

Lab 5 : BYOL [HOL2752]

Lab 5 : BYOL [HOL2752]

Introduction

Three Ways to work with Process

Process Setup

Step 1: Create a Process Alias

Step 2 : Start the process from a Visual Builder Action

Step 3: Show Tasks and task actions

Introduction

In this lab we will be connecting products to an order process that uses Oracle Process Cloud Service. It is necessary to use an instance of Visual Builder that is shipped within Oracle Integration Cloud for this lab. Additionally, this lab uses a prebuilt Process and Integration with which Visual Builder interacts. Creating the Process and Integration from scratch is beyond the scope of this lab.

Three Ways to work with Process

There are three primary ways to interact with Oracle Process from within Visual Builder.

1. Process Service APIs

- Process Service APIs are available as REST and can be called directly from within Visual Builder. You may setup service connections and use Service Data Providers and/or call APIs from within custom Javascript to populate Array Data Providers. When a Process alias is created within Visual Builder some doc of the most commonly used APIs along with code samples is provided

2. Visual Builder Process actions and objects

- Visual Builder includes a set of actions and objects for easier access to Process. Several common actions are presented as actions to be added into UI workflow.

Process



Get
Deployed
Process



Get
Deployed
Process
Collection



Get Process
Instance



Get Process
Instance
Collection



Get Task



Get Task
Collection



Perform
Task



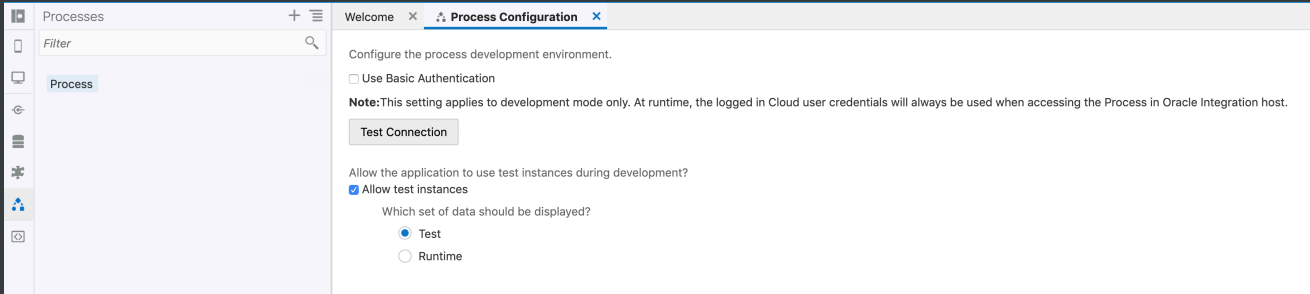
Start
Process

3. Process UI Components

- Process includes a set of UI components that can be used to add more information about your active processes and instance.

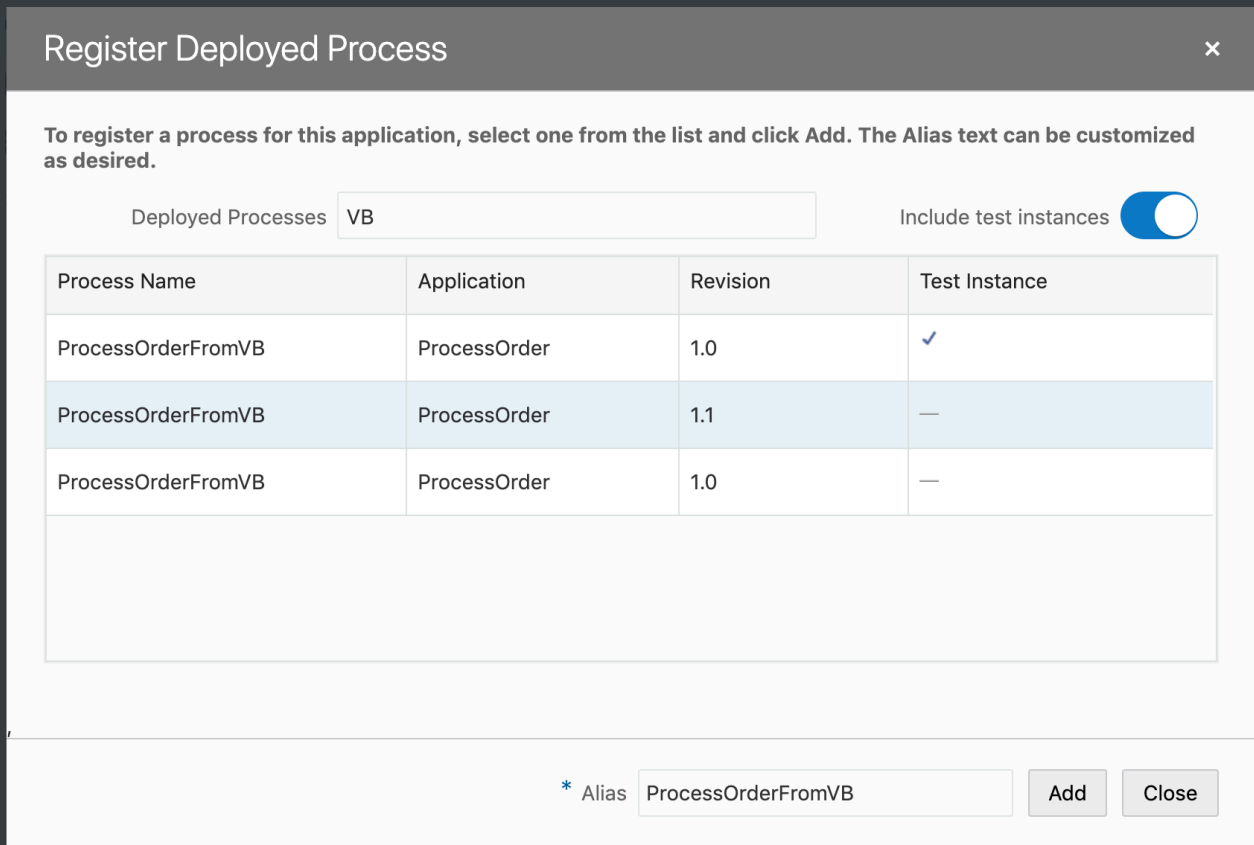
Process Setup

Step 1: Create a Process Alias

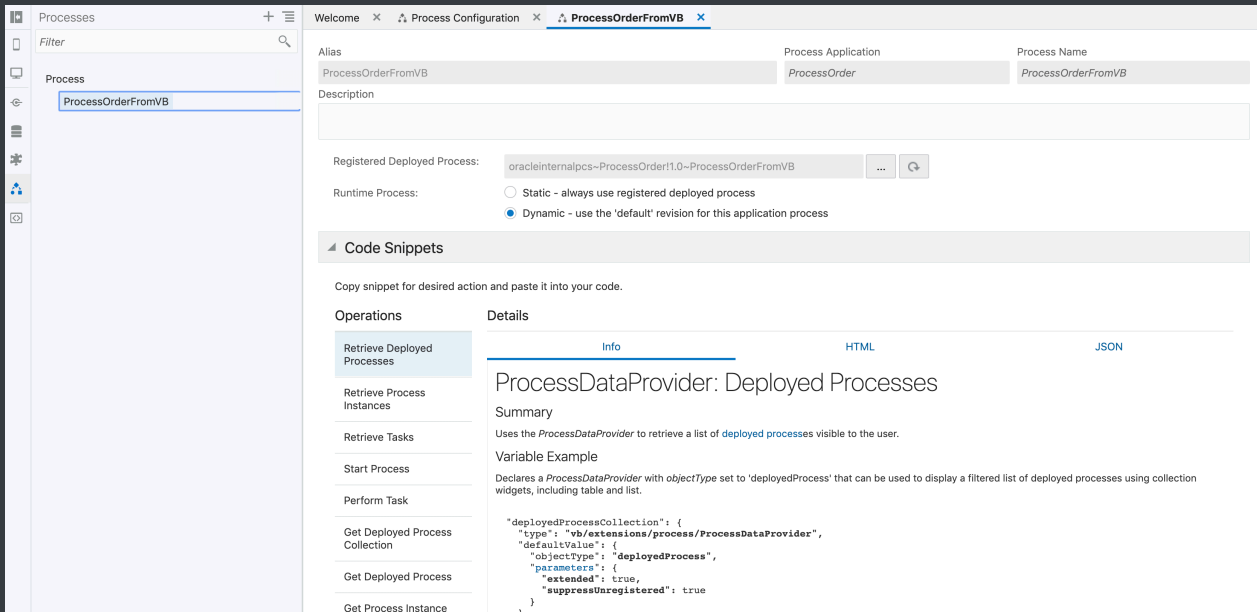


We need to first create an alias for our process.

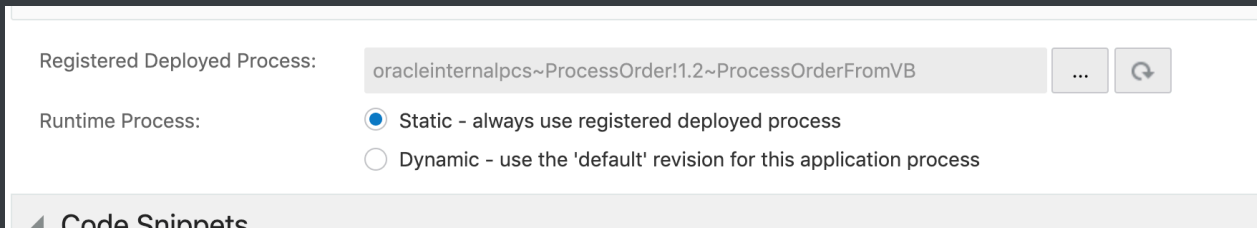
1. Select the Process tab from the left most navbar.
2. (Optional) Test Connection to ensure the Process server is configured for your logged in user account.
3. Click the + icon to select a Process to create an alias. Enter 'vb' into the search bar of that dialog to find the process for this lab and select the top ProcessOrderFromVB Revision 1.2 process and click the Add button.



4. Select the alias to see detailed information about it including code snippets for available APIs.

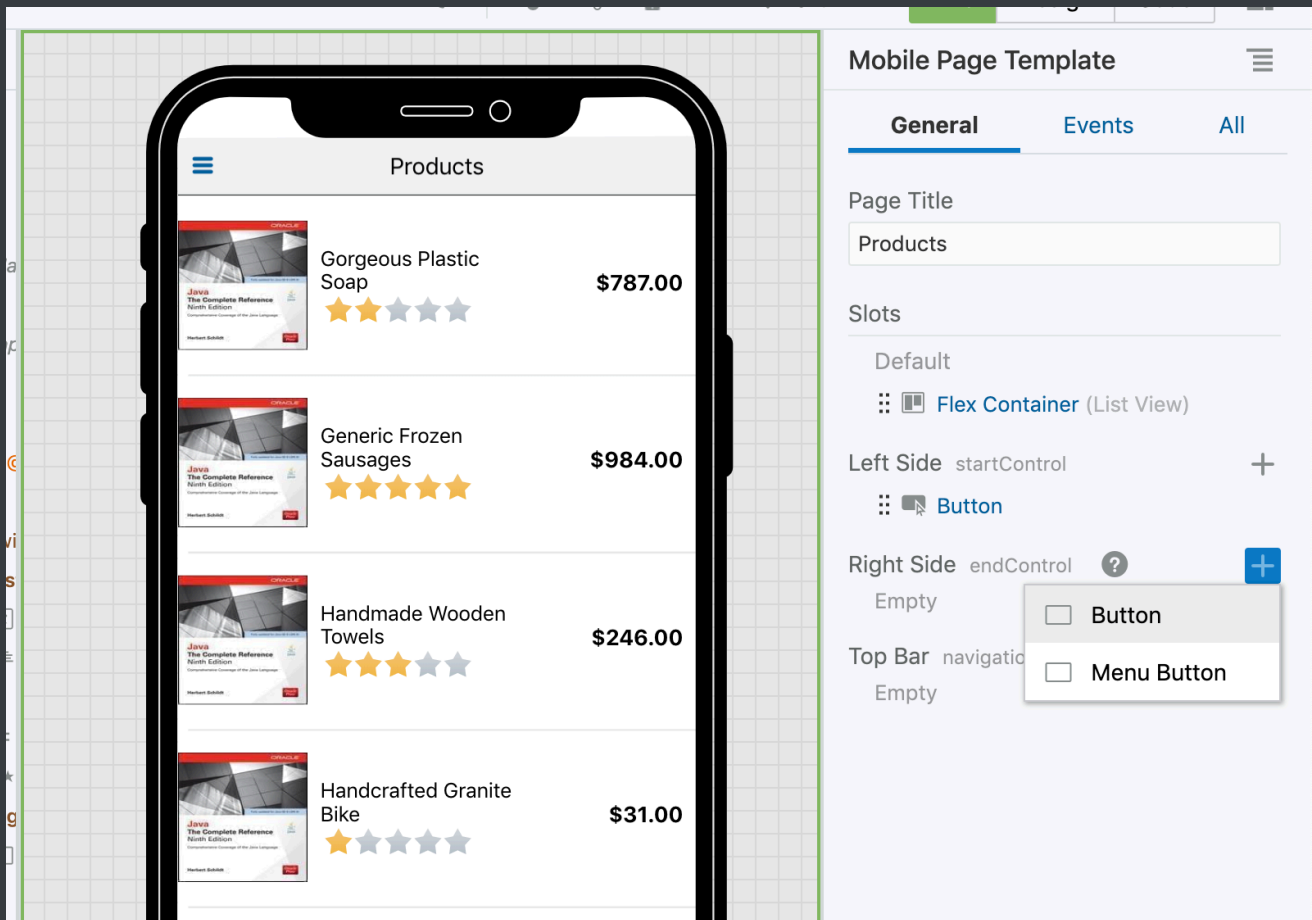


5. Select Static for the Runtime Process

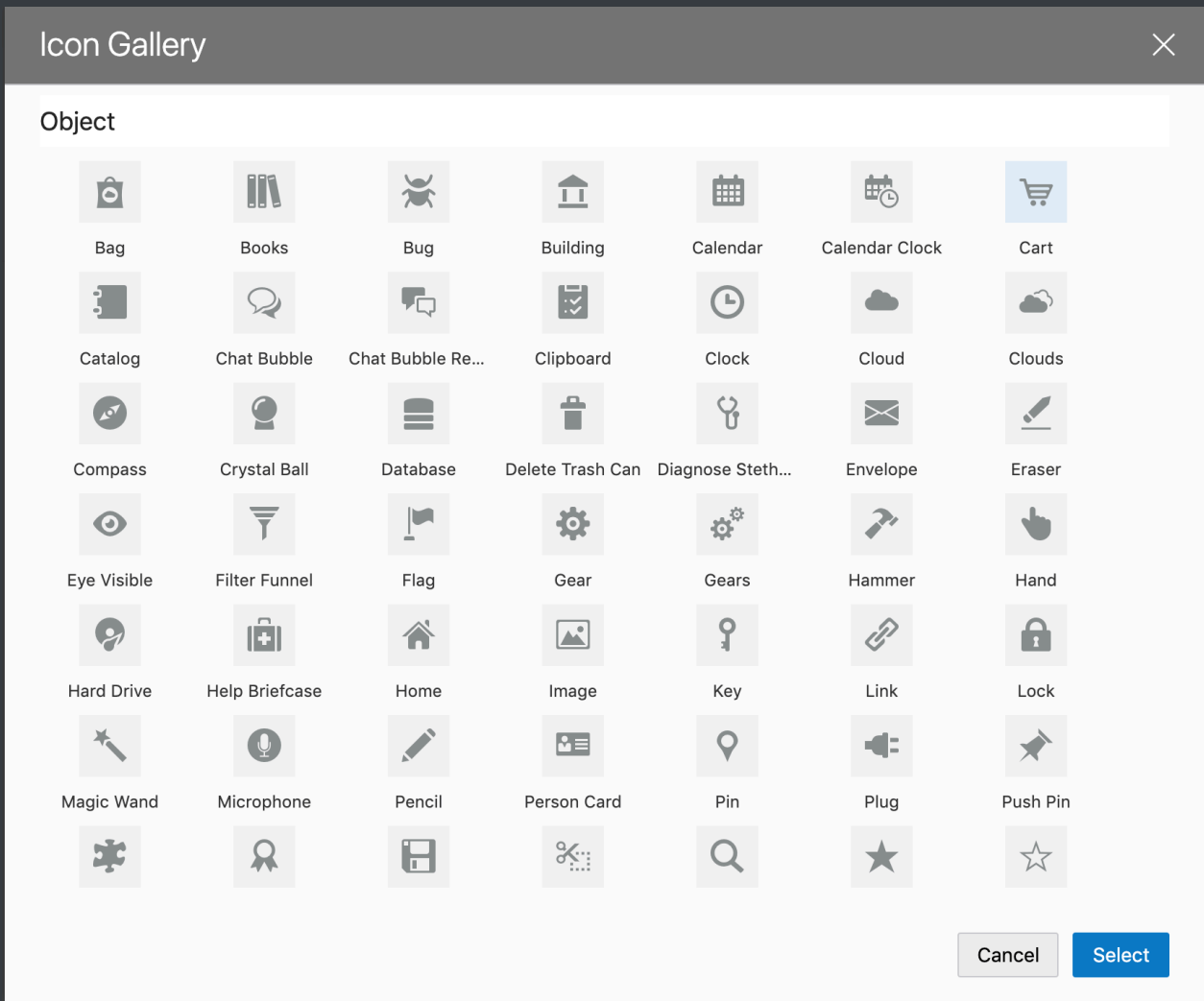


Step 2 : Start the process from a Visual Builder Action

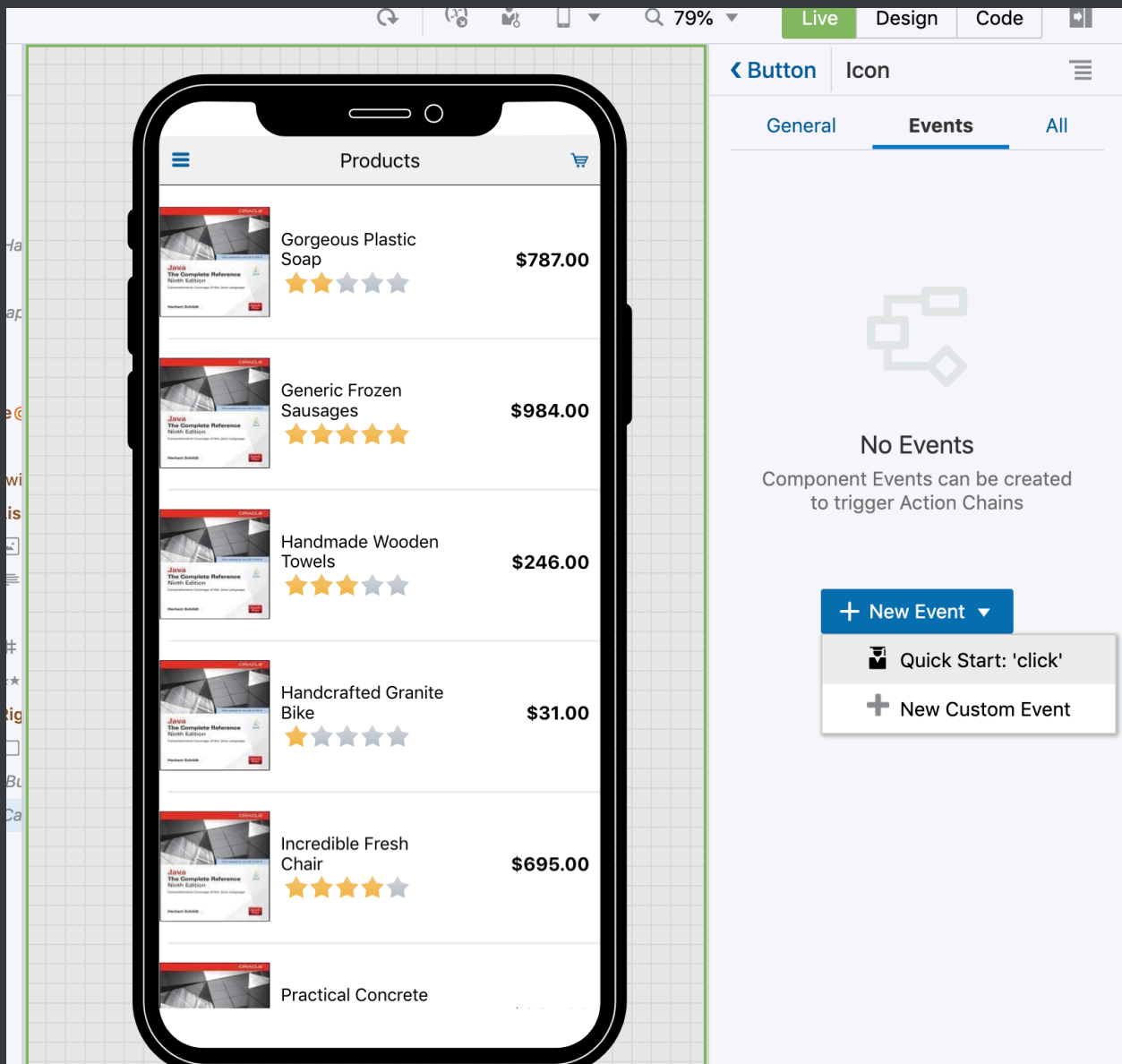
Now, we'll create a user interface action from which to kick off a process. On the Product page of the mobile application, we'll add a button to take an order.



1. Open the product-start page and go to the General tab of the object inspector.
2. Click on the Right Side '+' button and selecton Button
3. Click on the recently added '+' button on the Products page and change the icon to a cart within the Icon Gallery



4. Select the Cart icon on the Product page and move to the Events tab in the object inspector and click the New Event button and select 'Quick Start: 'click'



5. Now, you will be in the Action Chain editor. Scroll down the Actions palette to the Process actions.
6. Drag the Start Process action to the + sign after Start in the action chain editor
7. Click Select Process in the Action Chain object inspector and select Recieve Message from the Alias.

8. Notice the process is expecting an Input Parameter called productNumber.

Process Interface *	Select
<hr/>	
alias/ProcessOrderFromVB/start	
Input Parameters	Assign
<hr/>	
# productNumber *	NOT MAPPED

9. This was designed within Process Cloud as an input argument.

Configure ✕

How do you want to define your interface?

*** Operation Name**

start

Arguments Definition

Name	Type	
productNumber	int ▼	✕

10. So, we'll map the current selected list item product number as the input argument. However, we need to capture that value first when the row is selected. To do that we will create an event by selecting the the list item from the listview and creating an event for the 'click' event.

The image shows a mobile application interface on the left and a configuration panel on the right. The mobile app displays a list of products under the heading 'Products'. Each product entry includes a small image, the product name, a star rating, and a price. The products listed are:

- Gorgeous Plastic Soap: \$787.00 (3 stars)
- Generic Frozen Sausages: \$984.00 (5 stars)
- Handmade Wooden Towels: \$246.00 (4 stars)
- Handcrafted Granite Bike: \$31.00 (2 stars)
- Incredible Fresh Chair: \$695.00 (4 stars)
- Practical Concrete: (no price shown)

The configuration panel on the right is titled 'List Item' and has tabs for 'General', 'Events', and 'All'. The 'Events' tab is active, showing a configuration for a 'click' event. The event is named 'click → ListItemClickChain'. The configuration includes:

- Event *: click
- Action Chain *: ListItemClickChain
- Input Parameters: Assign
- Parameters:
 - * detail: mapped
 - * key: mapped
 - * current: mapped

Below the configuration panel, there is a '+ New Event' button and a 'Quick Start: 'click'' option.

11. Within that event we will capture the `$chain.variables.current.data.id` value by assigning it to a local Page variable called `productNumber`. You will need to create the variable for the page (as type int) called `Product Number` first. This variable will now get updated each time a new row within the listbox is selected.

Assign Variables: assignVariables1

Sources

- Action Chain
 - Variables
 - current
 - detail
 - key
 - Results
 - Page
 - Variables
 - productNumber
 - productsListSDP
 - endpoint
 - fetchChainId
 - filterCriterion

Target

- Action Chain
 - current
 - detail
 - key
- Page
 - productNumber
 - productsListSDP
- Flow
 - No flow variables
- Application
 - No application variables

\$page.variables.productNumber

Reset Target toDefault

1 \$chain.variables.current.data.id

Expression

Static content

1:33

Cancel Save

12. Now that we have that variable assigned, we can map it to our input argument.

Start Process: startProcess1

Sources

- Action Chain
 - Variables
 - detail
 - Results
 - Page
 - Variables
 - productNumber
 - productsListSDP
 - System
 - Flow
 - System
 - Application
 - System

Target

- Parameters
 - productNumber

productNumber

1 \$page.variables.productNumber

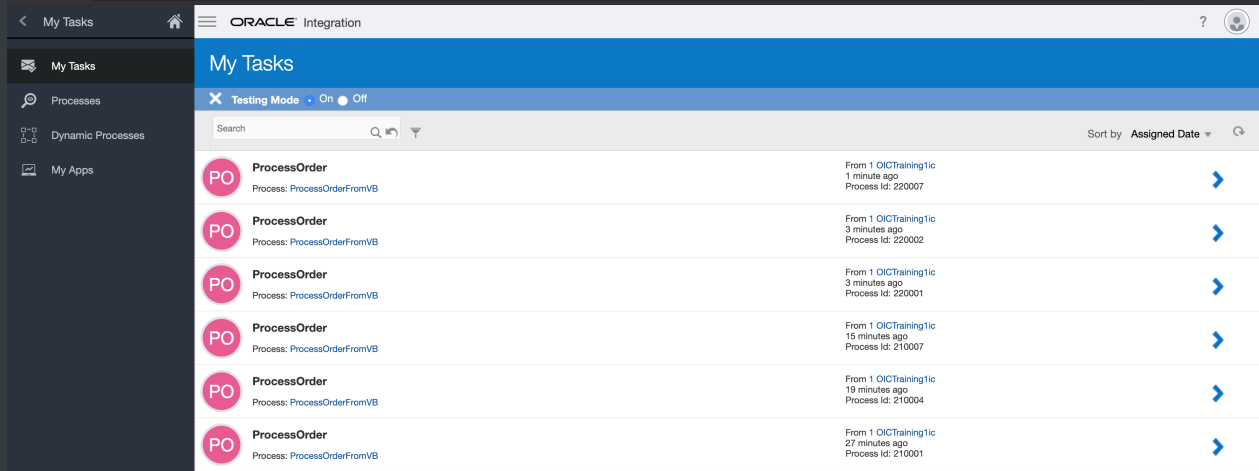
Expression

Static content

1:1

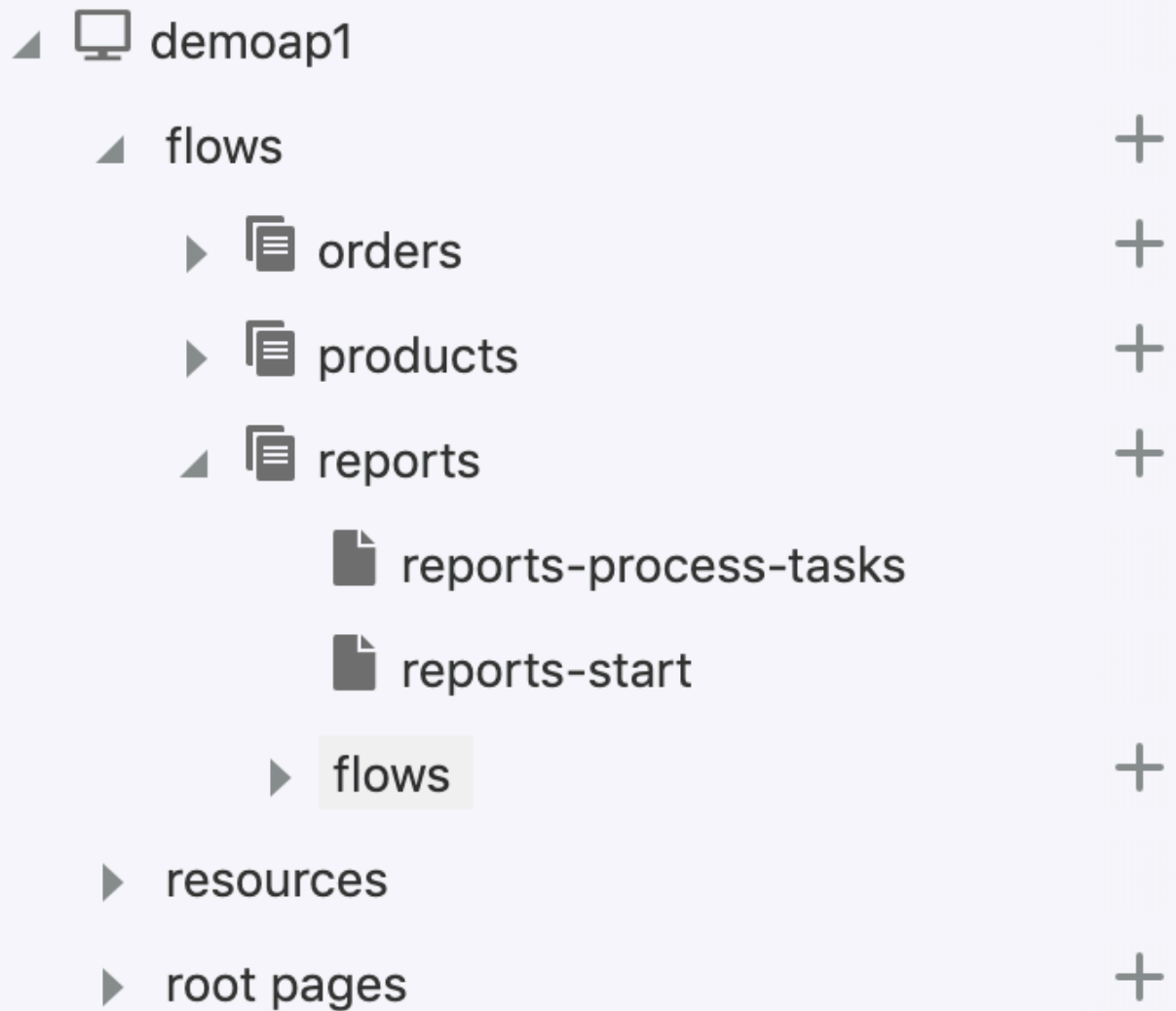
Cancel Save

13. Run the app and click the cart icon to kick off the process. You can verify that the process was properly started within the Process Cloud by looking at your open tasks.



Step 3: Show Tasks and task actions

1. We can also view the tasks within Visual Builder using a Table component. Go to your web app and add a new page under the Reports flow called reports-process-tasks.



2. On that page, drag a Table component and in the Object Inspector go to the Quick Start tab and Add Data. Select the 'task' service call under the Process Objects. Select some fields to display such as id, label, and priority. Click Finish.

Add Data

Cancel

Choose the source of your data

Business Objects



Orders



Products

Process Objects



deployed
process



process
instance



task

3. Back in the designer, select Add Task Actions and select the defaults by hitting Finish.
4. Now, set the designer to Live mode and select a row from the table. Then select the action combobox, you should see "Accept/Reject" displayed. You may complete the action at this point.

My Application

Choose an Action	Submit
Choose an Action	label
APPROVE	ProcessOrder
REJECT	ProcessOrder
201217	ProcessOrder
201219	ProcessOrder
201222	ProcessOrder
201223	ProcessOrder
201225	ProcessOrder

Table

General Data Events All

- Add Data**
Map the table to a data source, so we can populate it with data.
- Add Create Page**
Add a new action and a page with a form for creating a row.
- Add Edit Page**
Add a new action and a page updating the selected record.
- Add Detail Page**
Add a new action and a page with the record details.