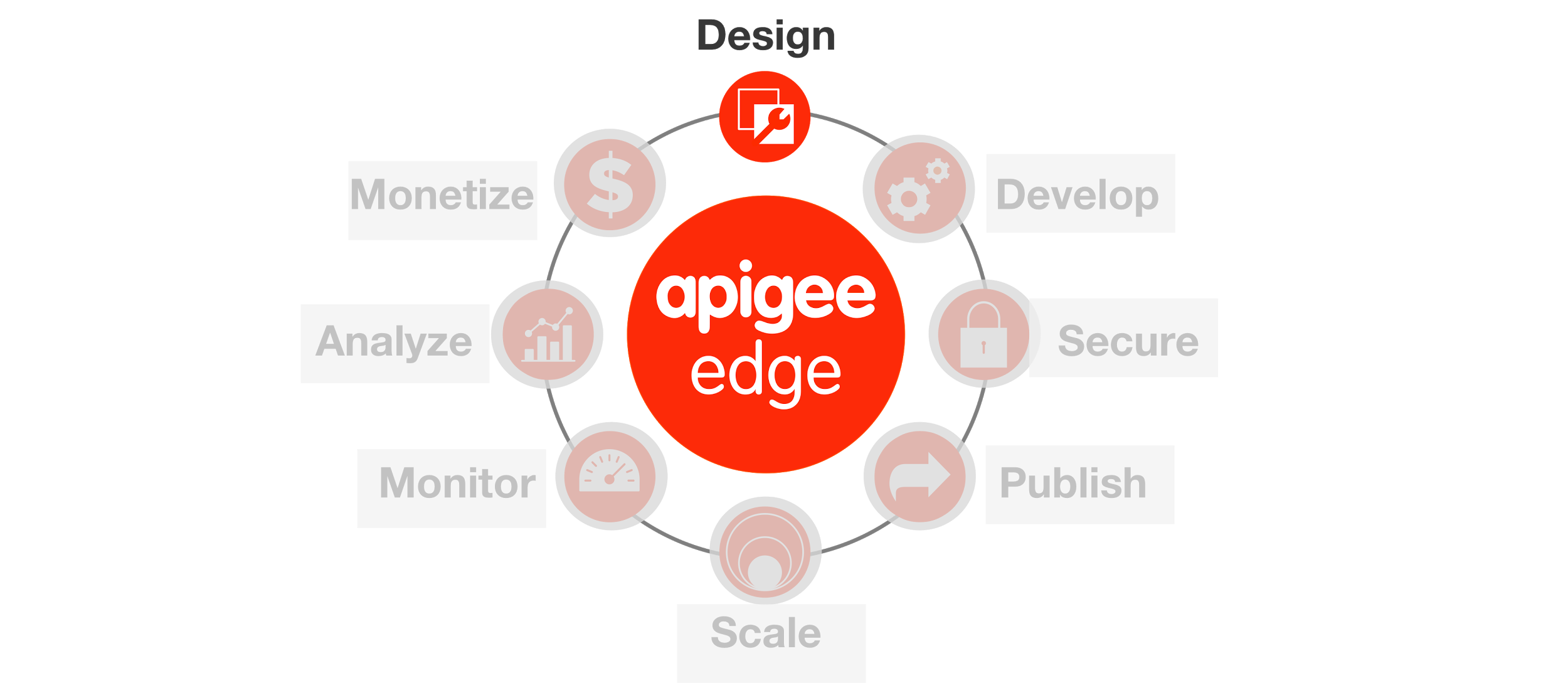
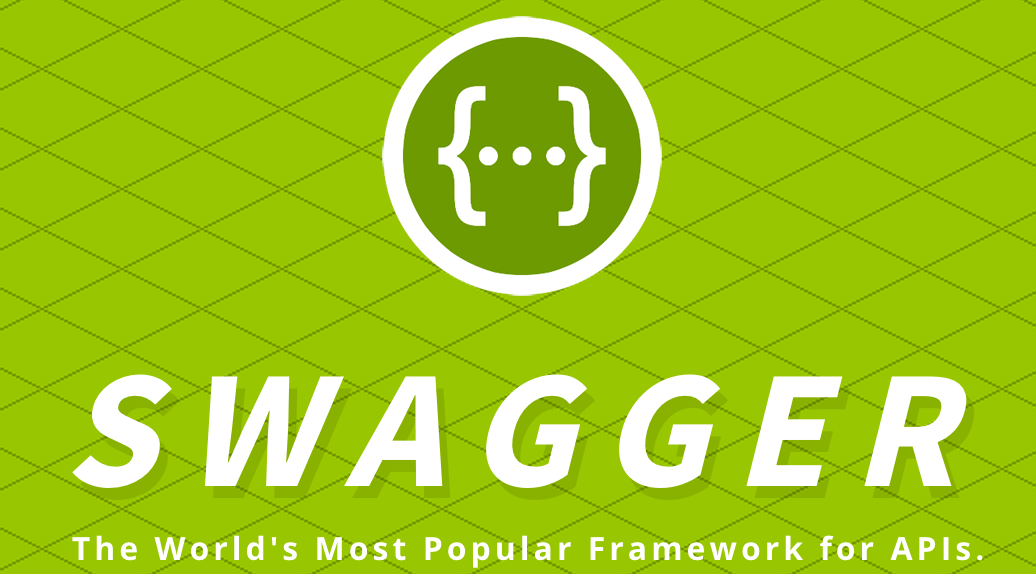


**Appendix 1 - Designing APIs using Swagger**

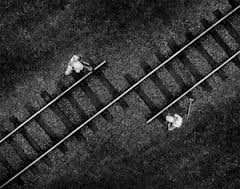
**Overview**



An API is an interface between the provider of some backend system(s) who wants to expose a set of services and the consumers of those services who want to do something with them. Adherence to a “contract” or a clear understanding of what the requests and responses should look like, makes life easier for all participants. In order to clearly communicate the terms of that contract, the industry has created a few formats for describing an API, the most popular of which is called [Swagger](http://swagger.io/).



By designing your API in Swagger, you allow the API developer and the API consumer to both do their jobs and meet successfully in the middle without unpleasant surprises.



Apigee is a key contributing member of and has partnered with a number of other companies to help drive the Swagger spec and contribute open source software to the community.

This is a partial list of the tools that Apigee has created, open-sourced, or contributed to related to API-first design, Swagger, or API deployment:

* [http://apistudio.io](http://apistudio.io/) - In-browser Swagger IDE including live documentation, code generation, mocking, and cloud hosting.
* <https://github.com/apigee-127/a127-documentation/wiki> - A toolkit for modeling and building rich, enterprise-class APIs in Node.js on your laptop.
* <http://editor.swagger.io> - This editor is the basis for the ones used in the above two projects, but it also includes code generators for a number of other languages.

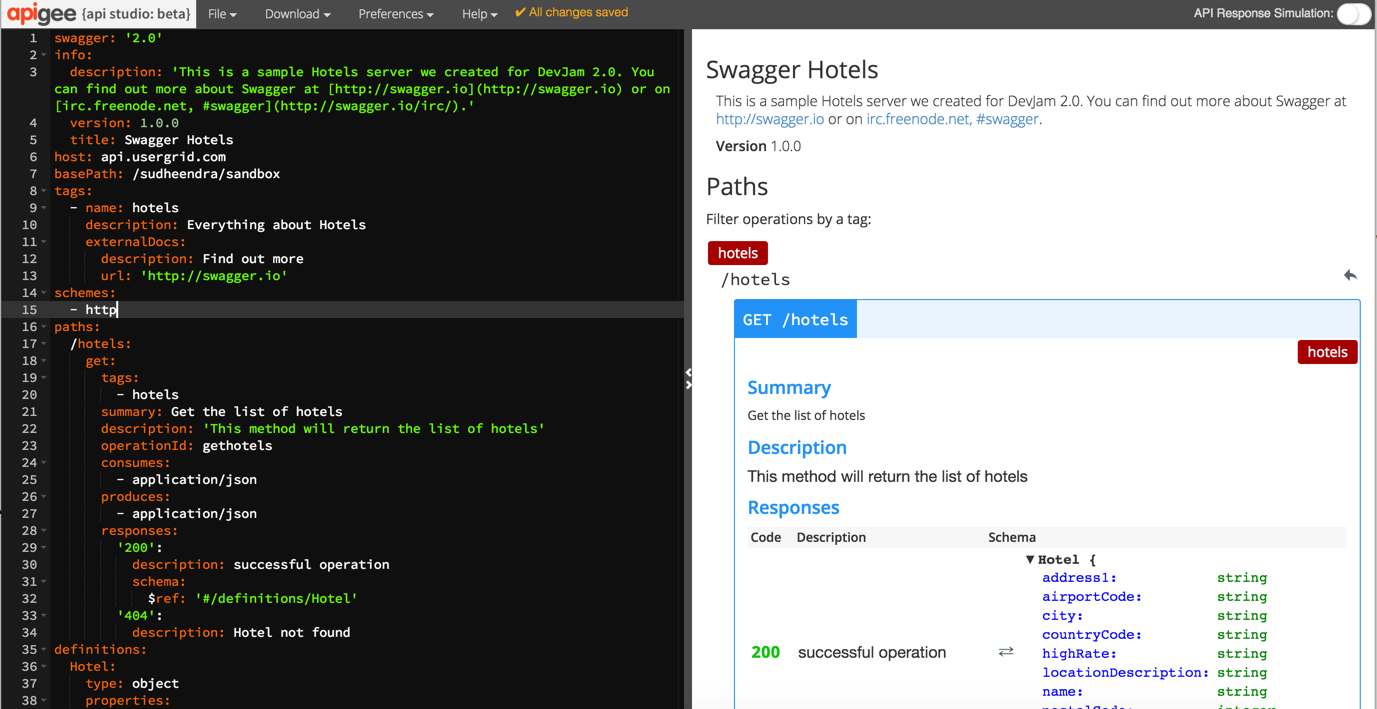
**Objectives**

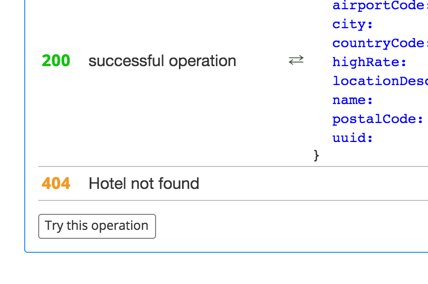
In this lab, we will describe an API contract in Swagger and invoke the API in the cloud. We will also see how an API contract in Swagger can be used to create an API proxy in Apigee Edge.

**Prerequisites**

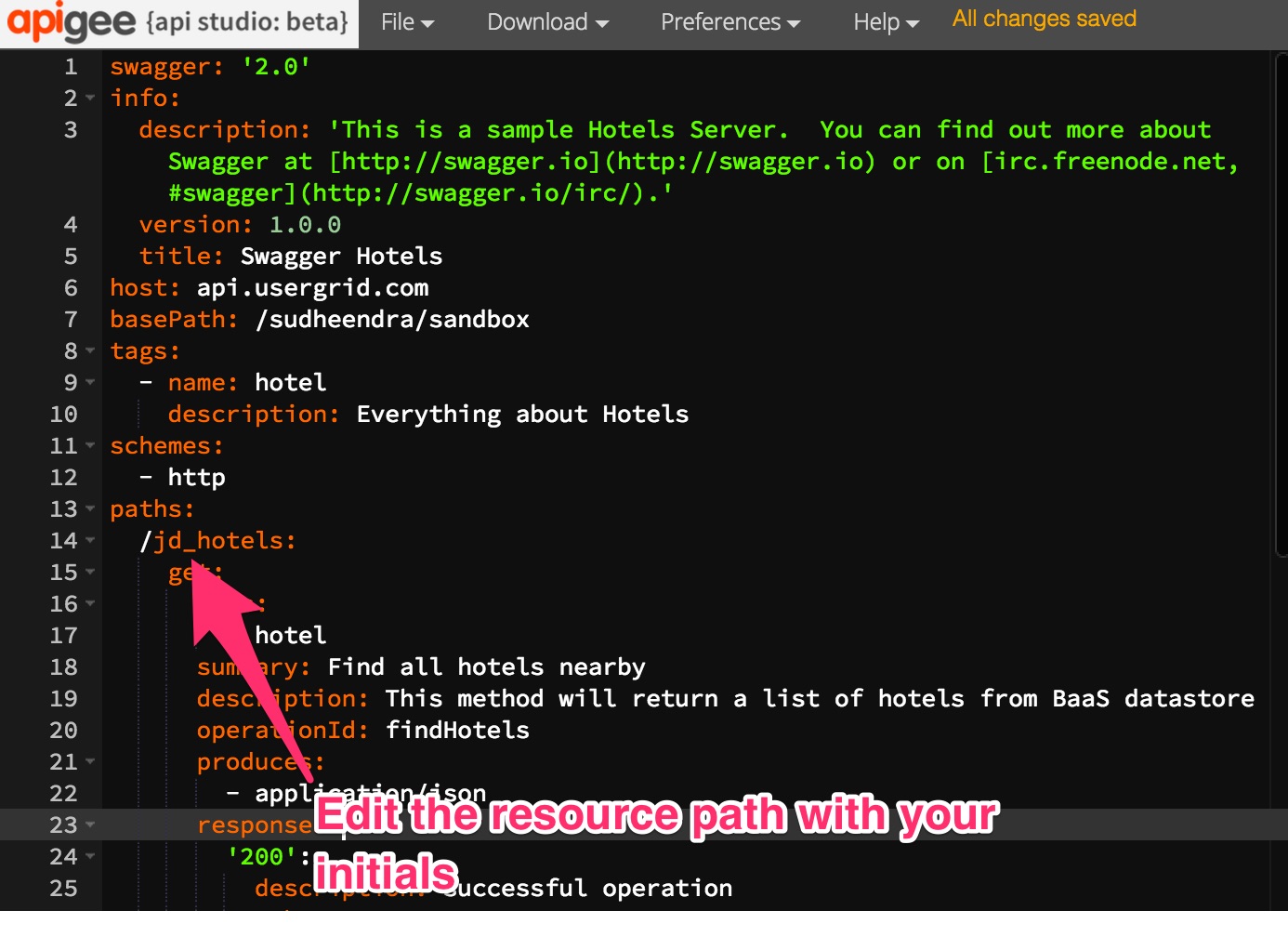
Your Apigee Edge organization name, user name, and the password. Ask your Instructor for organization and application details.

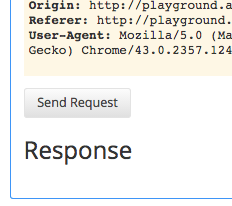
**Estimated Time: 45 mins**

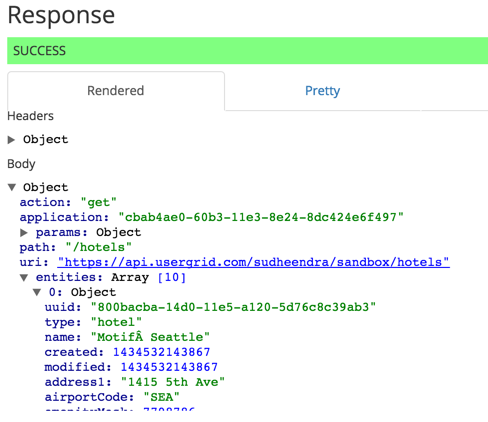
1. **Creating a Swagger specification** can be done with Apigee-127 on your local machine or in the cloud at <http://apistudio.io>, since both utilize the same Swagger editor. For this lab, we will assume that you are using <http://apistudio.io>.
   1. Copy the [Swagger YAML located here](https://gist.github.com/anonymous/90f30c5870c19e7980d9).
   2. Select the entire contents of the Swagger editor at API Studio in the browser and delete it. Paste the Swagger that you copied in its place.
   3. On the right side of the screen you will see the generated documentation updated to reflect the API described by the Swagger that you just pasted.  
        
      
   4. On the right side of the Swagger editor, in the documentation view, scroll down to the Responses section and click the Try this operation button:



* 1. Edit the Swagger to add your initials to the resource path.



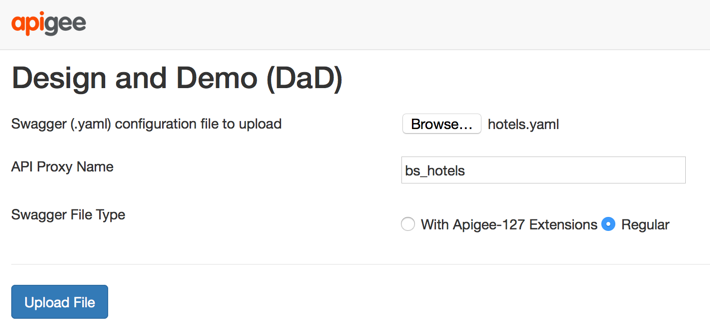
* 1. Scroll down past the request information in the yellow block, so that you can see the Send Request button. Click it to make an API request:  
     
  2. View the response below the button. Note the three tabs (Rendered, Pretty, and Raw) and the ability to view the Headers and Body :

  
   
 Please expand the Body section to view the entire response.

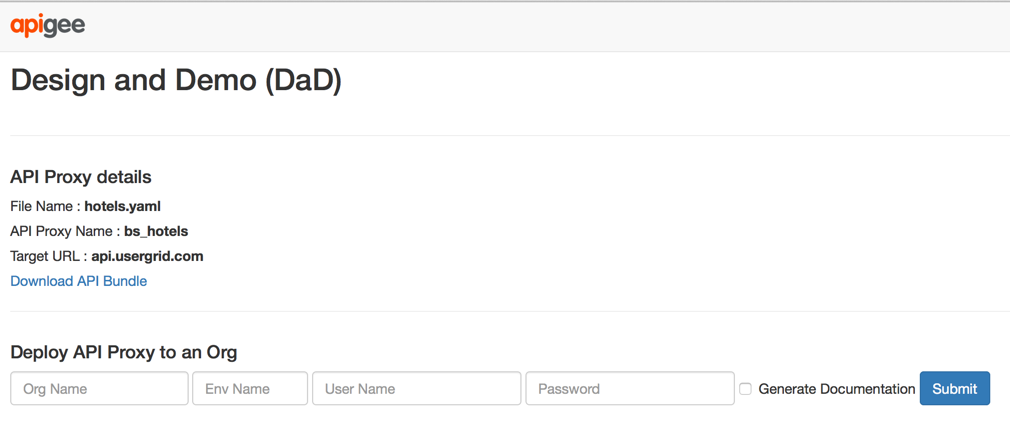
1. **Creating an API proxy from the Swagger specification** can be done using DaD (Design and Develop) utility.
   1. Copy the Swagger specification from <http://apistudio.io> editor or [from here](https://gist.github.com/anonymous/90f30c5870c19e7980d9)
   2. Edit the file and change the base path to /**{your\_baas\_org}**/sandbox/**{your\_initials}.**
   3. Replace **{your\_baas\_org}** with your BaaS org. Click on Download and save the content and rename the file as “**hotels.yaml**”.
   4. Open the DaD utility from a browser with this link - <http://54.211.26.23:8080/SwaggerFileUpload/>
   5. Browse and select “hotels.yaml” file and provide following values -

API Proxy Name : **{your\_initials}**\_hotels

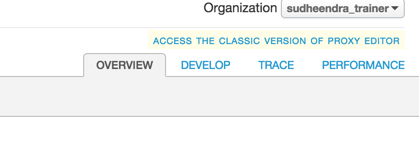
Swagger File Type : **Regular**



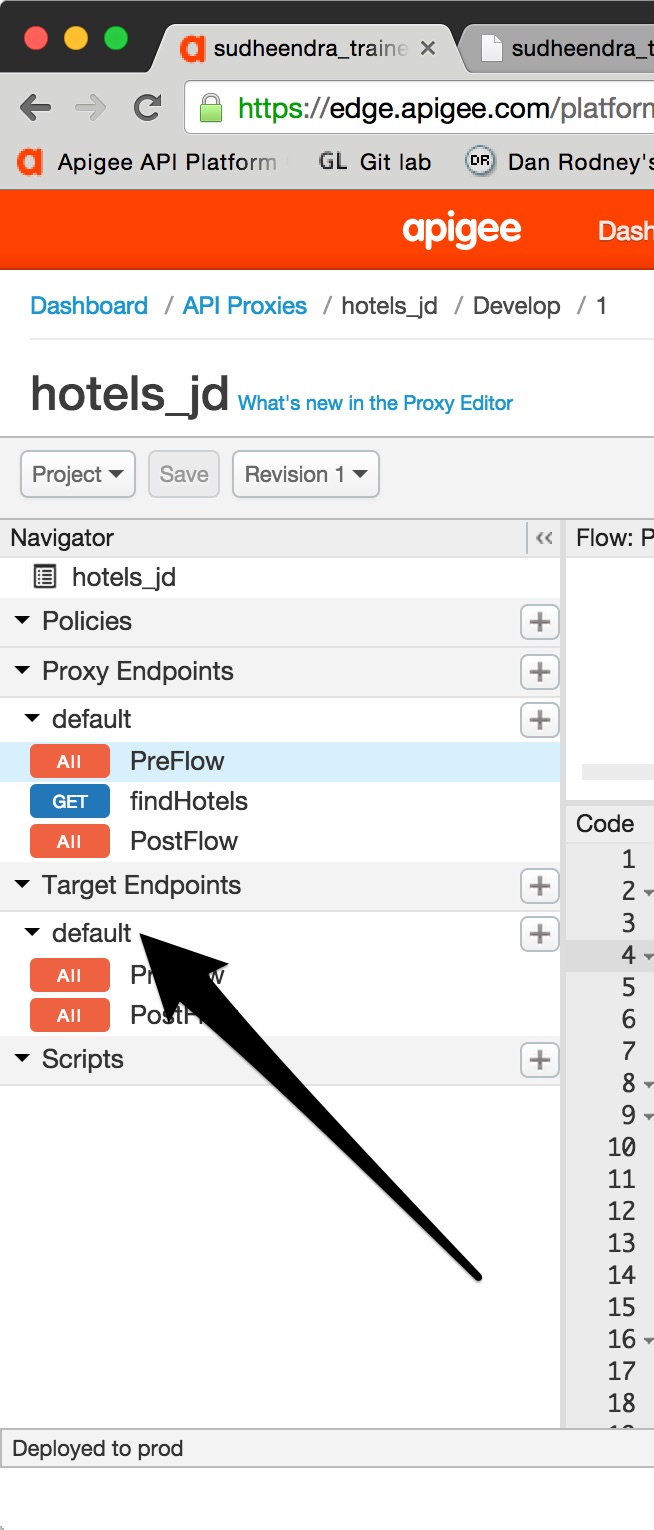
* 1. Click on Upload File.
  2. On the next screen, enter your Edge Org Name, environment name, your username and password. Uncheck “Generate Documentation”.

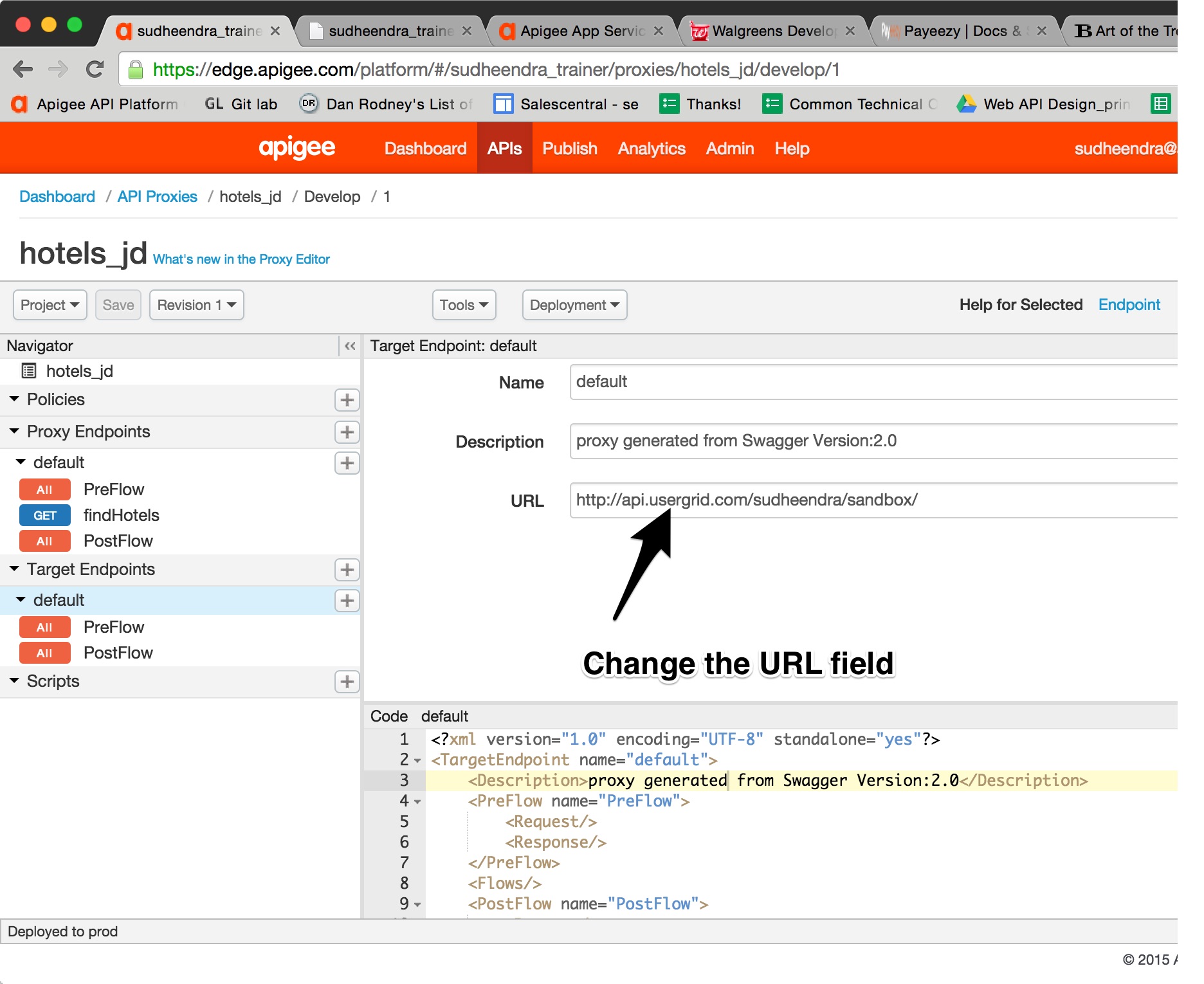


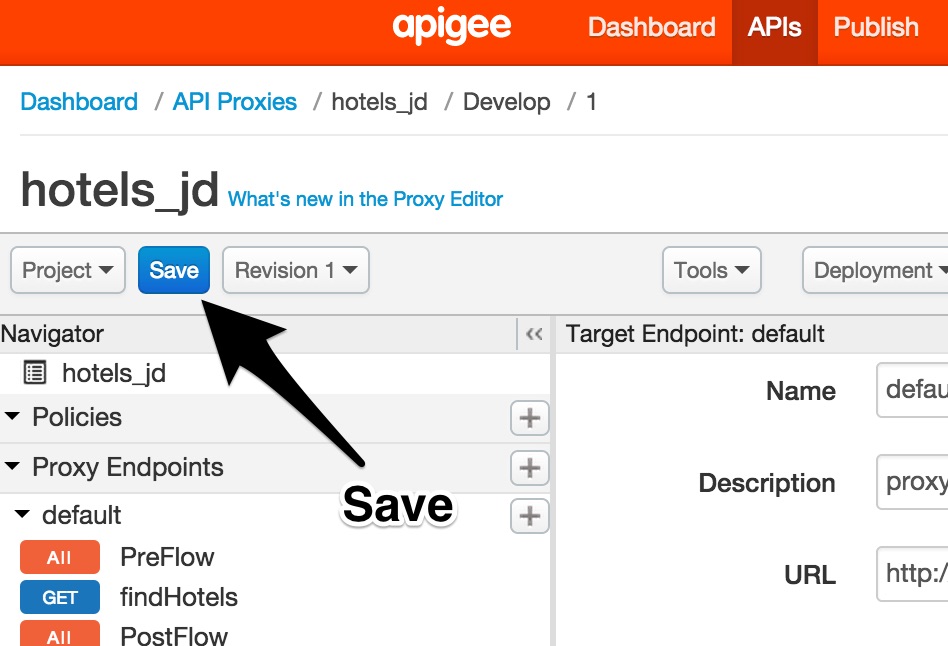
* 1. Click on Submit.
  2. This will create an API proxy on your Edge org. Wait till you see a success message.
  3. If you login to your Edge org, you will notice the new proxy with name - **{your\_initils}\_hotels.**
  4. From the main menu, select APIs → API Proxies
  5. From the ‘{your\_initials}\_hotels’ API Proxy page, click on the ‘Develop’ tab



* 1. Click on “default” section under “Target Endpoints”.



* 1. Change the URL to “http://api.usergrid.com/**{your\_baas\_org}**/sandbox/hotels”   
       
      
  2. Click on Save



* 1. You can test your API by invoking proxy from a browser.

For example -

https://**{your\_edge\_org}**-**{your\_environment}.**apigee.net/**{your\_baas\_org}**/sandbox/**{your\_initials}\_hotels/**

NOTE : Replace your Edge org name, environment name, BaaS org name and your initials before invoking.

**Summary**

That completes our first hands-on lesson. In this short lab, you learned the fundamentals of API design, creating Swagger specification for an API also understood how to create an API proxy in Apigee Edge from an existing Swagger specification.