

1. Add a user

URL: http://127.0.0.1:5000/add_user

Request Body:

```
{  
  "name": "JRD",  
  "email": "jrd.rt@example.com",  
  "age": 83,  
  "password": "Abcd@1234"  
}
```

Response Body:

```
{  
  "message": "User successfully added",  
  "user_id": "6713350d1b6a58882d43c195"  
}
```

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

- URL:** `http://127.0.0.1:5000/add_user`
- Method:** `POST`
- Request Body (JSON):**

```
{  
  "name": "JRD",  
  "email": "jrd.rt@example.com",  
  "age": 83,  
  "password": "Abcd@1234"  
}
```
- Response Body (JSON):**

```
{  
  "message": "User successfully added",  
  "user_id": "6713350d1b6a58882d43c195"  
}
```
- Status:** 201 CREATED
- Time:** 214 ms
- Size:** 254 B

2. Get a single user

URL: http://127.0.0.1:5000/get_user/6713350d1b6a58882d43c195

Response Body:

```
{
  "message": {
    "_id": "6713350d1b6a58882d43c195",
    "age": 83,
    "email": "jrd.rt@example.com",
    "name": "JRD",
    "password":
"script:32768:8:1$ZXsySdBPHrdknN7Y$853aa76fdd0541dc1018619c5c031e
7e5576a45518c3a29d617ae9efec79cc8e2a526ad77c7b347141e8609dab44
abc37a5703a34511bc1980b7a258f5897967"
  }
}
```

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

- URL:** `http://127.0.0.1:5000/get_user/6713350d1b6a58882d43c195`
- Method:** GET
- Body:** (Empty)
- Response:** 200 OK, 40 ms, 478 B
- Response Body (JSON):**

```
{  "message": {    "_id": "6713350d1b6a58882d43c195",    "age": 83,    "email": "jrd.rt@example.com",    "name": "JRD",    "password": "scrypt:32768:8:1$ZXsySdBPHrdknN7Y$853aa76fdd0541dc1018619c5c031e7e5576a45518c3a29d617ae9efec79cc8e2a526ad77c7b347141e8609dab44abc37a5703a34511bc1980b7a258f5897967"  }}
```

3. Get all users

URL: http://127.0.0.1:5000/get_users

Response Body:

```
{  "message": [    {      "_id": "6713350d1b6a58882d43c195",      "age": 83,      "email": "jrd.rt@example.com",      "name": "JRD",      "password": "scrypt:32768:8:1$ZXsySdBPHrdknN7Y$853aa76fdd0541dc1018619c5c031e7e5576a45518c3a29d617ae9efec79cc8e2a526ad77c7b347141e8609dab44abc37a5703a34511bc1980b7a258f5897967"
```

```
},  
  
{  
  "_id": "671336d81b6a58882d43c198",  
  "age": 51,  
  "email": "sach@example.com",  
  "name": "Sachin",  
  "password":  
"scrypt:32768:8:1$JcG0lc6qV3bP3DkQ$1dd1b5d8ab503e9c4d96740870b42  
94b142c9020f8cc4a5ceb1fc5886cc0dc891a26be684ed5a6e2d265050b8911  
a00099af8ad1cf23ee4015fb029c12182bd1"  
}  
]  
}
```

The screenshot displays a web browser's developer tools interface. At the top, the address bar shows the URL `http://127.0.0.1:5000/get_users`. Below the address bar, the 'Send' button is visible. The 'Body' tab is selected, showing the request body as an empty string. The 'Headers' tab is also visible, showing the request headers. The 'Test Results' tab is selected, showing the response status as '200 OK' with a response time of '51 ms' and a response size of '660 B'. The response body is displayed in the 'Pretty' view, showing a JSON object with a 'message' array containing one user object. The user object has the following properties: '_id': '6713350d1b6a58882d43c195', 'age': 83, 'email': 'jrd.rt@example.com', 'name': 'JRD', and 'password': 'scrypt:32768:8:1\$ZXsySdBPHrdknN7Y\$853aa76fdd0541dc1018619c5c031e7e5576a45518c3a29d617ae9efe c79cc8e2a526ad77c7b347141e8609dab44abc37a5703a34511bc1980b7a258f5897967'.

```
1  
2  "message": [  
3    {  
4      "_id": "6713350d1b6a58882d43c195",  
5      "age": 83,  
6      "email": "jrd.rt@example.com",  
7      "name": "JRD",  
8      "password":  
        "scrypt:32768:8:1$ZXsySdBPHrdknN7Y$853aa76fdd0541dc1018619c5c031e7e5576a45518c3a29d617ae9efe  
        c79cc8e2a526ad77c7b347141e8609dab44abc37a5703a34511bc1980b7a258f5897967"
```

4. Update a user

URL: http://127.0.0.1:5000/get_users

Request Body:

```
{  
  "name": "Dravid",  
  "email": "sach@example.com",  
  "age": 52,  
  "password": "Efgh@5678"  
}
```

Response Body:

```
{  
  "message": "User updated successfully"  
}
```

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

- URL Bar:** Shows the URL `http://127.0.0.1:5000/update_user/671336d81b6a58882d43c198` and a `PUT` method dropdown.
- Request Body:** A JSON object is entered in the body field:

```
{  
  "name": "Dravid",  
  "email": "sach@example.com",  
  "age": 52,  
  "password": "Efgh@5678"  
}
```
- Response:** The response status is `200 OK` with a response time of `27 ms` and a size of `210 B`. The response body is a JSON object:

```
{  
  "message": "User updated successfully"  
}
```
- Interface Elements:** Includes tabs for Params, Authorization, Headers (8), Body (selected), Pre-request Script, Tests, and Settings. There are also buttons for Save, Send, Beautify, and Save Response.

5. Delete a user

URL: http://127.0.0.1:5000/delete_user/6713350d1b6a58882d43c195

Response Body:

```
{  
  "message": "User deleted successfully"  
}
```

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

- URL Bar:** Shows the URL `http://127.0.0.1:5000/delete_user/6713350d1b6a58882d43c195` and a `DELETE` method dropdown.
- Send Button:** A blue button labeled `Send`.
- Request Body:** A large empty text area for the request body.
- Response Section:** Includes tabs for `Body`, `Cookies`, `Headers (5)`, and `Test Results`. The `Body` tab is active, showing a `JSON` response in `Pretty` format:

```
{  
  "message": "User deleted successfully"  
}
```
- Status Bar:** Displays the status `200 OK`, response time `19 ms`, and size `210 B`, along with a `Save Response` button.