**CMPE 207 : Network Programming and Applications**

**HW#5 Microservice**

**Title**: Microservice using Lambda and API Gateway

**Github:** <https://github.com/Shashank-Bheemanapalli/Network-Programming-Projects.git>

**Group:** Network Ninjas

**Members:** Shashank Bheemanapalli

Vidhatri Milind Joshi

Sahiti Shankar Varada

**Introduction:**

In this example, we have created a simple API using Amazon API Gateway. An Amazon API Gateway is a collection of resources and methods. For this example, we create one resource (pollution-stats) and define four methods (POST, GET, DELETE, PUT) on it. The method is backed by a Lambda function (lambda\_pollution). That is, when you call the API through an HTTPS endpoint, Amazon API Gateway invokes the Lambda function. Here we make use of the Postman to call these APIs.

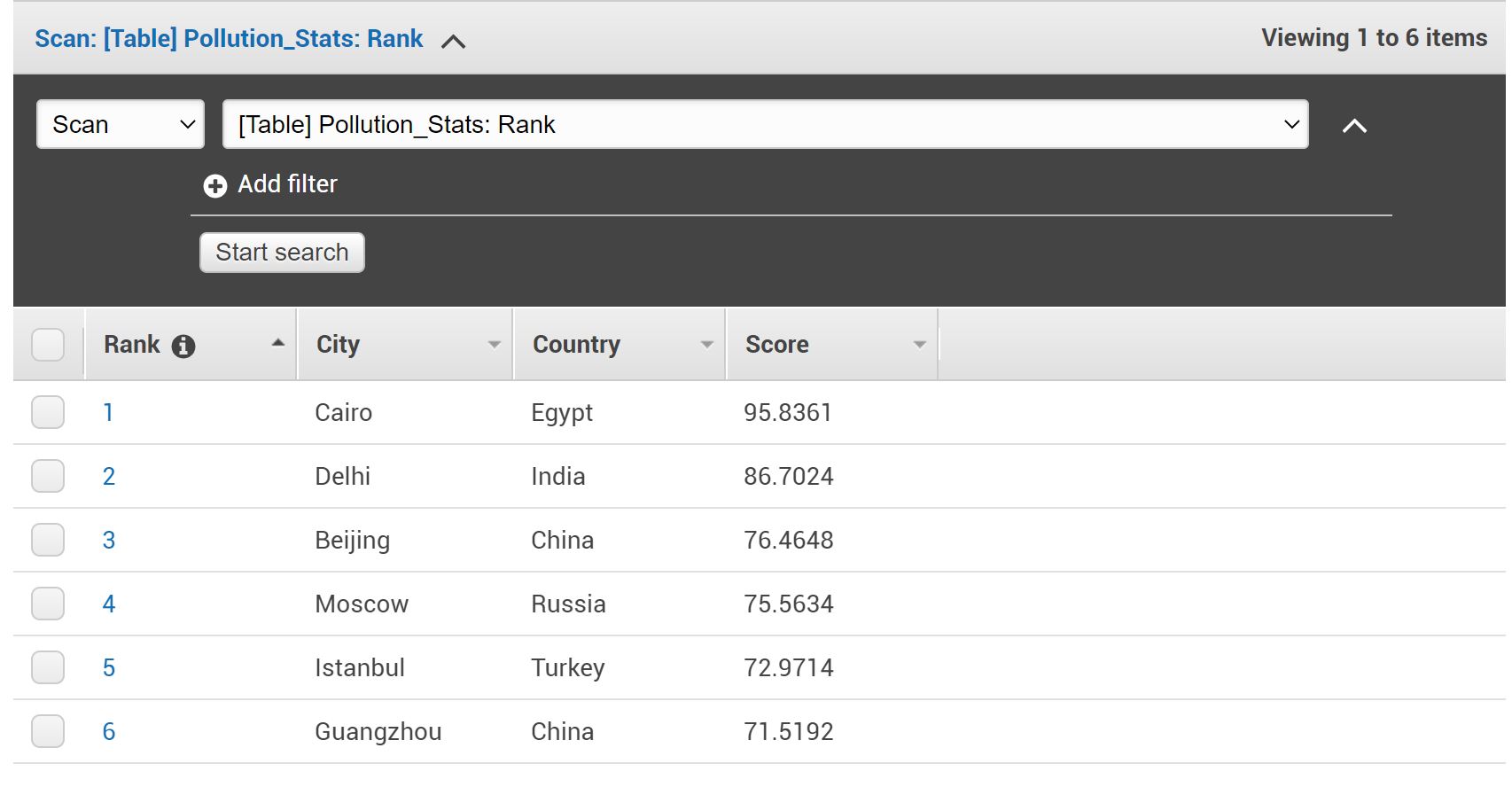
*Our API gateway can process 4 simple HTTP requests:*

**POST**: Create an item

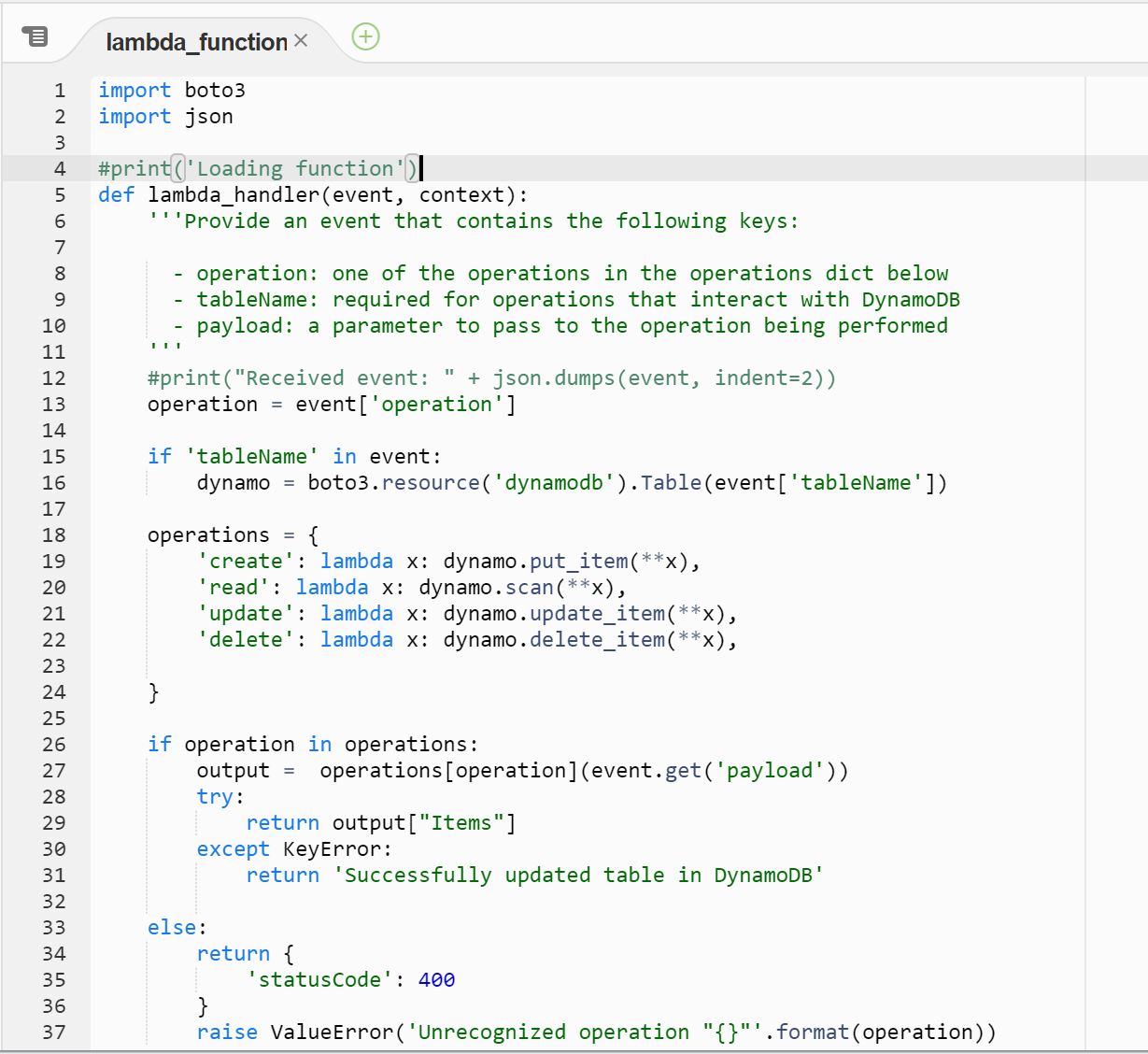
**PUT**: Update an item

**GET**: scan table and return all items

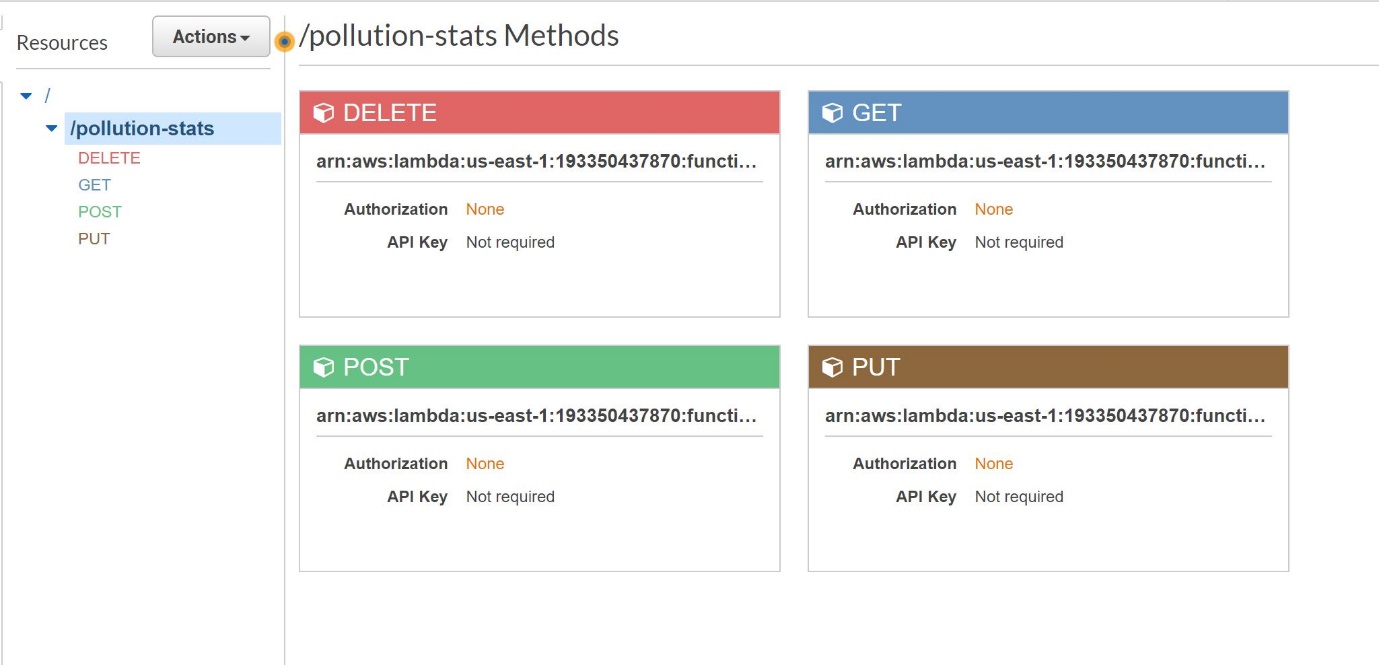
**DELETE**: delete an item

**Step1:** Table named ‘pollution-stats’ is created in DynamoDB and updated the table with records of data to perform actions on. 

**Step2:** Python function is written in Lambda to perform the 4 actions (create, update, get, delete) on DynamoDB. Here we used the ‘boto3’ module to communicate with DyanamoDB.



**Step3:** An API is created in API Gateway along with the 4 required methods and choose the Integration type: Lambda Function and attach it to the lambda function previously created.



**Step4:** ‘Postman’ is used to perform these four actions on DynamoDB which is triggered by Lambda functions which are attached to the API Gateway.

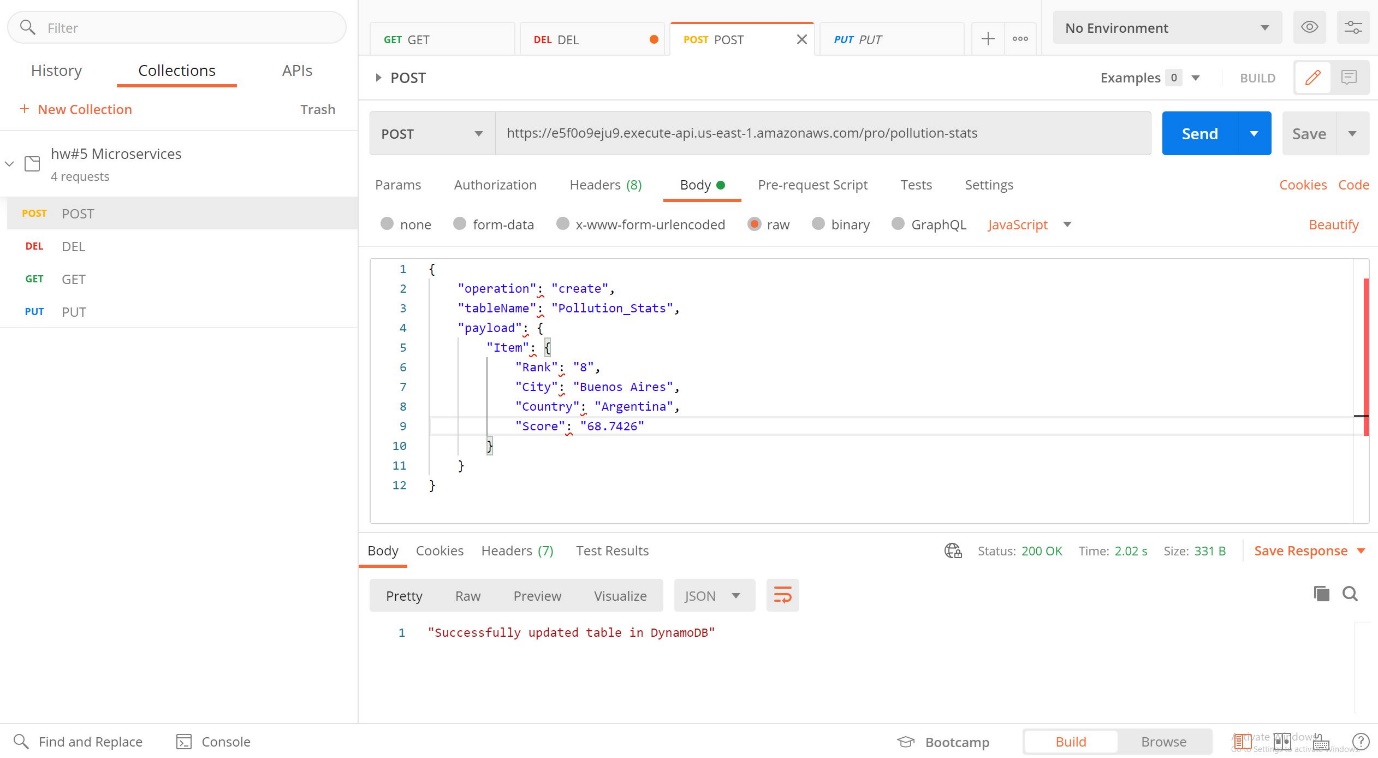


Fig. POST method

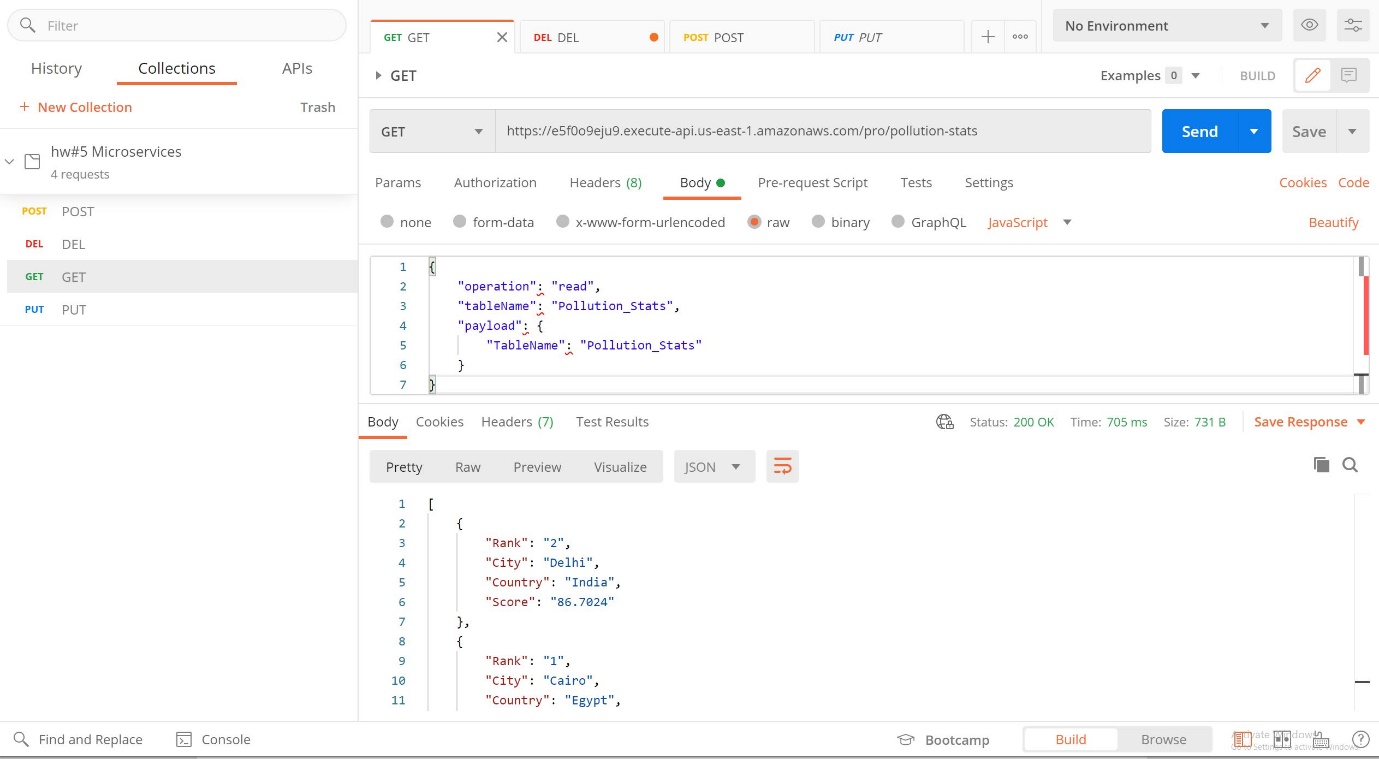


Fig. GET Method

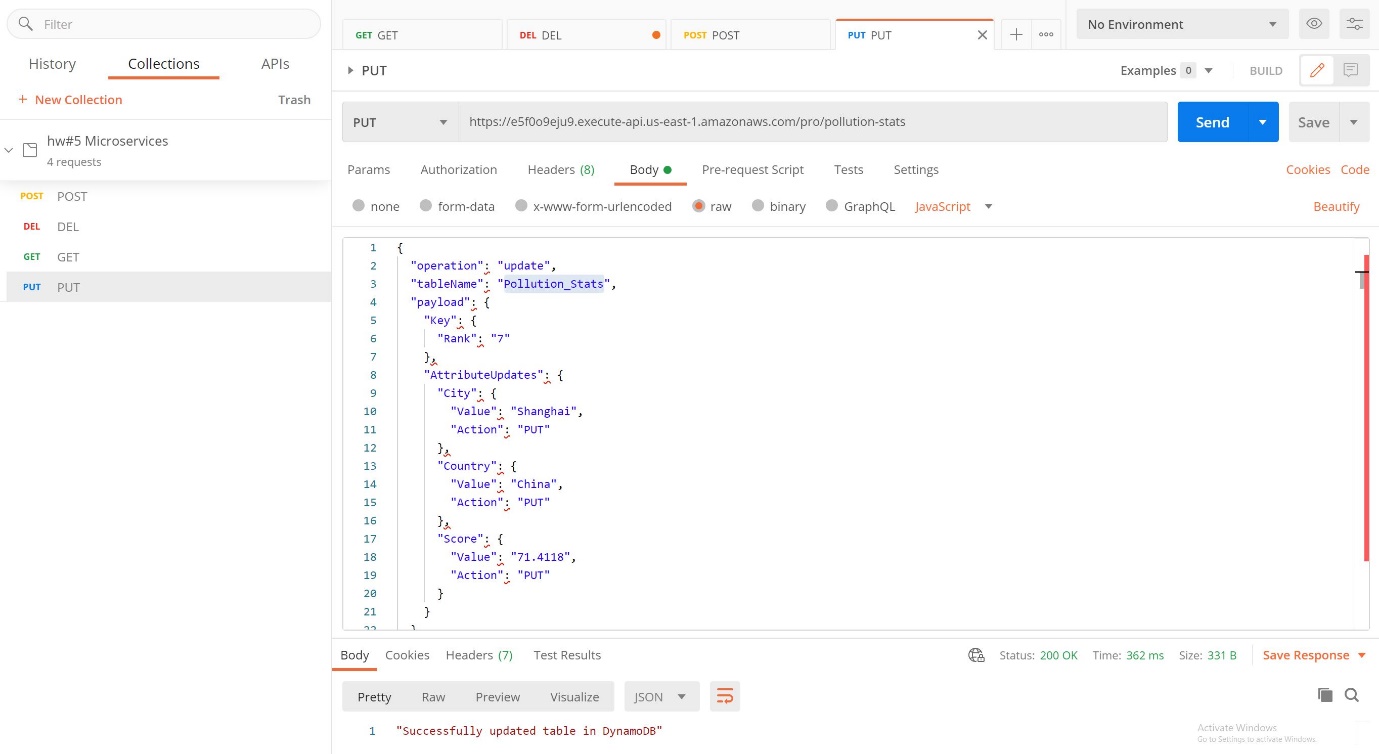


Fig. PUT Method

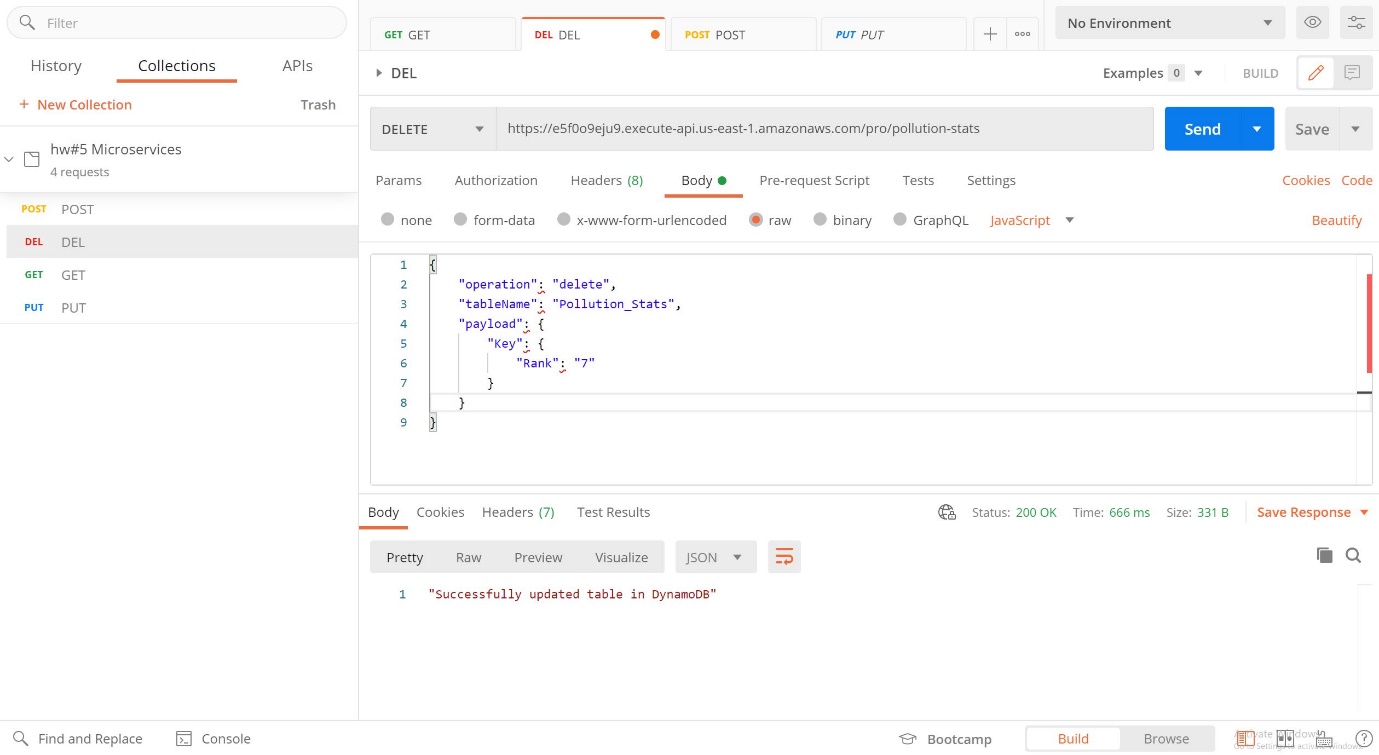


Fig. DELETE Method

**Conclusion:** All the 4 methods are seen in action using the Response Body of Postman and also the Updated DynamoDB table.