Module 3: The Modern API - Govern and Secure

Overview

This module consists of labs demonstrating how we can manage APIs. The Anypoint API Manager is an API policy management and governance tool that is integrated with the Mule Runtime. In these labs you will configure policies and SLAs that will allow you to manage security, quality of service, and compliance policies for your APIs, and monitor your API through API analytics.

With Anypoint API Manager, you can:

- Manage security and enforce policies
- Manage contracts associated with that API
- Monitor the API with analytics and KPI metrics in a dashboard view

The 4 labs in this unit are:

- Lab 1: Add your API to API Manager
- Lab 2: Apply Policies
- Lab 3: Monitor your API

Please proceed to Lab 1

Take me to the TOP

Lab 1: Govern your API

Overview

API management is essential to an API-Led architecture as it provides a governance framework to your APIs in all three layers.

One unique and powerful feature of MuleSoft is that, aside from creating API Proxies (Lab 4), we can apply governance directly to a MuleSoft implemented API. With governance directly on the service we eliminate the additional hop through an API gateway, and we provide last mile security directly on the service.

The API we will use to demonstrate this is an Omni Channel API that you deployed earlier.

We will use another unique feature of MuleSoft - API auto-discovery. The API auto-discovery capability is used in a Mule application to enable API management from Anypoint Platform on the application HTTP endpoint. When you run an app that incorporates auto-discovery, the auto-discovery process registers and starts the app in API Manager. Registration and start up occurs without user intervention. Your API will be discovered and governed whether it is running on your laptop, on-premises or in the cloud.

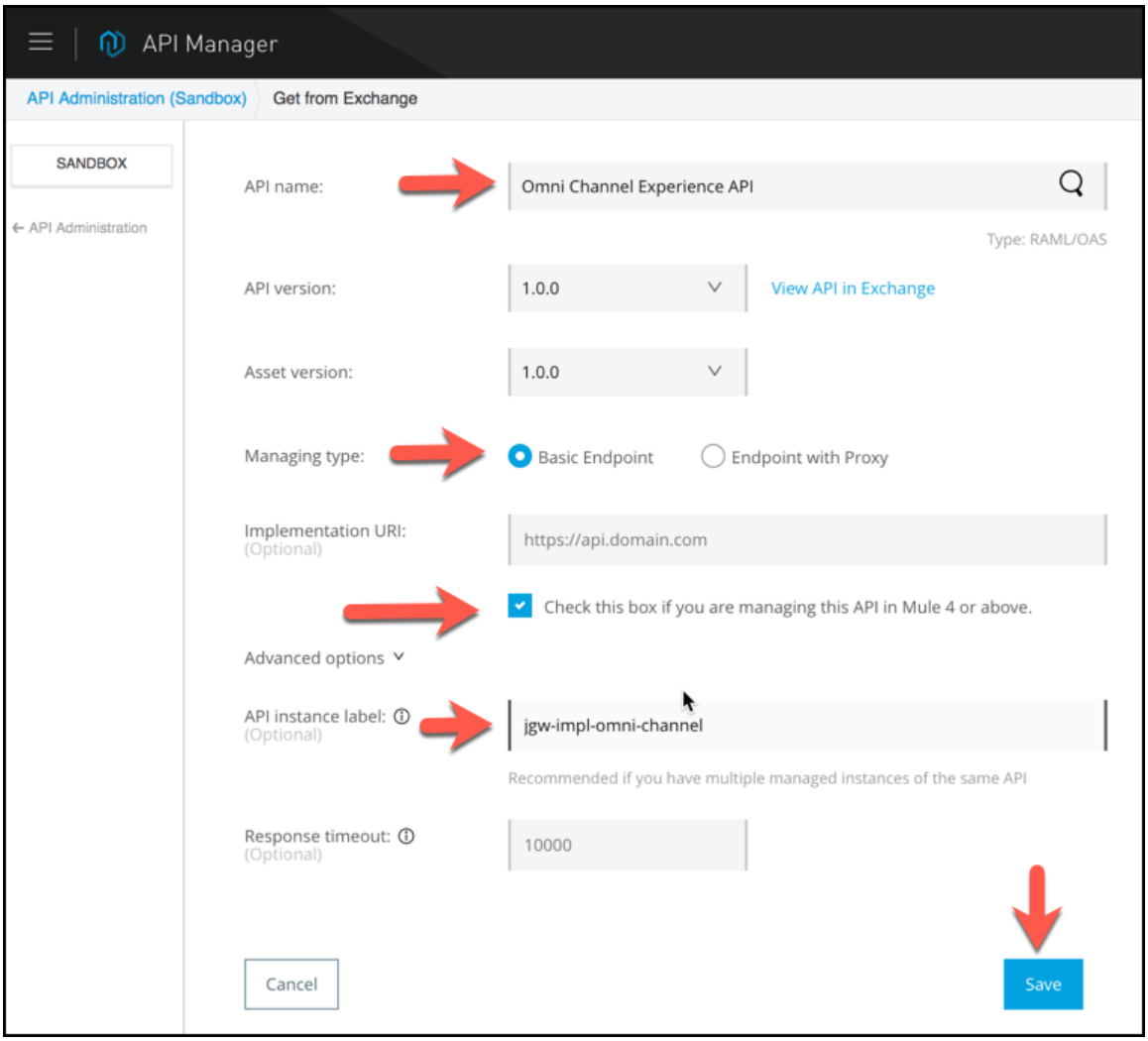
Clients will continue to access your deployed Omni Channel API through the same HTTP endpoint. API Manager will manage, control access and monitor the usage of the API, which we will look at in the succeeding labs.

Step 1: Create the API in the API manager

We are going to use the Omni Channel API to implement the API.

1. First go to the API Manager
2. Click on **Manage API** and select **Manage API from Exchange**
3. Configure the API with the following Configuration:
    a. **API name**: `Omni Channel Experience API`. Notice when you start to write the name, the field is autocompleted.
    b. **API version**: Select from the drop down list `1.0.0`.
    c. **Asset version**: Select from the drop down list `1.0.0`.
    d. **Managing type**: Select `Basic Endpoint`.
    e. **Implementation URI**: leave it blank.
    f. Advanced options

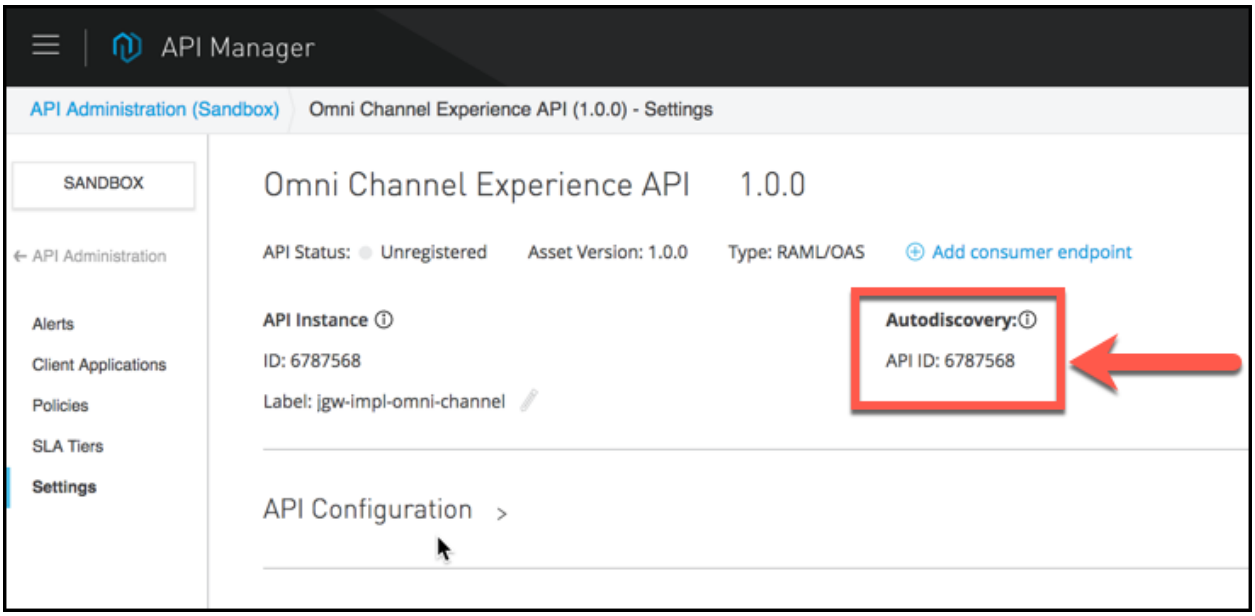
i. **API instance label**: `<username>-impl-omni-channel`. This Parameter is going to be use to identify the API in the API Manager.



4. Press **Save**

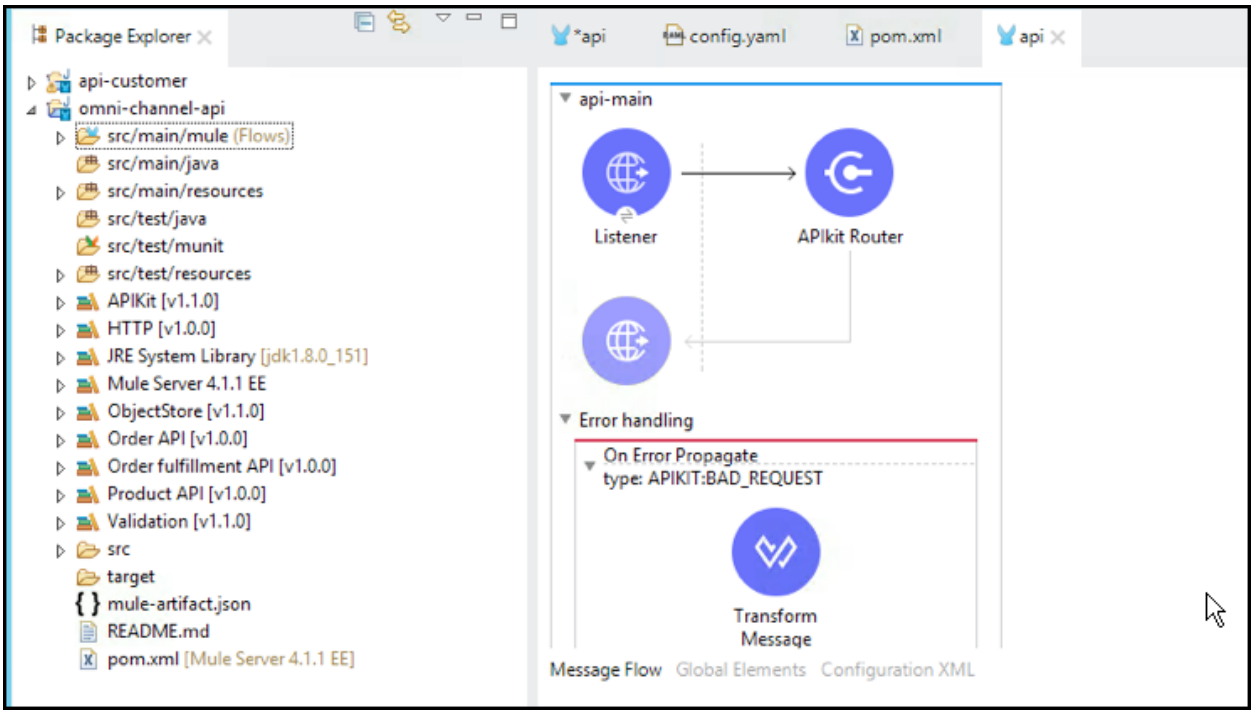
After pressing the save button, a screen with the API configuration will appear. Take note of the

API ID. It will be used to configure the autodiscovery component.



Step 2: Return to the complete Omni Channel Project in Anypoint Studio

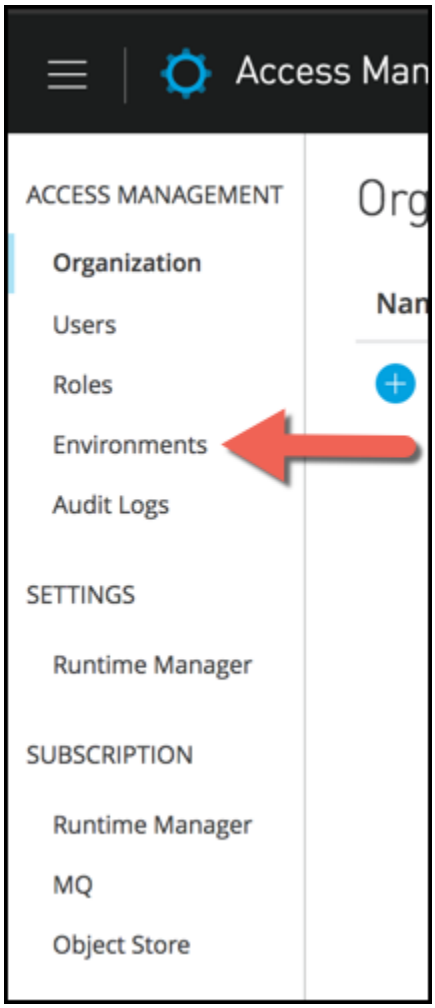
1. In Module 2, Lab 2 you downloaded the complete Omni Channel Project in Anypoint Studio. Open that project in Studio.



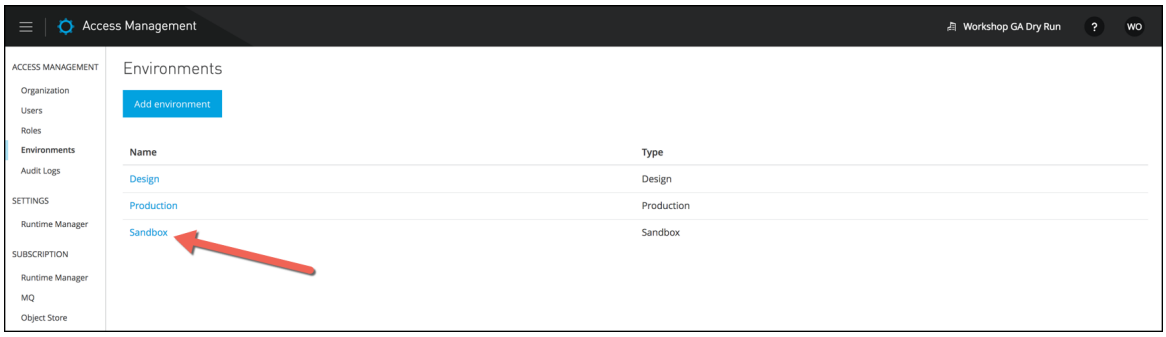
Step 3: Enable API Pairing with the Anypoint Platform

Your application must have the correct client credentials for your organization in order to be auto-discovered.

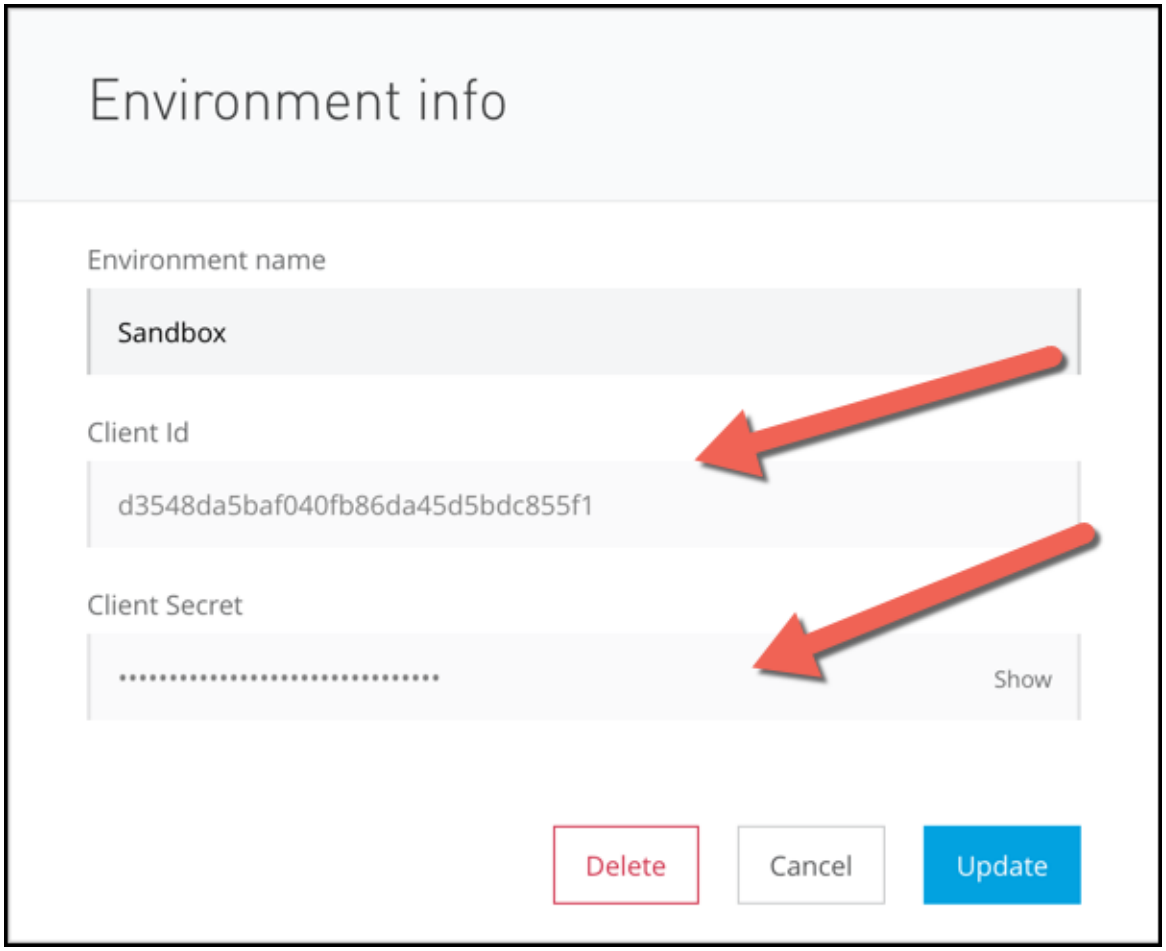
1. On the Anypoint platform select **Access Management**
2. Select **Environments** from the left menu



3. Select **Sandbox**

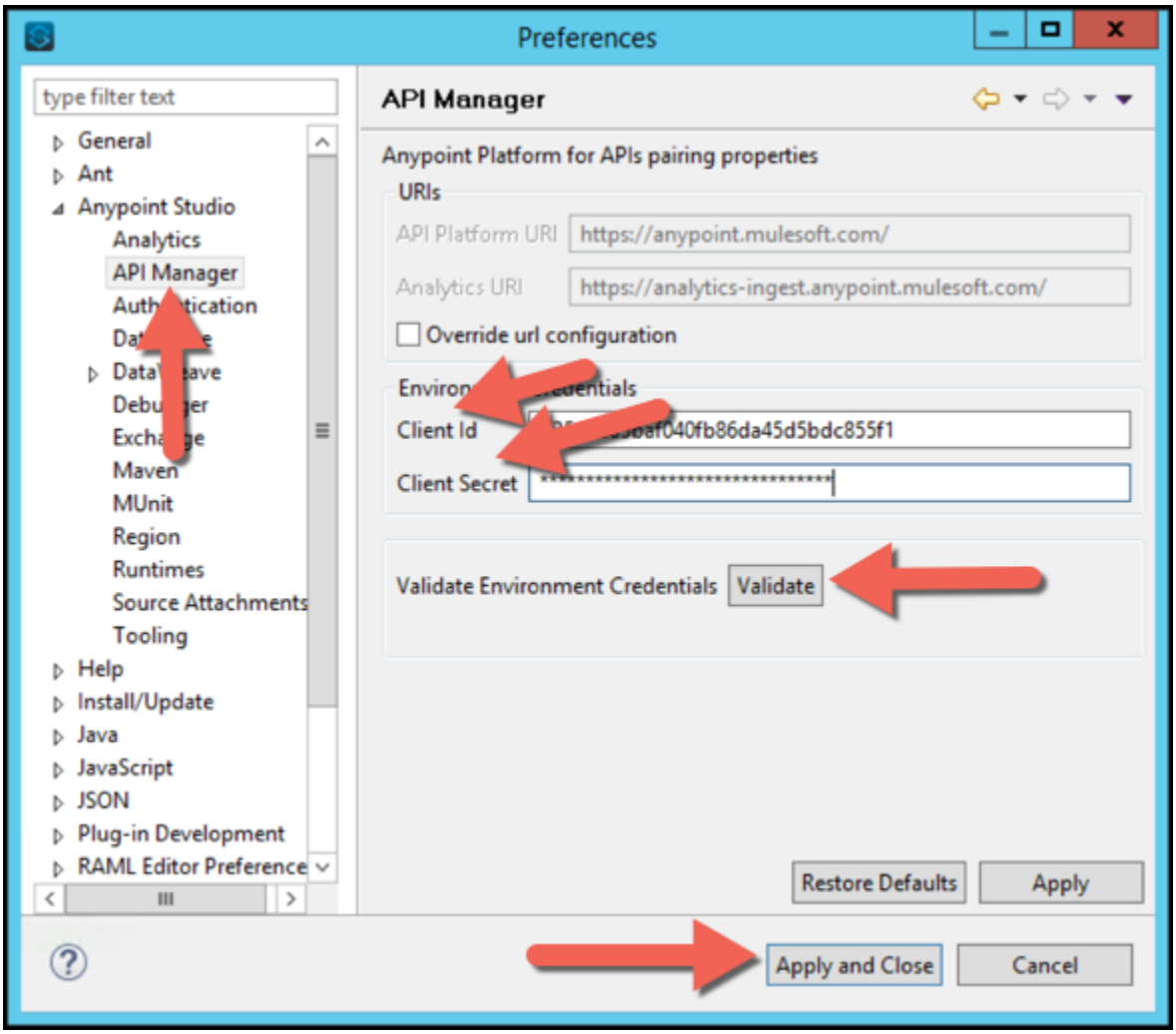


4. You use the **client_id** and **client_secret** in Studio for API pairing. To show the client_secret click on **Show**.



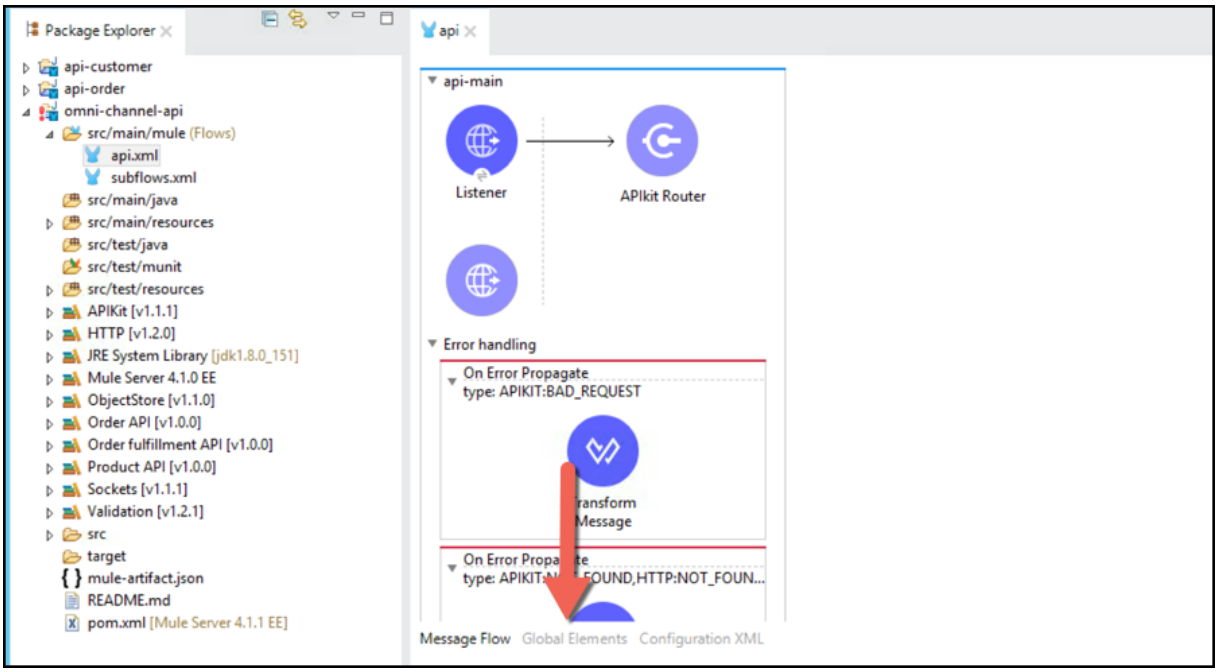
5. Back in Anypoint Studio select **Preferences** from the main menu (on a Mac it's under **Anypoint Studio-Preferences**. In Windows it's under **Window-Preferences**). Search for and select **API Manager**. Copy and paste the **client_id** and **client_secret** in the client credentials fields. Check

that they are correct by clicking **Validate**. Click **Apply and Close**.

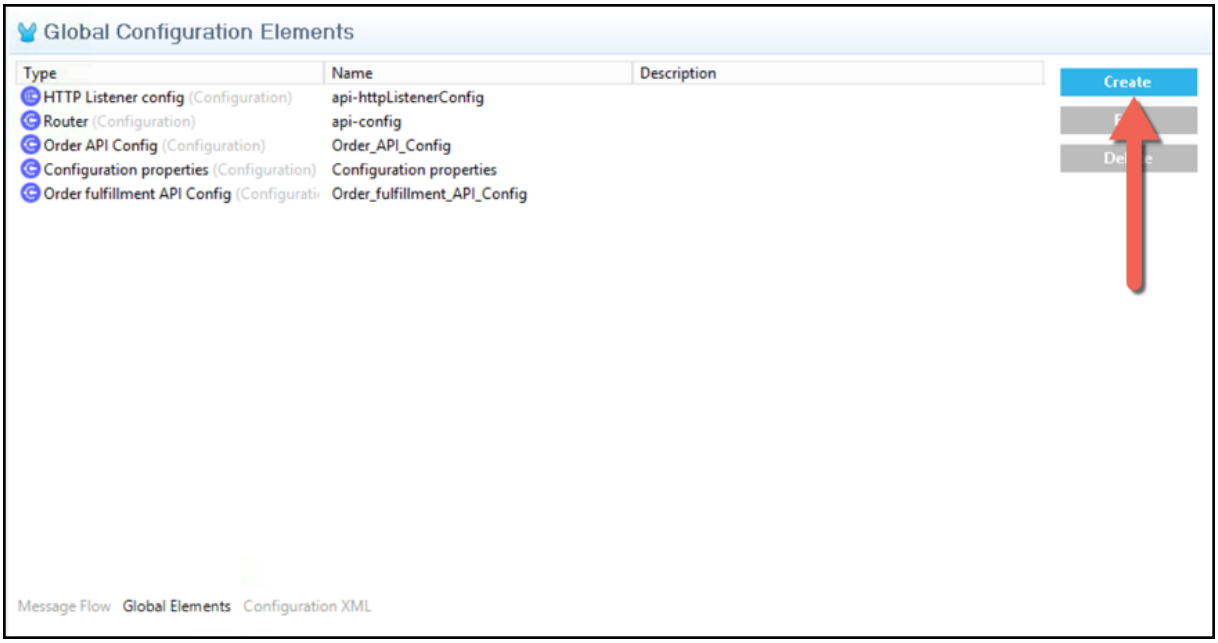


Step 4: Create the API Autodiscovery Global Element

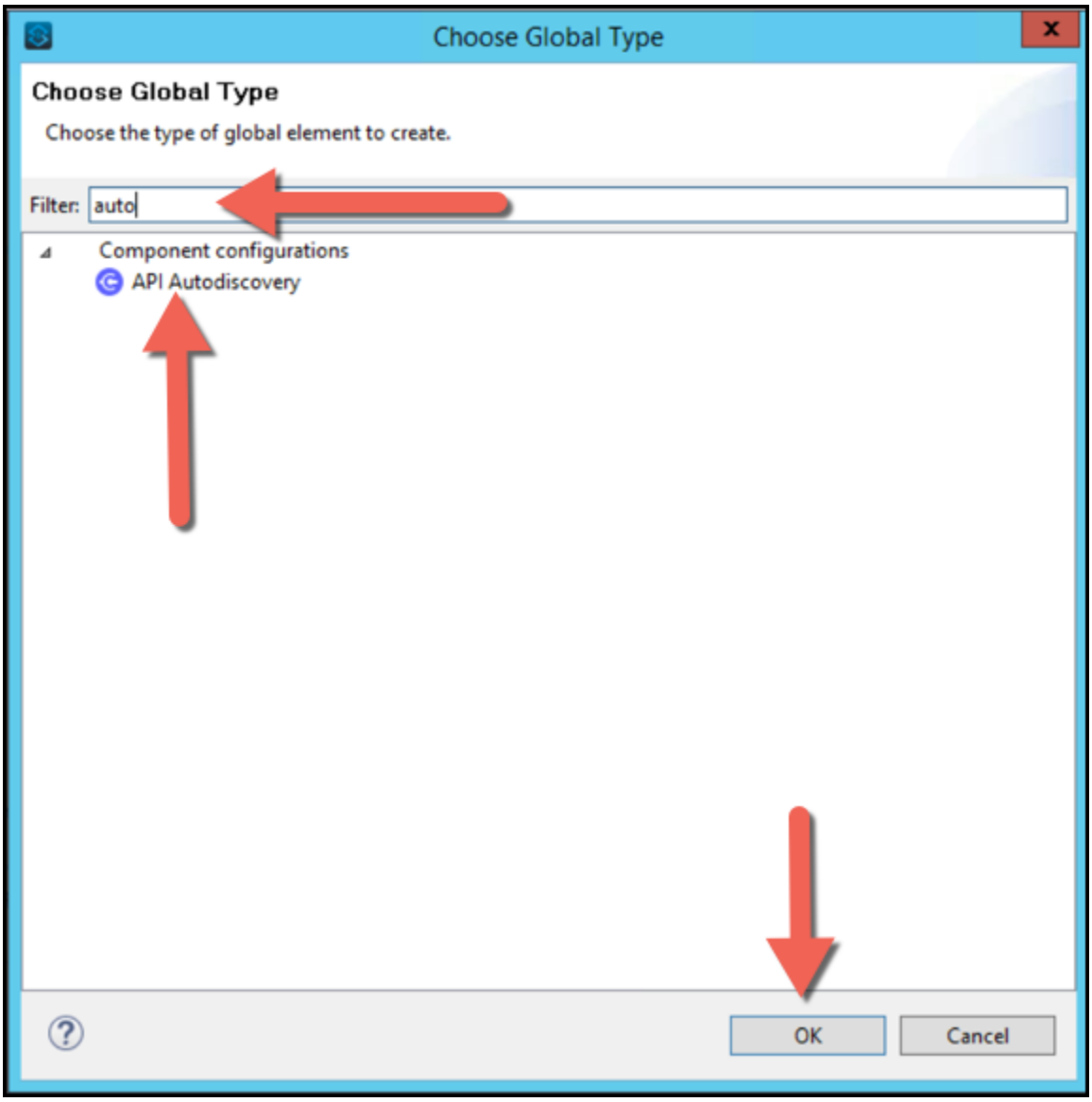
1. Click on the **Global Elements** tab just at the bottom of the Canvas.



2. Click on **Create**



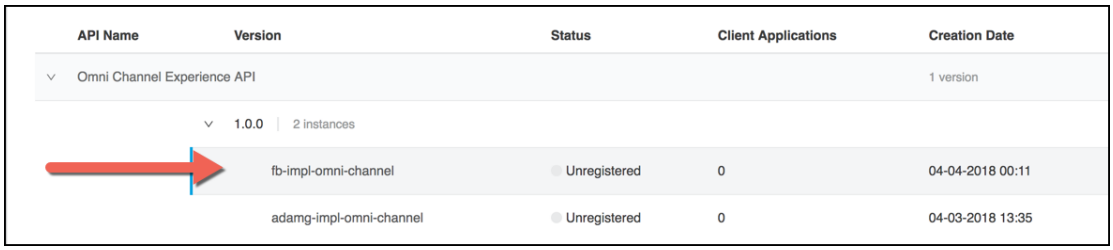
3. Using the **Filter**, search for **auto** and select **API Autodiscovery** and click **OK**



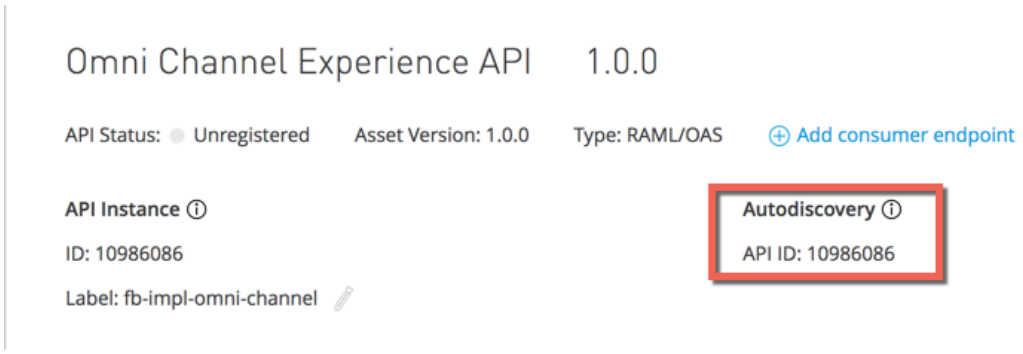
4. Fill in the **API Name** and **API Version**.

   To get this value, we need to get the information from the API Manager.

   a. Go to the API Manager and look for the API you created on the Step 1.

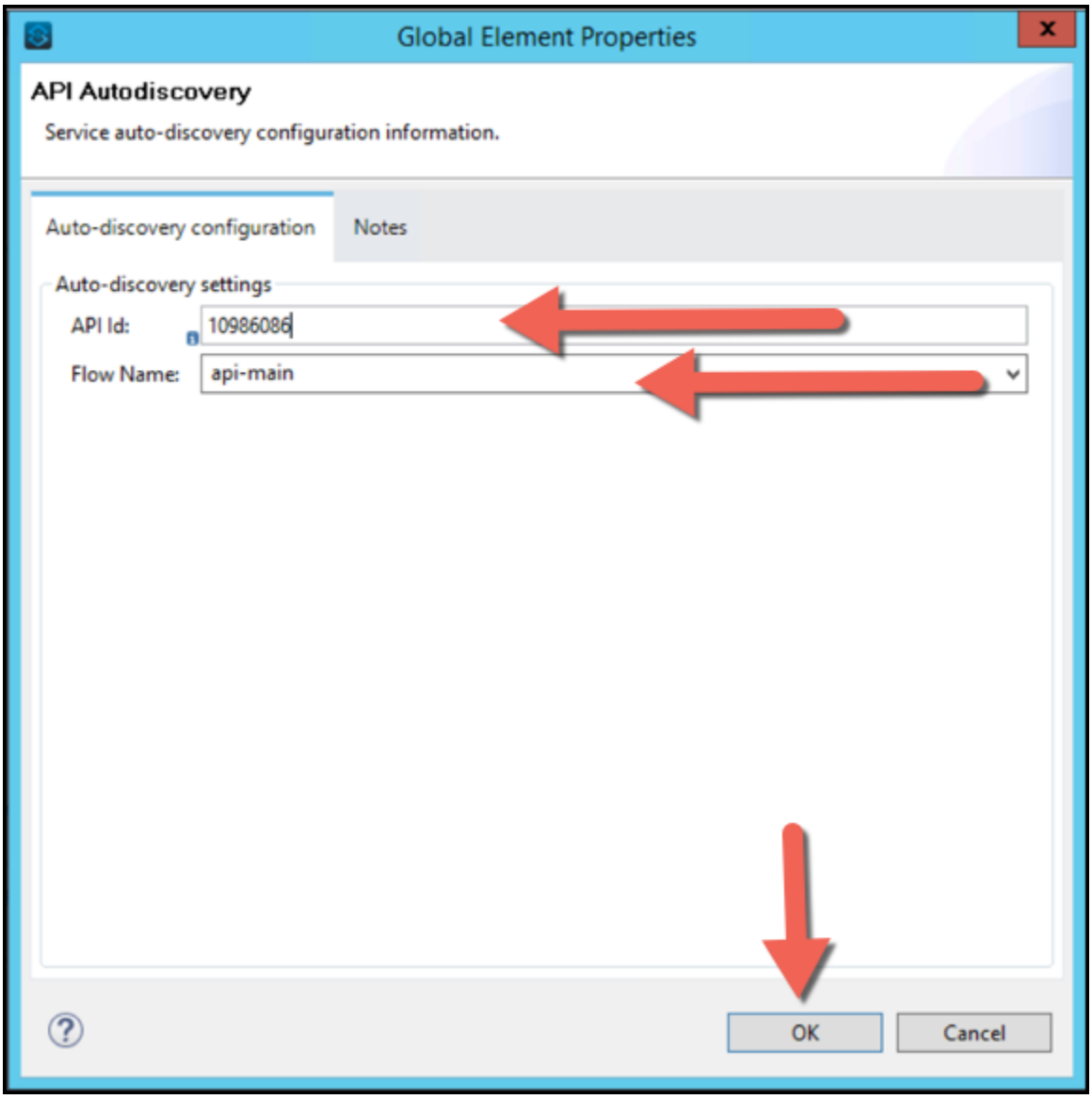


b. Take note of the **API ID**

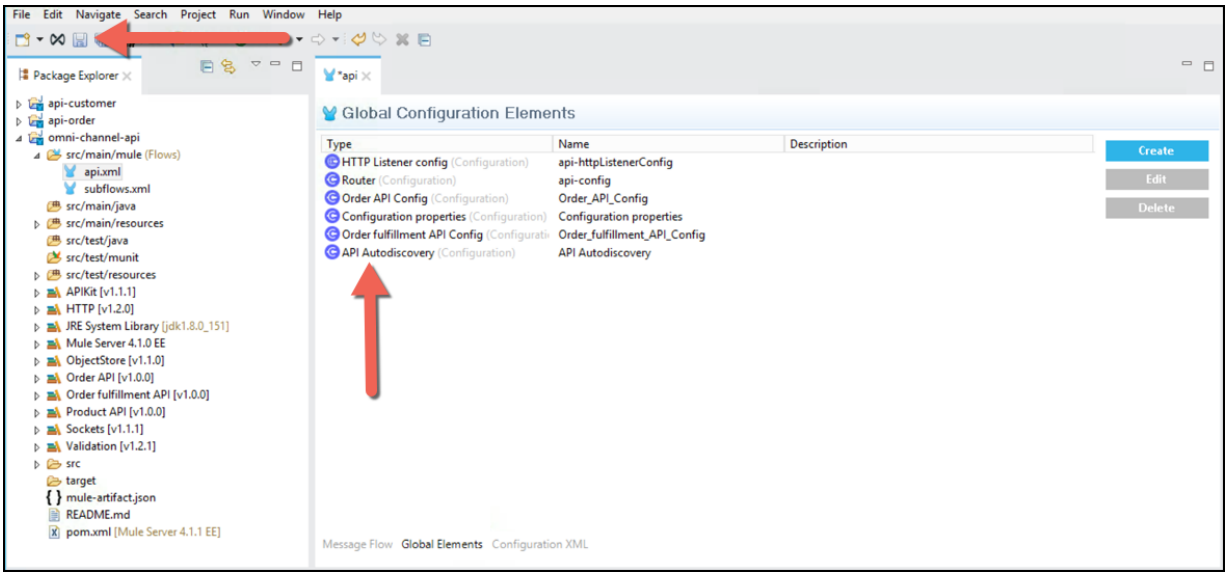


5. Complete the **API ID** with the API ID obtained in the last step.
6. Select **api-main** for **Flow Name**

7. Click **OK**

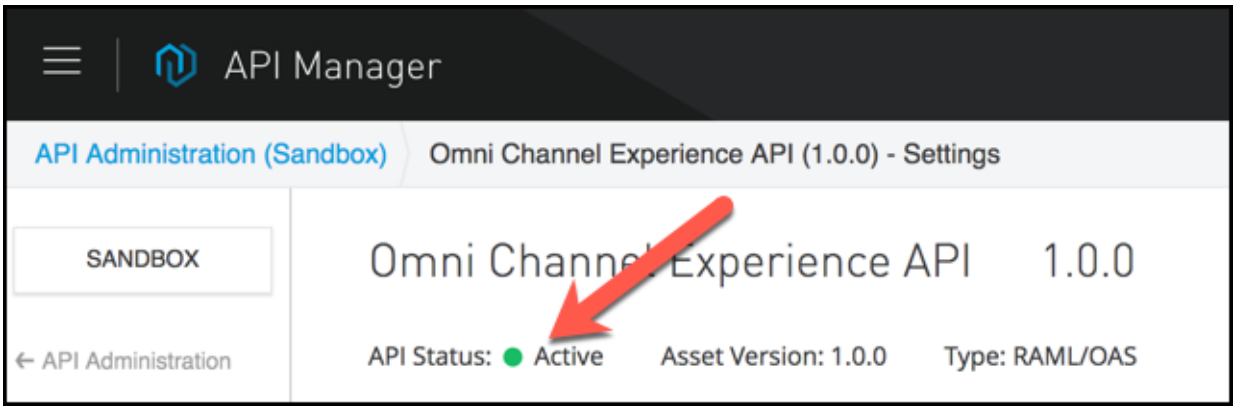

Global Element Properties
API Autodiscovery
Service auto-discovery configuration information.
Auto-discovery configuration
Notes
Auto-discovery settings
API Id:
10986086
Flow Name:
api-main
OK
Cancel

You should see the new **API autodiscovery** global element. **Save** your project.



Step 5: Test that the Application is Pairing Correctly with API Manager

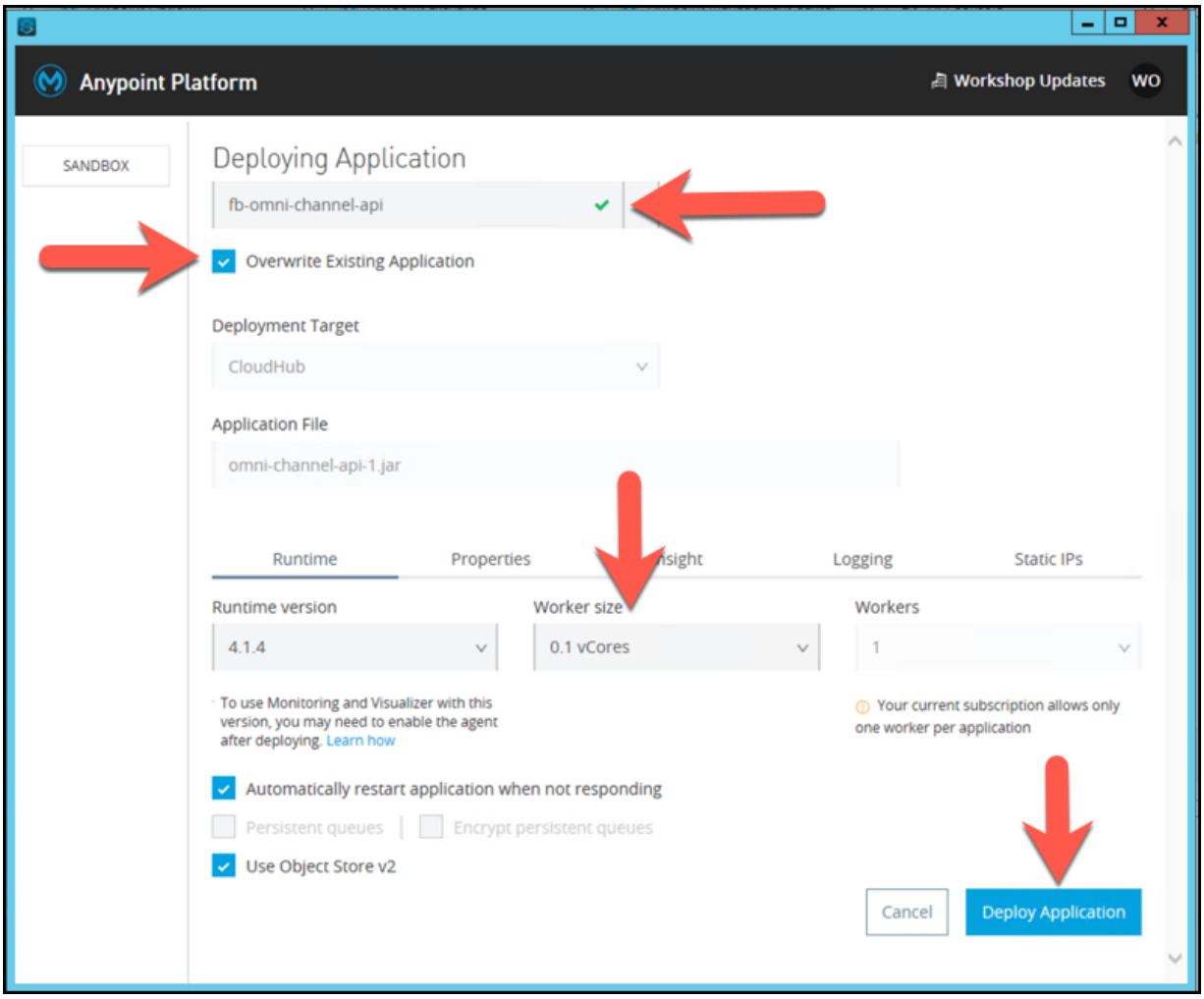
1. Start your application in test mode. See Module 2, Lab 2, Step 4 if you need a reminder on how to do this. After your application has started correctly, go to **API Manager** and see if your **API Status** is now **Active**. This may take a minute.



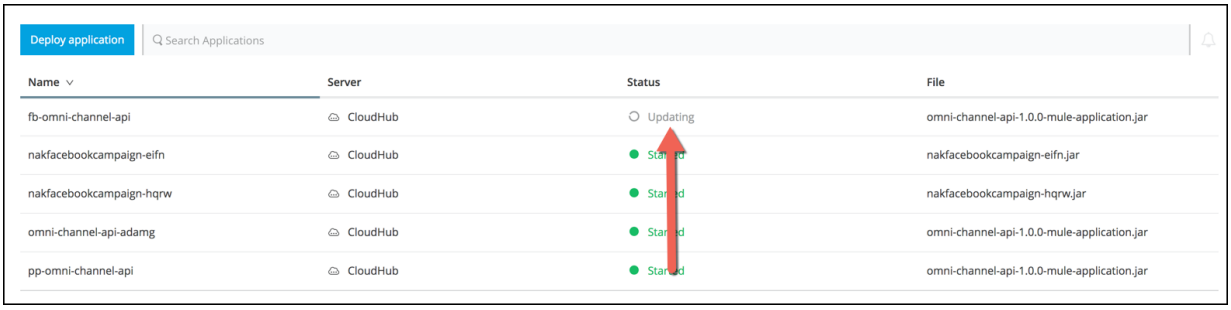
2. Congratulations! Your Omni Channel application will now be governed by **API Manager** whether it's running in test mode, on-premise or in the cloud.

Step 6: Redeploy Your API Omni Channel Application

1. Redeploy your application using the same steps as you did in Module 2, Lab 3. Note that you should see and check **Overwrite Existing Application** when re-deploying.



2. Ensure that your application has successfully redeployed using Anypoint **Runtime Manager**. The status should go from **Updating** to **Started**



When you see **Your application is started** you can continue.

3. Check in **API Manager** again that the API Status is Active (as in Step 4). You should see the green Deploy Successful in the status bar.

Summary

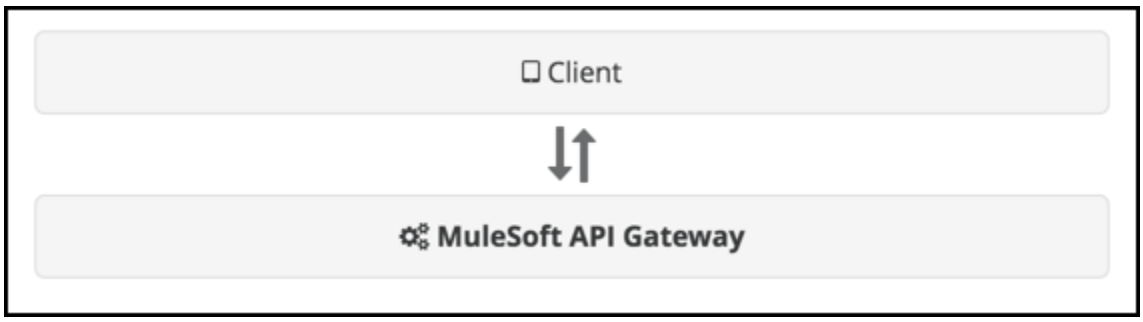
In this lab, we completed the following steps:

- Overview
- Step 1: Create the API in the API manager
- Step 2: Return to the complete Omni Channel Project in Anypoint Studio
- Step 3: Enable API Pairing with the Anypoint Platform
- Step 4: Create the API Autodiscovery Global Element
- Step 5: Test that the Application is Pairing Correctly with API Manager
- Step 6: Redeploy Your API Omni Channel Application

We saw how easy it is to enable your applications to be auto-discovered. This provides a **simplified** approach to manage MuleSoft impLemented APIs across your environment, whether on-premises or in the cloud. You are able to secure and govern your MuleSoft applications without an additional hop through a separate API Gateway. This also provides "last mile security" as policies are applied directly at the engine level. In this lab, we saw how we can deploy our API to CloudHub which can significantly **speed up your deployment** without having to manage and maintain infrastructure.

For further reading on deploying an API gateway and proxying your API, please refer to the following documentation:

- API Auto-Discovery



Congratulations! You have completed Lab 1.

Please proceed to Lab 2.

Take me to the TOP

Lab 2: Apply Policies

Overview

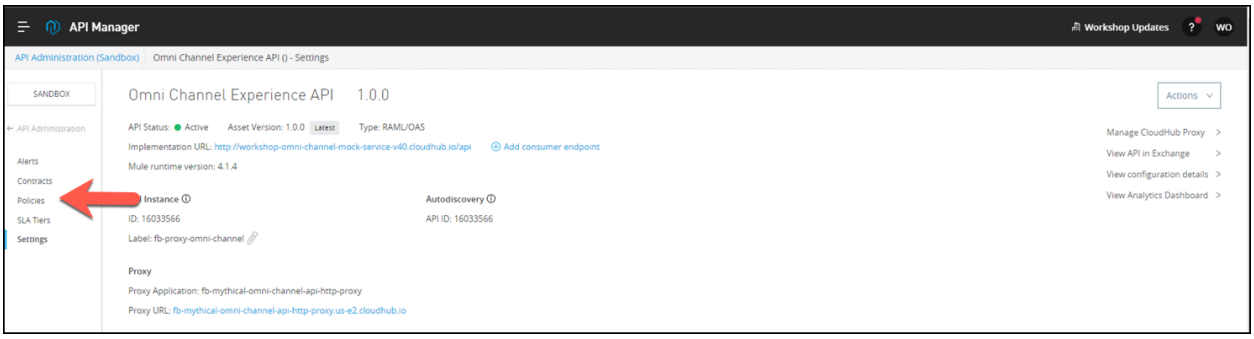
API Manager is an API policy management and governance tool that is integrated with the Mule runtime. In this lab, you will utilize the secure connection between the Omni Channel API on runtime manager and API manager to configure policies that will allow us to manage security, quality of service, and compliance policies for your APIs.

In last lab we deployed an autodiscovered Omni Channel API on CloudHub, now you can: In this lab you will apply policies to your API. **API Manager** provides many out-of-the-box (OOTB) policies related to compliance, quality of service and security. In addition, you can add your own policies.

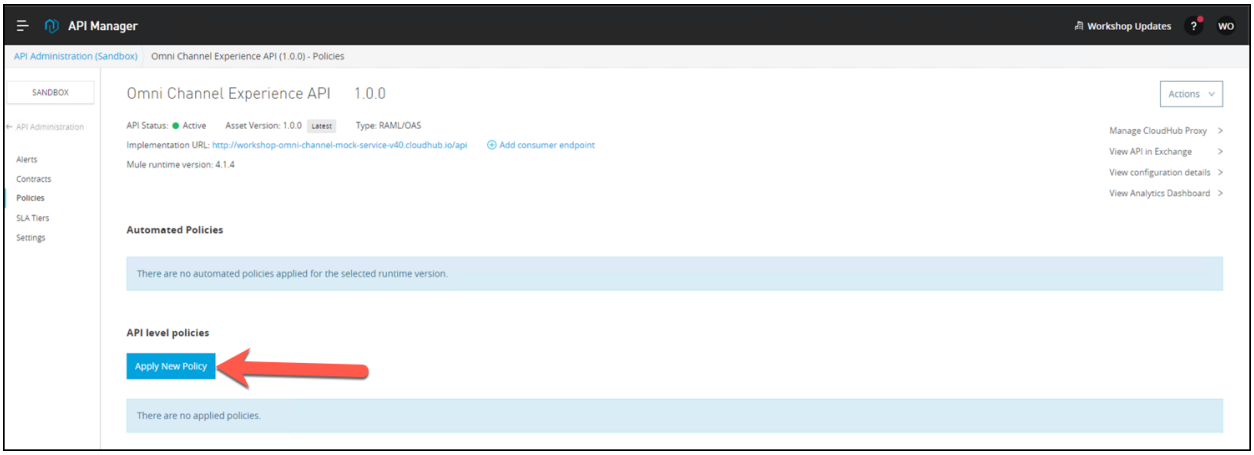
Step 1: Apply Rate Limiting Policy

To test policy management, you will add a **Rate Limiting** Policy to the API.

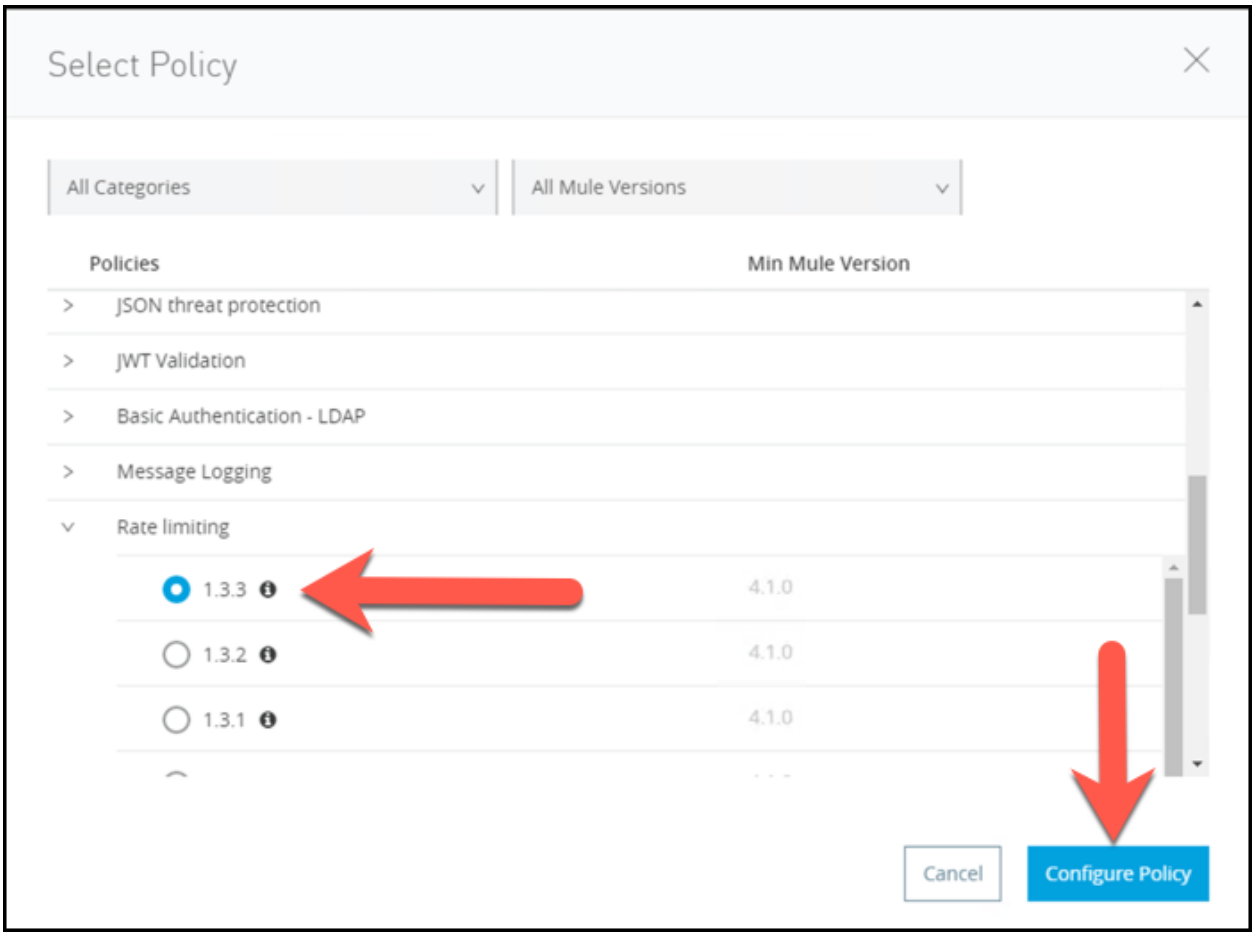
1. Go to the Omni Channel API Administration page.
2. Click the Policies tab under the API definitions.



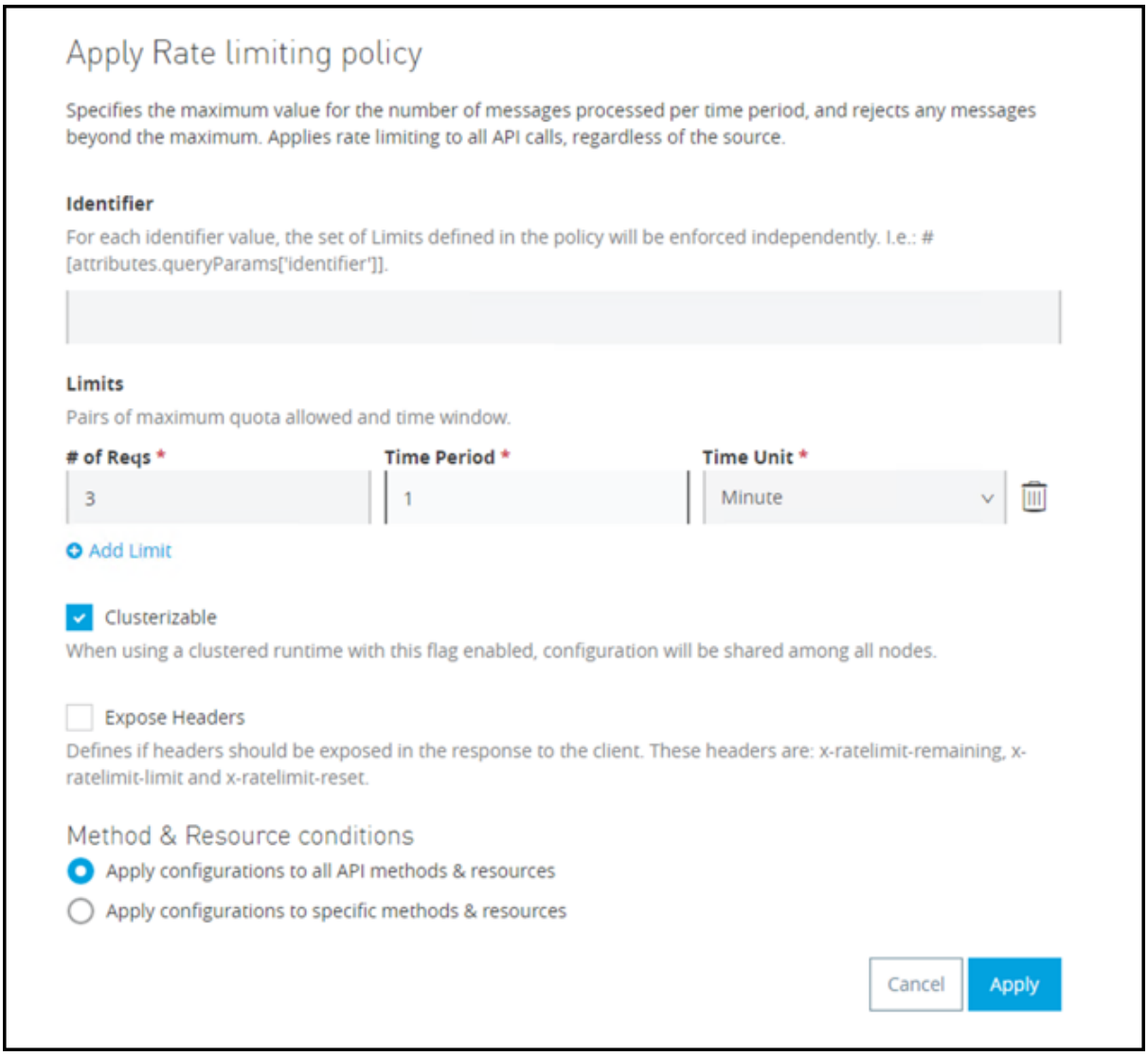
3. Click the **Apply New Policy** button.



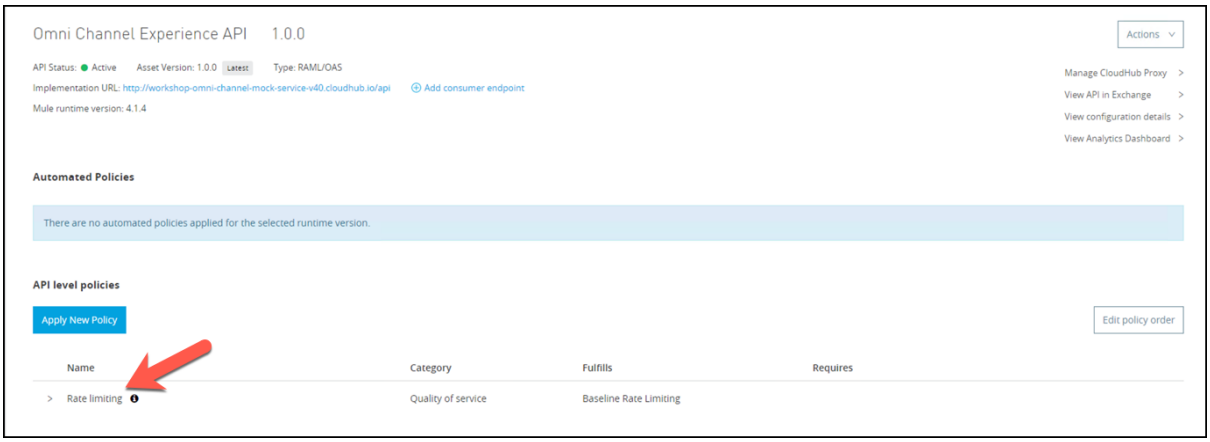
4. Click on the **Rate Limiting** Policy, select the version and **Configure Policy**.



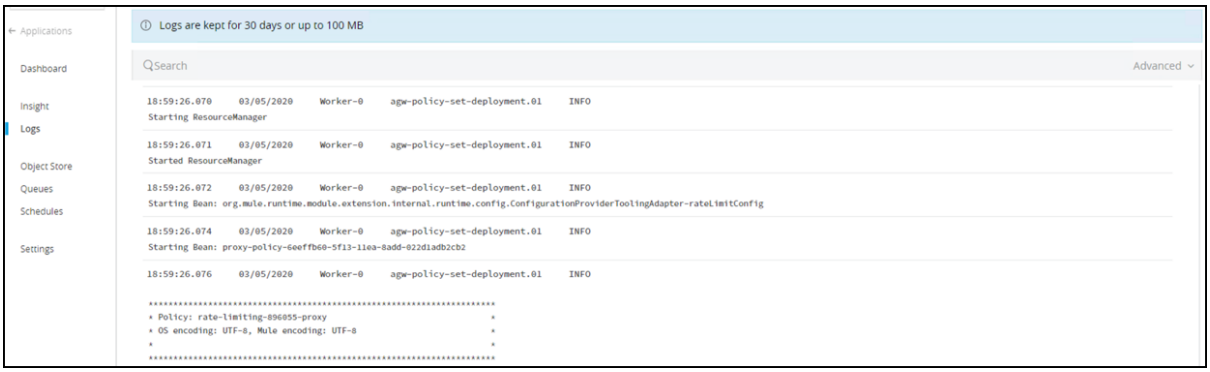
5. Enter a maximum requests of 3 per 1 Minute as shown and click **Apply**.



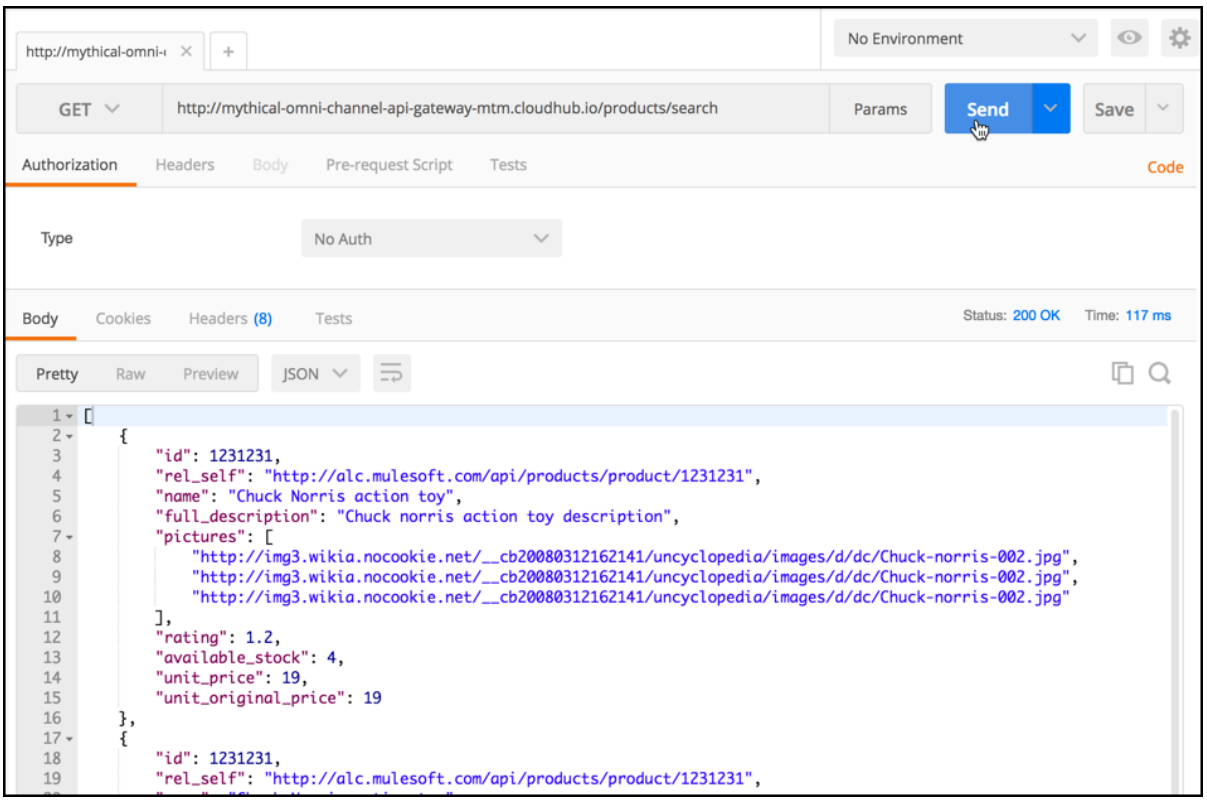
6. You should see the Rate limiting policy now under Applied Policies.



7. Wait for a minute (the Mulle Runtime receives policy updates every 60 seconds by default). You can also verify that your API has received the new policy by looking at the logs of your deployed application. Look under the logs tab in **Runtime Manager** for your application. Look for a log messages similar to the following **com.mulesoft.module.policies.**:

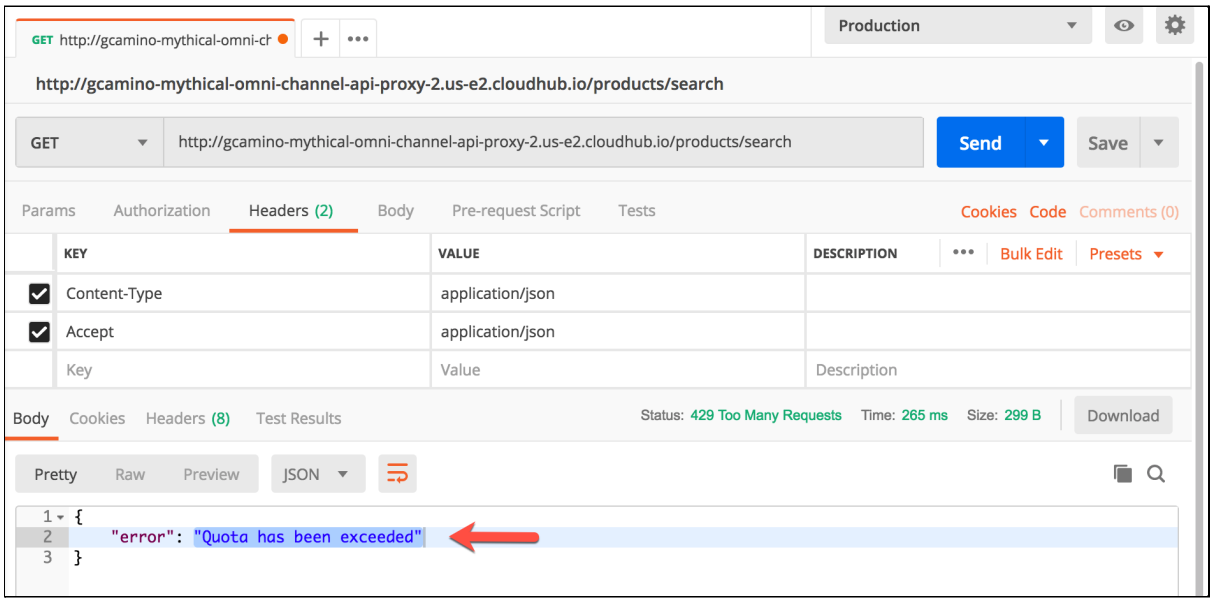


8. Test the API using your browser or using ARC or Postman and access your URL adding /products/search at the end (e.g. http://fb-mythical-omni-channel-api.us-e2.cloudhub.io/api/products/search).

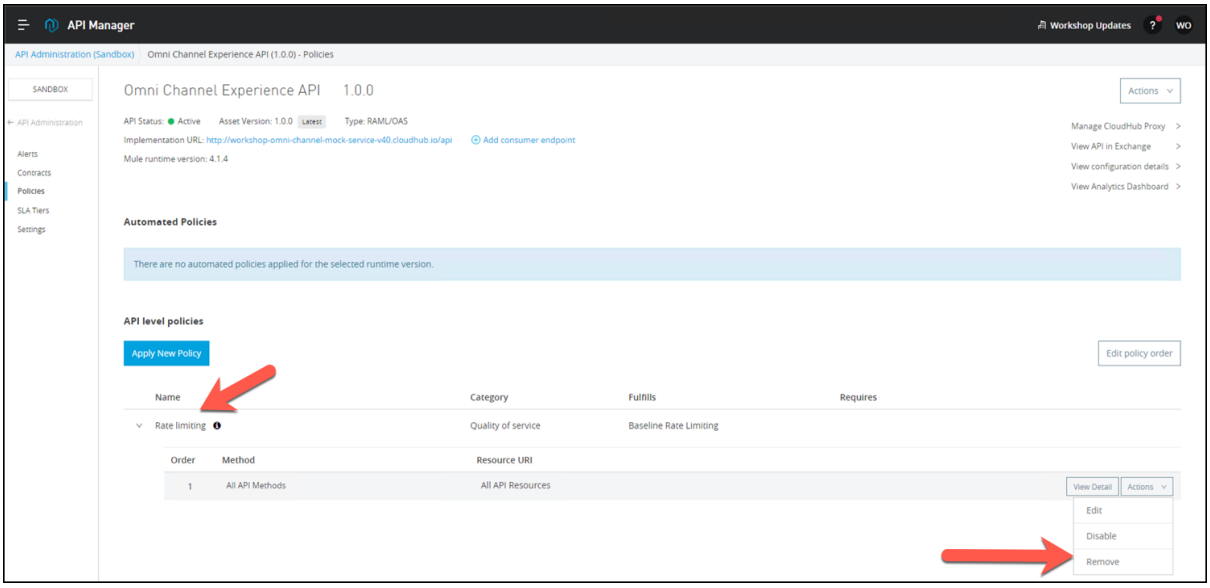


9. Test it again by reloading the page **3 times**.
10. On the 3rd invocation you will get a message indicating **Quota has been exceeded**. This demonstrates your Rate Limiting policy has been applied. (Note: If you do this with ARC, you will

see the X-RateLimit headers)



11. **Remove** the **Rate Limiting** Policy after your test by clicking the Remove button.



Step 2: Create SLA Tiers

In the API manager access can be based on SLA Tiers set by the API owner. However, this is optional. As we demonstrated in the previous step, it is possible to provide access without any SLA tiers. An API owner can establish pre-defined SLA tiers that consumers can view and choose from when they request access to the API. If no SLA tiers are defined for an API, the application owner can request access without an SLA tier.

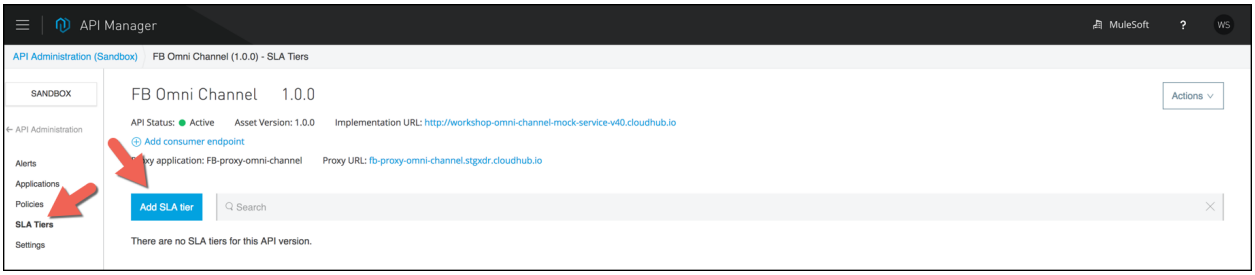
Let's define a new SLA tier for your API version.

1. Click the **SLA Tiers** link in the left toolbar of your API Version Details page

You are going to setup 3 SLA tiers:

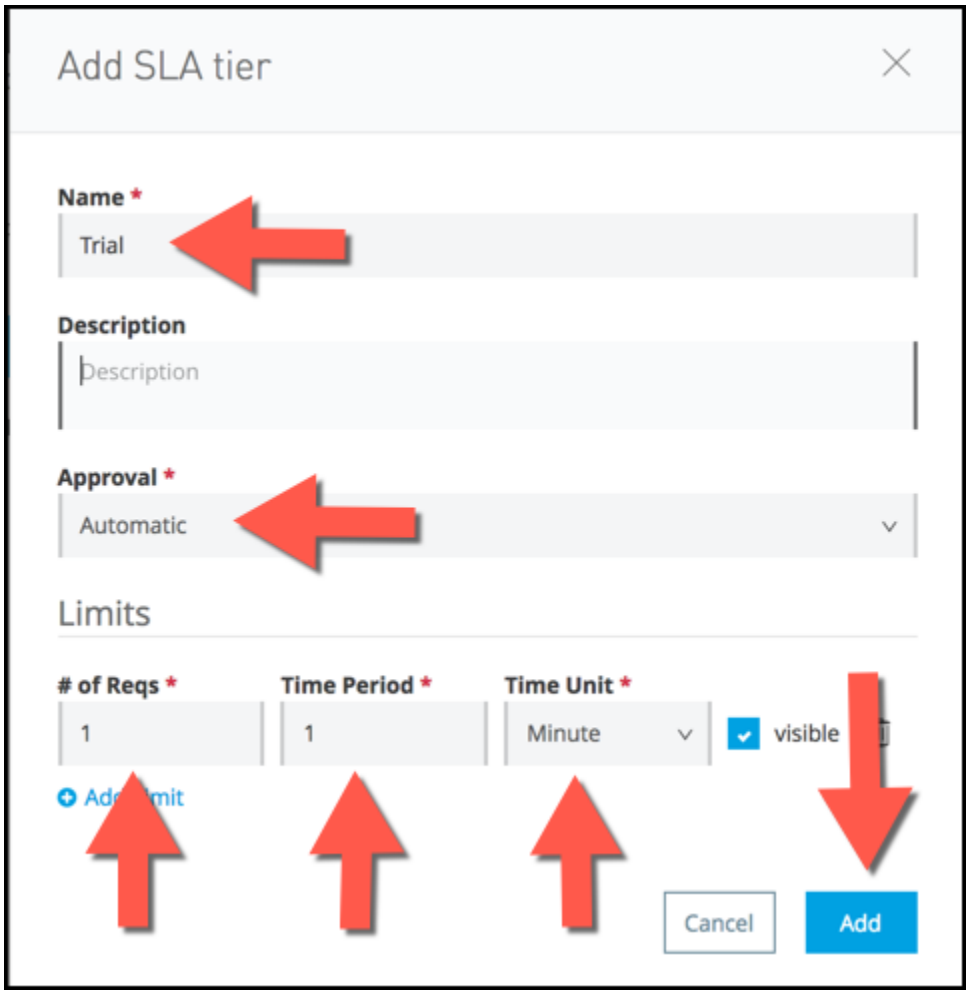
Tier	Approval	Throughput	Period
Trial	Automatic	1	Minute
Gold	Manual	10	Second
Platinum	Manual	100	Second

2. Click **Add SLA tier**.

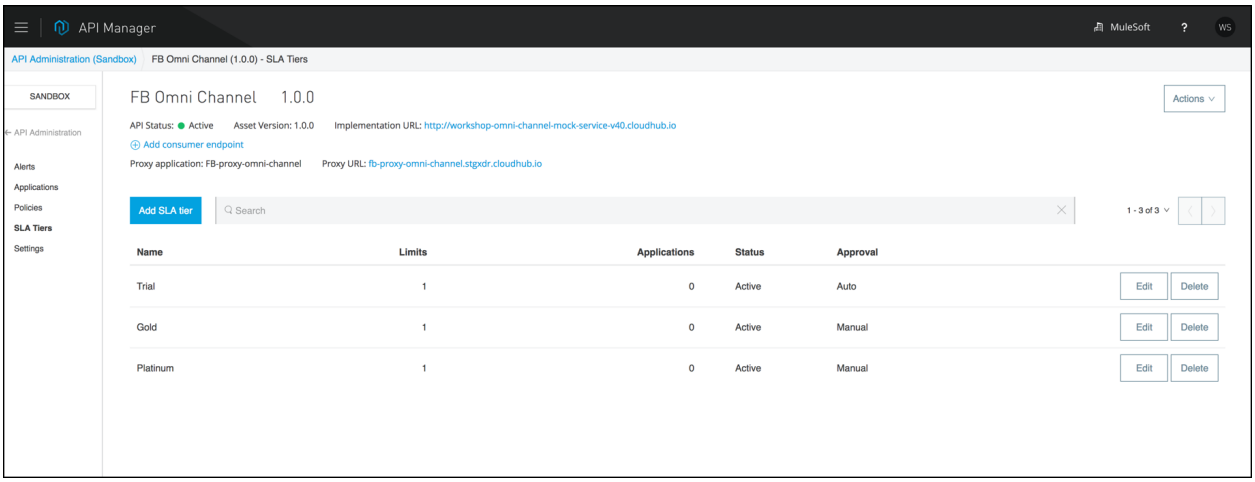


3. Fill in the fields to configure your tier
   - Give the tier a **Name**
   - Define the **Limits** by indicating the number of requests per time period that are allowed,
   - Indicate whether application access **Approval** at this tier level should be automatically approved or require manual approval.
   - Click **Add** to save your tier.

- *Repeat for all the tiers shows above.*



4. Your SLA tier is displayed with all of the information that you just defined. In addition, you have a column to indicate how many applications are registered on that tier. You can also edit or delete the tier using the links in the row.

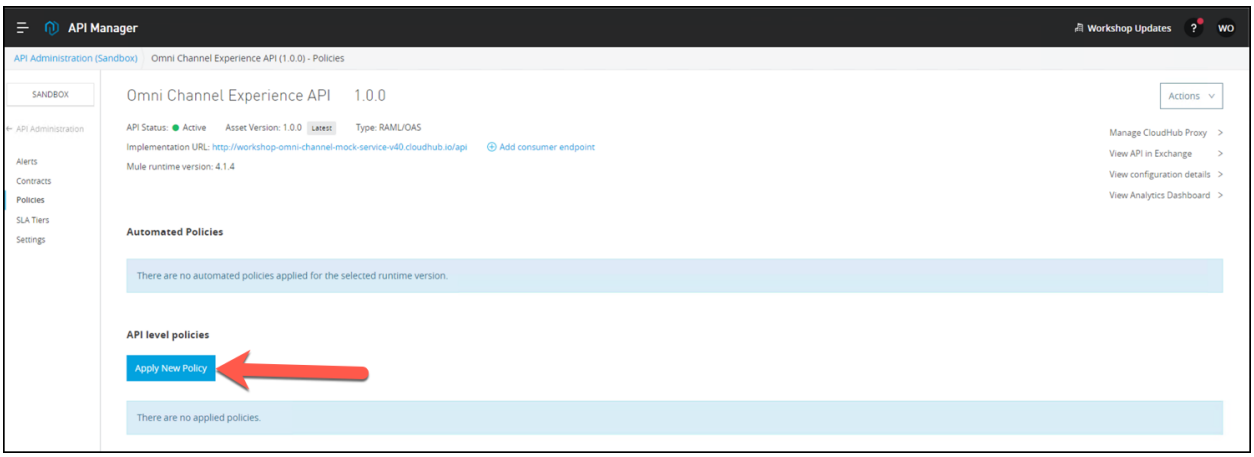


Step 3: Add a Rate Limiting SLA-based Policy

To enforce **SLA tiers**, you need to apply a rate-limiting or throttling policy that is SLA-based. These policies require all applications that consume your API to register for a specific tier. Their client credentials will be required for each API call so that Anypoint Platform can properly enforce the contracted tier.

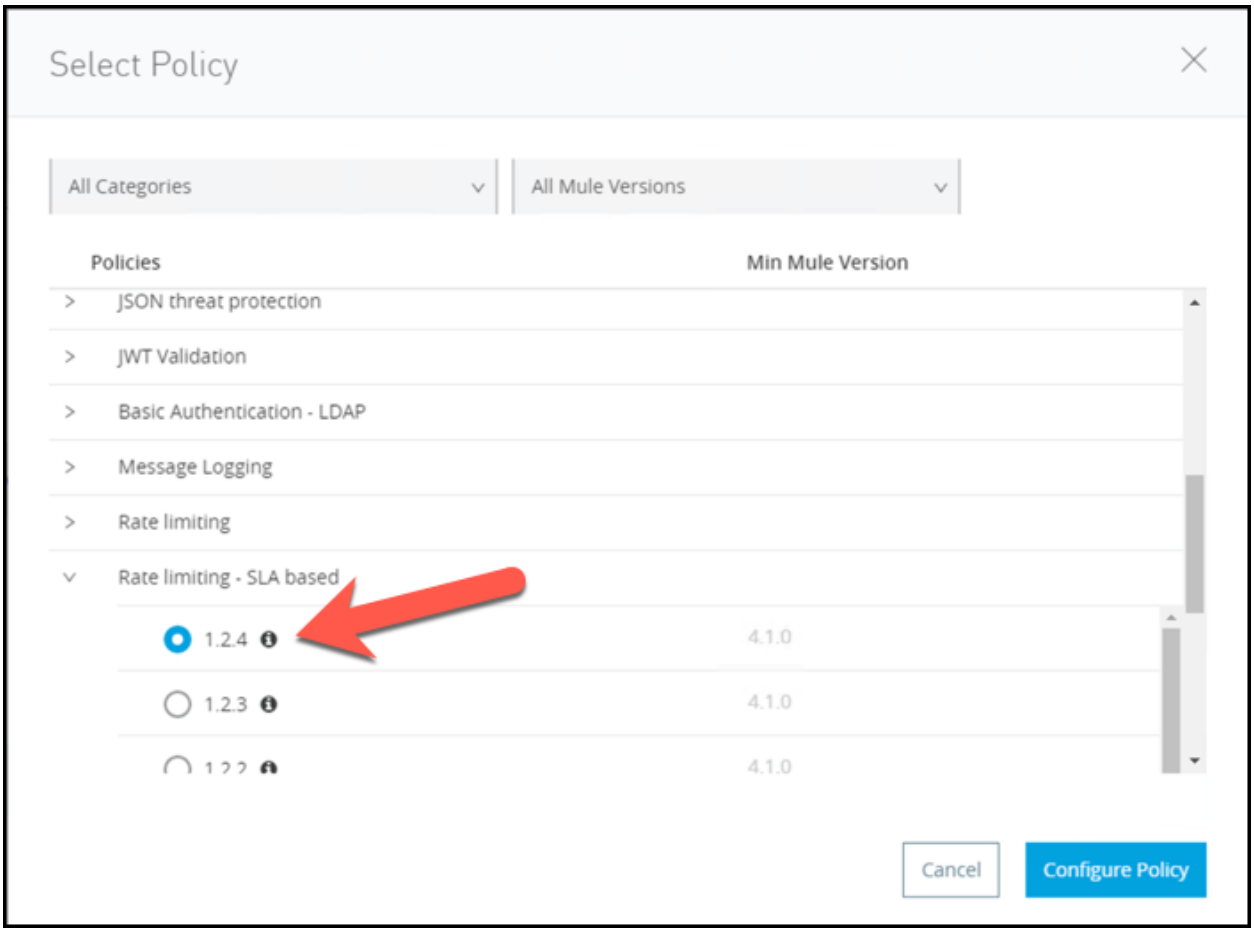
Let's apply a rate-limiting policy to your endpoint.

1. Click the **Policies** link in the left menu and **Apply New Policy** button to view the list of available policies for your organization.



2. Hover over the small **i** beside individual policies to read their descriptions in the pop up bubbles. Policies can be filtered by **Category** and **Fulfills** in the drop down boxes across the top.

3. Click on the Rate limiting - SLA based policy, select version and click **Configure Policy**.

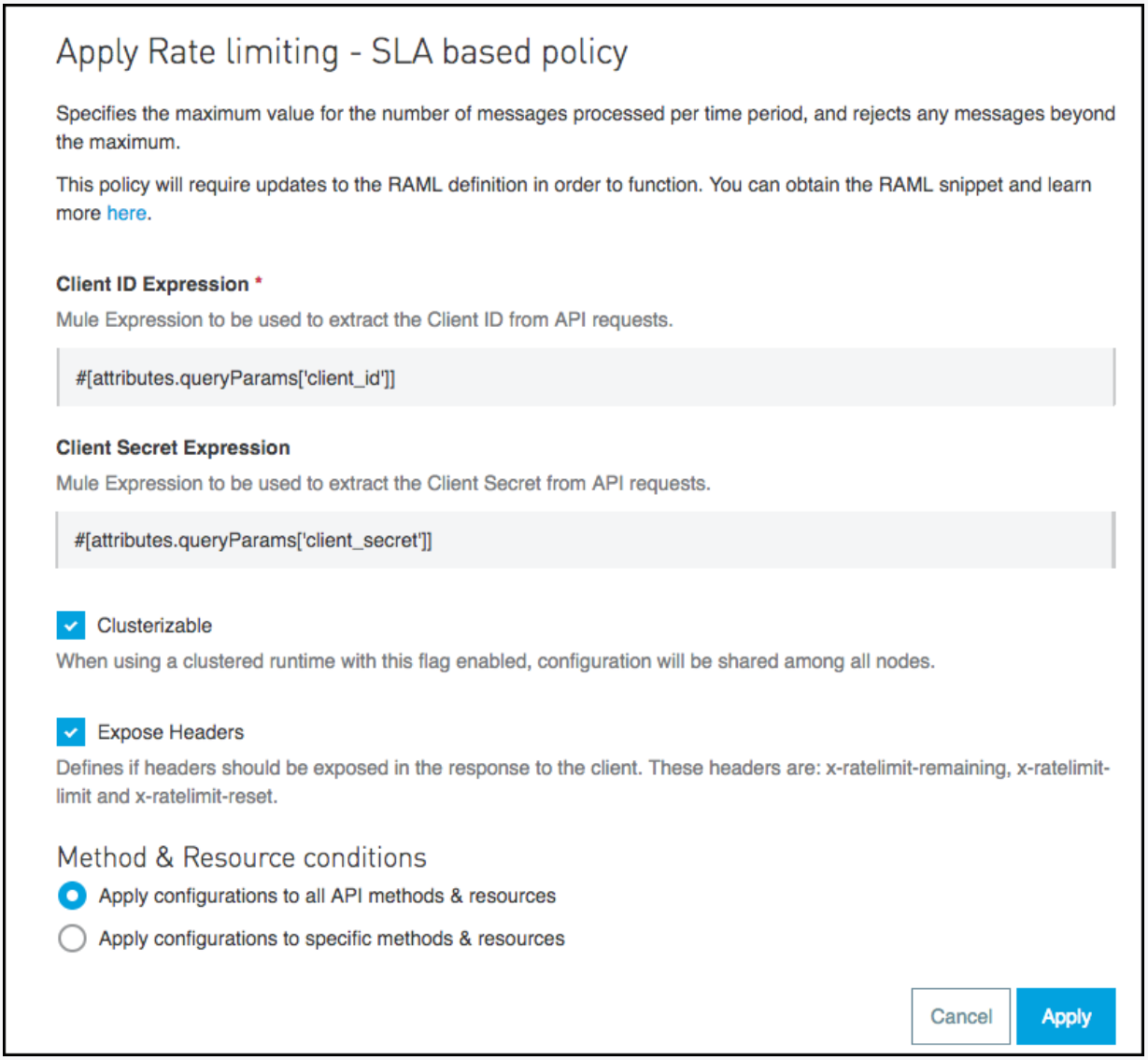


4. Change **Client ID Expression** value with the following

```
#[attributes.queryParams['client_id']]
```

5. Change **Client Secret Expression** value with the following

```
#[attributes.queryParams['client_secret']]
```

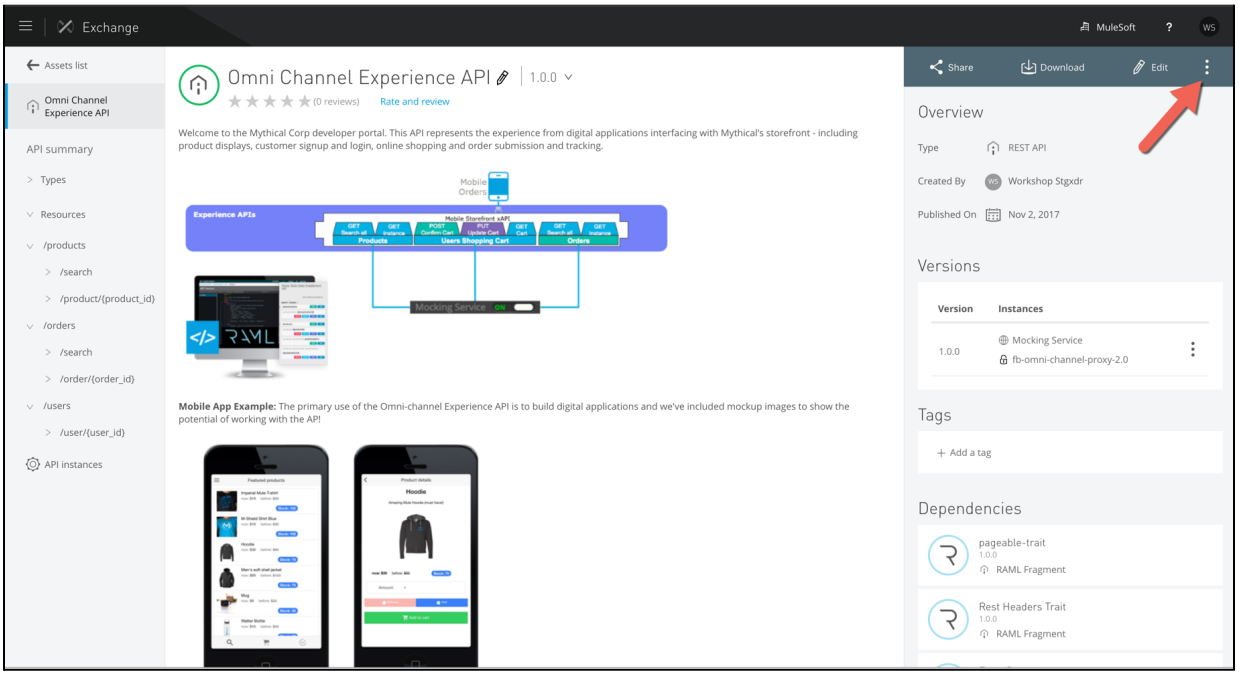


6. Click **Apply** to save configuration for the Rate limiting SLA based policy.
7.

Step 4: Request Access for the API

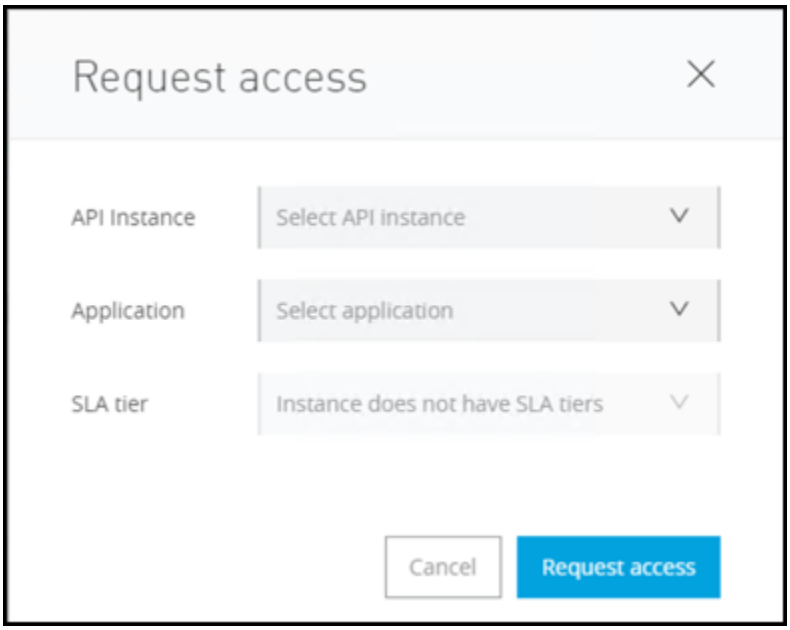
Remember that the APIs are designed to be discoverable and self served. That's why we need to go to the Exchange Portal to request access.

1. Go to the Exchange and select the **Omni Channel Experience API**
2. Press the ellipsis button that is on the right top.

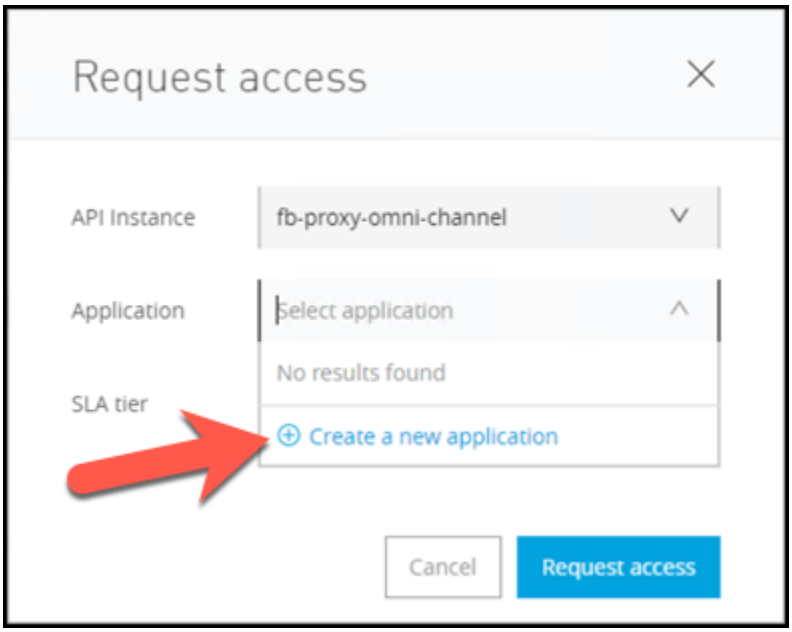


3. Press **Request Access**.

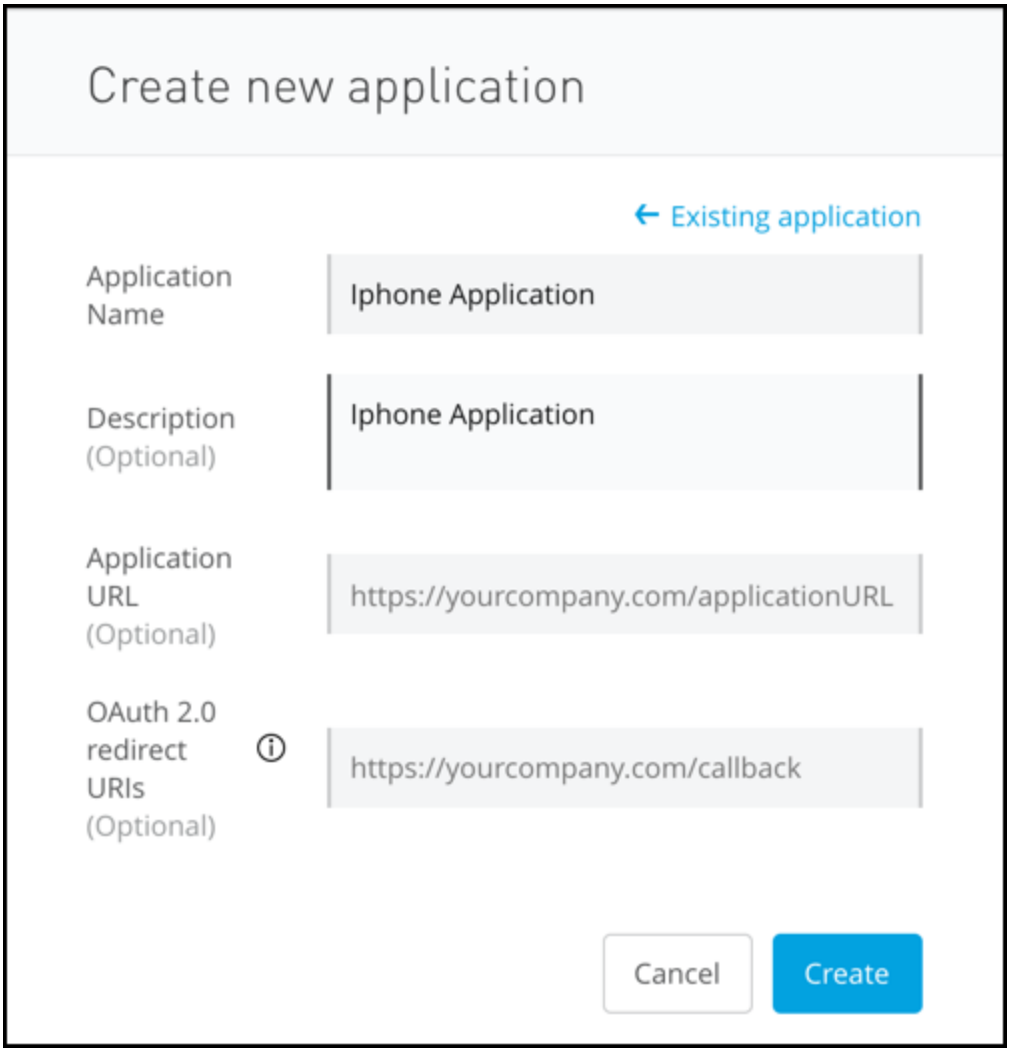
4. A popup window will appear.



5. In the API Instances, select your API.
6. In Application, choose **Create a new Application**.



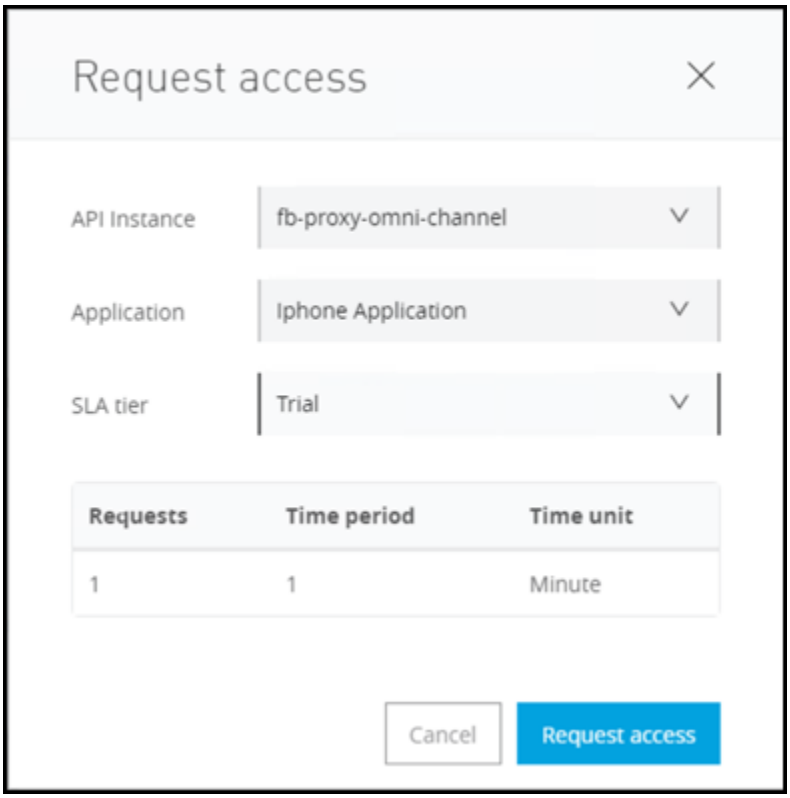
7. Complete the new application dialog as shown below (you must create a unique application name such as **mtm iPhone Application**). When finished, click **Create**.



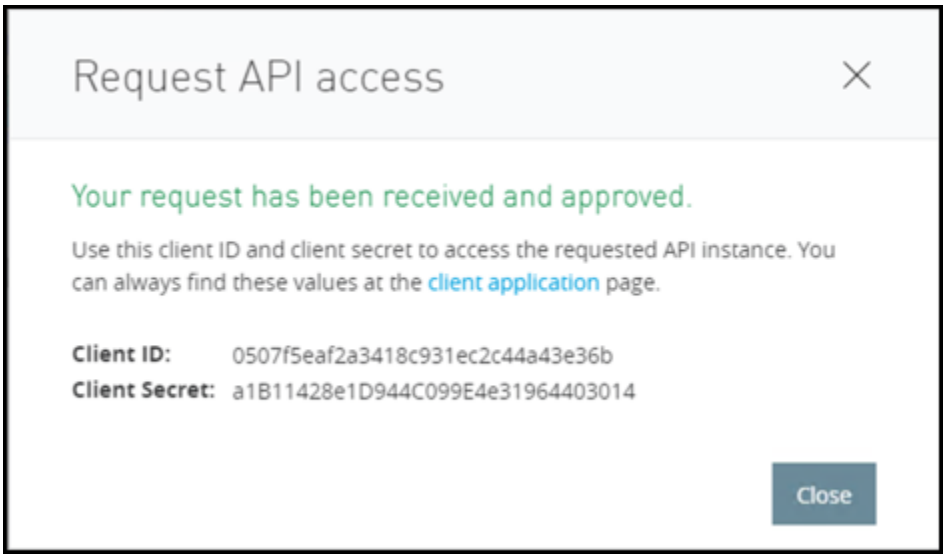
Since we have tiers associated with our API we also need to select a tier.

8. Choose the API Instance you deployed
9. Choose the **Trial** tier.

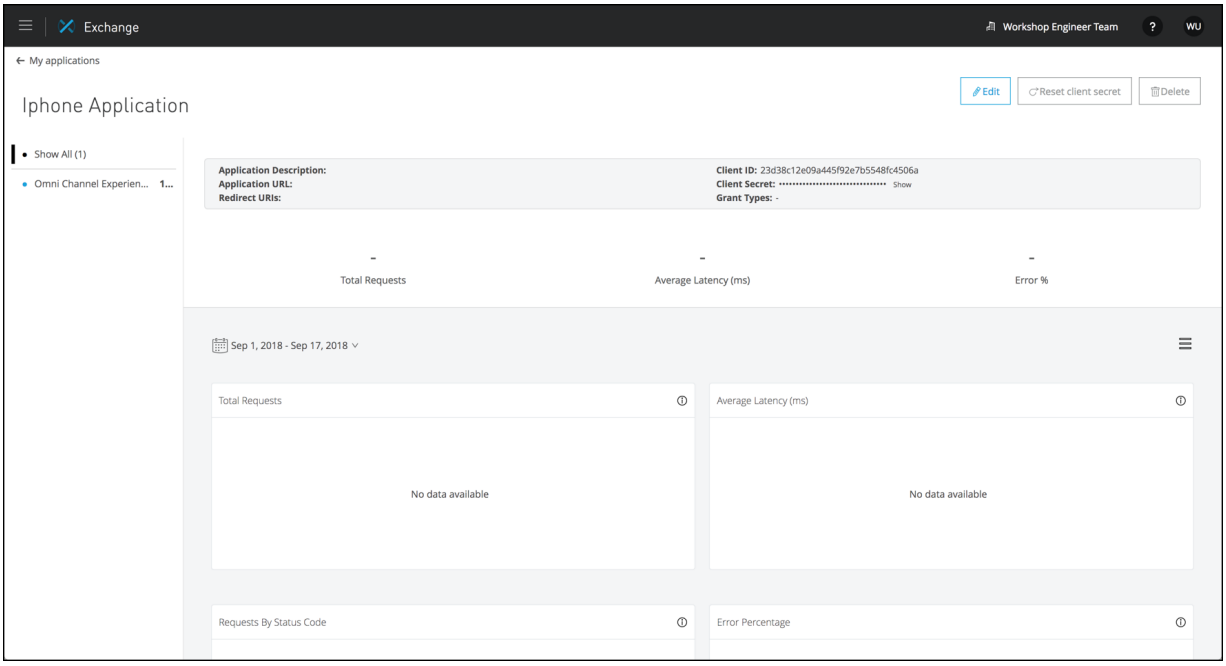
10. Click on the **Request API Access** button.



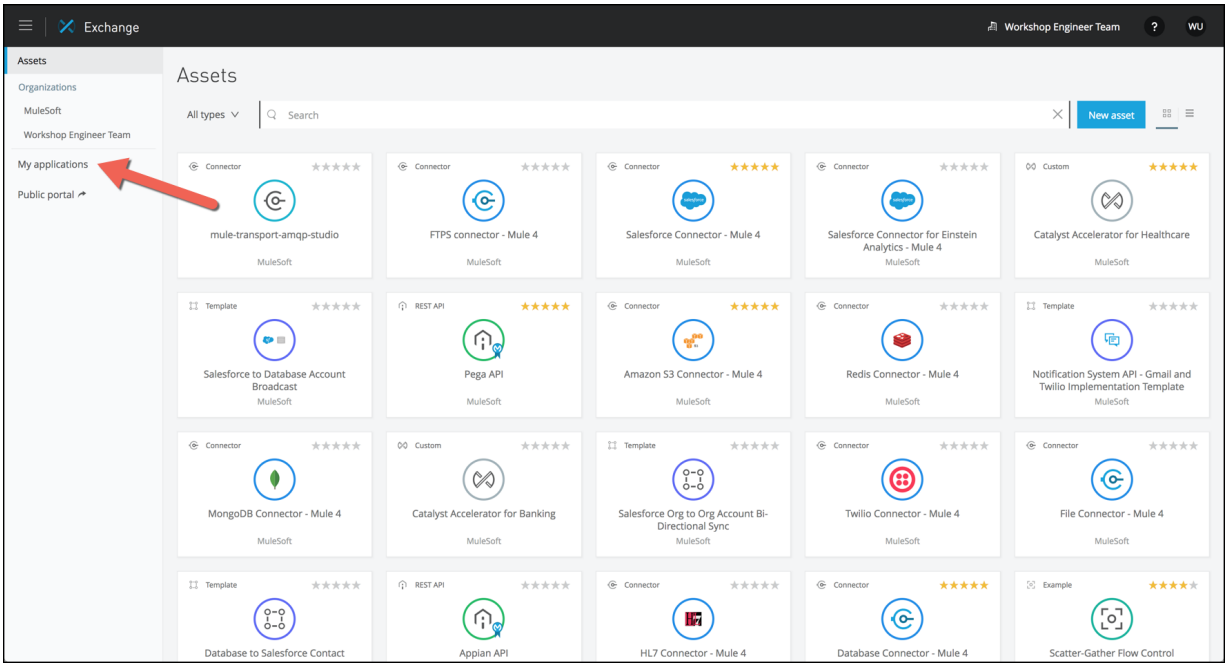
11. By default, all API requests will be approved for the *Trial* SLA tier. You'll see your Client ID and Client secret.



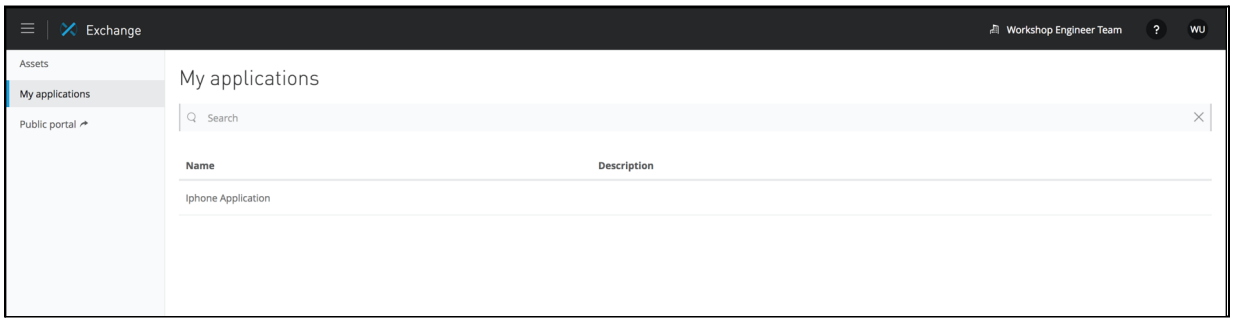
12. After you request access, a new page will open with your application and the APIs that is registered.



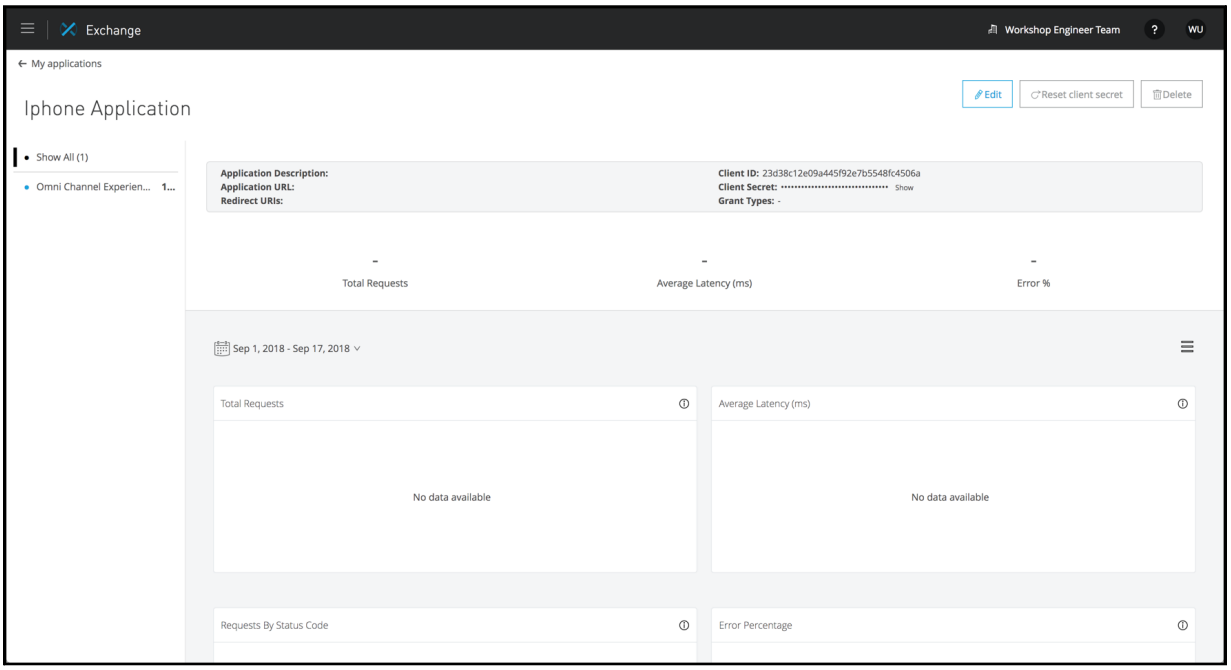
13. Close the window.
14. In Exchange you can click on **My Applications** to access all your registered applications.



15. Click on **My Applications**. You will find the API you've just created

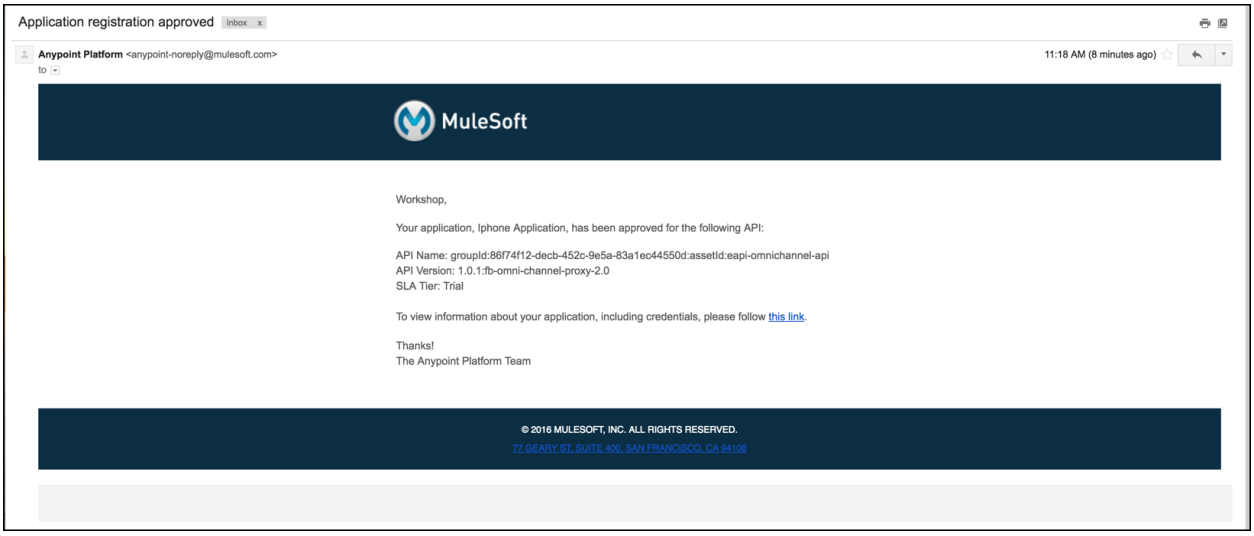


16. Click on the API you've just created and you will get the same page you saw when you created.



17. Go to your email

18. You should see the below registration email indicating you have been auto-approved.



Application registration approved Inbox x

Anypoint Platform <anypoint-noreply@mulesoft.com> 11:18 AM (8 minutes ago)
to

MuleSoft

Workshop,

Your application, Iphone Application, has been approved for the following API:

API Name: groupId:86f74f12-decb-452c-9e5a-83a1ec44550d:assetId:eapi-omnichannel-api
API Version: 1.0.1:fb-omni-channel-proxy-2.0
SLA Tier: Trial

To view information about your application, including credentials, please follow this link.

Thanks!
The Anypoint Platform Team

© 2016 MULESOFT, INC. ALL RIGHTS RESERVED.
77 GEARY ST, SUITE 400, SAN FRANCISCO, CA 94108

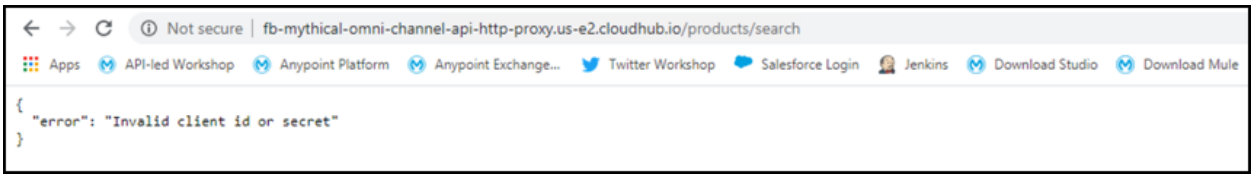
Step 5: Test the API

You will now test the Omni Channel Experience API.

1. Test the API again using your browser or using Postman and access your CloudHub URL with /products/search. For example: http://fb-mythical-omni-channel-api.us-e2.cloudhub.io/api/products/search.
2. You should see the response:

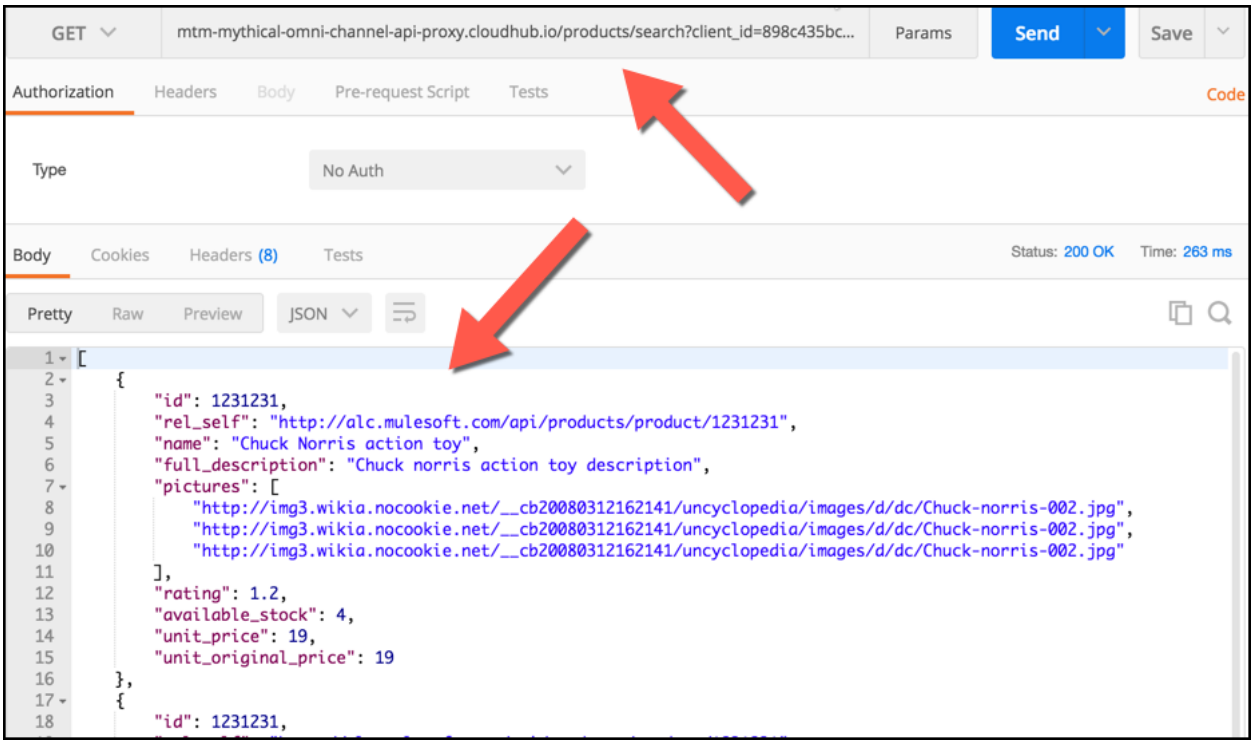
   `Invalid client id or secret`

   This is because the **Rate Limiting - SLA** policy id applied

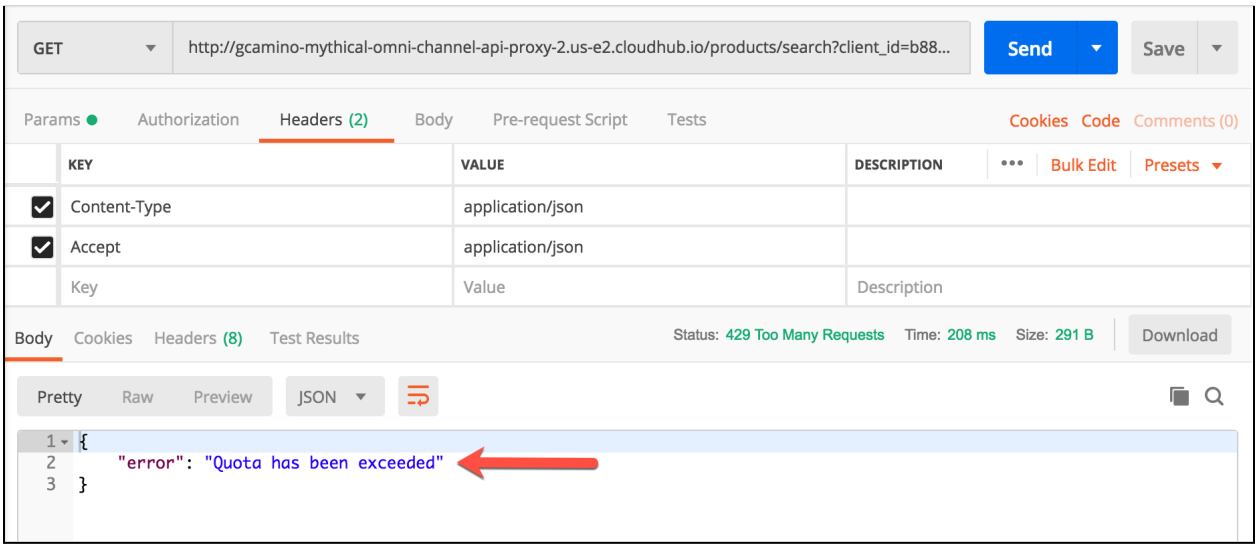


Step 6: Test the API with Credentials

1. Add **?client_id=** *<yourId>* **&client_secret=** *<yourSecret>* to the request URL. For example:
   http://<username>-mythical-omni-channel-api.us-e2.cloudhub.io/api/products/search?client_id=b466e22597b94689952bb77792cf7f8d&client_secret=53EDAFEDeA6f45dF95408596f846417F
   Change the url according to the proxy you deployed
2. You should now be able to access the product information because you entered your application credentials.



3. Execute the test again and you'll see you have exceeded the Rate Limit for the **Trial** tier.



Summary

In this lab, we completed the following steps:

- Overview
- Step 1: Apply Rate Limiting Policy
- Step 2: Create SLA Tiers
- Step 3: Add a Rate Limiting SLA-based Policy
- Step 4: Request Access for the API
- Step 5: Test the API
- Step 6: Test the API with Credentials

We saw the capabilities around managing APIs and applying policies to **enforce security and governance** around your API giving you better **control**. We easily applied rate limiting policies and added SLA tiers giving you the ability to **scale** with easier management and operations. We created a basic API portal for providing **easy access** to your APIs and we tested how to use consumer credentials to access your APIs that were provisioned based on SLAs.

Go Further:

- Learn more about Applying Runtime Policies
- Learn more about security policy configurations:
  - LDAP policy
  - OAuth 2.0 Access Token Enforcement Using External Provider Policy
  - OAuth Token Enforcement Policies
- In addition to the OOTB policies you saw in this lab, you can also Create Custom Policies
- Learn more about Managing API Versions

Congratulations! You have completed Lab 3.

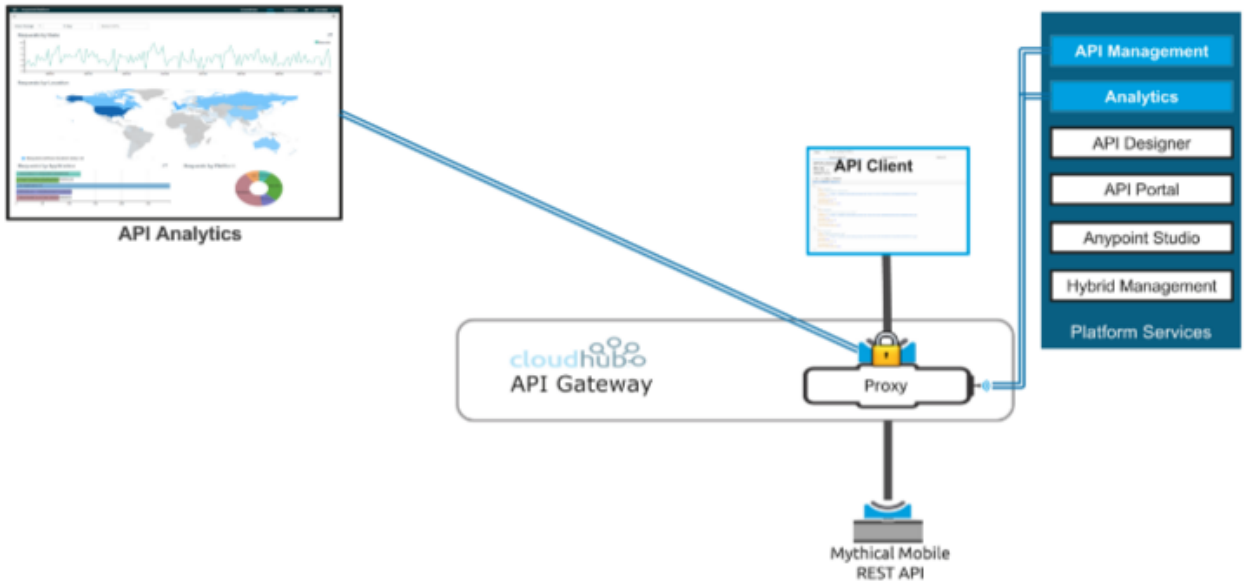
Please proceed to Lab 3

Take me to the TOP

Lab 3: Monitor your API (Analytics)

Overview

Once we have made an API deployment to the Mule Runtime it will start to push analytics data up to Anypoint Platform API analytics. This provides the ability to keep track of key metrics like top users, transactions by region, usage by tier and performance against SLAs in a set of interactive, graphical dashboards and charts. You can create Custom Dashboards in a few clicks and we provide the ability to extract metrics via an API for the API platform.



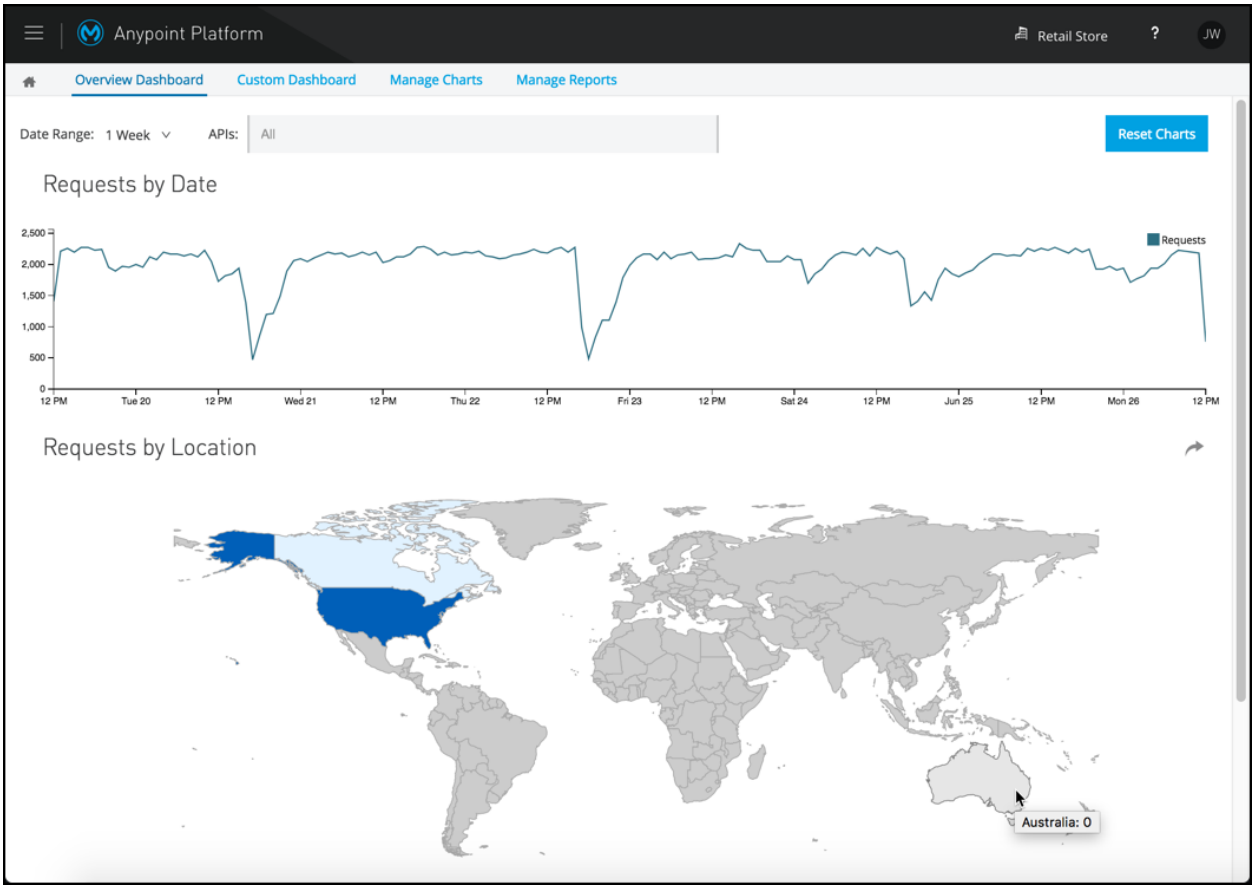
Step 1: Viewing API Analytics

API analytics allow users to keep track of key metrics like top users, transactions by region, usage by tier and performance against SLAs in a set of interactive, graphical dashboards and charts.

You can see the **API Analytics** in **API Manager**

The API Settings page of each API provides some summaries of the analytics data such as number of requests, top apps and latency.

1. As an **Organization Administrator**, you have access to the **Analytics Dashboards** for your organization. Go to **API Manager** and Click on **Analytics** to navigate to the overview dashboard.



When you access the Analytics for your organization, by default you land on your Overview Dashboard. This dashboard displays four standard charts:

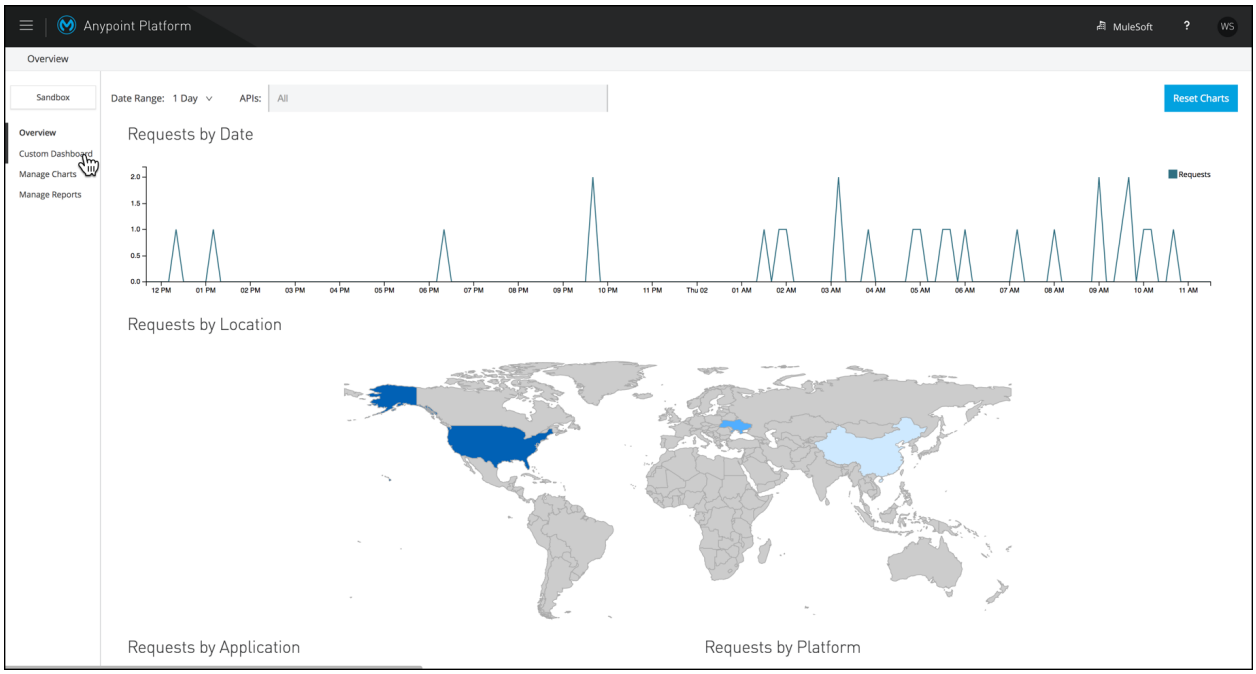
- **Requests by Date:** Line chart that shows the number of requests for all APIs in your organization.
- **Requests by Location:** Map chart that shows the number of requests for each country of origin.
- **Requests by Application:** Bar chart that shows the number of requests from each of the top five registered applications.
- **Requests by Platform:** Ring chart that shows the number of requests broken down by platform.

2. **Interactive Dashboard:** You can interact with the dashboards by clicking on the time range and dragging your mouse to create a custom time range. The overview dashboard graphs are linked together, so your custom time range could change the numbers and types of requests shown in

the graphs at the bottom. For example, you can select a specific country, like Canada, to see only the requests from that specific location.

Step 2: Create a Custom Dashboard

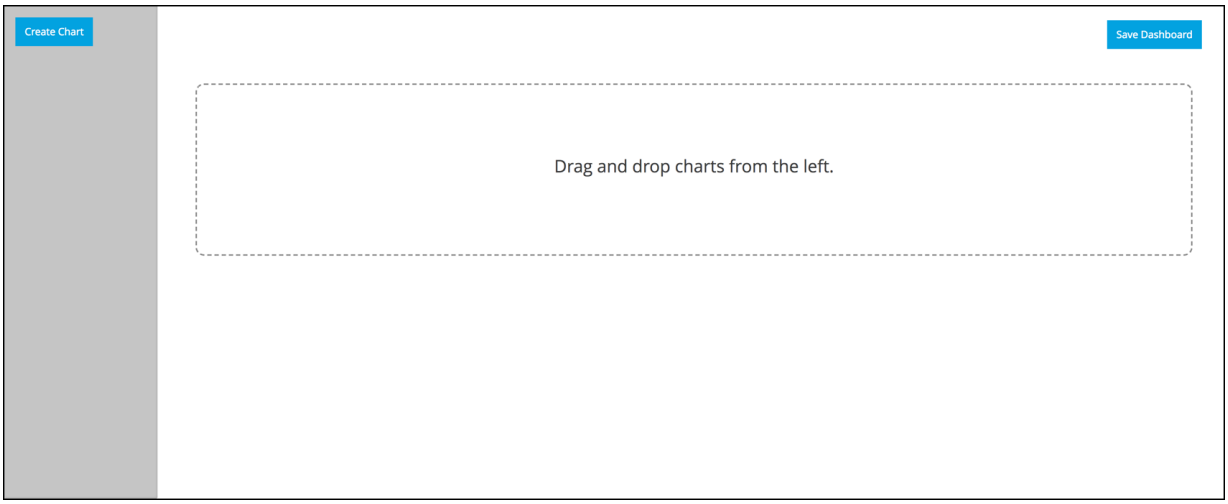
The overview dashboard provides an aggregate view of the API usage, but there is typically a need to see more detailed API utilization. The Anypoint Analytics platform allows user to create charts and build custom dashboards for more detailed information. In this step we'll create a custom dashboard by building 4 custom charts, including viewing response time, response codes, requests by city and policy violations.



To create a custom dashboard:

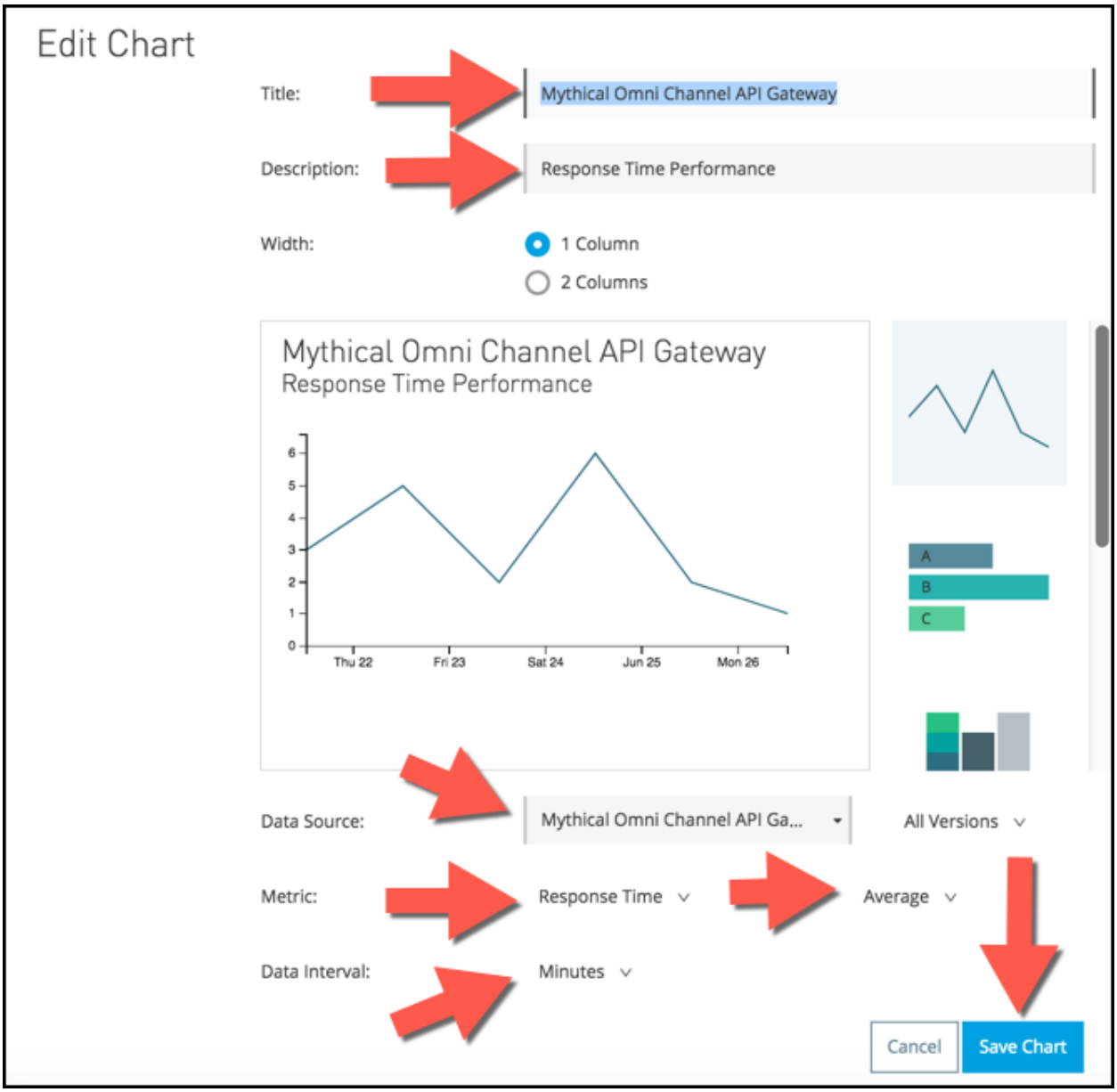
1. Click on the **Custom Dashboard** option in the menu on top

2. Select **Create Chart** (note - click **Edit Dashboard** if you don't see **Create Chart** on the left)



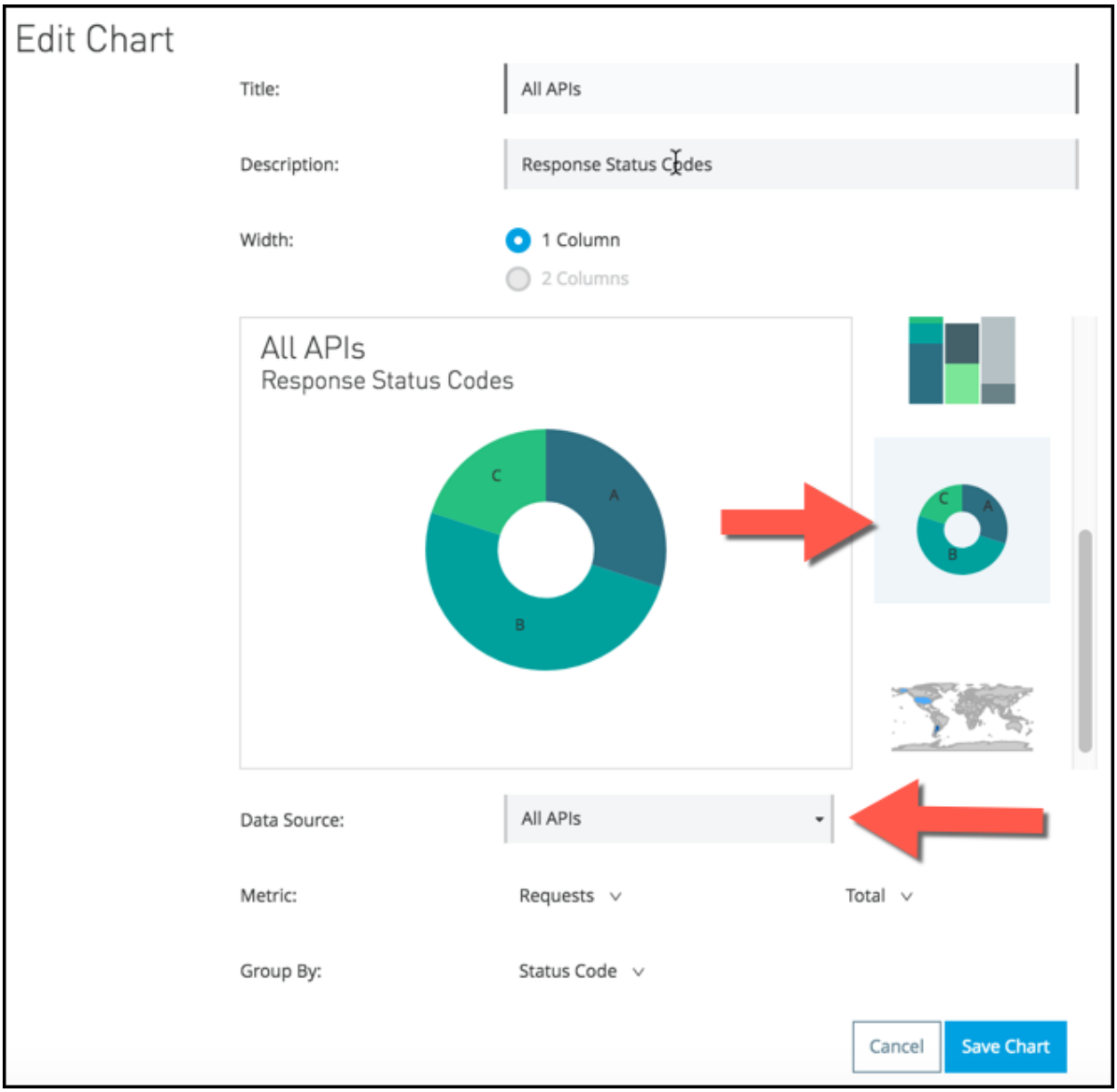
3. Add a chart **Title**.
4. Specify the metrics to chart the **Response Time** for your **Mythical Omni Channel API Gateway** and how to aggregate the values as shown below. Make sure to change the Data Interval to

Minutes.



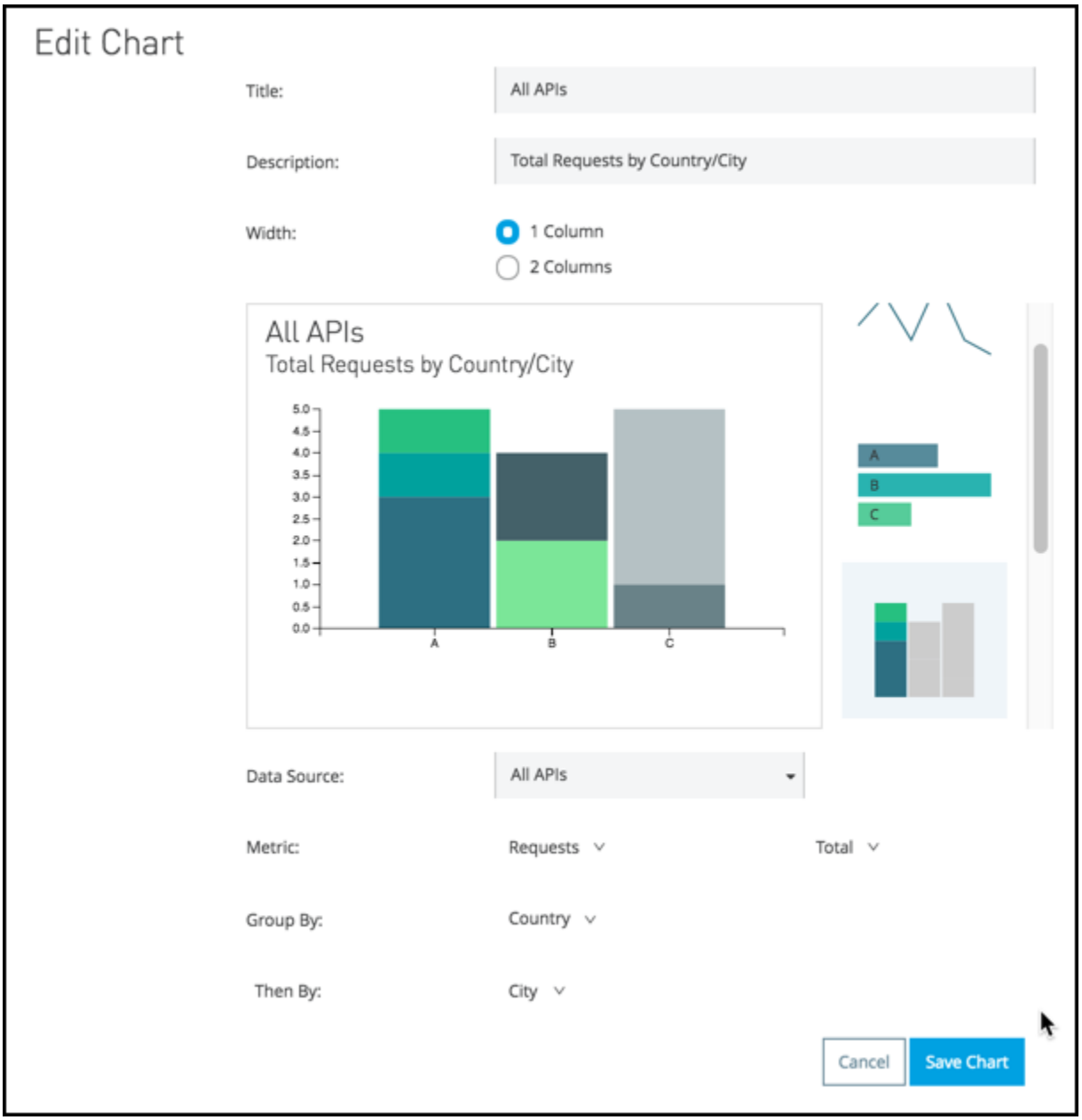
5. When you're complete select **Save Chart**.

6. Create a chart to track Response Status Codes **for all APIs** as shown below.



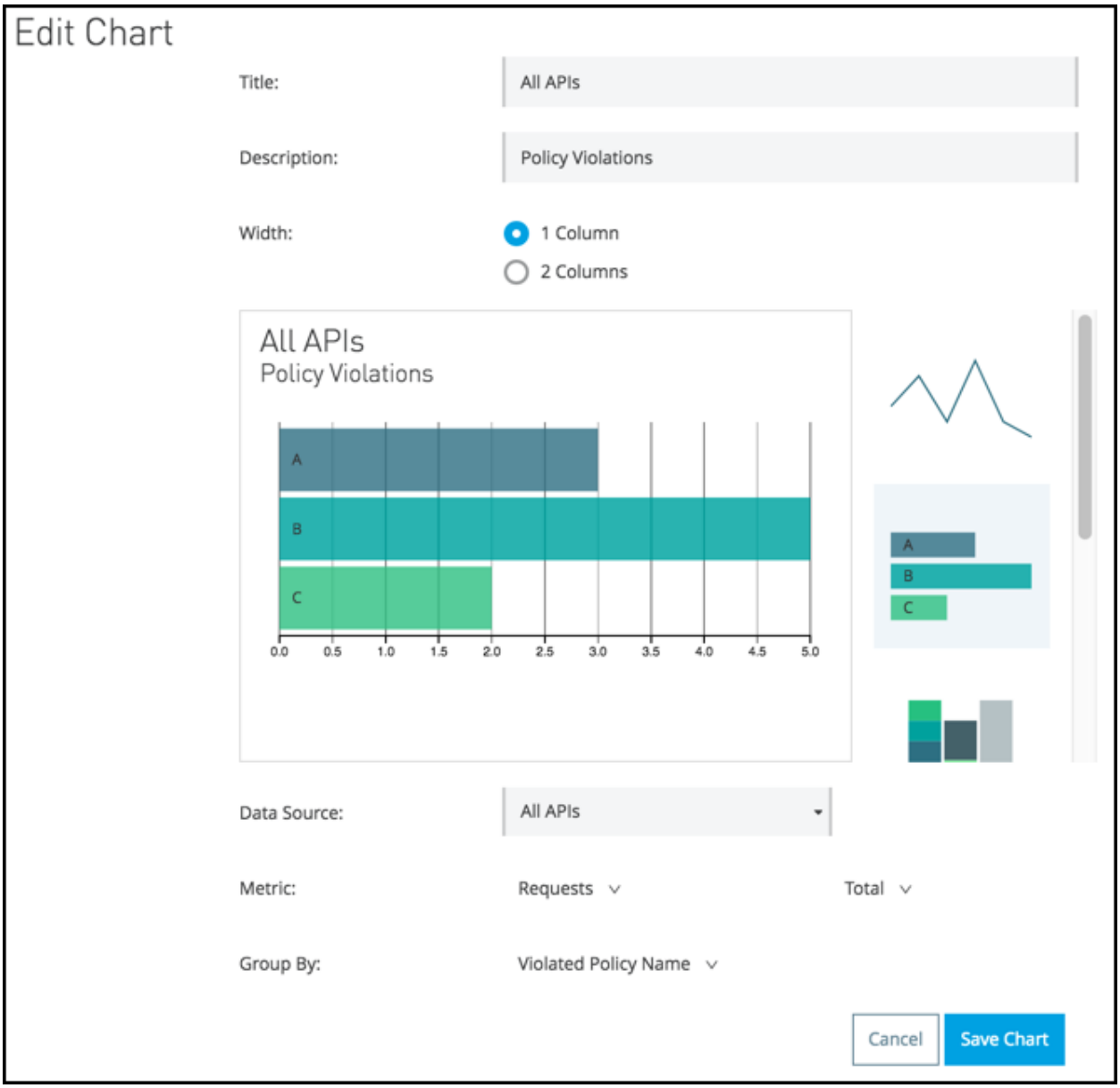
7. Click **Save Chart**

8. Create a chart with the grouping of requests by country and city.



9. Click **Save Chart**

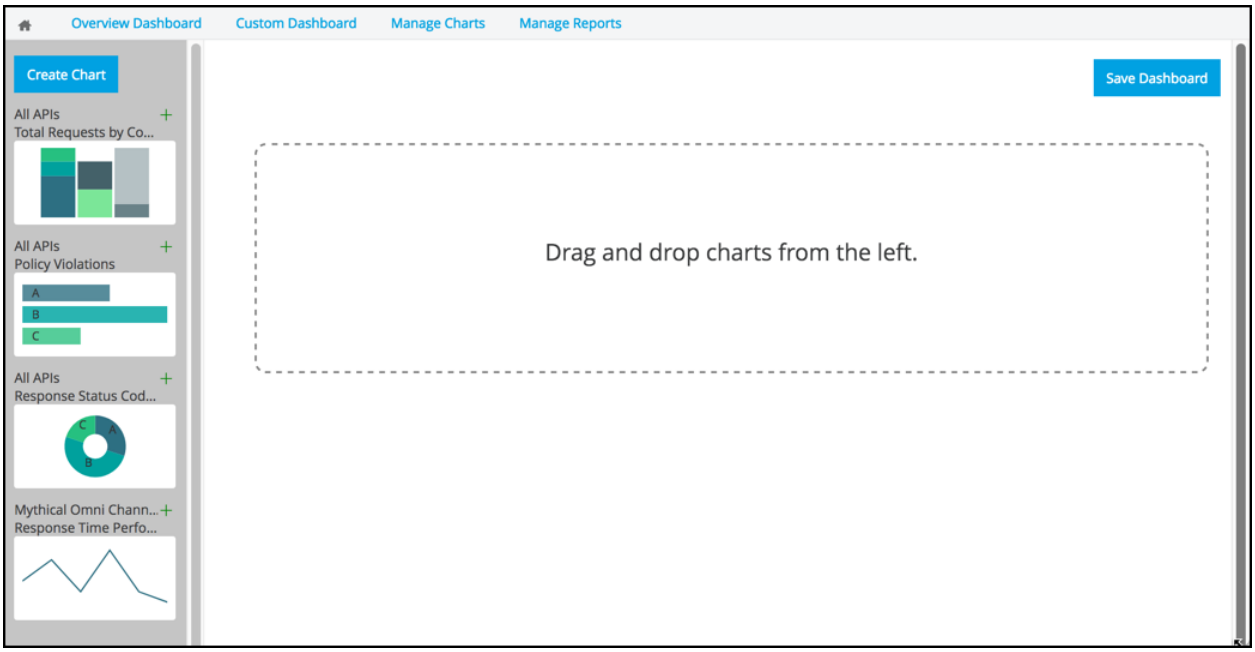
10. The last chart we will create is for measuring any requests that have policy violations.

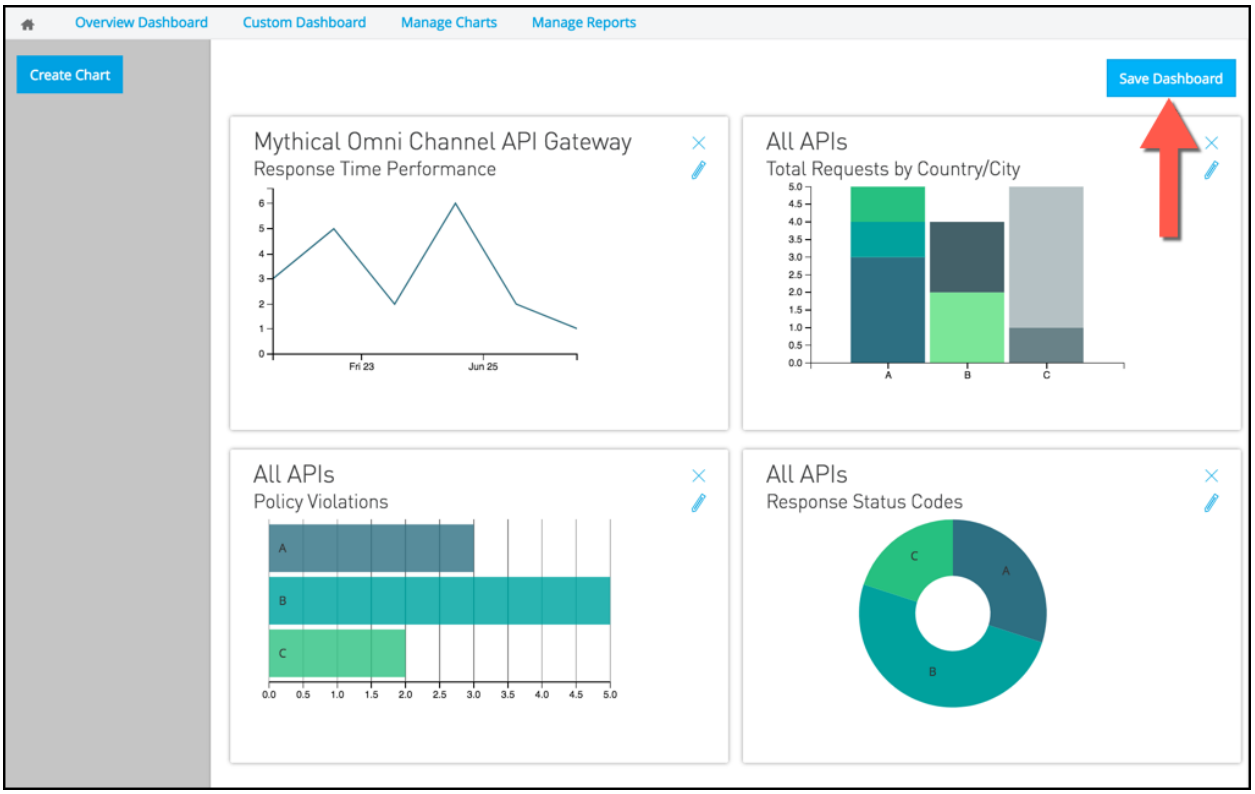


11. Click **Save Chart**

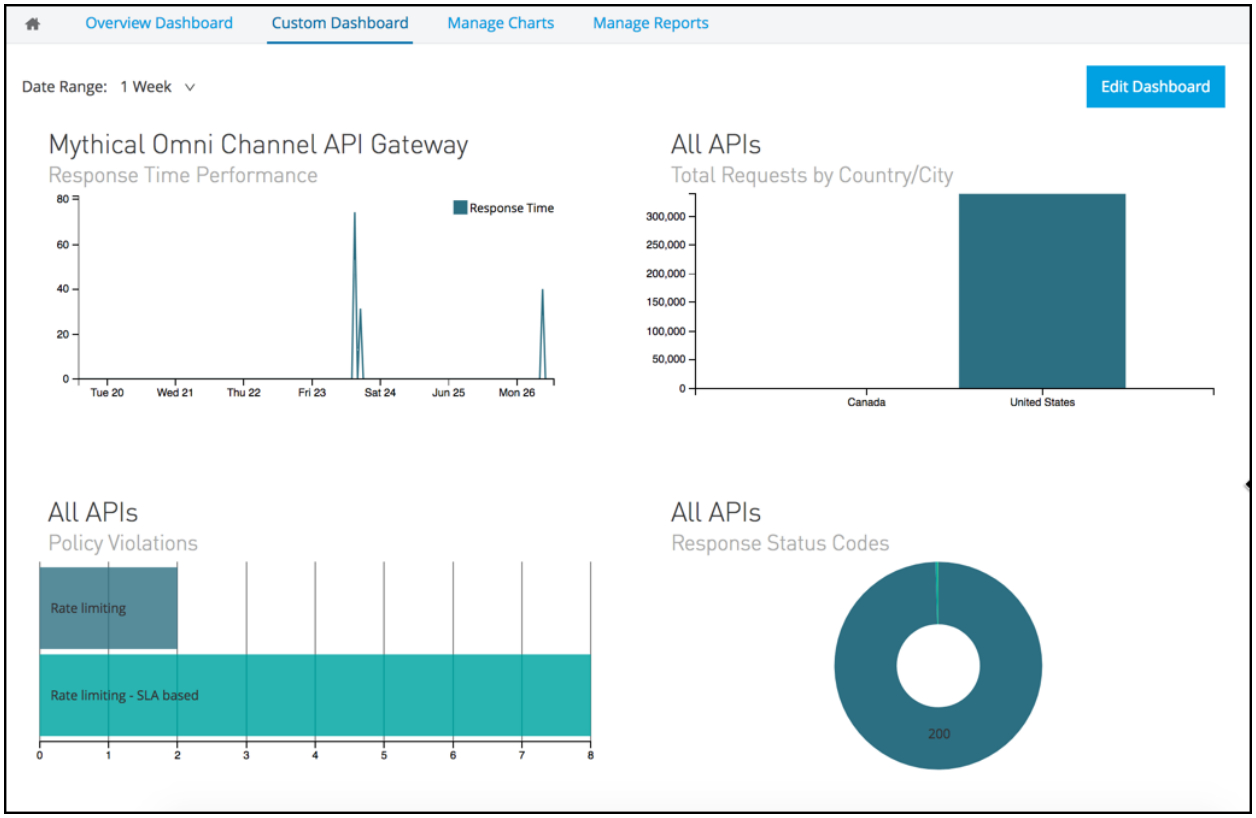
    When your chart has been saved, it is available to be added to the canvas of your custom dashboard.

12. Drag and drop your newly created charts to the dashboard canvas.





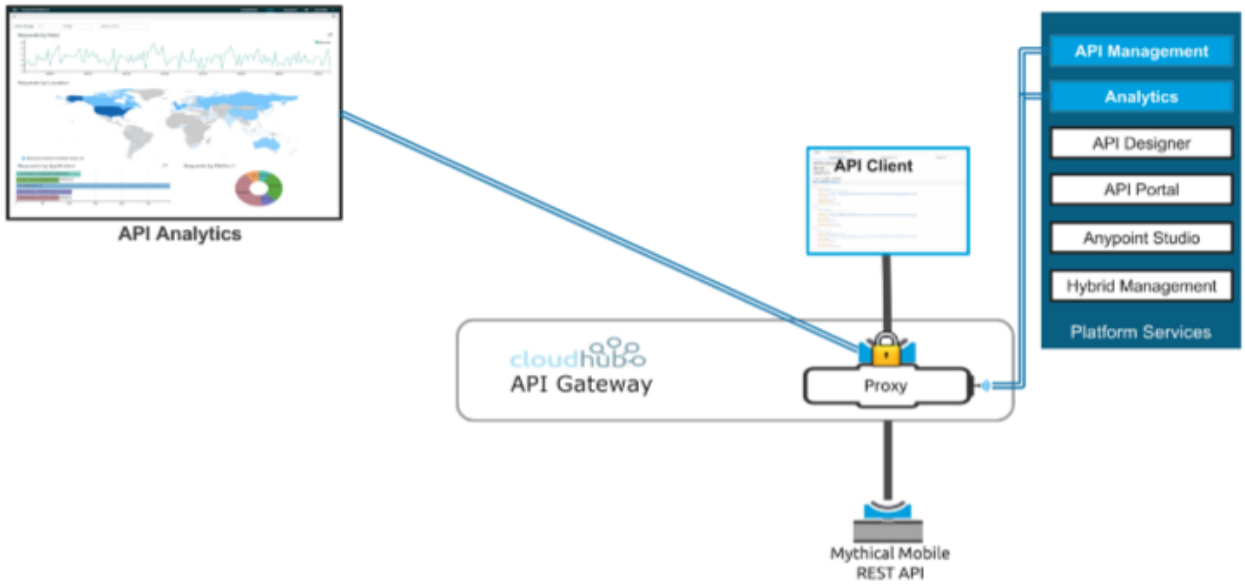
13. Click **Save Dashboard** in the top right of the Dashboard editor. You will now have a dashboard that will display metrics that are custom to your API.


Overview Dashboard
Custom Dashboard
Manage Charts
Manage Reports
Date Range: 1 Week
Edit Dashboard
Mythical Omni Channel API Gateway
Response Time Performance
Response Time
80
60
40
20
0
Tue 20
Wed 21
Thu 22
Fri 23
Sat 24
Jun 25
Mon 26
All APIs
Total Requests by Country/City
300,000
250,000
200,000
150,000
100,000
50,000
0
Canada
United States
All APIs
Policy Violations
Rate limiting
Rate limiting - SLA based
0
1
2
3
4
5
6
7
8
All APIs
Response Status Codes
200

Summary

- Overview
- Step 1: Viewing API Analytics
- Step 2: Create a Custom Dashboard

In this lab, you were able to monitor your APIs using the built in API Analytics and Custom Dashboards. This provides **visibility** into your APIs so you can understand how your APIs are being consumed and utilized.



Additional details on creating custom dashboards and custom chart aggregations can be found in the Mule doc at *Viewing API Analytics*

Congratulations! You have completed Module 3. Please proceed to Module 4

Take me to the TOP