**: An icon of a document with an upward arrow. Description: Publish translated applications.

The URL at the bottom of the page is "localhost:8080/apex/f?p=4000:4758:9293851071754::N".

2. Click **Upload Files >**.



3. Click **Browse** and locate the translated file. Click Open to upload the translated XLIFF file into the XLIFF repository. Click **Upload**.



XLIFF Upload

Upload up to ten translated XLIFF files into the XLIFF repository. XLIFF files uploaded into the XLIFF file repository should **only** be translated files which were first generated via the XLIFF Export operation.

* 1	<input type="button" value="Browse..."/> f101_60005_en_de.xlf
2	<input type="button" value="Browse..."/> No file selected.
3	<input type="button" value="Browse..."/> No file selected.
4	<input type="button" value="Browse..."/> No file selected.
5	<input type="button" value="Browse..."/> No file selected.
6	<input type="button" value="Browse..."/> No file selected.
7	<input type="button" value="Browse..."/> No file selected.
8	<input type="button" value="Browse..."/> No file selected.
9	<input type="button" value="Browse..."/> No file selected.
10	<input type="button" value="Browse..."/> No file selected.

4. To apply the XLIFF files, check the row against the file name, select a translation mapping from the Apply to Translation column, and then click **Apply Checked**.

<input type="checkbox"/>	Filename	Document Size	Created	Created By	Apply to Translation
<input checked="" type="checkbox"/>	f101_60005_en_de.xlf	144KB	Now	APEX	101 >> 60005 (de)

5. Click **Translate**.

<input type="checkbox"/>	Filename	Document Size	Created	Created By	Apply to Translation
<input type="checkbox"/>	f101_60005_en_de.xlf	144KB	42 seconds ago	APEX	- Select -

- d. Publish the translated application.
1. The final step is to publish the translated application from the translation repository. Click **Publish translated applications**.

The screenshot shows a web-based application interface for translating an application. At the top, there's a breadcrumb navigation: Application 101 > Shared Components > Trans. Below it, a section titled "How to Translate" contains three main steps:

- Download XLIFF translation files**: An icon of a downward arrow. Description: Download files with translatable text from the translation repository.
- Translate text**: An icon of a plus sign and a document. Description: Send XLIFF files for translation or manually edit translation repository.
- Apply XLIFF translation files**: An icon of an upward arrow. Description: Upload XLIFF files with translated text and apply translations to the translation repository.

At the bottom of the page, there's a prominent blue button labeled "Publish translated applications" with a right-pointing arrow icon. Below this button, the URL "localhost:8080/apex/f?p=4000:385:9293851071754::NO" is visible.

2. Check against the row containing the translated application and click **Publish**.

The screenshot shows the Oracle Translate interface. At the top, the navigation path is Application 101 > Shared Components > Translate > Seed and Publish. Below this, the primary language application is set to 101. There are four buttons: Cancel, Seed, Download >, and Publish, with the Publish button being highlighted. A table lists a single translated application: 60005 (German (Germany) (de)). The table columns are Translated Application, Language, Published, Strings, Distinct Strings, and Requires Synchronization. The 'Published' column for the row shows an empty status, while the other columns show values: Strings: 15,320, Distinct Strings: 1,141, and Requires Synchronization: Yes.

	Translated Application	Language ↑	Published	Strings	Distinct Strings	Requires Synchronization
<input checked="" type="checkbox"/>	60005	German (Germany) (de)		15,320	1,141	Yes

The application will be successfully published.

The screenshot shows the Oracle Translate interface after publication. At the top, a green checkmark icon and the message "Successfully published application(s)." are displayed. Below this, the primary language application is still set to 101. The table now shows the application has been published, indicated by the timestamp "17 seconds ago" in the "Published" column. The other columns remain the same: Strings: 15,320, Distinct Strings: 1,141, and Requires Synchronization: No.

	Translated Application	Language ↑	Published	Strings	Distinct Strings	Requires Synchronization
<input type="checkbox"/>	60005	German (Germany) (de)	17 seconds ago	15,320	1,141	No

Solution 14-2: Specifying the Primary Language for the Translated GMT Application

Overview

In this practice, you specify the primary language for the translated GMT application.

Steps

- a. Specify the primary language.
 1. Click Shared Components.

The screenshot shows the Oracle Database Shared Components interface. The navigation path is Application 101 > Shared Components > Translate > Seed and Publish. A success message 'Successfully published application(s)' is displayed. The primary language application is set to 101. There are buttons for Cancel, Seed, Download >, and Publish. Below is a table listing applications:

	Translated Application	Language	Published	Strings	Distinct Strings	Requires Synchronization
<input type="checkbox"/>	60005	German (Germany) (de)	17 seconds ago	15,320	1,141	No

2. In the Globalization section, click **Globalization Attributes**.

The screenshot shows the Oracle Database Globalization section. The navigation path is Globalization > Globalization Attributes > Text Messages > Translate Application. The 'Globalization Attributes' button is highlighted with a cursor icon.

3. On the Globalization page, select Application Primary Language as **German (Germany) (de)** and Application Language Derived From as **Application Primary Language**. Click **Apply Changes**.

The screenshot shows the 'Edit Globalization Attributes' page for 'Application 101'. The 'Globalization' tab is selected. The 'Application Primary Language' is set to 'German (Germany) (de)'. The 'Application Language Derived From' is set to 'Application Primary Language'. Other fields like 'Application Date Format', 'Application Date Time Format', 'Application Timestamp Format', and 'Application Timestamp Time Zone Format' are also visible.

Application 101 > Edit Globalization Attributes

Definition Security Globalization User Interface

Application 101

Globalization

Cancel Apply Changes

Application Primary Language: German (Germany) (de)

Application Language Derived From: Application Primary Language

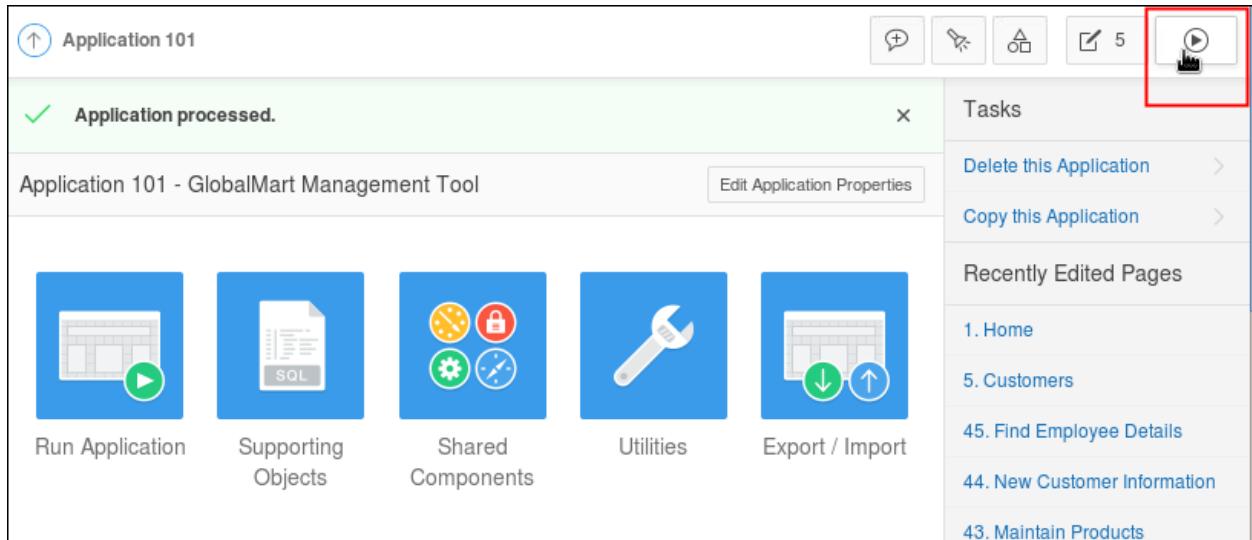
Application Date Format: DD-MON-YYYY

Application Date Time Format:

Application Timestamp Format:

Application Timestamp Time Zone Format:

4. Run the application.



5. You will notice that the application is in German and is based on the XLIFF file that was uploaded to the translation repository. Enter your login credentials for Username and Password on the Login page and click **Ordnung Login**.

The screenshot shows the "OrdnungLog In" login page. It has two input fields: "OrdnungUsername" containing "apex" and "OrdnungPasswort" containing "****". Below the fields is an orange "OrdnungLog In" button with a cursor icon pointing to it.

6. Notice that the application has been translated to German based on the XLIFF file that was uploaded.

The screenshot shows the GlobalMart Management Tool (GMT) interface. The top navigation bar includes links for OrdnungHelp, OrdnungSite Map, OrdnungFeedback, and a user icon. The left sidebar, titled "OrdnungHome", contains a dropdown menu with "OrdnungProduct List" and several other options: OrdnungCustomers, OrdnungOrders, OrdnungProducts, OrdnungHelp, OrdnungAdmin, and OrdnungData Load. The main content area is titled "OrdnungHome". It features a navigation bar with "OrdnungManage Customers", "OrdnungManage Products", and "OrdnungManage Orders". Below this is a summary section titled "OrdnungOrders By Mode Summary" showing two counts: 32 for "Online" and 73 for "Direct". Further down, there's a section titled "OrdnungIn the News" with "Ordnung", "News and Events", and a link to "Visit us at [www.oracle.com](#)".

Solution 14-3: Translating Messages Used in PL/SQL

Overview

If your application includes PL/SQL regions, PL/SQL processes, or calls PL/SQL package, procedures, or functions, you may need to translate generated HTML. First, you define each message on the Translatable Messages page. Second, you use the `APEX_LANG.MESSAGE` API to translate messages from PL/SQL stored procedures, triggers, or packaged procedures and functions. You create translatable messages on the Translate Messages page. To translate messages used in PL/SQL:

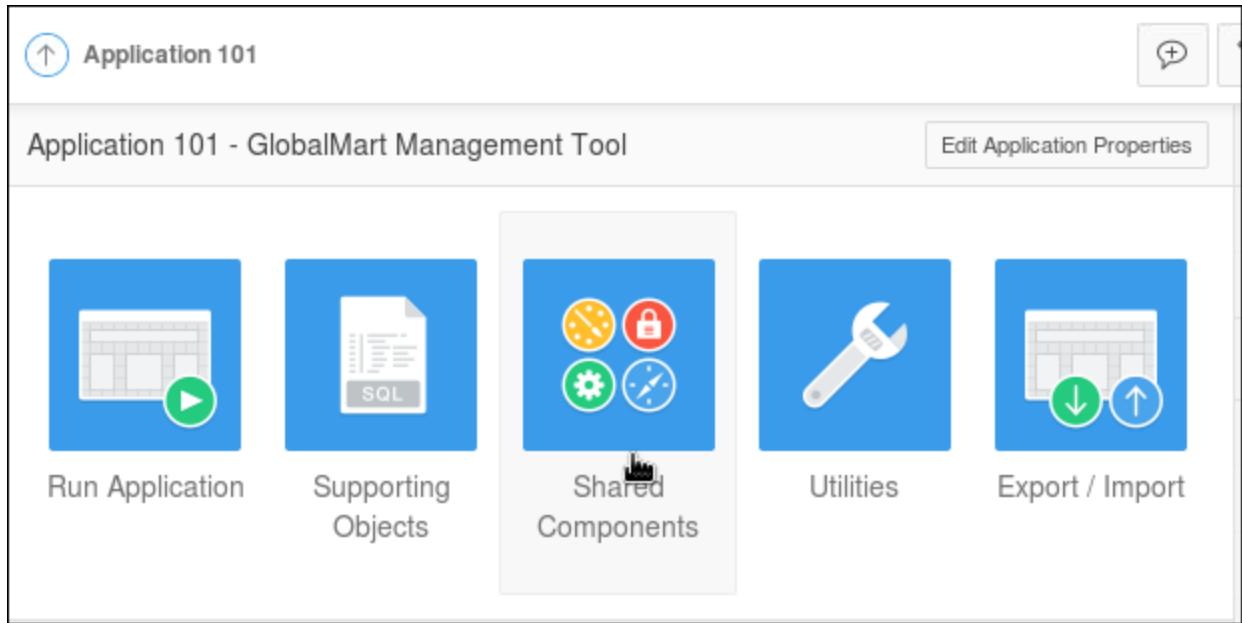
Steps

- Define each message on the Translatable Messages page. Use the `APEX_LANG.MESSAGE` API to translate messages from PL/SQL stored procedures, triggers, or packaged procedures and functions. Create translatable messages on the Translate Messages page.
- Click **Application<n>** on the Developer toolbar of the application.

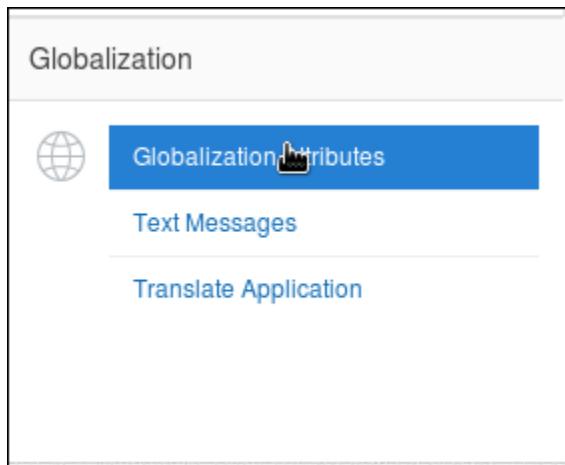
Note: Here, <n> refers to the Application ID of your application.

The screenshot shows the GlobalMart Management Tool application. The top navigation bar has tabs: OrdnungHelp, OrdnungSite Map, and OrdnungFeedback. The left sidebar, titled "GMT | GlobalMart Management Tool", contains a menu with the following items: OrdnungHome (selected), OrdnungProduct List, OrdnungCustomers, OrdnungOrders, OrdnungProducts, OrdnungHelp, OrdnungAdmin, and OrdnungData Load. The main content area displays the "OrdnungHome" page. At the top of this page are three buttons: "OrdnungManage Customers", "OrdnungManage Products", and "OrdnungManage Orders". Below these buttons is a summary section titled "OrdnungOrders By Mode Summary" showing two counts: 32 for "Online" and 73 for "Direct". Further down, there is a section titled "OrdnungIn the News" with links to "Ordnung" and "News and Events", and a note to "Visit us at www.oracle.com". The bottom of the screen shows the Oracle Application Express developer toolbar with various buttons like Home, Application 101, Edit Page 1, Session, View Debug, Debug, Show Grid, Quick Edit, Theme Roller, and a gear icon for settings.

2. Click **Shared Components**.



3. Under the Globalization section, click **Globalization Attributes**.



4. On the Globalization page, select Application Primary Language as **English (en)** and Application Language Derived From as **No NLS (Application not Translated)**, and click **Apply Changes**.

The screenshot shows the 'Edit Globalization Attributes' page for 'Application 101'. The 'Globalization' tab is selected. The 'Application Primary Language' is set to 'English (en)'. The 'Application Language Derived From' dropdown is set to 'No NLS (Application not translated)', which is highlighted with a blue border. Other fields include 'Application Date Format' (DD-MON-YYYY), 'Application Date Time Format' (empty), 'Application Timestamp Format' (empty), and 'Application Timestamp Time Zone Format' (empty). The 'Apply Changes' button is visible at the top right.

Application 101 > Edit Globalization Attributes

Definition Security Globalization User Interface

Application 101

Globalization

Cancel Apply Changes

Application Primary Language English (en)

Application Language Derived From No NLS (Application not translated)

Application Date Format DD-MON-YYYY

Application Date Time Format

Application Timestamp Format

Application Timestamp Time Zone Format

5. You will define a message called **GREETING_MESSAGE** in your application in English as "**Good morning %0**" and in German as "**Guten Tag %0**", and invoke this message from PL/SQL in the Home page by creating a region called Translation Region. Click **Home page**.

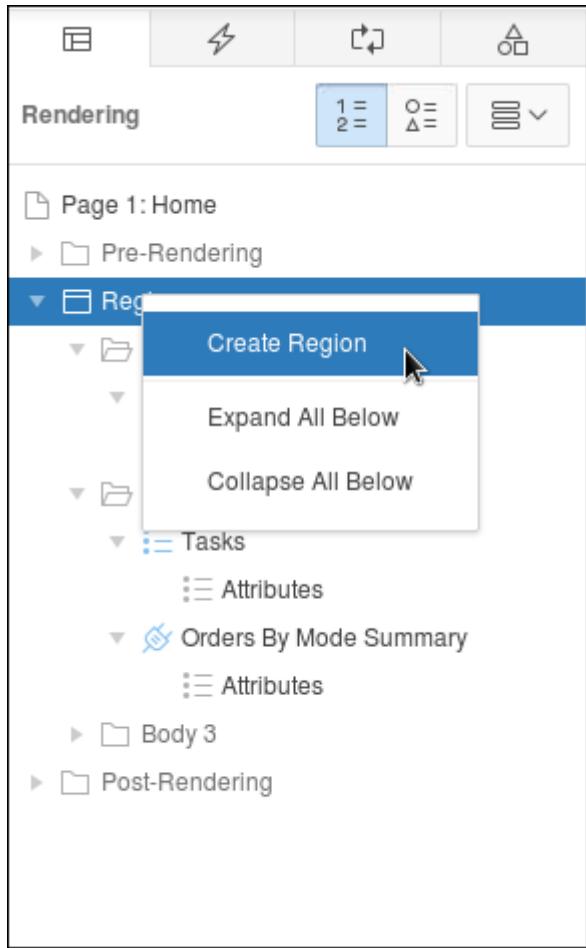
The screenshot shows the Oracle Application Express (APEX) interface for the "Application 101 - GlobalMart Management Tool". At the top, there is a success message: "Application processed." Below the title bar, there are five main navigation buttons:

- Run Application
- Supporting Objects
- Shared Components
- Utilities
- Export / Import

Below these buttons is a toolbar with search, go, actions, and create page buttons. Underneath the toolbar, four page thumbnails are displayed:

- 0 - Global Page - Mobile
- 1 - Home
- 2 - Home
- 3 - Top Tier Salary

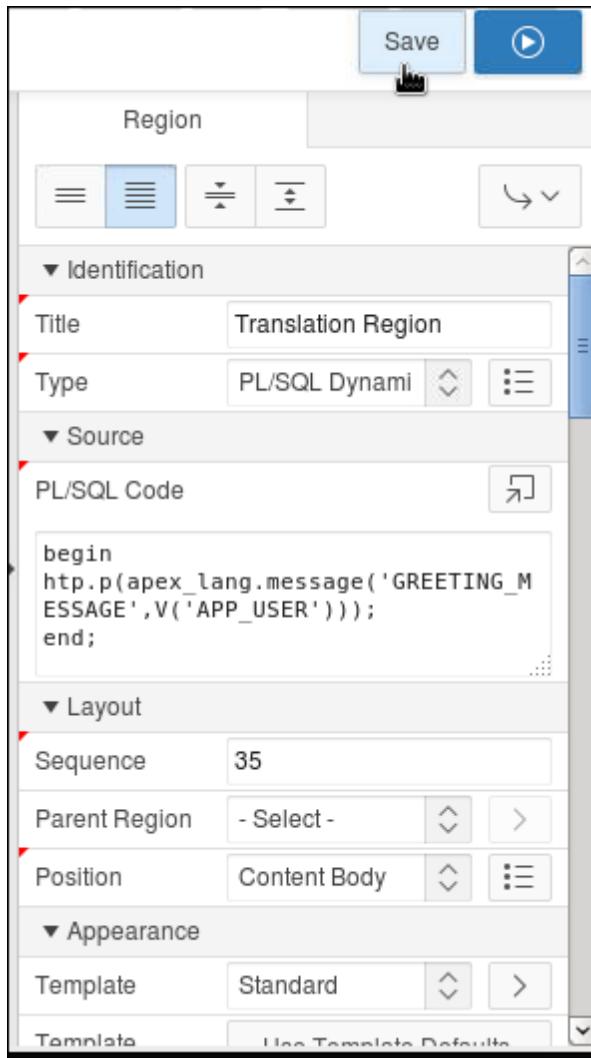
6. Right-click **Regions** and select **Create Region**.



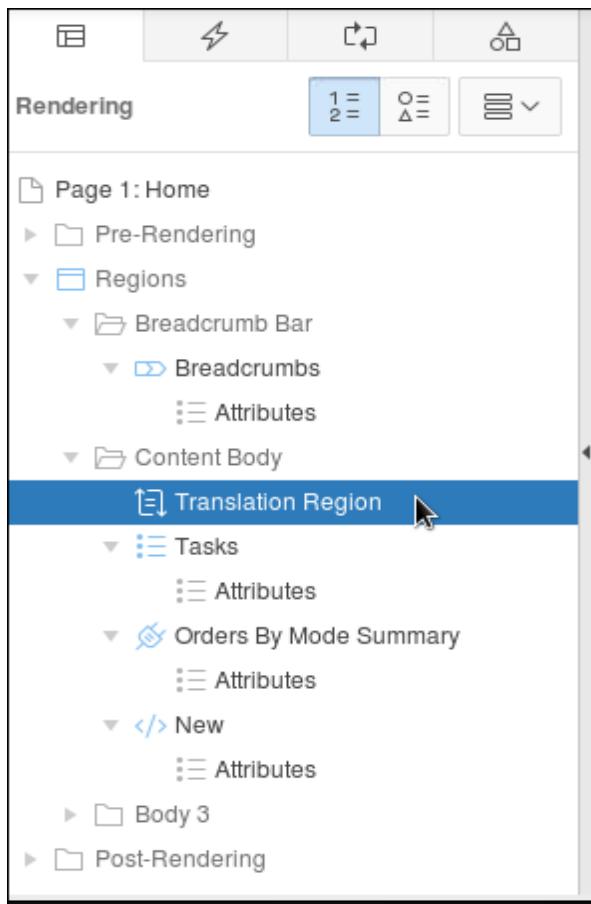
7. Enter **Translation Region** for Title.
8. Select **PL/SQL Dynamic Content** for the Type of region.

9. Enter the following PL/SQL. Click **Save**.

```
begin
http.p(apex_lang.message('GREETING_MESSAGE', V('APP_USER')));
end;
```



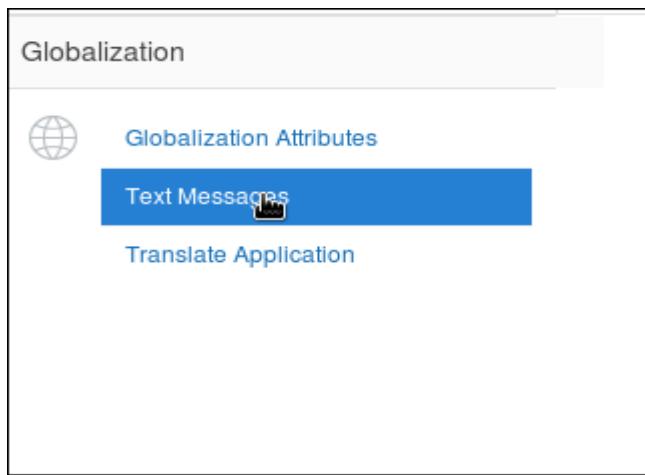
10. In the page definition of the Home page, move the Translation Region above the Tasks region.



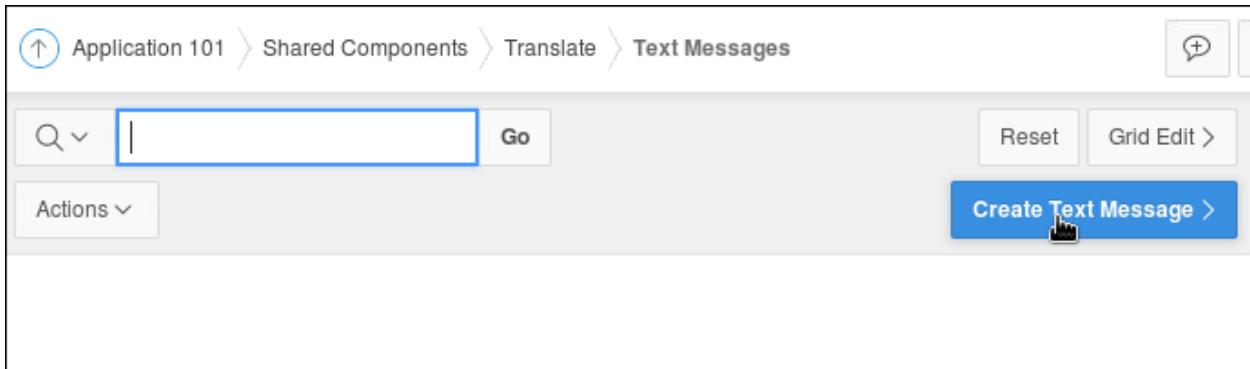
11. Click **Shared Components**.



12. Under Globalization, select **Text Messages**.



13. Click **Create Text Message**.



14. Enter **GREETING_MESSAGE** for Name and select **English (en)** for Language. Enter **Good Morning %0** for Text. Click **Create and Create Another**.

A screenshot of the 'Create/Edit Text Message' dialog box. The title is 'Create/Edit Text Message'. The application is set to '101 GlobalMart Management Tool'. The 'Name' field is marked with a red asterisk and contains 'GREETING_MESSAGE'. The 'Language' field is set to 'English (en)'. The 'Used in JavaScript' field is set to 'No'. The 'Text' field is marked with a red asterisk and contains 'Good Morning %0'. At the bottom, there are 'Cancel', 'Create And Create Another', and 'Create Text Message' buttons. The 'Create Text Message' button is highlighted with a cursor icon pointing at it.

15. Enter **GREETING_MESSAGE** for Name and select **German Germany (de)** for Language. Enter **Guten Tag %0** for Text. Click **Create Text Message**.

Application: 101 GlobalMart Management Tool

* Name: GREETING_MESSAGE

Language: German (Germany) (de)

Used in JavaScript: No

* Text (Example: Tax: %0 Total amount %1): Guten Tag %0

Create Text Message

16. Run the page.

Application 101 > Shared Components > Translate > Text Messages

Action processed.

Name	Language	Text	Used in JavaScript	Updated By	Date
GREETING_MESSAGE	de	Guten Tag %0	No	APEX	1 seconds ago
GREETING_MESSAGE	en	Good Morning %0	No	APEX	53 seconds ago

Create Text Message

Text Messages

Text Messages can be used to build translatable text strings with substitution variables that can be called from PL/SQL packages, procedures, and functions.

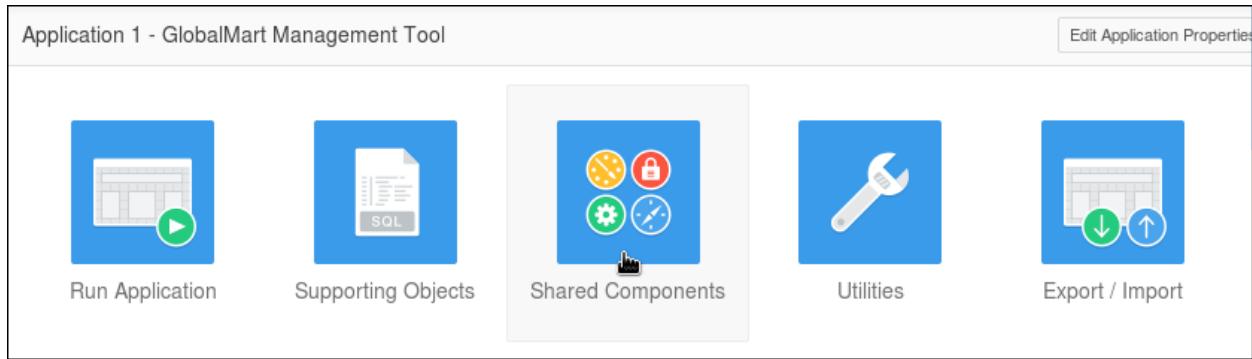
17. The Translation Region with the English greeting message is displayed.

The screenshot shows the GlobalMart Management Tool Home page. On the left, there is a navigation sidebar with orange buttons for Home, Product List, Customers, Orders, Products, Help, Admin, and Data Load. The main content area has a purple header "Translation Region". Below it, a message says "Good Morning APEX". There are three buttons: "Manage Customers", "Manage Products", and "Manage Orders". Underneath is a section titled "Orders By Mode Summary" with two circular icons: one labeled "32 Online" and another labeled "73 Direct". Further down is a section titled "In the News" and "News and Events". At the bottom, there is a toolbar with various icons and labels: Home, Application 101 (with a cursor icon), Edit Page 1, Session, View Debug, Debug, Show Grid, Quick Edit, Theme Roller, and a gear icon. The right side of the image has a vertical watermark: "Oracle University and Error : You are not a Valid Partner use only".

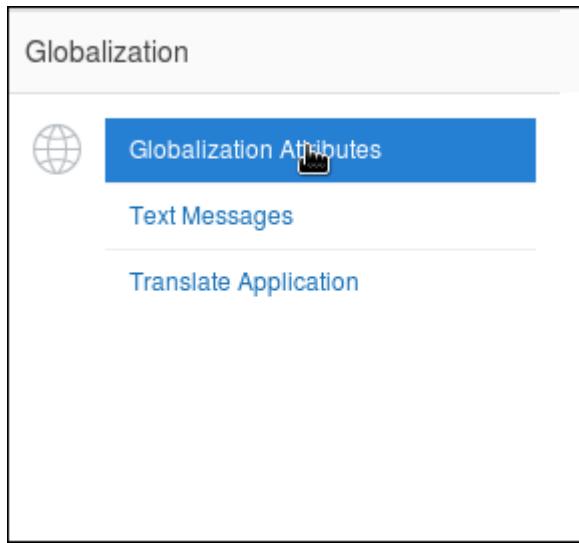
18. You want to display this greeting message in German. Click **Application<n>** on the Developer toolbar.

The screenshot shows the GlobalMart Management Tool home page. The left sidebar has an orange header "GMT | GlobalMart Management Tool" and a menu with items like Home, Product List, Customers, Orders, Products, Help, Admin, and Data Load. The main content area displays a "Translation Region" with the message "Good Morning APEX". Below it is a "Orders By Mode Summary" section showing 32 Online orders and 73 Direct orders. Further down are sections for "In the News" and "News and Events". At the bottom is a dark grey "Developer Toolbar" with icons for Home, Application 101 (which is selected), Edit Page 1, Session, View Debug, Debug, Show Grid, Quick Edit, Theme Roller, and a gear icon.

19. Click **Shared Components**.



20. Under Globalization, select **Globalization Attributes**.



21. Select **German (Germany)(de)** for Application Primary Language and click **Apply Changes**.

The screenshot shows the 'Edit Globalization Attributes' page for 'Application 1'. The 'Globalization' tab is active. The 'Application Primary Language' field is set to 'German (Germany) (de)'. The 'Apply Changes' button is highlighted with a cursor icon.

22. Run the page.

The screenshot shows the application processing results. It displays a green checkmark icon followed by the text 'Application processed.'

23. The Home page is displayed with the greeting message translated in German.

The screenshot shows the GlobalMart Management Tool interface. The top navigation bar includes links for Help, Site Map, Feedback, Sign Off, and View Cart. The left sidebar contains a navigation menu with Home (selected), Product List, Customers, Orders, Products, Help, Admin, and Data Load. The main content area displays a "Translation Region" with the message "Guten Tag APEX". Below this are three buttons: "Manage Customers", "Manage Products", and "Manage Orders". A section titled "Orders By Mode Summary" shows two circular icons: one for "Online" with the number 32 and another for "Direct" with the number 73. Further down, sections for "In the News" and "News and Events" are visible. The bottom navigation bar includes links for Home, Application 101, Edit Page 1, Session, View Debug, Debug, Show Grid, Quick Edit, Theme Roller, and a gear icon.

