# Lab 2: Manage and Consume the API

[Lab 2: Manage and Consume the API](#_unhd2nhxtatb)

[Overview](#_fs8583o0cdf6)

[Step 1: Apply Rate Limiting Policy](#_edetyfd26erb)

[Step 2: Create SLA Tiers](#_izq5ps7y6crt)

[Step 3: Add a Rate Limiting SLA-based Policy](#_kkj894bxexeo)

[Step 4: Create an API Portal](#_aec2o4k4jfsy)

[Step 5: Request Access for the API](#_8iei5doca5lj)

[Step 6: Get Your App Credentials](#_f4pou9yf65r2)

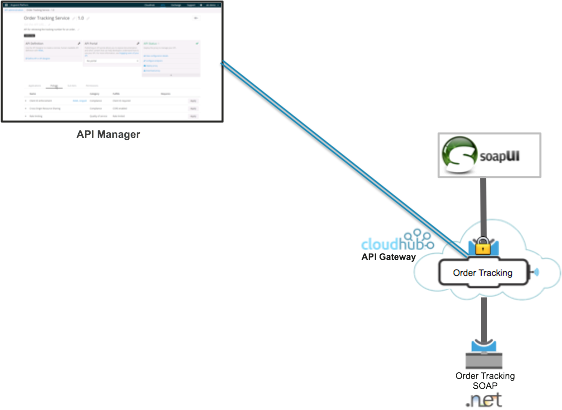
[Step 7: Test with SoapUI](#_8uybtme35shu)

[Step 8: Test the API with Credentials](#_haiur5lcz8k9)

[Summary](#_lqog6mfxdrs5)

## Overview

Now that we’ve deployed the API Gateway, the next step is to enforce governance around the API. The *Anypoint API Manager* is an API policy management and governance tool that is integrated with the *API Gateway*. In this lab, you will utilize the secure connection between the *API Gateway* and *API Manager* to configure policies that will allow us to manage security, quality of service, and compliance policies for your APIs.



The [*Proxy an Existing API lab*](https://docs.google.com/a/mulesoft.com/document/d/1lHlLBeonjGI8vAvuyQ3mP_maZ_n_2601XLUULSX8rP0/edit) added an existing SOAP API used for order tracking to the platform by generating a proxy for it. This proxy API was auto-deployed to the API Gateway and the service is now exposed as a *System Level API*. A CloudHub gateway is hosting your *API proxy application* and will route the requests to the existing .NET SOAP **Order Tracking Service**.

The API defined in the [*Proxy an Existing API lab*](https://docs.google.com/a/mulesoft.com/document/d/1lHlLBeonjGI8vAvuyQ3mP_maZ_n_2601XLUULSX8rP0/edit) was deployed to a CloudHub API Gateway worker using 0.1 vCores. Now that it has been deployed you can:

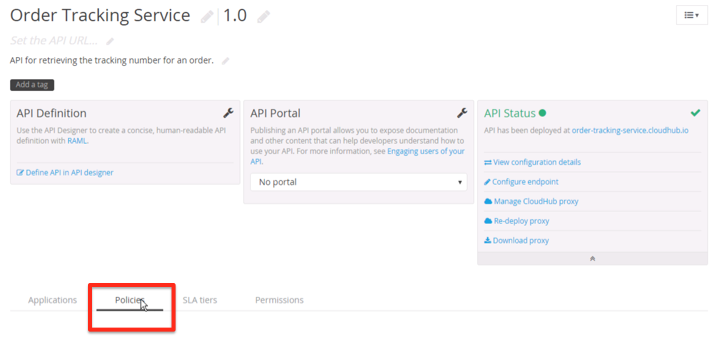
* [Create a portal](http://www.mulesoft.org/documentation/display/current/Engaging+Users+of+Your+API) for the API to engage developers
* [Manage security](http://www.mulesoft.org/documentation/display/current/Managing+Your+API) and enforce [policies](http://www.mulesoft.org/documentation/display/current/Applying+Runtime+Policies)
* Manage [contracts](http://www.mulesoft.org/documentation/display/current/Defining+SLA+Tiers) associated with that API
* Specify [terms & conditions](http://www.mulesoft.org/documentation/display/current/Adding+Terms+and+Conditions) associated with the API
* [Monitor the API](http://www.mulesoft.org/documentation/display/current/Viewing+API+Analytics)  with analytics and KPI metrics in a dashboard view

This makes it easy for the API manager to understand how the API is performing, how it’s being used and by whom, and for the API Administrator to identify potential issues before they arise.

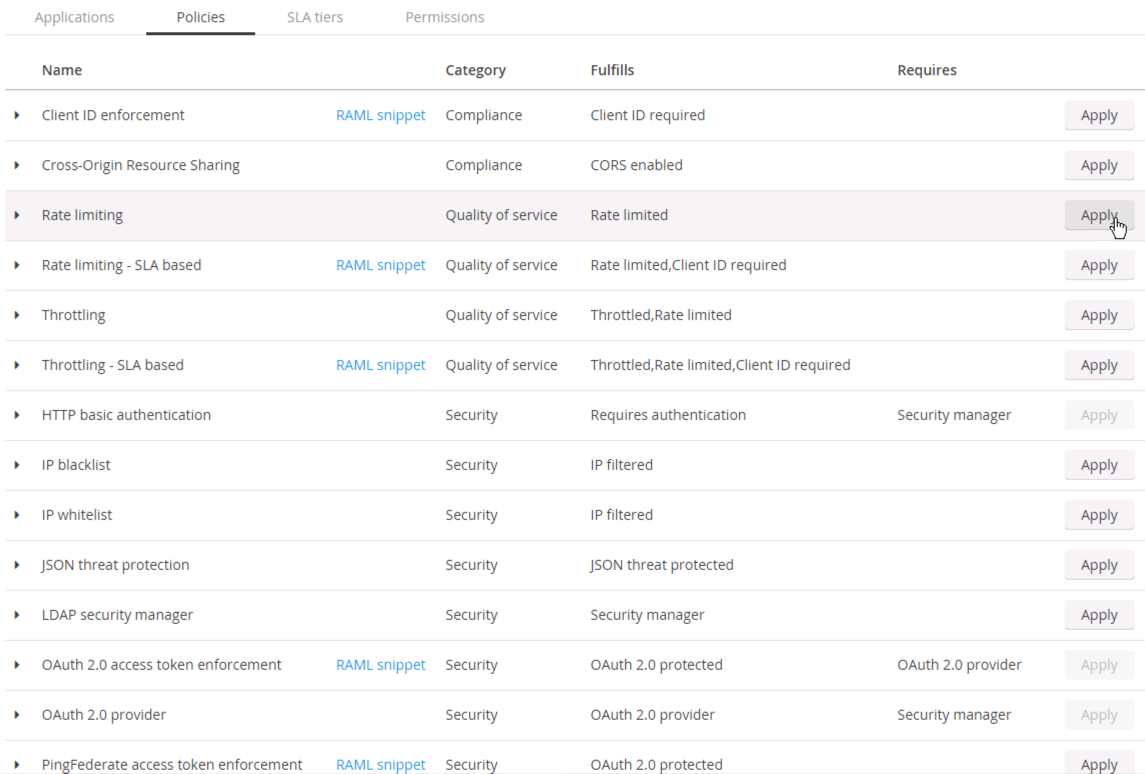
In this lab you will apply policies to your Order Tracking API. The *Anypoint API Manager*  provides many out-of-the-box (OOTB) policies related to compliance, quality of service and security. In addition, you can add your own policies.

## Step 1: Apply Rate Limiting Policy

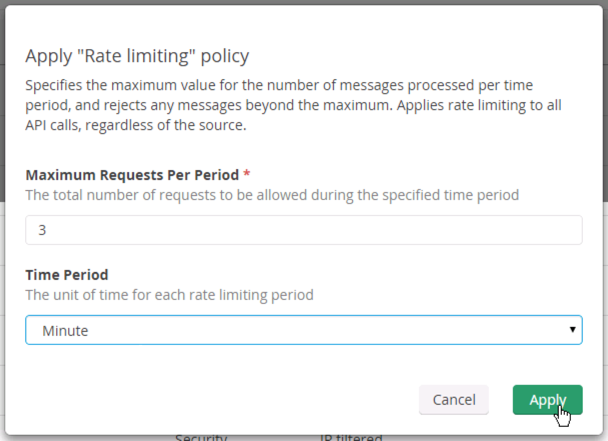
To test policy management, you will add a **Rate Limiting** Policy to the API.



1. Go to the Order Tracking Service **API Administration** page.
2. Click the **Policies** tab under the API definitions.

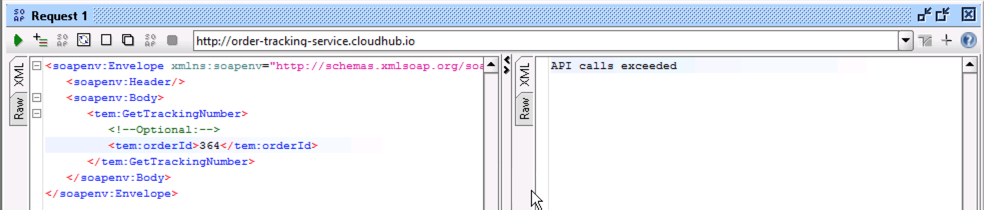


1. Click on **Apply** for the **Rate Limiting** Policy.



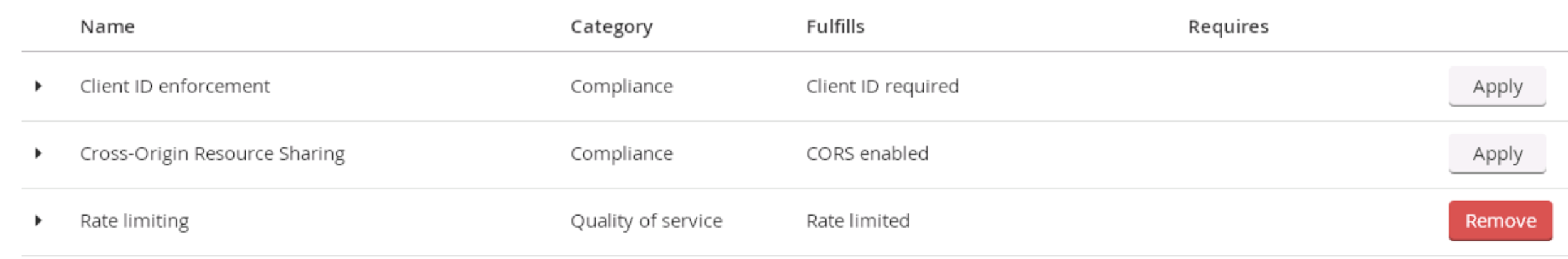
1. Enter a maximum requests of **3** per **Minute** as shown.

|  |
| --- |
| **DOUBLE CHECK**: Make sure you set the **Time Period** to **Minutes** to see the rate limit take effect. |



1. Wait for a few seconds (the API Gateway receive policy updates every 5 seconds by default).
2. **Invoke** (play) the Proxy **4 times**.

On the 4th invocation you will get a message indicating **API Calls Exceeded**. This demonstrates your *Rate Limiting* policy has been applied.



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

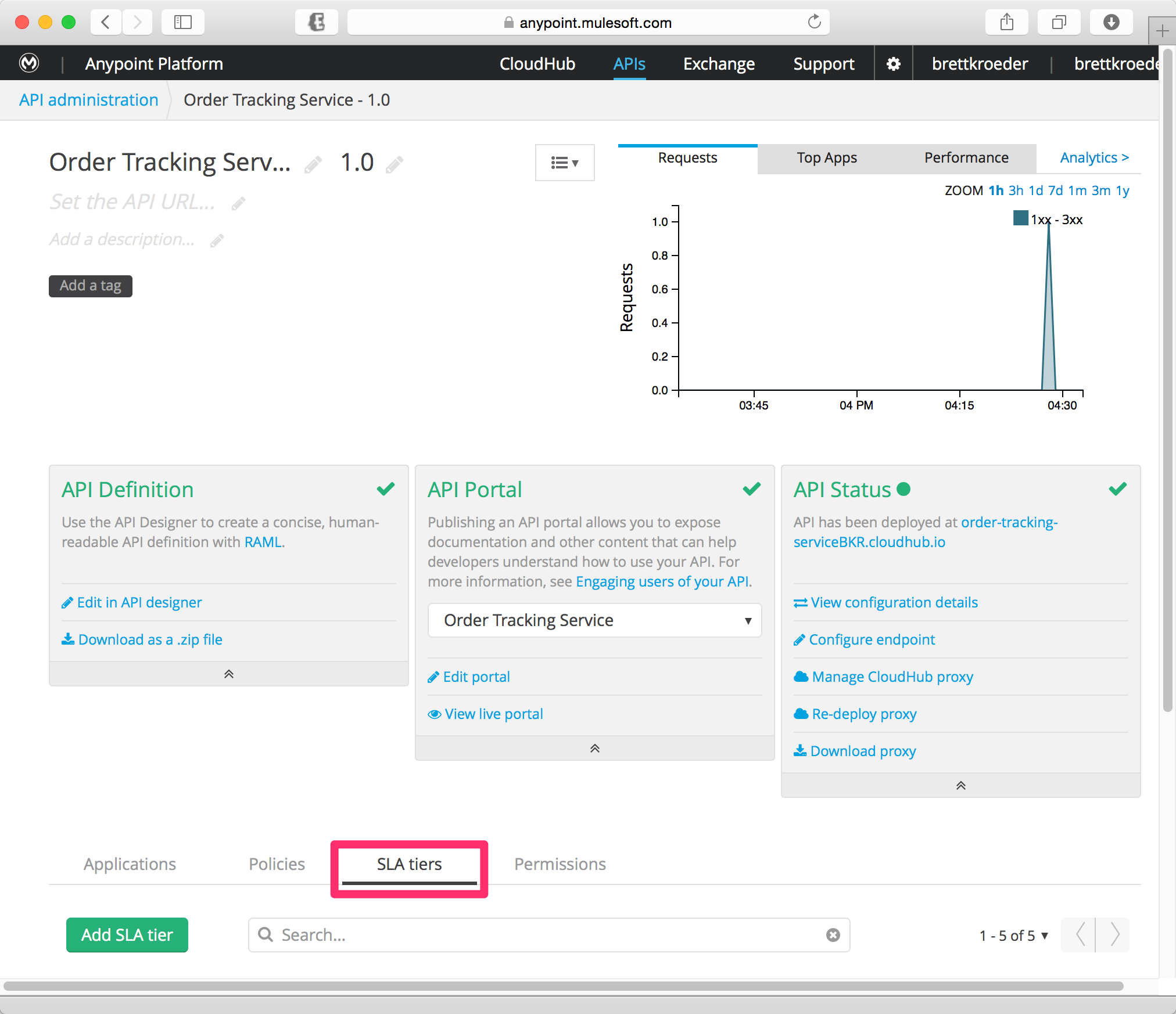
|  |
| --- |
| **DOUBLE CHECK**: Did you **REMOVE** the Rate Limiting policy? |

## Step 2: Create SLA Tiers

In the API Manager access can be based on [**SLA tiers**](http://www.mulesoft.org/documentation/display/current/Defining+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](http://www.mulesoft.org/documentation/display/current/Defining+SLA+Tiers) that consumers can view and choose from when they request access to the API. If no [SLA tiers](http://www.mulesoft.org/documentation/display/current/Defining+SLA+Tiers) are defined for an API, the application owner can request access without an SLA tier.

Let’s define a new [SLA tier](http://www.mulesoft.org/documentation/display/current/Defining+SLA+Tiers) for your API version.



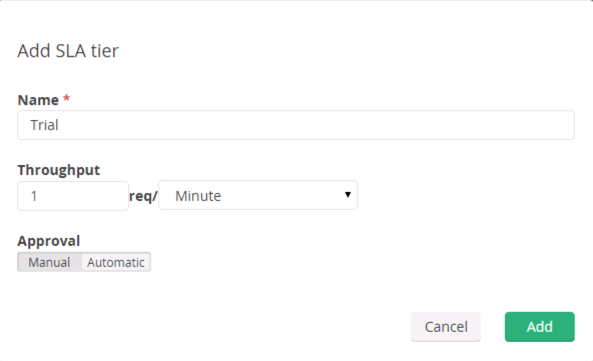
1. Click the [**SLA Tiers**](http://www.mulesoft.org/documentation/display/current/Defining+SLA+Tiers) tab on your API Version Details page

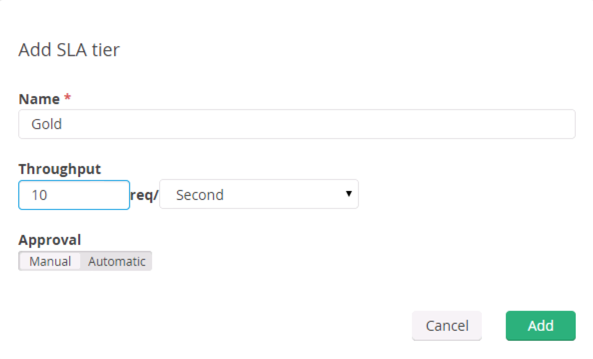
You are going to setup 3 SLA tiers:

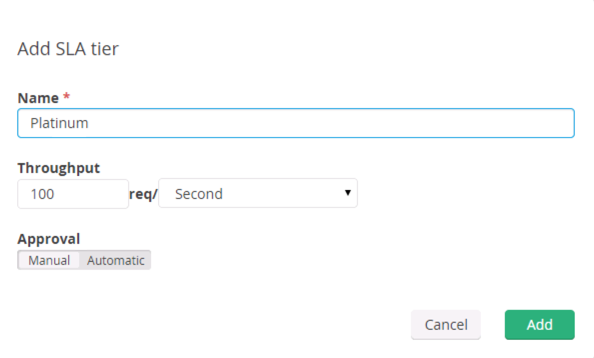
|  |  |  |  |
| --- | --- | --- | --- |
| Tier | Throughput | Period | Approval |
| Trial | 1 | **Minute** | **Automatic** |
| Gold | 10 | Second | Manual |
| Platinum | 100 | Second | Manual |



1. Click **Add SLA tier**.

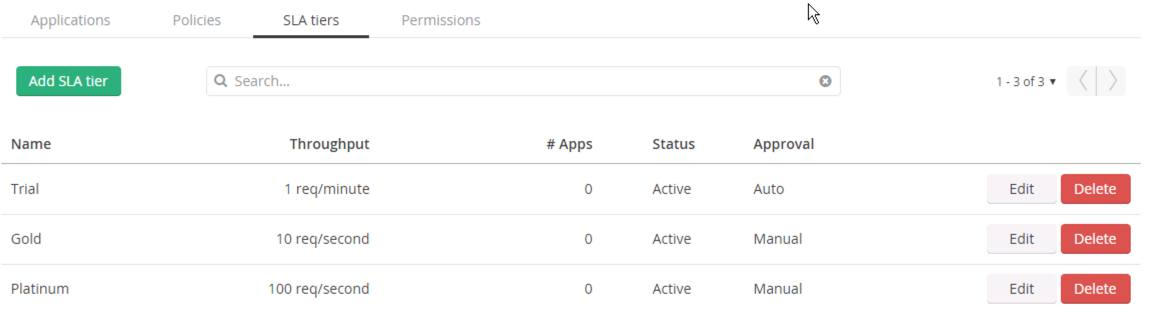






Fill in the fields to configure your tier

1. Give the tier a **Name**
2. Define the **Throughput** by indicating the number of requests per time period that are allowed,
3. Indicate whether application access **Approval** at this tier level should be automatically approved or require manual approval.
4. Click **Add** to save your tier.
5. Repeat for all the tiers shows above.

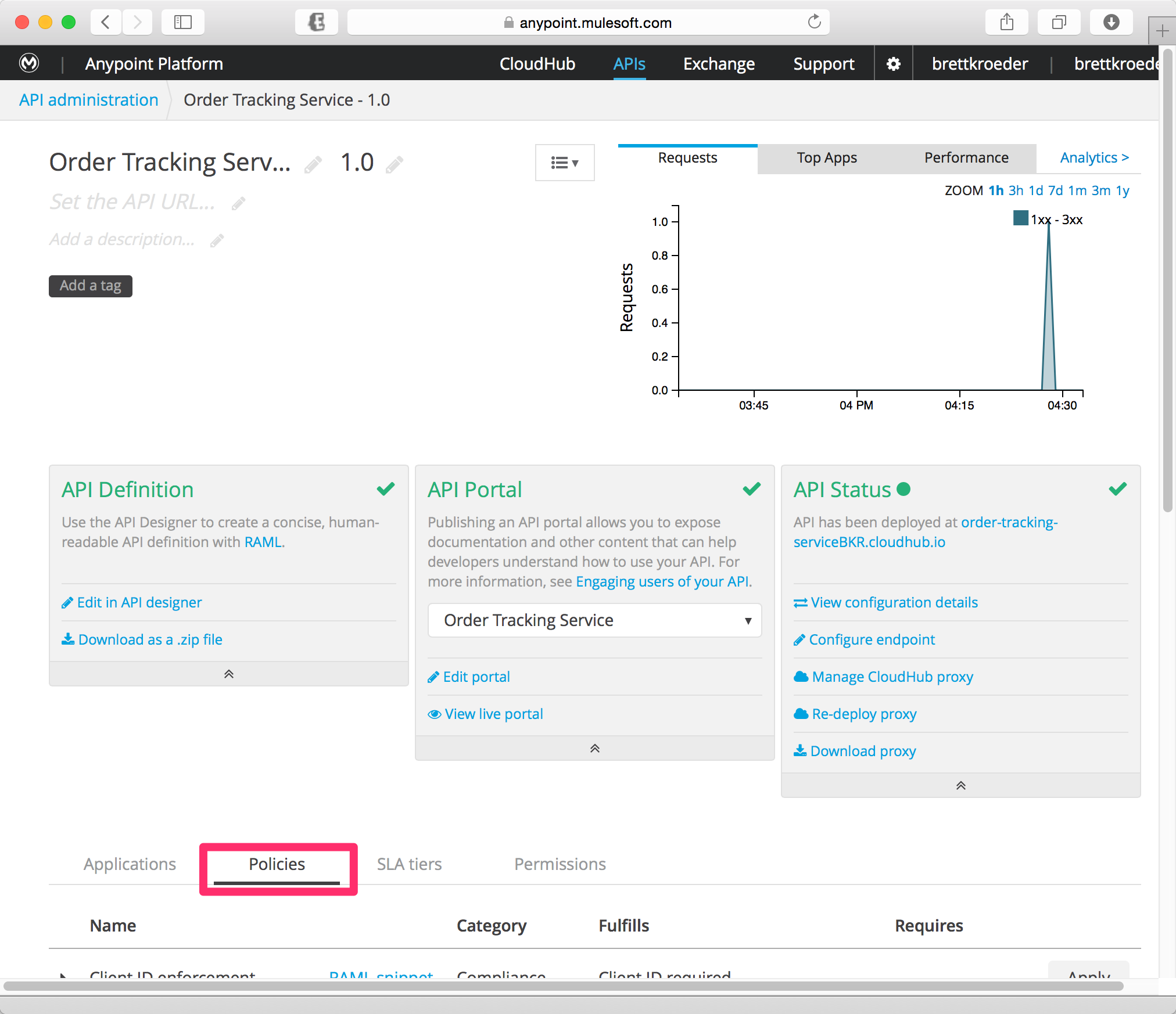


Your SLA tier is displayed with all of the information that you just defined. In addition, you have a column to indicate how many applications are registered on that tier. You can also edit or delete the tier using the links in the row.

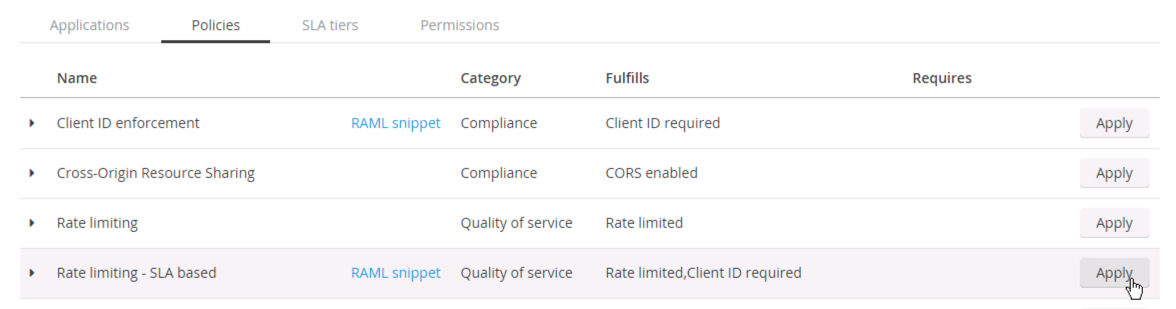
## Step 3: Add a Rate Limiting SLA-based Policy

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 the Anypoint Platform can properly enforce the contracted tier.

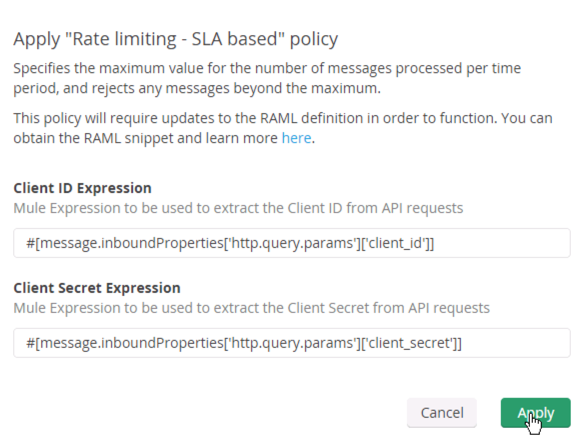
Let’s apply a rate-limiting policy to your endpoint.



1. Click the **Policies tab** to view the list of available policies for your organization. Select individual policies to read their descriptions.

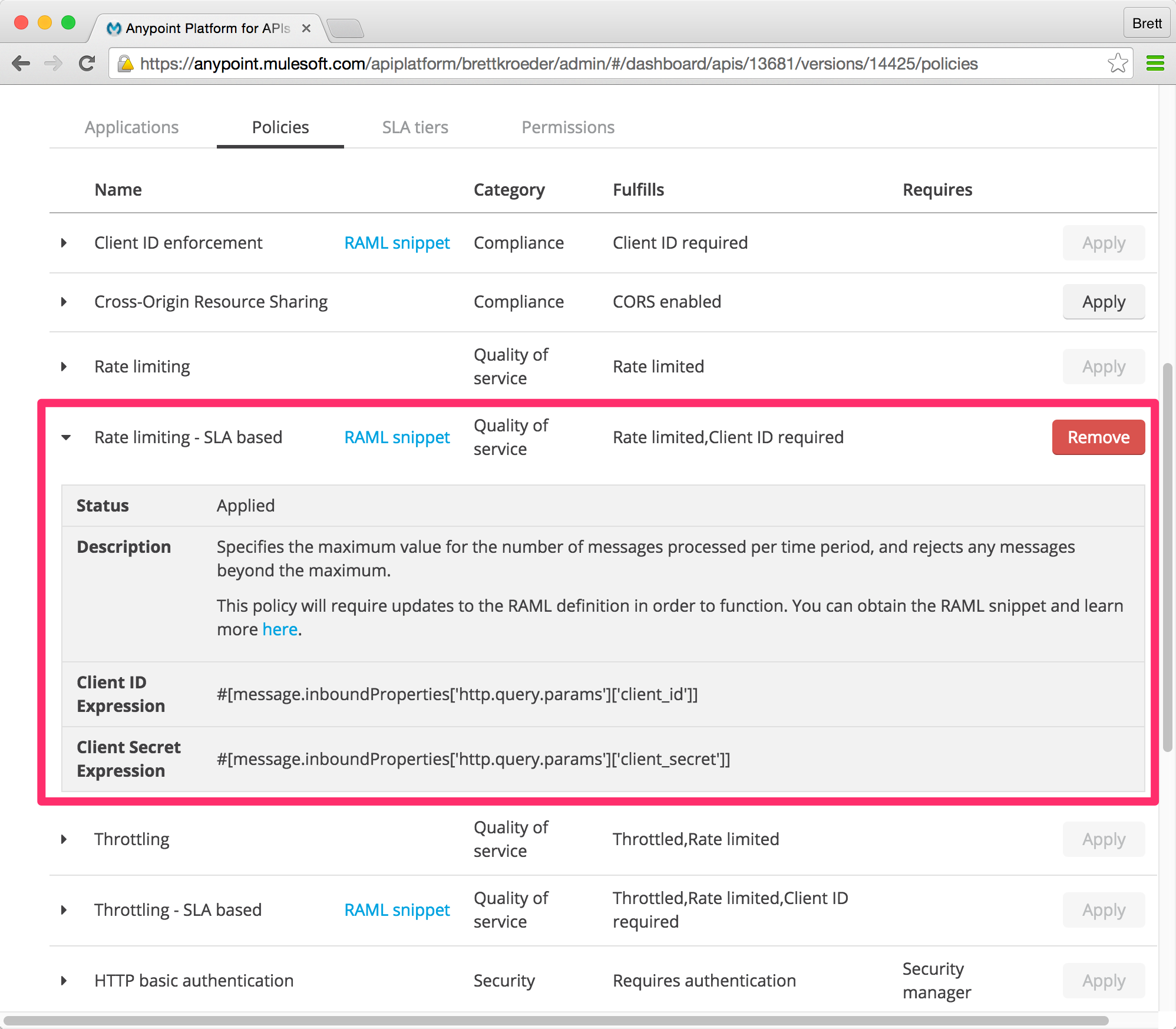


1. Click **Apply** on the *Rate limiting - SLA based* policy.



The Client ID and Client Secret Expressions are simple MEL statements. In this case, the policy is pulling the parameters from the HTTP header of the inbound request. But if you needed to pull them from somewhere else you could using MEL.

1. Click **Apply** to accept the default configuration for the Rate limiting SLA based policy.

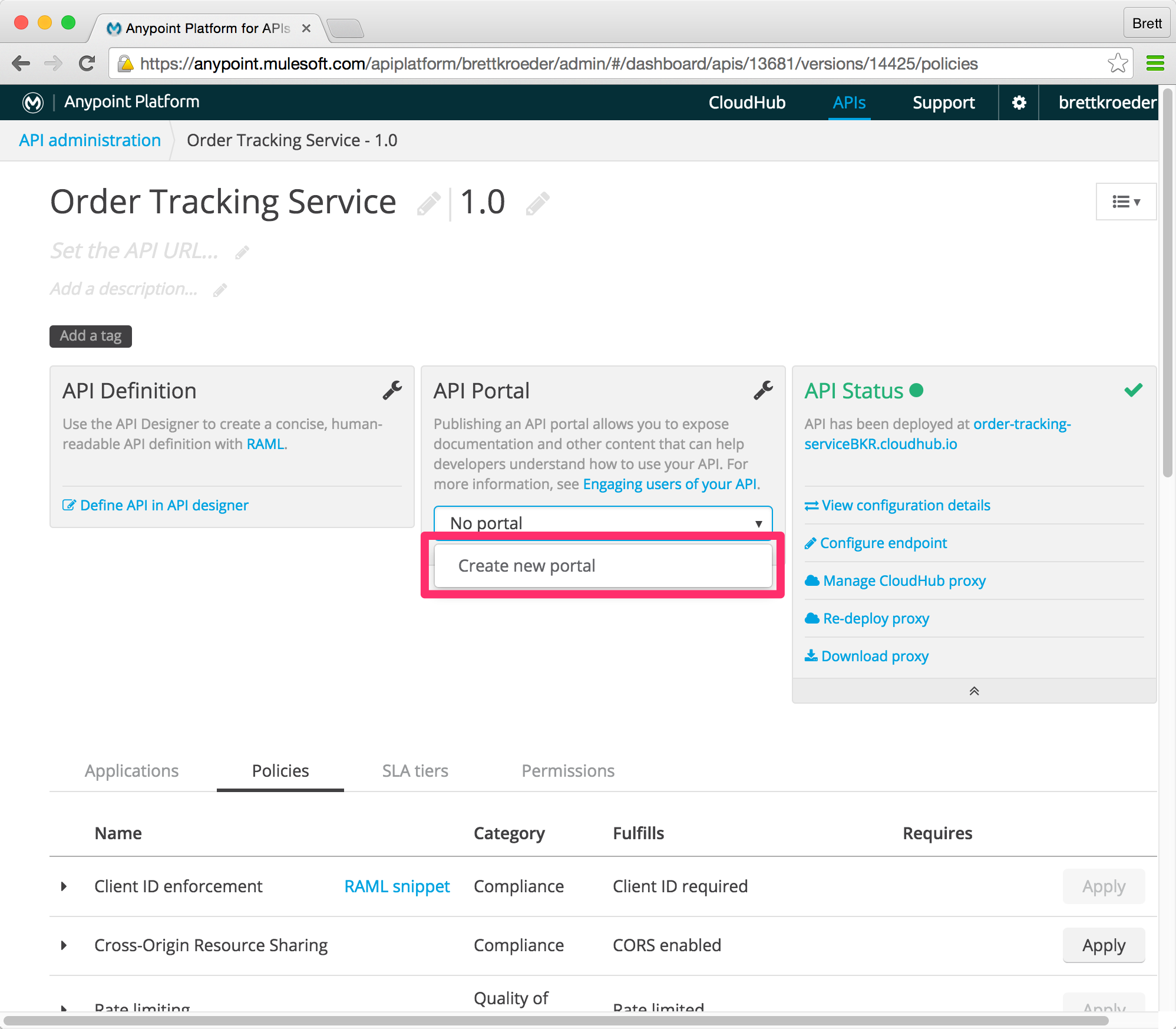


After you click Apply the  button will appear and you can expand the policy to see details about the policy. In this case you see the policy requires Client ID and Client Secret. You can see these properties are obtained from the http.query.params.

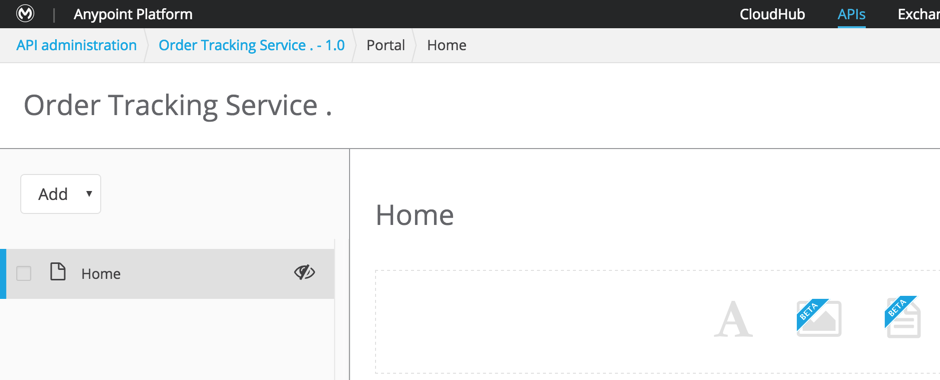
|  |
| --- |
| **NOTE**: Depending on the policy that you select, you may need to provide further configuration.  If the policy that you wish to apply has a disabled Apply link, it is not eligible to be applied to your endpoint. Either:   * you already have another policy applied which fulfills the same requirement (see the Fulfills column) or * the policy that you want to apply requires that another policy be applied first (see the Requires column)   To remove policies, you can click Remove. The policies are immediately removed from your endpoint. Note that if you wish to reapply the policy, you need to configure it again. Your previous configuration is not saved. |

## Step 4: Create an API Portal

Now we are going to create a basic **API portal**. The portal will enable a self-service API access registration process where consumers can request access with an associated service level. A later lab will explore more advanced API capabilities.

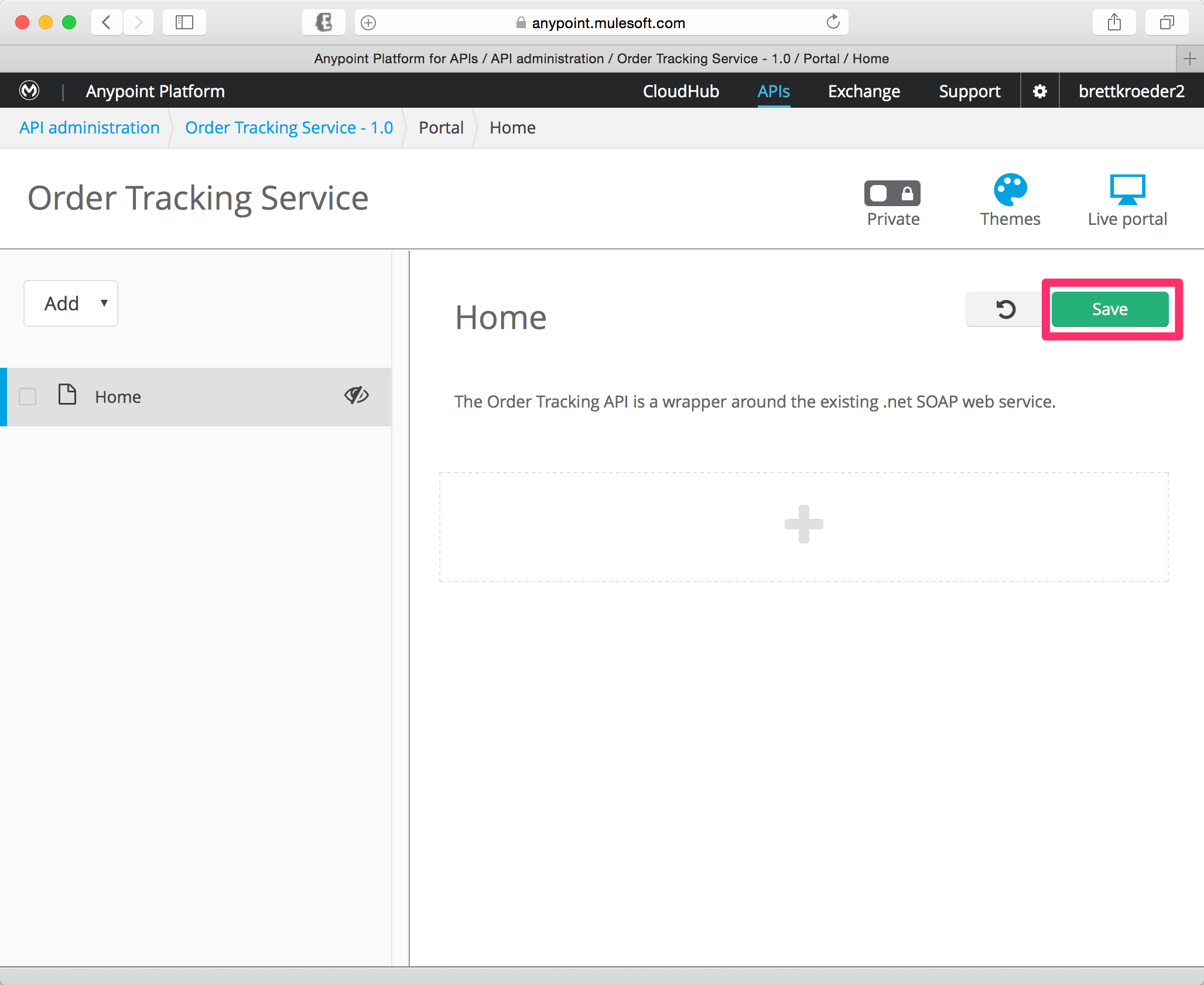


1. Scroll back up to the top of the API configuration page.
2. Click on the **No Portal** drop down and choose **Create new portal**. You will be brought to a page to edit your new portal.



This will create a new API portal page. Now you can add the components desired for your API.

1. Click the **Add** drop down to see a list of options that you have in your API Portal for managing content. At this point, we will not be creating anything. This is just to satisfy your curiosity about what can be added.



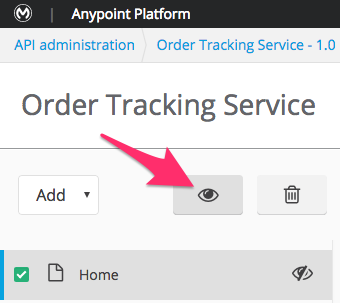
Your page is already in **Edit** mode.

1. Click on the to add text to the page
2. Add the following text to the page.

The Order Tracking API is a wrapper around the existing .net SOAP web service.

You’re page should look like above.

1. Click **Save**.

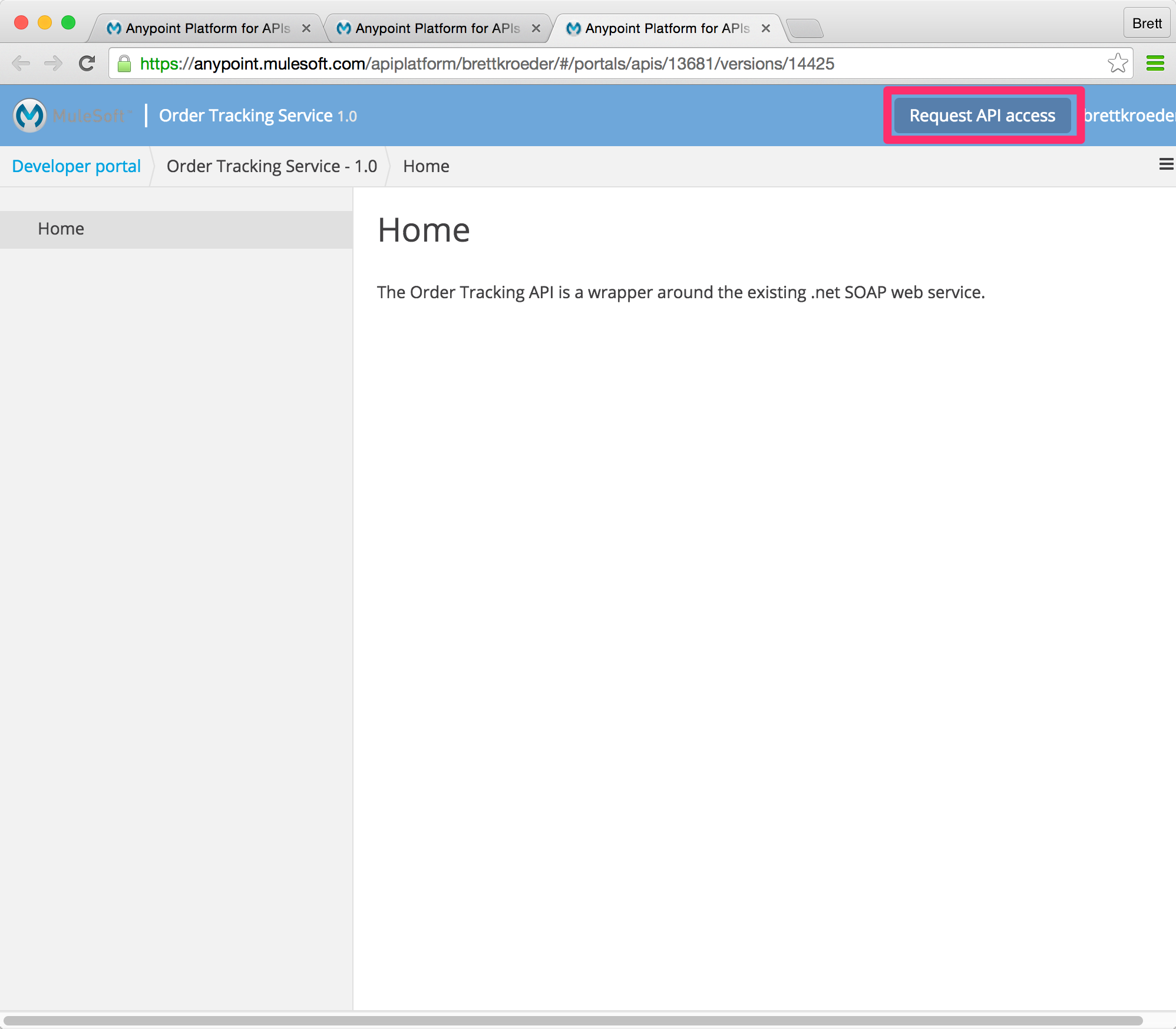


Now you need to make the *Home* component visible

1. Select the **checkbox** next to the **Home** page.
2. Make the page visible by clicking the  (visible) icon.
3. Make the Portal Public by clicking on  to toggle to 
4. Click on  to take the portal live and see your results.

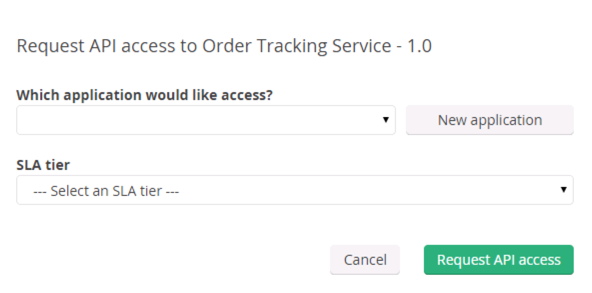
## Step 5: Request Access for the API

You now have a portal page for the API/SOAP Service that explains what it does and will allow Developers to register to use it.

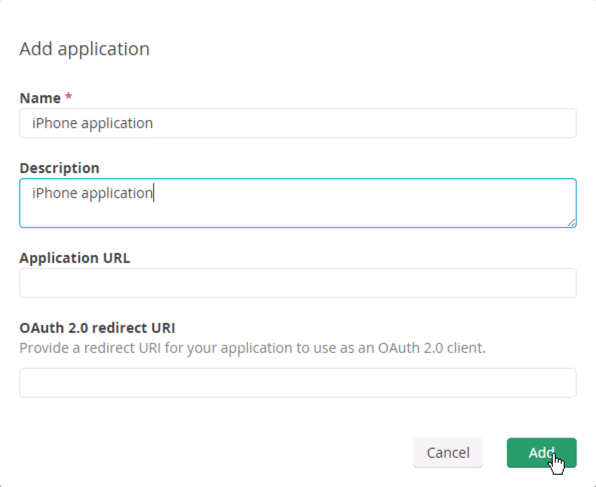


The live portal view will allow us to request a **client\_id** to access the *Employee Email Service API*.

1. In the top right corner of the portal, select **Request API Access.**



1. Click on **New application**.

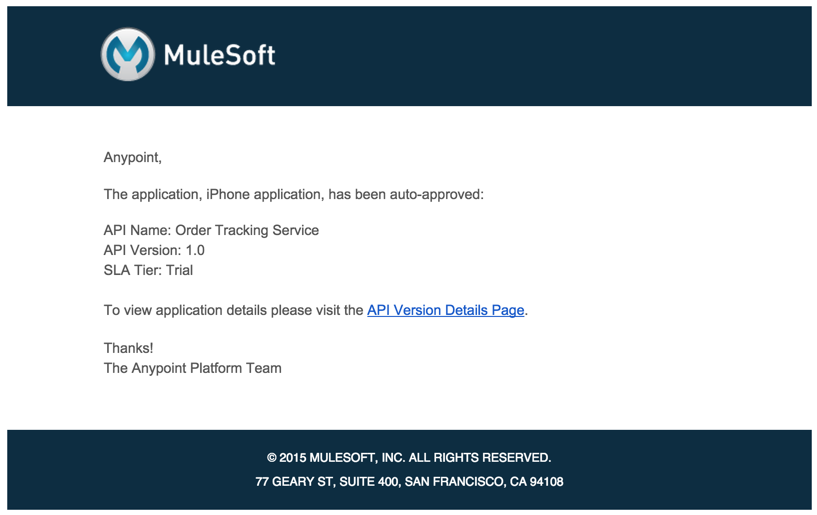


1. Complete the new application dialog.



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

1. Choose the **Trial** tier.
2. Click on the **Request** **API** **Access** button. By default, all API requests will be approved for the *Trial* SLA tier.

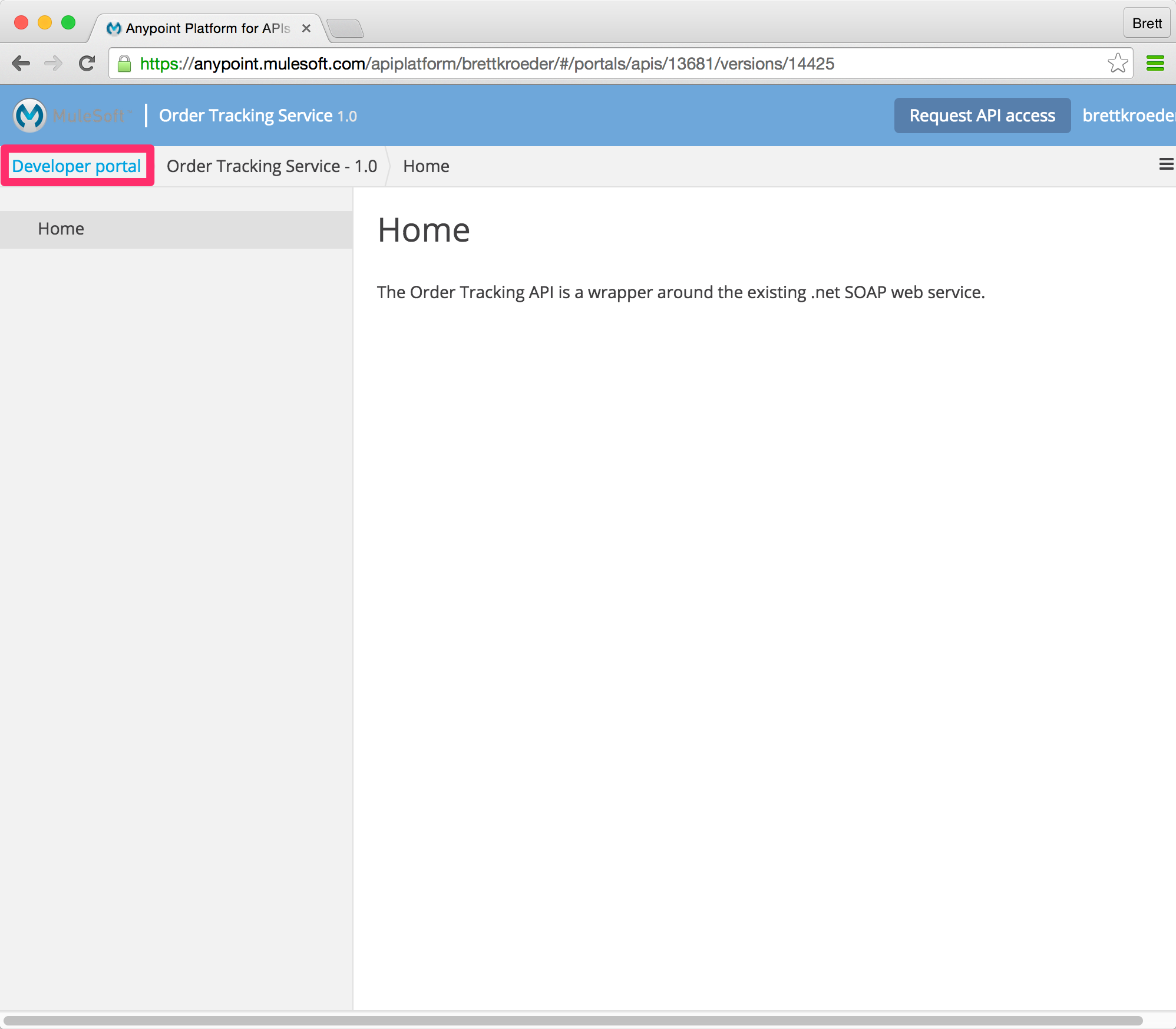


1. Go to your email
2. You should see the above registration email indicating you have been auto-approved.

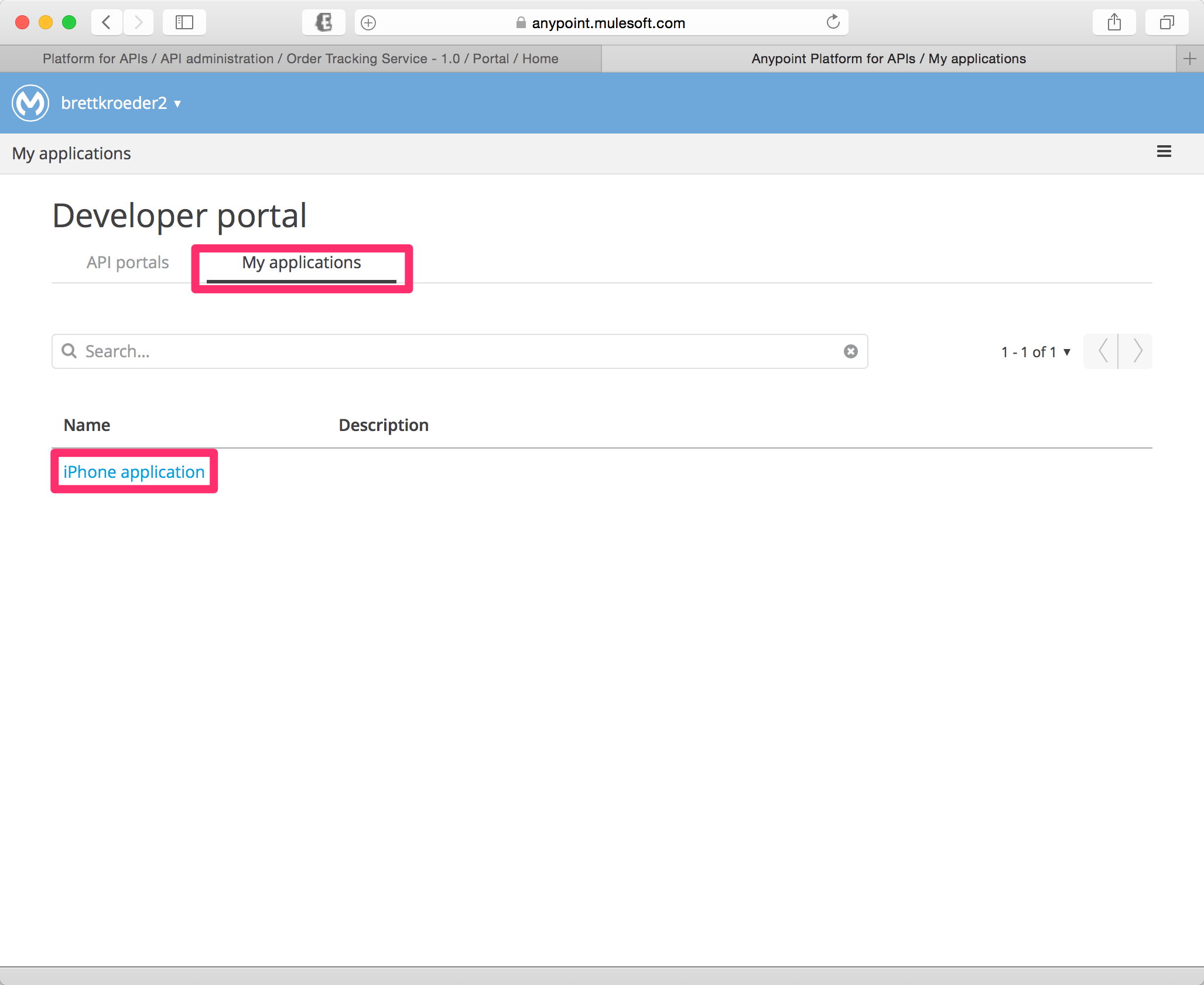
|  |
| --- |
| **NOTE**: If you set your tiers for manual approval, email notifications are sent to you when developers request access for their applications. You can review the applications on the Applications tab and approve, reject, or revoke requests. If a developer asks to change tiers, you can also review the change request and approve the application for the new tier or reject the change request. |

## Step 6: Get Your App Credentials

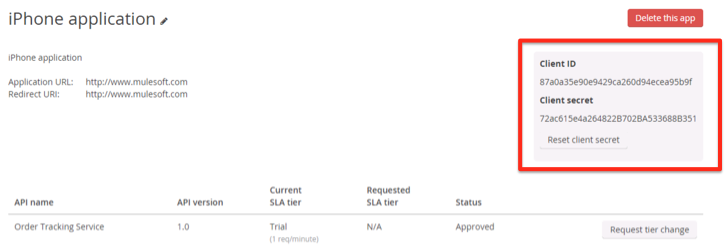
Each *application* has it’s own credentials. In order to call your managed API you need to supply these credentials which can be obtained from the Developer Portal.



1. Go back to the browser.
2. Click on **Developer portal**.

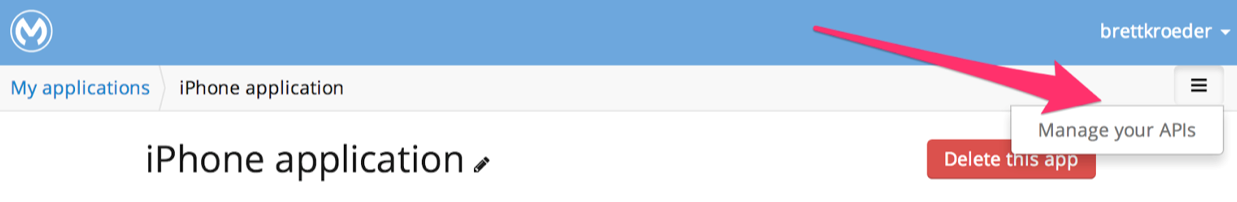


1. Click on **My applications**
2. Click on the application you just created.



You’ll find your **Client ID** and **Client secret** on this page.

1. Record these values as you will use them in the next step to invoke the API.



1. Select the **Admin** button at the top right of the page
2. Select **Manage your APIs**

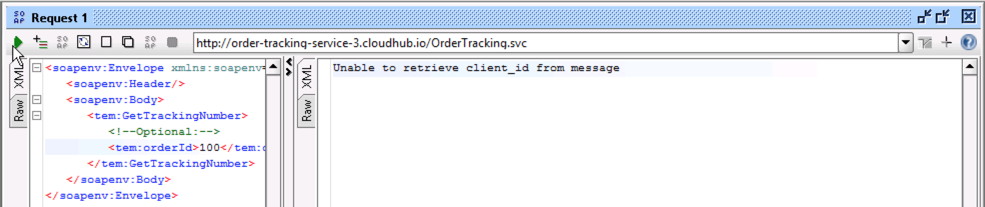
This will return you to the main API administration page.

## Step 7: Test with SoapUI

You will now test the SOAP service by using the SoapUI.

1. If not already open, open SoapUI from the Desktop 
2. **Re-open** the previous OrderTracking project to re-test the SOAP Service.
3. Select **Request 1** in the GetTrackingNumber operation

|  |
| --- |
| **DOUBLE CHECK**: Are you sure you opened the **GetTrackingNumber** operation? |



1. Enter an order ID of **100** in the Request.

<tem:orderId>100</tem:orderId>

|  |
| --- |
| **DOUBLE CHECK**: Are you sure you entered **100**? |

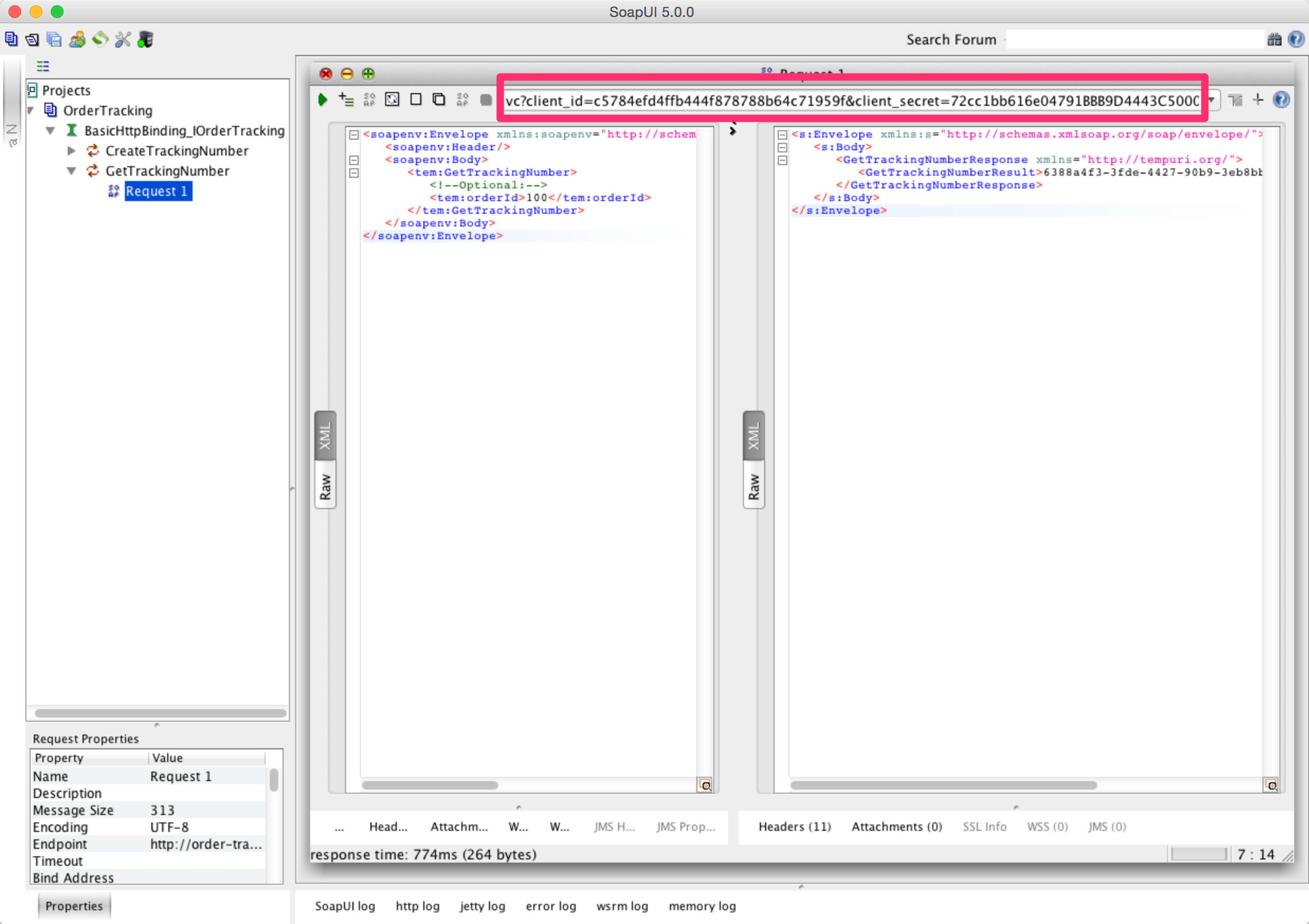
1. Click the Play icon

You should see the response:

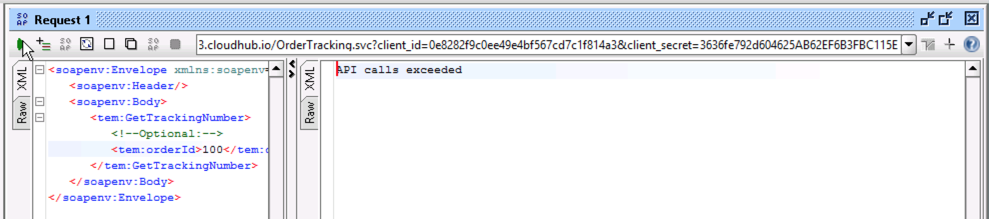
Unable to retrieve client\_id from message

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

## Step 8: Test the API with Credentials



1. Add ?client\_id=*<yourId>*&client\_secret=*<yourSecret>* to the request URL.
2. Click the Play button to test with the application credentials.



1. 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:

[Step 1: Apply Rate Limiting Policy](#_edetyfd26erb)

[Step 2: Create SLA Tiers](#_izq5ps7y6crt)

[Step 3: Add a Rate Limiting SLA-based Policy](#_kkj894bxexeo)

[Step 4: Create an API Portal](#_aec2o4k4jfsy)

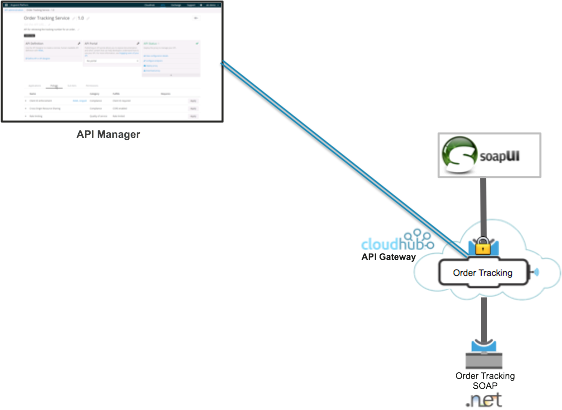
[Step 5: Request Access for the API](#_8iei5doca5lj)

[Step 6: Get Your App Credentials](#_f4pou9yf65r2)

[Step 7: Test with SoapUI](#_8uybtme35shu)

[Step 8: Test the API with Credentials](#_haiur5lcz8k9)

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](http://www.mulesoft.org/documentation/display/current/Applying+Runtime+Policies)
* Learn more about security policy configurations:
  + [LDAP policy](http://www.mulesoft.org/documentation/display/current/LDAP+Security+Manager)
  + [OAuth 2.0 Provider and AES Oauth 2.0 Token Enforcement](http://www.mulesoft.org/documentation/display/current/OAuth+2.0+Provider+and+AES+OAuth+2.0+Token+Enforcement+Policies)
  + [PingFederate](http://www.mulesoft.org/documentation/display/current/PingFederate+OAuth+Token+Enforcement+Policy)
  + [OpenAM OAuth Token Enforcement](http://www.mulesoft.org/documentation/display/current/OpenAM+OAuth+Token+Enforcement+Policy)
* In addition to the OOTB policies, you saw in this lab, you can also   
  [Create Custom Policies](http://www.mulesoft.org/documentation/display/current/Creating+a+Policy+Walkthrough)
* Learn more about [Managing API Versions](http://www.mulesoft.org/documentation/display/current/Managing+API+Versions)
* Learn more about [SLA Tiers](http://www.mulesoft.org/documentation/display/current/Defining+SLA+Tiers)

Congratulations! You have finished Lab 2.

Please update the spreadsheet indicating you have completed Lab 2.