

Challenge

Usuario CRUD

Para poder operar dentro de la API, lo primero que se debe hacer es crear un usuario.

POST Create

`http://127.0.0.1:5000/user`

Los campos obligatorios del body son:

*username

*mail

*password

Body raw (json)

json

```
{
  "username": "Prueba",
  "mail": "ejemplo@prueba.com",
  "password": "123"
}
```

GET Read one

`http://127.0.0.1:5000/user/1`

Para obtener los datos de un usuario es necesario colocar su id, en este caso "1"

GET Read many

http://127.0.0.1:5000/users?per_page=10&page=1

Para obtener varios usuarios de forma simultanea se necesitan definir dos parametros:

*per_page (default 10)

*page (default 1)

PARAMS

| | |
|----------|----|
| per_page | 10 |
| page | 1 |

PUT Update

http://127.0.0.1:5000/user/2

Para realizar un update en alguno de los campos del usuario, se necesita primero logearse y colocar el token de autorizacion que te devuelve ese loguien.

Los campos posibles a actualizar son:

*username

*mail

*password

HEADERS

| | |
|---------------|---|
| Authorization | eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjIifQ.65MESU9lIKj2LYUoTy79QVnyAGA3G1ke377h7NKHx2I |
|---------------|---|

Body raw (json)

json

```
{
  "password": "1234"
}
```

DELETE Delete (permanente)

http://127.0.0.1:5000/user/1

Para borrar un usuario de forma permanente es necesario logearse y colocar en el header el token que devuelve ese login.

Posteriormente, el usuario que se puede borrar es unicamente el que esta logueado, de ser cualquier otro no se podra borrar.

Para borrar el usuario actual es necesario colocar el id.

HEADERS

| | |
|----------------------|--|
| Authorization | eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjQifQ.gWkaICFV3JOEBtT3FwY2loJCxE_iSb3srHaYYmOcKs |
|----------------------|--|

PATCH Delete (logico)

http://127.0.0.1:5000/user/2

Para dar de baja un usuario de forma logica es necesario logearse y colocar en el header el token que devuelve ese login.

Posteriormente, el usuario que se puede borrar es unicamente el que esta logueado, de ser cualquier otro no se podra borrar.

Para borrar el usuario actual es necesario colocar el id.

HEADERS

| | |
|----------------------|---|
| Authorization | eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjMifQ.jlpcaFXI5ThAWQCy53vgcsAwis7j6elk9YVithEKK5U |
|----------------------|---|

Flujo ADMIN

GET Ver pedidos de permisos

http://127.0.0.1:5000/admin/obtenerPermisos

Los administradores tienen la posibilidad de ver que usuarios pidieron permisos de admin para posteriormente decidir si darles el permiso o no.

Para ello se necesita colocar la autorizacion que devuelve el login en el header del request.

Posteriormente en el body se debera de colocar el user y password del usuario admin con el que queremos acceder a

estos datos.

HEADERS

Authorization

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjEifQ.9654P10KPceeFNvZ
zScDkdqiVzKQ5VRTujjBXpyg0AQ

Body raw (json)

json

```
{  
  "username": "Admin",  
  "password": "123"  
}
```

POST Dar permisos

http://127.0.0.1:5000/admin/darPermisos/2

Un admin tiene la posibilidad de dar permisos de admin a otros usuarios.

Para realizar esta tarea es necesario primero loguearse y obtener el token de autorizacion.

posteriormente en el body colocar el usuario y contraseña del usuario con el que nos logueamos.

Por ultimo, en los params se debera colocar el id del usuario al cual se le quiere dar permisos de admin.

HEADERS

Authorization

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjEifQ.9654P10KPceeFNvZ
zScDkdqiVzKQ5VRTujjBXpyg0AQ

Body raw (json)

json

```
{  
  "username": "Admin",  
  "password": "123"  
}
```

POST solicitar permisos ADMIN

http://127.0.0.1:5000/user/solicitarPermisos/2

Un usuario puede solicitar permisos de admin, para ello debe primero de loguearse y colocar el token de la autorizacion en el header del request.

Posteriormente debera poner su id. No se permite poner ids de otros usuarios.

HEADERS

| | |
|----------------------|---|
| Authorization | eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjIifQ.65MESU9lIkj2LYUoTy79QVnyAGA3G1ke377h7NKHx2I |
|----------------------|---|

POST login

http://127.0.0.1:5000/login

Para realizar el login primeramente se debe estar registrado.

Dentro del body se debera poner el usuario y la password que se colocaron al momento del registro.

Body raw (json)

json

```
{
  "username": "Prueba",
  "password": "1234"
}
```

GET verificacion de acceso

http://127.0.0.1:5000/puedopasar?user=Prueba

Para verificar el acceso de un usuario al sistema, se debera poner la autorizacion que devolvio el login en el header del request. y como parametro el nombre de usuario.

De tener acceso el sistema devolvera True, en caso contrario devolvera False.

HEADERS

Authorization

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjEifQ.9654P10KPceeFNvZ
zScDkdqiVzKQ5VRTujjBXpyg0AQ

PARAMS

user

Prueba