

The screenshot shows the Postman application interface. At the top, the URL is set to `http://localhost:5000/signup`. The method is selected as `POST`. On the right side, there are buttons for `Save` and `Share`, and a `No environment` indicator.

In the main workspace, the `Body` tab is active, showing the raw JSON payload sent to the server:

```
1 {  
2   "name": "Test User",  
3   "email": "test@example.com",  
4   "password": "123456"  
5 }  
6
```

Below the body, the response is displayed under the `Test Results` tab, showing a `201 Created` status with a response time of `511 ms` and a size of `298 B`.

The bottom section shows the response body in JSON format:

```
1 {  
2   "message": "User created"  
3 }
```

The screenshot shows the Postman application interface. At the top, it displays a POST request to `http://localhost:5000/signin`. The 'Body' tab is selected, showing the following JSON payload:

```
1 {  
2   "email": "test@example.com",  
3   "password": "123456"  
4 }
```

Below the request, the response details are shown: a status of 200 OK, a response time of 318 ms, and a response size of 417 B. The response body is also displayed as JSON:

```
{ } JSON
```

```
1 {  
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJpZCI6MywiaWF0IjoxNzYyNzYyNTYzMjIeHAIoje3NjI3NjYxNjN9.  
iFLScVQHyaA74v-e4QQW6R-YMS7n3Jgere_UVSugYmk"  
3 }
```

GET http://localhost:5000/pri +

No environment

HTTP http://localhost:5000/protected

Save Share Send

GET http://localhost:5000/protected

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

Auth Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token: [REDACTED]

Body Cookies Headers (8) Test Results

200 OK • 12 ms • 354 B • [Copy](#)

{ } JSON ▾ Preview Visualize

```
1 {  
2   "message": "Protected data accessed",  
3   "user": {  
4     "id": 3,  
5     "iat": 1762762563,  
6     "exp": 1762766163  
7   }  
8 }
```

The screenshot shows the Postman application interface. At the top, there is a search bar labeled "Search Postman" and a "Ctrl K" keyboard shortcut. To the right are buttons for "Invite", "Upgrade", and "Minimize". Below the header, a URL bar shows "GET http://localhost:5000/users". The main workspace displays an "HTTP" request for "http://localhost:5000/users" with a "GET" method. The "Authorization" tab is selected, showing "Bearer Token" as the auth type with a token field containing a redacted string. Other tabs include "Params", "Headers (9)", "Body", "Scripts", and "Settings". On the right, there are "Cookies" and "Send" buttons. Below the request details, a status bar shows "200 OK" with a response time of "140 ms" and a size of "376 B". The "Body" tab is active, showing a JSON response:

```
1 [  
2   {  
3     "id": 1,  
4     "name": "Jerry",  
5     "email": "jerry@example.com"  
6   },  
7   {  
8     "id": 3,  
9     "name": "Test User",  
10    "email": "test@example.com"  
11  ]
```

GET http://localhost:5000/users/1

No environment

HTTP http://localhost:5000/users/1

Save Share

Send

GET http://localhost:5000/users/1

Params Authorization Headers (9) Body Scripts Settings Cookies

Auth Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token: [REDACTED]

Body Cookies Headers (8) Test Results 200 OK 122 ms 318 B

{ } JSON Preview Visualize

```
1 {  
2   "id": 1,  
3   "name": "Jerry",  
4   "email": "jerry@example.com"  
5 }
```

The screenshot shows the Postman application interface. At the top, there is a search bar labeled "Search Postman" and a "Ctrl K" keyboard shortcut. To the right are buttons for "Invite", "Upgrade", and "Minimizer". Below the header, a navigation bar indicates a "POST http://localhost:5000/users" request. The main workspace shows a "POST" method selected, with the URL "http://localhost:5000/users" in the request field. Below the method, tabs for "Params", "Authorization", "Headers (9)", "Body", "Scripts", and "Settings" are visible, with "Body" being the active tab. Under "Body", options for "none", "form-data", "x-www-form-urlencoded", "raw", "binary", "GraphQL", and "JSON" are shown, with "raw" selected. To the right of the body editor are "Cookies" and "Beautify" buttons. The body content is a JSON object:

```
1 {  
2   "name": "Nuevo User",  
3   "email": "nuevo@example.com",  
4   "password": "abc123"  
5 }  
6
```

At the bottom of the interface, a status bar displays "201 Created" with a timestamp of "291 ms" and a size of "328 B". Other icons in the status bar include a refresh, a magnifying glass, and a copy symbol.

The screenshot shows the Postman interface. At the top, it says "PUT http://localhost:5000/users/1". Below that, the URL is again shown as "http://localhost:5000/users/1". The "Body" tab is selected, showing the raw JSON payload:

```

1  {
2    "name": "Test User Updated",
3    "email": "testupdated@example.com"
4  }
5
6

```

At the bottom right, the response status is "200 OK" with a response time of "132 ms" and a size of "336 B". The response body is also displayed in JSON format:

```

1  {
2    "id": 1,
3    "name": "Test User Updated",
4    "email": "testupdated@example.com"
5  }

```

```

labo8(#   name TEXT NOT NULL,
labo8(#   email TEXT UNIQUE NOT NULL,
labo8(#   password TEXT NOT NULL
labo8(# );
CREATE TABLE
labo8=# \d users;
                                         Tabla %public.users%
 Columna |  Tipo   | Ordenamiento | Nullable |          Por omisi%n
-----+-----+-----+-----+-----+
 id     | integer |             | not null | nextval('users_id_seq'::regclass)
 name   | text    |             | not null |
 email  | text    |             | not null |
 password | text   |             | not null |
=ndices:
 "users_pkey" PRIMARY KEY, btree (id)
 "users_email_key" UNIQUE CONSTRAINT, btree (email)

labo8=# INSERT INTO users (name, email, password)
labo8-# VALUES (
labo8(#   'Jerry',
labo8(#   'jerry@example.com',
labo8(#   '$2b$10$XOHI0.vg73ve90y73lqU5.W//tVQuqMIAKJn3n25ZA2DUF.YGm6vq'
labo8(# );
INSERT 0 1

```

POST http://localhost:5000/signin

HTTP <http://localhost:5000/signin>

Save Share

POST http://localhost:5000/signin

Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

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

```
1 {  
2   "email": "jerry@example.com",  
3   "password": "1234"  
4 }
```

Body Cookies Headers (8) Test Results 200 OK 269 ms 417 B

{ } JSON Preview Visualize

```
1 {  
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
         eyJpZCI6NSwiaWF0IjoxNzYyNzYOMjE5LCJleHAiOjE3NjI3Njc4MTl9.  
         rdDnL2RqrPdIvg9BN5fVndD8_TZjfw_4uWvmjUsYDwc"  
3 }
```

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:5000/protected`. The response status is `200 OK`, time `63 ms`, size `354 B`.

Body (JSON)

```
1 {  
2   "message": "Protected data accessed",  
3   "user": {  
4     "id": 5,  
5     "iat": 1762764219,  
6     "exp": 1762767819  
7   }  
8 }
```

CRUD JERRY

GET http://localhost:5000/users

No environment

HTTP http://localhost:5000/users

Save Share Send

GET http://localhost:5000/users

Params Authorization Headers (9) Body Scripts Settings Cookies

Auth Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token: [REDACTED]

Body Cookies Headers (8) Test Results 200 OK 156 ms 433 B

{ } JSON Preview Visualize

```
7 {  
8   "id": 4,  
9   "name": "Nuevo User",  
10  "email": "nuevo@example.com"  
11 },  
12 {  
13   "id": 5,  
14   "name": "Jerry",  
15   "email": "jerry@example.com"  
16 }
```

GET http://localhost:5000/users/3

No environment

HTTP http://localhost:5000/users/3

Save Share Send

GET http://localhost:5000/users/3

Params Authorization Headers (9) Body Scripts Settings Cookies

Auth Type Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token

.....

Body Cookies Headers (8) Test Results 200 OK 219 ms 321 B

{ } JSON Preview Visualize

```
1 {  
2   "id": 3,  
3   "name": "Test User",  
4   "email": "test@example.com"  
5 }
```

The screenshot shows the Postman application interface. At the top, there is a search bar labeled "Search Postman" and a "Ctrl K" keyboard shortcut. On the right side of the header are buttons for "Invite", "Upgrade", and "Minimize". Below the header, the URL "http://localhost:5000/users/1" is entered into the address bar, with a "DELETE" method selected. To the right of the address bar are "Save", "Share", and "Send" buttons. The "Send" button is highlighted in blue.

The main workspace has tabs for "Params", "Authorization", "Headers (9)", "Body", "Scripts", and "Settings". The "Body" tab is currently active and set to "raw" mode, indicated by a blue dot. The body content is a JSON object:

```
1 {  
2   "name": "Test User Updated",  
3   "email": "testupdated@example.com"  
4 }  
5  
6
```

To the right of the body editor are "Cookies" and "Beautify" buttons. Below the body editor, there are tabs for "Body", "Cookies", "Headers (8)", and "Test Results". The "Test Results" tab is active, showing a green "200 OK" status, a response time of "140 ms", and a response size of "293 B". The response body is displayed as:

```
{ } JSON ▾ ▶ Preview ⌂ Visualize ▾
```

```
1 {  
2   "message": "User deleted"  
3 }
```

POST http://localhost:5000/users

No environment

HTTP http://localhost:5000/users

Save Share

POST http://localhost:5000/users

Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

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

1 {"name": "Jerry Friend", "email": "friend@example.com", "password": "12345"}

Body Cookies Headers (8) Test Results

201 Created 265 ms 331 B

[{}] JSON Preview Visualize

1 {
2 | "id": 7,
3 | "name": "Jerry Friend",
4 | "email": "friend@example.com"
5 }

GET http://localhost:5000/users/7

HTTP <http://localhost:5000/users/7>

Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

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

```
1 {"name": "Jerry Updated", "email": "jerry@example.com"}  
2
```

Body Cookies Headers (8) Test Results

200 OK 106 ms 326 B

```
{ } JSON Preview Visualize
```

```
1 {  
2   "id": 7,  
3   "name": "Jerry Friend",  
4   "email": "friend@example.com"  
5 }
```

DEL http://localhost:5000/users/3

HTTP <http://localhost:5000/users/3>

Send

Params Authorization Headers (7) Body Scripts Settings Cookies Beautify

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

```
1 Ctrl+Alt+P to Ask AI
```

Body Cookies Headers (8) Test Results

200 OK 115 ms 293 B

```
{ } JSON Preview Visualize
```

```
1 {  
2   "message": "User deleted"  
3 }
```