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 Anypoint Gateway to Mythical Corp’s Omni Channel API on CloudHub, now you can:

- Manage security and enforce policies
- Manage contracts associated with that API

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 proxy gateway for Mythical Corp’s Omni Channel 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**.
NOTE You can apply the policies to all or to specific methods and resources

IMPORTANT Make sure you set the Time Period to Minutes to see the rate limit take effect.

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

7. Wait for a minute (the API Gateway 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. 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 Proxy URL adding /products/search at the end (e.g. http://fb-mythical-omni-channel-api-proxy.cloudhub.io/products/search).

9. Test it again by reloading the browser 3 times.

10. On the 3th 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:

<table>
<thead>
<tr>
<th>Tier</th>
<th>Approval</th>
<th>Throughput</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trial</td>
<td>Automatic</td>
<td>1</td>
<td>Minute</td>
</tr>
<tr>
<td>Gold</td>
<td>Manual</td>
<td>10</td>
<td>Second</td>
</tr>
<tr>
<td>Platinum</td>
<td>Manual</td>
<td>100</td>
<td>Second</td>
</tr>
</tbody>
</table>

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**.
WARNING | Select the last version according to runtime 4.1.x

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

**NOTE**

Please note the following for other use cases: Depending on the policy that you select, you may need to provide further configuration.

If the policy that you wish to apply is greyed out, 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 filter)
- 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. Users can also edit applied policies.

**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.

**NOTE** You can see in the Versions panel that there is another instance added. That is the application we deployed.

3. Press **Request Access**.

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.

**Important**

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

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

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

14. Click on **My Applications**. You will find the API you’ve just created.
15. Click on the API you’ve just created and you will get the same page you saw when you created.

16. Go to your email

17. You should see the below 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 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: 

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-proxy/products/search?client_id=b466e22597b94689952bb77792cf7f8d&client_secret=53EDAFEDeA6f45dF95408596f846417F

   **NOTE** 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
  - OpenAM and PingFederate 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*
- Learn more about *SLA Tiers*
Congratulations! You have completed Lab 3.

Please proceed to Lab 3

Take me to the TOP