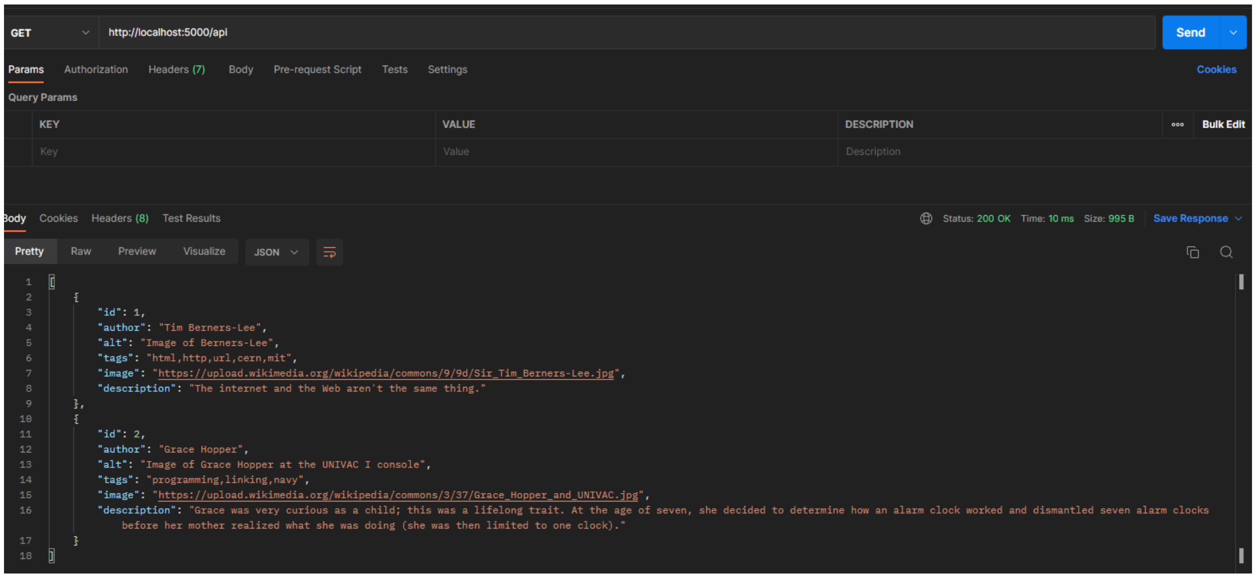
A3 Peer Review

1.Retrieving full data set

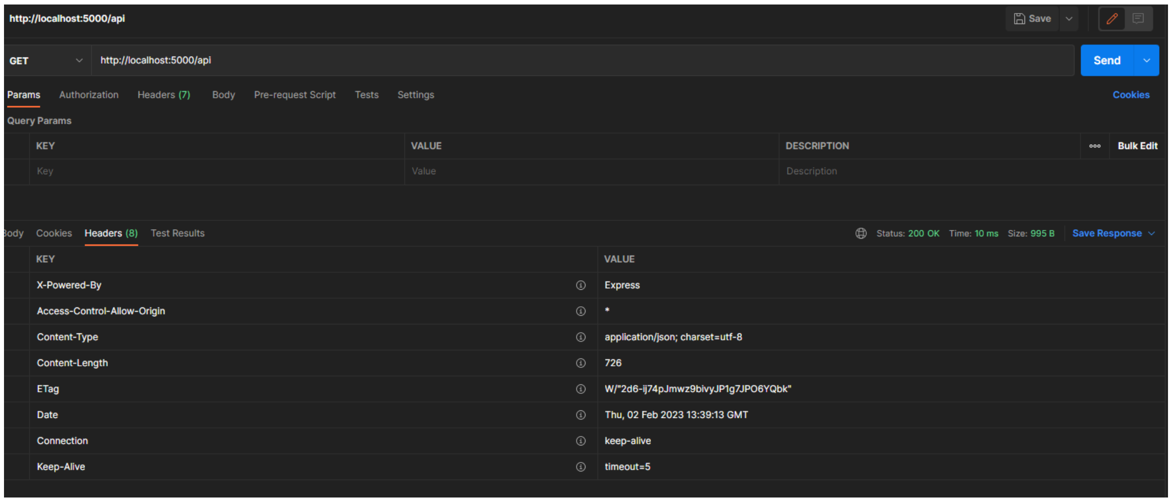


GET http://localhost:5000/api

Status: 200 OK Time: 10 ms Size: 995 B

```
[
    {
        "id": 1,
        "author": "Tim Berners-Lee",
        "alt": "Image of Berners-Lee",
        "tags": "html,http,url,cern,mit",
        "image": "https://upload.wikimedia.org/wikipedia/commons/9/9d/Sir_Tim_Berners-Lee.jpg",
        "description": "The internet and the Web aren't the same thing."
    },
    {
        "id": 2,
        "author": "Grace Hopper",
        "alt": "Image of Grace Hopper at the UNIVAC I console",
        "tags": "programming,linking,navy",
        "image": "https://upload.wikimedia.org/wikipedia/commons/3/37/Grace_Hopper_and_UNIVAC.jpg",
        "description": "Grace was very curious as a child; this was a lifelong trait. At the age of seven, she decided to determine how an alarm clock worked and dismantled seven alarm clocks
            before her mother realized what she was doing (she was then limited to one clock)."
    }
]
```

The database correctly returns all of the entries contained in the database, as well as the appropriate response code of 200. The documentation gives the address to perform this action as 'http://localhost/..../api' - it would have been more intuitive to understand if the '...' was instead replaced with something like '<port number>'.



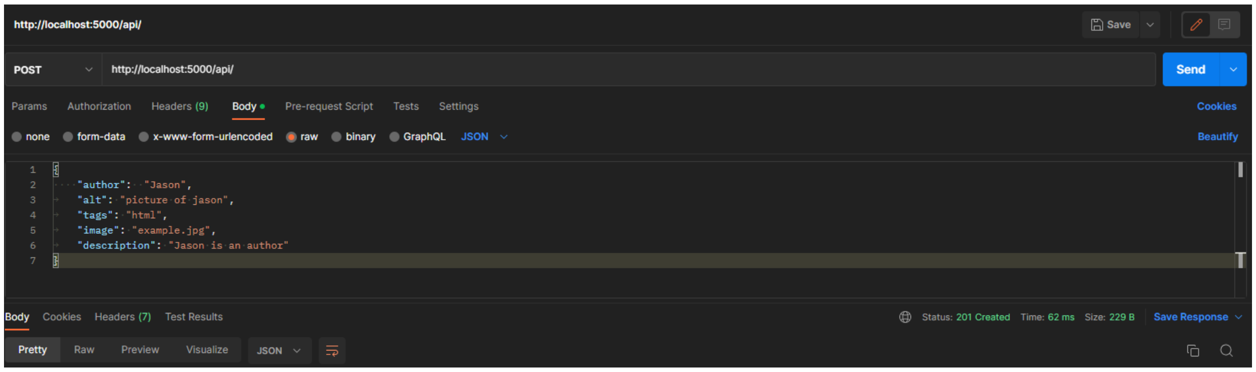
GET http://localhost:5000/api

Status: 200 OK Time: 10 ms Size: 995 B

KEY	VALUE
X-Powered-By	Express
Access-Control-Allow-Origin	*
Content-Type	application/json; charset=utf-8
Content-Length	726
ETag	W/"2d6-ij74pJmwz9bivyJP1g7JPO6YQbk"
Date	Thu, 02 Feb 2023 13:39:13 GMT
Connection	keep-alive
Keep-Alive	timeout=5

The Content-Type in the header is correctly set to application/json

2. Adding data (Create)

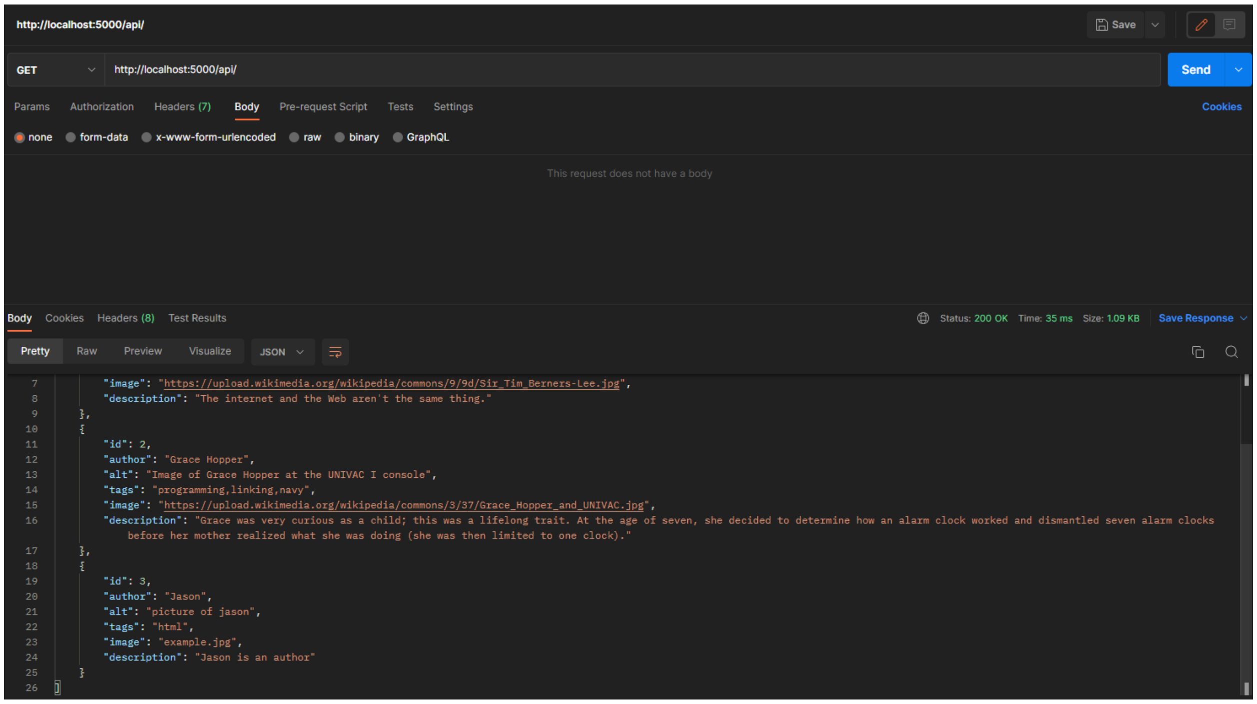


POST http://localhost:5000/api/

Body: none | form-data | x-www-form-urlencoded | [✓ raw] | binary | GraphQL — JSON

```
{
    "author": "Jason",
    "alt": "picture of jason",
    "tags": "html",
    "image": "example.jpg",
    "description": "Jason is an author"
}
```

Status: 201 Created Time: 62 ms Size: 229 B



GET http://localhost:5000/api/

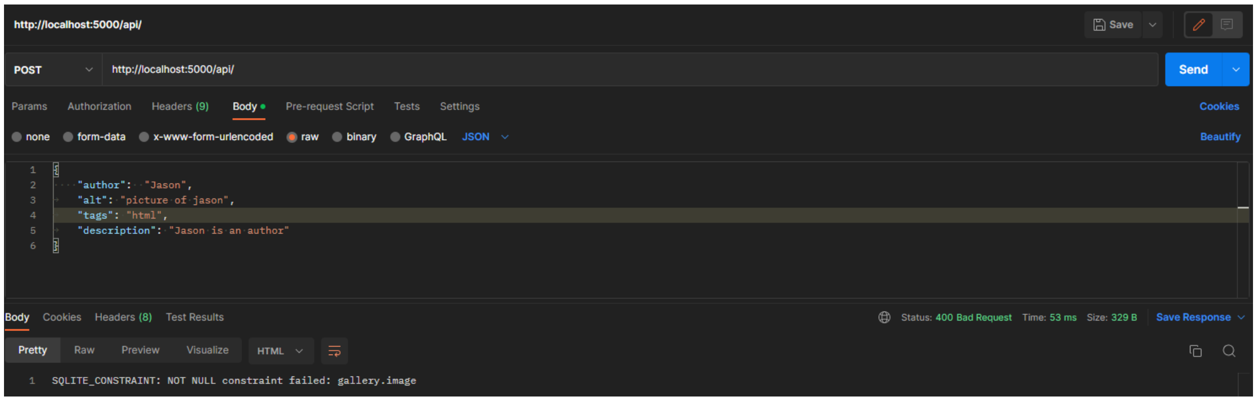
Body: [✓ none] | form-data | x-www-form-urlencoded | raw | binary | GraphQL

This request does not have a body

Status: 200 OK Time: 35 ms Size: 1.09 KB

```
            "image": "https://upload.wikimedia.org/wikipedia/commons/9/9d/Sir_Tim_Berners-Lee.jpg",
            "description": "The internet and the Web aren't the same thing."
        },
        {
            "id": 2,
            "author": "Grace Hopper",
            "alt": "Image of Grace Hopper at the UNIVAC I console",
            "tags": "programming,linking,navy",
            "image": "https://upload.wikimedia.org/wikipedia/commons/3/37/Grace_Hopper_and_UNIVAC.jpg",
            "description": "Grace was very curious as a child; this was a lifelong trait. At the age of seven, she decided to determine how an alarm clock worked and dismantled seven alarm clocks
                before her mother realized what she was doing (she was then limited to one clock)."
        },
        {
            "id": 3,
            "author": "Jason",
            "alt": "picture of jason",
            "tags": "html",
            "image": "example.jpg",
            "description": "Jason is an author"
        }
    ]
```

Upon entering all the fields and sending the request, the server gave a 201 response, and correctly added the submitted data to storage.



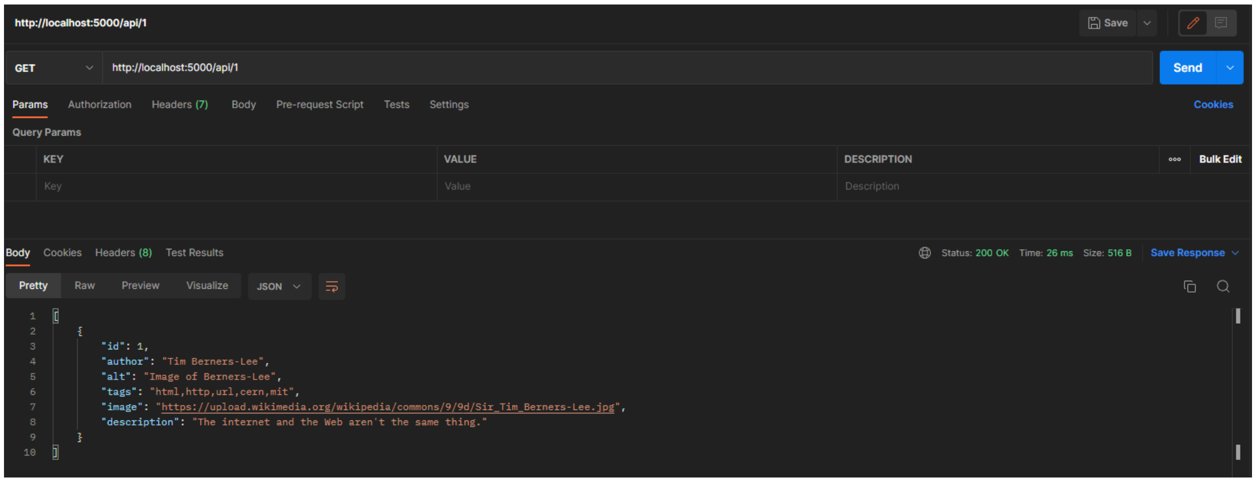
```
{
    "author": "Jason",
    "alt": "picture of jason",
    "tags": "html",
    "description": "Jason is an author"
}
```

```
SQLITE_CONSTRAINT: NOT NULL constraint failed: gallery.image
```

When a field was missing in the body, the api correctly gave a 400 error, along with a text error message.

The documentation gives an example of the JSON body, the endpoint and that the data needs to be sent as application/json - this makes this endpoint easier to understand and use.

3. Listing data for a specific item (Retrieve)



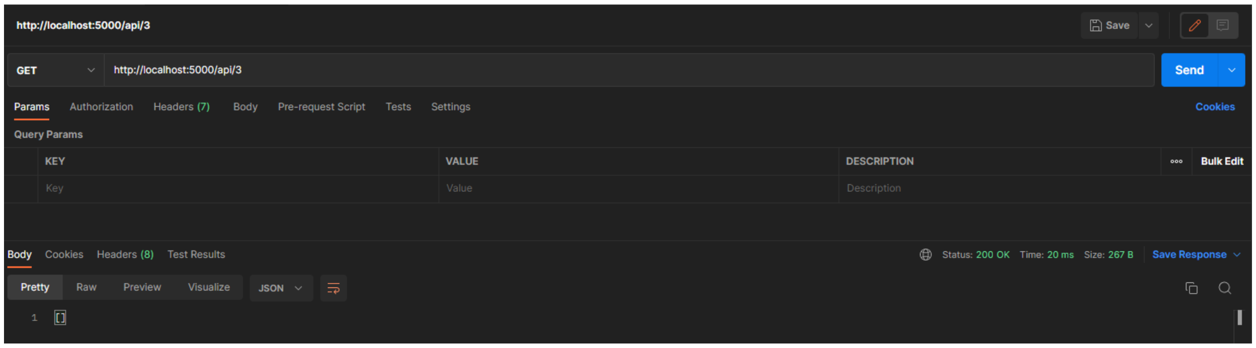
http://localhost:5000/api/1

GET http://localhost:5000/api/1

Status: 200 OK Time: 26 ms Size: 516 B

```
[
    {
        "id": 1,
        "author": "Tim Berners-Lee",
        "alt": "Image of Berners-Lee",
        "tags": "html,http,url,cern,mit",
        "image": "https://upload.wikimedia.org/wikipedia/commons/9/9d/Sir_Tim_Berners-Lee.jpg",
        "description": "The internet and the Web aren't the same thing."
    }
]
```

When the requested id is present in the database, the api correctly returns the requested data with response code 200.



http://localhost:5000/api/3

GET http://localhost:5000/api/3

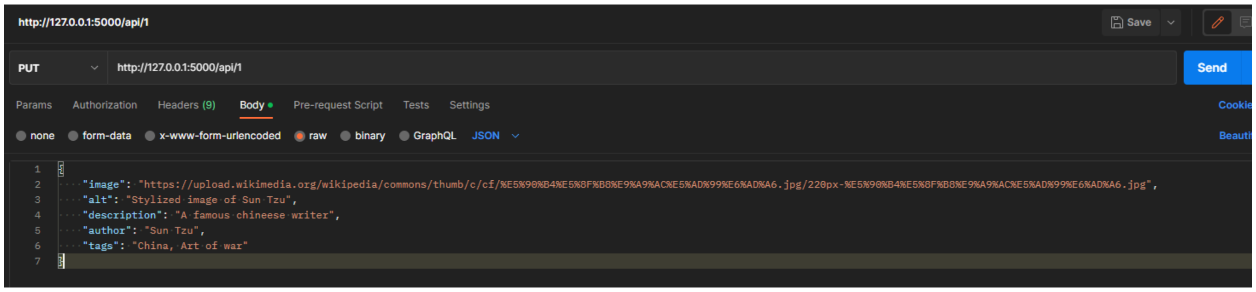
Status: 200 OK Time: 20 ms Size: 267 B

```
[]
```

When the requested id is not present in the database, the api returns an empty array with error code 200 - this endpoint does not function correctly, and should instead return a 404 error.

The documentation for this endpoint gives an example of the expected response and shows how to properly use the endpoint.

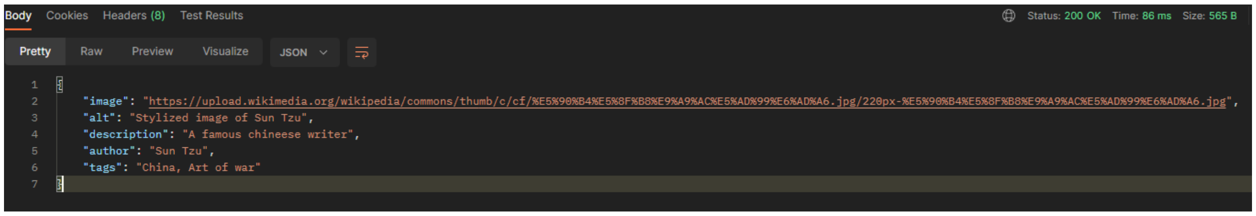
4. Changing data of specific item (Update)



http://127.0.0.1:5000/api/1

PUT http://127.0.0.1:5000/api/1

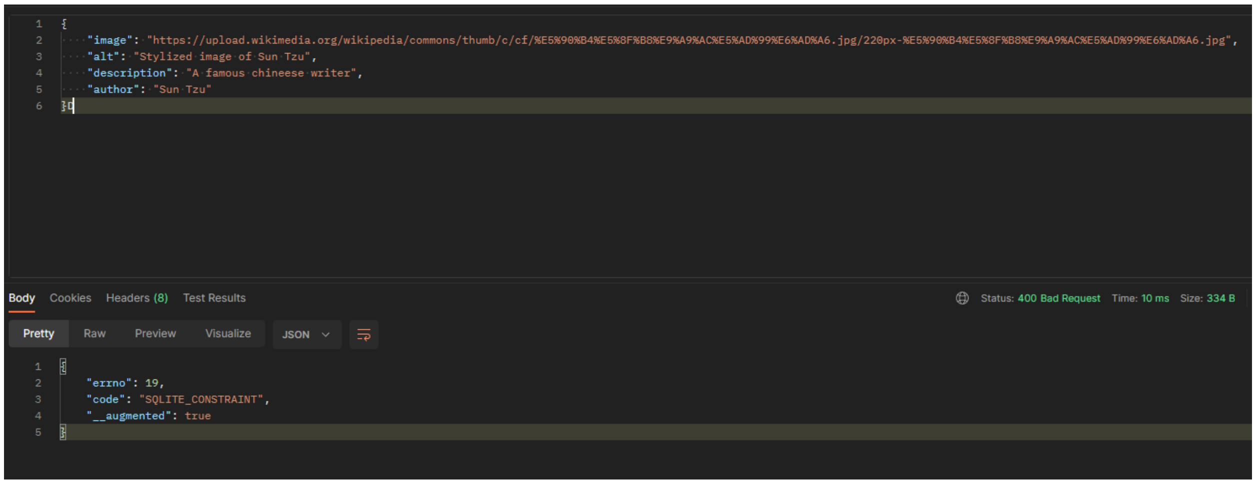
```
{
    "image": "https://upload.wikimedia.org/wikipedia/commons/thumb/c/cf/%E5%90%B4%E5%8F%B8%E9%A9%AC%E5%AD%99%E6%AD%A6.jpg/220px-%E5%90%B4%E5%8F%B8%E9%A9%AC%E5%AD%99%E6%AD%A6.jpg",
    "alt": "Stylized image of Sun Tzu",
    "description": "A famous chineese writer",
    "author": "Sun Tzu",
    "tags": "China, Art of war"
}
```



Status: 200 OK Time: 86 ms Size: 565 B

```
{
    "image": "https://upload.wikimedia.org/wikipedia/commons/thumb/c/cf/%E5%90%B4%E5%8F%B8%E9%A9%AC%E5%AD%99%E6%AD%A6.jpg/220px-%E5%90%B4%E5%8F%B8%E9%A9%AC%E5%AD%99%E6%AD%A6.jpg",
    "alt": "Stylized image of Sun Tzu",
    "description": "A famous chineese writer",
    "author": "Sun Tzu",
    "tags": "China, Art of war"
}
```

Given the expected data the put request successfully replaces the database information with the information given in the request body and returns the modified information. When presented with unexpected information the server sometimes crashes or returns a 400 error, however the response body does not contain any information about what is wrong with the request

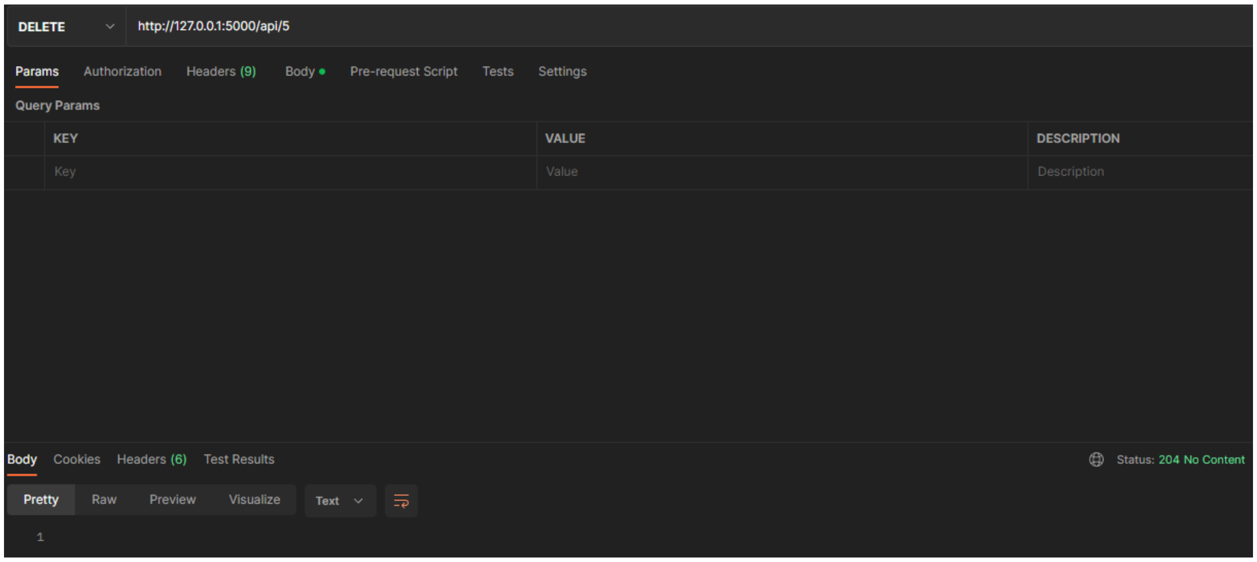


```
{
    "image": "https://upload.wikimedia.org/wikipedia/commons/thumb/c/cf/%E5%90%B4%E5%8F%B8%E9%A9%AC%E5%AD%99%E6%AD%A6.jpg/220px-%E5%90%B4%E5%8F%B8%E9%A9%AC%E5%AD%99%E6%AD%A6.jpg",
    "alt": "Stylized image of Sun Tzu",
    "description": "A famous chineese writer",
    "author": "Sun Tzu"
}
```

Status: 400 Bad Request Time: 10 ms Size: 334 B

```
{
    "errno": 19,
    "code": "SQLITE_CONSTRAINT",
    "__augmented": true
}
```

5. Removing a specific item (Delete)


DELETE http://127.0.0.1:5000/api/5

Params | Authorization | Headers (9) | Body | Pre-request Script | Tests | Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body | Cookies | Headers (6) | Test Results — Status: 204 No Content

Pretty | Raw | Preview | Visualize | Text

```
1
```

The delete requests have no body but do respond with 204 if the item has been deleted, however it will also respond with 204 if the item does not exist in the database. This makes it hard to distinguish whether the request was successfully processed on the server. Upon further inspection with the GET request the item appears to be deleted.

Bonus

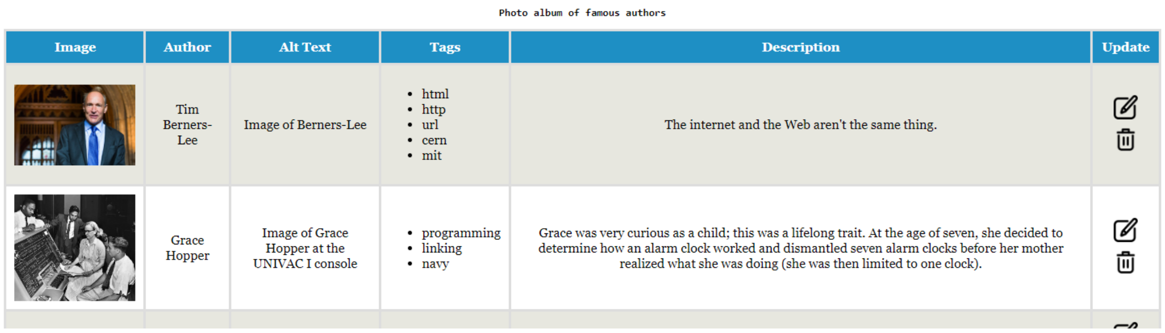


Photo album of famous authors

Image	Author	Alt Text	Tags	Description	Update
	Tim Berners-Lee	Image of Berners-Lee	• html • http • url • cern • mit	The internet and the Web aren't the same thing.	
	Grace Hopper	Image of Grace Hopper at the UNIVAC I console	• programming • linking • navy	Grace was very curious as a child; this was a lifelong trait. At the age of seven, she decided to determine how an alarm clock worked and dismantled seven alarm clocks before her mother realized what she was doing (she was then limited to one clock).	

Fields correctly loaded when page is opened

The bonus was implemented and appears to work when the server does not crash. GET, POST, PUT all work to modify the gallery data dynamically.

Actionable feedback

First, we would recommend that you review the style in which you write your documentation. Making the documentation as an HTML document would have been much easier to interpret than a comment due to all the different tags, such as <h1> and <h2> - this is because you could have utilised these tags to give your documentation more structure and make it more readable. Additionally, your inclusion of “...” to symbolise the port number is very unintuitive, and took some time to understand. Replacing it with “<port number>” or something akin to that would have greatly helped.

In addition, please test your API more thoroughly next time. A major problem we encountered is that the API would often crash after throwing a “MongooseServerSelectionError: connect ECONNREFUSED 127.0.0.1:27017” error - we are not sure if this is something unique to us, but given that we performed the expected “npm install” command, this error should not have appeared. This resulted in us having to constantly restart the API.

What we learnt

We liked that there was a brief welcome page when loading your API. In addition, despite the things we did not like about your documentation, it was more concise than our own, and thus would allow a developer to get started more quickly. We also learnt the importance of documentation for APIs, as it greatly helped us navigate around the API’s functionality.