

## About the system\_

The system consists of an API where the client will have access to methods that encompass the CRUD.

The information entered into the system through the insert methods will be persisted in a MongoDB cloud database and this data can be accessed again through queries performed through the API access methods.

## Instructions for consuming the API service\_

### Possible routes to do requests:

`route.post('/', controller.postElement); => http://localhost:3000/products`

-add elements

`route.put('/:id', controller.put); => http://localhost:3000/products/tags/id-of-item`

-update element

`route.delete('/:id', controller.delete); => http://localhost:3000/products/ id-of-item`

-delete element

`route.get('/:slug', controller.getBySlug); => http://localhost:3000/products/slug-value`

-get product by slug

`route.get('/id/:id', controller.getById); => http://localhost:3000/products/id/id-of-item`

-getbyid

`route.get('/tags/:tag', controller.getByTag); => http://localhost:3000/products/tags/tag-value`

-get by tag

Please see the JSON structure below:

**Json sctrcture:**

```
{  
  "title": "Astsah",  
  "description": "Testing175",  
  "slug": "teting-testing175",  
  "price": "955",  
  "active": false,  
  "tags": ["Dinam", "Chexv", "vion"]  
}
```

Json object with  
all its attributes.

## Test Tools

requests tests : POSTMAN

-data persistence: mLab, studio 3t

-if you want to access the **mLab database**, download **studio 3T** and pass the following user and password credentials

- **user : uemurarafa / password: uemurarafa1**

the database will be online if you need testing. (do not forget to check the connection to the database in the app.js file)

```
//conn database  
mongoose.connect('mongodb://uemurarafa:uemurarafa1@ds263460.mlab.com:63460/node_storeage');
```

If you would like to have easier access to the mongoDB database go to <https://mlab.com/login/> and enter

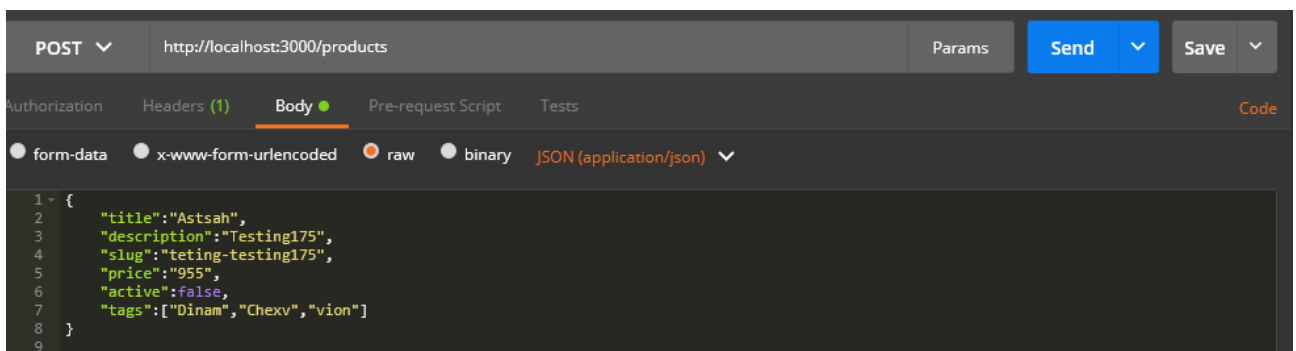
user: **uemurarafa** and password: **mlabbitcoin1**

This way you can have real-time access to the database

## Enter, read, update and delete data:

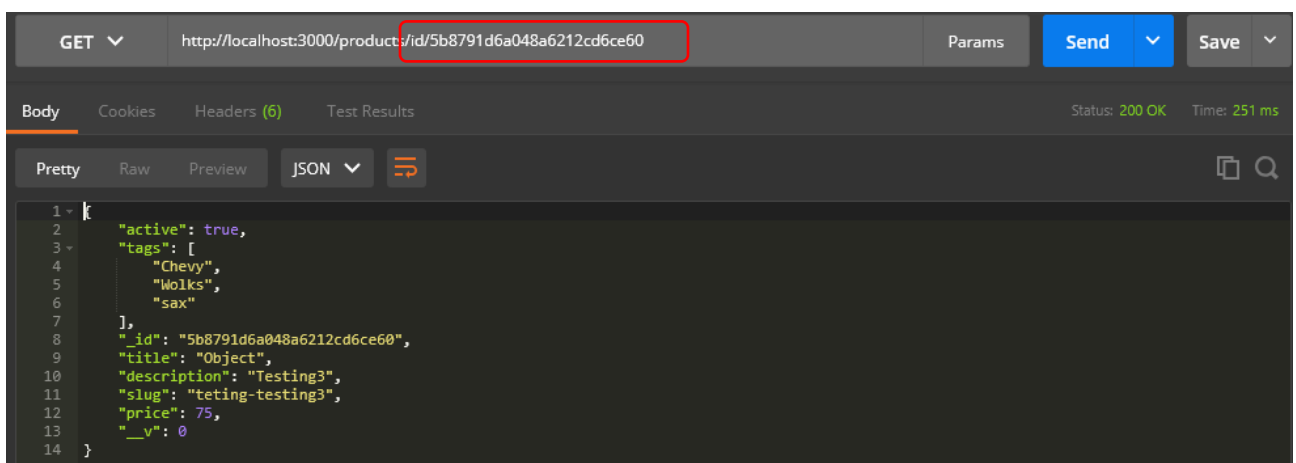
To test the API you can use the json object above to do insert operations in the mongoDB database using the post method

as the image below shows:

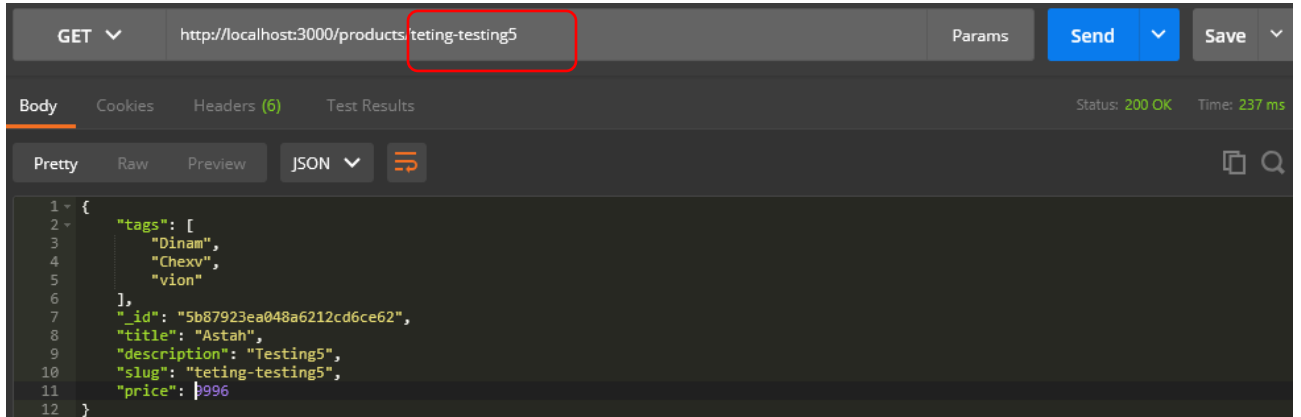


The system is configured to reproduce errors according to the requisition codes. In case of error when registering a post via registry the API will present an error

## Reading by id:



Reading by slug:



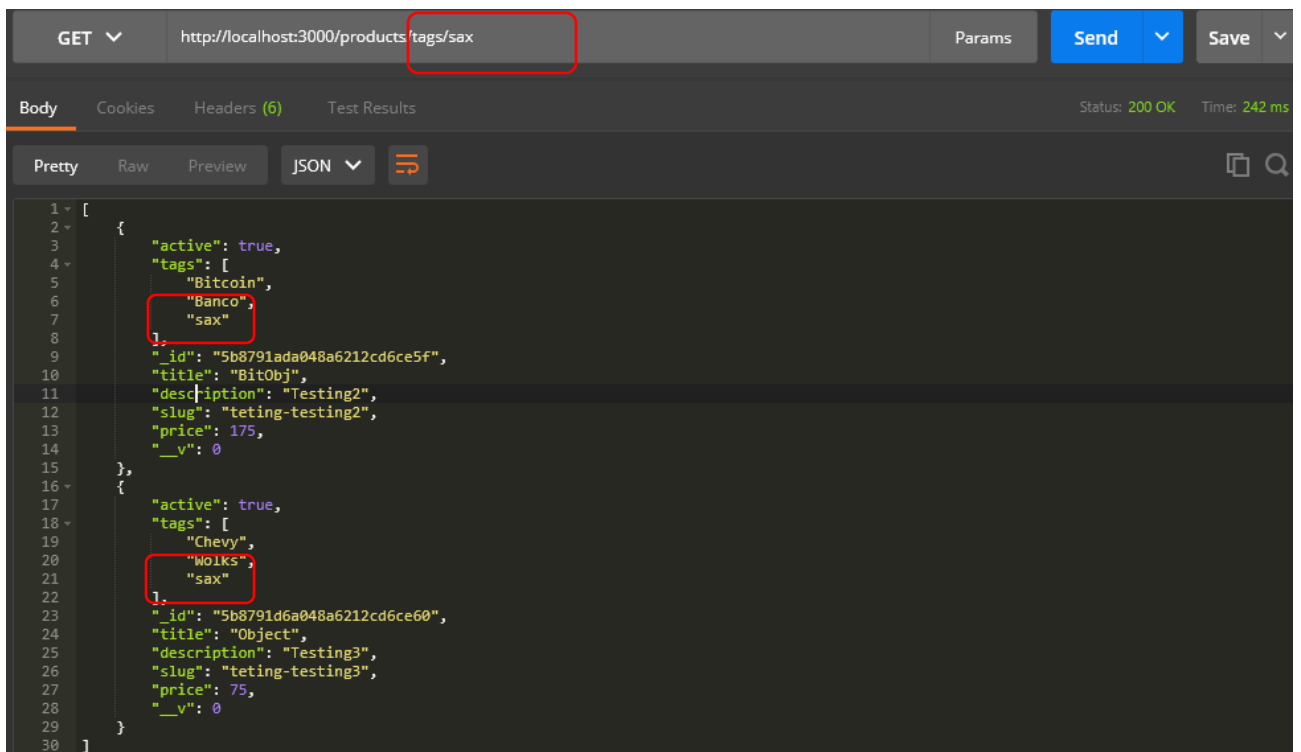
```
GET http://localhost:3000/products/teting-testing5 Params Send Save
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 237 ms

Pretty Raw Preview JSON

```
1 {
2   "tags": [
3     "Dinam",
4     "Chexv",
5     "vion"
6   ],
7   "_id": "5b87923ea048a6212cd6ce62",
8   "title": "Astah",
9   "description": "Testing5",
10  "slug": "teting-testing5",
11  "price": 996
12 }
```

Reading by tags:



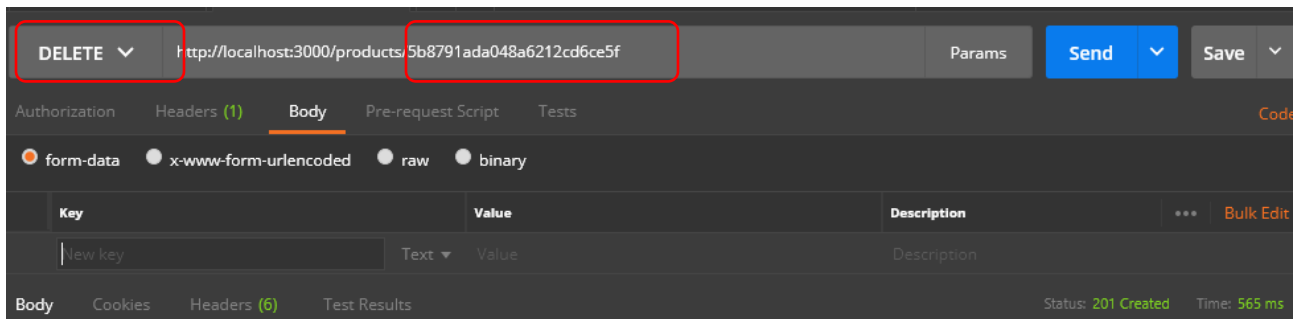
```
GET http://localhost:3000/products/tags/sax Params Send Save
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 242 ms

Pretty Raw Preview JSON

```
1 [
2   {
3     "active": true,
4     "tags": [
5       "Bitcoin",
6       "Banco",
7       "sax"
8     ],
9     "_id": "5b8791ada048a6212cd6ce5f",
10    "title": "BitObj",
11    "description": "Testing2",
12    "slug": "teting-testing2",
13    "price": 175,
14    "__v": 0
15  },
16  {
17    "active": true,
18    "tags": [
19      "Chevy",
20      "wo1ks",
21      "sax"
22    ],
23    "_id": "5b8791d6a048a6212cd6ce60",
24    "title": "Object",
25    "description": "Testing3",
26    "slug": "teting-testing3",
27    "price": 75,
28    "__v": 0
29  }
30 ]
```

Delete by id:



Updating by id:

