API Publishing : Packaging APIs

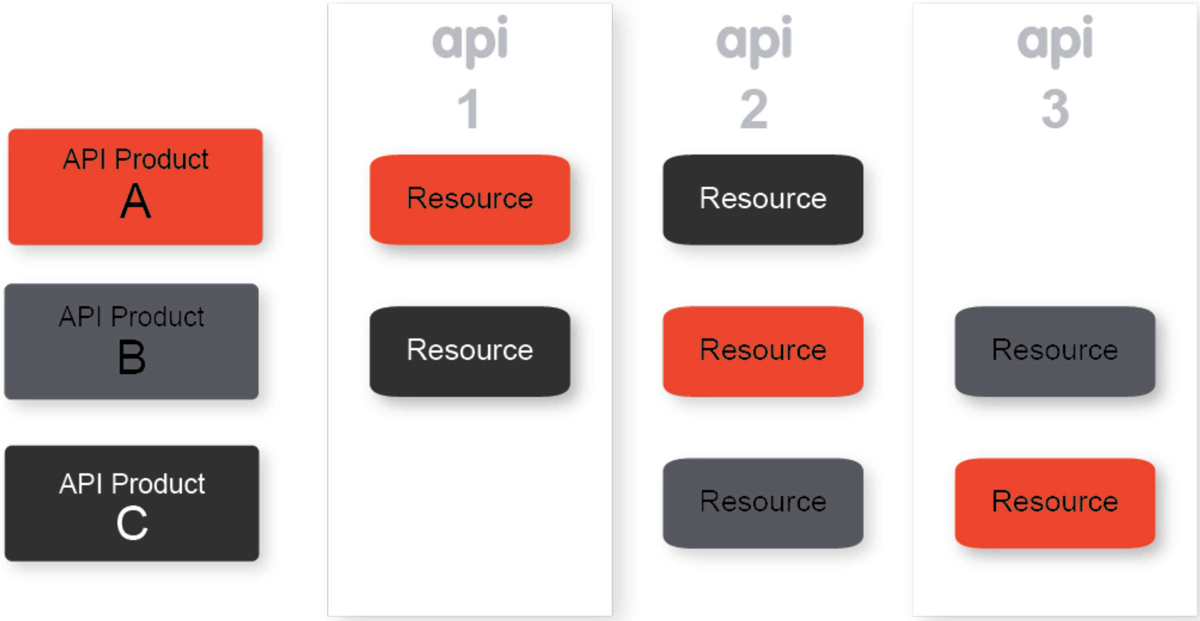
Duration : 20 mins
Persona : API Product Manager

Use case

You have a set of APIs that are to be consumed by trusted partners. You want to package the APIs into a meaningful context for the creation of a developer app, and you want to make that package available for a developer to subscribe to, in order to provide access to an API Key.

How can Apigee Edge help?

Apigee has the concept of 'API Products'. API products enable you to bundle and distribute your APIs to multiple developer groups simultaneously, without having to modify code. An API product consists of a list of API resources (URIs) combined with a Service Plan (rate-limiting policy settings) plus any custom metadata required by the API provider. API products provide the basis for access control in Apigee, since they provide control over the set of API resources that apps are allowed to consume.



As part of the app provisioning workflow, the developer selects from a list of API products. This selection of an API product is usually made from within the context of a developer portal, however for the purposes of this lab we will show you how to create a developer App and get access to an API Key via the Edge UI. The developer app is provisioned with a key and secret (generated by and stored on Apigee Edge) that enables the app to access the URIs bundled in the selected API product. To access API resources bundled in an API product, the app must present the API key issued by Apigee Edge. Apigee Edge will resolve the key that is presented against an API product, and then check associated API resources and quota settings.

The API supports multiple API products per app key - your developers can consume multiple API products without requiring multiple keys. Also, a key can be 'promoted' from one API product to another. This enables you to promote developers from 'free' to 'premium' API products seamlessly and without user interruption.

The following defines some of the terms used to register apps and generate keys:

API product A bundle of API proxies combined with a service plan that sets limits on access to those APIs. API products are the central mechanism that Apigee Edge uses for authorization and access control to your APIs. For more information, see API Products

Developer The API consumer. Developers write apps the make requests to your APIs.

App A client-side app that a developer registers to access an API product. Registering the app with the API product generates the API key for accessing the APIs in that product.

API key A string with authorization information that a client-side app uses to access the resources exposed by the API product. The API key is generated when a registered app is associated with an API product.

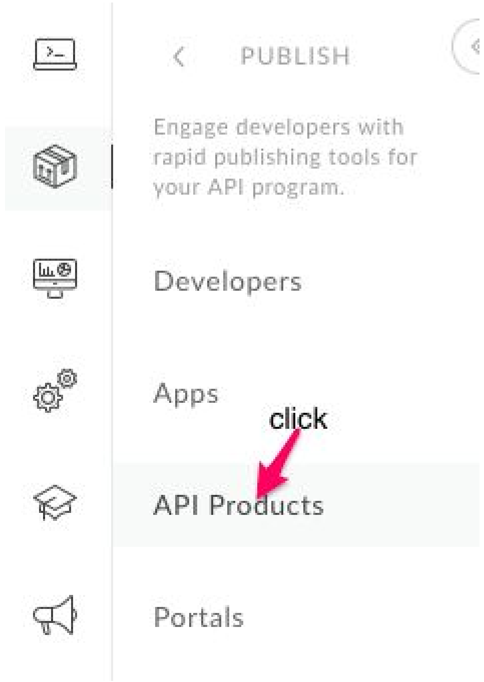
Pre-requisites

- You have an API proxy created in Apigee Edge. If not, jump back to "*API Development - Creating an API proxy*" lab.
- The API Proxy contains an API Key Verification policy. Follow the "*API Security - Secure with API Keys*" lab to enable this.

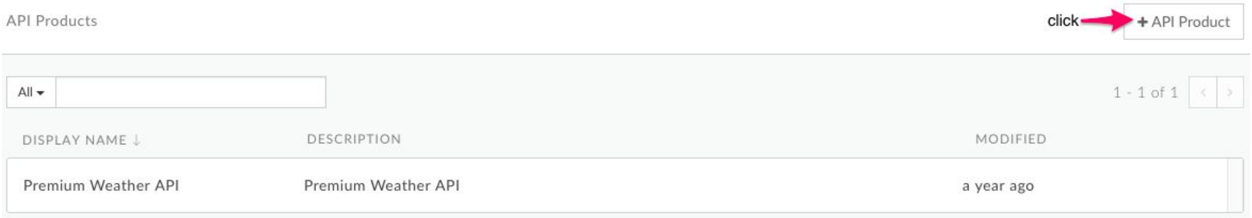
Instructions

Part 1 - Create API Product

- From the Apigee Edge Management UI, Using sidebar Navigation, go to **Publish → API Products**

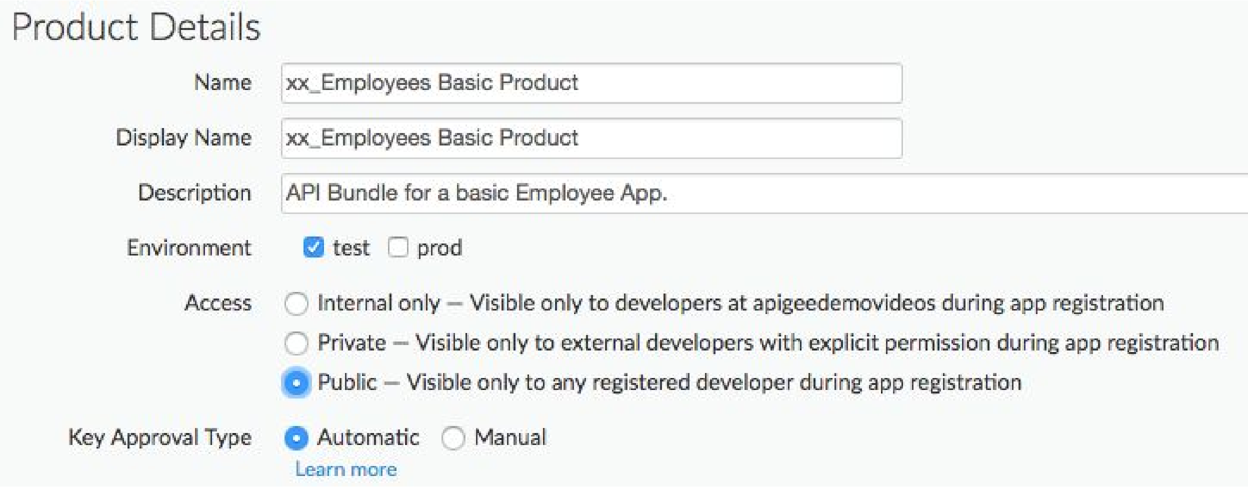

PUBLISH
Engage developers with rapid publishing tools for your API program.
Developers
Apps
click
API Products
Portals

- Click on ‘**+ API Product**’ button to add a new product



DISPLAY NAME ↓	DESCRIPTION	MODIFIED
Premium Weather API	Premium Weather API	a year ago

- In the ‘Product Details’ section of the new product screen, enter or select the following values for the various fields:
  - Display Name: {Your_Initials}_Employees Basic Product
  - Description: API Bundle for a basic Employee App.
  - Environment: Test
  - Access: Public
  - Key Approval Type: Automatic



Product Details

Name: xx_Employees Basic Product

Display Name: xx_Employees Basic Product

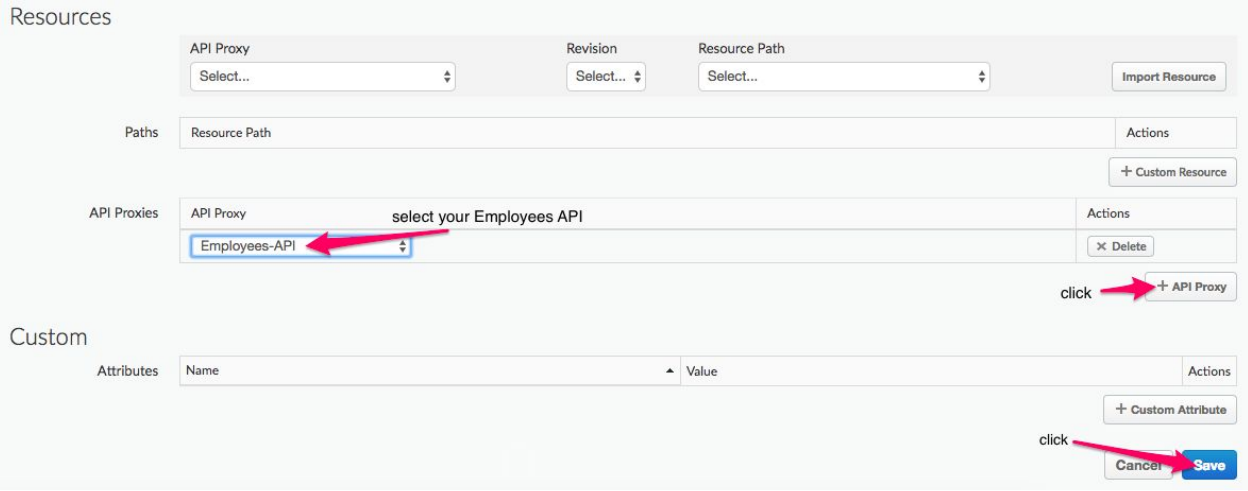
Description: API Bundle for a basic Employee App.

Environment: ☑ test □ prod

Access:
○ Internal only — Visible only to developers at apigeedemovideos during app registration
○ Private — Visible only to external developers with explicit permission during app registration
● Public — Visible only to any registered developer during app registration

Key Approval Type: ● Automatic ○ Manual

- In the ‘Resources’ section click the **‘+ API Proxy’** button, and in the dropdown select your ‘{your_initials} Employees API Proxy, and click ‘Save’ to save the API Product.



Resources

API Proxy	Revision	Resource Path	
Select...	Select...	Select...	Import Resource

Paths

Resource Path	Actions

API Proxies

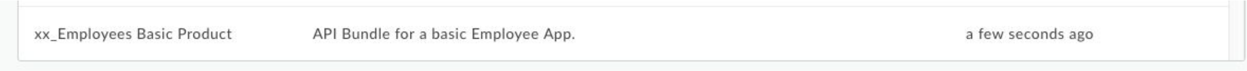
API Proxy	Actions
Employees-API	✕ Delete

Custom

Attributes

Name	Value	Actions

- The new product should now be listed on the ‘Products’ page.

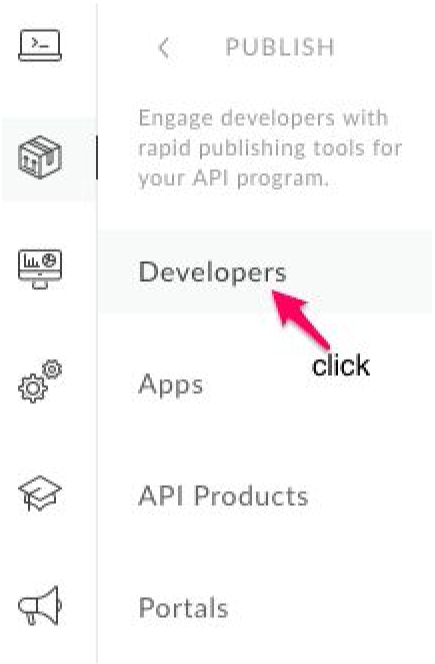


xx_Employees Basic Product	API Bundle for a basic Employee App.	a few seconds ago

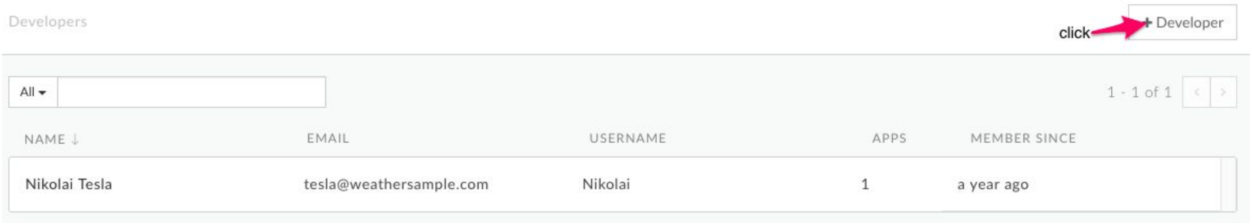
Part 2 - Create a Developer and a Developer App

In order to get access to an API key you must register a developer App with the API Product, and before you can register a developer App you must create a developer. Whilst this is typically one via the developer portal in a self service paradigm, for this lab we will do this via the Edge UI.

- From the Apigee Edge Management UI, Using sidebar Navigation, go to **Publish → Developers**

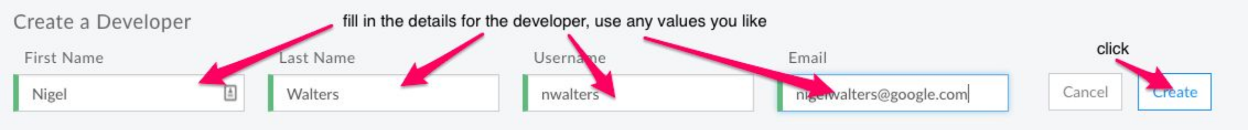

PUBLISH
Engage developers with rapid publishing tools for your API program.
Developers
click
Apps
API Products
Portals

- Click the '**+ Developer**' button to add a new developer



NAME ↓	EMAIL	USERNAME	APPS	MEMBER SINCE
Nikolai Tesla	tesla@weathersample.com	Nikolai	1	a year ago

- Fill in the developer details and click create



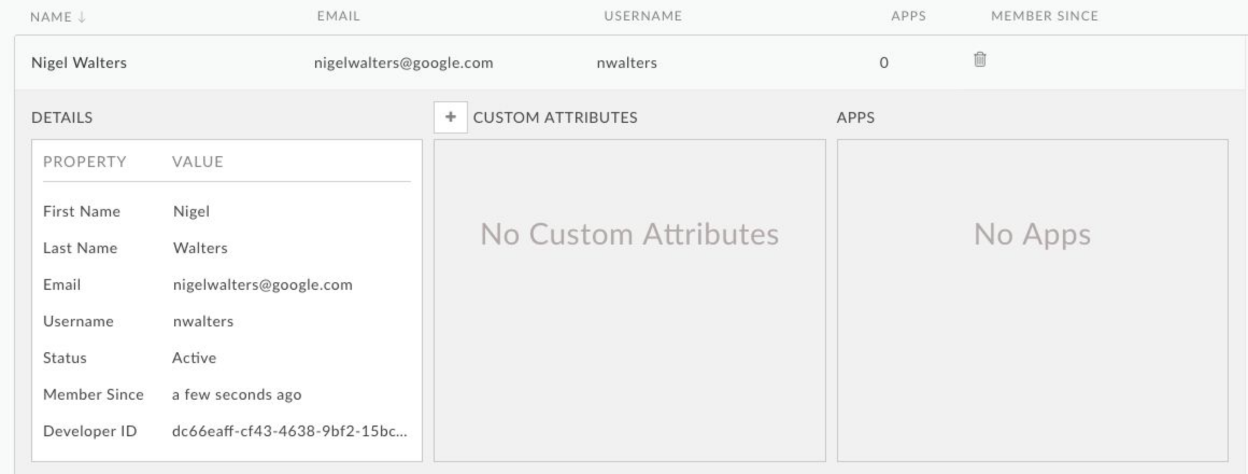
Create a Developer

fill in the details for the developer, use any values you like

First Name	Last Name	Username	Email		
Nigel	Walters	nwalters	nigelwalters@google.com	Cancel	Create

click

- The new developer should be listed on the ‘Developer’ page. Note that it shows the expanded view of the developer details.



NAME ↓	EMAIL	USERNAME	APPS	MEMBER SINCE
Nigel Walters	nigelwalters@google.com	nwalters	0	

DETAILS

PROPERTY	VALUE
First Name	Nigel
Last Name	Walters
Email	nigelwalters@google.com
Username	nwalters
Status	Active
Member Since	a few seconds ago
Developer ID	dc66eaff-cf43-4638-9bf2-15bc...

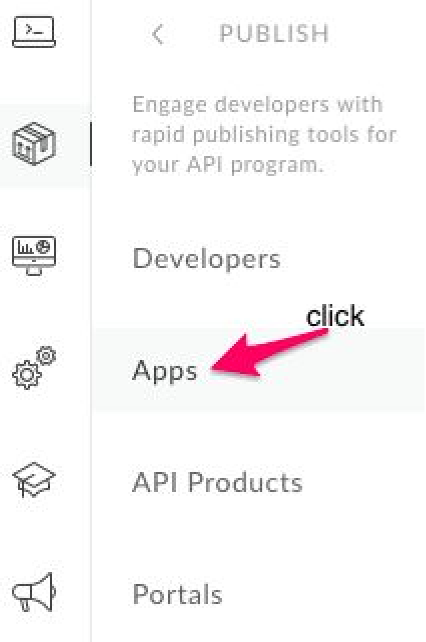
+ CUSTOM ATTRIBUTES

No Custom Attributes

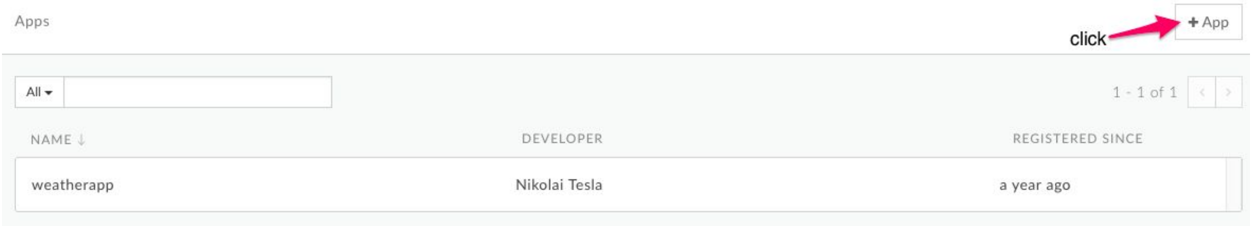
APPS

No Apps

- From the Apigee Edge Management UI, Using sidebar Navigation, go to **Publish** → **Apps**

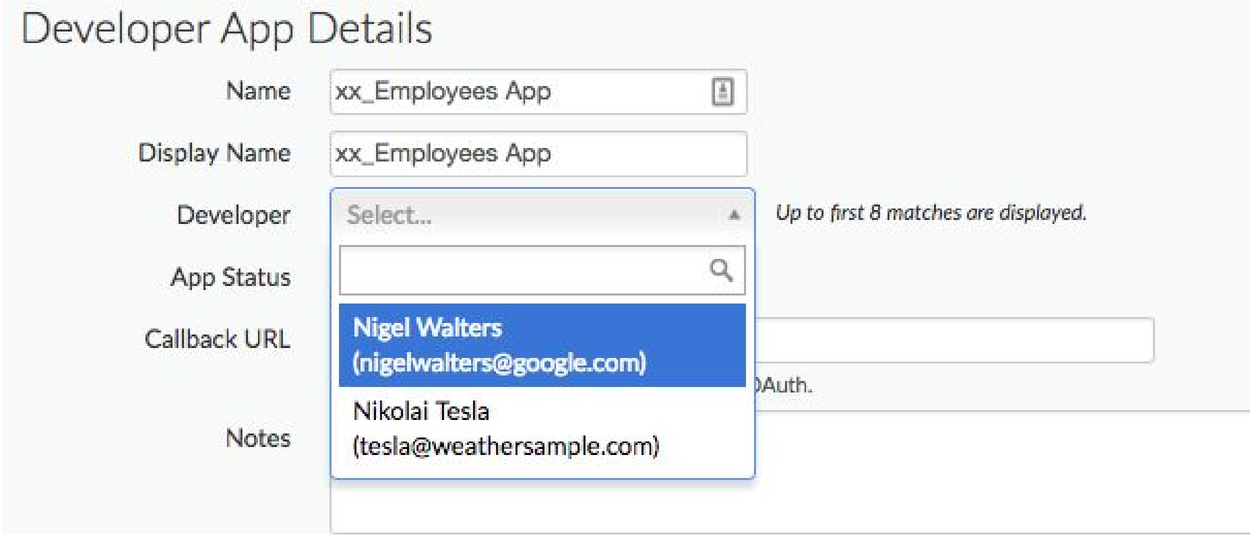

PUBLISH
Engage developers with rapid publishing tools for your API program.
Developers
click
Apps
API Products
Portals

- Click the ‘**+ Apps**’ button to add a new App



NAME ↓	DEVELOPER	REGISTERED SINCE
weatherapp	Nikolai Tesla	a year ago

- In the ‘Developer App Details’ section, specify {your_intitials)_Employees App as the App name, and select your developer from the developer drop down. **You may need to start typing your name in the text box to get it to appear.



Developer App Details

Name: xx_Employees App

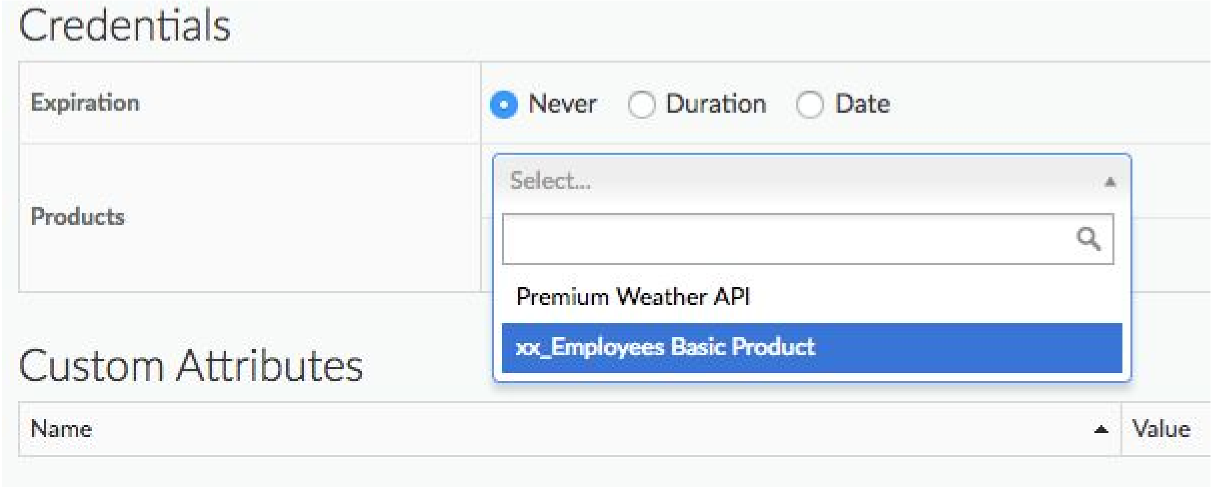
Display Name: xx_Employees App

Developer: Select...

Up to first 8 matches are displayed.

App Status
Callback URL
Notes
Nigel Walters (nigelwalters@google.com)
Nikolai Tesla (tesla@weathersample.com)
)Auth.

- In the credentials section, select ‘**+ API Product**’ and select your API Product

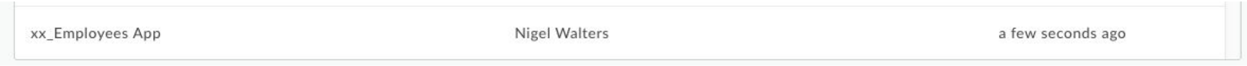


Credentials

Expiration: ● Never ○ Duration ○ Date

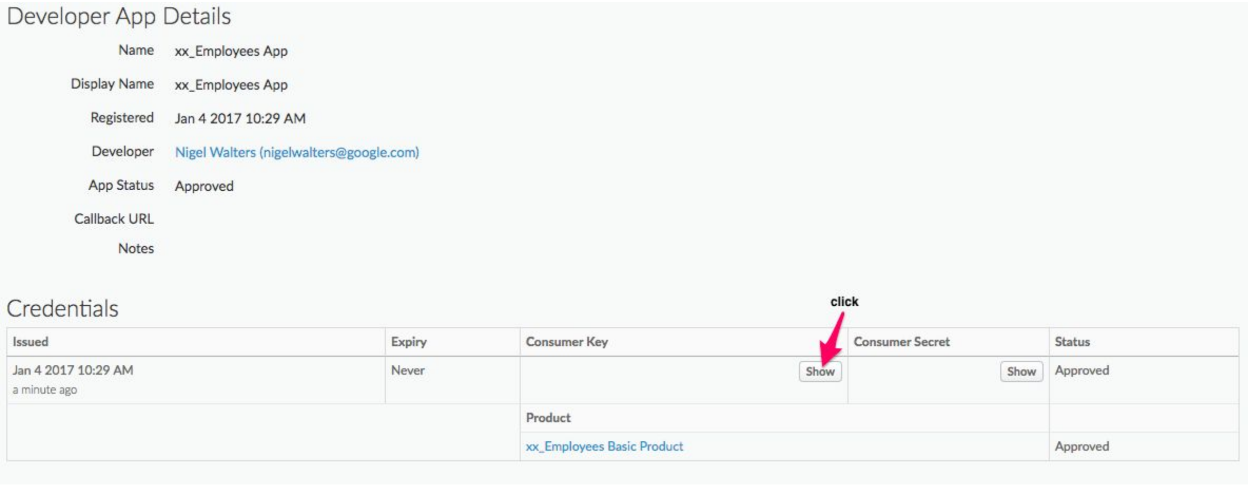
Products: Select...

- Hit the save button and verify that your new App appears in the ‘Apps’ page



xx_Employees App	Nigel Walters	a few seconds ago

- Click on your app to open it up, and then hit the show button next to 'Consumer Key' to reveal your API Key. Copy this value as we will need it to test the API.



Developer App Details

Name	xx_Employees App
Display Name	xx_Employees App
Registered	Jan 4 2017 10:29 AM
Developer	Nigel Walters (nigelwalters@google.com)
App Status	Approved
Callback URL	
Notes	

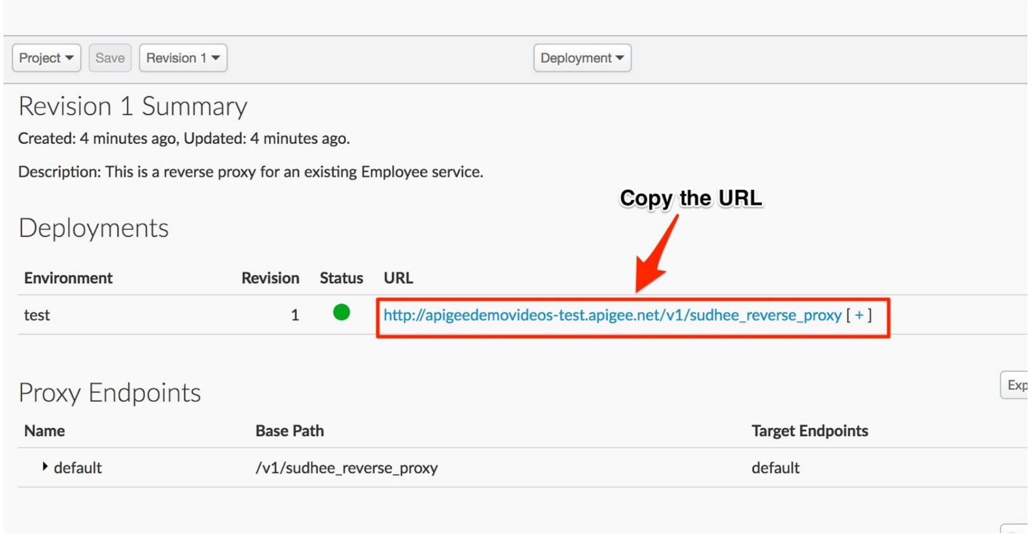
Credentials

click

Issued	Expiry	Consumer Key	Consumer Secret	Status
Jan 4 2017 10:29 AM a minute ago	Never	Show	Show	Approved
		Product		
		xx_Employees Basic Product		Approved

Part 3 - Test access to the API using the API Key

- Let us test the API proxy using the REST Client (https://apigee-rest-client.appspot.com/). Open the REST Client on a new browser window.
- Copy the URL for your API proxy.



Revision 1 Summary

Created: 4 minutes ago, Updated: 4 minutes ago.

Description: This is a reverse proxy for an existing Employee service.

Copy the URL

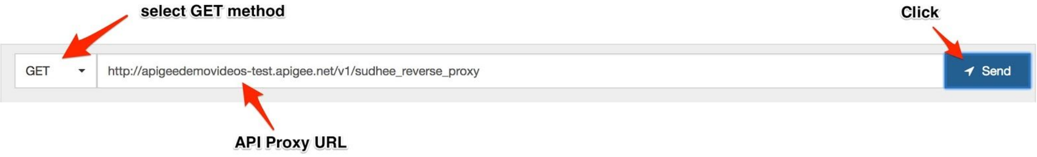
Deployments

Environment	Revision	Status	URL
test	1	●	http://apigeedemovideos-test.apigee.net/v1/sudhee_reverse_proxy [+]

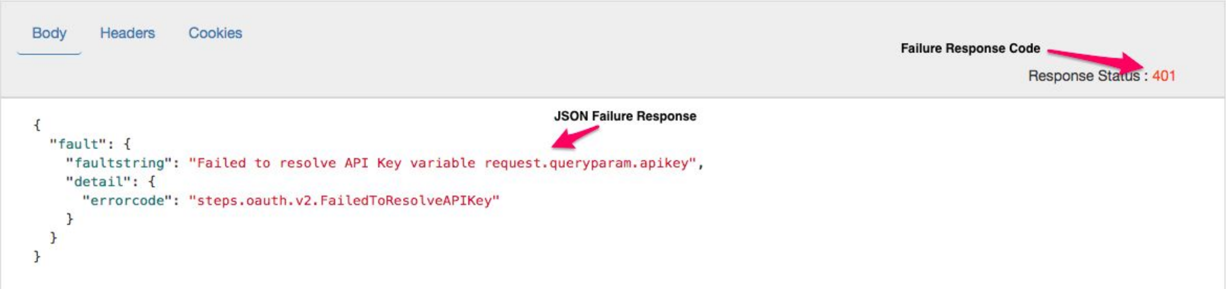
Proxy Endpoints

Name	Base Path	Target Endpoints
▸ default	/v1/sudhee_reverse_proxy	default

- Paste the link on the REST Client and make a GET call

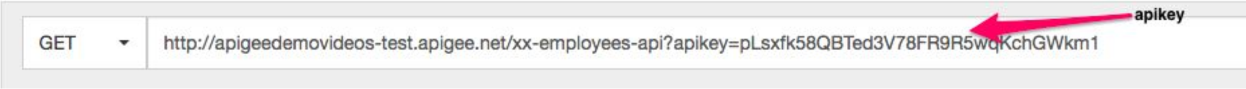

select GET method
Click
GET
http://apigeedemovideos-test.apigee.net/v1/sudhee_reverse_proxy
Send
API Proxy URL

- You should see a failure response similar to this, indicating that the API Key Verification has failed -

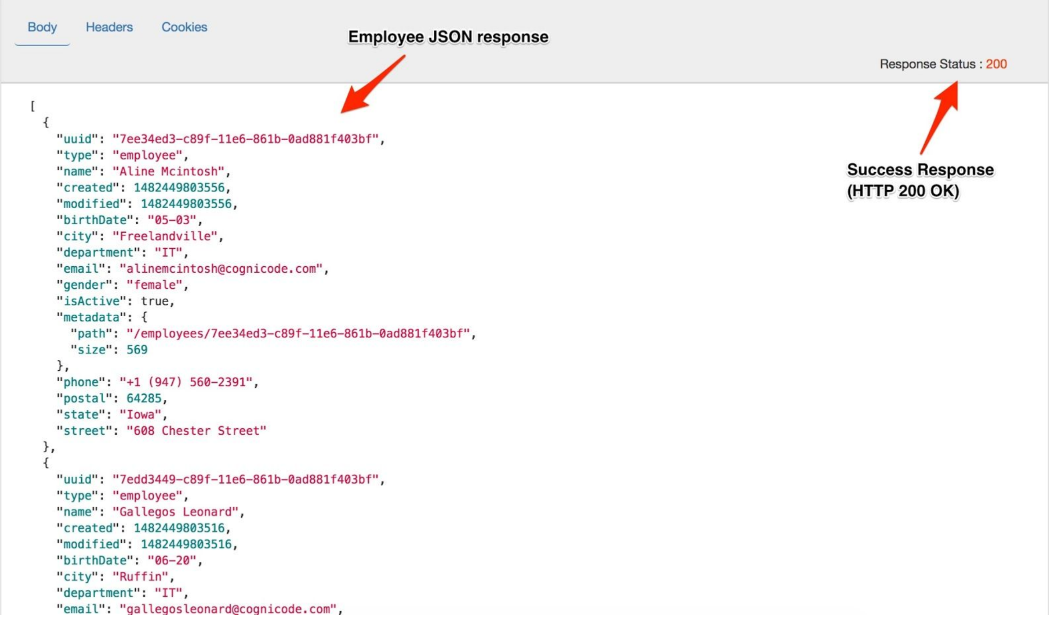


```
{
  "fault": {
    "faultstring": "Failed to resolve API Key variable request.queryparam.apikey",
    "detail": {
      "errorcode": "steps.oauth.v2.FailedToResolveAPIKey"
    }
  }
}
```

- Modify the Get call to include the API Key we copied from the previous section as a query parameter called ‘apikey’


GET
http://apigeedemovideos-test.apigee.net/xx-employees-api?apikey=pLsxfk58QBTed3V78FR9R5wqKchGWkm1
apikey

- You should see a success response similar to this -



```
[
  {
    "uuid": "7ee34ed3-c89f-11e6-861b-0ad881f403bf",
    "type": "employee",
    "name": "Aline Mcintosh",
    "created": 1482449803556,
    "modified": 1482449803556,
    "birthDate": "05-03",
    "city": "Freelandville",
    "department": "IT",
    "email": "alinemcintosh@cognicode.com",
    "gender": "female",
    "isActive": true,
    "metadata": {
      "path": "/employees/7ee34ed3-c89f-11e6-861b-0ad881f403bf",
      "size": 569
    },
    "phone": "+1 (947) 560-2391",
    "postal": 64285,
    "state": "Iowa",
    "street": "608 Chester Street"
  },
  {
    "uuid": "7edd3449-c89f-11e6-861b-0ad881f403bf",
    "type": "employee",
    "name": "Gallegos Leonard",
    "created": 1482449803516,
    "modified": 1482449803516,
    "birthDate": "06-20",
    "city": "Ruffin",
    "department": "IT",
    "email": "gallegosleonard@cognicode.com",
```

Lab Video

If you like to learn by watching, here is a short video on creating an API Product
http://www.youtube.com/watch?v=c1GWDey4DyM

Earn Extra-points

Now that you know how to create an API Product to include entire API Proxies, explore the finer grained control for packaging APIs. Modify your API Product to only allow access to specific revisions and / or resources within your API.

Quiz

1. What criteria should you use to choose which APIs should be added to any given API Product?
2. How would you use the Edge UI in order to revoke access to your APIs for a specific developer App?

Summary

That completes this hands-on lesson. In this simple lab you learned how to create an API Product, and how to use the Edge UI to create a developer App associated with that API and get access to an API Key.

References

- Useful Apigee documentation links on API Products -
  - API Product overview - http://docs.apigee.com/developer-services/content/what-api-product
  - How to create an API Product - http://docs.apigee.com/developer-services/content/creating-api-products
- Watch this 4minute video on “Anatomy of an API proxy” - https://www.youtube.com/watch?v=HWRSKZwCwkY - along with it’s community article http://bit.ly/2hWFSfZ

Rate this lab

How did you link this lab? Rate here.