

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:

Environment reachable here:

<https://tutorial-web-app-webapp.apps.openbanking-b7ca.openshiftworkshop.com>

End Users credentials:

user[1..100] / openshift

3scale dedicated instances here:

<https://userX-admin.apps.openbanking-b7ca.openshiftworkshop.com>

access using username and password

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. At the top, there's a header bar with the Red Hat logo and a developer account icon. Below the header, a banner reads "Welcome to the Red Hat Solution Explorer" and "Get started with an end-to-end solution walkthrough or use any of the available application services to create custom integrations." On the left, a sidebar titled "Start with a walkthrough" lists three options: "Integrating event-driven and API-driven applications (AMQ)" (preview, Get Started, 15 min), "Integrating event-driven and API-driven applications (EnMasse)" (Get Started, 21 min), and "Integrating API-driven applications" (preview, Get Started, 21 min). To the right, there's a section titled "Applications" listing seven items: 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, community), EnMasse (Ready for use), Red Hat Fuse (Ready for use, preview), and Red Hat Developer Launcher (Ready for use, community). At the bottom, there's a navigation bar with several tabs and a page number indicator "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

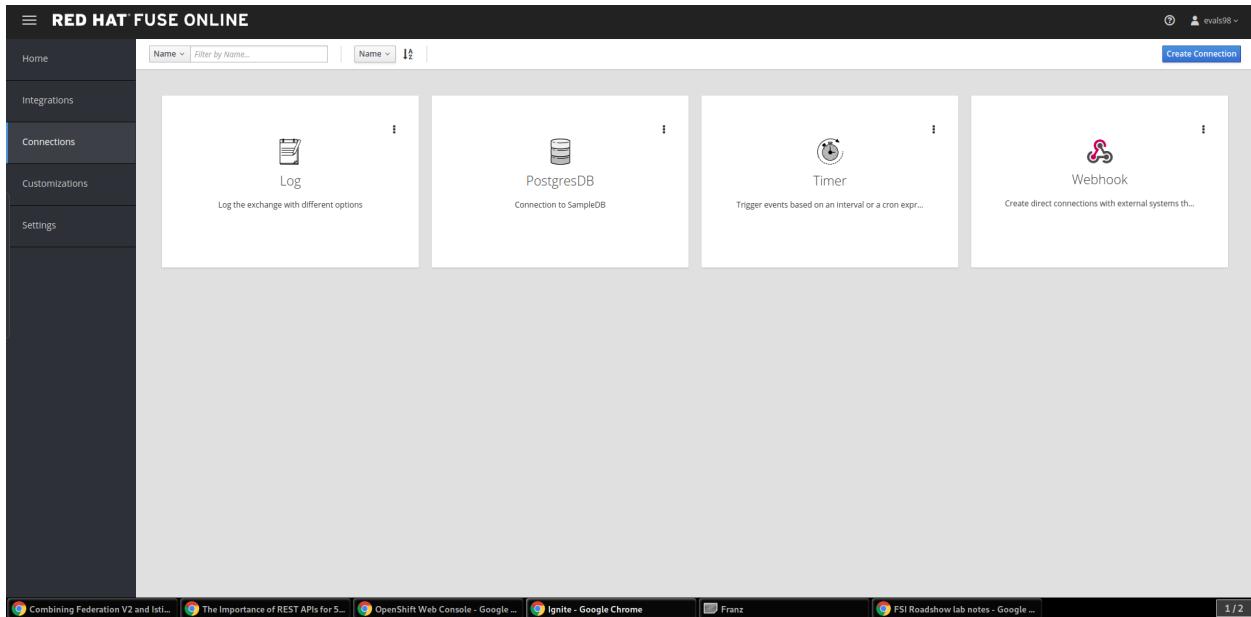
Then start by opening Red Hat Fuse.

The screenshot shows the Red Hat Fuse Online interface. On the left, a sidebar menu includes Home, Integrations, Connections, Customizations, and Settings. The main area displays 'System Metrics' with four cards: '0 Integrations' (green), '4 Connections' (grey), '0 Total Messages' (green), and 'Uptime Since Jan 10th 22:44 PM 2 minutes'. Below this is a central section with a large '+' button labeled 'Create an Integration'. A message states 'There are currently no Integrations. Click the button below to create one.' A blue 'Create Integration' button is present. At the bottom, there's a 'Connections' section with icons for PostgresDB, Log, Timer, and Webhook, along with 'View All Connections' and 'Create Connection' buttons. The browser taskbar at the bottom shows several open tabs, including 'Combining Federation V2 and Istio...', 'The Importance of REST APIs for E...', 'OpenShift Web Console - Google ...', 'Ignite - Google Chrome', 'Franz', and 'FSI Roadshow lab notes - Google ...'.

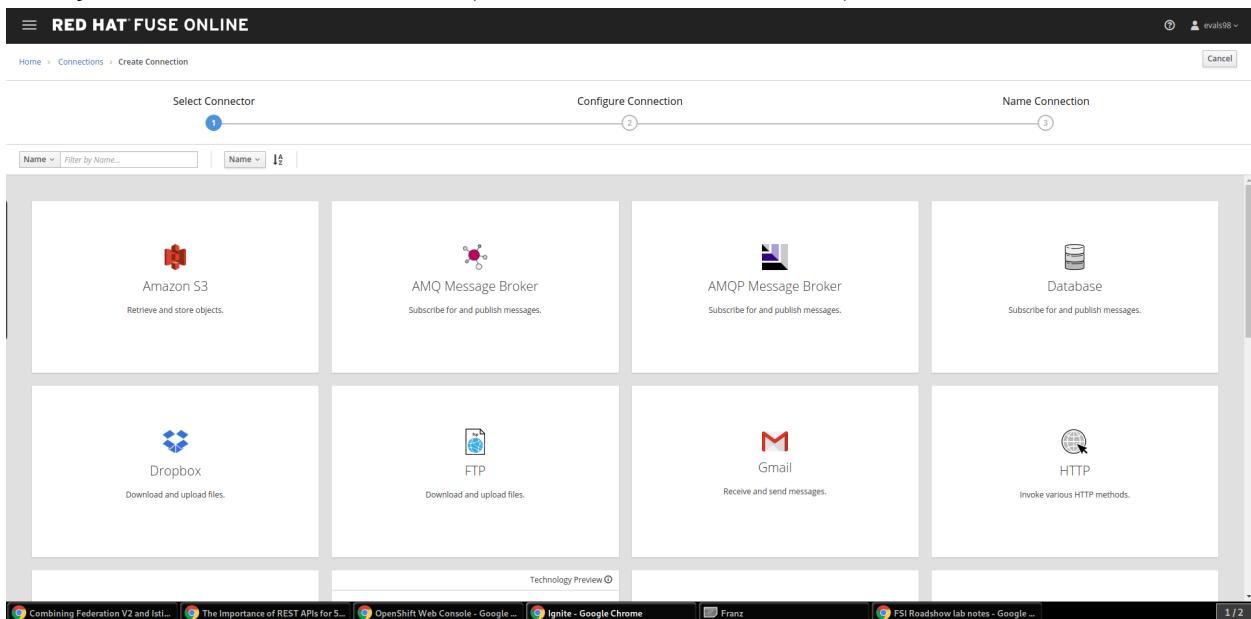
Explain the main dashboard, especially important when you then show messages coming request being sent.

The screenshot shows the 'Create an integration' window. It features a large '+' button and a message: 'There are currently no integrations available. Please click on the button below to create one.' A blue 'Create Integration' button is centered. The background shows the same Red Hat Fuse Online dashboard elements as the previous screenshot. The browser taskbar at the bottom is identical.

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, and Settings. The 'Customizations' option is currently selected. The main content area has a header 'API Client Connectors' with a sub-header 'ignite creates an API client connector when you upload a valid OpenAPI 2.0 specification that describes the API you want to connect to.' Below this is a search bar with dropdowns for 'Name' and 'Filter by Name...' and a sorting icon. In the center, there's a large circular button with a plus sign labeled 'Create API Connector'. Below it, the text 'Create API Connector' is displayed, followed by a note: 'There are currently no API connectors available. Please click on the button below to create one.' A blue 'Create API Connector' button is located at the bottom of this section. At the very bottom of the page, a standard browser navigation bar is visible.

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 remains the same as the previous screenshot. The main content area has a header 'Extensions' with a sub-header 'Extensions provide custom features for use in integrations. Find out more at [ignite Help](#)'. Below this is a search bar with dropdowns for 'Name' and 'Filter by Name...' and a sorting icon. In the center, there's a large circular button with a plus sign labeled 'Import Extension'. Below it, the text 'Import Extension' is displayed, followed by a note: 'There are no extensions available.' A blue 'Import Extension' button is located at the bottom of this section. At the very bottom of the page, a standard browser navigation bar is visible.

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 page in the Red Hat Fuse Online interface. It lists several 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 page in the Red Hat Fuse Online interface, specifically for the Gmail connector. The 'Client ID' and 'Client Secret' fields are highlighted with red asterisks, indicating they are required inputs. The 'Save' and 'Cancel' buttons are visible at the bottom of the form.

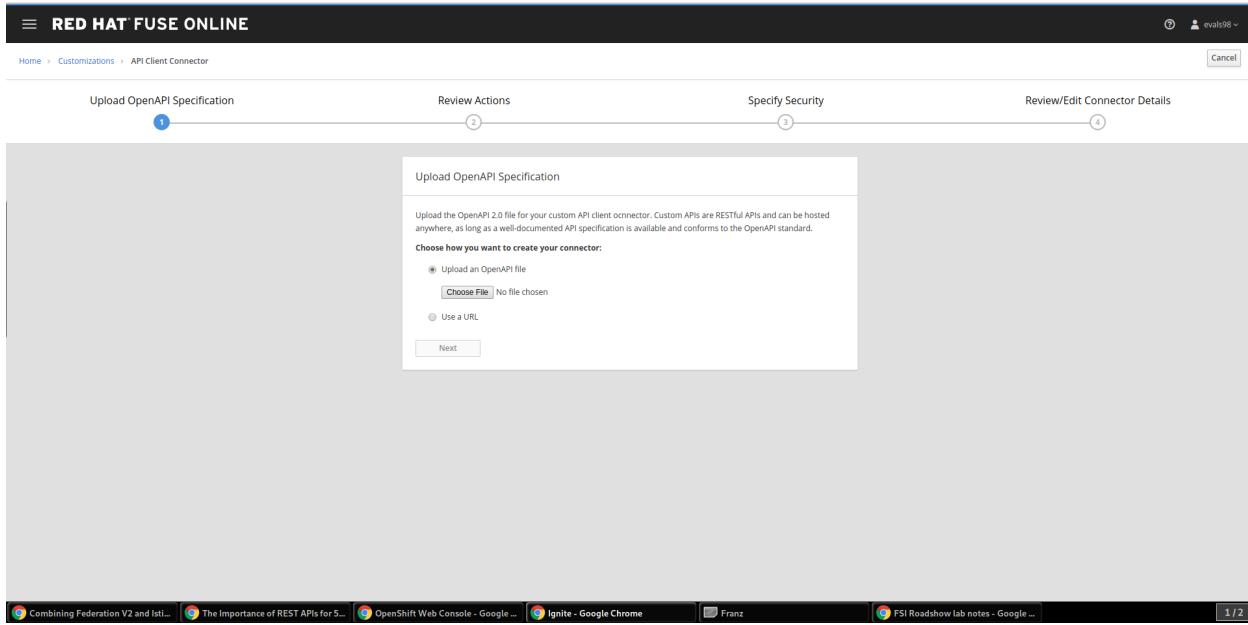
Make sure to explain a common use case of this, for example authenticating with Twitter account when accessing a news portal.

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

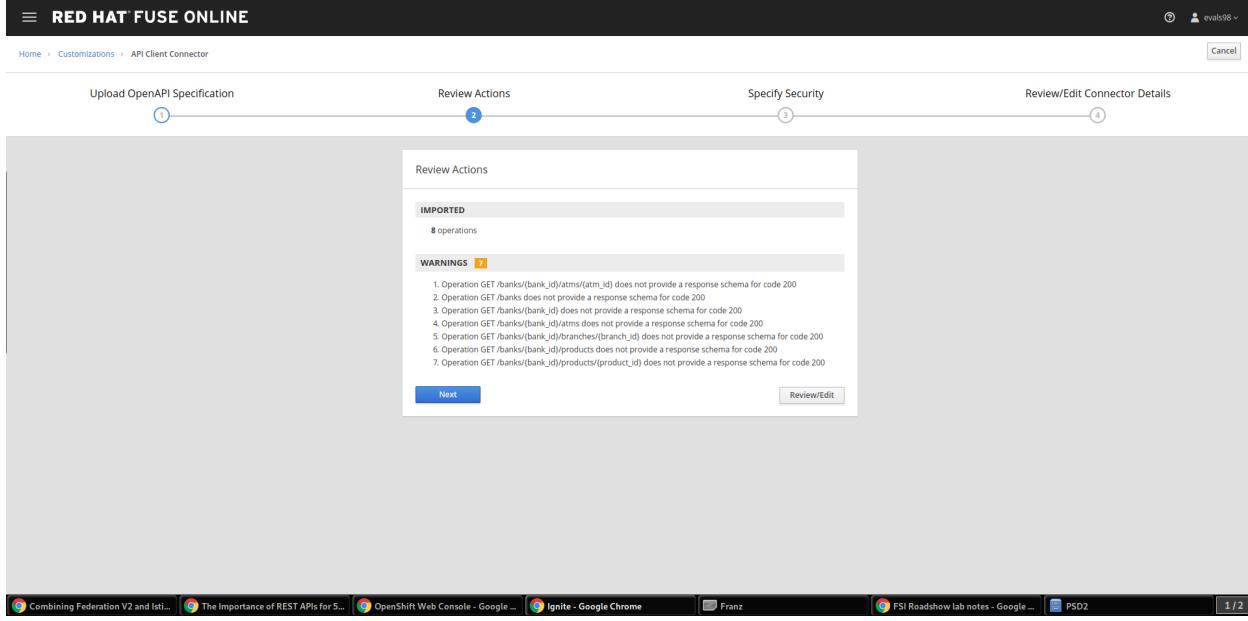


Combining Federation V2 and Isti... The Importance of REST APIs for S... OpenShift Web Console - Google ... Ignite - Google Chrome Franz FSI Roadshow lab notes - Google ... 1 / 2

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 ->**



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

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**

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

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

open-data-apis OpenBanking open data apis Used by integrations 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**

The screenshot shows the Red Hat Fuse Online interface. On the left, a sidebar menu includes Home, Integrations (which is selected), Connections, Customizations, and Settings. The main content area has a search bar at the top with dropdown menus for 'Name' and 'Filter by Name...'. A large central button with a plus sign and the text 'Create an integration' is present. Below it, a message states: 'There are currently no integrations available. Please click on the button below to create one.' At the bottom of the main area is a blue 'Create Integration' button. The browser's address bar shows the URL 'https://fuse.redhat.com/integrations'. The status bar at the bottom of the browser window displays various tabs and the number '1 / 2'.

This screenshot shows the 'Choose a Start Connection' step in the integration creation process. The left sidebar shows 'Enter Int...' with a plus icon. The main area has a breadcrumb trail: Home > Integrations > New Integration > Choose a Start Connection. It features a search bar with dropdowns for 'Name' and 'Filter by Name...'. A 'Create Connection' button is located in the top right. Below, four connection types are listed in cards: 'Technology Preview' (API Provider, Expose Restful APIs), 'PostgresDB' (Connection to SampleDB), 'Timer' (Trigger events based on an interval or a cron expr...), and 'Webhook' (Create direct connections with external systems th...). A dashed box highlights the 'Create Connection' button in the bottom-left corner of the card area. The browser's address bar shows the URL 'https://fuse.redhat.com/integrations/integration/new'. The status bar at the bottom of the browser window displays various tabs and the number '1 / 2'.

API provider. Upload the same definition as before.

RED HAT FUSE ONLINE

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

RED HAT FUSE ONLINE

Home > Integrations > expose financial open data > Choose operation

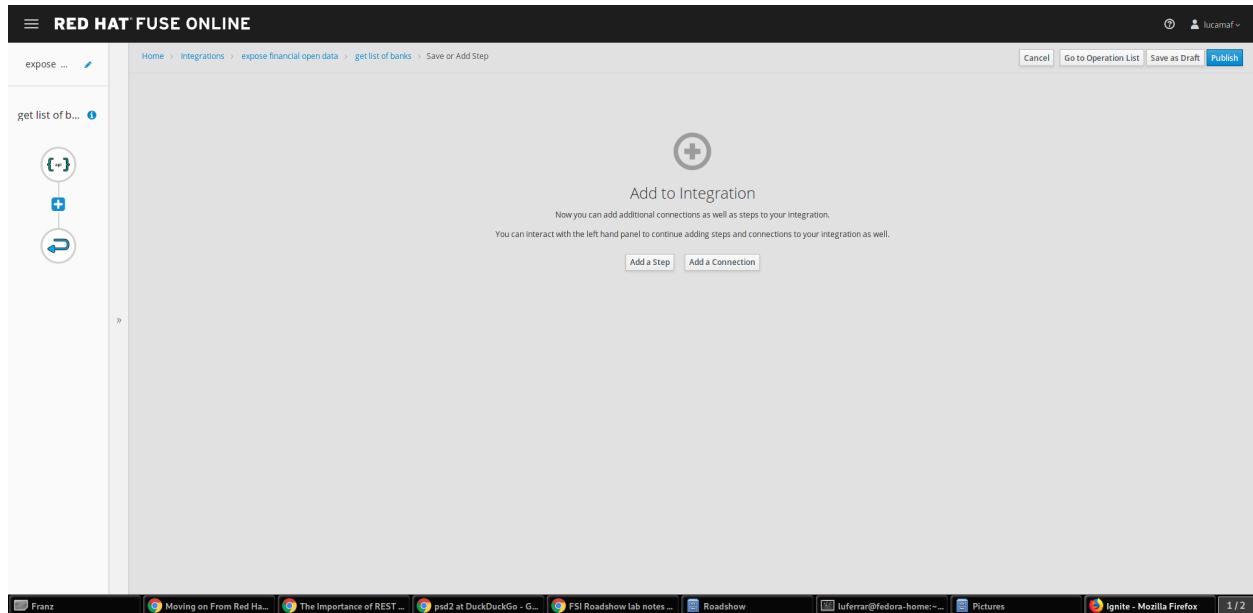
Choose operation

Click an operation to integrate an integration flow.

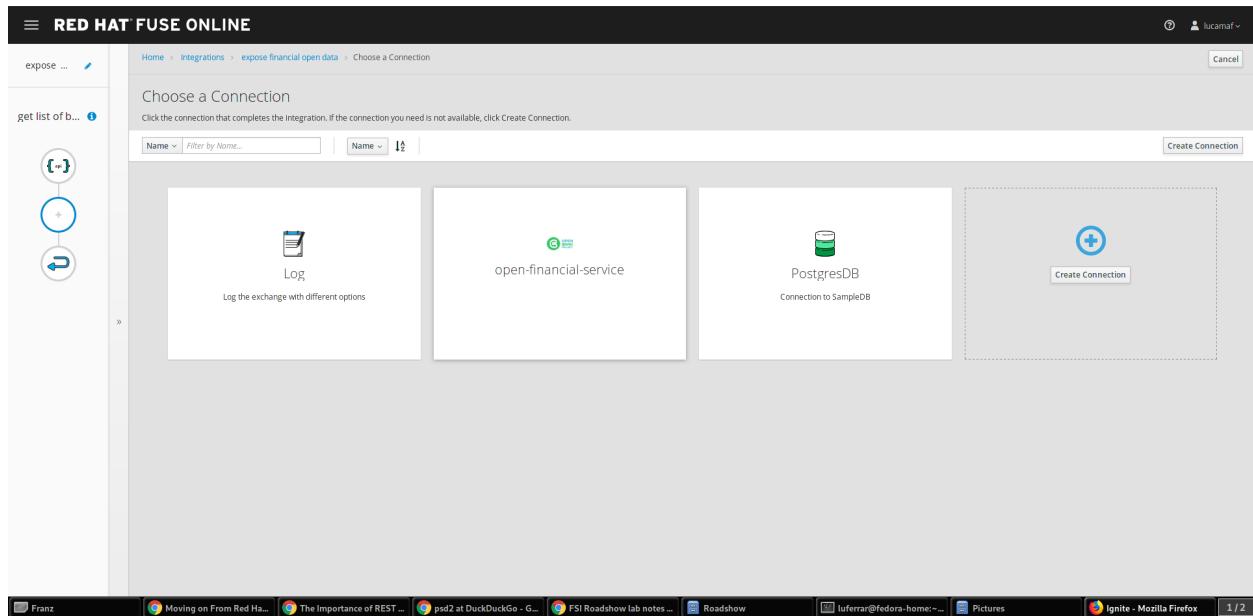
Name	Method	Description
get atm info	GET /banks/{bank_id}/atms/{atm_id}	501 Not Implemented
get bank atms	GET /banks/{bank_id}/atms	501 Not Implemented
get bank details	GET /banks/{bank_id}	501 Not Implemented
get branch info	GET /banks/{bank_id}/branches/{branch_id}	501 Not Implemented
get branches 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

Franz Moving on From Red Ha... The Importance of REST ... psd2 at DuckDuckGo - G... FSI Roadshow lab notes ... Roadshow lu ferrari@fedora-home:~ Pictures Ignite - Mozilla Firefox 1 / 2

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).



The screenshot shows the Red Hat Fuse Online interface. On the left, there's a sidebar with a tree view showing a connection named "get list of b...". The main area is titled "open-financial-service - Choose an Action" and contains a list of REST API endpoints:

- 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)

At the bottom right of the dialog, there are "Cancel", "Go to Operation List", "Save as Draft", and "Publish" buttons.

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

Add a Step | Add a Connection

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

Log
Log the exchange with different options

open-financial-service

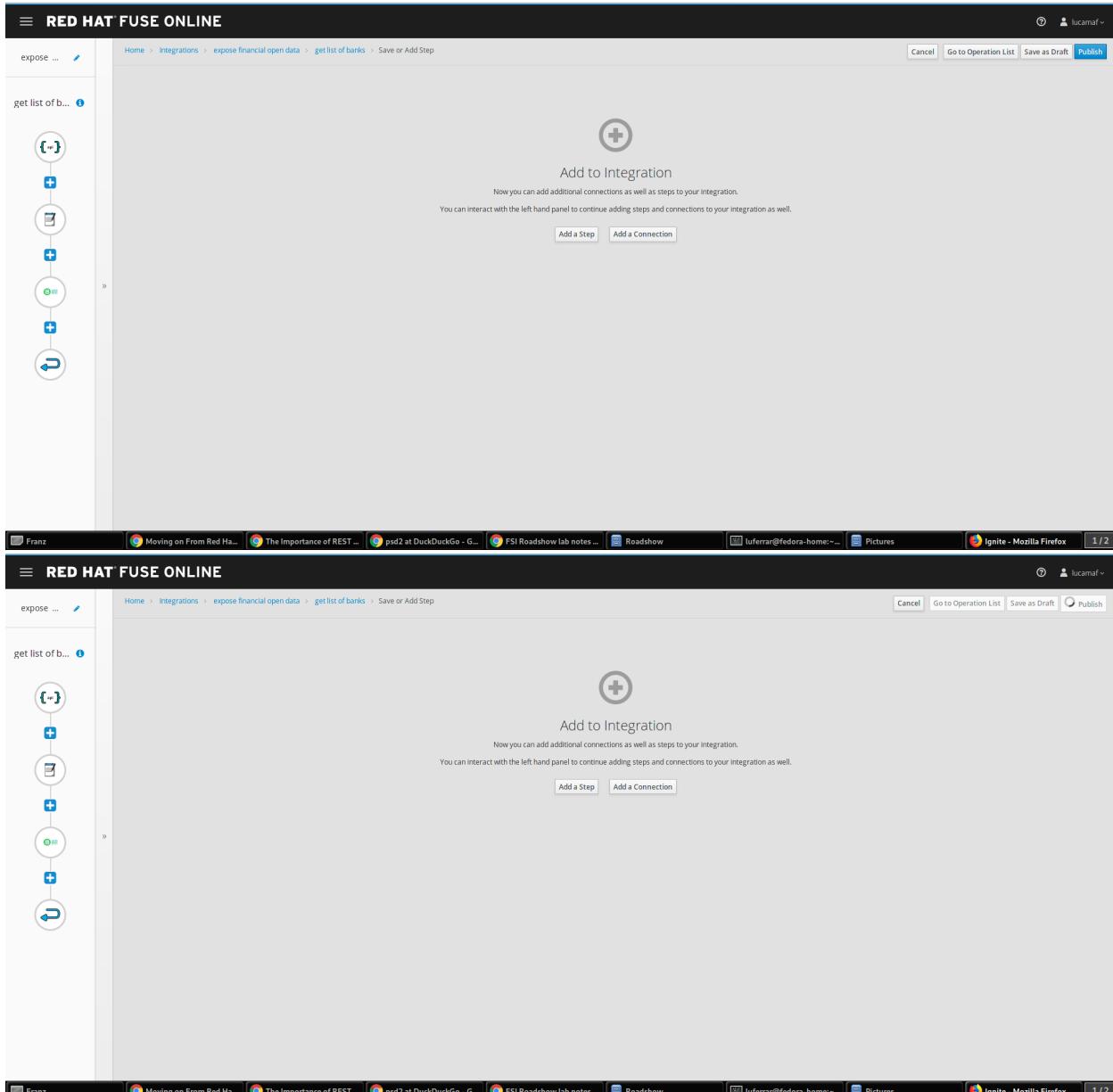
PostgresDB
Connection to SampleDB

Create Connection

The screenshot shows the Red Hat Fuse Online interface. On the left, there's a sidebar with icons for exposing services, getting lists of blocks, and a search bar. The main area is titled "Log - Choose an Action" and says "Choose an action for the selected connection." It contains a search bar and a list of actions. One action, "Simple Logger", is highlighted with a black border. The URL in the browser is "Home > Integrations > Choose Action".

The screenshot shows the configuration of the "Simple Logger" action. The title is "Configure Simple Logger". It has a "Log Level" section with "INFO" selected. Under "Log Level", there are four checkboxes: "Log everything" (checked), "Log Body", "Log message Id", and "Log Headers". At the bottom are "Cancel", "< Choose Action", and "Done" buttons. The URL in the browser is "Home > Integrations > expose_financial_open_data > Configure Simple Logger".

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.

```

{
  "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": "at02-2888-.01",
      "short_name": "Abanca",
      "full_name": "ABANCA CORPORACION BANCARIA, S.A.",
      "logo": "http://foo.gl/NStW1U",
      "website": "WWW.ABANCA.COM/ES",
      "bank_routing": {
        "scheme": "OBP"
      }
    }
  ]
}

```

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.

The screenshot shows the API Tester BETA interface. At the top, there's a navigation bar with links like 'Sign In' and 'Create Account'. Below it, a main area titled 'Build your test' has a sub-section 'View example'. A 'Request' step is selected, showing a 'GET' method and the URL 'https://open-data.b9ad.pro-us-east-1.openshiftapps.com/open-data/banks'. There are tabs for 'Headers' and '+ Add Request Header'. Below the request step, there are buttons for 'Test', 'Save to Account', and 'Share Test Config'. A tooltip box explains the benefits of saving tests to the account. At the bottom, there's a copyright notice '© 2019 RIGOR. ALL RIGHTS RESERVED' and a 'Powered by RIGOR' logo.

The screenshot shows the API Tester interface after a test run. The message box at the top says '[Step 1] no auth financial service passed'. Below it, the 'Connection Information' section shows a 'Request' tab with 'Request Headers' containing a GET request to the specified URL. The 'Response' section shows the full HTTP response headers and body. The response body is a JSON array: [{"id": "psd201-bank-x--uk", "short_name": "Bank X", "full_name": "The Bank of X", "logo": "https://static.openbankproject.com/images/sand"}]. The 'Response Body' section includes icons for copy, download, and eye (to toggle visibility).

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

```

{
  "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.com"
    }
  ]
}

```

Combining Federation V... The Importance of REST... OpenShift Web Console... https://uptime-diagnos... Franz FSI Roadshow lab notes... PSD2 lufer@fedora-home:~ Ignite - Google Chrome 1 / 2

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's a navigation bar with links for Documentation, Dashboard (which is active), Developers, Applications, Billing, Analytics, API, Developer Portal, and Settings. The main content area is divided into several sections:

- DEVELOPERS**: Shows 1 Signups (last 30 days) and 0 Potential Upgrades (today). It includes a note about adding usage limits to Application Plans and enabling Web Alerts for Admins.
- ACCOUNT**: Shows 1 ACCOUNT (TODAY) with the message "No news, good news."
- MESSAGES**: Shows 0 messages (TODAY) with the message "Blue skies ahead..."
- API**: Shows 0 Hits (last 30 days) and 1 APPLICATION (2 PLANS). It includes a note about sending traffic to the API and making test calls.

Support Version 2.2.0 - Powered by 3scale

Combining Federation V... The Importance of REST... OpenShift Web Console... Dashboards - Show | Re... Franz FSI Roadshow lab notes... PSD2 lufer@fedora-home:~ Ignite - Google Chrome 1 / 2

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**

API Definition, Integration and Settings

On the Integration page, add your API base URL to the Private Base URL field in the staging box and hit Update & Test. Once the staging box is green, hit Deploy in the production box and you have completed a basic integration, ready to go!

Grab after a delay of: 0 + seconds

Effects

- Include pointer
- Include the minimum border
- Apply effects: None

Create Service

Analytics

Hits: 0 hits

Latest alerts

There are no alerts.

Latest Apps

Developer's App from Developer

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**

Name	State	Account	Plan	Paid?	Created At	Traffic On
Developer's App	live	Developer	Basic	free	January 09, 2019	

Applications

Support

Version 2.2.0 - Powered by

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**

Developer's App

Description: Default application created on signup.

Service: API

State: Live suspend

API Credentials:

User Key: 13d113t578be9d43b296a566d1e24616

Regenerate Set Custom Key

Usage in last 30 Days:

Hits: 0 hits

Application Plan: Basic

Features:

- Unlimited Greetings: ✓
- 24/7 support: ✗
- Unlimited calls: ✗
- Customize: [Customize](#)

Change Plan

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**

Documentation Dashboard

Edit application: Developer's App

Name*: Developer's App

Description*: Default application created on signup.

Delete

Update Application

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 ...**

Configuration: configure & test immediately in the staging environment documentation

API

Private Base URL*: https://echo-api.3scale.net:443

API GATEWAY

Staging Public Base URL*: https://api-3scale-apicast-staging.apps.open-banking.opentry.me:443

Production Public Base URL*: https://api-3scale-apicast-production.apps.open-banking.opentry.me:443

MAPPING RULES

AUTHENTICATION SETTINGS

POLICIES

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>)

Configuration: configure & test immediately in the staging environment documentation

API

Private Base URL*: https://open-data.b9ad.pro-us-east-1.openshiftapps.com:443

API GATEWAY

Staging Public Base URL*: https://wt2-evalsX-example-com-3scale.apps.open-banking.opentry.me:443

Production Public Base URL*: https://wt2-evalsX-example-com-3scale.apps.open-banking.opentry.me:443

MAPPING RULES

Verb	Pattern	Metric or Method (Define)
GET	/open-data/banks	1 hits

AUTHENTICATION SETTINGS

Host Header

Secret Token

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 configuration interface for an API gateway. At the top, there are two URL fields: 'Staging Public Base URL' and 'Production Public Base URL', both set to `https://w2-eval98-example-com-3scale.apps.open-banking.opentrue.me:443`. Below these are sections for 'MAPPING RULES' and 'AUTHENTICATION SETTINGS'.

MAPPING RULES:

- Verb: GET, Pattern: `/open-data/banks$`, Metric or Method: hits
- Buttons: Add Mapping Rule, Save, Cancel

AUTHENTICATION SETTINGS:

- Host Header:** A field for defining a custom Host request header.
- Secret Token:** A field for a shared secret token used to block direct developer requests to the API backend.
- CREDENTIALS LOCATION:** Radio buttons for 'As HTTP Headers' (selected) and 'As query parameters (GET) or body parameters (POST/PUT/DELETE)'.
- Auth user key:** A field for the authentication user key, currently set to `api-key`.

OTHER SETTINGS:

- AUTHENTICATION FAILED ERROR:** Fields for Response Code (403), Content-type (text/plain; charset=us-ascii), and Response Body (Authentication failed).
- AUTHENTICATION MISSING ERROR:** Fields for Response Code (403), Content-type (text/plain; charset=us-ascii), and Response Body (Authentication parameters missing).

At the bottom of the interface, there is a toolbar with various icons and tabs, and a status bar indicating '1 / 2'.

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'

Host header

Lets you define a custom Host request header. This is needed if your API backend only accepts traffic from a specific host.

Secret Token

Shared_secret_sent_from_proxy_to_API_backend_b631809b9926e89a

Enables you to block any direct developer requests to your API backend: each 3scale API gateway call to your API backend contains a request header called X-3scale-proxy-secret-token. The value of this header can be set by you here. It's up to you to ensure your backend only allows calls with this secret header.

CREDENTIALS LOCATION

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

Auth user key key

AUTHENTICATION FAILED ERROR

Response Code* 403

Content-type text/plain; charset=us-ascii

Response Body Authentication failed

AUTHENTICATION MISSING ERROR

Response Code* 403

Content-type text/plain; charset=us-ascii

Response Body Authentication parameters missing

NO MATCH ERROR

Combining Federation ... The Importance of REST ... Documentation - Go... Integrations - Update ... Franz FSI Roadshow lab notes ... PSD2 Ignite - Google Chrome Trello Openbanking API Pr... 1 / 2

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

Policy Chain	Add Policy
APIcast	builtin - Main functionality of APIcast to work with the 3scale API manager.

CLIENT

API test GET request /

Optional GET request to a API gateway endpoint. We will use this call to validate your API gateway setup using credentials of the first live application. You can try it yourself by copying the following command into your shell:

```
curl "https://api-3scale-apicast-staging.apps.open-banking.openentry.me:443/" -H 'user_key: USER_KEY'
```

Hit the test button to check the connections between client, gateway & API.

Update & test in Staging Environment

Back to Integration & Configuration

Combining Federation V... The Importance of REST ... OpenShift Web Console ... Integrations - Edit | Red ... Franz FSI Roadshow lab notes ... PSD2 luferarr@fedora-home:~ Ignite - Google Chrome 1 / 2

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

Response Body: No Mapping Rule matched

POLICIES

Select a Policy

oauth2 token introspection policy
builtin - Configures OAuth 2.0 Token Introspection.

Upstream
builtin - Allows to modify the upstream URL of the request based on its path.

rate limit policy
builtin - Adds rate limit.

URL rewriting
builtin - Allows to modify the path of a request.

Echo
builtin - Prints the request back to the client and optionally sets a status code.

SOAP
builtin - Adds support for a small subset of SOAP.

Headers
builtin - Allows to include custom headers.

CORS
builtin - Enables CORS (Cross Origin Resource Sharing) request handling.

Caching
builtin - Controls how to cache authorizations returned by the 3scale backend.

CLIENT

API test GET request: /

Optional GET request to a API gateway endpoint. We will use this call to validate your API gateway setup using credentials of the first live application. You can try it yourself by copying the following command into your shell:
curl "https://api-3scale-apicast-staging.apps.open-banking.opentrace.me:443/open-data/banks" -H "user-key: USER_KEY"

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 \$).

built-in - Allows to modify the upstream URL of the request based on its path.

rate limit policy
builtin - Adds rate limit.

URL rewriting
builtin - Allows to modify the path of a request.

Echo
builtin - Prints the request back to the client and optionally sets a status code.

SOAP
builtin - Adds support for a small subset of SOAP.

Headers
builtin - Allows to include custom headers.

CORS
builtin - Enables CORS (Cross Origin Resource Sharing) request handling.

Caching
builtin - Controls how to cache authorizations returned by the 3scale backend.

CLIENT

API test GET request: /open-data/banks/

Optional GET request to a API gateway endpoint. We will use this call to validate your API gateway setup using credentials of the first live application. You can try it yourself by copying the following command into your shell:
curl "https://api-3scale-apicast-staging.apps.open-banking.opentrace.me:443/open-data/banks" -H "user-key: USER_KEY"

Hit the test button to check the connections between client, gateway & API.

Update & test in Staging Environment

Back to Integration & Configuration

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.

We will now fix the test request error.

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

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

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.

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 'Developer: Account Summary' page. At the top, there's a 'Custom Account Plan' section with a 'Edit' button and a 'Remove customization' link. Below it, the 'Billing Status' section indicates that monthly billing is enabled. On the left, there's a table with developer information: Organization/Group Name (Developer), Status (Approved), Administrator (John Doe, email: admin+test@3scale.apps.open-banking.opentry.me), and Signed up on (January 09, 2019 11:57). A 'Send message' and 'Edit' button are also present.

Support

Version 2.2.0 - Powered by 3scale

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 account. It features a table with columns: Name, State, Plan, Paid?, Created At, and Traffic On. A green 'Create Application' button is located at the top right of the table area. Below the table, a message says 'No applications'.

Support

Version 2.2.0 - Powered by 3scale

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

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

The screenshot shows the browser's toolbar with several tabs open, including 'Combining Federatio...', 'The Importance of RE...', 'Documentation - Go...', 'Applications - New | ...', 'Franz', 'FSI Roadshow lab no...', 'PSD2', 'Ignite - Google Chro...', 'Trello', and 'Openbanking API Pr...'. The tab for 'Applications - New | ...' is currently active.

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

The screenshot shows the 3scale platform's analytics page for the application 'my first finance app'. The top navigation bar includes 'Dashboard', 'Developers', 'Applications' (active), 'Billing', 'Analytics', 'API', 'Developer Portal', and 'Settings'. The page title is 'my first finance app'. It features sections for 'Description' (my first finance app), 'Service' (API), 'State' (Live suspend), 'API Credentials' (User Key: 4c571db87ab2b5aedccbd8877627a090), 'Application Plan: Basic' (Features: Unlimited Greetings (green checkmark), 24/7 support (red X), Unlimited calls (red X), Customize (green circle)), 'Change Plan' (button), 'Usage in last 30 Days' (Hits: 0 hits), and 'Current Utilization' (text: This is an unmetered application, there are no limits defined). The bottom of the screen shows the same browser toolbar as the previous screenshot.

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's integration configuration page. A red vertical bar on the left indicates a 'NO MATCH ERROR'. The 'Response Body' field contains 'Authentication parameters missing'. The 'Response Code*' field is set to '404'. The 'Content-type' field is 'text/plain; charset=us-ascii'. The 'Response Body' field below is 'No Mapping Rule matched'. Under 'POLICIES', there is a 'Policy Chain' section with an 'Add Policy' button. An 'APIcast' policy is listed with the note: 'builtin - Main functionality of APIcast to work with the 3scale API manager.' On the left, a 'CLIENT' icon is shown. Below it, an 'API test GET request' is defined with the URL '/open-data/banks'. The description states: 'Optional GET request to a API gateway endpoint. We will use this call to validate your API gateway setup using credentials of the first live application. You can try it yourself by copying the following command into your shell!'. A 'Test request failed with HTTP code 403: Authentication failed' message is displayed, along with a curl command: `curl "https://wt2-eval98-example-com-3scale.apps.open-banking.opentry.me:443/open-data/banks" -H 'key: 4c571db87a8205aedccbd8877627a090'`. A note says: 'Hit the test button to check the connections between client, gateway & API.' A blue button at the bottom right says 'Update & test in Staging Environment'. A link 'Back to Integration & Configuration' is at the bottom.

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

This screenshot is identical to the one above, but the 'Test request failed with HTTP code 403: Authentication failed' message is no longer present. Instead, a green vertical bar on the left indicates a 'Connection between client, gateway & API is working correctly as reflected in the analytics section.' The rest of the interface remains the same, including the 'Update & test in Staging Environment' button and the 'Back to Integration & Configuration' link.

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.

Echo API

Register **Get your API key** **Create your app**

Pick your plan

BASIC	UNLIMITED
Features ✓ Unlimited Greetings	Limits No limits
	Features ✓ Unlimited Greetings ✓ 24/7 support ✓ Unlimited calls
	Limits No limits
Signup to plan Basic	
Signup to plan Unlimited	

Combining Federatio... The Importance of RE... Documentation - Go... https://3scale.apps.o... Franz FSI Roadshow lab no... PSD2 Ignite - Google Chro... Trello Openbanking API Pr... 1 / 2

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

SIGNED IN SUCCESSFULLY

Echo API

Your API Key

This is your API key that should be kept secret. Use it to authenticate and report the calls you make to the Echo API.

CREDENTIALS

[See your Applications & their credentials](#)

Run your requests

```
$ curl -v https://echo-api.3scale.net
```

REQUEST	RESPONSE
> GET / HTTP/1.1 > User-Agent: curl/7.27.0 > Host: https://echo-api.3scale.net/echo > Accept: */*	< HTTP/1.1 200 OK < Content-Type: text/plain; charset=utf-8 < Connection: close echo

Combining Federatio... The Importance of RE... Documentation - Go... Homepage - Google... Franz FSI Roadshow lab no... PSD2 Ignite - Google Chro... Trello Openbanking API Pr... 1 / 2

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, a sub-menu for 'APPLICATIONS' is open, showing a table with one row. The row contains the application name 'my_first_finance_app', its credentials '4c571db87a82b5aedccbd8877627a090', and its state 'live'. There is also a blue button labeled 'Create new application'.

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 the text 'key' and a dropdown menu showing '4c571db87a82b5aedccbd8877627a090'. Below the search bar is a button labeled '+ Add Step'. At the bottom of the toolbar are buttons for 'Test' (highlighted in blue), 'Save to Account', and 'Share Test Config'.

The main area displays a green header with a checkmark icon and the text 'PASS'. Below this, it says 'Results 576 ms'. To the right, it shows 'N. Virginia' and 'Seconds elapsed: 2'. A progress bar indicates step 1 of 1.

Under the results, there is a 'Message' section containing '[Step 1] key financial service passed'. Below this is a 'Connection Information' section, which is collapsed. Under 'Request', there is a 'Request Headers' section showing:

```

GET /open-data/banks HTTP/1.1
Host: w12-eval198-example-com-3scale.apps.open-banking.opentry.me
Accept: /*
User-Agent: Mozilla/5.0 (compatible; Rigor/1.0.0; http://rigor.com)
key: 4c571db87a82b5aedccbd8877627a090
    
```

Below the request headers is a 'Response' section, which is collapsed. Under 'Response Headers', it shows:

```

HTTP/1.1 200 OK
Server: opentry/1.13.6.1
Date: Fri, 11 Jan 2019 18:04:18 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 5070
Last-Modified: Fri, 11 Jan 2019 18:04:18 GMT
Expires: Fri, 11 Jan 2019 18:04:18 GMT
Access-Control-Allow-Origin: *
Cache-Control: no-store, no-cache, must-revalidate
    
```

Notice the URL, the key Header and related value

Let's just test with a wrong key or path

PASS

Results 111 ms

N. Virginia
Seconds elapsed: 2

Viewing 3 Request Step

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: openresty/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 screenshot shows the Red Hat Single Sign-On Admin Console interface. On the left, there's a sidebar with sections like 'Configure', 'Manage', and 'Logs'. The main area is titled 'OpenShift' and shows the 'General' tab selected. It has fields for 'Name' (set to 'openshift'), 'Display name', 'HTML Display name', 'Enabled' (set to 'ON'), and 'Endpoints' (set to 'OpenID Endpoint Configuration'). At the bottom are 'Save' and 'Cancel' buttons.

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 the tokens default parameters. **Endpoints -> OpenID Endpoint Configuration**

The screenshot shows a browser window displaying the JSON configuration for the 'well-known/openid-configuration' endpoint. The JSON object includes fields like 'issuer', 'authorization_endpoint', 'token_endpoint', 'token_introspection_endpoint', 'userinfo_endpoint', 'end_session_endpoint', 'jwks_uri', 'check_session_iframe', 'grant_types_supported', 'response_types_supported', 'subject_types_supported', 'id_token_signing_alg_values_supported', 'userinfo_signing_alg_values_supported', 'request_object_signing_alg_values_supported', 'response_modes_supported', 'registration_endpoint', and 'token_endpoint_auth_methods_supported'. The 'Raw' and 'Parsed' buttons are visible at the top right of the JSON viewer.

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.

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**

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**

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**

ID	Username	Email	Last Name	First Name	Actions
0eb01f1ae-1292-4ca9-83d5-c15fc40a...	admin@example.com	admin@example.com			Edit Impersonate Delete
fe06099c-204c-46a4-b1ef-e119a0b8...	eval01@example.com	eval01@example.com			Edit Impersonate Delete
31dc7463-cd73-48fa-9f11-9119fb63...	eval02@example.com	eval02@example.com			Edit Impersonate Delete
5fb5c5a4-69a5-41d9-9f57-ab1208c7...	eval03@example.com	eval03@example.com			Edit Impersonate Delete
036ab9d-81f5-4a69-8cb8-6585e574f...	eval04@example.com	eval04@example.com			Edit Impersonate Delete
6ed01bed-10fd-4c29-a7fc-45923b3d...	eval05@example.com	eval05@example.com			Edit Impersonate Delete
14423744-1dbd-4072-a55b-429e209...	eval06@example.com	eval06@example.com			Edit Impersonate Delete
7776ef57-6c8d-44de-9a48-266e082...	eval07@example.com	eval07@example.com			Edit Impersonate Delete
97b191a8-1860-4b45-a3bb-2ea3799...	eval08@example.com	eval08@example.com			Edit Impersonate Delete
b3270190-e083-472d-9f16-2cdf0bf...	eval09@example.com	eval09@example.com			Edit Impersonate Delete
8c7a6b09-1771-4b2e-b15a-488176...	eval10@example.com	eval10@example.com			Edit Impersonate Delete
aba0b040-3c39-4793-b758-7a572b...	eval10@example.com	eval10@example.com			Edit Impersonate Delete
c77196a0-2043-4d9c-bc7d-23725d1...	eval11@example.com	eval11@example.com			Edit Impersonate Delete
e949ef9b-4b33-470e-a004-342d18b...	eval12@example.com	eval12@example.com			Edit Impersonate Delete
fd54725b-546d-4506-bc51-364348f1...	eval13@example.com	eval13@example.com			Edit Impersonate Delete
b420f03aff-0d49-4847-9e66-19674ac...	eval14@example.com	eval14@example.com			Edit Impersonate Delete
58814193-0c84-4aff-92e8-e09bf08f83...	eval15@example.com	eval15@example.com			Edit Impersonate Delete
9cae0759c-54ff-4564-a44c-973a4117...	eval16@example.com	eval16@example.com			Edit Impersonate Delete
332dc58c-a342-4f71-9292-12cef07...	eval17@example.com	eval17@example.com			Edit Impersonate Delete
37dfcf70-1bd3-4499-9f93-c9cb249...	eval18@example.com	eval18@example.com			Edit Impersonate Delete

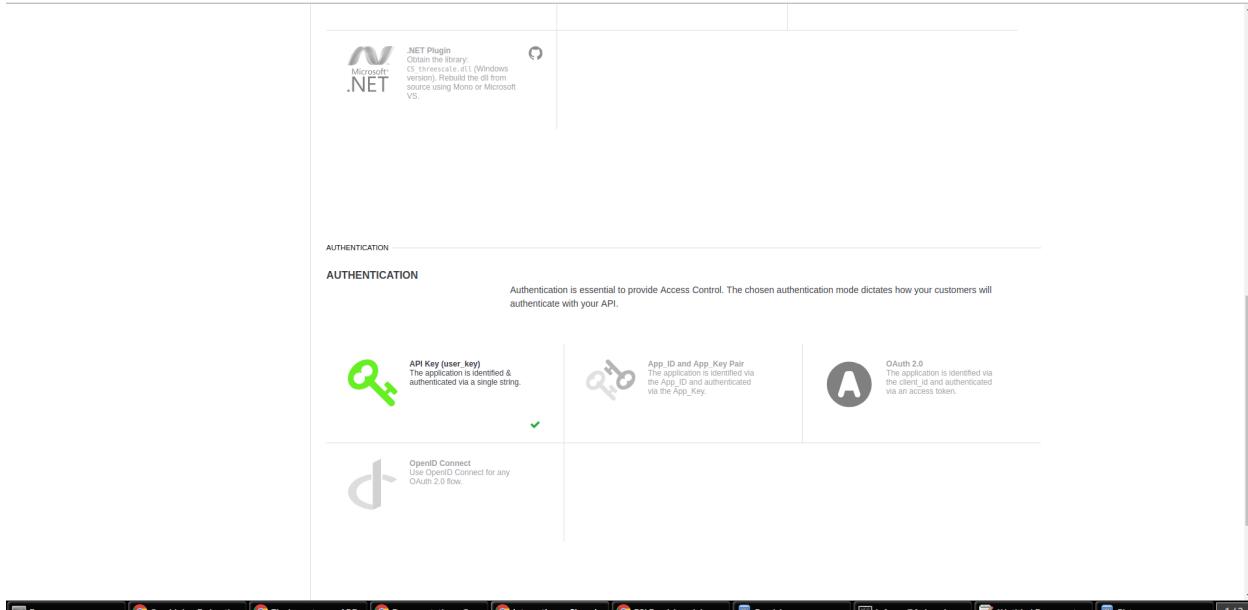
Here we can see all the user that are stored inside RH SSO, making it act as an IDM as well. These are the end users of the applications created in Clients. Let's explore one of these users.

The screenshot shows the Red Hat Single Sign-On (SSO) interface. On the left, there's a sidebar with 'Configure' and 'Manage' sections. Under 'Configure', 'Realm Settings' is selected. Under 'Manage', 'Users' is selected. The main area displays a user profile for 'evals98@example.com'. The profile includes fields for ID (236cd4a3-50c6-47b4-ac3d-987bb05e9836), Created At (1/9/19 12:48:50 PM), Username (evals98@example.com), Email (evals98@example.com), First Name, Last Name, User Enabled (ON), Email Verified (ON), Required User Actions (Select an action...), and Impersonate user (Impersonate). Buttons for 'Save' and 'Cancel' are at the bottom.

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.

The screenshot shows the 3scale API Integration & Configuration page. The left sidebar has 'Integration' selected under 'Definition'. The main content area shows 'Integration settings' with Deployment Option set to APicast and Authentication set to API Key (user_key). Below this is the 'APICAST Configuration' section, which contains Private Base URL: https://open-data.b9ad.pro-us-east-1.openshiftapps.com:443, Mapping rules: /open-data/banks\$ >> hits, Credential Location: query, and Secret Token: Shared_secret_sent_from_proxy_to_API_backend_id_b31809d8926e99a. The 'Environments' section lists 'Staging Environment' (https://w2-evals99-example-com-3scale.apps.open-banking.openhy.me:443, v. 41) and 'Production Environment' (https://api-3scale-apicast-production.apps.open-banking.openhy.me:443, v. 41). A 'Configuration history' link is shown next to each environment entry.

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

A modal dialog box is displayed, showing a warning message:

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?

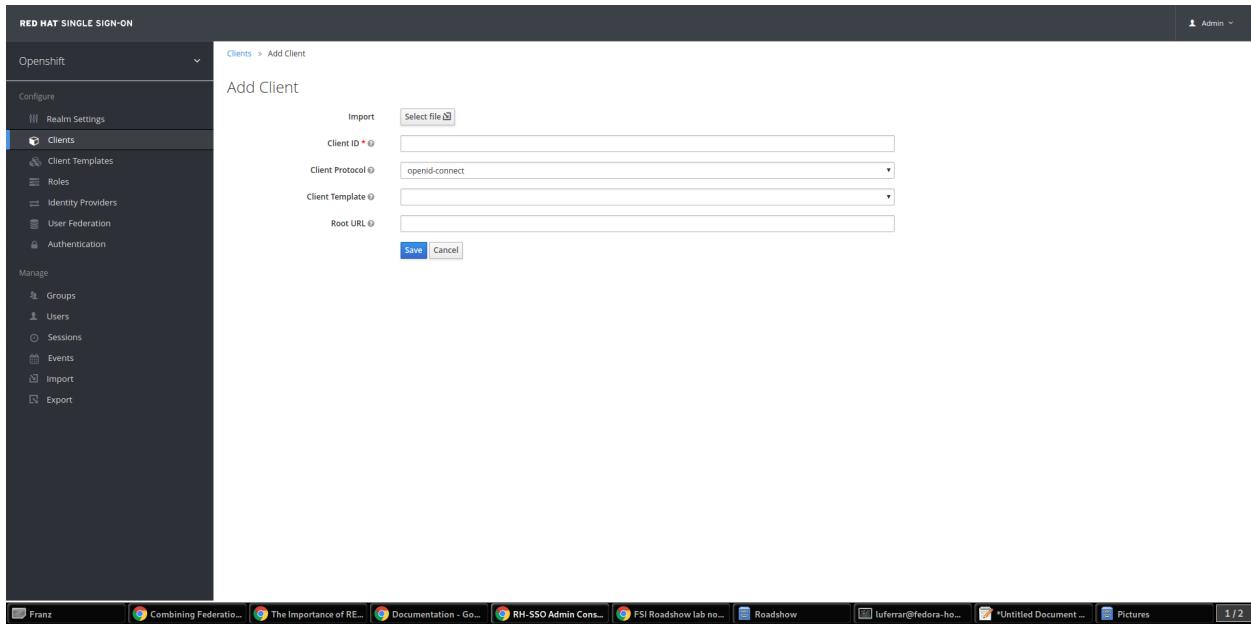
Buttons: Cancel, OK

Below the modal, the configuration interface shows the same four authentication modes again. The 'OpenID Connect' option is highlighted with a green checkmark.

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**



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

Client ID: sync-app

Name:

Description:

Enabled: ON

Consent Required: OFF

Client Protocol: openid-connect

Client Template: public

Access Type: public

Standard Flow Enabled: ON

Implicit Flow Enabled: OFF

Direct Access Grants Enabled: ON

Root URL:

Valid Redirect URIs:

Base URL:

Admin URL:

Web Origins:

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

Save -> Service account roles

Add manage-clients to the allowed roles for this client

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 mapping rule is defined for a GET request to the path `/open-data/banks$`. The rule has a metric of 1 hit. Below the mapping rule, there are sections for **AUTHENTICATION SETTINGS**, **CREDENTIALS LOCATION**, and **AUTHENTICATION FAILED ERROR**. In the **AUTHENTICATION SETTINGS** section, the OpenID Connect Issuer is set to `https://sync-apps.fcf7d61-8252-4078-ab82-52aa451dd487@secure-sso-sso.apps.open-banking.o`. The Host Header is left empty. The Secret Token is set to `Shared_secret_sent_from_proxy_to_API_backend_b631809d8926e89a`. In the **CREDENTIALS LOCATION** section, the credential location is set to **As HTTP Headers**. In the **AUTHENTICATION FAILED ERROR** section, the response code is set to 403, the Content-type is set to `text/plain; charset=us-ascii`, and the Response Body is set to `Authentication failed`.

Let's build a url of this format:
<https://client-id:client-secret@secure-sso-sso.apps.open-banking.opentry.me/auth/realmss/openshift>

And update the staging environment and promote the configuration to production

The screenshot shows the 3scale API Integration & Configuration page. The APICAST Configuration section displays the following details:

- Private Base URL: `https://open-data.b9ad.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_b631809d8926e89a`

Below this, the Environments section shows two environments:

- Staging Environment**: URL `https://rw2-eval99-example-com-3scale.apps.open-banking.opentry.me:443`, Version v. 42.
- Production Environment**: URL `https://api-3scale-apicast-production.apps.open-banking.opentry.me:443`, Version v. 42.

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 being viewed is named "my first finance app - sz". Key details include:

- Name:** my first finance app - sz
- Description:** my first finance app - sz
- Plan:** Basic > Review/Change
- Status:** Live
- Client ID:** 5bc94f6a (Note: This Client ID is absent from the configuration form below.)
- Client Secret:** 2b6db299118348dbae6134e97a8d0359 (Note: This Client Secret is present in the configuration form below.)
- Redirect URL:** https://openidconnect.net/callback

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 configuration has been updated. The application "my first finance app - sz" now includes the Redirect URL "https://openidconnect.net/callback". Other fields remain the same.

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

Screenshot of the 3scale by Red Hat application management interface showing the configuration and monitoring of an application named "my first finance app - sz".

Application Plan: Basic

Features	Status
Unlimited Greetings	✓
24/7 support	✗
Unlimited calls	✗
Customize	

API Credentials

Client ID	5bc94f6a	
Client Secret	2b6eb299110348dbae6134e97ab00359	Regenerate
Redirect URL	https://openidconnect.net/callback	Change

Usage in last 30 Days

7 hits

Current Utilization

This is an unmonitored application, there are no limits defined.

RED HAT SINGLE SIGN-ON

Clients

Client ID	Enabled	Base URL	Actions
3scale	True	Not defined	Edit Export Delete
5bc94f6a	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
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

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/>

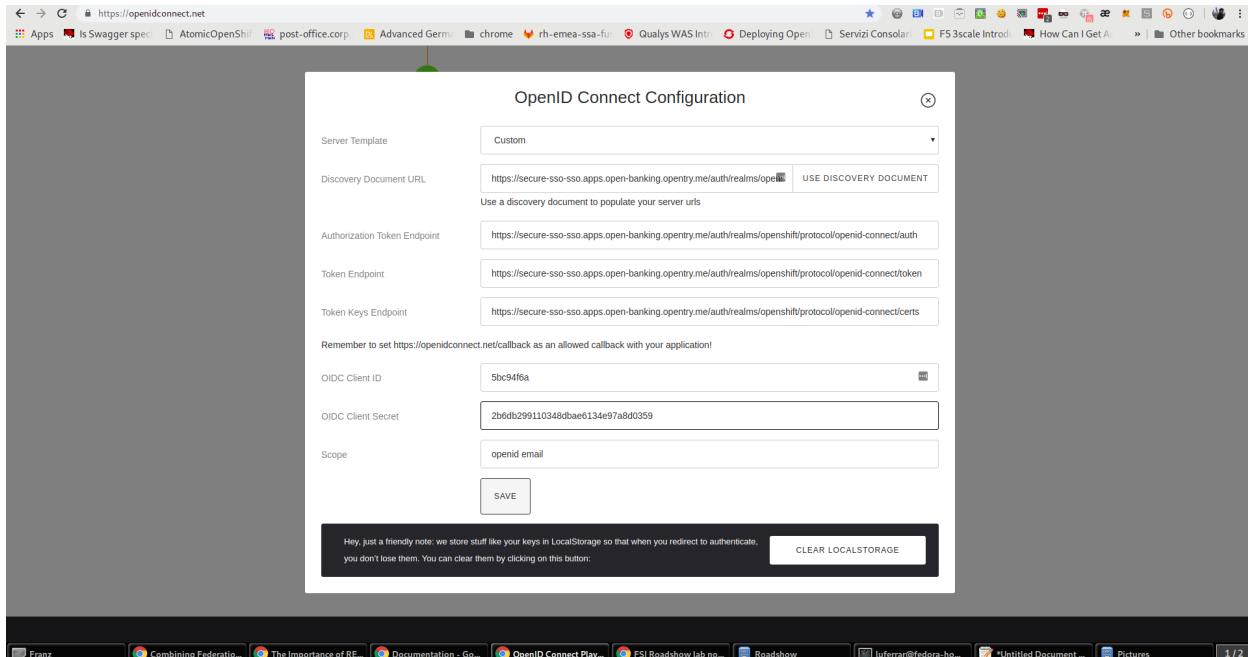
Let's configure it with the correct parameters from the previous steps. **Configuration**

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.opentrty.me/auth/realm/realms/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 the following details:

Request:

```
POST https://secure-sso-sso.apps.openbanking.opentrty.me/auth/realms/openshift/protocol/openid-connect/token
grant_type=authorization_code
&client_id=5bc54f6a
&client_secret=2b6db299110340dbae6134e97a0d0359
&redirect_url=https://openidconnect.net/callback
&code=eyJhbGciOiJkaXIlC1bmMj0iJBMTI4Q0JDLUtMjU2In0..VeAd
QLwD-
FZLOP0aP01mGwe7OKDJMI_32RqnrcpixM00YqePoPt5i6N1cKjSe8hZwin
Bj4RzQbMS5WcvkRf06m3tzLQz0t-X14eJgo8GPrSFPP37PuFtkZ-
3m7oubNDyf_ZK5msqY7Ir7xVb9K3vlKCP10F1ZRCY4_my_oyplHcbeJa.jl
```

HTTP/1.1 200

Content-Type: application/json

```
{
  "access_token": "eyJhbGciOiJSUzIiNiIsInRcCiIg0IAisldUiIwi",
  "expires_in": 300,
  "refresh_expires_in": 1800,
  "refresh_token": "eyJhbGciOiJSUzIiNiIsInRcCiIg0IAisldUiIwi",
  "token_type": "bearer",
  "id_token": "eyJhbGciOiJSUzIiNiIsInRcCiIg0IAisldUiIwi2lk",
  "not-before-policy": 0,
  "session_state": "fa149b8b-d4e9-49be-95b6-9613ff8a55bd",
  "scope": ""
}
```

NEXT

Address bar: Franz, Combining Federatio..., The Importance of RE..., Documentation - Go..., OpenID Connect Play..., FSI Roadshow lab no..., Roadshow, luferarr@fedora-ho..., *Untitled Document ..., Pictures

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.

3 Verify User Token

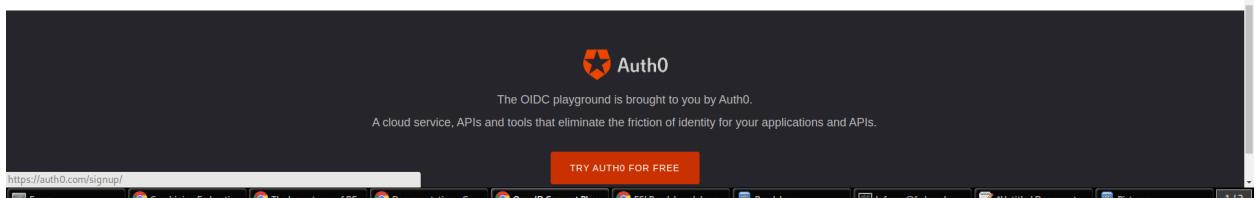
Now, we need to verify that the ID Token sent was from the correct place by validating the JWT's signature

Your “id_token” is [VIEW ON JWT.IO](#)

```
eyJhbGciOiJSUzIiNiIsInRcCiIg0IAisldUiIwi2lkIA6ICJRa1RX2Vw5
2G0Pzv517D-
CDJplXoCbVRr9gNkp0Bb9KFWviyWBI616YX6838JAJ5WTCXZkyfuaSruIma
LWlpEP0hiYxNg7m gjxKtV8bMUN80yb7WSVCGotvF_XBx9K3g
```

This token is cryptographically signed with the RS256 algorithm. We'll use the public key of the OpenID Connect server to validate it. In order to do that, we'll fetch the public key from <https://secure-sso-sso.apps.openbanking.opentrty.me/auth/realms/openshift/protocol/openid-connect/certs>, which is found in the discovery document or configuration menu options.

VERIFY



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

eyJhbGciOiJSUzI1NiIsInR5cCgOiAiSldeIwiia2IkIa6ICJRa1RX2VwS2IwNvpFsKp3ZTdlcnFQUWtjSERNRi1SMnhGeE1tZUJ2aC1Vn0.eyJqdGkiOiI3ZGNjNzU2MSB3MGvhLTQxWEYyWQzMS8xNTAyYThhY2E0Mzc1CJ1eHAi0jE1NDcz0TY8TcsIm51Zii6MCwiaWF0IjoxNTQ3Mzk2MTU3LCJpc3M1oIJo4HrwczovL3N1Y3VyzS1zc28tc3NvLmFwcHMub381b11Yw5raW5nLmW50cnkbUWVYXV0aC9yZWFsBXMvb3B1bnNoaZ0IiwiYXVkJoi0NwMj0TRkmMfiLCJzdW10IiMyMzjzDRhMy01MGm2LTQ3j0tYwUzZC050DdiYjA1ZTk4MzYiLCJ0eXAiOjJRCIsImF6cI61jViYzk0ZjzhIwiwYXV0af9baW11joxNTQ3Mzk10Dc2LCJzXNzaW9uX3N0YXR1IjoiZmExND1lOGItZDR1OS00OWJ1Tkt1YjAtOTYxM2ZmMGE1NWjkTiwiYVNyIjoiMCisInByZWZlcnJ1Zf91c2VybmfZS16ImV2YWx0zOThAZXhbXbsZS5jb20iLCJ1bWFpBC16ImV2YXz0ThAZXhbXbsZS5jb20ifQ.DxqOTuAT1vAWGMMFFje-2GP05v517D-CDJpLXoCbVr9gNkPbB9KFvviyW8I616YX6B38rJAj5WTCXZkyfUaSruImoM7fmPlfRfxwNAGT221CqHTCpTx6vQcmhg5Po..._cy1Tv6dwEcrxkeKVvn16FKkk0CpEUgVamElEy5Vkmlo1X8Rc0TRKfdnNZF6AT9LnhDzN5npHH-20qR1sWtZyYGJuuuQmxA272R7QKZTxXCNPjZBOKPUachbDvHmCbn5YGPQUPKKbCQTN2z9ufngVjLX01kg-a-LWNpEP0dh1YxNg7mmgjxKKYV8bMUW80yB7WsvCgo

Franz

Combining Federatio...

The Importance of RE...

Documentation - Go...

JSON Web Tokens - J...

FSI Roadshow lab no...

Roadshow

lferrari@fedora-ho...

Untitled Document...

Pictures

1 / 2

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

It should look something like this:

```
eyJhbGciOiJSUzI1NiIsInR5cCgOiAiSldeIwiia2IkIa6ICJRa1RX2VwS2IwNvpFsKp3ZTdlcnFQUWtjSERNRi1SMnhGeE1tZUJ2aC1Vn0.eyJqdGkiOiYzJmZjQ5ZS01MDY4LTQ0MjQtYTrINS05MWU30Tk3MTM0TMIcJlEHaI0jE1NDcz0Tc1NsIiCil6MCwiaWF0IjoxNTQ3Mzk2NjUyMzJzDRhMy01MGm2LTQ3j0tYwUzZC050DdiYjA1ZTk4MzYiLCJ0eXAiOjJRCIsImF6cI61jViYzk0ZjzhIwiwYXV0af9baW11joxNTQ3Mzk10Dc2LCJzXNzaW9uX3N0YXR1IjoiZmExND1lOGItZDR1OS00OWJ1Tkt1YjAtOTYxM2ZmMGE1NWjkTiwiYVNyIjoiMCisInByZWZlcnJ1Zf91c2VybmfZS16ImV2YXz0ThAZXhbXbsZS5jb20iLCJ1bWFpBC16ImV2YXz0ThAZXhbXbsZS5jb20ifQ.DxqOTuAT1vAWGMMFFje-2GP05v517D-CDJpLXoCbVr9gNkPbB9KFvviyW8I616YX6B38rJAj5WTCXZkyfUaSruImoM7fmPlfRfxwNAGT221CqHTCpTx6vQcmhg5Po..._cy1Tv6dwEcrxkeKVvn16FKkk0CpEUgVamElEy5Vkmlo1X8Rc0TRKfdnNZF6AT9LnhDzN5npHH-20qR1sWtZyYGJuuuQmxA272R7QKZTxXCNPjZBOKPUachbDvHmCbn5YGPQUPKKbCQTN2z9ufngVjLX01kg-a-LWNpEP0dh1YxNg7mmgjxKKYV8bMUW80yB7WsvCgo
```

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

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 eyJhbGciOiJSUzI1NiIsInR5cCgOiAiSldeIwiia2IkIa6ICJRa1RX2VwS2IwNvpFsKp3ZTdlcnFQUWtjSERNRi1SMnhGeE1tZUJ2aC1Vn0.eyJqdGkiOiYzJmZjQ5ZS01MDY4LTQ0MjQtYTrINS05MWU30Tk3MTM0TMIcJlEHaI0jE1NDcz0Tc1NsIiCil6MCwiaWF0IjoxNTQ3Mzk2NjUyMzJzDRhMy01MGm2LTQ3j0tYwUzZC050DdiYjA1ZTk4MzYiLCJ0eXAiOjJRCIsImF6cI61jViYzk0ZjzhIwiwYXV0af9baW11joxNTQ3Mzk10Dc2LCJzXNzaW9uX3N0YXR1IjoiZmExND1lOGItZDR1OS00OWJ1Tkt1YjAtOTYxM2ZmMGE1NWjkTiwiYVNyIjoiMCisInByZWZlcnJ1Zf91c2VybmfZS16ImV2YXz0ThAZXhbXbsZS5jb20iLCJ1bWFpBC16ImV2YXz0ThAZXhbXbsZS5jb20ifQ.DxqOTuAT1vAWGMMFFje-2GP05v517D-CDJpLXoCbVr9gNkPbB9KFvviyW8I616YX6B38rJAj5WTCXZkyfUaSruImoM7fmPlfRfxwNAGT221CqHTCpTx6vQcmhg5Po..._cy1Tv6dwEcrxkeKVvn16FKkk0CpEUgVamElEy5Vkmlo1X8Rc0TRKfdnNZF6AT9LnhDzN5npHH-20qR1sWtZyYGJuuuQmxA272R7QKZTxXCNPjZBOKPUachbDvHmCbn5YGPQUPKKbCQTN2z9ufngVjLX01kg-a-LWNpEP0dh1YxNg7mmgjxKKYV8bMUW80yB7WsvCgo.

+ 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 lferrari@fedora-ho... Untitled Document... Pictures 1 / 2

Let's hit Test

The screenshot shows a browser-based API testing interface. At the top, there is a 'Request' section with 'Request Headers' containing standard HTTP headers like GET, Host, Accept, User-Agent, and Authorization. Below it is a 'Response' section with 'Response Headers' showing various HTTP headers including Set-Cookie, Date, Content-Type, and Cache-Control. The 'Response Body' section displays a JSON array of bank objects:

```
{
  "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": "at02-2089--01",
      "short_name": "Abanca",
      "full_name": "ABANCA CORPORACION BANCARIA, S.A.",
      "logo": "https://www.abanca.es/logo"
    }
  ]
}
```

And success!

The screenshot shows a browser-based API testing interface with a JSON response body. The response is identical to the one shown in the previous screenshot, listing five bank objects with their details and routing information.

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

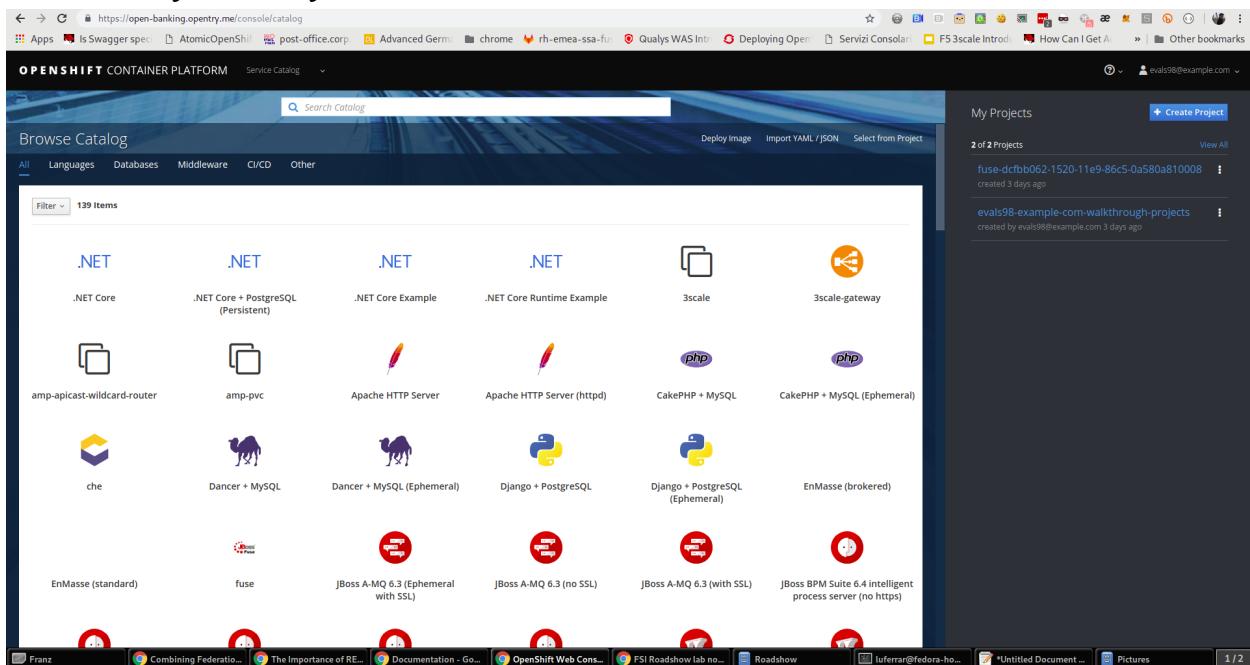
Checkpoint

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.



The screenshot shows the OpenShift Container Platform Service Catalog. At the top, there's a search bar labeled "Search Catalog". Below it, a navigation bar includes "OPENSHIFT CONTAINER PLATFORM", "Service Catalog", and a user profile icon. On the left, a sidebar lists categories: All, Languages, Databases, Middleware, CI/CD, and Other. A filter dropdown shows "139 Items". The main area displays a grid of project icons and names:

Category	Project Name	Description
.NET	.NET Core	
	.NET Core + PostgreSQL (Persistent)	
	.NET Core Example	
	.NET Core Runtime Example	
	3scale	
	3scale-gateway	
	amp-apicast-wildcard-router	
	amp-pvc	
	Apache HTTP Server	
	Apache HTTP Server (httpd)	
PHP	CakePHP + MySQL	
	CakePHP + MySQL (Ephemeral)	
	Django + PostgreSQL	
	Django + PostgreSQL (Ephemeral)	
	EnMasse (brokered)	
	EnMasse (standard)	
	fuse	
	JBoss A-MQ 6.3 (Ephemeral with SSL)	
	JBoss A-MQ 6.3 (no SSL)	
	JBoss A-MQ 6.3 (with SSL)	
Other	Roadshow	
	luferar@fedora-ho...	
	*Untitled Document ...	
	Pictures	
	Franz	
	Combining Federat...	
	The Importance of RE...	
	Documentation - Go...	
	OpenShift Web Cons...	
	FSI Roadshow lab no...	

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 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.

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

The screenshots show the OpenShift Cluster Console interface. The left sidebar contains navigation links for Home, Workloads (Pods, Deployments, Deployment Configs, Stateful Sets, Secrets, Config Maps, Cron Jobs, Jobs, Daemon Sets, Replica Sets, Replication Controllers, HPA), Networking, Storage, Builds, and Administration (Projects, Service Accounts, Roles, Role Bindings, Resource Quotas). The right pane lists several pods in the 'fuse-dcfbb062-1520-11e9-86c5-0a580a810008' namespace, including 'syndesis-operator-1-b5tlr', 'syndesis-prometheus-1-6dcfz', 'syndesis-server-1-jhmn', 'syndesis-ui-1-c6gc9', and 'todo-1-8bdk4'. Each pod entry shows its name, IP address (e.g., ip-172-31-0-6.ec2.internal), status (Running or Ready), and detailed configuration labels. A 'Delete Pod' dialog box is open in both screenshots, asking for confirmation to delete the 'syndesis-ui-1-c6gc9' pod. In the second screenshot, the dialog includes an error message: 'Are you sure you want to delete syndesis-ui-1-c6gc9 in namespace fuse-dcfbb062-1520-11e9-86c5-0a580a810008? pods "syndesis-ui-1-c6gc9" is forbidden: User "evals98@example.com" cannot delete pods in the namespace "fuse-dcfbb062-1520-11e9-86c5-0a580a810008": no RBAC policy matched'.

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

We can also open the relative Fuse Online installation.

The screenshot shows the OpenShift Container Platform Application Console interface. The left sidebar includes links for Overview, Applications, Builds, Resources, Storage, Monitoring, and Catalog. The main content area displays the 'syndesis' application with its deployment configurations and resource usage.

Deployment Config	Mib Memory	Cores CPU	Kib/s Network	Pods
DEPLOYMENT CONFIG broker-amq, #1	--	--	--	0
DEPLOYMENT CONFIG syndesis-db, #1	100	< 0.01	52	1
DEPLOYMENT CONFIG syndesis-meta, #1	340	< 0.01	0.1	1
DEPLOYMENT CONFIG syndesis-oauthproxy, #1	11	< 0.01	1.5	1
DEPLOYMENT CONFIG syndesis-operator, #1	16	< 0.01	1.8	1
DEPLOYMENT CONFIG syndesis-prometheus, #1	50	< 0.01	1.9	1
DEPLOYMENT CONFIG syndesis-server, #1	570	0.01	41	1
DEPLOYMENT CONFIG syndesis-ui, #1	--	--	--	0

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

The screenshot shows the OpenShift Container Platform Application Console interface. On the left, a sidebar lists navigation options: Overview, Applications, Builds, Resources, Storage, Monitoring, and Catalog. The main content area is titled "DEPLOYMENT CONFIG syndesis-oauthproxy, #1" and shows resource usage: 12 Mbit Memory, < 0.01 Cores CPU, 7.9 Kib/s Network, and 1 pod. Below this, other deployment configurations are listed: syndesis-operator, syndesis-prometheus, syndesis-server, and syndesis-ui. The syndesis-ui configuration is expanded, showing its container details: Image: fuse/fuse-ignite-ui@0c05f43 (100.0 MB), Ports: 8080/TCP, and resource usage: 5.7 Mbit Memory, 0 Cores CPU, 0.3 Kib/s Network, with Average Usage over the last 15 Minutes. A large blue circular icon on the right indicates 1 pod.

OPENSHIFT CONTAINER PLATFORM Application Console

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

Pods > syndesis-ui-1-c6gc9 created 3 days ago

Actions Add Storage Edit YAML Delete

Status

- Status: Running syndesis-ui #1
- 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 0c85f43 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

CONFIRM DELETE

Are you sure you want to delete the pod 'syndesis-ui-1-c6gc9'?

It cannot be undone. Make sure this is something you really want to do!

Delete pod immediately without waiting for the processes to terminate gracefully

Cancel **Delete**

OPENSHIFT CONTAINER PLATFORM Application Console

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

Pods > syndesis-ui-1-c6gc9 created 3 days ago

Actions Add Storage Edit YAML Delete

Status

- Status: Running syndesis-ui #1
- 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 0c85f43 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

CONFIRM DELETE

Are you sure you want to delete the pod 'syndesis-ui-1-c6gc9'?

It cannot be undone. Make sure this is something you really want to do!

Delete pod immediately without waiting for the processes to terminate gracefully

Cancel **Delete**

OPENShift CONTAINER PLATFORM Application Console

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

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 @e95f43 100.0 MiB
- Ports: 8080/TCP

Average Usage Last 15 Minutes

Mib Memory Cores CPU Kib/s Network

1 pod

DEPLOYMENT CONFIG syndesis-ui, #1

NETWORKING

Service - Internal Traffic syndesis-ui

Routes - External Traffic

Create Route

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



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

Franz Combinig Federatio... The Importance of RE... Documentation - Go... fuse-dcfbb062-1520... FSI Roadshow lab no... Roadshow luferar@fedora-ho... *Untitled Document ... Pictures 1 / 2

OPENSHIFT CONTAINER PLATFORM Application Console

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

DEPLOYMENT CONFIG

- syndesis-meta, #1
- syndesis-oauthproxy, #1
- syndesis-operator, #1
- syndesis-prometheus, #1
- syndesis-server, #1
- syndesis-ui, #1

CONTAINERS

syndesis-ui

- Image: fuse7/fuse-ignite-ui @e95f43 100.0 MB
- Ports: 8080/TCP

NETWORKING

Service - Internal Traffic
syndesis-ui

Routes - External Traffic
[Create Route](#)

Average Usage Last 15 Minutes

Mib Memory
Cores CPU
Kib/s Network

1 pod

RED HAT FUSE ONLINE

System Metrics

- 0 Integrations
- 5 Connections
- 11 Total Messages
- Uptime: Since Jan 13th 18:23 PM
2 days 19 hours 38 minutes

Create an Integration

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

[Create Integration](#)

Connections

- PostgresDB
- open-financial-service
- Log
- Timer

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

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>”