Memoria

Fabricio Flores

Users

Se implementaron los métodos faltantes, los cuales eran POST y PUT.

- **POST:** Para el método POST puede haber 3 casos:
  - username, email o password vacíos: El status de respuesta será `422` con el mensaje: `Username, e-mail or password is left out`.
  - username o email ya en uso: El status de la respuesta será `400` con el mensaje: `Username or email already exists`.
  - usuario creado satisfactoriamente. El status de la respuesta será `201` y la respuesta tendrá el objeto creado.

```
{
  "id": 42,
  "username": "fabricioflores",
  "email": "fabrifloresg@ff.com",
  "enabled": true,
  "token": "d0d1eb8bf2c72ca3d2dcaf1259cc6d42f942a15f"
}
```

- **PUT:** Para el métodoo PUT igualmente pueden haber 3 casos, cambiando la respuesta satisfactoria por un status `200`, y, además, una respuesta para cuando no se ecuentre el usuario con status `404`.

Results

En results se implementaron los siguientes métodos:

- **GET results/**: Este endpoint se encargará de listar los *Results* existentes y un status `200`. En caso de no existir ningún *Result* devolverá un status `404`.
- **GET results/{resultId}**: Este endpoint devolverá un *Result* basado en su id y un status `200`. En caso de no existir un *Result* con ese id devolverá un mensaje de error y el status `404`.
- **POST results/**: Este endpoint permitirá la inserción de un nuevo *Result* a la base de datos. En este caso, se podrá recibir dos tipos de respuesta:
  - Un status `201` y como cuerpo de la respuesta el objeto creado.

    - Un status `422` cuando no se hayan enviado el valor *result* o el valor *time*.

- **PUT results/{resultId}**: Este endpoint servirá para modificar un *Result*. Recibe en el url el id del result y en el cuerpo de la solicitud los datos. Las respuestas pueden ser las mismas del POST, cambiando la satisfatoria por un status `201` y agregando una de status `404` para cuando no se ha encontrado el *Result* a modificar.