

# Postman

## 1. Add New User

The screenshot displays the Postman application interface for a POST request. The top bar shows the URL `https://thinking-tester-contact-list.herokuapp.com/users` and a 'Send' button. Below the URL bar, tabs for 'Params', 'Authorization', 'Headers (11)', 'Body', 'Scripts', and 'Settings' are visible. The 'Scripts' tab is active, showing a JavaScript script for testing the response. The 'Post-response' tab is also visible. On the right, there are sections for 'Packages' and 'Snippets'. The bottom section shows the response body in 'Pretty' format, displaying a JSON object with user information and a token. The status bar at the bottom indicates a '201 Created' status, a response time of 1.09 s, and a size of 1.24 KB.

HTTP Telecom Domain / AddNewUser

POST `https://thinking-tester-contact-list.herokuapp.com/users` Send

Params Authorization Headers (11) Body Scripts Settings Cookies

Pre-request

Post-response

```
1 // Fixed reference error
2 pm.test("Test for Status code 201", ()=>{
3     pm.expect(pm.response.code).to.eql(201);
4 }
5
6
7 pm.test("Test for Status message", ()=>{
8     pm.expect(pm.response.status).to.eql("Created");
9 }
10
```

Packages + Add to package

Open package library →

Snippets

Get an environment variable

Get a global variable

Get a variable

Body Cookies (1) Headers (12) Test Results (2/2) 201 Created • 1.09 s • 1.24 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2     "user": {
3         "_id": "6763f56551604a001306a955",
4         "firstName": "Test",
5         "lastName": "User",
6         "email": "lekha1734604131631@gmail.com",
7         "__v": 1
8     },
9     "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfcm9udCI6I2NzYzZjU2NTUxNjA0YTAwMTMwNmE5NTUuIiwiaWF0IjE5MzQ2MDQxMzN9.6n3yXPNB235qnCWgoA40LqHubu7IeXYzLdvzvJZ1C-Q"
10 }
```

## 2. Get User Profile

HTTP Telecom Domain / GetUser1Profile Save Share

**GET** https://thinking-tester-contact-list.herokuapp.com/users/me Send

Params Authorization Headers (9) Body **Scripts** Settings Cookies

Pre-request

Post-response

```
1 pm.test("Test for Status code 200", ()=>{
2   pm.expect(pm.response.code).to.eql(200);
3   console.log(pm.response.code);
4 })
5
6 pm.test("Test for status message, OK", ()=>{
7   pm.expect(pm.response.status).to.eql('OK');
8 })
9
10
```

Packages

[Open package library](#) →

Snippets

Get an environment variable

Get a global variable

Get a variable

Body Cookies (1) Headers (11) Test Results (2/2) 200 OK • 901 ms • 876 B • Save Response

Pretty Raw Preview Visualize JSON ↔

```
1 {
2   "_id": "6763f56551604a001306a955",
3   "firstName": "Test",
4   "lastName": "User",
5   "email": "lekha1734604131631@gmail.com",
6   "__v": 1
7 }
```

### 3. Update User Profile : Patch

The screenshot shows the Postman interface for a PATCH request. The URL is `https://thinking-tester-contact-list.herokuapp.com/users/me`. The request is configured with the following details:

- Method:** PATCH
- URL:** `https://thinking-tester-contact-list.herokuapp.com/users/me`
- Scripts:** The following JavaScript code is present in the Scripts tab:

```
1 pm.test("Status code is 200", function () {
2     pm.response.to.have.status(200);
3 });
4 pm.test("Status message is 'OK'", function () {
5     pm.response.to.have.status('OK');
6 });
```
- Response:** The response is a JSON object, displayed in the "Body" tab:

```
1 {
2     "_id": "6763f56551604a001306a955",
3     "firstName": "Test",
4     "lastName": "Aravinda",
5     "email": "lekha1734604131631@gmail.com",
6     "__v": 1
7 }
```
- Status:** 200 OK, 948 ms, 876 B
- Actions:** Save Response, Share, Copy, Search

## 4. Login User

The screenshot shows the Postman interface for a POST request to `https://thinking-tester-contact-list.herokuapp.com/users/login`. The request is successful with a `200 OK` status. The response body is a JSON object containing user details and a token.

**Scripts**

```
1 // Fixed the test for Status code 200 to use the correct variable
2 pm.test("Test for Status code 200", ()=>{
3   pm.expect(pm.response.code).to.eql(200);
4 }
5
6
7 pm.test("Test for Status message", ()=>{
8   nm.expect(nm.response.status).to.eql("OK");
```

**Body**

```
1 {
2   "user": {
3     "_id": "6763f56551604a001306a955",
4     "firstName": "Test",
5     "lastName": "Aravinda",
6     "email": "lekha1734604131631@gmail.com",
7     "_v": 2
8   },
9   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NzYzZjU2NTUxNjA0YTAwMTMwNmE5NTUuIiwiaWF0IjE5MzQ2MDQzNzN9._Br3IteYQWLBjv1VH8m630xiuplQR-EvymvqxpWZcTs"
10 }
```

## New token got saved as environment variable

The screenshot shows the 'Telecom' environment in Postman. The table lists several variables that have been saved from the previous request.

	Variable	Type	Initial value	Current value	
<input checked="" type="checkbox"/>	useremail	default		Lekha1734021026016@gmail.com	
<input checked="" type="checkbox"/>	token	default		eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...	
<input checked="" type="checkbox"/>	id	default		67569b2b4c4d240013b83c27	
<input checked="" type="checkbox"/>	email	default		lekha1734021560683@gmail.com	
<input checked="" type="checkbox"/>	UserToken	default		eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...	
	Add new variable				

## 5. Add Contact

The screenshot displays the Postman application interface. At the top, the URL bar shows 'https://thinking-tester-contact-list.herokuapp.com/contacts' with a 'POST' method selected. The 'Send' button is visible. Below the URL bar, the 'Scripts' tab is active, showing a pre-request script with the following code:

```
1 pm.test("Test for Status code 201", ()=>{
2   pm.expect(pm.response.code).to.eql(201);
3   console.log(pm.response.code); // Fixed reference
4   // error by adding pm. before response.code
5 })
6
```

The 'Test Results' tab shows a successful response with a status of '201 Created', a response time of '243 ms', and a size of '1.06 KB'. The response body is displayed in the 'Body' tab, showing a JSON object:

```
1 {
2   "_id": "6763f6762e7a950013df4167",
3   "firstName": "Lekha",
4   "lastName": "Aravinda",
5   "birthdate": "1993-07-07",
6   "email": "lekha1734604131631@gmail.com",
7   "phone": "8005555555",
8   "street1": "1 Main St.",
9   "street2": "Apartment A",
10  "city": "Anytown",
11  "stateProvince": "KS",
12  "postalCode": "12345",
13  "country": "India",
14  "owner": "6763f56551604a001306a955",
15  "__v": 0
16 }
```

The interface also includes tabs for 'Params', 'Authorization', 'Headers (11)', 'Body', 'Scripts', 'Settings', and 'Cookies'. The 'Send' button is located to the right of the URL bar. The 'Test Results' tab shows a green status bar indicating '201 Created'.

## 6. Get Contact List

HTTP Telecom Domain / GetContactList

GET https://thinking-tester-contact-list.herokuapp.com/contacts

Send

Params Authorization Headers (9) Body Scripts Settings Cookies

Pre-request

Post-response

```
1 pm.test("Test for Status code 200", ()=>{
2   pm.expect(pm.response.code).to.eql(200);
3   console.log(pm.response.code);
4 })
5
6 pm.test("Test for status message, 0k", ()=>{
7   nm.expect(nm.response.status).to.eql('OK');
```

Packages

Open package library →

Snippets

Get an environment variable

Body Cookies (1) Headers (11) Test Results (2/2) 200 OK • 238 ms • 1.06 KB Save Response

Pretty Raw Preview Visualize JSON

```
3   "_id": "6763f6762e7a950013df4167",
4   "firstName": "Lekha",
5   "lastName": "Aravinda",
6   "birthdate": "1993-07-07",
7   "email": "lekha1734604131631@gmail.com",
8   "phone": "8005555555",
9   "street1": "1 Main St.",
10  "street2": "Apartment A",
11  "city": "Anytown",
12  "stateProvince": "KS",
13  "postalCode": "12345",
14  "country": "India",
15  "owner": "6763f56551604a001306a955",
16  "__v": 0
17 }
```

## 7. Get Contact

[HTTP](#) Telecom Domain / **Get Contact**

Save

Share

GET

https://thinking-tester-contact-list.herokuapp.com/contacts/

Send

Params

Authorization

Headers (9)

Body

Scripts

Settings

Cookies

Pre-request

Post-response

```
1 pm.test("Test for Status code 200", () => {
2   pm.response.to.have.status(200);
3   console.log(pm.response.code);
4 });
5
6 var res = pm.response.json();
7 console.log(pm.response.headers);
```

Packages

[Open package library](#)

Snippets

Get an environment variable

Body

Cookies (1)

Headers (11)

Test Results (2/2)

200 OK

237 ms

1.05 KB

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
2 {
3   "_id": "6763f6762e7a950013df4167",
4   "firstName": "Lekha",
5   "lastName": "Aravinda",
6   "birthdate": "1993-07-07",
7   "email": "lekha1734604131631@gmail.com",
8   "phone": "8005555555",
9   "street1": "1 Main St.",
10  "street2": "Apartment A",
11  "city": "Anytown",
12  "stateProvince": "KS",
13  "postalCode": "12345",
14  "country": "India",
15  "owner": "6763f56551604a001306a955",
16  "__v": 0
```

## 8. Update Contact(PUT)

The screenshot displays the Postman interface for a PUT request to update a contact. The sidebar on the left shows the API collection structure, with 'Telecom Domain' expanded and 'PUT Update Contact' selected. The main workspace shows the request details for the endpoint `https://thinking-tester-contact-list.herokuapp.com/contacts/{{contactID}}`. The 'Scripts' tab is active, showing a series of 12 test scripts. The 'Test Results' section at the bottom shows three passed tests: 'Test for Status code ,200', 'Test for status message: OK', and 'Test for email id matching'. The response status is '200 OK'.

**Request Details:**

- Method: PUT
- URL: `https://thinking-tester-contact-list.herokuapp.com/contacts/{{contactID}}`
- Params: Authorization, Headers (13), Body, Scripts, Settings

**Scripts:**

```
1 pm.test("Test for Status code ,200", () => {
2   pm.expect(pm.response.code).to.eql(200);
3 })
4 pm.test("Test for status message: OK", () => {
5   pm.expect(pm.response.status).to.eql("OK");
6 })
7 var emailid = pm.response.json().email;
8 console.log(emailid);
9 pm.test("Test for email id matching", () => {
10  pm.expect(emailid).to.eql("lekha@gmail.com");
11 })
12
```

**Test Results:**

- PASSED Test for Status code ,200
- PASSED Test for status message: OK
- PASSED Test for email id matching

**Response:** 200 OK



## 9. Update contact(Patch)

The screenshot displays the Postman application interface for testing an API endpoint. On the left, a sidebar lists various API collections, with 'Telecom Domain' expanded to show endpoints like 'AddNewUser', 'UpdateUser', and 'PartialUpdateContact'. The main workspace is configured for a PATCH request to the URL 'https://thinking-tester-contact-list.herokuapp.com/contacts/{{contactID}}'. The 'Scripts' tab is active, showing a series of PM (Postman) test scripts designed to verify the response status, message, and a specific attribute (firstName). The 'Test Results' section at the bottom indicates that all three tests passed successfully. The right-hand sidebar provides additional context with 'Packages' and 'Snippets'.

HTTP Telecom Domain / PartialUpdateContact

PATCH https://thinking-tester-contact-list.herokuapp.com/contacts/{{contactID}}

Send

Params Authorization Headers (11) Body Scripts Settings Cookies

Pre-request

Post-response

```
1 pm.test("Test for status code,200",()=>{
2   pm.expect(pm.response.code).to.eql(200);
3 }
4
5 pm.test("Test for status message,OK",()=>{
6   pm.expect(pm.response.status).to.eql("OK");
7 }
8 }
9
10 var fName=pm.response.json().firstName;
11 console.log(fName);
12
13 pm.test("Test for firstname validation ",()=>{
14   pm.expect(fName).to.eql("Anna123");
15 }
16 }
```

Packages

Open package library

Snippets

Get an environment variable

Get a global variable

Get a variable

Get a collection variable

Set an environment variable

Body Cookies (1) Headers (11) Test Results (3/3) 200 OK 902 ms 1.04 KB

Filter Results

PASSED Test for status code,200

PASSED Test for status message,OK

PASSED Test for firstname validation

Postbot

## 10. Logout User

The screenshot displays the Postman application interface for a REST client. At the top, the breadcrumb navigation shows 'Telecom Domain / Log Out User'. The main header area includes a 'POST' method dropdown, the URL 'https://thinking-tester-contact-list.herokuapp.com/users/logout', and buttons for 'Save' and 'Share'. Below this, a tabbed interface shows 'Params', 'Authorization', 'Headers (10)', 'Body', 'Scripts', and 'Settings'. The 'Scripts' tab is active, showing a pre-request script and a post-response script. The post-response script contains two PM tests: 'Test for Status code 200' and 'Test for Status message'. To the right of the scripts, there are sections for 'Packages' and 'Snippets'. The bottom status bar indicates the request was successful with a '200 OK' status, a response time of 994 ms, and a body size of 667 B. Below the status bar, the 'Test Results' section shows two passed tests: 'Test for Status code 200' and 'Test for Status message'.

Telecom Domain / Log Out User

POST [https://thinking-tester-contact-label.herokuapp.com/users/logout](https://thinking-tester-contact-list.herokuapp.com/users/logout) [Save](#) [Share](#)

Params Authorization Headers (10) Body **Scripts** Settings Cookies

Pre-request

Post-response

```
1 pm.test("Test for Status code 200", ()=>{
2   pm.expect(pm.response.code).to.eql(200);
3
4 })
5
6 pm.test("Test for Status message", ()=>{
7   pm.expect(pm.response.status).to.eql("OK");
8 })
9
```

Packages [+ Add to package](#) [Open package library](#) →

Snippets

- Get an environment variable
- Get a global variable
- Get a variable
- Get a collection variable
- Set an environment variable

Body Cookies (1) Headers (9) **Test Results (2/2)** [Refresh](#) **200 OK** • 994 ms • 667 B • [Download](#) [Save Response](#) [More](#)

Filter Results [Refresh](#)

**PASSED** Test for Status code 200

**PASSED** Test for Status message