A

REPORT Of

Project

On

**Azure Api Management**

Submitted in partial fulfilment of the requirements

for the award of the Degree of

Bachelor of Computer Applications

Of

Poornima University

Submitted To: Submitted By:

**Celebal Technologies Ujjawal Tripathi**

**Mansha singh**

**Pratap singh**

**Nikhil Yadav**

II Year, BCA

**ACKNOWLEDGEMENT**

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of my project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank **Ms. Anuradha Raheja**, for providing me an opportunity to do the project work and giving us all support and guidance which made me complete the project duly. I am extremely thankful to her for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs.

We are thankful to **Dr. Manoj Gupta, ProPresident, Poornima University** for providing us a platform to carry out this activity successfully.

We are also very grateful to **Mr. Shamneesh Sharma(HOD, BCA Deaprtment)** for his kind support and guidance.

We are thankful and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs which helped us in successfully completing our project work.

Ujjawal tripathi

Mansha singh

Nikhil yadav

Pratap Singh Shekhawat

**Abstract**

This project report is basically about how to create and deploy the APIM on Microsoft Azure and to communicate different api with it.

API Management (APIM) is a way to create consistent and modern API gateways for existing back-end services.

API Management helps organizations publish APIs to external, partner, and internal developers to unlock the potential of their data and services. Businesses everywhere are looking to extend their operations as a digital platform, creating new channels, finding new customers and driving deeper engagement with existing ones. API Management provides the core competencies to ensure a successful API program through developer engagement, business insights, analytics, security, and protection. You can use Azure API Management to take any backend and launch a full-fledged API program based on it.

**TABLE OF CONTENT**

Cover Page 1

Acknowledgment 2

Abstract 3

1.1 What is APIM …………………………………………………………….5

1.2 Api’s and operation ……………………………………………………….6-7

1.4 create an instance of Api management….…………………………………8-9

1.5 Import Api …………………..…………………………………………….9-11

1.6 Stages to Create & publish a product…………………………………….. 11-13

1.7 Stages to Mock Api response …………………………………………….13-16

1.8 access and customise developers portal…………………………………..17-18

References………………………………………………………….…………19

***What is Api Management***

 Azure API Management (APIM) is a cloud-based PaaS offering available in both commercial Azure and [Azure Government](https://www.appliedis.com/what-we-do/microsoft-azure-government-compliance-security-services/). APIM provides a one-stop-shop for API authority, with the ability to create products, enforce policies, and utilize a robust developer portal.

To use an API, developers subscribe to a product that contains that API, and then they can call the API's operation, subject to any usage policies that may be in effect. Common scenarios include

* **Securing mobile infrastructure**
* **Enabling ISV partner ecosystems**
* **Running an internal API program**

The system is made up of the following components:

* The **API gateway** is the endpoint that:
  + Accepts API calls and routes them to your backends.
  + Verifies API keys, JWT tokens, certificates, and other credentials.
  + Enforces usage quotas and rate limits.
* The **Azure portal** is the administrative interface where you set up your API program. Use it to:
  + Define or import API schema.
  + Package APIs into products.
* The **Developer portal** serves as the main web presence for developers, where they can:
  + Read API documentation.
  + Try out an API via the interactive console.
  + Create an account and subscribe to get API keys.

**APIs and operations**

APIs are the foundation of an API Management service instance. Each API represents a set of operations available to developers. Each API contains a reference to the back-end service that implements the API, and its operations map to the operations implemented by the back-end service.

**Products**

Products are how APIs are surfaced to developers. Products in API Management have one or more APIs, and are configured with a title, description, and terms of use. Products can be **open** or **protected**. Protected products must be subscribed to before they can be used, while open products can be used without a subscription.

**Groups**

Groups are used to manage the visibility of products to developers. API Management has the following immutable system groups:

* **Administrators** - Azure subscription administrators are members of this group. Administrators manage API Management service instances, creating the APIs, operations, and products that are used by developers.
* **Developers** - Authenticated developer portal users fall into this group. Developers are the customers that build applications using your APIs.
* **Guests** - Unauthenticated developer portal users, such as prospective customers visiting the developer portal of an API Management instance fall into this group.

**Developers**

Developers represent the user accounts in an API Management service instance. Developers can be created or invited to join by administrators, or they can sign up from the [Developer portal](https://docs.microsoft.com/en-us/azure/api-management/api-management-key-concepts#developer-portal). Each developer is a member of one or more groups, and can subscribe to the products that grant visibility to those groups.

**Policies**

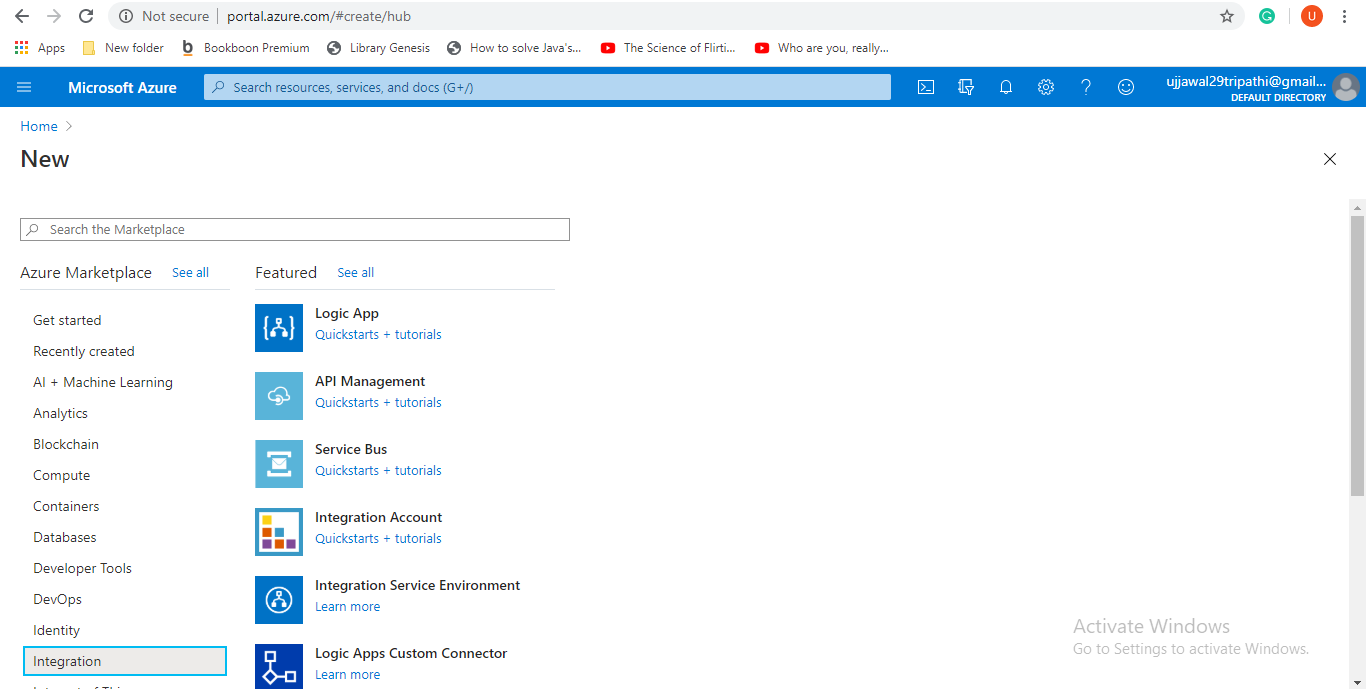
Policies are a powerful capability of API Management that allow the Azure portal to change the behavior of the API through configuration. Policies are a collection of statements that are executed sequentially on the request or response of an API. Popular statements include format conversion from XML to JSON and call rate limiting to restrict the number of incoming calls from a developer, and many other policies are available.

**Developer portal**

The developer portal is where developers can learn about your APIs, view and call operations, and subscribe to products. Prospective customers can visit the developer portal, view APIs and operations, and sign up. The URL for your developer portal is located on the dashboard in the Azure portal for your API Management service instance.

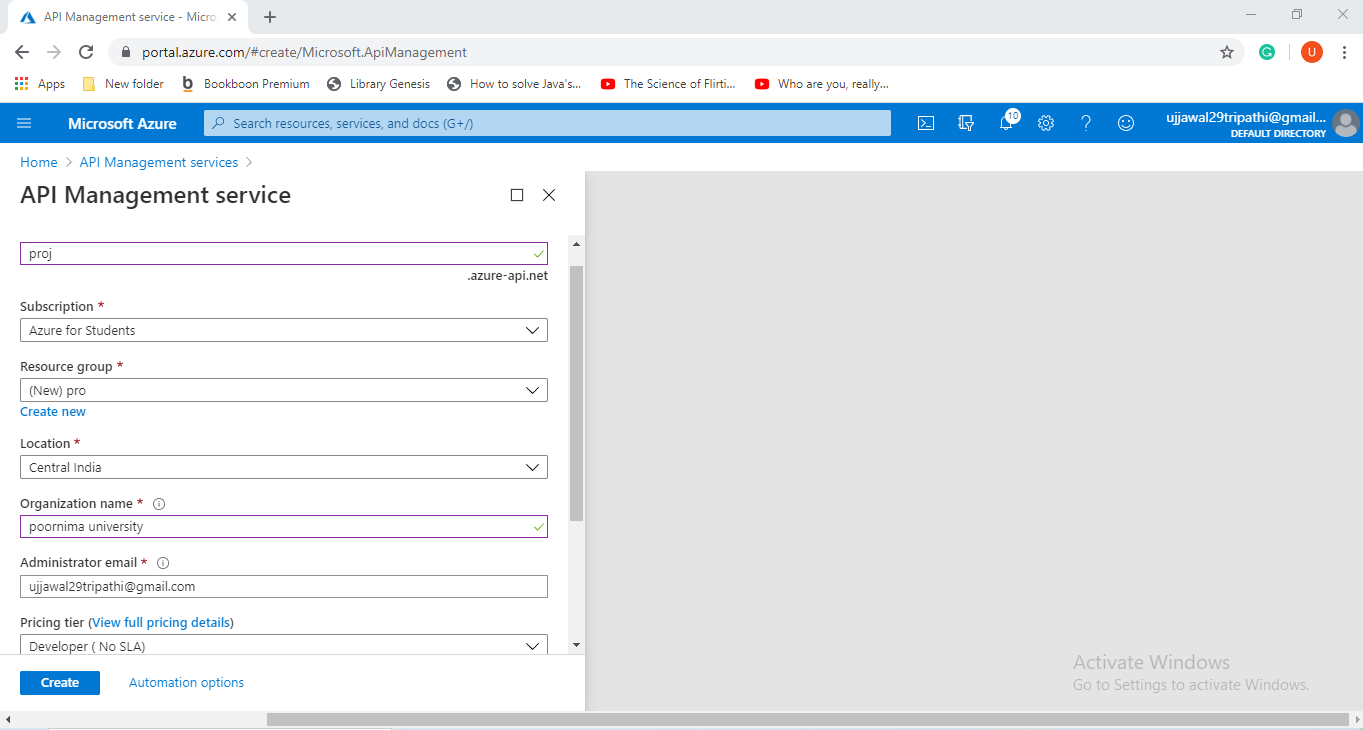
***Create instance of API Management***

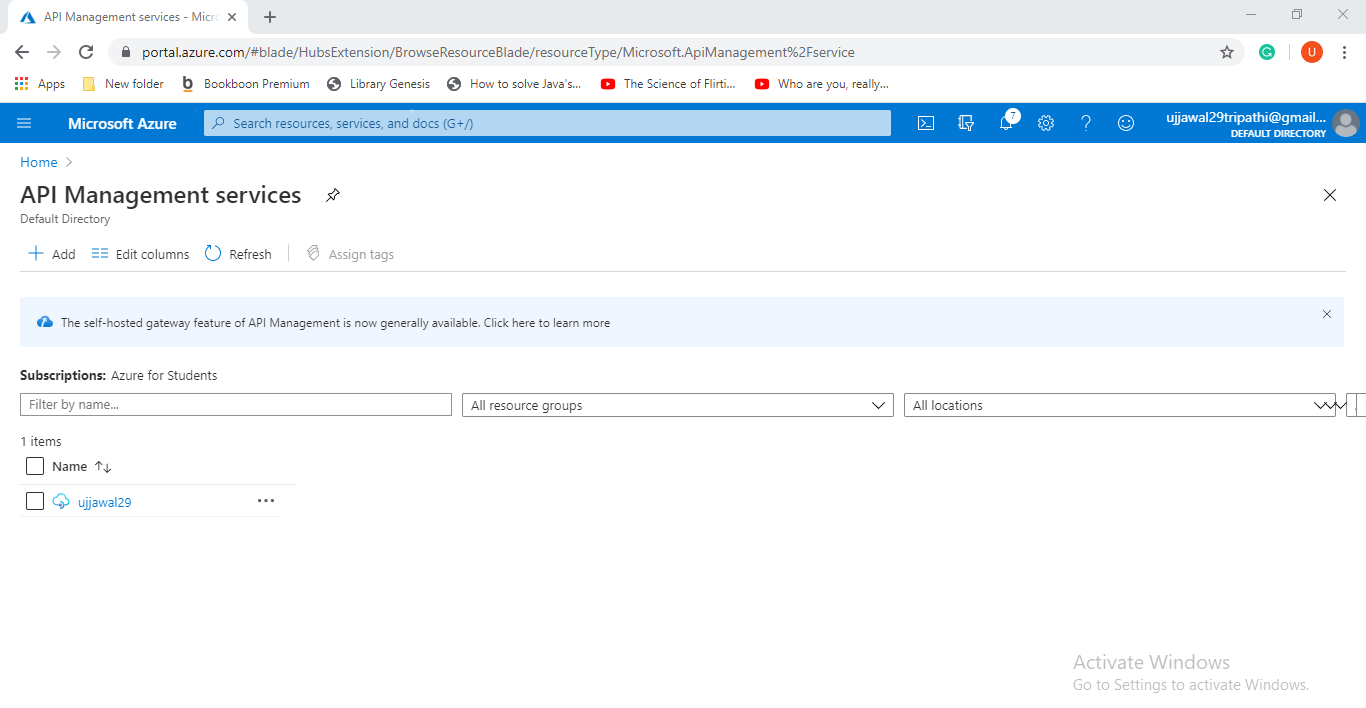
Stage 1 On the New screen, select Integration and afterward select API Management.



Stage 2 On the New screen, select Integration and afterward select API Management.

Stage 3 In the API Management administration screen, enter settings.

Stage 4 In the Azure entry, scan for and select API Management administrations. 

Stage 5 On the API Management screen, select your API Management example with resource pack.

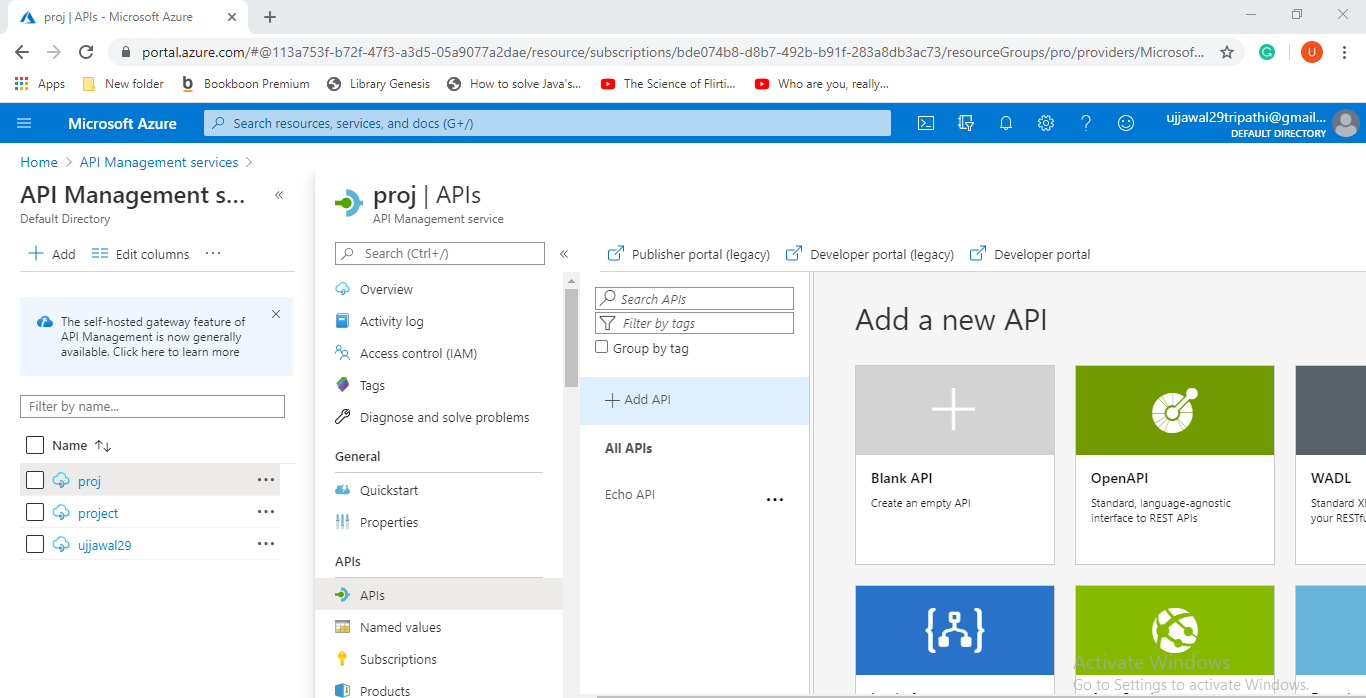
Stage 6 Pick Create.

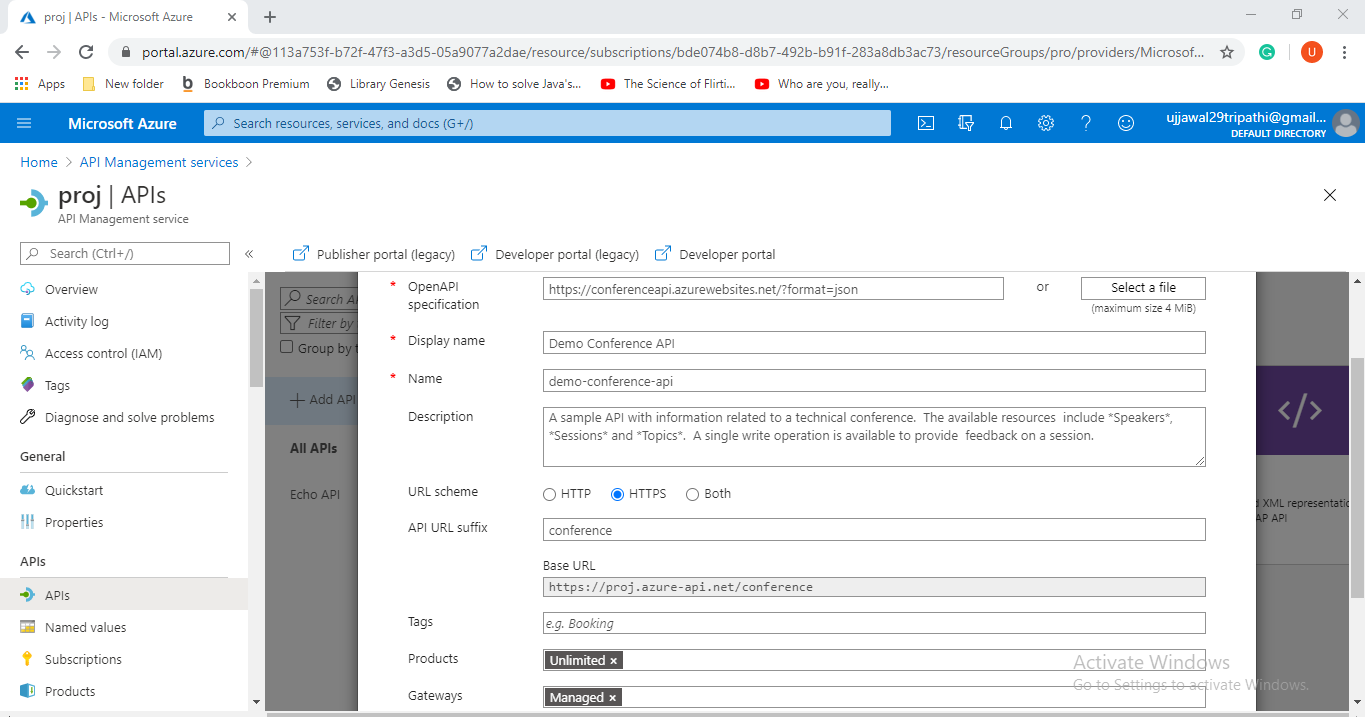
***Import API***

Stage 1 In the Azure entrance, scan for and select API Management administrations.

Stage 2 On the API Management screen, select your API Management case.

Stage 3 In the left route of your API Management case, select APIs from the API Management segment.



Stage 4 Select the OpenAPI tile, and afterward select Full on the spring up screen. 

Stage 5 On the Create from OpenAPI particular screen, utilize the qualities from the accompanying table to make your API.

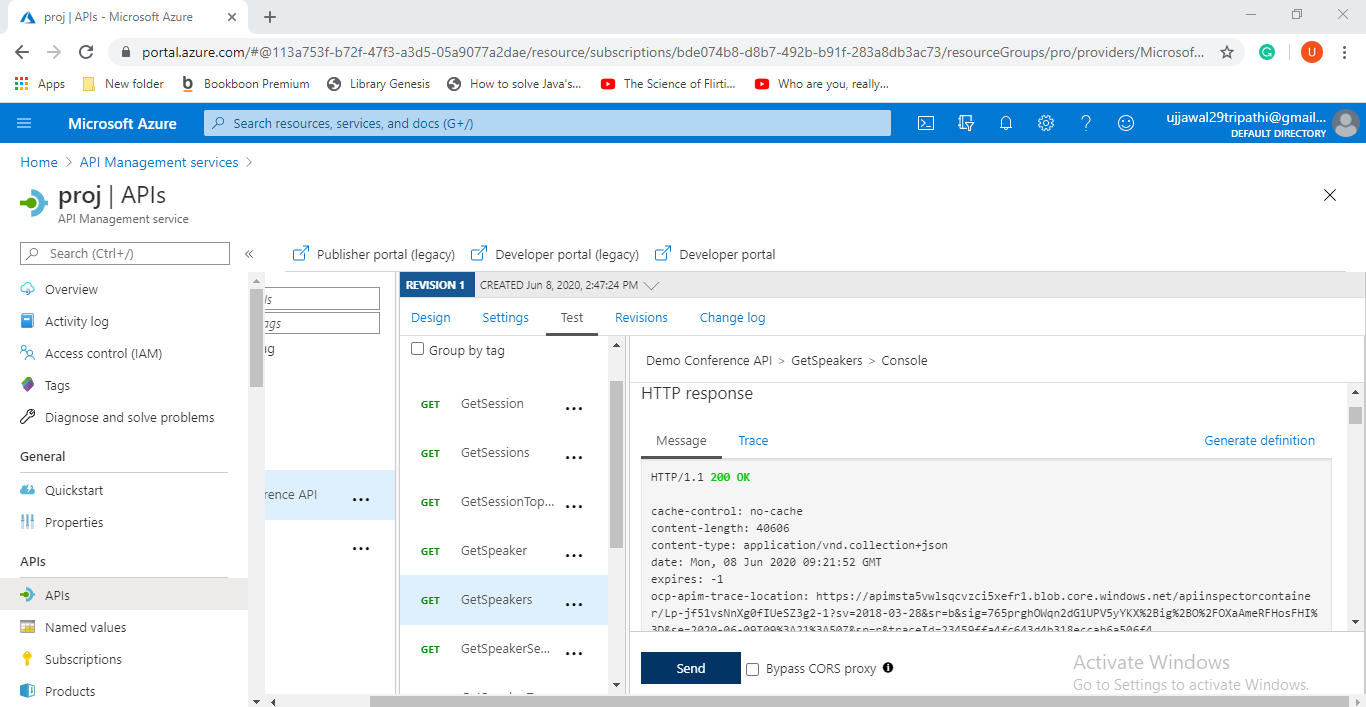
Stage 6 Select Create.

**Testing**

Stage 1 In the left route of your API Management case, select APIs from the API Management area, and afterward select Demo Conference API.

Stage 2 Select the Test tab, and afterward select GetSpeakers. The page shows Query parameters and Headers, assuming any. The Ocp-Apim-Subscription-Key is filled in consequently for the membership key related with this API.

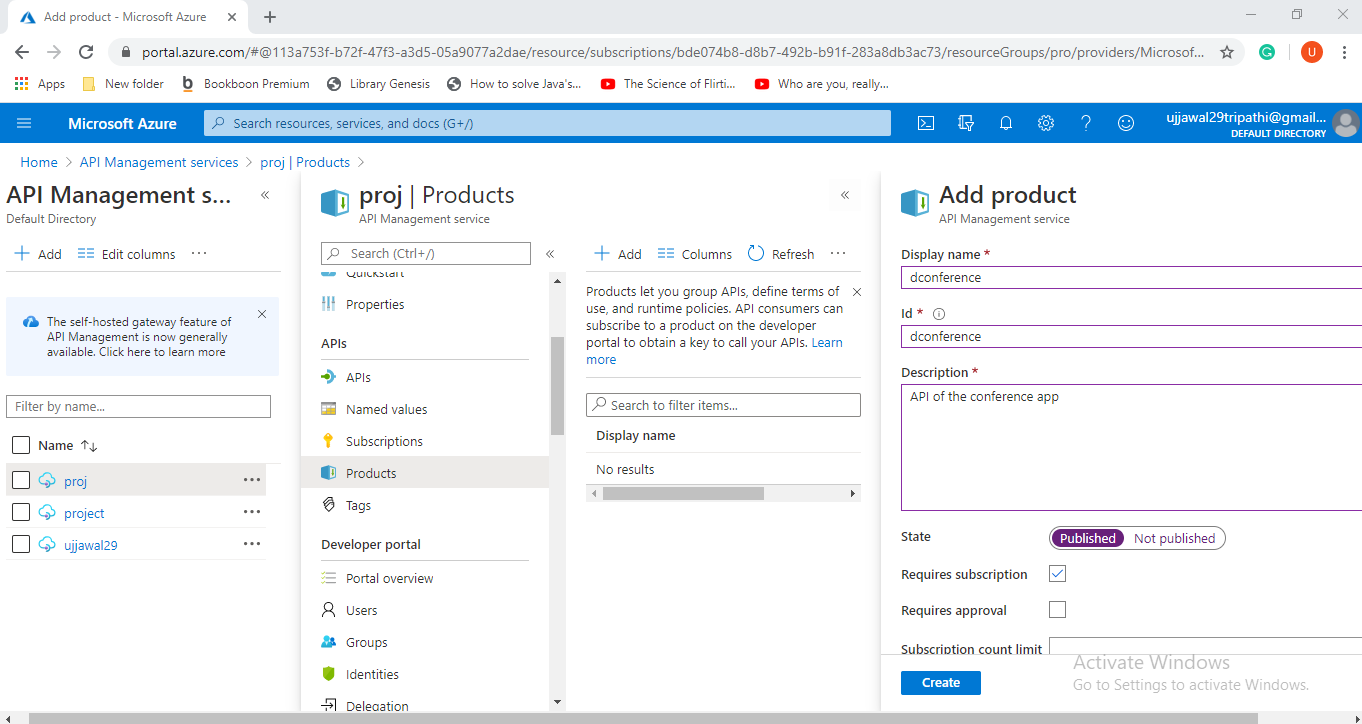
Stage 3 Select Send.



***Stages to Create & publish a product***

Stage 1 Click on Products in the menu on the left to show the Products page.

Stage 2 Click + Add.

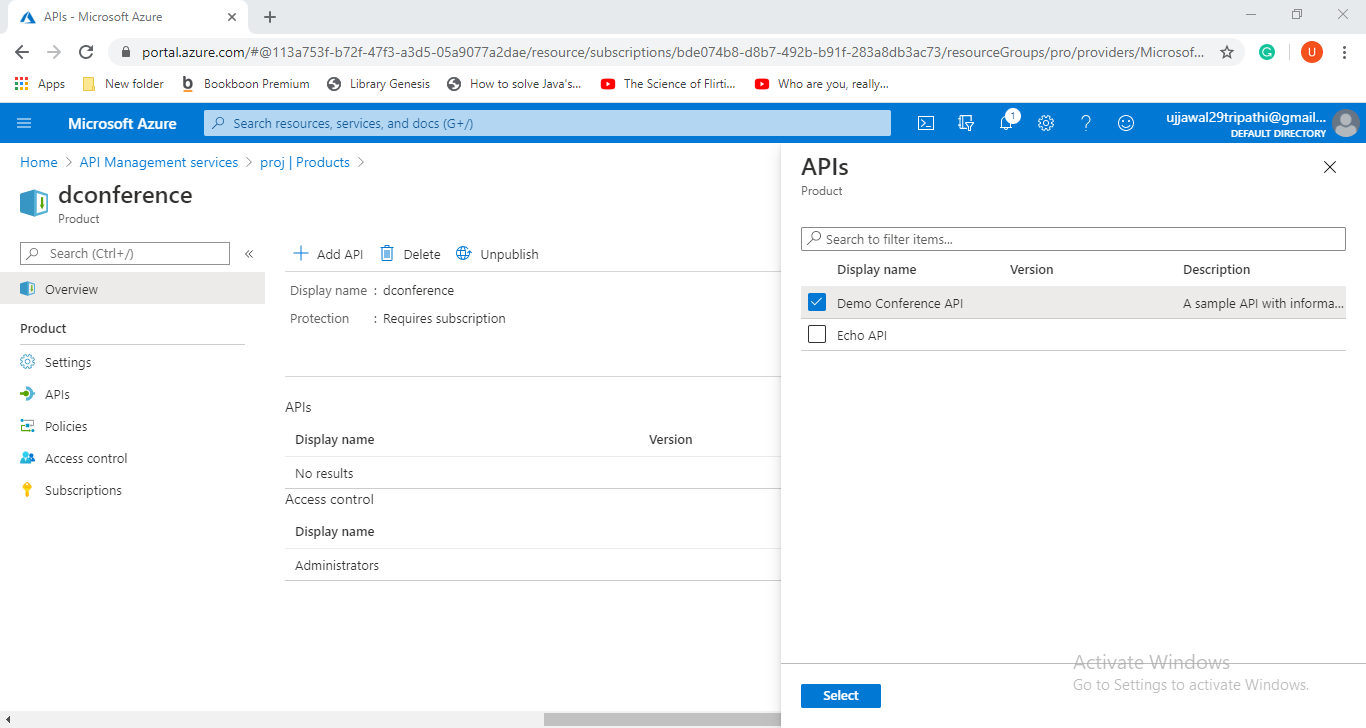


Stage 3 Click Create to make the new item.

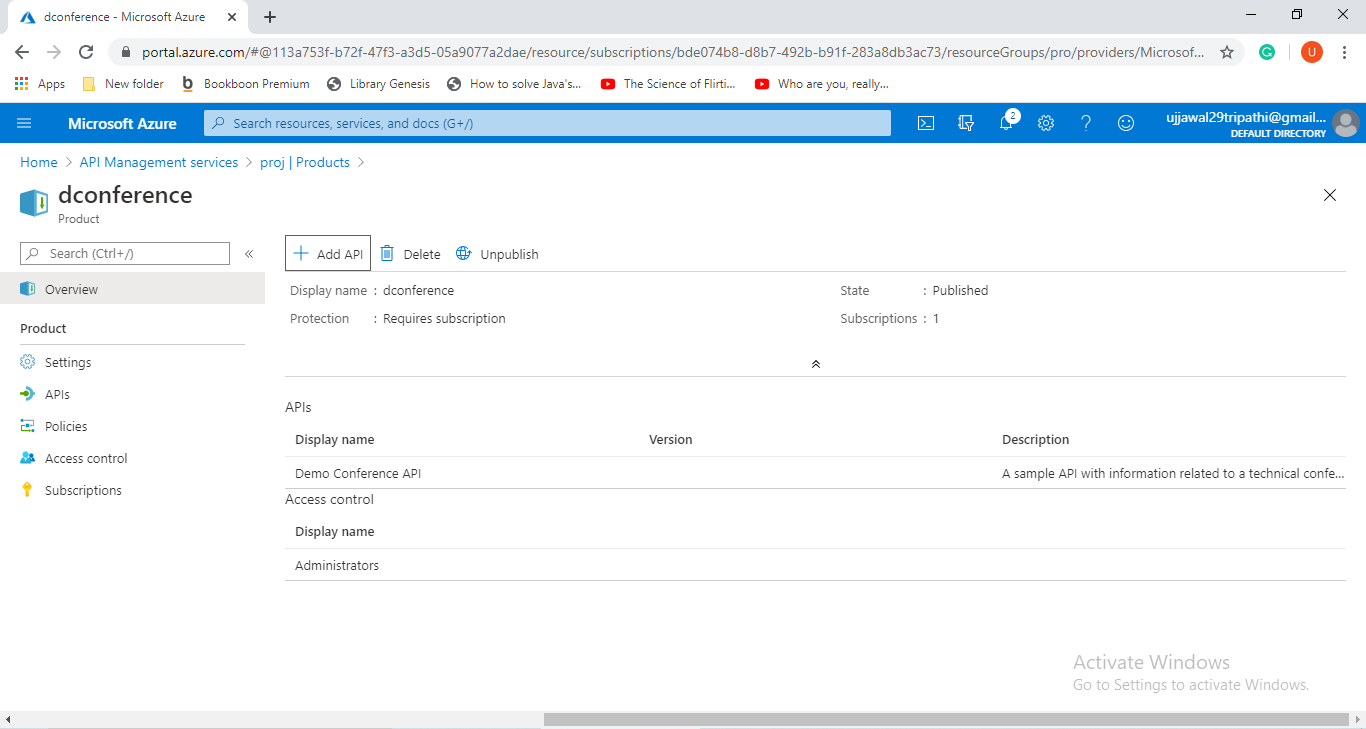
Stage 4 From the Products tab, select a product.

Stage 5 Navigate to the APIs tab.

Stage 6 Click + Add.



Stage 7 Choose an API and click Select.

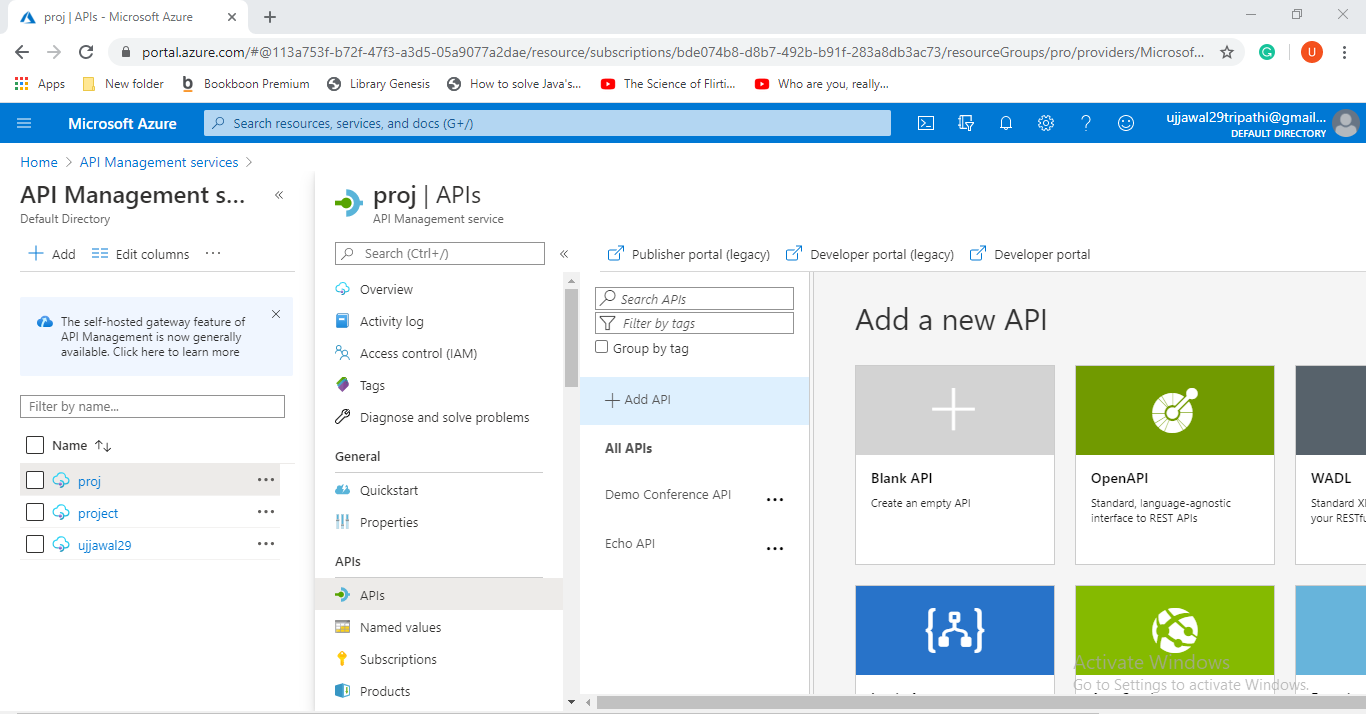


***Mock API RESPONSE***

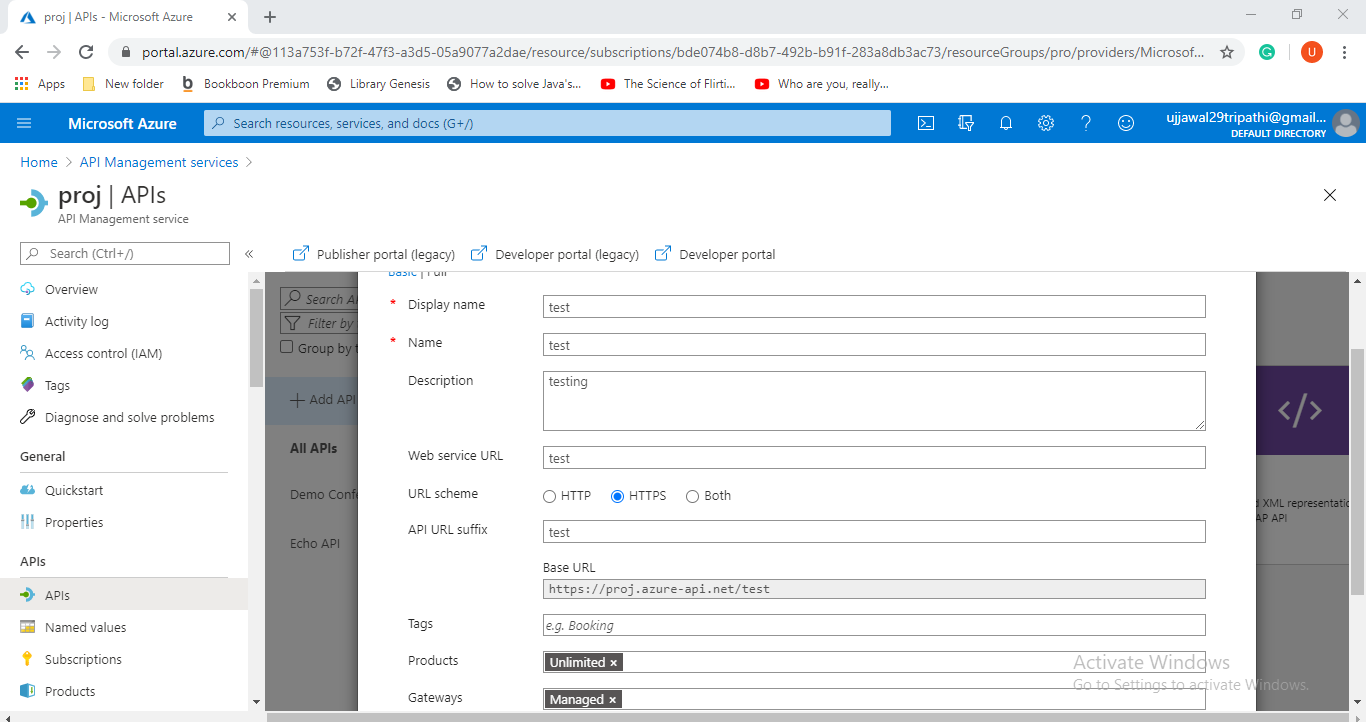
Stage 1 Select APIs from the API Management administration.

Stage 2 From the left menu, select + Add API.

Stage 3 Select Blank API from the rundown.

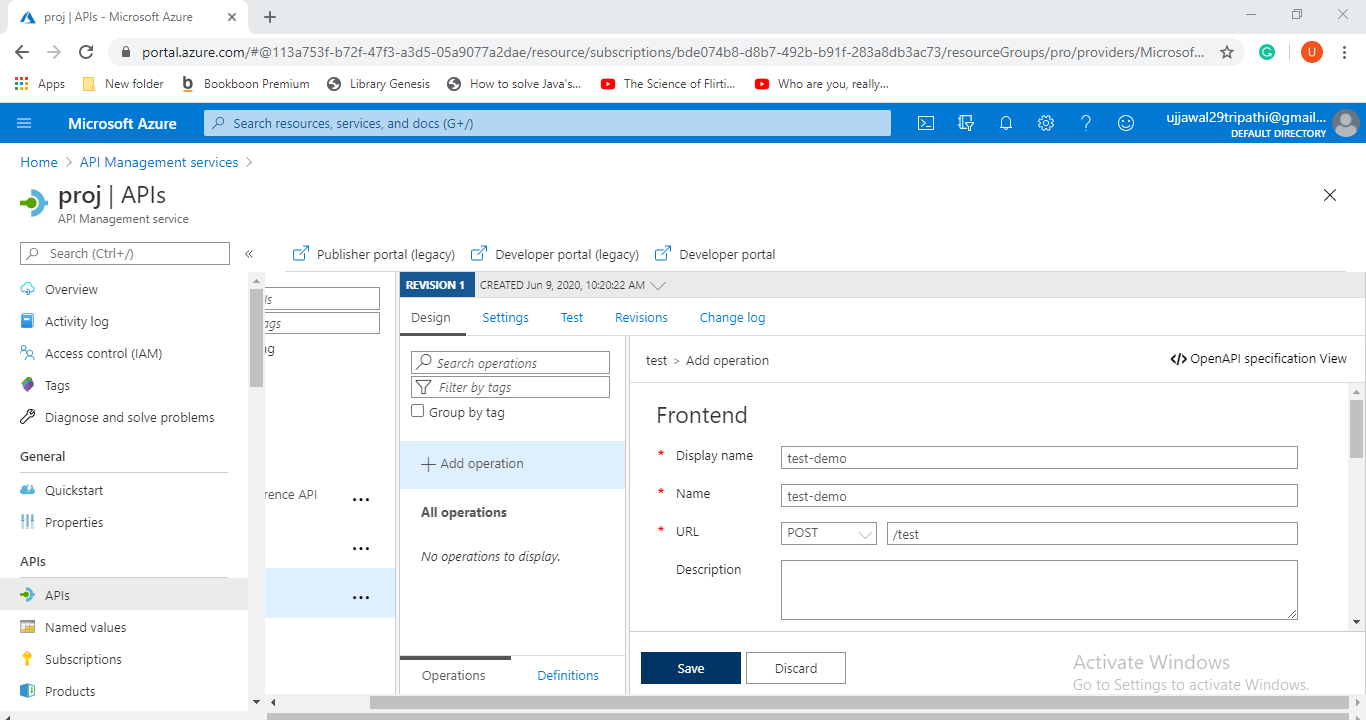


Stage 4 Enter "Test” for Display name.



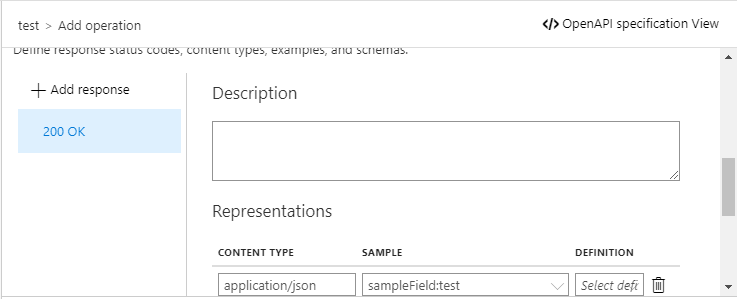
Stage 6 Select Create.

Stage 7 Select the API you made in the past advance.

Stage 8 Click + Add Operation. 

Stage 9 Select the Response tab, situated under the URL, Display name, and Description fields.

Stage 10 Snap + Add reaction.

Stage 11 Select 200 OK from the rundown. 

Stage 12 Under the Representations heading on the right, select + Add portrayal.

Stage 13 Enter "application/json" into the pursuit box and select the application/json content sort.

Stage 14 In the Sample content box, enter { "sampleField" : "test" }.

Stage 15 Select the API you made in the "Make a test API" Stage.

Stage 16 Select the test activity that you included.

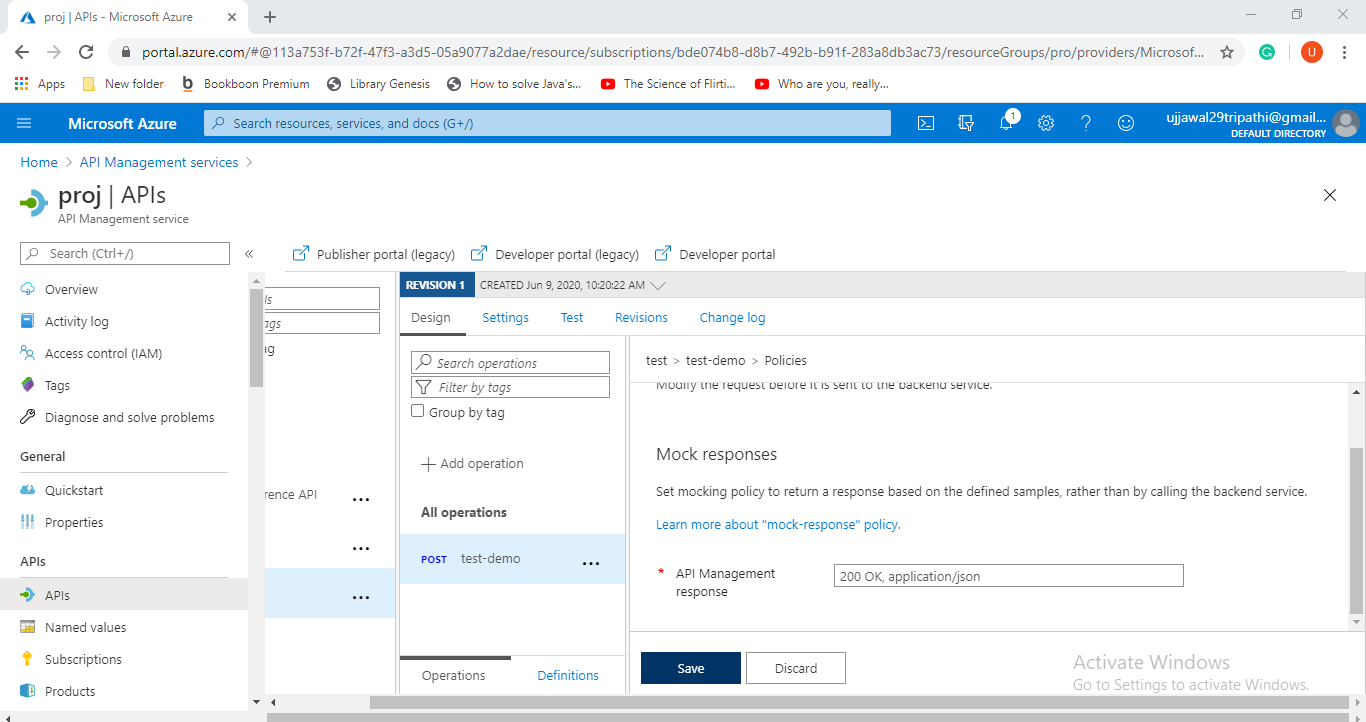
Stage 17 In the window on the right, click the Design tab.

Stage 18 In the Inbound handling window, click + Add approach.

Stage 19 Select the Mock reactions tile from the display.

Stage 20 In the API Management reaction textbox, type 200 OK, application/json. This choice demonstrates that your API should restore the reaction test you characterized in the past area.

Stage 21 Save.

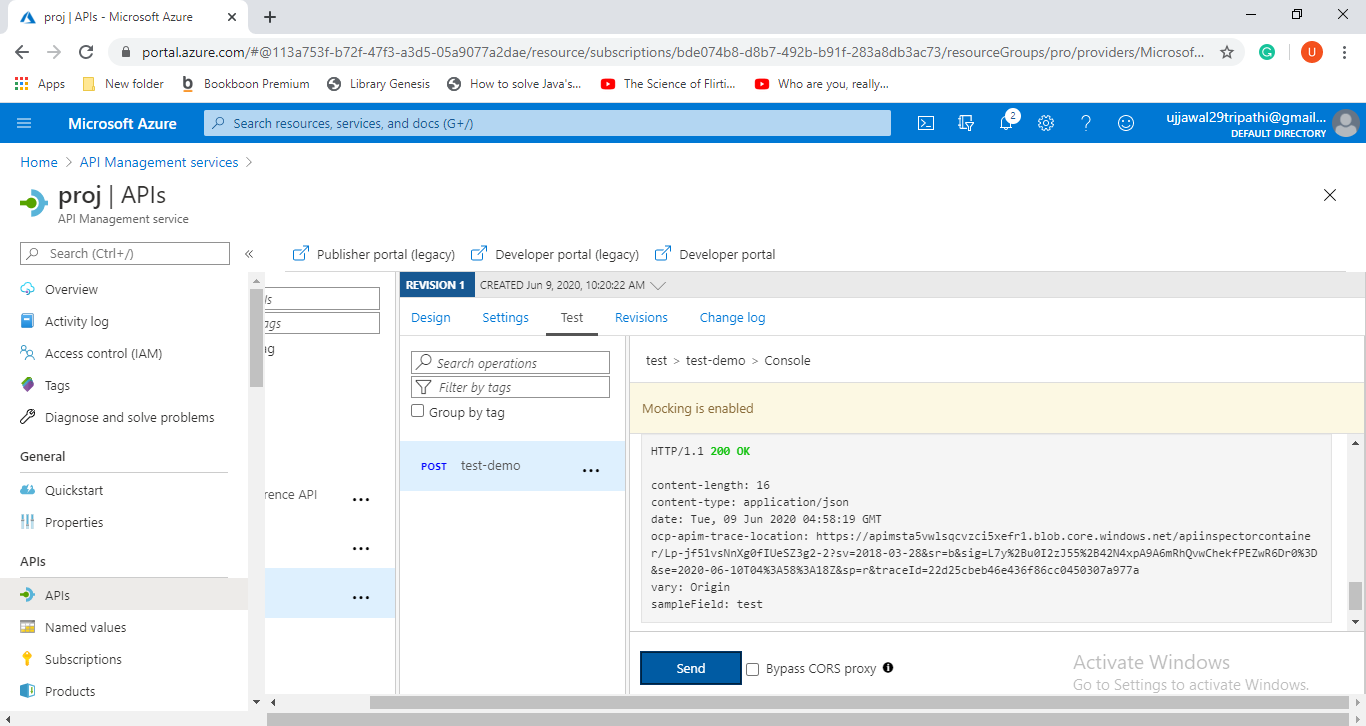


Access and customise developer’s portal

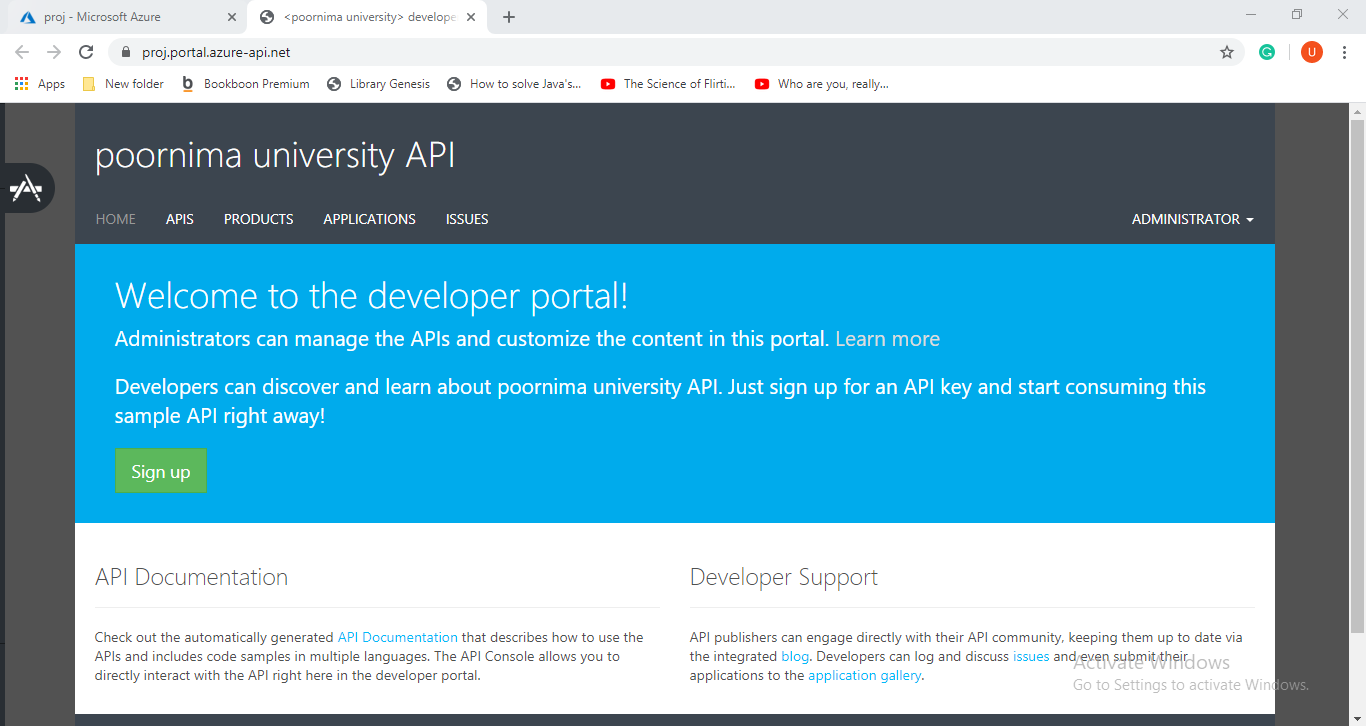
Stage 1 Select the API you made in the "Make a test API" Stage.

Stage 2 Open the Test tab.

Stage 3 Guarantee the Test call API is chosen.

Stage 4 Select Send to make a test call. 

Stage 6 Snap the Developer gateway (inheritance) button on the highest point of the Overview window.



Stage 7 on the upper left half of the screen, you see a symbol involved two paint brushes. Float over this symbol to open the gateway customization menu.

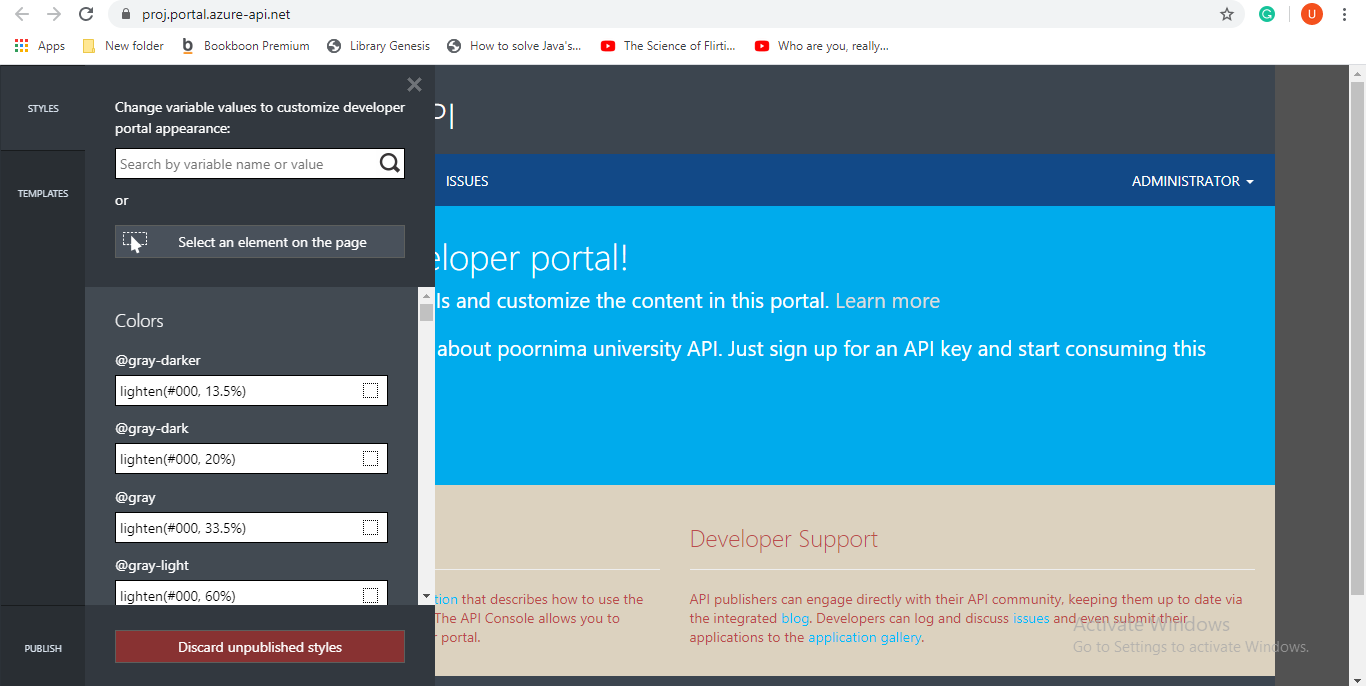
Stage 8 Select Styles from the menu to open the styling customization sheet.

Stage 9 All components that you can alter utilizing Styles show up on the page

Stage 10 Enter "headings-shading" in the Change variable qualities to redo engineer entry appearance: field.

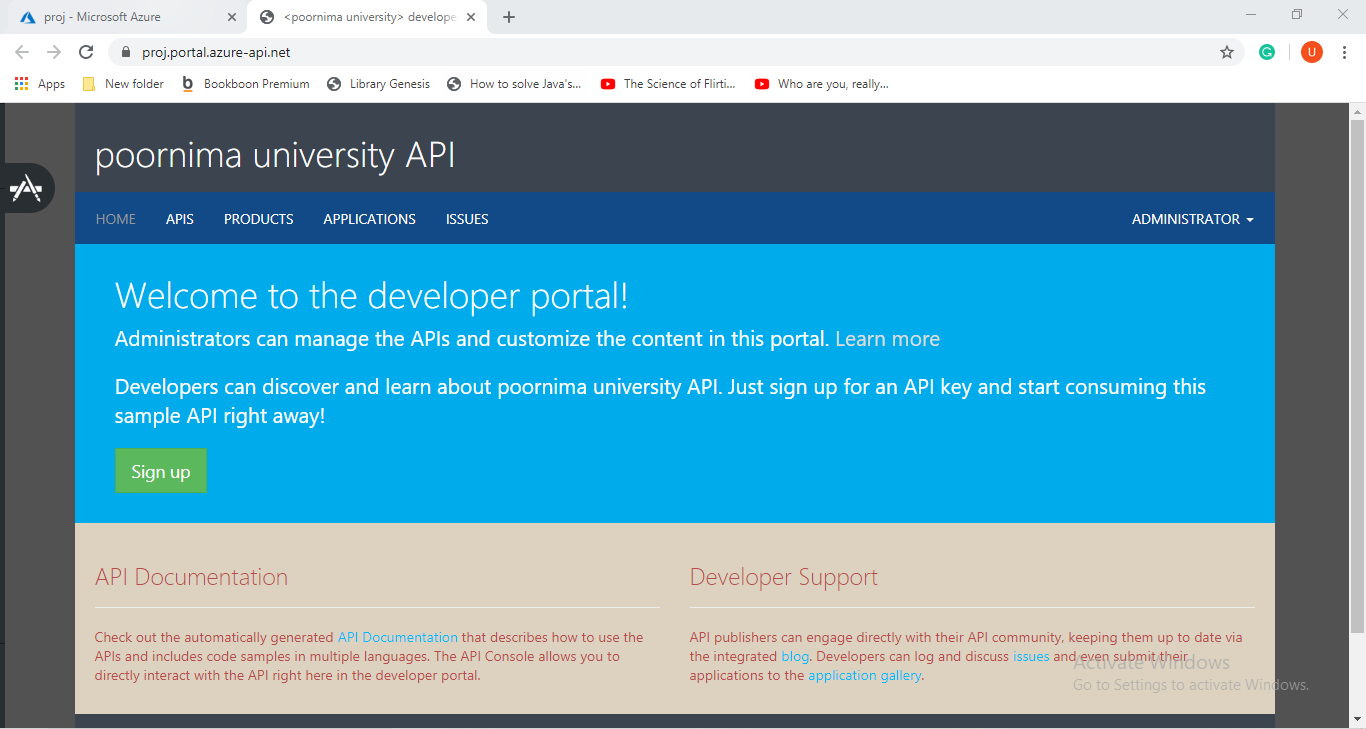
Stage 11 The @headings-shading component shows up on the page. This variable controls the shade of the content.

Stage 12 Snap on the field for the @headings-shading variable.



Stage 13 Select Publish from the lower left on the customization sheet menu.

Stage 14 Select Publish customizations to roll out the improvements openly accessible.



**References-**

* jakeydocs.readthedocs.io
* www.informit.com
* docs.microsoft.com
* microsoft.github.io
* azure.microsoft.com