

REPORT AWS - NODE JS - DYNAMODB

For the project was used [NodeJs Serverless Stack](#). In site serverless view the installation instructions for serverless

Previous Steps

- Clone this repo
- Have a aws account(free) or create
- Install or have an aws cli view site for installation instructions:

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

- Install or have an node js environment view site for installation instructions:

<https://www.journaldev.com/7402/node-js-environment-setup-node-js-installation>

View Documentation

View the file `manual.pdf` located in documentation folder

Import Postman collection

Import the file `report.postman_collection.json` located in documentation folder in postman

Commands for installation

aws configure

Configure your aws credentials

npm install

Execute the command in the root folder of the project for install the node packages located in `package.json` file

sls deploy

Deploy serverless stack to AWS cloud.

npm test

Execute jests tests

```

callback
  ✓ get report (90ms)

pdf_report tests
  ✓ pdf_report is object
  ✓ pdf download mock

14 passing (357ms)

```

Configure postman endpoint

```

Service Information
service: serverless-aws-nodejs
stage: dev
region: us-east-1
stack: serverless-aws-nodejs-dev
resources: 48
api keys:
  None
endpoints:
  POST - https://ppoziu3l24.execute-api.us-east-1.amazonaws.com/dev/report
  PUT - https://ppoziu3l24.execute-api.us-east-1.amazonaws.com/dev/report/{id}
  GET - https://ppoziu3l24.execute-api.us-east-1.amazonaws.com/dev/report/{id}
  GET - https://ppoziu3l24.execute-api.us-east-1.amazonaws.com/dev/reports
  GET - https://ppoziu3l24.execute-api.us-east-1.amazonaws.com/dev/csv
  DELETE - https://ppoziu3l24.execute-api.us-east-1.amazonaws.com/dev/report/{id}
  GET - https://sts3dnagf9.execute-api.us-east-1.amazonaws.com/api/pdf_report
functions:
  create: serverless-aws-nodejs-dev-create
  edit: serverless-aws-nodejs-dev-edit
  get: serverless-aws-nodejs-dev-get
  all: serverless-aws-nodejs-dev-all
  csv: serverless-aws-nodejs-dev-csv
  pdf_report: serverless-aws-nodejs-dev-pdf_report
  delete: serverless-aws-nodejs-dev-delete
layers:
  None

Monitor APIs by route with the Serverless Dashboard: run "serverless"
mac@MacBook-Pro-de-MAC:~/serverless-dynamodb-nodejs-starter% 

```

- take the url part of the endpoint similar to the following: <https://xxxxx.execute-api.us-east-1.amazonaws.com>
- copy the endpoint string before /dev/report then set the variable awsReportEndPoint in postman
- copy the endpoint string before /pdf_report then set the variable awsPdfReportEndPoint in postman

Services endpoints

Consume in postman any of the following services:

- Create Report
- Update Report
- Delete Report by Id
- Get Report by Id
- Get All Reports

- Generate CSV Report
- Generate PDF Report

Run integration tests

Press right click on the collection in postman, in the menu that is displayed select run collection:

The screenshot shows the Postman interface with the 'report' collection selected in the left sidebar. A context menu is open over the collection name 'report'. The 'Run collection' option is highlighted with a red box. Other options in the menu include 'Edit', 'Share collection', 'Add request', 'Add folder', 'Monitor collection', 'Create a fork', 'Create Pull Request', 'Merge changes', 'View documentation', 'Rename', 'Duplicate', 'Export', 'Manage roles', 'Remove from workspace', and 'Delete'.

The screenshot shows the Postman interface with the 'Runner' tab selected. The 'report' collection is selected in the left sidebar. In the main area, a 'RUN ORDER' table lists various API requests. The 'Run report' button at the bottom right of the table is highlighted with a red box. The table includes columns for method, URL, and description. The 'Iterations' field is set to 1. Other settings include 'Delay' (0 ms), 'Data' (Select File), and checkboxes for 'Save responses', 'Keep variable values', 'Run collection without using stored cookies', and 'Save cookies after collection run'.

GET Generate PDF Report {{awsPdfReportEndPoint}}/api/pdf_report / Generate PDF Report 200 OK 182 ms 2.212 KB

- Pass Content-Type header is application/pdf
- Pass Status code is 200

GET Generate CSV Report {{awsReportEndPoint}}/dev/csv / Generate CSV Report 200 OK 278 ms 671 B

- Pass Status code is 200
- Pass Content-Type header is text/csv

POST Create Report {{awsReportEndPoint}}/dev/report / Create Report 200 OK 272 ms 713 B

- Pass Status code is 200
- Pass Validate json response data created

GET Get Report by Id {{awsReportEndPoint}}/dev/report/{{id}} / Get Report by Id 200 OK 181 ms 713 B

- Pass Status code is 200
- Pass Get report by id

DELETE Delete Report by Id {{awsReportEndPoint}}/dev/report/{{id}} / Delete Report by Id 200 OK 602 ms 487 B

- Pass Status code is 200

GET Get All Reports {{awsReportEndPoint}}/dev/reports / Get All Reports 200 OK 266 ms 858 B

- Pass Status code is 200

PUT Update Report {{awsReportEndPoint}}/dev/report/{{id}} / Update Report 200 OK 232 ms 672 B

- Pass Status code is 200
- Pass Validate ison response data created

Postman

Some Examples of api responses:

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, History.
- Current Collection:** report
- API Endpoints:**
 - aws report
 - POST Create User
 - PUT Update User
 - DEL Delete User
 - GET Get Users
 - GET Generate report users CSV
 - GET Generate report users PDF
 - GET Get User
 - report
 - GET Generate PDF Report
 - GET Generate CSV Report
 - POST Create Report
 - GET Get Report by Id
 - DEL Delete Report by Id
 - GET Get All Reports
 - PUT Update Report
- Selected Endpoint:** GET {{awsReportEndPoint}}/dev/reports
- Request Headers:** Authorization, Headers (6), Body, Pre-request Script, Tests, Settings.
- Query Params:** Key, Value, Description.
- Body:** PRETTY, Raw, Preview, Visualize, JSON. The JSON response is displayed below:

```

1  {
2   "alias": "Alias",
3   "species": "species",
4   "company": "Company",
5   "name": "company name",
6   "team": "company team"
7 },
8   "dateCreation": 1636485873228,
9   "id": "NRTEOL0p4LAshf-7EieLInx29XTbnLM4AQLVytxh",
10  "name": "Name"
11 },
12  {
13   "alias": "Alias",
14   "species": "species",
15   "company": "Company",
16   "name": "company name",
17   "team": "company team"
18 },
19   "dateCreation": 1636485870531.
20

```

- Response Summary:** Status: 200 OK, Time: 2.14 s, Size: 3.58 KB, Save Response.

Screenshot of a REST API testing tool interface showing a successful response for a PDF report generation endpoint.

The left sidebar shows "My Workspace" with collections like "report" containing various API endpoints:

- GET Generate PDF Report
- GET Generate CSV Report
- POST Create Report
- GET Get Report by Id
- DELETE Delete Report by Id
- GET Get All Reports
- PUT Update Report

The main panel displays the "Generate PDF Report" endpoint details:

Method: GET
URL: {{awsPdfReportEndPoint}}/api/pdf_report
Params: Authorization, Headers (6), Body, Pre-request Script, Tests, Settings
Query Params: KEY, VALUE, DESCRIPTION, Bulk Edit
Body: Status: 200 OK, Time: 267 ms, Size: 2.21 KB, Save Response

The response body contains a table titled "Report" with the following data:

| Name | Alias | Species | Company Name | Company Team |
|--------------|---------------|-----------------|----------------------|----------------------|
| Cesar Vegas | cesarvega | special | The factory of users | Dandevils |
| Name example | Alias example | species example | company name example | company team example |