

Demo Introduction

Keywords:

Oracle Integration Cloud Service, Oracle Storage Cloud Service

Goal of Demo:

To demonstrate how easily to leverage Oracle Integration Cloud Service to build integration to pull Twitter feeds which the user cares about and store the data into Data Lake in Oracle Storage Cloud.

Pre-requisites:

User has registered both Oracle Integration Cloud Service and Oracle Storage Cloud Service. The storage containers for the demo have been configured properly.

User already has a Twitter account.

Demo Steps

Step 1: Create an Application in Twitter

This step will create an application in Twitter. However it is actually not an application running in Twitter. Instead it is for allocating OAuth 1.0 credentials for an remote application which can request Twitter services, for example retrieving feeds, creating Twitter message etc.

To create an application in Twitter, open a browser and access Twitter Application Management System with the URL: <https://apps.twitter.com/>. After a successful login with your Twitter account, you will see following Application Management page:

The screenshot shows the Twitter Application Management interface. At the top, there's a navigation bar with a Twitter icon, the text 'Application Management', and a user profile icon. Below the navigation bar, a blue header bar contains the text 'Twitter Apps'. To the right of the header, there's a 'Create New App' button. The main content area displays two application entries. The first entry is 'BigDataICS' with a blue gear icon, followed by the text 'list of hashtags from ICS'. The second entry is partially visible with a blue circular icon. At the bottom of the page, there's a footer bar with links for 'About', 'Terms', 'Privacy', and 'Cookies', and a copyright notice '© 2017 Twitter, Inc.' There's also a 'Tweet' button with a Twitter icon.

Click **Create New App** button to load the page below to create a new application.

The screenshot shows the Twitter Application Management interface. At the top, there's a blue header bar with the Twitter logo and the text "Application Management". On the right side of the header is a user profile icon with a dropdown arrow. Below the header, the main content area has a title "Create an application". The form is divided into sections:

- Application Details**:
 - Name ***: A text input field with a placeholder box below it. The placeholder text reads: "Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max."
 - Description ***: A text input field with a placeholder box below it. The placeholder text reads: "Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max."
 - Website ***: A text input field with a placeholder box below it. The placeholder text reads: "Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)"
 - Callback URL**: A text input field with a placeholder box below it. The placeholder text reads: "Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank."
- Developer Agreement**: A section containing a checkbox and a link. The checkbox is checked, and the link reads: "Yes, I have read and agree to the [Twitter Developer Agreement](#)".
- Having trouble creating your application?**: A section with a message: "If you're having trouble fulfilling application creation requirements, please contact our Platform Operations team by using the 'I have an API policy question not covered by these points' option of the contact form at <https://support.twitter.com/forms/platform>".

Name of the application must be unique among all users in Twitter. You'd better adding a user-specific prefix to the name. **Website** of the application must start with a protocol, either http or https in the demo. **Callback URL** is only necessary if the application need to request Twitter services as some other accounts. In the Oracle Integration Cloud Service, it only allows to use same account, so can leave the field empty.

After reading and accepting Twitter Developer Agreement, click **Create your Twitter application** button at bottom.

When the application is created, following application page will appear. You can review general information of the application. But the most critical information we need in the Oracle Integration Cloud Service is OAuth 1.0 credentials.



Your application has been created. Please take a moment to review and adjust your application's settings.

ICSBigData

[Test OAuth](#)[Details](#)[Settings](#)[Keys and Access Tokens](#)[Permissions](#)

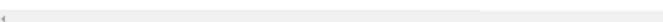
This app will be used to listen on #FlashbackFriday
<http://www.oracle.com>

Organization

Information about the organization or company associated with your application. This information is optional.

Organization None

Organization website None



Application Settings

Your application's Consumer Key and Secret are used to authenticate requests to the Twitter Platform.

Access level Read and write ([modify app permissions](#))

Consumer Key (API Key) [REDACTED] ([manage keys and access tokens](#))

Callback URL None

Callback URL Locked No

Select **Keys and Access Tokens** tab in the page. By default, only Consumer Key & Secret are generated. In this demo, we also need to allocate Access Token to access Twitter service from application.



ICSBigData

[Test OAuth](#)[Details](#)[Settings](#)[Keys and Access Tokens](#)[Permissions](#)

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key) [REDACTED]

Consumer Secret (API Secret) [REDACTED]

Access Level Read and write ([modify app permissions](#))

Owner oracleics

Owner ID 843873889017090049

Application Actions

[Regenerate Consumer Key and Secret](#)[Change App Permissions](#)

Your Access Token

You haven't authorized this application for your own account yet.

By creating your access token here, you will have everything you need to make API calls right away. The access token generated will be assigned your application's current permission level.

Token Actions

[Create my access token](#)

Select **Keys and Access Tokens** tab in the page. By default, only Consumer Key & Secret are generated. In this demo, we also need to allocate Access Token to access Twitter service from application. Click **Create my access token** button in Token Actions section to generate required credentials.

Keep Consumer Key/Secret and Access Token/Secret pairs confidentially for configuring connections to Twitter with Oracle Cloud Integration Service.

The screenshot shows the Oracle Cloud Integration Service Application Management interface. The application name is 'ICSBigData'. The 'Keys and Access Tokens' tab is selected. The 'Application Settings' section displays the Consumer Key (API Key) and Consumer Secret (API Secret), both of which are redacted. It also shows the Access Level as 'Read and write (modify app permissions)', Owner as 'oracleic', and Owner ID as '843873889017090049'. The 'Application Actions' section contains buttons for 'Regenerate Consumer Key and Secret' and 'Change App Permissions'. The 'Your Access Token' section displays the Access Token and Access Token Secret, both redacted. It also shows the Access Level as 'Read and write', Owner as 'oracleic', and Owner ID as '843873889017090049'. The 'Token Actions' section contains buttons for 'Regenerate My Access Token and Token Secret' and 'Revoke Token Access'.

Step 2: Configure Connections in Oracle Integration Cloud Service

Oracle Integration Cloud Service can leverage built-in adapters to connect various kinds of external services. In this step, we will configure connections to Twitter and Oracle Storage Cloud Services.

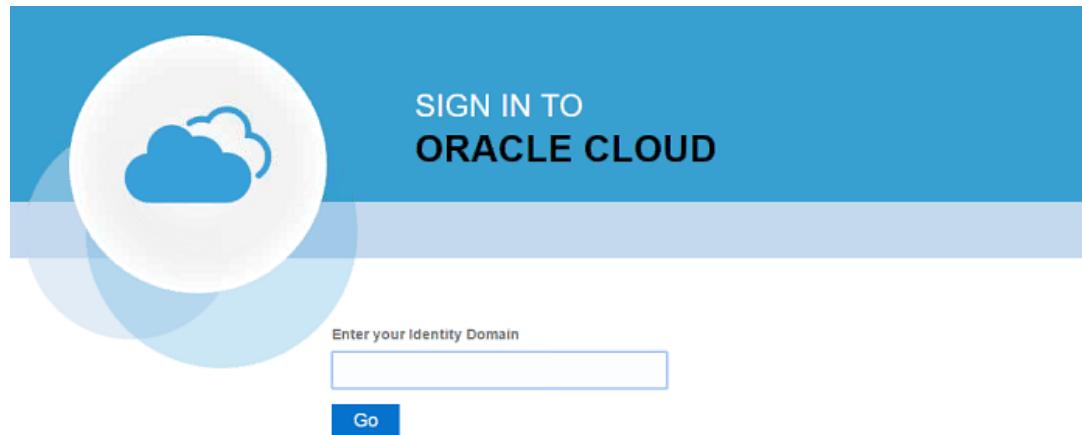
You need to first login to Oracle Cloud to configure Oracle Integration Cloud Service. Open a browser, and then enter the URL of Oracle Cloud. Click **Sign In** link on the top to access following page:

The screenshot shows the Oracle Cloud homepage. At the top, there's a navigation bar with links for Contact, Chat, English, Free Trial, Applications, Platform, Infrastructure, Resources, and a search bar. Below the navigation is a "Cloud Account" section with a user icon, a dropdown for "Traditional Cloud Account", and a "My Services" button. It also includes a "Select Data Center..." dropdown and sections for "Administrators" and "Users" with their respective permissions. To the right is a "Trouble Signing In?" section with an envelope icon, asking if users need help with account details, and a "Account Details" button. Below this is an "Order Management" section with its own user icon, a dropdown for "My Account", and a list of tasks: Complete orders and initiate cloud service provisioning, Activate paid services and trial requests, and Monitor cloud services across data centers.

The footer of the Oracle Cloud page includes a "Integrated Cloud Applications & Platform Services" link, followed by links for About Oracle, Contact Us, Legal Notices, Terms of Use, Your Privacy Rights, and Cookie Preferences. On the right side, there are social media icons for Facebook, LinkedIn, Twitter, YouTube, and others.

Oracle Cloud will leverage data center infrastructure in different spots across the world. After registering to Oracle Cloud, customers can request services from the data center near their location to get better performance.

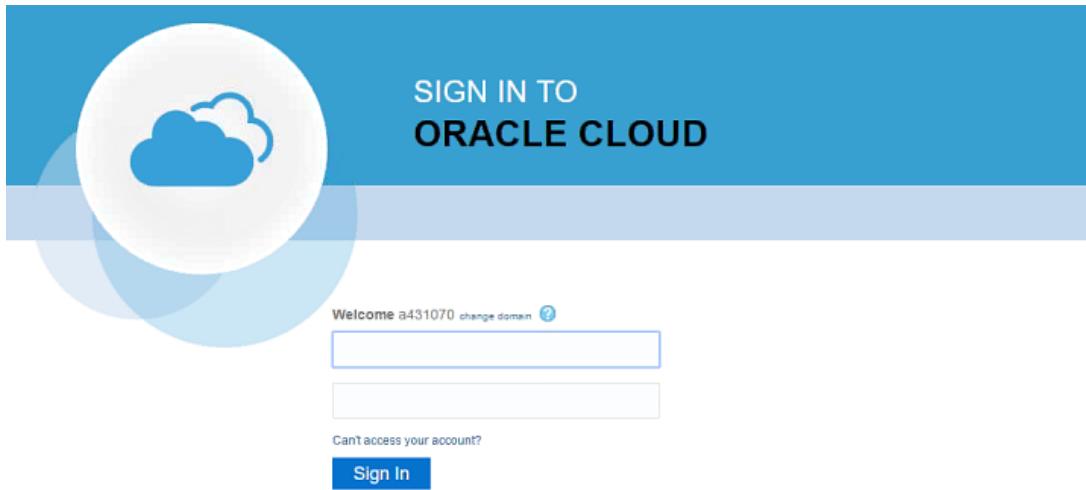
When login to Oracle Cloud, customers need to first select the data center where their services have been deployed.



ORACLE

Copyright © 2013, 2017, Oracle and/or its affiliates. All rights reserved.

Identity Domain is the container where accounts and Cloud Services are provisioned. Customers must select the correct Identity Domain to login and access their services. In this demo, user should select the Identity Domain where Oracle Integration Service is provisioned.



ORACLE®

Copyright © 2013, 2017. Oracle and/or its affiliates. All rights reserved.

Then, credential information need to input to login to Oracle Cloud platform. After login to Oracle Cloud, user can view and access cloud services from following page.

The image shows the Oracle Cloud My Services dashboard. The top navigation bar includes "ORACLE® CLOUD My Services", user info (a431070), and links for Preferences, Dashboard, Users, and Notifications. The main area is titled "Dashboard" and shows a "CLOUD SERVICES" section for the "Integration" catalog. It displays a chart for "# of Connections - ICS" with values 30, 1, 3, 5, and 0.20. Below the chart are sections for "ICS for SaaS (Instance per Month)" and "ICS for SaaS Number of Additional 10GB Data...". A "Create Instance" button is located in the top right of the dashboard area.

Select **Integration** catalog in this page to jump to following Oracle Integration Service page. From the **Overview** page, user can view current instances of Oracle Integration Cloud Service. Integrations are configured and deployed in separate service instances.

Service Details: Oracle Integration Cloud Service

Additional Information

- Service Start Date: 26-Jan-2017
- Identity Domain Name: a431070
- Subscription ID: 560039915
- Identity Domain Id: a431070
- Service Instance ID: 560039910
- Status: Active
- Customer Account: Global Prod Pool (US)
- Domain SFTP Host & Port: sftp.us2.cloud.oracle.com:22
- CS Number: Not available
- Domain SFTP User Name: a4310701
- Data Jurisdiction: US Commercial 2

Service Instances

Service Instance	Create Service Instance	Show: Active
icspmtestb		
icspmtesta		

Click **Open Service Console** link of the service instance where connections and integrations will be configured. Integration Service Console will display as below:

Welcome [REDACTED]!

Start Here

- Create Connections**
- Create and Activate your Integrations**
- Monitor Integrations on the Dashboard**

Connections
Define connections to the cloud and on-premize applications you want to integrate.
[Learn More](#) [Show Me](#)

Integrations
Connect two cloud applications, and define how they interact with each other.
[Learn More](#) [Show Me](#)

Dashboard
View the current state of your running integrations and fix any errors that occur.
[Learn More](#) [Show Me](#)

2.1 Configure connection to Twitter

Click **Create Connections** in the Console to load the Connection Designer page:

The screenshot shows the 'Connections' page in the Oracle Integration Cloud Service. The left sidebar includes links for Integrations, Connections, Agents, Lookups, Packages, and Adapters. The main area displays a table of connections:

Name	Type	Status
FlashBackFriday	TWITTER	Invoke
StorageCloud	REST	Trigger and Invoke
ERP Cloud 2	ORACLE ERP Cloud	Trigger and Invoke
ERP Cloud	ORACLE ERP Cloud	Invoke
ERP_Orchestration_Trigger	SOAP	Trigger
ERP Cloud - fap0122	ORACLE ERP Cloud	Trigger and Invoke

Click **New Connection** action in the page, following Wizard will appear to let you first select from pre-built adapters.

The screenshot shows the 'Create Connection - Select Adapter' dialog. It has a search bar and a 'Select' button. The adapters listed are:

- SRM Cloud Adapter
- Successfactors
- SurveyMonkey
- Trello
- Twilio
- Twitter

A 'Cancel' button is located at the bottom right.

Scroll down to the bottom of the list you can find **Twitter** adapter, click **Select** button to confirm your selection. A Twitter connection configuration page will appear, first following dialog will pop up to input general information of the connection.

New Connection - Information

Enter information that describes the connection. Use a meaningful name and description to help others find your connection when they create their own integrations. The Identifier must be unique and can be set only when the connection is created.

* Name	Enter a name for the connection
* Identifier	ENTER CONNECTION IDENTIFIER
Role	Invoke
Description	Enter a brief description...

Create **Cancel**

Specify a name of the connection, and select the **Role** to **Invoke** to allow request services to Twitter. Click **Create** button to create the connection.

After the connection is successfully created, you can see a light-green banner on top of the page. Next you need to configure parameters of the connection.

The screenshot shows the Oracle Integration Cloud Service interface for managing connections. At the top, a green banner indicates that the connection 'TwitterFlashBack' was created successfully. Below the banner, the connection details are listed: Identifier: TWITTERFLASHBACK, Adapter: Twitter, Connection Role: Invoke. There is a field for Description which is currently empty. In the bottom section, there is a 'Connection Administrator' section where users can enter an email address to receive notifications. Below this, there is a 'Security' section with a 'Configure Security' button. Under the security section, there are four fields for API keys: Consumer Key (API Key), Consumer Secret (API Secret), Access Token, and Access Secret, each represented by a lock icon.

Click **Configure Security** button in the page, following dialog will appear to let you input credential information to establish the Twitter connection.

Credentials

You can configure the Security Policy for this connection. Please select the Security Policy.

Security Policy: Custom Security Policy

Your application/endpoint requires that users and services provide security credentials for access. Specify the login credentials below.

Property Name	Property Value
* Consumer Key (API Key)	Consumer Key (API Key)
* Consumer Secret (API Secret)	Consumer Secret (API Secret)
* Access Token	Access Token
* Access Secret	Access Secret

OK Cancel

All credential information used in this dialog is from previous step. You can copy those parameters from Twitter Application Management page directly. After fill all required fields, click **OK** button.

Now all required parameters have been configured for the Twitter connection. However it is not sure that the connection will work. Click **Test** action on top of the page to trigger a test. If the test passed, a message will display in the light-green banner. Also the progress icon on the right will indicate that the connection is 100 percent correct.

After test is done, you can save the connection by clicking **Save** action. Then click **Exit Connection** action to go back to Connection Designer page.

The screenshot shows the Oracle Integration Cloud Service Connection Designer interface. At the top, there is a green banner indicating "Connection 'TwitterFlashBack' was created successfully". Below the banner, there are buttons for "Exit Connection" and "Editing". On the right side, there are buttons for "Save", "Delete", "Configure Security", and "Test". The main area displays the configuration details for the connection:

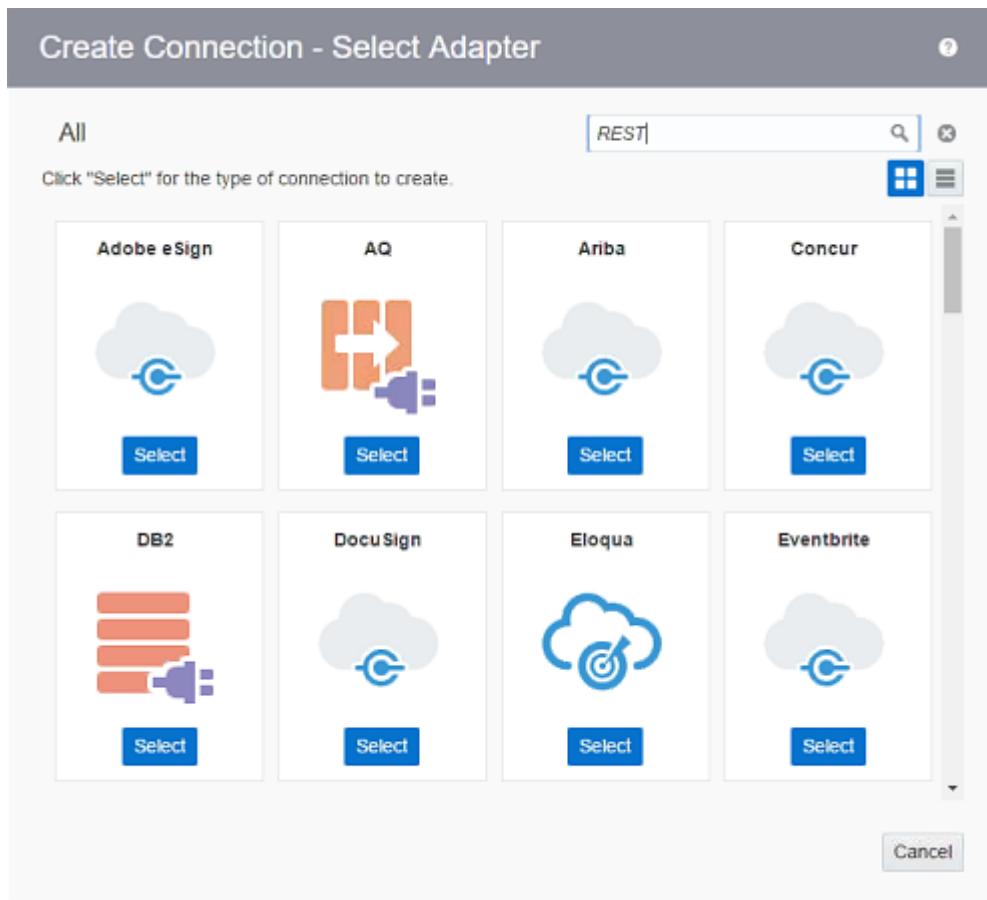
- Identifier:** TWITTERFLASHBACK
- Adapter:** Twitter
- Connection Role:** Invoke
- Description:** (empty)

Below the configuration details, there is a section for "Connection Administrator" with a note: "You can receive email notifications when problems or changes occur in this connection. Enter the email address to receive these notifications." There is a field for "Email Address" and a "Configure Security" button. At the bottom, there is a "Security" section with a lock icon and a "Configure Security" button.

2.2 Configure connection to Oracle Storage Cloud Service

In the Connection Designer page, click **New Connection** action to launch the Adapter Selection dialog. Oracle Integration Service does not provide built-in adapter for Oracle Storage Cloud Service, however user can leverage generic REST adapter to access REST API endpoint of the Oracle Storage Cloud.

In the search box of the dialog, input "REST" then click the search icon, the REST adapter will be the only available adapter in the list.



Click **Select** action in the REST adapter, the connection design page will show again to let you configure the connection to Oracle Storage Cloud Service.

First, following dialog will appear to collect general information of the connection.

Name	Enter a name for the connection
Identifier	ENTER CONNECTION IDENTIFIER
Role	Trigger and Invoke
Description	Enter a brief description...
<input type="button" value="Create"/> <input type="button" value="Cancel"/>	

Specify the name of the connection and select **Role** to **Trigger and Invoke** which means that the connection can both receive notification from Oracle Storage Cloud when Object Store is changed and make requests to the Oracle Storage Service. Click **Create** button to create the connection.

Next, we need to input the connection parameters to REST API endpoint of the Oracle Storage Cloud. Click **Configure Connectivity** button in the page to launch following dialog.

Connection Properties

Enter information so we can connect to your application/endpoint and process requests.

Property Name	Property Value
* Connection Type	< Please select an item from the list >
TLS Version	< Please select an item from the list >
* Connection URL	Please make sure that this value really corresponds to the type selected above.

OK **Cancel**

Set **Connection Type** to “REST API Base URL”, select **TLS Version** to “TLS V1.1” and copy the base url (without URI part) of the REST API endpoint of the Storage Cloud to the **Connection URL** field. Click **OK** to close the dialog.

The last parameters need to configure is the security information to login to REST API endpoint of the Storage Cloud. Click **Configure Security** button in the page to launch following dialog.

All the information in this dialog can be known from the Oracle Storage Cloud configuration. Format of the **Username** filed is “*Storage-<Identity Domain>:<login name>*”.

Credentials

You can configure the Security Policy for this connection. Please select the Security Policy.

Security Policy **Basic Authentication**

Your application/endpoint requires that users and services provide security credentials for access. Specify the login credentials below.

Property Name	Property Value
* Username	
* Password	
* Confirm Password	

OK **Cancel**

After specifying all required information, click **OK** to close the dialog.

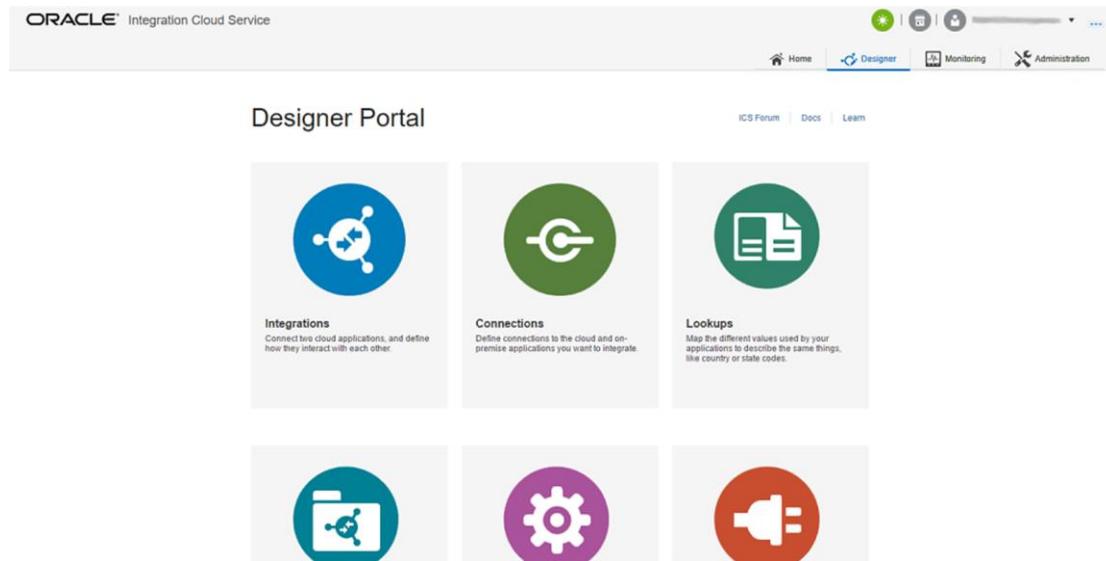
Now all required parameters have been configured for the Oracle Storage Cloud connection. A test is also required to ensure the configuration is correct. After test is done, click **Save** action to save the connection, and then click **Exit Connection** action to return to Connection Design page.

Step 3: Design the Integration

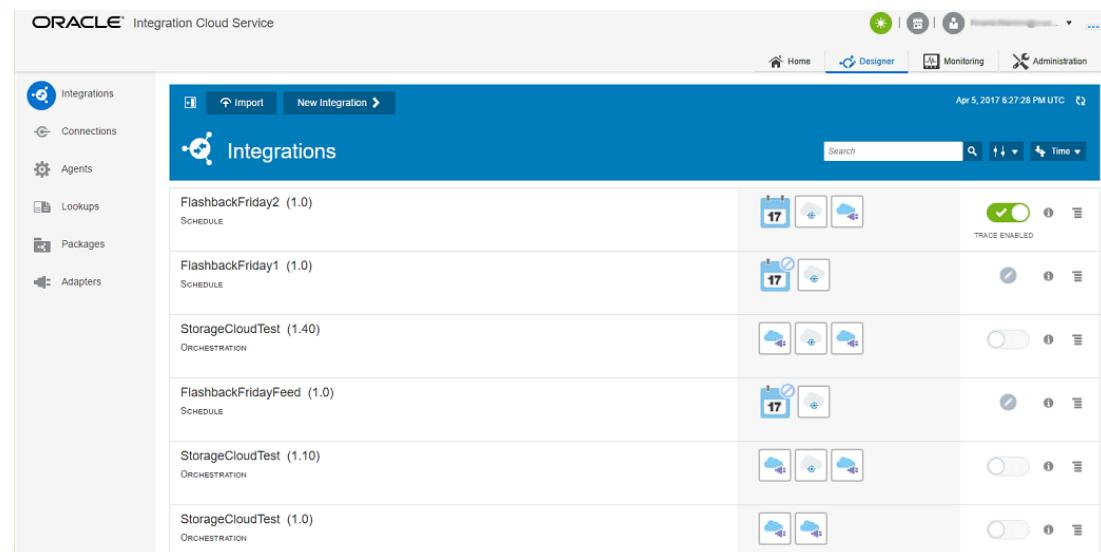
In this step, we will demonstrate how to design integrations with Oracle Integration Cloud Service to load Twitter feeds that the user is interested in and pull the data into Data Lake in Oracle Storage Cloud.

Oracle Integration Cloud Service provides an interactive tool for designing integrations with zero coding. The tool is available from the Designer Portal in Oracle Integration Cloud Service

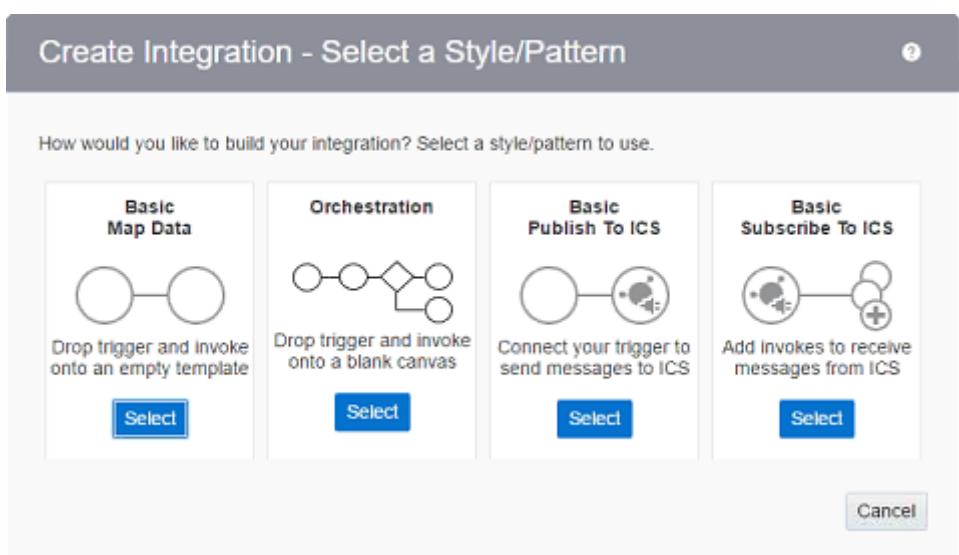
page. To go to Designer Portal, select **Designer** link on top of the Oracle Integration Cloud Service page.



From Designer Portal user can go directly to various designer tools, for example Integration Designer, Connection Designer etc., in Oracle Integration Cloud Service. Click **Integrations** icon to launch following Integration Designer.



User can manage and monitor integrations in the designer page. To create a new integration, click **New Integration** button on the top. Oracle Integration Cloud Service has pre-defined styles or patterns to design the integration.



In this demo, the integration is complex. It needs to handle connections to both Twitter and Oracle Storage Cloud carefully. User also needs to control data ingestion, transform and output at fine-grained level.

Select the **Orchestration** pattern for the integration, following dialog will pop up to allow user provide general information of the integration.

Name, version and an optional package are used to uniquely identify the integration. An integration could be triggered either by an event from a service connection or a schedule defined by Oracle Integration platform.

The dialog box is titled "Create New Integration". It has sections for "What triggers this integration?", "Describe this integration", and input fields for Identifier, Version, and Package.

What triggers this integration?

- Application event or business object
- Schedule

Describe this integration Use a meaningful name and description that will help others find and understand this integration. The Identifier and Version can be set only when the integration is created. The combination of Identifier and Version must be unique.

*** What do you want to call your integration?**
For example: Opportunity to Order

*** Identifier**
ENTER INTEGRATION IDENTIFIER

*** Version**
01.00.0000

What does this integration do?
Describe the integration's purpose and detail

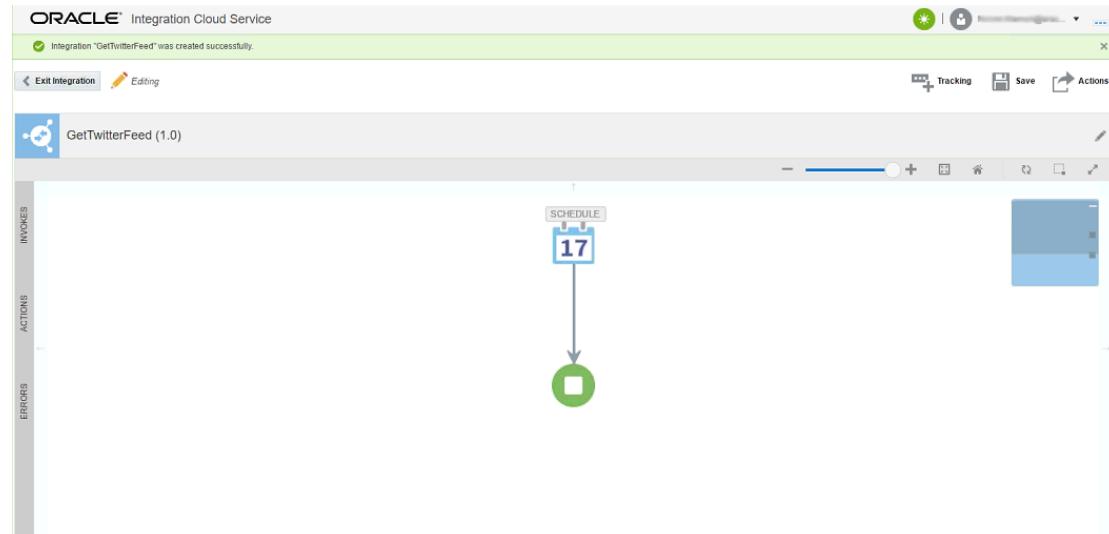
Which package does this integration belong to?
Enter a new or existing package name

Create **Cancel**

Select **Schedule** as trigger for the integration, and specify name, version and package for the

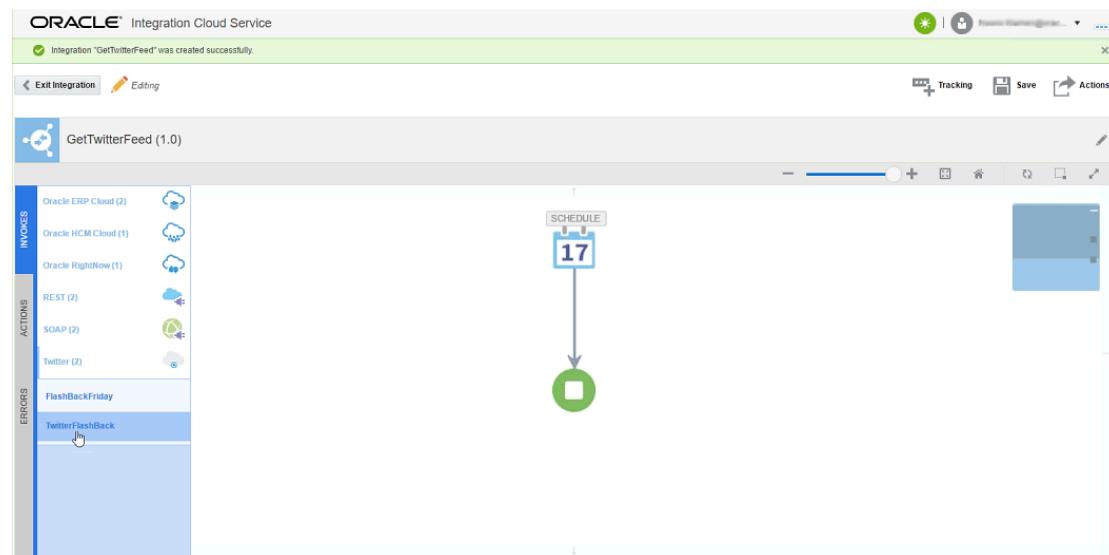
integration, then click **Create** button.

Following Designer page will appear to give user an interactive interface to design logic of the integration. On the left of the page are popup toolboxes. User can select to **Invoke** an available services, add an **Action** and handle **Error** in the integration. In the middle of the page are panel to view and edit the integration interactively. If the integration logic is very complex, user can also zoom and go to different part of the view with UI controls in the page.

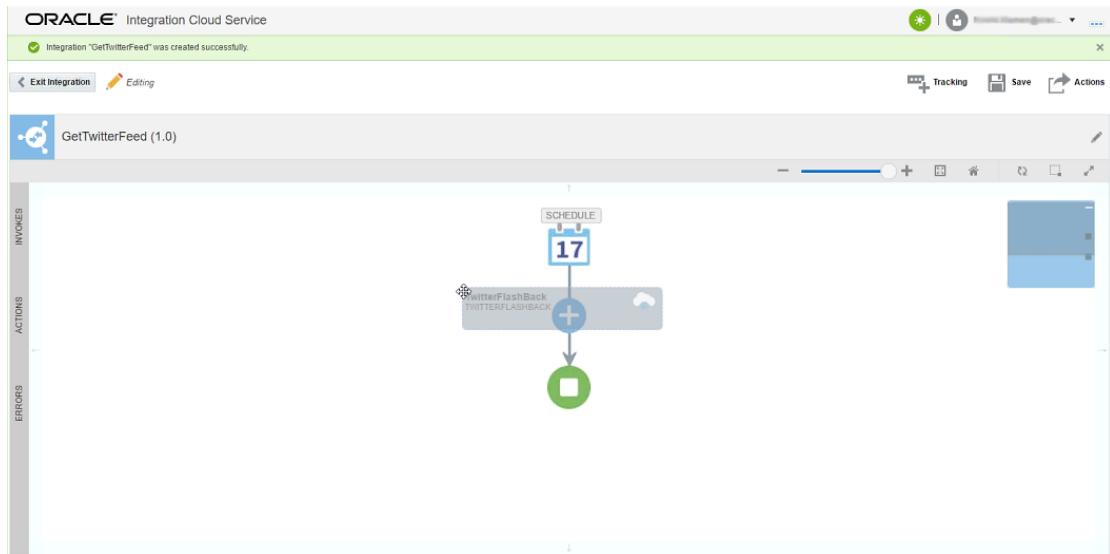


At initial, only a start and end node in the integration design page. User needs to add more logic to the integration.

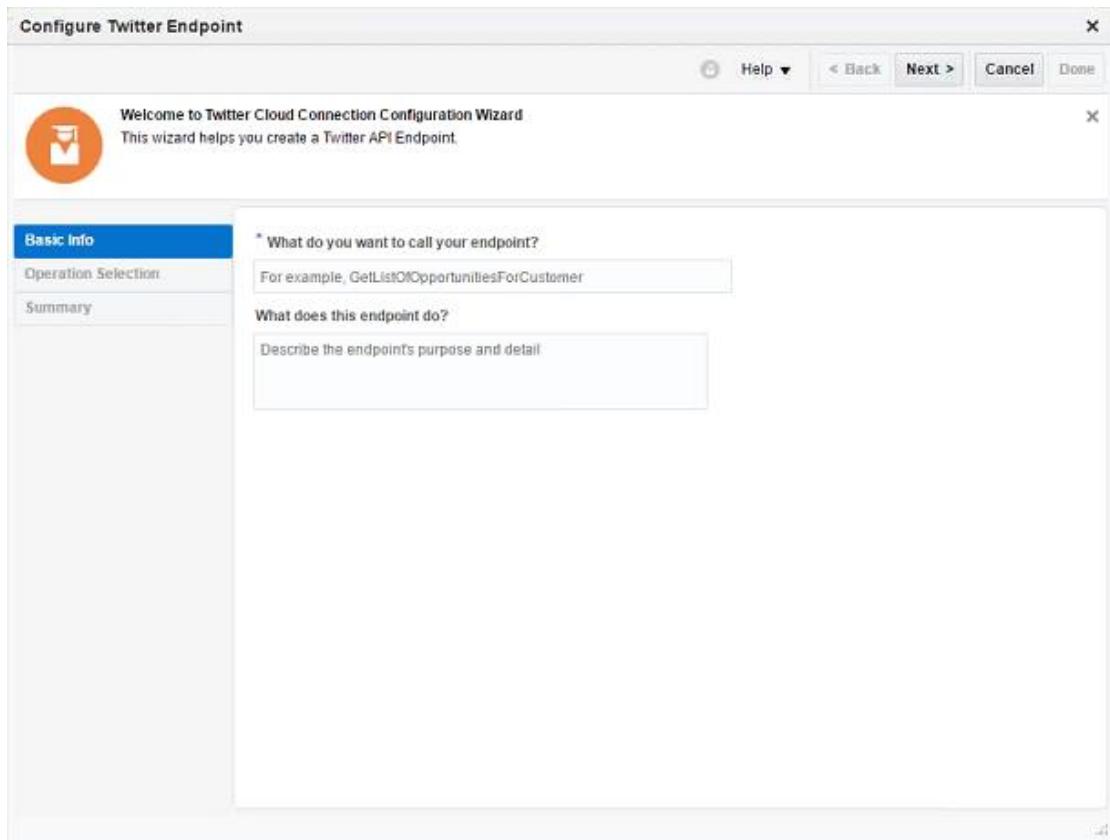
In the demo, we will first invoke Twitter to retrieve any new feeds. Move the mouse to INVOKES tab on the left, it will pop up a toolbox to let user select available services. Select Twitter catalog, all Twitter service connections will show just below the catalog.



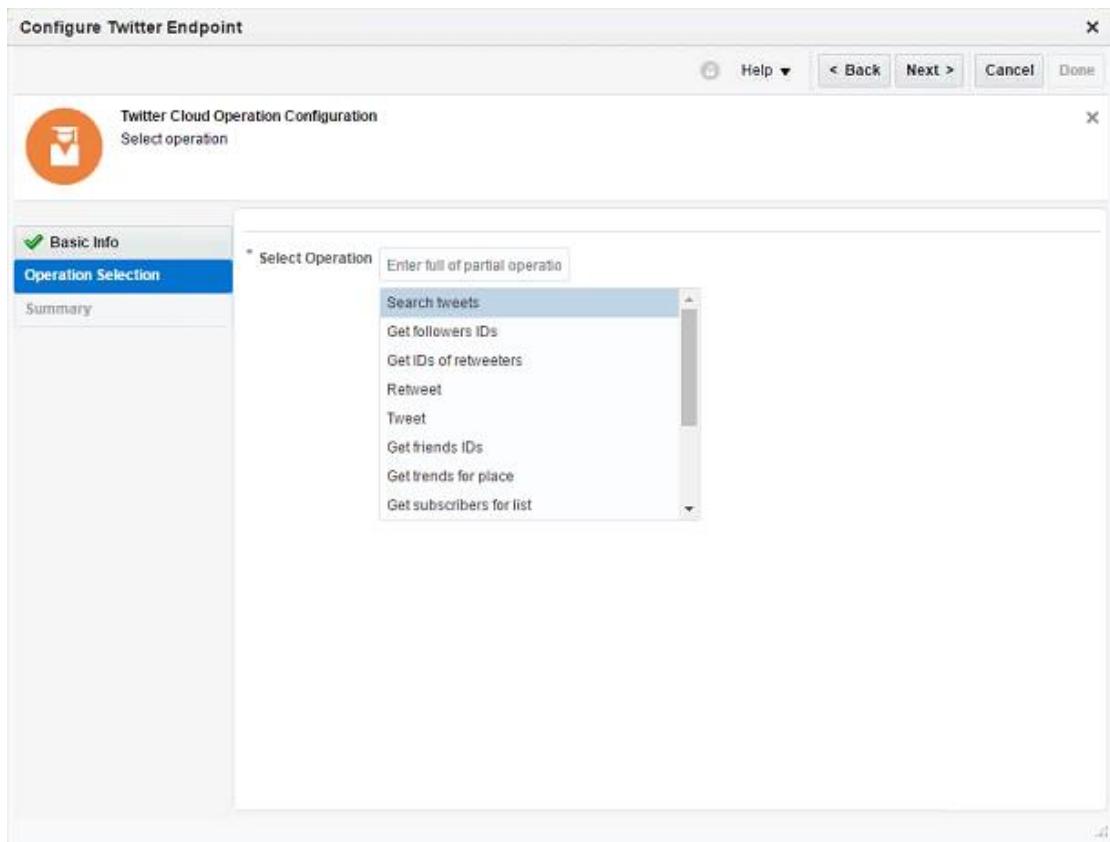
Drag the connection in between start and end node of the flow in the design panel.



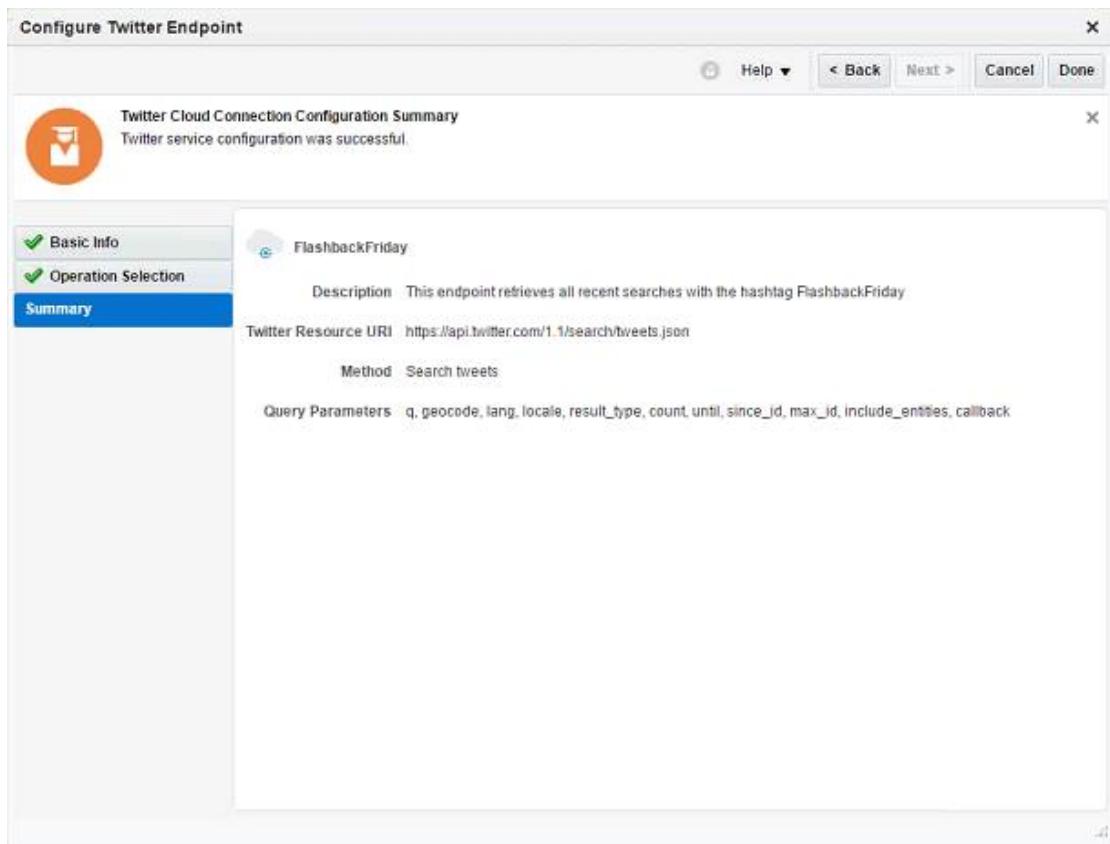
Following wizard will appear to let use customize the service invocation. The **Basic Info** tab let user assign a name of the invocation. The name will show as display name of the invocation node in the design panel.



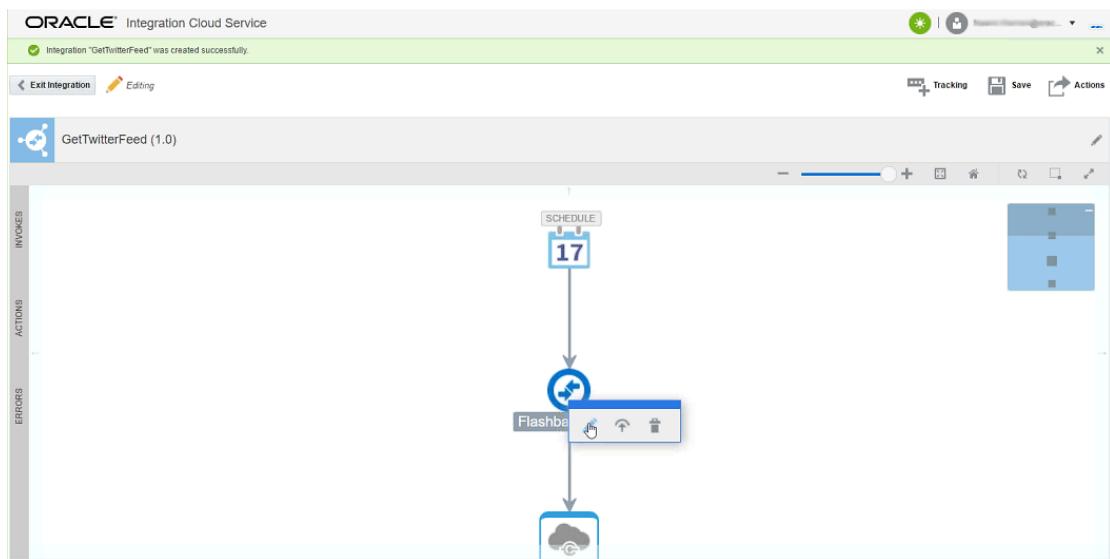
After filling the required information, click **Next** button to move to following page. In this page user can select which service function need to invoke from the integration.



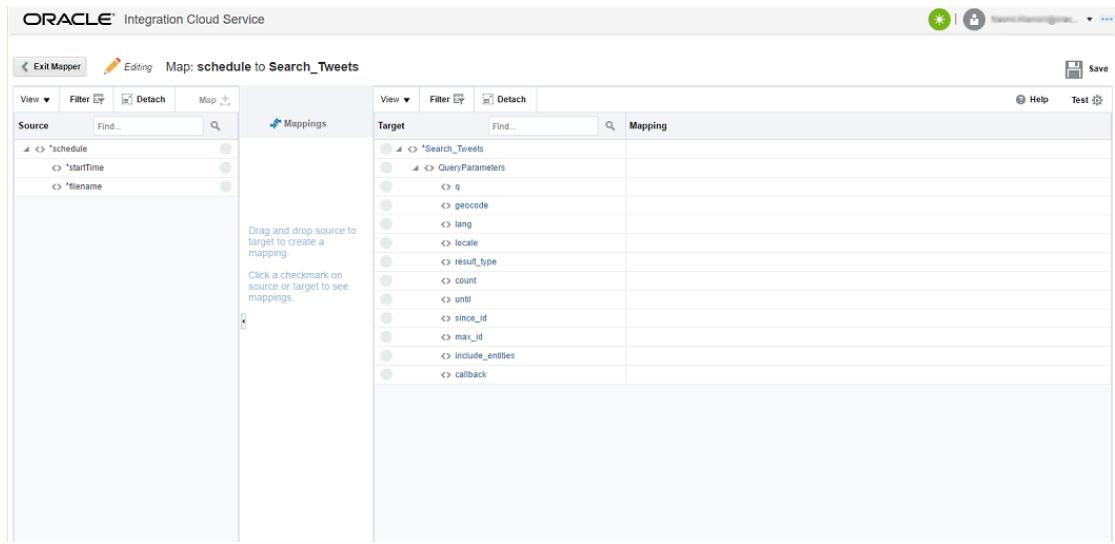
Select “Search tweets” from the function list, then click **Next** button to move to following page. This page will show overview information of the invocation. User has the opportunity to check the configuration, if anything is wrong, they can go back to previous page to make modification.



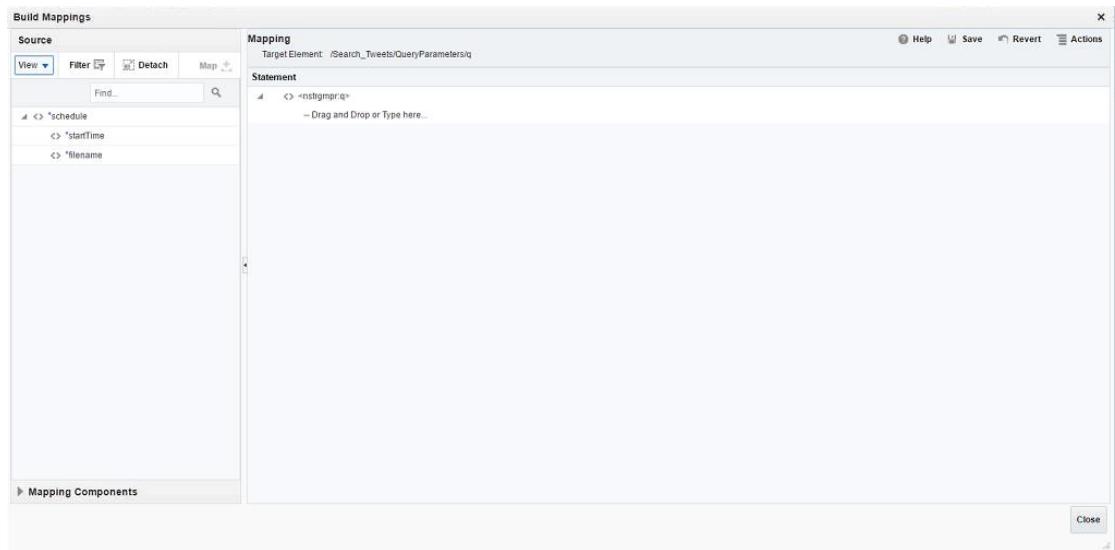
Click **Done** to accept the configuration. A new invocation node will appear in the design panel. However it is not done yet, user also needs to customize parameters of the invocation. Click the invocation node, a pop up toolbox will appear.



Click button in the toolbox to launch following dialog to map invocation parameters. Available parameters of the service call are listed on the right. User can design mapping for each individual parameter. A mapping is the way to assign runtime value of the target of existing sources. The source could be a static value, data from an available data source etc. Click a parameter in the list can modify the mapping.



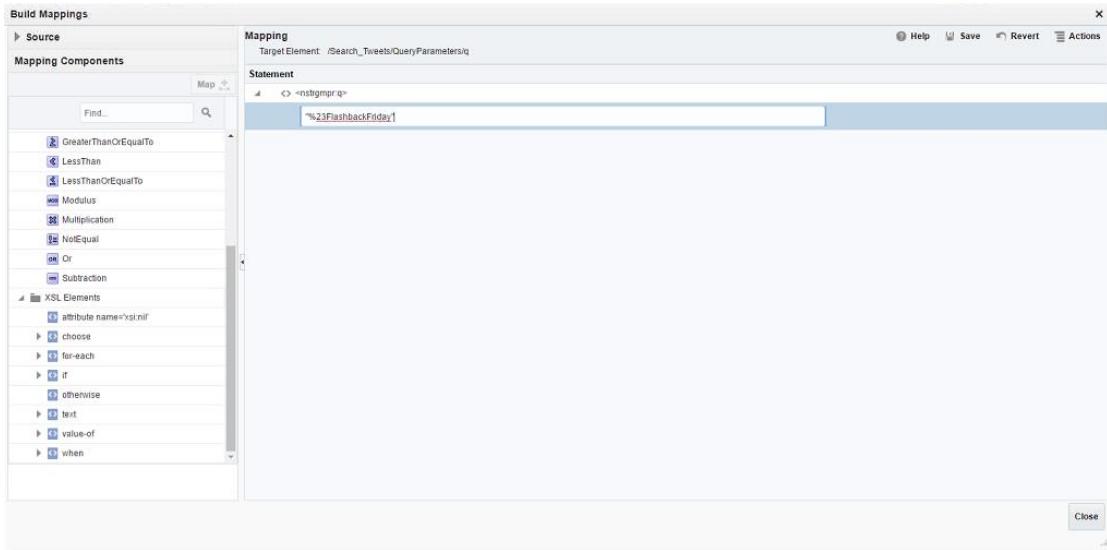
The first parameter needs to customize in the demo is **q**. Click the parameter will launch another dialog to design the mapping. User can select an entity from a source on the left, and drag it to link with the target. Also user can assign a fixed value to the target.



In this demo, the parameter will use a fixed value which indicates search tweets by a specific tag.

Click the target directly, a text box will appear to let you input the value. Input name of the interested tag in the tweets. You should encode the string according to XML standard.

Click **Save** button on the top to apply the change, then click **Close** button to return to previous dialog.

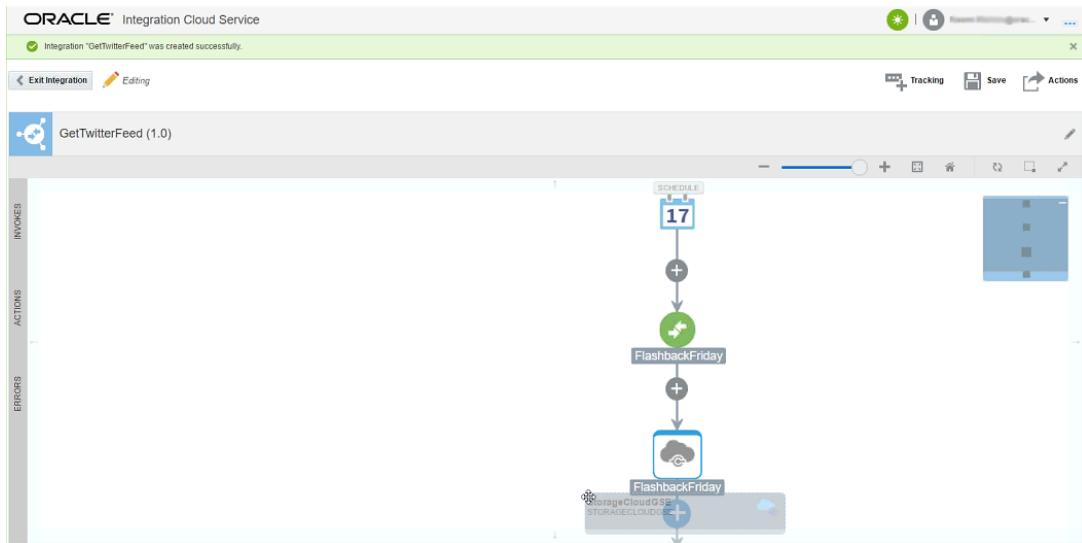


Another parameter needs to be customized is **result_type**. Use a fixed value “recent” for this parameter to only retrieve new tweets from user’s Twitter account.

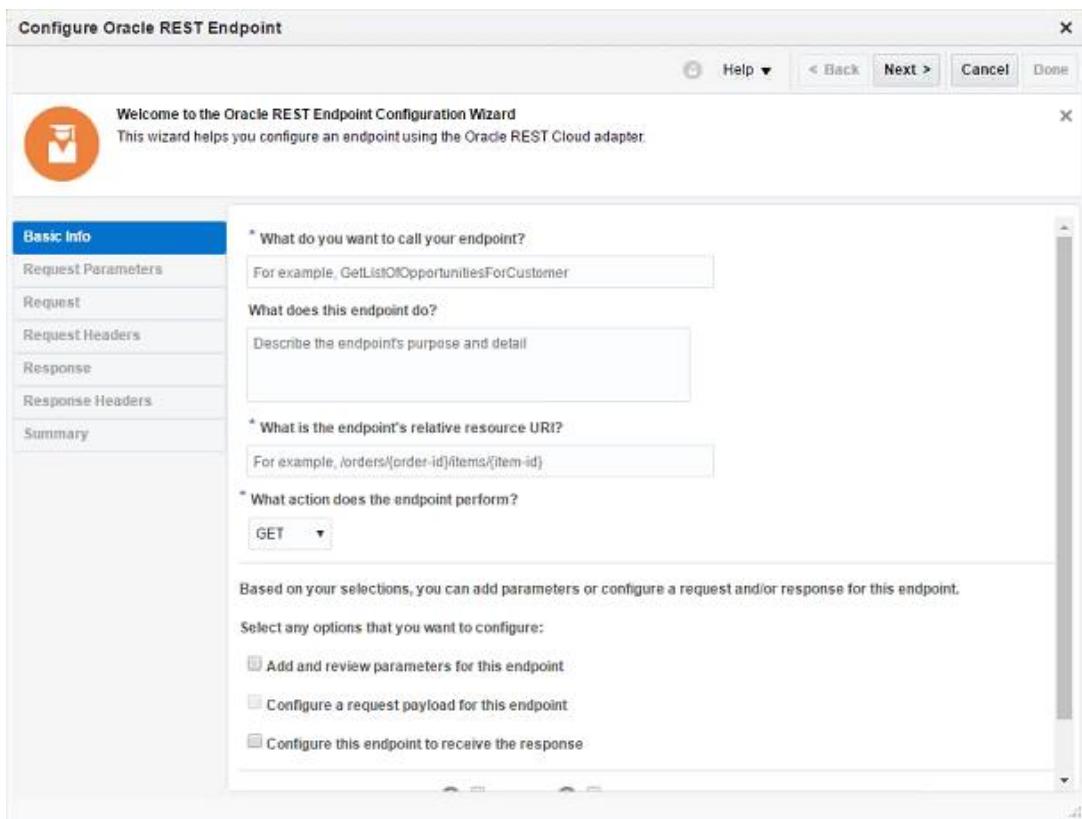
After both parameters are customized, use can view the configuration in the dialog. If properly configured, click **Exit Mapper** button to save the change.

Now the integration can invoke Twitter to retrieve interested tweets from user’s account. Next step is to enable writing those tweets to Data Lake in Oracle Storage Cloud. Services from the Oracle Storage Cloud can be leveraged implement the function.

The integration will first authenticate to the Oracle Storage Cloud to retrieve a security token for following requests. Select pre-defined Oracle Storage Cloud connection from **INVOKES** pop up toolbox in the left. Then drag the connection to the position between the Twitter node and the end node in the flow.

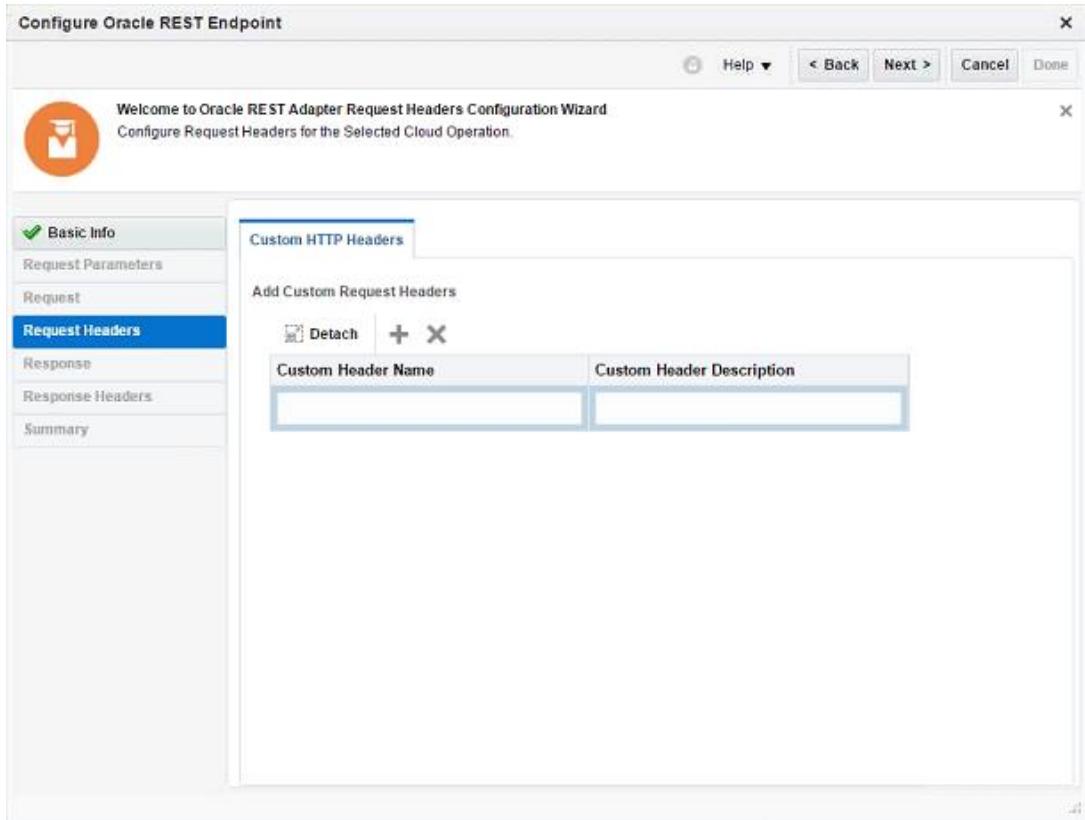


Following dialog will appear to configure the service invocation. The REST service endpoint at Oracle Storage Cloud is a HTTP service. The authentication process is wrapped in HTTP communication. Credential information is wrapped in a HTTP GET request, server returns authentication result in the HTTP response.



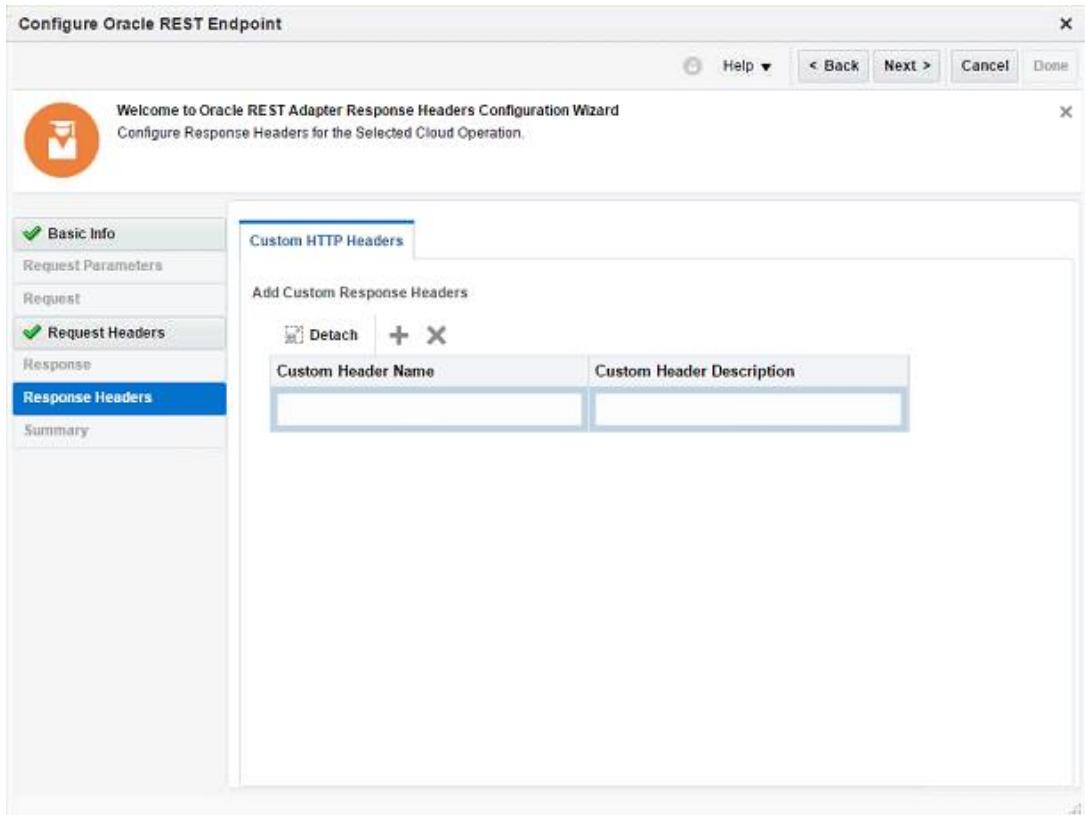
Specify the relative URI of the authentication service at the Oracle Storage Cloud, it is `/auth/v1.0` in this demo. Select to use GET method to send HTTP request. Customization need to be enabled for both preparing Request headers and analyzing Response headers.

Click **Next** button to move to following page. The page allows configure HTTP headers that are specific to the authentication process. Click **+** button to add a new header declaration. A name must be given in left column and an optional description in right column.



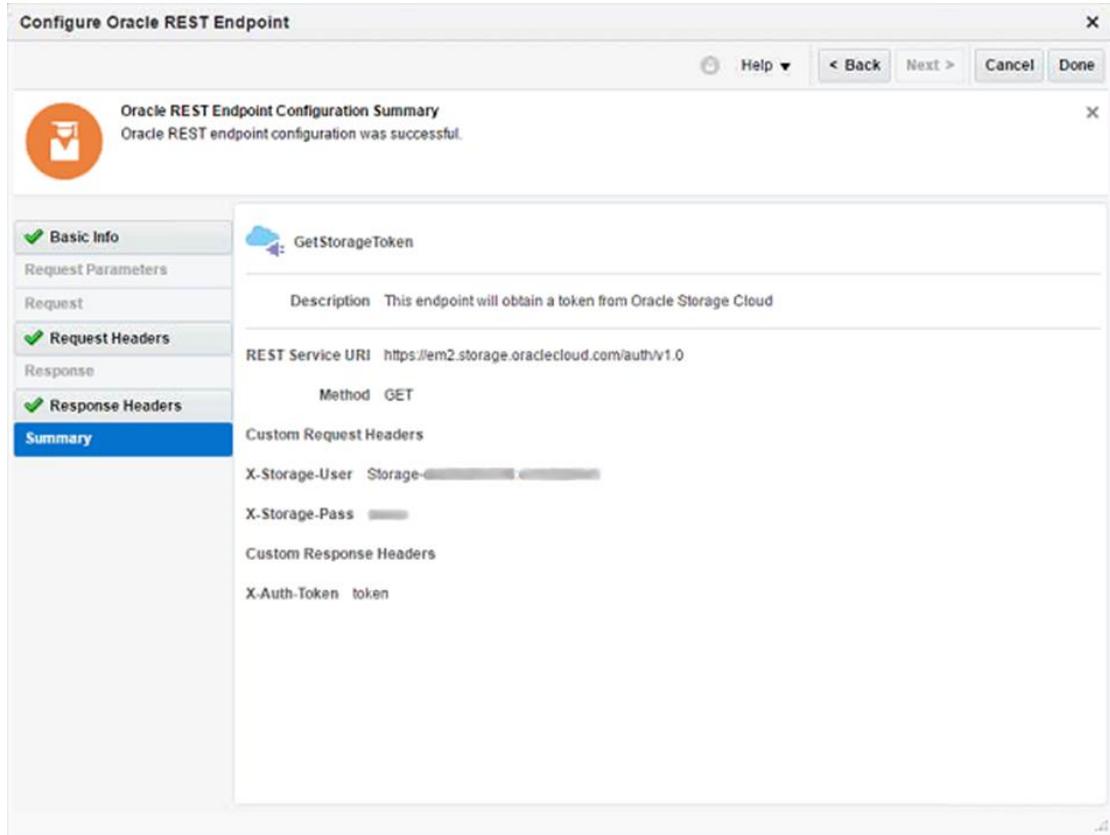
Two headers, **X-Storage-User** and **X-Storage-Pass**, are defined. They wrap username and password for authentication respectively.

After declaring both request headers, click **Next** button to move to following page. This page will declare response headers specific to the authentication process.

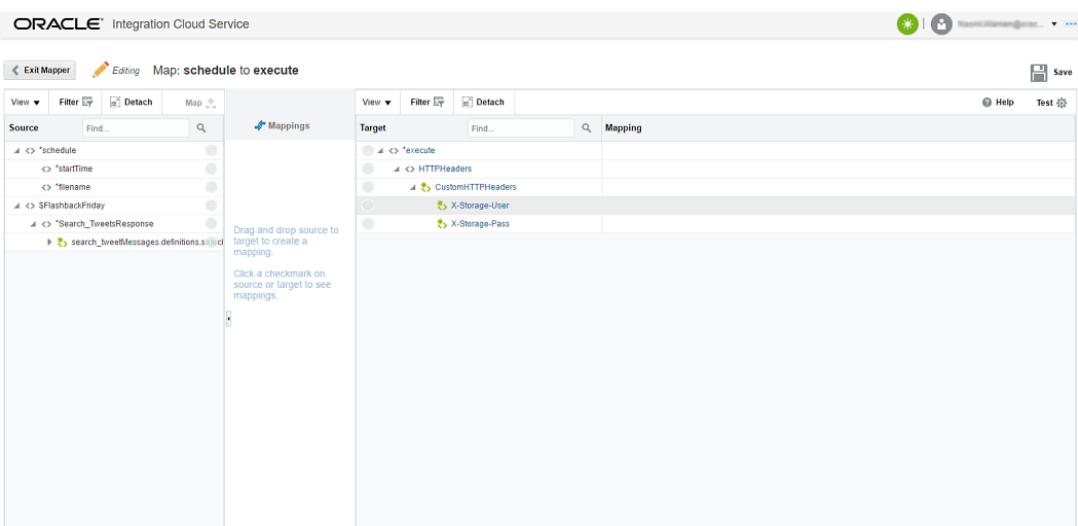


The response header **X-Auth-Token** is pre-defined to wrap the token if authentication succeeds.

After declaring the header click **Next** to move to following page. User can check the configuration in the page. If all configurations are correct, click **Done** button to commit the configuration.



Next, we will provide value for parameters (request headers) in the request. Select the invocation node and click button in the pop up toolbox to launch the Mapper dialog.



Click each HTTP header declared previously. In the demo, we will use static value for both headers. Values for both headers are from Oracle Storage Cloud configuration. Format for

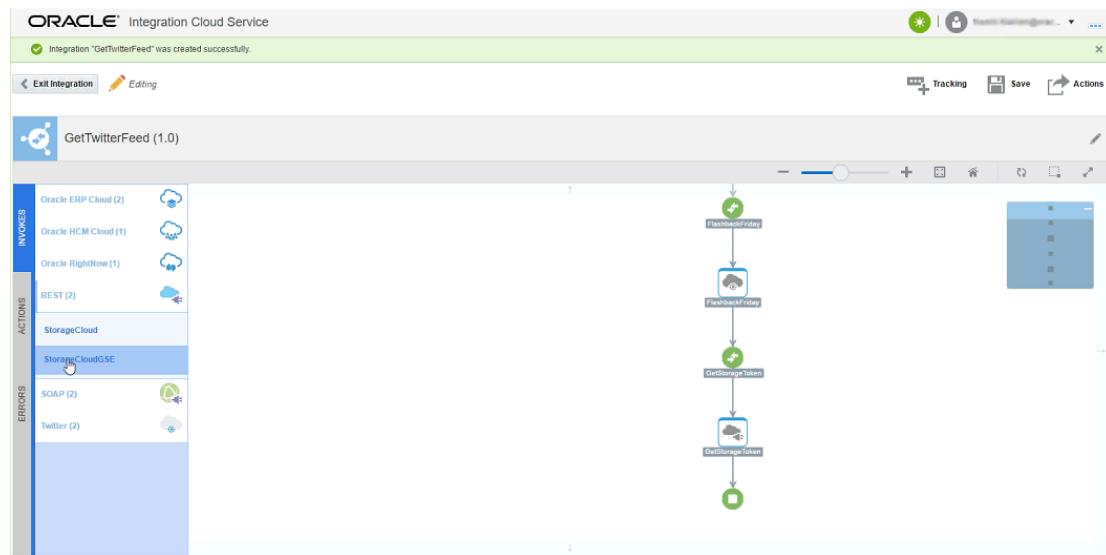
X-Storage-User is "Storage-<identity domain>:<login name>".

The screenshot shows the 'Build Mappings' interface in Oracle Integration Cloud Service. The 'Source' pane on the left lists various components: 'schedule', 'startTime', 'filename', '\$FlashbackFriday', and 'Search_TweetsResponse'. The 'Mapping' pane on the right shows a target element '/execute/HTTPHeaders/CustomHTTPHeaders/X-Storage-User'. A statement is defined under it, containing a mapping from 'nsmp2.X-Storage-User' to 'Storage-<identity domain>:<login name>'. The 'Actions' bar at the top includes Help, Save, Revert, and Actions buttons.

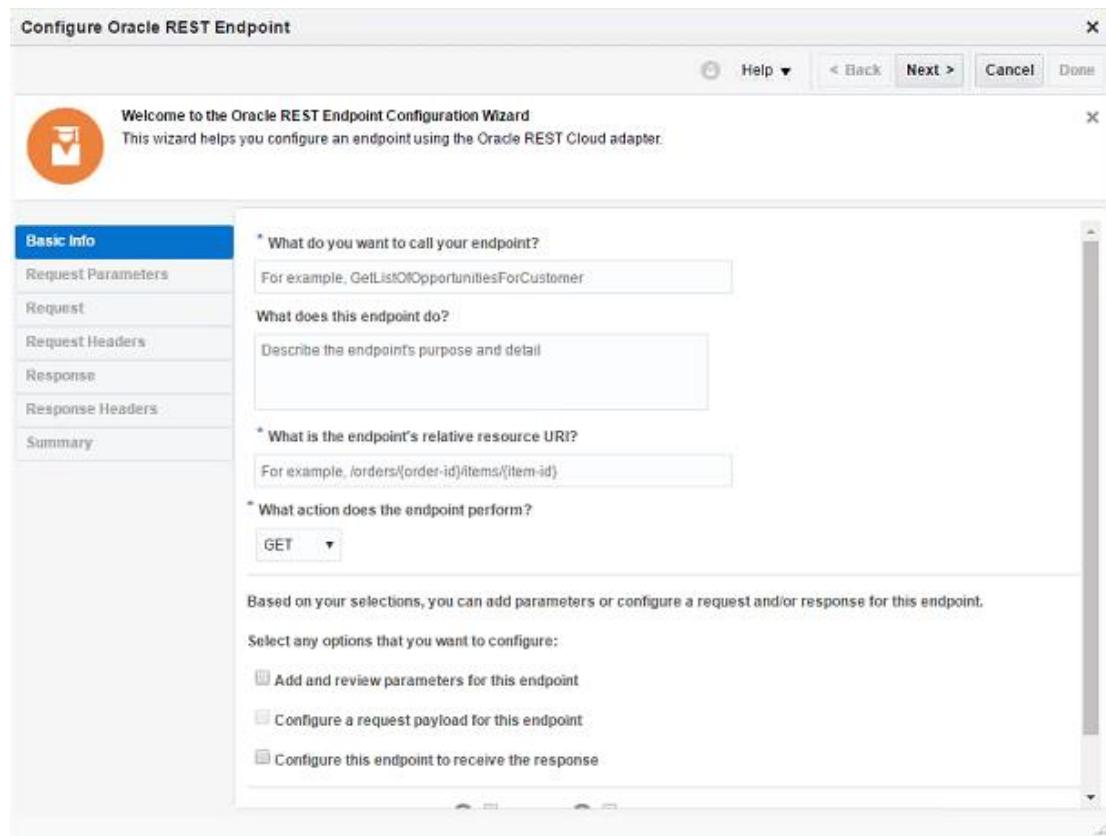
This screenshot shows another mapping configuration. The 'Source' pane lists the same components as the first one. The 'Mapping' pane shows a target element '/execute/HTTPHeaders/CustomHTTPHeaders/X-Storage-Pass'. A statement is defined under it, containing a mapping from 'nsmp2.X-Storage-Pass' to 'Storage-<identity domain>:<password>'. The 'Actions' bar at the top includes Help, Save, Revert, and Actions buttons.

This screenshot shows the 'Map: schedule to execute' interface. It displays two main panes: 'Source' on the left and 'Target' on the right. The Source pane lists the same components as before. The Target pane shows the target structure: '/execute/HTTPHeaders/CustomHTTPHeaders/X-Storage-User' and '/execute/HTTPHeaders/CustomHTTPHeaders/X-Storage-Pass'. Both of these mappings are checked and mapped to 'Storage-<identity domain>:<login name>' and 'Storage-<identity domain>:<password>' respectively. A central 'Mappings' section provides instructions for dragging and dropping components to create mappings. The 'Actions' bar at the top includes Exit Mapper, Editing, Save, Help, Test, and a three-dot menu.

After committing the values, click **Exit Mapper** button to return to Integration Designer. Next action in the integration flow is to write data to Oracle Storage Cloud.



Select the Oracle Storage Cloud connection from the INVOKES toolbox, and drag it between authentication result node and end node in the flow. Following dialog appears again to configure REST invocation.



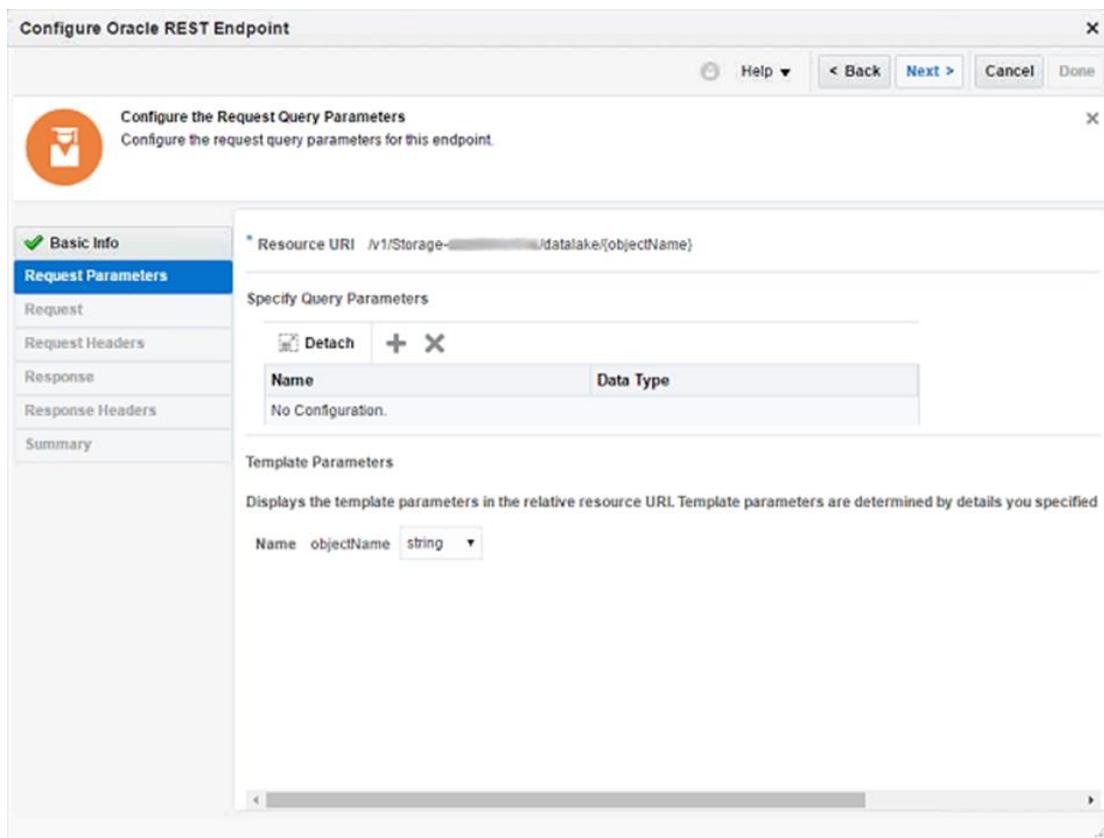
The URI for the data object upload service in Oracle Storage Cloud is in the format: `/v1/Storage-<Identity Domain>/<Container>/<Object Name>`. The service leverages HTTP PUT method to upload the data object. Content of the data object is sent as payload in a HTTP

request. **X-Auth-Token** header must be included in the HTTP request as the authentication proof to the service.

To enable the integration flow to control data object name at runtime, we use the template `/v1/Storage-<Identity Domain>/<Container>/<Object Name>` for the service URL. The container must be created in advance before the demo.

Select **PUT** as the request method, and check **Configure a request payload for this endpoint** to enable payload customization. Also need to enable request header customization in order to append **X-Auth-Token** header in the request.

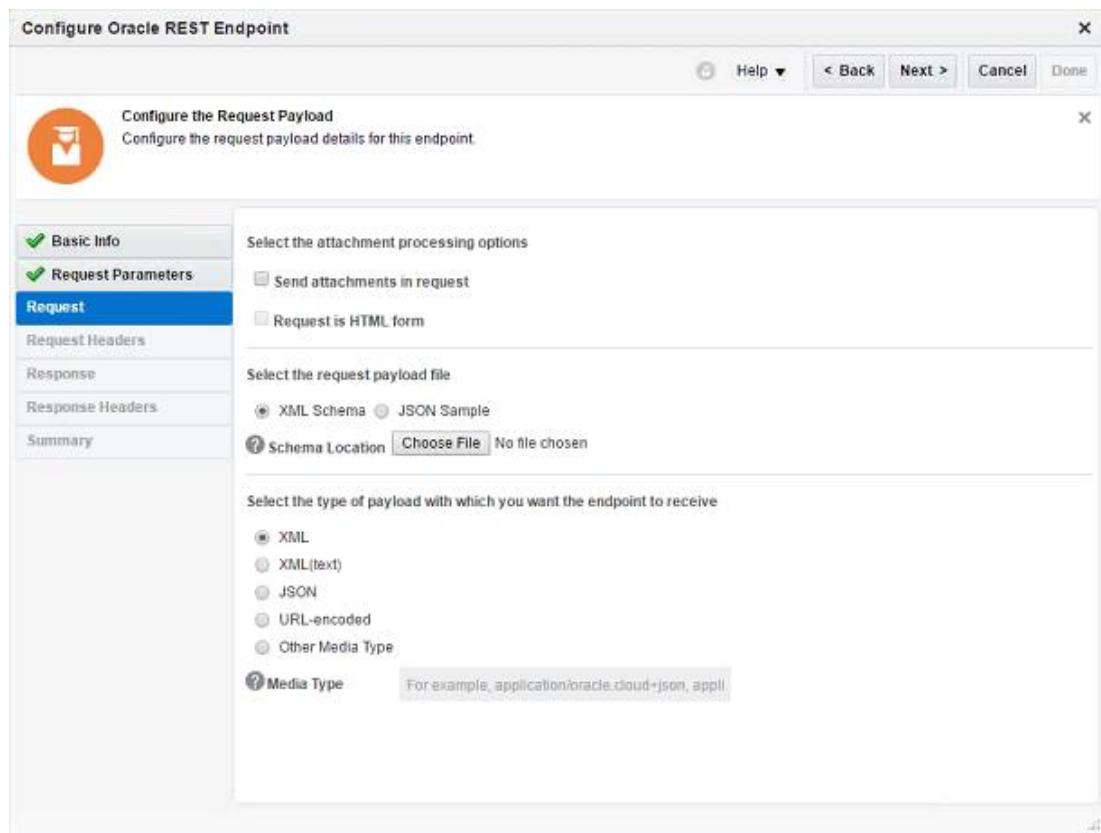
Click **Next** button to move to following page. User can add additional query parameters for the URI.



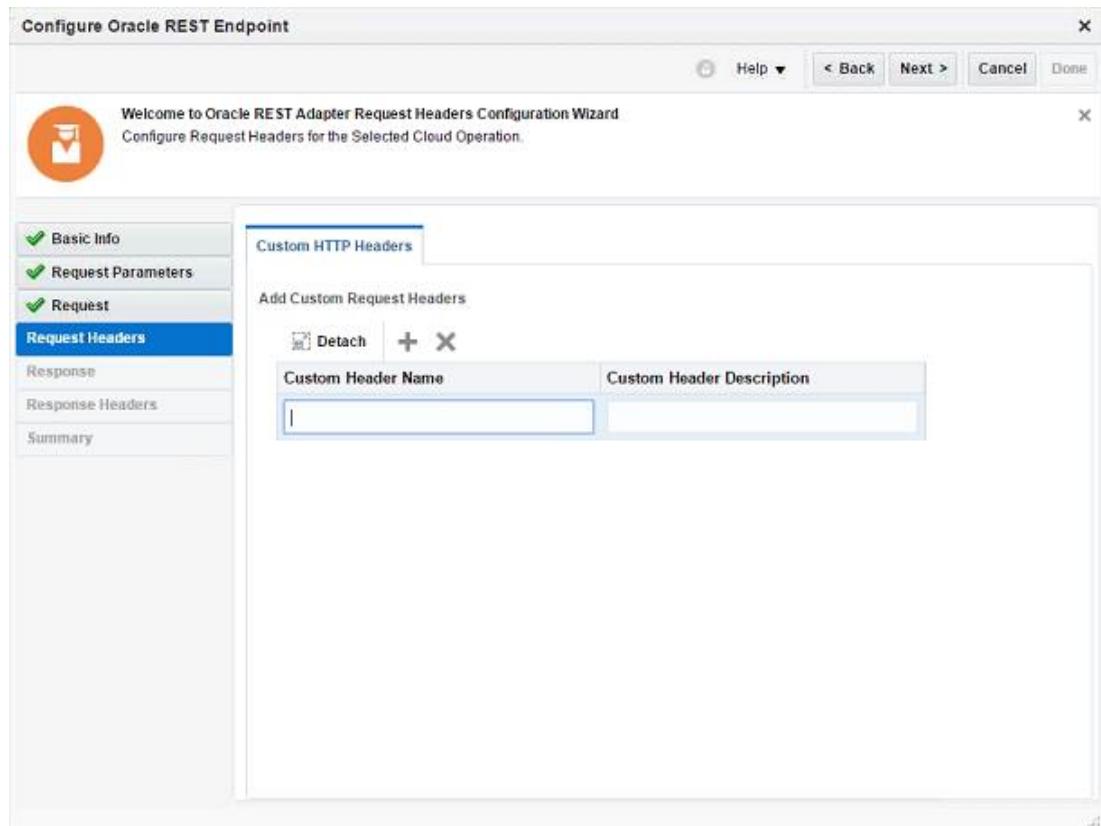
However, there is no query parameter defined for the service. So simply click **Next** button to move to the following page.

In the page, user needs to customize format of the payload. First, user need to select type of the payload. For different payload type, there are different ways to describe the format. For example, user can select an XSD schema file as format for a XML-based payload. Integration Platform will parse the format selected in the page to build data object at runtime.

In the demo, payload of the request is in XML format. A XSD schema file has been created to describe the XML document architecture.

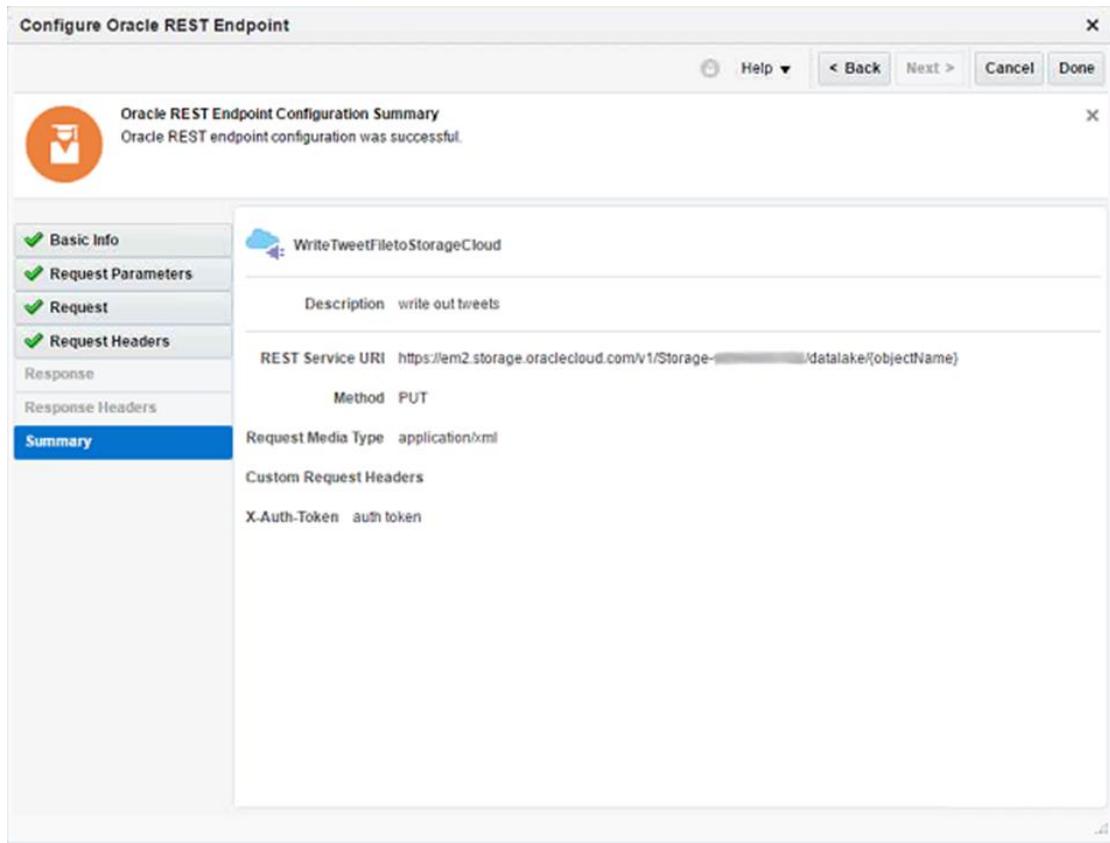


Select XML as type of payload and also upload the XSD schema file. Then click **Next** button to following page.



Add **X-Auth-Token** as a required HTTP header in the request. The exact value of the header will map the security token returned from the previous step.

Click **Next** button to the **Summary** page to check your configuration. If everything looks OK, click **Done** button to commit the invocation.

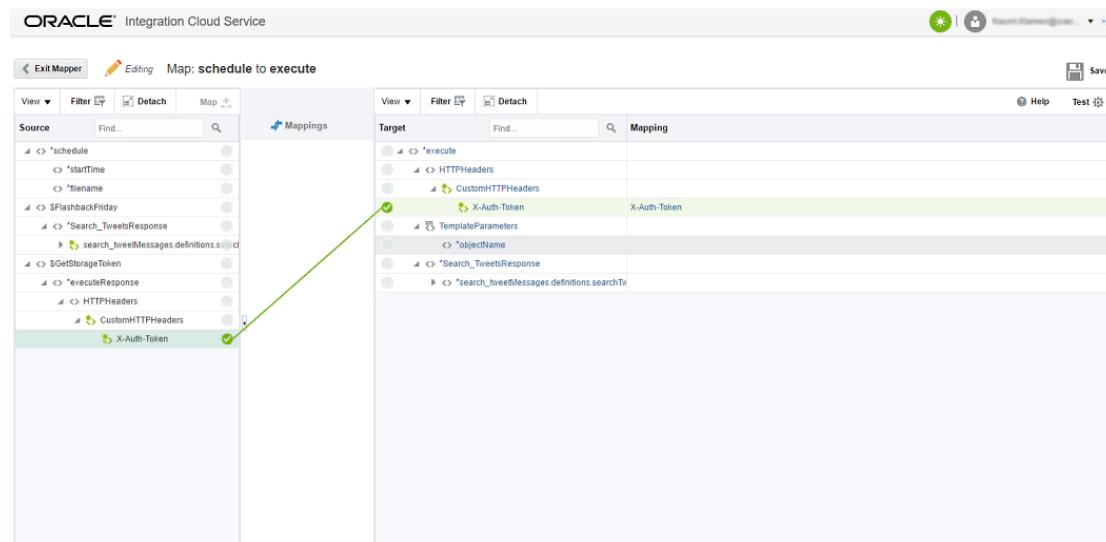


Next, we will customize mapping for parameters in the invocation. Click invocation node in the flow and click button in the pop up toolbox to launch following Mapper dialog.

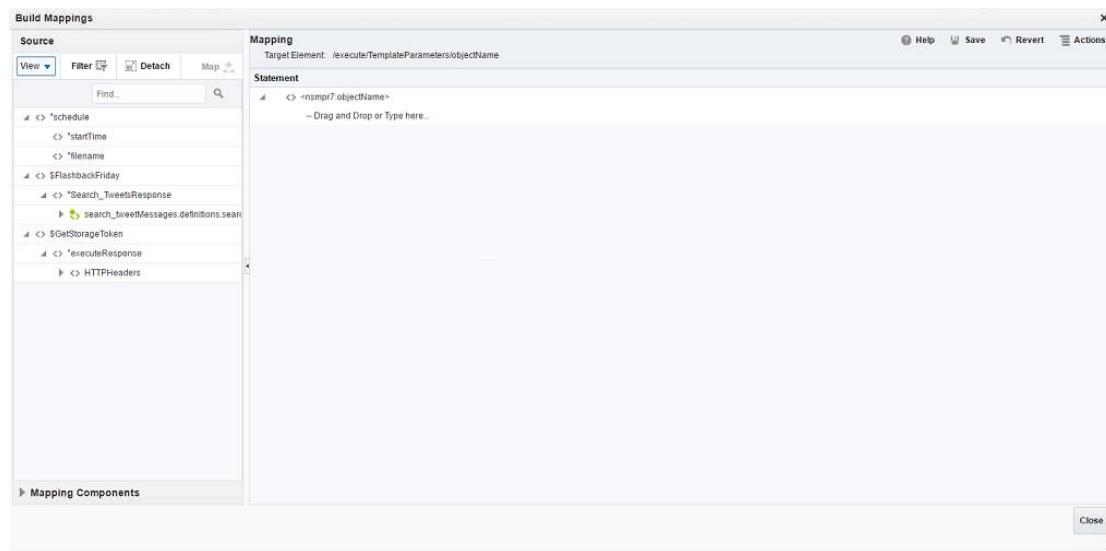
The first parameter is **X-Auth-Token** header in the HTTP request. The value for the request header should be identical as the token returned from the authentication action.

The result of invocation to authenticate to Oracle Storage Cloud is a source in the left. Collapse the source to find the custom response header **X-Auth-Token**, and then drag it to the

X-Auth-Token header as target in the right. A link is established between the source and target as below.

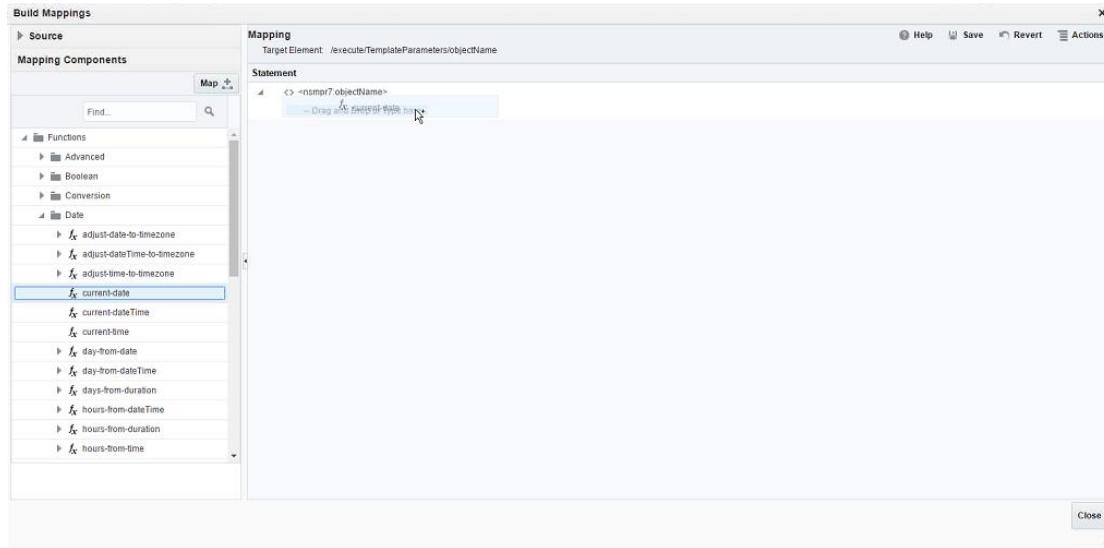


Next parameter must be customized is **objectName** in the URI template. It will dominate name of the data object in Oracle Storage Cloud. Click the parameter to launch following Mapping dialog.



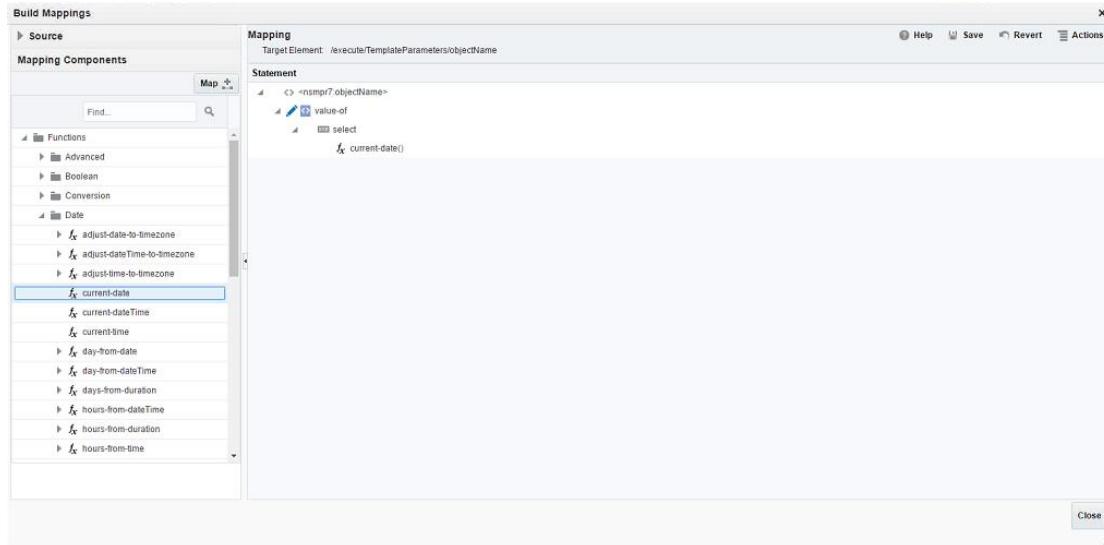
In the demo, current date will be used as name of the data object. To get value of current date, we will use pre-defined function in the Integration platform.

Function is a kind of Mapping Component which is defined to return a value. All available functions can be selected from Mapping Components catalog in the left.



Select **Mapping Components** catalog in the left, locate the **current-date** function from the *Functions -> Date* path. Drag the function to the target parameter on the right. Following page will indicate that the parameter will use the function to retrieve current date as name of the data object.

Click **Save** action, then click **Close** button to save the mapping.



The last parameter in the invocation is the payload in the request. In the demo, the payload will be generated from the result retrieved from Twitter previously. The result of that invocation is a XML document too. So the standard and easiest way is to transform the document into the payload with a XSL template.

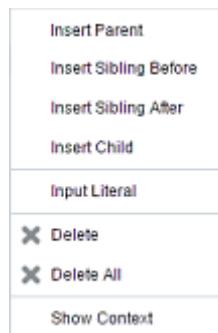
To make the demo simpler, we use same schema for the payload as that for the retrieved tweets. The XSL template only needs to iterate elements in the source document and generate elements with same name, with same number in the payload. All child elements are mapped one-to-one from source to target.

The screenshot shows the Oracle Integration Cloud Service Mapper Designer interface. On the left, the 'Source' schema tree includes elements like 'schedule', 'filename', '\$GetStorageToken', '\$FlashbackFriday', and 'Search_TweetsResponse'. On the right, the 'Target' schema tree includes 'execute', 'HTTPHeaders', 'CustomHTTPHeaders', 'TemplateParameters', and 'Search_TweetsResponse'. The central area is the 'Mappings' pane, which displays a mapping between the 'X-Auth-Token' source element and the 'X-Auth-Token' target element. A tooltip indicates: 'Drag and drop source to target to create a mapping.' and 'Click a checkbox on source or target to see mappings.'

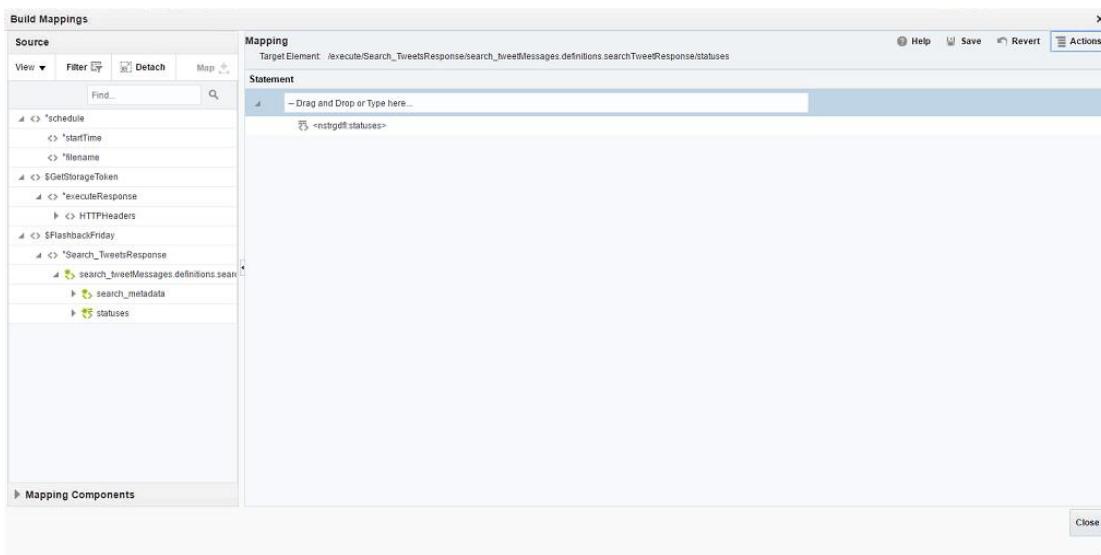
The Mapper Designer supports creating a XSL template interactively. In the schema tree, an element that can repeat more than once is decorated with an icon ahead. Those elements are always starting point of the template design. In the demo, we will start from the “**statuses**” element. Click the element in the target schema tree will launch following dialog.

The screenshot shows the 'Build Mappings' dialog. The 'Source' schema tree is identical to the one in the previous screenshot. The 'Mapping' tab shows a 'Statement' section with the XML code: <instgrdf:statuses>. The dialog has standard buttons for Help, Save, Revert, and Actions.

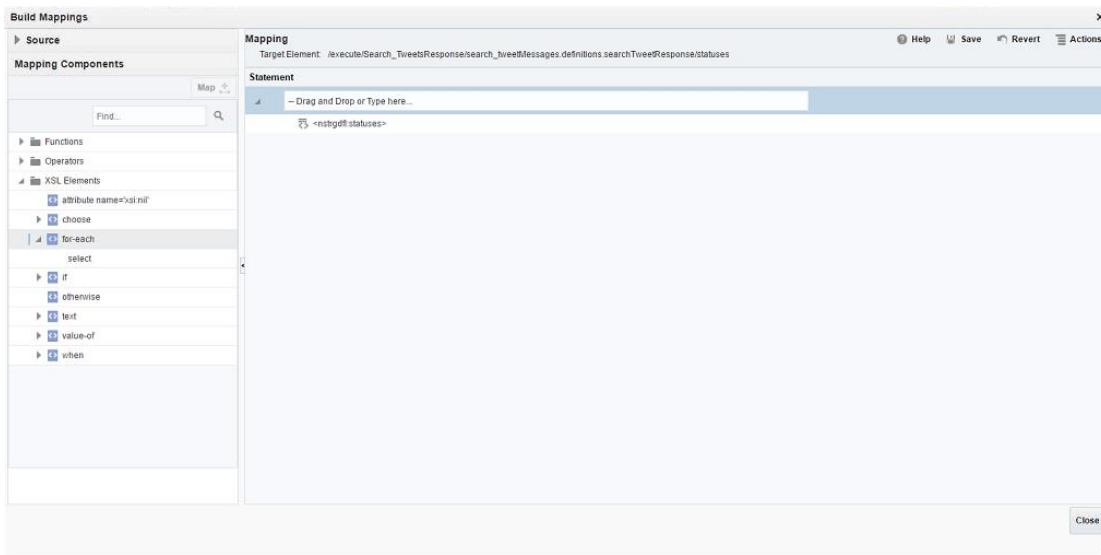
The transform process will iterate corresponding elements in the source document then generate elements in the target. To enable the iteration process, select the statement in the right, then click Actions button on the right-top of the dialog. Following menu will pop up.



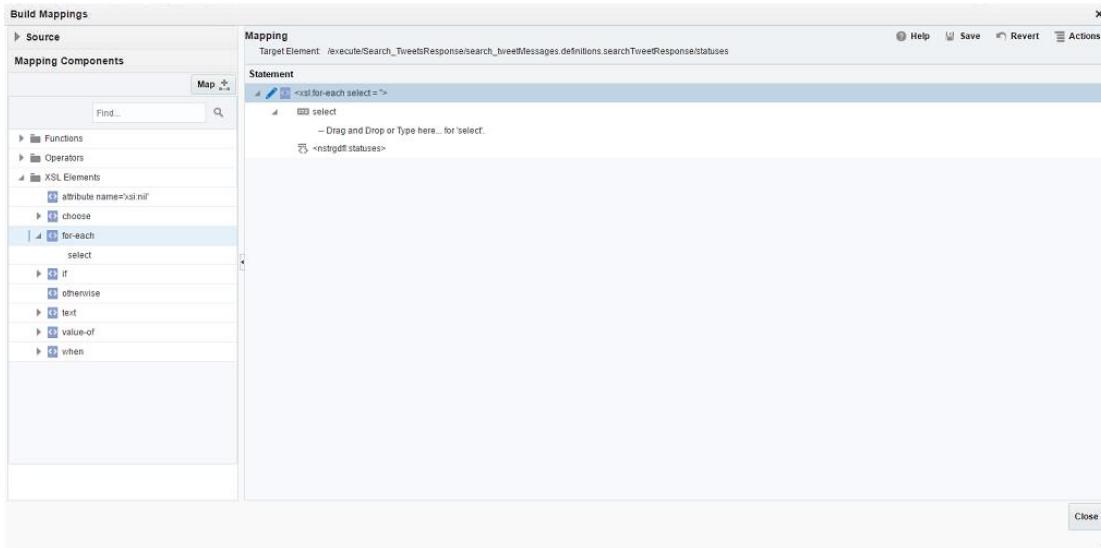
Select **Insert Parent** from the menu, an empty control is added as the parent.



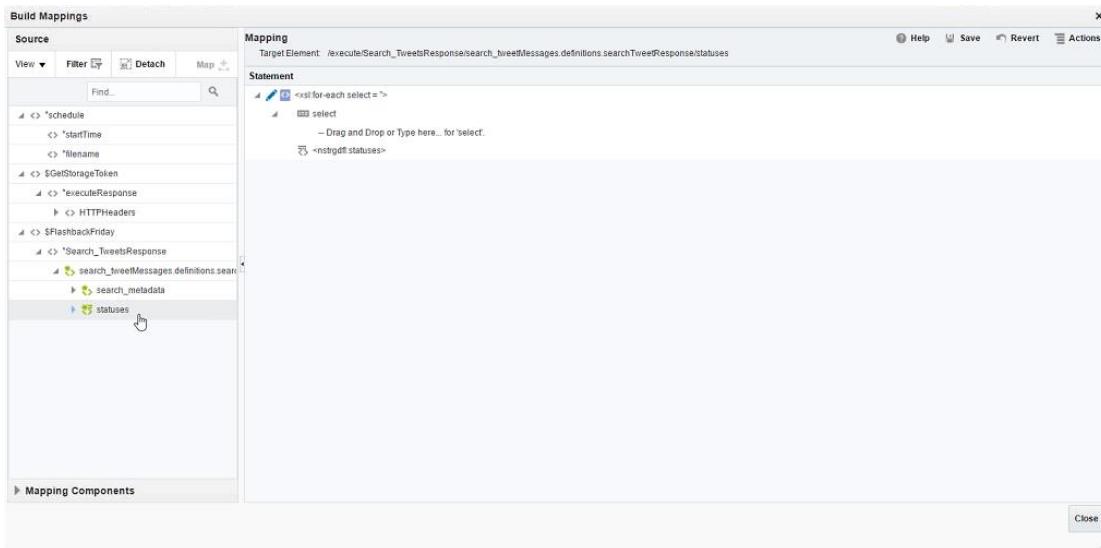
Next, you can add the iteration control to the parent statement. XSL Elements is a kind of Mapping Component in the tool. You can list available elements within **Mapping Components** catalog.



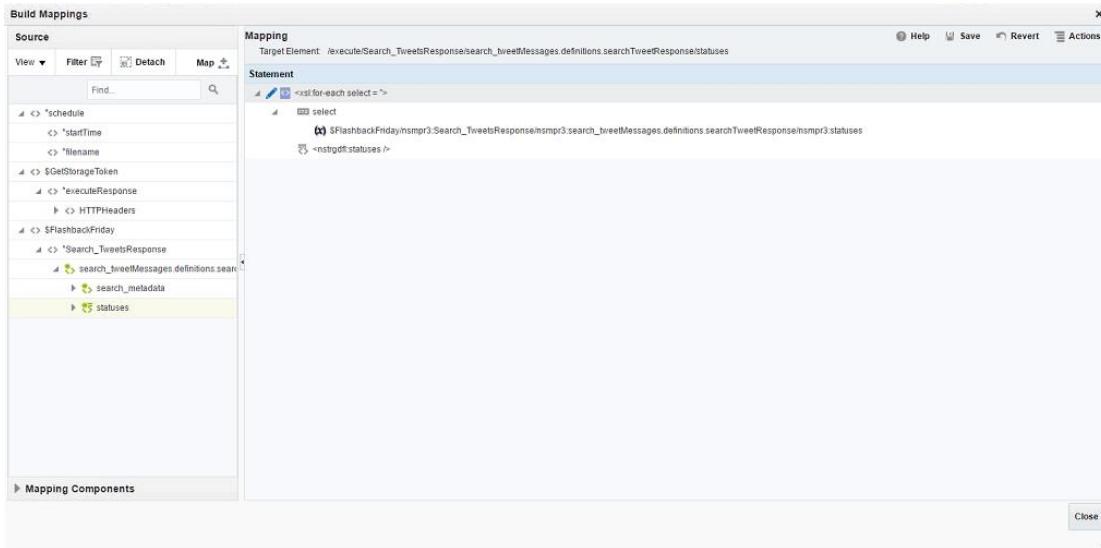
The **for-each** element is designed to iterate elements in a XML document. You can find it under the **XSL Elements** node. Drag and drop it onto the parent. The empty parent will be replaced by **for-each** element.



The **for-each** element requires a **select** attribute to specify name of the elements in the source document. User can map value of the attribute with an element in the schema of source document.

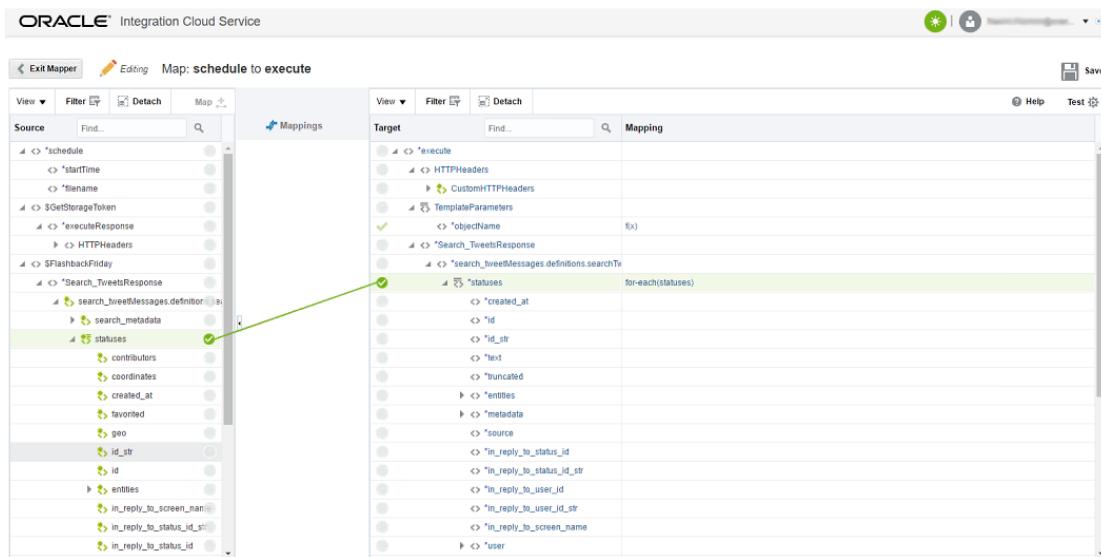


Find the source document schema from the **Source** catalog, and then locate the element that will be transformed to the target document. In this demo, it is the element with identical name. Drag and drop the **statuses** element to the **select** attribute under the **for-each** element in the right.



Click **Save** action on top of the dialog then click **Close** button and return to Mapper Designer page.

Next, we need to map child elements. Collapse the **statuses** element in the schema tree on both sides. All child elements are listed so that you can easily establish the map.



The mapping rules used in this demo include:

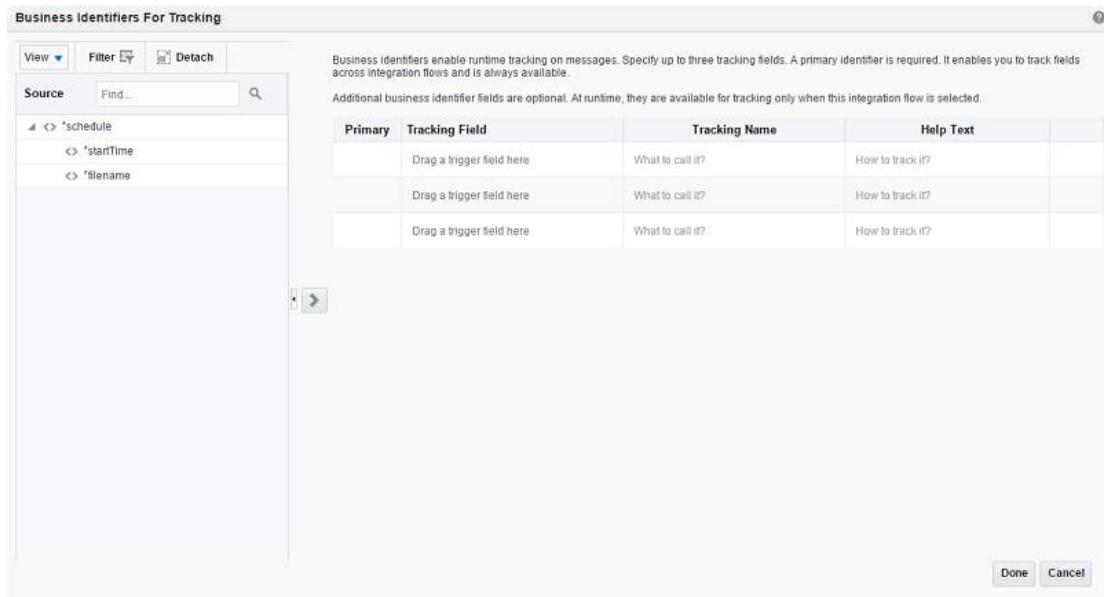
1. Map the elements with identical name between source and target.
2. Only need to map those elements you want them appear in target document.

To implement those rules, user can simply drag selected child elements in the source to the elements with identical name in the target.

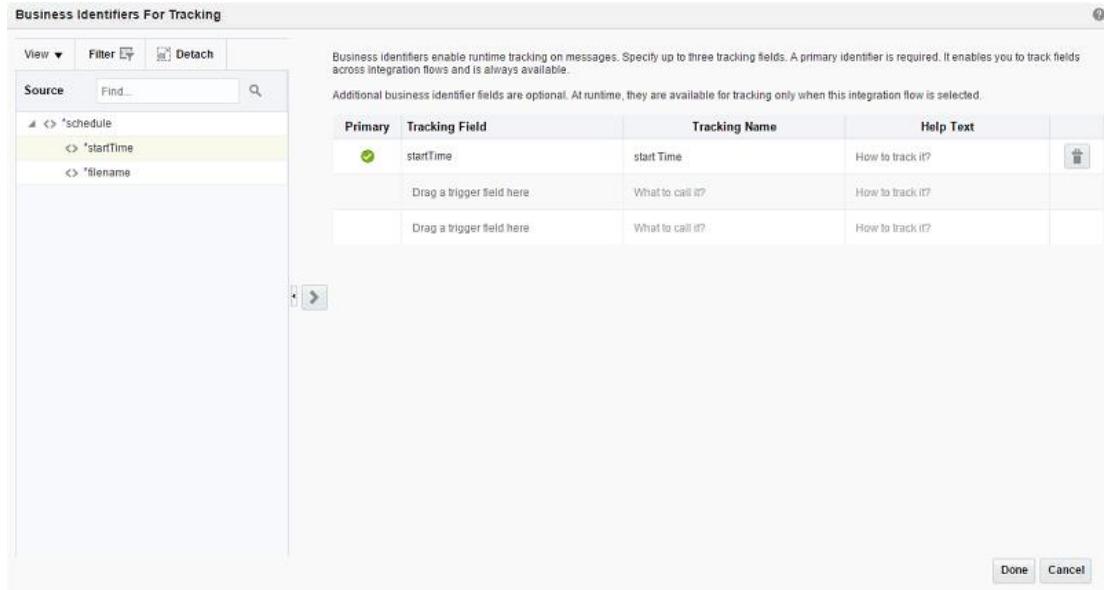
When required element mappings are done, click **Save** action to keep your modification safe. Then click **Exit Mapper** button and return to the Integration Designer.

Now the integration flow is completed. User can further customize tracking for the integration. Tracking is a mechanism of Oracle Integration Cloud Service to enable administrators to identify a running integration. Tracking record is composed of information or business identifier from the running integration.

Click **Tracking** action in the upper right to launch following dialog to customize tracking for the integration. User can select up to three business identifiers available from the list in the left as tracking fields. One of them must be marked as primary tracking field.



In this demo, we only rely on the time when the integration was launched to identify a running integration. Drag the schedule variable **startTime** from the left and drop it into Tracking Field column in the right, then click **Done** button in the dialog.



Congratulation, you have done all design efforts for the integration. Click **Save** action in the Integration Designer to save your work then click **Exit Integration** button.

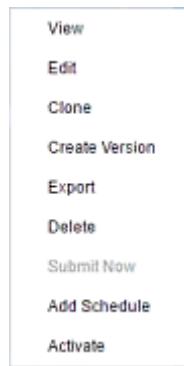
Step 4: Activate the Integration

The integration is ready to run. In this step, we will demonstrate how to activate an integration and monitor it after execution.

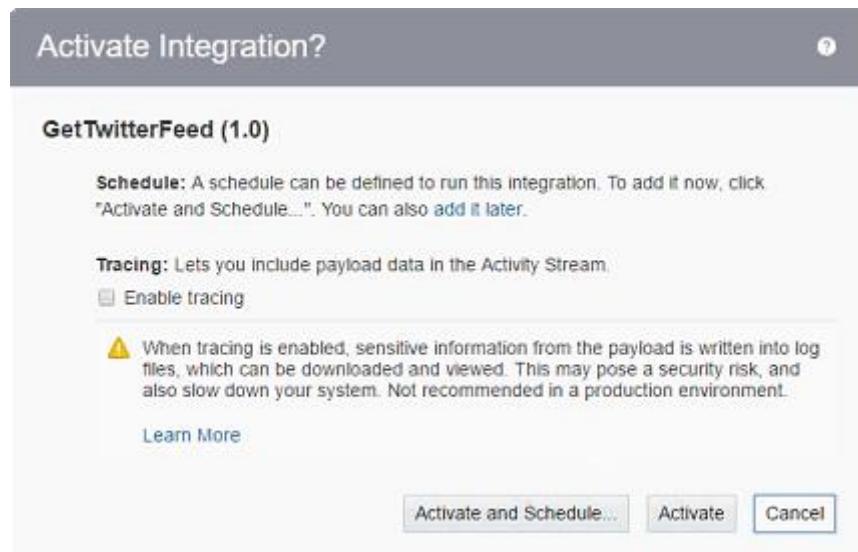
4.1 Activate an integration flow

In the following page, find the integration we just created. You may notice that it is not activated yet.

Click button in the right of the integration, a menu as below will pop up. Select **Activate** item in the menu.



Following dialog will appear to let user control how the integration is activated.



Click **Activate and Schedule...** button to schedule the integration right now. Following designer gives user additional control on the integration schedule.

The screenshot shows the Oracle Integration Cloud Service Scheduler interface. A message at the top indicates that the integration 'GetTwitterFeed (1.0)' was activated successfully. The main section displays a schedule named 'Run GetTwitterFeed' for the integration 'GetTwitterFeed'. The schedule is set to run 'Only Once' on April 5, 2017, at 6:54 PM UTC. The 'Frequency' dropdown is currently set to 'Only Once'. The 'Save' button is visible in the top right corner.

In this demo, we will customize **Frequency** of execution of the integration. From the drop-down menu you can see that the integration can be scheduled at fine-grained level.

The screenshot shows the same Oracle Integration Cloud Service Scheduler interface, but the 'Frequency' dropdown has been expanded. The 'Only Once' option is still selected, but the dropdown also lists 'Hours and Minutes', 'Days', 'Weeks', and 'Months' as alternative scheduling intervals. The 'Save' button is visible in the top right corner.

In this demo, select **Hours and Minutes** to schedule the integration repeatedly. Controls in the dialog as shown below let user customize the schedule.

The screenshot shows the Oracle Integration Cloud Service Scheduler interface with the 'Frequency' dropdown expanded to 'Hours and Minutes'. The configuration shows 'Every 0 hour(s) 0 minute(s)'. The 'Save' button is visible in the top right corner.

Increase the **hour(s)** field to 1 to schedule the integration every 1 hour. Click **Save** action in the upper right, then click **Exit Scheduler**. Following page will display and let user to review the schedule and decide when to submit the schedule.

The screenshot shows the 'Schedule and Future Runs' page. At the top, there's a navigation bar with links for Home, Designer, Monitoring (which is selected), and Administration. The date and time are shown as Apr 5, 2017 6:53:55 PM UTC. Below the navigation, there's a section for a scheduled run named 'Run GetTwitterFeed'. It includes fields for 'From' (Apr 5, 2017 6:54 PM UTC) and 'Until' (Never (repeat indefinitely)). A note says 'When this schedule is started, integration will run Every 1 hour(s) and 0 minute(s)'. Below this, there are tabs for 'Future Runs' (selected), 'All', and 'Scheduled', with a 'Next 24 Hours' dropdown. A message states 'There is no data to display.'

Click **Submit Now** button to schedule the integration immediately.

4.2 Track and monitor an integration flow

After the integration is scheduled, user can track and monitor the integration through Oracle Integration Cloud Service. All monitoring functions are consolidated in the following monitoring page. User can go to the page by clicking **Monitoring** link on the top.

The dashboard in the page will give user an overview of the whole Integration Service instance from workloads to performance. The panel on the left is a quick navigation to different monitoring functions.

The screenshot shows the 'Integrations Dashboard' page. On the left, there's a sidebar with navigation links for Monitoring (selected), Dashboard, System Health, Runtime Health, Design Time Metrics, Integrations (selected), Agents, Tracking, and Errors. The main area has a title 'Integrations Dashboard | Last 1 Hour'. It features several cards: 'SUCCESS RATE' (100.0%, ACCEPTABLE), 'MESSAGES' (2 INITIATED Last 1 Hour), 'FAILURES' (0 FAIL.MESSAGES Last 1 Hour), 'CONNECTIONS' (12 CURRENTLY USED 12 of 18), 'INTEGRATIONS' (23 CURRENTLY ACTIVATED 23 of 34), and 'SCHEDULED INTEGRATIONS' (6 total, 5 Paused, 1 Running). Below these are two charts: 'Hourly History for the last 24 hours' and 'Daily History for the last 30 days'. A legend on the right indicates activity levels: Activity Stream (yellow), Download Logs (orange), and Download Incident (green).

Select **Integrations** from the left panel will show following Integration Monitoring page. In the page, user can see clearly statistics of execution history of integrations.

ORACLE Integration Cloud Service

Integration	Last Message	Received	Processed	Success	Errors
GetTwitterFeed 1.0	Last Message just now Schedule not started	3	3	2	0
FlashbackFriday2 1.0	Last Message NO DATA Schedule not started	0	0	0	0
MFT ERP Load invoice fap1386 1.0	Last Message NO DATA	0	0	0	0
LoadAndSyncPersonClone 1.0	Last Message NO DATA	0	0	0	0

Select **Tracking** from the left panel will show following Integration Tracking page. In the page, user can see every scheduled execution of integrations. You may notice that the tracking fields you selected for integrations are displayed here. It helps users to uniquely identify an integration execution.

ORACLE Integration Cloud Service

Execution	Start Time	Run ID	Instance ID	Status
GetTwitterFeed (1.0)	2017-04-05T19:13:09.141+00:00	1200560	1200560	Completed just now
GetTwitterFeed (1.0)	2017-04-05T19:08:09.581+00:00	1200559	1200559	Completed 5 minutes ago
GetTwitterFeed (1.0)	2017-04-05T18:54:09.280+00:00	1200557	1200557	Failed 19 minutes ago

Click an integration execution in the list will jump to following page. User can view the progress and state of the integration flow.

ORACLE Integration Cloud Service

GetTwitterFeed (1.0) start Time : 2017-04-05T19:13:09.141+... Instance ID : 1200560

Completed just now

The diagram shows a vertical sequence of integration steps. From top to bottom, the steps are: 1. (green circle), 2. (blue square), 3. (green circle), 4. (blue square), 5. (green circle), 6. (blue square), and 7. (green circle). Each step has a small icon next to it, likely indicating its type or status.