**Oracle Cloud Test Drive for PaaS and IaaS**

Lab 1.0

Integration Cloud Service \ ICS Basic Mapping Lab

Contents

[2 Lab Scenario 2](#_Toc477421256)

[2.1 Logging in to ICS 2](#_Toc477421257)

[2.2 The ICS Environment 3](#_Toc477421258)

[2.2.1 Connections 4](#_Toc477421259)

[2.2.2 Integrations 4](#_Toc477421260)

[2.2.3 Dashboards 4](#_Toc477421261)

[2.2.4 Lookups 4](#_Toc477421262)

[2.2.5 Packages 4](#_Toc477421263)

[2.2.6 Agents 4](#_Toc477421264)

[3 Creating Connections 5](#_Toc477421265)

[3.1.1 Create Service Cloud connection 5](#_Toc477421266)

[3.1.2 Create REST Connection 9](#_Toc477421267)

[4 Creating Integrations 12](#_Toc477421268)

[4.1 Create the Basic Map Data Integration 14](#_Toc477421269)

[4.1.1 Map Request Data for Create Organization 22](#_Toc477421270)

[4.1.2 Map Response Data for Create Organization 23](#_Toc477421271)

[4.1.3 Add Tracking Information 25](#_Toc477421272)

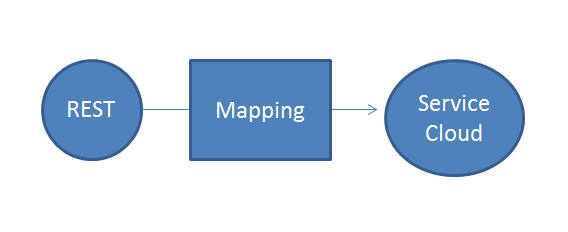
[4.2 Activate the Integration 25](#_Toc477421273)

[4.3 Test the Integration 27](#_Toc477421274)

Introduction to the Lab and Logging-in

# Lab Scenario

Our fictional company is using Oracle Service Cloud (RightNow) as their Field Service SaaS application. Business has requested that IT delivers a mobile application for the field service personnel – one that will enable them to quickly query and create customer records in RightNow whilst on the road.

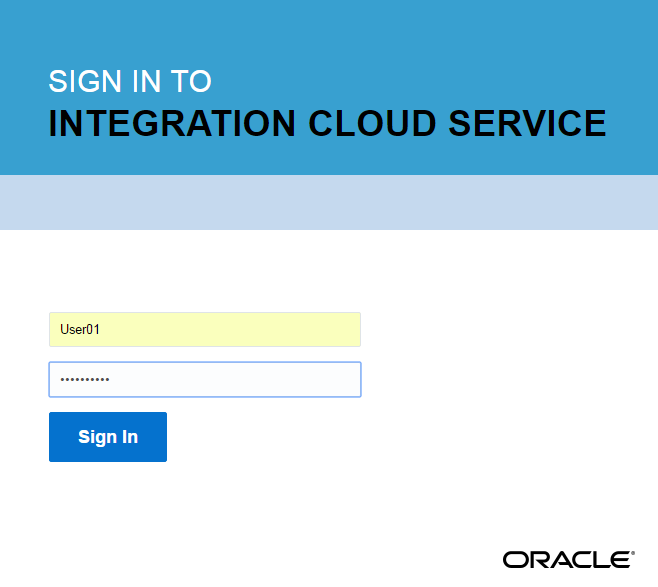
Note that in RightNow customers are known as “organizations”. To achieve their goal, our developers require a simple REST interface which masks the complexity of the “organization” data structure in Service Cloud. To this end, we will leverage Oracle Integration Cloud Service (ICS) to create a basic **Map Data** pattern integration between Rightnow and a mobile REST connection.

We will then use that to connection to create a new “organization” (i.e. customer) in Oracle Service Cloud.

## Logging in to ICS

Your instructor will assign you a URL and a username with a password to login. Please view the relevant “Access Credentials for all Labs” document.

Your instructor will also assign a Prefix – you will not need it for now, only for later.

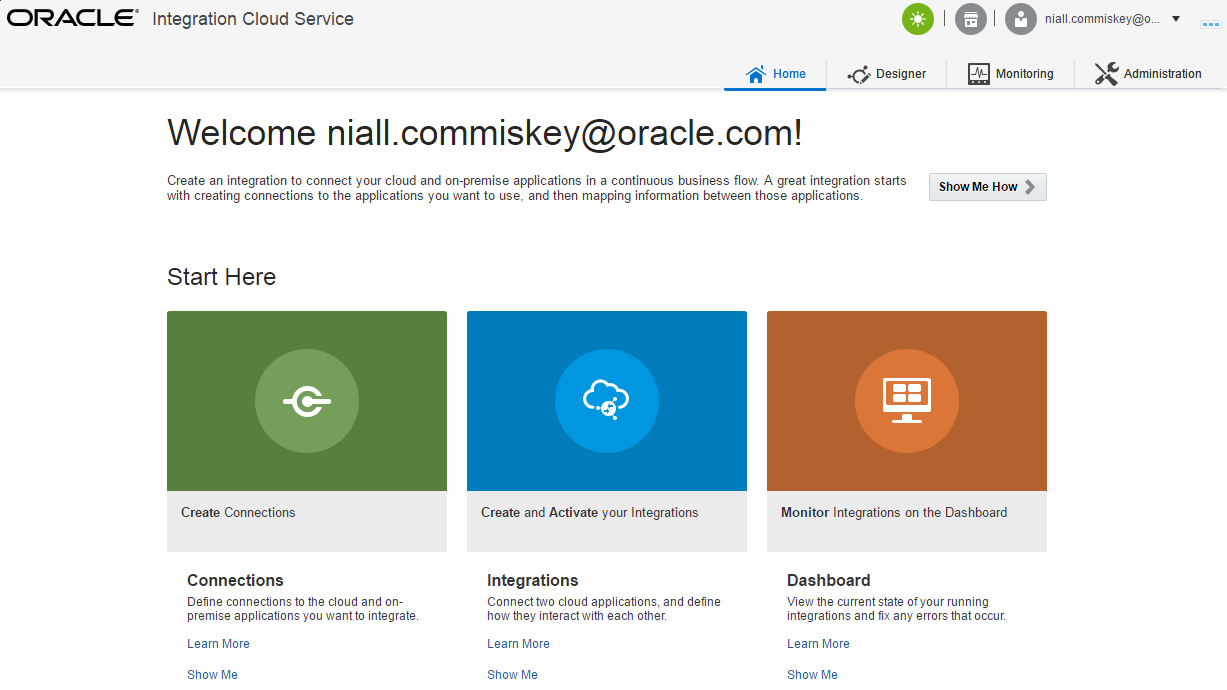
To avoid any possible cache issues:

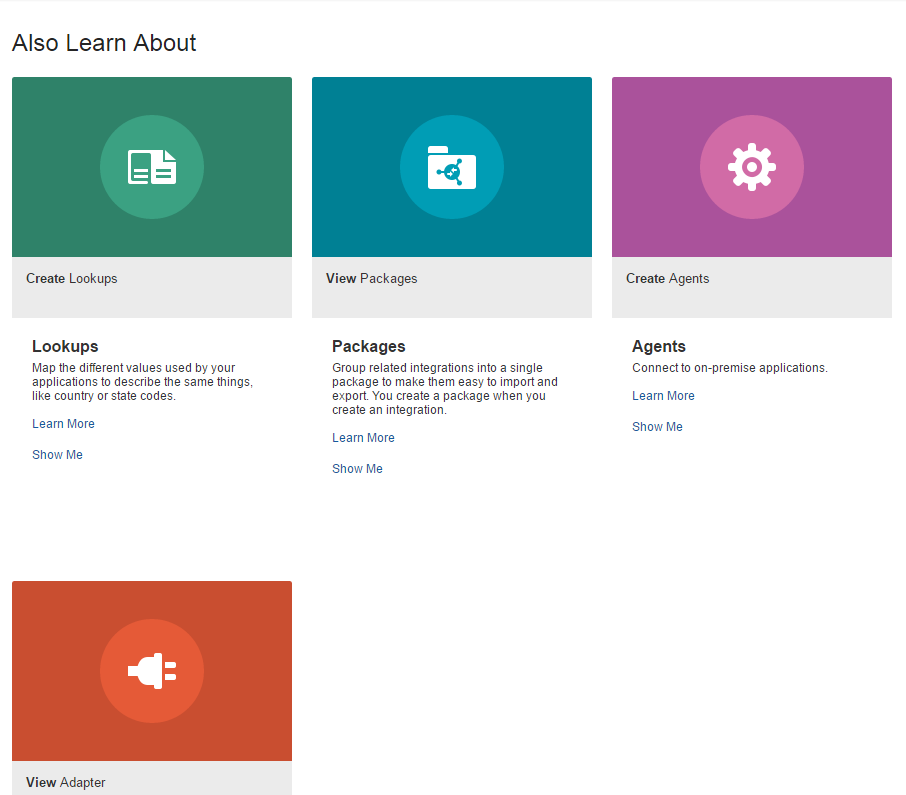
* In Chrome - Select New Incognito Window
* In Mozilla - Select New private window
* In Safari – Select File > New Private Window

Login with the credentials assigned to you

## The ICS Environment

Here is what you will see as soon as you login into Integration Cloud Service:





Let’s go through each component quickly. There is nothing to do here yet.

### Connections

Connections connect you to the apps with which you want to integrate. In our case, we will need two connections, one for the REST interface, the other for Service Cloud (RightNow).

Connections are based on Adapters. ICS provides 40+ adapters out of the box. Some of these are for SaaS apps such as Sales Cloud or Salesforce. Others are technical adapters such as SOAP and REST. ICS also provides on-premise adapters for E-Business Suite, Oracle DB, file etc.

### Integrations

Integrations connect applications to each other, in our case, the mobile interface to Service Cloud.

Integrations leverage connections and usually involve some mapping of data. In many cases, where we have App A connecting to App B, both may have the same logical business object e.g. Customer. However, the actual object field names may differ greatly.

### Dashboards

Dashboards provide us with the ability to monitor our integrations. Through the use of payload Tracking fields, we can see, for example, when a specific customer got created in Service Cloud.

### Lookups

Lookups help us address where different apps use different values for the same things e.g.

|  |  |
| --- | --- |
| Sales Cloud - AddressType | Service Cloud - AddressType |
| 1 | Billing |
| 2 | Shipping |

### Packages

Packages allow us to group related integrations and their artefacts to make it easier to import and export them.

### Agents

The ICS on-premise agent allows us to create secure hybrid integrations between cloud and on-premise. A possible use case includes sales account synchronization between Sales Cloud and on-premise Siebel.

Next, let’s start to implement our scenario.

Lab Execution

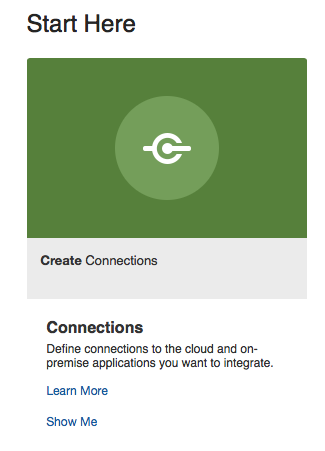
# Creating Connections

As already mentioned, you will need 2 connections for this lab. You will create them now. Ensure you are logged in as per the instructions above.

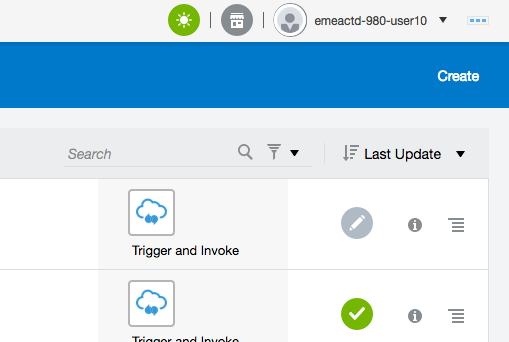
### Create Service Cloud connection

You will create a connection called **<Your\_Prefix>ServiceCloud** such as **AA1ServiceCloud**.

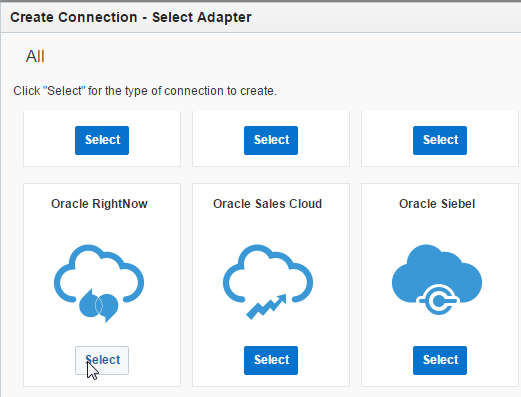
* Click **Create Connections**,



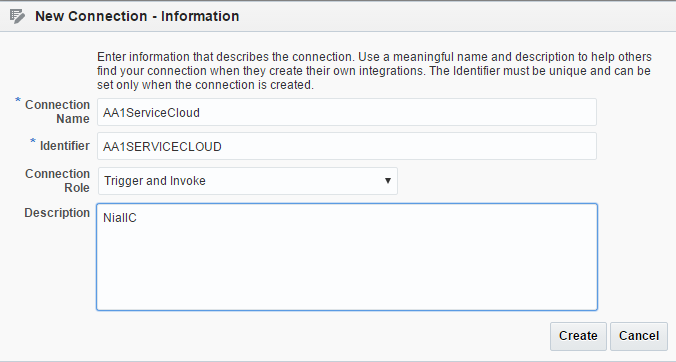
then **Create (upper right cornet of the window)**



* Select the **Oracle RightNow** adapter (You will need to scroll down the list of adapters)

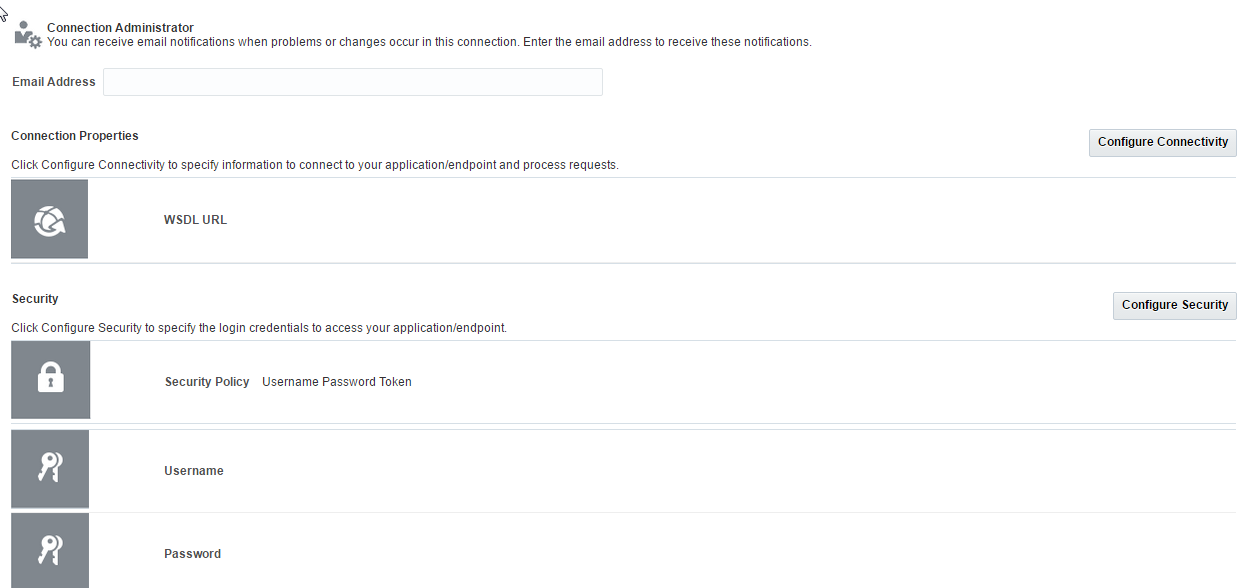


* Use the code assigned to you as the prefix of the connection name to give it a name of **<Your\_Prefix>ServiceCloud**
  + In my case below, I **used AA1 to give my connection the name AA1ServiceCloud**



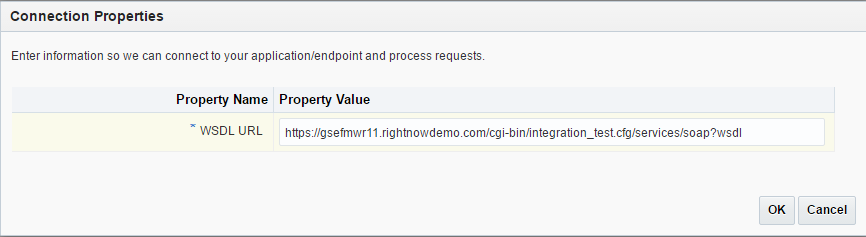
* Click **Create**

You will now see the following:

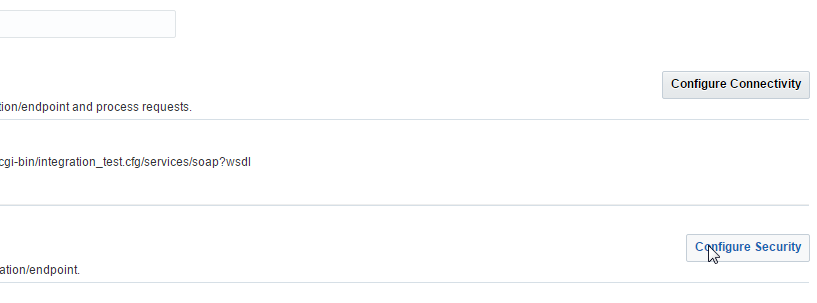


* Click **Configure Connectivity**.
* This will allow us to specify the enterprise WSDL. This contains a list of Service Cloud public business objects and the operations that can be performed on them e.g. to **create Organization**.
* Use the following:

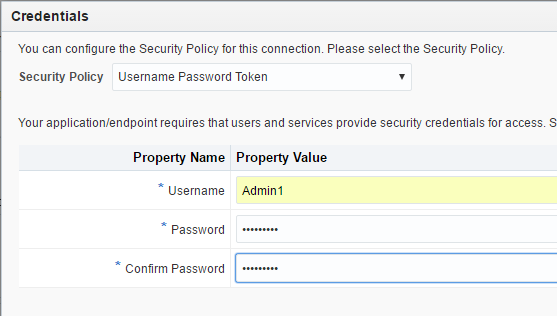
**https://gsefmwr11.rightnowdemo.com/cgi-bin/integration\_test.cfg/services/soap?wsdl**

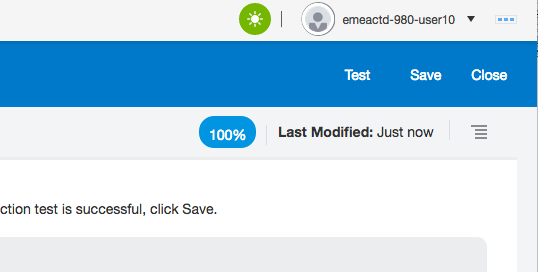


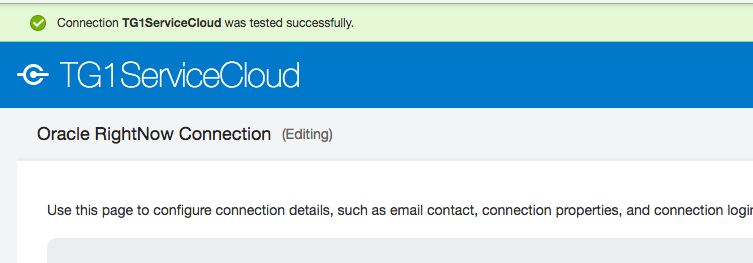
* Click **Configure Security**.
  + Here we will specify the Service Cloud user and password, which will be used to connect.



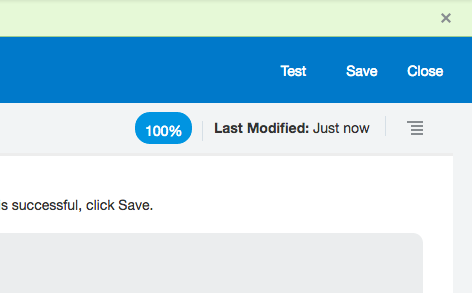
* + Enter **Admin1** for Username
  + Password – ***check your Access document for RightNow Adapter password***



* In the upper right Click **Save** and then, **Test**.
* 



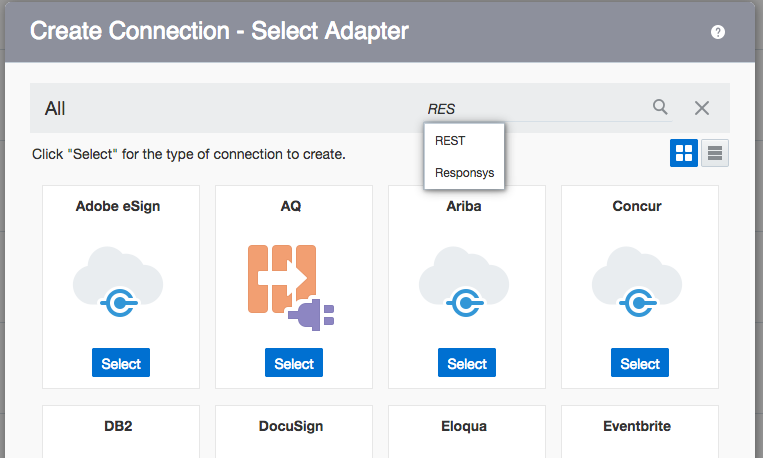
* If it was successful then on the upper right Click **Save** and **Close**



(If you forget to click save then you will be prompted if you want to save or discard your changes on closing the new connector window, chose save)

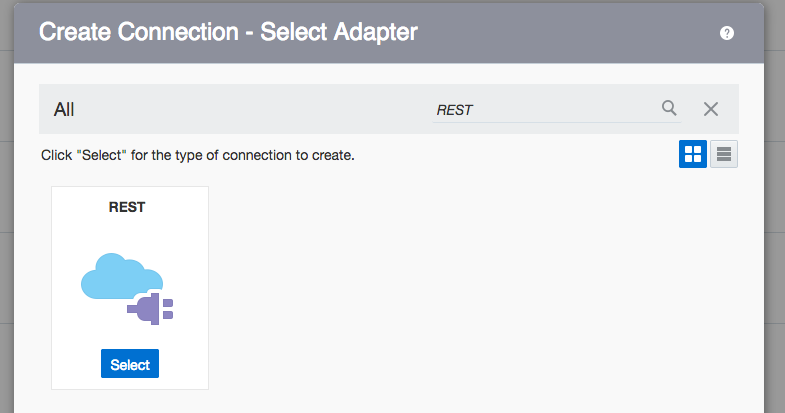
### Create REST Connection

* Click **Create** (upper right) to create another adaptor
* Select the **REST** adapter. To speed this up type the word RES into the Search field in the Create Connection – Select adaptor popup

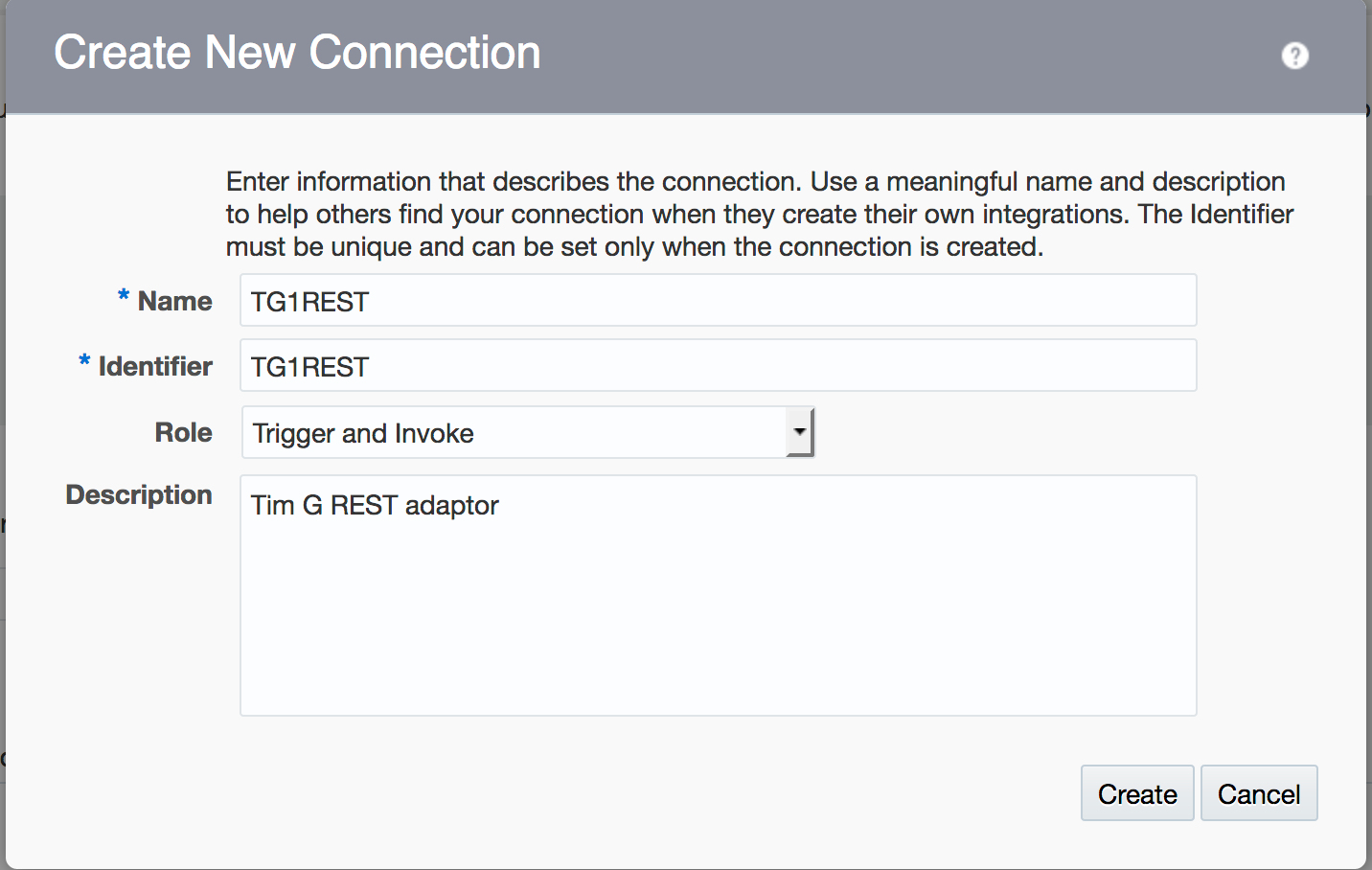


The UI will search and return possible names that match, select the REST option that’s returned

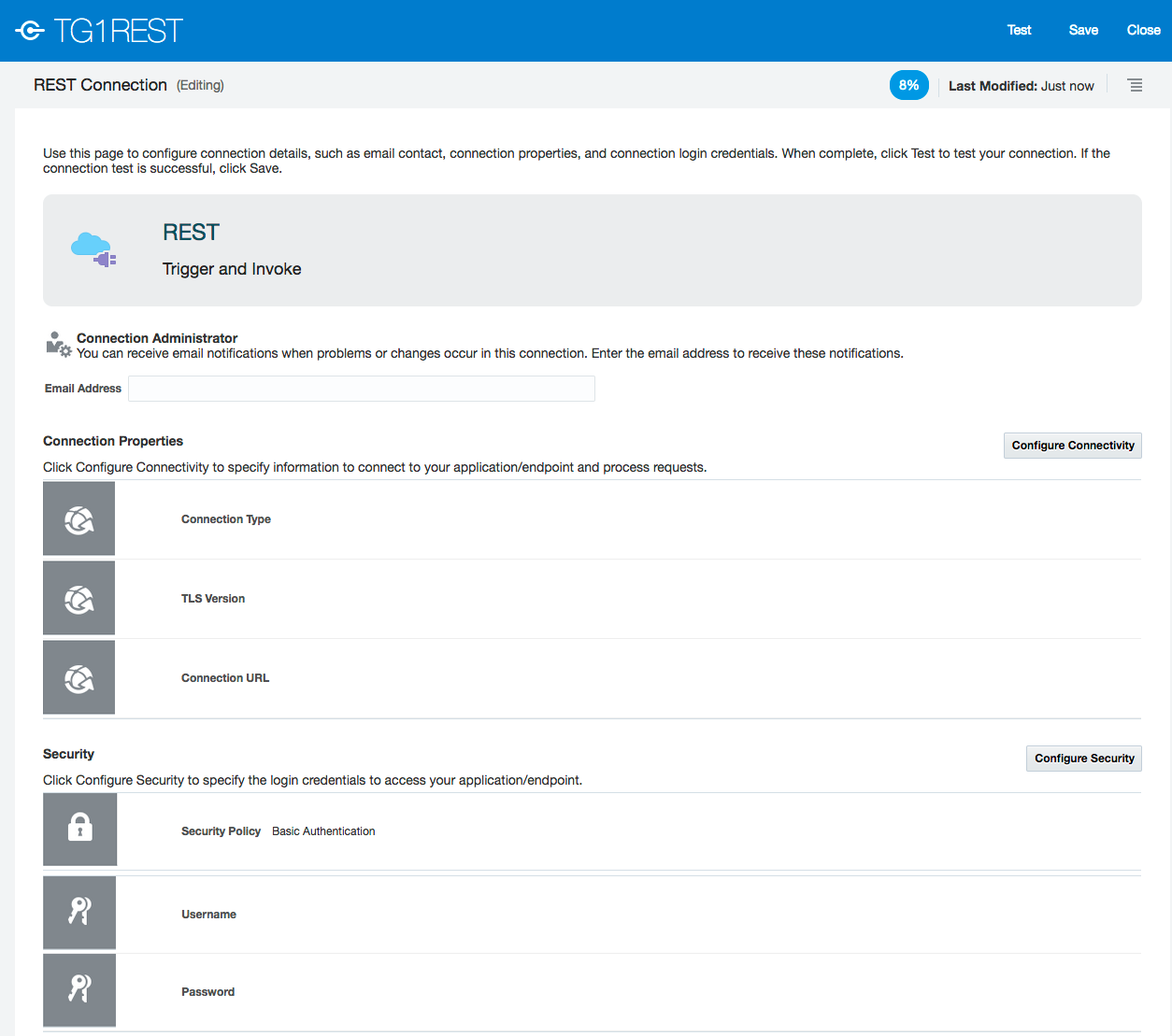
The UI will update to show adaptors that match, select the REST adaptor



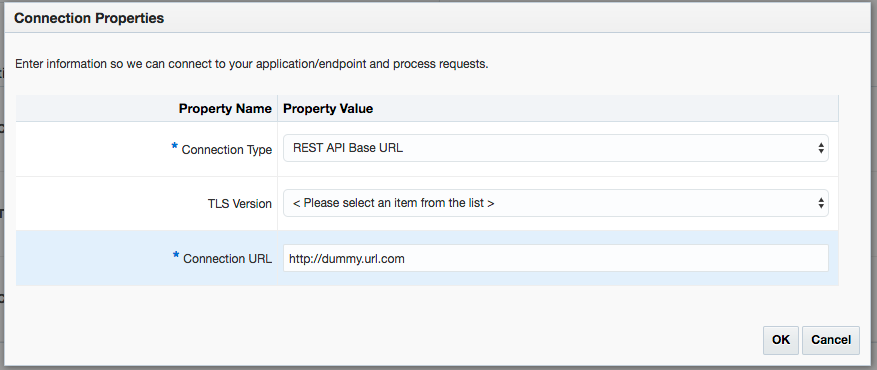
* Enter the relevant **Connection Name**, again, using the prefix allocated to you.
* Set the **Role** to be **Trigger & Invoke** (this is the default)
* Click **Create**

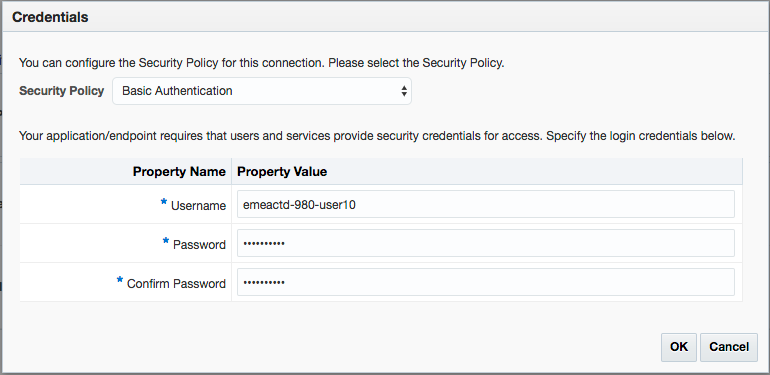


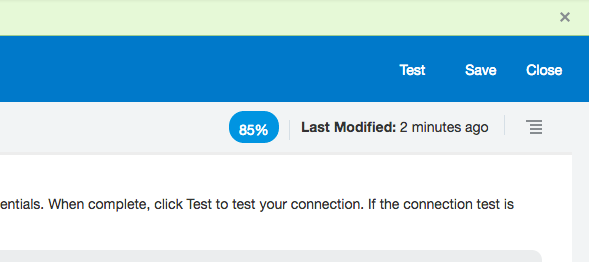
In the next screen showing your newly created adaptor

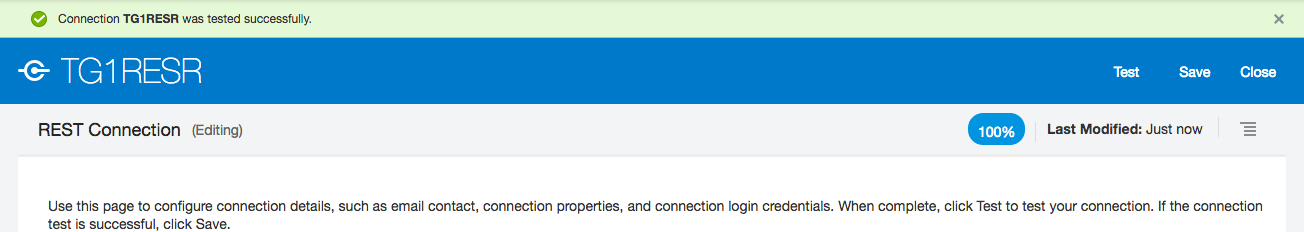


* Click **Configure Connectivity**.
* Select Connection Type as **REST API Base URL**
* Leave **TLS Version** blank
* For the **Connection URL** enter a dummy URL. We will specify the actual URL details later, when we leverage this connection at design time – [http://dummy.url.com](http://shaukatdesai.com)
* Click **OK**



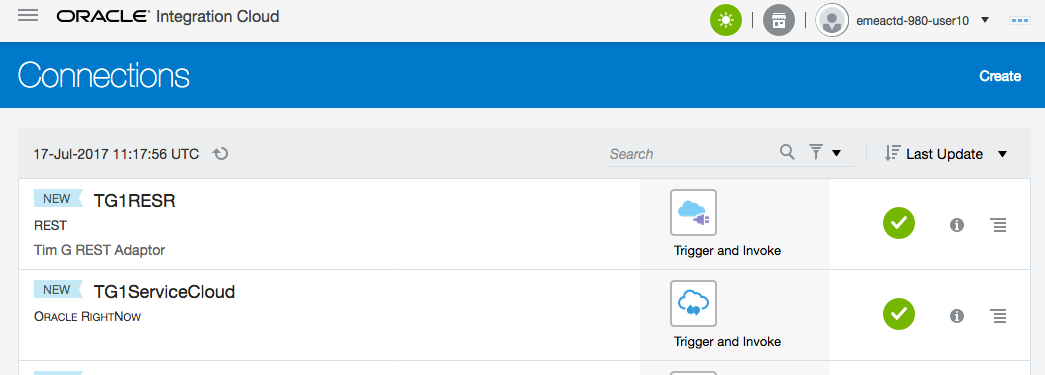
* Click **Configure Security** to set the security credentials.
* Set Security to **Basic Authentication**
* Enter your ICS username (e.g. emeactd-908-user10)
* Enter your ICS password in both fields (e.g. Paudge@000 )
* Click **OK**
* 
* Click **Save** and then **Test**. (Upper right of the window)





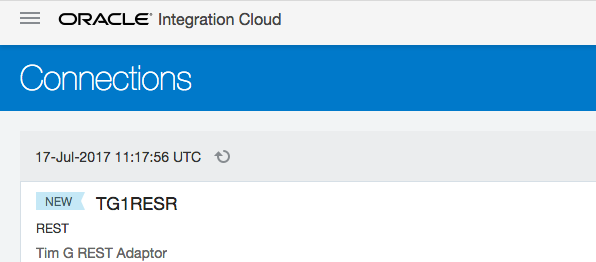
* Click **Save** and **Close**.

The two connections are now ready for use:

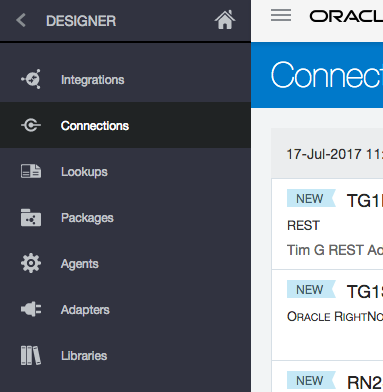


# Creating Integrations

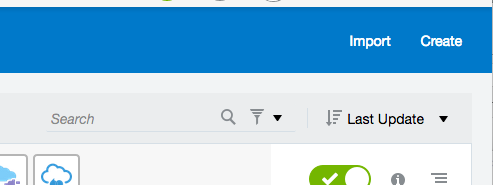
As already mentioned, we will do a simple data mapping integration. Begin by clicking on the “**Hamburger”** icon (Upper left) to “pull” the wide menu out



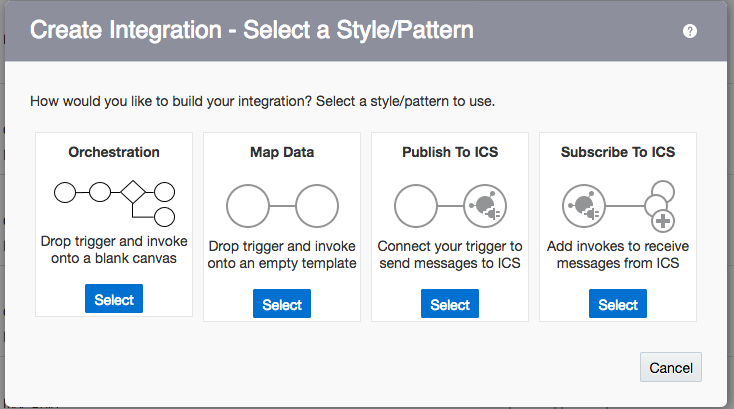
Then the **Integrations** option



* Click **Create** on the upper right of the window



You will see the following options to choose from.

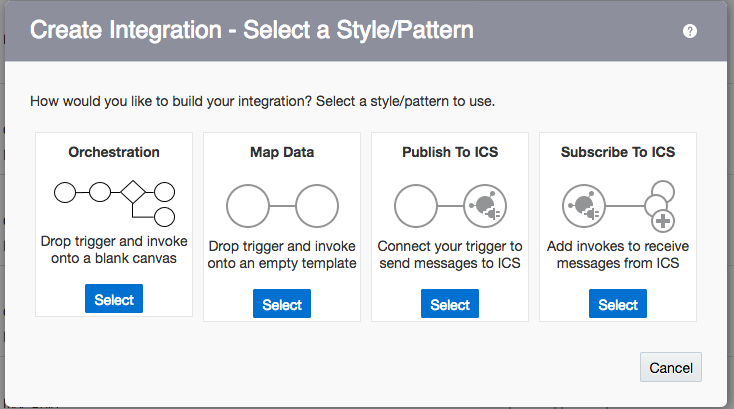


Let’s look at each of the above:

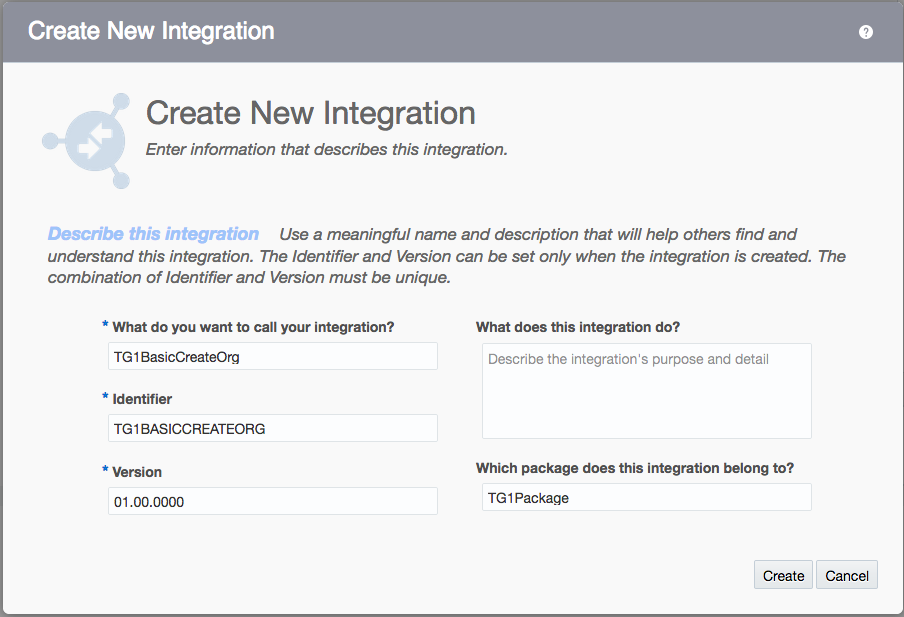
|  |
| --- |
| FYI (no steps to follow here)  **Map Data Pattern**  This is the pattern we will choose for this lab.  Use this for doing basic synchronisation between application business objects, e.g. syncing Sales Accounts from Sales Cloud with Organizations from Service Cloud. This pattern has the format:    We read the diagram above from left to right:   * ICS is called(Triggered) with a Sales Cloud Sales Account Business Object. * This is transformed into a Service Cloud Organization Business Object vua the Mapping. * Service Cloud is called(Invoked) to create the new Organization.   This pattern can also include content based routing as shown in the next page    **Orchestration Pattern**  Allows us to do more complex processing, such as checking whether the organization exists, before we attempt to create it.  **Publish/Subscribe to ICS Patterns**  Allows us to implement Pub/Sub style integrations. |

## Create the Basic Map Data Integration

* Select the **“Map Data”** option from the list of options

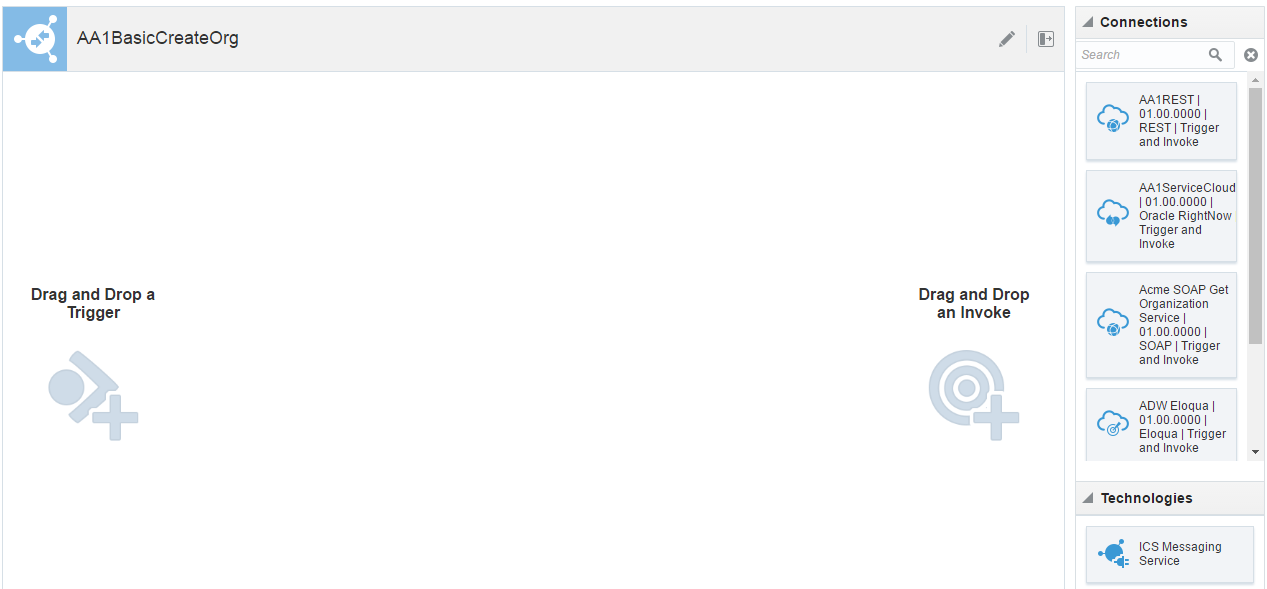


* Enter a name for your integration using your prefix and “BasicCreateOrg” or anything else you wish.
  + Example: **TG1BasicCreateOrg**
* Type in a package such as **<Your\_Prefix>OrgPackage**

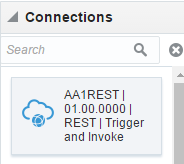


* Click **Create**

You will now see the integration canvas where you can drag and drop your connections.



On the far-right side, we see the connections available e.g. your REST and Service Cloud connections. We will drop the REST connection as the **Trigger** and the Service Cloud connection as the **Invoke**.

* Select your **REST** connection (You may have to scroll down the list and click the Load more option, or you can search on your prefix like with the adaptors) and drag and drop it on the Trigger icon. A popup wizard will appear

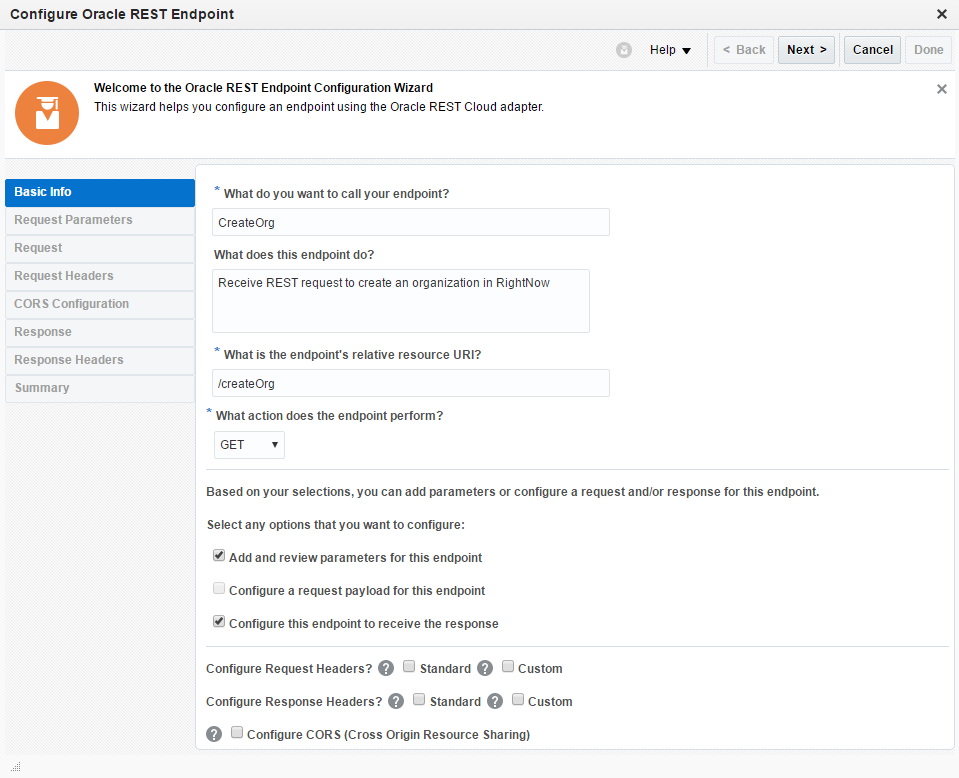


* Provide the following information on the wizard’s Basic Info page:

|  |  |
| --- | --- |
| **Field** | **Enter** |
| What do you want to call your endpoint | CreateOrg |
| What does this endpoint do? | Receive REST request to create an organization in RightNow. |
| What is the endpoint’s relative resource URI | /createOrg |
| What action does the endpoint perform | GET |
| Add and review parameters for endpoint. | Check this box |
| Configure this endpoint to receive the response | Check this box |

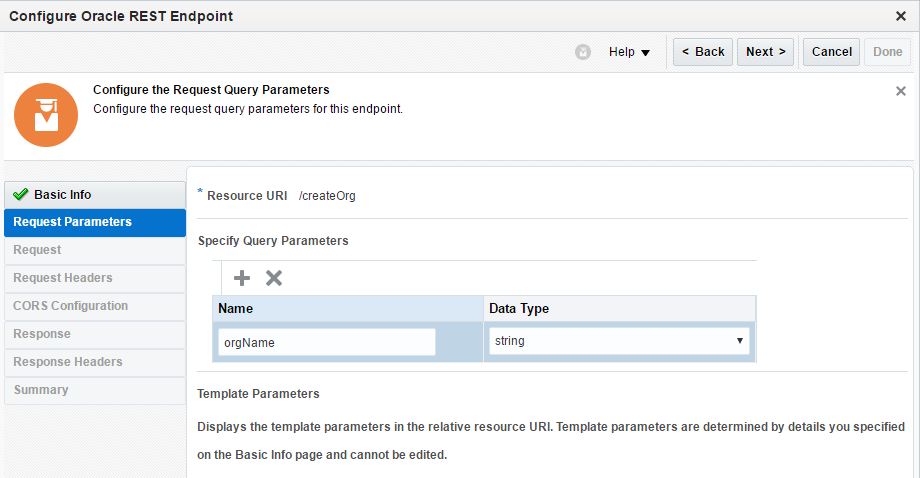
Your screen should look like the screenshot in the next page.

* If your screen looks like the below, click **Next**



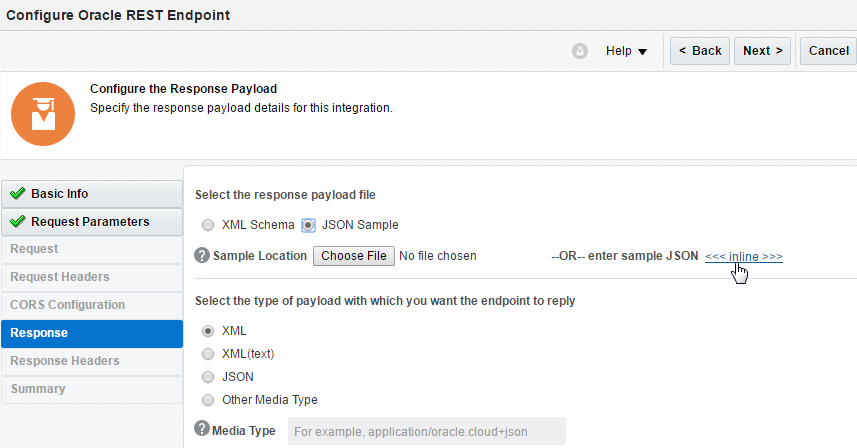
Case Sensitive

* Click the **+** and provide a parameter **Name** “orgName” **Data Type** “string” as per the screenshot below.
* Click **Next**.



You should now be in the Response panel:

* Select the **JSON Sample** radio button.
* Click on the **<<<<< inline >>>>>** link as shown below

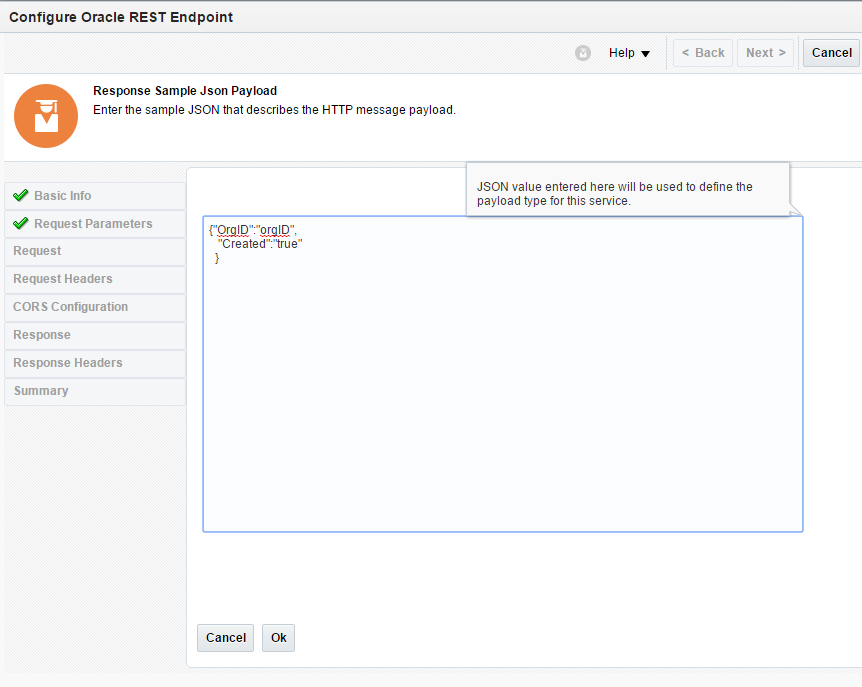


* Copy and paste in the **JSON** code below

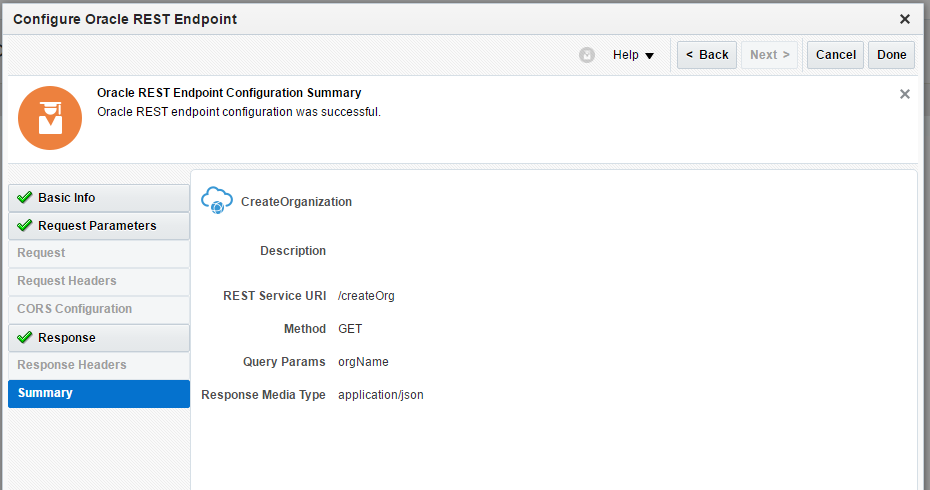
{"OrgID":"orgID",

"OrgName":"orgName"

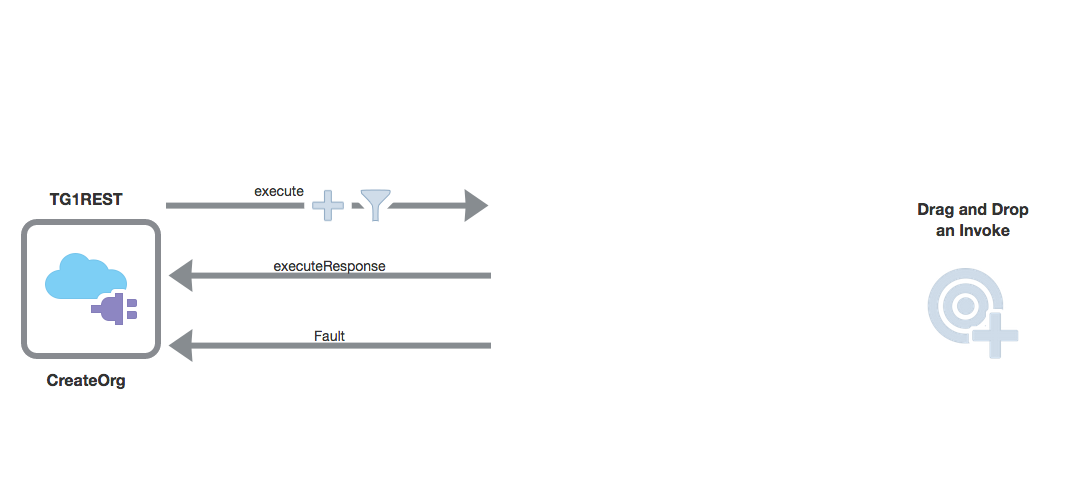
}



* Click **Ok** 🡪 then click **Next**.
* On the Summary screen, click **Done**.



Your **Integration** now should look like this:



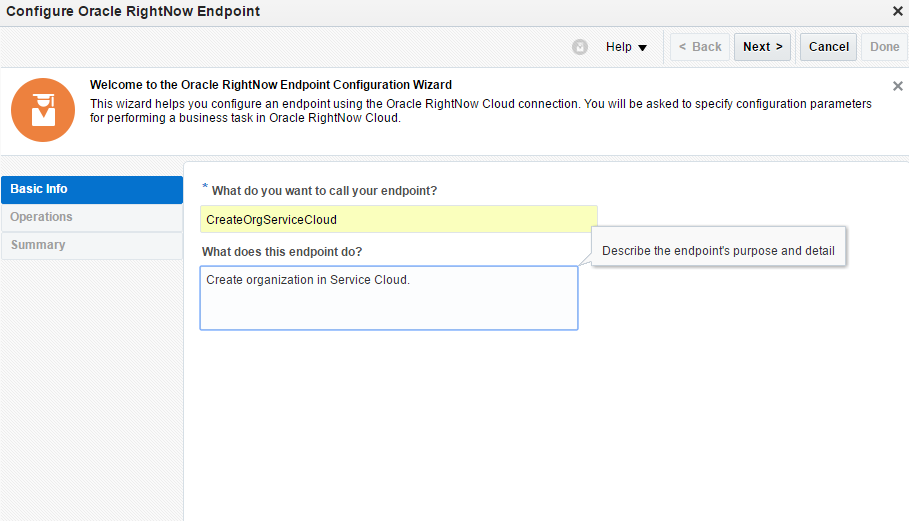
Next, we will leverage the Service Cloud (Rightnow) connector. **Drag and drop the Service Cloud connection** from the right panel and onto the **Invoke** area to the right.



* Provide the following information on the Basic Info page:

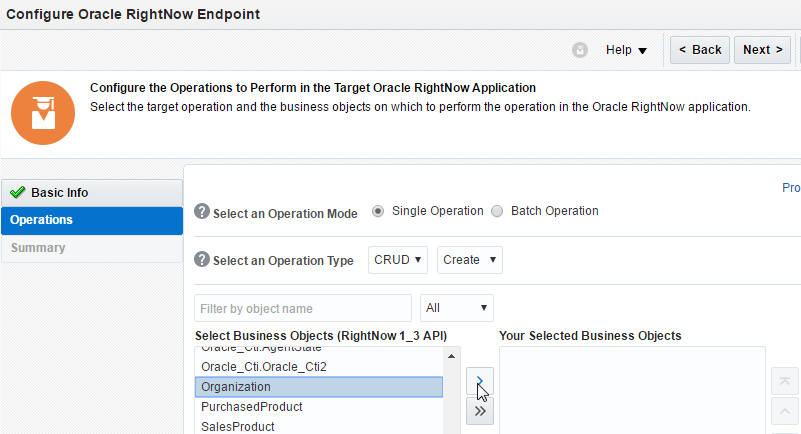
|  |  |
| --- | --- |
| **Field** | **Enter** |
| What do you want to call your endpoint | **CreateOrgServiceCloud** |
| What does this endpoint do? | **Create organization in Service Cloud.** |

Your screen should look like the below. If so, click **Next**

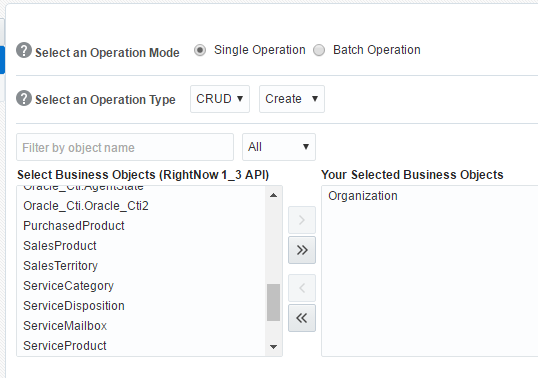


* Provide the following information on the Operations page, and click **Next**.

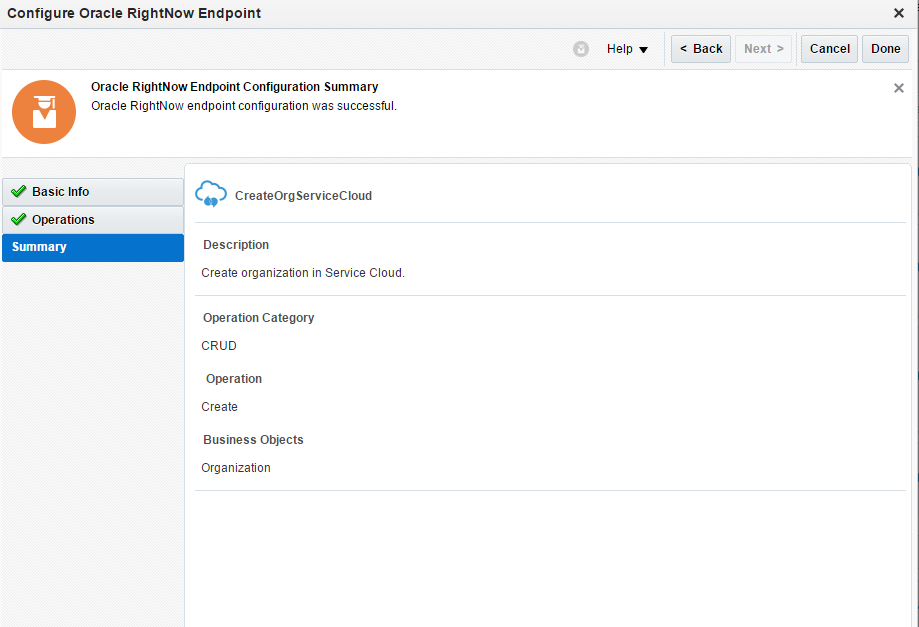
|  |  |
| --- | --- |
| **Field** | **Enter** |
| Select an Operation Mode | **Single Operation** |
| Select an Operation Type | **CRUD**, then **Create**. |
| Your Selected Business Objects | Select **Organization** from Select Business Objects (RightNow 1\_3 API) and use the right-arrow button to move it to the right listbox. |



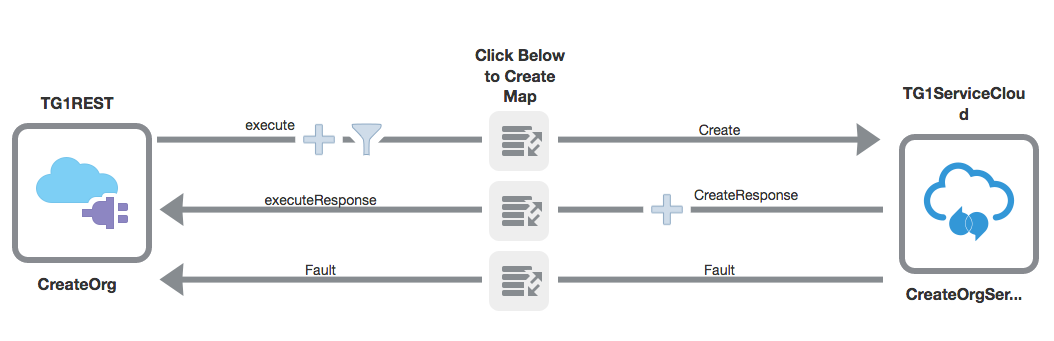
Your screen should look like the below:



* Click **Next** and then review your selections on the Summary page.
* Click **Done**.

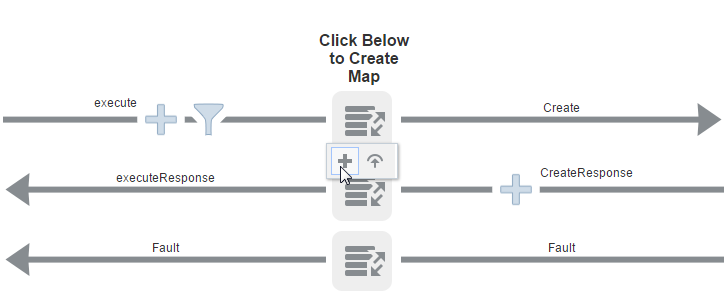


Your **Integration** should look as follows:



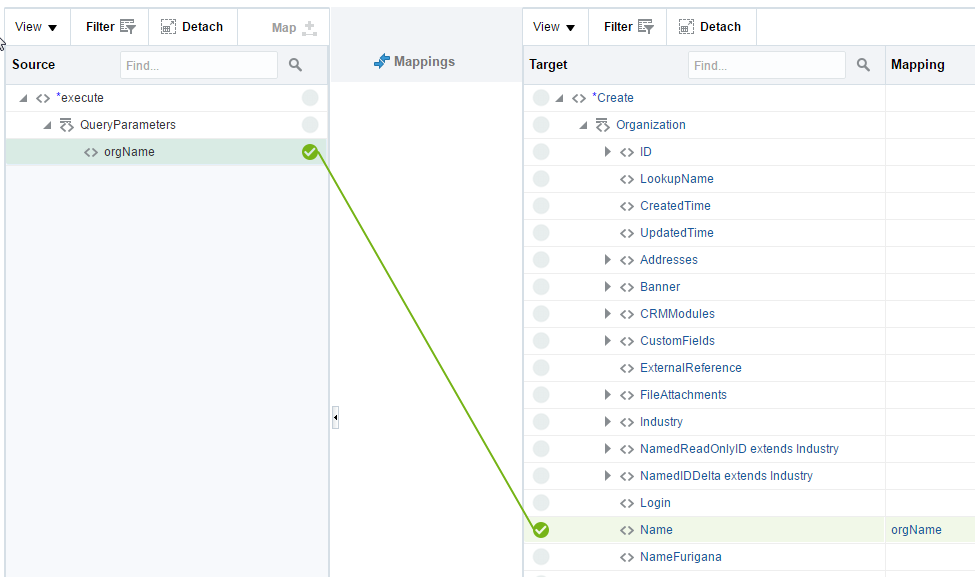
### Map Request Data for Create Organization

Click (left button) on the topmost map icon  soon a small plus button appears. Click the **+** icon to go into the data mapper.



Here we map the **orgName** from the direction of the REST request to the Organization name in Service Cloud.

* Drag and drop the **orgName** to **Name** using the grey circles 



* Click **Validate** to make sure the mapping is sensible (upper right)
* 
* Assuming the validation passes chose **Close**. To exit the mapper

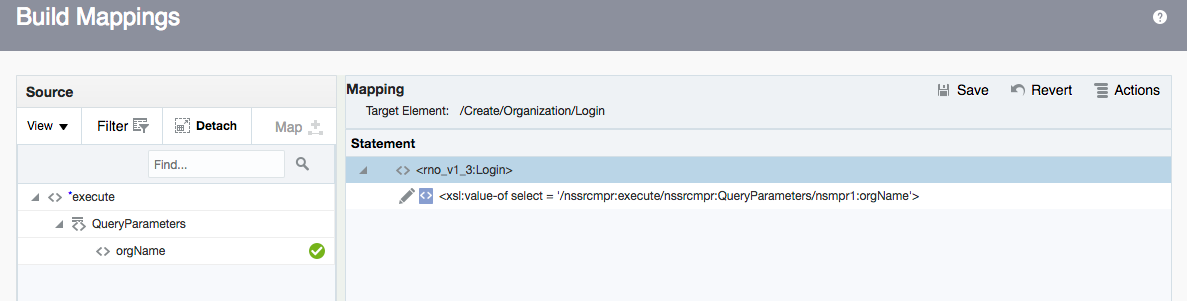
Correcting a mapping mistake

If you make a mistake with a mapping, for example shown connecting the orgName to the Login property select the mapping target (on the left column)

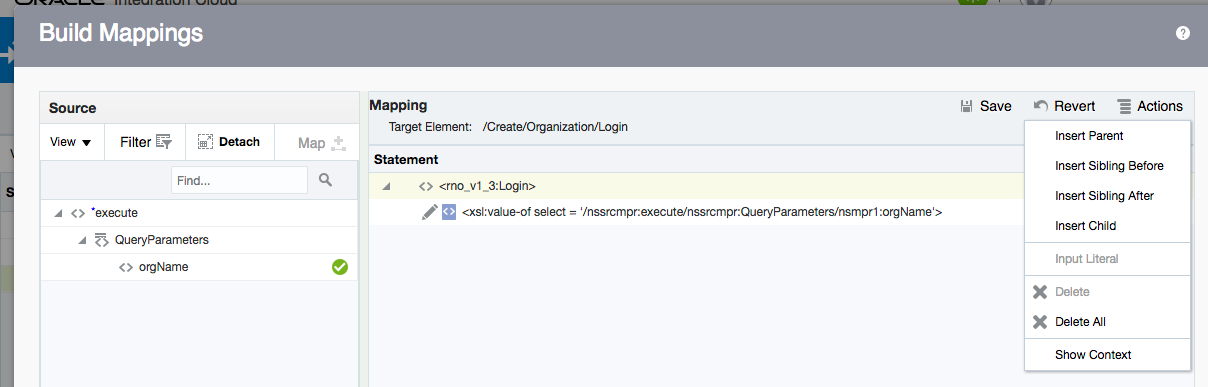


This will open up the Build mappings window (remember do this ONLY if you’ve made a mistake and want to remove a mapping)

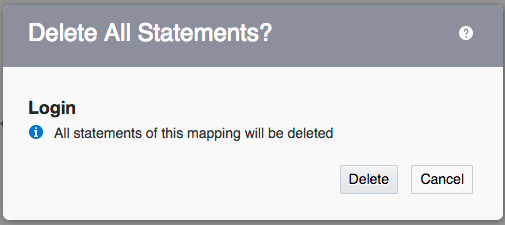
Select the Top level target element (in this case <rno\_v1\_3:Login>



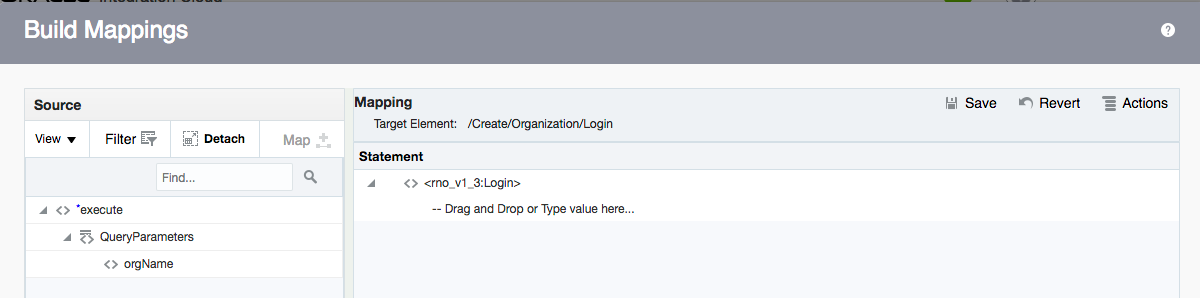
The in the Actions menu chose Delete All



In the resulting confirmation window confirm you want to delete the mappings

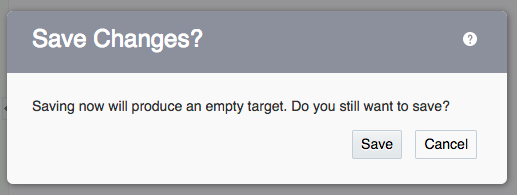


The mapping will be deleted



Click the Save option (upper right) to save the updated mapping with the mistake removed

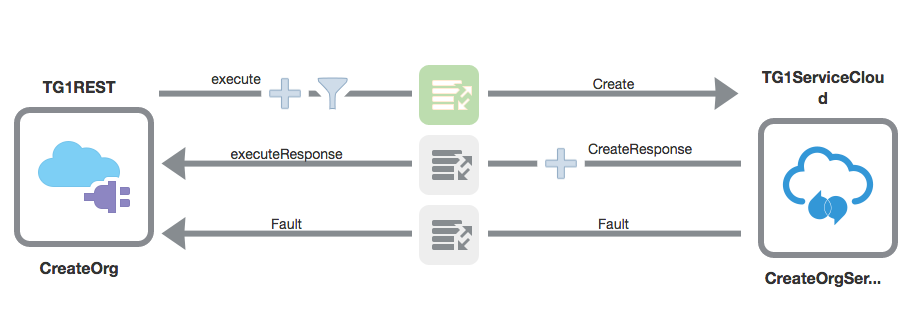
In There may be save confirmation popup, advising you that the target is now empty, click save to proceed



Then click close (Lower right) to exit the low level mapping builder

In the high level mapping screen you’ll see the mapping has been removed.

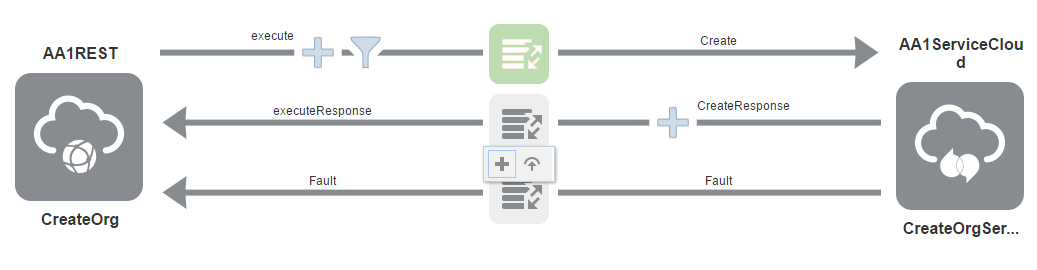
Now create the correct mappings as required and Validate and save as before



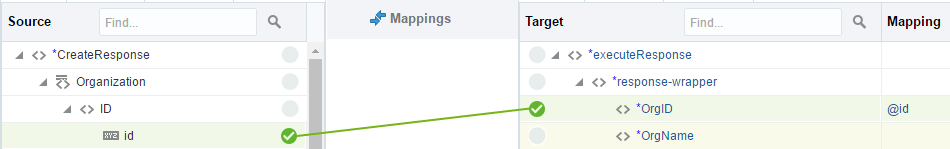
Note: the color of the mapping icon has changed to green, indicating it has been implemented.

### Map Response Data for Create Organization

Here we will map things in the opposite direction, from the direction of the Service Cloud application towards the REST connection. **Click on the second map icon** and select the plus button.

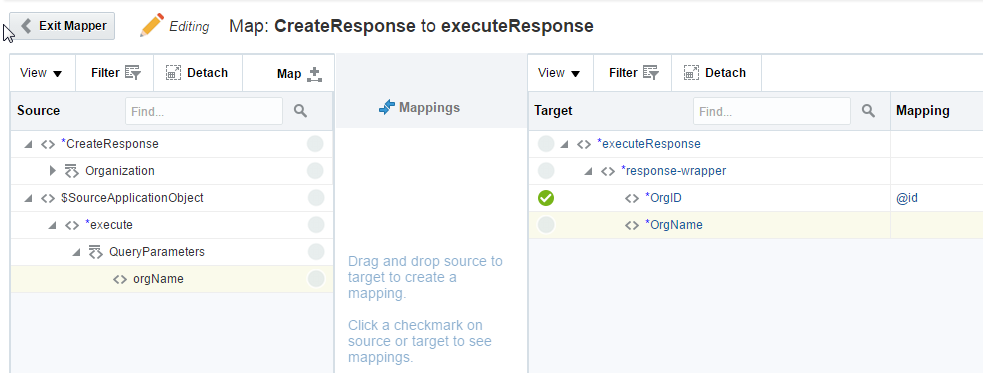


* Expand **CreateResponse 🡪 Organization 🡪 ID** 
  + Then map **id** (make sure you select the correct **id** !!) to **OrgID**

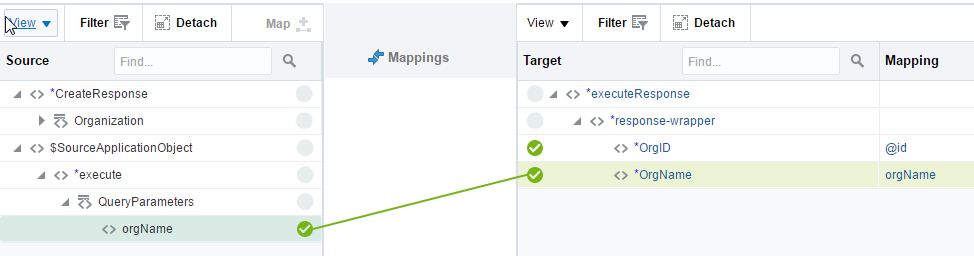


Now we will map the OrgName and for this we will use the orgName value from the REST request. It is still available to us in the Response Mapping.

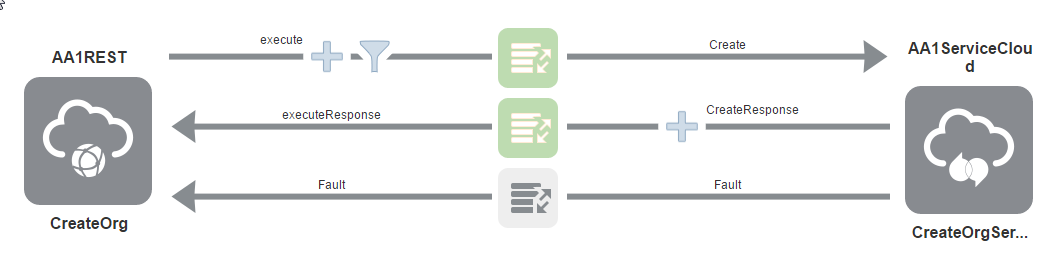
* Collapse the **CreateResponse** data structure.
* Expand **$SourceApplicationObject 🡪 execute** 🡪 **QueryParameters** so that you see **orgName**



* Map **orgName** to **OrgName**



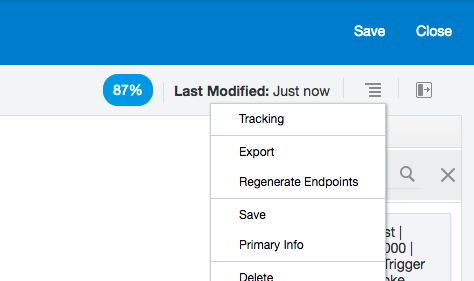
* Click **Save** and then click **Exit Mapper**.



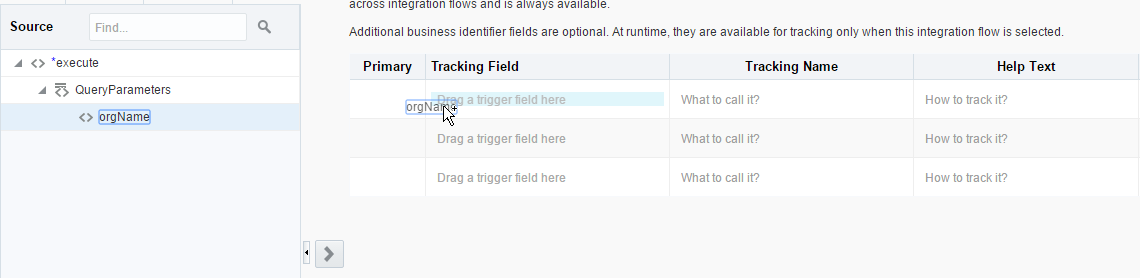
### Add Tracking Information

Setting Tracking field(s) allows us to track integration instances as they pass through ICS. Usually this will be a business-related key from the payload. We will use the orgName as our tracking field.

* Click the Manu drop down (upper right) and select tracking **Tracking** from the menu to bring up the tracking details screen



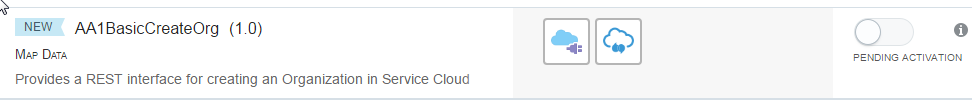
* Now drag **orgName** to **Tracking Field**



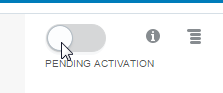
* Click **Done** 🡪 **Save**
* Click **Save,** then **Close** (Both upper right) to exit the integration

## Activate the Integration

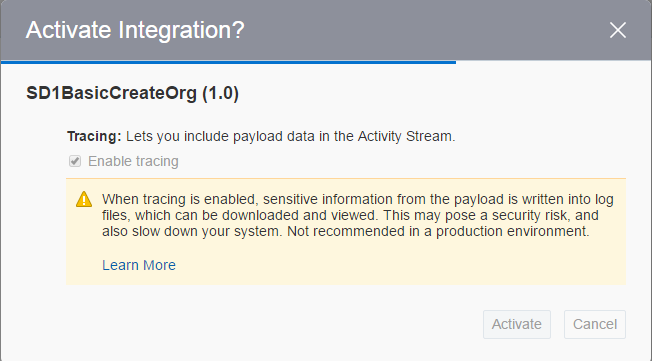
You should now be at the integrations list page where you should see the integration you just created.



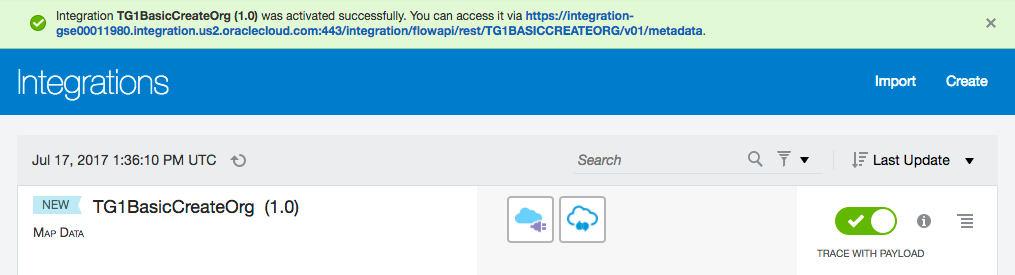
* Click the relevant button to **Activate**



* You will see a confirmation dialog. Before selecting **Activate**, make sure to **Enable Tracing**. And the sub option of **Include Payload**. This is useful for test purposes.



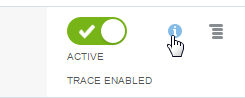
You will see your integration with Green Tick

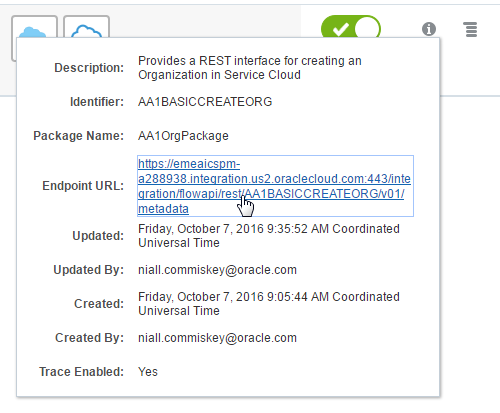


Note that a link to the integration is displayed in the top of the screen

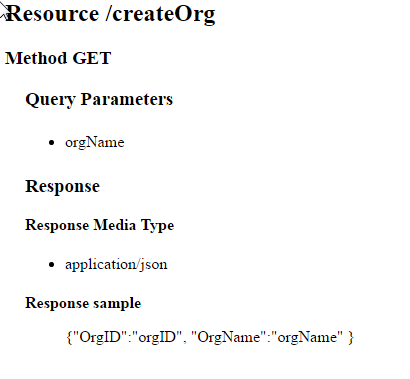
## Test the Integration

* Click on the Access linik in the top of the screen to go direct to the integration, if this is no longer available click the **info** icon – this will bring up a popup screen. Copy the **Endpoint URL**





* Open another window in your browser and visit that URL
  + You will be asked to enter your **ICS** username and password - **UserXX/Paudge2009**
* You should see something like the below

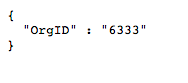


1. Copy the URL into notepad
2. Replace **metadata** in the URL with **createOrg?orgName=MyNewOrg*YourPrefix***

Example URL from another instance (yours will be pointing to different instance)

[https://emeaicspm-a288938.integration.us2.oraclecloud.com/integration/flowapi/rest/AA1BASICCREATEORG/v01/**createOrg?orgName=MyNewOrgAA1**](https://emeaicspm-a288938.integration.us2.oraclecloud.com/integration/flowapi/rest/AA1BASICCREATEORG/v01/createOrg?orgName=MyNewOrgAA1)

1. Now copy and paste new URL in another broswer tab and hit enter
2. The response is as follows.



What you have done is to call the REST connection with the new information and in turn call ICS to create the new organization in Oracle Service Cloud (RightNow).

Congratialtions you have completed the ICS lab.

If you finished with spare time as an optional bit of work you can test out your skills and update the mappings to include additional informartion from the, in this case including the query params

You will need to deactivate the integration to make any changes (The green slider, then confirm)

Edit the Integration by clicking on it’s name.

Edit the mapping by clicking left on the mapping icon, then the pencil to actually edit the mapping



Add a mapping of the $SOurceApplicationObject -> execute->QueryParameters->orgName to the response orgName

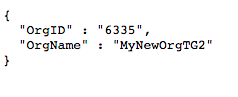
Remember to validate and close the mapping and to save and close the integration

Activate the Integration again with the slider switch

Access the integration again, specify a different org name in your URL

<https://integration-gse00011980.integration.us2.oraclecloud.com/integration/flowapi/rest/TG1BASICCREATEORG/v01/createOrg?orgName=MyNewOrgTG2>

See that the ordname is now in the response (and the orgId is different from the previous one)



This shows how the integration can make use of data not just from the back end service, but also from the input, this is important as not all back end servcies will include the input data in their response.