

VBCS Training Lab

Working with Related Data

Introduction

Our list of Service Requests has a problem: It displays the Lookup Code instead of the display label for the Service Request Status. In this lab, we will add logic to fetch all of the display names for Service Request Status and use the correct display name in the list.

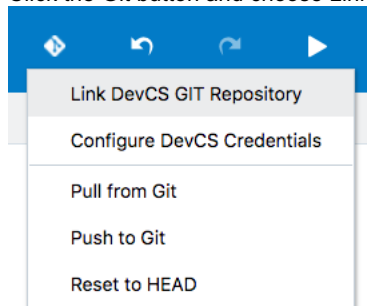
Specifically, we will:

- Create a flow-level JavaScript function that returns the display value for a given lookup code
- Create a flow-level type and variable to hold all of the possible lookup codes and display labels
- Add an action to populate the list of lookup codes and display labels to the page with the list of Service Requests
- Change the <oj-bind-text> logic to call the decoder function

Hands on Lab Instructions

Updating the Project to

1. Click the Git button and choose Link DevCS Git Repository



2. Click Edit Link
3. In the Branch Selection field, select relatedData.
4. Click Save Configuration
5. Click the Git button and choose Reset to HEAD
6. Reload your browser

Adding the getLOVMeaningValue Function

1. Open the main servicerequests Editor by opening the Web Apps panel and clicking crmCrud > flows > servicerequests
2. Open the JavaScript panel of the Flow Editor by clicking the JS button
3. Add the **getLOVMeaningValue** function as shown in bold below:

```
define([], function () {  
    'use strict';  
    var FlowModule = function FlowModule() {};  
  
    FlowModule.prototype.getLOVMeaningValue = function (key, lovs) {
```

```

var res;
var keyProperty = Object.keys(lobs[0])[0];
lobs.forEach(function(record) {
    if (record[keyProperty] === key) {
        res = record;
    }
});

if (res) {
    return res[Object.keys(res)[1]];
}

};

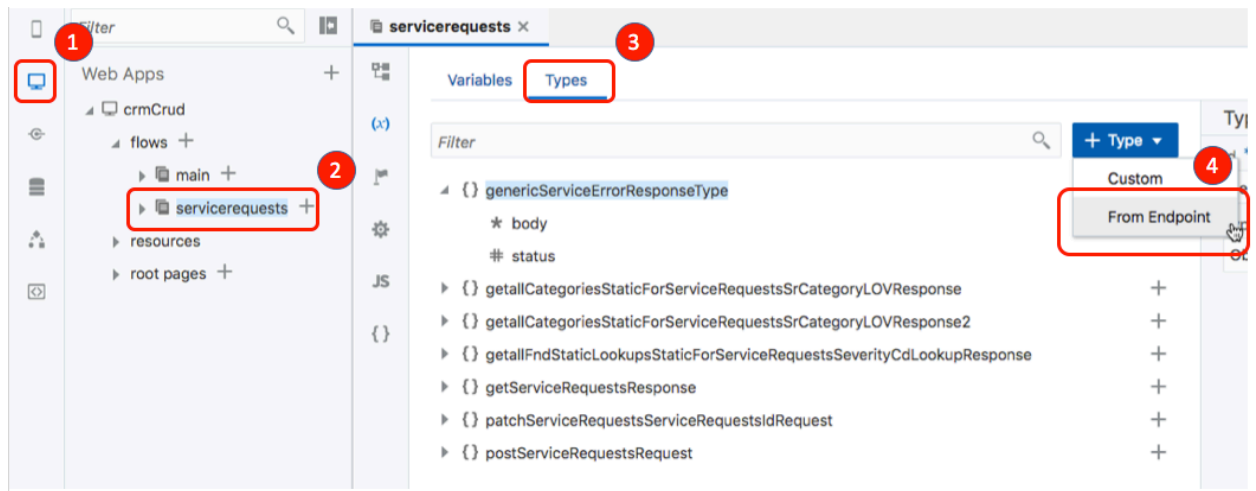
return FlowModule;
});

```

Create the Type and Variable for the Status Code Options

Note: Make sure you do this on the servicerequests Flow Editor and not in the servicerequests-main Page Editor. The two are almost identical.

1. In the servicerequests Page Flow Editor, open the Variables editor, then click the Types tab.
2. Create a new Type from Endpoint based on the /fndStaticLookups;for=servicerequests-StatusTypeLookup.



3. Change the name of the type to statusCodeOptions. Add LookupCode and Meaning to the Type and click Finish.

Create Type From Endpoint

Define Type
Select attributes from the endpoint you want to use in your application.

statusCodeOptions ☒ Create From Endpoint Root

Endpoint Structure

- ☐ # totalResults
- ☐ # count
- ☐ hasMore
- ☐ # limit
- ☐ # offset
- ☒ [] items
 - ☐ A LookupType
 - ☒ A LookupCode
 - ☒ A Meaning

Type

- [-] [] items
 - A LookupCode
 - A Meaning

Cancel Previous **Finish**

- Switch to the Variables tab.
- Create a new variable called statusCodes of type statusCodeOptions.

New Variable

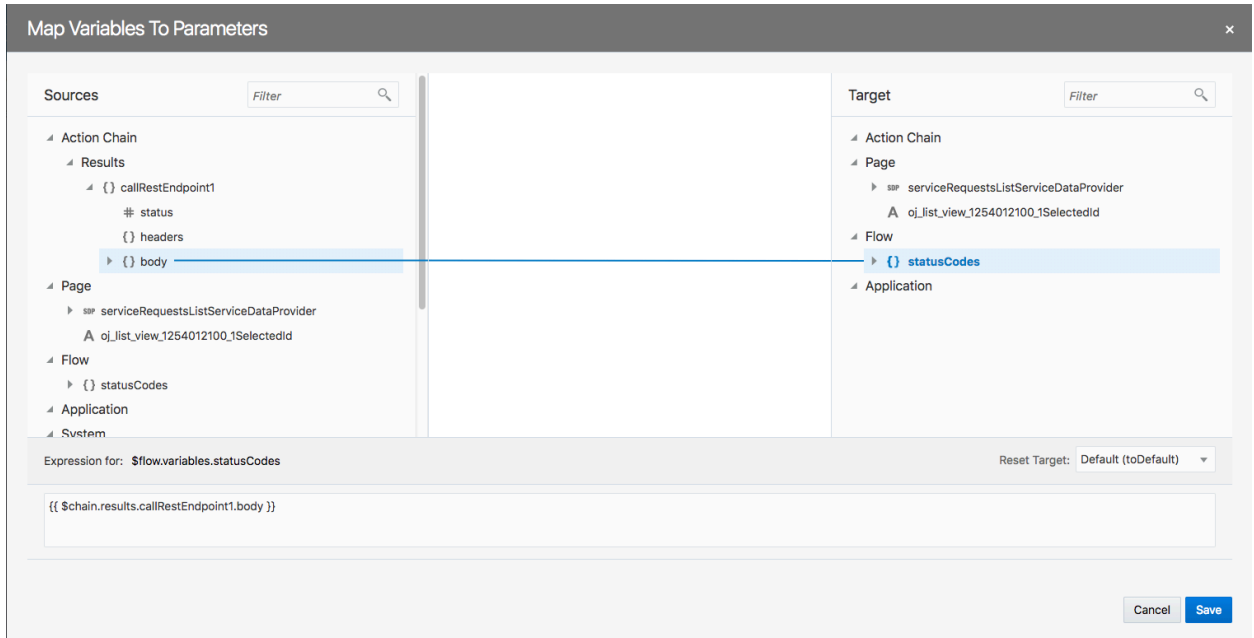
Id *
statusCodes

Type *
statusCodeOptions

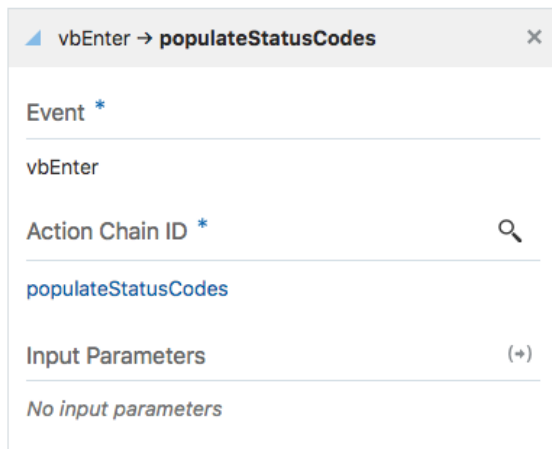
☐ Create Another Cancel **Create**

Create an Action to Populate Account Type Options

- Open servicerequests-start.
- Create a new Action Chain called populateStatusCodes.
- Drag a Call REST Endpoint action from the left panel to underneath the Start node.
- In the Property Inspector, click Select Endpoint and select `crmRestApi > Service Requests > /findStaticLookups;for=serviceRequests-StatusTypeLookup`.
- Add an Assign Variables action that maps the body returned by the Call REST action to the statusCodes flow variable.

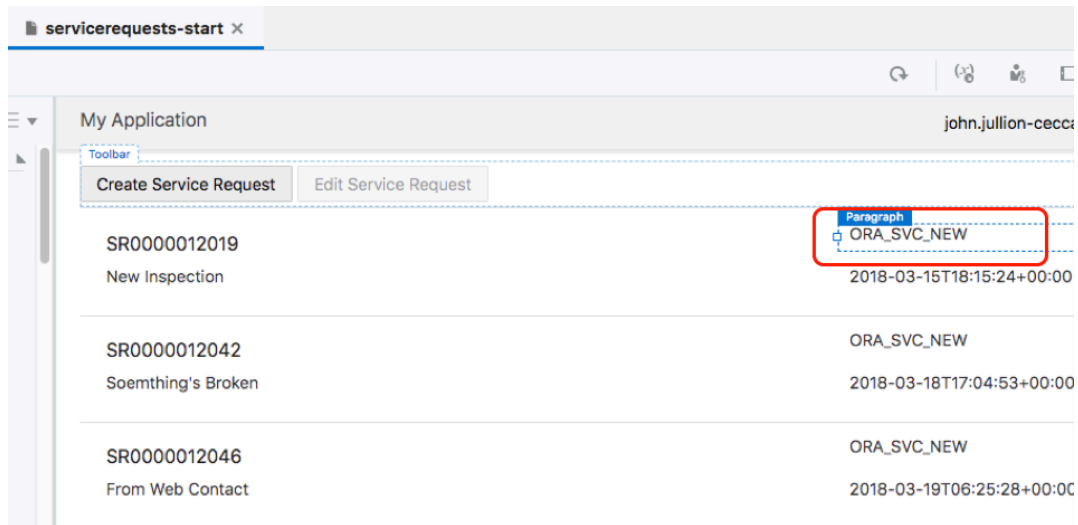


6. Return to the servicerequests-start tab and open the Settings panel
7. Create a new event:
 - Event: vbEnter
 - Action Chain: populateStatusCodes



Update the List to Call the getLOVMeaningValue Function

1. Open the Page Designer for servicerequests-start.
2. Select one of the Type fields in the list



3. In the General tab of the Property Inspector, change the Text to:

```
{{ [[ $flow.functions.getLOVMeaningValue($current.data.StatusCd,$flow.variables.statusCodes.items)]] }}
```