

REST API para votaciones entre empleados

Autor: Pablo Landeta

Fecha: 03-07-2020

1. Introducción

Se desea tener un servidor BE que exponga una API Rest segura. La misma será utilizada por un sistema web para registrar votos de los empleados de una empresa hacia sus compañeros.

Notar que no es requisito implementar la interfaz gráfica y es se puede seleccionar la tech o framework libremente. Esto es un plus.

Debe haber dos roles de usuario con distintos niveles de acceso: Admin puede acceder a todo y Employee solo a algunos recursos

Lista de funcionalidades

Employee:

- Los votos realizados son para otro compañero, no se permite votos sobre uno mismo
- Los votos son sobre una persona en distintas áreas: Team player, Technical referent, Key Player, Client satisfaction, Motivation, Fun, etc...
- El voto puede ir opcionalmente con un comentario
- Una persona posee 1 voto disponible para cada área
- Los votos caducan a fin de mes y no se acumulan si fueron utilizados

Admin:

- Obtener los empleados más votados por mes
- Obtener cantidad de empleados registrados
- Obtener los empleados más votados por área

2. Diseño e Implementación

2.1 The REST API Specification

Todos:

- | | |
|---------------|--------------|
| - Registrarse | POST /signup |
| - Loguear | POST /login |
| - Salir | GET /logout |

Admin:

- | | |
|--------------------------------------|-------------------|
| - Lista de usuarios: | GET /users |
| - Cantidad de empleados registrados: | GET /countusers |
| - Listar votos de un empleado | GET /votes/userId |
| - Listar todos los votos | GET /votes |

Empleado

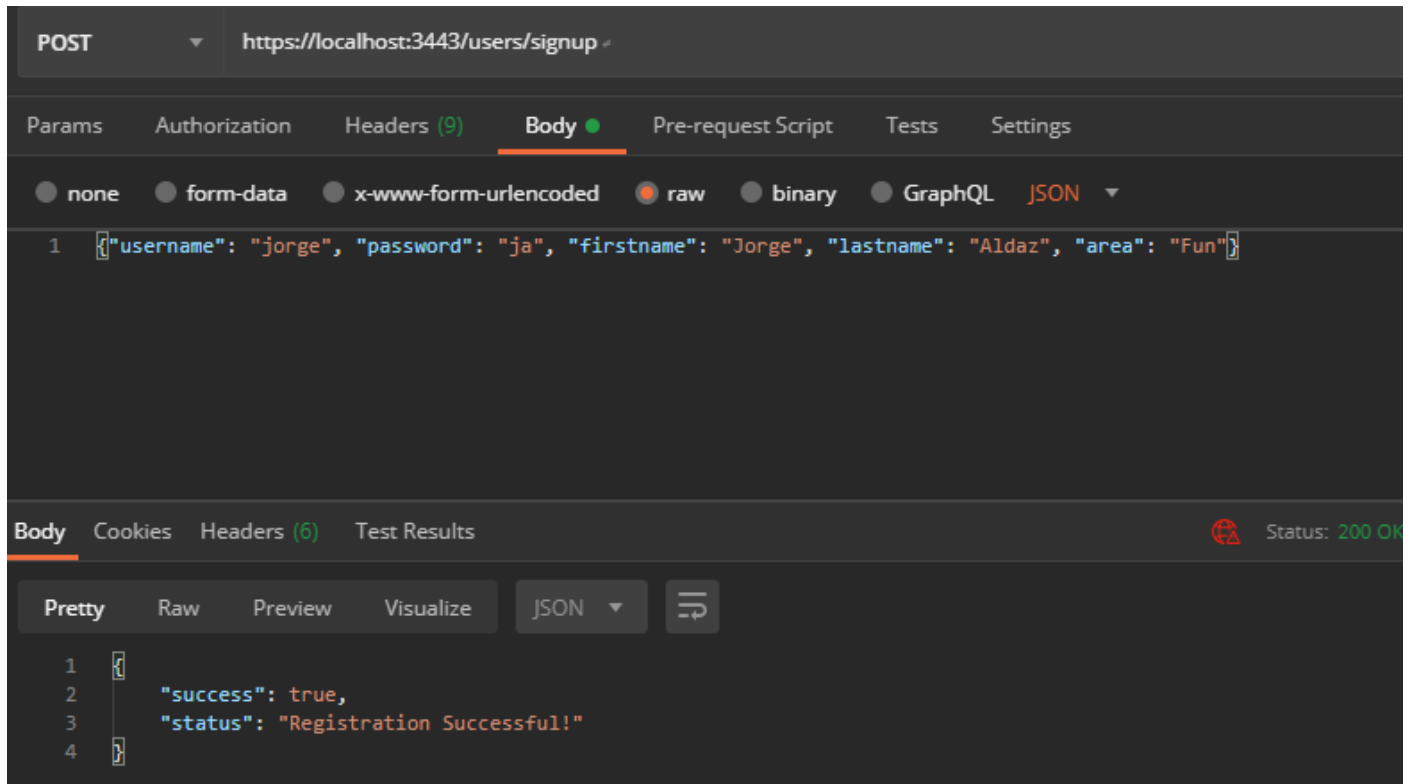
- Votar

POST /votes/userId

2.2 Schemas de Base de Datos, Diseño y Estructura

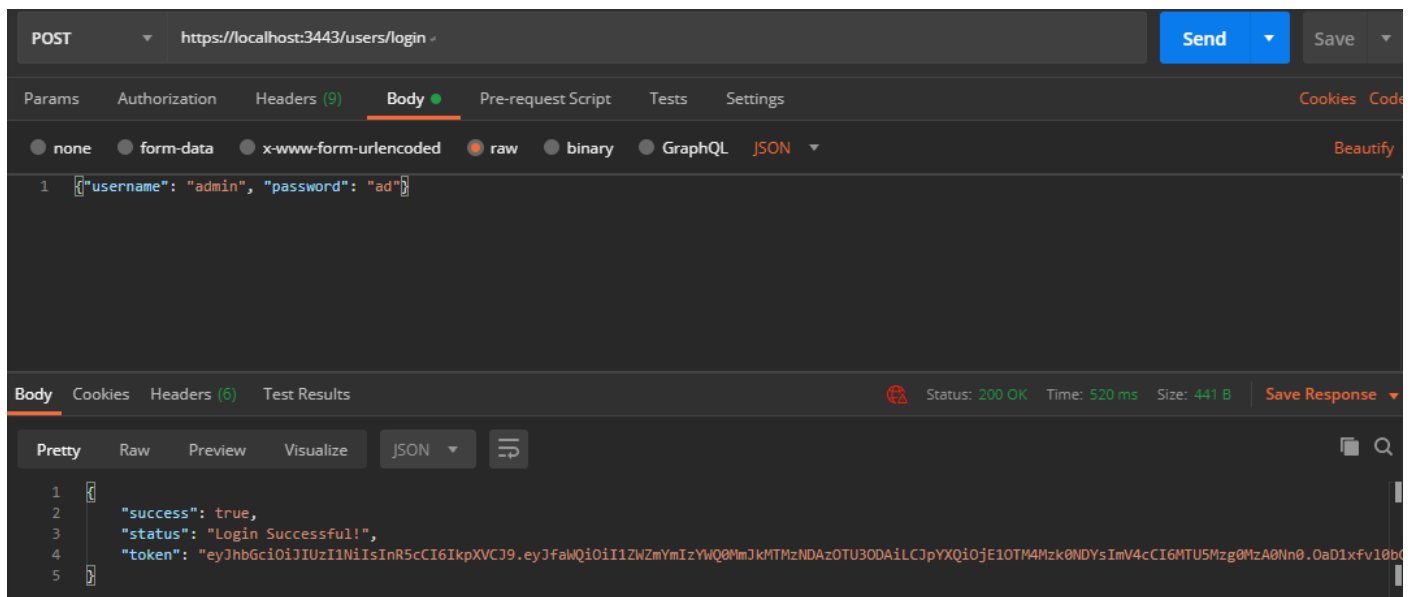
Registrarse POST /signup

```
{"username": "pablo", "password": "pa", "firstname": "Pablo", "lastname": "Landeta", "area": "Motivation"}
```



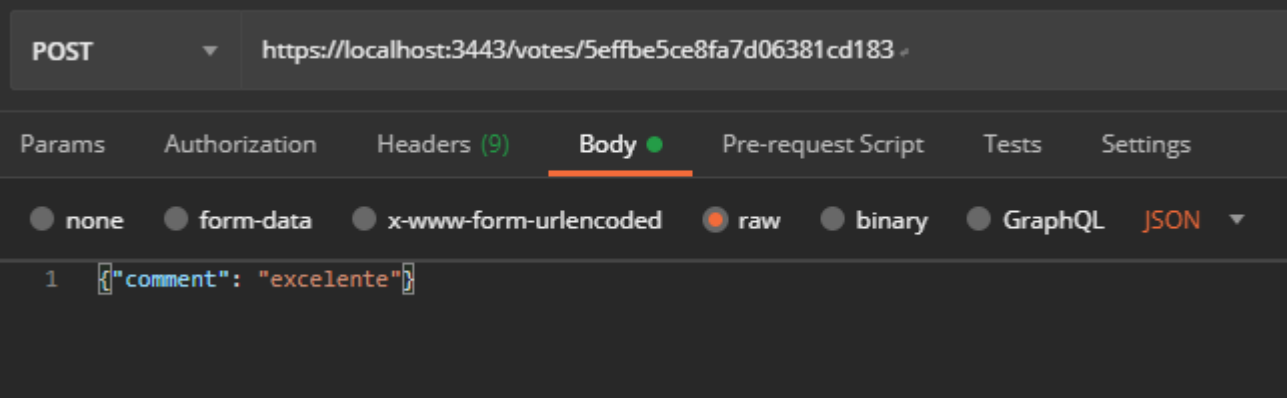
Loguear POST /login

```
{"username": "pablo", "password": "pa"}
```

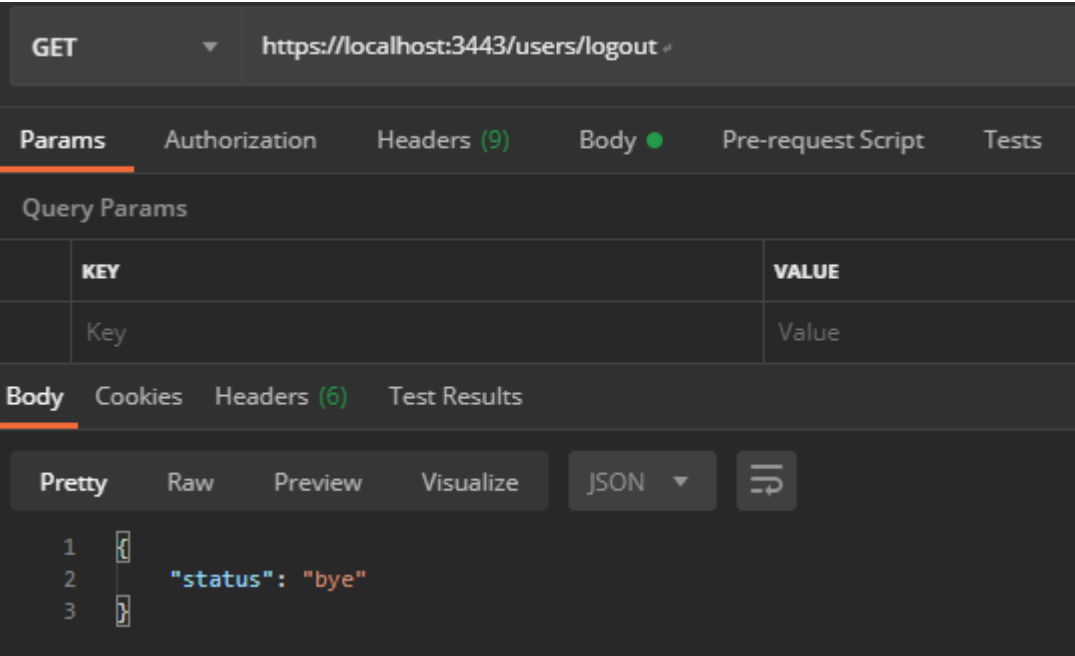


Votar POST /votes/userId

`{"comment": "ok"}`



Salir GET `/logout`



Lista de usuarios

```
[
  {
    "firstname": "adminis",
    "lastname": "trator",
    "area": "",
    "admin": true,
    "_id": "5effbb3ad42bd13340395780",
    "username": "admin",
    "__v": 0
  },
  {
    "firstname": "Pablo",
    "lastname": "Landeta",
    "area": "Motivation",
    "admin": false,
    "_id": "5effbc6dd42bd13340395781",
    "username": "pablo",
    "__v": 0
  },
  {
```

```

    "firstname": "Juan",
    "lastname": "Cevallos",
    "area": "Team Player",
    "admin": false,
    "_id": "5effbe5ce8fa7d06381cd183",
    "username": "juan",
    "__v": 0
  },
  {
    "firstname": "Jorge",
    "lastname": "Aldaz",
    "area": "Fun",
    "admin": false,
    "_id": "5f000432efbd5325d4c13a60",
    "username": "jorge",
    "__v": 0
  }
]

```

GET <https://localhost:3443/users>

<input checked="" type="checkbox"/>	Accept	*/*
<input checked="" type="checkbox"/>	Accept-Encoding	gzip, deflate, br
<input checked="" type="checkbox"/>	Connection	keep-alive
<input checked="" type="checkbox"/>	Authorization	bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiJ1b3R5b250IiwiaWF0IjoiMTU5OTU0MjU0In0
	Key	Value

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```

1  [
2    {
3      "firstname": "adminis",
4      "lastname": "trator",
5      "area": "",
6      "admin": true,
7      "_id": "5effbb3ad42bd13340395780",
8      "username": "admin",
9      "__v": 0
10   },
11   {
12     "firstname": "Pablo",
13     "lastname": "Landeta",
14     "area": "Motivation",
15     "admin": false,
16     "_id": "5effbc6dd42bd13340395781",

```

Votos

```

[
  {
    "_id": "5f0006ee08e8462c2c8854db",
    "comment": "nice",
    "elector": "5effbb3ad42bd13340395780",
    "elected": "5effbe5ce8fa7d06381cd183",
    "month": "july",
    "area": "Motivation",
    "createdAt": "2020-07-04T04:34:54.841Z",
    "updatedAt": "2020-07-04T04:34:54.841Z",

```

```

    "__v": 0
  }
]

```

GET <https://localhost:3443/votes>

Key	Value
Accept	*/*
Accept-Encoding	gzip, deflate, br
Connection	keep-alive
Authorization	bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiJ1b3R5IiwiaWF0IjoiMTY1MjY0MjY0In0.eyJfaWQiOiJ1b3R5IiwiaWF0IjoiMTY1MjY0MjY0In0.

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```

1  [
2    {
3      "_id": "5f0006ee08e8462c2c8854db",
4      "comment": "nice",
5      "elector": "5effbb3ad42bd13340395780",
6      "elected": "5effbe5ce8fa7d06381cd183",
7      "month": "july",
8      "area": "Motivation",
9      "createdAt": "2020-07-04T04:34:54.841Z",
10     "updatedAt": "2020-07-04T04:34:54.841Z",
11     "__v": 0
12   }
13 ]

```

2.3 Dependencias

Se ha utilizado la herramienta NPM para las dependencias:

```

"dependencies": {
  "cookie-parser": "~1.4.4",
  "cors": "^2.8.5",
  "debug": "~2.6.9",
  "express": "~4.16.1",
  "http-errors": "~1.6.3",
  "jade": "~1.11.0",
  "jsonwebtoken": "^8.5.1",
  "mongoose": "^5.9.19",
  "mongoose-currency": "^0.2.0",
  "morgan": "~1.9.1",
  "passport": "^0.4.1",
  "passport-jwt": "^4.0.0",
  "passport-local": "^1.0.0",
  "passport-local-mongoose": "^6.0.1"
}

```

2.4 Base de datos

Se ha utilizado la Base de Datos MongoDB

Nombre de la base de datos: votoEmpleados

3 Arquitectura

La arquitectura se basa en MVC, ya que se ha creado los modelos que están basados en esquemas, los controladores que están basados en rutas, y las vistas son los diferentes endpoints creados para comprobar el funcionamiento de las API

3 Seguridad

Esta creado con generación de claves privadas y certificados usando Openssl, listo para ser usado en una conexión segura HTTPS.

Para el acceso a los usuarios se ha usado JSON Web Token, de esta manera es necesario ingresar el token de autorización en la cabecera de los requerimientos dependiendo del tipo de usuario que solicite un endpoint.