

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

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

- Install serverless

`npm i serverless`

- Install newman

`npm install -g newman`

- 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>

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

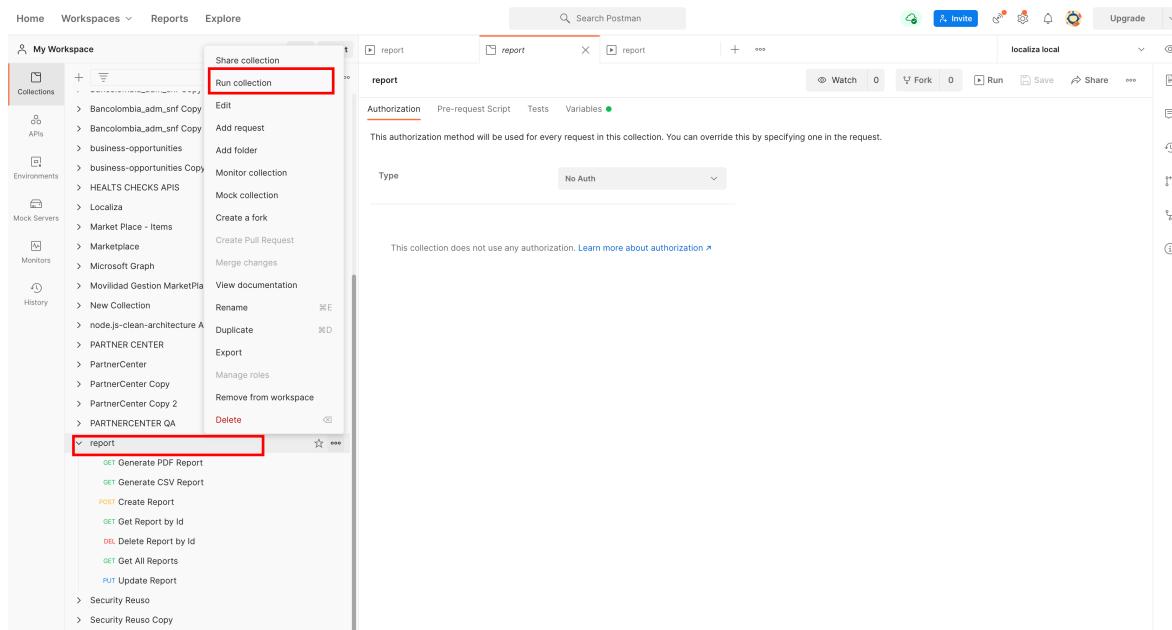
Services endpoints

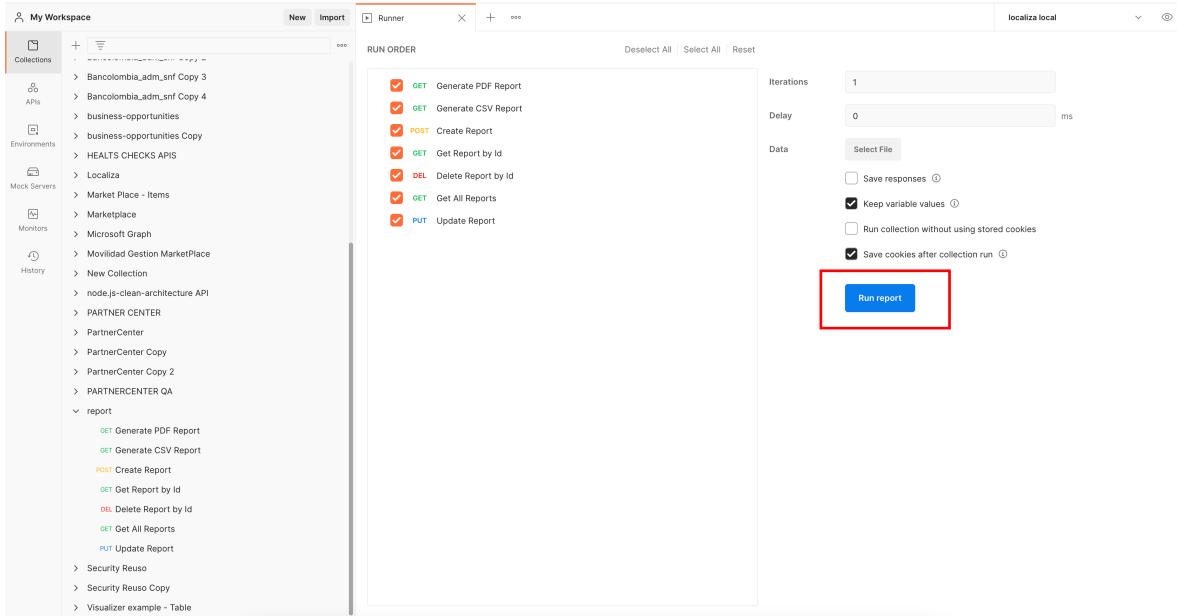
Consume in postman any of the following services:

- 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:





Run integration test in console with newman

Run the next command replace the vars `awsPdfReportEndPoint` and `awsReportEndPoint` with the endpoint url

```
newman run documentation/report.postman_collection.json --global-var
"awsPdfReportEndPoint=https://xxxx.execute-api.us-east-1.amazonaws.com" -
-global-var "awsReportEndPoint=https://xxxx.execute-api.us-east-
1.amazonaws.com"
```

The screenshot shows a Visual Studio Code interface with the following details:

- Left Sidebar:** Shows icons for Source Control, Search, and other development tools.
- Top Bar:** Displays the title "package.json — serverless-dynamodb-nodejs-starter".
- File Explorer:** Shows a tree view with "Changes" selected, indicating no changes have been made.
- Editor:** Displays the contents of the package.json file, specifically the "dependencies" section, which includes "body-parser", "dynamodb-doc-client-wrapper", "express", "generate-safe-id", "jest", "json2csv", "mocha", "pdfkit-table", "save", and "save-dev". Lines 12 through 21 are highlighted.
- Terminal:** Shows the output of a command, likely a test or report generation, detailing various API requests and their results (e.g., GET, PUT, POST) with status codes like 200, 208, and 209, along with response times and assertion counts.
- Bottom Status Bar:** Shows the path "macBook-Pro-de-MAC serverless-dynamodb-nodejs-starter %".

Some Examples of api responses:

My Workspace

New Import

aws report

report / Get All Reports

GET {{awsReportEndPoint}}/dev/reports

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 214 s Size: 3.58 KB Save Response

```
1 [
2   {
3     "alias": "Alias",
4     "species": "Species",
5     "company": [
6       {
7         "name": "Company name",
8         "team": "Company team"
9       },
10      {
11        "alias": "Alias",
12        "species": "Species",
13        "company": [
14          {
15            "name": "Company name",
16            "team": "Company team"
17          },
18        ],
19        "dateCreation": 1636485870531
20      }
21    ]
22 }
```

My Workspace

New Import

report

report / Generate PDF Report

GET {{awsPdfReportEndPoint}}/api/pdf_report

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 267 ms Size: 2.21 KB Save Response

Report				
Name	Alias	Species	Company Name	Company Team
Cesar Vegas	cesarvega	special	The factory of users	Dandemis
Name example	Alias example	species example	company name example	company team example