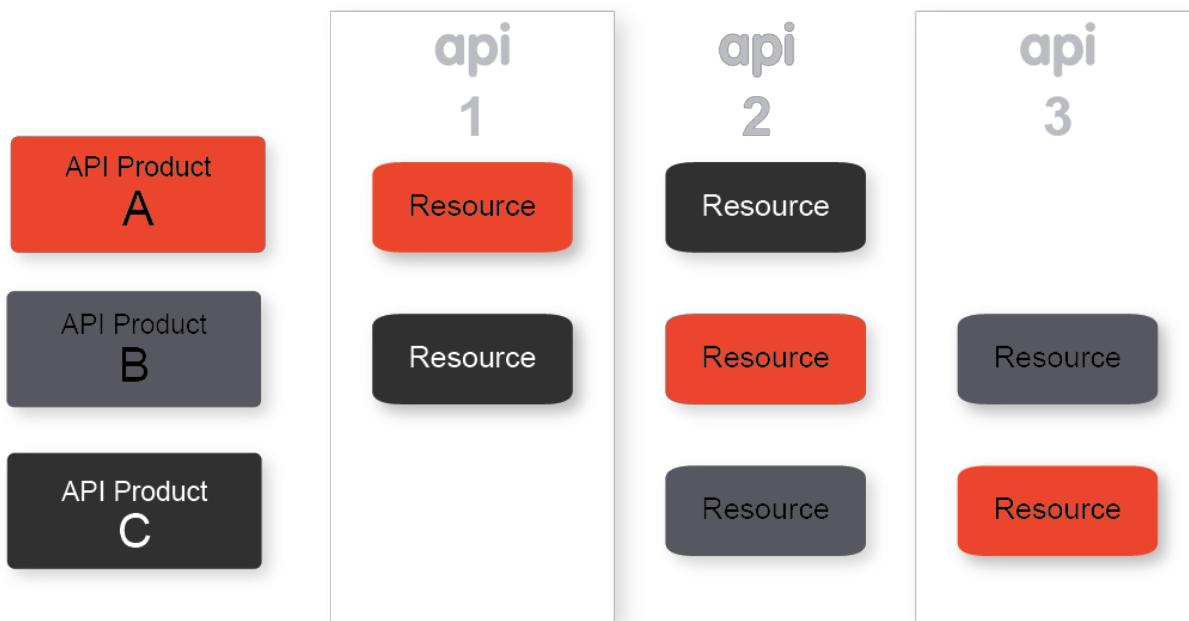


Lab 2: Developer Services - API Products & Developer Management

Overview

As an API provider, you need to create an API product. The API product is the mechanism through which your APIs are bundled and published so that developers can consume them. An API product is a collection of API resources (URIs) combined with a service plan and presented to developers as a bundle. The API product can also include some metadata specific to your business for monitoring or analytics. You can think of API products as your product line. You can create different products to provide features for different use cases. So instead of just giving developers a list of resources, you can bundle specific resources together to create a product that solves a specific user need.

API products are also a good way to control access to a specific bundle of resources. For example, you can bundle resources that can only be accessed by internal developers, or bundle resources that can only be accessed by paying customers. The API resources bundled in a product can come from one or more APIs, so you can mix and match resources to create specialized feature sets.

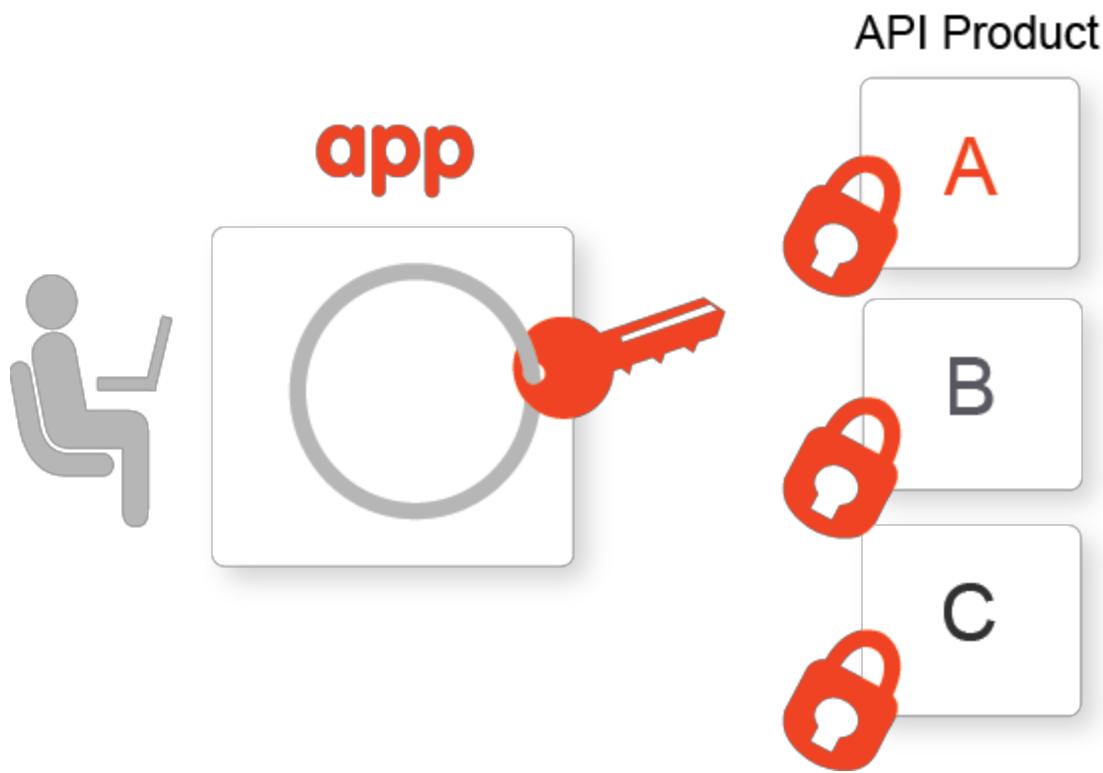


Developers access your APIs through apps. When the developer registers an app, they receive a single API key that allows them to access all of the API products associated with the app. However, developers must be registered before they can register an app.



Developers register apps to access your API products. When a developer registers an app, the developer selects the API products to associate with the app and Edge generates an API key. Each app has a single API key that provides access to all API products associated with the app.

Apps allow you to control who can access your APIs. You can revoke an app's key, preventing it from accessing all API products. Or you can revoke access to a single API product associated with the app.



In this lab you will use the Apigee API Console to create API Products, Developers, Developer Apps, and allow the apps to access API products. You will also configure the API Product to apply quotas on calls to all APIs contained in that product. Finally you will add API Key Verification to the proxy and invoke the proxy with the API Key to access the API.

Objective

After this lab you should be able to create API Products, add developers, register apps and add API Key verification to secure the proxy and more importantly identify what app is accessing the proxy. This information can be used to provide dynamic behaviour for different apps trying to access the same API Proxy.



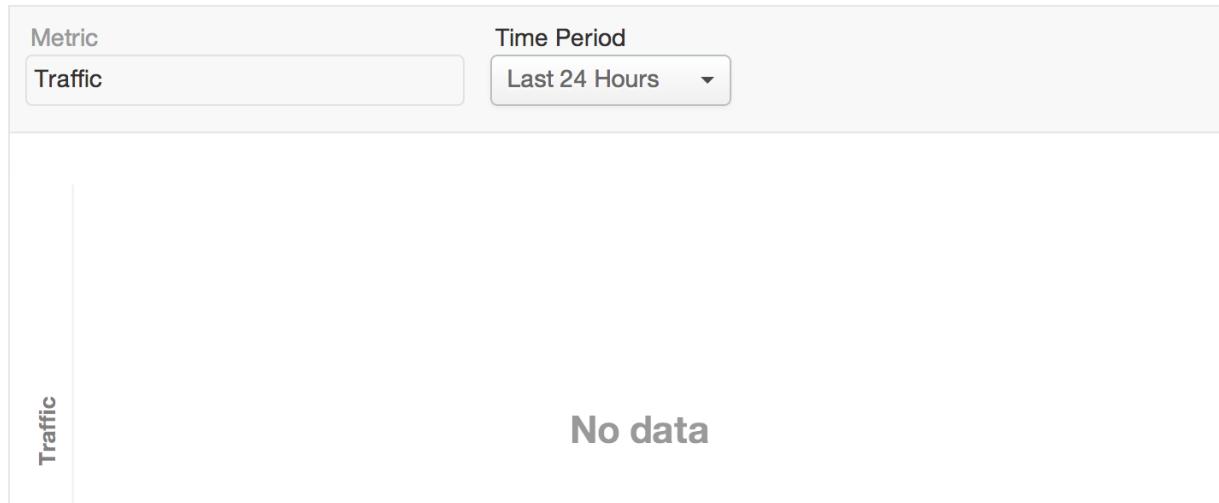
Estimated Time: 30 mins

1. Login to the Apigee Edge Management User Interface (Management UI). On the top menu, click on the Publish item and then click on “Products”. You should see something like this.

The screenshot shows the Apigee Management UI. The top navigation bar has tabs: apigee, Dashboard, APIs, Publish (which is highlighted in orange), Analytics, Admin, and Help. Below the navigation, a breadcrumb trail shows 'Dashboard / Products'. A dropdown menu under 'Products' is open, showing 'Products' (which is also highlighted in orange), 'Developers', and 'Developer Apps'. The main content area is titled 'Products' and contains a search bar. A table lists one product: 'Premium Weather API' with 1 key, created on Aug 7, 2014 at 1:21:34 PM, and last modified on the same date and time.

| Product | Keys | Created On | Last Modified |
|---------------------|------|------------------------|---------------------|
| Premium Weather API | 1 | Aug 7, 2014 1:21:34 PM | Aug 7, 2014 1:21:34 |

Performance



2. Create an API Product - In this step we will create an API product and assign a quota to it. Click on the '+ Product' button on the top right. Enter the following values in the fields:



- a. **Display Name:** `{Your Initials}_Open_Weather_Bronze`
- b. **Description:** This is the bronze product offering for the Open Weather API
- c. **Environment:** Select test .
- d. **Access:** Public
- e. **Key Approval Type:** Automatic
- f. **Quota:** 5 requests every 1 minute
- g. **Allowed OAuth Scopes:** Don't enter anything in OAuth scopes.

- h. **Resources:** In the API Proxies section select the `{Your Initials}_Open_Weather API Proxy`.

Product Details

| | |
|--|--|
| Display Name | <input type="text" value="PB_Open_Weather_Bronze"/> |
| Description | <input type="text" value="This is the bronze offering for the Open Weather API"/> |
| Environment | <input checked="" type="checkbox"/> test <input type="checkbox"/> prod |
| Access | <input type="radio"/> Internal only — Visible only to developers at devjam_admin during app registration <input type="radio"/> Private — Visible only to external developers with explicit permission during app registration <input checked="" type="radio"/> Public — Visible only to any registered developer during app registration |
| Key Approval Type | <input checked="" type="radio"/> Automatic <input type="radio"/> Manual <small>Learn more</small> |
| Quota | <input type="text" value="5"/> <input type="button" value=""/> requests every <input type="text" value="1"/> <input type="button" value=""/> minute <input type="button" value=""/> |
| Allowed OAuth Scopes | <input type="text"/> |
| <small>Comma separated scope names. Learn more</small> | |

Resources

| | API Proxy | Revision | Resource Path | |
|-------------|-----------------|-----------|---------------|--|
| | Select... | Select... | Select... | <input type="button" value="Import Resource"/> |
| Paths | Resource Path | | | <input type="button" value="+ Custom Resource"/> |
| API Proxies | API Proxy | | | <input type="button" value="Actions"/> |
| | PB_Open_Weather | | | <input type="button" value="Delete"/> |
| | | | | <input type="button" value="+ API Proxy"/> |

- i. Finally click on the blue Save button

3. Create a Developer - In this step we will add a Developer. From the Management UI, click on the Publish item and then click on “Developers”. You should see something like this:



apigee Dashboard APIs Publish Analytics Admin Help pbhogill@apigee.com API Management ▾

Dashboard / Developers Products Developers Developer Apps Organization pbhogill Environment prod Dev Portal

Developers 1—5 of 5 Export ▾ + Developer

| Developer | Email | Username | Apps | Keys | Actions |
|-------------------|--------------------------|------------|------|------|---------------------------|
| Thomas Edison | thomas@weathersample.com | tom_edison | 1 | 1 | <button>X Delete</button> |
| Nikolai Tesla | tesla@weathersample.com | Nikolai | 1 | 1 | <button>X Delete</button> |
| Prithpal Bhogill | pbhogill@apigee.com | pbhogill | 3 | 5 | <button>X Delete</button> |
| Catalog Developer | pbhogill+cdev@apigee.com | cdev | 1 | 2 | <button>X Delete</button> |
| Josiah Carberry | joe@weathersample.com | joe | 1 | 1 | <button>X Delete</button> |

Performance Export ▾ Traffic

Metric: Traffic Time Period: Last 24 Hours Show Moving Averages Show Alerts Investigate Anomalies

No data

Click on the “+ Developer” button and enter the following values:

- First Name: {Your Initials}
- Last Name: Developer.
- Email: {Your Email Address}
- Username: {Your Initials}_developer
- Click on the blue “Save” button and you should see something like this:



New Developer

Developer Details

| | |
|------------|----------------------------|
| First Name | PB |
| Last Name | Developer |
| Email | pbhogill+devjam@apigee.com |
| Username | pb_developer |

Custom Attributes

| Name | Value | Actions |
|------|-------|------------------------------------|
| | | + Custom Attribute |

[Cancel](#) [Save](#)

- f. You should now see yourself in the list of developers.
4. Create an Developer App - From the Management UI, click on the Publish item and then click on “Developer Apps”. You should see something like this:

| App | Developer | App Family | Company | Key | Metrics for Last 24 Hours | | Registered | Actions |
|------------------------|-------------------|------------|---------|-----|---------------------------|----------------|--------------------------|------------------------|
| | | | | | Traffic | Error Rate (%) | | |
| SFApp | Prithpal Bhogill | default | | 1 | | | Jul 21, 2014 9:19:55 PM | Delete |
| Catalog Developers App | Catalog Developer | default | | 1 | | | Jun 9, 2014 10:54:25 AM | Delete |
| Product Developers App | Prithpal Bhogill | default | | 1 | | | Jun 9, 2014 10:24:45 AM | Delete |
| Josiah's Weather App | Josiah Carberry | default | | 1 | | | Apr 30, 2014 11:22:47 AM | Delete |
| Tom's Weather App | Thomas Edison | default | | 1 | | | Apr 30, 2014 11:22:46 AM | Delete |
| Weather App | Nikolai Tesla | default | | 1 | | | Apr 29, 2014 12:56:49 PM | Delete |

Performance

Metric: Traffic Time Period: Last 24 Hours

Show Moving Averages
 Show Alerts
 Investigate Anomalies



Click on the “+ Developer App” button and enter the following values:

- a. Display Name: `{Your Initials}_OpenWeatherApp`
- b. Developer: Select yourself as the Developer
- c. Callback URL:
- d. Notes: This is the main developer app for the Open Weather API
- e. Products: Pick the `{Your Initials}_OpenWeather_Bronze` product and make sure to click on the “check” button
- f. Click on the blue “Save” button

New Developer App

Developer App Details

| | |
|---|------------------------------------|
| Display Name | PB_Open_Weather_App |
| Developer | Prithpal Bhogill (pbhogill+dev...) |
| Callback URL | |
| A callback URL is only required for 3-legged OAuth. | |
| Notes | |

Products

| Product | Status | Consumer Key | Consumer Secret | Actions |
|------------------------|--------|--------------|-----------------|-------------------------|
| PB_Open_Weather_Bronze | | | | <button>Remove</button> |

+ Product

Custom Attributes

| Name | Value | Actions |
|------|-------|-------------------------------------|
| | | <button>+ Custom Attribute</button> |

Cancel Save

- g. Once you save the developer app, you will see your app in the list of other developer apps. Click on your recently created developer app and click on the “Show” button in the Products section to reveal the Consumer Key. (Save this value on a textpad or somewhere else. You will need to supply this value when making a call to the API later). You will see something like this:



PB_Open_Weather_App

Regenerate Key Edit Delete

Developer App Details

Display Name PB_Open_Weather_App
Registered Oct 18, 2014 3:46:23 PM
Developer Prithpal Bhogill (pbhogill+devjam@apigee.com)

Callback URL

Notes

Products

| Product | Status | Consumer Key | Consumer Secret |
|------------------------|----------|----------------------------------|---|
| PB_Open_Weather_Bronze | Approved | Yz71TmsBwZik0qzdnS04N7wkm8czB9lp | <button>Hide</button> <button>Show</button> |

Custom Attributes

| Name | Value |
|------|-------|
| | |

5. Add a “Verify API Key” Policy. The Verify API Key policy lets you enforce verification of API keys at runtime, letting only approved apps access your API resources. This policy ensures that keys are valid, have not been revoked, and are approved to consume specific API resources contained in your API products.

When you add your API resources to API products (best practice), a unique key is generated for each registered developer app that uses those products. But API keys are meaningless unless you use this policy to verify them. Without this policy, your API is wide open. Visit this [link](#) for more details.

- From the Management UI, click on the APIs menu item and click on the “{Your Initials}_Open_Weather” API proxy.
- Click on the “Develop” tab from the top right.
- From the proxy editor, select “New Policy” from the top and select “Verify API Key” policy.



PB_Open_Weather

OVERVIEW DEVELOP TRACE

Navigator

- Policies
 - Extract Variables 1
 - Response Cache 1
 - Spike Arrest 1
 - Statistics Collector 1
- Proxy Endpoints
 - default
 - All PreFlow
 - All PostFlow
- Target Endpoints
 - default
 - All PreFlow
 - All PostFlow
- Scripts

Deployed to Environment: test

Help for Selected Flow

PreFlow
PreFlow name PreFlow
PreFlow / Request
PreFlow / Request / Step
FaultRules
Name Spike-Arrest-1
PreFlow / Request / Step
FaultRules
Name Response-Cache-1
PreFlow / Response
PreFlow / Response / Step
FaultRules
Name Extract-Variables-1
PreFlow / Response / Step
FaultRules
Name Statistics-Collector-1

© 2014 Apigee Corp. All rights reserved. Version 141015

- d. Attach it to the “Request” segment of the Proxy Pre Flow.

New Policy: Verify API Key

| | |
|---------------------|---|
| Policy Display Name | Verify API Key 1 |
| Policy Name | Verify-API-Key-1 |
| Attach Policy | <input checked="" type="checkbox"/> |
| Flow | Flow PreFlow, Proxy Endpoint default |
| Segment | <input checked="" type="radio"/> Request <input type="radio"/> Response |

[Cancel](#)

[Add](#)

- e. Click on the “Verify API Key” policy and make sure the XML snippet in the bottom window looks like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<VerifyAPIKey async="false" continueOnError="false"
enabled="true" name="Verify-API-Key-1">
    <DisplayName>Verify API Key 1</DisplayName>
    <FaultRules/>
    <Properties/>
    <APIKey ref="request.queryparam.apikey" />
</VerifyAPIKey>
```

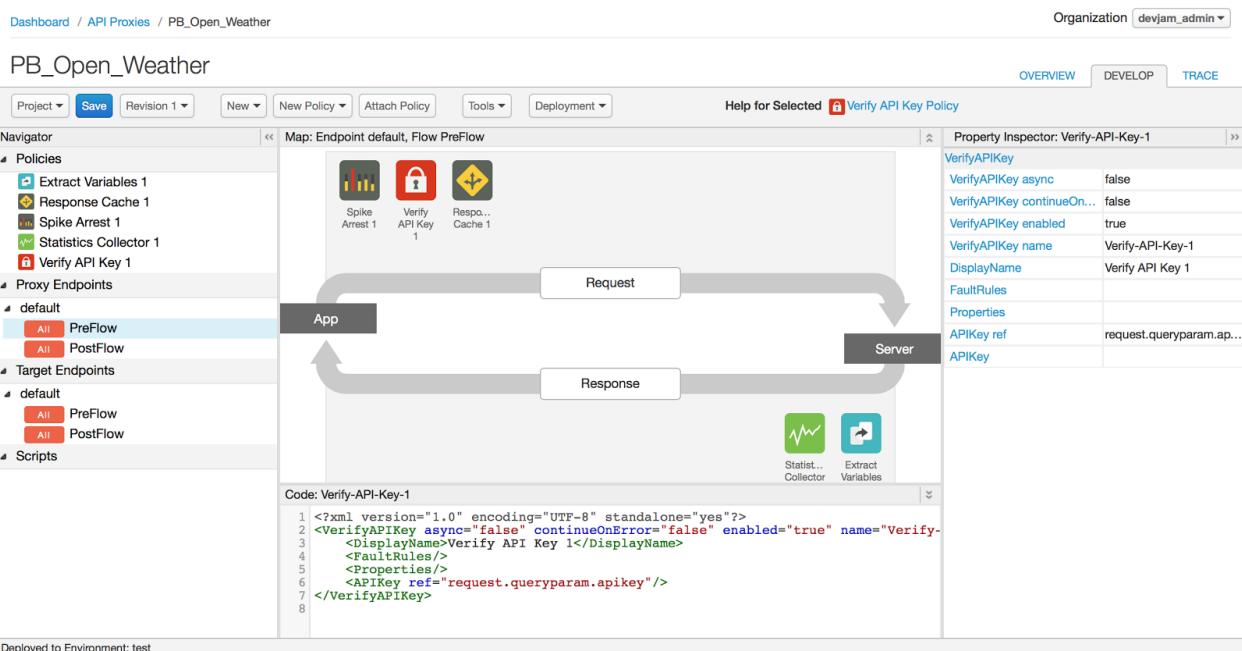


You can also copy/paste the xml snippet from [here](#) into your policy editor.

(<https://gist.github.com/prithpal/60ed9e528bae06ce7a0b/raw/18b6a3d1979285811cb9f9ae8bb849c2a94db122/Developer%20Services%20-%20API%20Products%20&%20Developer%20Management%20-%20Verify%20API%20Key>)

In this case we have used `request.queryparam.apikey` as the place where the proxy should expect to see the API key, you can also use `request.header.apikey` if you want to read the API key from the request header instead.

- Drag the “Verify API Key” policy so it appears right after the “Spike Arrest Policy”



- Lets add a “Quota” policy to the proxy. You can use the Quota policy to configure the number of request messages that an app is allowed to submit to an API over the course of an hour, day, week, or month. For additional details, visit this [link](#).

- When in side the proxy editor, select “New Policy” from the top and select “Quota”.



Dashboard / API Proxies / PB_Open_Weather

Organization devjam_admin

PB_Open_Weather

OVERVIEW DEVELOP TRACE

Navigator

- Policies
 - Extract Variables 1
 - Response Cache 1
 - Spike Arrest 1
 - Statistics Collector 1
 - Verify API Key 1
- Proxy Endpoints
- default
 - All PreFlow
 - All PostFlow
- Target Endpoints
- default
 - All PreFlow
 - All PostFlow
- Scripts

TRAFFIC MANAGEMENT

- Quota
- Spike Arrest
- Concurrent Rate Limit
- Response Cache
- Lookup Cache
- Populate Cache
- Invalidate Cache
- Reset Quota

SECURITY

- Basic Authentication
- XML Threat Protection
- JSON Threat Protection
- Regular Expression Protection
- OAuth v2.0
- Get OAuth v2.0 Info
- Set OAuth v2.0 Info
- OAuth v1.0a
- Get OAuth v1.0a Info

Help for Selected Verify API Key Policy

Property Inspector: Verify-API-Key-1

| | |
|------------------------------|---------------------------|
| VerifyAPIKey | |
| VerifyAPIKey async | false |
| VerifyAPIKey continueOnError | false |
| VerifyAPIKey enabled | true |
| VerifyAPIKey name | Verify-API-Key-1 |
| DisplayName | Verify API Key 1 |
| FaultRules | |
| Properties | |
| APIKey ref | request.queryparam.apikey |
| APIKey | |

Request → Server → Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Quota async="false" continueOnError="false" enabled="true" name="Verify-
    API Key 1"><DisplayName>Quota 1</DisplayName>
    <Request><!-- request parameters -->
        <!-- request headers -->
        <!-- request body -->
        <!-- query parameters -->
        <!-- cookie parameters -->
    </Request>
    <!-- response headers -->
    <!-- response body -->
</Quota>

```

request.queryparam.apikey />

Deployed to Environment: test

© 2014 Apigee Corp. All rights reserved. Version 141015

- b. Attach it to the Request segment of the Pre-Flow of the Proxy.

New Policy: Quota

Policy Display Name

Policy Name

Attach Policy

Flow

Segment Request Response

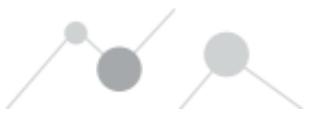
Add

- c. Click on the “Quota” policy and make sure the XML snippet in the bottom window looks like this:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Quota async="false" continueOnError="false"
enabled="true" name="Quota-1">
    <DisplayName>Quota 1</DisplayName>

```



```
<Identifier ref="request.queryparam.apikey" />
<Interval
ref="verifyapikey.Verify-API-Key-1.apiproduct.develop
er.quota.interval" />
<TimeUnit
ref="verifyapikey.Verify-API-Key-1.apiproduct.develop
er.quota.timeunit" />
<Allow
countRef="verifyapikey.Verify-API-Key-1.apiproduct.de
veloper.quota.limit" />
<Distributed>true</Distributed>
<Synchronous>true</Synchronous>
</Quota>
```

You can also copy/paste the xml snippet from [here](#) into your policy editor.

(<https://gist.githubusercontent.com/prithpal/71cc0fae36582bfb0ccb/raw/10acfb039c761e0c43de025f14d0797f7da9e5b7/Developer%20Services%20-%20API%20Products%20and%20Developer%20Management%20-%20Quota>)

This configuration allows the API Keys to reference the quota intervals etc. from the corresponding API products definition.

- d. Drag the newly added Quota policy to make sure that it appears right after the “Verify API Key” policy on the pre-flow. Using the Quota policy you can control how many times the API gets invoked over a time duration depending on whether its being accessed directly or as part of an API product / App.



Dashboard / API Proxies / PB_Open_Weather

Organization devjam_admin

PB_Open_Weather

Project Save Revision 1 New Policy Attach Policy Tools Deployment Help for Selected Quota Policy

Navigator

- Variables Extract Variables 1
- Quotas Quota 1
- Response Cache Response Cache 1
- Spike Arrest Spike Arrest 1
- Statistics Collector Statistics Collector 1
- Verify API Key Verify API Key 1
- Proxy Endpoints
- default
 - All PreFlow
 - All PostFlow
- Target Endpoints
- default
 - All PreFlow
 - All PostFlow
- Scripts

Map: Endpoint default, Flow PreFlow

Property Inspector: Quota-1

| | |
|-----------------------|--|
| Quota | |
| Quota async | false |
| Quota continueOnError | false |
| Quota enabled | true |
| Quota name | Quota-1 |
| DisplayName | Quota 1 |
| Identifier ref | request.queryparam.apikey |
| Identifier | |
| Interval ref | verifyapikey.Verify-API-Key-1.apiproduct.developer.quota.interval |
| Interval | |
| TimeUnit ref | verifyapikey.Verify-API-Key-1.apiproduct.developer.quota.timeunit |
| TimeUnit | |
| Allow countRef | verifyapikey.Verify-API-Key-1.apiproduct.developer.quota.allowCountRef |
| Allow | |
| Distributed | true |
| Synchronous | true |

Code: Quota-1

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <Quota async="false" continueOnError="false" enabled="true" name="Quota-1">
3   <displayName>Quota 1</displayName>
4   <identifier ref="request.queryparam.apikey" />
5   <interval ref="verifyapikey.Verify-API-Key-1.apiproduct.developer.quota.interval" />
6   <timeUnit ref="verifyapikey.Verify-API-Key-1.apiproduct.developer.quota.timeunit" />
7   <allow countRef="verifyapikey.Verify-API-Key-1.apiproduct.developer.quota.allowCountRef" />
8   <distributed>true</distributed>
9   <synchronous>true</synchronous>

```

Deployed to Environment: test

© 2014 Apigee Corp. All rights reserved. Version 141015

7. Test the API Proxy - We will test the updated {Your Initials}_Open_Weather API Proxy in two separate ways, one with the Consumer key & one without.

- Click on the Trace tab (top right) of the OpenWeather proxy and click on the green “Start Trace” button to start a new trace session.
- Switch to the browser and open the POSTMAN client if not already open.
- Access the proxy using the following URL
http://{org_name}-{env_name}.apigee.net/v1/{your_initials}_open_weather?q=Los%20Angeles,%20USA
- You should see an error like this:



Body Headers (3) STATUS 401 Unauthorized TIME 1914 ms

Pretty Raw Preview JSON XML

```

1  {
2      "fault": {
3          "faultstring": "Failed to resolve API Key variable request.queryparam.apikey",
4          "detail": {
5              "errorcode": "steps.oauth.v2.FailedToResolveAPIKey"
6          }
7      }
8  }

```

- f. Switch back to the browser tab where you have the trace tool running. Click on the request that just came through and you will see that the Verify API Key just clicked in and blocks access to the API as no Consumer Key is specified as part of the request.

Dashboard / API Proxies / PB_Open_Weather Organization devjam_admin ▾

PB_Open_Weather

Deployment to Trace Environment test, Revision 1 ▾ Stop Trace Session Remaining Time: 07:21 Download Trace Session Node.js Logs ACCESS THE CLASSIC VERSION OF TRACE

Send Requests Method URL Status

Method: GET URL: /v1/pb_open_weather? q=Los%20Angeles%20US

Transactions Status Method URI Elapsed

| | Status | Method | URI | Elapsed |
|---|--------|--------|--------------------------------------|---------|
| 1 | 401 | GET | /v1/pb_open_wea... q=Los%20Angele... | 5 ms |

Transaction Map

Phase Details

| Phase | Description | Details |
|------------------|--------------------------|-------------------------|
| Spike Arrest 1 | GET /v1/pb_open_weather? | q=Los%20Angeles%2C%20US |
| Verify API Key 1 | GET /v1/pb_open_weather? | q=Los%20Angeles%2C%20US |

Variables

| Variable | Value | Value |
|--------------------|-----------------|-----------------|
| apiproxy.name | PB_Open_Weather | PB_Open_Weather |
| apiproxy.revision | 1 | 1 |
| environment.name | test | test |
| flow.resource.name | | |

Output from all Transactions
© 2014 Apigee Corp. All rights reserved. Version 141015

- g. Switch back to the browser tab with the POSTMAN client and this time add the “apikey” parameter (in the request) with the value of the app Consumer Key (the one you had saved in an earlier step in this lab) and you will see that the request returns successfully this time.



q Los Angeles, US

apikey Yz71TmsBwZlk0qzdnS04N7wkm8czB

URL Parameter Key Value

Body Headers (9) STATUS 200 OK TIME 363 ms

Pretty Raw Preview

```
1 {  
2   "coord": {  
3     "lon": -118.24,  
4     "lat": 34.05  
5   },  
6   "sys": {  
7     "type": 1,  
8     "id": 396,  
9     "message": 0.1112,  
10    "country": "US",  
11    "sunrise": 1413640868,  
12    "sunset": 1413681290  
13  },  
14  "weather": [  
15    {  
16      "id": 800,  
17      "main": "Clear",  
18      "description": "sky is clear",  
19      "icon": "01d"  
20    }  
21  ],  
22  "base": "cmc stations",  
23  "main": {  
24    "temp": 297.02,  
25    "pressure": 1011,  
26    "humidity": 53,  
27    "temp_min": 294.15,  
28    "temp_max": 300.15  
29  },  
30}
```

- h. Switch back to the browser tab which has the trace tool open and click on the specific request that came in.



Dashboard / API Proxies / PB_Open_Weather

Organization devjam_admin ▾

PB_Open_Weather

Deployment to Trace Environment test, Revision 1 ▾ Stop Trace Session Remaining Time: 04:03 Download Trace Session Node.js Logs ACCESS THE CLASSIC VERSION OF TRACE

Transactions

| Method | URL | Status | Elapsed |
|--------|--------------------------------------|--------|---------|
| GET | /v1/pb_open_wea... q=Los%20Angele... | 200 | 175 ms |
| GET | /v1/pb_open_wea... q=Los%20Angele... | 401 | 5 ms |

Send Requests

Method URL Status

Transaction Map

Phase Details

- Request Received from Client: GET /v1/pb_open_weather?q=Los%20Angeles%2C%20US &apikey=Yz71TmsBwZik0q2dnS04N7wkm8czB9lp
- Response Sent to Client: 200 OK

Request Headers

| | |
|-----------------|--------------------|
| Accept | * |
| Accept-Encoding | gzip,deflate, sdch |
| Accept-Language | en-US,en;q=0.8 |

Response Headers

| | |
|----------------------------------|------|
| Access-Control-Allow-Credentials | true |
| Access-Control-Allow-Methods | GET |
| Access-Control-Allow-Origin | * |

View Options Transaction Map

- Show Disabled Policies (none)
- Show Skipped Phases (2)
- Show All FlowInfo's (4)
- Automatically Compare Selected Phase
- Show Variables
- Show Properties

© 2014 Apigee Corp. All rights reserved. Version 141015

8. Now lets test the “Quota” Policy. Remember you had set a quota of 5 requests every 1 minute. So this time we will hit the proxy URL (along with the API Key in the query param) a few times, but before you do that you have do couple more steps.
 - a. Remember that you have a Spike Arrest Policy in place which can kick in if you keep hitting the proxy too fast. At this point stop the trace session, go the proxy editor and change the Spike Arrest rate to 100ps so it doesn't kick in during our testing of the quota.

PB_Open_Weather

OVERVIEW DEVELOP Help for Selected Spike Arrest Policy

Project ▾ Save Revision 2 ▾ New ▾ New Policy ▾ Attach Policy Tools ▾ Deployment ▾

Navigator

- Policies
 - Extract Variables 1
 - Quota 1
 - Response Cache 1
 - Spike Arrest 1
 - Statistics Collector 1
 - Verify API Key 1
- Proxy Endpoints
 - default
 - All PreFlow
 - All PostFlow
- Target Endpoints
 - default
 - All PreFlow
 - All PostFlow
- Scripts

Map: Endpoint default, Flow PreFlow

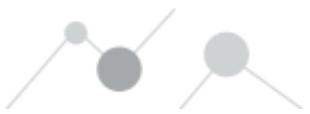
Code: Spike-Arrest-1

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <SpikeArrest async="false" continueOnError="false" enabled="true" name="Spike Arrest 1">
3   <DisplayName>Spike Arrest 1</DisplayName>
4   <Rate>100ps</Rate>
5 </SpikeArrest>
    
```

Deployed to Environment: test

© 2014 Apigee Corp. All rights reserved. Version 141015



- b. After making the change and saving it, go back to the Trace window and start another trace session
- c. Now go back to POSTMAN and hit the proxy URL along with the API key in the query param about 6 times, you should see that after the 5th time the quota policy kicks in.

The screenshot shows the POSTMAN interface with the following details:

- URL: `q` (query parameter) and `Los Angeles, US` (location).
- Headers: `apikey` set to `Yz71TmsBwZik0qzdnS04N7wkm8czB`.
- Body: URL Parameter Key and Value fields are empty.
- Action buttons: `Send` (highlighted in blue), `Preview`, and `Add to collection`.
- Status bar: `Body`, `Headers (3)`, `STATUS 500 Internal Server Error`, `TIME 5307 ms`.
- Response tab: `Pretty` (selected), `Raw`, `Preview`, `JSON`, `XML`. The response body is displayed in JSON format:

```
1 {
2   "fault": {
3     "faultstring": "Rate limit quota violation. Quota limit exceeded. Identifier : Yz71TmsBwZik0qzdnS04N7wkm8czB9Ip",
4     "detail": {
5       "errorcode": "policies.ratelimit.QuotaViolation"
6     }
7   }
8 }
```

- d. Now go back to the Trace tool and see the trace log to see the Quota policy returning an error.



PB_Open_Weather

Deployment to Trace Environment test, Revision 2 ▾ Stop Trace Session Remaining Time: 08:55 Download Trace Session ▾ Node.js Logs ACCESS THE CLASSIC VERSION OF TRACE

Transactions

| | Status | Method | URI | Elapsed |
|---|--------|--------|---|---------|
| 6 | 500 | GET | /v1/pb_open_wea... q=Los%20Angele... | 5033 ms |
| 5 | 200 | GET | /v1/pb_open_wea... q=Los%20Angele... | 33 ms |
| 4 | 200 | GET | /v1/pb_open_wea... q=Los%20Angele... | 36 ms |
| 3 | 200 | GET | /v1/pb_open_wea... q=Los%20Angele... | 35 ms |
| 2 | 200 | GET | /v1/pb_open_wea... q=Los%20Angele... | 50 ms |
| 1 | 200 | GET | /v1/pb_open_wea... q=Los%20Angele... | 71 ms |

Send Requests

Method URL Status

GET http://devjam_admin-test.apigee.net/v1/pb_open_weather Send Or Send with the API Console

Transaction Map

Phase Details

Verify API Key 1
GET /v1/pb_open_weather?q=Los%20Angeles%2C%20US
&apikey=Yz71TmsBwZik0qzdnS04N7wkm8czB9lp

View Options

- Show Transaction Map
- Show Disabled Policies (none)
- Show Skipped Phases (2)
- Show All Flowinfos (1)
- Phase Details
- Automatically Compare Selected Phase
- Show Variables
- Show Properties

Variables

| | |
|---------------------------|------------------------------------|
| apigee.apiproduct.name | = PB_Open_Weather_Bronze |
| apigee.client_id | = Yz71TmsBwZik0qzdnS04N7wkm8czB9lp |
| apigee.developer.app.name | = PBOpenWeatherApp |
| apigee.developer.email | = pbhogill+devjam@apigee.com |

Output from all Transactions © 2014 Apigee Corp. All rights reserved. Version 141015

Summary

In this lab you created an API Product, added a Developer, created a Developer App, and added Quota and Verify API Key policies to the API Proxy. You also tested that the API proxy now only allows access to the target end-point when a valid Consumer Key is sent as part of the request and also only the number of times that is allowed by the Quota policy. Please visit the [documentation](#) to see the different ways of securing your API proxies using API Key verification, OAuth etc.