TUTORIAL:

Building a Stateful Decision Service using ODM and Cloudant

Author: Nigel T. Crowther, IBM

Date: 15th March 2018

Introduction

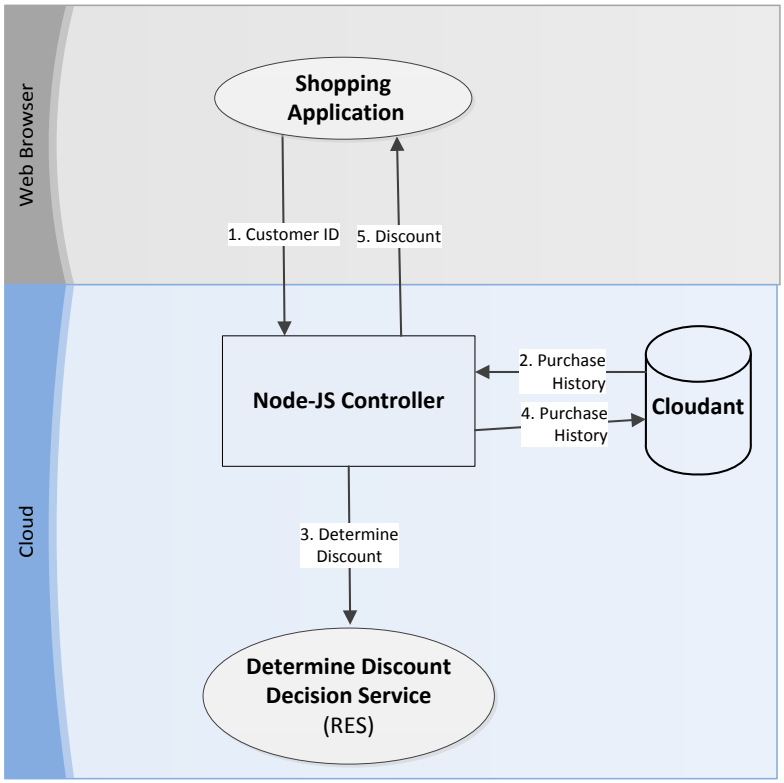
This tutorial walks you through the implementation of a stateful decision service using ODM on Cloud and Cloudant. It uses the customer discount example described in the article found here:

https://developer.ibm.com/odm/wp-content/uploads/sites/38/2018/03/odmCloudant.pdf

The tutorial can be run using IBM trial accounts (see prerequisites below).

Solution Design

The diagram below shows the high-level design of the solution. ODM on Cloud provides the stateless decision service and Cloudant persists the purchase history. Node-JS serves as the 'glue' tying the components together. Node-JS also provides a web GUI that invokes the service. This design is shown below:



The sequence of events is as follows:

1. A customer selects a product
2. The customer purchase history is retrieved from Cloudant
3. The Decision Service is invoked, passing in purchase history. If the rule fires, a discount is applied
4. The purchase history is saved back to Cloudant
5. The discount is shown to the customer.

The next sections provide more details on the design of the business rules, Cloudant and Node-JS

Business Rules

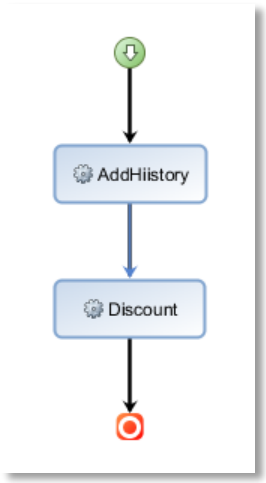
The decision service contains two business rules. The first simply adds the current purchase to the purchase history:

```
then
    add the purchase of 'the customer' to the purchase history of 'the customer';
```

The second checks whether the total number of purchases in the last sixty days is more than two. If so, a discount is applied:

```
definitions
    set purchasesInLastSixtyDays to the purchase history of 'the customer'
            where the purchase timestamp of each purchase is after 60 days before
now
if
    the number of purchases in purchasesInLastSixtyDays is more than 2
then
    set the discount of 'the customer' to 10;
    set the discount reason of 'the customer' to "10% discount for 3 purchases in
last 60 days";
```

The rule flow is as follows:



The decision service uses a *customer* json object as the input/output parameter.

Using the Stateful Decision Service Façade Design Pattern described in the article, the approach is to pass the purchase history (the state) into the decision service via the json payload.

Persisting State using Cloudant

Cloudant is an IBM software product, which is primarily delivered as a cloud-based service. Cloudant is a non-relational, distributed database. Cloudant is ideal for persisting ODM state for the following reasons:

- The JSON model generated by ODM is the database schema
- Cloudant is a document store. There are no complex queries or relationships between tables to worry about.
- Cloudant is low cost, robust, scalable and easy to use

Node-JS is the glue

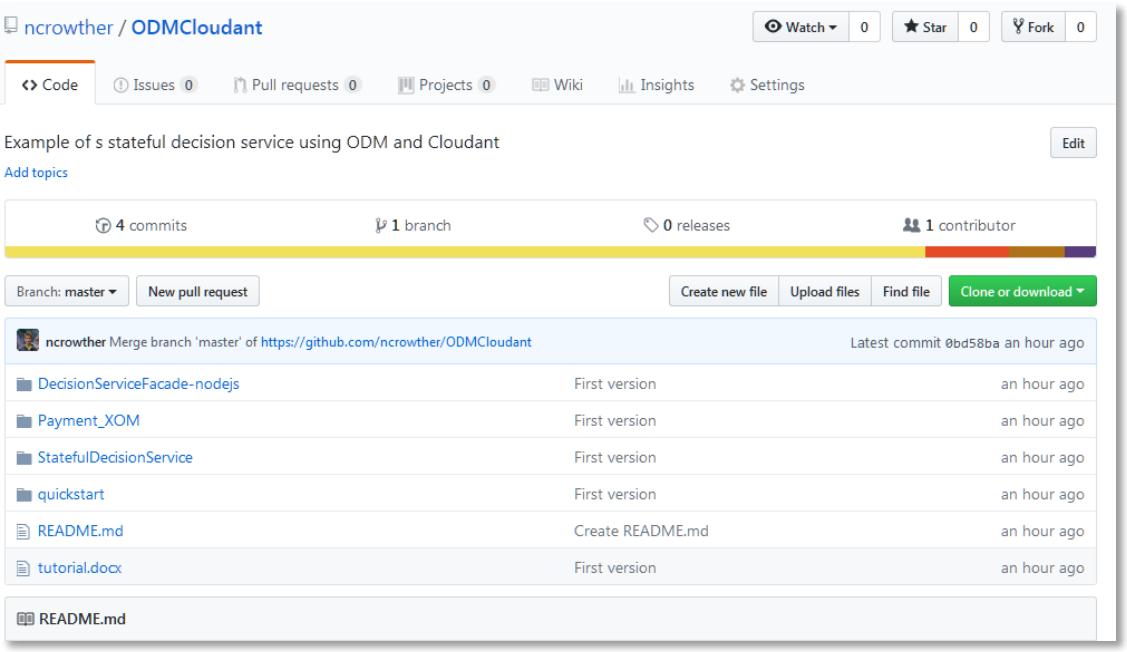
Node.js® is server side JavaScript. *Node.js* uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node-js is well suited to run on IBM Cloud. In this tutorial, Node-js implements the controller in the Stateful Decision Service Façade Design Pattern. The Model is Cloudant, the View is implemented in Jade, and the rules are ODM

Prerequisites: What you need to start the tutorial

You will now be shown how to build and run the customer discount service using the following free trials from IBM:

- ODM on Cloud. See ODM on Cloud Trial
- Cloudant. See Cloudant Trial
- ODM Cloud. See IBM Cloud Trial

The tutorial materials are provided on GitHub here: **Get the code**

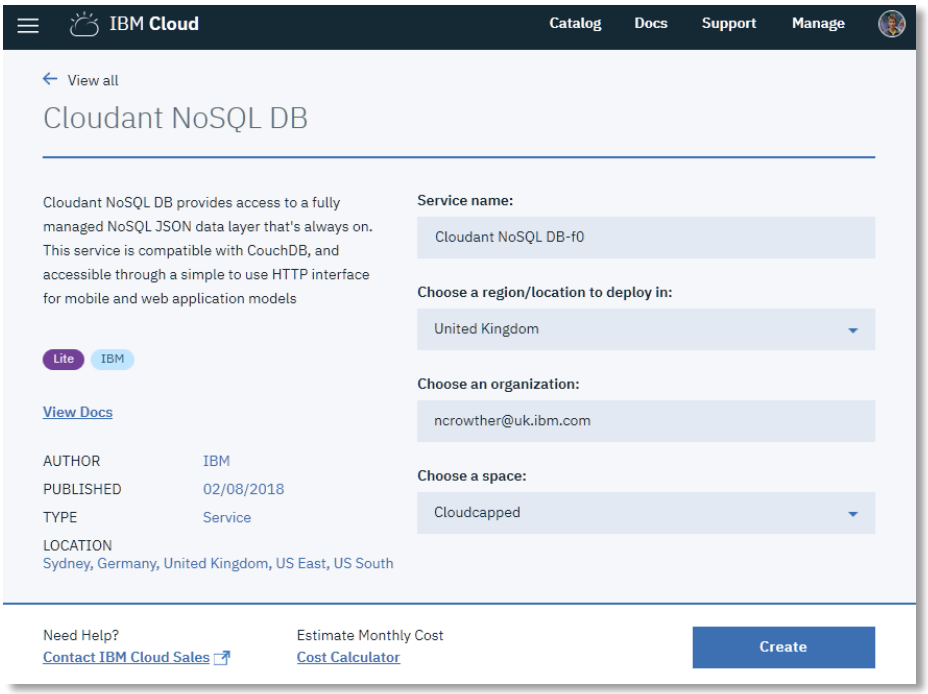


Click the Clone or Download button and save the code as a .zip.

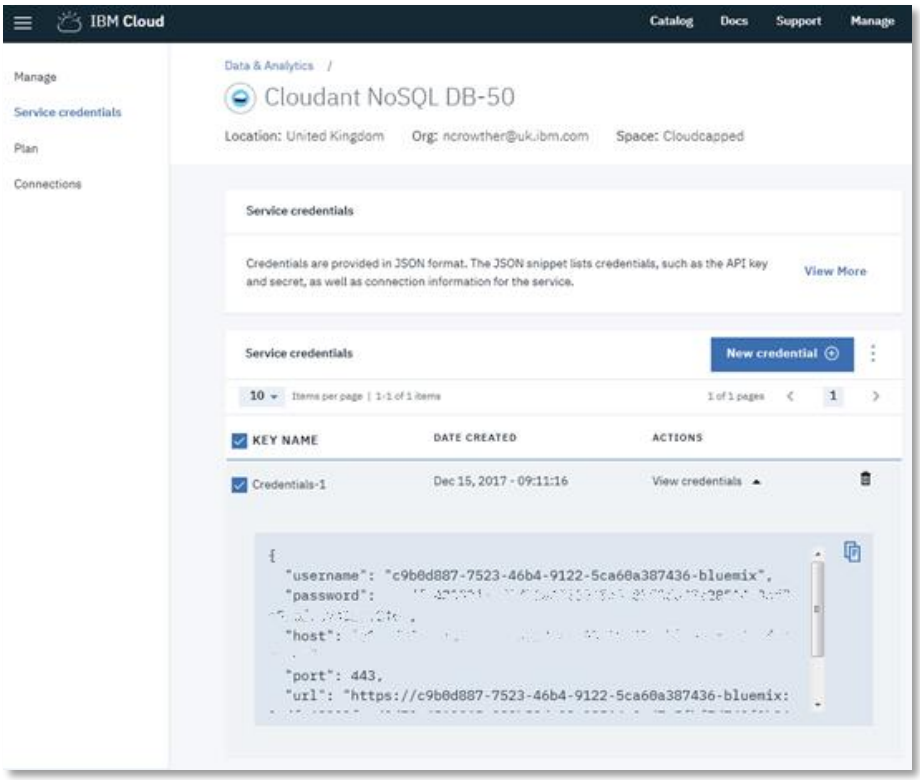
Open the zip and extract the artifacts from the *quickstart* directory within the zip.

Step 1. Create Cloudant Service

1. Log in to your IBM Cloud account and create the Cloudant service as shown in the following screen shot from the IBM Cloud dashboard.

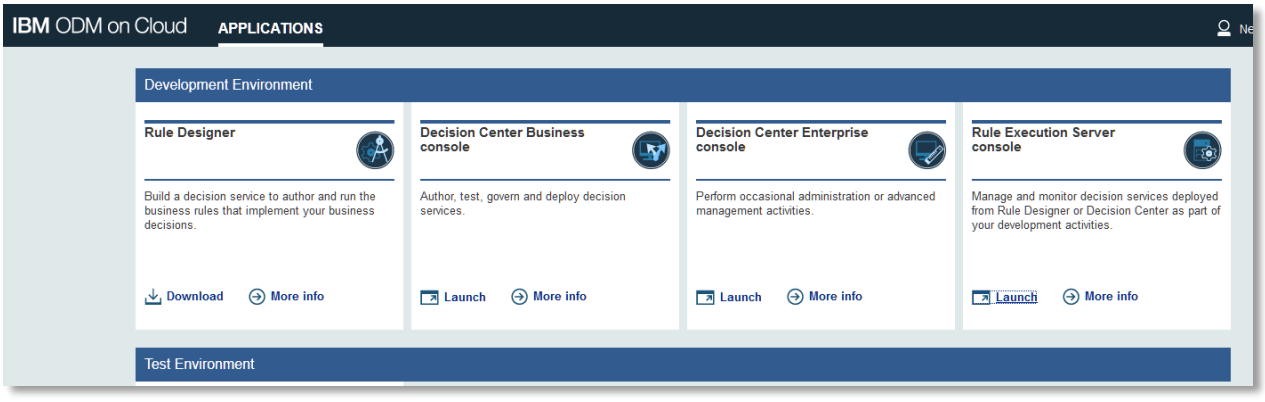


2. From the Cloudant service, click on the *Service Credentials* tab and press the New Credential button. Take note of the user name, password, host, port and URL. See following example:



Deploy the Decision Service

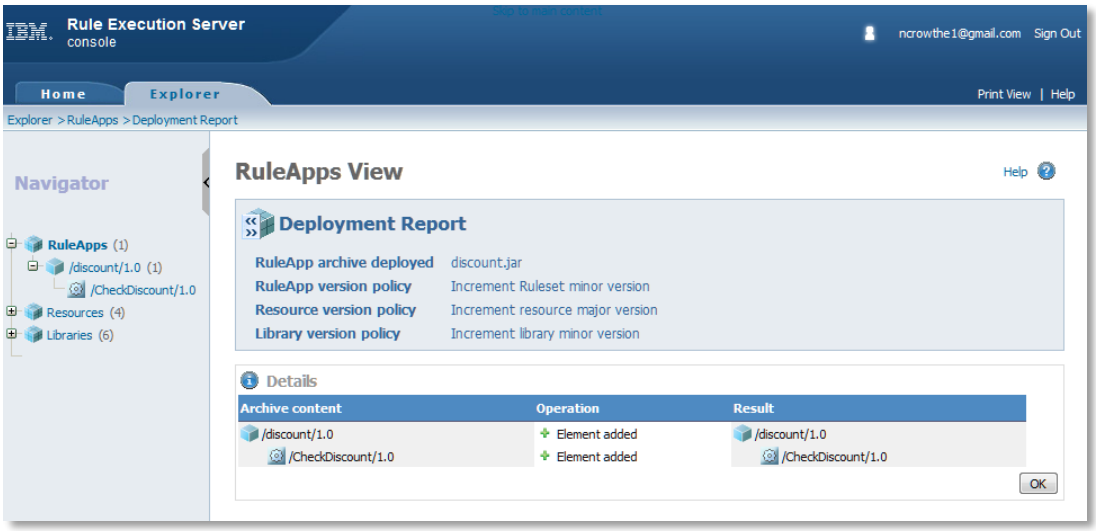
On your ODM on Cloud trial, go to the following screen:



Launch the Rule Execution Server Console. Go to the **Explorer** tab and select **Deploy RuleApp Archive**, as shown in the following example.

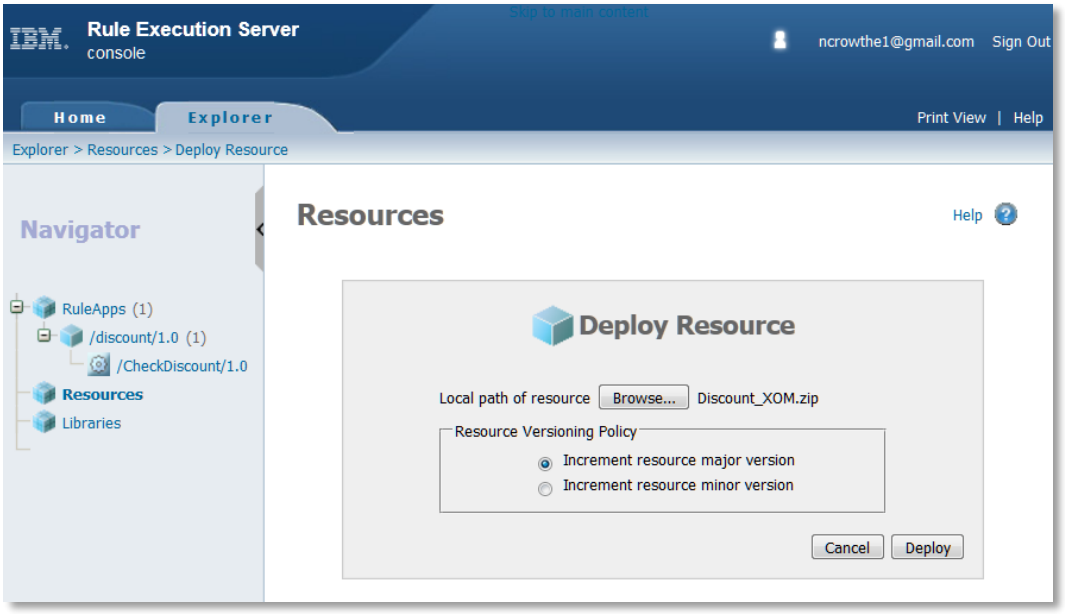
In the **Deploy RuleApp Archive** window, click the **Browse** button and select the `checkDiscount.jar` that you downloaded from the quickstart folder of the GIT repo. Leave the default versioning policy and click **Deploy**.

You should see the successful deployment of the discount rule app as shown below:



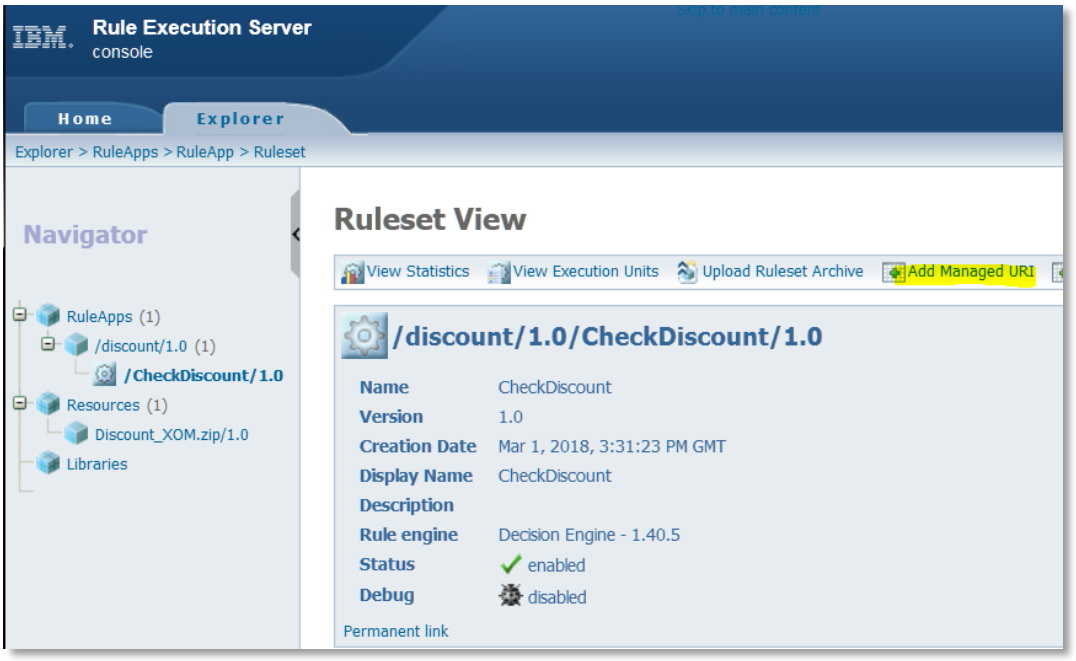
Now deploy the XOM. In the Explorer tab, navigate to the Resources folder and click 'Deploy Resource'. See below:

Now browse to the *Discount_XOM.zip* downloaded from the quickstart folder of the GIT repo. Press the Deploy button:

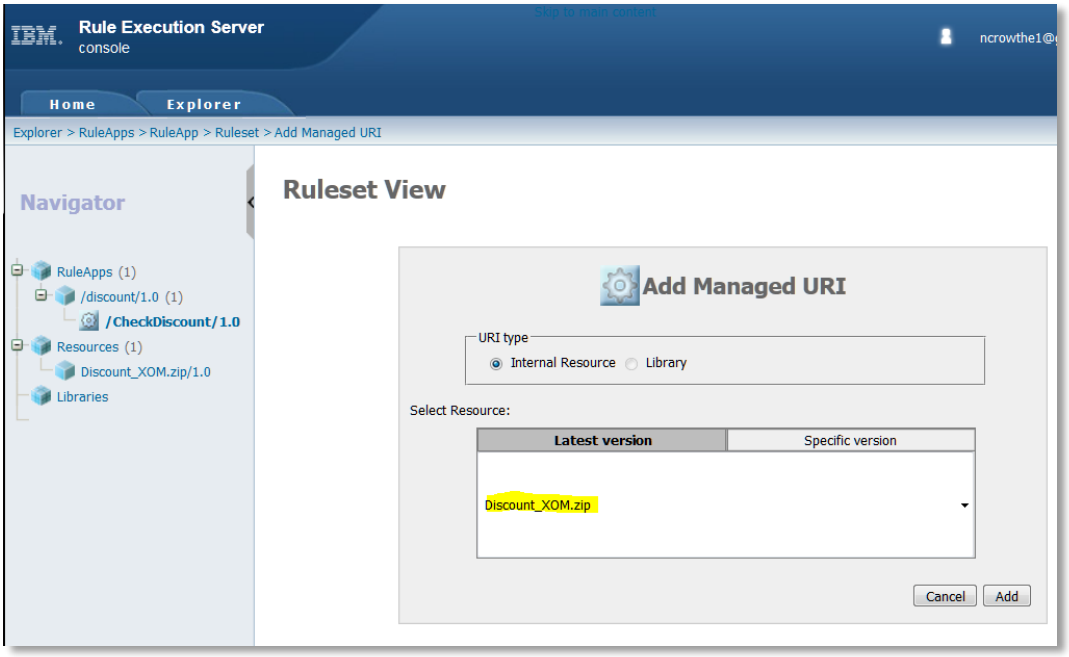


To make development changes to this decision service, you will need to download Rule Designer. This is beyond the scope of this tutorial, but the source code for the rules is also provided in section z

Go to the CheckDiscount service and select Add Managed URI:

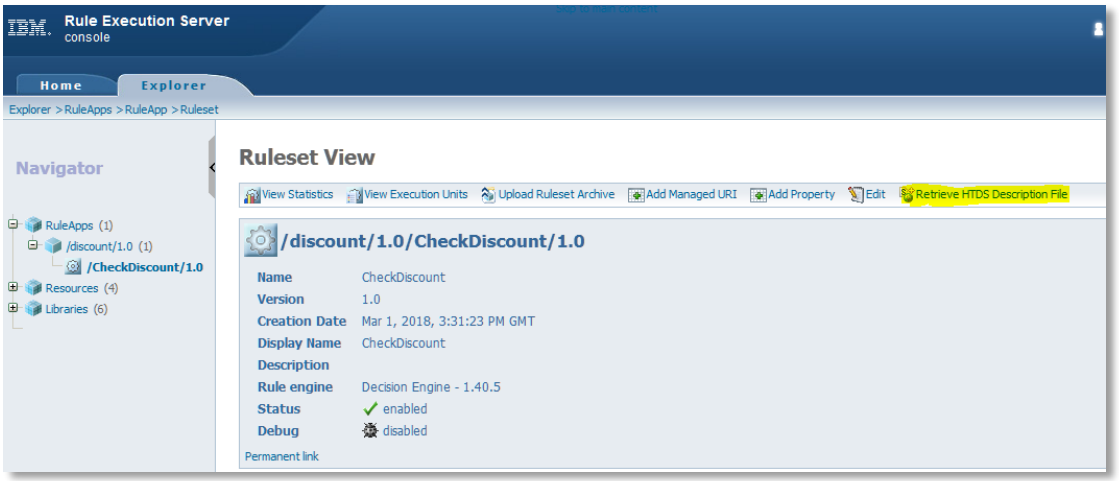


Add Discount_XOM.zip:

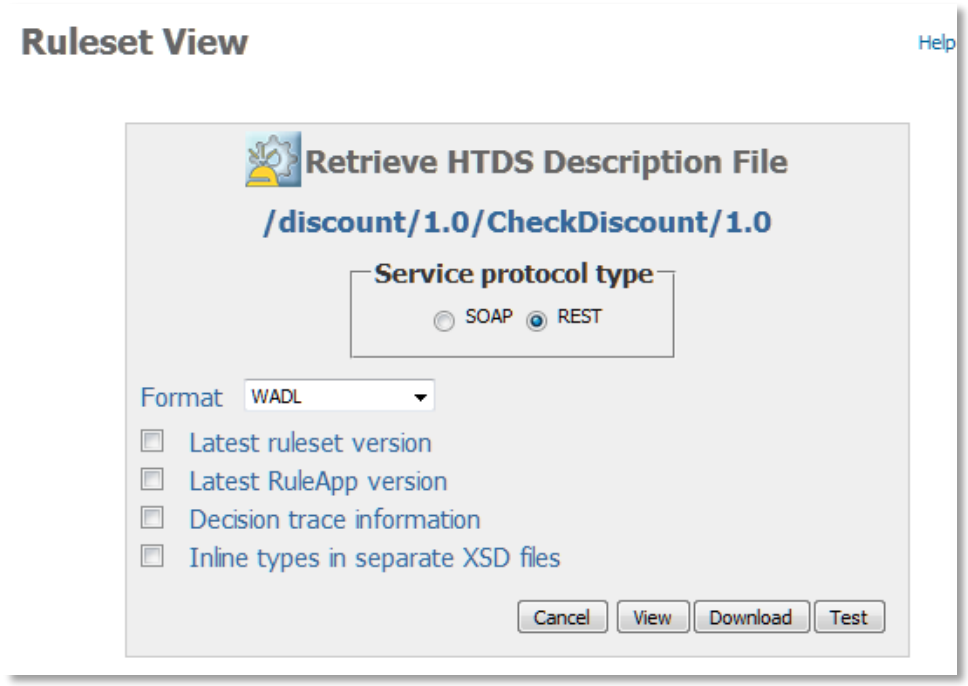


Test the decision service

Go back to the RES Explorer tab and select *Retrieve HTDS Description File*, highlighted below:



Select REST as the Service Protocol Type:



Press the *Test* button. Select JSON as the Execution Request type. Paste the following JSON payload in to the request:

```
{
  "__DecisionID__": "0",
  "customer": {
    "customerId": "123456",
    "purchase": {
      "sku": "3",
      "purchaseTimestamp": "2018-03-08T01:49:45.000+0000"
    },
    "discount": 0,
    "discountReason": "",
    "purchaseHistory": [
      {
        "sku": "1",
        "purchaseTimestamp": "2018-03-01T01:49:45.000+0000"
      },
      {
        "sku": "2",
```

```
        "purchaseTimestamp": "2018-02-28T01:49:45.000+0000"
      }
    ]
  }
}
```

You should see the following result:

```
{
  "__DecisionID__": "0",
  "customer": {
    "customerId": "123456",
    "purchase": {
      "sku": "1",
      "purchaseTimestamp": "2018-03-08T01:49:45.000+0000"
    },
    "discount": 10,
    "discountReason": "10% discount for 3 purchases in last 60 days",
    "purchaseHistory": [
      {
        "sku": "1",
        "purchaseTimestamp": "2018-03-01T01:49:45.000+0000"
      },
      {
        "sku": "2",
        "purchaseTimestamp": "2018-02-28T01:49:45.000+0000"
      },
      {
        "sku": "3",
        "purchaseTimestamp": "2018-02-08T01:49:45.000+0000"
      }
    ]
  }
}
```

This response proves the discount has been correctly applied.

Node-JS

If you are not familiar with node-js, refer the tutorial below:

Node-js Tutorial

Edit the app.js

Extract the DecisionServiceFacade-nodejs project from Git and save to your local drive. Open *app.js* in the top level of this folder.

Go to line 35:

```
// The section below provides the credentials to bind to the Cloud services when running node-js locally
{
        var rules = {
                "executionRestUrl" : "https://xxxxxx/odm/dev/DecisionService/rest",
                "user" : "xxxxxxx",
                "password" : "xxxxxxxx"
        };

        var appEnv = {
                        "services" : {
                                "cloudantNoSQLDB" : [ {
                                        "credentials" : {
                                                "url" : "xxxxx"
                                        },
                                        "label" : "cloudantNoSQLDB"
                                } ]
                        }
                }
}
```

Replace the credentials blanked out with x's with your ODM on Cloud and Cloudant credentials.

Initialise the libraries

Connect to cloud foundry (see Node-js Tutorial) and install the required node-js libraries:

```
npm install
```

Run the application locally

Run your application locally:

```
npm start
```

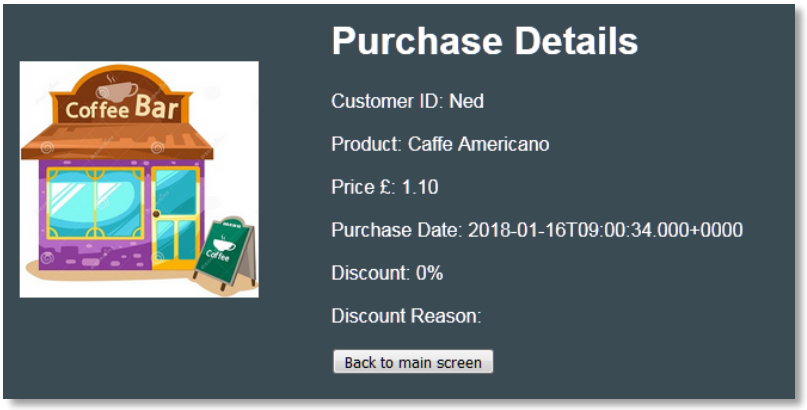
After a few seconds you should see the application start on port 3000.

In a web browser, go to the following URL:

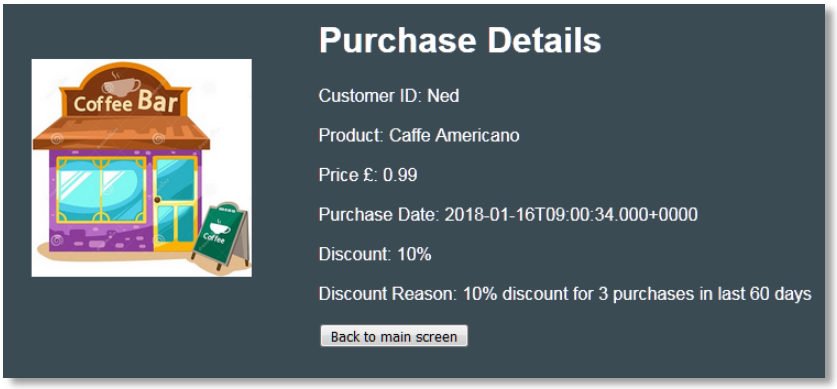
http://localhost:3000/

You should see the following screen:

Click the *Submit* button. On the first two attempts no discount will be applied. See below:



On the third attempt the stateful rule fires, applying a 10% discount:



Where to go now

Now that you have successfully run your first ODM Cloudant application you can start to look at developing your own. We suggest changing the sample code first to get a feel for what it is doing. Then feel free to create your own stateful ODM Decision Services using Node-JS and Cloudant!

Contact the author if you need assistance: *ncrowther@uk.ibm.com*