

Preparing to deliver the workshop

Instructor requirements:

- Being very comfortable with 3scale both in terms of configuration and of operations.
Knowing the basics of the Developer Portal
- Being comfortable with RH SSO and especially around OpenID Connect protocol
- Being able to use Postman (or similar REST client) to build REST request and REST authentication
- Having a basic knowledge of OpenShift
- For the Fuse Online part 2 options are available:
 - A Fuse Online account, sign up here using your Employee sub:
<https://www.openshift.com/products/fuse>
 - Using the local installation with minishift following these instructions:
<https://github.com/syndesisio/syndesis/tree/master/install>. Once you have minishift running use this commands to create the syndesis app:
oc new-project syndesis
oc create -f support/serviceaccount-as-oauthclient-restricted.yml
oc create -f syndesis.yml
oc new-app --template=syndesis -p
ROUTE_HOSTNAME=syndesis.192.168.42.73.nip.io -p OPENSHIFT_MASTER=\$(oc whoami --show-server) -p OPENSHIFT_PROJECT=\$(oc project -q) -p
OPENSHIFT_OAUTH_CLIENT_SECRET=\$(oc sa get-token syndesis-oauth-client) -p
DEMO_DATA_ENABLED=true -p CONTROLLERS_EXPOSE_VIA3SCALE=true -p
SAR_PROJECT=\$(oc project -q)

● Once syndesis is running locally you can use ngrok (<https://ngrok.com/>) to expose the integration externally

Attendees requirements:

- To use the GUI of OpenShift:
https://docs.openshift.com/container-platform/3.11/architecture/infrastructure_components/web_console.html#browser-requirements
- Basic knowledge of REST protocol
- Basic knowledge of containers
- Basic understanding of OAuth social applications

Environment:

[TODO](#)

Duration and format

This is a workshop designed for approximately 2 hours delivery. The focus is on the adoption of the products in the financial industry. This is not an introduction to the different products used in the PSD2 showcase environment, but it is focused on the modules and functionalities relevant for the FSI case. Typically, instructors would talk through the introduction slides, and then for each hands-on lab, explain the steps needed to achieve the objective of the lab with reference to content in the slides. At the end of each activity there will be a checkpoint.

Timeline

30 minutes	Architecture and key features
30 minutes	Start of Practical Part 1 Reference Fuse features Financial Backend service demo and building blocks on Fuse Online [simple demo]
25 minutes	Reference 3scale features Basics of Service Configuration and Integration in 3scale. Basics of Developer portal content publishing [hands-on of the attendees in parallel]
5 minutes	CHECKPOINT
10 minutes [here]	Introducing OIDC & OAuth
30 minutes	Reference RH SSO features Basics of RH SSO and 3scale OIDC API authentication [hands-on of the attendees in parallel]
5 minutes	CHECKPOINT
15 minutes	Break
10 minutes	Start of Practical Part 2 Reference OpenShift 10.000 feet view

	OpenShift high level walkthrough [simple demo]
5 minutes	Q&A and closing

Practical Part 1

Integr8ly

Login into the integr8ly environment main page

The screenshot shows the Red Hat Solution Explorer interface. On the left, there's a sidebar titled "RED HAT SOLUTION EXPLORER" with a "developer" dropdown. The main content area has two sections: "Start with a walkthrough" and "Applications".

Start with a walkthrough:

- Integrating event-driven and API-driven applications (AMQ) - Preview button, Get Started button, 15 min duration.
- Integrating event-driven and API-driven applications (EnMasse) - Get Started button, 21 min duration.
- Integrating API-driven applications - Preview button, Get Started button, 21 min duration.

Applications:

- Red Hat OpenShift - Ready for use
- Red Hat 3scale API Management Platform - Ready for use
- Red Hat AMQ - Ready for use
- Eclipse Che - Ready for use
- EnMasse - Ready for use
- Red Hat Fuse - Ready for use
- Red Hat Developer Launcher - Ready for use

At the bottom, there are browser tabs for various documents and a navigation bar with "1 / 2".

Explain why we choose integr8ly environment following the presentation you can find in the links. Explain how it provides out of the box integration between products but it doesn't change the base product. How you can still install this type of environment starting from a “clean” openshift installation.

Quick recap of the platforms or products installed and their main functionality.

Fuse Online

Then start by opening Red Hat Fuse.

☰ RED HAT FUSE ONLINE

evals98

Home

Integrations

Connections

Customizations

Settings

System Metrics

0 Integrations Green: 0 Red: 0	4 Connections	0 Total Messages Green: 0 Red: 0	Uptime Since Jan 10th 22:44 PM 2 minutes
-------------------------------------	---------------	---------------------------------------	--

 Create an Integration

There are currently no Integrations. Click the button below to create one.

[Create Integration](#)

Connections

[View All Connections](#) [Create Connection](#)

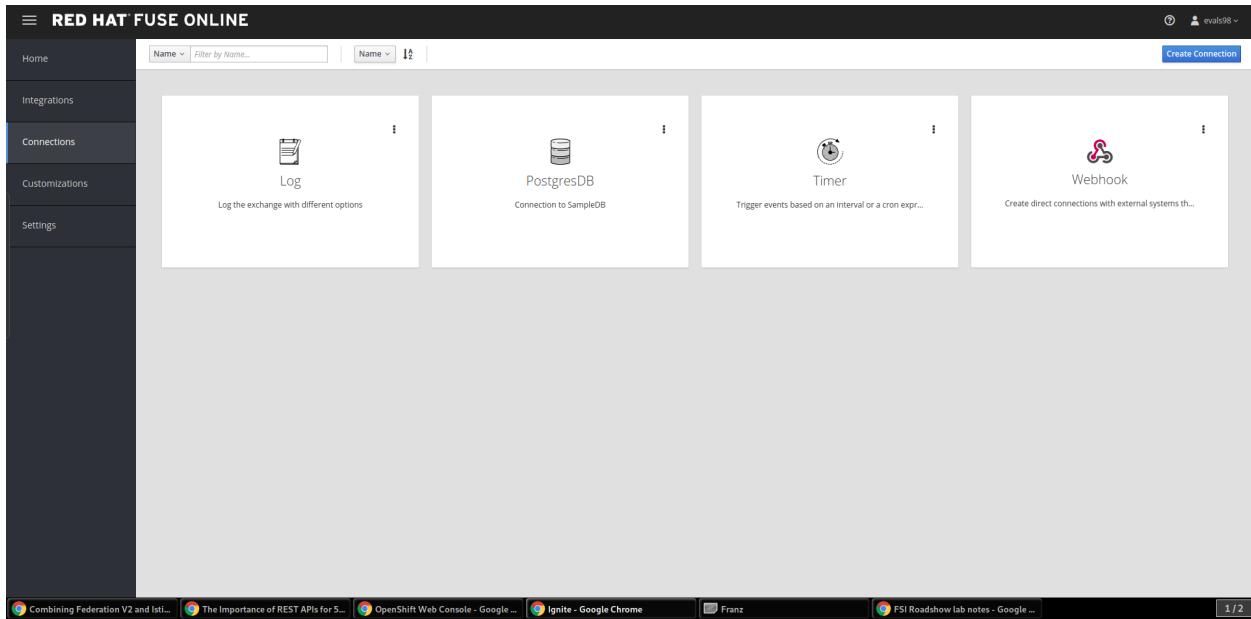
			
PostgresDB	Log	Timer	Webhook

Combining Federation V2 and Isti... The Importance of REST APIs for... OpenShift Web Console - Google ... Ignite - Google Chrome Franz FSI Roadshow lab notes - Google ...

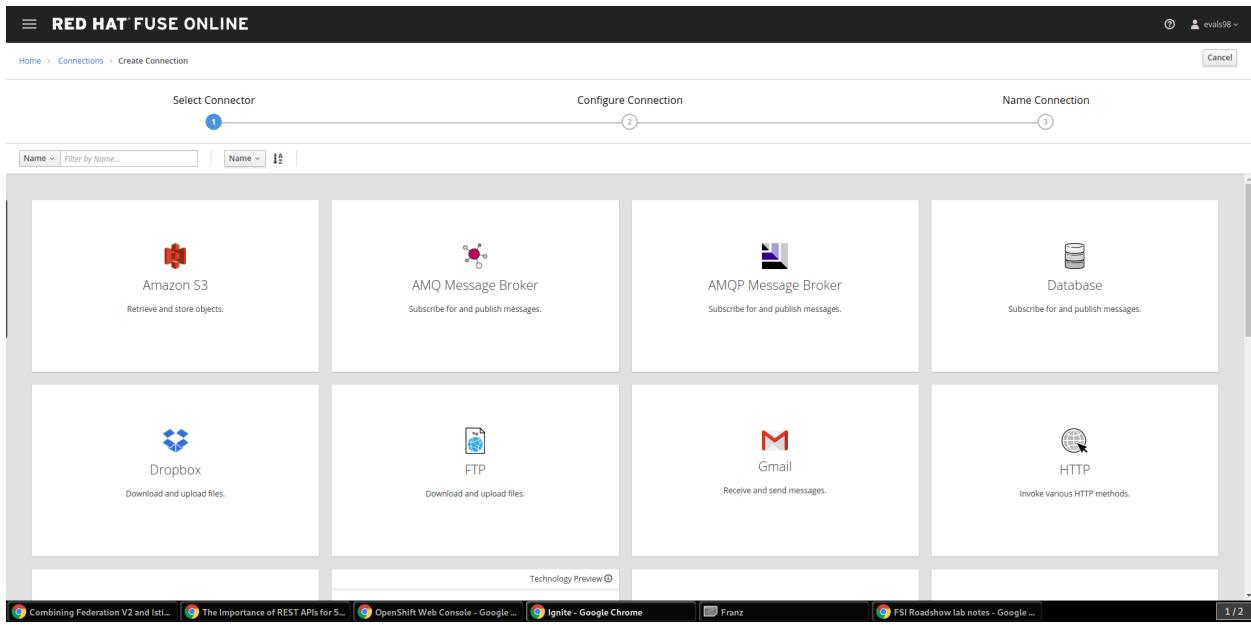
Explain the main dashboard, especially important when you then show messages coming request being sent.

The screenshot shows the Red Hat Fuse Online web application. The left sidebar contains navigation links: Home, Integrations (which is selected), Connections, Customizations, and Settings. The main content area has a search bar at the top with dropdowns for 'Name' and 'Filter by Name...', and a sorting section with 'Name' and 'Type'. A large central area features a 'Create an integration' button with a plus sign icon. Below it, a message states: 'There are currently no integrations available. Please click on the button below to create one.' At the bottom right of the main area, there is a 'Create Integration' button. The top right corner shows user information: 'evalrh98' and a profile icon.

Explain the role of the integration window (seeing graphically the mediation/transformation) and the relevance for citizen developers. Little detour on what behind (Camel) to reassure people that's not new technology but well tested one and with 100s of possible connectors.



Connections window: those connections are pre-configured for the scenario shown on the main page of integr8ly. Explain that you can start or end (or sometimes interpose) connections which are really configured connectors. Anticipate that later on we will be showing the configuration of 2 connectors and open the Create connection option.



Explain the various capabilities available here and tell audience new connectors are coming in the future and will be ported onto Fuse Online. Quick overview of “services” available here. Make a couple examples about possible integrations or mashups. Cancel button.

The screenshot shows the 'API Client Connectors' section of the Red Hat Fuse Online interface. On the left, a dark sidebar lists navigation options: Home, Integrations, Connections, Customizations (which is selected), and Settings. The main content area has tabs for 'API Client Connectors' (selected) and 'Extensions'. Below the tabs, there's a search bar with dropdowns for 'Name' and 'Filter by Name...'. A large central area features a plus sign icon and the text 'Create API Connector'. It includes a message: 'There are currently no API connectors available. Please click on the button below to create one.' A blue 'Create API Connector' button is centered at the bottom of this area. At the very bottom of the page, there's a browser-style address bar showing multiple tabs and a page number indicator '1 / 2'.

Customization view. Explain the 2 panes available here, the first one dedicated to API connections, which we will be using afterwards as Connection in the Integration.

The screenshot shows the 'Extensions' section of the Red Hat Fuse Online interface. The left sidebar is identical to the previous screenshot, with 'Customizations' selected. The main content area has tabs for 'API Client Connectors' and 'Extensions' (selected). Below the tabs, there's a search bar with dropdowns for 'Name' and 'Filter by Name...'. A large central area features a plus sign icon and the text 'Import Extension'. It includes a message: 'There are no extensions available.' A blue 'Import Extension' button is centered at the bottom of this area. At the very bottom of the page, there's a browser-style address bar showing multiple tabs and a page number indicator '1 / 2'.

Extensions pane: this is the part where you explain how to extend the functionalities of the platform. Ideally you would have prepared one compiled extension and you can import it live to show how easy it is to add transformation functionalities to the product.

Conclude on settings

The screenshot shows the 'OAuth Application Management' section of the Red Hat Fuse Online interface. It lists four connectors: Gmail, Salesforce, SAP Concur, and Twitter. Each connector entry includes a status message indicating 'Access to this application is not configured.' and a 'Register' button.

Important to highlight here is the fact that to retrieve end user data for this connectors you have to first authenticate, using OAuth2.0 protocol. This authentication is configured in this window

The screenshot shows the 'OAuth Application Management' section of the Red Hat Fuse Online interface, specifically for the Gmail connector. The 'Client ID' and 'Client Secret' fields are populated with values, and the 'Save' button is visible at the bottom of the configuration panel.

Make sure to explain a common use case of this, for example authenticating with Twitter account when accessing a news portal.

THE FUSE ONLINE PART IS DEMO ONLY, YOU WILL BE USING AN INSTANCE OF FUSE

ONLINE THAT YOU CAN CREATE WITH YOUR EMPLOYEE SKY RED HAT ACCOUNT HERE

<https://www.openshift.com/products/fuse>

OR

Running syndesis on minishift on your local machine

The screenshot shows the 'RED HAT FUSE ONLINE' interface for creating an API client connector. The top navigation bar includes 'Customizations > API Client Connector'. The main area has a progress bar with four steps: 1. Upload OpenAPI Specification (highlighted), 2. Review Actions, 3. Specify Security, and 4. Review/Edit Connector Details. Step 1 is titled 'Upload OpenAPI Specification' and contains instructions: 'Upload the OpenAPI 2.0 file for your custom API client connector. Custom APIs are RESTful APIs and can be hosted anywhere, as long as a well-documented API specification is available and conforms to the OpenAPI standard.' It offers two options: 'Upload an OpenAPI file' (selected) and 'Use a URL', with a 'Choose File' button showing 'No file chosen'. A 'Next' button is at the bottom.

Now start with the configuration of the integration. **home -> customizations -> API client connector.** Upload the following OpenAPI file:

<https://raw.githubusercontent.com/lucamaf/open-banking-roadshow/master/open-data-apis-nokey.json>

Explain that this is the definition already exposed on the Open Banking portal for Open Data APIs as demoed before. **Next ->**

The screenshot shows the 'RED HAT FUSE ONLINE' interface, step 2: 'Review Actions'. The progress bar is at step 2. The main area displays the imported OpenAPI specification. It shows 8 operations under the 'IMPORTED' section and 7 warnings under the 'WARNINGS' section. The 'IMPORTED' section lists operations like 'GET /banks/(bank_id)/atms/(atm_id)' and 'GET /banks/(bank_id)/branches/(branch_id)'. The 'WARNINGS' section lists issues such as 'Operation GET /banks/(bank_id)/atms/(atm_id) does not provide a response schema for code 200'. Buttons for 'Next' and 'Review/Edit' are at the bottom.

Explain that there is a validation happening on the API definition, but that no error was identified so we can proceed with the configuration of the connector.

To show how easy it is to correct definitions, -> review/edit

The screenshot shows the Red Hat Fuse Online interface for editing an API definition. The left sidebar has sections for 'Paths' (listing endpoints like /banks, /banks/(bank_id), etc.) and 'Definitions'. The main area has tabs for 'INFO', 'CONTACT', 'LICENSE', 'TAGS', and 'SECURITY SCHEMES'. The 'LICENSE' tab shows the GNU GPL v3 license with its terms. The 'TAGS' section lists several tags with their descriptions.

This opens a window on Apicurito which is a scaled down version of Apicurio, our API Design platform. Show how easy it is to change elements graphically and also explain that with the help of this tool an API team can start with a Design First approach when configuring API. Also the same team doesn't need to know about the rule of OpenAPI specifications thanks to this tool.

If you want modify one element (not the /banks endpoint). -> back -> next

The screenshot shows the Red Hat Fuse Online API configuration wizard with four steps: 'Upload OpenAPI Specification', 'Review Actions', 'Specify Security', and 'Review/Edit Connector Details'. The 'Specify Security' step is active, showing a form with an 'Authentication Type' section where 'No Security' is selected. A 'Next' button is at the bottom. The status bar at the bottom shows multiple browser tabs open.

Explain that the service is exposed without any protection (that's one of the reasons we are going to be using 3scale later on). -> next

Review/Edit Connector Details

Connector Icon: Choose File (No file chosen)

* Connector Name: open-data-apis

Description: OpenBanking open data apis

Host: https://open-data.b9ad.pro-us-east-1.openshiftapps.com

Base URL: /open-data

Create API Connector

Finalize the configuration, explaining that the various fields are populated from the OpenAPI definition. (don't touch the field, especially host or base url!). You can change the icon of the connector if you want, to make it more personal. -> **create API connector**

API Client Connectors

ignite creates an API client connector when you upload a valid OpenAPI 2.0 specification that describes the API you want to connect to.

Name	Description	Used by integrations	Action
open-data-apis	OpenBanking open data apis	0 times	Delete

Highlight that we have also an indicator showing if the connector is being used somewhere.
Create connection

RED HAT FUSE ONLINE

Home > Connections > Create Connection

Select Connector Configure Connection Name Connection

1 2 3

Name	Filter by Name...	Name	12
 HTTPS	Invoke various HTTPS methods.	 IRC	Send and receive messages to/from an IRC chat.
 MQTT Message Broker	Subscribe for and publish messages.	 open-data-apis	OpenBanking open data apis
 Salesforce	Manage customer relations in the cloud.	 SAP Concur	Manage invoices, travel, and expenses.
 ServiceNow		 SFTP	
 Slack		 Telegram	

Technology Preview

Cancel Back Next 1 / 2

Open-data-apis

RED HAT FUSE ONLINE

Home > Connections > Create Connection

Select Connector Configure Connection Name Connection

1 2 3

open-data-apis Configuration

The fields marked with * are required.

Authentication Type: No Security

* Host: https://open-data.b9ad.pro-us-east-1.openshiftapps.com

* Base path: /open-data

API basePath for example /v2. Default is unset if set overrides the value present in OpenAPI specification.

Cancel Back Next 1 / 2

You can still modify some parameters of the connectors if you want before proceeding. -> next

RED HAT FUSE ONLINE

Home > Connections > Create Connection

Select Connector Configure Connection Name Connection

Add Connection Details

The fields marked with * are required.

* Connection Name: Name is required

Description:

Cancel Back Create

Give the Connection a name. -> create

RED HAT FUSE ONLINE

Home Integrations Connections Customizations Settings

Name Filter by Name... Name

Create Connection

Log Log the exchange with different options

open-financial-service Table Overview: Create the initial service instance, Create the connection instance, Create the connector instance. Effects: Include parent, Include the connection instance, Apply effects:

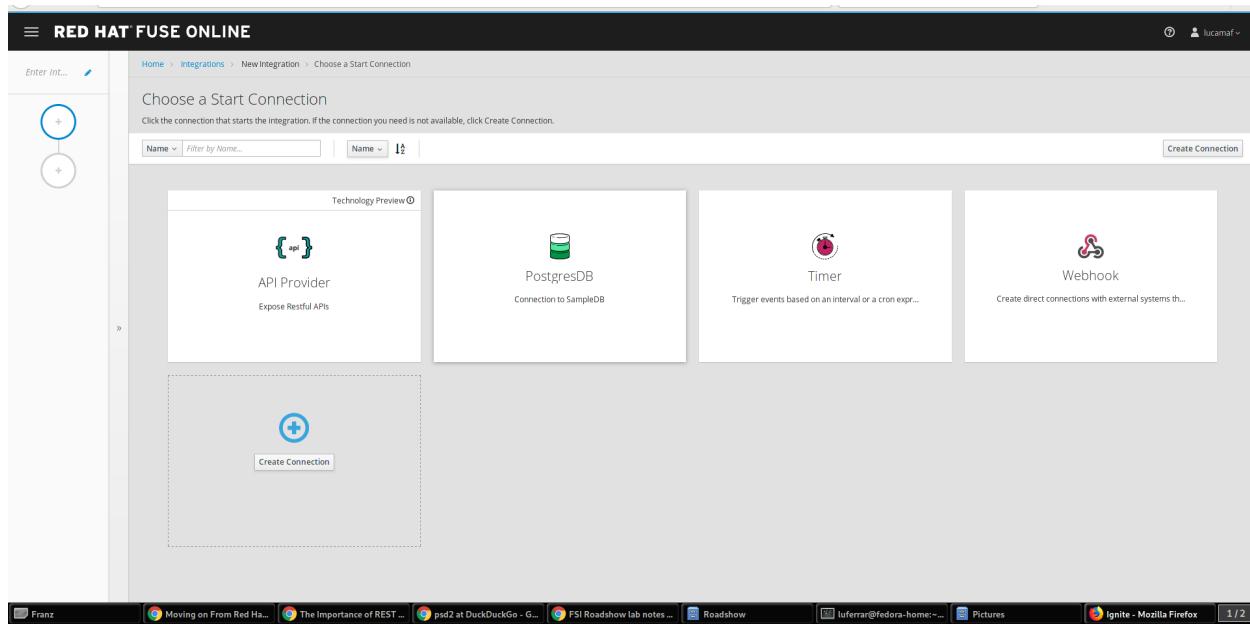
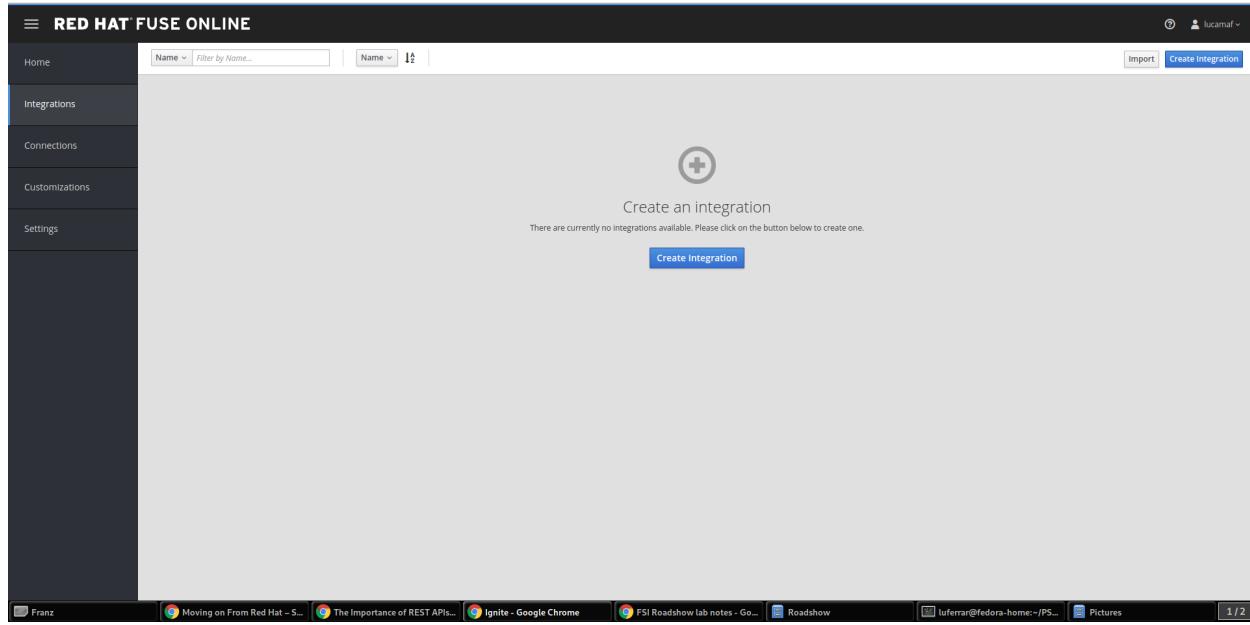
PostgresDB Connection to SampleDB

Timer Trigger events based on an interval or a cron expr...

Webhook Create direct connections with external systems th...

Combining Federation V2 and Isti... The Importance of REST APIs for S... OpenShift Web Console - Google ... Ignite - Google Chrome Franz FSI Roadshow lab notes - Google ... PSD2 1 / 2

Now that we have configured the start and the end of the integration path, we can start building the complete integration block. **Integrations -> create**



API provider. Upload the same definition as before.

RED HAT FUSE ONLINE

Enter int...

Home > Integrations > New Integration > Start integration with an API call

Start Integration with an API call

Execute this integration when a client invokes an operation defined by this API.

Review Actions

API DEFINITION

- Name: open-data-api
- Description: OpenBanking open data api

IMPORTED

8 operations

WARNINGS 8

- Some operations (8) have no operationid
- Operation GET /banks/{bank_id}/atms/{atm_id} does not provide a response schema for code 200
- Operation GET /banks does not provide a response schema for code 200
- Operation GET /banks/{bank_id} does not provide a response schema for code 200
- Operation GET /banks/{bank_id}/atms does not provide a response schema for code 200
- Operation GET /banks/{bank_id}/branches/{branch_id} does not provide a response schema for code 200
- Operation GET /banks/{bank_id}/products does not provide a response schema for code 200
- Operation GET /banks/{bank_id}/products/{product_id} does not provide a response schema for code 200

Review/Edit **Next** **Cancel**

Give the transformation a name and save and continue

RED HAT FUSE ONLINE

expose ...

API Provider

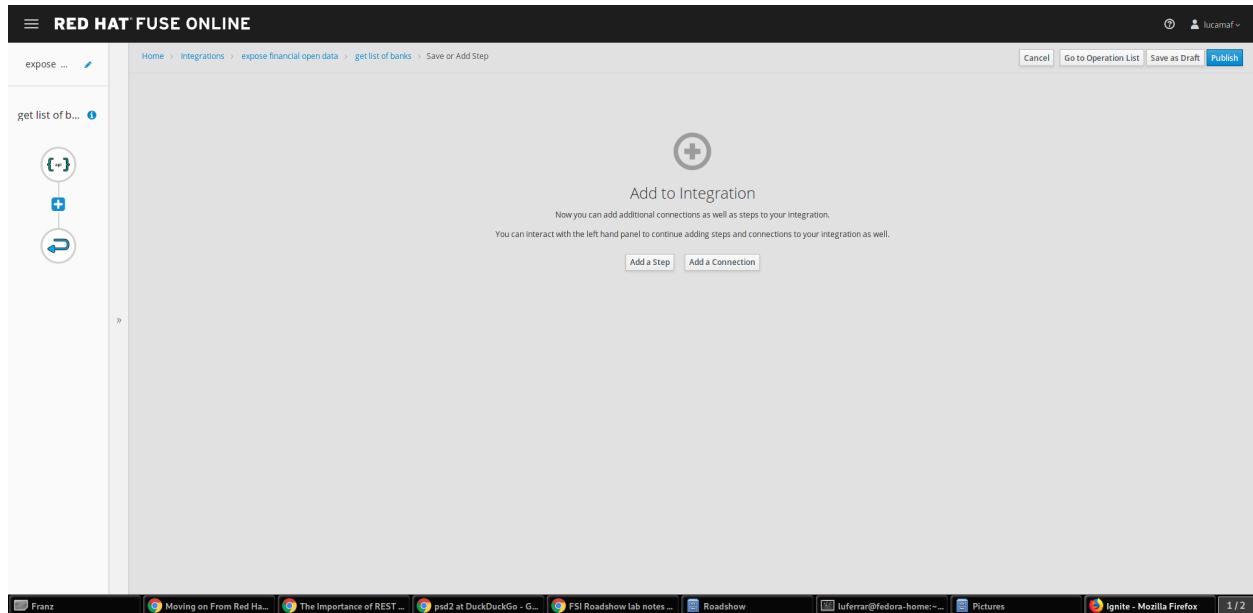
Home > Integrations > expose financial open data > Choose operation

Choose operation

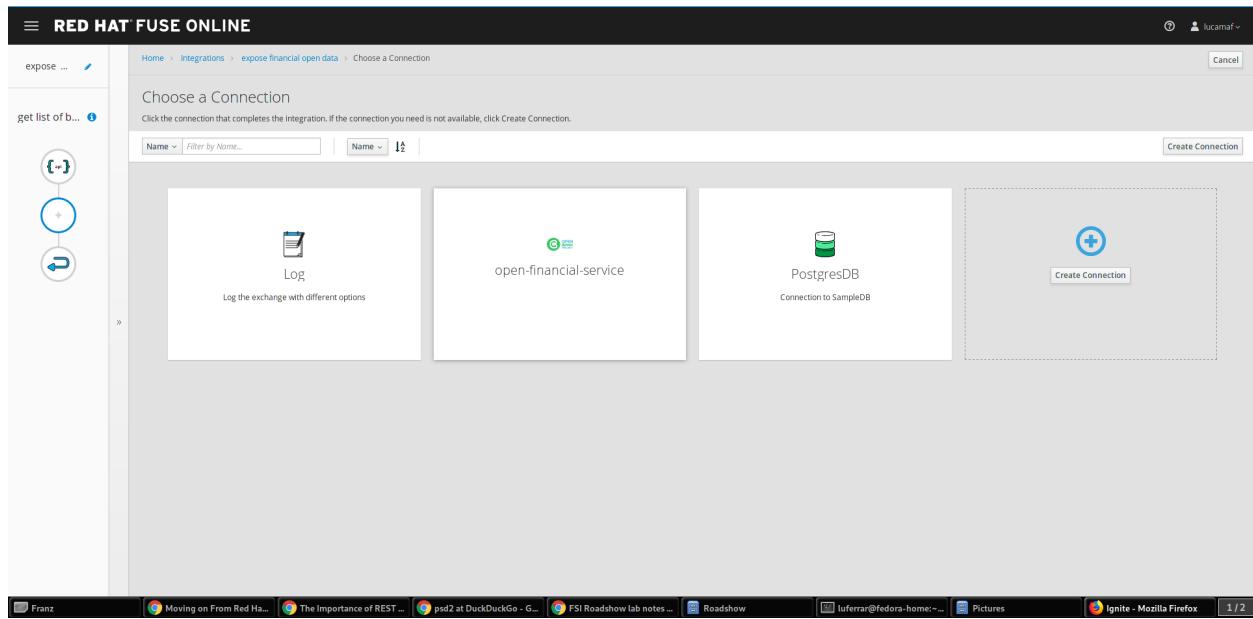
Click an operation to integrate an integration flow.

Name	Description	Status
get atm info	GET /banks/{bank_id}/atms/{atm_id}	501 Not Implemented
get bank atms	GET /banks/{bank_id}/atms	501 Not Implemented
get bank details	GET /banks/{bank_id}	501 Not Implemented
get branch info	GET /banks/{bank_id}/branches/{branch_id}	501 Not Implemented
get branches available by a specific bank	GET /banks/{bank_id}/branches	501 Not Implemented
get list of banks	GET /banks	501 Not Implemented
get product info	GET /banks/{bank_id}/products/{product_id}	501 Not Implemented
get products	GET /banks/{bank_id}/products	501 Not Implemented

Expose just one resource on Fuse Online, the endpoint to get a list of banks.



Add a connection to link this external endpoint to an existing connection (as we configured before).



RED HAT FUSE ONLINE

Home > Integrations > Choose Action

open-financial-service - Choose an Action

Choose an action for the selected connection.

Name Filter by Name... Name

GET /banks	get list of banks
GET /banks/(bank_id)	get bank details
GET /banks/(bank_id)/atms	get bank atms
GET /banks/(bank_id)/atms/(atm_id)	get atm info
GET /banks/(bank_id)/branches	get branches available by a specific bank
GET /banks/(bank_id)/branches/(branch_id)	get branch info
GET /banks/(bank_id)/products	get products
GET /banks/(bank_id)/products/(product_id)	get product info

Cancel

This screenshot shows the 'Choose Action' step in the Red Hat Fuse Online interface. It lists various REST endpoints for a 'open-financial-service' integration. The left sidebar shows a tree structure with 'get list of b...' selected. The main area displays a table of actions with their descriptions. At the bottom right, there are buttons for 'Cancel', 'Go to Operation List', 'Save as Draft', and 'Publish'.

RED HAT FUSE ONLINE

Home > Integrations > expose financial open data > get list of banks > Save or Add Step

Add to Integration

Now you can add additional connections as well as steps to your integration.
You can interact with the left hand panel to continue adding steps and connections to your integration as well.

Add a Step Add a Connection

Cancel Go to Operation List Save as Draft Publish

This screenshot shows the 'Add to Integration' step. It features a central 'Add to Integration' button with a plus sign. Below it is a message encouraging users to add connections and steps. At the bottom are 'Add a Step' and 'Add a Connection' buttons. The top navigation bar includes 'Save or Add Step' and other integration-related links. The bottom navigation bar shows the standard Firefox interface.

And now add a simple log of the request coming through. Add another connection.

RED HAT FUSE ONLINE

Home > Integrations > expose financial open data > get list of banks > Save or Add Step

Add to Integration

You can add additional connections as well as steps to your integration.

You can interact with the left hand panel to continue adding steps and connections to your integration as well.

Add a Step | Add a Connection

Add a Step | Add a Connection

1 / 2

RED HAT FUSE ONLINE

Home > Integrations > expose financial open data > Choose a Connection

Choose a Connection

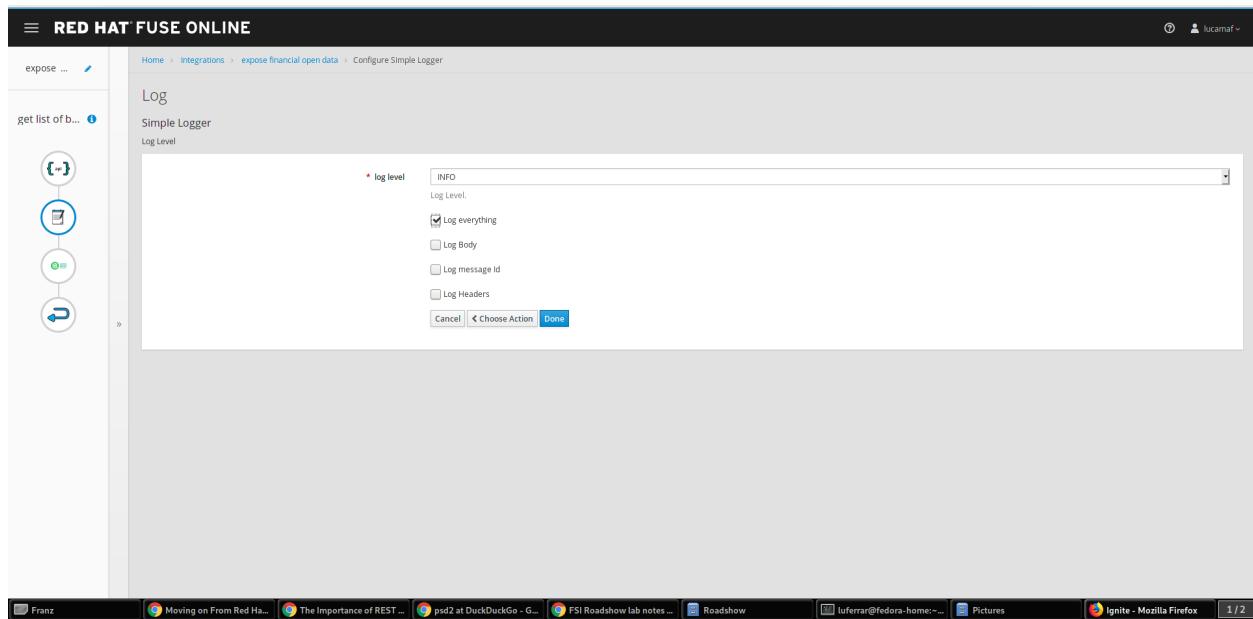
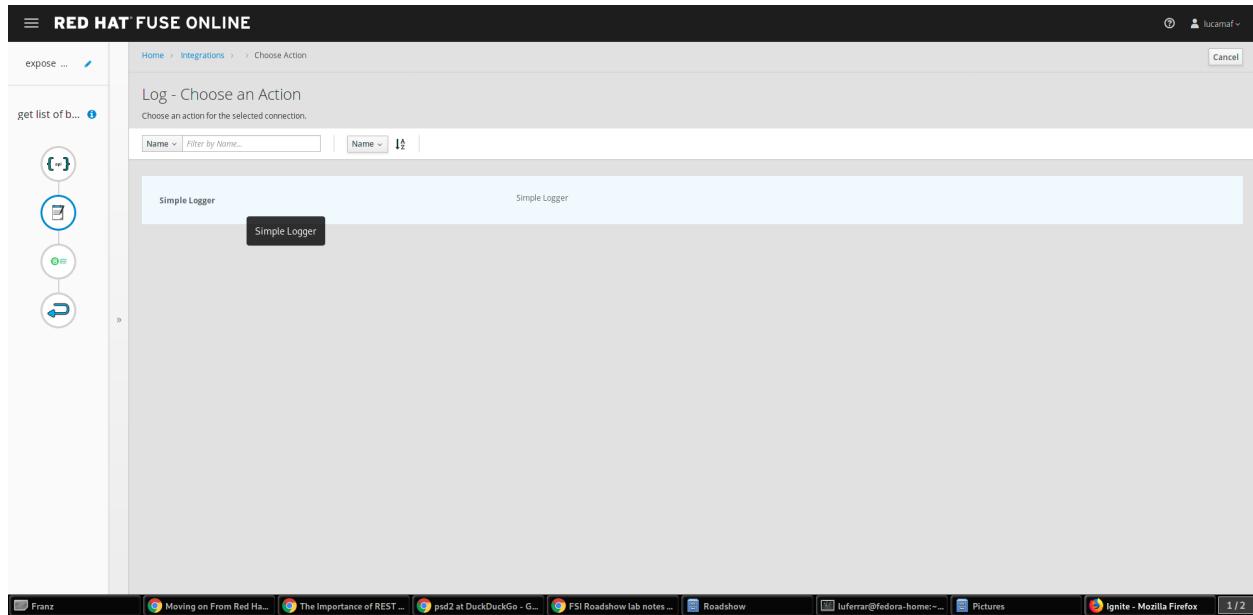
Click the connection that completes the integration. If the connection you need is not available, click Create Connection.

Name: Filter by Name... | Name:

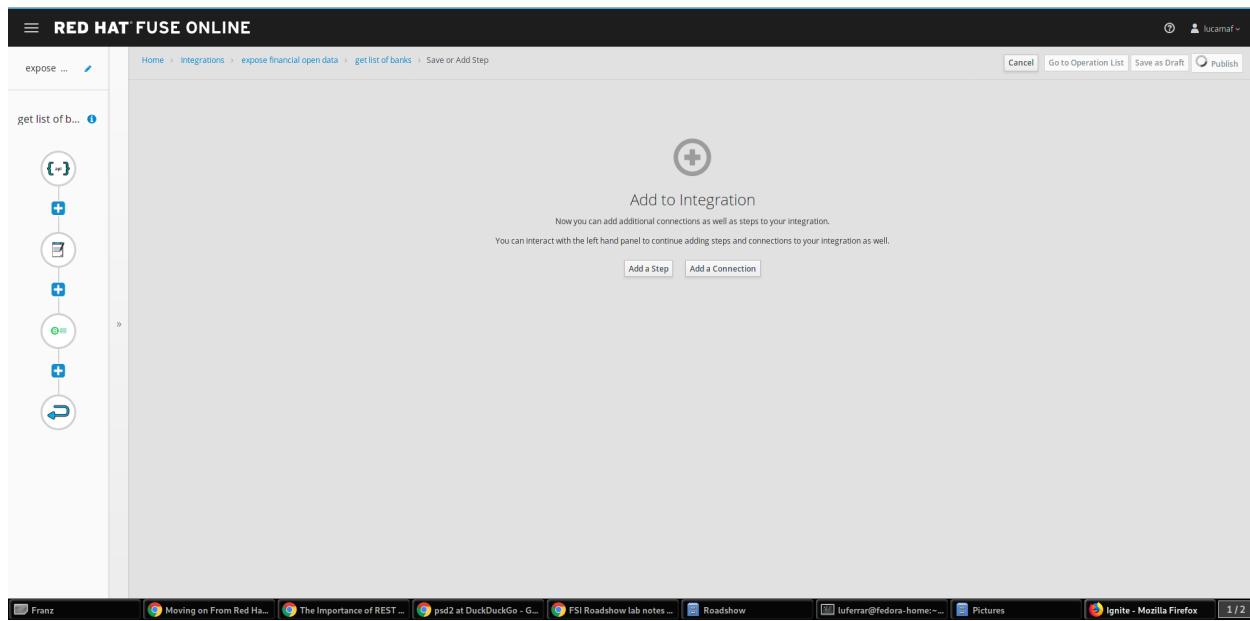
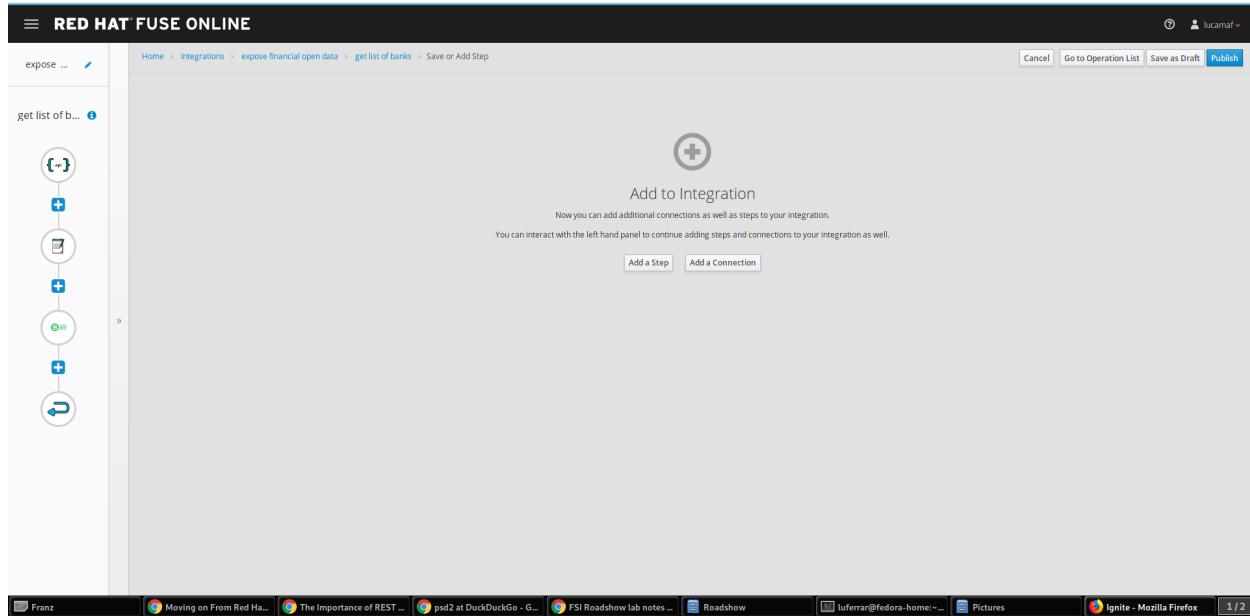
Create Connection

Create Connection

1 / 2



We are now ready to deploy this block in our platform. Hit **Publish**



We can notice the platform getting the required components and constructing the block. When the building is completed we will receive an External URL that we can use to test the Integration block.

The screenshot shows the Red Hat Fuse Online interface. On the left, there's a sidebar with 'Home', 'Integrations' (which is selected), 'Connections', 'Customizations', and 'Settings'. The main content area is titled 'expose financial open data' and shows a 'Published version 1'. Below this, there are tabs for 'Details', 'Activity', and 'Metrics'. Under 'Details', there are two circular icons: one for 'API Provider' and another for 'Flow'. An 'External URL' field contains the link <https://i-expose-financial-open-data-proj370378.6a63.fuse-ignite.openshiftapps.com>. A 'History' section indicates the integration was last published on Jan 13, 2019, at 20:09:25.

```

[{"id": "psd201-bank-x--uk", "short_name": "Bank X", "full_name": "The Bank of X", "logo": "https://static.openbankproject.com/images/sandbox/bank_x.png", "website": "https://www.example.com", "bank_routing": {"scheme": "OBP", "address": "psd201-bank-x--uk"}, "version": 1}, {"id": "psd201-bank-y--uk", "short_name": "Bank Y", "full_name": "The Bank of Y", "logo": "https://static.openbankproject.com/images/sandbox/bank_y.png", "website": "https://www.example.com", "bank_routing": {"scheme": "OBP", "address": "psd201-bank-y--uk"}, "version": 1}, {"id": "at02-bank-x--01", "short_name": "Bank X", "full_name": "The Bank of X", "logo": "https://static.openbankproject.com/images/sandbox/bank_x.png", "website": "https://www.example.com", "bank_routing": {"scheme": "OBP", "address": "at02-bank-x--01"}, "version": 1}, {"id": "at02-bank-y--01", "short_name": "Bank Y", "full_name": "The Bank of Y", "logo": "https://static.openbankproject.com/images/sandbox/bank_y.png", "website": "https://www.example.com", "bank_routing": {"scheme": "OBP", "address": "at02-bank-y--01"}, "version": 1}, {"id": "at02-2008..01", "short_name": "Abanca", "full_name": "ABANCA CORPORACION BANCARIA, S.A.", "logo": "http://poo.gl/NS7wIU", "website": "WWW.ABANCA.COM/ES", "bank_routing": {"scheme": "OBP", "address": "at02-2008..01"}, "version": 1}

```

Now that you have shown the attendees how to build the full integration you can test the integration just built, using this online tool <https://apitester.com/>. For attendees, you can also alternatively share the following URL for them to use <https://open-data.b9ad.pro-us-east-1.openshiftapps.com/open-data/banks>

Explain the role of the API tester as mimicking a full Web or Mobile application but stripped down of the GUI.

Build your test

View example

Click to add or remove steps

Request no auth financial service

GET https://open-data.b9ad.pro-us-east-1.openshiftapps.com/open-data/banks

Headers + Add Request Header

+ Add Step

Test Save to Account Share Test Config

T Saving tests to your account allows you to:

- Rerun this test
- Share test results with others
- View the run history
- Create and re-run multiple tests

© 2019 RIGOR. ALL RIGHTS RESERVED

Powered by RIGOR

Message [Step 1] no auth Financial service passed

Connection Information >

Request

Request Headers

```
GET /open-data/banks HTTP/1.1
Host: open-data.b9ad.pro-us-east-1.openshiftapps.com
Accept: */*
User-Agent: Mozilla/5.0 (compatible; Rigor/1.0.0; http://rigor.com)
```

Response

Response Headers

```
HTTP/1.1 200 OK
Server: openresty/1.13.6.2
Date: Fri, 11 Jan 2019 09:40:08 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 5070
Set-Cookie: JSESSIONID=jd2hwsnk3h9en1f562s4q1fh; Path=/
Expires: Fri, 11 Jan 2019 09:40:08 GMT
Access-Control-Allow-Origin: *
Cache-Control: no-cache, private, no-store
Correlation-Id: p2hwmk3h9en1f562s4q1fh
Pragma: no-cache
X-Frame-Options: DENY
Strict-Transport-Security: max-age=31536000; includeSubdomains
X-Frags-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self'; script-src 'self' 'unsafe-inline' 'unsafe-eval'; img-src 'self' https://static.openbankproject.org
Referrer-Policy: no-referrer-when-downgrade
Set-Cookie: 4e0f3rdf5272673fd1ac68bd37832b37=020029bc3fa9aa2bdf56883d95b5e541; path=/; HttpOnly; Secure
```

Response Body

```
{"banks": [{"id": "psd201-bank-x--uk", "short_name": "Bank X", "full_name": "The Bank of X", "logo": "https://static.openbankproject.com/images/sandboxes/banks/psd201-bank-x--uk/logo.png"}]}
```

Show that everything went 200 fine and open the Response Body (eye icon)

```

v {
  "banks": [
    {
      "id": "psd2b1-bank-x--uk",
      "short_name": "Bank X",
      "full_name": "The Bank of X",
      "logo": "https://static.openbankproject.com/images/sandbox/bank_x.png",
      "website": "https://www.example.com",
      "bank_routing": [
        {
          "scheme": "OBP",
          "address": "psd2b1-bank-x--uk"
        }
      ],
      "id": "psd2b1-bank-y--uk",
      "short_name": "Bank Y",
      "full_name": "The Bank of Y",
      "logo": "https://static.openbankproject.com/images/sandbox/bank_y.png",
      "website": "https://www.example.com",
      "bank_routing": [
        {
          "scheme": "OBP",
          "address": "psd2b1-bank-y--uk"
        }
      ],
      "id": "at02-bank-x--01",
      "short_name": "Bank X",
      "full_name": "The Bank of X",
      "logo": "https://static.openbankproject.com/images/sandbox/bank_x.png",
      "website": "https://www.example.com",
      "bank_routing": [
        {
          "scheme": "OBP",
          "address": "at02-bank-x--01"
        }
      ],
      "id": "at02-bank-y--01",
      "short_name": "Bank Y",
      "full_name": "The Bank of Y",
      "logo": "https://static.openbankproject.com/images/sandbox/bank_y.png",
      "website": "https://www.example.com",
      "bank_routing": [
        {
          "scheme": "OBP",
          "address": "at02-bank-y--01"
        }
      ],
      "id": "at02-2080--01",
      "short_name": "Abanca",
      "full_name": "ABANCA CORPORACION BANCARIA, S.A.",
      "logo": "https://www.abanca.es/logo"
    }
  ]
}

```

The screenshot shows a JSON response structure for bank data. It includes arrays for 'banks' containing individual bank objects. Each bank object has properties like 'id', 'short_name', 'full_name', 'logo', 'website', and 'bank_routing'. The 'bank_routing' array contains objects with 'scheme' (set to 'OBP') and 'address'.

Show the structure of the JSON response, with basic information given around banks (dummy data).

3scale

To reach 3scale as a user you can just refer back to the main integr8ly dashboard and click on 3scale.

Introducing now 3scale, for the purpose of managing these exposed APIs, securing them and tracking usage. **Dashboard**

The screenshot shows the 3scale dashboard interface. At the top, there are navigation links: Documentation, Dashboard (which is active), Developers, Applications, Billing, Analytics, API, Developer Portal, and Settings. The main area is divided into two sections: DEVELOPERS and API.

DEVELOPERS: This section displays metrics for signups and potential upgrades.

- 1 Signups** (last 30 days)
- 0 Potential Upgrades** (today)
- Information about accounts hitting usage limits and the need for Web Alerts for Admins.

API: This section displays metrics for hits and top applications.

- 0 Hits** (last 30 days)
- 0 Top Applications** (today)
- Information about top applications and how to make test calls.

Explain that you are now logged in as an administrator or API provider. The dashboard will

show a summary of the trends in usage of the platform, in terms of developers signups, usage of APIs and message sent and received. API

The screenshot shows the 3scale API management platform interface. The top navigation bar includes links for Applications, Places, Google Chrome, Documentation, and a list of open tabs. The main menu has options for Dashboard, Developers, Applications, Billing, Analytics, API, Developer Portal, and Settings. Below the menu, there are two sections: 'API Definition, Integration and Settings' and 'Published Application Plans'. The 'API' section contains fields for 'Private Base URL', 'Staging Box', 'Production Box', and 'Effects'. It also lists permissions: 'Authenticated by API key', 'ID for API calls is 2 and system name is api', 'Users can manage application keys', 'Users can manage applications', 'Users can request plan change', and 'Users cannot select a plan when creating an application'. The 'Analytics' section shows a chart with 'Hits' (0 hits) over time. The 'Latest alerts' section indicates 'There are no alerts.' The 'Published Application Plans' section shows 'Basic - 1 application' and 'Unlimited - 0 applications'. A note at the bottom says 'You have 2 application plans (2 published) with a total of 1 live application.'

You explain that by default there is an API created and on this screen you are able to see a summary of the APIs, in terms of configuration, API consumer contracts (or Application Plans) and usage (reporting and created Application on top of this API).

We will now delete as API provider the default application since we are going to change the API configuration. Applications

The screenshot shows the 3scale API management platform interface. The top navigation bar includes links for Applications, Places, Google Chrome, Documentation, and a list of open tabs. The main menu has options for Dashboard, Developers, Applications, Billing, Analytics, API, Developer Portal, and Settings. Below the menu, there is a section titled 'Applications' with a table. The table columns are: Name, State, Account, Plan, Paid?, Created At, and Traffic On. There is one entry: 'Developer's App' with 'live' state, 'Developer' account, 'Basic' plan, 'free' paid status, and 'January 09, 2019' created date. A 'Search' button is located at the bottom right of the table. A link 'Export all Applications' is also present.

This screenshot is identical to the one above, showing the 'Applications' management page with the same table and data.

We will now inform the audience that part of the platform also covers Developers and Application management. Several approval workflow and rules can be configured for these 2 modules. We will now click on **Developer's App**

The screenshot shows the 3scale developer application management interface. The top navigation bar includes links for Dashboard, Developers, Applications, Billing, Analytics, API, Developer Portal, and Settings. The current page is 'Applications > Application 'Developer's App' > Analytics'. The main content area is titled 'Developer's App' and contains several sections:

- Description:** Default application created on signup.
- Service:** API
- State:** Live suspend
- API Credentials:** User Key (13c113b578be9d43b296a566d1e24616), with options to Regenerate or Set Custom Key.
- Application Plan: Basic** (Features: Unlimited Greetings (green checkmark), 24/7 support (red X), Unlimited calls (red X), Customize).
- Change Plan** button.
- Usage in last 30 Days:** Hits (0 hits).
- Current Utilization:** This is an unmetered application, there are no limits defined.

The bottom of the screen shows a toolbar with various application icons and the text '1 / 2'.

We can see the details of the predefined application, we can see that its use has been approved and that we can also regenerate key in case we think it's been compromised. We will now proceed on deleting it. Click on **Edit**

The screenshot shows the 'Edit application: Developer's App' form. The top navigation bar includes links for Documentation, Dashboard, Developers, Applications, Billing, Analytics, API, Developer Portal, and Settings. The current page is 'Applications - Edit | Red Hat'. The form fields are:

- Name***: Developer's App
- Description***: Default application created on signup.

At the bottom of the form is a blue 'Update Application' button. Below the form, there is a red 'Delete' link. The bottom of the screen shows a toolbar with various application icons and the text 'Version 2.2.0 - Powered by 3scale'.

Delete

The application was successfully deleted.



We can now proceed on changing the configuration of the API and publish the update on the Developer Portal so that the public Developers can sign up for the open financial API.

API-> Integration

To get started with this service on APICAST, add the base URL of your API and save the configuration.



This is where you configure the details of the mapped Service. add the base URL ...

Explain the structure of the configuration. We have the section where we map the Backend API and then 2 URLs where we expose the managed API on the staging first and production gateways or infrastructure. We are going to change the private base URL with the one from the public service before (for attendees this will be <https://open-data.b9ad.pro-us-east-1.openshiftapps.com/open-data/banks>). We will also be changing the Staging and Public address of the gateway. In this case we are not going to use separate staging or public infrastructure so it can be the same address. (please notice the format of the staging and public base url for the gateways <https://wt2-evalsX-example-com-3scale.apps.open-banking.opentry.me>)

We will now make sure we map a single endpoint or resource in 3scale and disallow any other endpoint (i.e. the other endpoints have not been implemented yet).

The screenshot shows the 3scale API configuration interface. At the top, there are fields for 'Staging Public Base URL' and 'Production Public Base URL'. Below these are sections for 'MAPPING RULES' and 'AUTHENTICATION SETTINGS'. In the 'MAPPING RULES' section, a rule is defined for 'GET /open-data/banks\$' with a metric of 'hits'. In the 'AUTHENTICATION SETTINGS' section, 'Secret Token' is set to 'Shared_secret_sent_from_proxy_to_API_backend_bb31809e8926e89a'. The 'CREDENTIALS LOCATION' section shows 'As HTTP Headers' selected. The 'AUTHENTICATION FAILED ERROR' section includes fields for 'Auth user key' ('api-key'), 'Response Code*' ('403'), 'Content-type' ('text/plain; charset=utf-8'), and 'Response Body' ('Authentication failed'). The 'AUTHENTICATION MISSING ERROR' section includes fields for 'Response Code*' ('403'), 'Content-type' ('text/plain; charset=utf-8'), and 'Response Body' ('Authentication parameters missing').

(notice the \$ at the end of the path). We can now check the configuration of authentication settings.

This screenshot shows the same 3scale configuration interface as above, but with different values. In the 'AUTHENTICATION SETTINGS' section, 'Host Header' is empty. 'Secret Token' is set to 'Shared_secret_sent_from_proxy_to_API_backend_bb31809e8926e89a'. 'CREDENTIALS LOCATION' is set to 'As query parameters (GET) or body parameters (POST/PUT/DELETE)'. The 'AUTHENTICATION FAILED ERROR' section has 'Auth user key' ('user_key'), 'Response Code*' ('403'), 'Content-type' ('text/plain; charset=utf-8'), and 'Response Body' ('Authentication failed'). The 'AUTHENTICATION MISSING ERROR' section has 'Response Code*' ('403'), 'Content-type' ('text/plain; charset=utf-8'), and 'Response Body' ('Authentication parameters missing').

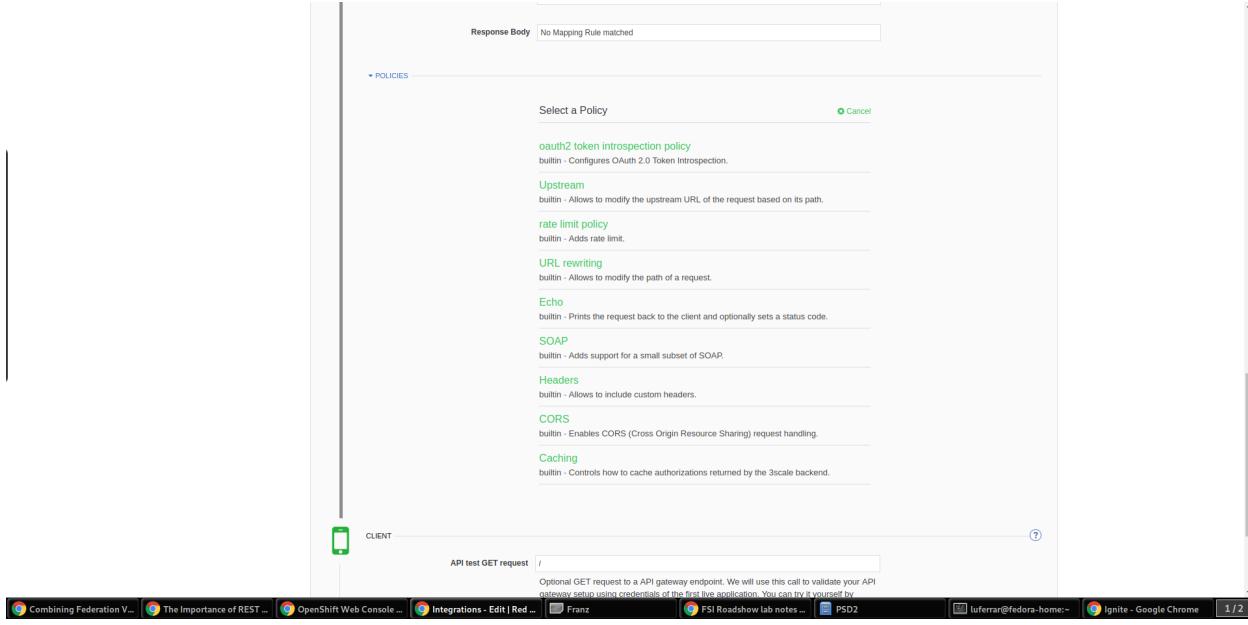
We see that we have already api key protection enabled, but we might want to pass this information as HTTP Header instead of HTTP parameter. We will also change the header name to 'key'

The screenshot shows the configuration interface for error handling in the 3scale API Manager. It includes sections for 'Host header', 'Secret Token' (with a note about accepting traffic from a specific host), 'CREDENTIALS LOCATION' (set to 'As HTTP Headers'), and 'Auth user key' (set to 'key'). Below these are sections for 'AUTHENTICATION FAILED ERROR' (Response Code: 403, Content-type: text/plain; charset=us-ascii, Response Body: Authentication failed) and 'AUTHENTICATION MISSING ERROR' (Response Code: 403, Content-type: text/plain; charset=us-ascii, Response Body: Authentication parameters missing). A 'NO MATCH ERROR' section is also present. At the bottom, a navigation bar includes links like 'Combining Federation...', 'The Importance of REST...', 'Documentation - Go...', 'Integrations - Update...', 'Franz', 'FSI Roadshow lab notes...', 'PSD2', 'Ignite - Google Chrome', 'Trello', and 'Openbanking API Pr...'. The page number '1 / 2' is visible.

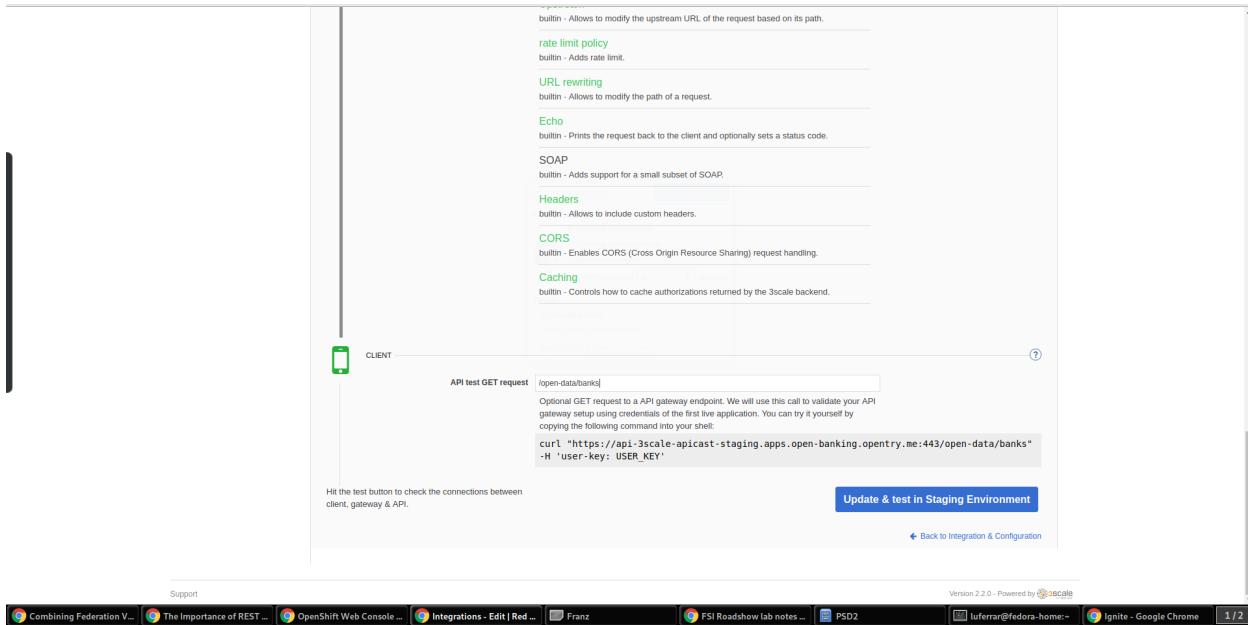
Explain that you can also customize the error returned to the end user. Policies

The screenshot shows the configuration interface for policies and client testing. It includes a 'NO MATCH ERROR' section (Response Body: Authentication parameters missing) and a 'POLICIES' section. Under 'POLICIES', there is a 'Policy Chain' with an 'Add Policy' button. A detailed view of the 'APICast' policy is shown, stating it's built-in and used for APICast to work with the 3scale API manager. Below this is a 'CLIENT' section with an 'API test GET request' input field containing a curl command to validate the API gateway setup. A note says to hit the test button to check connections between client, gateway & API. A 'Update & test in Staging Environment' button is available. At the bottom, a 'Support' link and a 'Version 2.2.0 - Powered by 3scale' note are visible. The same navigation bar as the previous screenshot is at the bottom.

These are like additional plugin that you can configure to adapt the service to your own preference. **Add Policy**



Several policies are available and the list is always longer. Highlight the presence of SOAP policy to map SOAP services and advanced rate limit functionalities, rate limit policy. Update the API test GET request field with the same pattern you are mapping above (excluding the \$).



Hitting the big blue button will allow you to 2 things at once:

- Updating the service configuration
- Testing the configuration just uploaded to the gateway.

The second one will fail since we are not providing any valid key, so we will get unauthorized

request but the gateway will receive the updated configuration in any case.

The screenshot shows the 3scale API Manager's 'Test Requests' section. A red vertical line highlights the 'Response Body' field, which contains the error message 'Authentication parameters missing'. Below this, the 'NO MATCH ERROR' section shows a 'Response Code' of 404 and a 'Content-type' of text/plain; charset=us-ascii. The 'Response Body' field here says 'No Mapping Rule matched'. Under the 'POLICIES' section, there is a 'Policy Chain' with an entry for 'APICast' (builtin). At the bottom, there is an 'API test GET request' input field containing '/open-data/banks', a note about optional GET requests for validation, and a 'Test request failed with HTTP code 403: Authentication failed' message with a curl command. A button labeled 'Update & test in Staging Environment' is present. The footer includes a navigation bar with links like 'Combining Federated...', 'The Importance of RE...', 'Documentation - Go...', 'Integrations - Update...', 'Franz', 'FSI Roadshow lab no...', 'PSD2', 'Ignite - Google Ch...', 'Trello', and 'Openbanking API Pr...', and a page number '1 / 2'.

We will now fix the test request error.

Let's switch to explaining the role of API contracts of Application Plans.

The screenshot shows the 'Application Plans' section of the 3scale API Manager. On the left, a sidebar lists 'Definition', 'Integration', and 'Application Plans'. The main area shows a table for 'Default Plan' with two rows: 'Basic' and 'Unlimited'. Both rows have 0 applications and are in a 'published' state. There are 'Hide', 'Copy', and 'Delete' actions for each row. A 'Create Application Plan' button is at the top right of the table. The footer includes a navigation bar with links like 'Combining Federated...', 'The Importance of RE...', 'Documentation - Go...', 'Application plans ...', 'Franz', 'FSI Roadshow lab no...', 'PSD2', 'luferran@fedora-h...', 'Ignite - Google Ch...', 'Trello', and 'Openbanking API Pr...', and a page number '1 / 2'.

We see that we have 2 already configured API contracts (or Application Plans), but no application associated to it. Both application plans are in **published** state, so they are both usable and visible on the Developer portal. Let's open the **Basic** plan

The screenshot shows the 3scale API Management platform's 'Application Plan Basic' configuration page. The left sidebar has tabs for Overview, ActiveDocs, Definition, Integration, Application Plans (selected), Settings, and Alerts. The main content area has tabs for 'API > Application Plan Basic'. It contains fields for 'Name*' (Basic), 'System name*' (basic), a checkbox for 'Applications require approval' (unchecked), 'Trial Period (days)' (empty), 'Setup fee' (0.00 USD), and 'Cost per month' (0.00 USD). Below these are sections for 'Metrics, Methods, Limits & Pricing Rules' (with a table for Hits) and 'Features' (with a table for Unlimited Greetings). A blue 'Update Application plan' button is at the bottom right.

Main elements:

- Monetization settings (trial, setup, cost per month)
- Endpoint mapped (in this case generic Hits) and relative monetization and rate limiting settings

We can now switch to the Developers tab to create an Application to test the Configuration.

The screenshot shows the 3scale API Management platform's 'Developers' tab. The left sidebar has tabs for Documentation, Accounts (selected), and Messages. The main content area shows a table of accounts with columns for Group/Org, Admin, Signup Date, Apps, State, and a 'Create' button. One account, 'Developer' (John Doe), is highlighted with a modal window showing its details: 'John Doe', '9 Jan, 2019', '0 Apps', and 'Approved'. The modal also has tabs for 'Details' and 'Effects'. A blue 'Export all Accounts' button is at the bottom right.

From here we can see how we can, as Provider, approve or deny Developers' Accounts registrations. Let's click on the default **Developer**

The screenshot shows the 3scale account dashboard for a developer named 'Developer'. At the top, there's a navigation bar with links for Documentation, Dashboard, Developers (which is active), Applications, Billing, Analytics, API, Developer Portal, and Settings. Below the navigation is a breadcrumb trail: Accounts > Account 'Developer' > 0 Applications | 1 User | 0 Invitations | 0 Group Memberships | 0 Invoices | 1 Service Subscription. The main content area is titled 'Developer: Account Summary' and contains sections for 'Organization/Group Name' (Developer, Approved), 'Administrator' (John Doe, admin+test@3scale.apps.open-banking.openty.me), 'Signed up on' (January 09, 2019 11:57), 'Custom Account Plan' (Edit, Remove customization), and 'Billing Status' (Monthly billing is enabled, Disable). A 'Send message' and 'Edit' button are also present. At the bottom, there's a support link and a note about version 2.2.0.

<https://3scale-admin.apps.open-banking.openty.me/accounts/3>

We can see that the Developer has no associated application, but it's subscribed to one Service.
We can also see the Developer user details.

Let's click on **0 Applications** and **Create application**

The screenshot shows the 'Applications' page for the developer. The top navigation bar includes a 'Create Application' button. The main content area lists columns for Name, State, Plan, Paid?, Created At, and Traffic On. A note below states 'No applications'. At the bottom, there's a support link and a note about version 2.2.0.

<https://3scale-admin.apps.open-banking.openty.me/buyers/accounts/3/applications?direct...>

The screenshot shows the 3scale platform's application creation interface. At the top, there's a navigation bar with links for Documentation, Dashboard, Developers (which is active), Applications, Billing, Analytics, API, Developer Portal, and Settings. Below the navigation is a breadcrumb trail: Accounts > Account 'Developer' > 0 Applications | 1 User | 0 Invitations | 0 Group Memberships | 0 Invoices | 1 Service Subscription. The main form is titled 'New Application'. It has fields for 'Application plan*' (set to 'Basic'), 'Service plan' (set to 'Default'), 'Name*' (empty), and 'Description*' (empty). At the bottom right of the form is a blue 'Create Application' button.

Let's fill in with some basic details and click the big blue button

The screenshot shows the 3scale platform's analytics view for an application named 'my first finance app'. The top navigation bar includes links for Dashboard, Developers, Applications (active), Billing, Analytics, API, Developer Portal, and Settings. The breadcrumb trail indicates the user is viewing the analytics for their first application. The main content area contains several sections: 'my first finance app' (with an 'Edit' link), 'API Credentials' (User Key: 4c571db87ab2b5aeadccbd8877627a090, with 'Regenerate' and 'Set Custom Key' buttons), 'Usage in last 30 Days' (Hits: 0 hits), and 'Current Utilization' (Note: This is an unmetered application, there are no limits defined). On the right side, there's a 'Change Plan' section for the 'Application Plan: Basic' (Features: Unlimited Greetings (green checkmark), 24/7 support (red X), Unlimited calls (red X), and a 'Customize' button). A '1 / 2' page indicator is at the bottom right.

We now have an assigned key so we can go back to the Configuration window of the API and make a successful test call. **API -> Integration -> edit Apicast configuration**

The screenshot shows the 3scale API Manager interface. At the top, there is a message: "Response Body Authentication parameters missing". Below it, under "NO MATCH ERROR", there are fields for "Response Code*" (set to 404) and "Content-type" (set to text/plain; charset=us-ascii). A "Response Body" field contains the message "No Mapping Rule matched". On the left, a vertical bar labeled "CLIENT" has a red line segment above it. Under "POLICIES", there is a section for "APIcast" with the note "builtin - Main functionality of APIcast to work with the 3scale API manager." At the bottom, there is a "Test" section for an "API test GET request" to "/open-data/banks". It includes a curl command: `curl "https://wt2-evals98-example-com-3scale.apps.open-banking.opentry.me:443/open-data/banks" -H 'key: 4c571db87a02b5aedccbd8077627a090'`. A button "Update & test in Staging Environment" is present. The footer includes a "Support" menu and links to various documentation and tools.

We now have a pre-populated key, let's try again testing the deployed configuration.

This screenshot is identical to the one above, except the "Test" section now shows a green line segment above the "CLIENT" label, indicating a successful connection. The message "Connection between client, gateway & API is working correctly as reflected in the analytics section." is displayed below the curl command. The rest of the interface, including policies and the footer, remains the same.

As we can see we turned the testing into a success.

Let's switch to the developers' point of view by accessing the Developer portal. **Developer portal -> Visit Developer Portal**

The sidebar allows us to edit pages of the Developer Portal live, but we are not interested into it so we can close it.

The screenshot shows the homepage of the Echo API developer portal. At the top, there's a navigation bar with links for 'PROVIDER NAME', 'DOCUMENTATION', 'PLANS', 'SIGN IN', and a 'DRAFT | PUBLISHED' switch. Below the header, the main title 'Echo API' is prominently displayed. Underneath the title, there are three main calls-to-action: 'Register' (with a 'Sign up' button), 'Get your API key' (with a magnifying glass icon), and 'Create your app' (with a code editor icon). A section titled 'Pick your plan' follows, showing two plan options: 'BASIC' and 'UNLIMITED'. Both plans are described as having 'Unlimited Greetings', '24/7 support', and 'Unlimited calls', with 'No limits' indicated for the number of calls. Each plan has a 'Signup to plan [Plan Type]' button. The bottom of the page features a standard browser toolbar with various tabs and icons.

Let's sign in with the default user credentials provided in the sidebar. john / 123456

The screenshot shows the developer portal after signing in. The top navigation bar now includes the user's name 'SIGNED IN SUCCESSFULLY'. The main title 'Echo API' is centered. Below it, a 'Your API Key' section displays the user's API key, with a note: 'This is your API key that should be kept secret. Use it to authenticate and report the calls you make to the Echo API.' A call-to-action button 'See your Applications & their credentials' is shown. Further down, a 'Run your requests' section contains a terminal command: '\$ curl -v https://echo-api.3scale.net'. Below this, a 'REQUEST' section shows the curl command, and a 'RESPONSE' section shows the resulting JSON output. The bottom of the page features a standard browser toolbar with various tabs and icons.

We are now logged in the developer's dashboard. Let's see the Applications I have

The screenshot shows a provider management interface. At the top, there are tabs for 'PROVIDER NAME', 'APPLICATIONS', 'STATISTICS', and 'DOCUMENTATION'. Below these are 'MESSAGES', 'SETTINGS', and a refresh icon. A central table displays one application entry:

NAME	CREDENTIALS	STATE
my_first_finance_app	4c571db87a82b5aedccbd8877627a090	live

A blue button labeled 'Create new application' is located at the bottom right of the table area. The background of the page is light blue.

I can now use the credential that I have associated with the application and test the protected service.

The screenshot shows an API testing tool interface. At the top, there is a search bar with 'key' and a value '4c571db87a82b5aedccbd8877627a090'. Below the search bar are buttons for 'Test', 'Save to Account', and 'Share Test Config'.

The main area has a green header bar with a 'PASS' status and 'N. Virginia' and 'Seconds elapsed: 2' information. Below this, there are sections for 'Results 576 ms', 'Message' (containing '[Step 1] key financial service passed'), 'Connection Information', 'Request' (with 'Request Headers' showing a GET request to /open-data/banks), and 'Response' (showing the HTTP response headers and body). The response body includes:

```
HTTP/1.1 200 OK
Server: openresty/1.13.6.1
Date: Fri, 11 Jan 2019 18:04:18 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 5070
Set-Cookie: JSESSIONID=y0ravw644k1us8lkk5f1uy;Path=/
Expires: Fri, 11 Jan 2019 18:04:18 GMT
Access-Control-Allow-Origin: *
```

The bottom of the interface shows a toolbar with various icons and a status bar indicating '1 / 2'.

Notice the URL, the key Header and related value
Let's just test with a wrong key or path

PASS

Results 111 ms

Message
[Step 1] key financial service passed

Connection Information

Request

Request Headers

```
GET /open-data/banks HTTP/1.1
Host: wt2-evals98-example-com-3scale.apps.open-banking.opentry.me
Accept: */*
User-Agent: Mozilla/5.0 (compatible; Rigor/1.0.0; http://rigor.com)
key: c571db87a82b5aedccbd8877627a099
```

Response

Response Headers

```
HTTP/1.1 403 Forbidden
Server: openentry/1.13.6.1
Date: Fri, 11 Jan 2019 18:16:42 GMT
Content-Type: text/plain; charset=us-ascii
Transfer-Encoding: chunked
Set-Cookie: 6ec552533d9895c2bc361d784adcd8db:fd85f9e6e21ee8a00f6cad673c7c9e5b; path=/; HttpOnly; Secure
```

Response Body

As expected we receive a Forbidden error.

Checkpoint

RH SSO and 3SCALE OIDC

After introducing content around OAuth and RH SSO, let's take a quick walk through RH SSO itself.

SINCE IN INTEGR8LY USERS DON'T HAVE ACCESS TO THE RELATED RH SSO REALM, YOU ARE GOING TO SHOW HOW TO CONFIGURE A RH SSO CLIENT THAT WILL THEN BE USED BY ALL THE OTHER USERS IN THEIR 3SCALE CONFIGURATION. YOU WILL NEED TO GET THE USER AND PASSWORD FROM THE ENVIRONMENT VARIABLES OF SSO TO ACCESS TO THE ADMIN CONSOLE OF RH SSO. YOU WILL NEED TO LOGIN TO OPENSHIFT AS ADMIN

Let's start by RH SSO main dashboard (logging in as admin).

<https://secure-sso-sso.apps.open-banking.opentry.me/auth/admin/master/console/#/realms/openshift/token-settings>

The realms are like separate instances of the platform, dedicated to separating users and applications. As we can see we can customize several aspect of the realm like the theme of the login page or the tokens default parameters. **Endpoints -> OpenID Endpoint Configuration**

```

{
  "issuer": "https://secure-sso-sso.apps.open-banking.opentry.me/auth/realms/openshift",
  "authorization_endpoint": "https://secure-sso-sso.apps.open-banking.opentry.me/auth/realms/openshift/protocol/openid-connect/auth",
  "token_endpoint": "https://secure-sso-sso.apps.open-banking.opentry.me/auth/realms/openshift/protocol/openid-connect/token",
  "token_introspection_endpoint": "https://secure-sso-sso.apps.open-banking.opentry.me/auth/realms/openshift/protocol/openid-connect/token/introspect",
  "end_session_endpoint": "https://secure-sso-sso.apps.open-banking.opentry.me/auth/realms/openshift/protocol/openid-connect/logout",
  "jwks_uri": "https://secure-sso-sso.apps.open-banking.opentry.me/auth/realms/openshift/protocol/openid-connect/certs",
  "check_session_iframe": "https://secure-sso-sso.apps.open-banking.opentry.me/auth/realms/openshift/protocol/openid-connect/login-status-iframe.html",
  "grant_types_supported": [
    "authorization_code",
    "implicit",
    "refresh_token",
    "password",
    "client_credentials"
  ],
  "response_types_supported": [
    "code",
    "none",
    "id_token",
    "token",
    "id_token token",
    "code id_token",
    "code token",
    "code id_token token"
  ],
  "subject_types_supported": [
    "public",
    "pairwise"
  ],
  "id_token_signing_alg_values_supported": [
    "RS256"
  ],
  "userinfo_signing_alg_values_supported": [
    "RS256"
  ],
  "request_object_signing_alg_values_supported": [
    "none",
    "RS256"
  ],
  "response_modes_supported": [
    "query",
    "fragment",
    "form_post"
  ],
  "registration_endpoint": "https://secure-sso-sso.apps.open-banking.opentry.me/auth/realms/openshift/clients-registrations/openid-connect",
  "token_endpoint_auth_methods_supported": [
    "private_key_jwt",
    "client_secret_basic",
    "client_secret_post"
  ]
}

```

This is where we can find the public endpoints of the APIs exposed by RH SSO. Let's now take a look at the Clients section.

RED HAT SINGLE SIGN-ON

Admin

OpenShift

Configure

- Realm Settings
- Clients**
- Client Templates
- Roles
- Identity Providers
- User Federation
- Authentication

Manage

- Groups
- Users
- Sessions
- Events
- Import
- Export

Clients ?

Client ID	Enabled	Base URL	Actions
3scale	True	Not defined	Edit Export Delete
account	True	/auth/realm/openshift/account	Edit Export Delete
admin-cli	True	Not defined	Edit Export Delete
broker	True	Not defined	Edit Export Delete
faucner-openshift-users	True	Not defined	Edit Export Delete
openshift-client	True	Not defined	Edit Export Delete
realm-management	True	Not defined	Edit Export Delete
security-admin-console	True	/auth/admin/openshift/console/index.html	Edit Export Delete

Franz Combinig Federatio... The Importance of RE... Documentation - Go... RH-SSO Admin Cons... FSI Roadshow lab no... Roadshow lu ferr@fedora-ho... *Untitled Document ... Pictures 1 / 12

Here we can configure the applications that will authenticate using RH SSO as IDP. As we can see there are some default clients dedicated to authentication in the integr8ly environment. Let's now walk through the other main modules. **Identity Providers**

RED HAT SINGLE SIGN-ON

Admin

Openshift

Configure

- Realm Settings
- Clients
- Client Templates
- Roles
- Identity Providers**
- User Federation
- Authentication

Manage

- Groups
- Users
- Sessions
- Events
- Import
- Export

Identity Providers

Through identity Brokering it's easy to allow users to authenticate to Keycloak using external Identity Providers or Social Networks.

We have built-in support for OpenID Connect and SAML 2.0 as well as a number of social networks such as Google, GitHub, Facebook and Twitter.

To get started select a provider from the dropdown below:

Add provider...

- User-defined**
- SAML 2.0
- OpenID Connect v1.0
- Keycloak OpenID Connect
- Social**
- GitHub
- Twitter
- Facebook
- OpenShift v3
- Google
- GitLab
- LinkedIn
- Microsoft
- BitBucket
- PayPal
- StackOverflow

Here we can see that RH SSO can also act as an Identity Broker, allowing our end users to authenticate with the application by leveraging standard identity protocols, like OpenID Connect or SAML2.0 . It is also possible to re-use some templated connectors towards social networks or services. **User federation**

RED HAT SINGLE SIGN-ON

Admin

Openshift

- Configure
 - Realm Settings
 - Clients
 - Client Templates
 - Roles
 - Identity Providers
 - User Federation**
 - Authentication
- Manage
 - Groups
 - Users
 - Sessions
 - Events
 - Import
 - Export

User Federation

Keycloak can federate external user databases. Out of the box we have support for LDAP and Active Directory.

To get started select a provider from the dropdown below:

Add provider...

kerberos

ldap

In user federation we can federate external user base with RH SSO and cache the user base locally. Supported protocols include LDAP and Active Directory. **Users -> View all users**

RED HAT SINGLE SIGN-ON

Admin

Openshift

- Configure
 - Realm Settings
 - Clients
 - Client Templates
 - Roles
 - Identity Providers
 - User Federation
 - Authentication
- Manage
 - Groups
 - Users**
 - Sessions
 - Events
 - Import
 - Export

Users

Lookup

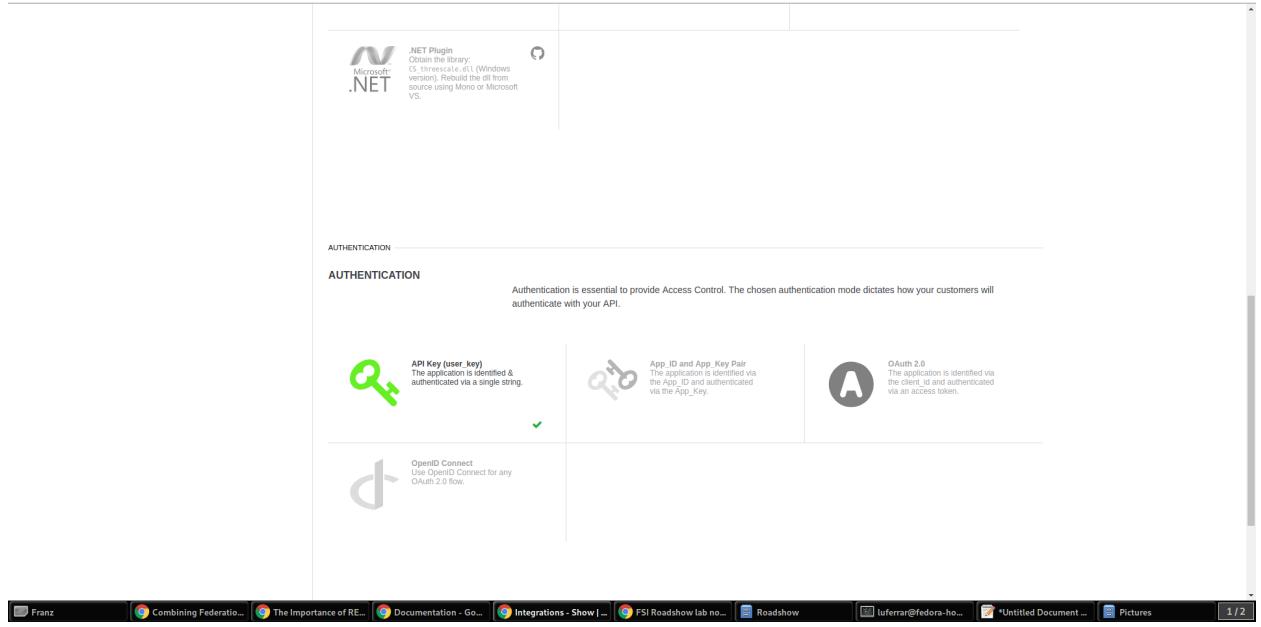
ID	Username	Email	Last Name	First Name	Actions
0eb01f1ae-1292-4ca9-83d5-c15fc5ba...	admin@example.com	admin@example.com			Edit Impersonate Delete
1e060595-2c4c-46a4-b16f-e11940bb...	eval01@example.com	eval01@example.com			Edit Impersonate Delete
31dc7463-c73-48fe-9f19-41191b63...	eval02@example.com	eval02@example.com			Edit Impersonate Delete
5fb5c550-69a5-41b9-9537-812128c7...	eval03@example.com	eval03@example.com			Edit Impersonate Delete
036eafab-91f5-a6f9-8c8b-6558574f...	eval04@example.com	eval04@example.com			Edit Impersonate Delete
6ed01bed-1fd6-4c26-a7fe-4592b2d...	eval05@example.com	eval05@example.com			Edit Impersonate Delete
142c3744-1b00-4072-a55b-a29b20b...	eval06@example.com	eval06@example.com			Edit Impersonate Delete
77756e4-6c8d-4ade-9a88-2b6d382...	eval07@example.com	eval07@example.com			Edit Impersonate Delete
97b191c8-1860-4b4b-81bb-2ea3799...	eval08@example.com	eval08@example.com			Edit Impersonate Delete
b3270190-e0b3-472d-9f16-2cfcd0f...	eval09@example.com	eval09@example.com			Edit Impersonate Delete
8c7a6bd5-1771-4b2e-b15a-4881e7b...	eval10@example.com	eval10@example.com			Edit Impersonate Delete
ab4a0d04-4059-479b-b758-7a572b...	eval10@example.com	eval10@example.com			Edit Impersonate Delete
c7719640-2d43-409c-bc7d-23725df...	eval11@example.com	eval11@example.com			Edit Impersonate Delete
e9489dfb-4c3-470e-a04-342618b...	eval12@example.com	eval12@example.com			Edit Impersonate Delete
fd547255-546d-4506-bc51-364348f...	eval13@example.com	eval13@example.com			Edit Impersonate Delete
b42903df-a049-4847-9eb6-1967c4c...	eval14@example.com	eval14@example.com			Edit Impersonate Delete
58814183-0c84-4aff-9268-979b0f83...	eval15@example.com	eval15@example.com			Edit Impersonate Delete
9cae975b-6aff-4564-844-97a34117...	eval16@example.com	eval16@example.com			Edit Impersonate Delete
3326158c-342-47f1-9292-12eeb07...	eval17@example.com	eval17@example.com			Edit Impersonate Delete
37dfc701-1b03-4499-9953-03b2249...	eval18@example.com	eval18@example.com			Edit Impersonate Delete

First Page Previous Page Next Page

Here we can see all the user that are stored inside RH SSO, making it act as an IDM as well. These are the end users of the applications created in Clients. Let's explore one of these users.

We can see here the type of information stored along with basic user details. The user profile can be customized with additional attributes as well. Let's now switch back 3scale.

We can see that we have a fully configured API with API key as the Authentication method. We are going to change it to the more secure OpenID Connect, to ensure our financial data are protected from attacks performed when key is compromised. **Edit integration settings**



We are going to change it to OpenID Connect. Update service

https://3scale-admin.apps.open-banking.opentry.me/api/config/services/2/integration

3scale-admin.apps.open-banking.opentry.me says
It is not advisable to change the authentication mode
when you are serving customers in production. If you do
proceed, you will need to update your APIcast
configuration to reflect this change. Do you really want to
change the authentication mode?

AUTHENTICATION

API Key (user, key)
The application is identified &
authenticated via a single string.

App ID and App_ Key Pair
The application is identified via
the App_ID and authenticated
via the App_Key.

OAuth 2.0
The application is identified via
the client_id and authenticated
via an access token.

OpenID Connect
Use OpenID Connect for any
OAuth 2.0 flow.

Update Service

Clearly the platform is warning us that we have customer using this API and it might break their application, changing the authentication method. In a real world case, we would inform the developer in advance by using the messaging and notification functionality available within the platform.

We have now changed the authentication method, we are just left with configuring the correct IdP inside 3scale to make sure it is authenticating the requests with RH SSO. **edit apicast configuration**

As we see we have a dedicated field for this. We will take advantage of one of the features available in OIDC and not in OAUTH which is dynamic client registration. Normally to make sure an API application authenticates with RH SSO, we would need to create manually the application on both platforms. With this feature, we let 3scale sync the application list with RH

SSO, beside obviously authenticating our API calls. Let's create a special type of such Client in RH SSO. Clients -> Create

RED HAT SINGLE SIGN-ON

Clients > Add Client

Add Client

Import

Client ID * sync-app

Client Protocol * openid-connect

Client Template * public

Root URL

Let's call it sync-app and configure the other details required to let it communicate with 3scale.

RED HAT SINGLE SIGN-ON

Clients > sync-app

Sync-app

Settings Roles Mappers Scope Revocation Sessions Offline Access Installation

Client ID * sync-app

Name

Description

Enabled

Consent Required

Client Protocol * openid-connect

Client Template

Access Type * public

Standard Flow Enabled

Implicit Flow Enabled

Direct Access Grants Enabled

Root URL

Valid Redirect URIs +

Base URL

Admin URL

Web Origins +

> Fine Grain OpenID Connect Configuration

Franz Combinig Federat... The Importance of RE... Documentation - Go... RH-SSO Admin Cons... FSI Roadshow lab no... Roadshow lufer@fedora-ho... *Untitled Document... Pictures 1 / 2

We are going to give it the rights to create applications on behalf of 3scale.

Sync-app

Settings Roles Mappers Scope Revocation Sessions Offline Access Installation

Client ID: sync-app
Name:
Description:
Enabled: ON
Consent Required: OFF
Client Protocol: openid-connect
Client Template:
Access Type: confidential
Standard Flow Enabled: OFF
Implicit Flow Enabled: OFF
Direct Access Grants Enabled: OFF
Service Accounts Enabled: ON
Root URL:
Base URL:
Admin URL:

> Fine Grain OpenID Connect Configuration
> OpenID Connect Compatibility Modes

Save **Cancel**

Save -> Service account roles

Add manage-clients to the allowed roles for this client

RED HAT SINGLE SIGN-ON

OpenShift Admin

Clients > sync-app

Sync-app

Settings Credentials Roles Mappers Scope Revocation Sessions Offline Access Clustering Installation Service Account Roles

Sync-app Service Accounts

Realm Roles Available Roles: Add selected >

Assigned Roles: offline_access uma_authorization
Effective Roles: offline_access uma_authorization

Client Roles realm-management Available Roles: create-client impersonation manage-authorization manage-events manage-identity-providers
Assigned Roles: manage-clients
Effective Roles: manage-clients

And now we are ready to use these application credentials inside 3scale IDP configuration.
Let's copy the client id, client secret and authorization endpoint.

The screenshot shows the 3scale API configuration interface. A new endpoint is being defined:

- Verb:** GET
- Pattern:** /open-data/banks\$
- Metric or Method:** hits

AUTHENTICATION SETTINGS

- OpenID Connect Issuer:** https://sync-apps-cf7fd61-8252-4078-ab2c-52aa451dd487@secure-sso-sso.apps.open-banking.o...
- Host Header:** (empty)
- Secret Token:** Shared_secret_sent_from_proxy_to_API_backend_bb31809d8926e89a

CREDENTIALS LOCATION

- As HTTP Headers
- As query parameters (GET) or body parameters (POST/PUT/DELETE)

AUTHENTICATION FAILED ERROR

- Response Code:** 403
- Content-type:** text/plain; charset=us-ascii
- Response Body:** Authentication failed

AUTHENTICATION MISSING ERROR

Let's build a url of this format:

<https://client-id:client-secret@secure-sso-sso.apps.open-banking.opentry.me/auth/realms/openshift>

And update the staging environment and promote the configuration to production

The screenshot shows the 3scale developer portal under the API > Integration & Configuration section. The APICAST configuration is displayed:

- Deployment Option:** APICAST
- Authentication:** OpenID Connect
- Private Base URL:** https://open-data.bb9ad.pro-us-east-1.openshiftapps.com:443
- Mapping rules:** /open-data/banks\$ => hits
- Credential Location:** headers
- Secret Token:** Shared_secret_sent_from_proxy_to_API_backend_bb31809d8926e89a

Environments

- Staging Environment:** https://lw2-eval99-example-com-3scale.apps.open-banking.opentry.me:443
- Production Environment:** https://api-3scale-apicast-production.apps.open-banking.opentry.me:443

Let's now switch perspective and get in the shoes of the developer and open his Applications section.

The screenshot shows the 3scale admin interface for managing applications. The application 'my first finance app - sz' is selected. The configuration page includes fields for Name, Description, Plan, Status, Client ID, Client Secret, and Redirect URL. The Client ID is listed as '5bc94f6a'. The Client Secret field is empty. The Redirect URL field contains the value 'https://openidconnect.net/callback'. A 'Submit' button is visible at the bottom.

We can see the secret of his application is absent as the redirect URL. We are going to generate the first and add as redirect url the following <https://openidconnect.net/callback> (we are going to explain it in a moment).

The screenshot shows the 3scale admin interface after the application has been updated. The message 'APPLICATION WAS SUCCESSFULLY UPDATED.' is displayed. The application 'my first finance app - sz' is selected. The configuration page includes fields for Name, Description, Plan, Status, Client ID, Client Secret, and Redirect URL. The Client ID is listed as '5bc94f6a'. The Client Secret field now contains a regenerated value: '2b6d0299110348dbae6134e97a8d0359'. The Redirect URL field now contains the correct value 'https://openidconnect.net/callback'. A 'Submit' button is visible at the bottom.

Let's make sure that is application is now aligned in terms of credentials both in 3scale and RH SSO.

https://3scale-admin.apps.open-banking.opentry.me/buyers/applications/9

Documentation

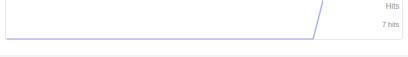
3scale BY RED HAT

Dashboard Developers Applications Billing Analytics API Developer Portal Settings

Account Developer > Application 'my first finance app - sz' > Analytics

my first finance app - sz

Edit

Description	my first finance app - sz
Service	API
State	
Live suspend	
API Credentials	
Client ID	5bc94f6a
Client Secret	2b6eb299110348dbae6134e97ab00359
Redirect URL	https://openidconnect.net/callback
Usage in last 30 Days	
 7 hits	
Current Utilization	
This is an unmetered application, there are no limits defined	

Application Plan: Basic

Features

- Unlimited Greetings ✓
- 24/7 support ✗
- Unlimited calls ✗
- Customize

Change Plan

1 / 2

https://secure-sso-sso.apps.open-banking.opentry.me/auth/admin/master/console/#/realms/openshift/clients

RED HAT SINGLE SIGN-ON

Admin

OpenShift

Configure

- Realm Settings
- Clients**
- Client Templates
- Roles
- Identity Providers
- User Federation
- Authentication

Manage

- Groups
- Users
- Sessions
- Events
- Import
- Export

Clients

Client ID	Enabled	Base URL	Actions		
3scale	True	Not defined	Edit	Export	Delete
5bc94f6a	True	Not defined	Edit	Export	Delete
account	True	/auth/realms/openshift/account	Edit	Export	Delete
admin-cli	True	Not defined	Edit	Export	Delete
broker	True	Not defined	Edit	Export	Delete
launcher-openshift-users	True	Not defined	Edit	Export	Delete
openshift-client	True	Not defined	Edit	Export	Delete
realm-management	True	Not defined	Edit	Export	Delete
security-admin-console	True	/auth/admin/openshift/console/index.html	Edit	Export	Delete
sync-app	True	Not defined	Edit	Export	Delete

Franz Combinig Federatio... The Importance of RE... Documentation - Go... RH-SSO Admin Cons... FSI Roadshow lab no... Roadshow lufer@fedora-ho... *Untitled Document ... Pictures 1 / 2

The screenshot shows the RHSSO developer portal interface. On the left, a sidebar menu includes 'Configure', 'Realm Settings', 'Clients' (selected), 'Client Templates', 'Roles', 'Identity Providers', 'User Federation', and 'Authentication'. Under 'Clients', there are links for 'Groups', 'Users', 'Sessions', 'Events', 'Import', and 'Export'. The main content area is titled '5bc9f6a' and shows the 'Settings' tab selected. The configuration details are as follows:

- Client ID:** 5bc9f6a
- Name:** my first finance app - sz
- Description:** my first finance app - sz
- Enabled:** ON
- Consent Required:** OFF
- Client Protocol:** openid-connect
- Client Template:** (empty)
- Access Type:** confidential
- Standard Flow Enabled:** ON
- Implicit Flow Enabled:** OFF
- Direct Access Grants Enabled:** OFF
- Service Accounts Enabled:** OFF
- Root URL:** (empty)
- Valid Redirect URIs:** https://openidconnect.net/callback
- Base URL:** (empty)
- Admin URL:** (empty)
- Web Origins:** (empty)

All seems good! Let's now try to authenticate the end user, using OpenID Connect.

We are going to need a special web client, a little bit more intelligent than just an API tester:

<https://openidconnect.net/>

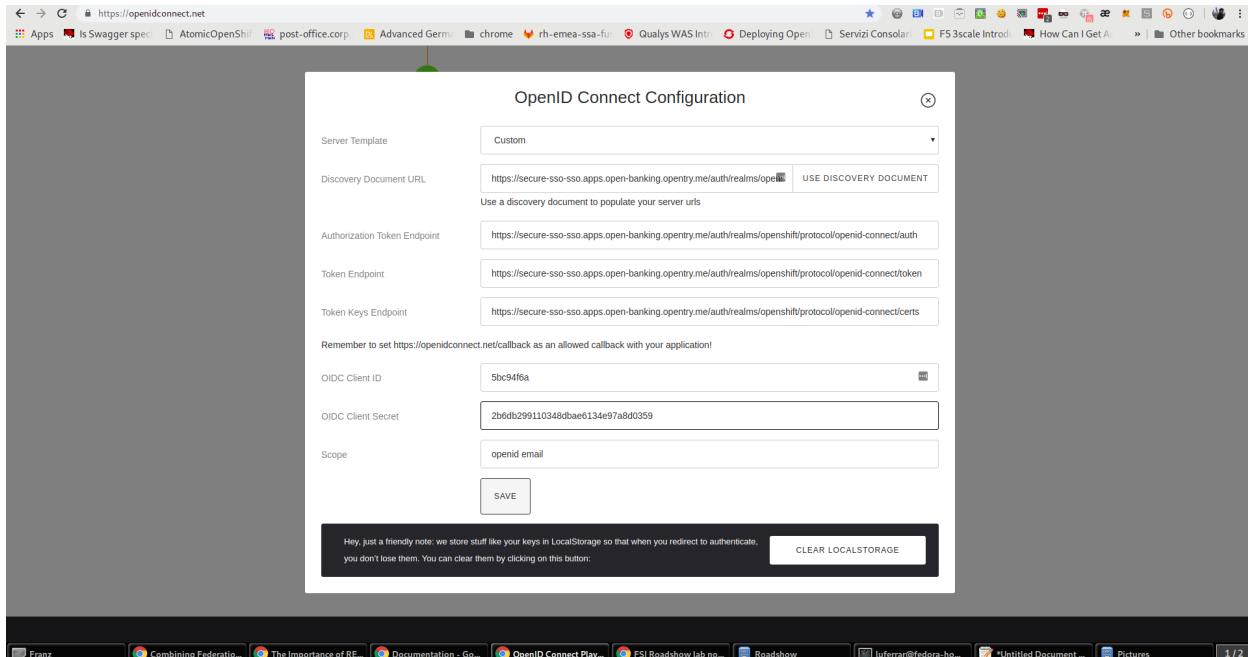
Let's configure it with the correct parameters from the previous steps. **Configuration**

Let's change the server template to custom and input in the discovery URL the one we opened before in our RH SSO realm

<https://secure-sso-sso.apps.open-banking.opentry.me/auth/realms/openshift/.well-known/openid-configuration>

We are going to use the client id and secret from the previous application in the developer portal.

And lastly as scope we are going to add openid and email. **SAVE**



Start the authentication flow by hitting start. You are going to be redirected to the RH SSO login interface where you can use your standard user details (evalsXX@example.com). Once you login you will receive a temporary code to be exchanged for the final credentials or access token.

2 Exchange Code from Token

Your Code is

```
eyJhbGciOiJkaXIIICJ1bmMl01JBMTI4Q0JDLUhTMjU2In0..VeAdYXdZhkLT
QLw0-
FZLQP0aPD1mGwe7OKDjMI_32RqnrrpcixM08YqPoPt5i6IkKjSe8hZwinWah
Bj4RzQbM53WCukRf0em3ltzLQz0t-XI4eJg08Gr5PP37PuFtkZ-
3m7ouBnDyF_ZK5msqY7Ir7x8v0K3v1KCP10F12RCY4_my_oyplhCbeJa.jlTf
```

Now, we need to turn that access code into an access token, by having our server make a request to your token endpoint

Request

```
POST https://secure-sso-sso.apps.open-
banking.opentr.y.me/auth/realm.openshift/protocol/openid-
connect/token
grant_type=authorization_code
&client_id=5bc94f6a
&client_secret=2b6db299110348dbae6134e97a8d0359
&redirect_url=https://openidconnect.net/callback
&code=eyJhbGciOiJkaXIIICJ1bmMl01JBMTI4Q0JDLUhTMjU2In0..VeAd
QLw0-
FZLQP0aPD1mGwe7OKDjMI_32RqnrrpcixM08YqPoPt5i6IkKjSe8hZwinWah
Bj4RzQbM53WCukRf0em3ltzLQz0t-XI4eJg08Gr5PP37PuFtkZ-
3m7ouBnDyF_ZK5msqY7Ir7x8v0K3v1KCP10F12RCY4_my_oyplhCbeJa.jlTf
```

EXCHANGE

Hit Exchange

The screenshot shows a browser window with a request and response pane. The request is a POST to <https://secure-sso-sso.apps.open-banking.opentrty.me/auth/realms/openshift/protocol/openid-connect/token> with parameters: client_id=5bc54f6a, client_secret=2b6db299110340dbae6134e97a0d0359, redirect_url=https://openidconnect.net/callback, code=eyJhbGciOiJkaXIlC1bmMl0lBMlI400JDUtMjU2In0..VeAdQLwD-, and FZL0PoAP01mGwe7OKDJMI_32RqnrcpixM00YqePoPt5i6N1ckjSe8hZwinBj4RzQbMS5WcUkF06m3ltzLQZot-X14eJgo8GPrSFPP37PuFtkZ-3m7oubNDoyf_ZK5msq7Ir7x8v0K3vlKCP10F1ZRCY4_my_oyplHcbeJa.jl. The response is HTTP/1.1 200 with Content-Type: application/json, containing a JSON object with access_token, expires_int, refresh_expires_in, refresh_token, token_type, id_token, not-before-policy, session_state, and scope fields.

You will receive the “access_token” which is the temporary credentials that we will be using to authenticate with 3scale to get access to the configured API. We can see that another important information is shown there regarding when this credential will expire.

We can hit **NEXT** and id_token will also be shown, which contains the more user related details.

The screenshot shows a step 3 titled "Verify User Token". It says "Now, we need to verify that the ID Token sent was from the correct place by validating the JWT's signature". Below is a "VIEW ON JWT.IO" button and a copyable token string: eyJhbGciOiJSUzI1NiIsInR5cIg0IA1SldUiIiwi2lkIiA6ICJRaIRJX2w52G0Pv5v17D-CDJp1XcbRrgNkp0b9Kfwviy8I616YX6B38rJAJ5WTCXZkyfUsruImaLWnpEP0DhiYxNg7mmgjxKKVVBmIN0Byt7WsvCGotvF_XBx9K3g. A note explains it's cryptographically signed with RS256. At the bottom is a "VERIFY" button and the Auth0 logo.

We can decode the information on JWT.io and found our user details once again as passed to the Backend service.

The screenshot shows a detailed view of an OAuth access token. The token is presented in a structured JSON format across three main sections: HEADER, PAYLOAD, and VERIFY SIGNATURE.

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT",
  "kid": "QKTI_...epKb05ZJwe7urqPQkchDMF-R2xFpMmeBvh-U"
}
```

PAYOUT: DATA

```
{
  "jti": "7dec7531-70ea-41ea-ad31-1502a0aca437",
  "exp": 1547396457,
  "nbf": 8,
  "iat": 1547396157,
  "iss": "https://secure-sso.sso.apps.openshift.openebs.io",
  "aud": "Spc94f6a",
  "sub": "236c4443-58c6-47b4-ae3d-987bb05e9836",
  "typ": "ID",
  "azp": "Spc94f6a",
  "auth_time": 1547395876,
  "session_state": "fa149b80-d4e9-49be-95b0-9613ff0a55bd",
  "acr": "9",
  "preferred_username": "evals98@example.com",
  "email": "evals98@example.com"
}
```

VERIFY SIGNATURE

```
RSASHA256(
  base64UR1Encode(header) + "." +
  base64UR1Encode(payload),
  Public Key or Certificate. Enter it in plain text only if you want to verify it later
```

Let's now go back to our OpenID client and copy the access token long string.

It should look something like this:

```
eyJhbGciOiJSUzI1NiInR5cClgOiAiSldeUliwia2lkiA6ICJRa1RX2VwS2lwNVpFsKp3TdlcnFQUWtjSERNRi1SMnhGcEtzUJ2aC1VIn0eyJqdGkiOilyYzJmZjQ5Z01MDY4LTQ0MjQYTRIN05MWU30Tk3MTM0YTMIcJleHAiOjE1NDczOTc1NTlsIm5iZl6MCviaWF0!joxNTQ3Mzk2NjuYLCJpc3MjOjodHRwczoL3NIY3VyzS1zc28tc3NvLmFwcHMub3BibiliYW5raW5nLm9wZW50cnkubWuvYXVo9yZWFsbXmvb3BlnNaoWZ0liwiYXVkjoiNWJjOTrmNmElCJzdWliOilyMzzjZDRhMy01MGm2LTQ3YQtYUuZC05ODdiYjA1ZT4MzYiLCJ0eXaiOjJCZWfYZxiILCJhenAiOiiYm5N5mg2Y5lslmFldGhfdGtZS16MTU0Nz5NTg3Niic2Vzc2lvb9zdgFO2z16lmZhMTQ5jhiLWQ0ZTkndiZs05NWiLtk2MTNmZjBhNTViZCIsImFjci6ljAiLCJhbGxdv2VklW9yaWdpbnMiOltLCJyZWFsbV9hY2Nlc3MiOnsicm9sZXMiOlsidWihX2F1dGhvcml6YXRpb24iXX0slnJlc291cmNIX2FjY2VzcyI6eyJhY2NvdW50lp7InJvbGVzljpbmlhbmnFzS1hY2NvdW50liwbWFuYwdILWFjY291bnQtbGlua3MiLCJ2aWV3LXByb2ZpbGUiX19LCJwcmVmZXJyZWRfdXNlcml5hbWUiOjIdmfsczk4QGV4YWlwGUuY29tiwiZWhiawwiOjldmfsczk4QGV4YWlwGUuY29tn0.07y6GDFq5CajAT0DkywEuQqEuD5H7_YMqrVC4AMPthZ-m_xZ_DAPBEqj3mmzp1o1Joo0_4pMxNgKpqqCQiy79GRSS1JE6aVrZK53rQkud5laZAE1-ryiD8Ctp_MrqtsTS7bVKbaFyCNyFxy3c-TER8GnGG900IYPxpy5M954slcp4CWxXa7ZwVEuQNRRs5w2G2TCJrFyQjCzsINFwDRtADjbMiY7kq1cwRB5qM9ipdEEigDnH8dietiOZgY24sK10vtowjz_CHuWr5W3474dAZVF
```

We are going to use this as a Header in our call towards the OpenID protected service.

Let's go back to our api tester and add this as an Authorization header. The format is
Authorization : Bearer <access_token_value_here>

API TESTER
BETA

Sign in Create Account ⓘ

Build your test

View example

Click ⚙ to add or remove steps

Request Step Name: GET https://wt2-evals99-example-com-3scale.apps.open-banking.opentry.me Headers + Add Request Header Authorization: Bearer eyJhbGciOiJSUzI1NiB

+ Add Step

Test Save to Account Share Test Config

⚠ Saving tests to your account allows you to:

- Rerun this test
- Share test results with others
- View the run history
- Create and re-run multiple tests

© 2019 RIGOR. ALL RIGHTS RESERVED

Powered by RIGOR

Franz Combinig Federatio... The Importance of RE... Documentation - Go... API Tester - Google C... FSI Roadshow lab no... Roadshow lufer...@fedor... *Untitled Document... Pictures 1 / 2

Let's hit Test

Request

Request Headers

```
GET /open-data/banks HTTP/1.1
Host: wt2-evals99-example-com-3scale.apps.open-banking.opentry.me
Accept: /*
User-Agent: Mozilla/5.0 (compatible; Rigor/1.0.0; http://rigor.com)
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cC1g0iAiSldUUiwia2lkIA6ICJRaiRjX2VwS2IwNvpFskp3ZTdicnFQUWtjSERNRiisMnhGcE1tZUJ2aC1VIn0.eyJqd6k
```

Response

Response Headers

```
HTTP/1.1 200 OK
Server: openbank/1.13.6.1
Date: Sun, 13 Jan 2019 16:26:57 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 5070
Set-Cookie: JSESSIONID=hbqayb65vua5991ip06hh3ny; Path=/;
Expires: Sun, 13 Jan 2019 16:26:57 GMT
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST
Access-Control-Allow-Headers: private, no-store
Correlation-Id: hbqayb65vua093j1qphh3ny
Pragma: no-cache
X-Frame-Options: DENY
Strict-Transport-Security: max-age=31536000; includeSubdomains
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'self'; script-src 'self' 'unsafe-inline' 'unsafe-eval'; img-src 'self' https://static.openbankproject.
Referrer-Policy: no-referrer-when-downgrade
Set-Cookie: 4e0f3rdf85272673fd1ac6837832b3=7e20029bc3fa9aa2bdf56683d95b5e541; path=/; HttpOnly; Secure
Set-Cookie: 0ce625c053b39cab8739dead9521874=fd85f9e6e21e8a0f6cad673c7c9e5b; path=/; HttpOnly; Secure
```

Response Body

```
{"banks": [{"id": "psd201-bank-x--uk", "short_name": "Bank X", "full_name": "The Bank of X", "logo": "https://static.openbankproject.com/images/sandb"}]
```

Variables

Search variables...

Franz Combinig Federatio... The Importance of RE... Documentation - Go... API Tester - Google C... FSI Roadshow lab no... Roadshow lufer...@fedor... *Untitled Document... Pictures 1 / 2

And success!

```

v {
  "banks": [
    {
      "id": "psd201-bank-x--uk",
      "short_name": "Bank X",
      "full_name": "The Bank of X",
      "logo": "https://static.openbankproject.com/images/sandbox/bank_x.png",
      "website": "https://www.example.com",
      "bank_routing": {
        "scheme": "OBP",
        "address": "psd201-bank-x--uk"
      }
    },
    {
      "id": "psd201-bank-y--uk",
      "short_name": "Bank Y",
      "full_name": "The Bank of Y",
      "logo": "https://static.openbankproject.com/images/sandbox/bank_y.png",
      "website": "https://www.example.com",
      "bank_routing": {
        "scheme": "OBP",
        "address": "psd201-bank-y--uk"
      }
    },
    {
      "id": "at02-bank-x--01",
      "short_name": "Bank X",
      "full_name": "The Bank of X",
      "logo": "https://static.openbankproject.com/images/sandbox/bank_x.png",
      "website": "https://www.example.com",
      "bank_routing": {
        "scheme": "OBP",
        "address": "at02-bank-x--01"
      }
    },
    {
      "id": "at02-bank-y--01",
      "short_name": "Bank Y",
      "full_name": "The Bank of Y",
      "logo": "https://static.openbankproject.com/images/sandbox/bank_y.png",
      "website": "https://www.example.com",
      "bank_routing": {
        "scheme": "OBP",
        "address": "at02-bank-y--01"
      }
    },
    {
      "id": "at02-2080--01",
      "short_name": "Abanca",
      "full_name": "ABANCA CORPORACION BANCARIA, S.A.",
      "logo": "https://www.abanca.es/logo"
    }
  ]
}

```

Franz Combinig Federatio... The Importance of RE... Documentation - Go... https://uptime-diagn... FSI Roadshow lab no... Roadshow lu ferr@fedora-ho... Untitled Document ... Pictures 1 / 2

The work done by the API management behind the curtain is quite impressive:

- Check for the validity of the access token credentials (not expired, legit and associated to the correct application)
- Check for rate limits on the application triggering the call
- Apply monetization rules to the call
- Apply any additional policy that might modify the call in real time

Checkpoint

Break

Practical Part 2

OpenShift

As user you will login into openshift and it already looks evident that the end user has been profiled as developer on OpenShift and it has access only to Objects and Projects he created or he has the right to see given his role.

If we click on the fuse project we will be able to access to the Fuse Online installation dedicated to the user. We would also be able to see any integration project running alongside Fuse installation.

If we switch to the cluster console, this will give us some Operations details on the project created or assigned to our user.

This type of console is also used by Operations administrators to check the health of OpenShift. We can see the RBAC in action if we click on **Home -> Status**

The screenshot shows the OpenShift Container Platform Cluster Console interface. On the left is a navigation sidebar with options like Home, Status, Search, Events, Workloads, Networking, Storage, Builds, Administration, and more. The main content area is titled "Status of default". It features a large "Restricted Access" icon (a sign with a red circle and a slash) and the text "Restricted Access". Below this, a message states "You don't have access to this section due to cluster policy." A red error box contains the text "projects.project.openshift.io \"default\" is forbidden: User \"evals98@example.com\" cannot get projects.project.openshift.io in the namespace \"default\": no RBAC policy matched". At the bottom of the page, there is a browser toolbar with various tabs and icons.

The Project default is excluded from the scope of any evals users, since it can contain system components.

We can just switch to the Fuse project to see if there anything wrong with it in the cluster.

The screenshot shows the OpenShift Container Platform Cluster Console interface, specifically for the project "fuse-dcfbb062-1520-11e9-86c5-0a580a810008". The left sidebar is identical to the previous screenshot. The main content area shows the "Status of fuse-dcfbb062-1520-11e9-86c5-0a580a810008" page. It includes sections for "Health" (Kubernetes API and OpenShift Console status are both "UP" and "All good"), "Events" (streaming events, showing 0 events in the past hour), and "Software Info" (Kubernetes v1.11.0+d4cacc0, OpenShift Container Platform v3.11.51). There are also sections for "Documentation", "Full Documentation" (with a link to the CLI), and "Additional Support" (with links to the Interactive Learning Portal, Local Development, YouTube, and Blog). The bottom of the page has a browser toolbar.

We will now try as bad intentioned user to change some parameters around the installed products.

OPENSHIFT CONTAINER PLATFORM Cluster Console

evals@88@example.com

Workloads

Pods

Deployment Configs

Stateful Sets

Secrets

Config Maps

Cron Jobs

Jobs

Daemon Sets

Replica Sets

Replication Controllers

HPAs

Networking

Storage

Builds

Administration

Projects

Service Accounts

Roles

Role Bindings

Resource Quotas

Franz

Combining Federation...

The Importance of RE...

Documentation - Go...

Pods - OpenShift Con...

Issues - syndesisio/t...

Roadshow

lufarr@fedora-ho...

*Untitled Document -

Pictures

1 / 2

syndesis-operator-1-b5tir

use-dcfbb062-1520-11e9-86c5-0a580a810008

syndesis-prometheus-1-6dcfz

use-dcfbb062-1520-11e9-86c5-0a580a810008

syndesis-server-1-jhmn

use-dcfbb062-1520-11e9-86c5-0a580a810008

syndesis-ui-1-c6gc9

use-dcfbb062-1520-11e9-86c5-0a580a810008

todo-1-8bd4k

use-dcfbb062-1520-11e9-86c5-0a580a810008

syndesis.io/type=operator

deployment=syndesis-operator-1

deploymentconfig=syndesis-operator

syndesis.io/app=syndesis

syndesis.io/component=syndesis-operator

syndesis.io/type=infrastructure

app=syndesis

deployment=syndesis-prometheus-1

deploymentconfig=syndesis-prometheus

syndesis.io/app=syndesis

syndesis.io/component=syndesis-prometheus

syndesis.io/type=infrastructure

app=syndesis

deployment=syndesis-server-1

deploymentconfig=syndesis-server

syndesis.io/app=syndesis

syndesis.io/component=syndesis-server

syndesis.io/type=infrastructure

app=syndesis

deployment=syndesis-ui-1

deploymentconfig=syndesis-ui

syndesis.io/app=syndesis

syndesis.io/component=syndesis-ui

syndesis.io/type=infrastructure

app=syndesis

deployment=todo-1

deploymentconfig=todo

syndesis.io/app=todo

syndesis.io/component=todo

syndesis.io/type=infrastructure

Running

Running

Running

Running

Running

Ready

Ready

Ready

Ready

OPENSHIFT CONTAINER PLATFORM Cluster Console

evals@88@example.com

Workloads

Pods

Deployment Configs

Stateful Sets

Secrets

Config Maps

Cron Jobs

Jobs

Daemon Sets

Replica Sets

Replication Controllers

HPAs

Networking

Storage

Builds

Administration

Projects

Service Accounts

Roles

Role Bindings

Resource Quotas

Franz

Combining Federation...

The Importance of RE...

Documentation - Go...

Pods - OpenShift Con...

Issues - syndesisio/t...

Roadshow

lufarr@fedora-ho...

*Untitled Document -

Pictures

1 / 2

Delete Pod

Are you sure you want to delete syndesis-ui-1-c6gc9 in namespace fuse-dcfbb062-1520-11e9-86c5-0a580a810008?

Cancel

Confirm

syndesis-operator-1-b5tir

use-dcfbb062-1520-11e9-86c5-0a580a810008

syndesis-prometheus-1-6dcfz

use-dcfbb062-1520-11e9-86c5-0a580a810008

syndesis-server-1-jhmn

use-dcfbb062-1520-11e9-86c5-0a580a810008

syndesis-ui-1-c6gc9

use-dcfbb062-1520-11e9-86c5-0a580a810008

todo-1-8bd4k

use-dcfbb062-1520-11e9-86c5-0a580a810008

Running

Ready

Running

Ready

Running

Ready

Running

Ready

Running

Ready

As we can see we tried to kill one of the running components of our integration platform with no success, because of the roles assigned to my user.

Let's see the magic introduced by OpenShift and login as administrator of the platform once again.

We now have full access to all the platforms from all users. We will open as admin one of the Fuse project

syndesis

- DEPLOYMENT CONFIG broker-amq, #1
- DEPLOYMENT CONFIG syndesis-db, #1
- DEPLOYMENT CONFIG syndesis-meta, #1
- DEPLOYMENT CONFIG syndesis-oauthproxy, #1
- DEPLOYMENT CONFIG syndesis-operator, #1
- DEPLOYMENT CONFIG syndesis-prometheus, #1
- DEPLOYMENT CONFIG syndesis-server, #1
- DEPLOYMENT CONFIG syndesis-ui, #1

We can also open the relative Fuse Online installation.

syndesis

- DEPLOYMENT CONFIG broker-amq, #1
- DEPLOYMENT CONFIG syndesis-db, #1
- DEPLOYMENT CONFIG syndesis-meta, #1
- DEPLOYMENT CONFIG syndesis-oauthproxy, #1
- DEPLOYMENT CONFIG syndesis-operator, #1
- DEPLOYMENT CONFIG syndesis-prometheus, #1
- DEPLOYMENT CONFIG syndesis-server, #1
- DEPLOYMENT CONFIG syndesis-ui, #1

We are going to test the auto healing capabilities of the platform by killing one if its running components, in particular the one providing the UI service.

OPENSHIFT CONTAINER PLATFORM Application Console

fuse-dcfbb062-1520-11e9-86c5-0a580a810008

Overview

- DEPLOYMENT CONFIG syndesis-oauthproxy, #1
- DEPLOYMENT CONFIG syndesis-operator, #1
- DEPLOYMENT CONFIG syndesis-prometheus, #1
- DEPLOYMENT CONFIG syndesis-server, #1
- DEPLOYMENT CONFIG syndesis-ui, #1

CONTAINERS

syndesis-ui

- Image: fuse7/fuse-ignite-ui 0c05f43 100.0 MB
- Ports: 8080/TCP

Average Usage Last 15 Minutes

NETWORKING

Service - Internal Traffic

syndesis-ui
80/TCP → 8080

Routes - External Traffic

Create Route

Deployment Config

todo, #1

OPENSHIFT CONTAINER PLATFORM Application Console

fuse-dcfbb062-1520-11e9-86c5-0a580a810008

Pods > syndesis-ui-1-c6gc9

Actions

- Add Storage
- Edit YAML
- Delete

app syndesis deployment syndesis-ui-1 deploymentconfig syndesis-ui More labels...

Details Environment Metrics Logs Terminal Events

Status

Status:	Running
Deployment:	syndesis-ui, #1
IP:	10.130.6.14
Node:	ip-172-31-3-6.ec2.internal (172.31.3.6)
Restart Policy:	Always

Container syndesis-ui

State:	Running since Jan 10, 2019 10:44:45 PM
Ready:	true
Restart Count:	0

Template

Containers

syndesis-ui

- Image: fuse7/fuse-ignite-ui 0c05f43 100.0 MB
- Ports: 8080/TCP
- Mount: config-volume ... /usr/share/nginx/html/config read-write
- Mount: default-token-qv2w ... /var/run/secrets/kubernetes.io/serviceaccount read-only
- Memory: 50 MiB to 255 MiB
- Readiness Probe: GET / on port 8080 (HTTP) 1s delay, 1s timeout
- Liveness Probe: GET / on port 8080 (HTTP) 30s delay, 1s timeout

Volumes

config-volume

Type: config map (populated by a config map)
Config Map: syndesis-ui-config

default-token-qv2w

Type: secret (populated by a secret when the pod is created)
Secret: default-token-qv2w

Add Storage to syndesis-ui | Add Config Files to syndesis-ui

Show Annotations

javascript:void(0)

OPENSHIFT CONTAINER PLATFORM Application Console

Pods > syndesis-ui-1-c6gc9

syndesis-ui-1-c6gc9 created 3 days ago

app deployment syndesis syndesis-ui-1 deploymentconfig

Details Environment Metrics Logs Terminal

Status

Status:	Running
Deployment:	syndesisui, #1
IP:	10.120.6.14
Node:	ip-172-31-3-6.ec2.internal (172.31.3.6)
Restart Policy:	Always

Container syndesis-ui

State:	Running since Jan 10, 2019 10:44:45 PM
Ready:	true
Restart Count:	0

Template

Containers

syndesis-ui

- Image: fuse7/fuse-ignite-ui 0:c05f43 100.0 MB
- Ports: 8080/TCP
- Mount: config-volume ... /usr/share/nginx/html/config read-write
- Mount: default-token-qvf2w ... /var/run/secrets/kubernetes.io/serviceaccount read-only
- Memory: 50 MiB to 255 MiB
- Readiness Probe: GET / on port 8080 (HTTP) 1s delay, 1s timeout
- Liveness Probe: GET / on port 8080 (HTTP) 30s delay, 1s timeout

Volumes

config-volume

Type: config map (populated by a config map)
Config Map: syndesis-ui-config

default-token-qvf2w

Type: secret (populated by a secret when the pod is created)
Secret: default-token-qvf2w

Add Storage to syndesis-ui | Add Config Files to syndesis-ui

Show Annotations

Cancel Delete

OPENSHIFT CONTAINER PLATFORM Application Console

DEPLOYMENT CONFIG syndesis-meta, #1

DEPLOYMENT CONFIG syndesis-oauthproxy, #1

DEPLOYMENT CONFIG syndesis-operator, #1

DEPLOYMENT CONFIG syndesis-prometheus, #1

DEPLOYMENT CONFIG syndesis-server, #1

DEPLOYMENT CONFIG syndesis-ui, #1

CONTAINERS

syndesis-ui

- Image: fuse7/fuse-ignite-ui 0:c05f43 100.0 MB
- Ports: 8080/TCP

Average Usage Last 15 Minutes

Mb Memory Cores CPU Kib/s Network

NETWORKING

Service - Internal Traffic
syndesis-ui

Routes - External Traffic
Create Route

Cancel Delete



This page isn't working

`fuse-dcfbb062-1520-11e9-86c5-0a580a810008.apps.openbanking.opentry.me` is currently unable to handle this request.

HTTP ERROR 502

[Reload](#)



OPENSHIFT CONTAINER PLATFORM Application Console ▾

fuse-dcfbb062-1520-11e9-86c5-0a580a810008

DEPLOYMENT CONFIG syndesis-meta, #1

340	< 0.01	0.1
Mib Memory	Cores CPU	Kib/s Network

DEPLOYMENT CONFIG syndesis-oauthproxy, #1

12	< 0.01	1.5
Mib Memory	Cores CPU	Kib/s Network

DEPLOYMENT CONFIG syndesis-operator, #1

16	< 0.01	2.8
Mib Memory	Cores CPU	Kib/s Network

DEPLOYMENT CONFIG syndesis-prometheus, #1

50	< 0.01	1.9
Mib Memory	Cores CPU	Kib/s Network

DEPLOYMENT CONFIG syndesis-server, #1

570	< 0.01	89
Mib Memory	Cores CPU	Kib/s Network

DEPLOYMENT CONFIG syndesis-ui, #1

CONTAINERS

syndesis-ui

- Image `fusat7/fuse-ignite-ui:0c05f43` 100.0 MIB
- Ports: 8080/TCP

Average Usage Last 15 Minutes

MEMORY

syndesis-ui

- Image `fusat7/fuse-ignite-ui:0c05f43` 100.0 MIB
- Ports: 8080/TCP

NETWORKING

Service - Internal Traffic

syndesis-ui

Routes - External Traffic

Create Route

1 pod

Franz Combinig Federatio... The Importance of RE... Documentation - Go... OpenShift Web Cons... FSI Roadshow lab no... Roadshow lufer...@fedora-ho... *Untitled Document... Pictures 1 / 2

The screenshot shows the Red Hat Fuse Online web application. The top navigation bar includes links for Home, Swagger spec, AtomicOpenShift, post-office.corp, Advanced German, chrome, rh-emea-ssa-fu, Qualys WAS Int..., Deploying Open..., Servizi Consolari, F53scale Intro!, How Can I Get..., and Other bookmarks. The main header says "RED HAT FUSE ONLINE". On the left, a sidebar menu lists Home, Integrations, Connections, Customizations, and Settings. The main content area has a "System Metrics" section with four cards: "0 Integrations" (green), "5 Connections" (green), "11 Total Messages" (green), and "Uptime" (2 days 19 hours 38 minutes). Below this is a "Create an Integration" button with a plus sign icon. The "Connections" section shows four components: PostgresDB, open-financial-service, Log, and Timer, each with a small icon. At the bottom, there's a toolbar with various browser tabs and a status bar showing "1 / 2".

As we can see the component auto-healed based and in a few seconds we have a GUI running once again for the integration platform.

Q&A

Common issues

- While using Fuse Online it I have experienced some timeouts and slowness. If that happens too often you can always use the same URL that all the other attendees are using (from our open-bank demo portal) to test the API. Also make sure you use a different browser for Fuse Online scenario than the one used for Integr8ly demos
- openidconnect.net client might have an additional space in the redirect_uri field. That's a client bug, you can fix it by adding an additional redirect URIs in RH SSO with a space preceding the URL: “<https://openidconnect.net/callback>”