Assignment 7 – Lambda

Serverless is awesome and consts $0. You don't have to delete them.

Check out the "Be a better dev" channel. It is a great channel on serverless services that this course covers in the second half.

Task 0. Practice version, alias, and weighted (canary) deployment

Task 1. Create a Lambda, "CourseLambda"

- Crete a IAM policy that allow CRUD operations on the table.
- Create a DynamoDB table, "CourseTable".
    a. courseCode -> Partition key
    b. teacherName -> Sort key
- Implement the PutItem in the Lambda.

You will implement the rest of the CRUD operations in the coming days. You can do your own research on the DynamoDB CRUD APIs by referring

- Official documentation
- SDK
- Internet blogs
- ChatGPT
- Sample code on Sakai (lambda-helper.mjs and lambda-index.mjs in Slides folder).

I highly recommend you to start doing the rest of CRUD operations now.

Sample code for creating an item. The rest of the code is on Sakai

```
import {
    DynamoDBClient,
    PutItemCommand,
} from "@aws-sdk/client-dynamodb";

const dynamodb = new DynamoDBClient({
    apiVersion: "2012-08-10"
});

export const handler = async(event) => {
    const saveParameters = {
        TableName: 'prep',
        Item: {
            "courseCode": {
                S: 'CS516'
            },
            "courseName": {
                S: 'CC'
            },
            "teacherName": {
                S: 'Uno'
            }
        }
    };

    const command = new PutItemCommand(saveParameters);
    await dynamodb.send(command);

    const response = {
        statusCode: 200,
        body: 'success'
    };

    return response;
};
```