

Documentación de uso ST-Backend

1- Pre-Requisitos para el despliegue de la aplicación en AWS

A- Tener un usuario con los tipos de acceso: Acceso programático y Acceso a la consola de administración de AWS.

B- Se le debe agregar al usuario la política de directiva: AdministratorAccess.

C- Registrar las variables AWS IAM en el sistema operativo (AWS Access Key ID, AWS Secret Access Key, etc.), usando el comando “aws configure”.

2- Despliegue de la aplicación en AWS

A- Descargar la aplicación desde la página de GitHub.

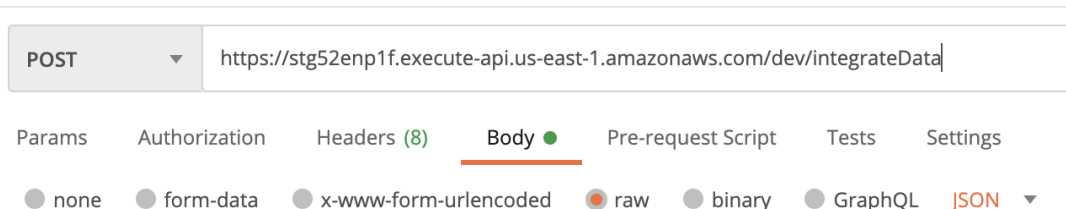
B- Instalar las librerías en la carpeta “node_modules”(npm install).

C- Desplegar la aplicación al AWS ejecutando el comando: sls deploy -v. La aplicación generara las tablas en “DynamoDB”. Se generarán los endpoints get y post.

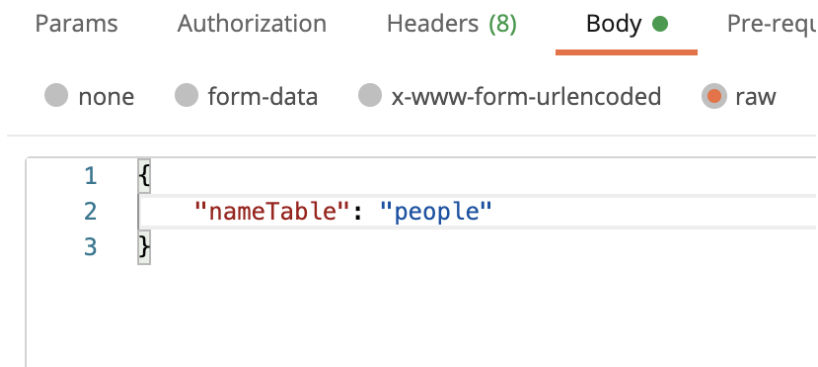
```
endpoints:
  POST - https://dhz9e87xs7.execute-api.us-east-1.amazonaws.com/dev/integrateData
  GET - https://dhz9e87xs7.execute-api.us-east-1.amazonaws.com/dev/people
  GET - https://dhz9e87xs7.execute-api.us-east-1.amazonaws.com/dev/person/{id}
functions:
  integrateData: st-backend-ljc-dev-integrateData
  getPeople: st-backend-ljc-dev-getPeople
  getPerson: st-backend-ljc-dev-getPerson
layers:
  None
```

3- Migrando datos de "The Star Wars API" a "DynamoDB"

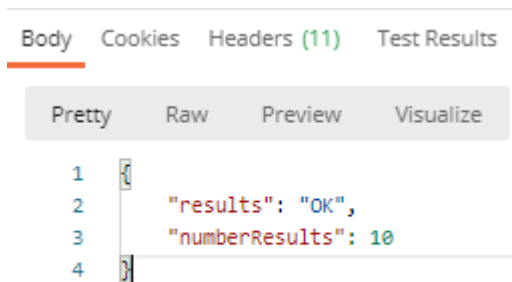
A- Abrir el programa “Postman” y pegar el endpoint post “integrateData”.



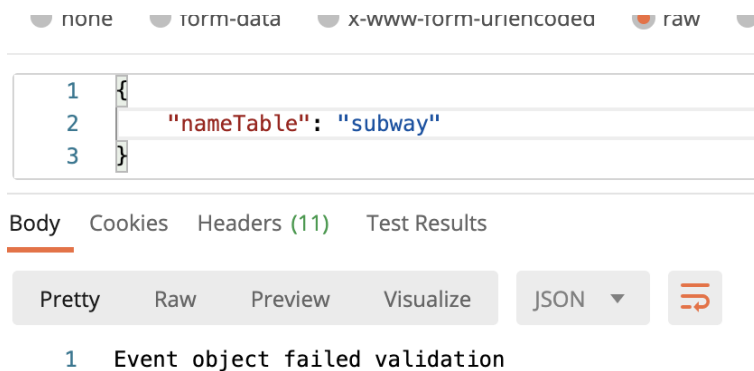
B- Le pasaremos como parámetro los nombres de las tablas: People o Planets.



C- Ejecutamos la aplicación y si la migración de datos sale correcta obtendremos la cantidad de objetos migrados.

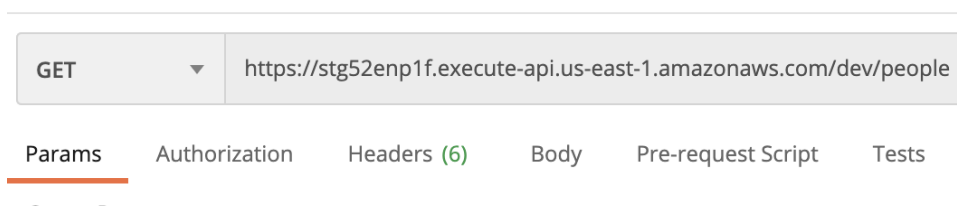


D- Si no ponemos los parámetros correctos (People o Planets) o no ponemos ningún parámetro, se nos mostrara un error de validación de objetos.



4- Obtención de datos de todos los personajes en “DynamoDB”

A- Abrir el programa “Postman” y pegar el endpoint get “people”.

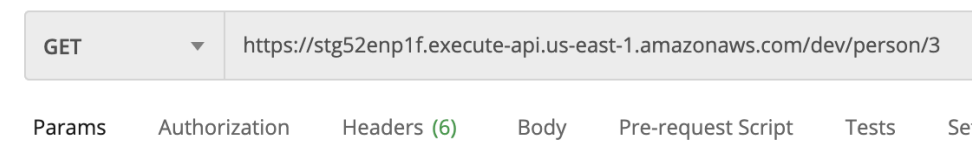


B- Ejecutamos la aplicación y luego se nos mostrara la información de todos los personajes.

```
35     "anio_nacimiento": "unknown",
36     "genero": "n/a"
37   },
38   {
39     "naves": "[\"12\"]",
40     "nombre": "Biggs Darklighter",
41     "color_cabello": "black",
42     "tamano": "183",
43     "masa": "84",
44     "vehiculos": "[]",
45     "peliculas": "[\"1\"]",
46     "editado": "2014-12-20T21:17:50.323000Z",
```

5- Obtención de datos de un solo personaje “DynamoDB”

A- Abrir el programa “Postman”, pegar el endpoint get “person” y escribir el “ID” del personaje que desee.



B- Ejecutamos la aplicación y luego se nos mostrara la información del personaje seleccionado.

```

1
2  "naves": "[]",
3  "nombre": "R2-D2",
4  "color_cabello": "n/a",
5  "tamano": "96",
6  "masa": "32",
7  "vehiculos": "[]",
8  "peliculas": "[\"1\", \"2\", \"3\", \"4\", \"5\", \"6\", \"7\"]",
9  "editado": "2014-12-20T21:17:50.311000Z",
10 "color_piel": "white, blue",
11 "especies": "[\"2\"]",
12 "creado": "2014-12-10T15:11:50.376000Z",
13 "id": "3",
14 "color_ojos": "red",
15 "mundoNatal": "8",
16 "anio_nacimiento": "33BBY",
17 "genero": "n/a"
18

```

C- En el caso de que no exista el personaje con el “ID” seleccionado, se no mostrara un mensaje de error.

```

Body  Cookies  Headers (11)  Test Results
Pretty  Raw  Preview  Visualize  JSON
1  {
2    "error": "Person with ID \"100\" not found!"
3  }

```

6- Pruebas unitarias usando “Jest”

A- Abrir la consola desde la carpeta raíz del proyecto.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19042.1348]
(c) Microsoft Corporation. Todos los derechos reservados.
D:\Personal\Proyectos\st-backend>

```

B- Ejecutar el grupo de test para las funciones que tiene el archivo “service.support.js”, usando el comando “npm test integrateData/supports/service.support.spec.js”

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19042.1348]
(c) Microsoft Corporation. Todos los derechos reservados.

D:\Personal\Proyectos\st-backend>npm test integrateData/supports/service.support.spec.js_
```

C- Se ejecutarán las cuatro pruebas que abarcan todas las funciones del archivo “service.support.js”. Se mostrará la descripción de cada función con su marca de validación.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19042.1348]
(c) Microsoft Corporation. Todos los derechos reservados.

D:\Personal\Proyectos\st-backend>npm test integrateData/supports/service.support.spec.js

> st-backend@1.0.0 test D:\Personal\Proyectos\st-backend
> jest "integrateData/supports/service.support.spec.js"

PASS test/integrateData/supports/service.support.spec.js
  Test de generacion del objeto "Persona" que se guardara en dynamoDB
    ✓ Generar el objeto "Persona" usando el objeto "Person" que se obtiene de Swapi (4 ms)
    ✓ Convertir el objeto "Persona" a otro objeto "PersonaES" cuyos campos deben estar en español (1 ms)
  Test de generacion del objeto "Planeta" que se guardara en dynamoDB
    ✓ Generar el objeto "Planeta" usando el objeto "Planet" que se obtiene de Swapi (1 ms)
    ✓ Convertir el objeto "Planeta" a otro objeto "PlanetaES" cuyos campos deben estar en español

Test Suites: 1 passed, 1 total
Tests: 4 passed, 4 total
Snapshots: 0 total
Time: 1.457 s, estimated 2 s
Ran all test suites matching /integrateData\\supports\\service.support.spec.js/i.
```