VBCS Lab
Master-Detail Pattern

Introduction

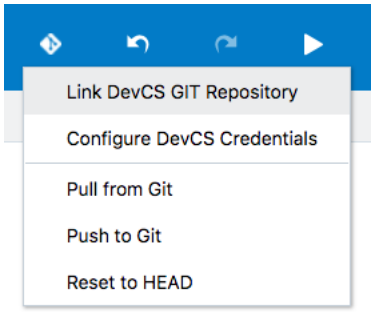
In this lab we will examine the Master-Detail pattern. We will add a list of child Line Items to the Edit Expense Report page.

Hands on Lab Instructions

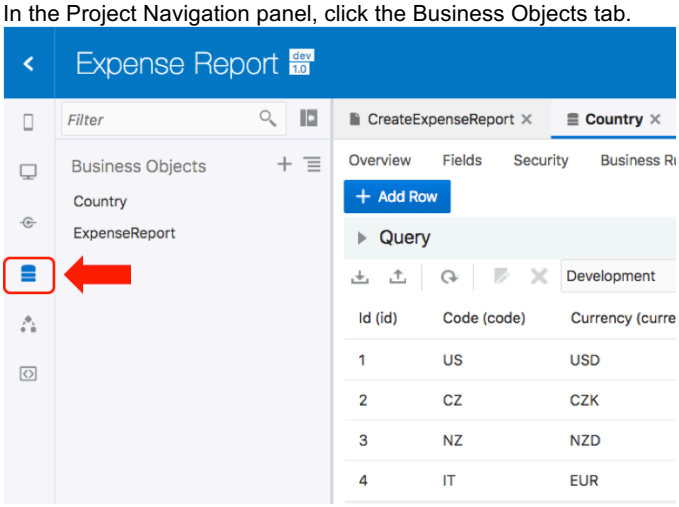
Checking Out the Project

First we will check out the project from Git. As Git exports don't contain data we will also have to upload the data for the all the business objects from one ZIP file.

1. Open your Expense Report *YourName* project from the Dynamic UI lab.
2. Click the Git button and choose Link DevCS Git Repository



3. Click Edit Link
4. In the Branch Selection field, select ***solution***.
5. Click Save Configuration.
6. Click the Git button and choose Reset to HEAD.
7. Reload your browser.
8. In the Project Navigation panel, click the Business Objects tab.

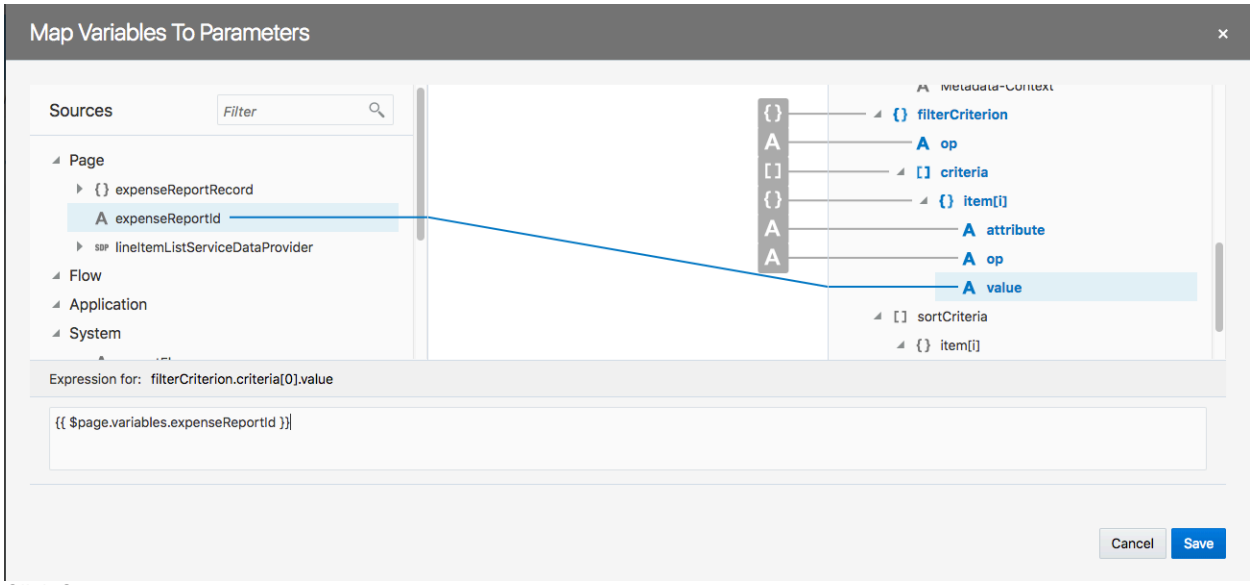


9. Click the Business Objects menu and click Data Manager.

10. Click Import from File.
11. Upload ExpenseReportData.zip

Adding a Child Table

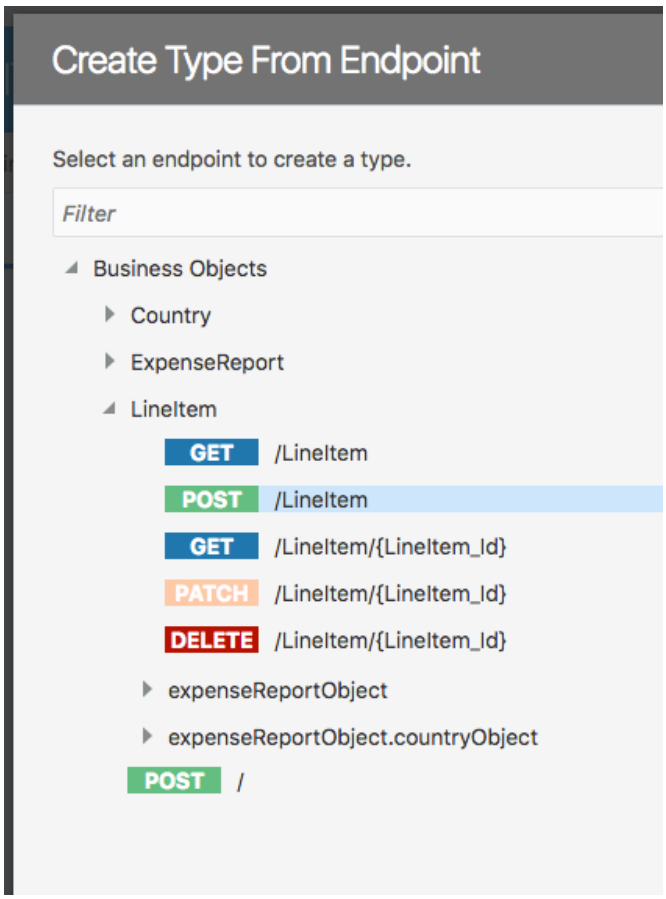
1. Open EditExpenseReport. Use the trick where you use the Live mode from the list of Expense Reports so it loads with some data in it.
2. Add a Heading between the form and the buttons and call it Line Items
3. Add a List View below the heading.
4. In the Property Inspector, click Add Data
5. In the Choose an Endpoint tab, select Business Objects > LineItem > GET /LineItem. Click Next.
6. Select Custom Template. Click Next.
7. Choose the ***name*** and ***amount*** fields. Click Next. Then click Finish.
8. Open Variables tab. Select lineItemListServiceDataProvider variable and in the Property Inspector click the (->) button for Input Parameters.
9. and set the following in filterCriterion:
    - op: "$or"
    - attribute: "expenseReport"
    - op: "$eq"
    - value: $page.variables.expenseReportId (you can drag expenseReportId from the left panel to bind it)



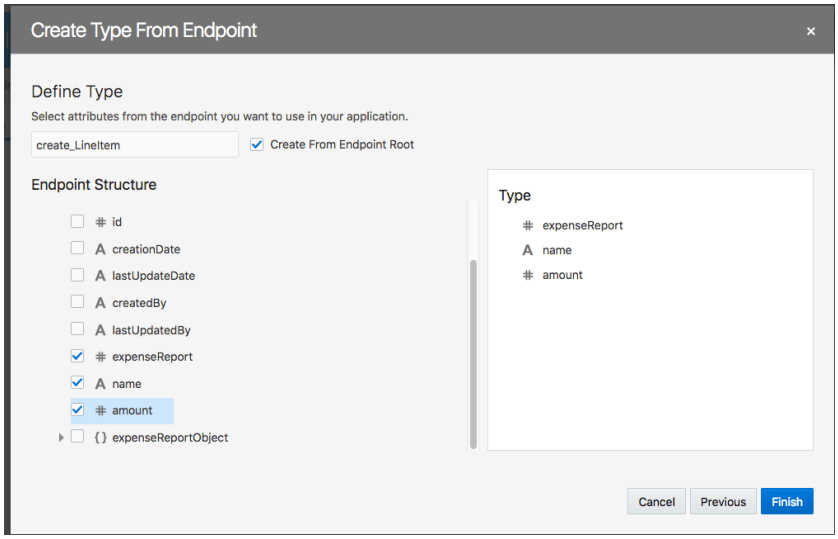
10. Click Save

Adding a New Line Item Form

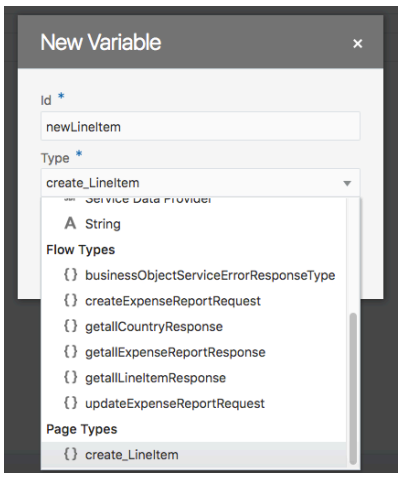
1. Go to the Variables panel of the EditExpenseReport page.
2. Click the Types tab. Create a New Type from Endpoint.
3. Select Business Objects > LineItem > POST /LineItem



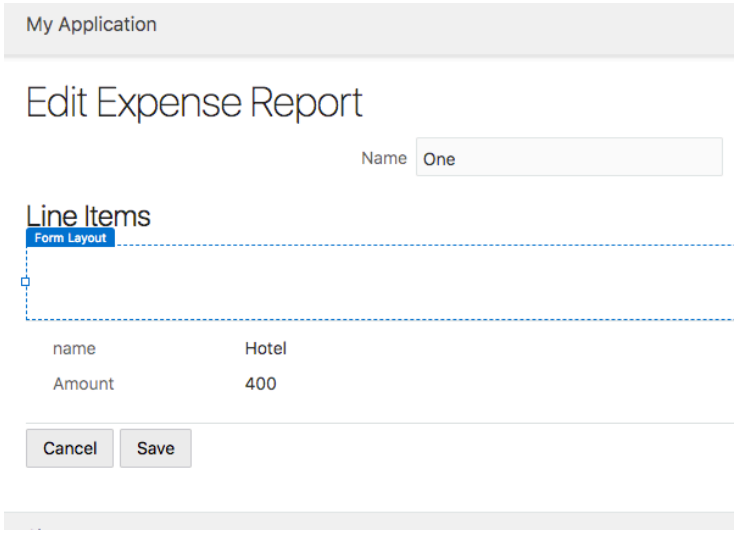
4. Click Next.
5. Add the ***expenseReport***, ***name***, and ***amount*** fields.



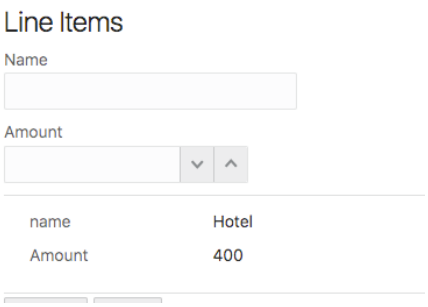
6. Click Finish.
7. Click the Variable Tab and add a new variable called newLineItem of type createLineItem



8. Expand the newLineItem variable and select the expenseReport property.
9. In the Property Inspector, change Default Value to {{ $page.variables.expenseReportId }}
10. Go back to the Page Designer.
11. Drop a Form Layout component between the Line Items heading and the list of Line Items

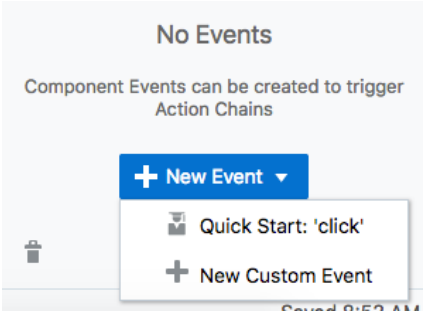


12. Add an Input Text component and a Currency component to the Form Layout. Change the display labels to Name on the Input Text and Amount on the Currency component.

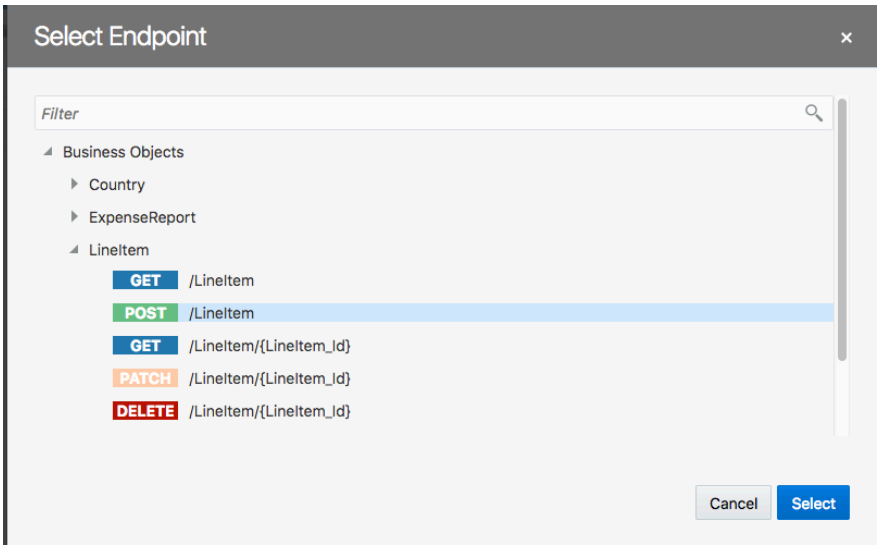


13. Select the Name field in the Page Designer. In the Data tab of the Property Inspector, bind Value to {{ $page.variables.newLineItem.name }}.
14. Do the same to bind the Amount field to {{ $page.variables.newLineItem.amount }}
15. Add a Button below the Form Component. Change the Text to Add.

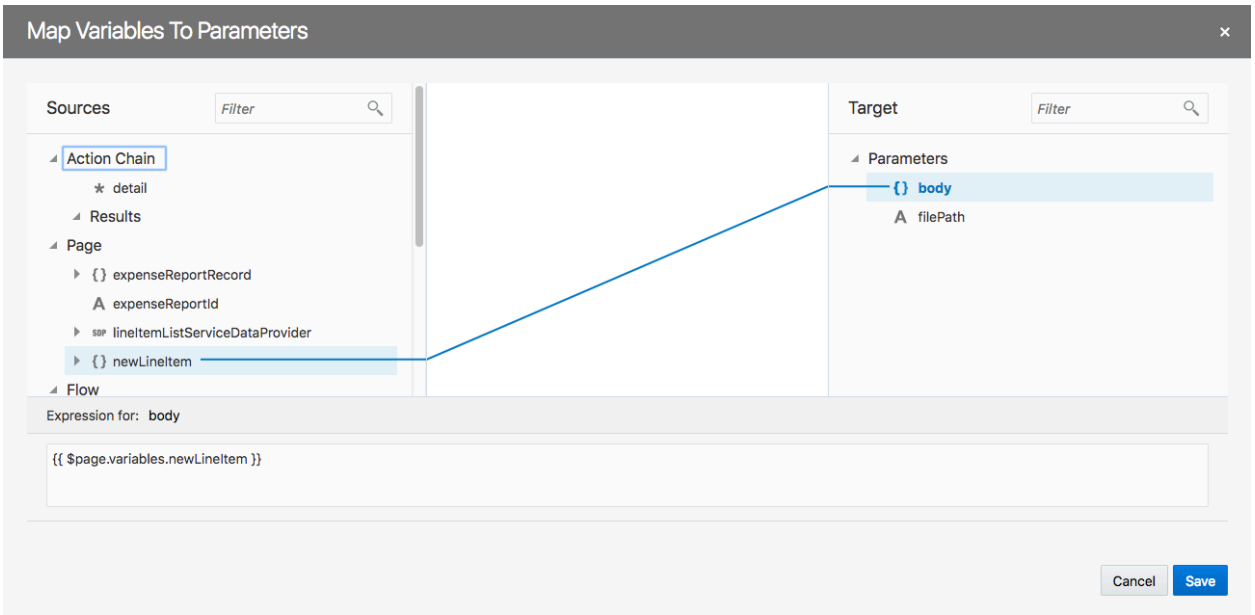
16. Select the Button and click the Events tab of the Property Inspector. Click New Event > Quick Start: 'click'



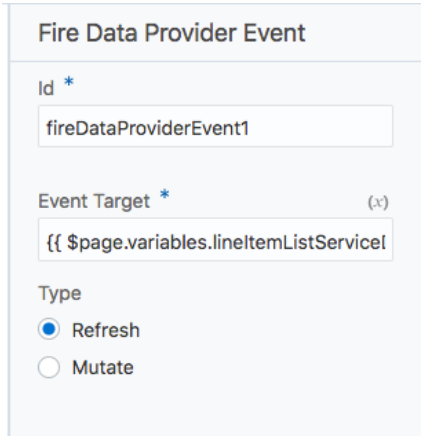
17. Drop a Call REST Endpoint action under Start. Click Select Endpoint and select Business Objects > LineItem > POST /LineItem.



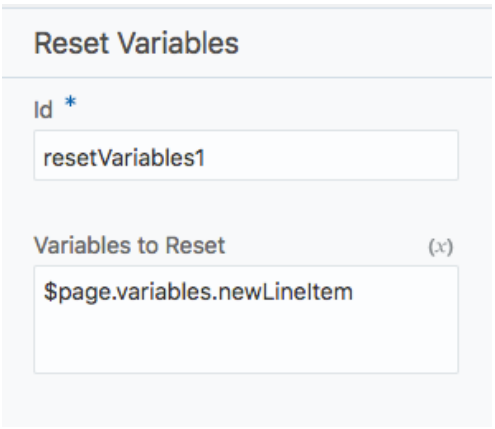
18. Click (->) for Parameters. Map {} newLineItem on the left panel to {} body on the right panel. Click Save.



19. Drag a Fire Data Provider Event under the Call REST Endpoint action. Select {{ $page.variables.lineItemListServiceDataProvider }} as the Event Target. Set Refresh as the Type.



20. Drag a Reset Variables action to under the Fire Data Provider Event action. Set $page.variables.newLineItem as the Variables to Rest.

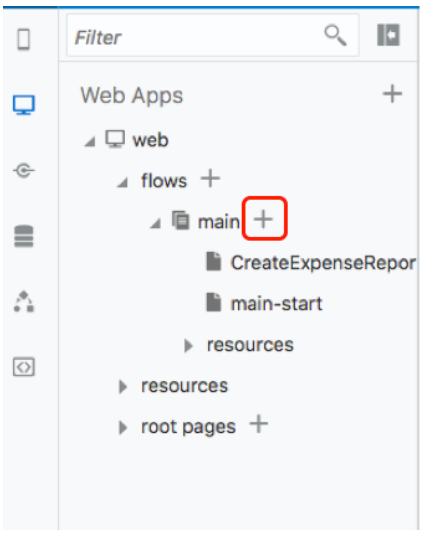


21. Run and test the app.

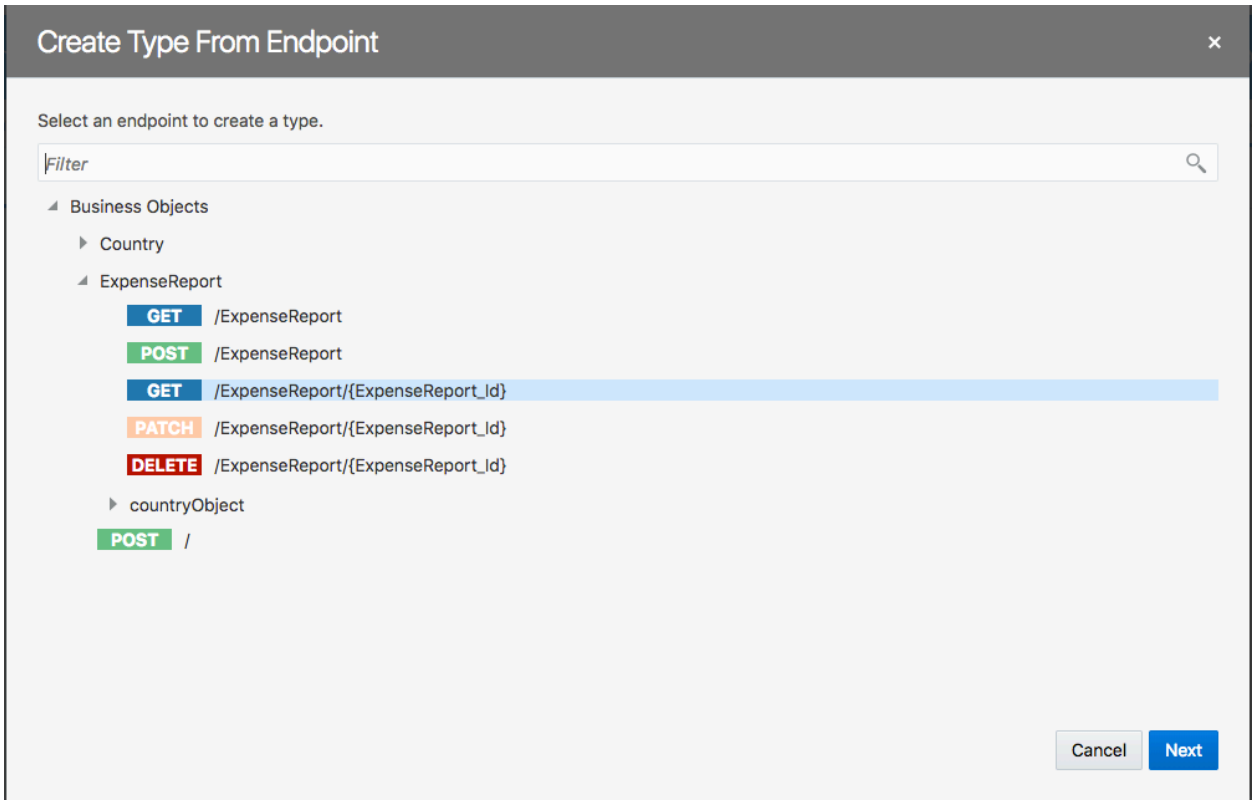
Creating a Single-Page Master-Detail

Now let's create a single page that shows a list of expense reports on the left and the details on the right.

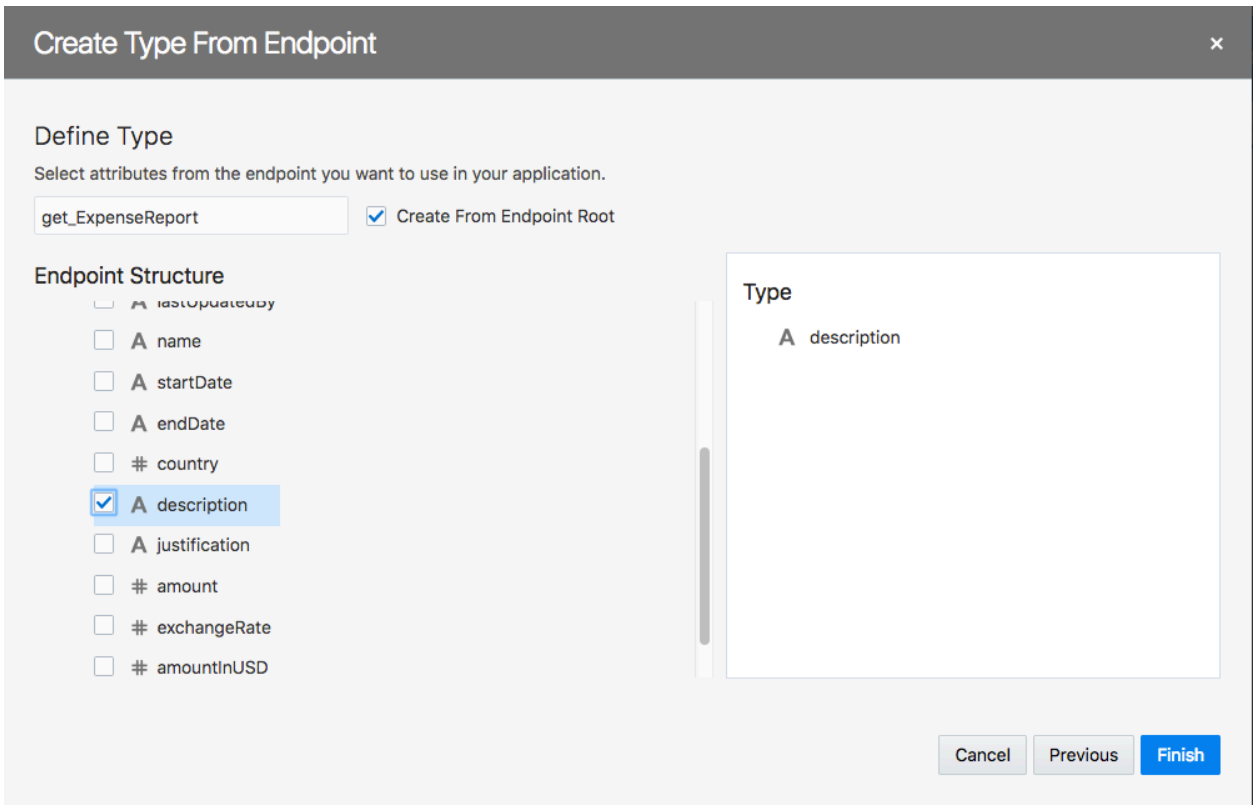
1. In the Web Apps panel, expand web > flows and click the + button next to main.



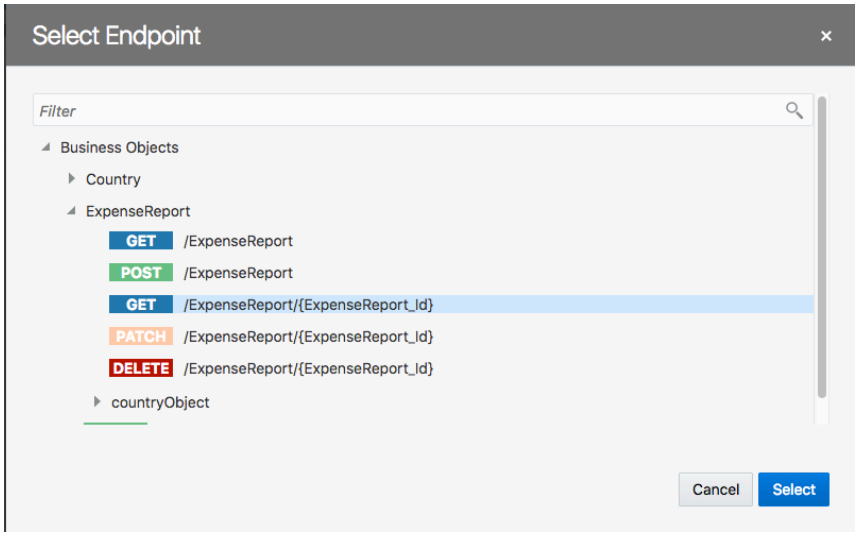
2. Name the page ***main-singlePage***.
3. Drop a List View on the page. Use the Add Data Quickstart to bind the list to the ***Expense*** business object and display the ***name*** field.
4. Shrink the List View to only take the left third of the page. You can play with the formatting, like removing the ***name*** label and expanding the field to take up the full width of the list.
5. Go to the Variables panel for ***main-singlePage***.
6. In the Types tab, create a Type from Endpoint based on the GET /ExpenseReport/{ExpenseReport_Id} endpoint.



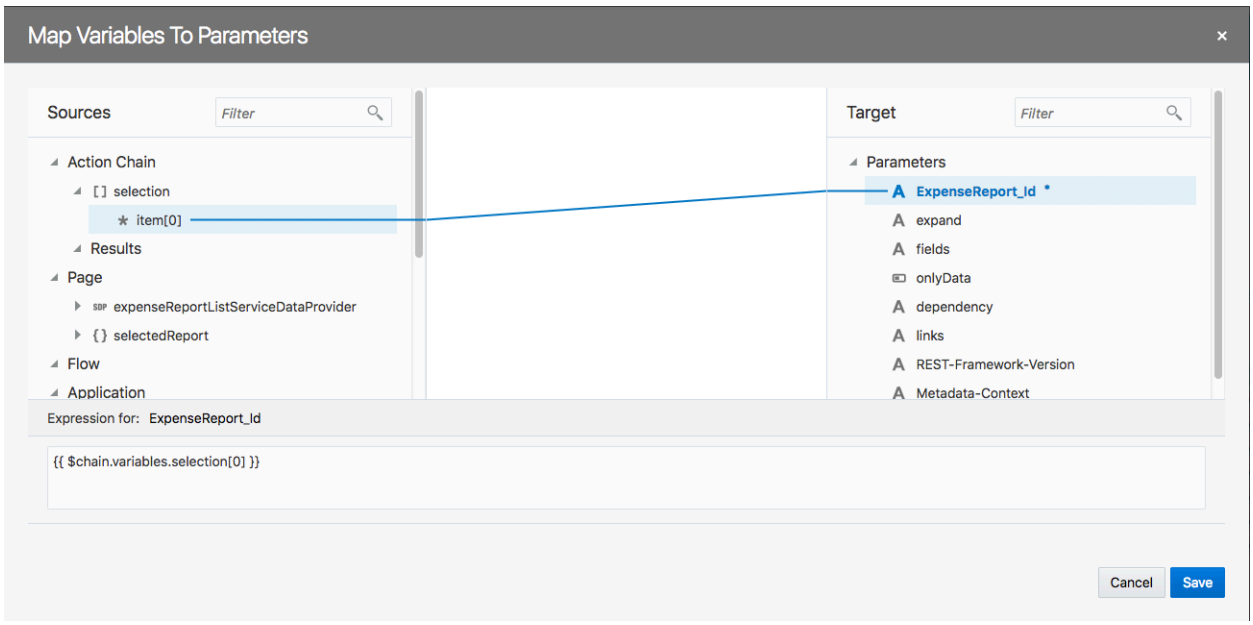
7. Add the description field to the Type and click Finish.



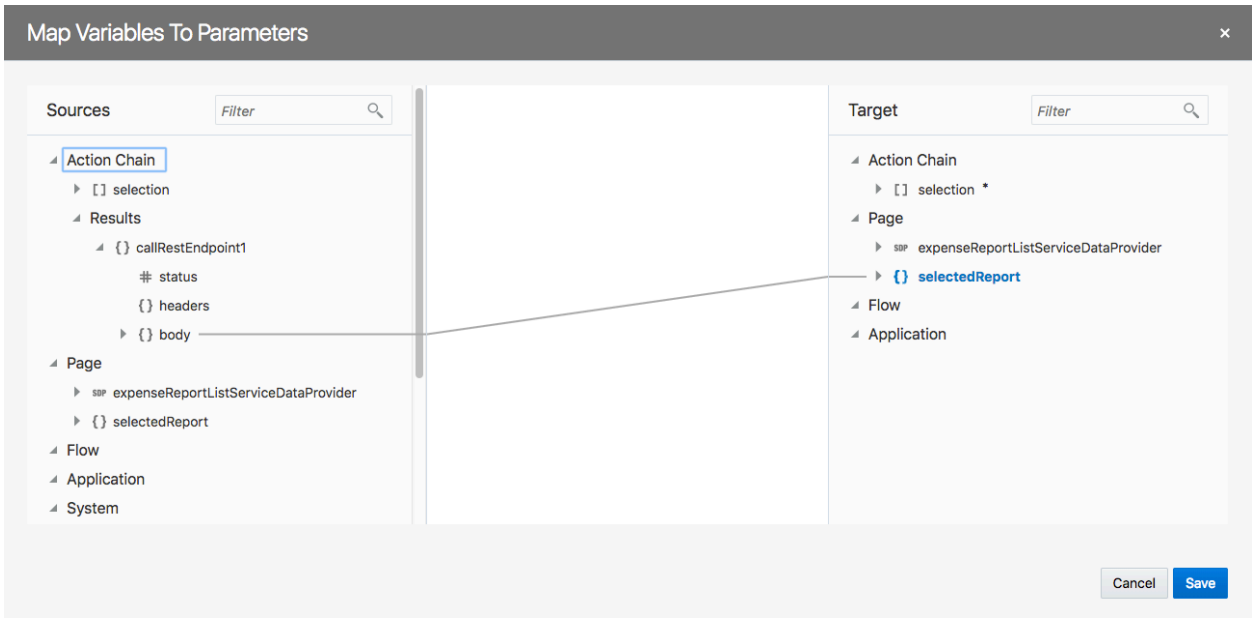
8. Switch back to the Variables tab and add a variable called selectedReport of type get_ExpenseReport.
9. Drag a Form Layout on the right side of the List.
10. Drag a Text Area to the page. In the Property Inspector, set the Label Hint to Description.
11. Go to the Data tab and bind the Text Area to **{{ $page.variables.selectedReport.description }}**
12. Select the List View. In the Events tab of the Property Inspector, click New Event > Quick Start: 'selection'. This opens the ListViewSelectionAction in the editor.
13. Drop a Call REST Endpoint under Start and select the GET /ExpenseReport/{ExpenseReport_Id} endpoint.



14. Under Input Parameters, click ExpenseReport_Id to open the variable mapper.
15. Map Action Chain > selection > item[0] to ExpenseReport_Id and click Save.



16. Drop an Assign Variables action under the Call REST Endpoint action.
17. Click the -> arrow for Variables and map Action Chain > Results > callRestEndpoint1 > body to selectedReport.



18. Go back to the Page Designer for main-singlePage and select the listView.
19. In the General tab of the Properties Inspector, switch Selection Mode to Single.