

# Ayudantía 8

API, Mongo y usarlo con NodeJS



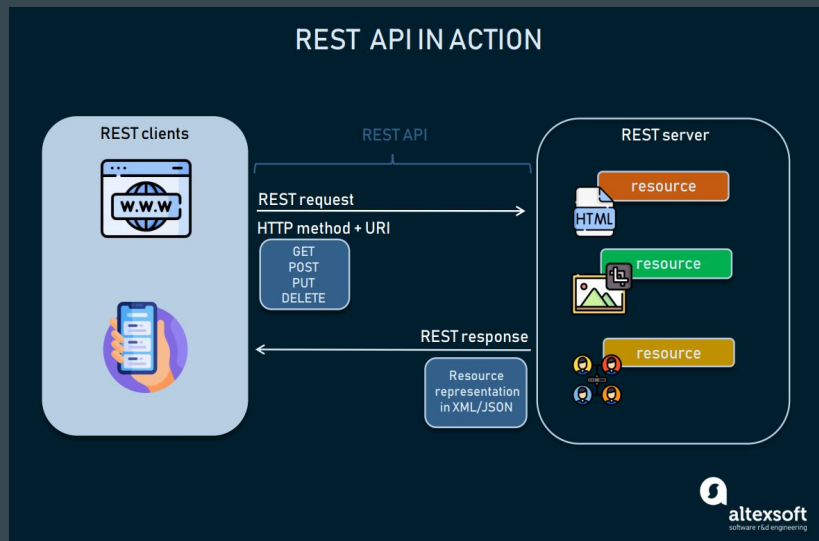
Sebastián Riquelme

# API

Anteriormente les hablé de lo que era un api, ayudanta 5.pdf :

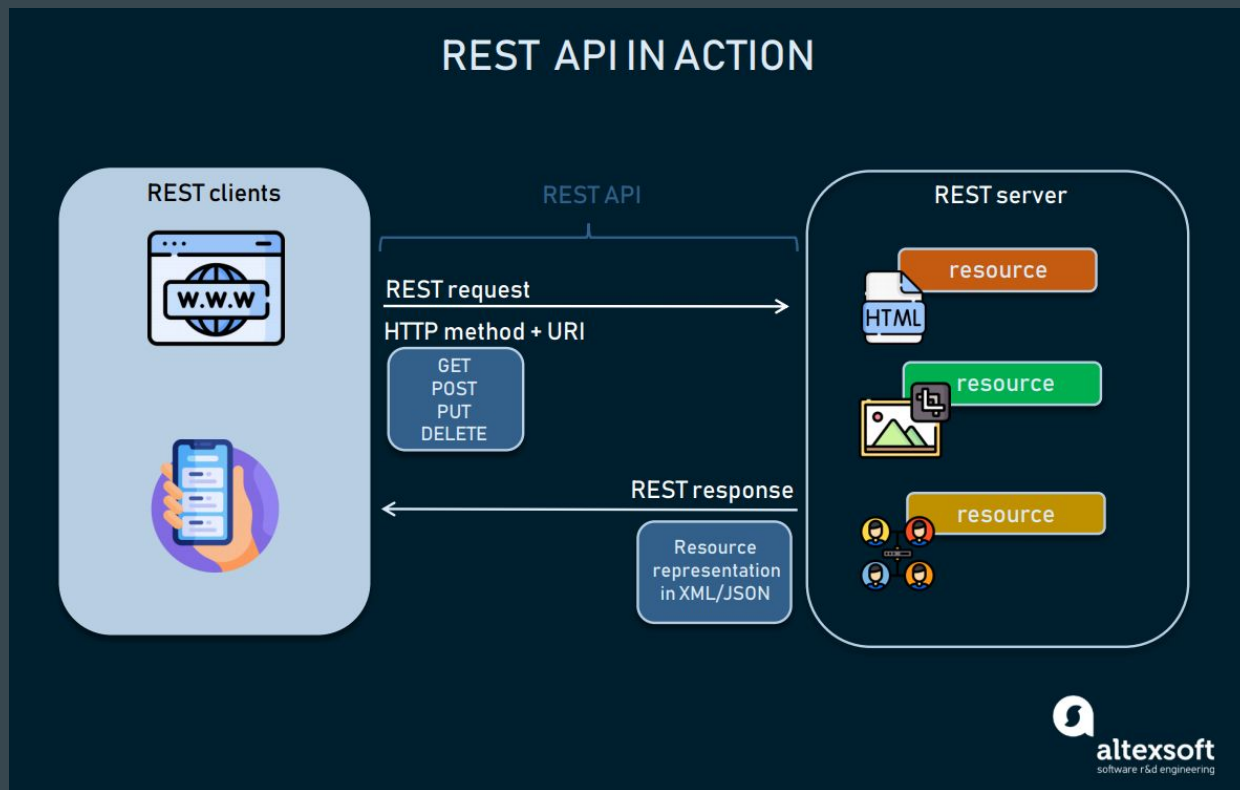
## ¿Qué es un API?

- GET = Obtener
- POST = Crear
- PUT = Actualizar
- DELETE = Eliminar



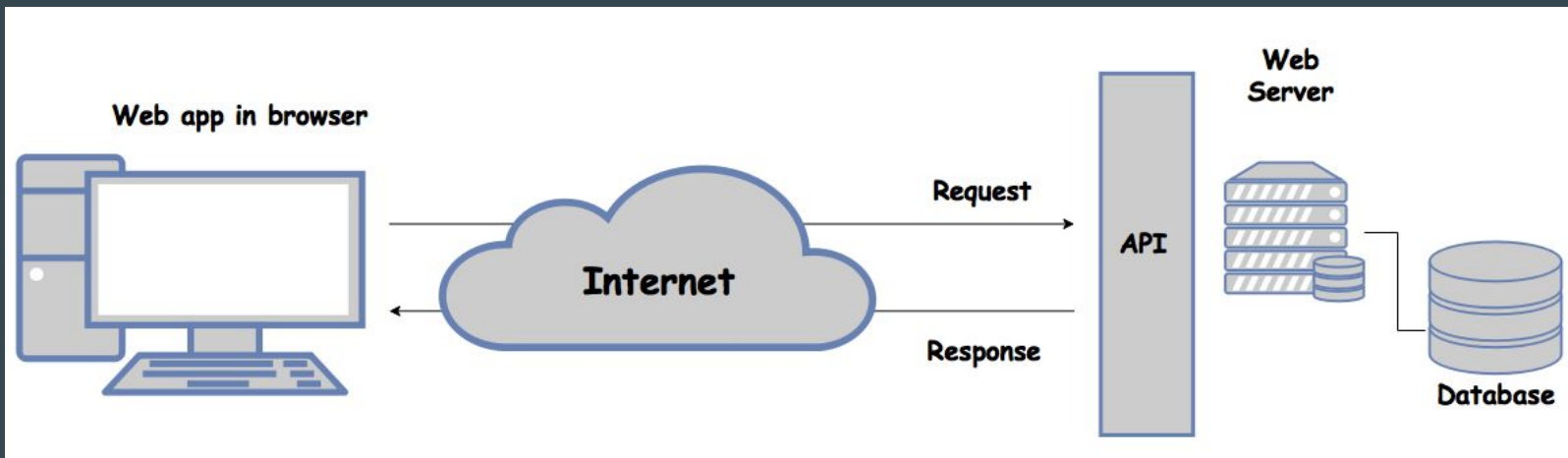
# ¿Qué es un API?

- GET = Obtener
- POST = Crear
- PUT = Actualizar
- DELETE = Eliminar



# En palabras simples

Básicamente es un sistema de intercambio de datos entre un servidor y un cliente.



# Ejemplo en NodeJS

API básica hecha en nodeJS

con express

Console.log se muestra en el servidor y res se envía al cliente.



```
1  const express = require('express');
2
3  const app = express();
4
5  //app.use(cors());
6  app.use(express.json());
7
8  app.get('/', (req, res) => {
9      console.log("API funcionando")
10     res.send("API funcionando!");
11 });
12
13 app.post('/formulario', (req, res) => {
14     console.log("API recibió: ", req.body.nombre)
15     res.status(200).json({ success: true });
16 });
17
18
19 app.listen(3001, () => {
20     console.log("\nServer corriendo en puerto: 3001\n");
21 });
```

# MongoDB

Mongo es un sistema de bases de datos de código abierto no sql, que guarda los datos en documentos parecidos a JSON.



```
1 {
2   "address": {
3     "building": "1007",
4     "coord": [ -73.856077, 40.848447 ],
5     "street": "Morris Park Ave",
6     "zipcode": "10462"
7   },
8   "borough": "Bronx",
9   "cuisine": "Bakery",
10  "grades": [
11    { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
12    { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
13    { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
14    { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
15    { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
16  ],
17  "name": "Morris Park Bake Shop",
18  "restaurant_id": "30075445"
19 }
```

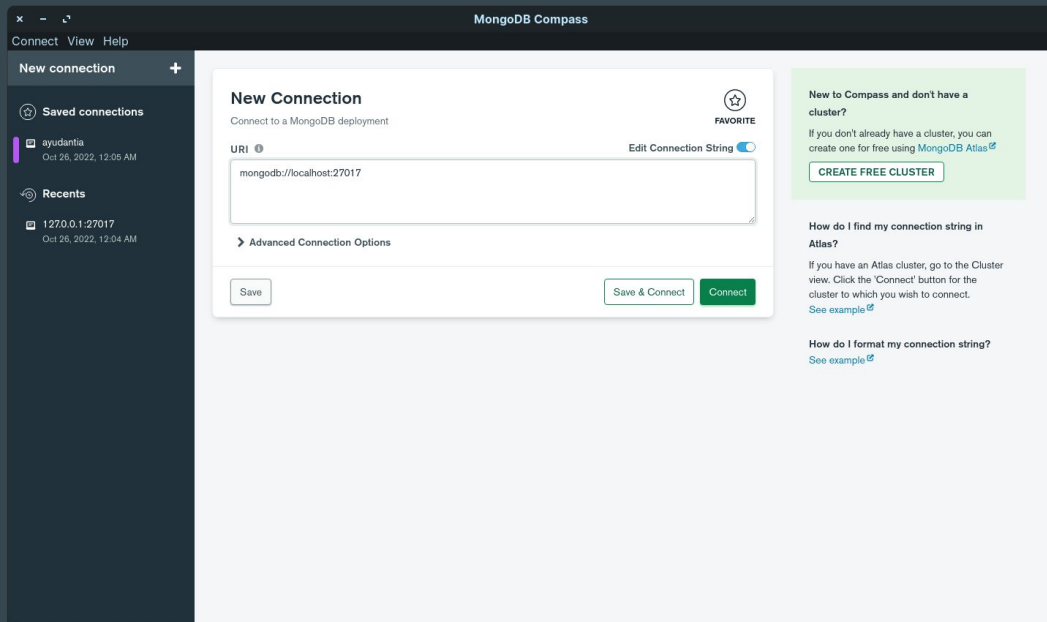
# Instalar MongoDB

Les recomiendo seguir esta guía de la documentación oficial:

<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/>

# Compass para MongoDB (Opcional)

Compass es un administrador de bases de datos para MongoDB, es una interfaz gráfica para un uso más cómodo e intuitivo.

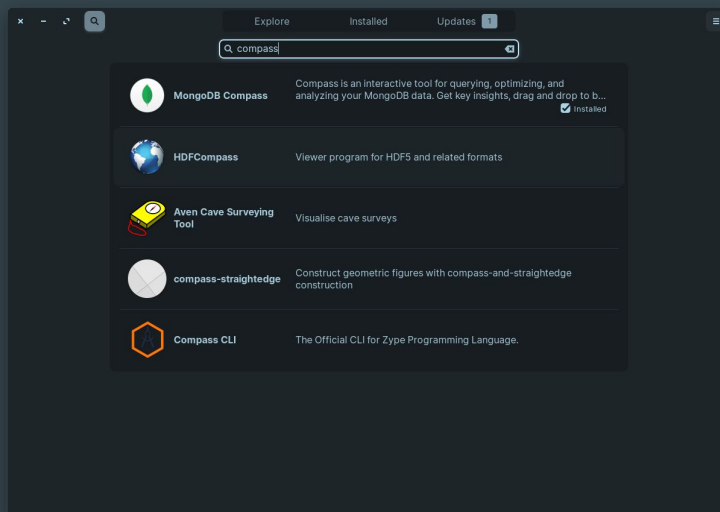




# Descargar Compass

Pueden descargarlo desde la web oficial de MongoDB, está disponible para Linux, Windows y Mac:

<https://www.mongodb.com/docs/compass/current/install/>



En mi caso lo descargué de la tienda de mi linux

# Conectar compass a nuestro MongoDB

Ejecutamos el comando “mongosh” y copiamos la URI en Compass:

```
seba@seba-thinkpad-x1-carbon: ~/Desktop/Ayudantia/test_vps
>
mongosh

Current Mongosh Log ID: 6358a32122fcd2b2ef4e953b
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeout
MS=2000&appName=mongosh+1.6.0
Using MongoDB:      6.0.2
Using Mongosh:      1.6.0

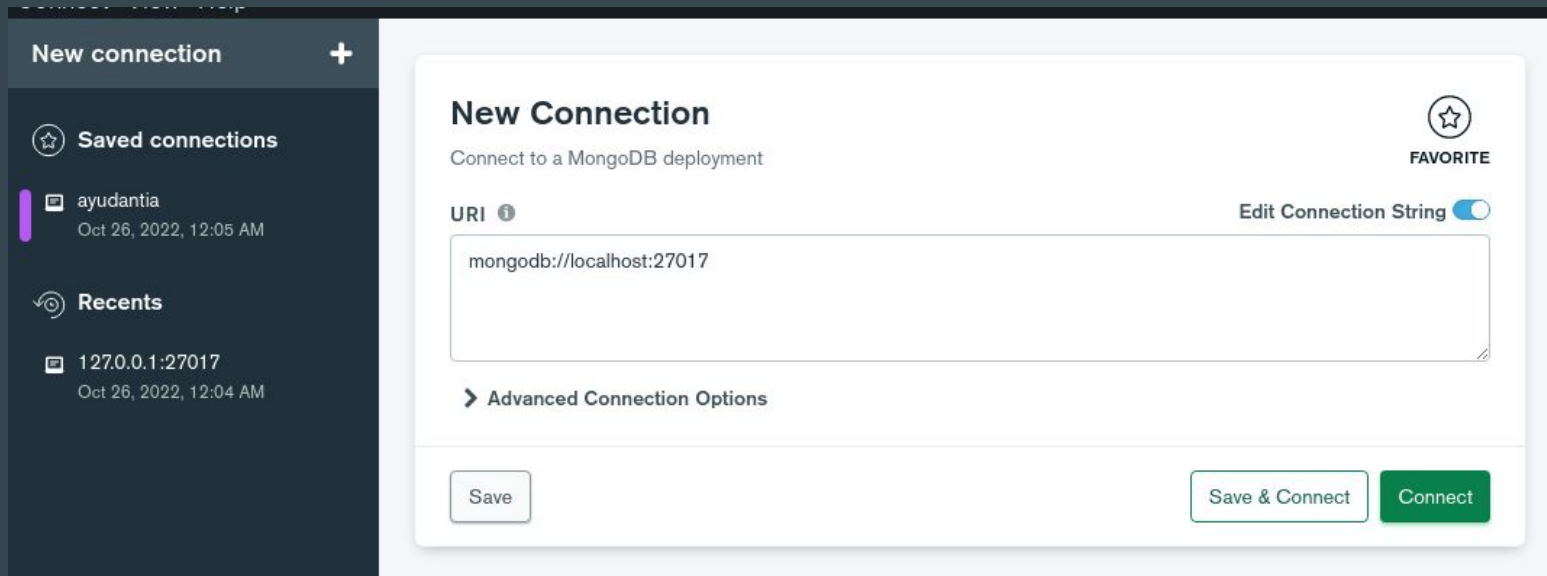
For mongosh info see: https://docs.mongodb.com/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
2022-10-26T00:01:47.899-03:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-10-26T00:01:48.655-03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2022-10-26T00:01:48.656-03:00: vm.max_map_count is too low
-----
```

# En compass

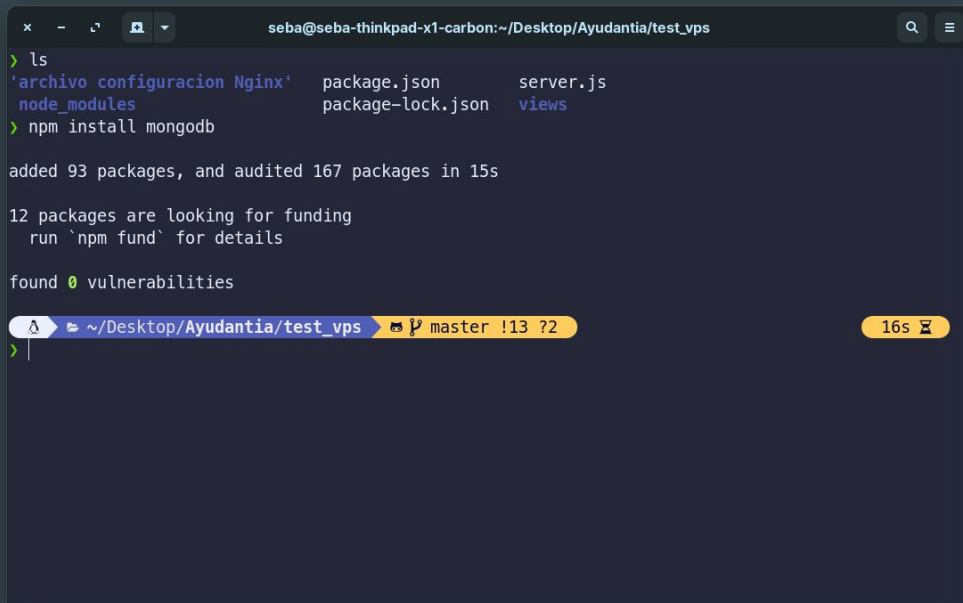
Pueden pegar la URI y dar en Save & conect para guardar la conexión como se ve a la izquierda:



# Añadir MongoDB a NodeJS

En el directorio de nuestro proyecto instalamos:

```
npm install mongodb
```



```
seba@seba-thinkpad-x1-carbon: ~/Desktop/Ayudantia/test_vps
> ls
'archivo configuracion Nginx'  package.json      server.js
node_modules                  package-lock.json  views
> npm install mongodb

added 93 packages, and audited 167 packages in 15s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

The terminal window shows the execution of `npm install mongodb`. It lists the files in the current directory, including `package.json`, `package-lock.json`, `server.js`, `views`, and the `node_modules` directory. The output of the command shows that 93 packages were added and 167 packages were audited in 15 seconds. It also mentions that 12 packages are looking for funding and that no vulnerabilities were found.

# Esto agrega el paquete a nuestro proyecto en package.js

```
1  {
2    "name": "express-handlebars-basic-example",
3    "version": "0.0.0",
4    "description": "",
5    "private": true,
6    "main": "server.js",
7    "type": "module",
8    "scripts": {
9      "test": "echo \"Error: no test specified\" && exit 1",
10     "start": "node server.js"
11   },
12   "author": "Eric Ferraiuolo <eferraiuolo@gmail.com>",
13   "license": "BSD",
14   "dependencies": {
15     "express": "^4.17.1",
16     "express-handlebars": "^6.0.6",
17     "mongodb": "^4.11.0"
18   }
19 }
20
```

# Importarlo a nuestro server.js

Lo veremos en vivo en ayudantía y quedará en una de las carpetas vps...

