**Amazon Lambda and API Gateway**

Benjamin M. Brandhorst

University of Maryland Global Campus

SDEV 400 7980 Secure Programming in the Cloud

Doctor James Robertson

April 22nd, 2020

**Rest API with Lamda backend and DynamoDB**

The easiest way for me to explain how this REST API works is to start backwards which is the order that everything was written. Figure 1 shows the DynamoDB table I created which holds 75 items. These items detail the results of the last five games played by five National Hockey League (NHL) teams, five Major League Baseball (MLB) teams, and five National Basketball Association (NBA) teams.

**A screenshot of a cell phone

Description automatically generated**

Figure 1 – Sports DynamoDB table

Figure 2 shows the Python 3.7 Lambda I wrote to query the DynamoDB. In hindsight, there could have been some more modularity if I could have worked out how to create a function to format the query returns better. I had trouble passing a str and the contents of a variable as function arguments, which is why much of it is written within the queryDB() function. I could have also future proofed better by working in a query return showing what teams are in the database under a specific sport when a user searches for a team that isn’t present in the database. Instead, there are hardcoded messages in their place. If I had more time to play around, I would definitely implement that better. As it is, I spent 80 percent of my time trying to work out how to best format the resulting query returns. In the end, I went with using a json.dumps to make a single string and then using a regex I wrote to strip out extra characters.

A screenshot of a social media post

Description automatically generated

**Figure 2 – Python 3.7 Lambda used to query DynamoDB**

Figures 3 and 4 show the test configuration and the results of testing the Lambda.

A screenshot of a cell phone

Description automatically generated

**Figure 3 – Lambda test configurationA screenshot of a cell phone

Description automatically generated**

**Figure 4 – Lambda test results**

The actual REST API was created following the instructions in the “Using AWS API Gateway” document included in the week 6 instruction. It used a GET method as shown in Figure 5. Figure 6 shows the GET Integration Request with an application/json mapping template.A screenshot of a social media post

Description automatically generated

**Figure 5 – REST API GET method overview**

**A screenshot of a cell phone

Description automatically generated**

**Figure 6 – Get Integration Request**

Finally comes the actual API deployment. In the interest of preventing someone from flooding the API with requests and quickly draining the remaining AWS Educate account funds for this class, I changed the Default Method Throttling from 10,000 requests per second with a burst of 5,000 to 100 requests per second with a burst of 50 as seen in Figure 7. A screenshot of a cell phone

Description automatically generated

**Figure 7 – API Deployment settings.**

From there, I used the Invoke URL provided and added ?Sport=Baseball&Team=Reds to the end. The results can be seen in Figure 8.

**A screenshot of a cell phone

Description automatically generated**

**Figure 8 – API use**

The API URL is: <https://gf9wm3v2ik.execute-api.us-east-1.amazonaws.com/Submission>

References

How to write regular expressions? (2016, July 8). Retrieved from https://www.geeksforgeeks.org/write-regular-expressions/

What is the difference between json.dumps and json.load? (2016, October). Retrieved from https://stackoverflow.com/questions/32911336/what-is-the-difference-between-json-dumps-and-json-load