

Guía Backend

Preparación:

1. Instalar node JS
2. Instalar npm
3. Clonar el repo
4. Abrirlo en VS code
5. Ejecutar en la terminal del proyecto: npm i (Solo se debe hacer la primera vez, para instalar las librerías)
6. Ejecutar en la terminal del proyecto npm start (Para ejecutarlo)

```
PS C:\Users\nicol\OneDrive - Universidad de los Andes\Notebooks\Moviles\backend> npm start

> backend@1.0.0 start
> nodemon ./bin/www

[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node ./bin/www`
<===== Database connected =====
```

Que se puede hacer:

1. Get de todos los usuarios: GET localhost:3000/api/users/

The screenshot shows a REST client interface with a GET request to `localhost:3000/api/users/`. The response is a JSON array of two user objects. The status is 200 OK, and the response size is 1.34 KB.

KEY	VALUE	DESCRIPTION
Key	Value	Description

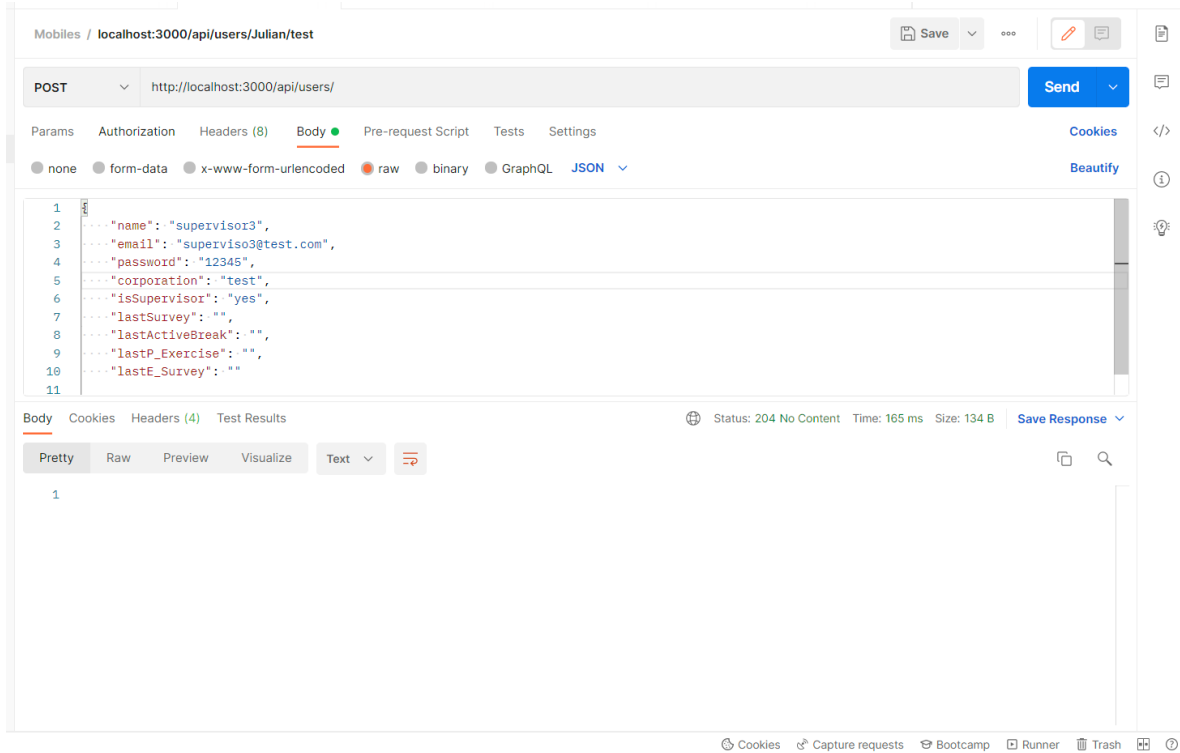
```
1  {
2    "users": [
3      {
4        "_id": "623bab8c269772ede5c62c58",
5        "name": "supervisor",
6        "email": "supervisor@test.com",
7        "password": "$2b$10$NPZz/5PSB83zTtpj_MD1Nu1L6oY5D1J8JN9vxiuy880v/KDPAHmZu",
8        "cooperation": "test",
9        "isSupervisor": "yes",
10       "lastSurvey": "2022-03-23T23:24:13.886Z",
11       "lastActiveBreak": "",
12       "lastP_Exercise": "",
13       "lastE_Survey": ""
14     },
15     {
16       "_id": "623baba7269772ede5c62c59",
17       "name": "supervisor2",
18       "email": "supervisor2@test2.com",
19       "password": "$2b$10$uU4NFZcQ8a3BNbg2MMYe91FDhwX/CaKZi6U10cZimLh0fUvS..m",
```

2. Registro de un usuario: POST <http://localhost:3000/api/users/>

Se deben pasar todos los parámetros que están ahí como un json. Los primeros 5 hasta isSupervisor deben venir del formulario de registro.

Los últimos cuatro deben estar siempre como un string vacío "", se usan para inicializar atributos como la última vez que el supervisor reviso las health surveys.

Como es un post la petición no devuelve nada.



3. Autenticar un usuario GET localhost:3000/api/users/email/password

Si existe un usuario con ese email y password devuelve todos sus datos en un JSON, si no, no devuelve nada.

The screenshot shows a REST client interface with the following details:

- URL:** localhost:3000/api/users/supervisor@test.com/12345
- Method:** GET
- Status:** 200 OK
- Time:** 158 ms
- Size:** 537 B
- Response Body (JSON):**

```
1 {
2   "_id": "623bab8c269772ede5c62c58",
3   "name": "supervisor",
4   "email": "supervisor@test.com",
5   "password": "$2b$10$NPZz/5PSB03zTtpj.MD1Nu1L6oY5D1J0JN9vx1uy000v/KDPAHmzu",
6   "cooperation": "test",
7   "isSupervisor": "yes",
8   "lastSurvey": "2022-03-23T23:24:13.806Z",
9   "lastActiveBreak": "",
10  "lastP_Exercise": "",
11  "lastE_Survey": ""
12 }
```

4. Obtener todos los usuarios de tipo employee de una corporacion: GET

localhost:3000/api/users/CORPORACION

En el ejemplo se devuelve una lista de usuarios employee de la corporación test

The screenshot shows a REST client interface with the following details:

- URL:** localhost:3000/api/users/test
- Method:** GET
- Status:** 200 OK, Time: 105 ms, Size: 794 B
- Response Format:** JSON
- Response Body:**

```
1  {
2    "users": [
3      {
4        "_id": "623babca269772ede5c62c5a",
5        "name": "employee",
6        "email": "employee@test.com",
7        "password": "$2b$10$piXmfeQ21891afbt0Ahbup9UizEEw9zhBgnFqRiLbDk/VqY1.9K0",
8        "corporation": "test",
9        "isSupervisor": "no",
10       "lastSurvey": "",
11       "lastActiveBreak": "",
12       "lastP_Exercise": "",
13       "lastE_Survey": ""
14     },
15     {
16       "_id": "623babca269772ede5c62c5b",
17       "name": "employee2",
18       "email": "employee2@test.com",
19       "password": "$2b$10$0jQmVom8yL6DQME7w2SjFenziHwe6DqYNkJWu1C1evu0dvaa0Xvif",
20       "corporation": "test",
21       "isSupervisor": "no",
22       "lastSurvey": "",
23       "lastActiveBreak": "",
24       "lastP_Exercise": "",
25       "lastE_Survey": ""
26     }
27   ]
28 }
```

5. Actualizar contadores, existen 4 contadores:

lastSurvey: Ultima vez que el supervisor reviso las health surveys.

`http://localhost:3000/api/users/lastSurvey/EMAIL`

lastActiveBreak: Ultima vez que el employee hizo una pausa activa.

`http://localhost:3000/api/users/lastActiveBreak /EMAIL`

lastP_Exercise: Ultima vez que el employee hizo un ejercicio personalizado

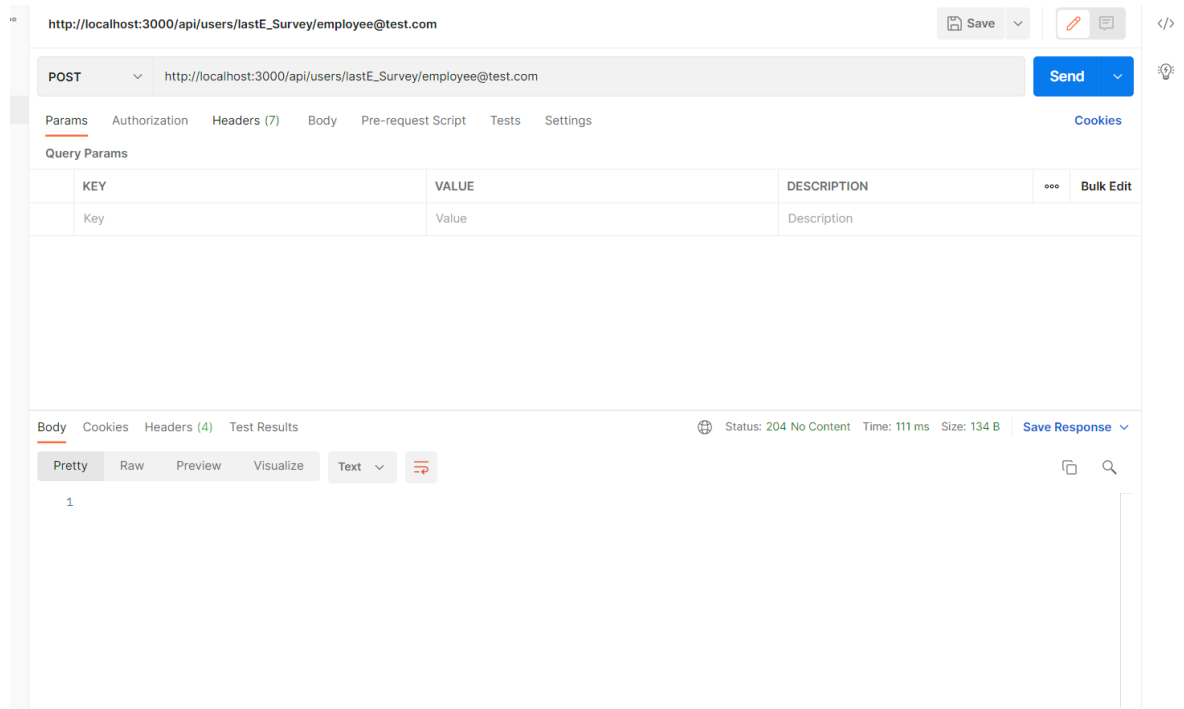
`http://localhost:3000/api/users/lastP_Exercise/EMAIL`

lastE_Survey: Ultima vez que el employee hizo una health survey:

`http://localhost:3000/api/users/lastE_Survey/EMAIL`

- Estos guardar como un string la fecha y hora de ese momento.
- Como es un post no devuelve nada.

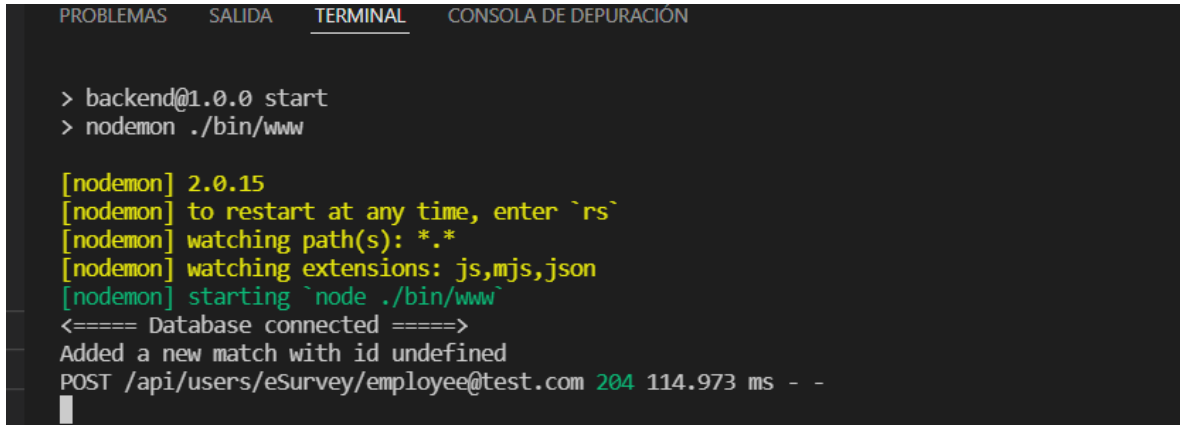
En el ejemplo se guarda la última vez que el empleado con email employee@test.com hizo una health survey.



Verificar:

Para verificar que las peticiones funcionaron se pueden hacer 2 cosas:

- 1) Revisar lo que imprime la consola del back. Esto imprimirá con cada petición.



```
PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN

> backend@1.0.0 start
> nodemon ./bin/www

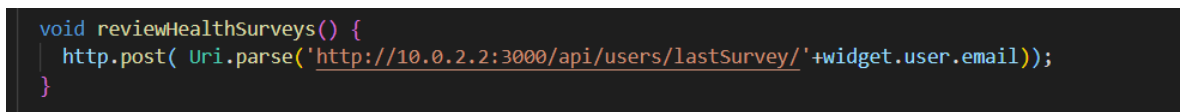
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node ./bin/www`
<===== Database connected =====>
Added a new match with id undefined
POST /api/users/eSurvey/employee@test.com 204 114.973 ms - -
```

- 2) Hacer desde postman la petición con todos los datos de todos los usuarios y verificar si cambiaron.

Extra:

En flutter las peticiones no se hacen a localhost puesto que el emulador del celular corre en una maquina virtual y su localhost no es el mismo. Por tal motivo se debe usar la ip: 10.0.2.2

Un ejemplo de implementación en flutter se puede ver cuando se hace el post para indicar la ultima vez que el supervisor reviso las health surveys.



```
void reviewHealthSurveys() {
  http.post( Uri.parse('http://10.0.2.2:3000/api/users/lastSurvey/'+widget.user.email));
}
```

Este método se llama cuando el botón de las health surveys es oprimido.