

fuego-de-quasar by Paula

Se puede ejecutar el programa con los siguientes datos:

- ★ Api Url: <https://geyc73n98h.execute-api.us-east-2.amazonaws.com/default/topsecret>
- ★ Método: POST
- ★ Body:

```
{
  "satellites": [
    {
      "name": "kenobi",
      "distance": 100.0,
      "message": [ "este", "", "", "mensaje", "" ]
    },
    {
      "name": "skywalker",
      "distance": 115.5,
      "message": [ "", "es", "", "", "secreto" ]
    },
    {
      "name": "sato",
      "distance": 142.7,
      "message": [ "este", "", "un", "", "" ]
    }
  ]
}
```

Para obtener el 404 es necesario "borrar" alguna de las palabras de los mensajes. No logré que falle por no encontrar coordenadas, las encuentra siempre.

Como las coordenadas conocidas de los satélites están preseleccionadas, es importante ingresarlos en el mismo orden del ejemplo. Se matchean en ese orden con su ubicación hardcodeda.

Los endpoints pedidos en el nivel 3 son:

- ★ Api Url: https://geyc73n98h.execute-api.us-east-2.amazonaws.com/default/topsecret_split/{nombre de la nave}
- ★ Método: POST
- ★ Body:

```
{
  "distance": 143.7,
  "message": [ "este", "", "un", "", "" ]
}
```

- ★ Api Url: https://geyc73n98h.execute-api.us-east-2.amazonaws.com/default/topsecret_split
- ★ Método: GET

Como las posiciones están hardcodedas para los tres satélites del ejemplo, deben estar esas tres guardadas para que funcione todo el proceso. Los datos que se envían en el post se guardan en una dybamodb. A medida que se cargan nuevas distancias y mensajes se pisan sobre el mismo nombre de satélite. Se pueden cargar otras naves con otros nombres y quedan guardadas, pero no se usan para encontrar la ubicación de la nave emisora del mensaje.

Notas

- Para hallar las coordenadas de la nave emisora del mensaje utilicé una librería que resuelve problemas de trilateración. Esa librería siempre encuentra una respuesta con ciertos parámetros de proximidad que no tuve en cuenta en mi algoritmo. Dejo el link a la librería: <https://github.com/lemmingapex/trilateration>
- Los servicios de AWS que utilicé fueron: tres lambdas, una base de datos dynamo y un API Gateway.
- El lambda que se ejecuta en el GET del topsecret_split, recupera los datos de la base y arma el request para el otro lambda que busca la posición y el mensaje en el endpoint /topsecret
- Sólo hice tests para el lambda rebellion-finder-lambda.

Lambda rebellion-finder-lambda

Lambda > Functions > rebellion-finder-lambda

rebellion-finder-lambda

ThrottleCopy ARNActions

Function overview Info

rebellion-finder-lambda

Layers (0)

API Gateway (3)

+ Add trigger

+ Add destination

Description-

Last modified3 hours ago

Function ARNarn:aws:lambda:us-east-2:271420477283:function:rebellion-finder-lambda

Lambda rebellion-finder-add-split-lambda

rebellion-finder-add-split

ThrottleCopy ARNActions

Function overview Info

rebellion-finder-add-split

Layers (0)

API Gateway

+ Add trigger

+ Add destination

Description-

Last modified20 hours ago

Function ARNarn:aws:lambda:us-east-2:271420477283:function:rebellion-finder-add-split

Lambda rebellion-finder-get-split-lambda

rebellion-finder-get-split

ThrottleCopy ARNActions

Function overview Info

rebellion-finder-get-split

Layers (0)

API Gateway

+ Add trigger

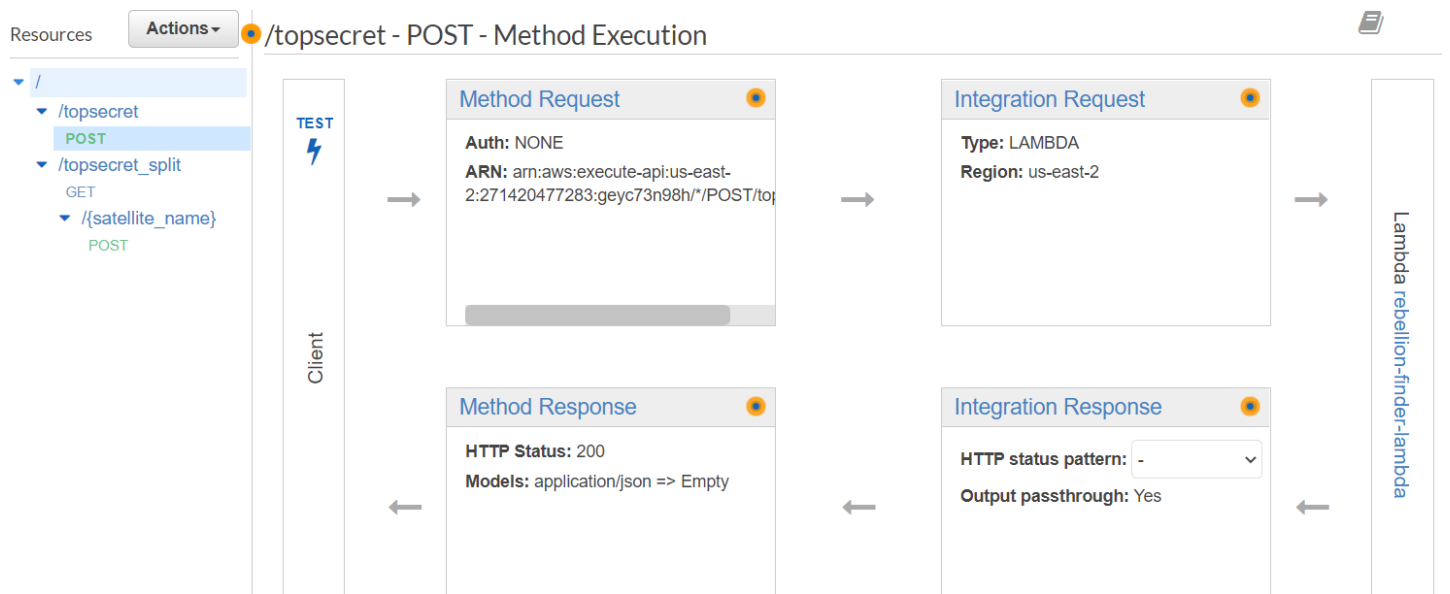
+ Add destination

Description-

Last modified19 hours ago

Function ARNarn:aws:lambda:us-east-2:271420477283:function:rebellion-finder-get-split

API Gateway
Endpoint: /topsecret (post)



Respuesta 200 (en Postman)

POST https://geyc73n98h.execute-api.us-east-2.amazonaws.com/default/topsecret

Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "satellites": [
3     {
4       "name": "kenobi",
5       "distance": 100.0,
6       "message": [ "este", "", "", "mensaje", "" ]
7     },
8     {
9       "name": "skywalker",
10      "distance": 115.5,
11      "message": [ "", "es", "", "", "secreto" ]
12    },
13    {
14      "name": "sato",
15      "distance": 142.7,
16      "message": [ "este", "", "un", "", "" ]
17    }
18  ]
19 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 1866 ms Size: 366 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "position": {
3     "x": -58.32,
4     "y": -69.55
5   },
6   "message": "este es un mensaje secreto"
7 }
```

Respuesta 404 (en Postman)

POST <https://geyc73n98h.execute-api.us-east-2.amazonaws.com/default/topsecret> Send Save

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Code

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** Beautify

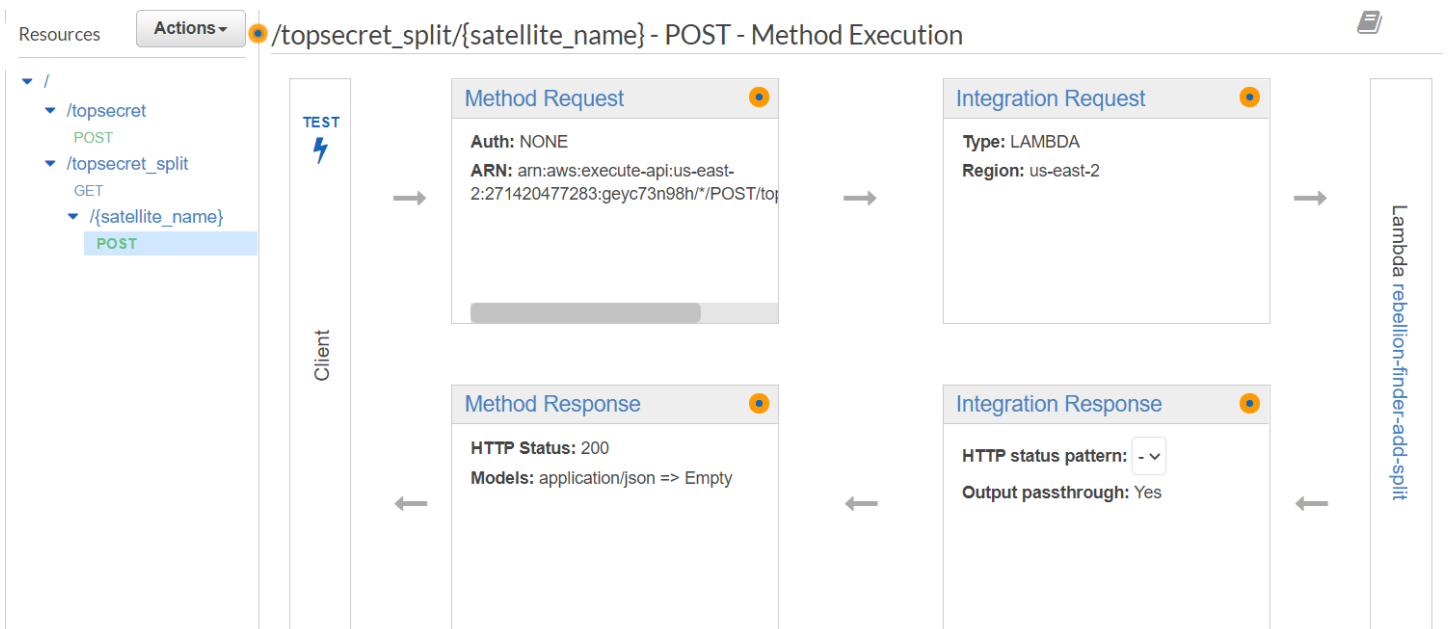
```
1 {
2   "satellites": [
3     {
4       "name": "kenobi",
5       "distance": 100.0,
6       "message": [ "este", "", "", "mensaje", "" ]
7     },
8     {
9       "name": "skywalker",
10      "distance": 115.5,
11      "message": [ "", "es", "", "", "" ]
12    },
13    {
14      "name": "sato",
15      "distance": 142.7,
16      "message": [ "este", "", "un", "", "" ]
17    }
18  ]
19 }
```

Body Cookies Headers (7) Test Results Status: 404 Not Found Time: 2.69 s Size: 297 B Save Response

Pretty Raw Preview Visualize JSON

1

Endpoint: /topsecret_split/{nombre de la nave} (post)



Respuesta 200 (en Postman)

POST https://geyc73n98h.execute-api.us-east-2.amazonaws.com/default/topsecret_split/millennium_falcon Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON**

```
1 {
2   "distance": 143.7,
3   "message": [ "este", "", "un", "", "" ]
4 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 233 ms Size: 294 B Save

Pretty Raw Preview Visualize JSON

Respuesta 404 (en Postman)

POST ▼ https://geyc73n98h.execute-api.us-east-2.amazonaws.com/default/topsecret_split/millennium_falcon Send Save ▼

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Code

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▼ Beautify

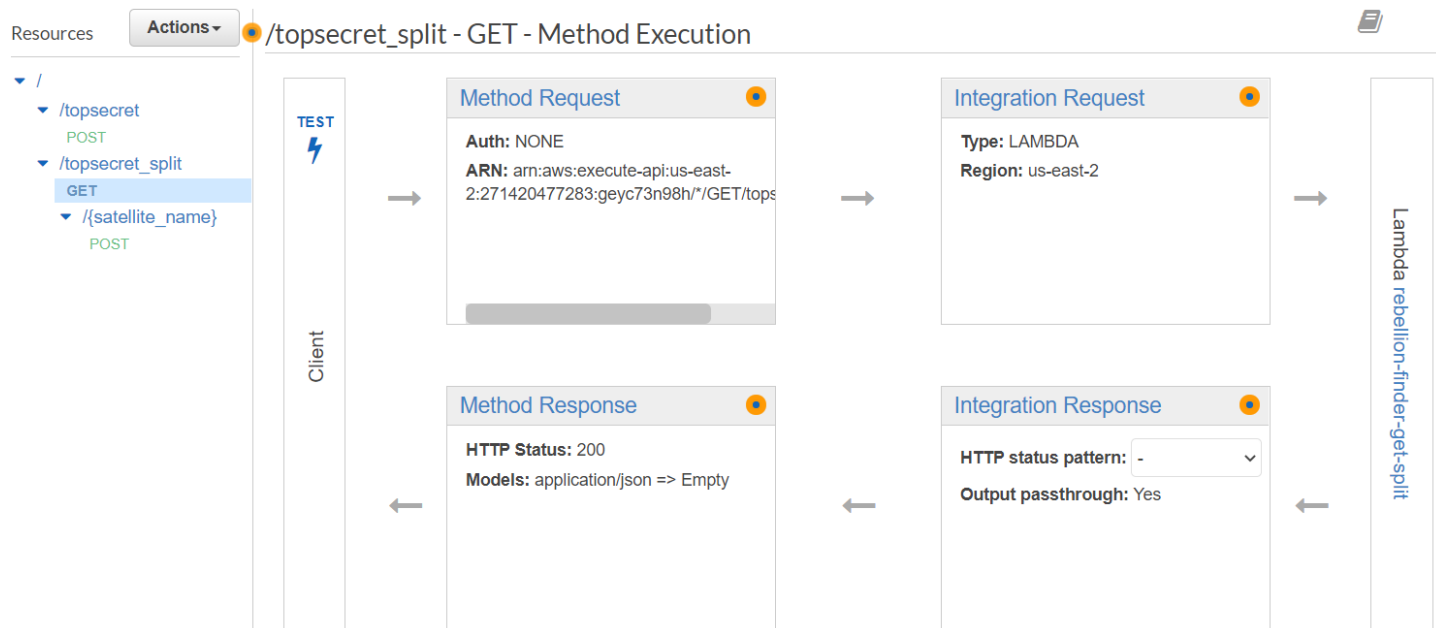
```
1 {
2   "message": [ "este", "", "un", "", "" ]
3 }
```

Body Cookies Headers (7) Test Results 🌐 Status: 400 Bad Request Time: 676 ms Size: 302 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 🔍

```
1 {}
```

Endpoint: /topsecret_split (get)



Respuesta 200

GET ▼ https://geyc73n98h.execute-api.us-east-2.amazonaws.com/default/topsecret_split Send Save ▼

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

Query Params

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
| Key | Value | Description | | |

Body Cookies Headers (7) Test Results 🌐 Status: 200 OK Time: 14.69 s Size: 366 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 🔍

```
1 {
2   "position": {
3     "x": -58.32,
4     "y": -69.55
5   },
6   "message": "este es un mensaje secreto"
7 }
```

Respuesta 418

GET

https://geyc73n98h.execute-api.us-east-2.amazonaws.com/default/topsecret_split

Send

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

Query Params

| | KEY | VALUE | DESCRIPTION | ... |
|--|-----|-------|-------------|-----|
| | Key | Value | Description | |

- Body
- Cookies
- Headers (7)
- Test Results
- Status: 418 I'm a teapot (RFC 2324)
- Time: 1253 ms
- Size: 361 B
- Save f

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "message": "No hay suficiente informacion"
3 }
```