

Post Man:

Creating User Account:

The screenshot displays the Postman interface for a POST request to `http://localhost:3001/api/v1/user/signup`. The request body is a JSON object with the following fields:

```
1 {
2   "username": "Mobina.zargary",
3   "email": "Mobina.zargary@gmail.com",
4   "password": "13812002"
5 }
6
```

The response is a 201 Created status, indicating successful user creation. The response body is a JSON object:

```
1 {
2   "message": "User created successfully.",
3   "user_id": "670a95c12d42a4524dbdc453"
4 }
```

The interface also shows tabs for Params, Authorization, Headers (9), Body, Scripts, and Settings. The Body tab is active, and the response is displayed in the bottom section with tabs for Body, Cookies, Headers (7), and Test Results. The response is formatted as JSON and includes a 'Save Response' button.

## Login User:

The screenshot displays a REST client interface with a tab for a POST request to `http://localhost:3001/api/v1/user/login`. The request body is a JSON object containing email and password. The response is a 200 OK status with a JSON body indicating a successful login and a user ID.

**Request:**

```
POST http://localhost:3001/api/v1/user/login
```

Body (raw):

```
1 {
2   "email": "Mobina.zargary@gmail.com",
3   "password": "13812002"
4 }
```

**Response:**

```
200 OK - 203 ms - 303 B
```

Body (pretty):

```
1 {
2   "message": "Login successful.",
3   "user_id": "670a95c12d42a4524dbdc453"
4 }
```

## Create Employees:

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

- URL:** `http://localhost:3001/api/v1/emp/employees`
- Method:** `POST`
- Body (Request):**

```
1 {  
2   "first_name": "Mobina",  
3   "last_name": "Zargary",  
4   "email": "Mobina.zargary@gmail.com",  
5   "position": "Designer",  
6   "salary": 85000,  
7   "date_of_joining": "2023-08-10",  
8   "department": "Design"  
9 }  
10
```
- Response:**
  - Status:** `201 Created`
  - Time:** `89 ms`
  - Size:** `325 B`
  - Body (Response):**

```
1 {  
2   "message": "Employee created successfully.",  
3   "employee_id": "670a9b3de3f97e42b034a30e"  
4 }
```

rt

OverviewGET http://lo. •GET Untitled Iproduct-:POST Create •GET Get Proc •PUT Update I •DEL Delete P •POST http://k •+ •No environment •

100

http://localhost:3001/api/v1/emp/employees

Save •Share

</>

POST •http://localhost:3001/api/v1/emp/employees

Send •

ParamsAuthorizationHeaders (9)Body •ScriptsSettings

cookies

none •form-data •x-www-form-urlencoded •raw •binary •GraphQLJSON •

Beautify

1 {  
2   "first\_name": "Mohammad",  
3   "last\_name": "smith",  
4   "email": "M.smith@gmail.com",  
5   "position": "Designer",  
6   "salary": 85000,  
7   "date\_of\_joining": "2023-08-10",  
8   "department": "Design"  
9 }  
10

BodyCookiesHeaders (7)Test Results

201 Created • 96 ms • 325 B • Save Response •••

PrettyRawPreviewVisualizeJSON •

1 {  
2   "message": "Employee created successfully.",  
3   "employee\_id": "670a9b61e3f97e42b034a310"  
4 }

## Get All Employees:

The screenshot displays a REST client interface with a tab for the endpoint `http://localhost:3001/api/v1/emp/employees`. The request method is set to `GET`. Below the URL bar, tabs for `Params`, `Authorization`, `Headers (7)`, `Body`, `Scripts`, and `Settings` are visible. The `Body` tab is active, showing radio buttons for `none`, `form-data`, `x-www-form-urlencoded`, `raw`, `binary`, and `GraphQL`. A message states: "This request does not have a body".

Below the request configuration, the response section shows a status of `200 OK`, a response time of `86 ms`, and a size of `763 B`. The response body is displayed in the `Pretty` view, showing a JSON array with one employee object:

```
1 [
2   {
3     "_id": "670a9b3de3f97e42b034a30e",
4     "first_name": "Mobina",
5     "last_name": "Zargary",
6     "email": "Mobina.zargary@gmail.com",
7     "position": "Designer",
8     "salary": 85000,
9     "date_of_joining": "2023-08-10T00:00:00.000Z",
10    "department": "Design",
11    "created_at": "2024-10-12T15:52:29.757Z",
12    "__v": 0
13  },
14 ]
```

OverviewGET http://lo...GET Untitled Iproduct-tPOST CreateGET Get ProcPUT Update !DEL Delete PGET http://lo...+vNo environment

http://localhost:3001/api/v1/emp/employeesSaveShare</>

GEThttp://localhost:3001/api/v1/emp/employeesSend

ParamsAuthorizationHeaders (7)BodyScriptsSettingsCookies

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

This request does not have a body

BodyCookiesHeaders (7)Test Results200 OK86 ms763 BSave Response

PrettyRawPreviewVisualizeJSON

```
11  {
12    "created_at": "2024-10-12T15:53:05.918Z",
13    "__v": 0
14  },
15  {
16    "_id": "670a9b61e3f97e42b034a310",
17    "first_name": "Mohammad",
18    "last_name": "smith",
19    "email": "M.smith@gmail.com",
20    "position": "Designer",
21    "salary": 85000,
22    "date_of_joining": "2023-08-10T00:00:00.000Z",
23    "department": "Design",
24    "created_at": "2024-10-12T15:53:05.918Z",
25    "__v": 0
26  }
27 ]
```

## Get Employees by Their Id:

The screenshot displays a REST client interface with a tab for the endpoint `http://localhost:3001/api/v1/emp/employees/670a9b61e3f97e42b034a310`. The request method is set to `GET`. The response status is `200 OK` with a response time of `83 ms` and a body size of `495 B`. The response body is shown in the `JSON` format, displaying the following data:

```
1 {
2   "_id": "670a9b61e3f97e42b034a310",
3   "first_name": "Mohammad",
4   "last_name": "smith",
5   "email": "M.smith@gmail.com",
6   "position": "Designer",
7   "salary": 85000,
8   "date_of_joining": "2023-08-10T00:00:00.000Z",
9   "department": "Design",
10  "created_at": "2024-10-12T15:53:05.918Z",
11  "__v": 0
12 }
```

Update Employees:

(Updated Salary)

The screenshot shows a REST client interface with a PUT request to the endpoint `http://localhost:3001/api/v1/emp/employees/670a9b61e3f97e42b034a310`. The request body is a JSON object with the following fields: `first_name` (Mohammad), `last_name` (smith), `email` (M.smith@gmail.com), `position` (Designer), `salary` (15000), `date_of_joining` (2023-08-10), and `department` (Design). The response is a 200 OK status with a message: "Employee details updated successfully."

Overview GET http://lo... GET Untitled I product-t... POST Create GET Get Proc PUT Update I DEL Delete P PUT http://lo... + No environment

PUT http://localhost:3001/api/v1/emp/employees/670a9b61e3f97e42b034a310 Save Share

PUT http://localhost:3001/api/v1/emp/employees/670a9b61e3f97e42b034a310 Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

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

```
1 {
2   "first_name": "Mohammad",
3   "last_name": "smith",
4   "email": "M.smith@gmail.com",
5   "position": "Designer",
6   "salary": 15000,
7   "date_of_joining": "2023-08-10",
8   "department": "Design"
9 }
10
```

Body Cookies Headers (7) Test Results 200 OK • 97 ms • 287 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Employee details updated successfully."
3 }
```



## Delete Employee:

The screenshot displays a REST client interface with a top toolbar containing tabs for various HTTP methods: Overview, GET http://lo..., GET Untitled I, product--, POST Create, GET Get Proc, PUT Update I, DEL Delete P, and the active tab, DEL http://loc... The interface is set to 'No environment'.

The main area shows a DELETE request to the URL `http://localhost:3001/api/v1/emp/employees?eid=670a9b61e3f97e42b034a310`. The request is configured with the following options:

- Method: DELETE
- URL: `http://localhost:3001/api/v1/emp/employees?eid=670a9b61e3f97e42b034a310`
- Body: none (selected)
- Other options: form-data, x-www-form-urlencoded, raw, binary, GraphQL

A message states: "This request does not have a body".

The response section at the bottom shows a successful status: **200 OK** with a response time of 89 ms and a size of 279 B. The response body is displayed in JSON format:

```
1 {
2   "message": "Employee deleted successfully."
3 }
```

## Example of Express Validator:

(Missing email and salary for creating employees and get error)

The screenshot shows a Postman interface for a POST request to `http://localhost:3001/api/v1/emp/employees`. The request body is a JSON object:

```
1 {
2   "first_name": "reza",
3   "last_name": "abedi",
4   "position": "Designer",
5   "date_of_joining": "2023-08-10",
6   "department": "Design"
7 }
8
```

The response is a **400 Bad Request** with a status of 13 ms and 422 B. The response body is a JSON object with an `errors` array:

```
1 {
2   "errors": [
3     {
4       "type": "field",
5       "msg": "Valid email is required",
6       "path": "email",
7       "location": "body"
8     },
9     {
10      "type": "field",
11      "msg": "Salary must be a number",
12      "path": "salary",
13      "location": "body"
14    }
15  ]
16 }
```

Incorrect Date format:

POST http://localhost:3001/api/v1/emp/employees

Params Authorization Headers (9) Body Scripts Settings

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

```
1 {
2   "first_name": "Mohammad",
3   "last_name": "smith",
4   "email": "M.smith@gmail.com",
5   "position": "Software Engineer",
6   "salary": 90000,
7   "date_of_joining": "2023-07-157822-",
8   "department": "Engineering"
9 }
10
```

Body Cookies Headers (7) Test Results

400 Bad Request • 62 ms • 388 B •

Pretty Raw Preview Visualize JSON

```
1 {
2   "errors": [
3     {
4       "type": "field",
5       "value": "2023-07-157822-",
6       "msg": "Date of joining must be a valid date",
7       "path": "date_of_joining",
8       "location": "body"
9     }
10  ]
11 }
```

Mock Postman API:

POST

http://localhost:3001/api/v1/mock/user/signup

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (7)

Test Results

201 Created

6 ms

296 B

...

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "message": "User created successfully (mock response)."
3 }
```

POST http://localhost:3001/api/v1/mock/user/login Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (7) Test Results

200 OK • 10 ms • 282 B • 🌐 📄 ⋮

Pretty Raw Preview Visualize JSON ⌵ ⌵

```

1  {
2    |  "message": "Login successful (mock response).\"
3    |  }

```

POST

http://localhost:3001/api/v1/mock/emp/employees

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON

Beautify

1

body

Cookies

Headers (7)

Test Results

201 Created

9 ms

307 B

Pretty

Raw

Preview

Visualize

JSON

1

{

2

"message": "Employee added (mock response).",

3

"employee": {

4

"id": "3"

5

}

6

}

GET  Send

Params Authorization Headers (7) Body Scripts Settings Cook

Query Params

	Key	Value	Description	...	Bulk Ed
	Key	Value	Description		

Body Cookies Headers (7) Test Results

200 OK • 8 ms • 427 B •

Pretty Raw Preview Visualize JSON

```

1  [
2    {
3      "id": "1",
4      "first_name": "Alice",
5      "last_name": "Johnson",
6      "position": "Developer",
7      "salary": 80000
8    },
9    {
10     "id": "2",
11     "first_name": "Bob",
12     "last_name": "Smith"

```

GET
http://localhost:3001/api/v1/mock/emp/employees/1
Send

Params Authorization Headers (7) Body Scripts Settings

Cookies

#### Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (7) Test Results

200 OK • 10 ms • 326 B • e.g.

```

Pretty Raw Preview Visualize JSON
1  {
2    "id": "1",
3    "first_name": "Alice",
4    "last_name": "Johnson",
5    "position": "Developer",
6    "salary": 80000
7  }
```



mock / mock-employee

PUT http://localhost:3001/api/v1/mock/emp/employees/1 Send

Params Authorization Headers (9) Body Scripts Settings

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

```
1 {
2   "first_name": "Alice",
3   "last_name": "Johnson",
4   "position": "Developer",
5   "salary": 20000
6 }
```

ody Cookies Headers (7) Test Results 200 OK • 8 ms • 386 B • e.g.

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Employee updated (mock response).",
3   "employee": {
4     "id": "1",
5     "first_name": "Alice",
6     "last_name": "Johnson",
7     "position": "Developer",
8     "salary": 20000
9   }
10 }
```

DELETE http://localhost:3001/api/v1/mock/emp/employees?eid=1

Send

Parameters Authorization Headers (7) Body Scripts Settings

Cookies

## Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	eid	1			
	Key	Value	Description		

More actions

Cookies Headers (7) Test Results

200 OK • 11 ms • 282 B • 🌐 📄 ⋮

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "message": "Employee deleted (mock response)."  
3 }
```

