AWS-Serverless-typescript-dynamodb

This is a sample project using serverless to deploy lambda to AWS to interact with DynamoDb

Installation

Clone the git repo

Use the node package manager to install the dependencies.

```
npm i
```

Usage

Create AWS user with the appropriate permissions to create the cloud formation stack

To deploy the lambda to AWS.

```
npm run deploy
```

Once the lambda is deployed. Copy the API key to consume the API. The API key needs to be added to the headers for authorization as X-API-KEY while making a call to the URL in the service information.

```
Service Information
service: employees
stage: dev
region: eu-west-2
stack: employees-dev
resources: 22
api keys:
  first_dev_key: dhq0IQ5FfO3UwarotoCsa8p962uJJthK3IYb0VA2
  second_dev_key: gywXlevQrY4MI7MrcSZ8e2Mm6cDEKCjf3bLYT3DC
endpoints:
  POST - https://jq1h3y4qh3.execute-api.eu-west-2.amazonaws.com/dev/employee
functions:
  addEmployee: employees-dev-addEmployee
```

To remove the lambda from AWS

```
npm run remove
```

Other options

Running Tests

```
npm run test
```

Running Tests with coverage

```
npm run coverage
```

Request Payload

The lambda can handle both single or multiple objects in the request body

Single object

```
{
"id": "UID-123",
"name": "robert",
"vat-number": "DE123456",
"user-id": "gid:robert"
}
```

multiple objects

```
    [{
        "id": "UID-127",
        "name": "kartik",
        "vat-number": "DE3456",
        "user-id": "kartik"
    },
    {
        "id": "UID-124",
        "name": "john",
        "vat-number": "DE123456",
        "user-id": "gid:john"
    }]
```

Logs

Logs are available in cloud watch and x-ray

HTTP Response codes

```
200 success
400 bad request
500 internal server error
```

TODOs

```
Integration test to be added
Request JSON to be validated using a schema validator
Security to be increases using cognito
Third party logging libraries to be used to improve the logging
```