Umut Zeren

Cloud Computing Hw1 Documentation

Used Cloud Service: Cloud mongo dB Atlas Cluster

Cluster Tier: AWS /MO Sandbox Germany Frankfurt eu-central-1

Unique Record Id's provided by mongoDb everytime we post a record.

MongoDb Schema:

{_id:int, username:String, password:String,date:Date,name:String,surname,String}

Test Environment:

Versions of libraries data can be seen in package.json

Bakcend app written in JavaScript

Following libraries are used: Express: to manage call such as post,delete,get,patch,post

Mongoose: to iteract with cloud mongoDb,

body-parser:to parse request bodies,

dotenv: to hide database connection string {password etc} as a global variable

nodemon:to restart script everytine the code has changed while coding in visual studio

Scripts:

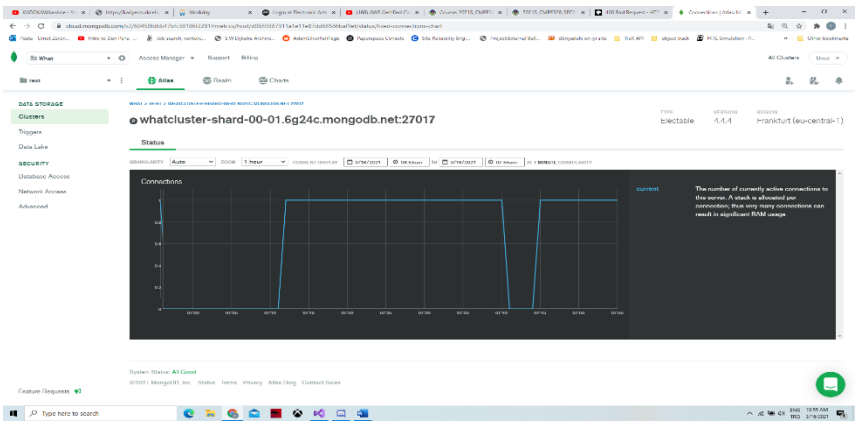
MongoModels/Post.js:To design MongoDb schema

App.js: Main script that can access rotues and imports other scripts when neccasariy

Starts local server : Example: require('./routes/posts');
Makes db connection using mongoose.connect(process.env.DB_Connection, at line ,32-33

Roots/posts.js: makes routes like post,get,patch,put
.env: file for database connection string to hide in code in commercial use
Package.json: contains used libraries and their versions

Connection made ss:



Testing:

Postman is an option but my tests shows that command line curl like testing is more healthy. Therefore I suggest using **httpi** : https://httpie.io/ best use of this app is by downloading as command line tool.

If you have python package manager; “pip install httpie” would do the work.

Or if you on a linux machine can use apt-get install httpie

On macos: brew install httpie

Note: I made db ip authorization to all ips so if you can connect to db in some way from its gui you can also see changing data.

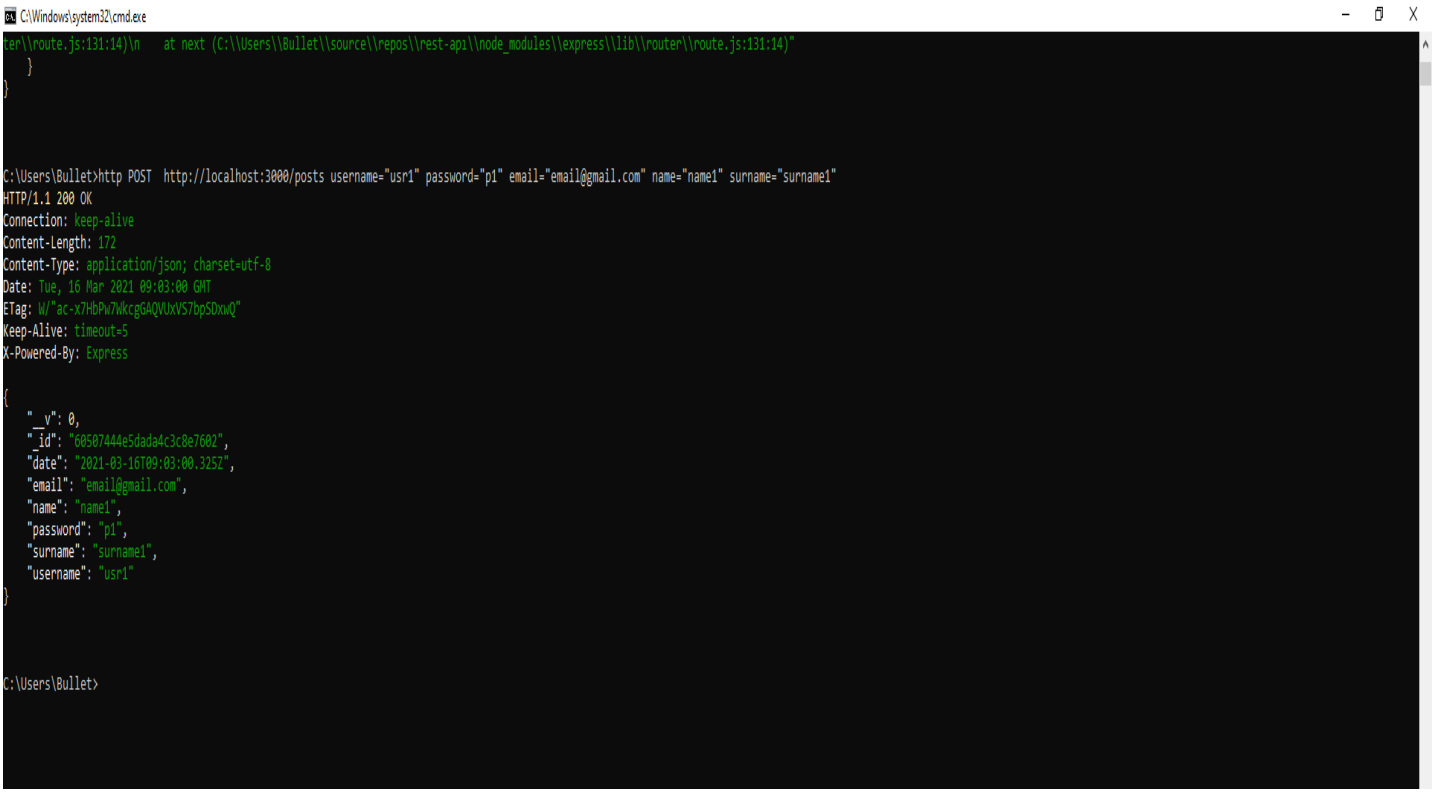
To test Functionality:

Copy paste this example requests after starting the server via double click app.js or if you use visual studio code write “npm start” to terminal while in the project directory

POST: post a user

Example test command: http POST http://localhost:3000/posts username="usr1" password="p1" email="email@gmail.com" name="name1" surname="surname1"

Zoom to see more precisely.



GET: get all posts

```
C:\Windows\system32\cmd.exe
C:\Users\Bullet>http GET  http://localhost:3000/posts
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 348
Content-Type: application/json; charset=utf-8
Date: Tue, 16 Mar 2021 09:06:44 GMT
ETag: W/"15c-awnFrgZ94FPbC7KRJ2sOv5xZm20"
Keep-Alive: timeout=5
X-Powered-By: Express

[
    {
        "__v": 0,
        "_id": "60507444e5dada4c3c8e7602",
        "date": "2021-03-16T09:03:00.325Z",
        "email": "email@gmail.com",
        "name": "name1",
        "password": "p1",
        "surname": "surname1",
        "username": "usr1"
    },
    {
        "__v": 0,
        "_id": "605074b3e5dada4c3c8e7603",
        "date": "2021-03-16T09:04:51.271Z",
        "email": "email2@gmail.com",
        "name": "name2",
        "password": "p2",
        "surname": "surname2",
        "username": "usr2"
    }
]
```

Get spesific post with the **user name**:

http GET http://localhost:3000/posts/usr1

```
C:\Windows\system32\cmd.exe
C:\Users\Bullet>http GET  http://localhost:3000/posts/usr1
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 174
Content-Type: application/json; charset=utf-8
Date: Tue, 16 Mar 2021 09:13:49 GMT
ETag: W/"ae-ZNPEekuwZaoy8+ADmsPTtpq7qwQ"
Keep-Alive: timeout=5
X-Powered-By: Express

[
  {
    "__v": 0,
    "_id": "60507444e5dada4c3c8e7602",
    "date": "2021-03-16T09:03:00.325Z",
    "email": "email@gmail.com",
    "name": "name1",
    "password": "p1",
    "surname": "surname1",
    "username": "usr1"
  }
]
```

İf get request for non existing user or username server returns status 404

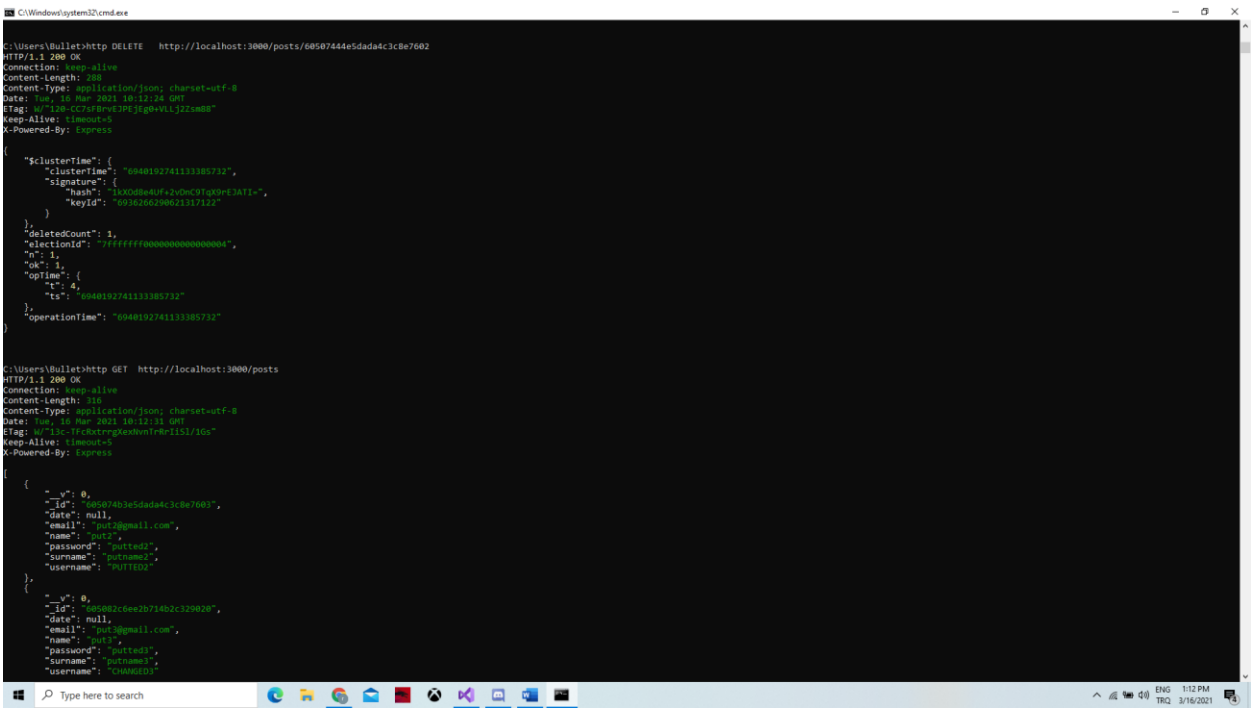
PUT: http PUT http://localhost:3000/posts/605082c6ee2b714b2c329020 username="CHANGED3" password="putted3" email="put3@gmail.com" name="put3" surname="putname3"

Before:

After put:

DELETE: Now I delete the first record with the id “60507444e5dada4c3c8e7602" from the ss above we continue.

http DELETE http://localhost:3000/posts/60507444e5dada4c3c8e7602



As you can see first records name was put1 and now its put2 because first record is deleted.

PATCH:

Now I will update just the name attribute of the record with id “605074b3e5dada4c3c8e7603” which its name is put2 wright now.

After patch its name is “CloudCompute”

http PATCH http://localhost:3000/posts/605074b3e5dada4c3c8e7603 name=”CloudCompute”