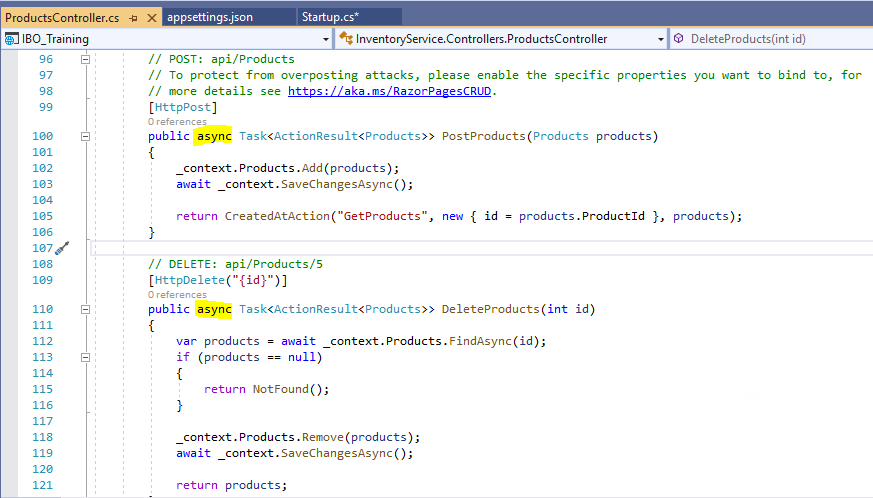
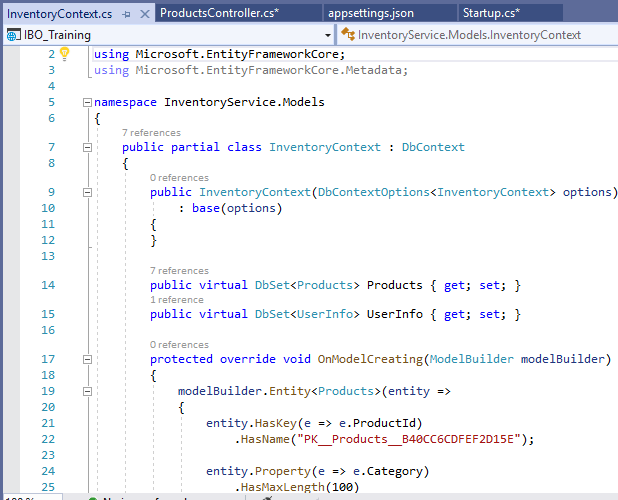
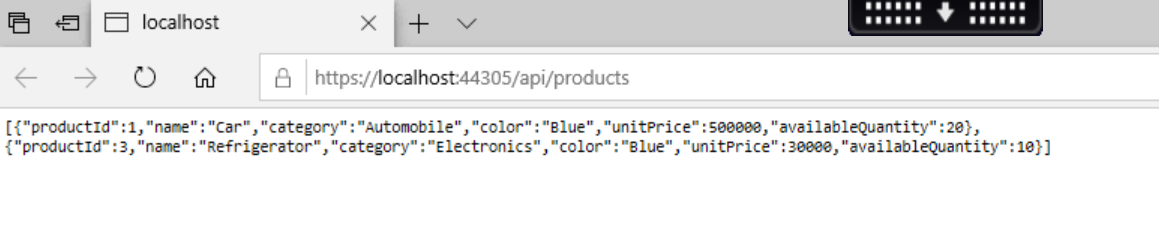
**Implemented CRUD REST API as Asynchronous APIs with version v1.0**

****

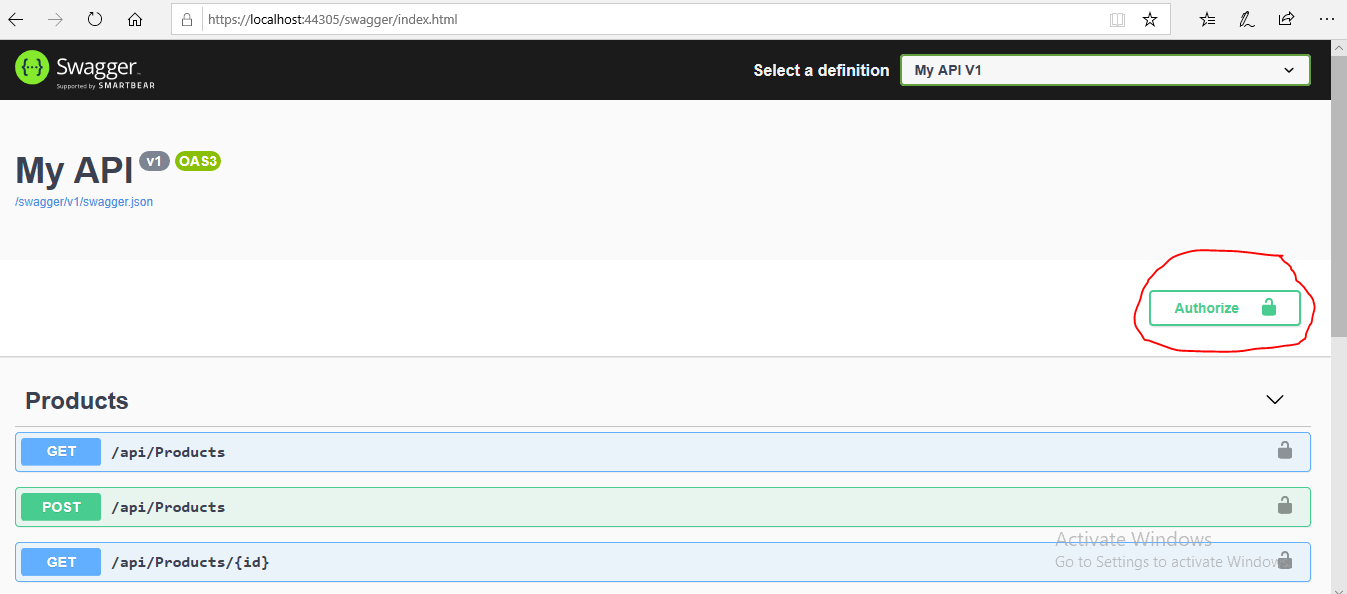
**Used Entity Framework Core ORM and Entity Framework migration to work with Database:**



**Used Browser for API Call:**



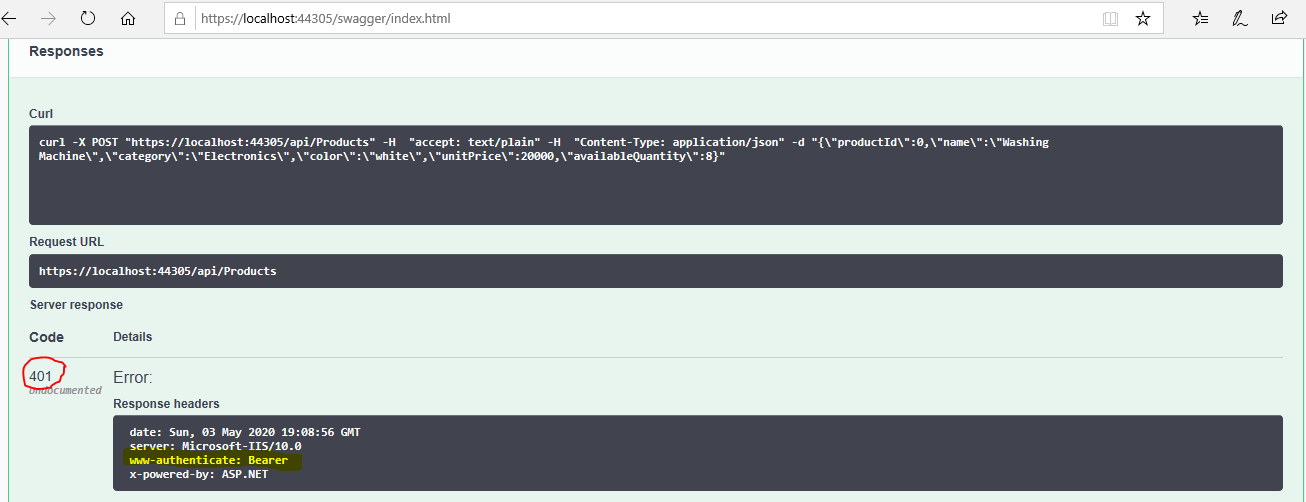
**Enabled Swagger with Authentication for performing CRUD operation:**



**JWT Authentication for Microservices in .NET:**

The tokens that I used are [JSON Web Tokens](https://tools.ietf.org/html/rfc7519)(JWT, which is “a compact, URL-safe means of representing claims to be transferred between two parties.”) Basically, a JWT is an encoded JSON object, which is then signed either with a secret key, or a public/private key pair.

A JWT is composed of three different parts: the header, the payload and the signature.



**Credentials I used for authentications:**

{

"userId": 0,

"firstName": "Inventory",

"lastName": "Admin",

"userName": "InventoryAdmin",

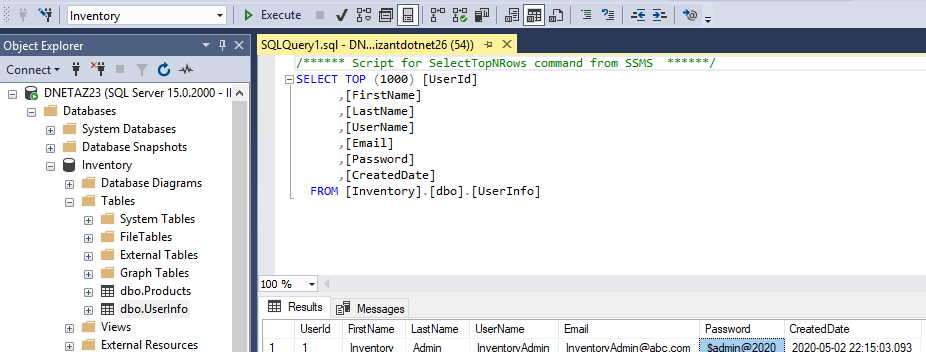
"email": "InventoryAdmin@abc.com",

"password": "$admin@2020",

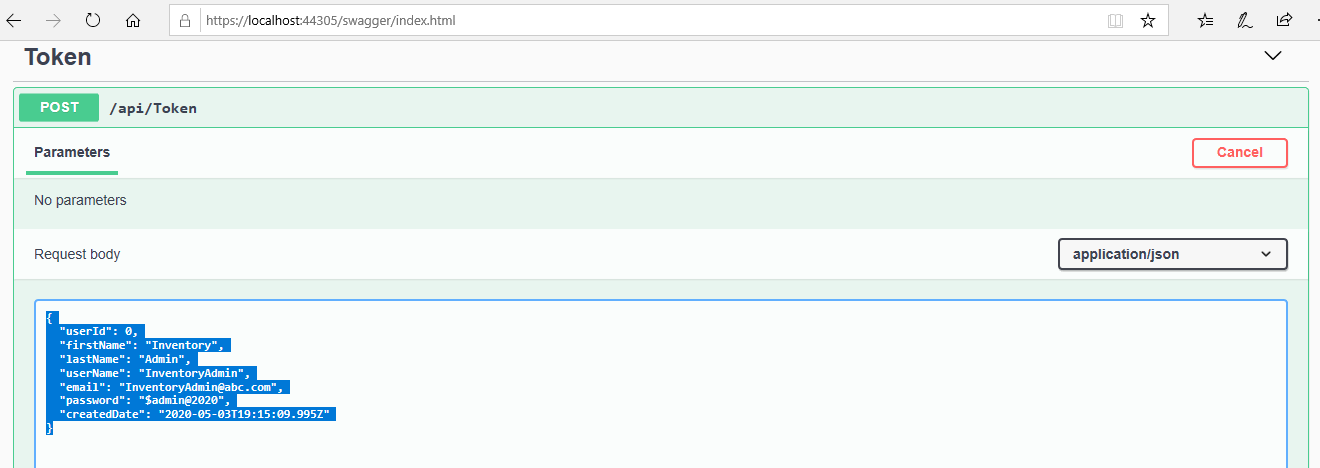
"createdDate": "2020-05-03T19:15:09.995Z"

}

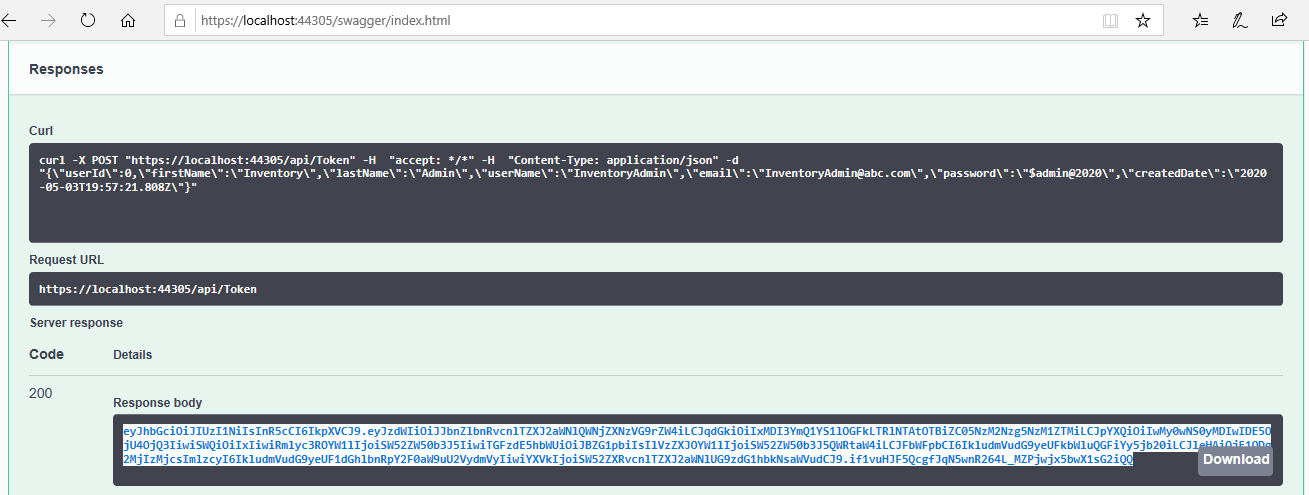
**Credentials for authentications are fetched from Database Table – ‘UserInfo’:**



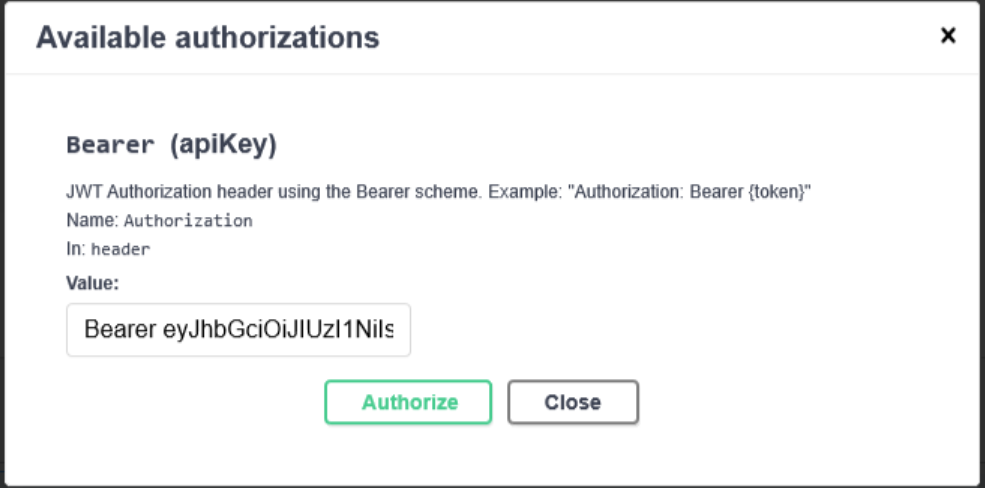
**In the Token section, enter the above details to generate the token:**



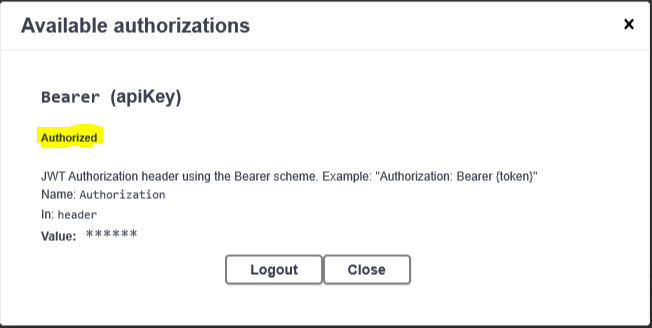
**Token generated can be copied/downloaded:**



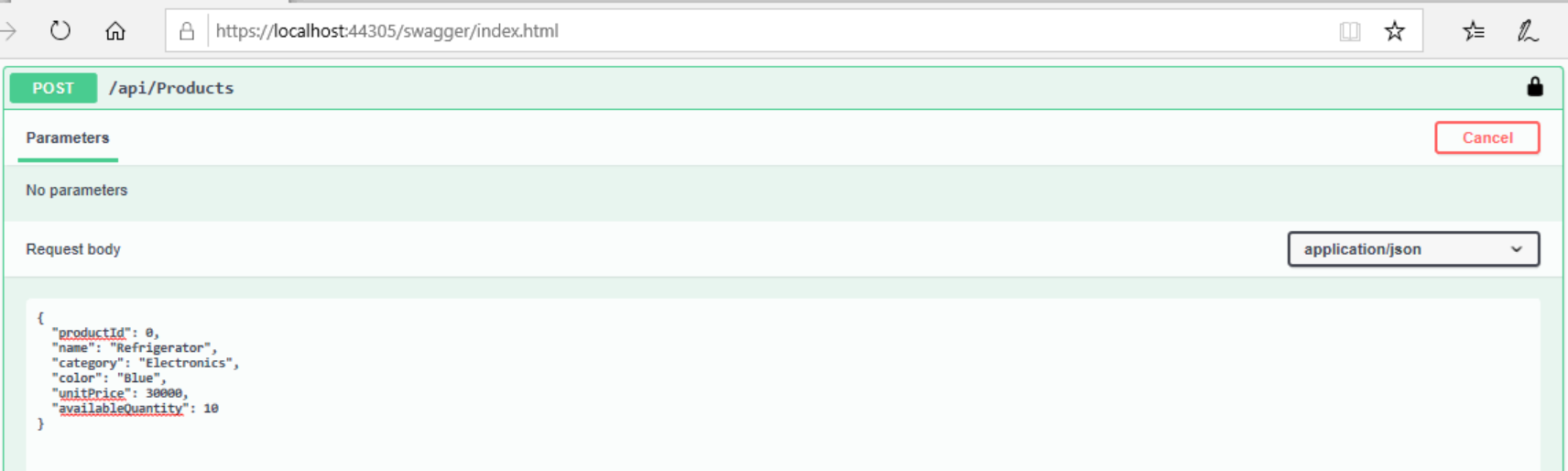
**Click on the ‘Authorize’ button at the top right corner of the Swagger UI and paste the token in the Value field with prepending ‘Bearer’ with it, Click on ‘Authorize’**

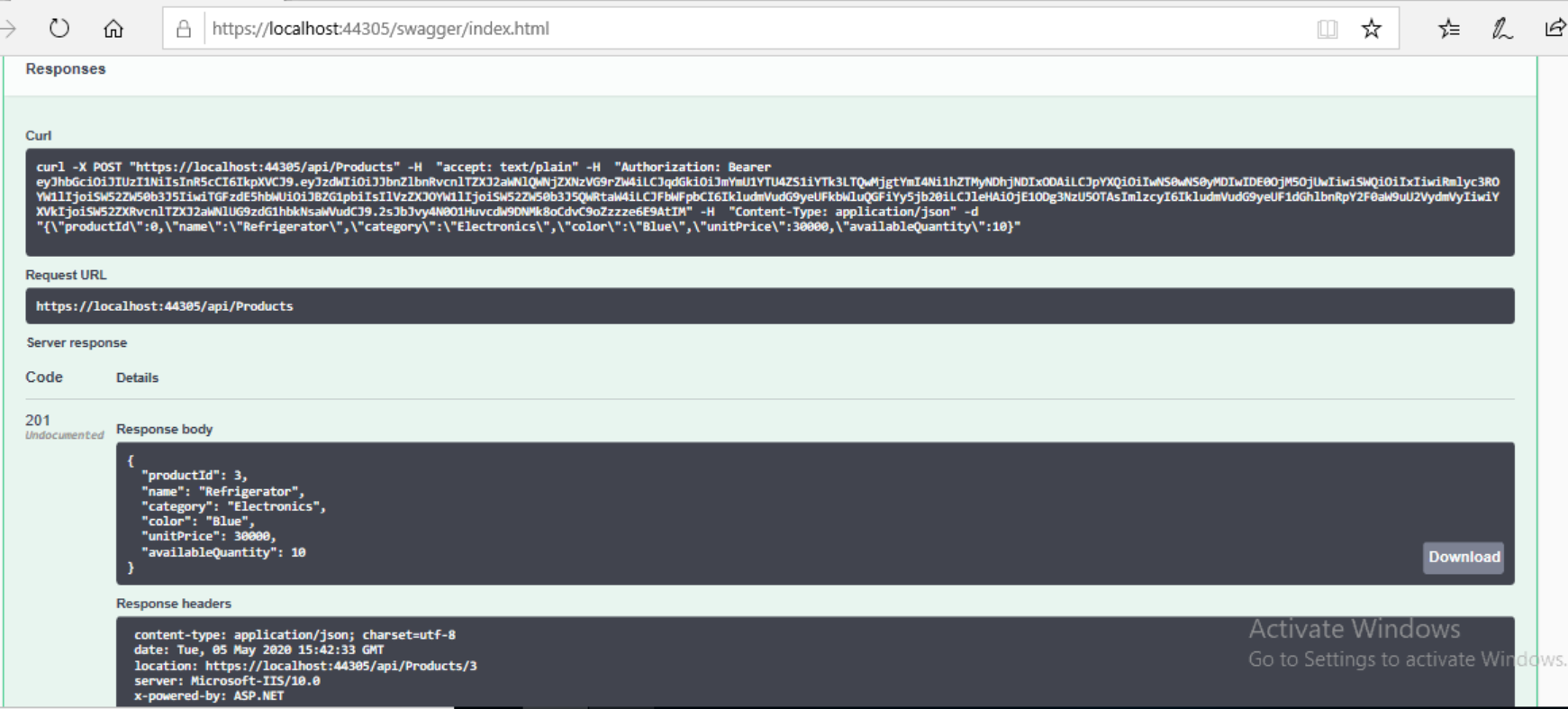


It will show you as ‘Authorized’.

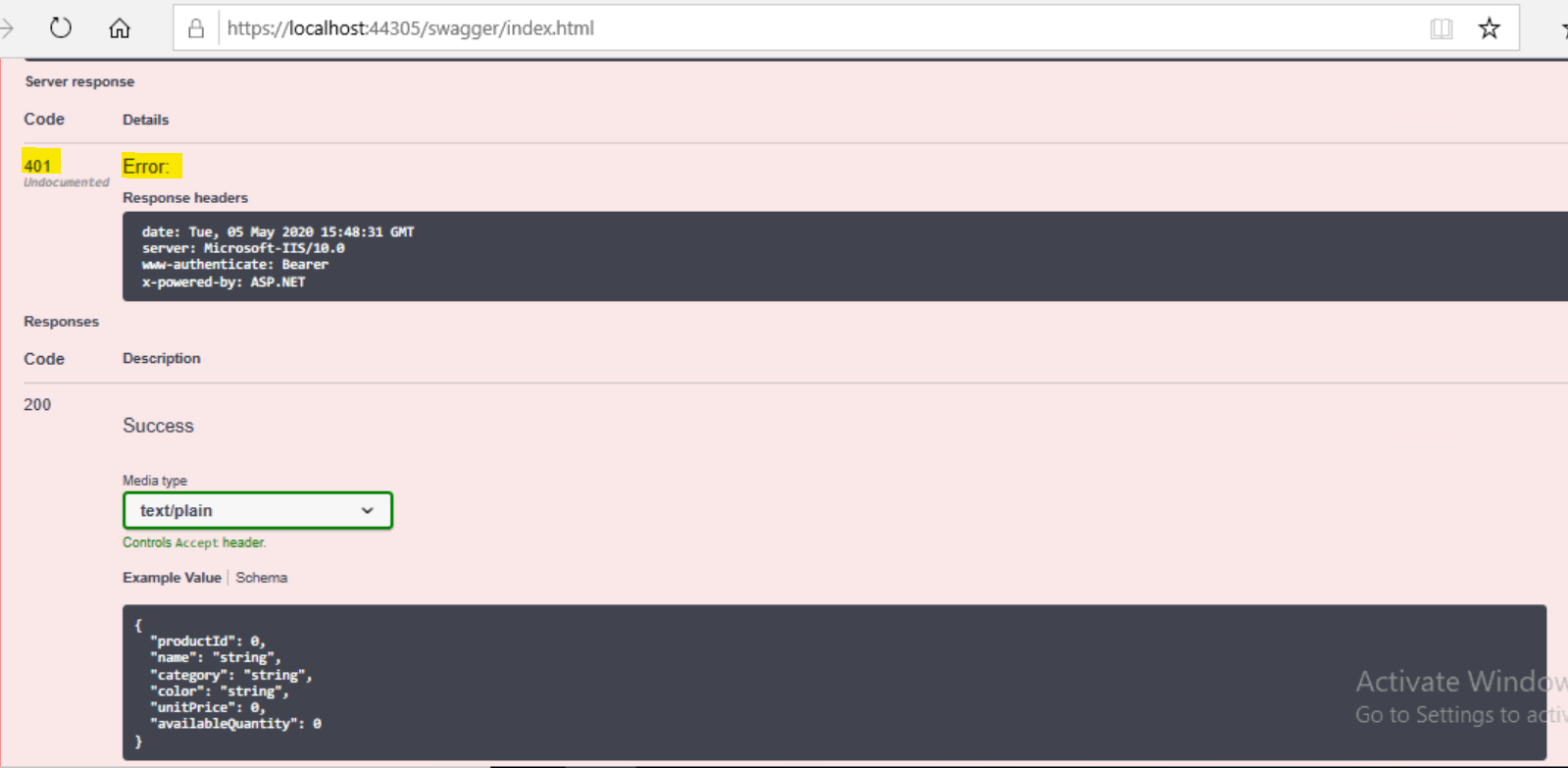


‘**POST’ request is now working successfully after successful authetication:**

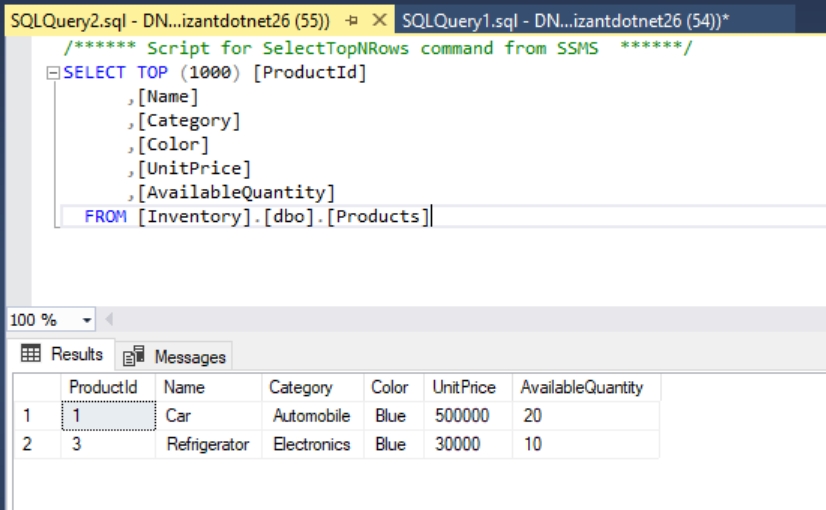




**Without Authentication it will show a 401 Unauthorized error:**



**Data we have executed for ‘POST’ gets inserted into Database:**



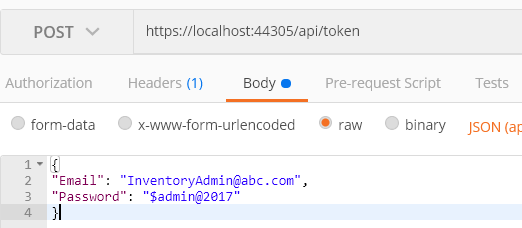
**Used Postman for API Call: Enter the authentication**

Follow these steps to check the token endpoint using Postman:

**Step 1**: Enter this endpoint <https://localhost:44305/api/token>.

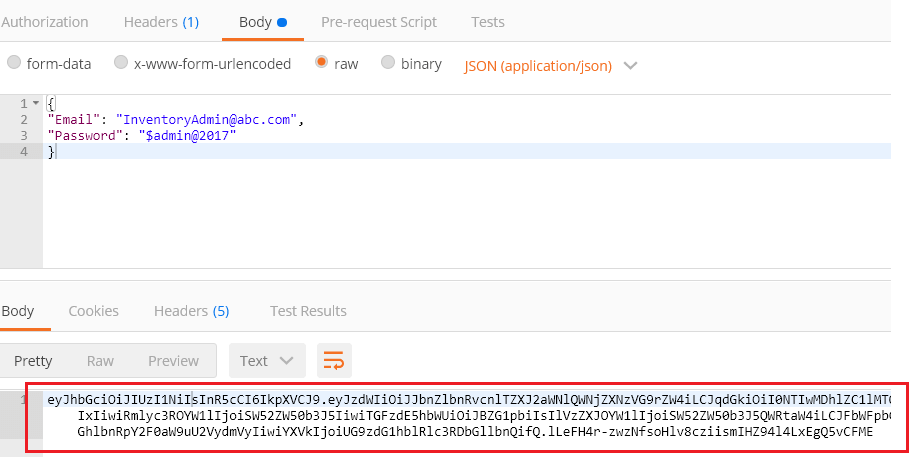
**Step 2**: Choose the **POST** method and set the header to **‘Content-Type’: ‘application/json’**.

**Step 3**: Under Body > Raw, choose type **JSON (application/javascript)** and paste the product details.



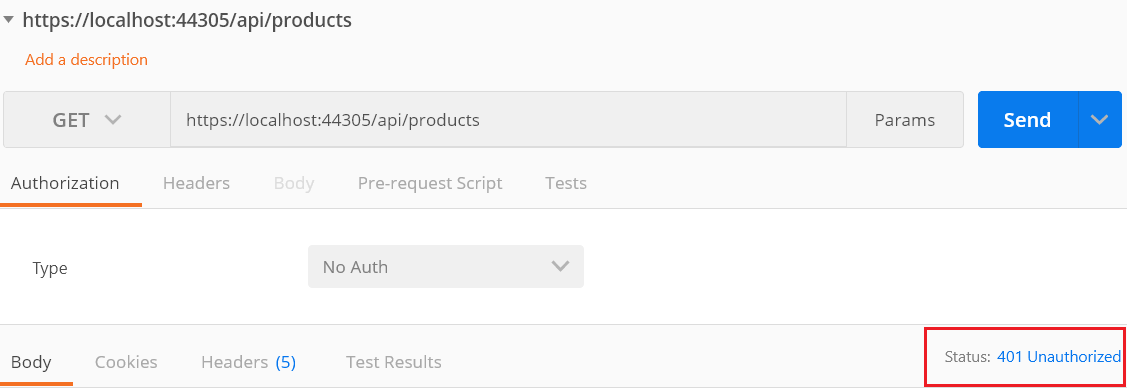
Choosing type JSON (application or javascript) and paste the product details to check the token end point

By clicking **Send**, user credentials will be checked, and a token will be generated and returned. See the highlighted body section.



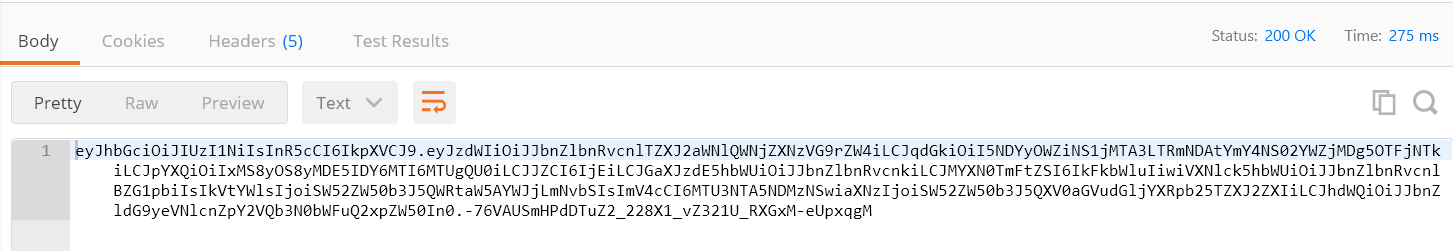
**Step 1:** In Postman, enter this endpoint: <https://localhost:44305/api/products>.

**Step 2:** Choose the **GET** method and then click **Send**. Now, you can see the Status code is **401 Unauthorized.**

Status code 401 Unauthorized

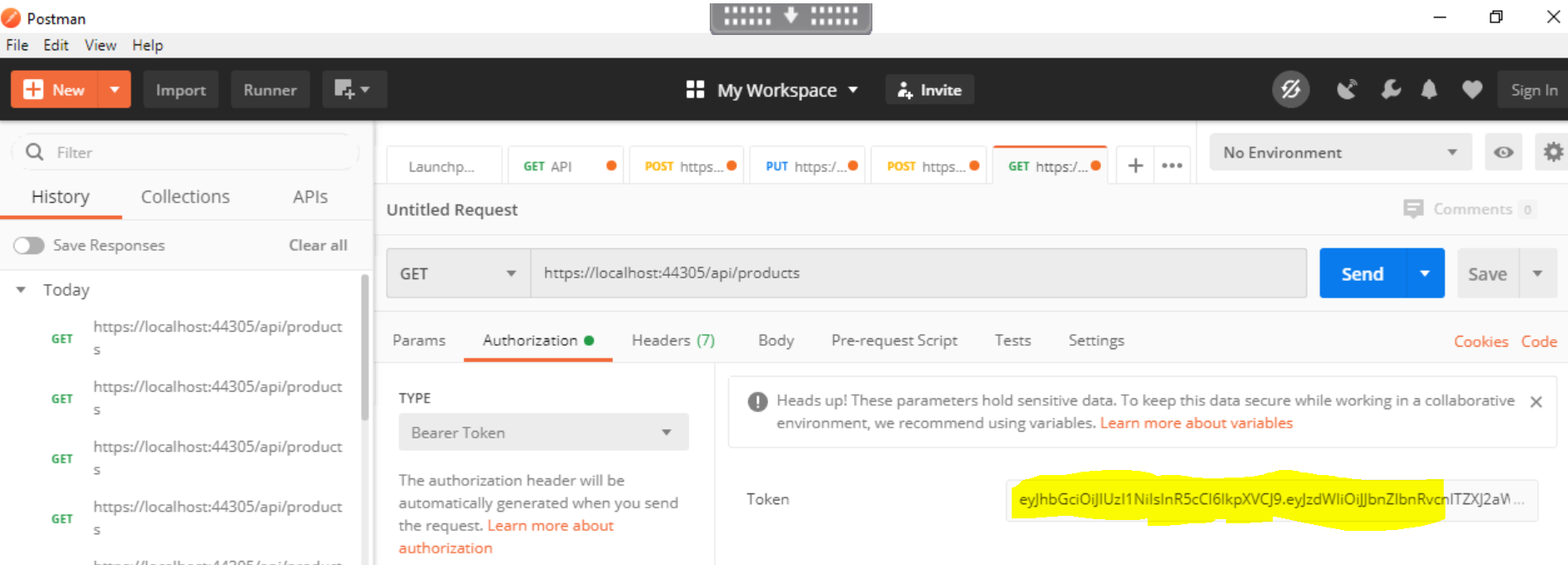
The anonymous access has been blocked and the APIs have been secured. Now, we will see how to access the APIs using the token.

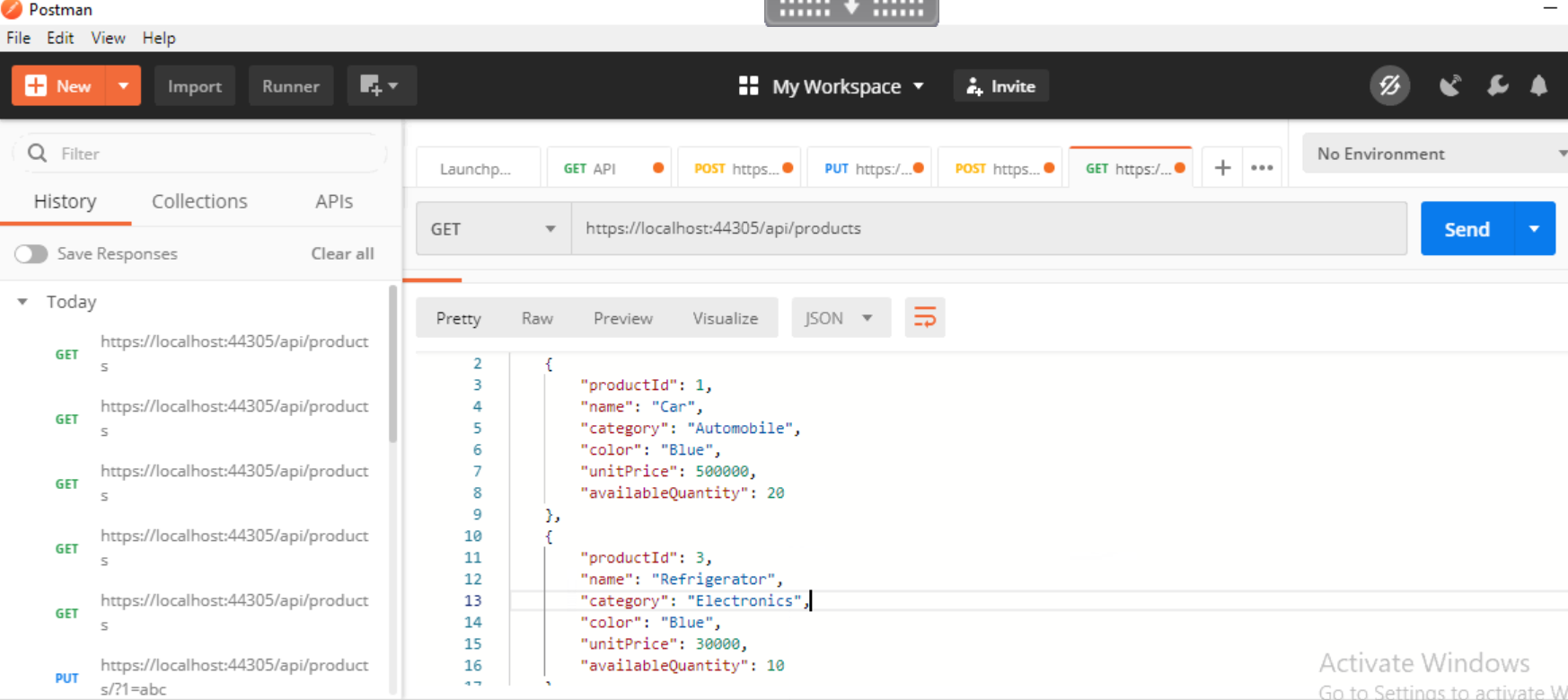
**Step 3:**Copy the token that was created in earlier steps.

Copying the token created earlier

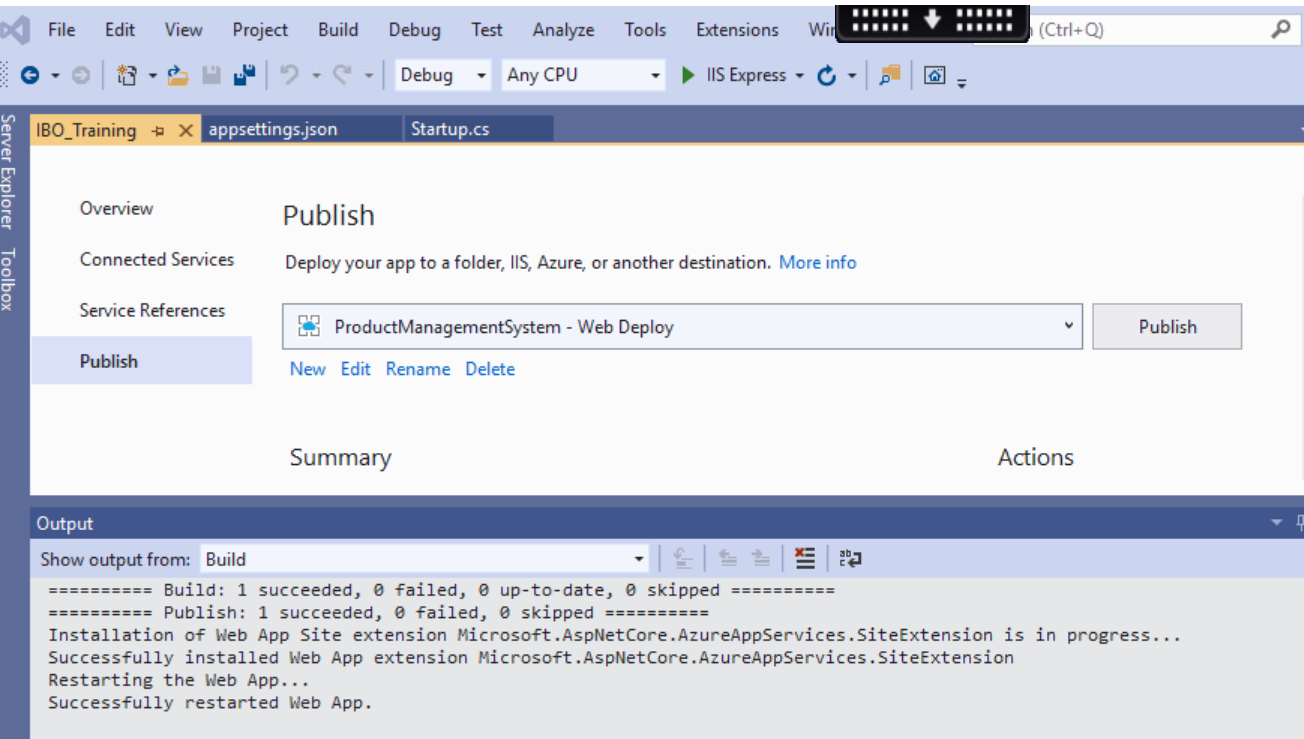
**Step 4:**Prepend text**Bearer**with this token, and finally you will have a token as follows.

**Step 5:**Now, come back to the product list. Under the**Authorization**header, paste the previous token in the **Value** field, and then click **Send.**Now you can see the products from our API.

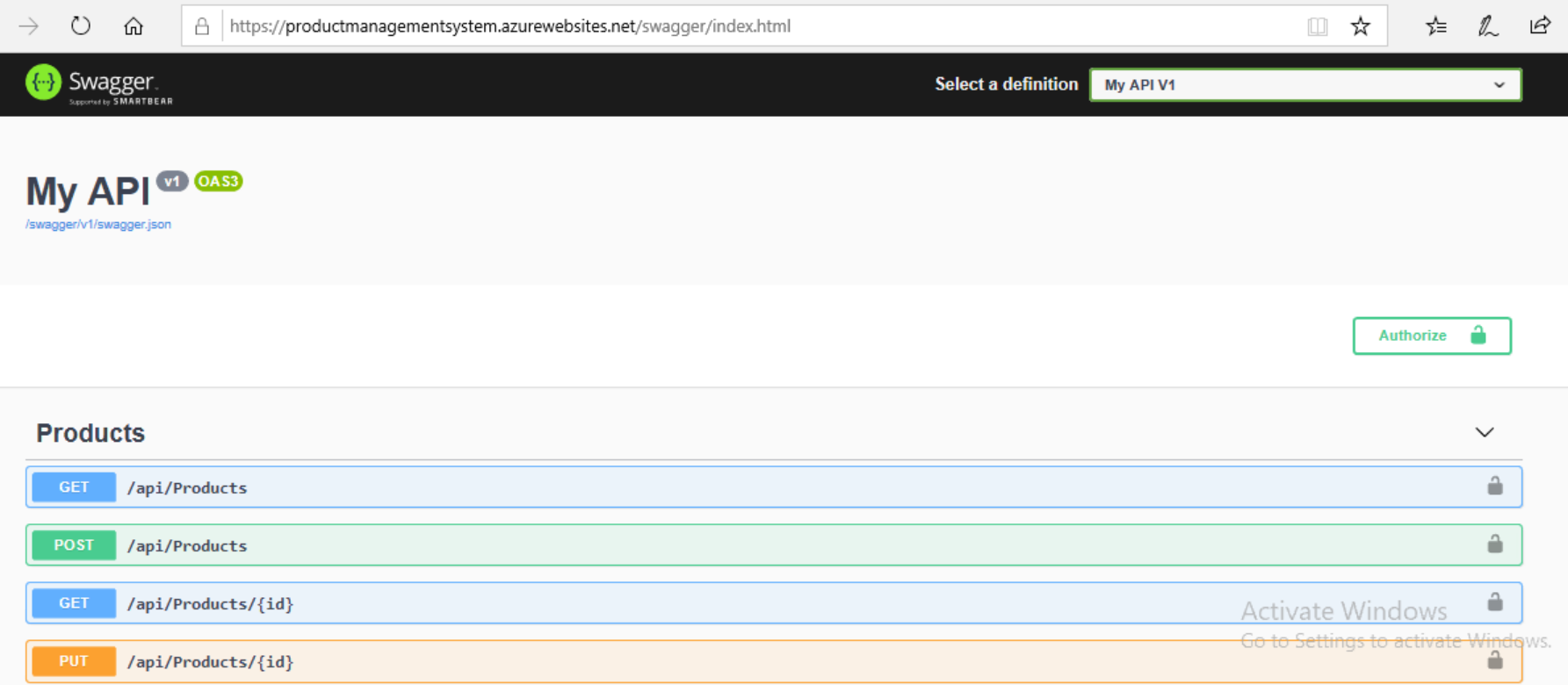


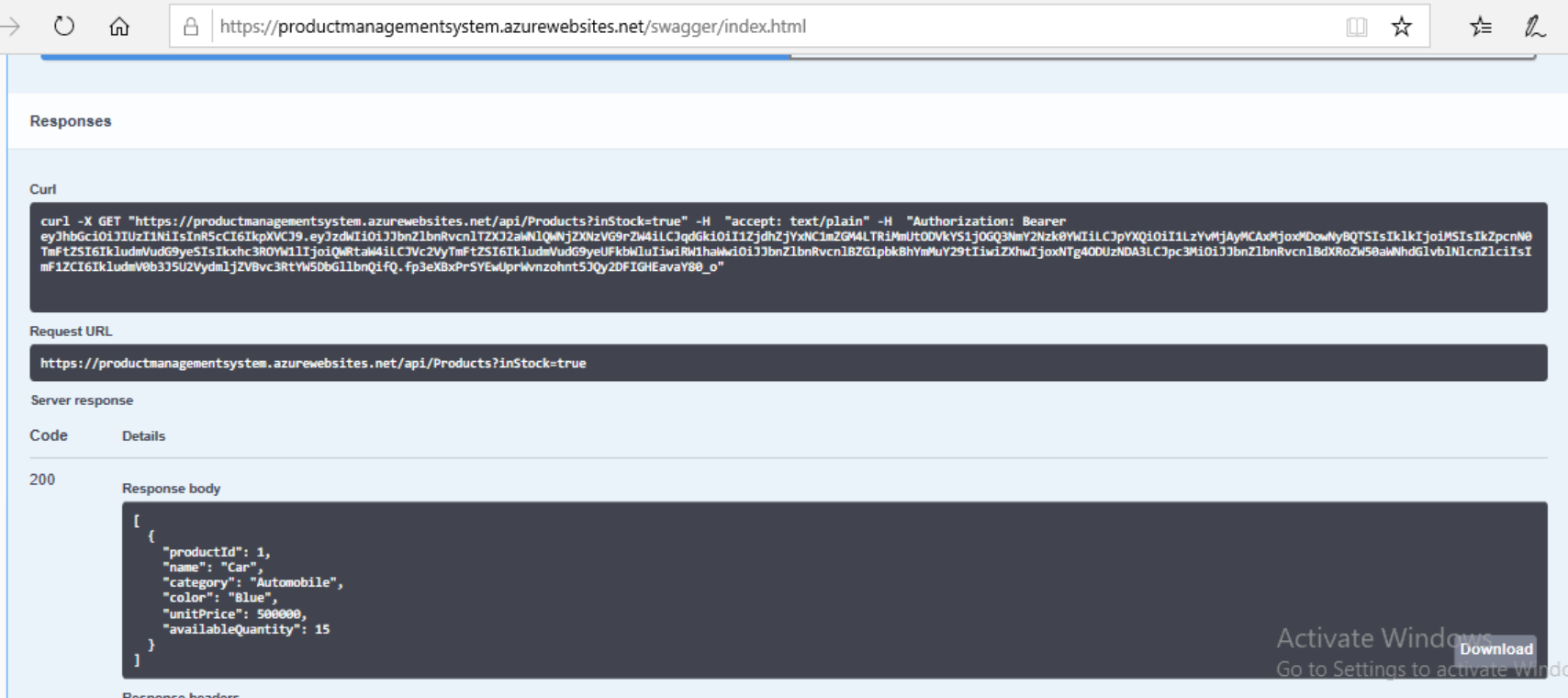


**Published and Hosted the Application in Azure Web Apps:**

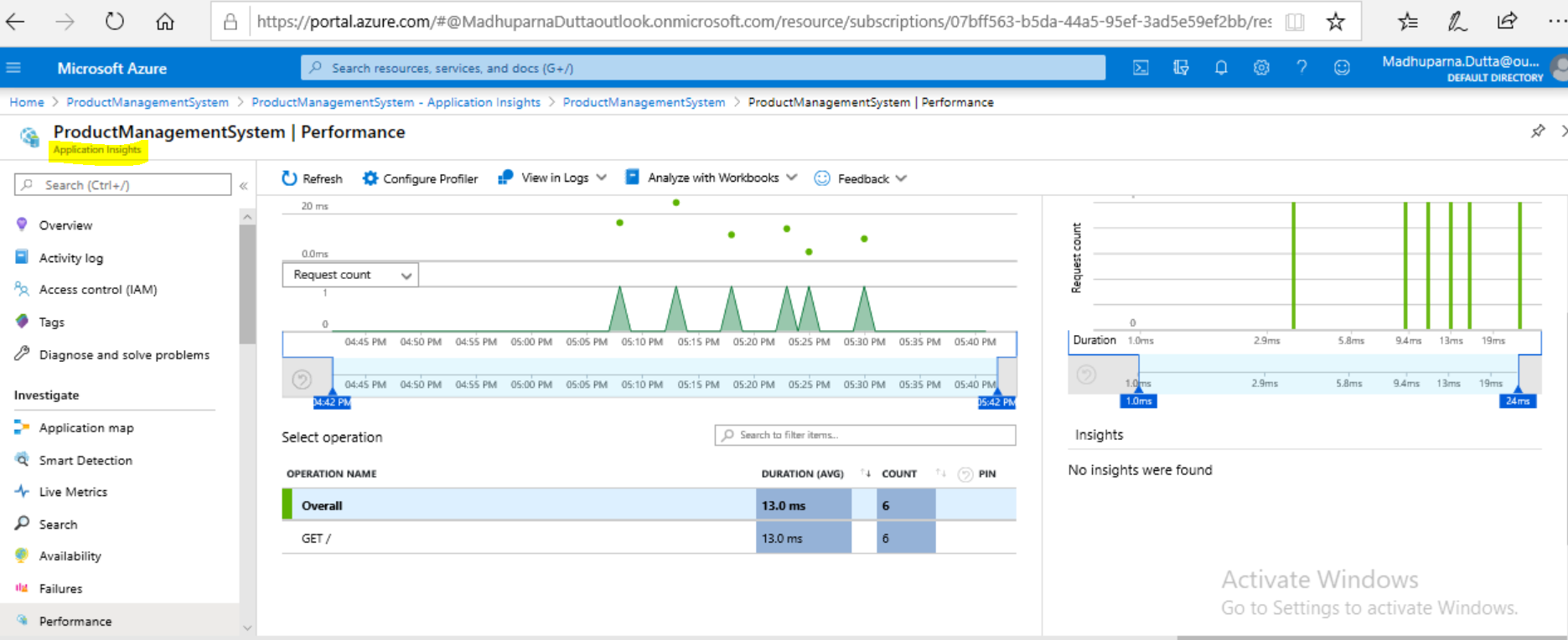


**Application running in Azure Web App:**

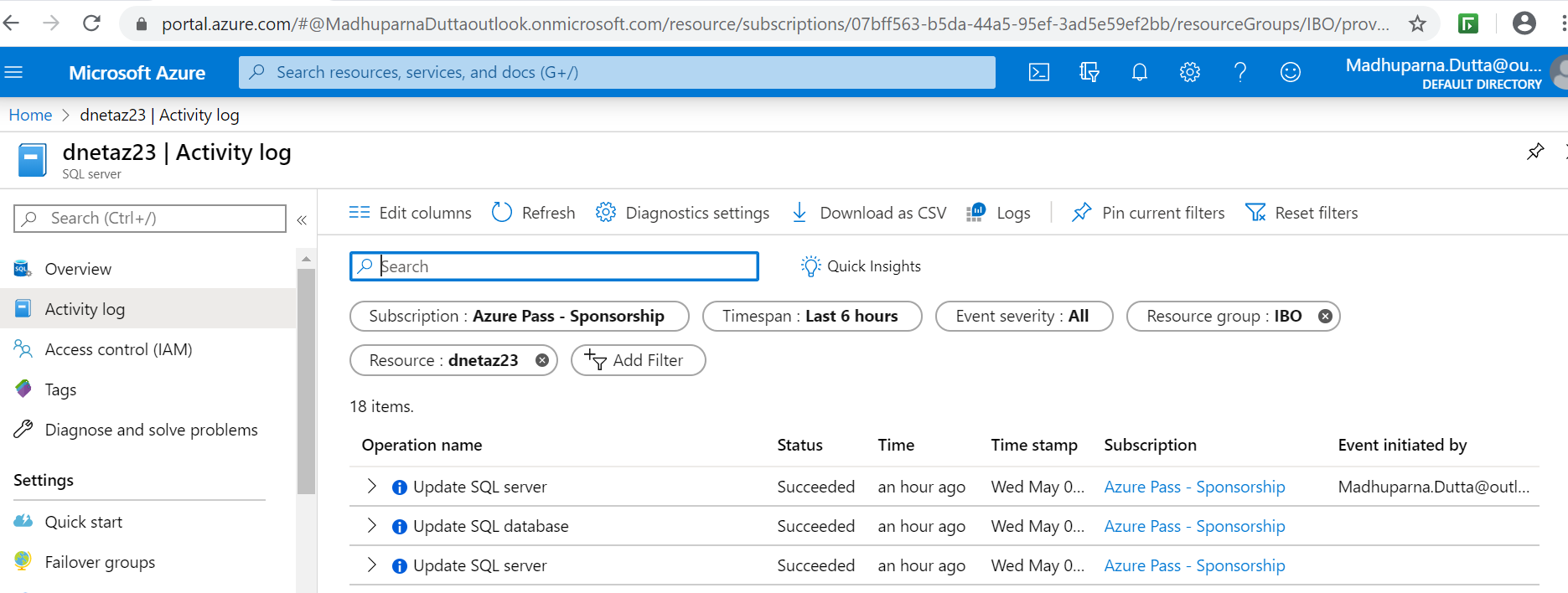




**Integrated Azure App Insights for the API service hosted:**



**Established connectivity to Azure SQL database:**



**Azure SQL database able to fetch data from the web application and vice-versa:**

