

HAPPY TRAVEL API DOC

Hackers Cats

Crear una base de datos MySQL : happy_travel
Migraciones y seeding: php artisan migrate --seed

Levantar la app : php artisan serve
Abrir la base de datos con Xampp : Apache : start / admin

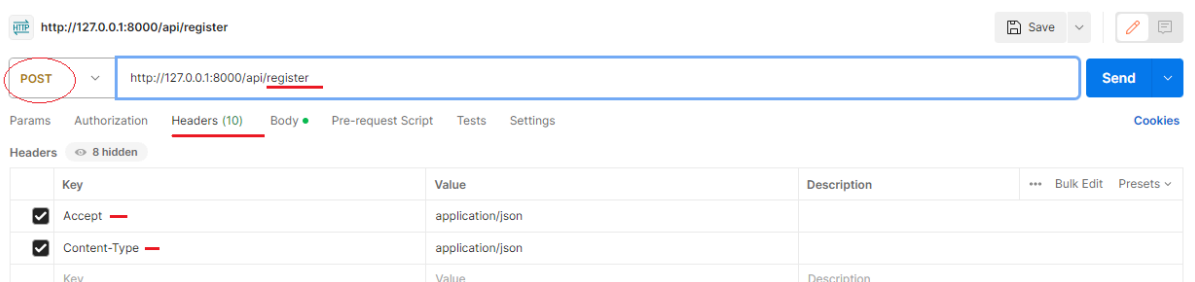
I - REGISTER / LOGIN / LOGOUT

1 - Register

Metodo POST

URL : http://127.0.0.1:8000/api/register

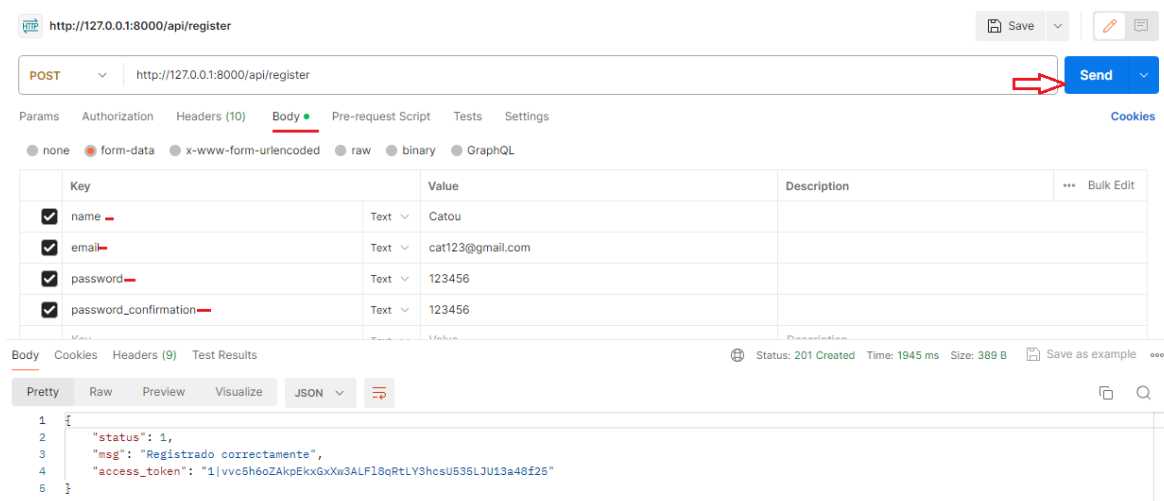
Rellenar la pestaña “Headers” según la foto siguiente



The screenshot shows the 'Headers' tab of a REST client interface. The URL is set to 'http://127.0.0.1:8000/api/register'. The method is 'POST'. The 'Headers' tab is selected, showing a table with the following headers:

Key	Value	Description
Accept	application/json	
Content-Type	application/json	

Rellenar la pestaña “Body” según la foto siguiente y Send. En la parte baja, se ve el nuevo usuario con el token



The screenshot shows the 'Body' tab of the REST client interface. The URL is 'http://127.0.0.1:8000/api/register'. The method is 'POST'. The 'Body' tab is selected, showing a table with the following headers:

Key	Value	Description
name	Catou	
email	cat123@gmail.com	
password	123456	
password_confirmation	123456	

Below the table, the 'Send' button is highlighted with a red arrow. At the bottom, the 'Test Results' tab is selected, showing the JSON response:

```
{
  "status": 1,
  "msg": "Registrado correctamente",
  "access_token": "1|vvc5h6oZakpEkxGxXw3ALF18qRtLY3hcsU535LJU13a48f25"
}
```

2 - Login

Metodo POST

URL : http://127.0.0.1:8000/api/login

Rellenar la pestaña “Headers” como se hizo en la parte Register

Rellenar la pestaña “Body” según lo siguiente. La sesión se inicia con un token

The screenshot shows a Postman interface for a POST request to `http://127.0.0.1:8000/api/login`. The request body is set to `form-data`. The body contains two fields: `email` with value `cat123@gmail.com` and `password` with value `123456`. The response status is `200 OK` with a time of `1928 ms` and a size of `395 B`. The response body is shown in JSON format:

```
1 {
2   "status": 1,
3   "msg": "Inicio de sesión correctamente",
4   "access_token": "2|bzKTM31ydWAMFsG4CuaesVc5Usj2YBfA5yK4AvFK79542d8a"
5 }
```

3 - Logout

Metodo GET

URL : http://127.0.0.1:8000/api/logout

Copiar el token que recibí al hacer el login. Ir a la pestaña “Authorization”, seleccionar

“Type”: “Bearer Token” y pegar el token copiado

Cambiar el método http a GET y clicar “Send”. La sesión está cerrada y el token destruido.

The screenshot shows a Postman interface for a GET request to `http://127.0.0.1:8000/api/logout`. The request method is set to `GET`. The Authorization tab is selected, and the type is set to `Bearer Token`. The token value is `3|HjJR49rjxAwEJODoBlh1OXUE6meKwrG2N...`, which is highlighted with a red arrow. The response status is `200 OK` with a time of `812 ms` and a size of `311 B`. The response body is shown in JSON format:

```
1 {
2   "status": 1,
3   "msg": "Sesión cerrada"
4 }
```

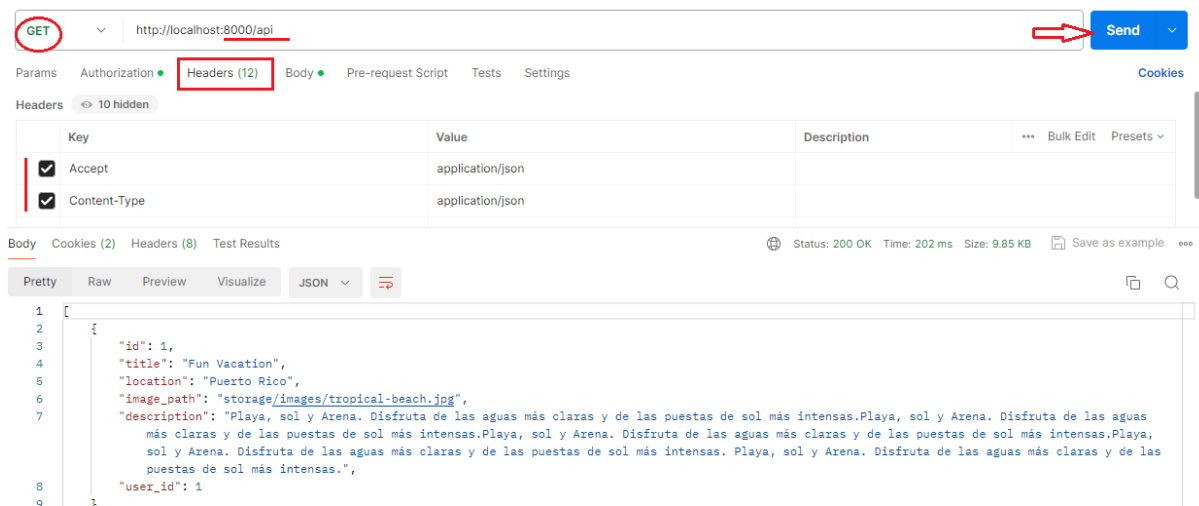
II - C.R.U.D.

1 - Index : recuperar todos los viajes

Metodo GET

URL : <http://127.0.0.1:8000/api>

Rellenar la pestaña “Headers” según el modelo siguiente y clicar en “Send”

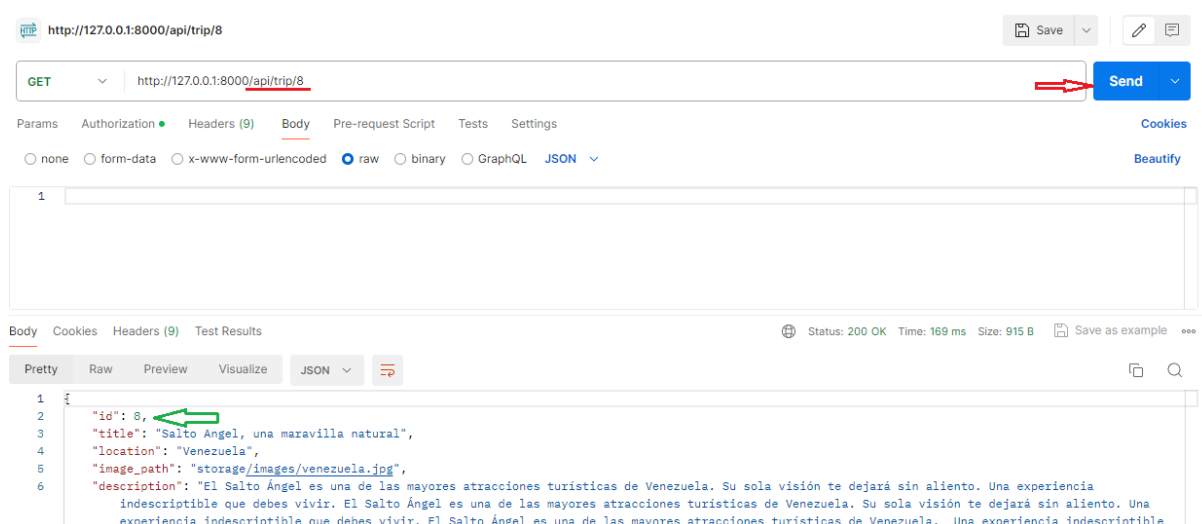


2 - getByld : encontrar un viaje por id

Metodo GET

URL : <http://127.0.0.1:8000/api/trip/id>

El usuario tiene que estar logueado. Copiar el token que recibió al hacer el login. Ir a la pestaña "Authorization", seleccionar "Type" : "Bearer Token" y pegar el token copiado



3 - Search : hacer una búsqueda según título o ubicación

Metodo GET

URL : <http://127.0.0.1:8000/api/search>

Rellenar la pestaña “Body” según el modelo siguiente (introduciendo search en Key y la palabra buscada en Value) y clicar en “Send”

Postman interface showing a GET request to <http://localhost:8000/api/search>. The request is configured with the method GET and the URL <http://localhost:8000/api/search>. The 'Body' tab is selected, showing a JSON object with the key 'search' and the value 'grecia'. The 'Send' button is highlighted with a green arrow.

Query Params

Key	Value	Description
search	grecia	Description

Body

```
{
  "id": 4,
  "title": "Escapada Soñada",
  "location": "Grecia",
  "image_path": "storage/images/grecia.jpg",
}
```

4 - Store : añadir un viaje

Metodo POST

URL : <http://127.0.0.1:8000/api/store>

Copiando el token en la pestaña “Authorization” (cf Login).

Rellenar la pestaña “Body” según la foto siguiente y clicar “Send”. El viaje está creado.

Postman interface showing a POST request to <http://127.0.0.1:8000/api/store>. The request is configured with the method POST and the URL <http://127.0.0.1:8000/api/store>. The 'Body' tab is selected, showing a JSON object with the following fields: title (Disfruta de la literatura en Barcelona), location (Spain), image (Interior-Biblioteca-Gabriel-Garcia-Marquez-Mari...), and description (Ven a visitar la biblioteca Gabriel García Marquez, la biblioteca...). The 'Send' button is highlighted with a green arrow.

Body

```
{
  "message": "Viaje se ha creado correctamente"
}
```

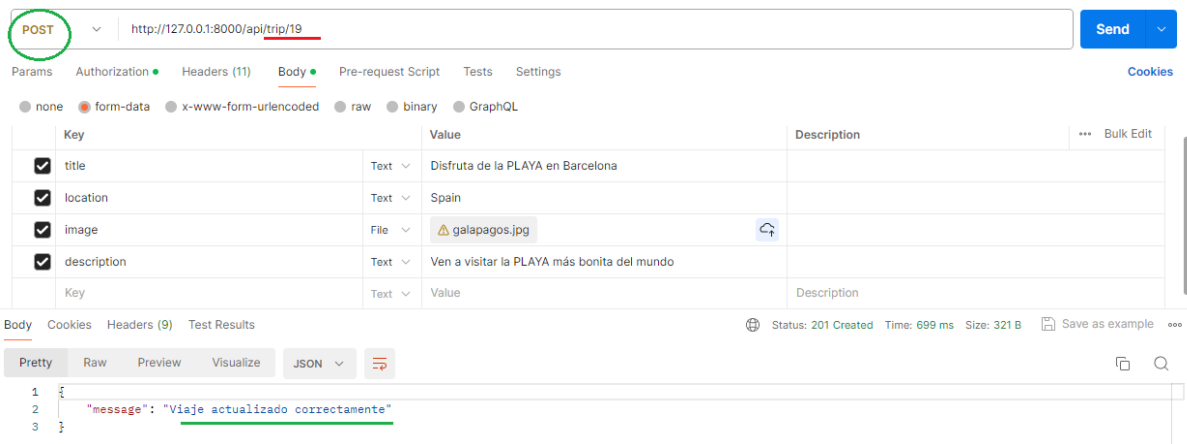
5 - Update

Metodo POST

URL : <http://127.0.0.1:8000/api/trip/id>

Copiando el token en la pestaña "Authorization" (cf Login).

Rellenar la pestaña "Body" según la foto siguiente y clicar "Send". El viaje está actualizado.



5 - Destroy: borrar un viaje

Metodo DELETE

URL : <http://127.0.0.1:8000/api/trip/id>

