

Lambda Demo

Employee-Data CRUD Operations

1. Download and install Postman API client from below link -

<https://learning.postman.com/docs/getting-started/installation-and-updates/>

2. Download application code-base from Github URL -

https://github.com/Ritesh-Goyal/Lambda_CRUD

3. Case-Study

You need to create a backend API system for employee data management. We will perform CRUD operation and below flows should be covered -

- New Employee record insertion.
- Update existing employee record based on employeeid
- Get specific employee record based on employeeid
- Delete employee record based on employeeid
- Fetch all the employee record
- A monitoring API to check the health

Table items should be - employeeid (primary-key), empname, domain, projectname, location.

4. Demo: AWS services

For this case study, you will use below AWS services -

- DynamoDB - We will maintain all the employee data in DynamoDB table
- AWS Lambda - One Lambda function supporting 6 API's
- IAM roles - For access permission in between API Gateway, DynamoDB and Lambda.
- AWS API Gateway - We will create 6 REST API's.
- Cloudwatch - The API request & response will be tracked using Cloudwatch

5. Creating Services:

- Create DynamoDB Table -
 - Table name: employee-data
 - Partition (Primary) key: employeeid
- Create Lambda Function -
 - name: employee-data
 - Runtime: Node.js 16.x

- API Gateway -
 - Create resources and HTTP methods:
 - /
 - complete
 - GET
 - record
 - GET
 - POST
 - PATCH
 - DELETE
 - status
 - GET

6. Sample API Request Data

- Check Health of the API:

GET: <https://kchw8wbdt4.execute-api.us-east-1.amazonaws.com/devops/status>

- Employee Records

GET: <https://kchw8wbdt4.execute-api.us-east-1.amazonaws.com/devops/record>

POST: <https://kchw8wbdt4.execute-api.us-east-1.amazonaws.com/devops/record>

Body Raw Data:

```
{
  "employeeId": "1005",
  "empname": "Ritesh Goyal",
  "domain": "DevOps",
  "projectname": "Cloud Training",
  "location": "Pune"
}
```

PATCH:

<https://kchw8wbdt4.execute-api.us-east-1.amazonaws.com/devops/record>

Body Raw Data:

```
{
  "employeeId" : "1005",
  "updateKey" : "projectname",
  "updateValue" : "DevOps Training"
}
```

DELETE:

<https://kchw8wbd4.execute-api.us-east-1.amazonaws.com/devops/record>

Body Raw Data:

```
{  
  "employeeId": "1005"  
}
```

- Get complete employee records

GET:

<https://kchw8wbd4.execute-api.us-east-1.amazonaws.com/devops/complete>

Name: Ritesh Goyal

Contact: 9960930111

Email-ids: ritesh.devopstrainer@gmail.com