VBCS Lab
Dynamic UI and Client-Side Validations

Introduction

In this lab we will take a simple form for submitting an expense report and add dynamic display, validators, and converters to it. Specifically, we will:

- Set up our form to check validity before submitting

- Add a Date validation, both using HTML properties and a custom JavaScript validator

- Add a custom converter to a currency field

- Make a field conditionally required

- Make some fields dynamically hide and display based on other values on the 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 Countries business object from a CSV.

1. Click Import > Import from Git
2. Click Add Credentials
3. Enter https://developer.em2.oraclecloud.com/developer20509-cloud01/ for the DevCS URL and your username and password. Click Finish.
4. Enter the following information:
   - DevCS URL with Credentials - https://developer.em2.oraclecloud.com/developer20509-cloud01/
   - Project Selection - VBCS Training
   - Repository Selection - expenseReport.git
   - Branch Selection - master
   - Application Name - expenseReport*YourName*
   - Application ID - accept default
5. Click Import. The application is opened in the list of apps.
6. Click your app in the list of apps to open it.

7. In the Project Navigation panel, click the Business Objects tab.
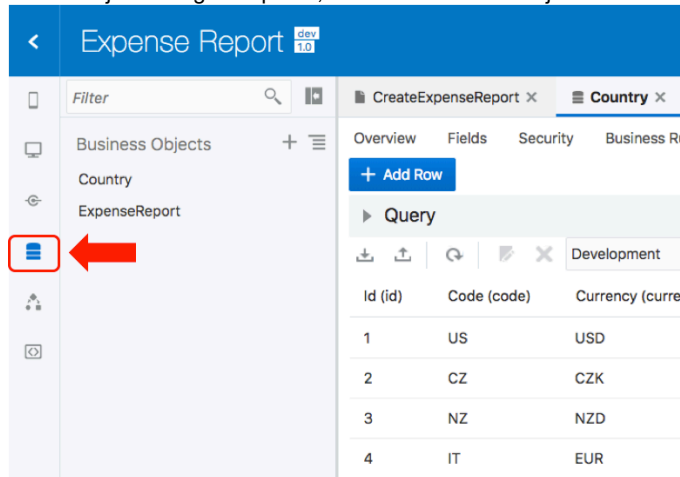
8. Click Countries.
9. In the Data tab, click the Upload from CSV ⤓ button.
10. Upload Country.csv. The Data tab should now shows some entries.

Setting Up Validation

First we have to set up our form to check the validity of its contents before submitting. We do this by surrounding the form with an <oj-validation-group> element, adding a custom isFormValid Javascript function that returns a boolean, and then calling that function before submitting the form.

1. Open CreateExpenseReport
2. Switch to the Code view
3. Surround the expense-form div with <oj-validation-group id="tracker">. Don't forget the closing tag.

```
1   <div class="oj-flex">
2     <h1 id="h1--862244262-1" class="oj-flex-item oj-sm-12 oj-md-12">New Expense Report</h1>
3   </div>
4   <oj-validation-group id="tracker">
5     <div class="oj-flex" id="expense-form">
6       <oj-form-layout id="oj-form-layout--862244262-1" class="oj-flex-item oj-sm-12 oj-md-6" max-columns="1" labe
13      <oj-form-layout id="oj-form-layout--862244262-2" class="oj-flex-item oj-sm-12 oj-md-6" label-edge="start">.
20      </div>
21    </oj-validation-group>
22  <div class="oj-flex">
23    <oj-toolbar id="oj-toolbar--862244262-1" chroming="full" class="oj-flex-item oj-sm-12 oj-md-12">
24      <oj-button id="oj-button--862244262-2" on-click="[[$page.listeners.saveButtonClicked]]">Save</oj-button>
25      <oj-button id="oj-button--862244262-1" on-click="[[$page.listeners.backButtonClicked]]">Cancel</oj-button>
26    </oj-toolbar>
27  </div>
```

4. Switch to the JS view

5. Add the isFormValid function shown in bold:

```
define([], function() {
  'use strict';

  var PageModule = function PageModule() {};

  PageModule.prototype.isFormValid = function() {
    var tracker = document.getElementById("tracker");

    if (tracker.valid === "valid") {
       return true;
    } else {
      tracker.showMessages();
      tracker.focusOn("@firstInvalidShown");
      return false;
    }
  }
}
...
```
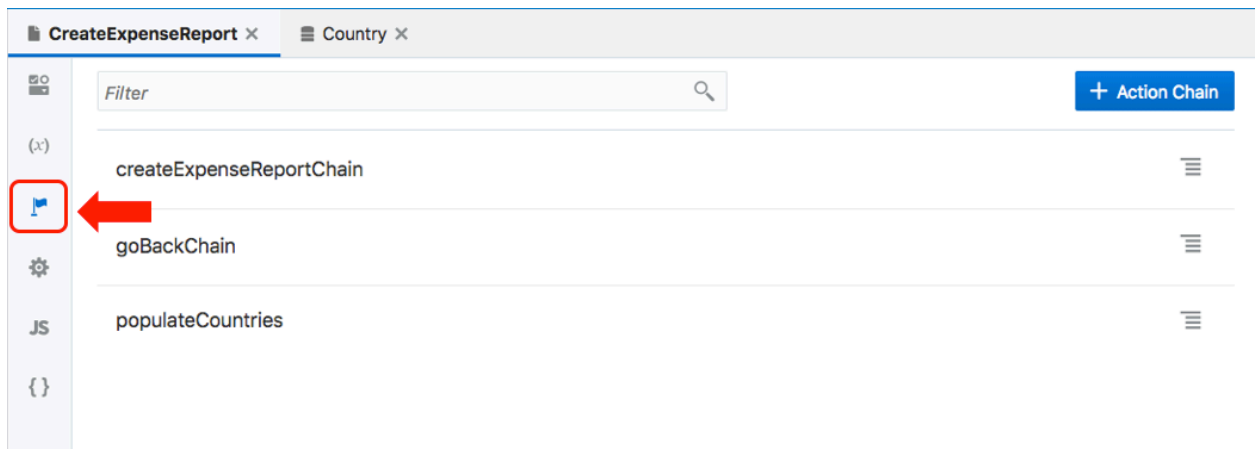
6. Go to the Actions list and open createExpenseReportChain

7. Add an If action after Start.
8. Set the Condition to {{ $page.functions.isFormValid() }}

If

Id *

if1

Condition *   (x)

{{ $page.functions.isFormValid() }}

9. Move the Call businessObjects/cre... node to the true branch of the If action
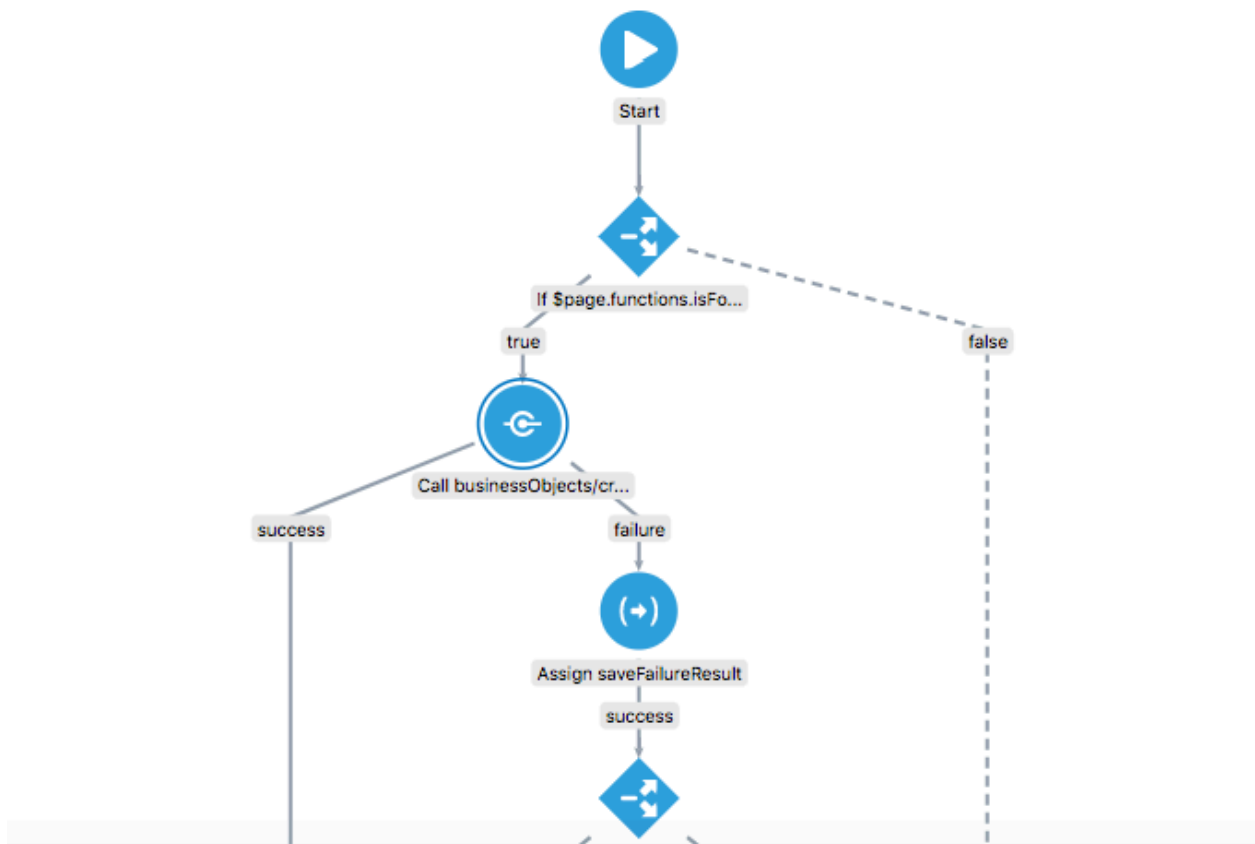
Setting a Date Validation

Note that we are going to use the Expression Editor here to create the expression. The Expression Editor gives you code completion over all in-scope variables as well as JS error highlighting.

1. Go back to the Page Designer for CreateExpenseReport.
2. Select the End Date field in the form.

3. In the General panel of the Property Inspector, click (x) > Expression Editor for the the Min property.
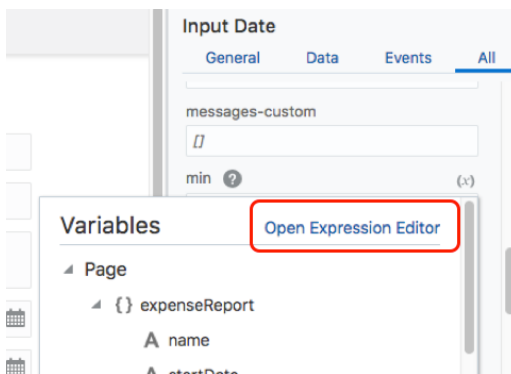
4. In the left panel, expand {} expenseReport and double-click startDate. The editor enters:

   $page.variables.expenseReport.startDate

5. Click Save.
6. Run the page and try entering a date before the Start Date. The DatePicker for End Date makes all dates before the Start Date unavailable. If you manually enter a dat before the Start Date it shows an error.

Using a Custom Validator

1. Open the JS panel for CreateExpenseReport.

2. Add the following function after the formIsValid function:

```
PageModule.prototype.endDateAfterStartDateValidator = function (startDate) {
    return [{
        validate: (endDate) => {
            if (endDate) {
                const valid = endDate >= startDate;
                if (!valid) {
                    throw new Error('End date must be start date or later');
                }
                return valid;
            }
        },
    }];
};
```

3. Return to the Page Designer panel and select End Date
4. In the General panel of the Property Inspector, delete the value for the **Min** property.
5. In the All panel of the Property Inspector, change the **validators** property to:

{{ $page.functions.endDateAfterStartDateValidator($page.variables.expenseReport.startDate) }}

Making a Field Conditionally Required

Let's make the Justification field required if the Amount is more than $500.

1. Select Justification
2. In the All panel of the Property Inspector, change the **required** property to:

{{ $page.variables.expenseReport.amount * $page.variables.expenseReport.exchangeRate >= 500 }}

3. Run the page and try entering 600 for the Amount. Justification becomes conditional.

Creating a Computed Field

Let's change Amount in USD to be calculated by multiplying the Exchange Rate by the Amount. Note that this should be done for display purposes only.

1. Select Amount in USD
2. In the General panel of the Property Inspector, change the **Value** property to:

{{$page.variables.expenseReport.exchangeRate * $page.variables.expenseReport.amount }}

3. Run the page and try entering different values for the Exchange Rate. Amount in USD is updated automatically.

Using a Custom Field Converter

Now let's create a converter that will take the Amount field and display the correct currency symbol for the selected Country.

1. Open the JS panel for CreateExpenseReport.

```
CreateExpenseReport ×     Country ×

1  ◢ define([], function() {
2       'use strict';
3
4       var PageModule = function PageModule() {};
5
6  ◢    PageModule.prototype.isFormValid = function() {
7          var tracker = document.getElementById("tracker");
8  ◢       if (tracker.valid === "valid") {
9            return true;
10 ◢       } else {
11           tracker.showMessages();
12           tracker.focusOn("@firstInvalidShown");
1            return false;
1          }
15       }
16
17 ◢    PageModule.prototype.endDateAfterStartDateValidator = function(startDate) {
18 ◢       return [{
19           validate: (endDate) => {
20             if (endDate) {
```

2. Add the following functions after the endDateAfterStartDateValidator function:

PageModule.prototype.getCurrency = function(countries, selectedCountry) {
  const res = countries.filter(country => country.value === selectedCountry);
  if (res.length === 0) {
   return "USD";
  }
  return res[0].country.currency;
 }

PageModule.prototype.getCurrencyConvertor = function(countries, selectedCountry) {
    return {
      type: "number",
      options: {
        style: "currency",
        currency: this.getCurrency(countries, selectedCountry),
        currencyDisplay: "symbol",
        pattern: '¤ ##,##0.00'}};
  }

3. Go back to the Page Designer panel and select Amount.
4. Set the ***convertor*** property to:

[[$page.functions.getCurrencyConvertor($page.variables.countryOptions,
$page.variables.expenseReport.country)]]

5. Run the page and test switching the Country field. The Amount field is updated with the currency.

Dynamically Displaying Fields

Now let's wrap Exchange Rate and Amount in USD in an <oj-bind-if> that only displays if the currency is not USD.

1. Go back to the Page Designer for CreateExpenseAccount and switch into Code view.
2. Insert the following <oj-bind-if> statement above the Exchange Rate <oj-input-number>

<oj-bind-if test="[[$page.variables.expenseReport.country !== '1' &&
      $page.variables.expenseReport.country !== undefined &&
      $page.variables.expenseReport.country !== null &&
      $page.variables.expenseReport.country !== "]]"
      id="oj-bind-if--862244262-2">

3. Put a closing </oj-bind-if> tag after the Amount in USD tag.

```
4  ▲ <oj-validation-group id="tracker">
5  ▲    <div class="oj-flex" id="expense-form">
6  ▲      <oj-form-layout id="oj-form-layout--862244262-1" class="oj-flex-item oj-sm-12 oj-md-6" max-columns="1" label-edge
7           <oj-input-text label-hint="Name" id="oj-input-text--862244262-1" value="{{ $page.variables.expenseReport.name }
8           <oj-input-text label-hint="Description" id="oj-input-text--862244262-3" value="{{ $page.variables.expenseReport
9           <oj-text-area id="oj-text-area--862244262-1" label-hint="Justification" value="{{ $page.variables.expenseReport
10          <oj-input-date label-hint="Start Date" id="oj-input-date--862244262-1" value="{{ $page.variables.expenseReport.
11          <oj-input-date label-hint="End Date" id="oj-input-date--862244262-2" value="{{ $page.variables.expenseReport.en
12        </oj-form-layout>
13 ▲      <oj-form-layout id="oj-form-layout--862244262-2" class="oj-flex-item oj-sm-12 oj-md-6" label-edge="start">
14          <oj-select-one label-hint="Country" id="oj-input-number--862244262-2" value="{{ $page.variables.expenseReport.c
15          <oj-input-number label-hint="Amount" id="oj-input-number--862244262-3" value="{{ $page.variables.expenseReport.
16          <oj-bind-if test="[[ $page.variables.expenseReport.country !== '1' &amp;&amp; $page.variables.expenseReport.cou
17            &amp;&amp; $page.variables.expenseReport.country !== null &amp;&amp;
18 ▲          $page.variables.expenseReport.country !== '']]" id="oj-bind-if--862244262-2">
19            <oj-input-number label-hint="Exchange Rate" id="oj-input-number--862244262-1" value="{{ $page.variables.expen
20            <oj-input-number converter="{&quot;options&quot;:{&quot;style&quot;:&quot;currency&quot;,&quot;currency&quot;
21            readonly="true"></oj-input-number>
22          </oj-bind-if>
23        </oj-form-layout>
24      </div>
```