

Rate Limiting for API Proxy (Quota Policy):

To configure the number of request messages that an API proxy allows over a period of time, such as a second, minute, hour, day, week, or month. You can set the quota to be the same for all apps accessing the API proxy, or you can set the quota based on:

- The product that contains the API proxy
- The app requesting the API
- The app developer
- Many other criteria

Using Quota policies you can, for example, limit apps to 1 request per minute, or to 10,000 requests per month. For example, if you have three API proxies in an API product, a single quota is not shared across all three even if all three use the same quota policy configuration.

Step 1:

Initially, create an API Proxy by providing Backend Service URL and Project Base path.

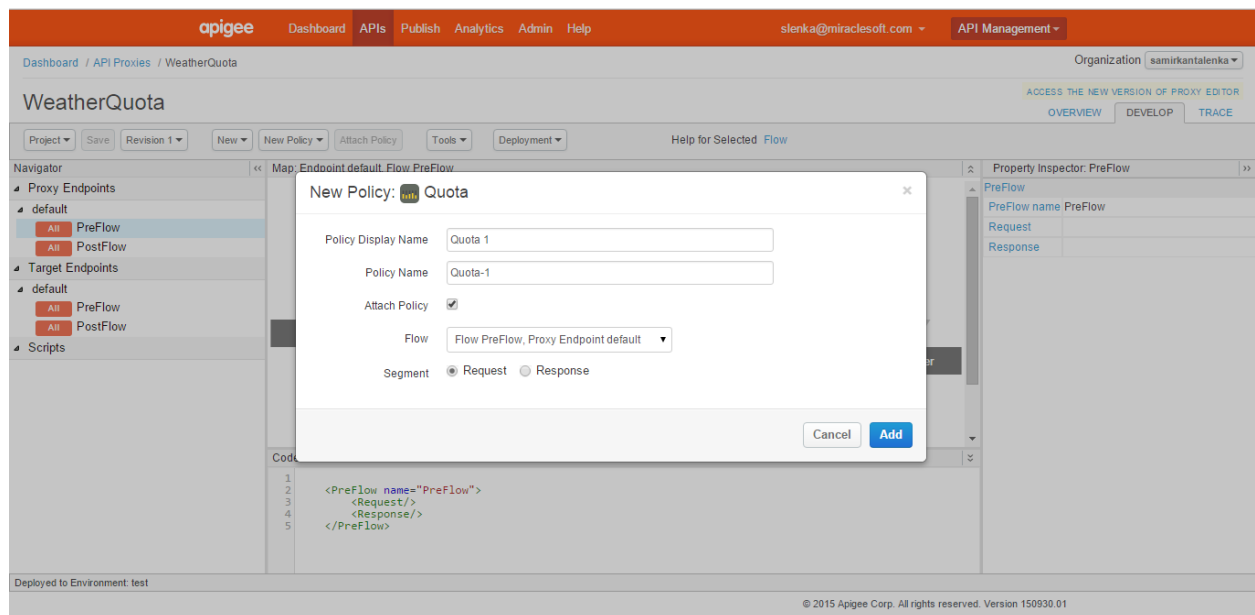
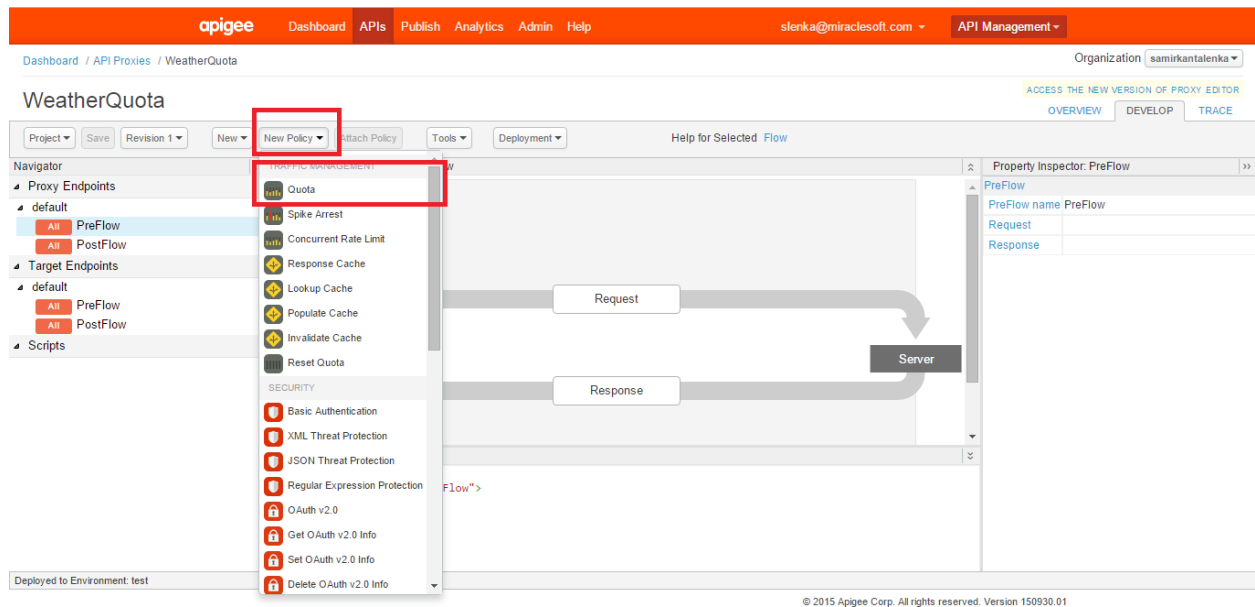
The screenshot shows the 'New API Proxy' dialog box with three main sections:

- 1 Choose Your Starting Point**
 - Starting Point Type: ☒ Backend Service, ☐ API Bundle, ☐ WSDL, ☐ No Target, ☐ New Node.js, ☐ Existing Node.js
 - Backend Service URL:
Defines the target URL invoked on behalf of this API proxy. Any URL that is accessible over the open Internet can be used. Example: https://weather.yahooapis.com
- 2 Identify Your API Proxy**
 - Name:
Valid characters are letters, numbers, dash (-), and underscore (_).
 - Project Base Path:
A path component that uniquely identifies this API proxy. The public-facing URL of this API proxy is comprised of your organization name, an environment where this API proxy is deployed, and this Base Path. Example URL: http://samirkantalekha-test.apigee.net/v1/weatherquota
 - Description:
- 3 Add Features**
 - Security: ☒ None, ☐ Secure with API Keys, ☐ Secure with OAuth v2.0 Access Tokens, ☐ Impose Quota per Developer, ☐ Publish API Product
 - Browser Access: ☐ Enable Direct Browser Access for Your API — Allow direct requests from a browser via CORS.

Buttons: Cancel, Build

Step 2:

In order to give restriction for the user based on the hits, select **Quota Policy** in **New Policy** tab



Step 3:

Here select the *Quota Policy* and edit the code and specify the *Time limit* based on our requirement.

The screenshot shows the Apigee API Management console for the 'WeatherQuota' API. The 'Quota Policy' is selected, and the 'Code: Quota-1' section displays the following XML code:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Quota async="false" continueOnError="false" enabled="true" name="Quota-1" type="calendar">
  <DisplayName>Quota 1</DisplayName>
  <Properties/>
  <Allow count="20" countRef="request.header.allowed_quota"/>
  <Interval ref="request.header.quota_count">1</Interval>
  <Distributed>false</Distributed>
  <Synchronous>false</Synchronous>
  <TimeUnit ref="request.header.quota_timeout">month</TimeUnit>
  <StartTime>2015-12-15 12:00:00</StartTime>
  <AsynchronousConfiguration>
    <SyncIntervalInSeconds>20</SyncIntervalInSeconds>
    <SyncMessageCount>5</SyncMessageCount>
  </AsynchronousConfiguration>
</Quota>
```

The 'Property Inspector: Quota-1' section shows the following configuration details:

Property	Value
Quota async	false
Quota continueOnError	false
Quota enabled	true
Quota name	Quota-1
Quota type	calendar
DisplayName	Quota 1
Allow count	20
Allow countRef	request.header.allowed...
Interval ref	request.header.quota_c...
Interval	1
Distributed	false
Synchronous	false
TimeUnit ref	request.header.quota_l...
TimeUnit	month
StartTime	2015-12-15 12:00:00

Step 4:

Change the *Interval* and *Allow Count* based on our requirement.

In this example, it is changed to 20 requests per minute.

The screenshot shows the Apigee API Management console for the 'WeatherQuota' API. The 'Quota Policy' is selected, and the 'Code: Quota-1' section displays the following XML code:

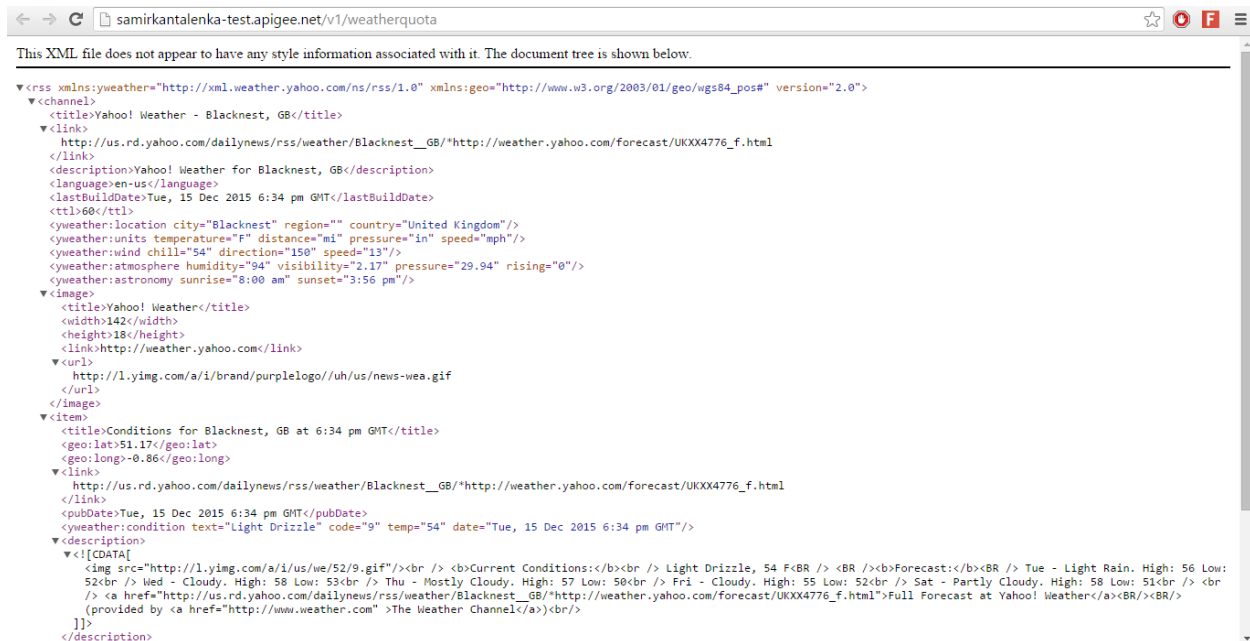
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Quota async="false" continueOnError="false" enabled="true" name="Quota-1" type="calendar">
  <DisplayName>Quota 1</DisplayName>
  <Properties/>
  <Allow count="2000" countRef="request.header.allowed_quota"/>
  <Interval ref="request.header.quota_count">1</Interval>
  <Distributed>false</Distributed>
  <Synchronous>false</Synchronous>
  <TimeUnit ref="request.header.quota_timeout">month</TimeUnit>
  <StartTime>2015-12-15 12:00:00</StartTime>
  <AsynchronousConfiguration>
    <SyncIntervalInSeconds>20</SyncIntervalInSeconds>
    <SyncMessageCount>5</SyncMessageCount>
  </AsynchronousConfiguration>
</Quota>
```

The 'Property Inspector: Quota-1' section shows the following configuration details:

Property	Value
Quota async	false
Quota continueOnError	false
Quota enabled	true
Quota name	Quota-1
Quota type	calendar
DisplayName	Quota 1
Allow count	2000
Allow countRef	request.header.allowed...
Interval ref	request.header.quota_c...
Interval	1
Distributed	false
Synchronous	false
TimeUnit ref	request.header.quota_l...
TimeUnit	month
StartTime	2015-12-15 12:00:00

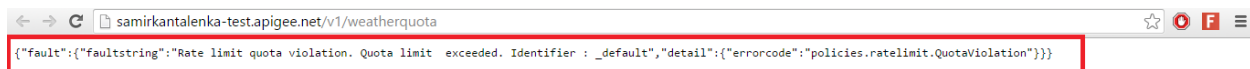
Step 5:

In overview tab hit the backened URL for viewing the response.



Step 6:

After hitting the URL for 20 times, it will show an error as follows



After completion of 1minute, the data will be shown.

