

# Blog Website Backend

*By- Kaifur Rahman*

**Github Repository Link:** <https://github.com/kaifur-rahman/blog-website-REST-API-MVC-Based.git>

**Note:** forking might not work due to environment variables security setup.

**Live Link:** <https://blog-website-rest-api.onrender.com/>

## Documentation

**Tech Used:** NodeJs, ExpressJs, MongoDB, Mongoose

### **Features:**

- All the passwords are hashed as per industry standard.
- REST API developed on MVC pattern

### **Users Features:**

- 1) Users can sign up
- 2) Users can login, backend will identify user exists or not or password is incorrect
- 3) Users can post/delete a comment on any post
- 4) Users can like /unlike the post. It will identify whether user is liking the post for first time or already liked it.

### **Admin Features:**

- 1) Admin cannot sign up directly from frontend his/her credentials is provided from backend
- 2) Admin can log in backend identifies correctness of email and password

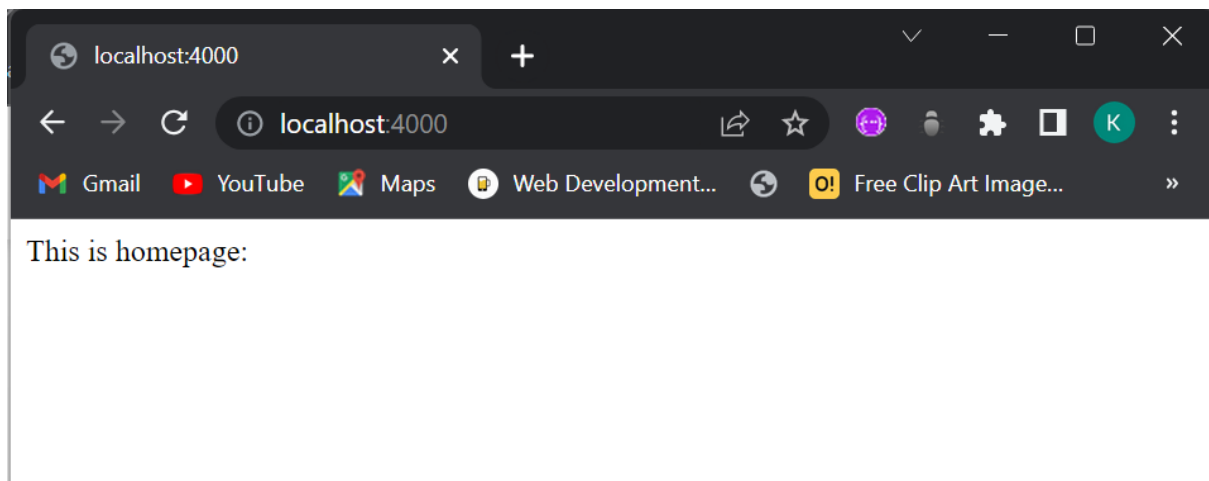
- 3) Admin can get all the blog posts posted by him/her
- 4) Admin can create new blogpost
- 5) Admin can update/modify existing blogpost's title/body
- 6) Admin can delete any existing blogpost

## **Https Requests:**

### **HomePage:**

**URL:** <http://localhost:4000/>

### **Output:**



## Users:

**Common Endpoint:** <http://localhost:4000/users/>

### 1) To sign up user:

**POST Request:** <http://localhost:4000/users/signup>

#### Body Parameters:

name:

email:

pass:

#### Output:

http://localhost:4000/users/signup

POST <http://localhost:4000/users/signup> [Send](#)

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings [Cookies](#)

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	name	Rahul Kumar			
<input checked="" type="checkbox"/>	email	rahul@gmail.com			
<input checked="" type="checkbox"/>	pass	abcd			

Body [Cookies](#) [Headers \(7\)](#) [Test Results](#) [200 OK](#) [89 ms](#) [257 B](#) [Save Response](#)

Pretty Raw Preview Visualize [HTML](#) [↻](#)

1 New user created successfully

*What if another user tries to create account with same email id*

http://localhost:4000/users/signup

POST http://localhost:4000/users/signup

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	name	Rahul Kumar			
<input checked="" type="checkbox"/>	email	rahul@gmail.com			
<input checked="" type="checkbox"/>	pass	abcd			
	Key	Value	Description		

Body Cookies Headers (7) Test Results 200 OK 67 ms 266 B Save Response

Pretty Raw Preview Visualize HTML

```
1 User already exist with this email id
```

**1) To log in user:**

**POST Request:** <http://localhost:4000/users/login>

**Body Parameters:**

email:

pass:

**Output:**

*Example of wrong password entered*

http://localhost:4000/users/login

Save

POST http://localhost:4000/users/login Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

<input type="checkbox"/>	name	Rahul Kumar	
<input checked="" type="checkbox"/>	email	rahul@gmail.com	
<input checked="" type="checkbox"/>	pass	abcdd	
	Key	Value	Description

Body Cookies Headers (7) Test Results 200 OK 55 ms 246 B Save Response

Pretty Raw Preview Visualize HTML

```
1 incorrect password
```

*Example of wrong email entered:*

http://localhost:4000/users/login

Save

POST http://localhost:4000/users/login Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

<input type="checkbox"/>	name	Rahul Kumar	
<input checked="" type="checkbox"/>	email	rahul123@gmail.com	
<input checked="" type="checkbox"/>	pass	abcdd	
	Key	Value	Description

Body Cookies Headers (7) Test Results 200 OK 51 ms 246 B Save Response

Pretty Raw Preview Visualize HTML

```
1 No such user exist
```

*Example of correct credentials:*

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:4000/users/login`
- Method:** `POST`
- Body Type:** `x-www-form-urlencoded`
- Body Data:**

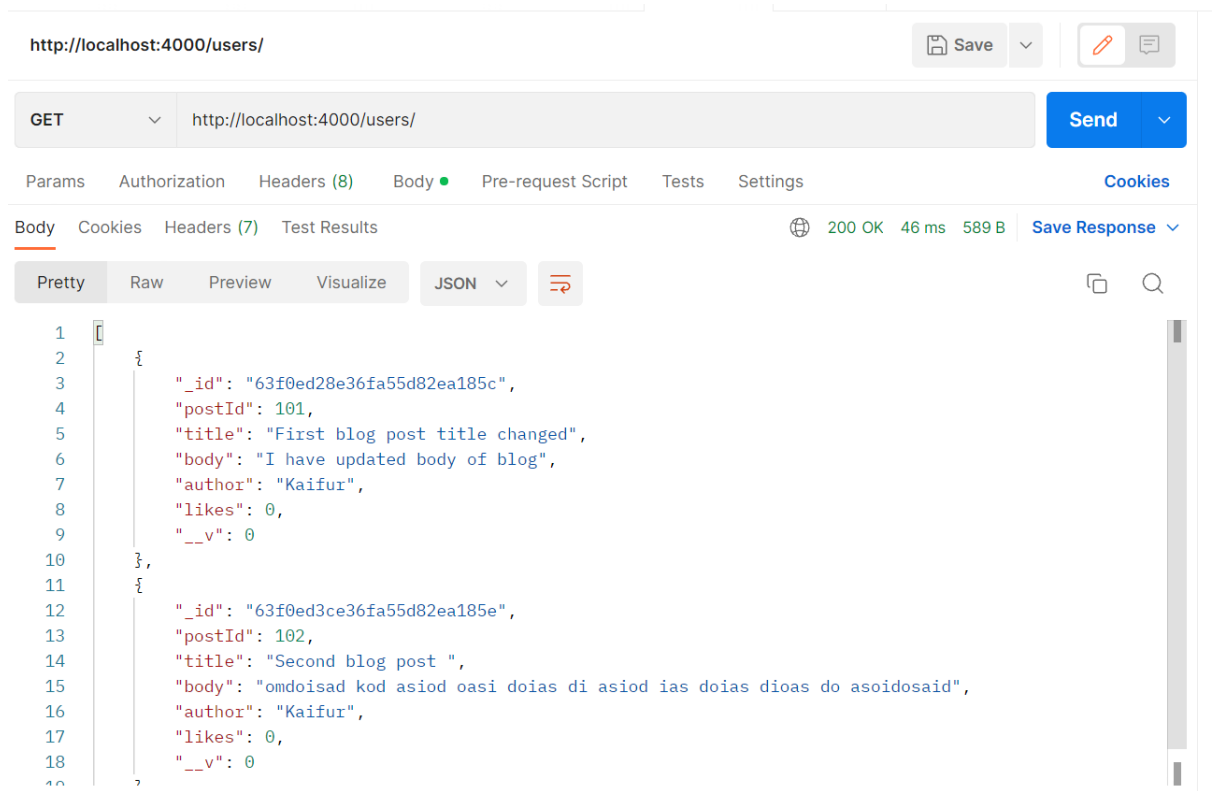
Key	Value	Description
<input type="checkbox"/> name	Rahul Kumar	
<input checked="" type="checkbox"/> email	rahul@gmail.com	
<input checked="" type="checkbox"/> pass	abcd	
- Status:** `200 OK`, `92 ms`, `240 B`
- Response Body:**

```
1 login success
```

**2) To display all the blog posts after success full log in :**

**GET Request:** <http://localhost:4000/users/>

**Output:**



**Note:** I have separated comment and like/unlike features as comment

**Common end point:** <http://localhost:4000/comments/>

**2) To make user post a comment :**

**POST Request:**

<http://localhost:4000/comments/UserName/postComment>

### Body Parameters:

userEmail:

comment:

commentId: datatype numeric, (Later can be generated from backend)

postId (Existing post): datatype numeric, (Later can be generated from backend)

### Output:

http://localhost:4000/comments/Rahul/postComment

POST http://localhost:4000/comments/Rahul/postComment

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data **x-www-form-urlencoded** raw binary GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	userEmail	rahul@gmail.com			
<input checked="" type="checkbox"/>	comment	Nice post			
<input checked="" type="checkbox"/>	commentId	91			
<input checked="" type="checkbox"/>	postId	101			

Body Cookies Headers (7) Test Results 200 OK 70 ms 265 B Save Response

Pretty Raw Preview Visualize HTML

```
1 Successfully posted a comment on blog
```

### 3) To make user delete a comment :

#### DELETE Request:

<http://localhost:4000/users/username/deleteComment>



**Body Parameters:**  
commentId:

**Output:**

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:4000/comments/Rahul/deleteComment`
- Method:** DELETE
- Body Parameters Table:**

	KEY	VALUE	DESCRIPTION
<input type="checkbox"/>	userEmail	rahul@gmail.com	
<input type="checkbox"/>	comment	Nice post	
<input checked="" type="checkbox"/>	commentId	91	
- Response Status:** 200 OK, 84 ms, 430 B
- Response Body (JSON):**

```
1 Successfully deleted a comment{
2   _id: new ObjectId("63f0ee35e36fa55d82ea1862"),
3   userName: 'Rahul',
4   userEmail: 'rahul@gmail.com',
5   comment: 'Nice post ',
6   commentId: 91,
7   postId: 101,
8   __v: 0
9 }
```

**4) To make user like a post :**

**POST Request:** `http://localhost:4000/users/:userName/likePost`

## Body Parameters:

email:

postId: datatype numeric

## Output:

*Suppose user is liking a post with id 101 for first time*

http://localhost:4000/comments/Priya/likePost

Save

POST

http://localhost:4000/comments/Priya/likePost

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	email	priya@hotmail.com			
<input checked="" type="checkbox"/>	postId	101			
	Key	Value	Description		

Body

Cookies

Headers (7)

Test Results

200 OK

855 ms

274 B

Save Response

Pretty

Raw

Preview

Visualize

HTML

1

Successfully pushed liked post in liked array

*Now what if user again tries to like the post which he/she already liked*

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:4000/comments/Priya/likePost`
- Method:** POST
- Body:** `http://localhost:4000/comments/Priya/likePost`
- Form Data:**
  - ☒ email: priya@hotmail.com
  - ☒ postId: 101
- Response:** 200 OK, 94 ms, 260 B. The response body is: `1 You have already liked this post`

**5) To make user unlike a comment:**

**PUT Request:** <http://localhost:4000/users/username/unlikePost>

**Body Parameters:**

email:

postId: datatype numeric

## Output:

http://localhost:4000/comments/Priya/unlikePost

PUT http://localhost:4000/comments/Priya/unlikePost

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data **x-www-form-urlencoded** raw binary GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	email	priya@hotmail.com			
<input checked="" type="checkbox"/>	postId	101			
	Key	Value	Description		

Body Cookies Headers (7) Test Results 200 OK 85 ms 253 B Save Response

Pretty Raw Preview Visualize HTML

```
1 successfully unliked post
```

## Admin:

### Common endpoint:

1) To make admin log in :

POST Request: <http://localhost:4000/admin/login>

#### Body Parameters:

email:

pass:

## Output:

### Example of wrong email

http://localhost:4000/admin/login

Save

POST http://localhost:4000/admin/login Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

<input type="checkbox"/>	name	Rahul Kumar	
<input checked="" type="checkbox"/>	email	rahul@gmail.com	
<input checked="" type="checkbox"/>	pass	abcd	
	Key	Value	Description

Body Cookies Headers (7) Test Results 200 OK 118 ms 247 B Save Response

Pretty Raw Preview Visualize HTML

```
1 No such admin exist
```

### Example of wrong password:

http://localhost:4000/admin/login

Save

POST http://localhost:4000/admin/login Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

<input type="checkbox"/>	name	Rahul Kumar	
<input checked="" type="checkbox"/>	email	kaifur@yahoo.com	
<input checked="" type="checkbox"/>	pass	1222	
	Key	Value	Description

Body Cookies Headers (7) Test Results 200 OK 64 ms 246 B Save Response

Pretty Raw Preview Visualize HTML

```
1 incorrect password
```

## Example of correct credentials

http://localhost:4000/admin/login

POST http://localhost:4000/admin/login

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

<input type="checkbox"/>	name	Rahul Kumar	
<input checked="" type="checkbox"/>	email	kaifur@yahoo.com	
<input checked="" type="checkbox"/>	pass	1234	
	Key	Value	Description

Body Cookies Headers (7) Test Results 200 OK 51 ms 240 B Save Response

Pretty Raw Preview Visualize HTML

```
1 login success
```

## 2) To display all the blogposts posted by admin :

GET Request: <http://localhost:4000/admin/AdminName>

## Output:

http://localhost:4000/admin/Kaifur/

GET http://localhost:4000/admin/Kaifur/

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

<input checked="" type="checkbox"/>	postId	101	
-------------------------------------	--------	-----	--

Body Cookies Headers (7) Test Results 200 OK 67 ms 445 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "_id": "63f0e968e36fa55d82ea1853",
4     "postId": 101,
5     "title": "First blog post",
6     "body": "jfdskf sjkf dsjf kdsjf jds fkjds fkjds jf dsjfk dskjf jdsf fjkds fkjds fkds jfdskf
7       jdsfk",
8     "author": "Kaifur",
9     "likes": 0,
10    "__v": 0
11  }
12 }
```

### 3) To post a new blog :

**POST Request:** <http://localhost:4000/admin/:AdminName/createBlog>

#### Body Parameters:

postId: datatype: numeric (later this can be generated from backend only)

title:

body:

#### Output:

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:4000/admin/Kaifur/createBlog`
- Method:** POST
- Body Parameters:**

Key	Value	Description
<input checked="" type="checkbox"/> postId	101	
<input checked="" type="checkbox"/> title	First blog post	
<input checked="" type="checkbox"/> body	jfdskf sjkf dsjf kdsjf jds fkjds fkjds jf dsjkd ...	
- Response:** 200 OK, 110 ms, 267 B. The response body is: `1 New blog posted successfully by+ Kaifur`

### 4) To update existing blog :

**PUT Request:** <http://localhost:4000/admin/:AdminName/updateBlog>

#### Body Parameters:

postId:

title:

body:

## Output:

*Before Update (Fetched using get request ref #2 for details)*

http://localhost:4000/admin/Kaifur/

GET http://localhost:4000/admin/Kaifur/ Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

☒ postId 101

Body Cookies Headers (7) Test Results 200 OK 67 ms 445 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "63f0e968e36fa55d82ea1853",
3   "postId": 101,
4   "title": "First blog post",
5   "body": "jfdskf sjkf dsjf kdsjf jds fkjds fkjds jf dsjfk dskjf jdsf fjkds fkjds fkdskf
6     jdsfk",
7   "author": "Kaifur",
8   "likes": 0,
9   "__v": 0
10 }
11
```

## Update request:

http://localhost:4000/admin/Kaifur/updateBlog

PUT http://localhost:4000/admin/Kaifur/updateBlog Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

<input type="checkbox"/>	name	Rahul Kumar	
<input checked="" type="checkbox"/>	postId	101	
<input checked="" type="checkbox"/>	title	First blog post title changed	
<input checked="" type="checkbox"/>	body	I have updated body of blog	
	Key	Value	Description

Body Cookies Headers (7) Test Results 200 OK 146 ms 253 B Save Response

Pretty Raw Preview Visualize HTML

```
1 successfully updated blog
```



After update:

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:4000/admin/Kaifur/`
- Method:** `GET`
- Body:** `name` (input field) and `Rahul Kumar` (text input)
- Response:** `200 OK`, `59 ms`, `398 B`
- Response Body (JSON):**

```
{  "_id": "63f0e968e36fa55d82ea1853",  "postId": 101,  "title": "First blog post title changed",  "body": "I have updated body of blog",  "author": "Kaifur",  "likes": 0,  "__v": 0}
```

## 5) To delete existing blog:

**DELETE Request:** <http://localhost:4000/admin/:AdminName/deleteBlog>

### Body Parameters:

postId:

### Output:

The screenshot displays a REST client interface with the following details:

- URL:** `http://localhost:4000/admin/Kaifur/deleteBlog`
- Method:** `DELETE`
- Body Parameters:**

Key	Value	Description
<input type="checkbox"/> name	Rahul Kumar	
<input checked="" type="checkbox"/> postId	101	
<input type="checkbox"/> title	First blog post title changed	
<input type="checkbox"/> body	I have updated body of blog	
- Response:**
  - Status:** 200 OK
  - Time:** 74 ms
  - Size:** 258 B
  - Message:** Successfully deleted blog post