

## ScreenShot Home

The screenshot displays the API Network application interface. The top navigation bar includes a search icon, a user icon, a settings icon, a bell icon, a GitHub icon, and an "Upgrade" button. The main workspace is titled "My Workspace" and contains a "New" button and an "Import" button. The left sidebar shows a "Collections" section with a tree view of "API\_USERS" containing several endpoints: "GET Home", "GET ListUser", "GET SpecificUserList", "POST NewUser", "PUT UpdateUser", and "DEL DeleteUser". Below this is a "New Collection" section with a message "This collection is empty" and a link "Add a request" to start working.

The main area shows a selected endpoint "GET Home" with a URL "http://localhost:3000". The "Send" button is visible. Below the URL bar, there is a "Params" section with a "Query Params" table. The table has columns for "Key", "Value", "Des...", and "Bulk Edit". The first row contains the text "Key", "Value", and "Description".

At the bottom, the response body is displayed, showing a status of "200 OK" and a response time of "5 ms". The response is in HTML format, showing a single line of text: "1 Bienvenido a la API REST con Express.js".

## ScreenShot todos los usuarios

The screenshot displays the API Network application interface. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. The left sidebar shows 'My Workspace' with a list of collections: 'API\_USERS' (containing 'GET Home', 'GET ListUser', 'GET SpecificUserList', 'POST NewUser', 'PUT UpdateUser', and 'DEL DeleteUser') and 'New Collection'. The main panel shows the 'GET ListUser' endpoint with the URL 'http://localhost:3000/api/users'. The response status is '200 OK' with a response time of '4 ms' and a size of '341 B'. The response body is a JSON array of three user objects.

**API\_USERS / ListUser**

**GET** **Send**

URL: `http://localhost:3000/api/users`

**Params**

**Query Params**

Key	Value	Description
Key	Value	Description

**Body** **200 OK** • 4 ms • 341 B

**JSON** **Preview** **Visualize**

```
1 [
2   {
3     "id": 1,
4     "nombre": "Juan",
5     "edad": 28
6   },
7   {
8     "id": 2,
9     "nombre": "Ana",
10    "edad": 22
11  },
12  {
13    "id": 3,
14    "nombre": "Luis",
15    "edad": 35
16  }
17 ]
```

## ScreenShot usuario especifico

The screenshot displays the API Network application interface. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. The left sidebar shows 'My Workspace' with a list of collections: 'API\_USERS' (containing 'GET Home', 'GET ListUser', and 'GET SpecificUserList'), 'POST NewUser', 'PUT UpdateUser', and 'DEL DeleteUser'. The main panel shows the 'GET SpecificUserList' endpoint selected. The request is a GET method to the URL 'http://localhost:3000/api/users/1'. The response is a 200 OK status with a response time of 4 ms and a body size of 269 B. The response body is displayed in JSON format:

```
{
  "id": 1,
  "nombre": "Juan",
  "edad": 28
}
```

## ScreenShot nuevo usuario

The screenshot displays the Postman API client interface. The top navigation bar includes a search icon, a user profile icon, a settings icon, a notification bell, a GitHub logo, and an 'Upgrade' button. The main workspace is titled 'My Workspace' and contains a sidebar with 'Collections', 'Environments', 'Flows', and 'History'. The 'Collections' sidebar shows a collection named 'API\_USERS' with several requests: 'GET Home', 'GET ListUser', 'GET SpecificUserList', 'POST NewUser', 'PUT UpdateUser', and 'DEL DeleteUser'. The 'POST NewUser' request is selected. The main panel shows the details of this request, including the method 'POST', the URL 'http://localhost:3000/api/users', and the body content in JSON format: 

```
{  "nombre": "Carlos",  "edad": 30}
```

. The 'Response' tab at the bottom is currently empty, displaying a cartoon astronaut and the text 'Click Send to get a response'.

API\_USERS / NewUser

POST http://localhost:3000/api/users

Body

raw JSON

```
1 {
2   "nombre": "Carlos",
3   "edad": 30
4 }
```

Response History

Click Send to get a response

## ScreenShot actualizar usuario

The screenshot displays the API Network application interface. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. The left sidebar shows 'My Workspace' with a list of collections: 'API\_USERS' (containing 'GET Home', 'GET ListUser', 'GET SpecificUserList', 'POST NewUser', 'PUT UpdateUser', and 'DEL DeleteUser') and 'New Collection'. The main panel shows the 'PUT UpdateUser' endpoint with the URL 'http://localhost:3000/api/users/1'. The 'Send' button is highlighted. Below the URL bar, the 'Query Params' section is visible, showing a table with columns 'Key', 'Value', 'Des...', and 'Bulk Edit'. The response body is displayed as JSON, showing a successful '200 OK' status with a response time of 15 ms and a size of 270 B. The JSON body is: 

```
{  "id": 1,  "nombre": "Oscar",  "edad": 28}
```

API\_USERS / UpdateUser

PUT http://localhost:3000/api/users/1

Send

Params

Query Params

Key	Value	Des...	Bulk Edit
Key	Value	Description	

Body

200 OK • 15 ms • 270 B

JSON

```
1 {
2   "id": 1,
3   "nombre": "Oscar",
4   "edad": 28
5 }
```

## ScreenShot borrar usuario

The screenshot shows the Postman API client interface. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. The left sidebar shows 'My Workspace' with a list of collections: 'API\_USERS' (containing 'GET Home', 'GET ListUser', 'GET SpecificUserList', 'POST NewUser', 'PUT UpdateUser', and 'DEL DeleteUser') and 'New Collection'. The main panel displays the 'DeleteUser' endpoint with a 'DELETE' method and the URL 'http://localhost:3000/api/users/3'. The 'Send' button is highlighted. Below the URL bar, the 'Query Params' table is empty. The response section shows a '200 OK' status with a response time of '3 ms' and a size of '271 B'. The response body is displayed in JSON format:

```
1 [
2   {
3     "id": 3,
4     "nombre": "Luis",
5     "edad": 35
6   }
7 ]
```