Serverless Development 101

**HOP09 – Build an app with REST API - UPDATE**

5/16/2020 Created by Apiwat Chuaphan

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

  

**Learning Outcomes**

* Learn what REST API is and how it works
* Learn basic CRUD
* Learn how items store in DynamoDB
* Build an app with AWS Lambda, API Gateway, DynamoDB, REST API

**Build the U from CRUD paradigm.**

1. Open the VSCode and open the “**myproject**” project folder that we developed a web application using serverless computing in the previous module.
2. Go to **todos** folder and get inside **functions** folder.
3. Create a new file named **update.js** to update our task in the database in case we make a mistake, or we want to mark it as done. Then update the file with the content [here](https://bit.ly/3e2BEzX)

A screenshot of a cell phone

Description automatically generated

**update(params, callback)**

* Edits an existing item's attributes or adds a new item to the table if it does not already exist.

**# is a syntax for expression attribute**

* An expression attribute name is a placeholder that you use in an Amazon DynamoDB expression as an alternative to an actual attribute name. An expression attribute name must begin with a pound sign (#), and be followed by one or more alphanumeric characters.

The **ReturnValues** parameter instructs DynamoDB to return only the updated attributes

\*\*\*Read more of these expression [here](https://www.dynamodbguide.com/expression-basics/)

1. After we added new function, we need to let Lambda know our new function by updating **serverless.yml** as follow

A screenshot of a cell phone

Description automatically generated

1. Run **sls deploy -v** I terminal to deploy our function to Lambda. This time we have to deploy the entire process because we updated **serverless.yml** (*Make sure you’re in the* ***“todos” f****older)*

A screenshot of a cell phone

Description automatically generated

See that you have new endpoint “PUT” with the {id} at the end which means the specific item you want to make changes.

1. Verify that new function has been created and deployed to Lambda page.

A screenshot of a cell phone

Description automatically generated

1. We will test the function if it does what it’s supposed to, which is updating the data in the database. First, we need to get the id of the item we want to update. There are several ways to do this step, but I will use **curl** command to get the id by running **curl <your-api-url>**

After you ger your id of the item you want to update, run this command in terminal

>>> **curl -X PUT -d** '{"text":"I am done with this HOP", "isDone":true}' **<your-api-url>/id**

1. Then, check if the update request is successfully complete. Run **curl <your-api-url>**

You will see the output response has been changed to the new text that you just updated.



1. To confirm that update has been update the data in the database, go to DynamoDB page.

A screenshot of a social media post

Description automatically generated

Then click the id, you will notice that the has been changed.

A screenshot of a social media post

Description automatically generated

**Push your work to GitHub**

Open the terminal from the VSCode by hit the control + ~ key and type the following command:

Run the following commands to push your work to the GitHub repository:

>>> git add .

>>> git commit -m “Submission for Module 9”

>>> git push origin YOUR\_BRANCH\_NAME

**Note**: you should change the YOUR\_BRANCH\_NAME to your own branch name. It should be firstname-lastname (e.g. maria-gracia).

If you cannot remember, run the command “git status” to check