

The screenshot displays the Postman REST client interface. At the top, the address bar shows 'localhost:3000/api/user/register'. The request method is set to 'POST'. The request body is raw JSON, containing a user registration payload. The response is a 200 OK status with a response time of 111 ms and a body size of 397 B. The response body is displayed in JSON format, showing the user's details and a token.

Request:

```
POST localhost:3000/api/user/register
{
  "username": "Molly",
  "email": "molly@cloud.com",
  "password": "123456"
}
```

Response:

```
200 OK • 111 ms • 397 B
{
  "username": "Molly",
  "email": "molly@cloud.com",
  "password": "$2b$05$pyW/JV1/i.P7IBR06BQRM.PgHkghb73D5o5znAE3ILjJYvgDj/wlu",
  "_id": "67dd84a340f95498e69f1845",
  "__v": 0
}
```

The screenshot displays the Postman application interface. At the top, the workspace is named 'API Network'. The main area shows a REST client request configuration for the endpoint `localhost:3000/api/user/login`. The request method is **POST**, and the body is set to **raw** with JSON content: `{ "email": "molly@cloud.com", "password": "123456" }`. The request is successfully executed, resulting in a **200 OK** status with a response time of 75 ms and a body size of 565 B. The response body is displayed in JSON format: `{ "auth-token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjaWQiOiI1N2RkODRhMzQwZjkiNDk4ZTY5ZjE4NDUuIiwiaWF0IjE5NDI1NzA2Znd9.BXSkNDsZDDwvXU6a_g8_v0zi0bn_KYWIqF95icYw1M" }`. The interface includes various tabs for request configuration (Params, Authorization, Headers, Body, Scripts, Tests, Settings) and response viewing (Body, Cookies, Headers, Test Results). The bottom status bar shows the Postman logo and several utility icons.

3. Creating a 'first' post

The screenshot shows the Postman interface with a POST request to `localhost:3000/api/posts`. The request body is a JSON object:

```
1 {
2   "title": "First post",
3   "description": "This is my first post"
4 }
```

The response is a 200 OK status with a 73 ms response time and 386 B of data. The response body is a JSON object:

```
1 {
2   "title": "First post",
3   "description": "This is my first post",
4   "likes": 0,
5   "createdBy": "67dd84a340f95498e69f1845",
6   "_id": "67dd84d040f95498e69f1848",
7   "__v": 0
8 }
```

4. Fetching all posts (first and second)

The screenshot shows the Postman interface with a GET request to `localhost:3000/api/posts/`. The response is a 200 OK status with a 45 ms response time and 542 B of data. The response body is a JSON array of two objects:

```
1 [
2   {
3     "_id": "67dd84d040f95498e69f1848",
4     "title": "First post",
5     "description": "This is my first post",
6     "likes": 0,
7     "createdBy": "67dd84a340f95498e69f1845",
8     "__v": 0
9   },
10  {
11    "_id": "67dd84e440f95498e69f184a",
12    "title": "Second post",
13    "description": "This is my second post",
14    "likes": 0,
15    "createdBy": "67dd84a340f95498e69f1845",
16    "__v": 0
17  }
18 ]
```

5. Updating 'second' post to 'third' post

The screenshot shows the Postman interface with a PUT request to `localhost:3000/api/posts/67dd84e440f95498e69f184a`. The request body is a JSON object:

```
1 {
2   "title": "Third post",
3   "description": "This is my third post"
4 }
```

The response is a 200 OK status with a response time of 101 ms and a body size of 327 B. The response body is a JSON object:

```
1 {
2   "acknowledged": true,
3   "modifiedCount": 1,
4   "upsertedId": null,
5   "upsertedCount": 0,
6   "matchedCount": 1
7 }
```

The screenshot shows the Postman interface with a GET request to `localhost:3000/api/posts/`. The response is a 200 OK status with a response time of 47 ms and a body size of 540 B. The response body is a JSON array of two post objects:

```
1 [
2   {
3     "_id": "67dd84d040f95498e69f1848",
4     "title": "First post",
5     "description": "This is my first post",
6     "likes": 0,
7     "createdBy": "67dd84a340f95498e69f1845",
8     "__v": 0
9   },
10  {
11    "_id": "67dd84e440f95498e69f184a",
12    "title": "Third post",
13    "description": "This is my third post",
14    "likes": 0,
15    "createdBy": "67dd84a340f95498e69f1845",
16    "__v": 0
17  }
18 ]
```

6. Deleting 'third' post

POST localhost POST localhost GET localhost POST localhost GET localhost PUT localhost DEL localhost + No environment

localhost:3000/api/posts/67dd84e440f95498e69f184a

DELETE localhost:3000/api/posts/67dd84e440f95498e69f184a Send

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Headers 6 hidden

Key	Value	Description	Bulk Edit	Presets
auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...			
Key	Value	Description		

Body Cookies Headers (7) Test Results 200 OK 80 ms 273 B

JSON Preview Visualize

```
1 {
2   "acknowledged": true,
3   "deletedCount": 1
4 }
```

POST localhost POST localhost GET localhost POST localhost GET localhost PUT localhost DEL localhost + No environment

localhost:3000/api/posts/

GET localhost:3000/api/posts/ Send

Params Authorization Headers (6) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (7) Test Results 200 OK 45 ms 388 B

JSON Preview Visualize

```
1 [
2   {
3     "_id": "67dd84d040f95498e69f1848",
4     "title": "First post",
5     "description": "This is my first post",
6     "likes": 0,
7     "createdBy": "67dd84a340f95498e69f1845",
8     "__v": 0
9   }
10 ]
```

