Server less Web App using DynamoDB

**High level process**

1. Create IAM roles: Theses roles will be used by lambda functions to gain access to our DynamoDB table items
2. Create DynamoDB table: This table is hosting our items. Web application will be requesting this information.
3. Create Lambda functions: These functions will run and query the items in DynamoDb and retrieve them for us.
4. API Gateway: This is an API that will accept HTTP requests from the Internet and trigger the function when it receives the request.
5. HTML page: This is the client front end that will be used to send a request the API gateway

Step 1 Create IAM Roles:

1. Create a policy and call it SuperDynamoDBScanPolicy. The Json code to create the policy is the file named Superheropolicies.txt
2. Create a policy and call it SuperDynamoDBQueryPolicy. The Json code to create the policy is the file named Superheropolicies.txt

Step2 Create DynamoDB table:

1. Create a table and name it “SuperMission” with a primary key of “SuperHero”.
2. Add items using the text input option. Key value pairs are saved in file named “Super heros dynamoDB items.txt”

Step3 Create Lambda functions:

1. Create Lambda function to scan for all items in DynamoDB table. The Lambda code is stored in a file named “Lambda scan and query code.txt”.
2. Create a function named “getheroeslist” using Nodejs 12.x and IAM “SuperDynamoDBScanRole”. Copy and paste the code inside the function.
3. Create a test and run it. You should see the function retrieve items from DynamoDB
4. Create a function named “getmissiondetails” using Nodejs 12.x and IAM “SuperDynamoDBQueryRole”. Copy and paste the code inside the function.
5. Create a test and run it. You should see the function retrieve items from DynamoDB. For this test to run successfully you must place the following as input in the test.

{ "superhero": "Batman" }

Step3 Create API Gateway:

1. Open API Gateway Service, select API’s and click “**Create API**”
2. Select “REST API” and click “Build”
3. Under new API, Provide and new API name of “SuperheroesMission”
4. Under “/”, open actions menu and click “**Create Resource**” and name it “getheroeslist”
5. Under “/”, open actions menu and click “**Create Resource**” and name it “getmissiondetails”
6. Select the “getheroeslist” resource and **create method**. From drop down select “Post”, and click the check mark on the right side. Place the name of the Lambda Function as “getheroeslist” and save.
7. Select the “getmissiondetails” resource and **create method**. From drop down select “Post”, and click the check mark on the right side. Place the name of the Lambda Function as “getmissiondetails” and save.
8. Select the resource named “getheroeslist” and **enable CORS** from the Action menu.
9. Select the resource named “getmissiondetails” and **enable CORS** from the Action menu.
10. Select “/”, and in the actions menu select “**Deploy API**”. For deployment stage select “New Stage” and name it Demo1 and hit deploy.
11. Select Stages form the left, click SDK Generation tab, select javascript platform an click “**Generate SDK”.**
12. This will download a zip file on our computer that we need to extract.
13. Place our HTML file inside the folder named “apigateway-js-sdk” and run the file. (Client Code saved in index.html.txt)