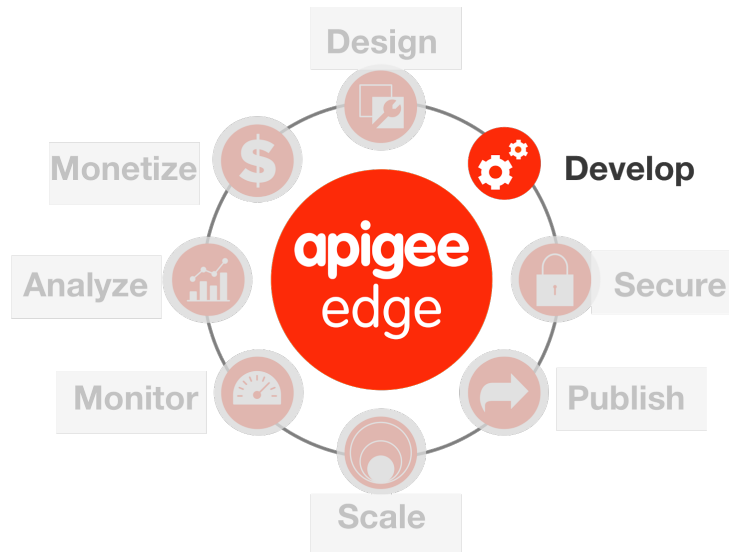




Lab 1 - Building a simple API Proxy



Overview

Apigee Edge enables you to expose APIs that can be easily consumed by developers who build apps. You expose APIs on Apigee Edge by building API proxies that act as managed 'facades' for backend services.

You expose APIs on Apigee Edge by implementing *API proxies*. An API proxy is a bundle of XML configuration files and code (such as JavaScript and Java) that implements the facade for your backend HTTP services. API proxies decouple the developer-facing API from your backend services, shielding developers from code changes and enabling you to innovate at the edge without impacting internal applications and development teams. As development teams make backend changes, developers continue to call the same API without any interruption.

API proxies manage request and response messages using a 'pipeline' processing model that defines 'Flows'. To customize the behavior of your API, you attach Policies to request and response Flows.

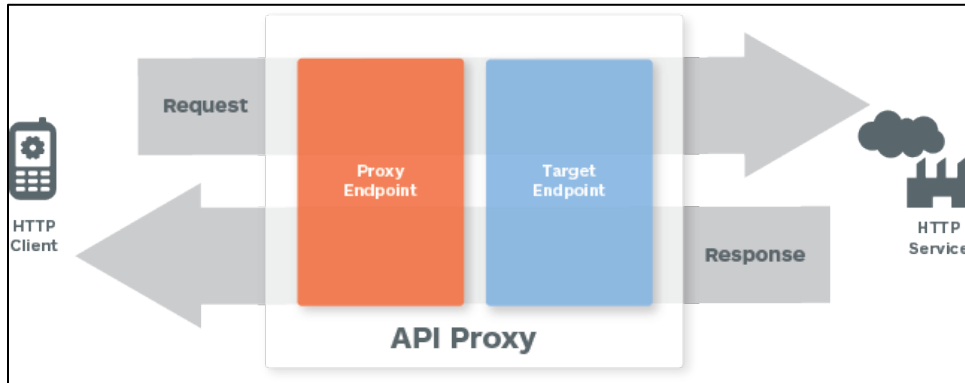
In an API proxy configuration, there are two types of endpoints:

- **ProxyEndpoint:** This configuration manages interactions with apps that consume your API. You configure the ProxyEndpoint to define the URL of your API. You usually attach Policies to the ProxyEndpoint to enforce security, quota checks, and other types of access control and rate-limiting.
- **TargetEndpoint:** This configuration manages interactions with your backend services on behalf of consumer apps. You configure the TargetEndpoint to forward request messages to the proper backend service. You usually attach Policies to the



TargetEndpoint to ensure that response messages are properly formatted for the app that made the initial request.

You can visualize API proxies as shown by the graphic below. A basic request and response exchange between an app (HTTP client) and a backend service is managed in an API proxy by a ProxyEndpoint and TargetEndpoint.



You can build API proxies using the Apigee Edge management UI. You can also implement API proxies on your local machine, and then import them to your organization on Apigee Edge. For an overview of the UI and API, see here - <http://apigee.com/docs/api-services/content/using-apigee-edge-development-environment>

Objectives

In this lab you will get familiar with the Apigee Edge Management UI by creating a simple passthrough facade to the GET API for the 'hotels' data collection in API BaaS. After creating the facade you will also learn how to deploy and undeploy a proxy. Finally you will also learn how to use the Trace capabilities in the Management UI.

Prerequisites

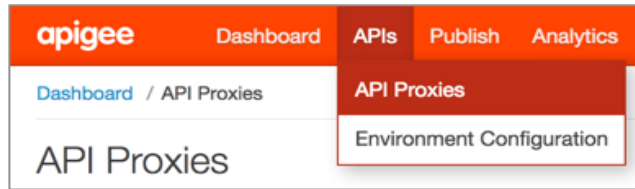
- You have registered with Apigee and have an account in Apigee Edge.

Estimated Time: 15 mins

- 1) **Creating an API Proxy** for a backend service that you want to expose requires you to provide the network address for the backend service, along with some information that API Services uses to model the API that will be exposed to developers.
 - a) Open up a browser tab and log in to <http://enterprise.apigee.com>
 - b) From the Organization drop-down in the top-right corner, select the organization assigned to you.
 - c) From the Environment drop-down, select 'test'



- d) From the main menu, select APIs → API Proxies



- e) To create a new API proxy, select the **+ API Proxy** button to add a new proxy.
f) On the New API Proxy form that is displayed select “Reverse proxy” and click Next.

Build a Proxy

☒ **Reverse proxy (most common)**
Route inbound requests to backend services.
 Optionally associate the proxy with an OpenAPI (Swagger) document

☐ **Node.js App**
Create a new app in JavaScript and optionally add policies.

☐ **SOAP service**
Create a RESTful or pass-through proxy for a SOAP service.

☐ **No Target**
Create a simple API proxy that does route to any backend target.

☐ **Proxy bundle**
Import an existing proxy from a zip archive.

- g) Provide the following information under Proxy Details section and click Next.
Proxy Name: **{your_initials}_hotels**
Proxy Base Path: **/v1/{your_initials}_hotels**
Existing API: <http://api.usergrid.com/apigeedemovideos/sandbox/hotels>
Description: **Facade to the BaaS hotels data collection**



Replace **{your_initials}** with your initials while configuring the API proxy. *Example - If you name is 'John Doe', your API proxy name would be 'jd_hotels'. We will follow this naming convention throughout the DevJam.*

- h) Select the Authorization as “Pass through (none)”.



Build a Proxy

TYPE
DETAILS
SECURITY

Secure access for users and clients.

Authorization ☒ Pass through (none)
☐ API Key
☐ OAuth 2.0

Browser ☐ Add CORS headers

- i) Select both “default” and “secure” virtual hosts and click Next.

Build a Proxy

TYPE
DETAILS
SECURITY
VIRTUAL HOSTS

Select the virtual hosts this proxy will bind to when it is deployed. You must select at least one v

<input checked="" type="checkbox"/> Name	Environment	Host Aliases
<input checked="" type="checkbox"/> default	prod	http://devjammingorg-prod.apigee.net
	test	http://devjammingorg-test.apigee.net
<input checked="" type="checkbox"/> secure	prod	https://devjammingorg-prod.apigee.net
	test	https://devjammingorg-test.apigee.net

- j) Select the Deployment Environment as “test”. Click on ‘Build and Deploy’ button to create and deploy the proxy to test environment.
- k) Once the proxy has been deployed, click on the view **{your_initials}_hotels**.
- l) Review the information on the resulting API Proxy page



sud_hotels What's new in the Proxy Editor ACC OVERVIEW

Project Save Revision 2 Deployment

Revision 2 Summary
Created: 30 minutes ago, Updated: 3 minutes ago.
No description for this proxy.

Deployments

Environment	Revision	Status	URL
test	2	●	http://devjammingorg-test.apigee.net/v1/sud_hotels [+]

Proxy Endpoints Expand All Collapse All

Name	Base Path	Target Endpoints
▶ default	/v1/sud_hotels	default

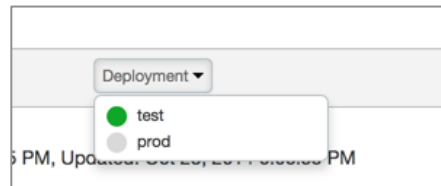
Target Endpoints Expand All Collapse All

Name	Target	Used by Proxy Endpoints
------	--------	-------------------------

m) That's it! With those few steps, you've deployed a simple passthrough API Proxy which acts as a facade to our BaaS 'hotels' data collection

2) **Deploying and Undeploying** a Proxy to a given environment from the Management UI is simple too.

a) Click on the 'Deploy' drop-down on the API Proxy page



b) Notice that the proxy we just created is deployed to the 'test' environment

c) Click on the 'test' environment name. This un-deploys the proxy from the 'test' environment

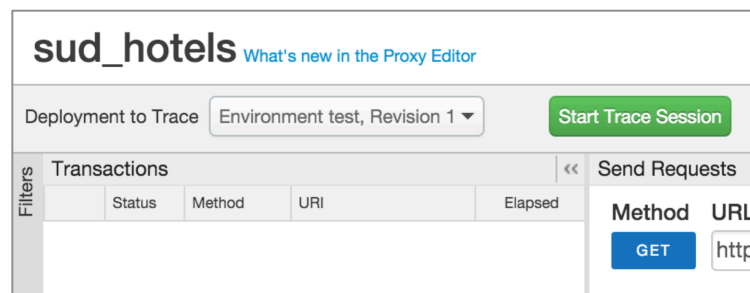
d) Click on the 'Deploy' drop-down again and select the 'test' environment. This re-deploys the proxy to the 'test' environment

3) **Tracing the execution** of a proxy can also be done easily from the Management UI. This is a very powerful capability of the product since it makes it easy to troubleshoot and monitor API proxies running on Apigee Edge. Trace lets you probe the details of each step through an API proxy flow.

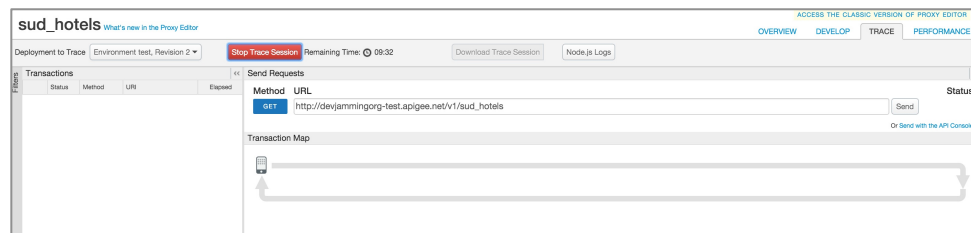
a) From the '{your_initials}_hotels' API Proxy page, click on the 'Trace' tab



- b) Once on the 'Trace' tab, pick the correct environment from the 'Deployment to Trace' drop-down. In your case there should only be one environment to pick - 'Environment test, Revision 1'



- c) Click on the 'Start Trace Session' button.
d) After the trace session starts, click on the 'Send' button in the 'Send Requests' section.



- e) You will see the a transaction appear in the 'Transactions' section and a visual representation of the executed proxy flow in the 'Transaction Map' section.
f) The 'Transaction Map' is interactive - click on the various execution steps within the flow and review the information provided in the 'Phase Details' section. You will notice that for each step information such as headers, variables, payload, properties and other relevant information is available. This helps you quickly understand exactly what is happening within the proxy flow.



Send Requests

Method

URL

GET

http://devjammingorg-test.apigee.net/sud_hotels

Send

Status

200

Or Send with the API Console

Transaction Map

AX

2ms

Back

Next

Phase Details

Request Received from Client

GET /sud_hotels

Response Sent to Client

200 OK

Request Headers

Response Headers

Summary

That completes this hands-on lesson. Simple and quick, but you learned the fundamentals of how to create an API Proxy, how to deploy and undeploy it to/from an environment, and how to trace the proxy flow. Throughout the rest of the hands-on labs you will be working with this Management UI to add policies to proxies, testing and tracing these proxies, and learning about the power of the Apigee Edge platform.

Devjam 2.0 - ©2015-2017 Apigee, All rights reserved.

7 of 7