public_API

Public API is a basic and easy to use API developed with front-end-development in mind. The API will allow front end developers to test their apps during development phase.

Requirements

To run the API the user will need NODE.JS, MongoDB and Bash(or any other console);

Install Node.JS, MongoDB and GIT

Git and Node.JS do not require any special configuration to install. Just follow the links. Download and install. (restart your pc after).

- GIT: https://gitforwindows.org/
- Node: https://nodejs.org/en/ (Download LTS version)

Mongo requires some configuration.

- Download Link: https://www.mongodb.com/try/download/community (select your platform and Download)

Suggested videos:

https://www.youtube.com/watch?v=FwMwO8pXfq0
https://www.youtube.com/watch?v=Ph1Z97X6xno
https://www.youtube.com/watch?v=BwVOIRX3VXk

Clone and Initialize the app.

1 - Open Bash on the folder you want to save your repository. Copy the following lines one by one and paste them on Bash("ctrl + v" wont work. You will have to use the right button of the mouse and select paste).

```
$ mkdir public_api
```

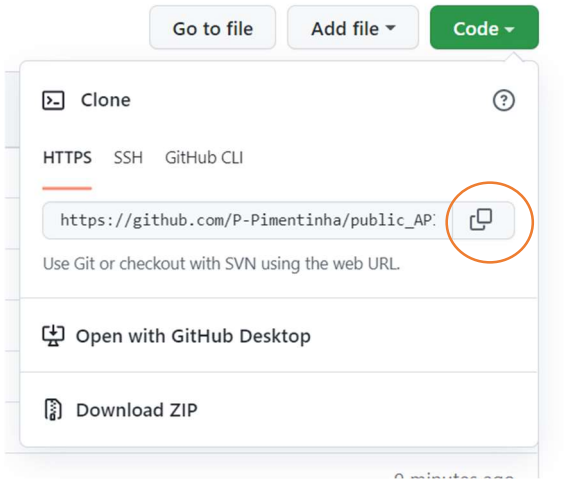
mkdir public_api

```
$ cd public_api
```

cd public_api

2 - Navigate to the Link: github.com/P-Pimentinha/public_API

3 - Copy the link to clone the repository



4 - Head to Bash again and paste the following commands

```
$ git clone https://github.com/P-Pimentinha/public_API.git
```

git clone https://github.com/P-Pimentinha/public_API.git

```
$ cd public_api
```

cd public_api

```
$ npm install
```

npm install

5 – Initialize the Server.

```
$ node server.js
```

node server.js

Your server is now up and running on port 8000;

Routes

Get - http://localhost:8000/api/users - returns all the users.
Post(1) - http://localhost:8000/api/users - creates a new user.
Put - http://localhost:8000/api/users/:id – updates any field of the user object.
Delete - http://localhost:8000/api/users/:id – deletes the user with the passed ID

Post(1):
Required:

- user_name (string) // UserNames are Unique values.
- name (string)
- age (number)
- address (string)

Populating Mongo

While the server is running you can use the following objects to populate the database. (You can use Postman).

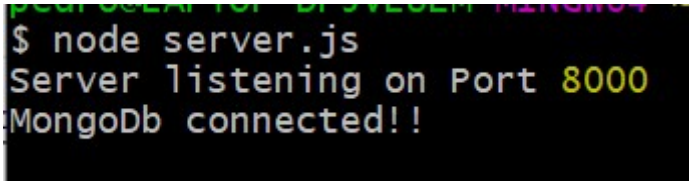
```
{
    "user_name": "Sakore",
    "name": "noName",
    "age": 21,
    "address": "far far away"
}

{
    "user_name": "Narotu",
    "name": "noName",
    "age": 21,
    "address": "far far away"
}

{
    "user_name": "Sasoki",
    "name": "noName",
    "age": 21,
    "address": "far far away"
}

{
    "user_name": "Etache",
    "name": "noName",
    "age": 21,
    "address": "far far away"
}

{
    "user_name": "Laara",
    "name": "noName",
    "age": 21,
    "address": "far far away"
}

{
    "user_name": "Yoje Etadure",
    "name": "noName",
    "age": 21,
    "address": "far far away"
}
```

Postman

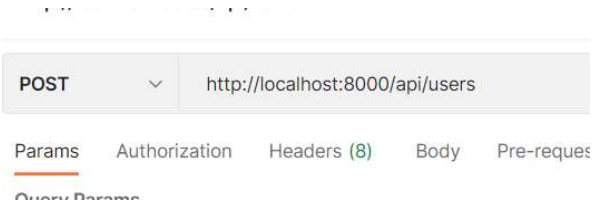
If you have never user Postman before you can check lesson 3 and 4 on the following video:
https://www.youtube.com/watch?v=VywxIQ2ZXw4

Populating the database:

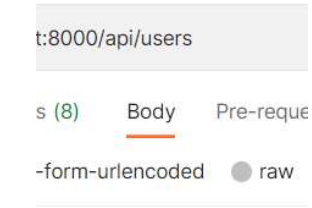
While the server is running Open Postman



Go to your workspace select post and paste the following url:
http://localhost:8000/api/users

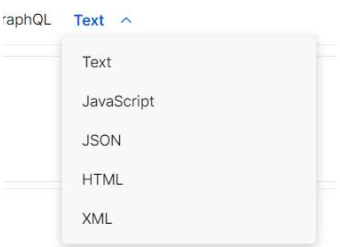


Select Body



Raw

Select JSON



Using the objects above populate the database by pasting them one by one and clicking on sending. You are free to choose any values you want. THE USERNAME has a UNIQUE value.