

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: node
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

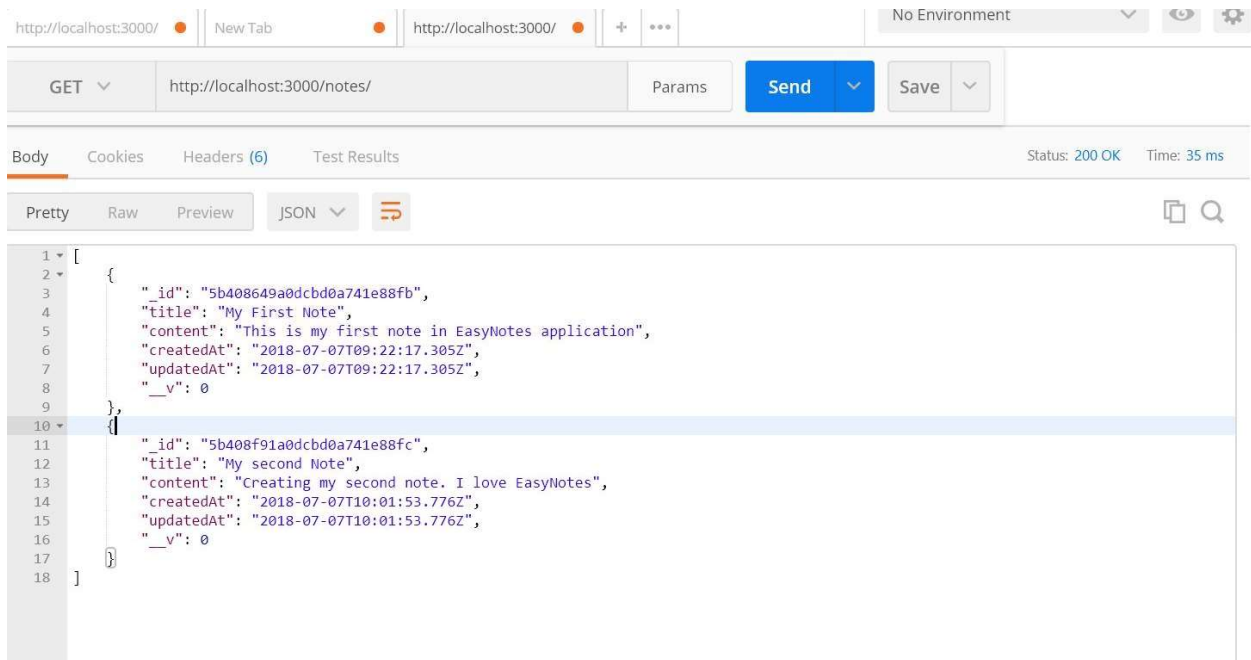
PS C:\Acadgild_fswd_21apr1\day22-jul14\assignment\noteApp> node server.js
Server is listening on port 3000
Successfully connected to the database
```

## 1. Creating a new NOTE using POST / notes API

The screenshot shows a web browser with a REST client interface. The URL bar shows `http://localhost:3000/`. The interface is set to a POST request to `http://localhost:3000/notes/`. The request body is a JSON object: `{ "title": "My First Note", "content": "This is my first note in EasyNotes application" }`. The response is a JSON object: `{ "_id": "5b408649a0dcbd0a741e88fb", "title": "My First Note", "content": "This is my first note in EasyNotes application", "createdAt": "2018-07-07T09:22:17.305Z", "updatedAt": "2018-07-07T09:22:17.305Z", "__v": 0 }`.

```
POST http://localhost:3000/notes/
Authorization Headers (1) Body Pre-request Script Tests
form-data x-www-form-urlencoded raw binary JSON (application/json)
1 [{"title": "My First Note", "content": "This is my first note in EasyNotes application"}]
Pretty Raw Preview JSON
1 {
2   "_id": "5b408649a0dcbd0a741e88fb",
3   "title": "My First Note",
4   "content": "This is my first note in EasyNotes application",
5   "createdAt": "2018-07-07T09:22:17.305Z",
6   "updatedAt": "2018-07-07T09:22:17.305Z",
7   "__v": 0
8 }
```

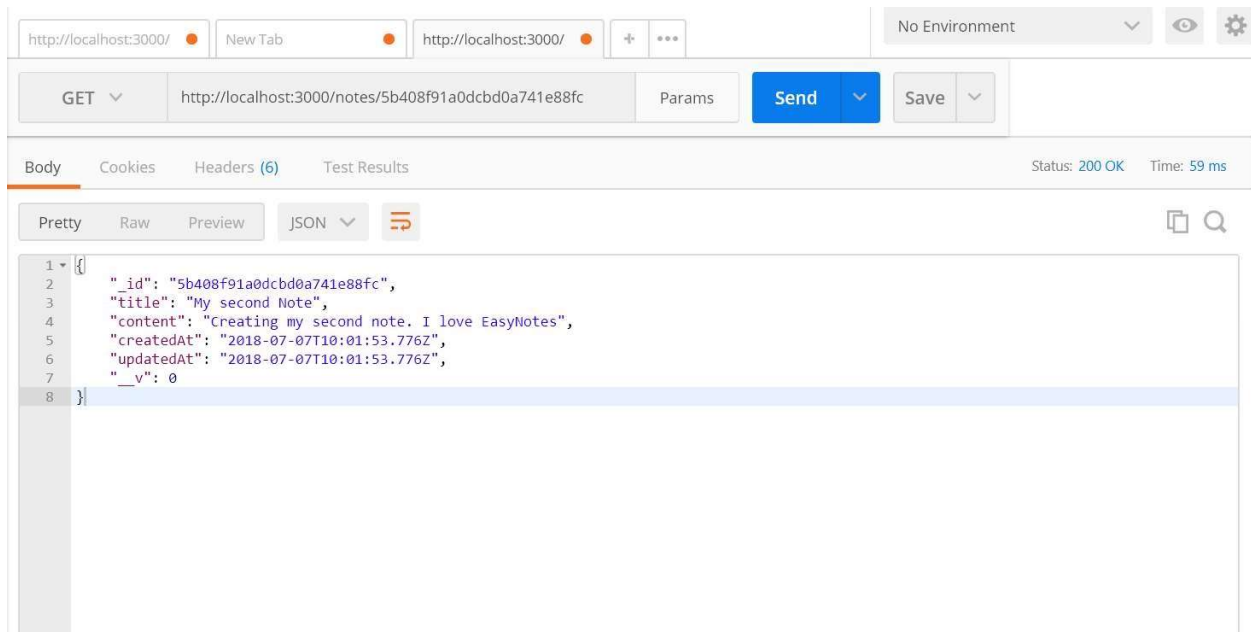
## 2. Retrieving ALL Notes using GET / notes API



The screenshot shows a REST client interface with a GET request to `http://localhost:3000/notes/`. The response is a JSON array of two notes. The status is 200 OK and the time is 35 ms.

```
1 [
2   {
3     "_id": "5b408649a0dcdbd0a741e88fb",
4     "title": "My First Note",
5     "content": "This is my first note in EasyNotes application",
6     "createdAt": "2018-07-07T09:22:17.305Z",
7     "updatedAt": "2018-07-07T09:22:17.305Z",
8     "__v": 0
9   },
10  {
11    "_id": "5b408f91a0dcdbd0a741e88fc",
12    "title": "My second Note",
13    "content": "Creating my second note. I love EasyNotes",
14    "createdAt": "2018-07-07T10:01:53.776Z",
15    "updatedAt": "2018-07-07T10:01:53.776Z",
16    "__v": 0
17  }
18 ]
```

## 3. Retrieving single Note using GET /notes/:noteId API



The screenshot shows a REST client interface with a GET request to `http://localhost:3000/notes/5b408f91a0dcdbd0a741e88fc`. The response is a single note object. The status is 200 OK and the time is 59 ms.

```
1 {
2   "_id": "5b408f91a0dcdbd0a741e88fc",
3   "title": "My second Note",
4   "content": "Creating my second note. I love EasyNotes",
5   "createdAt": "2018-07-07T10:01:53.776Z",
6   "updatedAt": "2018-07-07T10:01:53.776Z",
7   "__v": 0
8 }
```

#### 4. Updating a Note using PUT /notes/ :noteId API

The screenshot shows a REST client interface with a browser tab at `http://localhost:3000/`. The request method is `PUT` and the URL is `http://localhost:3000/notes/5b408f91a0dcbd0a741e88fc`. The request body is set to `JSON (application/json)` and contains the following JSON:

```
1 {"title": "Edited title of second Note", "content": "Edited content of second Note."}
```

The screenshot shows the response of the PUT request. The status is `200 OK` and the time taken is `40 ms`. The response body is displayed in `JSON` format and contains the following JSON:

```
1 {
2   "_id": "5b408f91a0dcbd0a741e88fc",
3   "title": "Edited title of second Note",
4   "content": "Edited content of second Note.",
5   "createdAt": "2018-07-07T10:01:53.776Z",
6   "updatedAt": "2018-07-07T10:24:47.388Z",
7   "__v": 0
8 }
```

## 5. Deleting a Note using DELETE /notes/:noteId API

The screenshot shows a web browser with two tabs, both at `http://localhost:3000/`. The active tab displays a REST client interface. The method is set to **DELETE** and the URL is `http://localhost:3000/notes/5b408f91a0dcbd0a741e88fc`. The **Send** button is highlighted in blue. Below the URL bar, the **Body** tab is selected, showing a JSON response in **Pretty** format. The response is `{ "message": "Note deleted successfully!" }`. The status is **200 OK** and the time taken is **78 ms**. The **Headers** tab shows **6** headers. The **Test Results** tab is also visible.

DELETE `http://localhost:3000/notes/5b408f91a0dcbd0a741e88fc` Params Send Save

Body Cookies Headers (6) Test Results Status: 200 OK Time: 78 ms

Pretty Raw Preview JSON

```
1 {  
2   "message": "Note deleted successfully!"  
3 }
```