

Serverless

RETO, REST API CON SERVERLESS FRAMEWORK DESPLEGADA EN AWS

Integración con DynamoDB

HOST: <https://8fx3fyijv1.execute-api.us-east-1.amazonaws.com/dev> (<https://8fx3fyijv1.execute-api.us-east-1.amazonaws.com/dev>)

PRODUCTS

POST createOne

{{HOST}}/products

Método para crear productos y almacenarlos en una base de datos DynamoDB

BODY raw

```
{
  "productId": "3",
  "name": "Laptop",
  "description": "Core i7"
}
```

Example Request

createOne

```
curl --location -g --request POST '{{HOST}}/products' \
--data-raw '{
  "productId": "3",
  "name": "Laptop",
  "description": "Core i7"
}'
```

Documentation Settings ▼

Example Response

200 OK

Body Headers (14)

```
{
  "success": true,
  "message": "usuario creado con éxito"
}
```

GET getOne

```
{{HOST}}/products/:productId
```

Método para recuperar los items creados en la tabla de productos. Debes conocer el ID de producto para poder recuperarlo y mostrar la información.

PATH VARIABLES

productId

3

String, id del producto a buscar

Example Request

getOne

```
curl --location -g --request GET '{{HOST}}/products/3'
```

Example Response

200 OK

Body Headers (14)



```
"success": true,  
  "data": {  
    "description": "Core i7",  
    "name": "Laptop",  
    "productId": "3"  
  }  
}
```

USERS

Tabla almacenada en DynamoDB, se pueden crear nuevos usuarios y también recuperarlos.

GET `getOne`

{{HOST}}/users/:userId

Método para recuperar la data guardada en la base de datos de DynamoDB. Debes conocer el ID del usuario para poder recuperar sus datos.

PATH VARIABLES

userId

7

String, id de usuario

Example Request

`getOne`

```
curl --location -g --request GET '{{HOST}}/users/7'
```

Example Response

200 OK

Body	Headers (14)
------	--------------



```
"success": true,
"data": {
  "email": "angel@barrio.com",
  "firstName": "Angel",
  "lastName": "Barrios",
  "userId": "7"
}
```

POST createOne

```
{{HOST}}/users
```

Método para crear un nuevo usuario. La data enviada se guarda en la base de datos para su posterior recuperación.

BODY raw

```
{
  "userId": "11",
  "firstName": "Alvaro",
  "lastName": "fontela",
  "email": "y@gmail.com"
}
```

Example Request

[createOne](#)

```
curl --location -g --request POST '{{HOST}}/users' \
--data-raw '{
  "userId": "11",
  "firstName": "Alvaro",
  "lastName": "fontela",
  "email": "y@gmail.com"
}'
```

Example Response

200 OK

Body Headers (14)



```
"success": true,  
"message": "usuario creado con éxito"  
}
```

PEOPLE

Fuente SWAPI

GET `getPerson`

```
{{HOST}}/people/:personId
```

Hace un llamado a Stars Wars API para recuperar la información de personas (people). Debes conocer el id de la persona que deseas recuperar. Como respuesta te muestra todos los datos de esta persona.

PATH VARIABLES

personId

1

String, Id de la persona

Example Request

`getPerson`

```
curl --location -g --request GET '{{HOST}}/people/1'
```

Example Response

200 OK

Body Headers (14)



```
"success": true,
"data": {
  "nombre": "Luke Skywalker",
  "altura": "172",
  "peso": "77",
  "color_cabello": "blond",
  "color_piel": "fair",
  "color_ojo": "blue",
  "nacimiento": "1989Y".
```

[View More](#)

PLANETS

Fuente: SWAPI

GET getPlanets

```
{{HOST}}/planet/
```

Este método hace un llamado a SWAPI para recuperar la información de los planetas. Trae la primera página que contiene los planetas. No recibe parámetros.

Example Request

[getPlanets](#)

```
curl --location -g --request GET '{{HOST}}/planet/'
```

Example Response

200 OK

Body Headers (14)

```
{
  "success": true,
  "data": {
    "total": 60,
    "siguiente": "http://swapi.dev/api/planets/?page=2",
    "anterior": null,
    "resulados": [
      {
        "nombre": "Tatooine",
```

[View More](#)

