ORACLE

APPLICATION EXPRESS



Oracle Application Express: Developing Database Web Applications

Hands-On-Labs Guide

Unit 15: Creating and Using Dynamic Actions and Plug-ins

This exercise includes eight hands-on-labs and uses the Demo Projects application.

HOL 15-1 Creating and Using a Dynamic Action on the Maintain Project Page: In this lab, you add a dynamic action on the Maintain Project page to show or hide the Completed Date based on the value of Status.

HOL 15-2 Creating and Using a Dynamic Action on the Maintain Task Page: In this lab, you add a dynamic action on the Maintain Task page to populate the Due Date whenever the Milestone is changed.

HOL 15-3 Creating and Using Dynamic Actions on the Project Tree Page: This lab covers creating two buttons that you use to expand and contract the project tree nodes. You create a dynamic action on each of these buttons. Then, for each of the button, you set the action as defined by the dynamic action.

HOL 15-4 Creating and Using a Plug-in: In this hands-on lab you create a copy of the Badge List Plug-in from the Sample Charts packaged app. Then, on the Reports page, you create a badge list region. On this region, you display three different badges: Open Projects, Upcoming Milestones, and Open Tasks.

HOL 15-5 Adding Additional Reports: In this lab, you clean up the Dashboard page by moving two regions as separate pages. Later, you add in new regions to the Dashboard page.

HOL 15-6 Updating the Reports Page: In this lab, you add new components to the Reports page of the Demo Projects application.

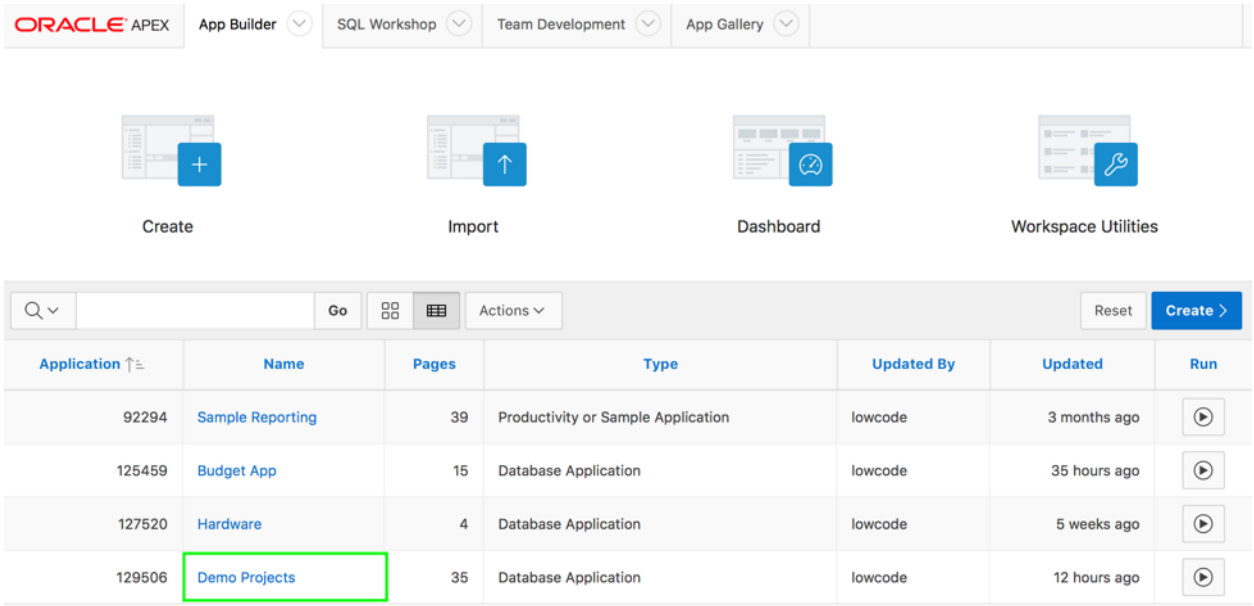
HOL 15-7 Updating the Dashboard Page: In this lab, you update the Dashboard page of the Demo Projects application. The Dashboard should display the Project Status chart, Upcoming Milestones report and Incomplete Tasks.

HOL 15-8 Updating the Home Page: In this lab, you update the navigation menu of the Demo Projects application and also update the List region on the Home page.

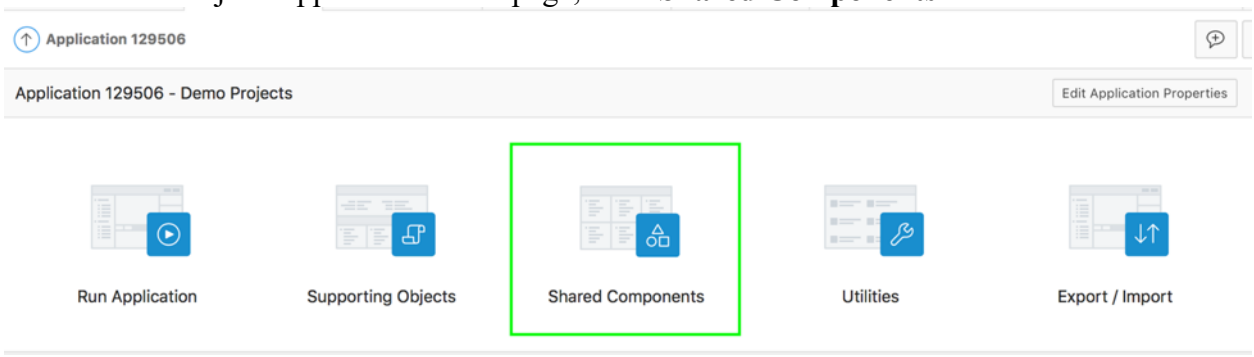
HOL 15-1: Creating and Using a Dynamic Action on the Maintain Project Page

On the Maintain Project page, the Completed Date field is displayed irrespective of the value of Status. In this lab, you add a dynamic action on the page to show or hide the Completed Date based on the value of Status.

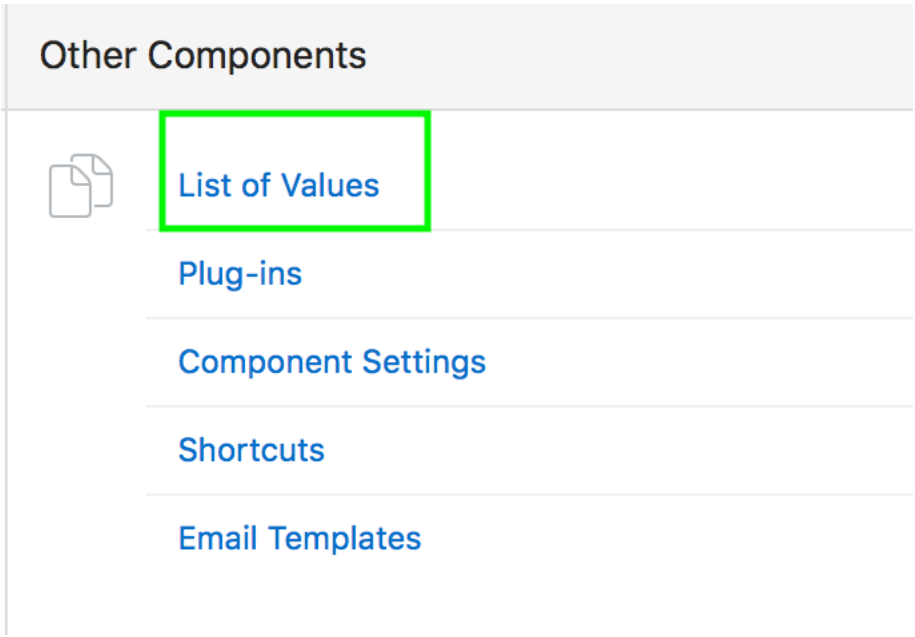
1. Navigate to App Builder and click the **Demo Projects** application.



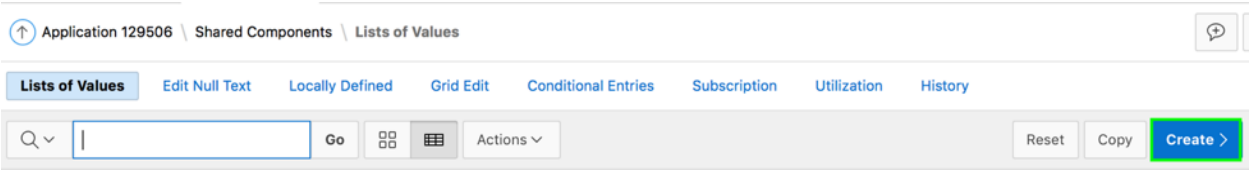
2. In the Demo Projects application home page, click **Shared Components**.



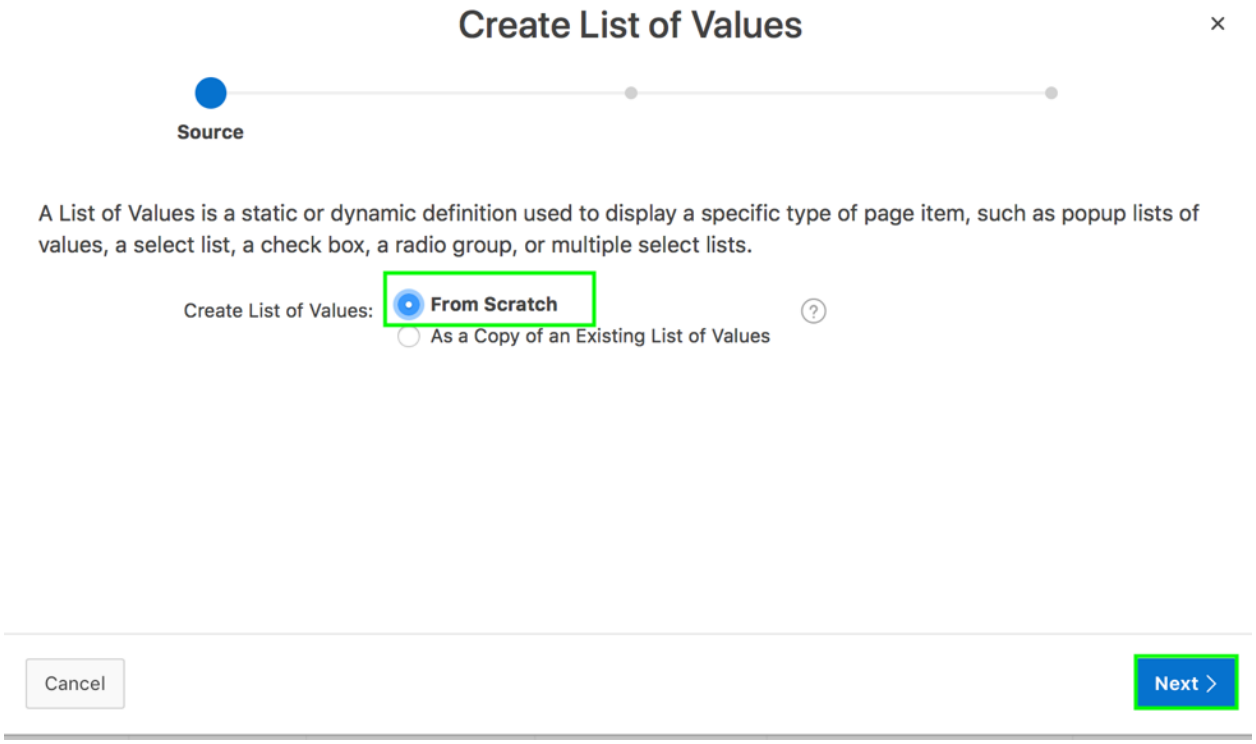
3. You want to create a dynamic LOV for Status. Under Other Components, click **List of Values**.



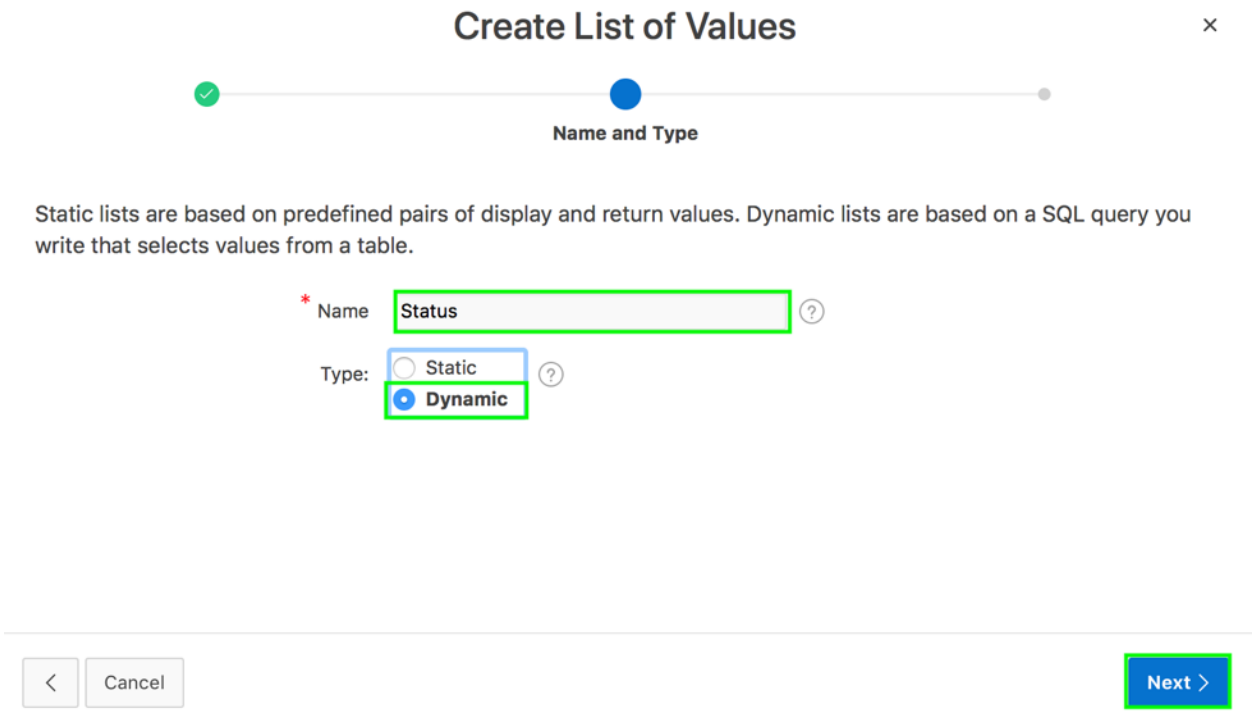
4. Click **Create**.



5. Accept the default for Source and click **Next**.



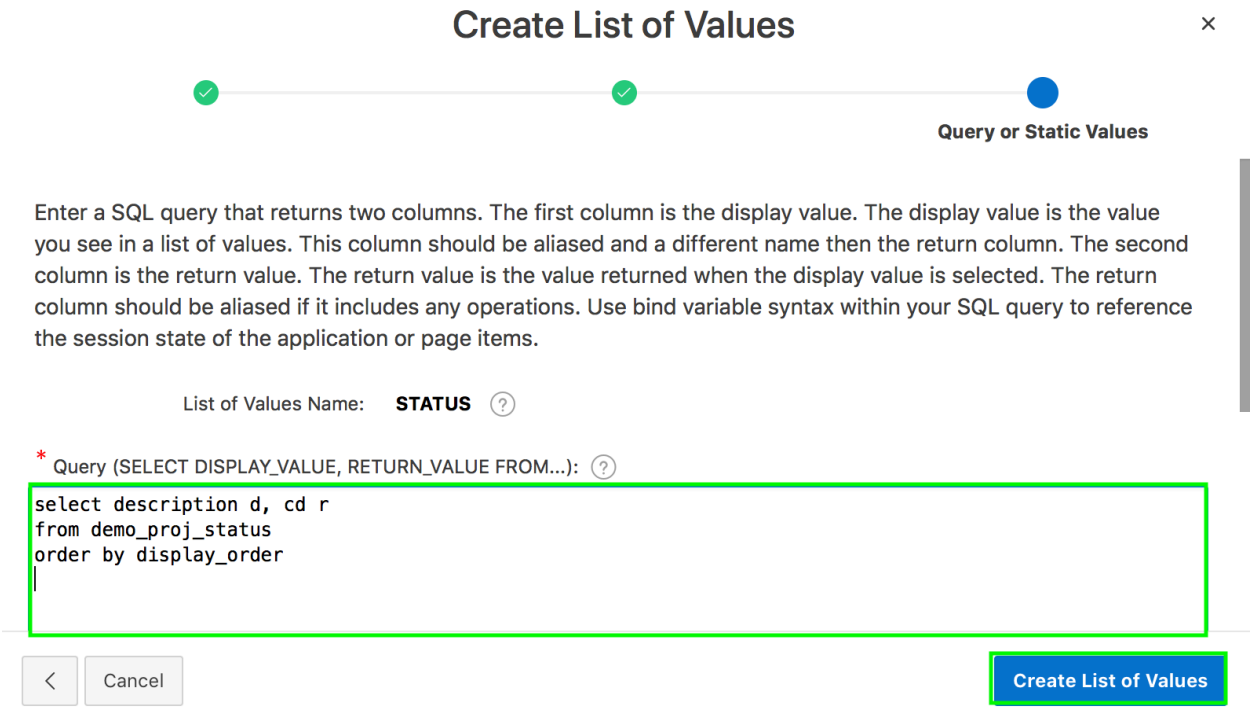
6. For Name, enter **STATUS**, select **Dynamic** for Type and click **Next**.



7. For Query, copy and paste the following SQL:

```
select description d, cd r
from demo_proj_status
order by display_order
```

Click **Create List of Values**.



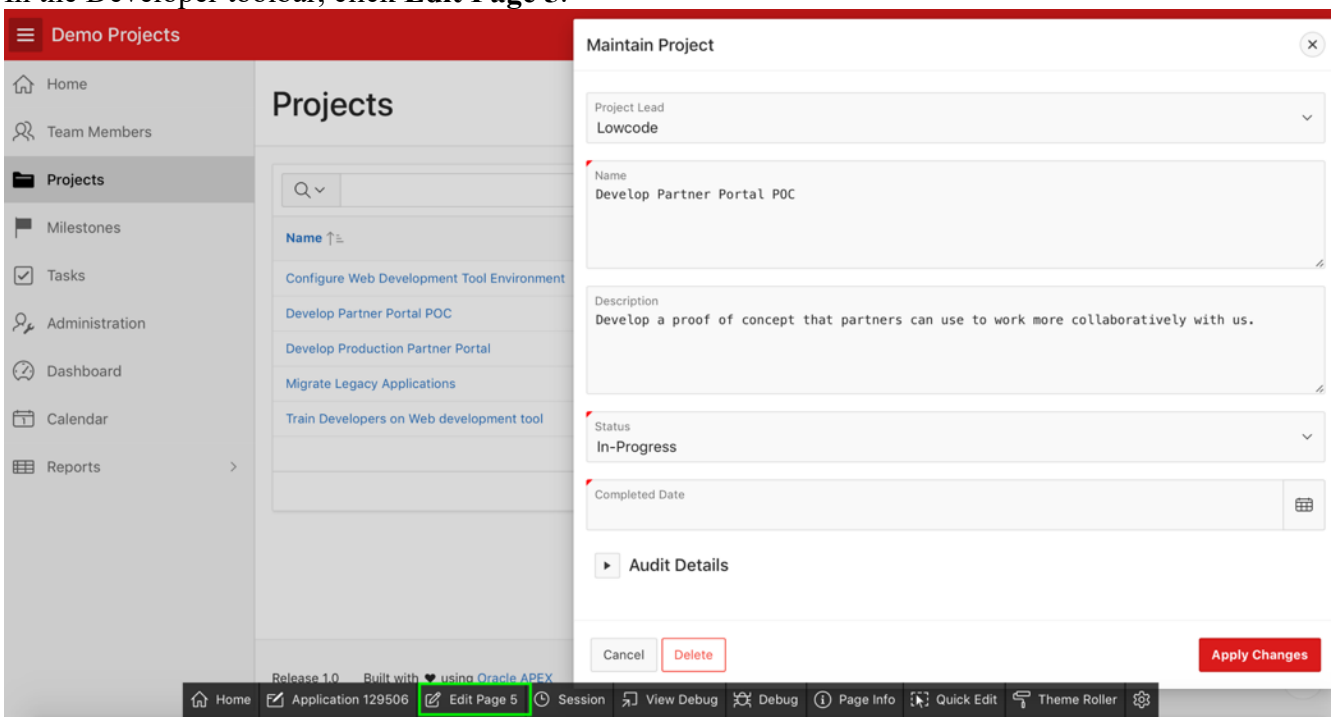
8. In the Application Express toolbar, click **Run Page 1**.

9. In the Demo Projects application runtime environment, click **Projects** in the navigation menu.
   Then, in the interactive report, select a project name with status as IN-PROGRESS or ASSIGNED.

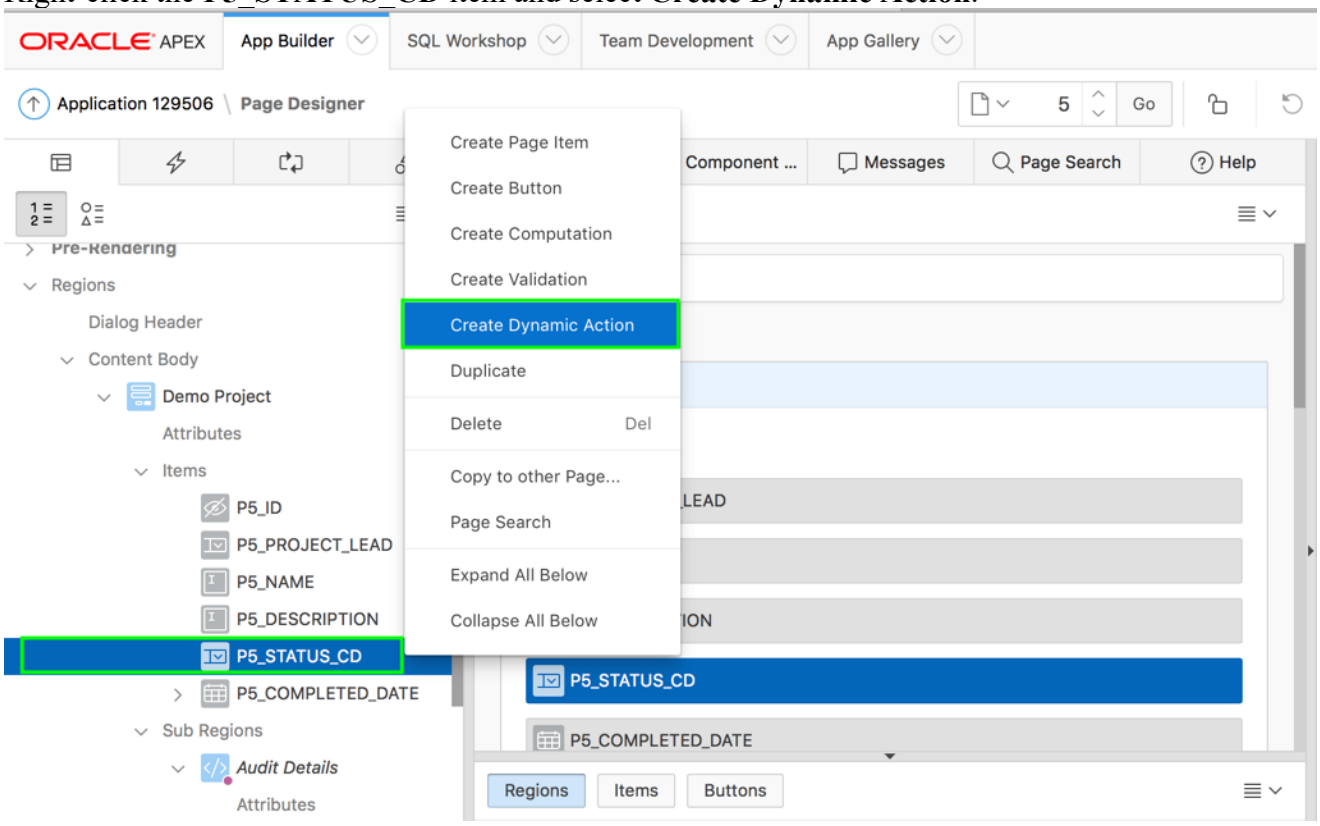
   In the dialog page, for Status, select each of the values Assigned, Completed, and In-Progress. Notice that the Completed Date field is displayed irrespective of the status.

   You want to create a dynamic action that will allow the display of the Completed Date field only if the status is *Completed*.

In the Developer toolbar, click **Edit Page 5**.

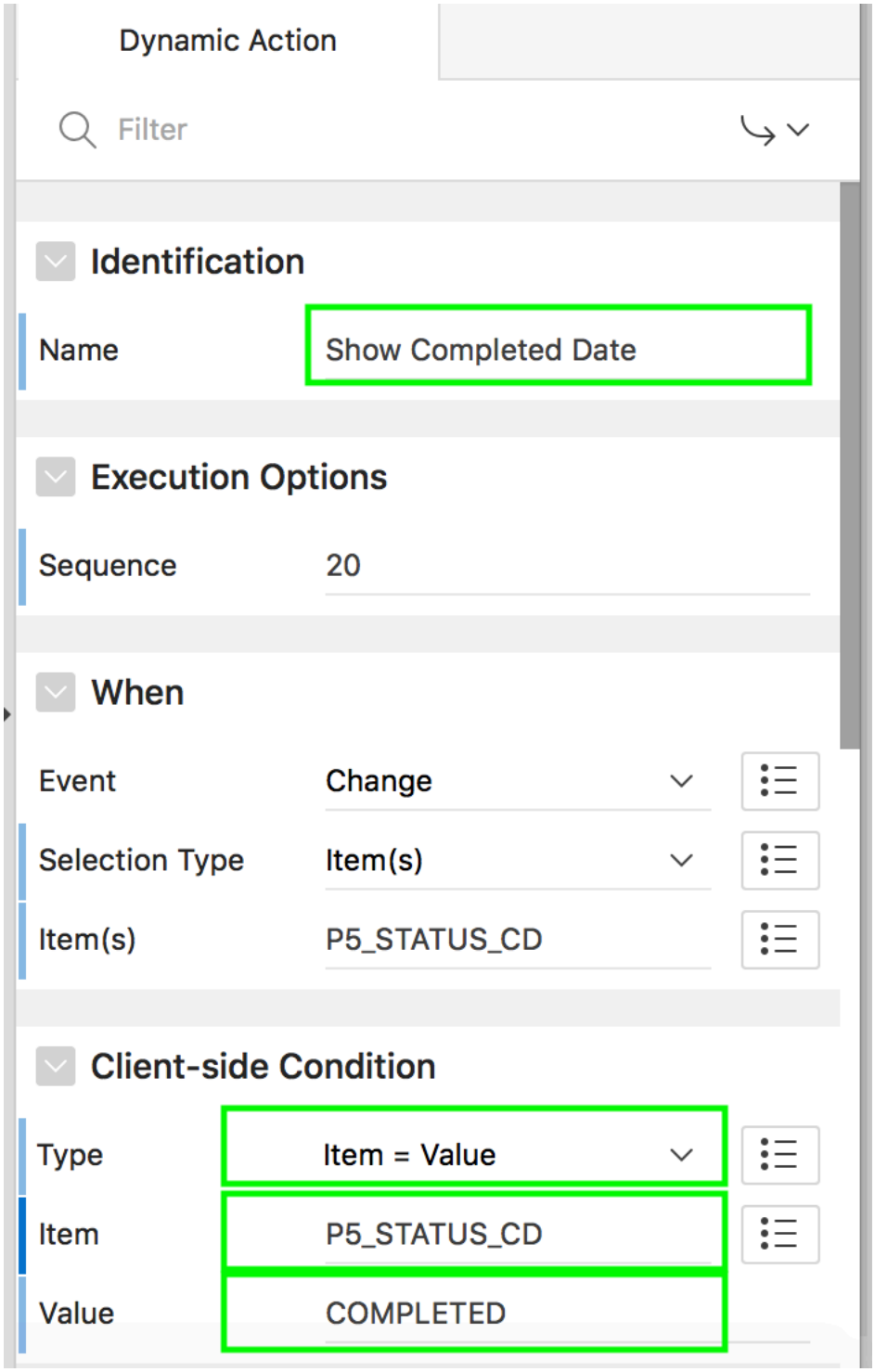


10. In the page designer, expand **Regions > Demo Project > Items**.
Right-click the **P5_STATUS_CD** item and select **Create Dynamic Action**.

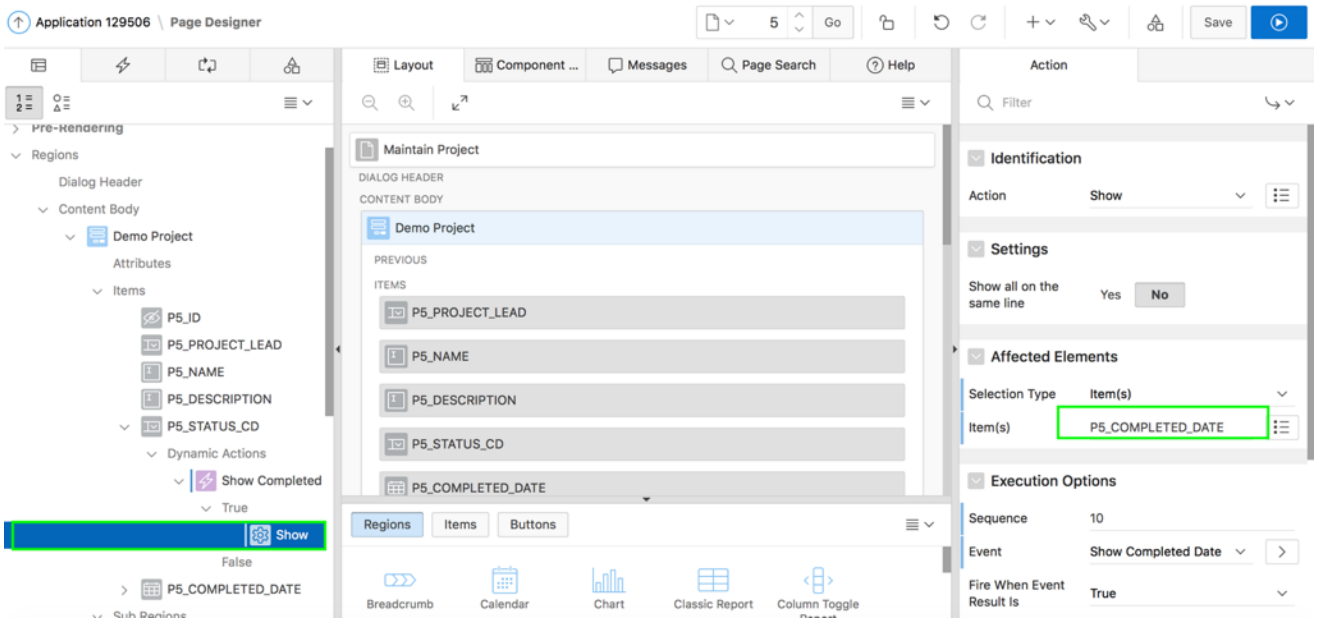


11. In the property editor:
    - Identification > Name: Enter **Show Completed Date**
    - Client-side Condition > Type: Select **Item = Value**
    - Client-side Condition > Item: Select **P5_STATUS_CD**
    - Client-side Condition > Value: Enter **COMPLETED**

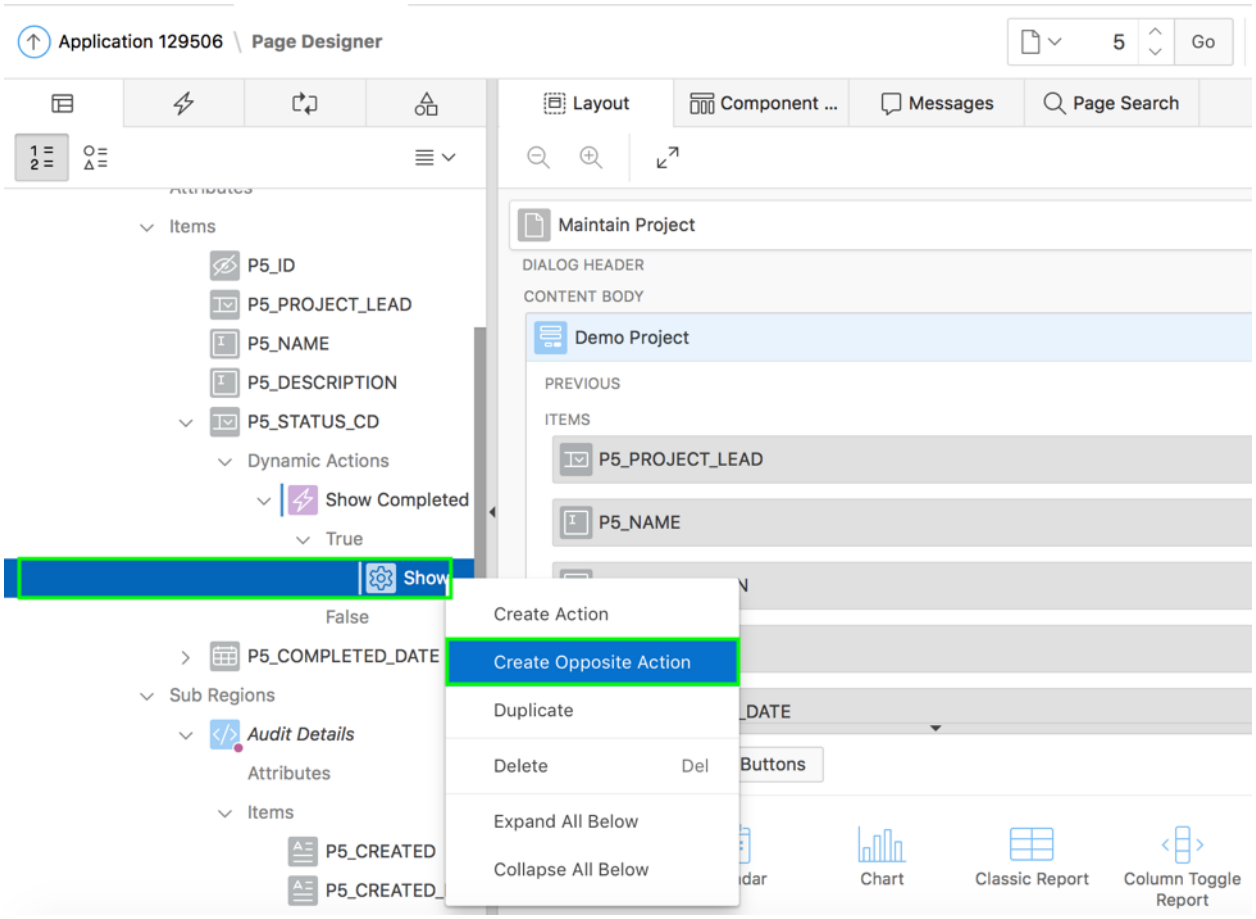
    Note: The capitalization and spelling of the value must match the data entry value exactly in order for the dynamic action to fire.



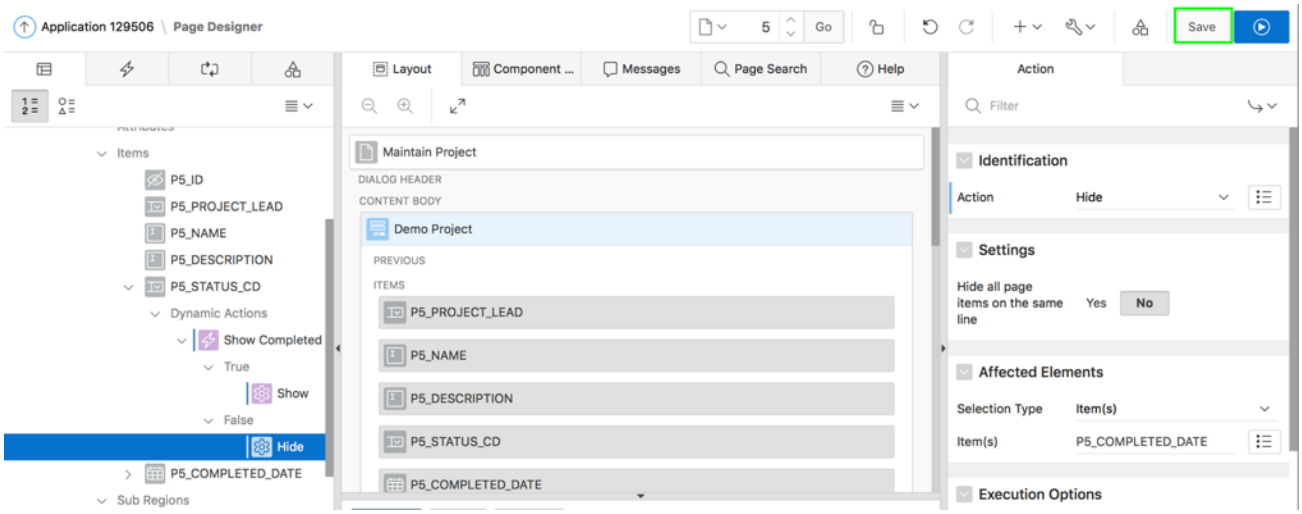
12. In the Rendering tree, under the Show Completed Date dynamic action, expand the **True** node and select **Show**.
In the Property Editor, for Item(s) select **P5_COMPLETED_DATE**.



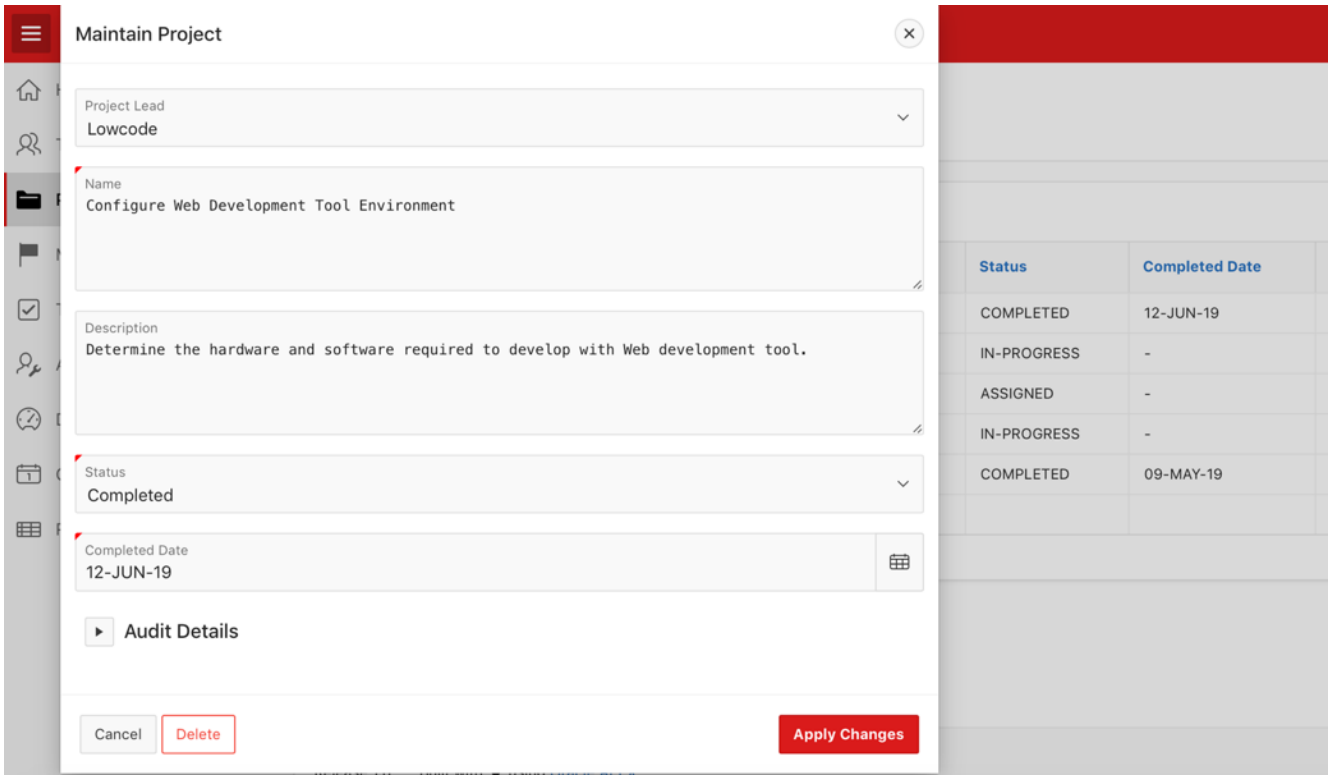
13. Right-click **Show** and select **Create Opposite Action**.



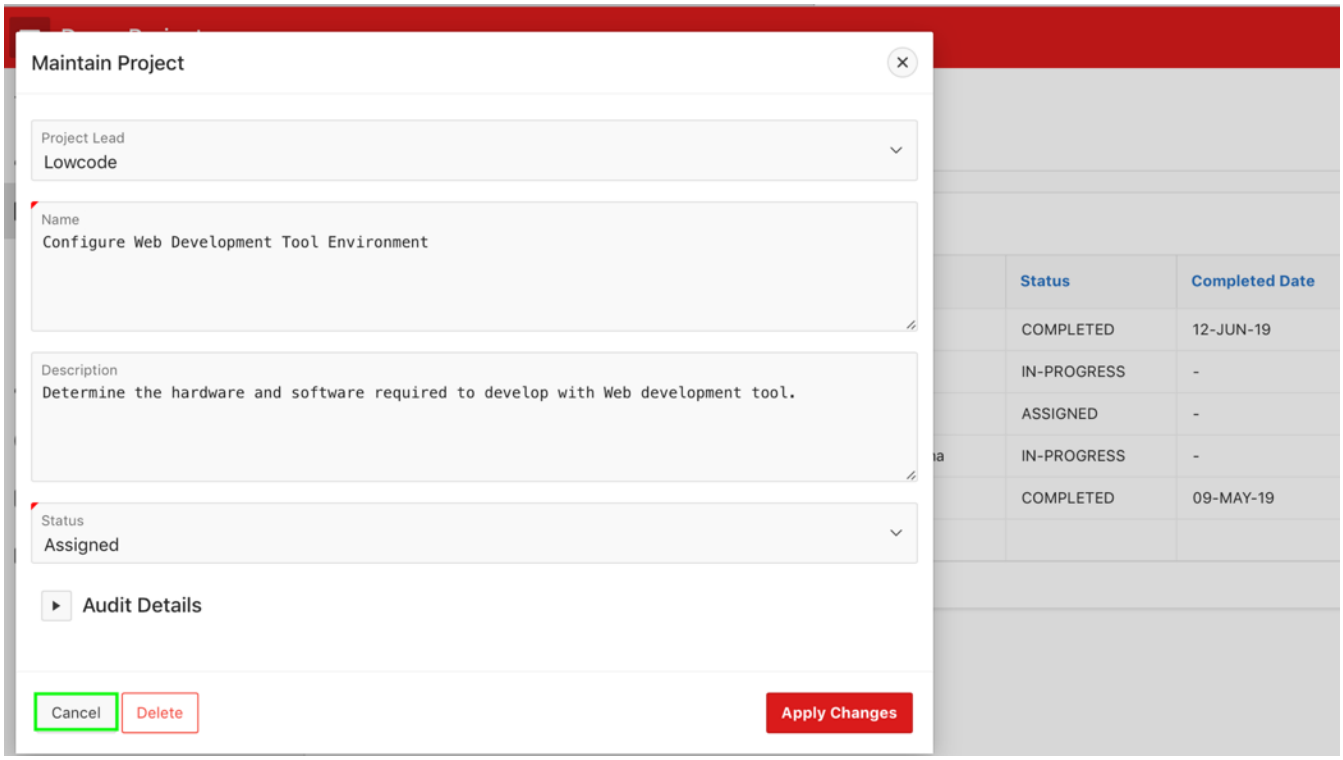
14. Notice that the opposite action is created now. The Completed Date field will be hidden if the Status includes a value other than COMPLETED.
    Click **Save**.



15. Now, you can verify to see if the dynamic action works as expected.
    Navigate to the Demo Projects application runtime environment. If the Maintain Project dialog is open, click **Cancel**.
    In the navigation menu, click **Projects** and then in the report, click any project name with the Status as COMPLETED.
    Notice that the Completed Date field is displayed.



16. In the Maintain Project dialog, for Status, select either Assigned or In-Progress.
Notice that this time, the Completed Date field is hidden.
Click the **Cancel** button.



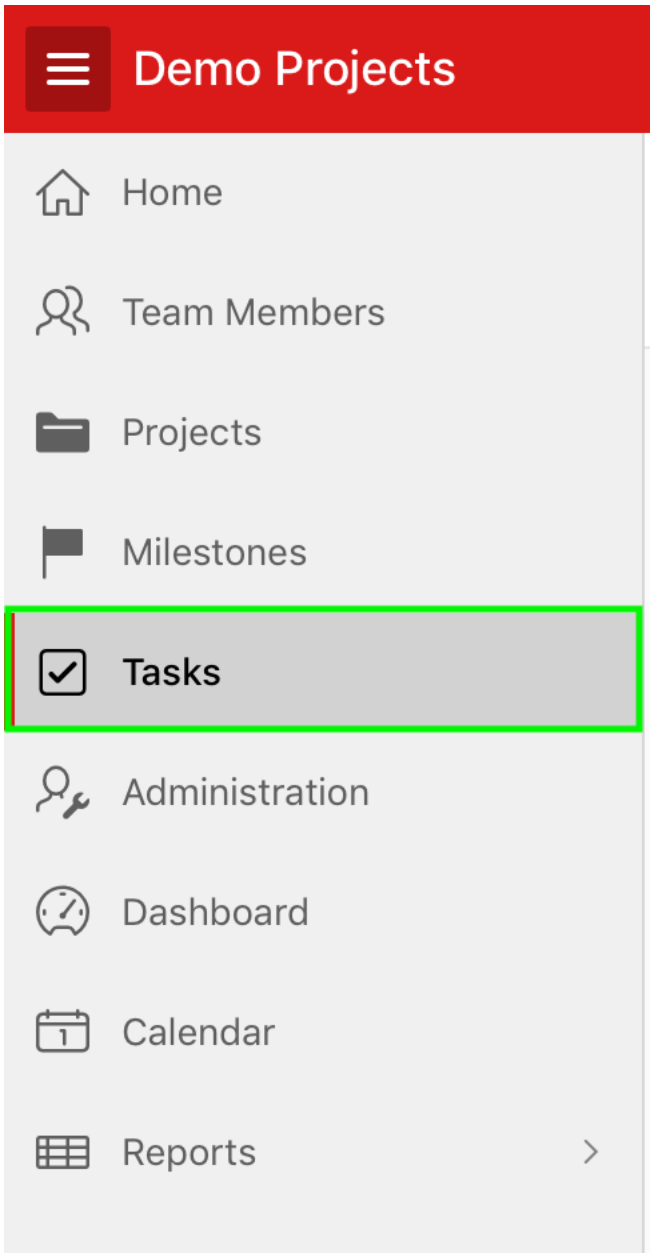
HOL 15-2: Creating and Using a Dynamic Action on the Maintain Task Page

In this hands-on lab, you add a dynamic action on the Maintain Task page to populate the Due Date whenever the Milestone is changed.

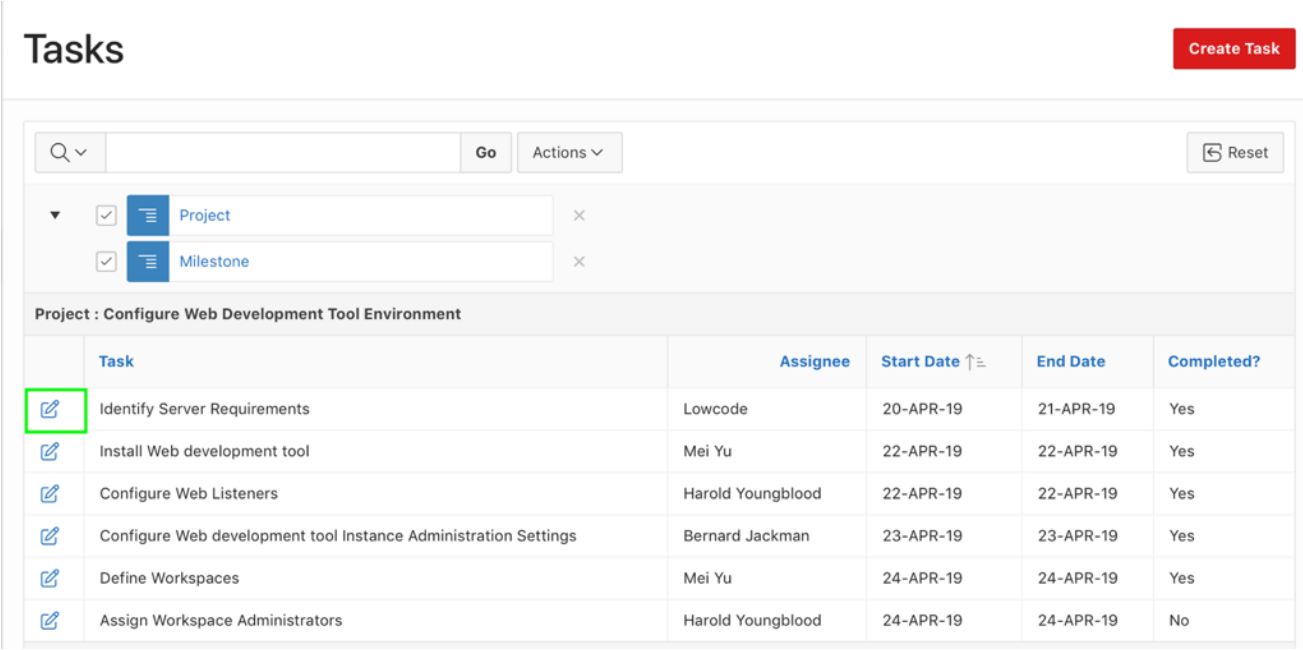
Note: If you have not renamed the breadcrumb entry for Page 9 in a previous lab, you need to rename it now. Perform the following steps:

a) In the Demo Projects application runtime environment, click Tasks in the navigation menu.
b) In the interactive report, click the Edit icon for a task name
c) If the breadcrumb entry still shows Demo Proj Tasks, click **Edit Page 9** in the Developer Toolbar.
d) In the page designer, click **Page Shared Components**.
e) Expand Breadcrumbs and select **Breadcrumb**.
f) In the property editor, click **Edit Component**.
g) Click Demo Proj Tasks.
h) Under Entry, for Short Name, enter **Maintain Task**.
i) Click **Apply Changes**.

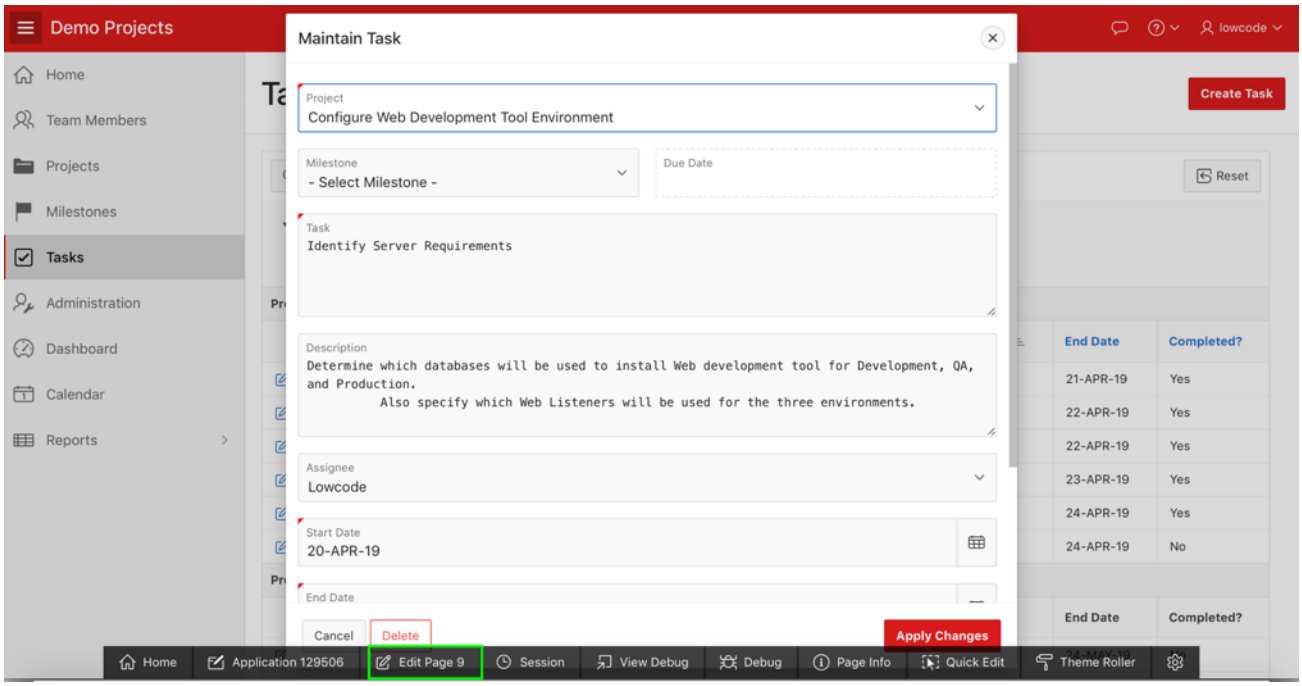
1. In the Demo Projects application runtime environment, click **Tasks** in the navigation menu.


Demo Projects
Home
Team Members
Projects
Milestones
Tasks
Administration
Dashboard
Calendar
Reports

2. In the interactive report, click the Edit icon for a task name.



3. In the Developer Toolbar, click **Edit Page 9**.
On this page, you want to populate the Due Date whenever the Milestone is changed. In order to achieve this you will use a Dynamic Action with an action of Set Value, which can execute an AJAX call to retrieve data from the database.



4. Under Rendering > Regions > Maintain Task, expand items.
   Right-click **P9_MILESTONE_ID** and select **Create Dynamic Action**.

5. For the new Dynamic Action, in the Property Editor, for Name enter **Get Due Date**.

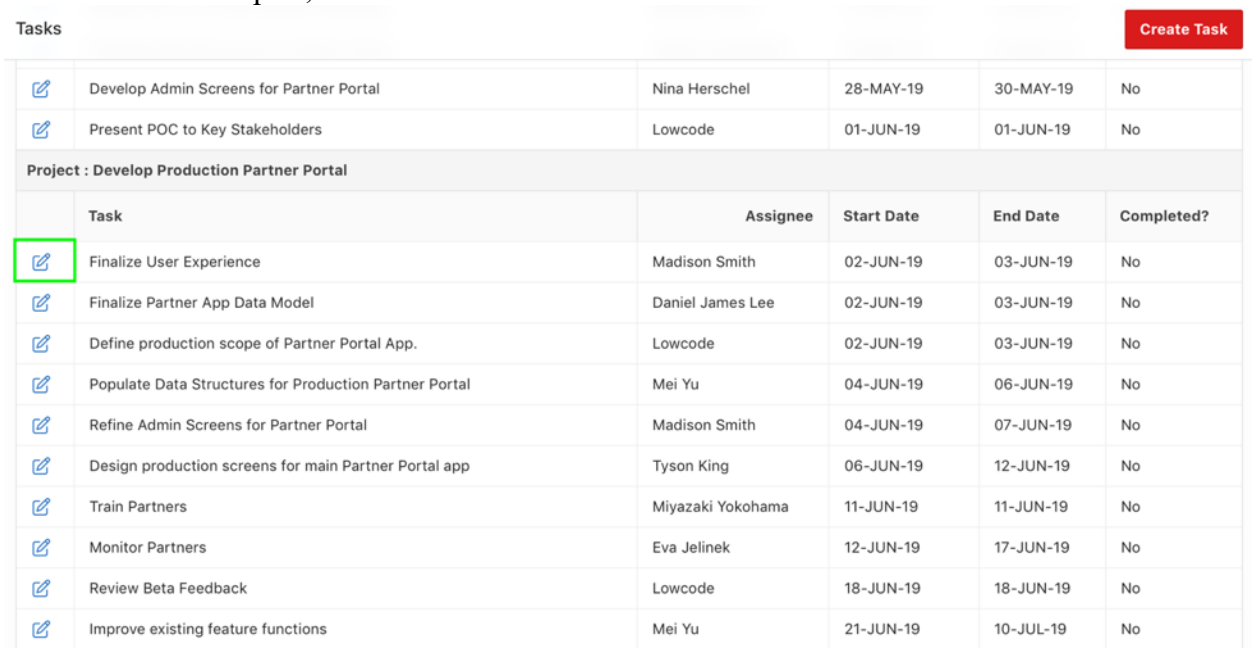
6. In the Rendering tree, under the P9_MILESTONE_ID item, select the **Action** under the True node (currenly labeled X Show).
   In the Property Editor:
   - Identification: Action - select **Set Value**
   - Settings: Set Type - select **SQL Statement**
   - Settings: SQL Statement - copy and paste the following code:

```
select due_date
from demo_proj_milestones
where id = :P9_MILESTONE_ID
```

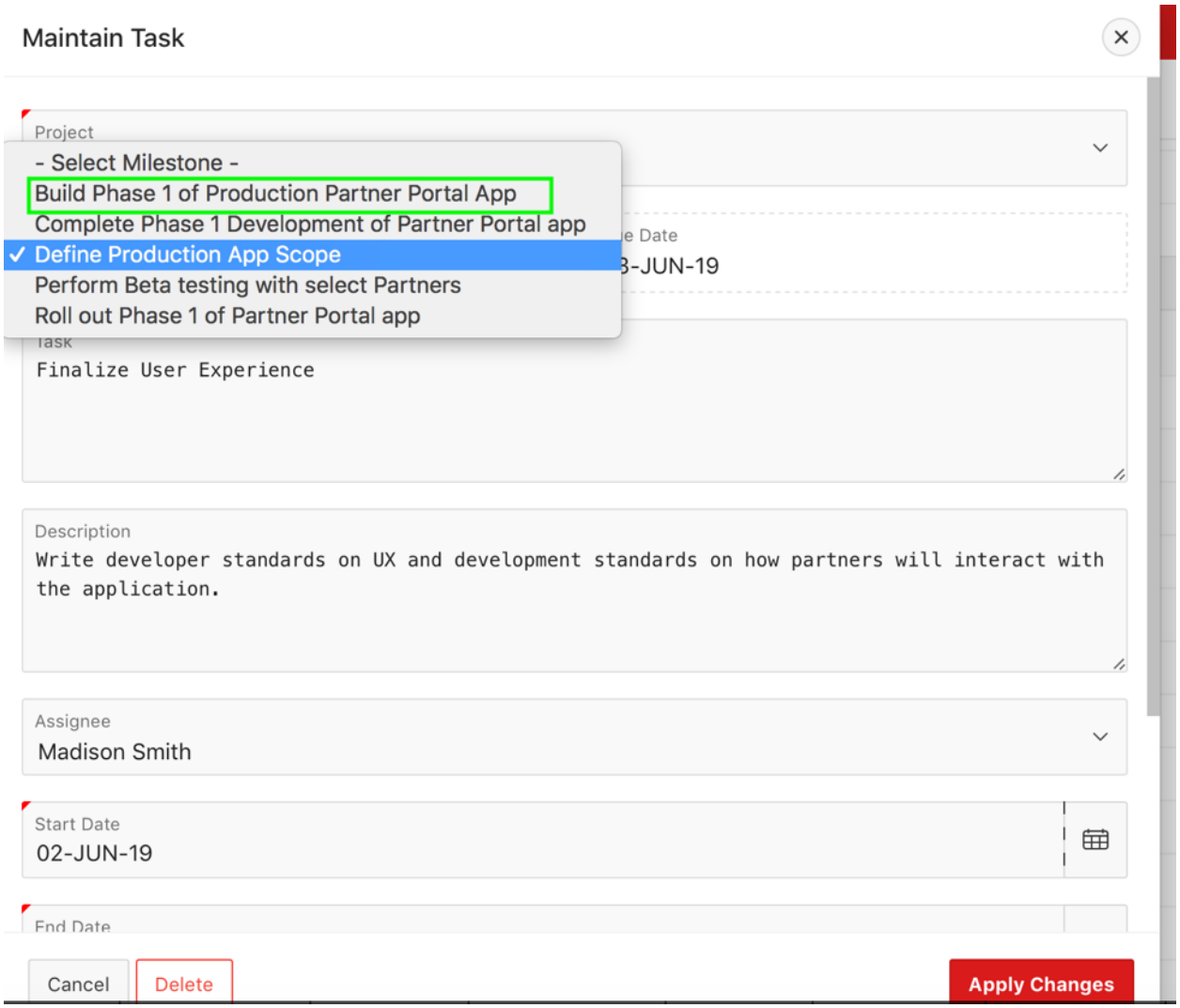
   - Settings: Items to Submit - select **P9_MILESTONE_ID**
   - Affected Elements: Item(s) - select **P9_MILESTONE_DUE_DATE**

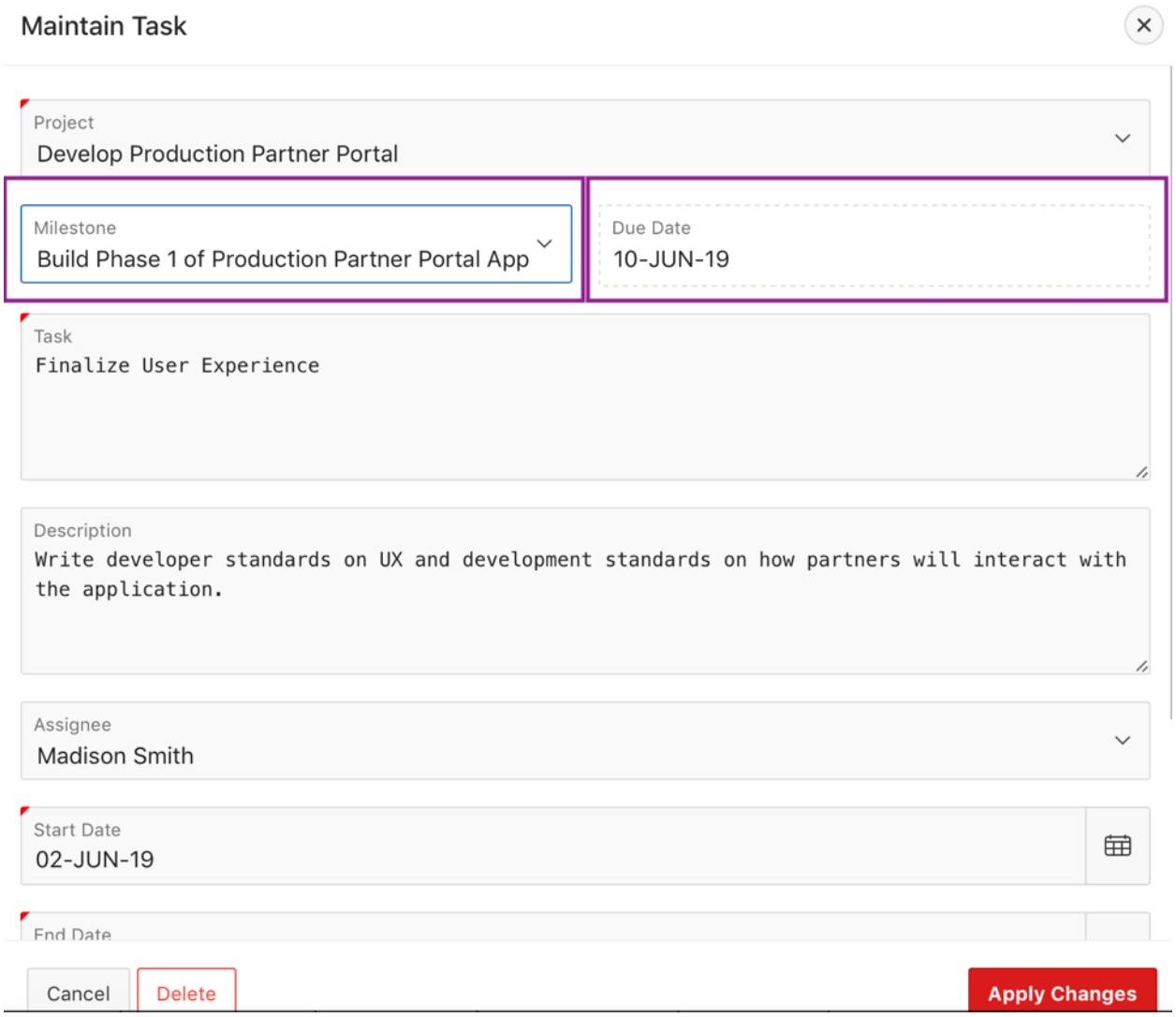
   Click **Save**.

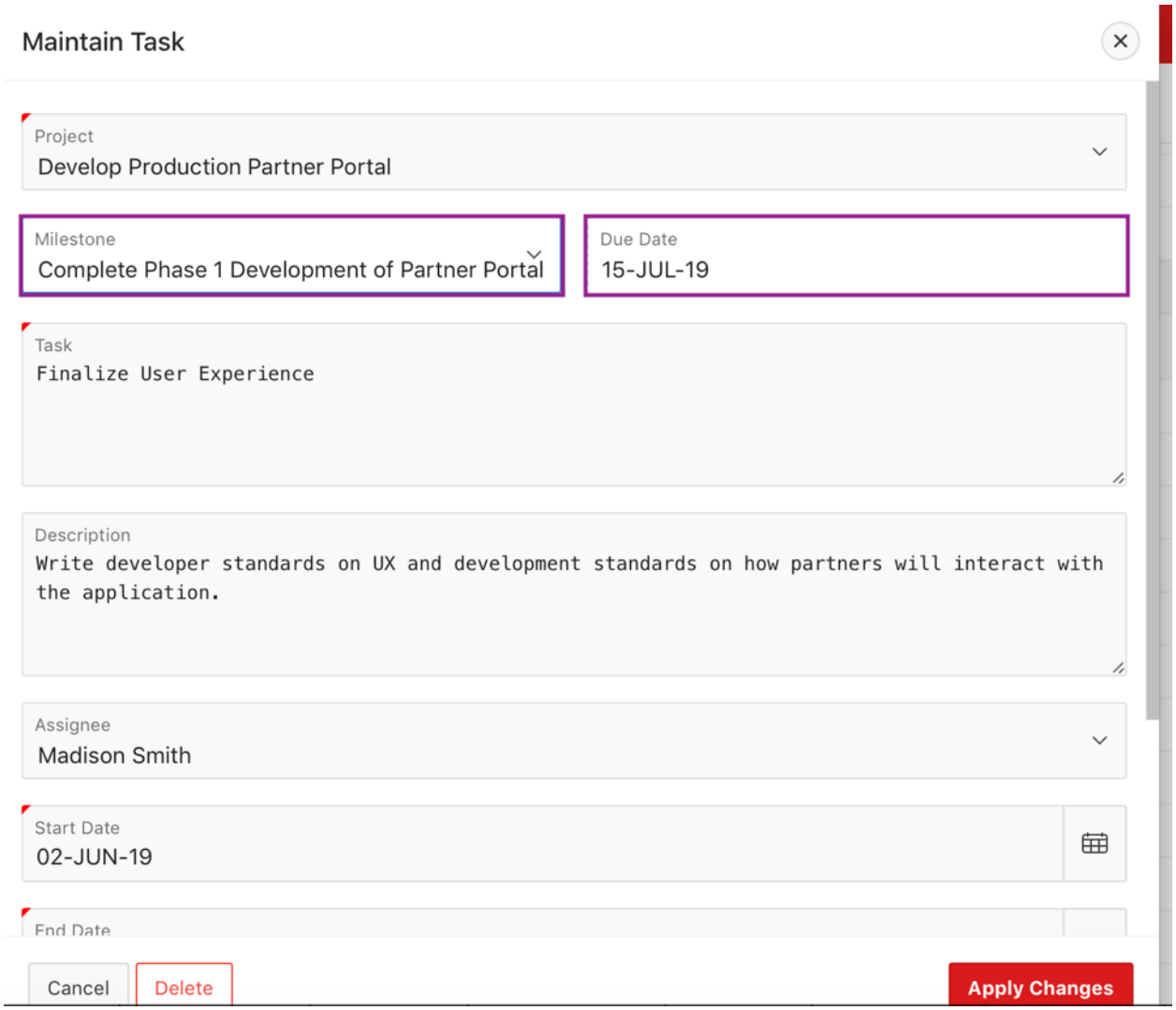
7. Navigate to Demo Projects application runtime environment.
   In the navigation menu, click **Tasks**.
   In the interactive report, click the Edit icon for a task with milestones.



8. Try selecting different values for **Milestone** to see how the Due Date is updated based on the selection.


Maintain Task
Project
- Select Milestone -
Build Phase 1 of Production Partner Portal App
Complete Phase 1 Development of Partner Portal app
Define Production App Scope
Perform Beta testing with select Partners
Roll out Phase 1 of Partner Portal app
e Date
3-JUN-19
Task
Finalize User Experience
Description
Write developer standards on UX and development standards on how partners will interact with the application.
Assignee
Madison Smith
Start Date
02-JUN-19
End Date
Cancel
Delete
Apply Changes


Maintain Task
Project
Develop Production Partner Portal
Milestone
Build Phase 1 of Production Partner Portal App
Due Date
10-JUN-19
Task
Finalize User Experience
Description
Write developer standards on UX and development standards on how partners will interact with the application.
Assignee
Madison Smith
Start Date
02-JUN-19
End Date
Cancel
Delete
Apply Changes


Maintain Task
Project
Develop Production Partner Portal
Milestone
Complete Phase 1 Development of Partner Portal
Due Date
15-JUL-19
Task
Finalize User Experience
Description
Write developer standards on UX and development standards on how partners will interact with the application.
Assignee
Madison Smith
Start Date
02-JUN-19
End Date
Cancel
Delete
Apply Changes

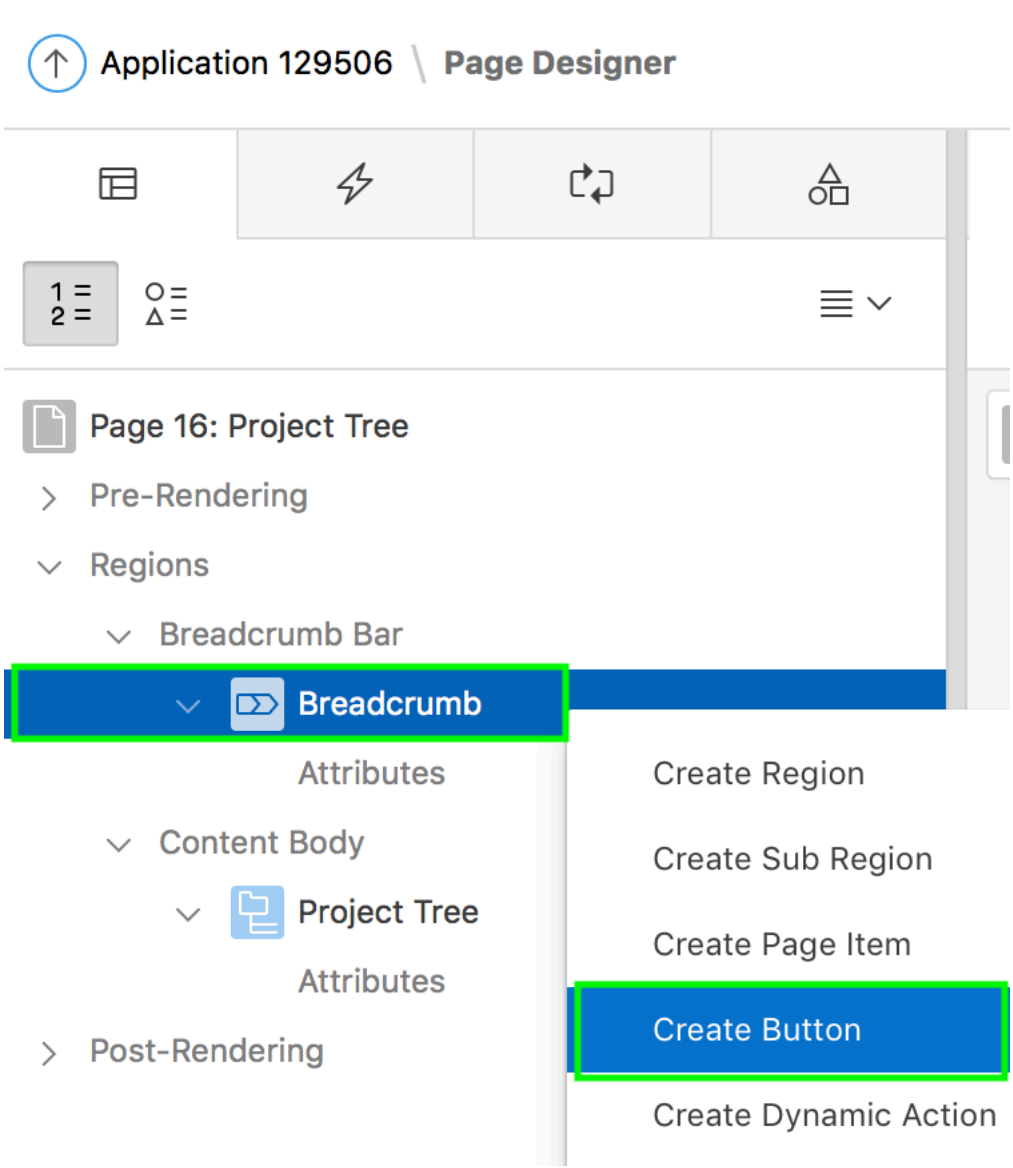
HOL 15-3: Creating and Using Dynamic Actions on the Project Tree Page

In this lab, you update the Project Tree page that you created in HOL 14-6. You add two buttons: Collapse All and Expand All. You use them to expand and contract the project tree nodes. You create a dynamic action on each of these buttons. Then, for each of the button, you set the action as defined by the dynamic action.

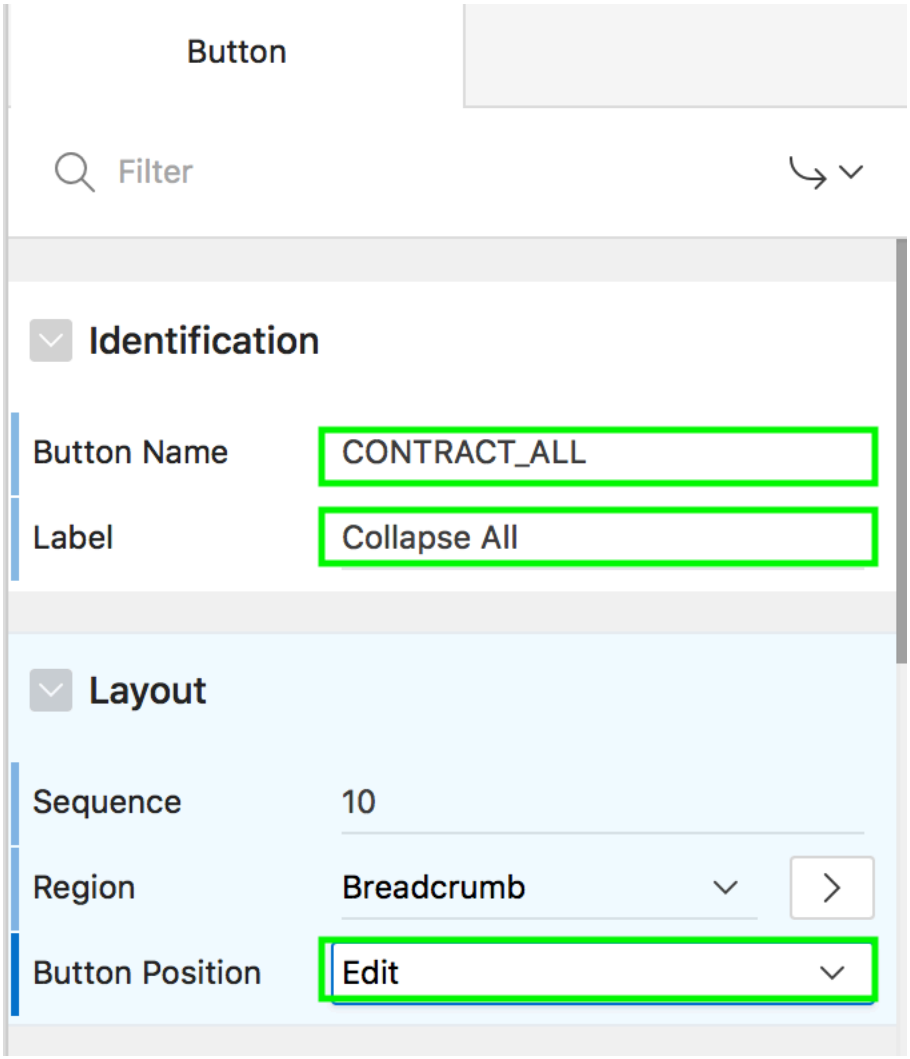
1. Navigate to Demo Projects application runtime environment. In the navigation menu, under Reports, click **Project Tree**.
   In the Developer Toolbar, click **Edit Page 16**.

2. Now, to expand and collapse the project tree nodes, you want to create Expand All and Collapse All buttons on the Project Tree button. These buttons' actions are defined by dynamic actions.
   Under Rendering, expand **Regions > Breadcrumb**.

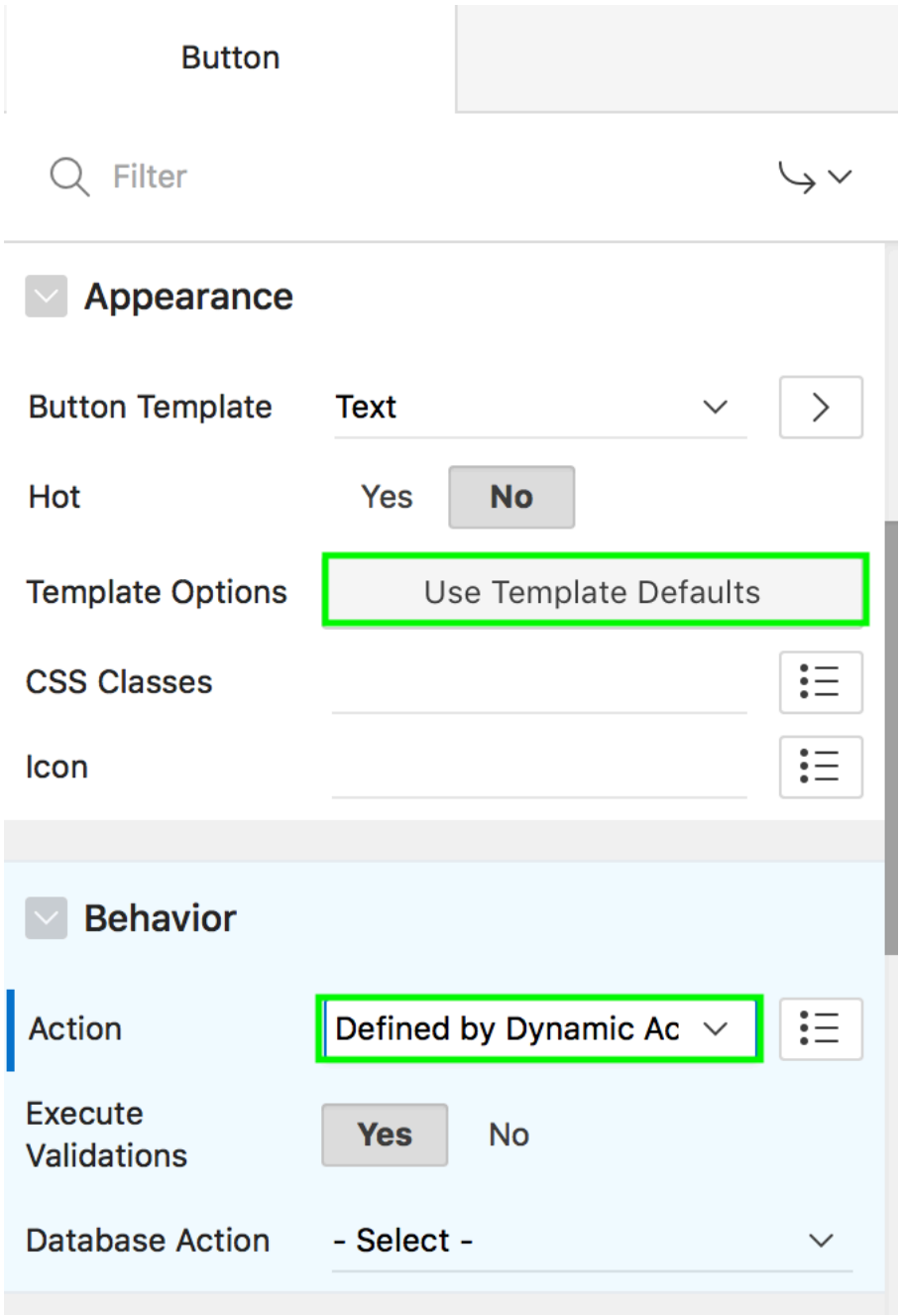
Right-click **Breadcrumb** and select **Create Button**.



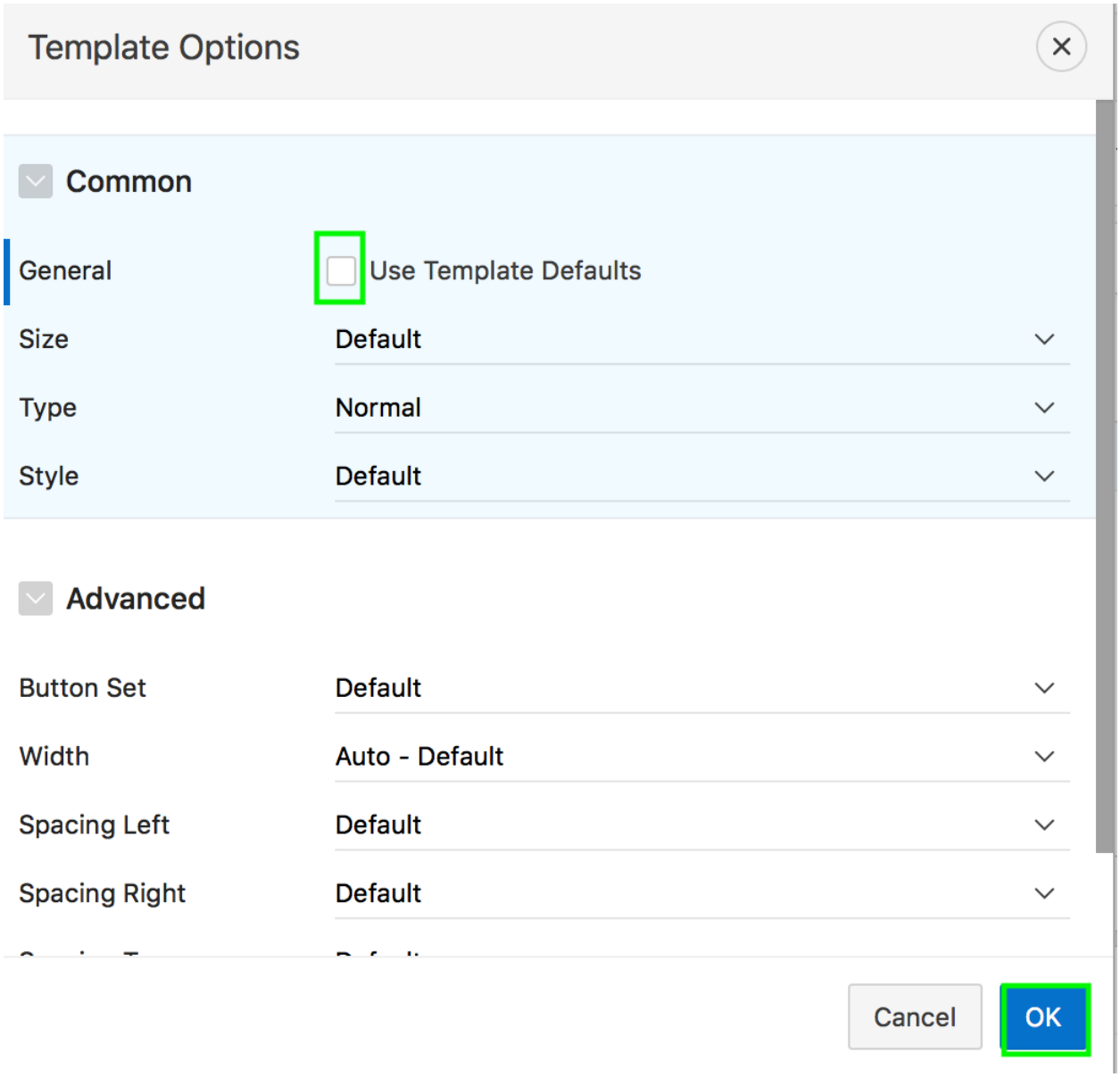
3. In the Property Editor:
   - Identification > Button Name: Enter **CONTRACT_ALL**
   - Identification > Label: Enter **Collapse All**
   - Layout > Button Position: Select **Edit**



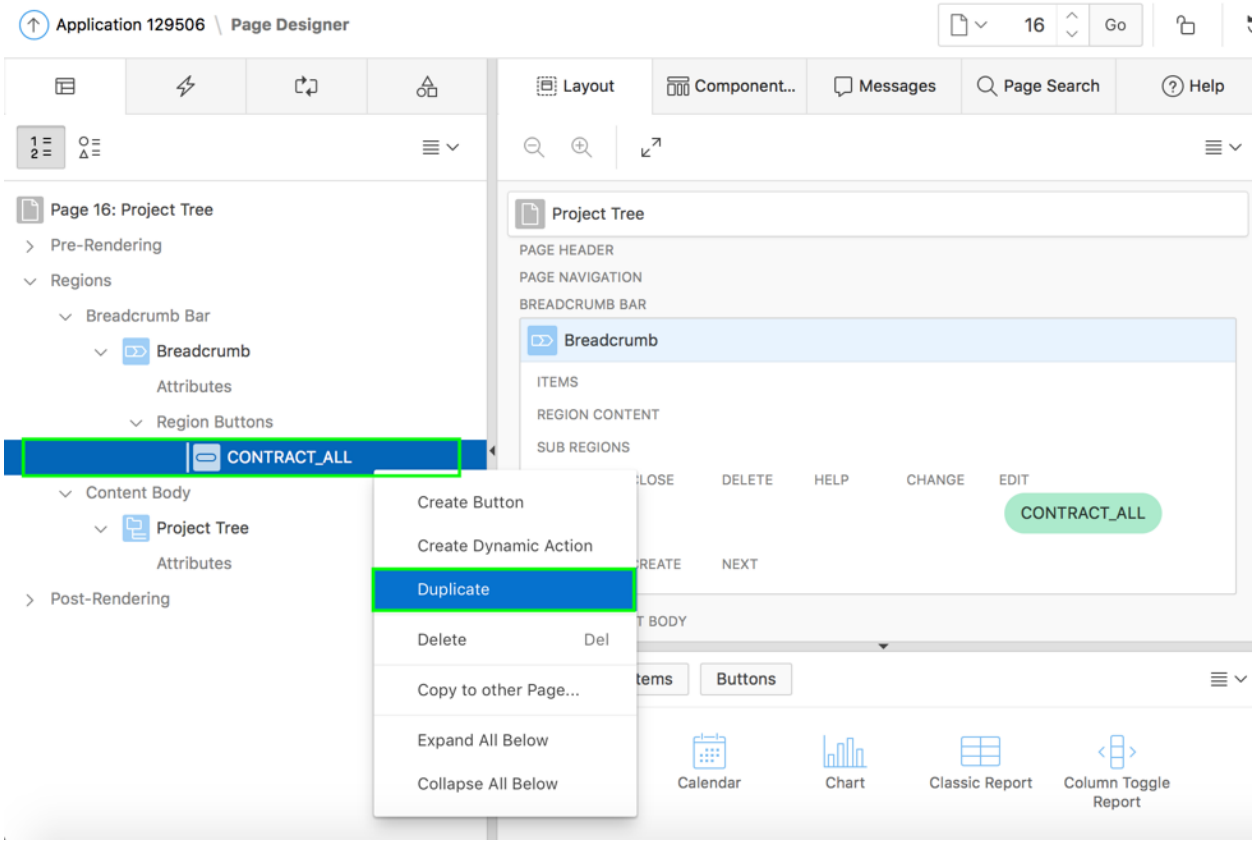
4. In the Property Editor, for Behavior > Action: Select **Defined by Dynamic Action**. Under Appearance, click **Template Options** button.


Button
Filter
Appearance
Button Template
Text
Hot
Yes
No
Template Options
Use Template Defaults
CSS Classes
Icon
Behavior
Action
Defined by Dynamic Ac
Execute Validations
Yes
No
Database Action
- Select -

5. Deselect the **Use Template Defaults** check box and click **OK**.



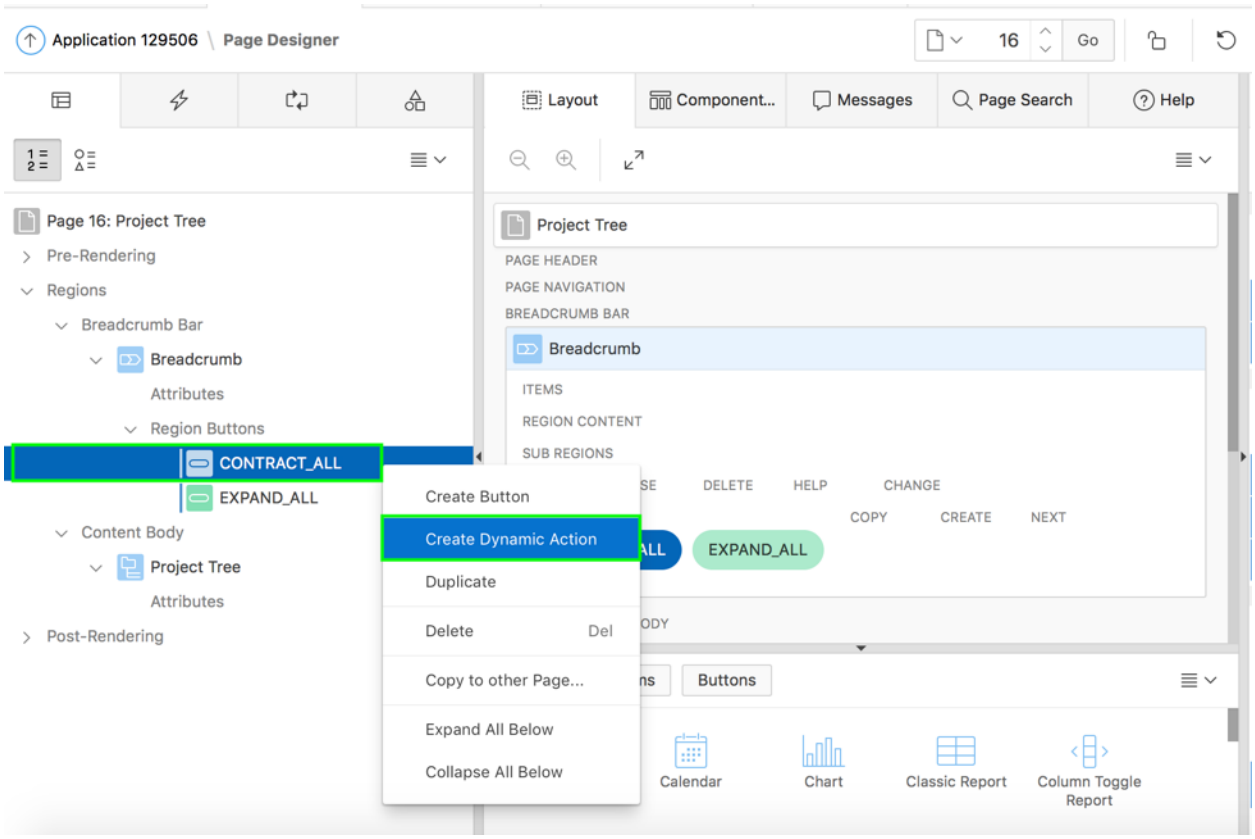
6. Now, right-click the **CONTRACT_ALL** button and select **Duplicate**.



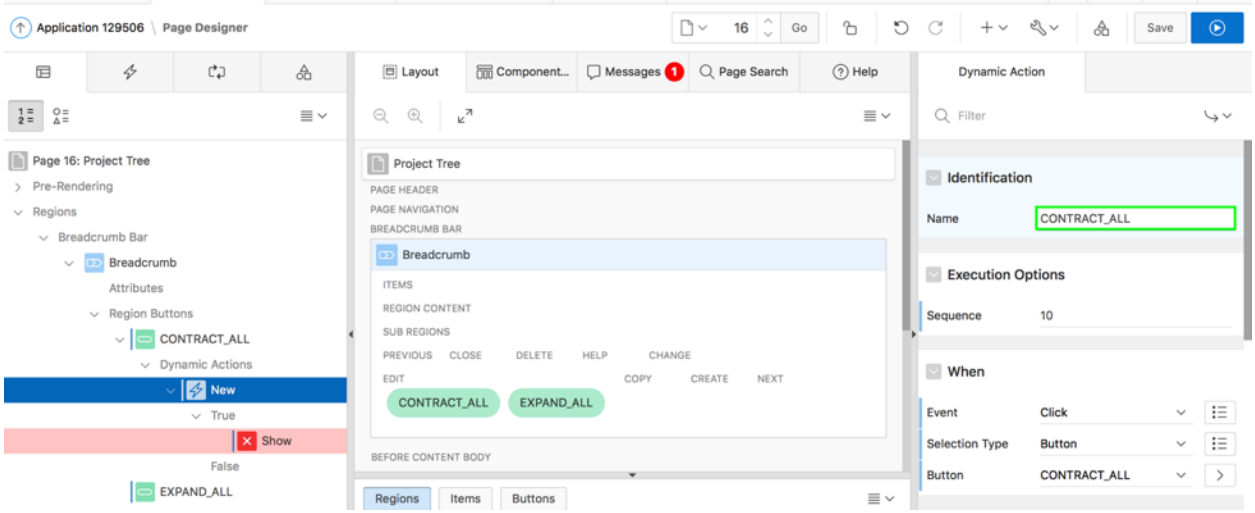
7. In the Property Editor:
   - Identification > Button Name: Enter **EXPAND_ALL**
   - Identification > Label: Enter **Expand All**
   - Layout > Button Position: Select **Edit**
   - Behavior > Action: Select **Defined by Dynamic Action**

   Under Appearance, click **Template Options** button. Deselect the **Use Template Defaults** check box and click **OK**.

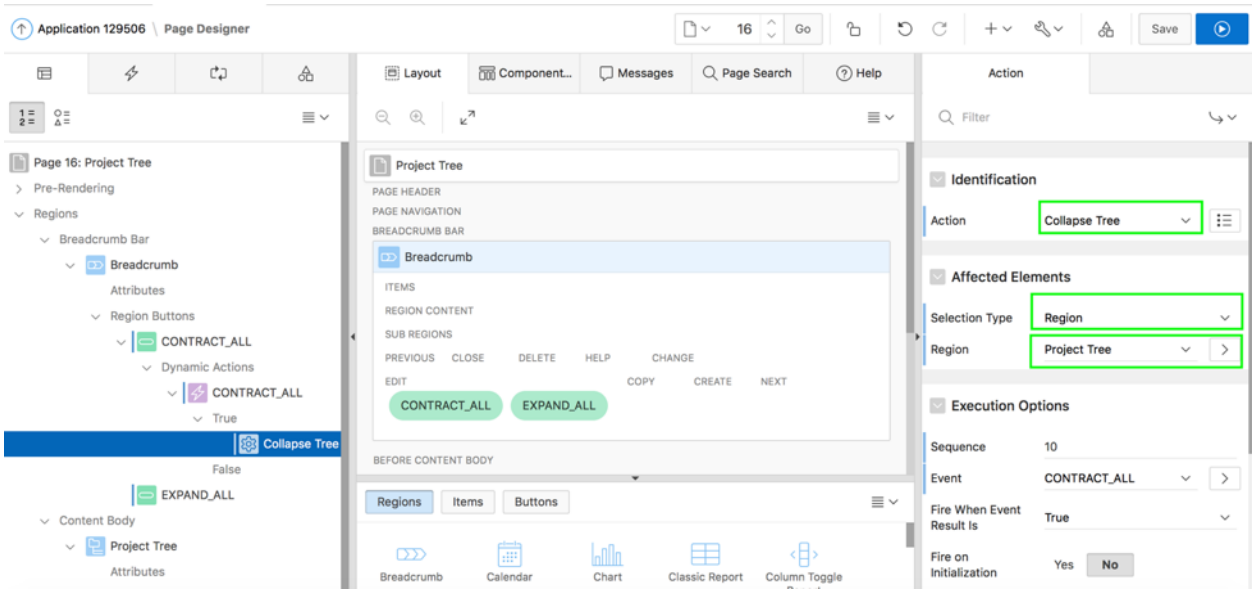
8. Now you want to create a dynamic action for each of these two buttons.
   Under Rendering, expand Regions > Breadcrumb > Region Buttons.
   Right-click the **CONTRACT_ALL** button and select **Create Dynamic Action**.



9. In the property editor, for Identification > Name, enter **CONTRACT_ALL**.



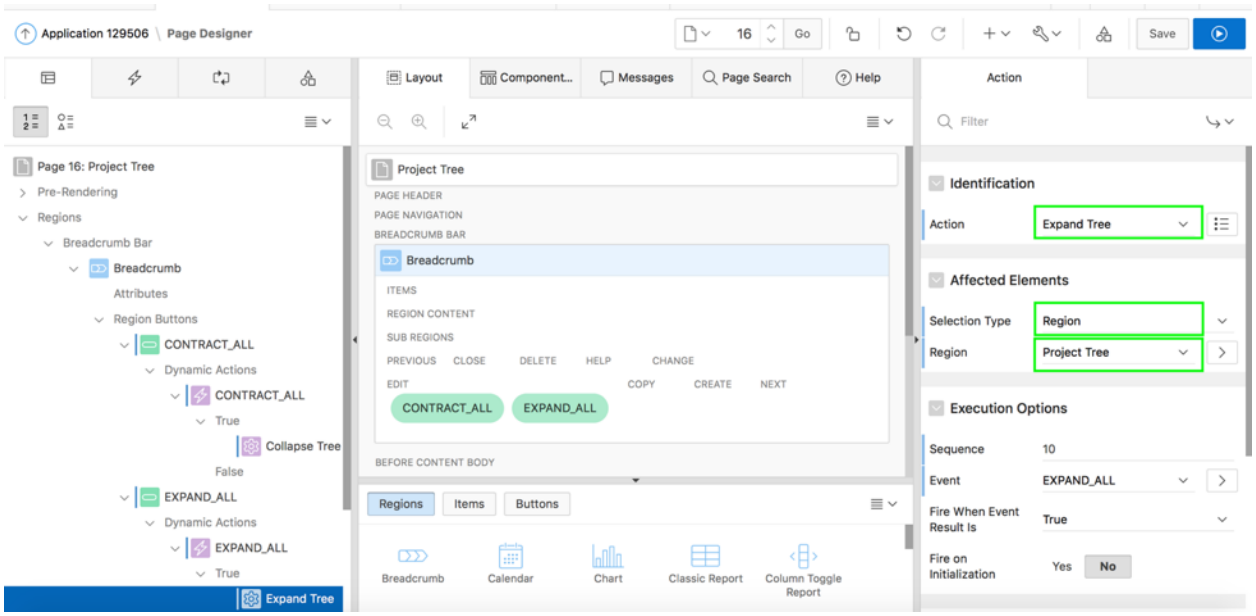
10. Under Rendering, navigate to the CONTRACT_ALL region button and expand Dynamic Actions. Select **Show**.
    In the property editor, for Identification > Action, select **Collapse Tree**.
    For Affected Elements > Selection Type, select **Region**.
    For Affected Elements > Region, select **Project Tree**.



11. Under Rendering, expand Regions > Breadcrumb > Region Buttons.
    Right-click the **EXPAND_ALL** button and select **Create Dynamic Action**.

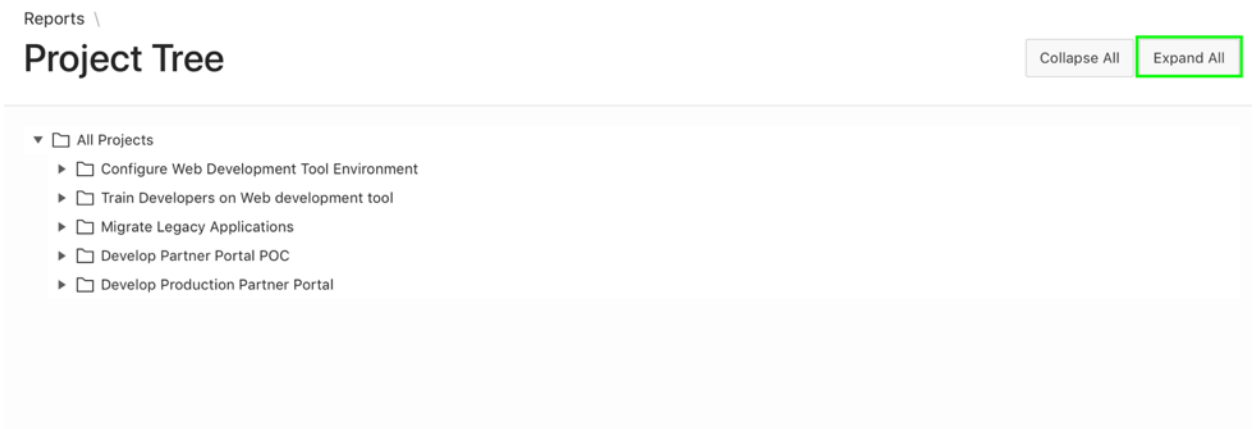
12. In the property editor, for Identification > Name, enter **EXPAND_ALL**.

13. Under Rendering, navigate to the EXPAND_ALL region button and expand Dynamic Actions. Select **Show**.
    In the property editor, for Identification > Action, select **Expand Tree**.
    For Affected Elements > Selection Type, select **Region**.
    For Affected Elements > Region, select **Project Tree**.

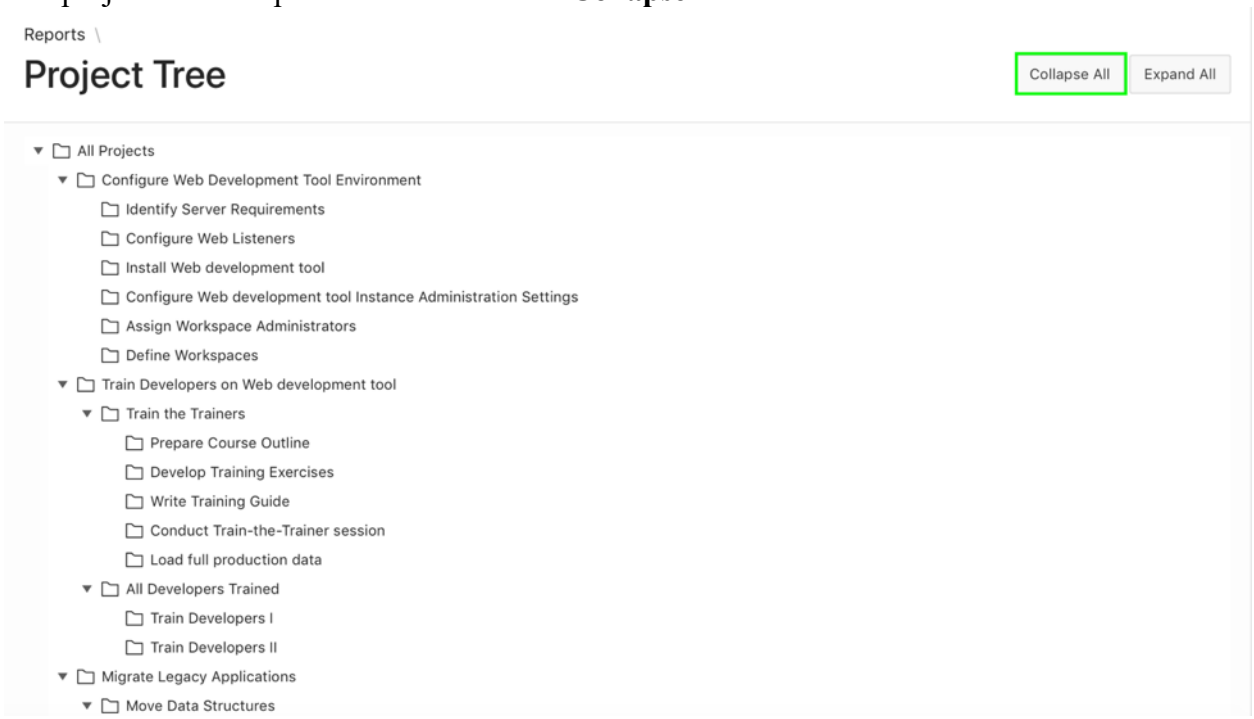


14. Click **Save and Run Page**.

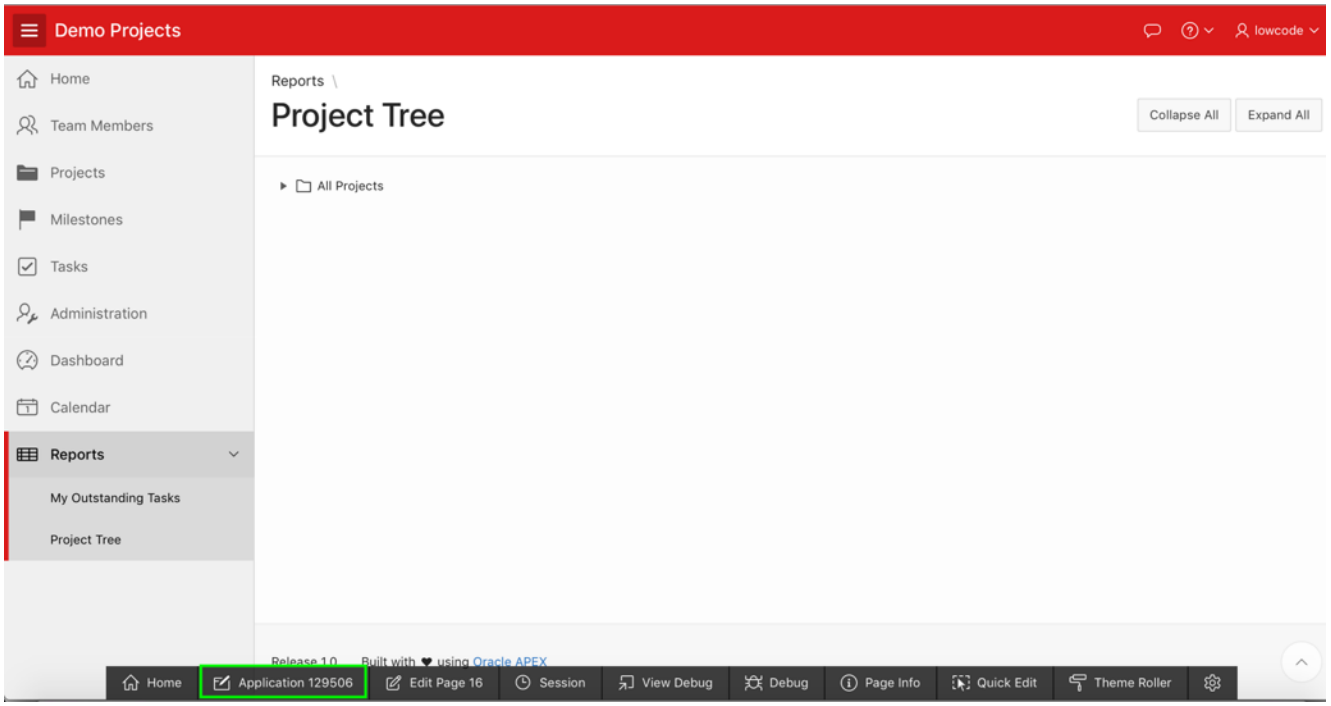
15. The project tree is now displayed in collapsed mode. Click the **Expand All** button.



16. The project tree is expanded now. Click the **Collapse All** button.



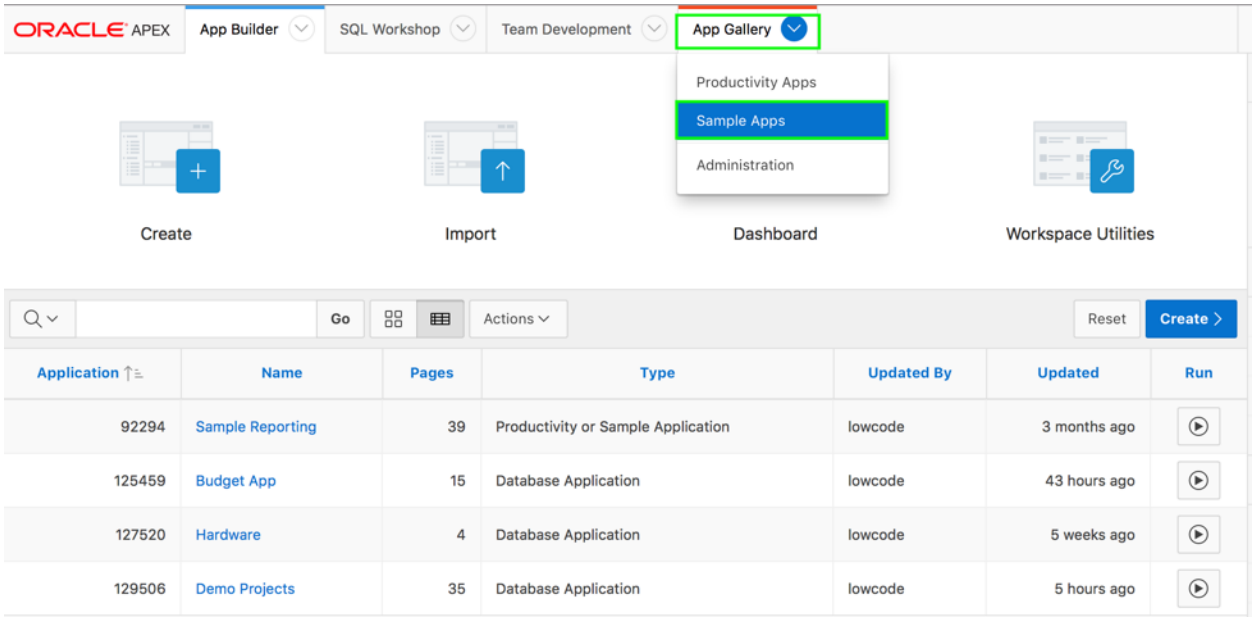
17. In the Developer Toolbar, click **Application<*n*>**.



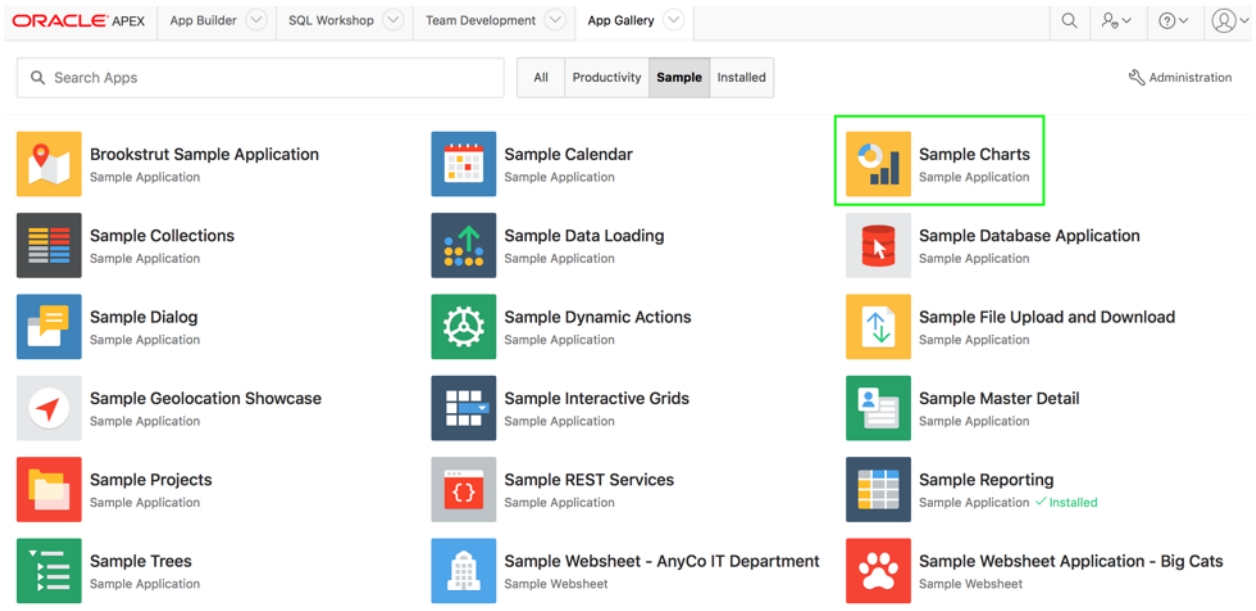
HOL 15-4: Creating and Using a Plug-in

In this hands-on lab, you create the Badge List Plug-in as a copy of the plug-in from the Sample Charts application. On the Reports page, you create a badge list region. On this region, you display three different badges: Open Projects, Upcoming Milestones, and Open Tasks.

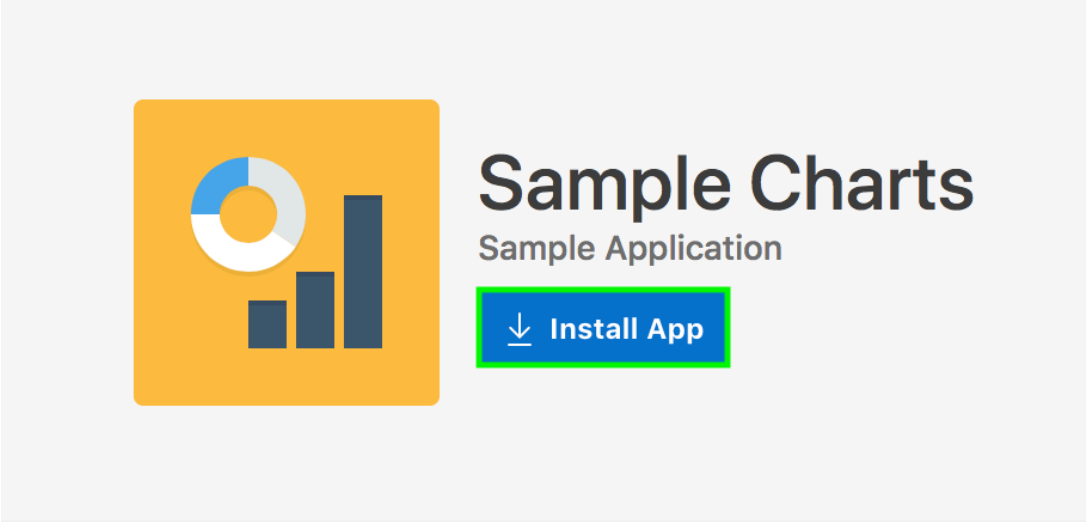
1. Install the Sample Charts application so that you can subscribe to the badge list plug-in. Navigate to App Builder and click **App Gallery > Sample Apps**.



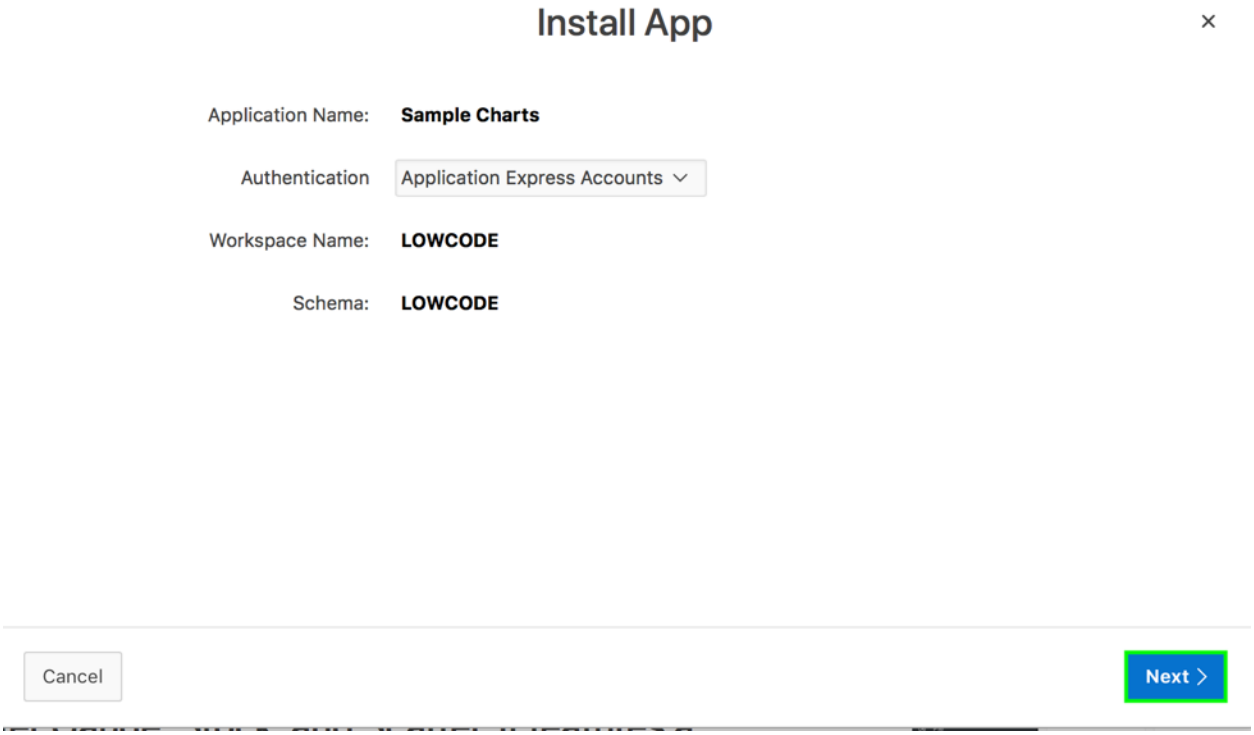
2. Click **Sample Charts**.



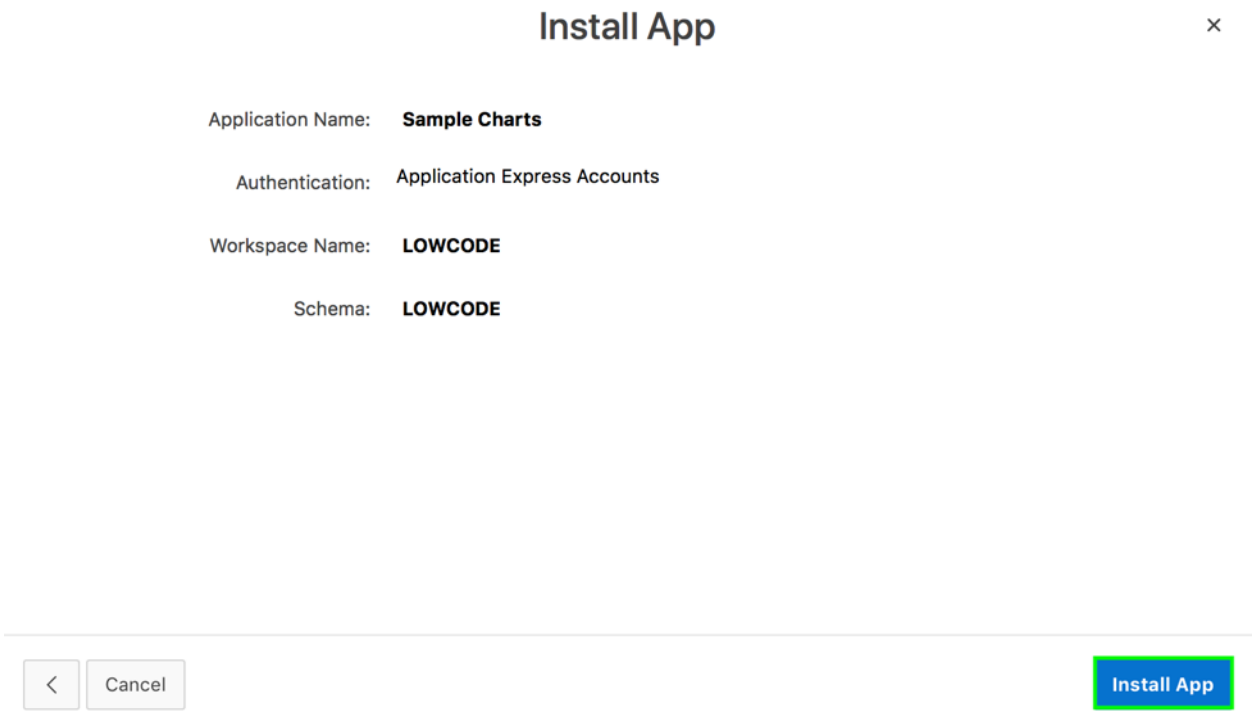
3. Click **Install App**.



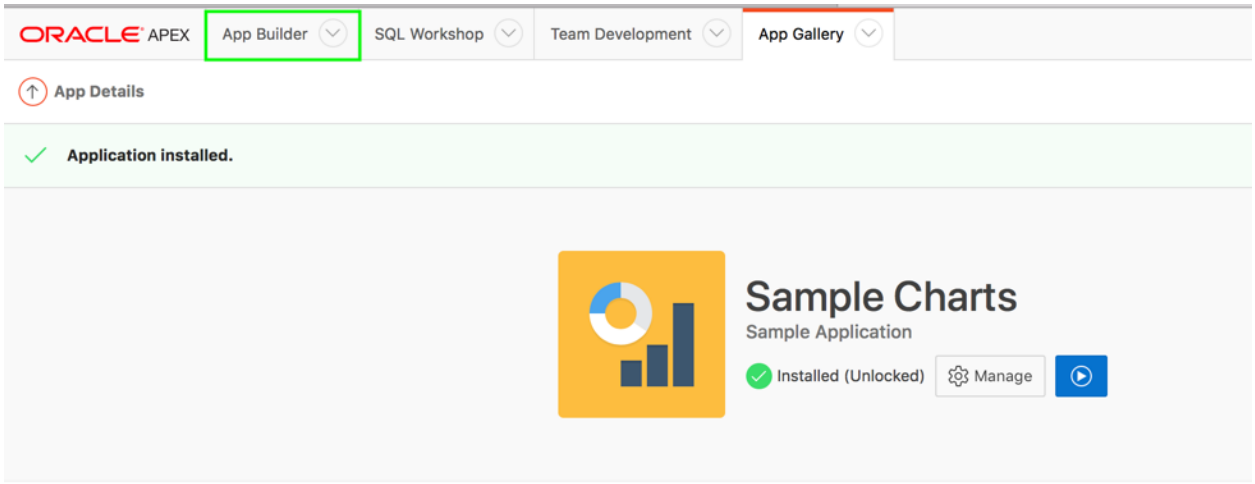
4. Accept the default authentication and click **Next**.



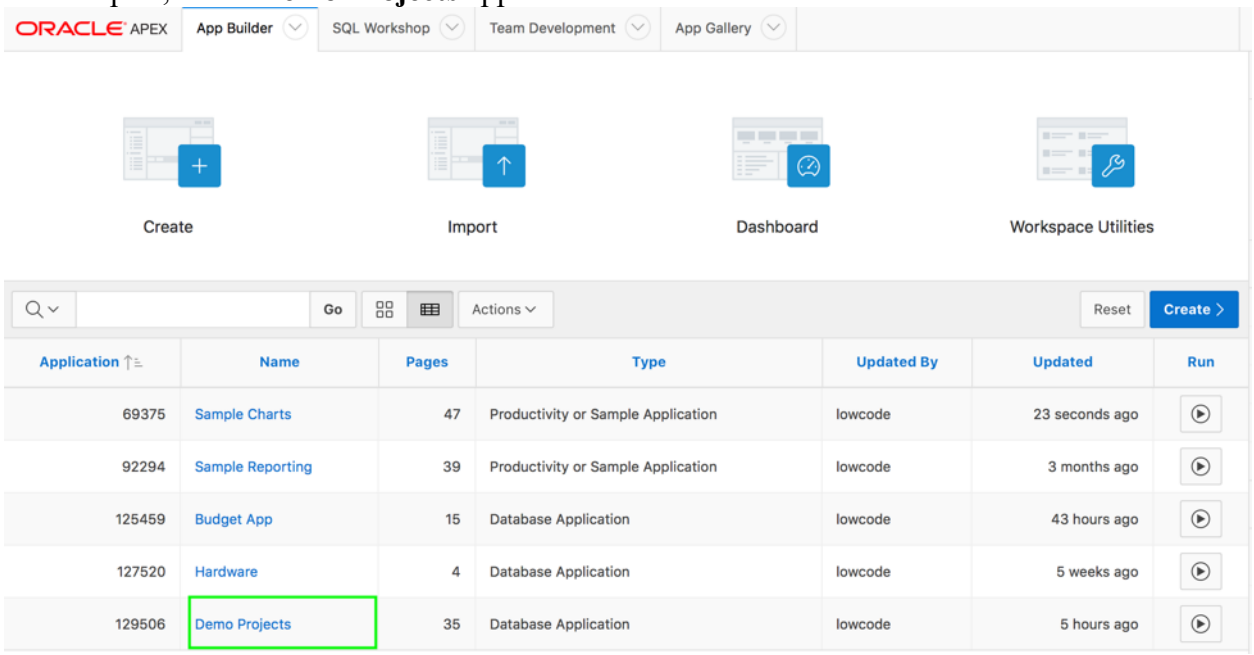
5. Click **Install App**.



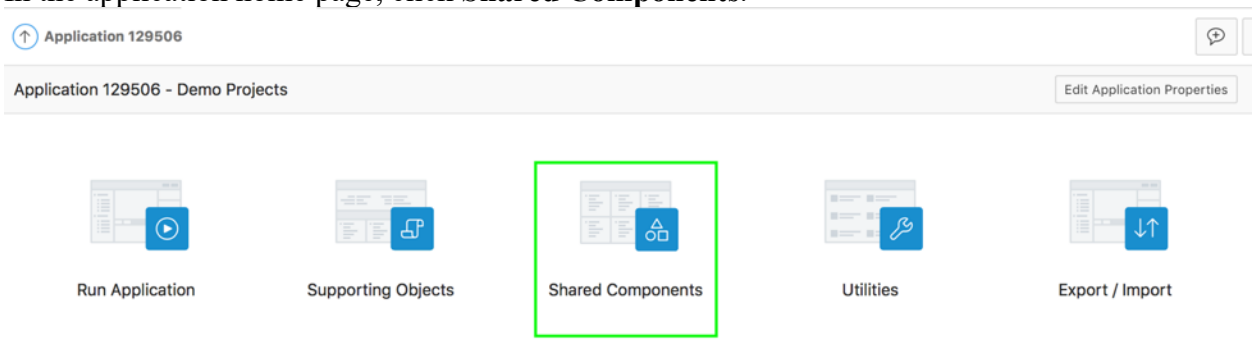
6. The Sample Charts application is now installed.
Click **App Builder**.



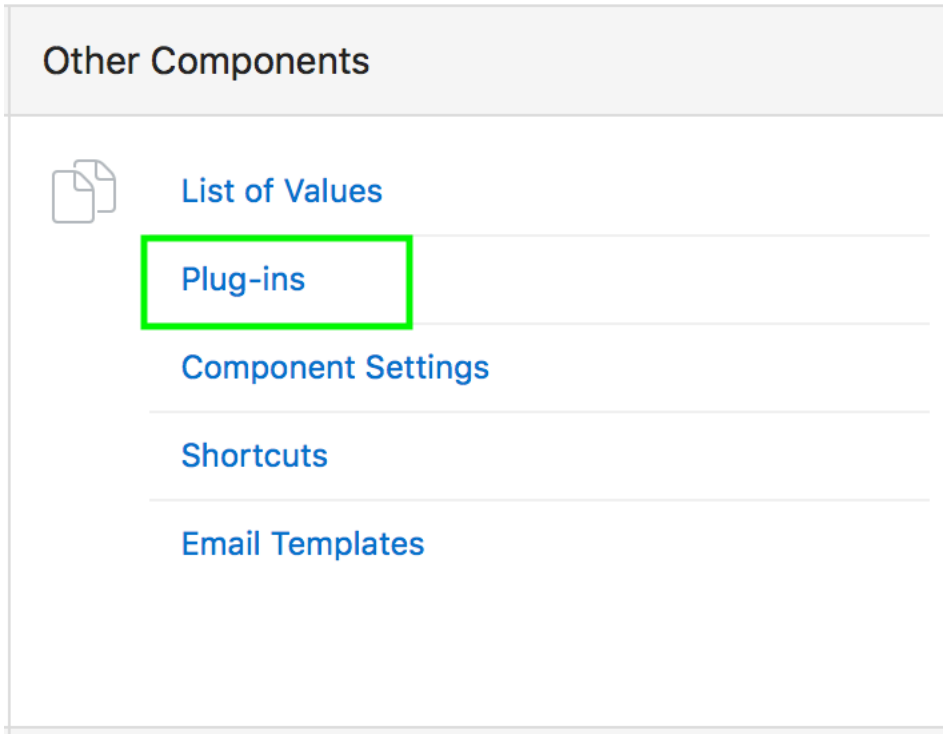
7. In the report, select **Demo Projects** application.



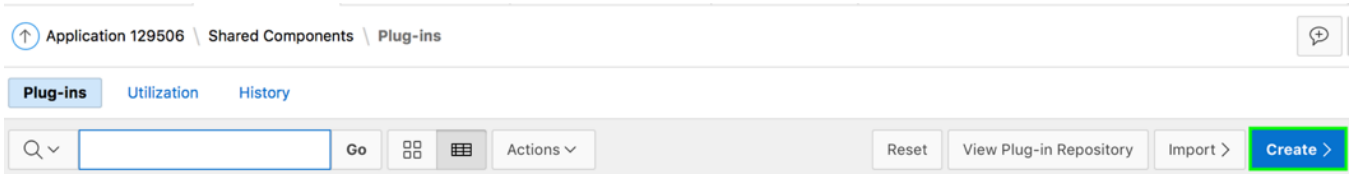
8. In the application home page, click **Shared Components**.



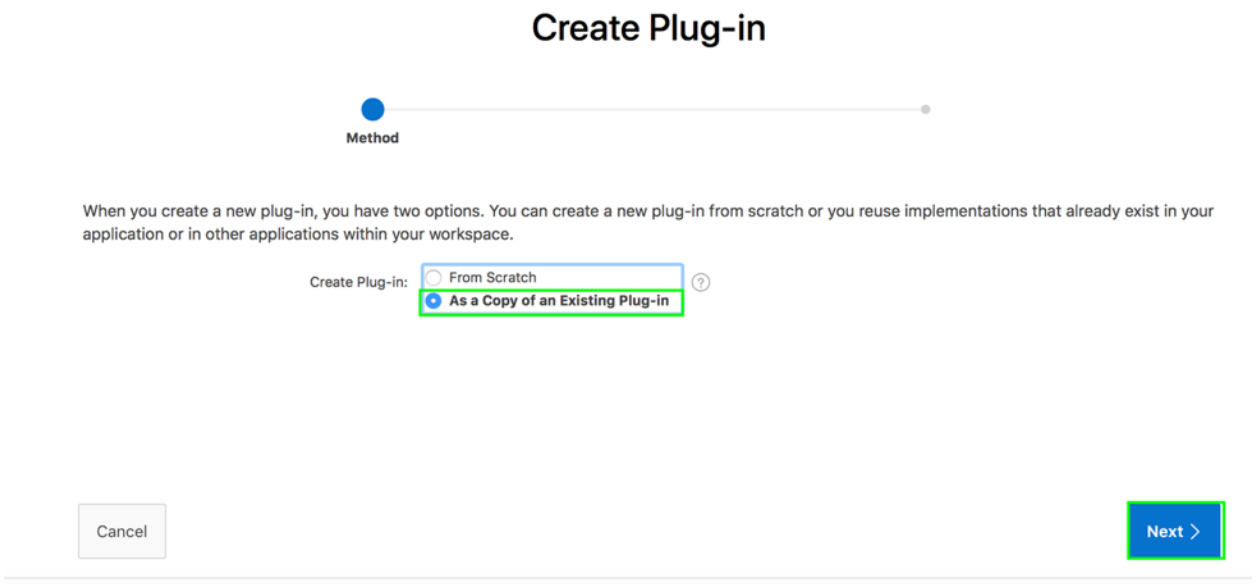
9. Under Other Components, click **Plug-ins**.



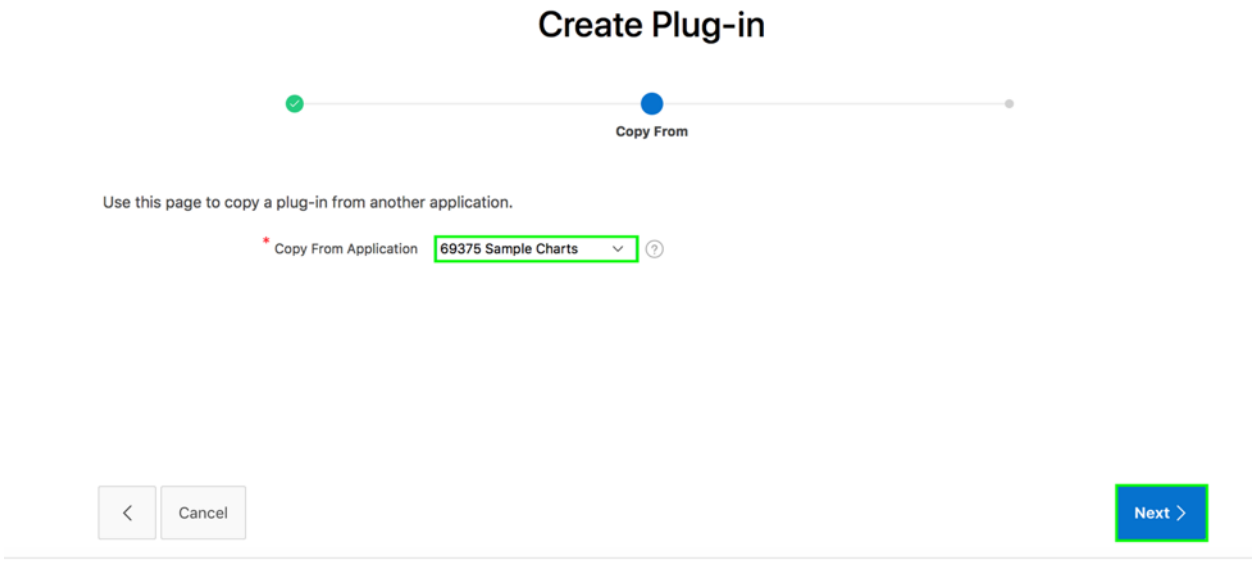
10. In the Plug-ins page, click **Create**.



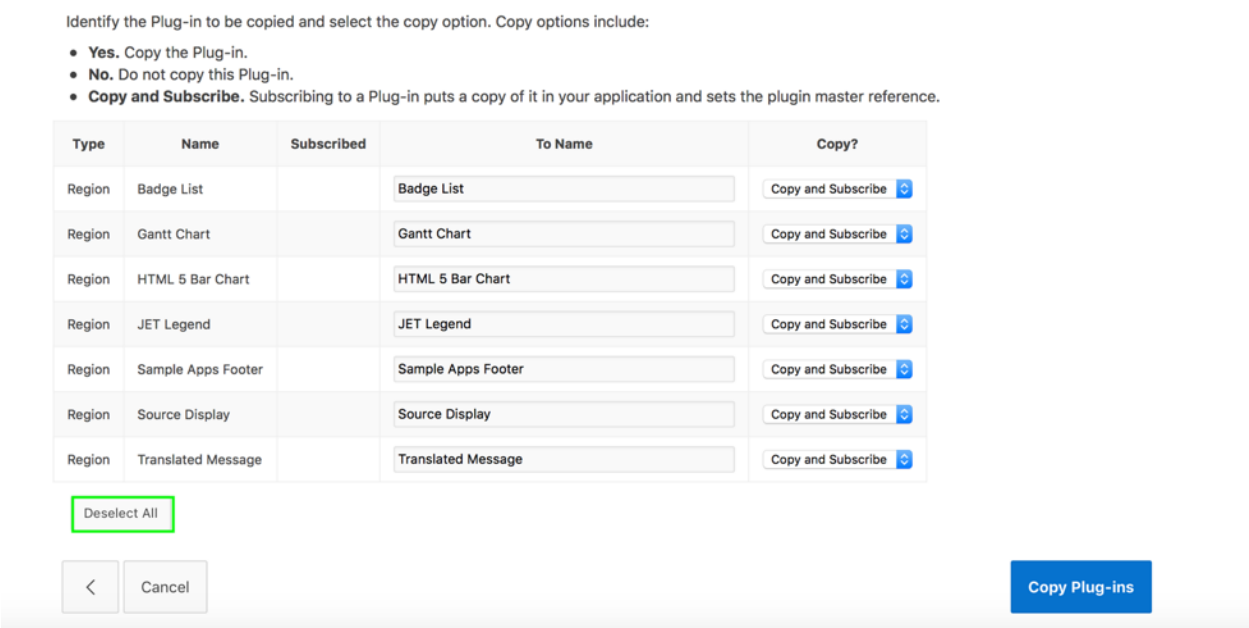
11. For Create Plug-in, select **As a Copy of an Existing Plug-in** and click **Next**.



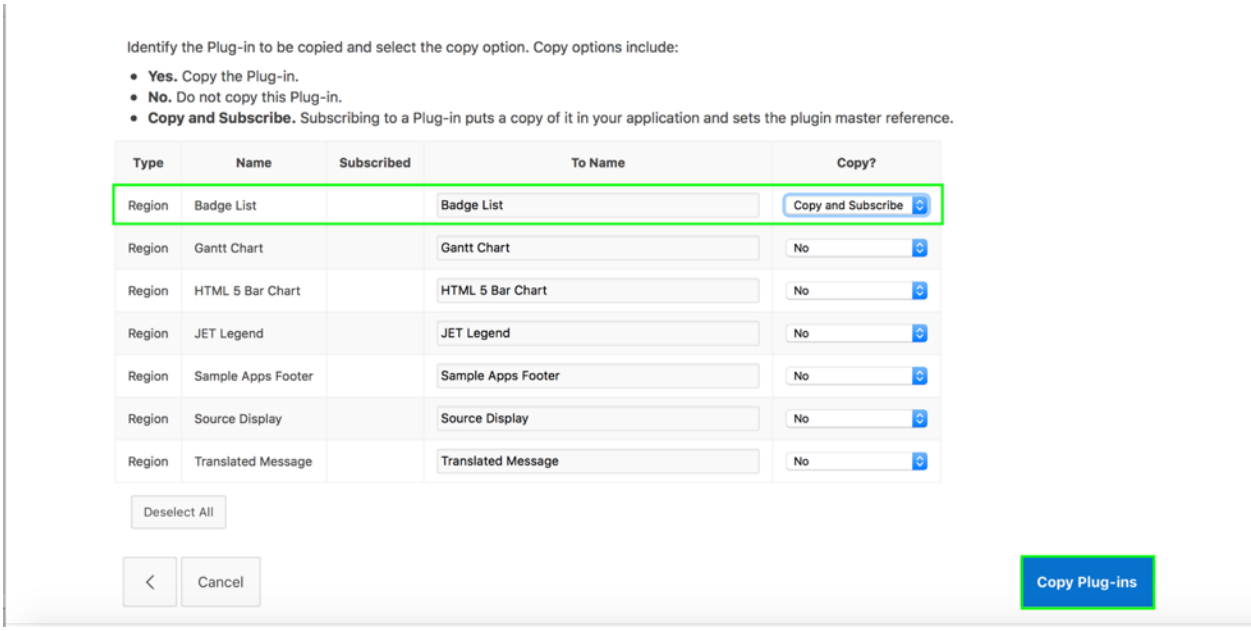
12. For Copy From Application, select <xxx>**Sample Charts** and click **Next**.



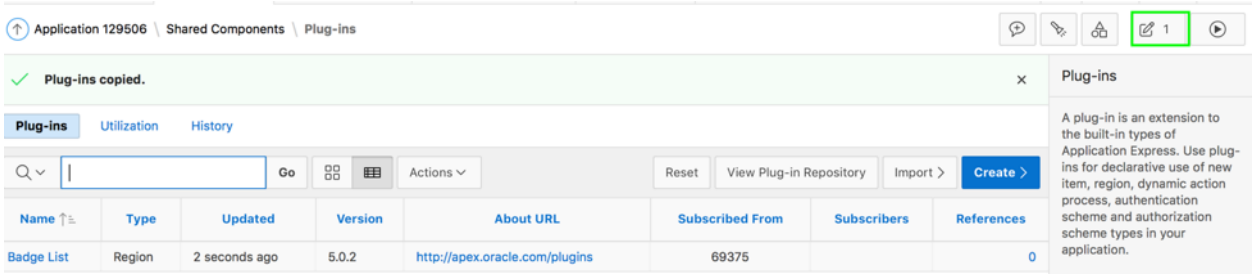
13. At the bottom of the page, click the **Deselect All** button.



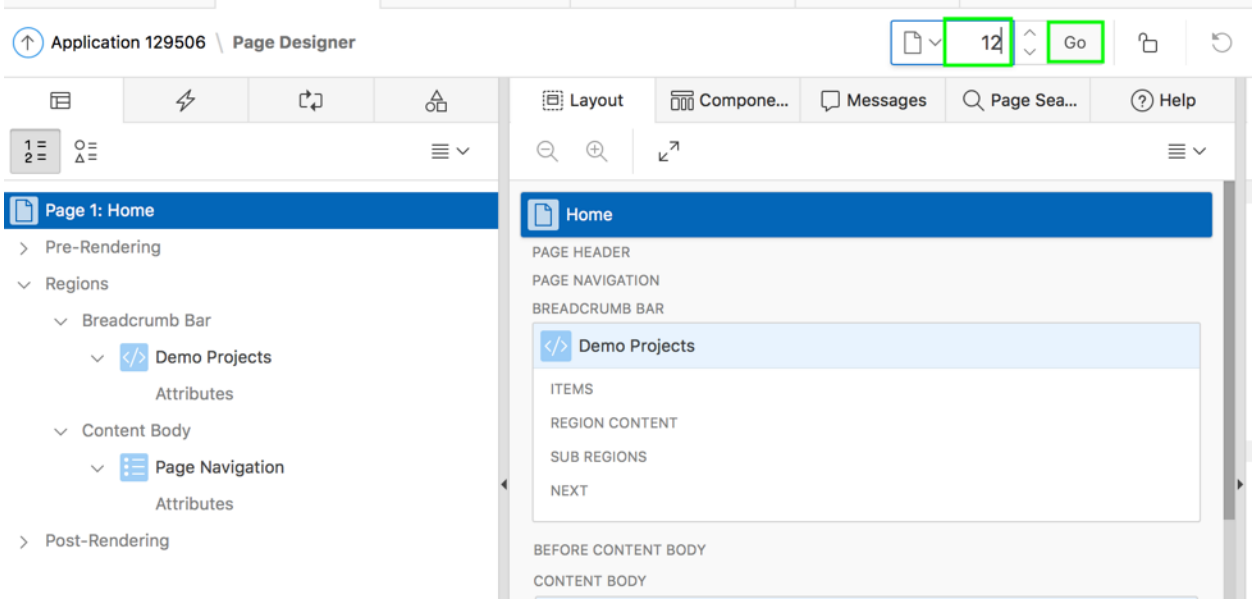
14. Now, for Badge List, select **Copy and Subscribe**.
    Then, click **Copy Plug-ins**.



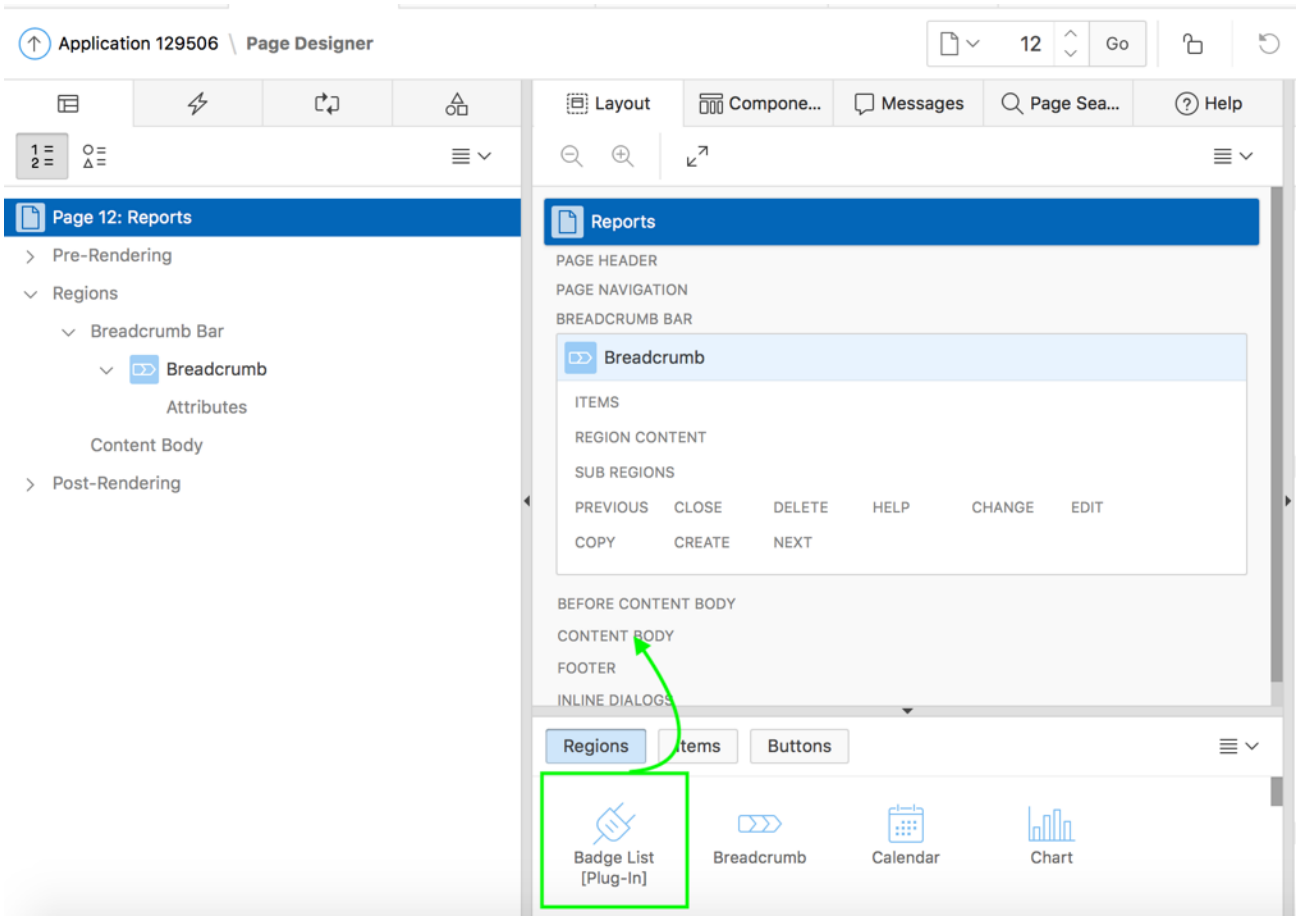
15. In the Application Express toolbar, click **Edit Page 1**.

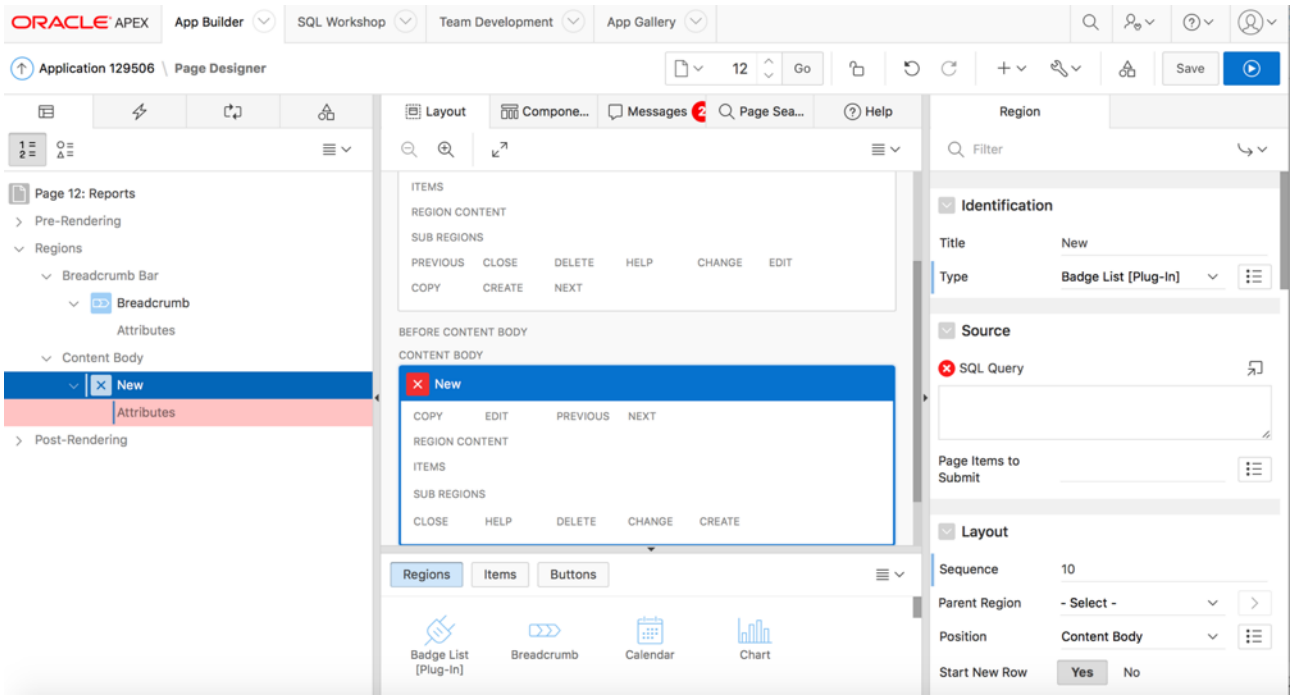


16. In the page finder, enter the Reports page number and click **Go**.



17. Use Page Designer's drag-and-drop functionality to quickly add a Badge List from the Gallery to the Layout.
    In Page Designer, within the Gallery (directly below the Layout), click **Regions**, and locate **Badge List**.
    Click and hold **Badge List [Plug-In]** and drag it into **Content Body**. It should appear as a darkened tile before you drop it into place.
    Note: When you drag the region up, and hover over the small yellow section, below Content Body, the yellow section will expand. A darker yellow section, with a black box around it, will indicate where the region will be placed.

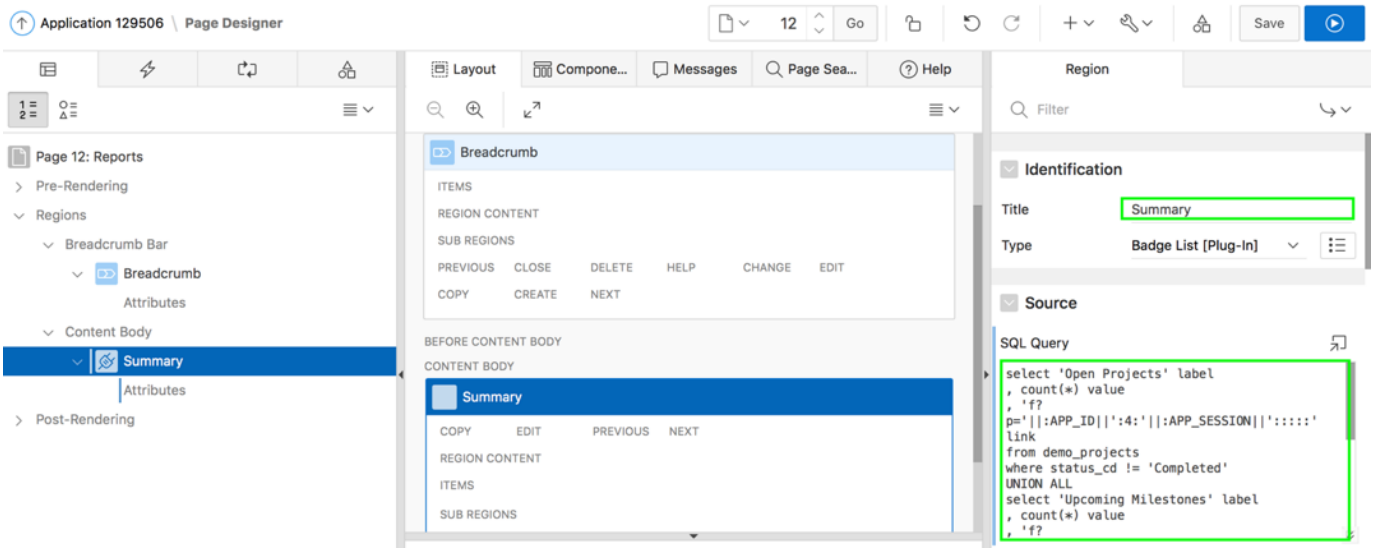

Application 129506 \ Page Designer
12
Go
Layout
Compone...
Messages
Page Sea...
Help
Page 12: Reports
Pre-Rendering
Regions
Breadcrumb Bar
Breadcrumb
Attributes
Content Body
Post-Rendering
Reports
PAGE HEADER
PAGE NAVIGATION
BREADCRUMB BAR
Breadcrumb
ITEMS
REGION CONTENT
SUB REGIONS
PREVIOUS CLOSE DELETE HELP CHANGE EDIT
COPY CREATE NEXT
BEFORE CONTENT BODY
CONTENT BODY
FOOTER
INLINE DIALOGS
Regions
Items
Buttons
Badge List [Plug-In]
Breadcrumb
Calendar
Chart


ORACLE APEX
App Builder
SQL Workshop
Team Development
App Gallery
Application 129506 \ Page Designer
12
Go
Save
Page 12: Reports
Pre-Rendering
Regions
Breadcrumb Bar
Breadcrumb
Attributes
Content Body
New
Attributes
Post-Rendering
ITEMS
REGION CONTENT
SUB REGIONS
PREVIOUS CLOSE DELETE HELP CHANGE EDIT
COPY CREATE NEXT
BEFORE CONTENT BODY
CONTENT BODY
New
COPY EDIT PREVIOUS NEXT
REGION CONTENT
ITEMS
SUB REGIONS
CLOSE HELP DELETE CHANGE CREATE
Regions
Items
Buttons
Badge List [Plug-In]
Breadcrumb
Calendar
Chart
Region
Filter
Identification
Title
New
Type
Badge List [Plug-In]
Source
SQL Query
Page Items to Submit
Layout
Sequence
10
Parent Region
- Select -
Position
Content Body
Start New Row
Yes
No

18. In the Property Editor:
    - Identification: Title - enter **Summary**
    - Source: SQL Query - copy and paste the following:

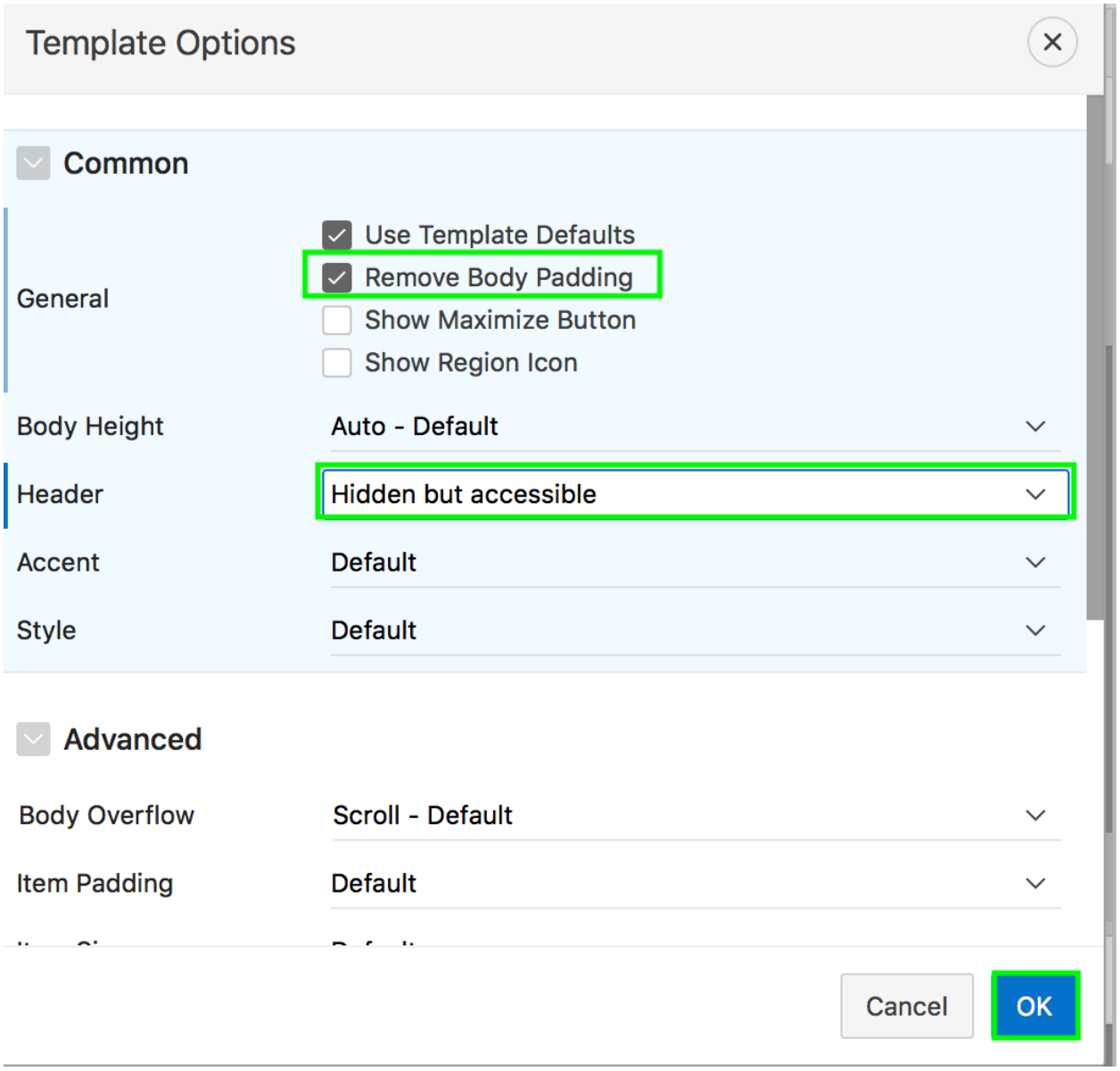
```
select 'Open Projects' label
, count(*) value
, 'f?p='||:APP_ID||':4:'||:APP_SESSION||'::::' link
from demo_projects
where status_cd != 'Completed'
UNION ALL
select 'Upcoming Milestones' label
, count(*) value
, 'f?p='||:APP_ID||':4:'||:APP_SESSION||'::::' link
from demo_proj_milestones
where due_date > sysdate
UNION ALL
select 'Open Tasks' label
, count(*) value
, 'f?p='||:APP_ID||':4:'||:APP_SESSION||'::::' link
from demo_proj_tasks
where is_complete_yn = 'N'
```

    Note: This query has three sub-queries, separated by UNION ALL that will be displayed as three separate badges within the region.



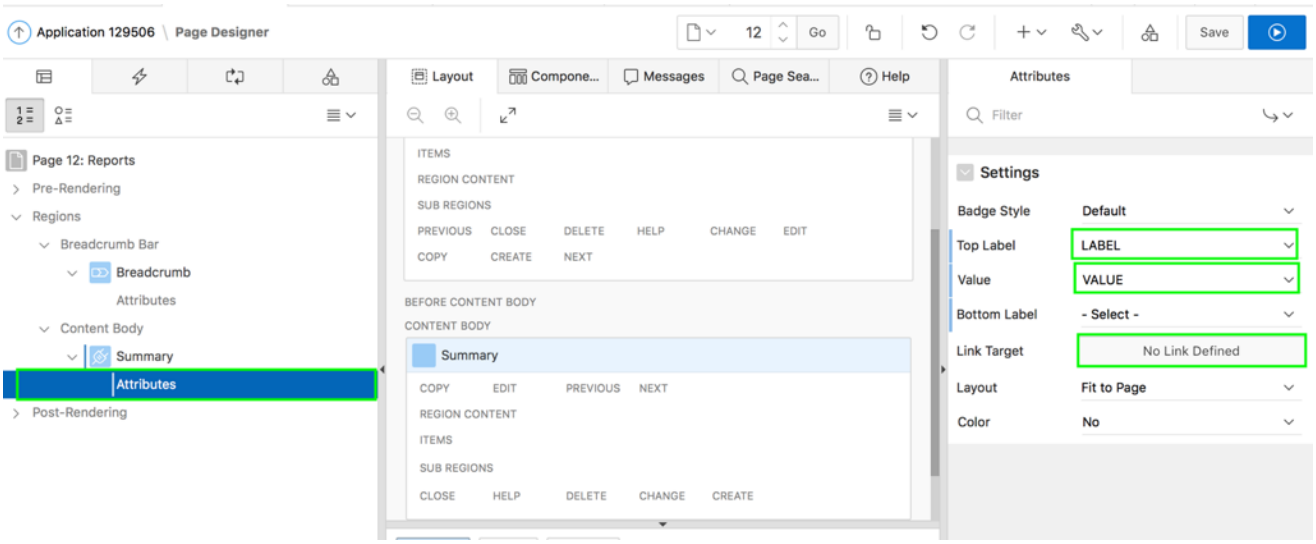
19. In the property editor, navigate to Appearance and click **Template Options**.

20. In the Template Options dialog, select the **Remove Body Padding** check box.
    For Header, select **Hidden but accessible**.
    Click **OK**.

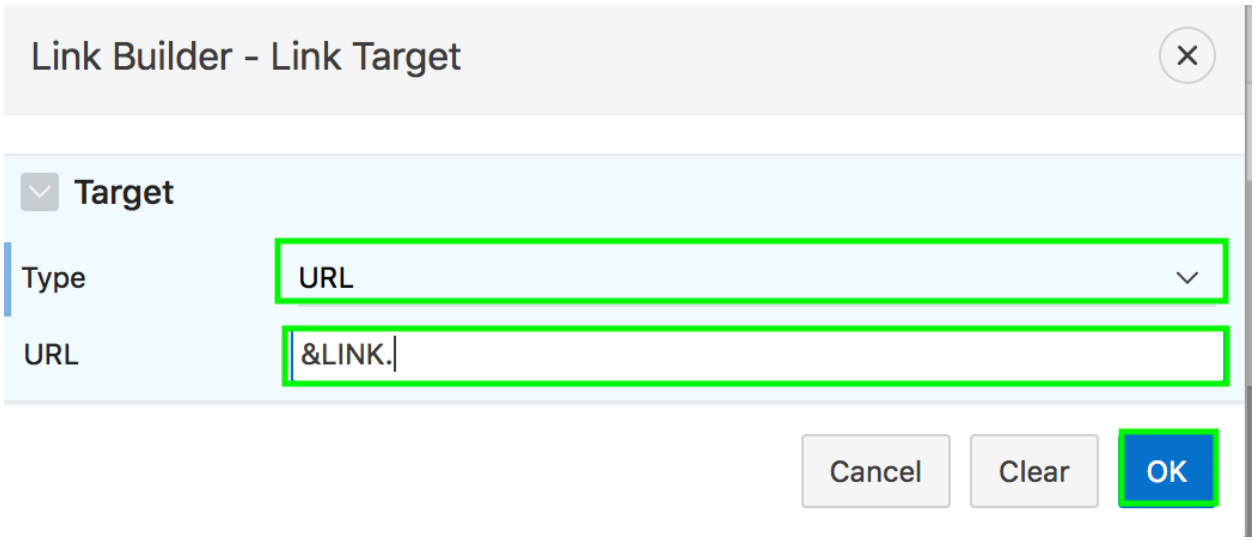


21. In the Rendering tree, under the Summary region, click **Attributes**.
    In the Property Editor:
    - Settings: Top Label - select **LABEL**
    - Settings: Value - select **VALUE**

    For Link Target, click **No Link Defined**.



22. In the Link Builder – Link Target dialog, select URL for Type. For URL, enter **&LINK.**.
Click **OK**.
Note: When you enter **&LINK.** ensure you include the period.



23. Click **Save and Run Page**.

24. In the Developer Toolbar, click **Application<n>.**

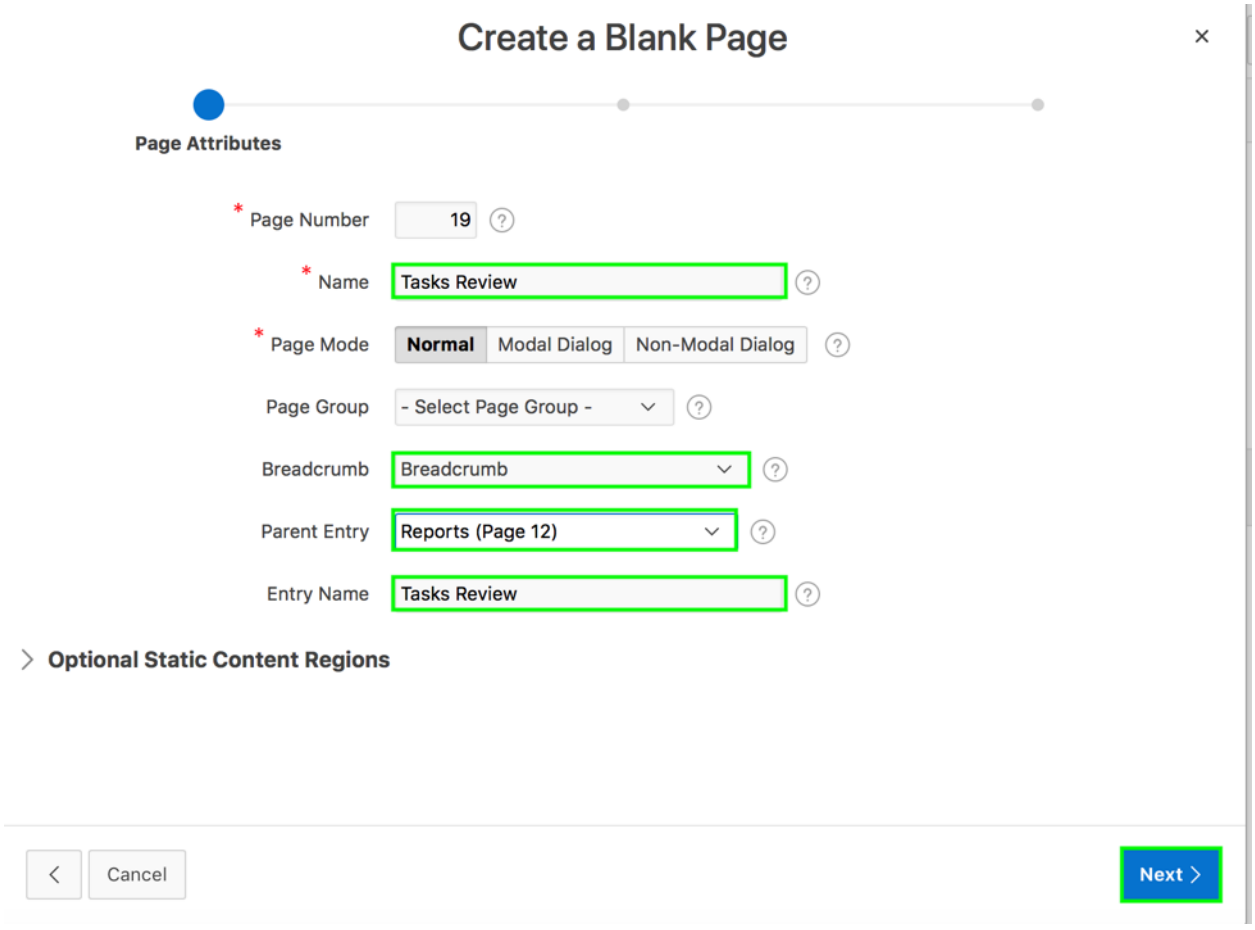
HOL 15-5: Adding Additional Reports

The Dashboard page includes the Tasks Review and Milestones Review chart regions. You want to clean up the Dashboard page by moving these two regions as separate pages. Later, you add in new regions to the Dashboard page. In this lab, you create two new pages as Tasks Review and Milestones Review. You also move the My Outstanding Tasks report as a region on the Reports page.

1. In the application home page, click **Create Page**.

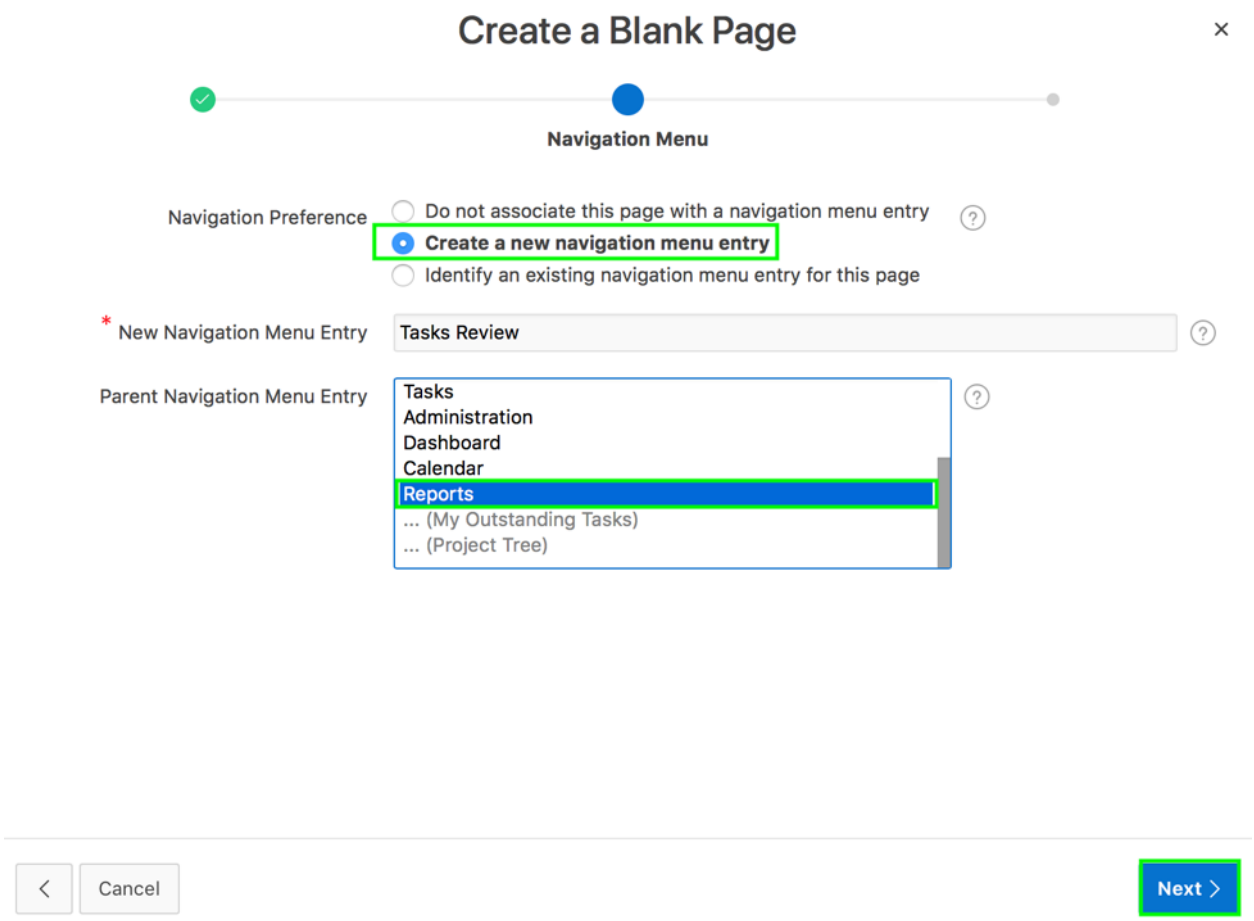
2. Select Blank Page and click Next.

3. Enter / select the following for Page Attributes and click **Next**.
   - Name: **Tasks Review**
   - Breadcrumb: **Breadcrumb**
   - Parent Entry: **Reports**
   - Entry Name: **Tasks Review**



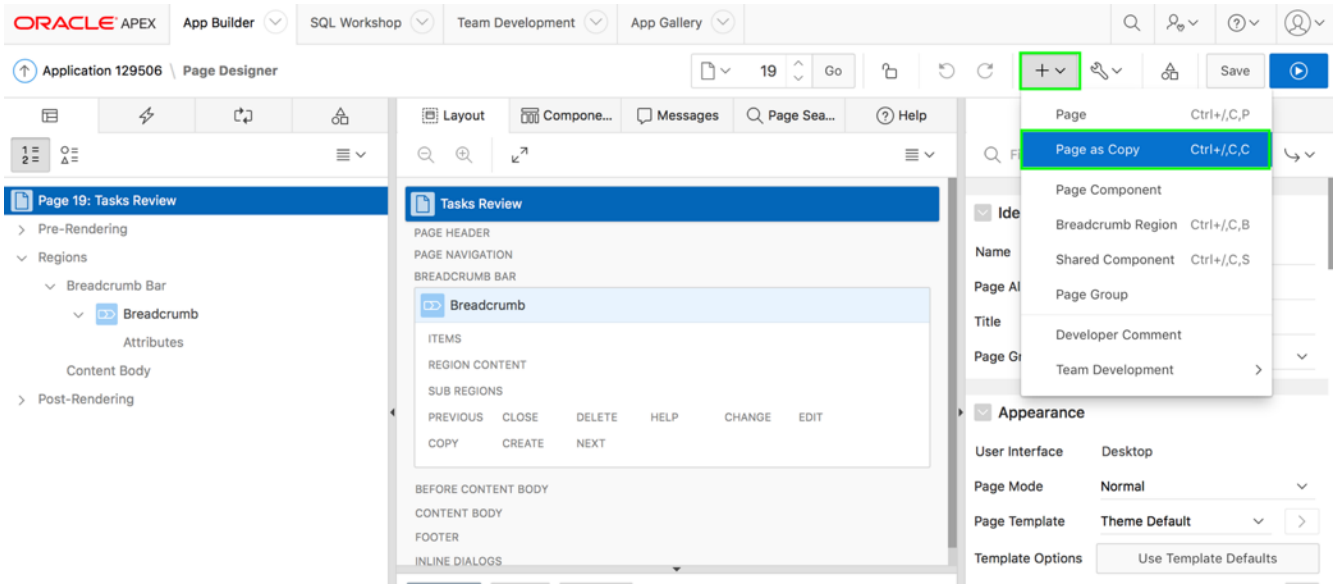
4. Enter / select the following for Navigation Menu and click **Next**.
   - Navigation Preference: **Create a new navigation menu entry**

- Parent Navigation Menu Entry: **Reports**

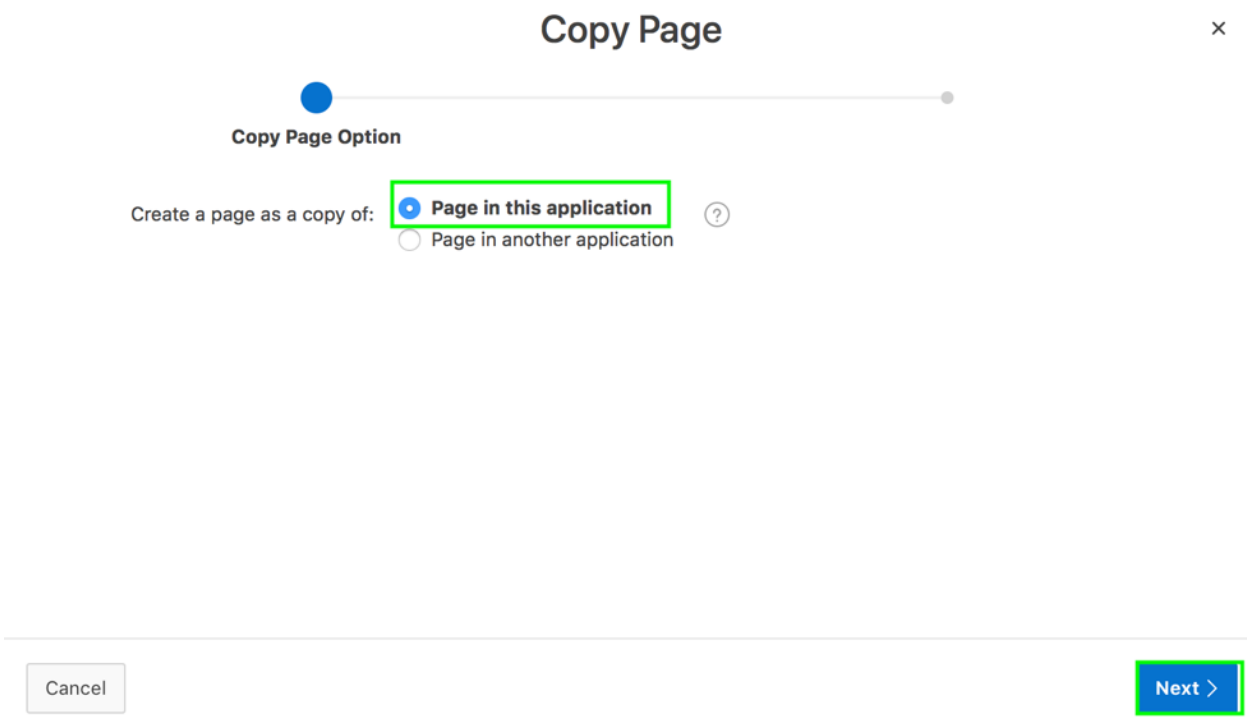


5. On the confirm page, click **Finish**.

6. In the Page Designer, click **Create** and select **Page as Copy**.



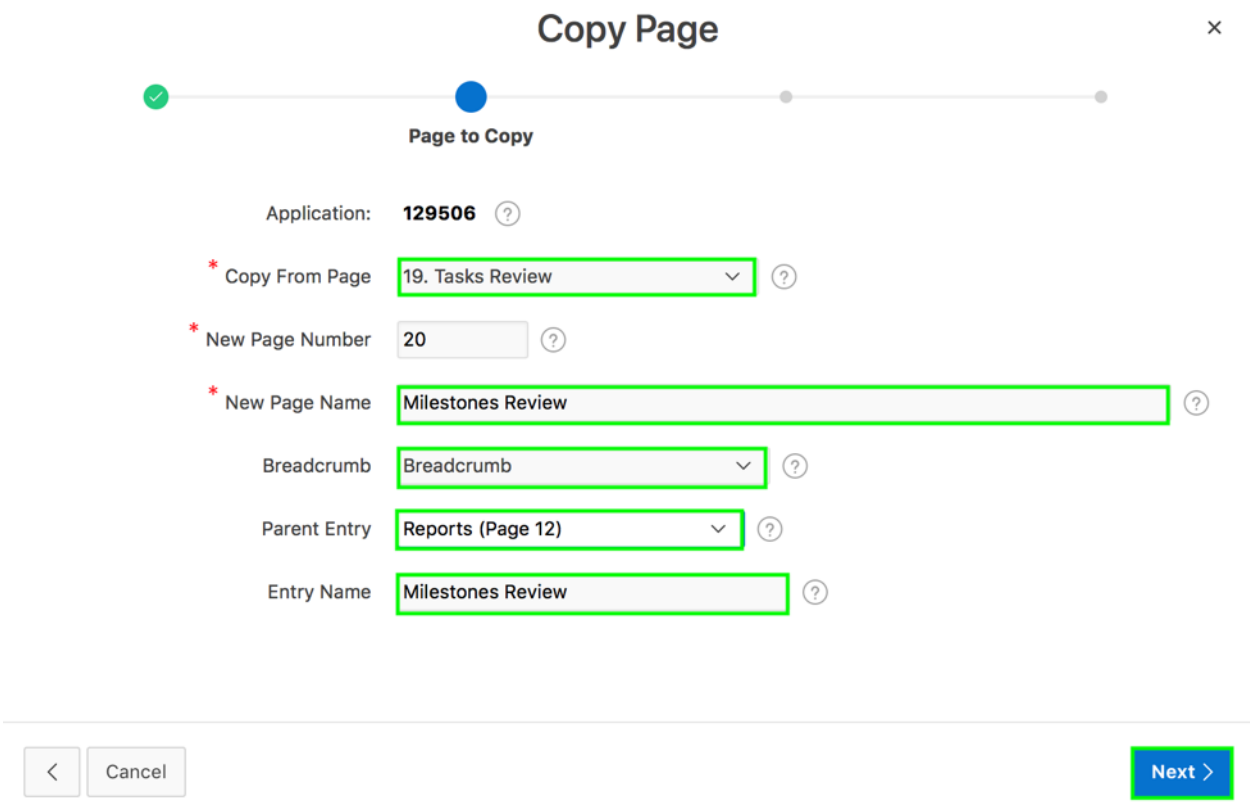
7. For Create a page as a copy of select **Page in this application** and click **Next**.



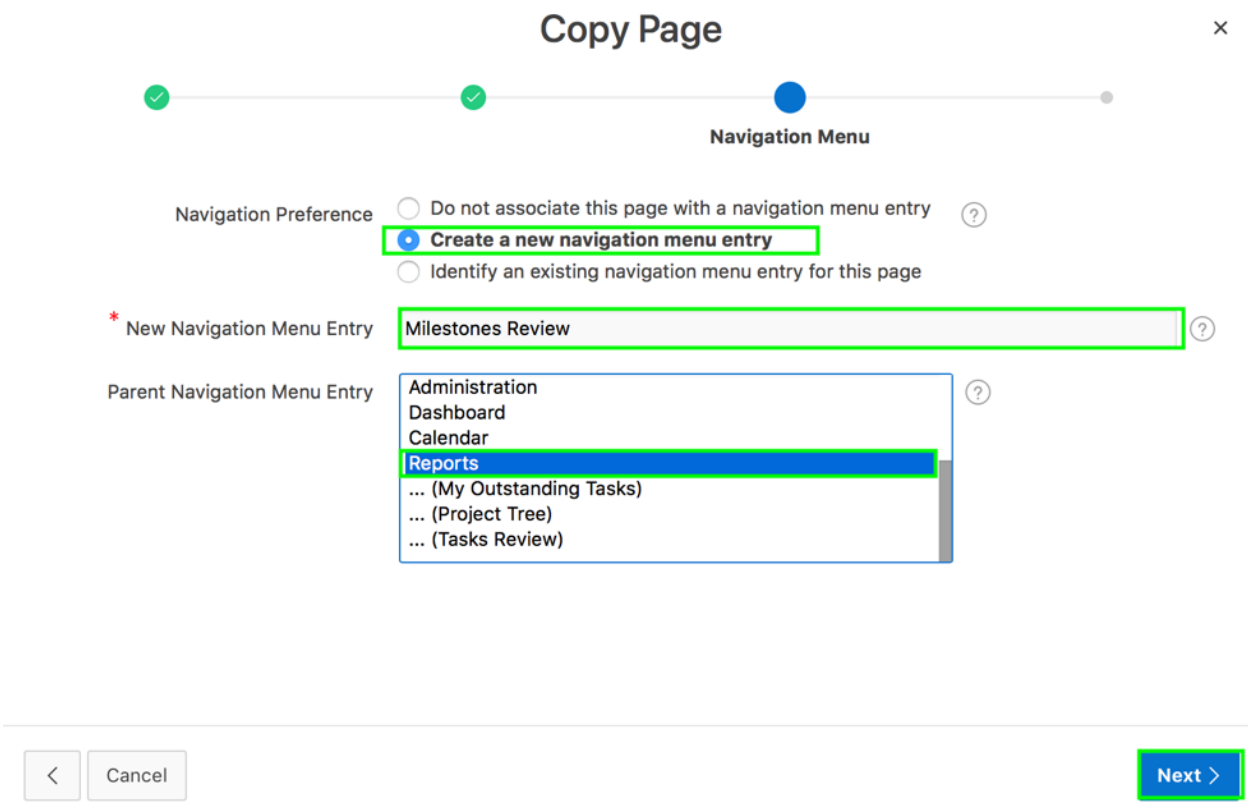
8. Enter / select the following and click **Next**.

- Copy From Page: **<n>. Tasks Review**
- New Page Name: **Milestones Review**
- Breadcrumb: **Breadcrumb**

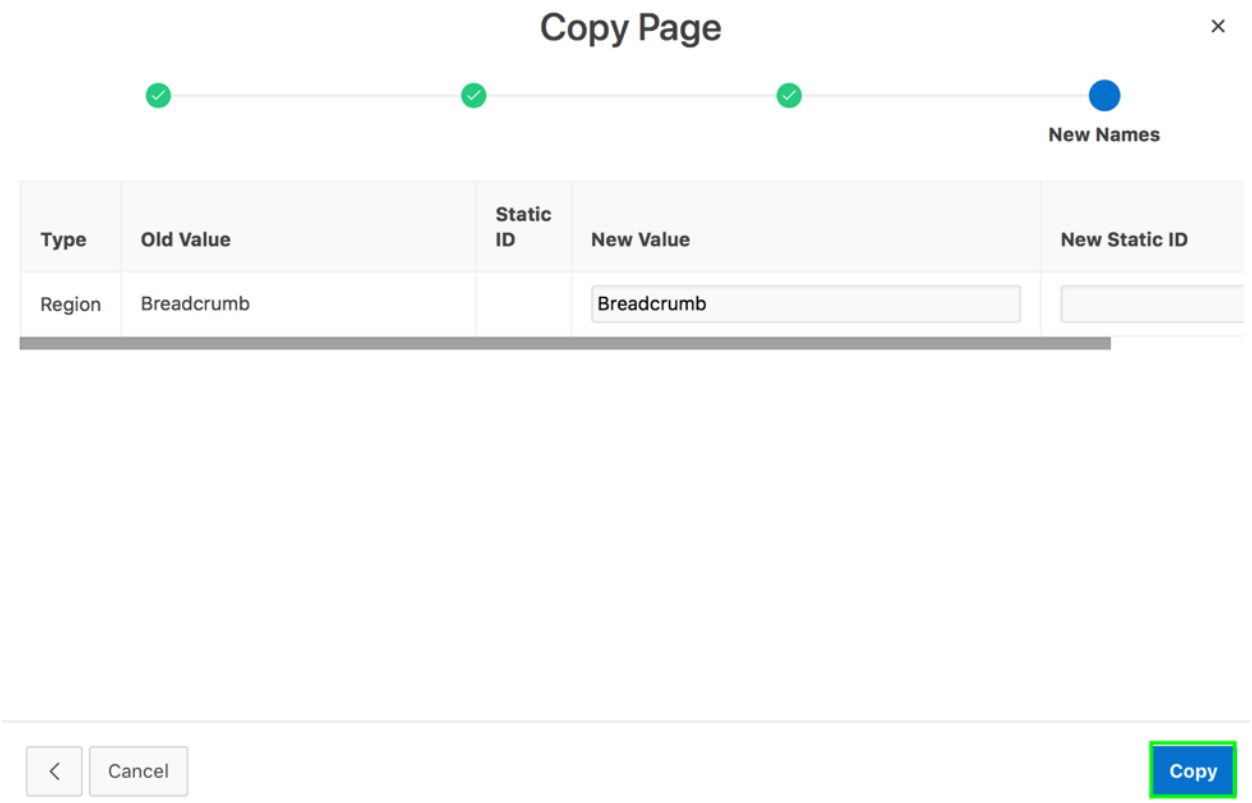
- Parent Entry: **Reports**
- Entry Name: **Milestones Review**



9. For Navigation Menu, enter / select the following and click **Next**.
   - Navigation Preference: **Create a new navigation menu entry**
   - New Navigation Menu Entry: **Milestones Review**
   - Parent Navigation Menu Entry: **Reports**


Copy Page
Navigation Menu
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
New Navigation Menu Entry
Milestones Review
Parent Navigation Menu Entry
Administration
Dashboard
Calendar
Reports
... (My Outstanding Tasks)
... (Project Tree)
... (Tasks Review)
Cancel
Next

10. Click **Copy**.

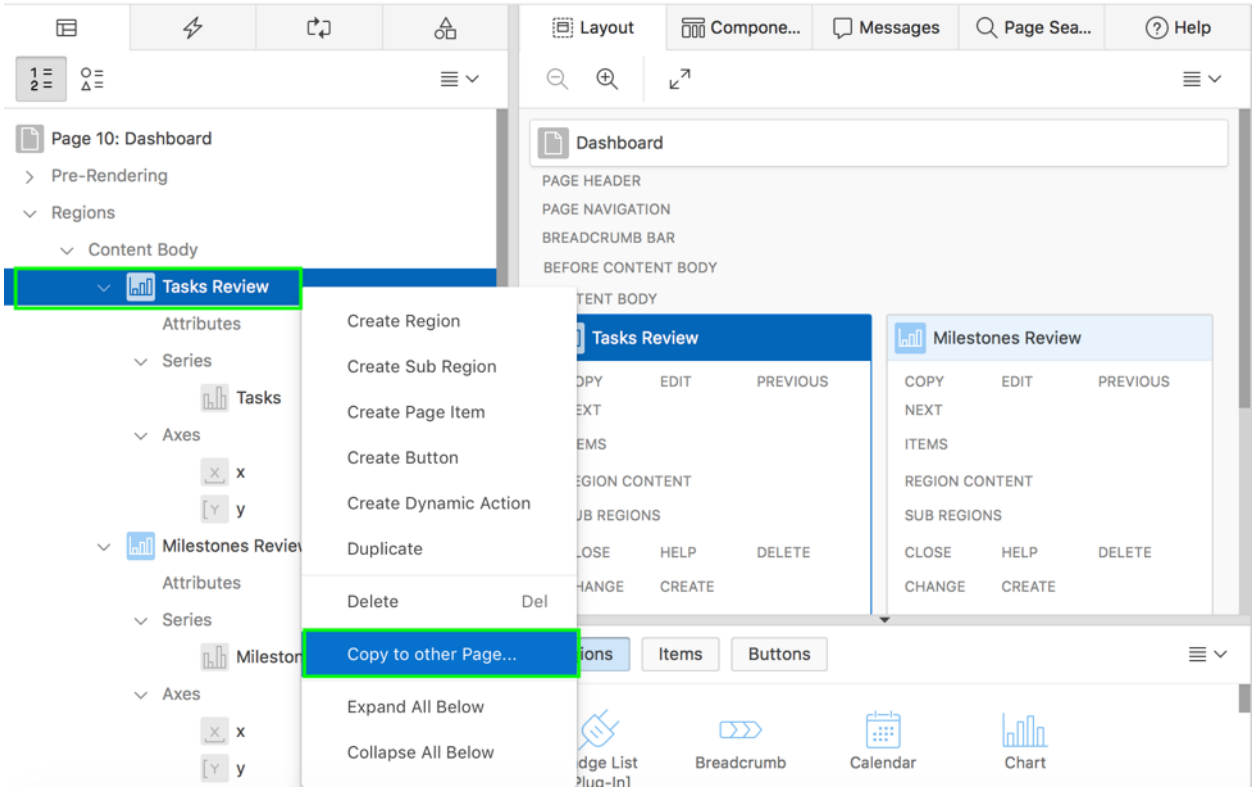


11. Now, navigate to application runtime environment and click **Dashboard** in the navigation menu.

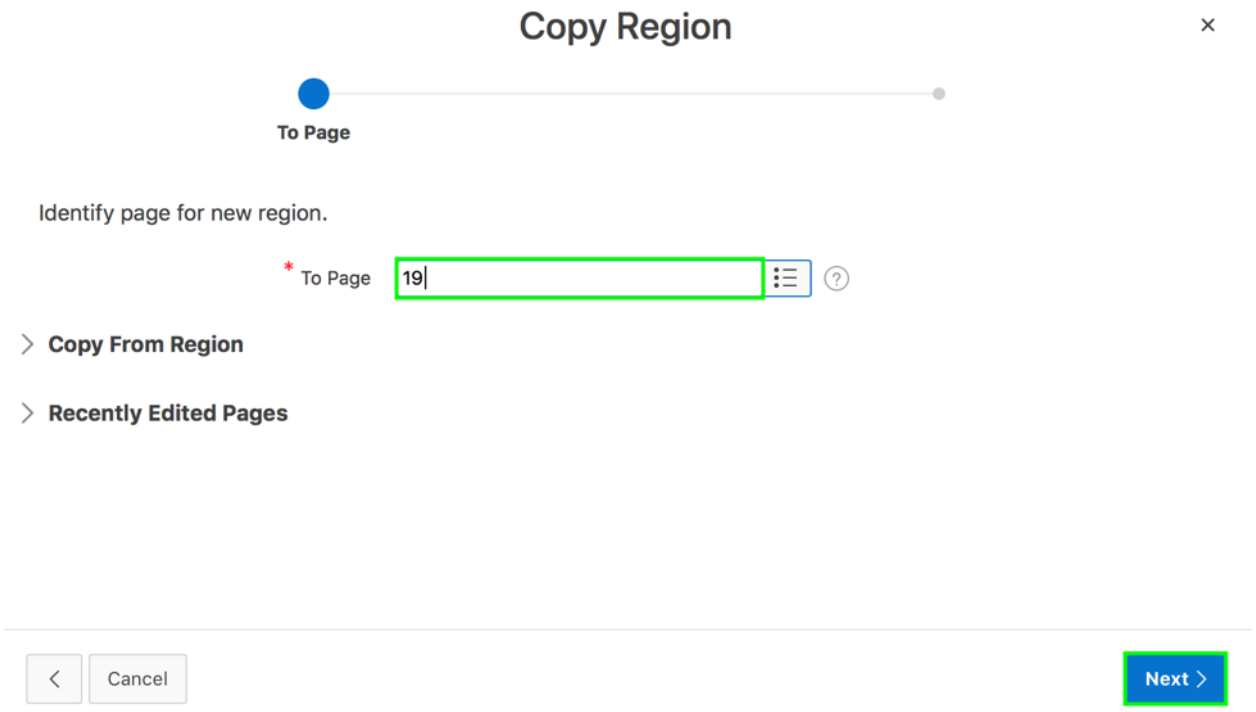
12. In the Developer Toolbar, click **Edit Page 10**.



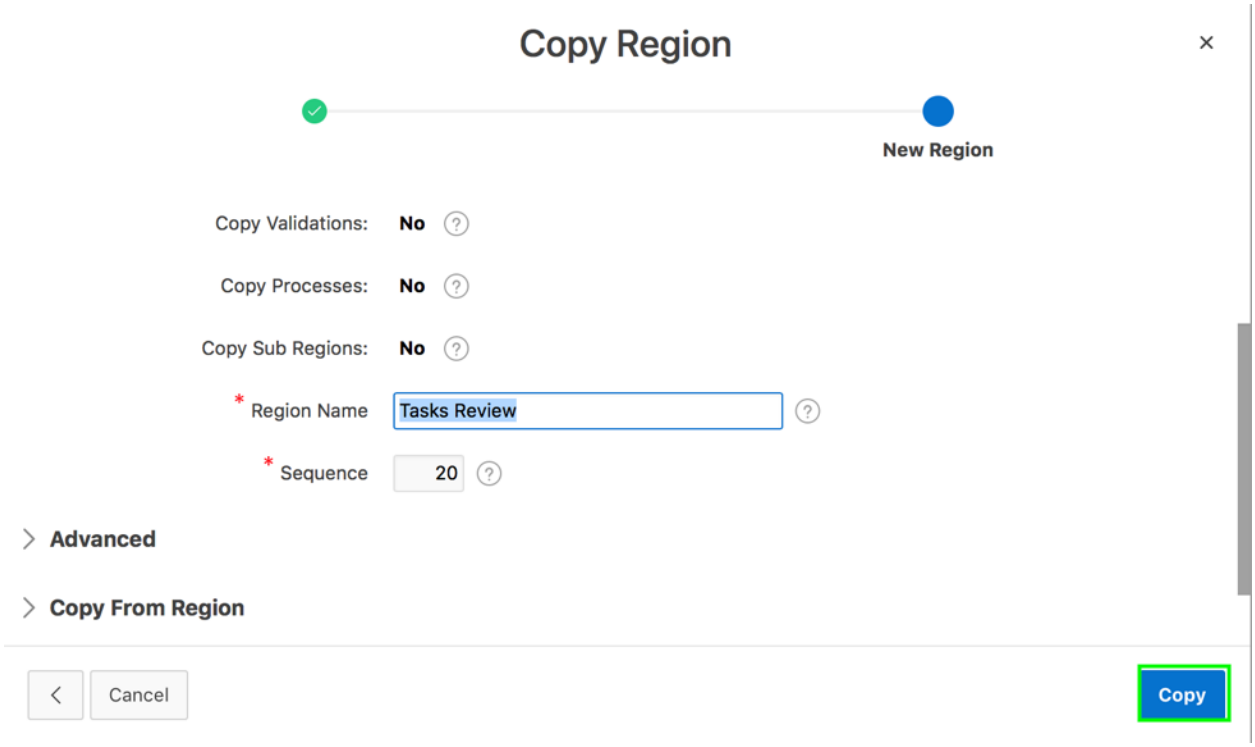
13. In the Page Designer, right-click **Tasks Review** chart region and select **Copy to other page**.



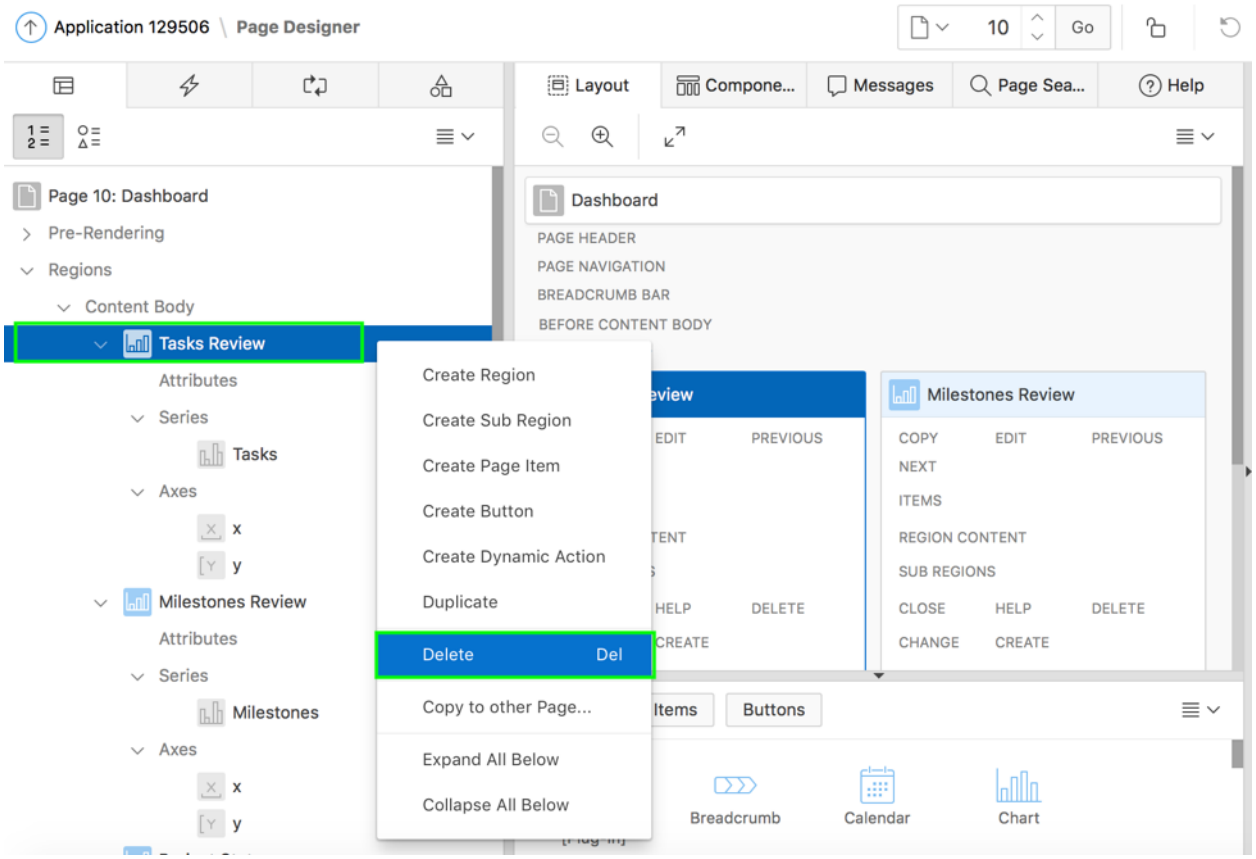
14. For To Page, select the **Tasks Review** page that you created earlier in this lab. You can use the Search dialog to locate the page.
    Click **Next**.



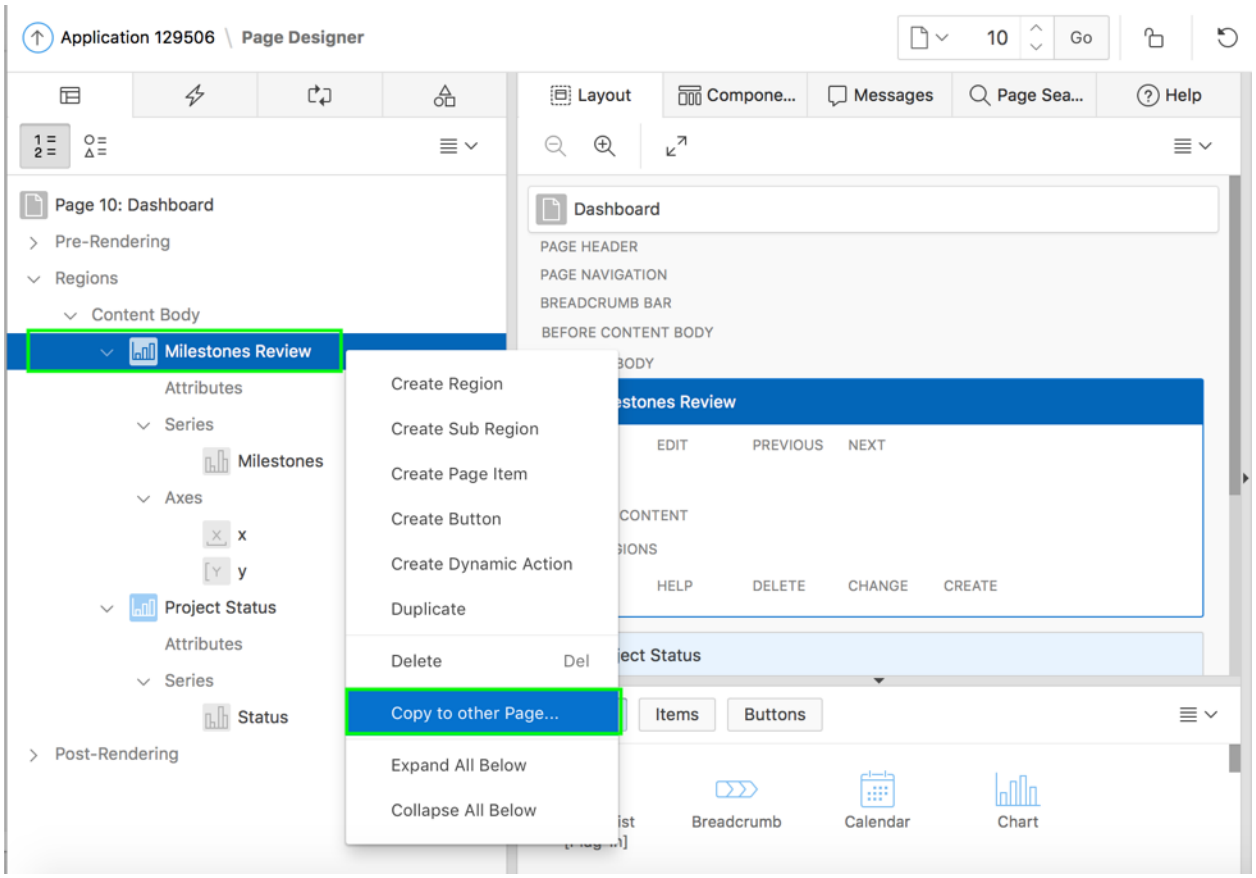
15. Accept the defaults and click **Copy**.



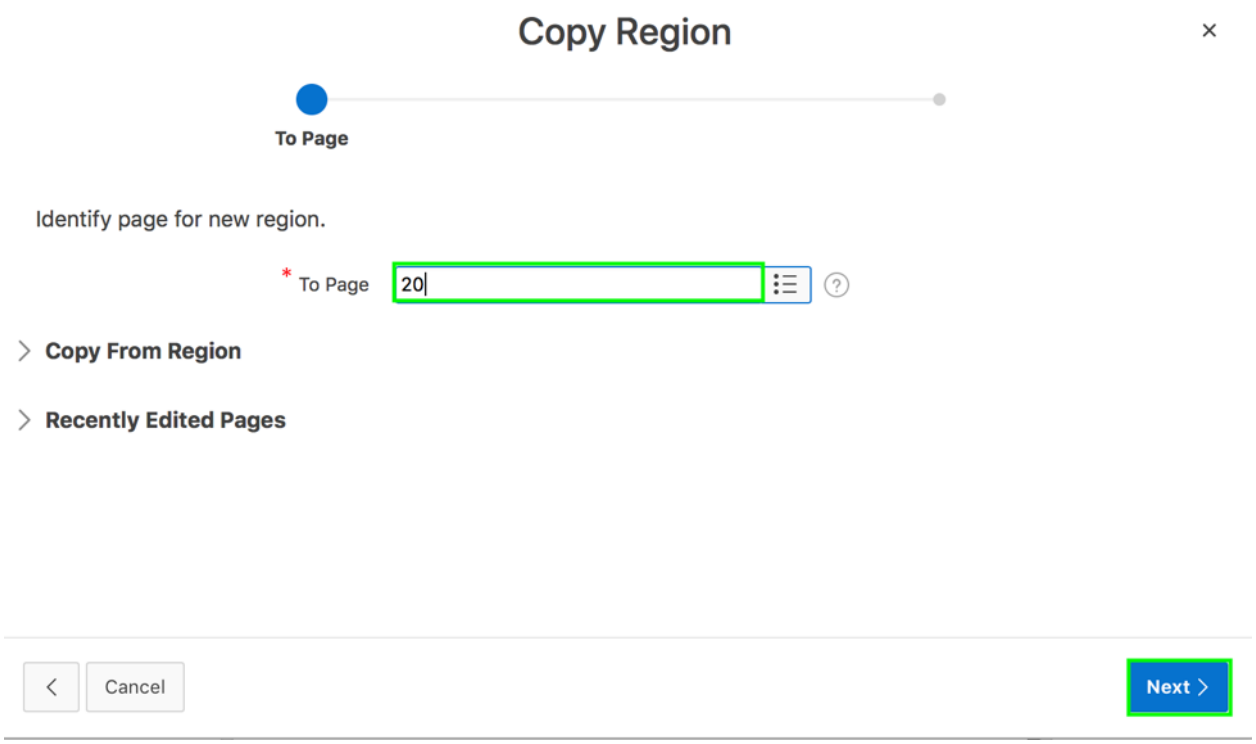
16. Now, under Rendering, right-click **Tasks Review** chart region and select **Delete**.



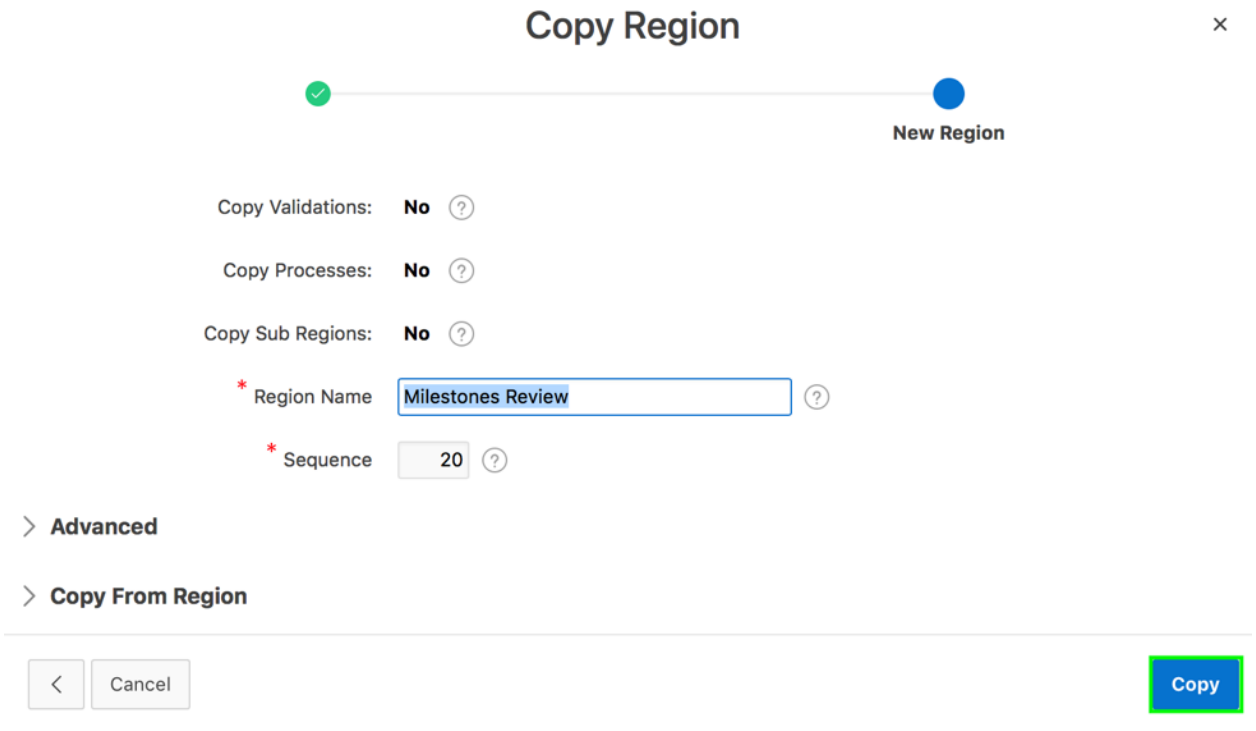
17. In the Page Designer, right-click **Milestones Review** chart region and select **Copy to other page**.



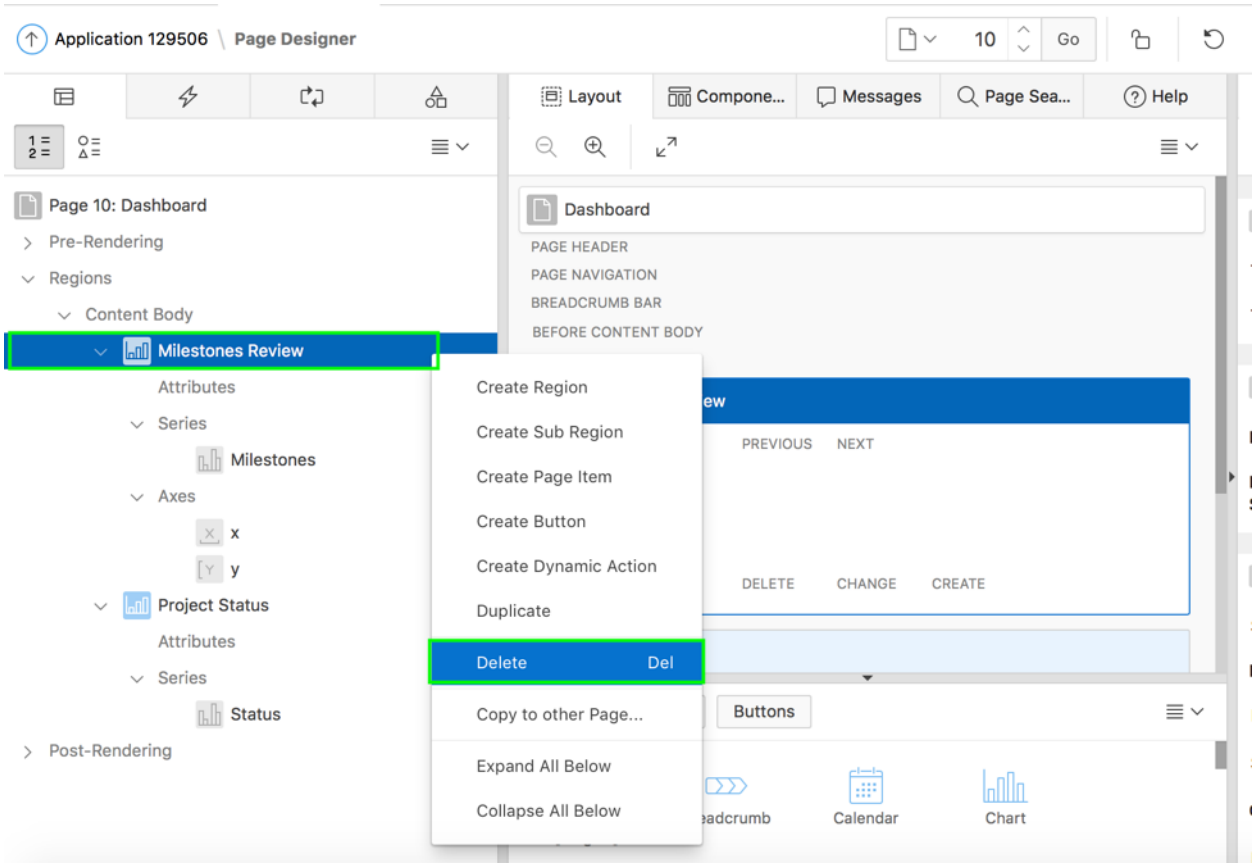
18. For To Page, select the **Milestones Review** page that you created earlier in this lab. You can use the Search dialog to locate the page.
Click **Next**.



19. Accept the defaults and click **Copy**.

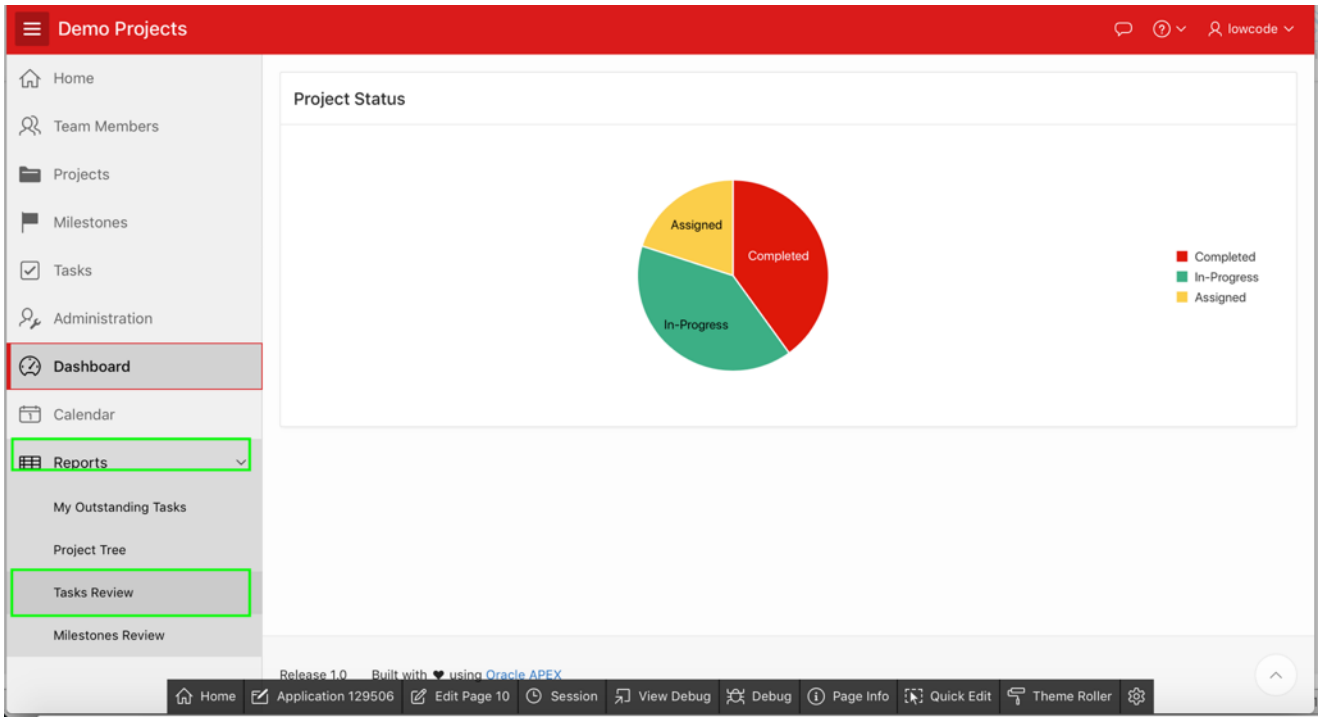


20. Now, under Rendering, right-click **Milestones Review** chart region and select **Delete**.

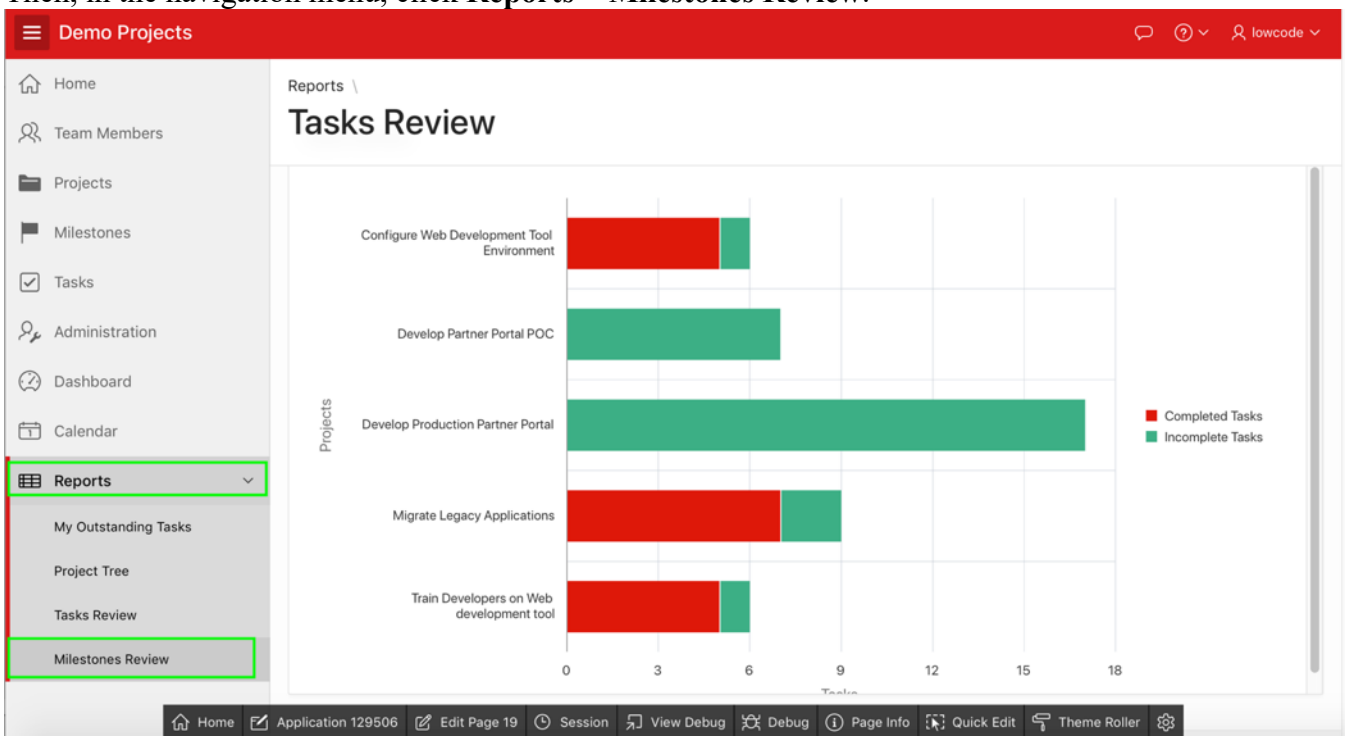


21. Click **Save and Run Page**.

22. Notice that the Dashboard page now shows only the Project Status region.
    In the navigation menu, click **Reports** and then click **Tasks Review**.



23. Then, in the navigation menu, click **Reports > Milestones Review**.

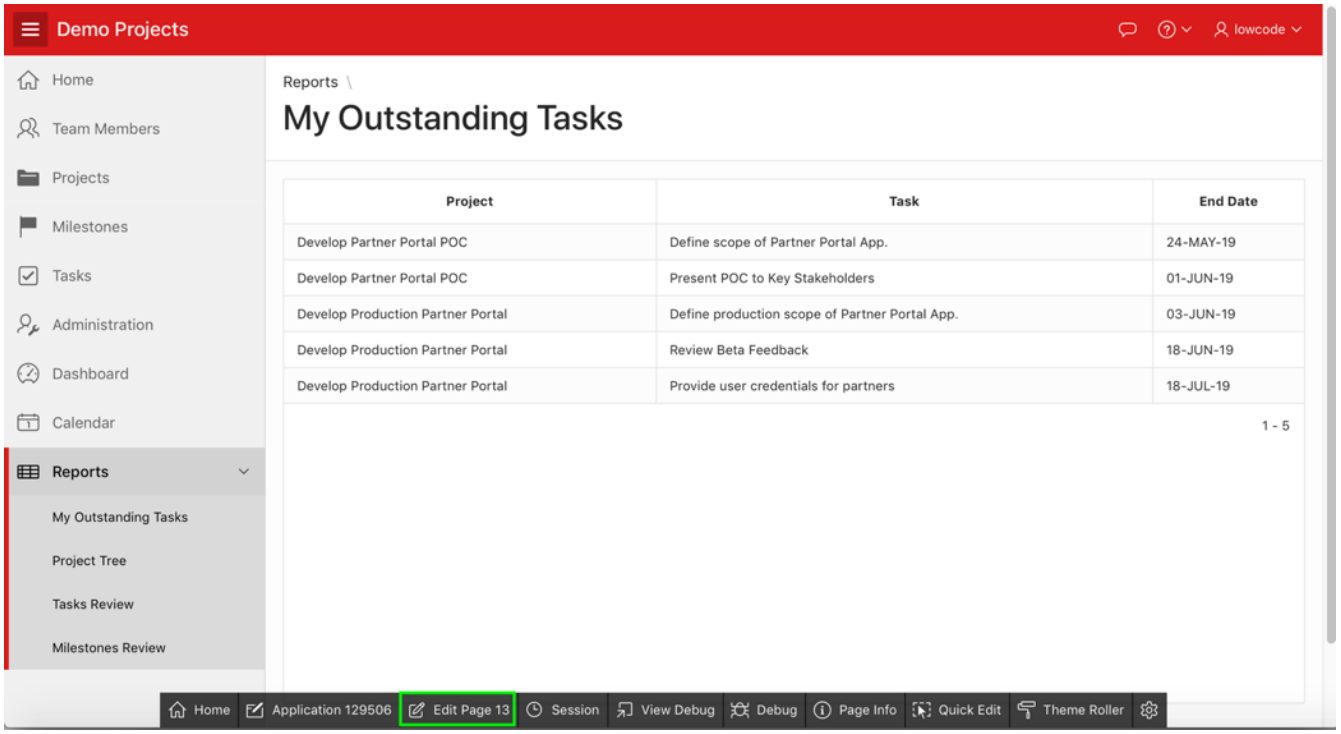


24. The Milestones Review chart page is now displayed.

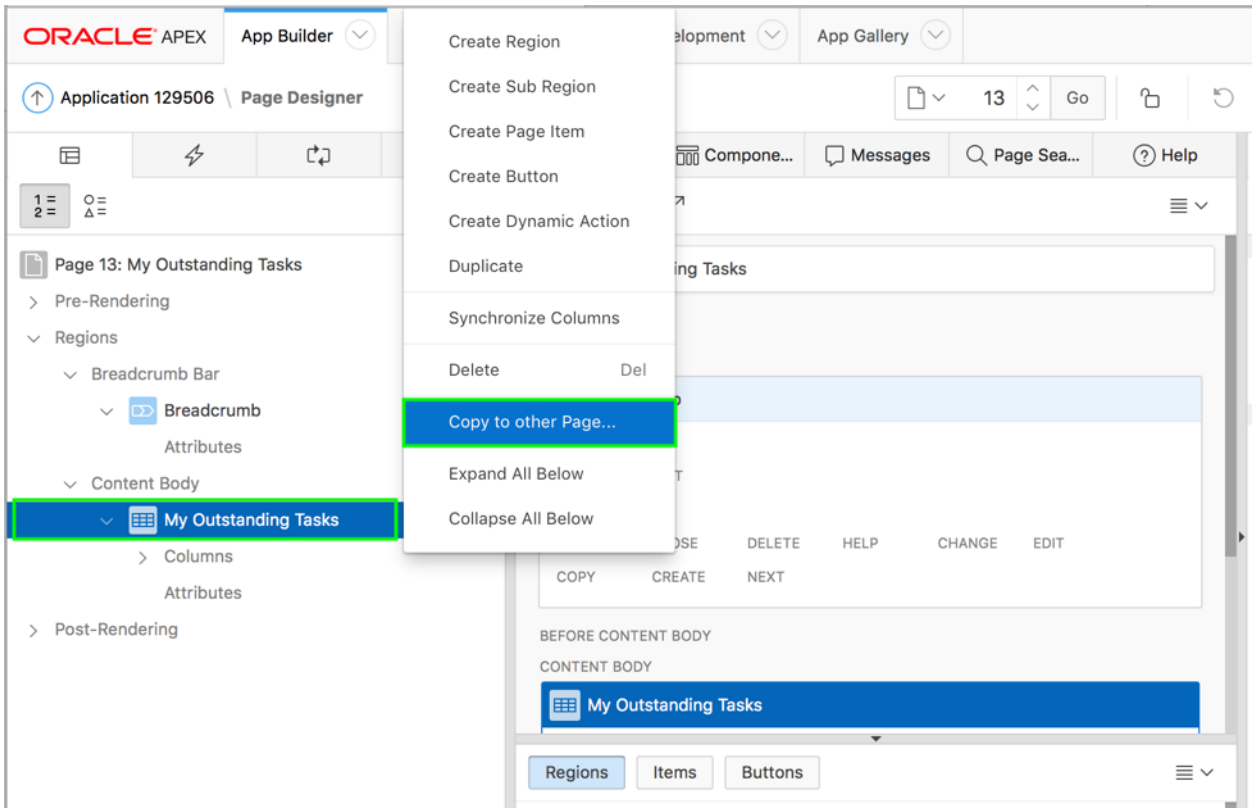


25. You do not need the My Outstanding Tasks report as a separate page. In stead, you want to move that report region to the Reports page.
In the Developer Toolbar, click **Reports > My Outstanding Tasks**.

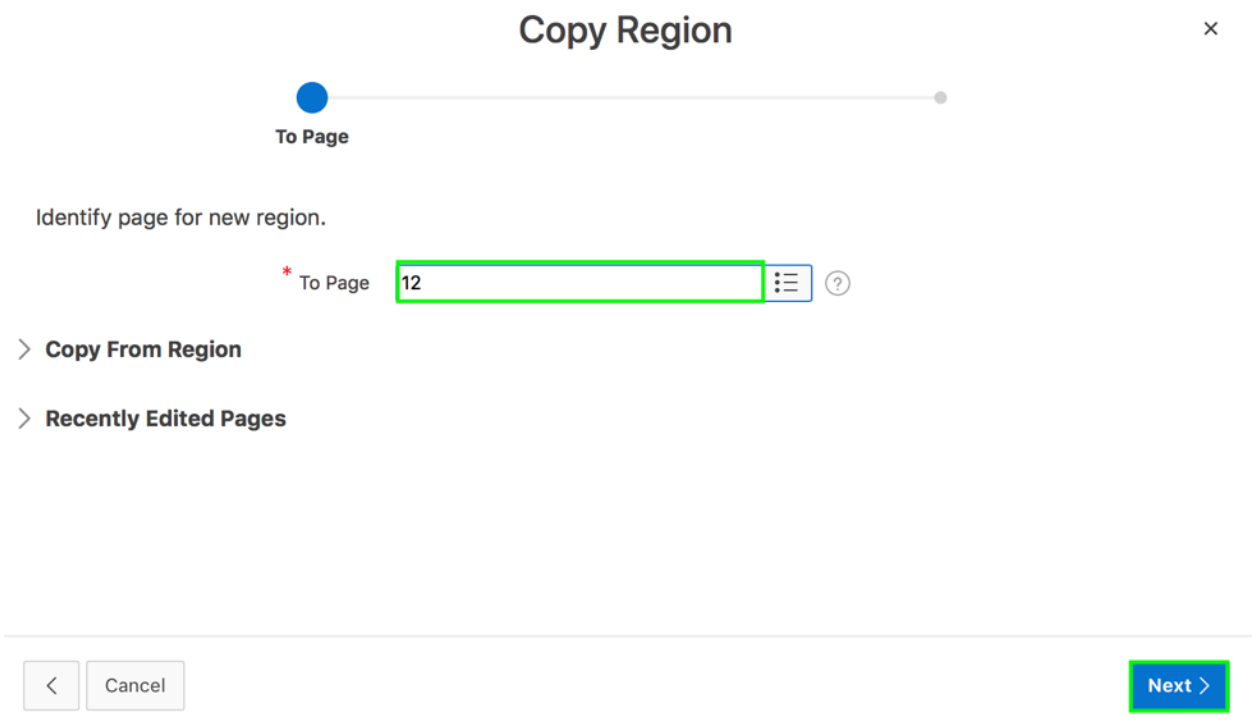
26. In the Developer Toolbar, click **Edit Page<n>**.



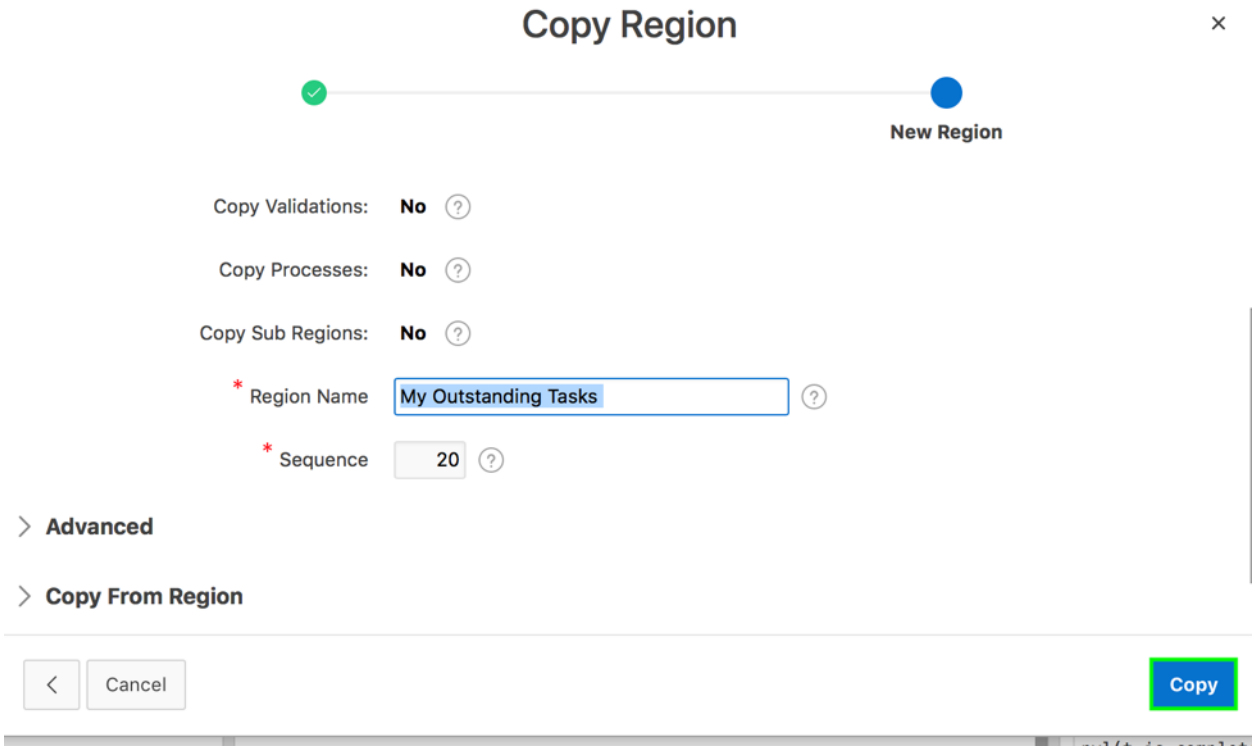
27. In the Page Designer, right-click **My Outstanding Tasks** region and select **Copy to other page**.



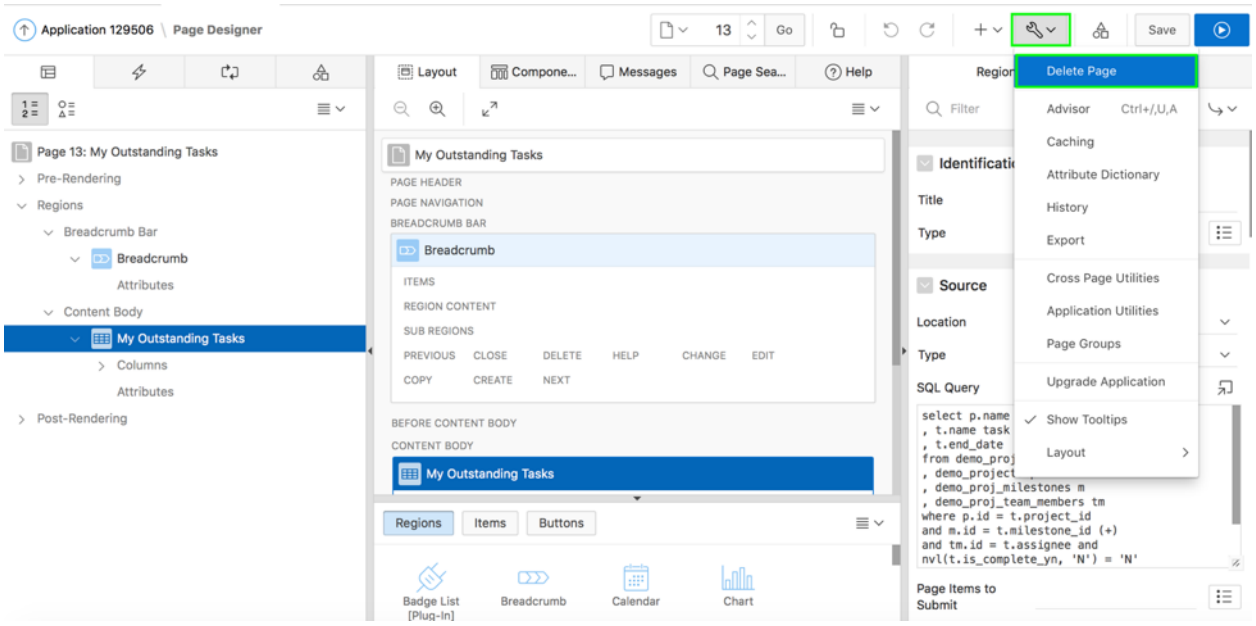
28. For To Page, select the **Reports** page. You can use the Search dialog to locate the page. Click **Next**.



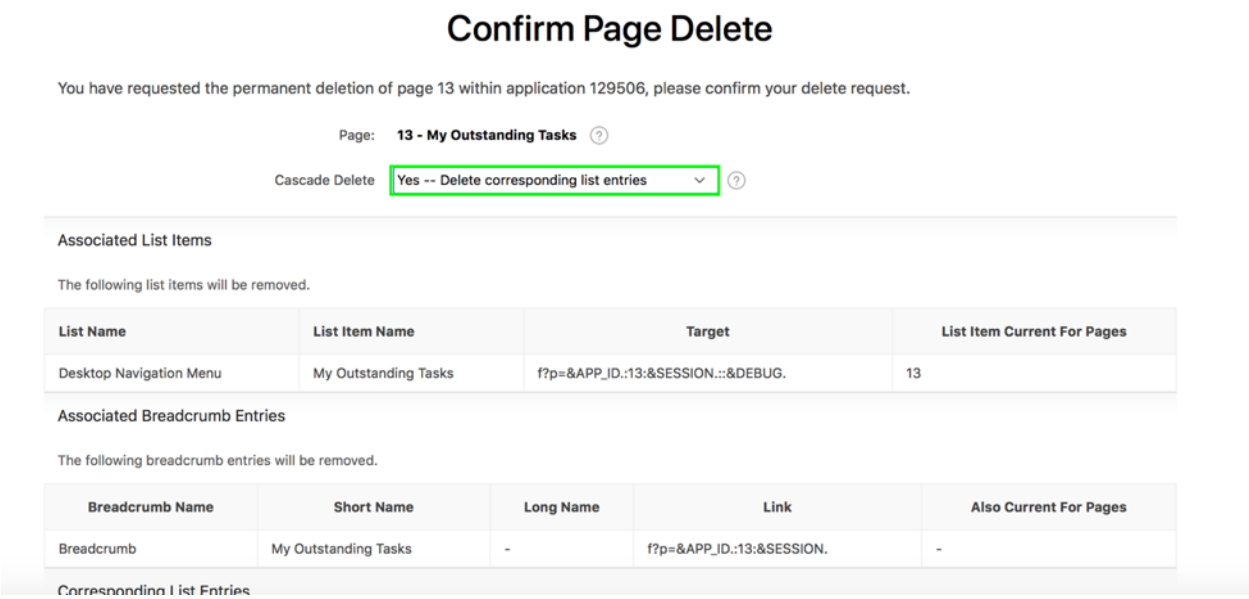
29. Accept the defaults and click **Copy**.



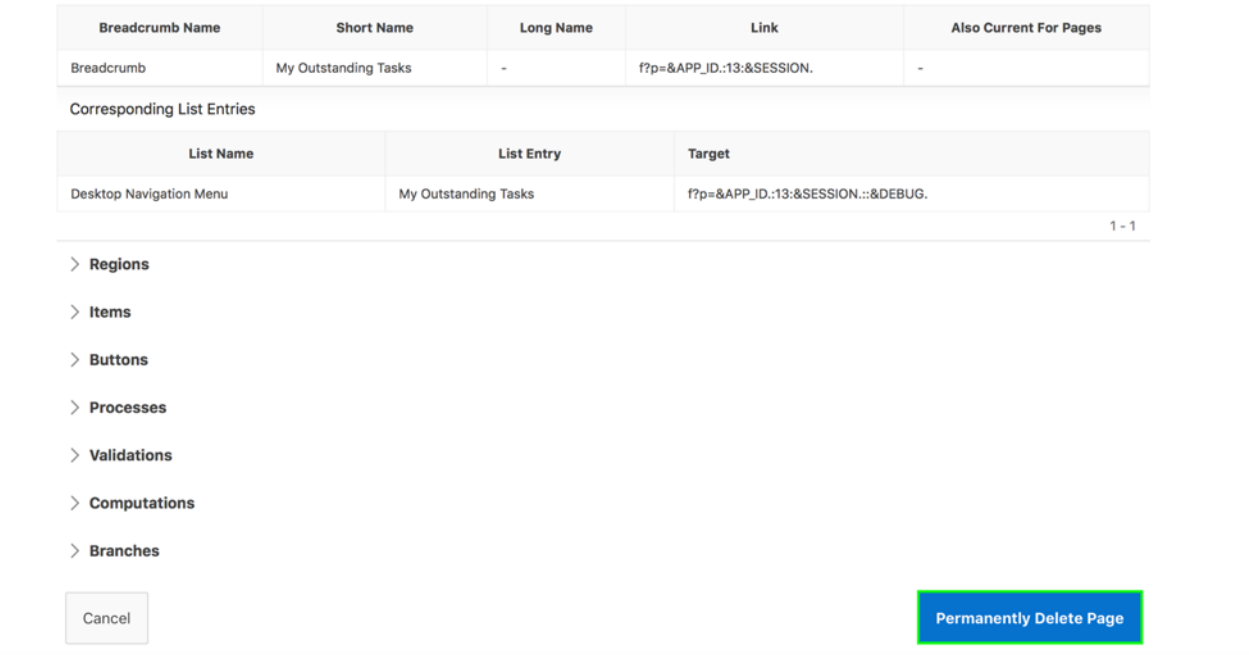
30. Now, click Utilities and select **Delete Page**.



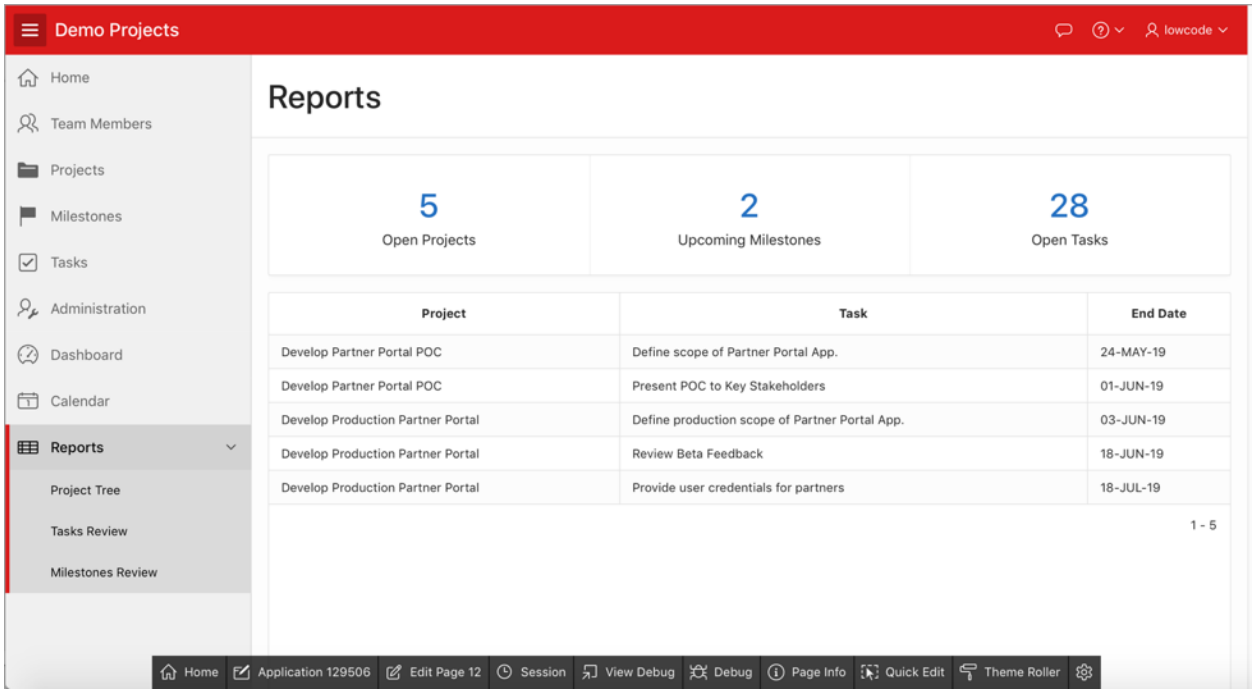
31. On the Confirm Page Delete page, select **Yes – Delete corresponding list entries** for Cascade Delete.



32. Click **Permanently Delete Page**.



33. Navigate to application runtime environment and click **Reports** in the navigation menu. You now see the My Outstanding Tasks region on the Reports page.

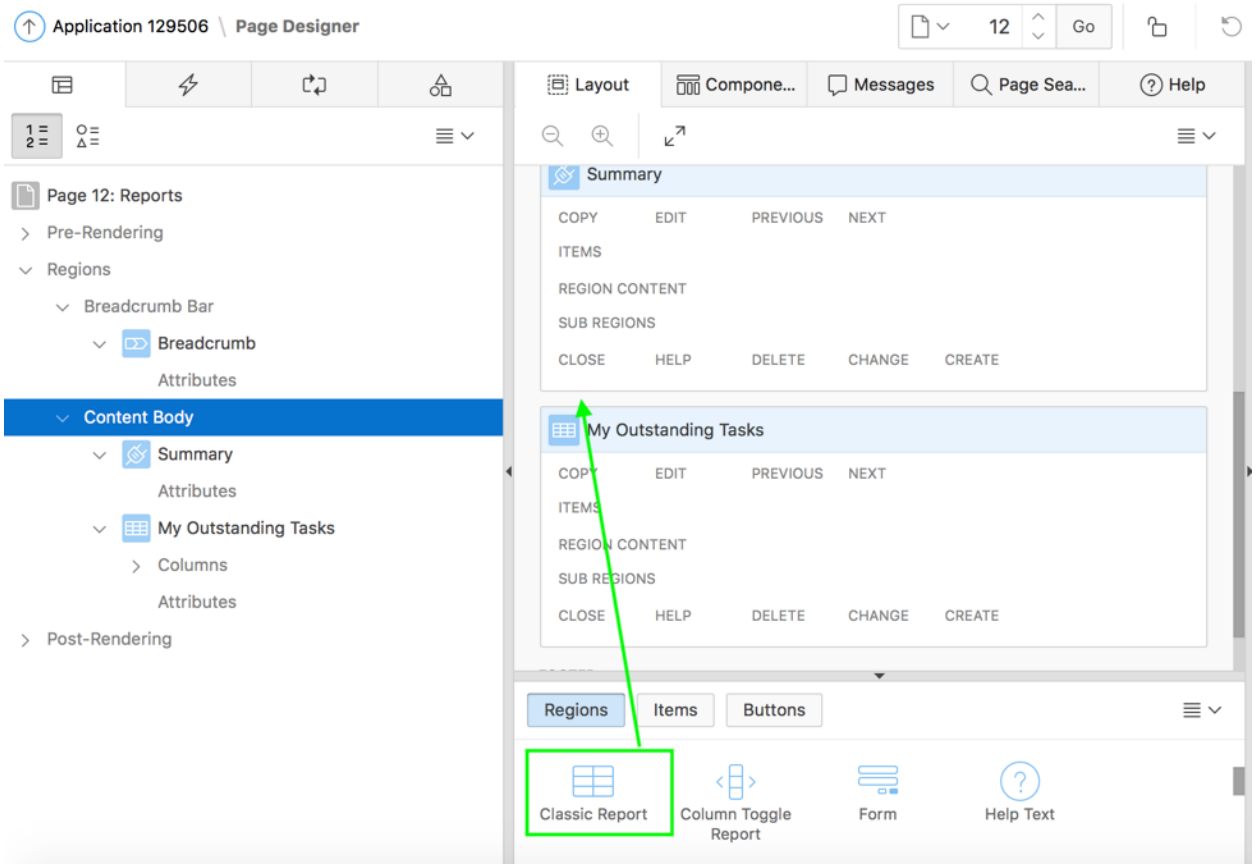

Demo Projects
lowcode
Home
Team Members
Projects
Milestones
Tasks
Administration
Dashboard
Calendar
Reports
Project Tree
Tasks Review
Milestones Review
Reports
5
Open Projects
2
Upcoming Milestones
28
Open Tasks
Project
Task
End Date
Develop Partner Portal POC
Define scope of Partner Portal App.
24-MAY-19
Develop Partner Portal POC
Present POC to Key Stakeholders
01-JUN-19
Develop Production Partner Portal
Define production scope of Partner Portal App.
03-JUN-19
Develop Production Partner Portal
Review Beta Feedback
18-JUN-19
Develop Production Partner Portal
Provide user credentials for partners
18-JUL-19
1 - 5
Home
Application 129506
Edit Page 12
Session
View Debug
Debug
Page Info
Quick Edit
Theme Roller

HOL 15-6: Updating the Reports Page

In this lab, you add new components to the Reports page of the Demo Projects application. You already created a badge list region. This lab covers adding the Recent Projects classic report region and modifying the Summary region attributes.

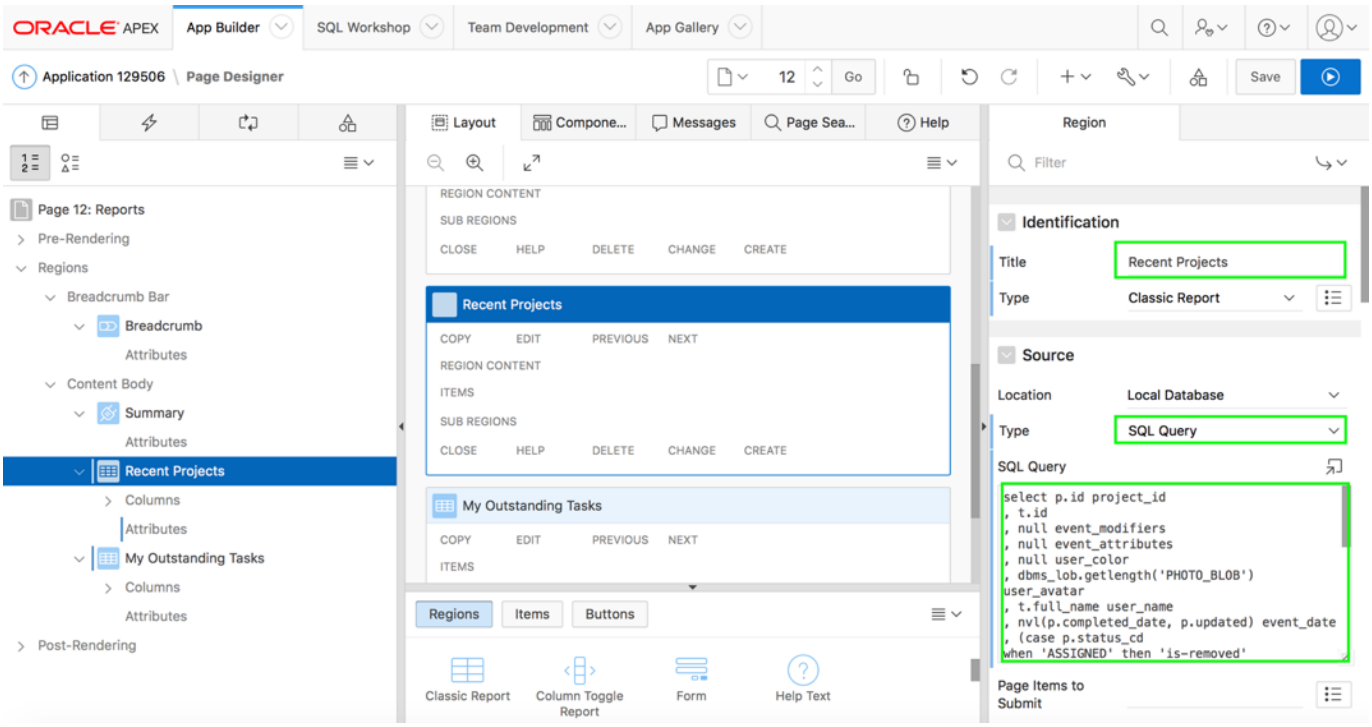
1. You add a new classic report region to the Reports page.
   In the Developer Toolbar, click **Edit Page 12**.

   Now, in the Layout, navigate to Gallery and click **Regions**.
   Click and hold **Classic Report** and drag it into **Content Body**, to a position directly between the Summary and the My Outstanding Tasks regions.



2. In the property editor:
   - Identification > Title: Enter **Recent Projects**
   - Source > Type: **SQL Query**
   - Identification > Source > SQL Query: Copy and paste the following code:

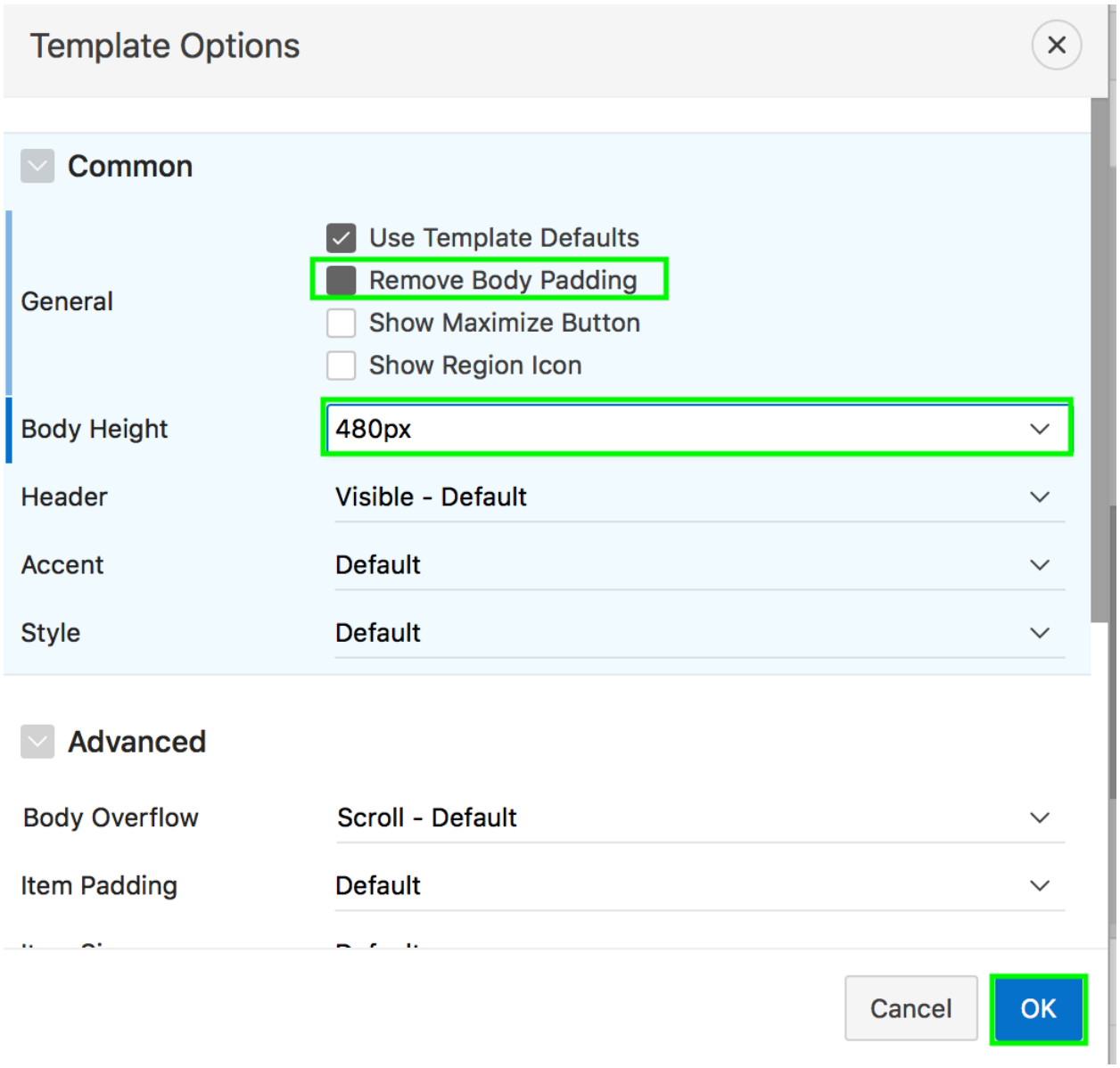
```
select p.id project_id
, t.id
, null event_modifiers
, null event_attributes
, null user_color
, dbms_lob.getlength('PHOTO_BLOB') user_avatar
, t.full_name user_name
, nvl(p.completed_date, p.updated) event_date
, (case p.status_cd
when 'ASSIGNED' then 'is-removed'
when 'IN-PROGRESS' then 'is-updated'
when 'COMPLETED' then 'is-new'
end) event_status
, (select s.description
   from demo_proj_status s
   where s.cd = p.status_cd
  ) event_type
, (case p.status_cd
when 'ASSIGNED' then 'fa fa-clock-o'
when 'IN-PROGRESS' then 'fa fa-refresh'
when 'COMPLETED' then 'fa fa-check'
end) event_icon
, p.name event_title
, p.description event_desc
from demo_projects p
, demo_proj_team_members t
where p.project_lead = t.id
order by p.updated desc
```



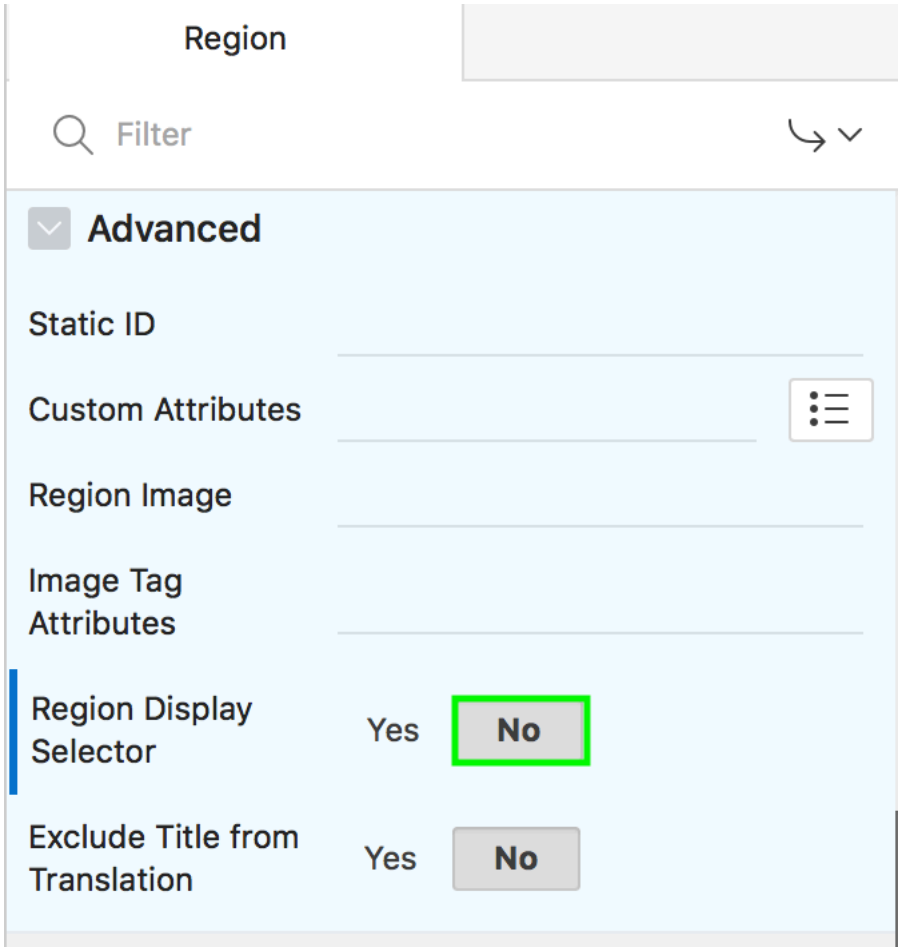
3. In the property editor, locate Appearance and click **Template Options**.

4. In the Template Options dialog:
   - For General, select the **Remove Body Padding** check box.
   - For Body Height, select **480px**.

   Click **OK**.

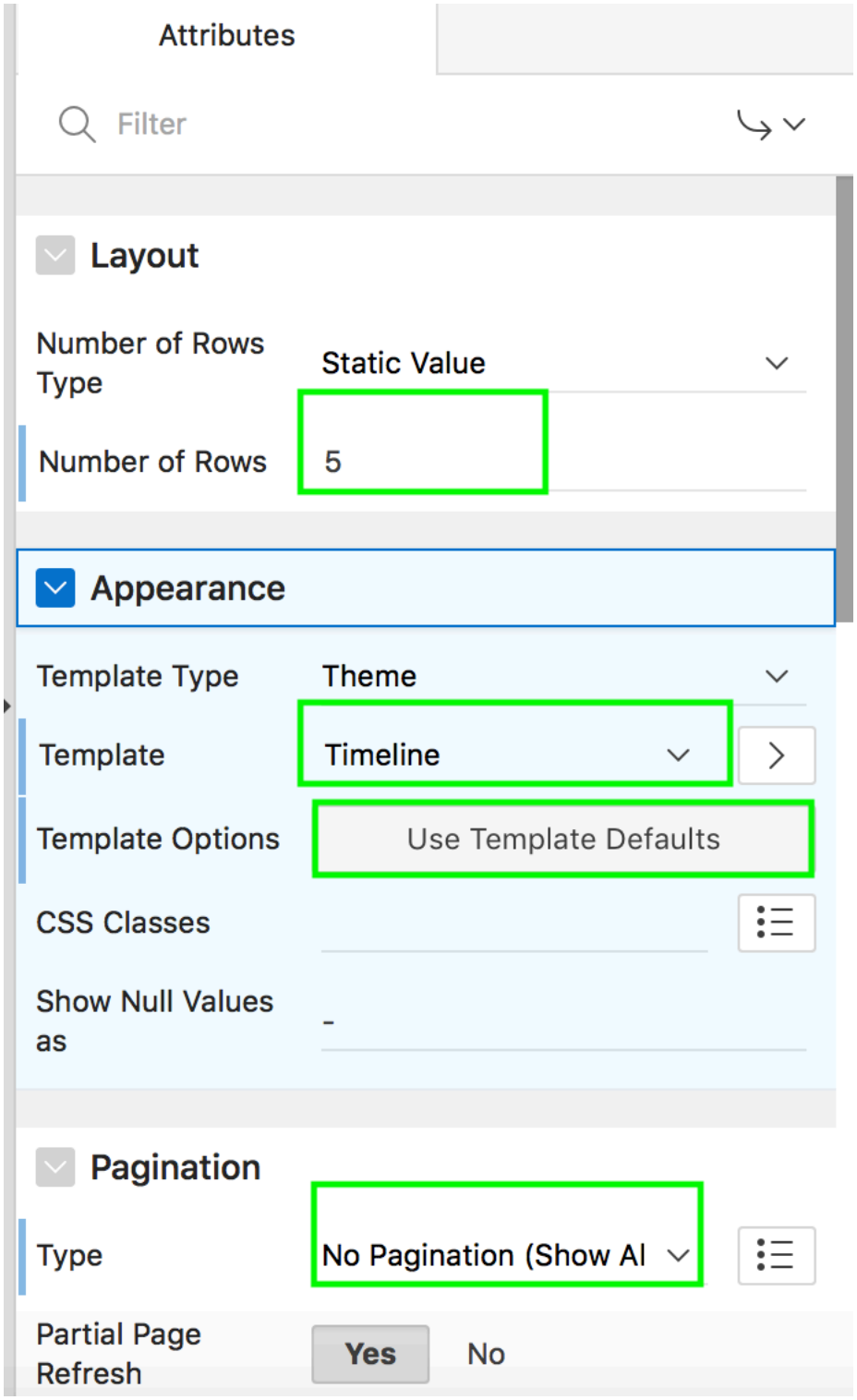


5. In the property editor, under Advanced, select **No** for Region Display Selector.

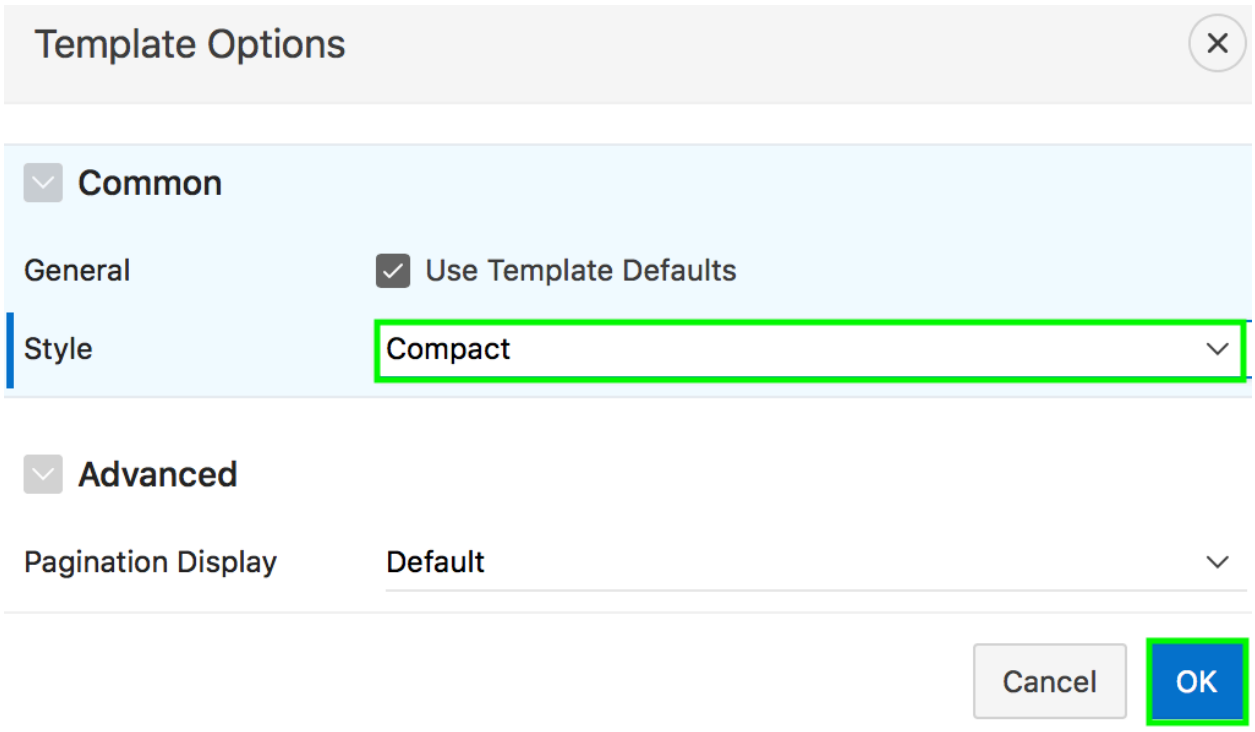


6. Under Rendering, expand Regions > Recent Projects and select **Attributes**.
   In the property editor:
   - Layout > Number of Rows: Enter **5**
   - Appearance > Template: Select **Timeline**
   - Pagination > Type: Select **No Pagination (Show All Rows)**

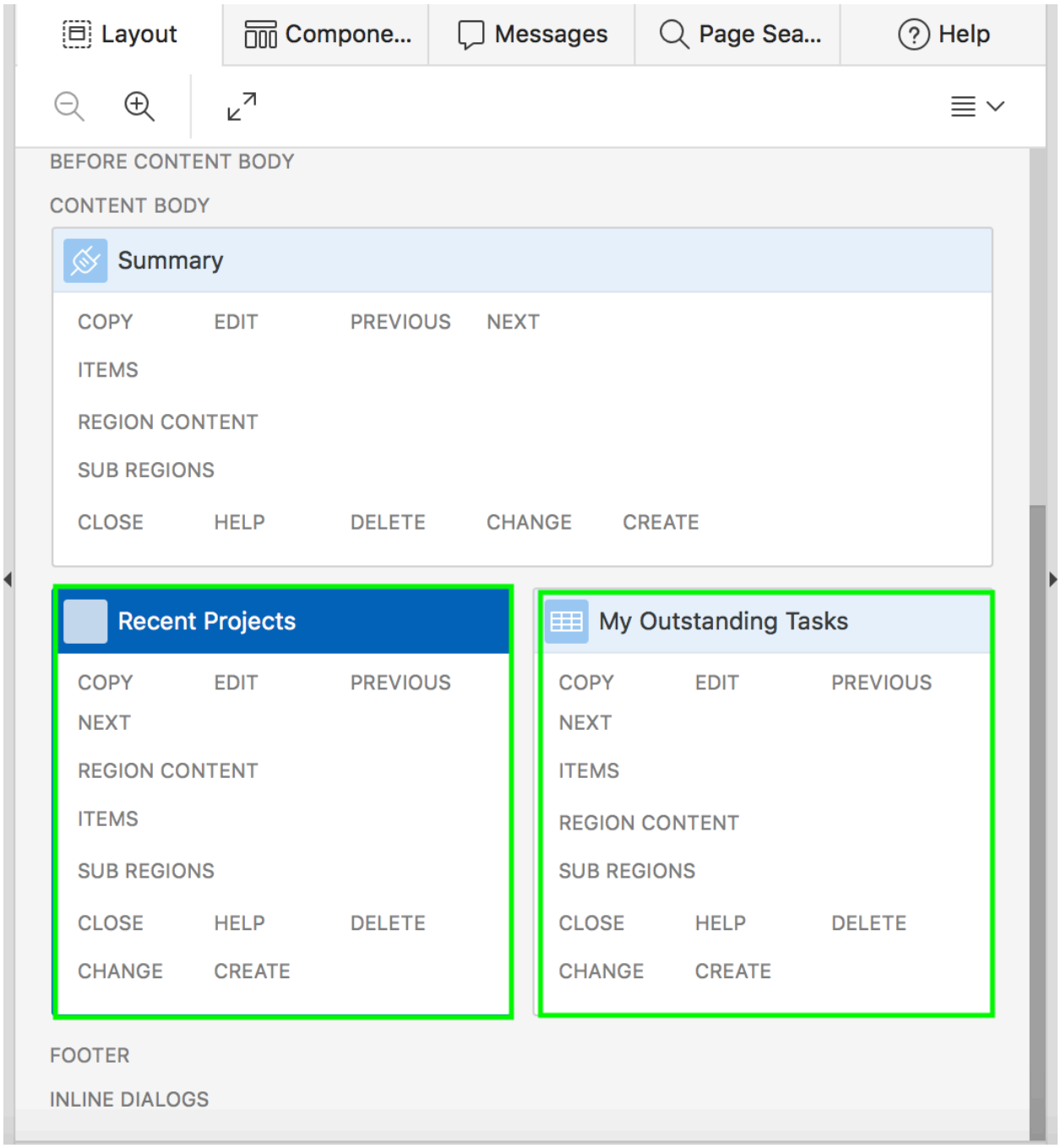
   Under Appearance, click **Template Options**.



7. In the Template Options dialog, select **Compact** for Style.
Click **OK**.

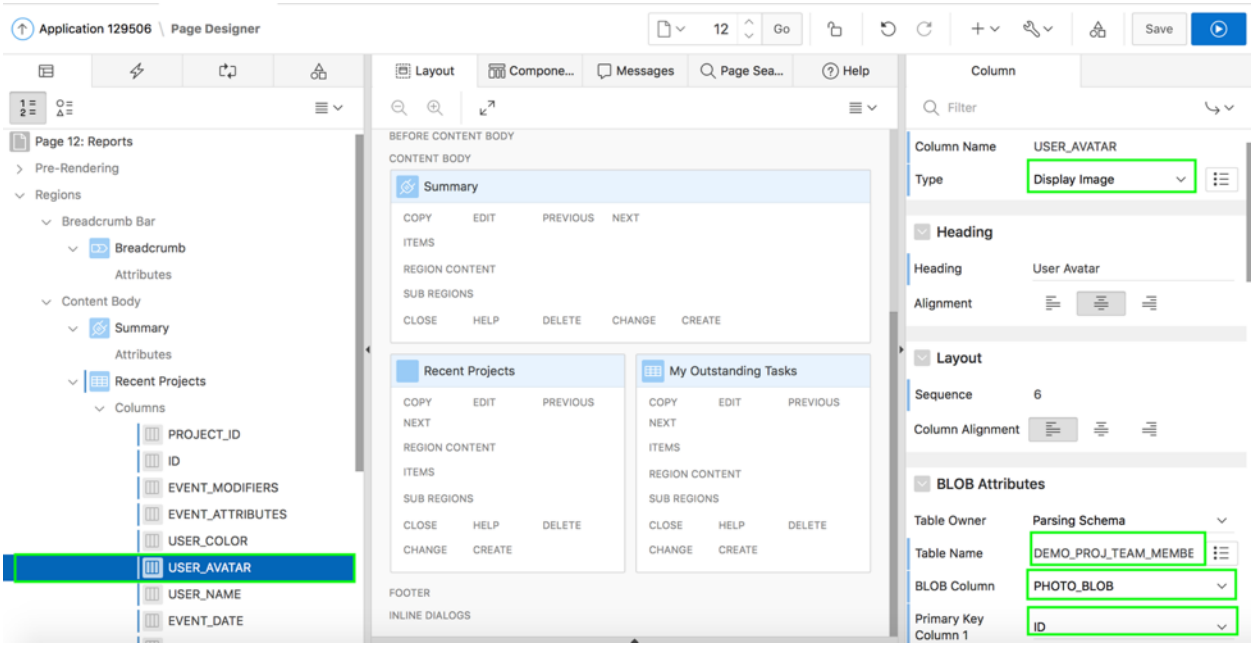


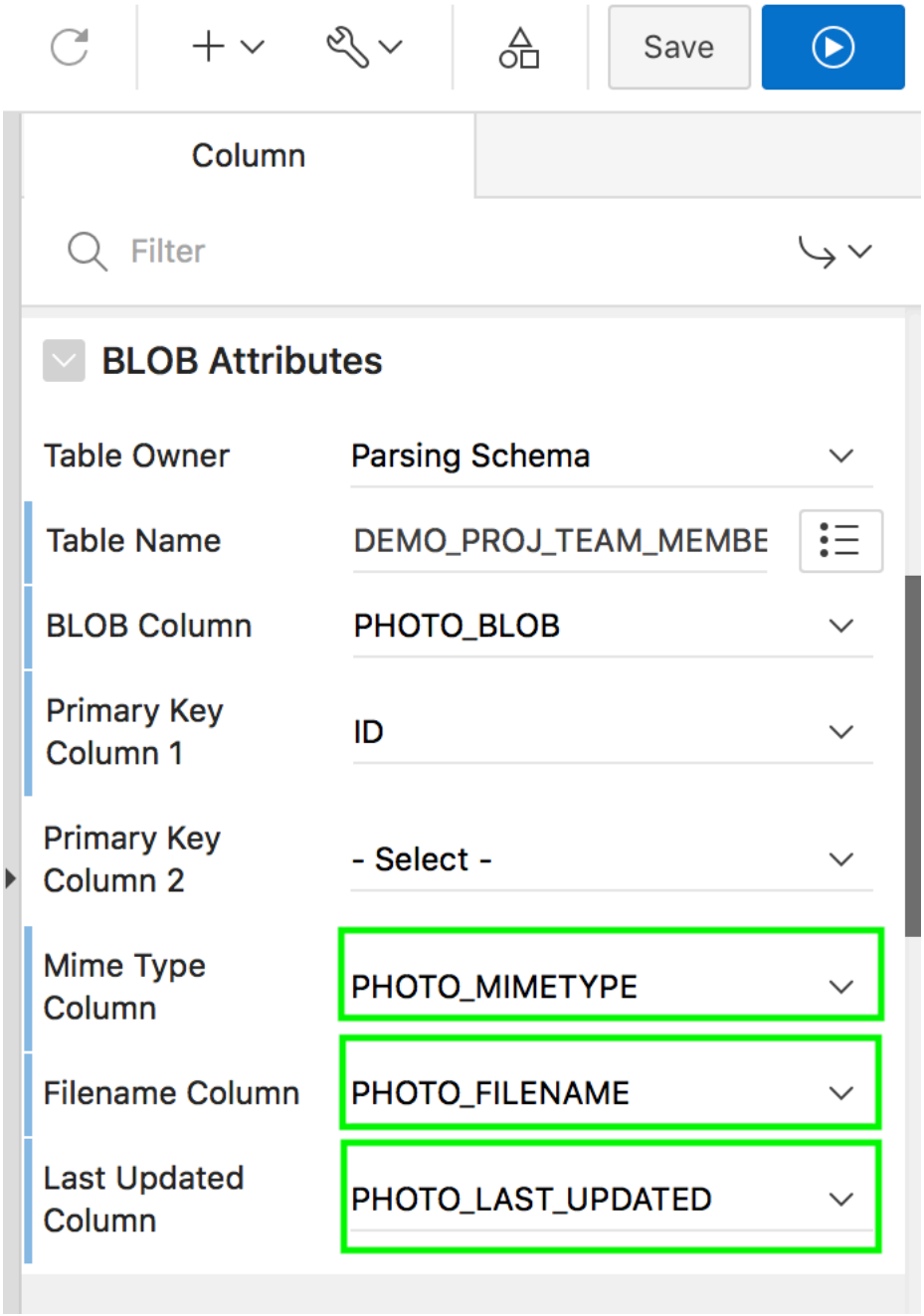
8. In the Layout, click and hold the **My Outstanding Tasks** region and drag it into the position next to the Recent Projects region. Both the Recent Projects and My Outstanding Tasks regions should now appear on the same line horizontally.



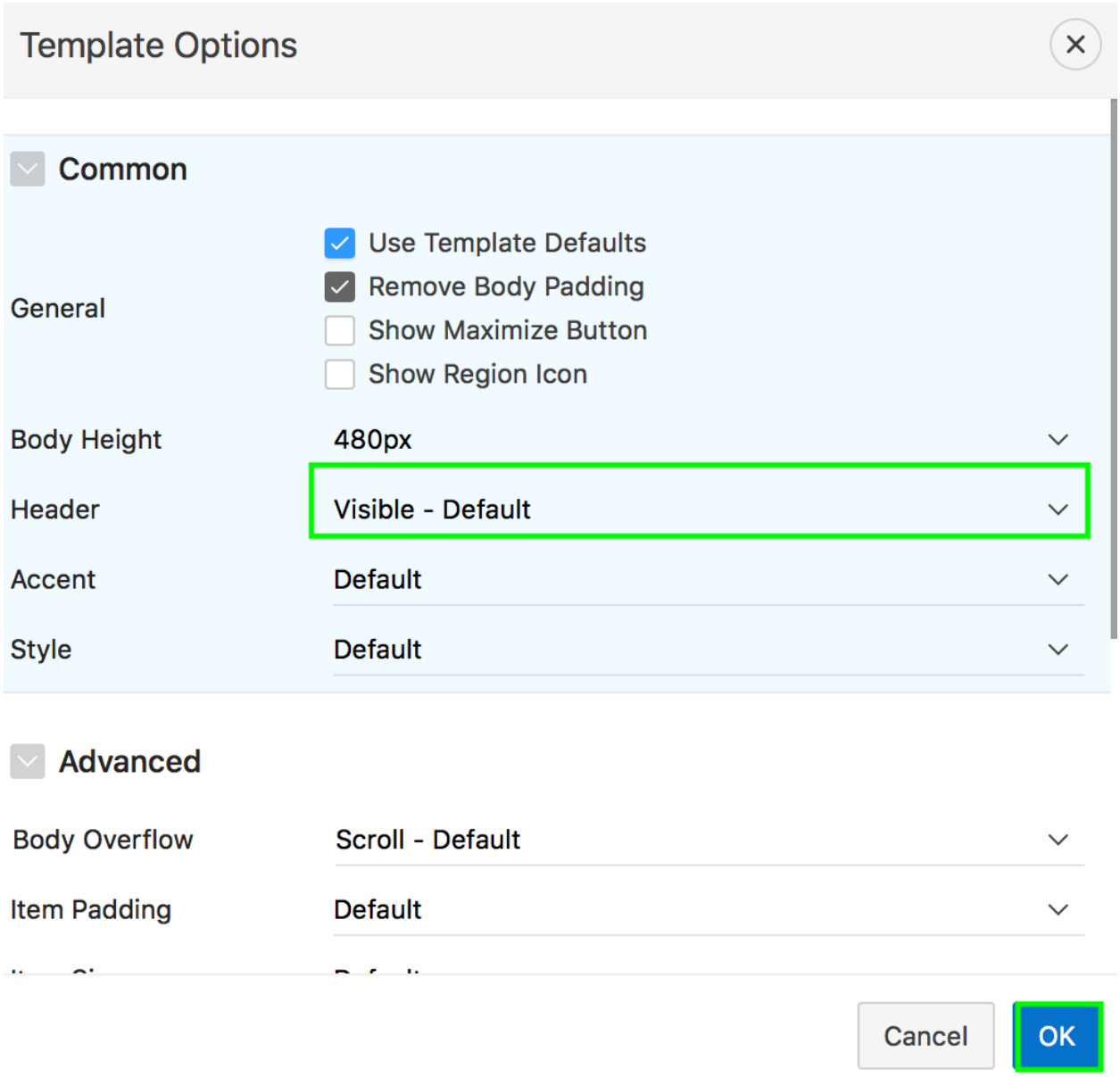
9. Under Rendering, expand **Regions > Recent Projects > Columns**.
   Click **USER_AVATAR**.
   In the property editor:
   - Identification > Type: Select **Display Image**
   - BLOB Attributes > Table Name: Select **DEMO_PROJ_TEAM_MEMBERS**
   - BLOB Attributes > BLOB Column: Select **PHOTO_BLOB**
   - Primary Key Column 1: Select **ID**
   - BLOB Attributes > Mime Type Column: Select **PHOTO_MIMETYPE**

- BLOB Attributes > Filename Column: Select **PHOTO_FILENAME**
- BLOB Attributes > LAST_UPDATED Column: Select **PHOTO_LAST_UPDATED**

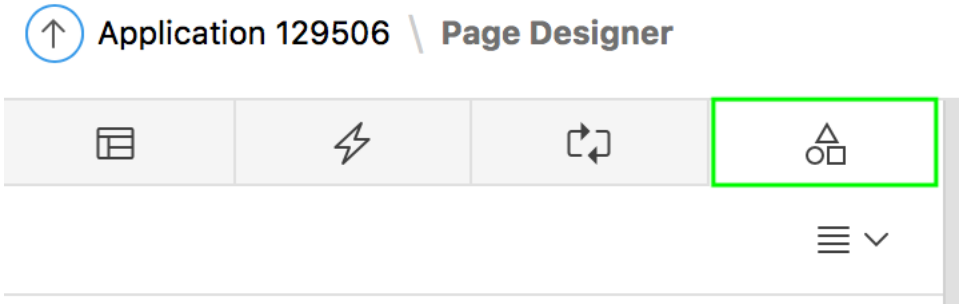




10. Under Rendering, select **My Outstanding Tasks**.
    In the property editor, under Appearance, click **Template Options**.
    For Header, select **Visible – Default**.
    Click **OK**.

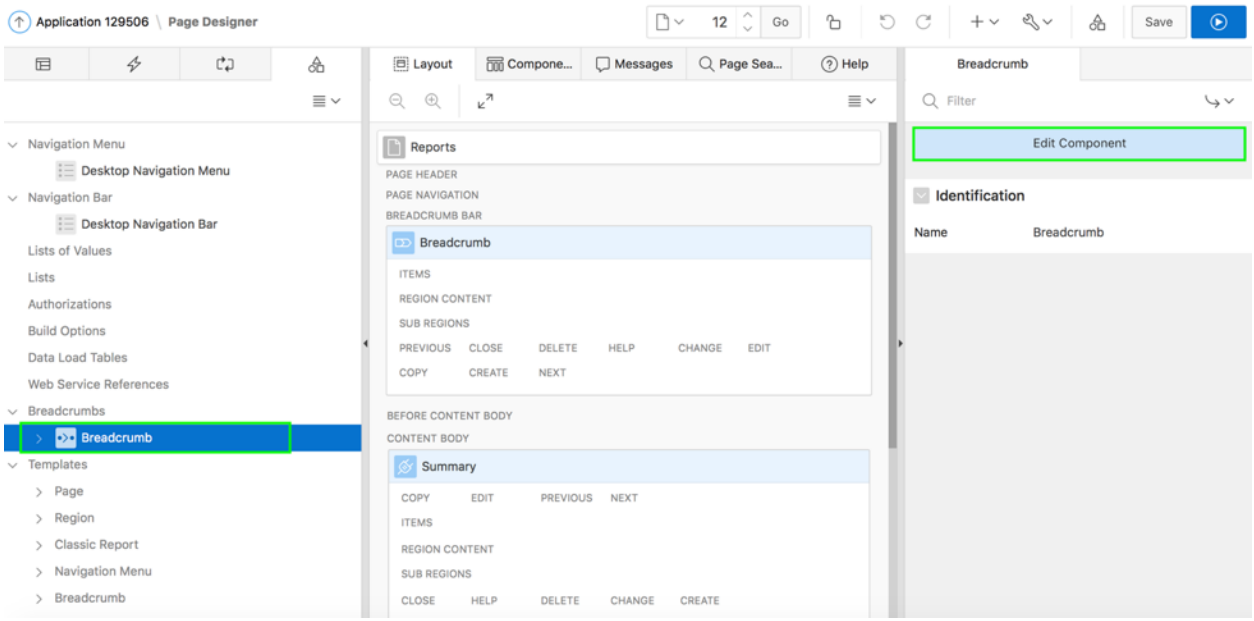


11. Click **Page Shared Components**.

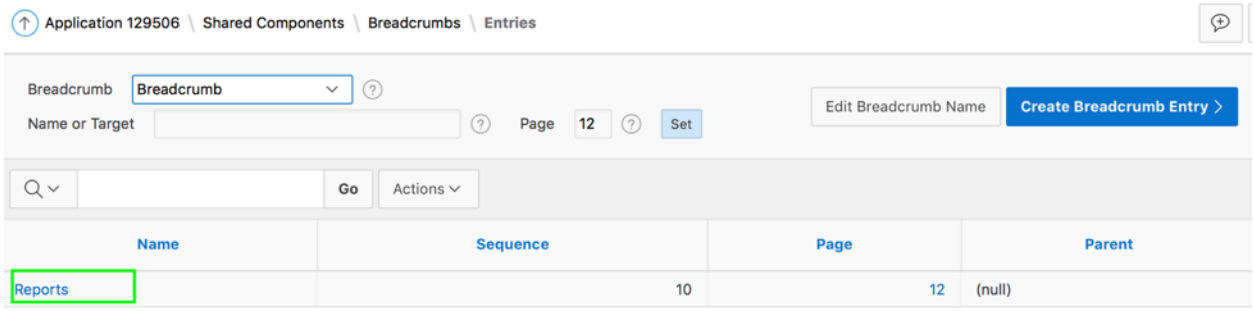


12. Expand Breadcrumbs and select **Breadcrumb**.

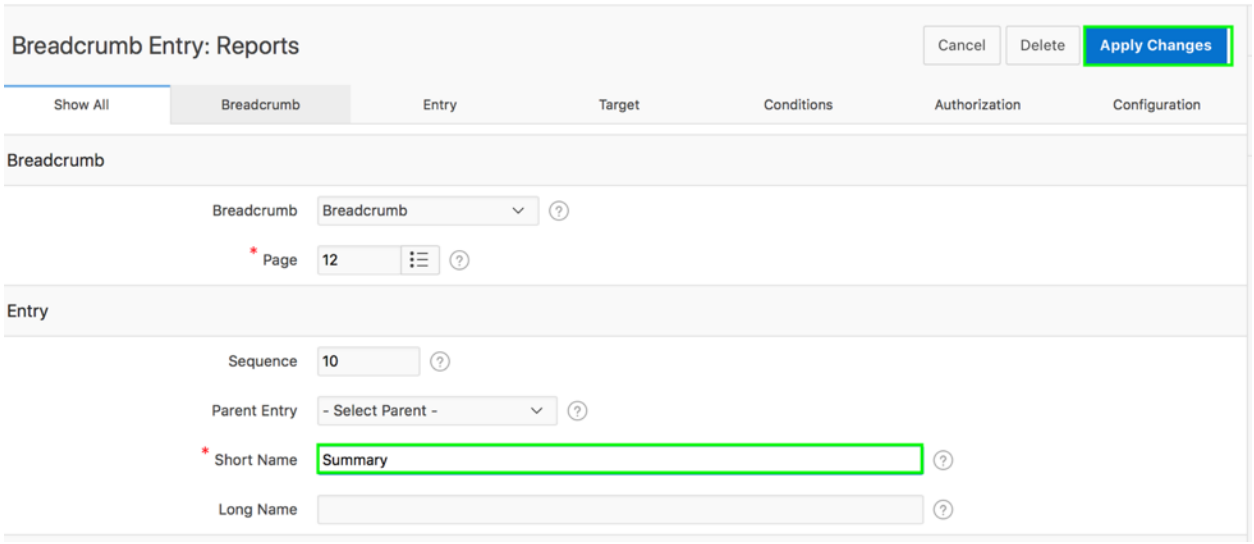
    In the property editor, click **Edit Component**.



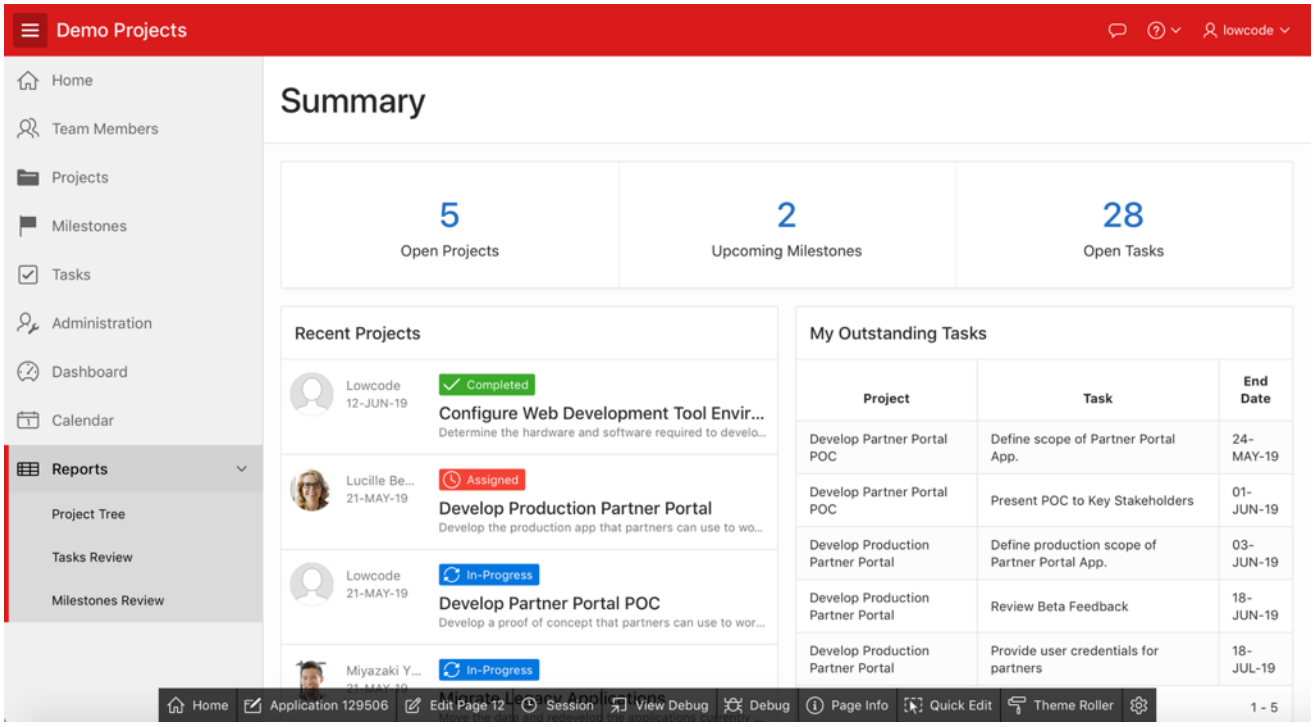
13. Click **Reports**.



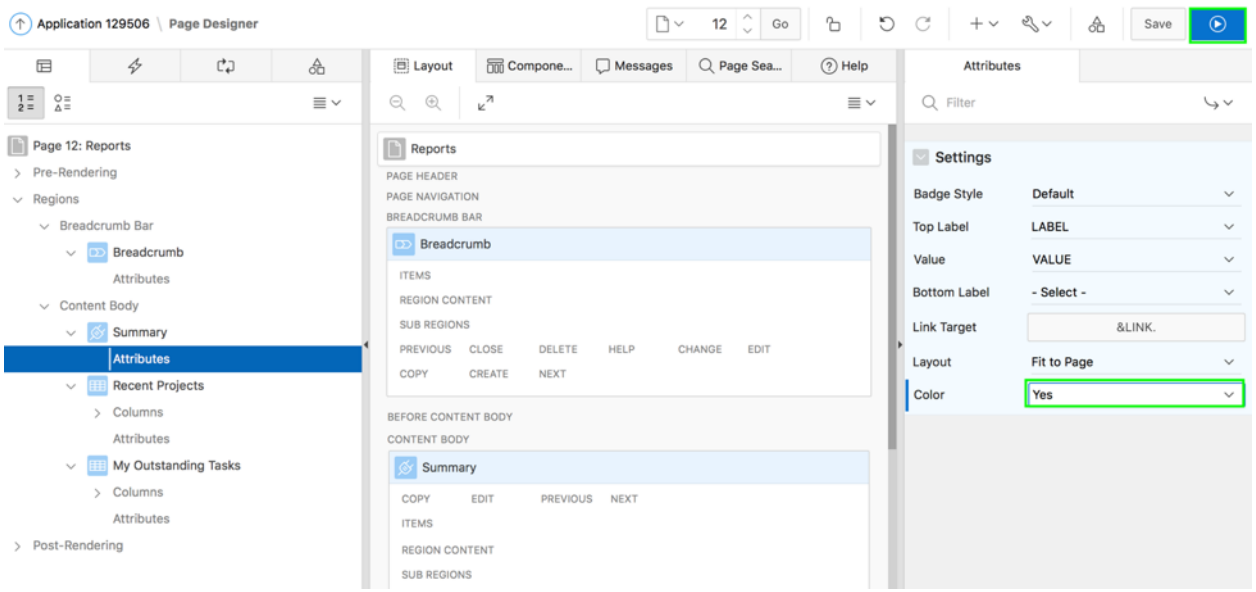
14. Under Entry, enter **Summary** for Short Name.
    Click **Apply Changes**.



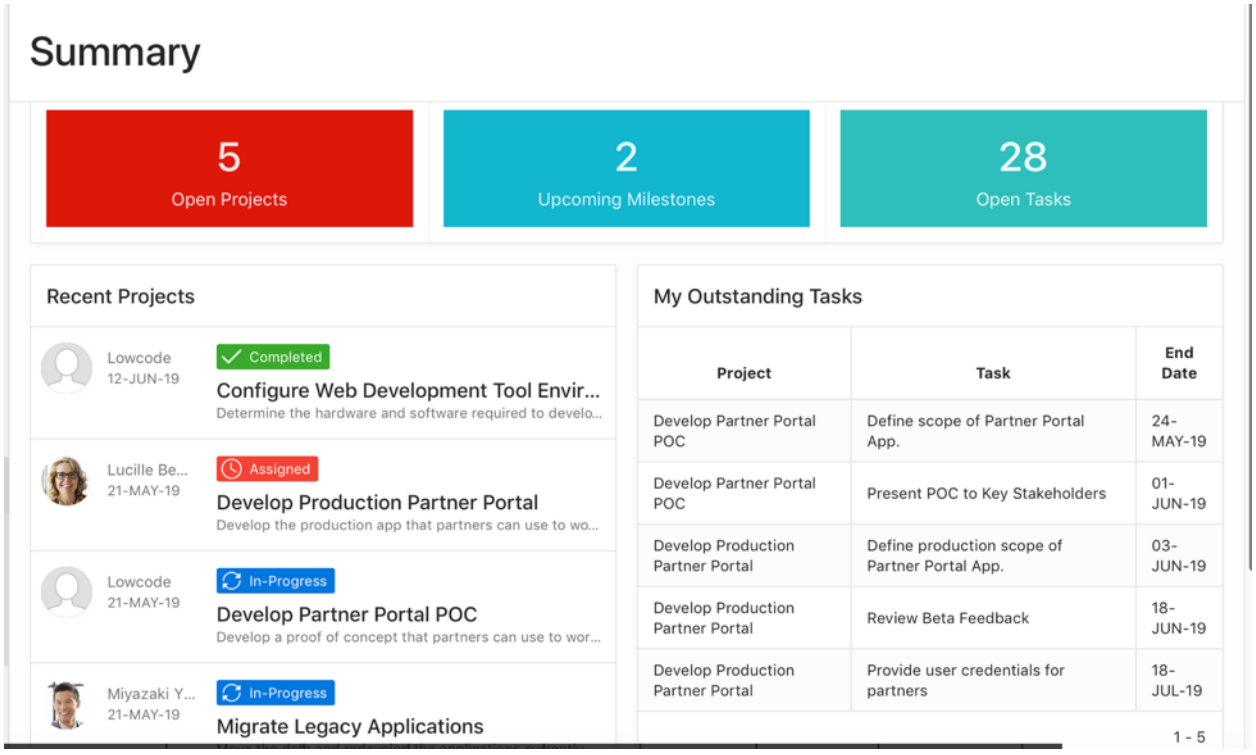
15. Click **Save and Run Page**.
The updated page is displayed now.



16. You can set color for the badges.
    Navigate to page designer and under Rendering, expand Regions > Summary. Then, select **Attributes**.
    In the property editor, under Settings, for Color, select **Yes**.
    Click **Save and Run Page**.



17. The Reports page now is displayed with the colored badges.

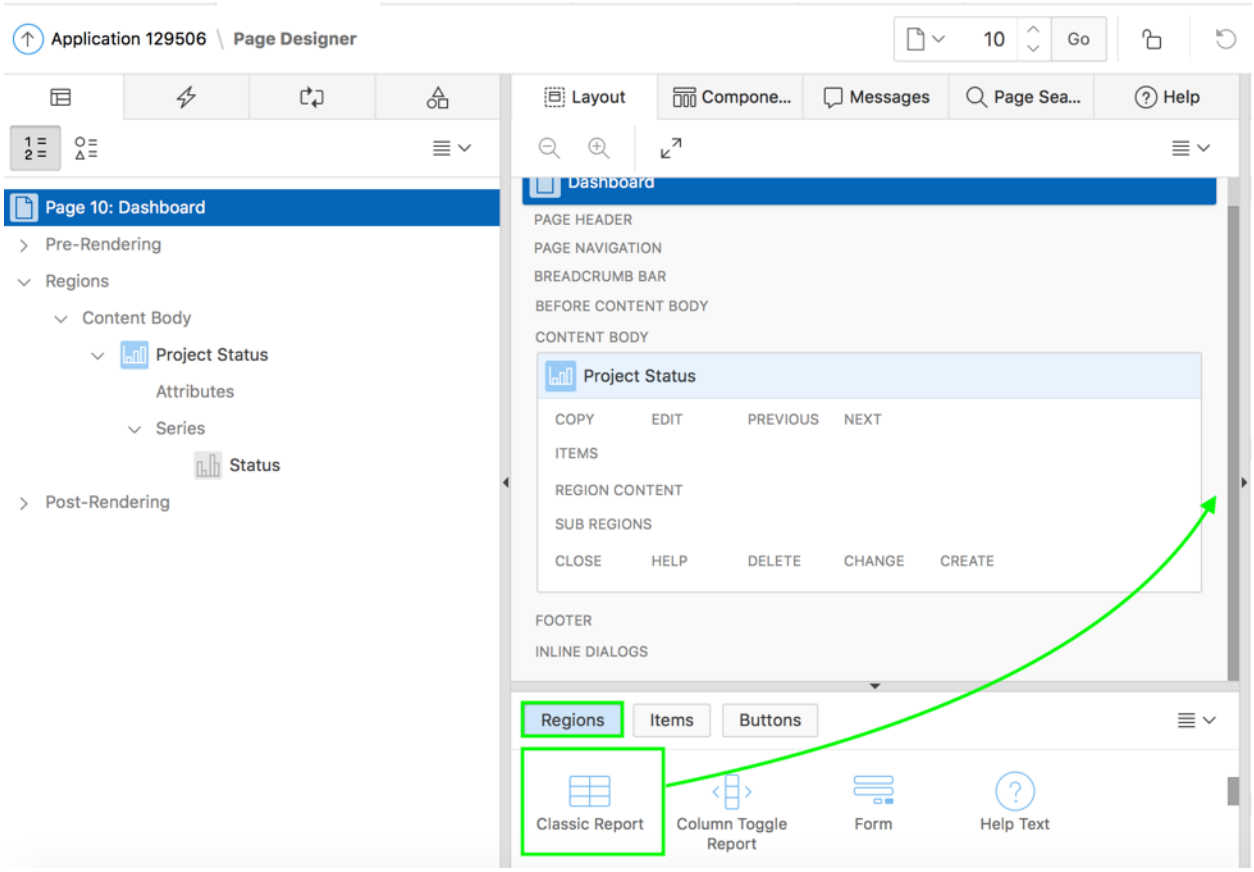


HOL 15-7: Updating the Dashboard Page

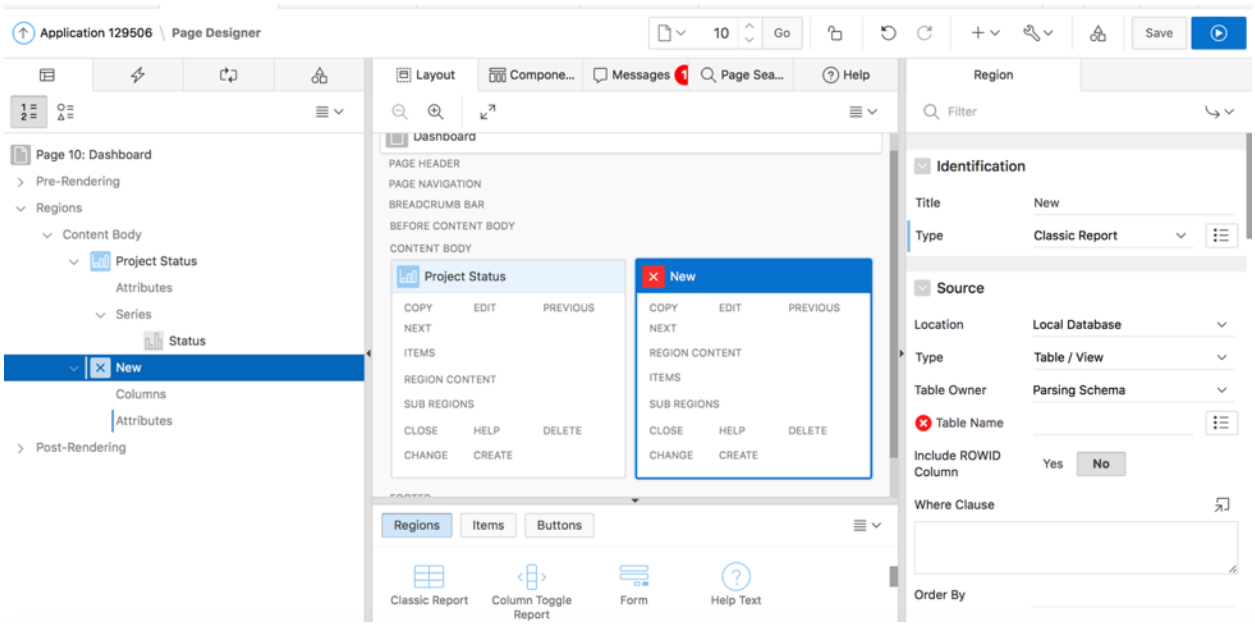
In this lab, you update the Dashboard page of the Demo Projects application. The Dashboard should show Project Status chart, Upcoming Milestones report. At the bottom of the page it would be good to show incomplete tasks to round off the dashboard page.

1. In the left navigation menu, click **Dashboard**.
   In the Developer Toolbar, click **Edit <Page *n*>**.

2. Under Layout, navigate to Gallery and click **Regions**.
   Select **Classic Report**, and then drag and drop this region to the right side of Project Status region.



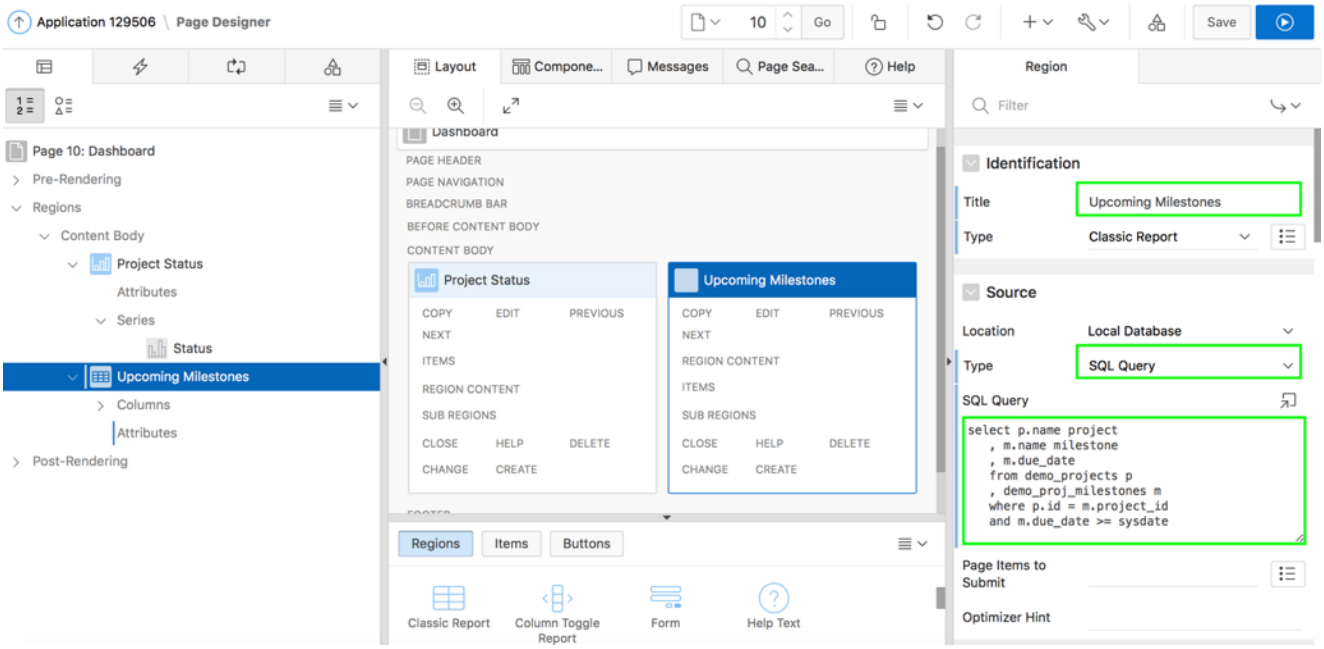
The new region should be placed as displayed in the screenshot below:



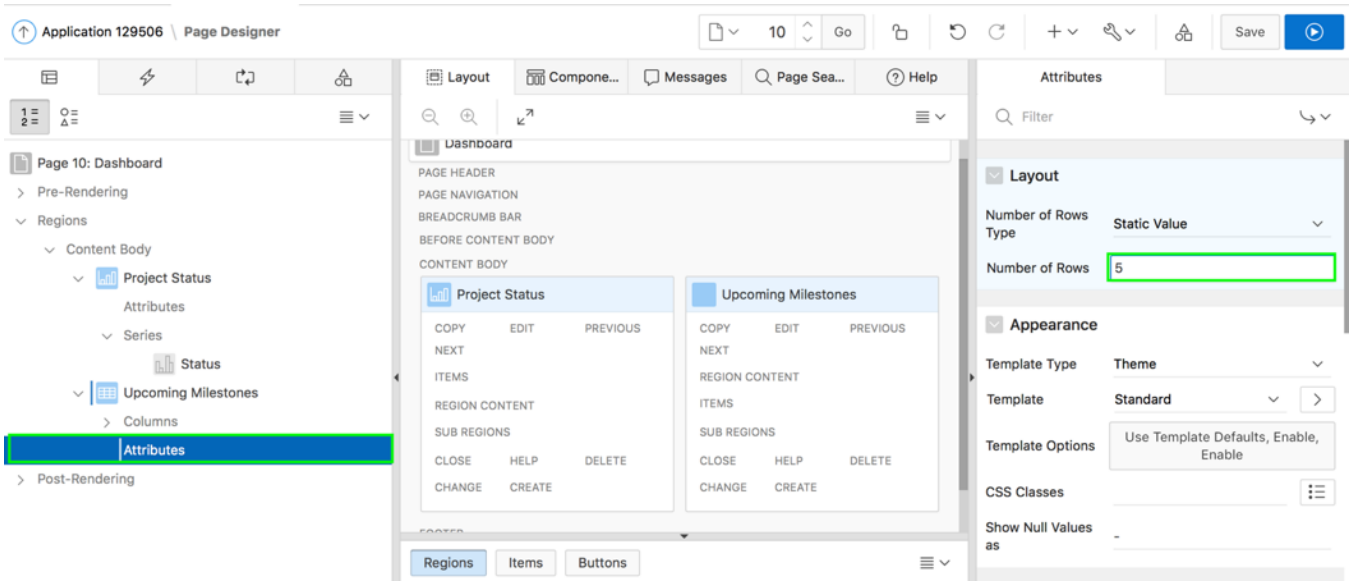
3. In the property editor, enter / select the following:
   - Identification > Title: **Upcoming Milestones**
   - Source > Type: **SQL Query**
   - Source > SQL Query:

```
select p.name project
    , m.name milestone
    , m.due_date
    from demo_projects p
    , demo_proj_milestones m
    where p.id = m.project_id
    and m.due_date >= sysdate
```

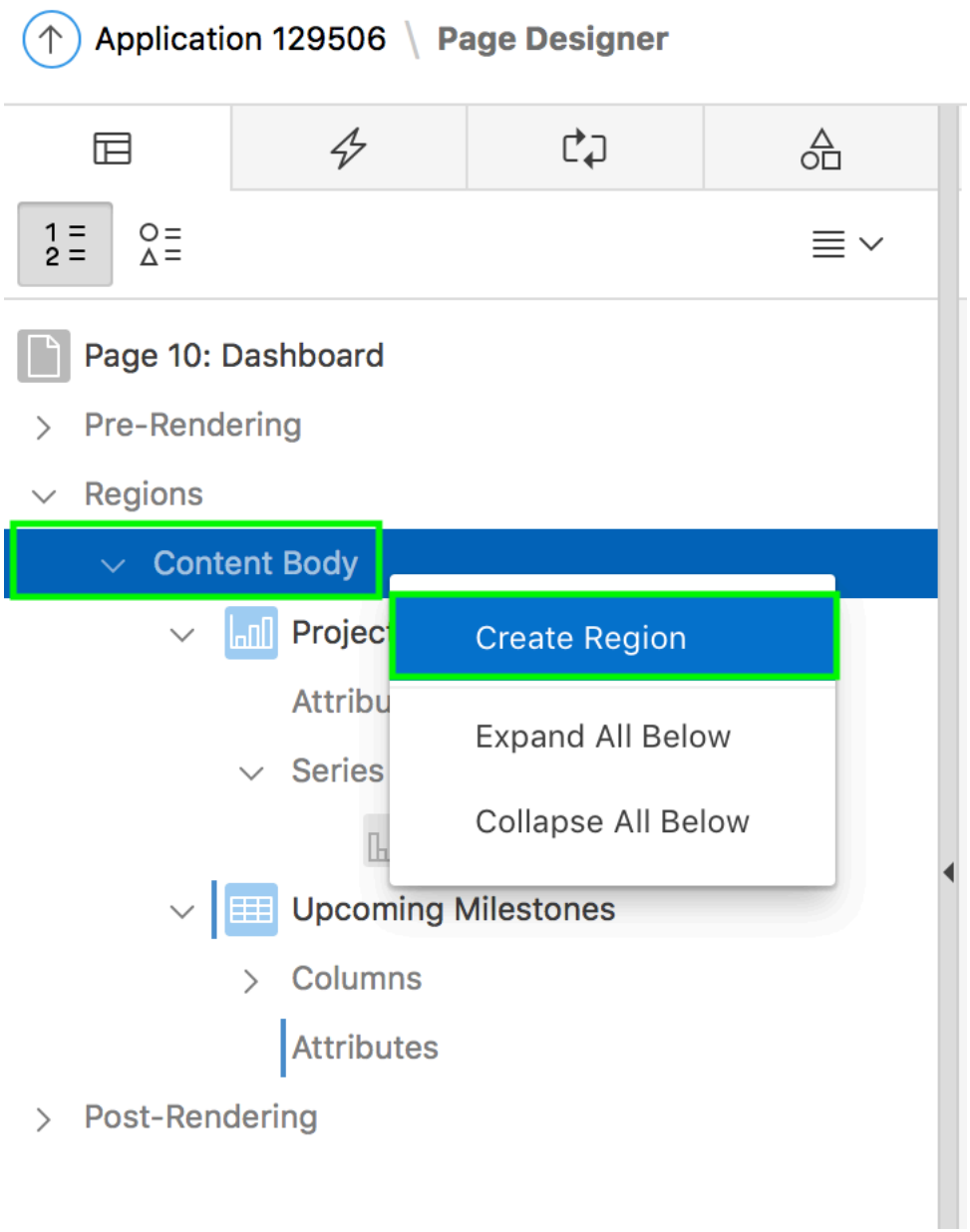
Note: To ensure only future milestones are returned, the milestone due date is compared to sysdate which represents the current date in the Oracle database.



4. Update the report attributes, to only display 5 records at once.
   In the Rendering tab, under **Upcoming Milestones**, click **Attributes**. In the Property Editor, for Layout > Number of Rows, enter **5**.



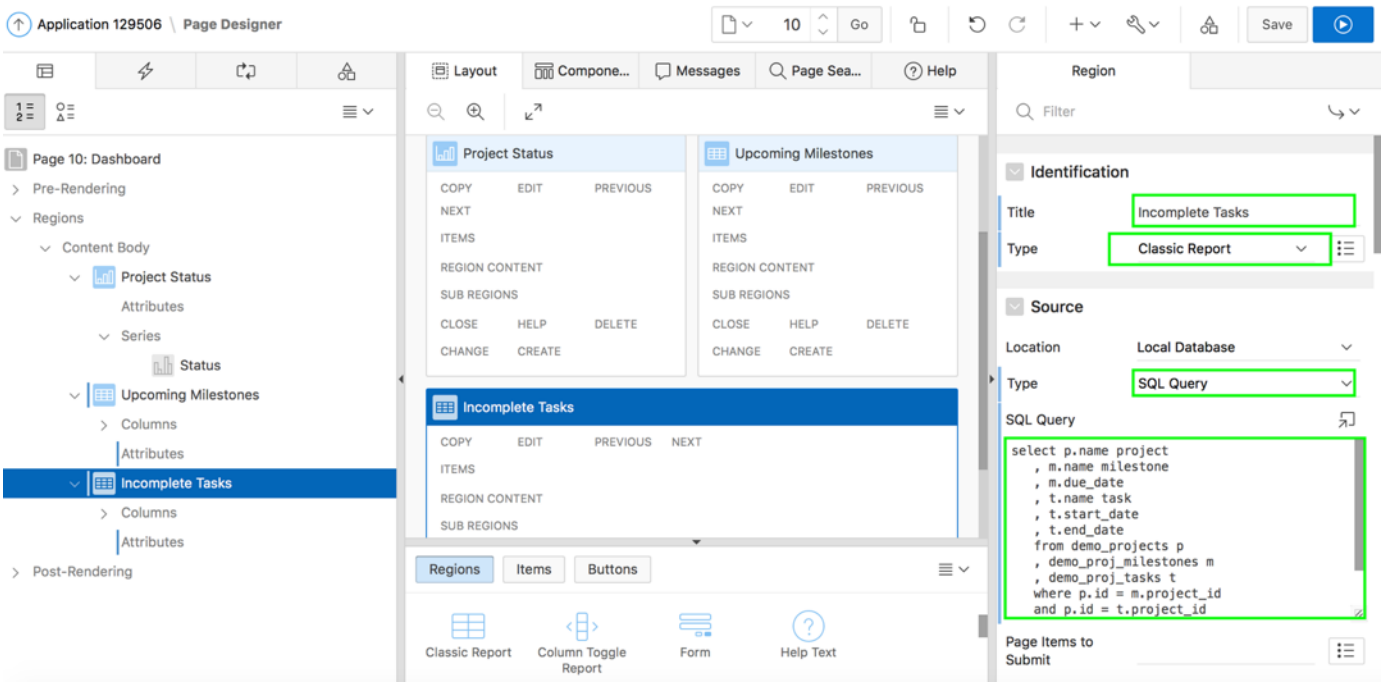
5. It would be very beneficial to also list the incomplete tasks on the dashboard page. To do so you will add another classic report at the bottom of the page.
In the Rendering tab, right-click **Content Body**, and select **Create Region**.



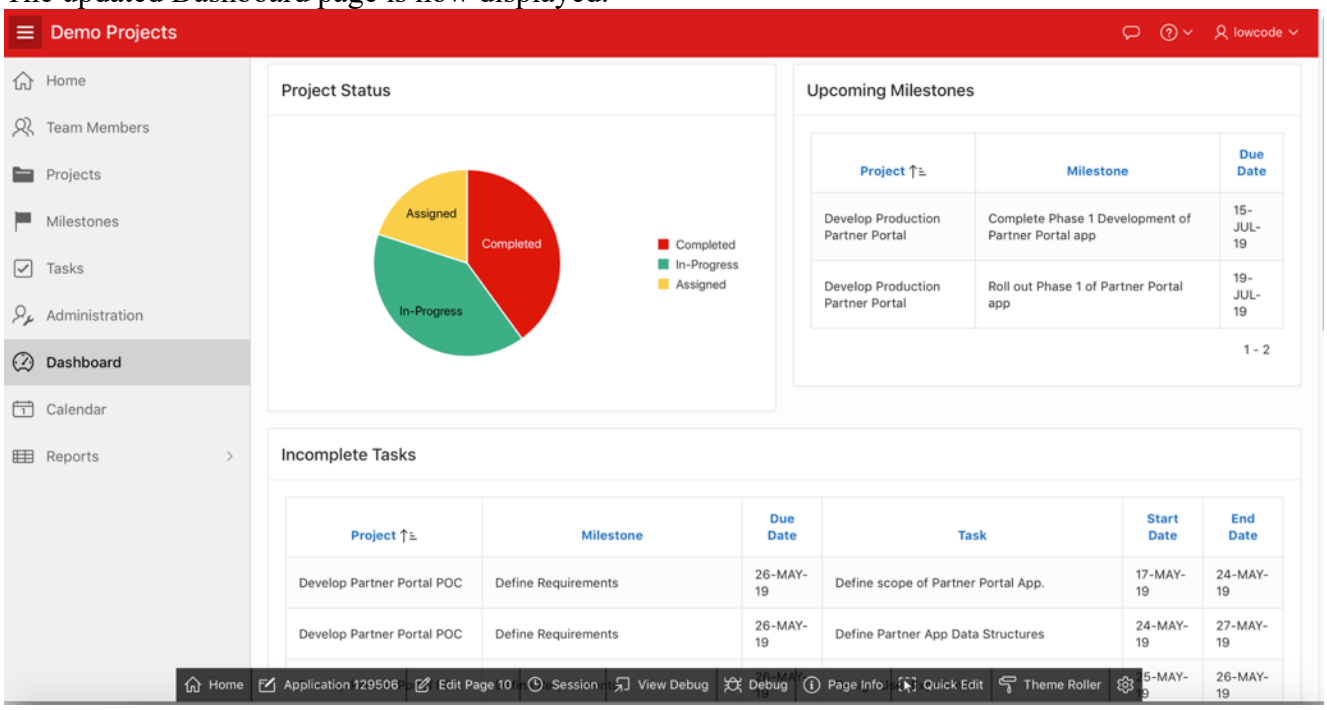
6. Update the report properties.
   In the Property Editor, update the following:
   - Identification: Title - enter Incomplete Tasks
   - Identification: Type - select **Classic Report**
   - Source: Type - select **SQL Query**
   - Source: SQL Query - enter the following code:

```
select p.name project
   , m.name milestone
   , m.due_date
   , t.name task
   , t.start_date
   , t.end_date
   from demo_projects p
   , demo_proj_milestones m
   , demo_proj_tasks t
   where p.id = m.project_id
   and p.id = t.project_id
   and t.milestone_id = m.id (+)
   and nvl(is_complete_yn, 'N') = 'N'
   and nvl(is_complete_yn, 'N') = 'N'
```

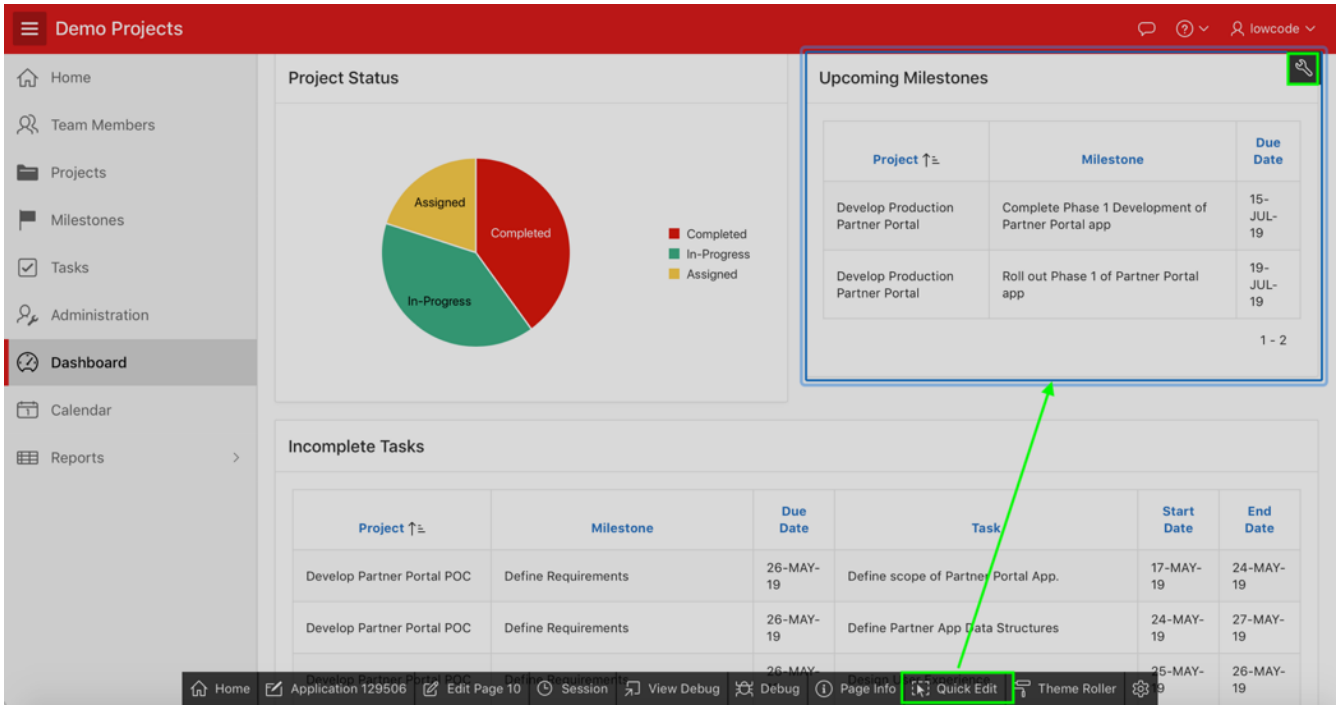
Note: Milestones are only optional for tasks, therefore, when joining the tasks and milestones tables it uses an outer join (+) so that all tasks are returned, even if they don't have an associated milestone.
Only incomplete tasks are to be displayed, irrespective of their start and end dates, so the Tasks column is_complete_yn is checked.



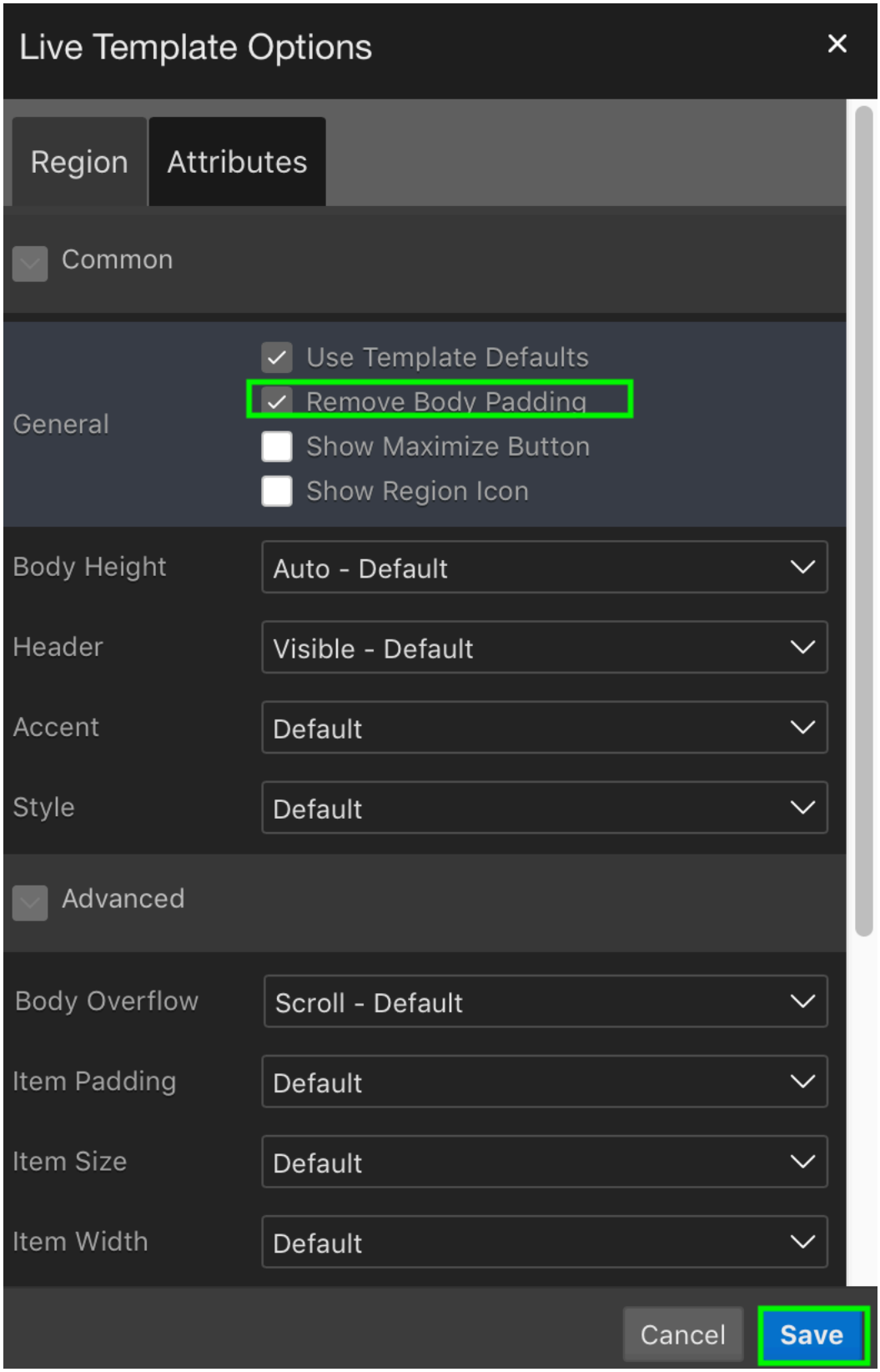
7. In the Page Designer, click **Save and Run Page**.
   The updated Dashboard page is now displayed.



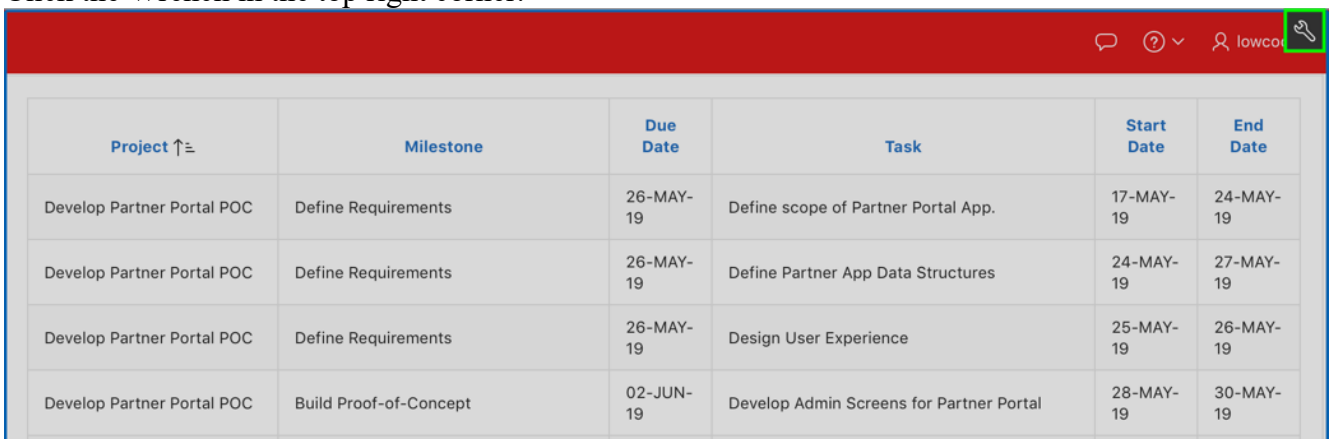
8. Both reports can readily be improved by removing body padding.
   In the Runtime Environment, within the Developer Toolbar, click **Quick Edit**.
   Hover over the **Upcoming Milestones** region until a blue box appears around the outside of the region.
   Click the Wrench in the top right corner.



9. In the Live Template options dialog, for General, select **Remove Body Padding**.
   For Header, select **Visible – Default**.
   Click **Save**.

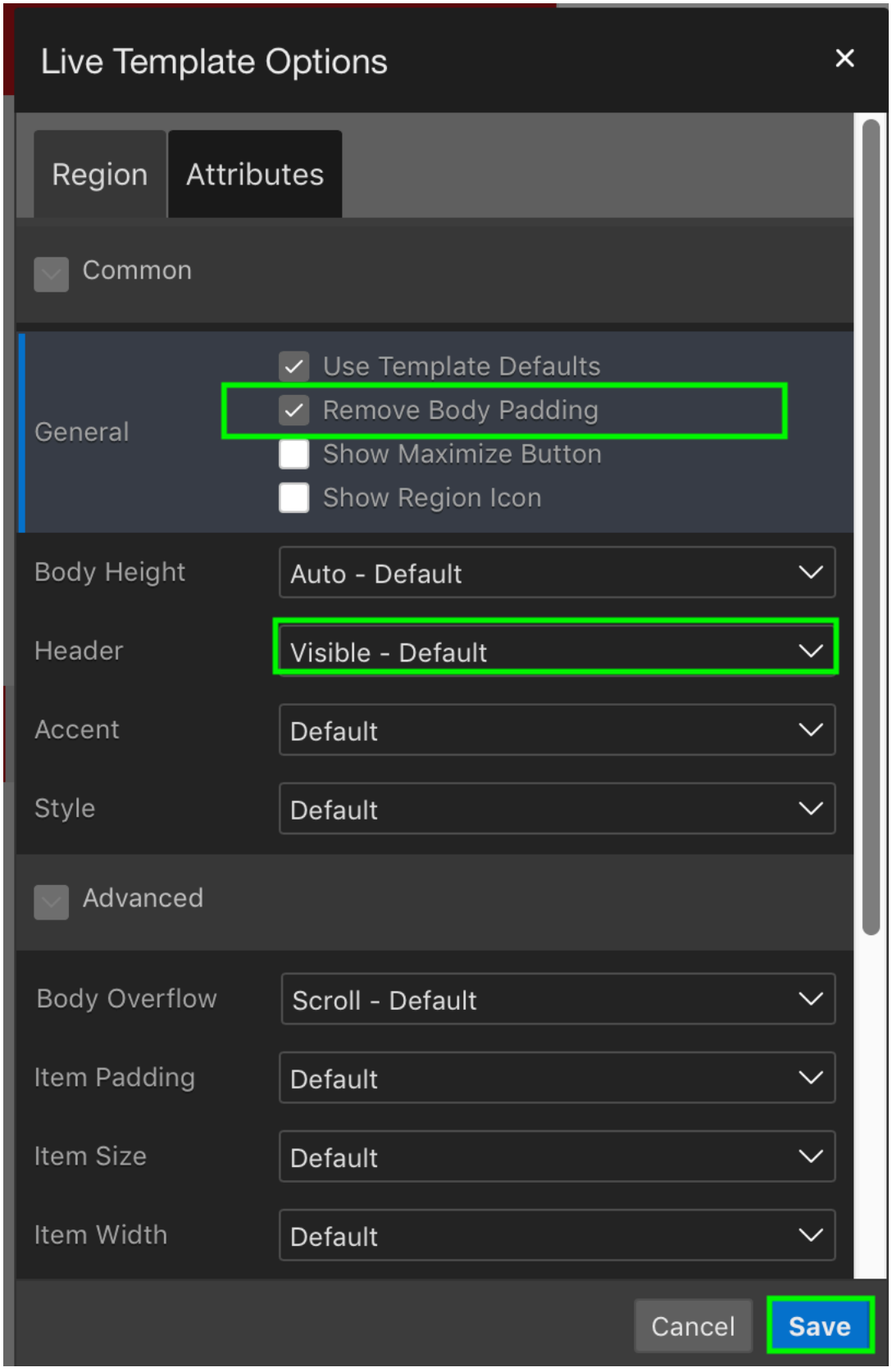

Live Template Options
Region
Attributes
Common
General
Use Template Defaults
Remove Body Padding
Show Maximize Button
Show Region Icon
Body Height
Auto - Default
Header
Visible - Default
Accent
Default
Style
Default
Advanced
Body Overflow
Scroll - Default
Item Padding
Default
Item Size
Default
Item Width
Default
Cancel
Save

10. In the Runtime Environment, within the Developer Toolbar, click **Quick Edit**.
    Hover over the **Incomplete Tasks** region until a blue box appears around the outside of the region.
    Click the Wrench in the top right corner.



Project	Milestone	Due Date	Task	Start Date	End Date
Develop Partner Portal POC	Define Requirements	26-MAY-19	Define scope of Partner Portal App.	17-MAY-19	24-MAY-19
Develop Partner Portal POC	Define Requirements	26-MAY-19	Define Partner App Data Structures	24-MAY-19	27-MAY-19
Develop Partner Portal POC	Define Requirements	26-MAY-19	Design User Experience	25-MAY-19	26-MAY-19
Develop Partner Portal POC	Build Proof-of-Concept	02-JUN-19	Develop Admin Screens for Partner Portal	28-MAY-19	30-MAY-19

11. In the Live Template options dialog, for General, select **Remove Body Padding**.
    For Header, select **Visible – Default**.
    Click **Save**.

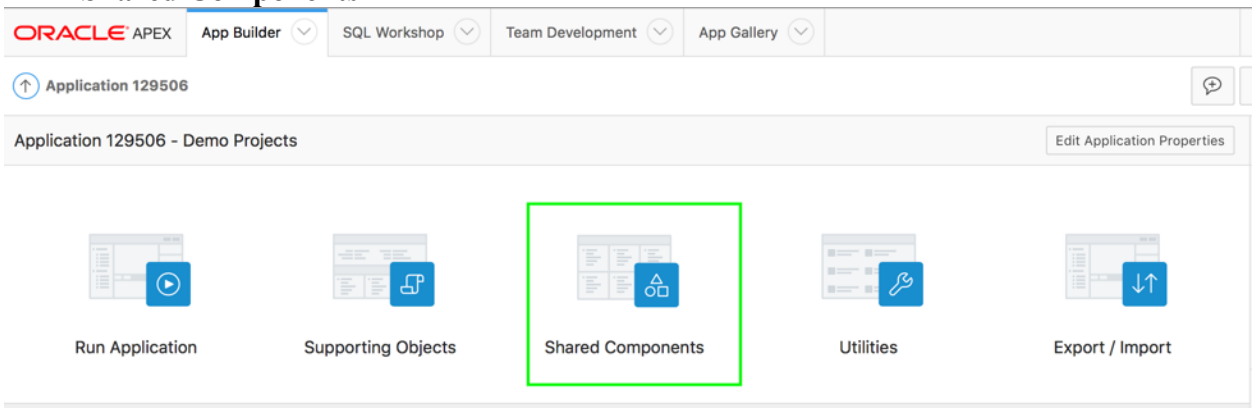

Live Template Options
Region
Attributes
Common
General
Use Template Defaults
Remove Body Padding
Show Maximize Button
Show Region Icon
Body Height
Auto - Default
Header
Visible - Default
Accent
Default
Style
Default
Advanced
Body Overflow
Scroll - Default
Item Padding
Default
Item Size
Default
Item Width
Default
Cancel
Save

HOL 15-8: Updating the Home Page

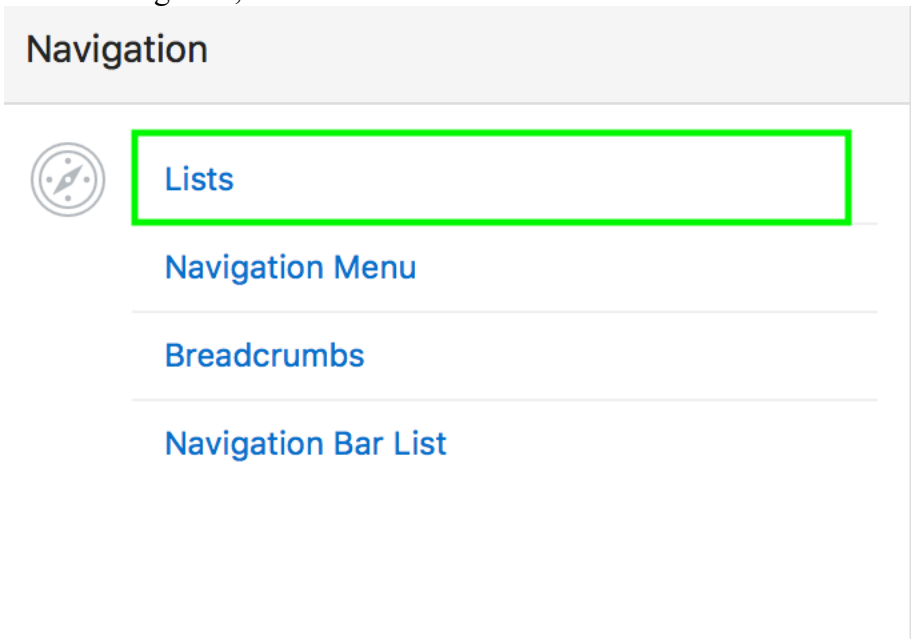
In this lab, you update the navigation menu of the Demo Projects application. You also add/update entries on the Home page.

1. In the Developer Toolbar, click **Application <*n*>**.

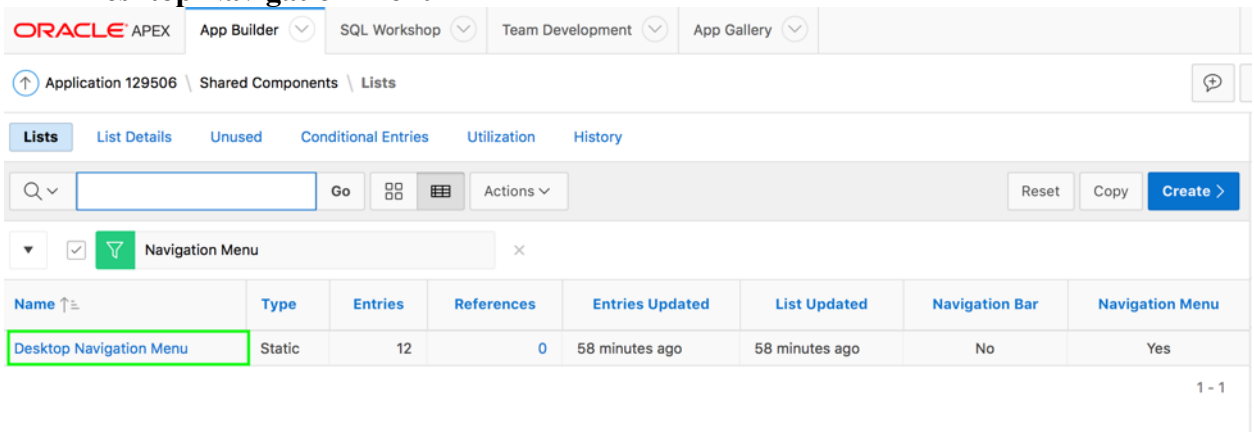
   Click **Shared Components**.



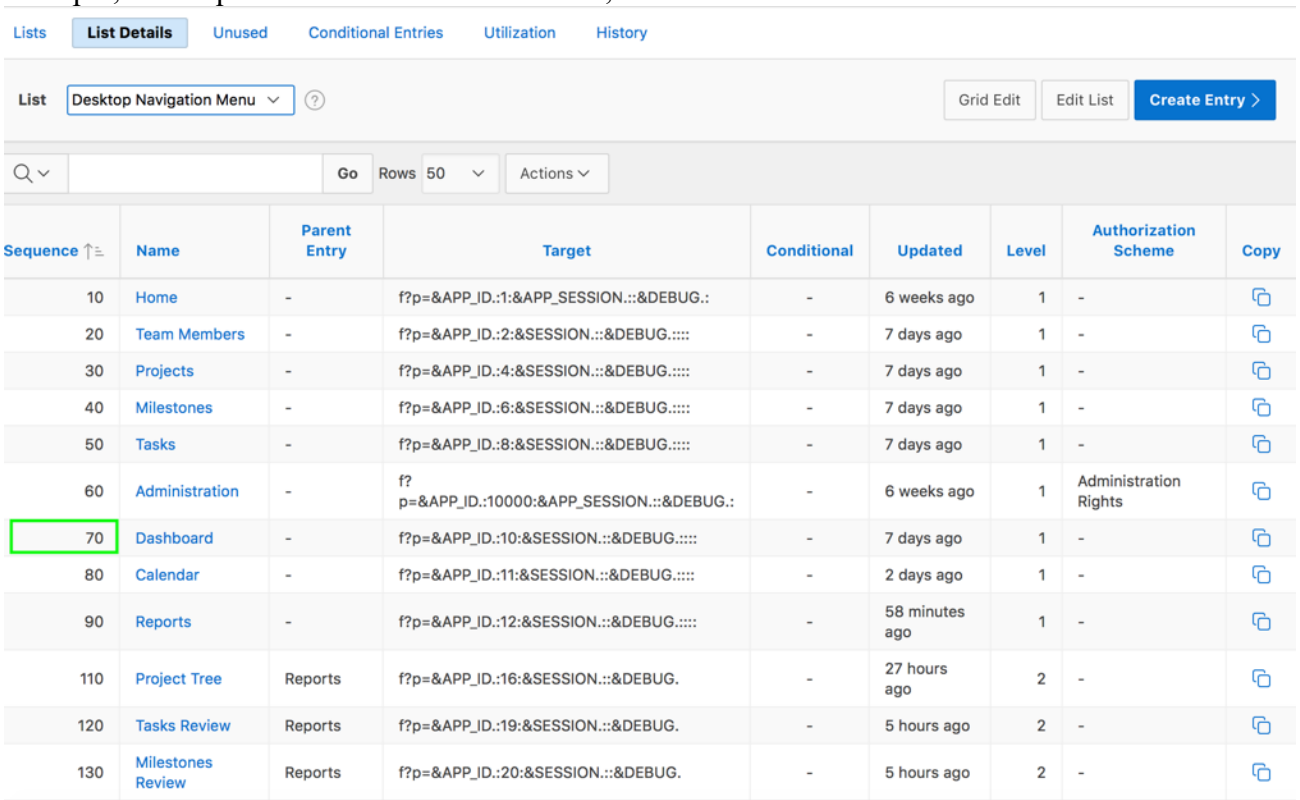
2. Under Navigation, click **Lists**.



3. Click **Desktop Navigation Menu**.

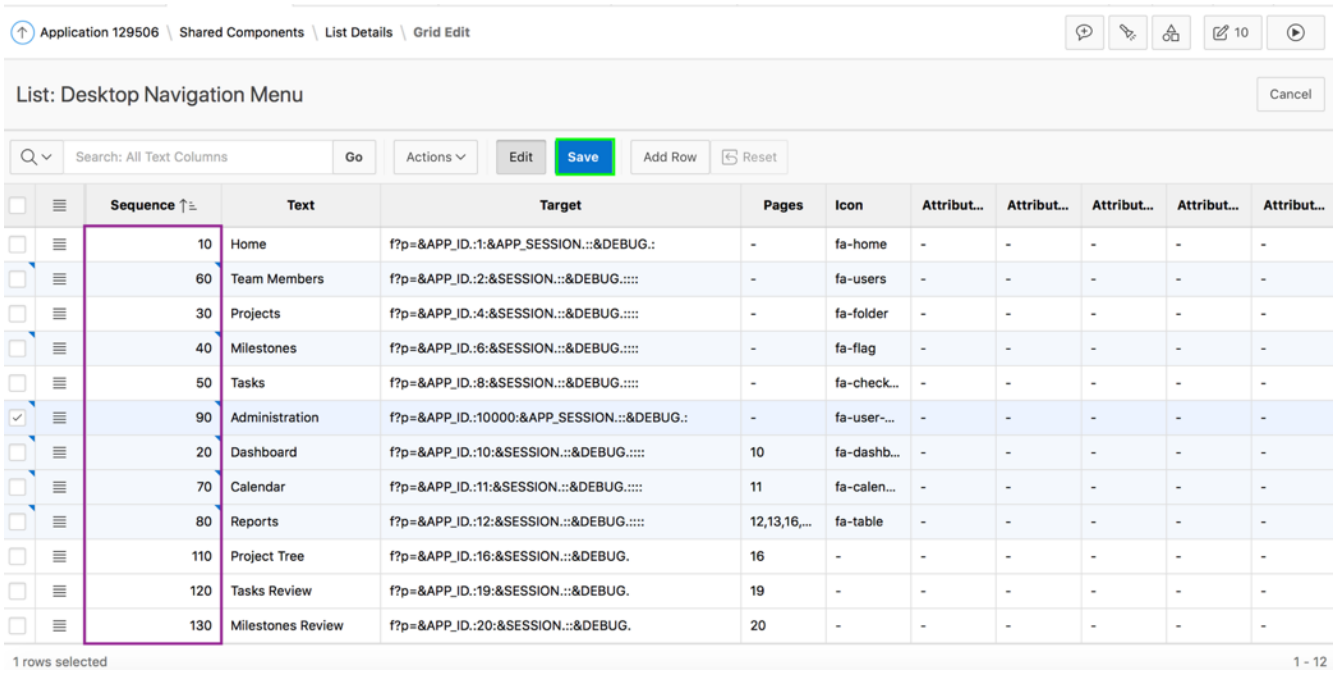


4. You now see an editable interactive grid.
Click in the Sequence field for each of the entries and adjust the sequence so that the order is: Home, Dashboard, Projects, Milestones, Tasks, Team Members, Calendar, and Reports. For example, the Sequence number for Home is 10, and for Dashboard is 20.

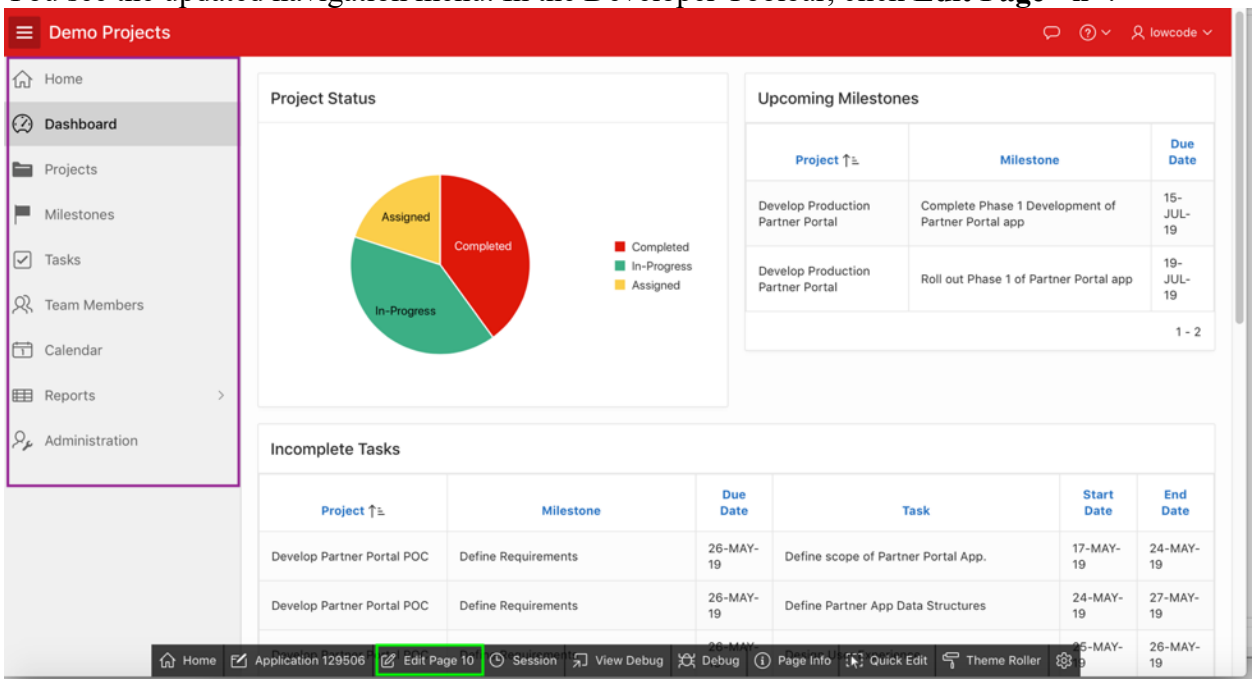


Sequence	Name	Parent Entry	Target	Conditional	Updated	Level	Authorization Scheme	Copy
10	Home	-	f?p=&APP_ID.:1:&APP_SESSION.::&DEBUG.:	-	6 weeks ago	1	-	
20	Team Members	-	f?p=&APP_ID.:2:&SESSION.::&DEBUG.:::	-	7 days ago	1	-	
30	Projects	-	f?p=&APP_ID.:4:&SESSION.::&DEBUG.:::	-	7 days ago	1	-	
40	Milestones	-	f?p=&APP_ID.:6:&SESSION.::&DEBUG.:::	-	7 days ago	1	-	
50	Tasks	-	f?p=&APP_ID.:8:&SESSION.::&DEBUG.:::	-	7 days ago	1	-	
60	Administration	-	f?p=&APP_ID.:10000:&APP_SESSION.::&DEBUG.:	-	6 weeks ago	1	Administration Rights	
70	Dashboard	-	f?p=&APP_ID.:10:&SESSION.::&DEBUG.:::	-	7 days ago	1	-	
80	Calendar	-	f?p=&APP_ID.:11:&SESSION.::&DEBUG.:::	-	2 days ago	1	-	
90	Reports	-	f?p=&APP_ID.:12:&SESSION.::&DEBUG.:::	-	58 minutes ago	1	-	
110	Project Tree	Reports	f?p=&APP_ID.:16:&SESSION.::&DEBUG.	-	27 hours ago	2	-	
120	Tasks Review	Reports	f?p=&APP_ID.:19:&SESSION.::&DEBUG.	-	5 hours ago	2	-	
130	Milestones Review	Reports	f?p=&APP_ID.:20:&SESSION.::&DEBUG.	-	5 hours ago	2	-	

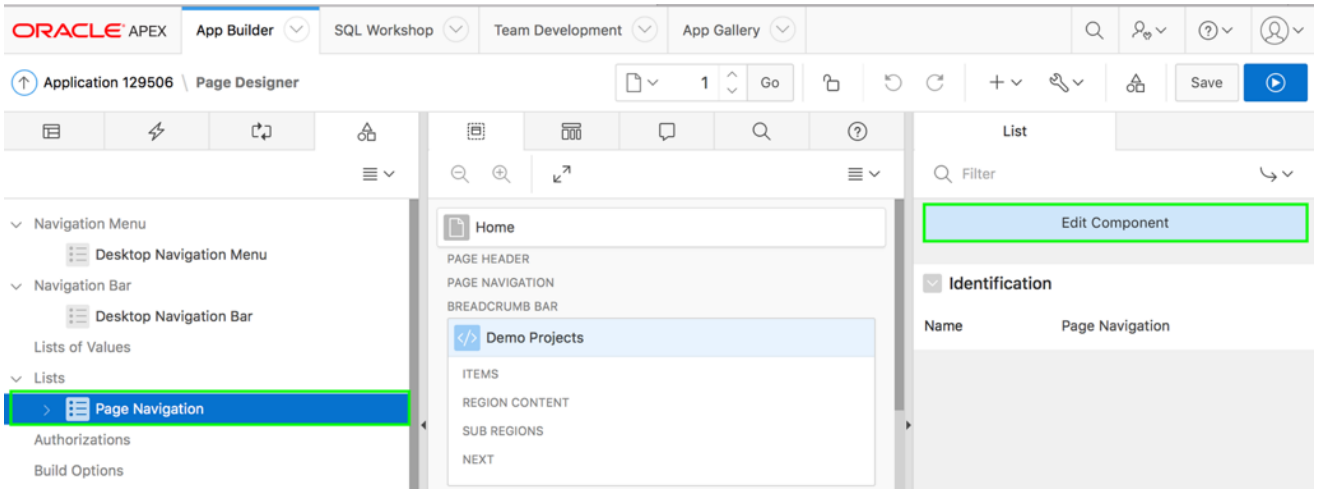
5. Adjust the sequence numbers for all of the pages and the screen should look like this:
Then, click **Save**.



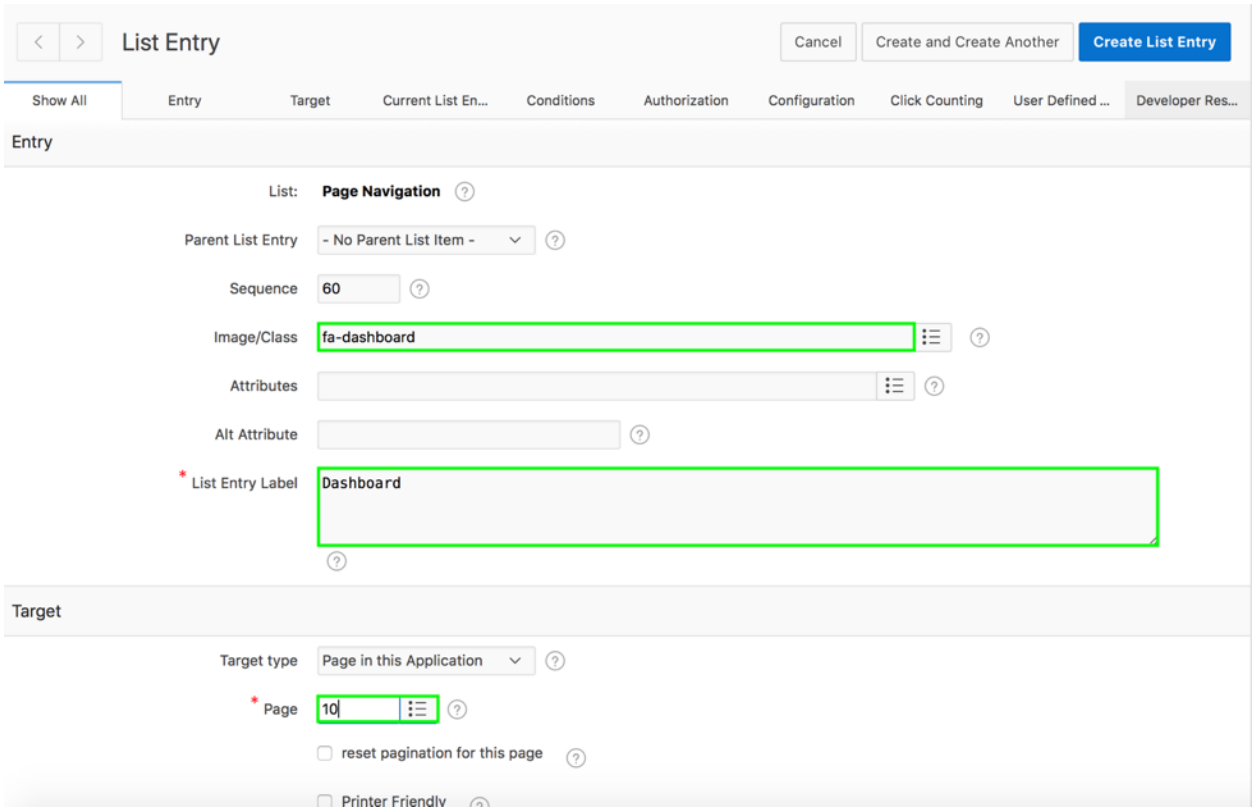
6. Now, run the page.
You see the updated navigation menu. In the Developer Toolbar, click **Edit Page <n>**.



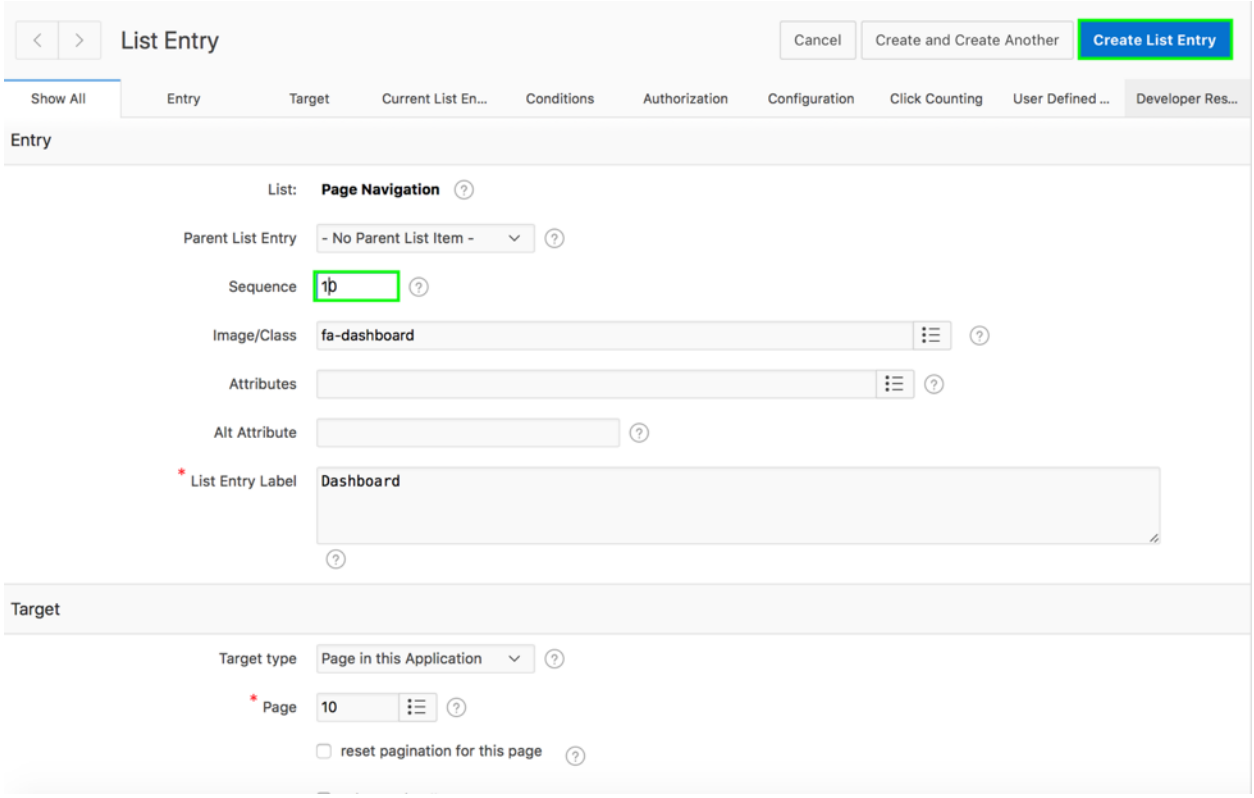
7. You want to adjust the entries on the Home page. In the Page Designer, navigate to **Page 1**.
Under Rendering, click **Page Shared Components**.
Expand **Lists** and select **Page Navigation**.
In the property editor, click **Edit Component**.



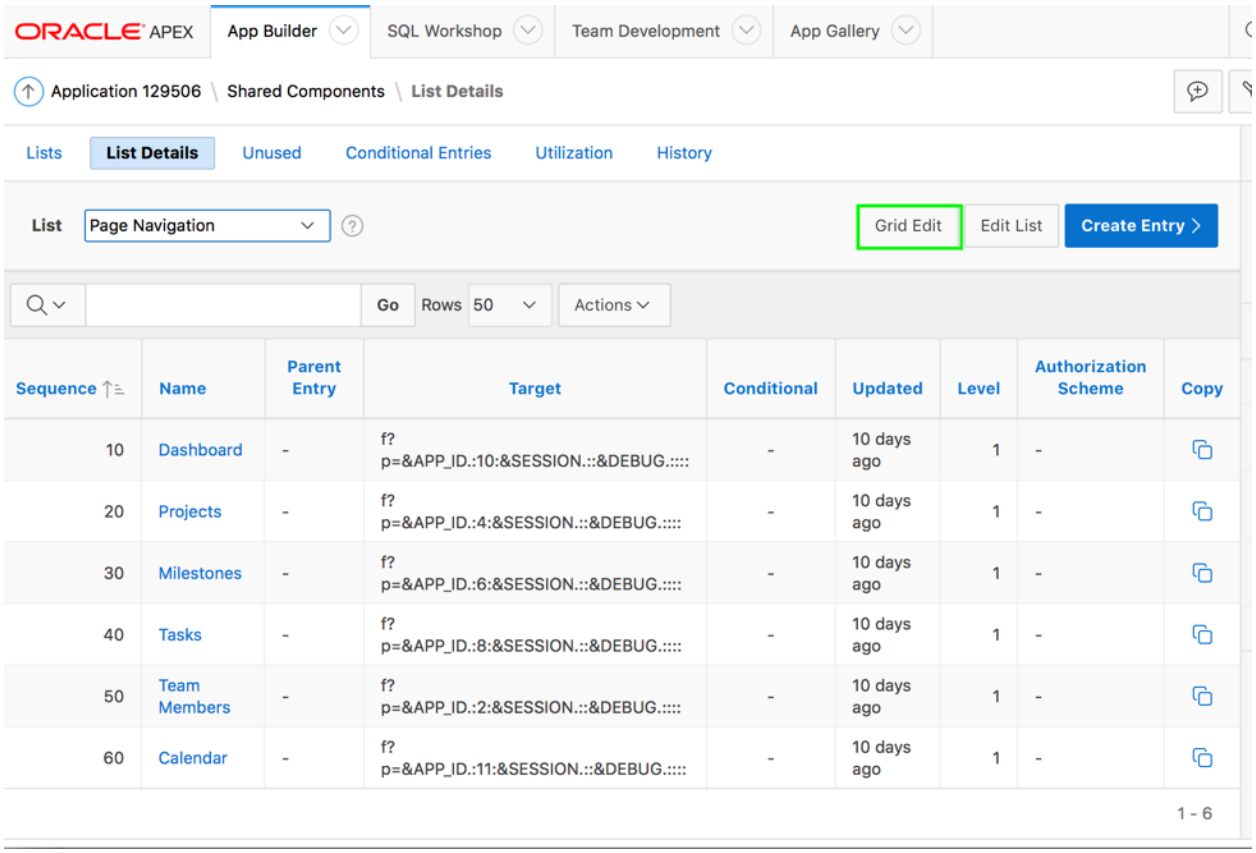
8. Click **Create Entry**.
   Under Entry, for Image/Class, enter **fa-dashboard** and for List Entry Label, enter **Dashboard**.
   Under Target, for Page, select the Dashboard page number. For example, Page 10 in the screenshot below.



9. For Sequence, enter **10** and click **Create List Entry**.

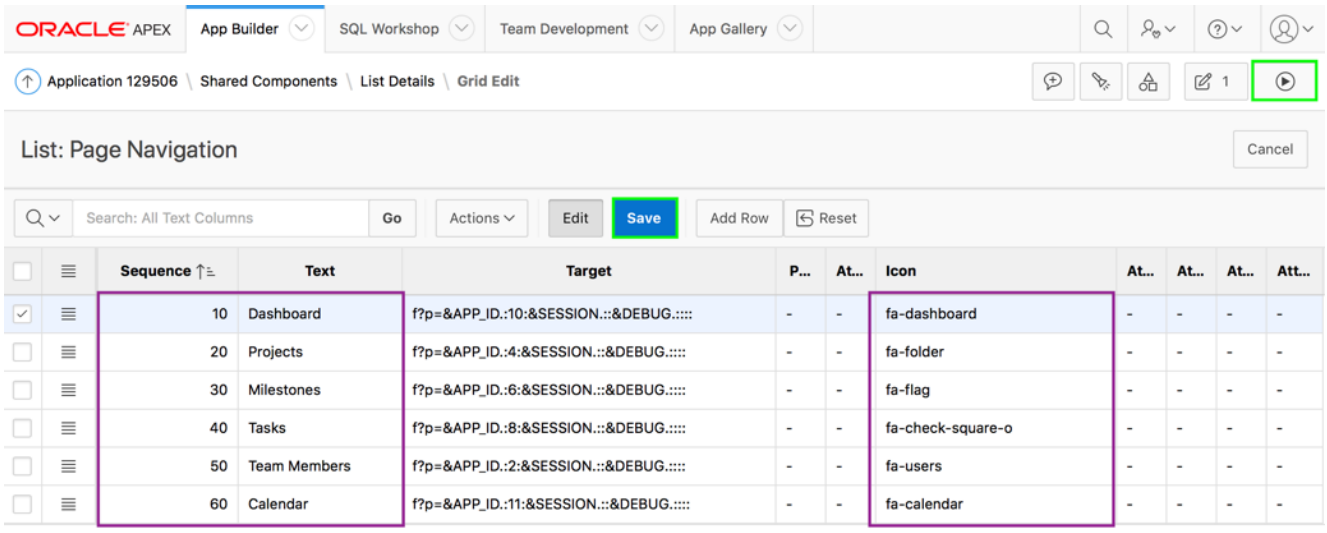


10. Similarly create/update list entries for other pages that you want to display on the Home page.
    You want to have the Dashboard, Projects, Milestones, Tasks, Team Members, and Calendar page in sequence. Select the image/class as appropriate for each of these pages. Once done, click **Grid Edit**.

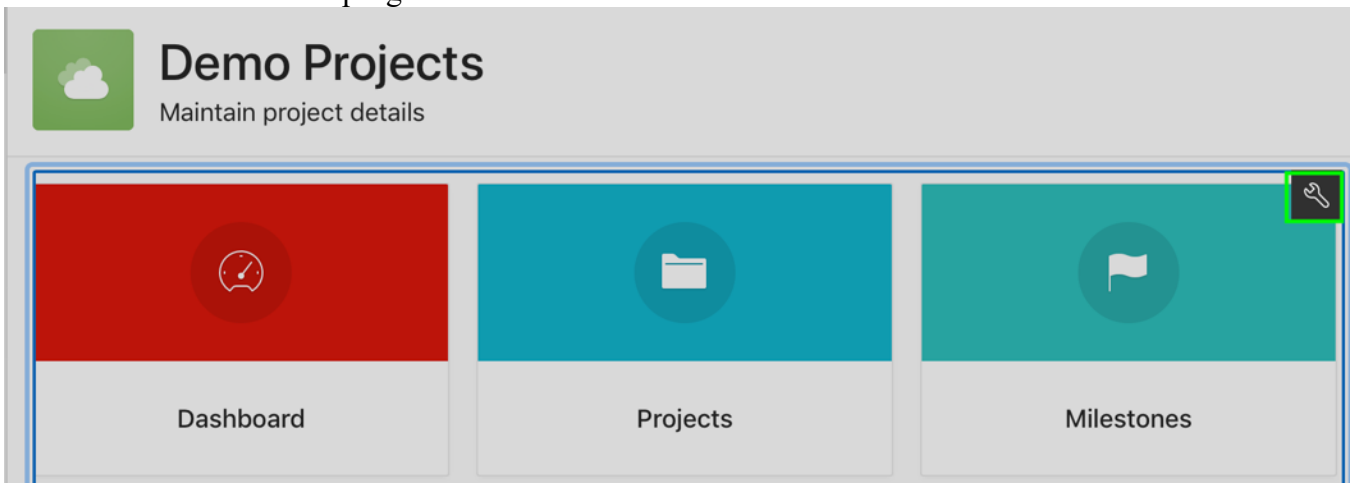


11. In this editable interactive grid, verify the sequence of the pages and also the image/class. If you have any adjustments, click the corresponding field and then click **Save**.

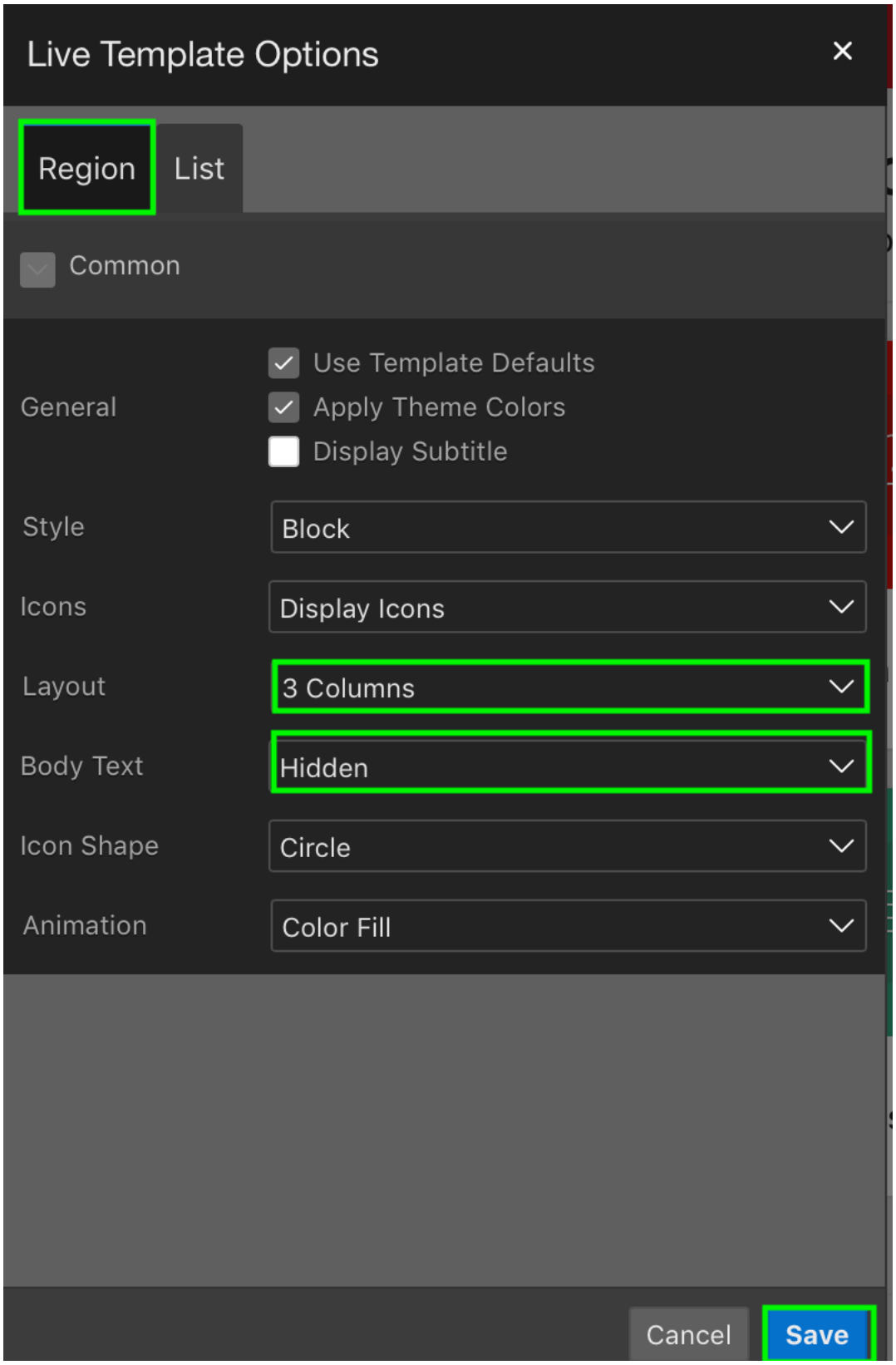
    Click **Run Page 1**.



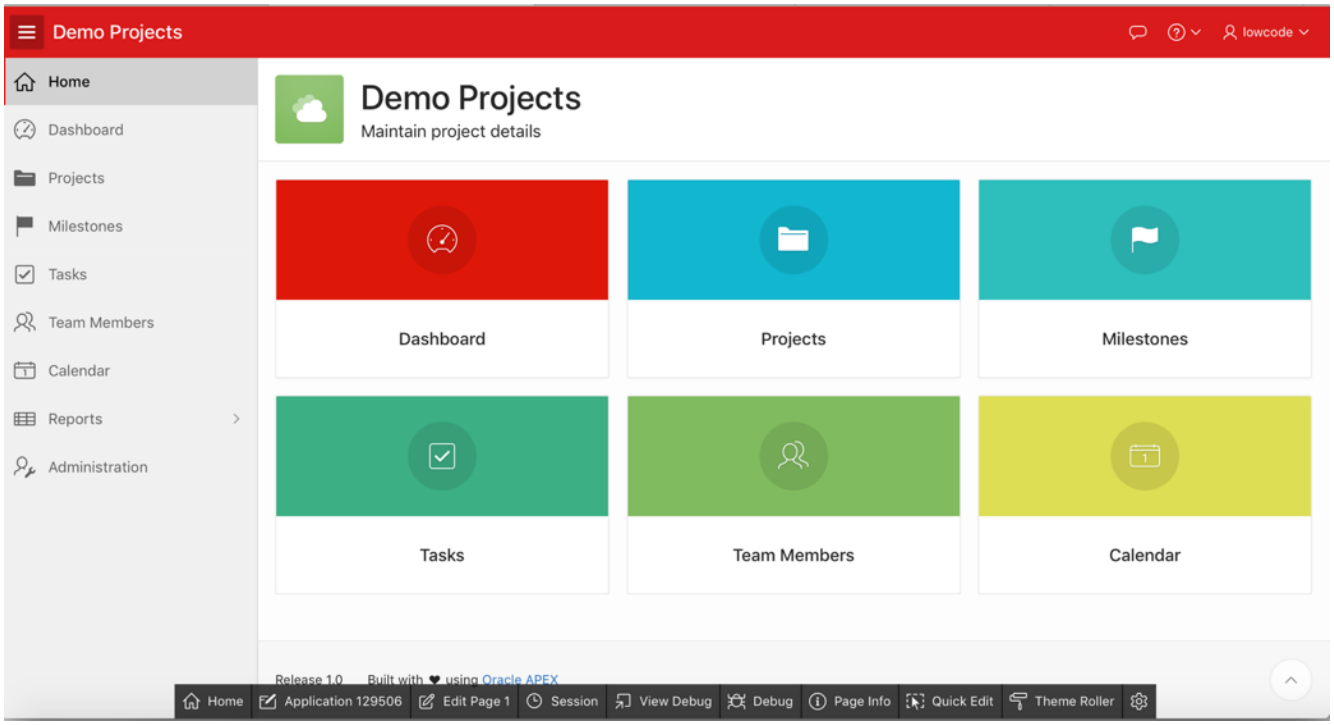
12. In the Developer Toolbar, click **Quick Edit**. Hover over the **List** region (under the Demo Projects breadcrumb region) until a blue box appears around the outside of the region. Click the Wrench in the top right corner.



13. In the Live Templates dialog, click **Region**. For Layout, select **3 Columns** and for Body Text, select **Hidden**.
Click **Save**.


Live Template Options
Region
List
Common
General
Use Template Defaults
Apply Theme Colors
Display Subtitle
Style
Block
Icons
Display Icons
Layout
3 Columns
Body Text
Hidden
Icon Shape
Circle
Animation
Color Fill
Cancel
Save

14. Now you see the updated Home page.







Oracle is committed to developing practices and products that help protect the environment

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. Licensed under the Creative Commons Attribution 4.0 International License as shown at https://creativecommons.org/licenses/by/4.0/legalcode.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document.

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

Hardware and Software, Engineered to Work Together