

## POST

The screenshot shows the Postman application interface. On the left, the sidebar includes 'Team Workspace' (highlighted), 'New', 'Import', 'Collections' (selected), 'Environments', 'Flows', and 'History'. The main area displays a collection named 'My Collection' with two items: 'GET Get data' and 'POST Post data'. The 'POST Post data' item is selected, showing a POST request to 'http://localhost:8080/api/users'. The 'Body' tab is active, showing the following JSON payload:

```
1 {  
2   "firstName": "mark",  
3   "lastName": "mayor",  
4   "email": "markyymr@gmail.com"  
5 }
```

The 'Body' tab also includes options for 'Params', 'Authorization', 'Headers (8)', 'Scripts', and 'Settings'. Below the body, the 'Body' tab is again active, showing the response body:

```
{ } JSON ▾
```

```
1 {  
2   "id": 1,  
3   "firstName": "mark",  
4   "lastName": "mayor",  
5   "email": "markyymr@gmail.com"  
6 }
```

The response status is '201 Created' with a duration of '421 ms' and a size of '244 B'. At the bottom, there are links for 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', and 'Trash'.

GET

The screenshot shows the Postman application interface. On the left, the sidebar includes 'Team Workspace' (highlighted), 'New', 'Import', 'Collections' (selected), 'Environments', 'Flows', and 'History'. The main workspace displays a collection named 'My Collection' containing a 'GET Get data' request. The request URL is `http://localhost:8080/api/users/1`. The 'Params' tab is selected, showing a table for 'Query Params' with one row: 'Key' and 'Value'. Below the table, the 'Body' tab is selected, showing a JSON response with the following content:

```
1 {  
2   "id": 1,  
3   "firstName": "mark ronniel",  
4   "lastName": "mayor",  
5   "email": "markyymr@gmail.com"  
6 }
```

The status bar at the bottom shows '200 OK' with a response time of '14 ms' and a size of '247 B'. Other status icons include 'Online', 'Find and replace', 'Console', 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', and 'Trash'.

## PUT / UPDATE

The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', a search bar, and various tool icons. The left sidebar features 'Team Workspace' with 'Collections', 'Environments', 'Flows', and 'History'. The main workspace displays a 'My Collection' section with 'GET Get data' and 'POST Post data' requests. A specific 'PUT http://localhost:8080/api/users/1' request is selected, showing its configuration details: method 'PUT', URL 'http://localhost:8080/api/users/1', headers (8), body (raw JSON), and settings. The body contains the following JSON payload:

```
1 {
2   "firstName": "mark",
3   "lastName": "mayor",
4   "email": "markyymr@gmail.com"
5 }
```

The response section shows a successful '200 OK' status with a response time of '81 ms' and a size of '239 B'. The response body is identical to the request body. The bottom navigation bar includes links for 'Online', 'Find and replace', 'Console', 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', and 'Trash'.

## GET ALL USERS

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Team Workspace' selected, showing 'My Collection' with a 'Get data' item. The main area has a 'GET Get data' tab selected, pointing to the URL `http://localhost:8080/api/users`. The 'Params' tab is active, showing a table for 'Query Params'. Below this, the 'Body' tab is selected, showing a JSON response with two user objects:

```
[{"id": 1, "firstName": "mark", "lastName": "mayor", "email": "markyymr@gmail.com"}, {"id": 2, "firstName": "juan", "lastName": "delacruz", "email": "juandelacruz@gmail.com"}]
```

The status bar at the bottom indicates a 200 OK response with 13 ms latency and 324 B size.

## DELETE

The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', a search bar, and various icons for invite, settings, and upgrade. The left sidebar features 'Team Workspace' with 'Collections' (selected), 'Environments', 'Flows', and 'History'. The main workspace displays a collection named 'My Collection' with two items: 'GET Get data' and 'POST Post data'. A specific DELETE request is selected, targeting the URL `http://localhost:8080/api/users/1`. The 'Params' tab is active, showing a single entry for 'Key' and 'Value'. Below the request details, the response section shows a status of '200 OK' with a response time of '72 ms' and a size of '190 B'. The response body contains the message 'User successfully deleted!'. The bottom navigation bar includes 'Online', 'Find and replace', 'Console', 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', and 'Trash'.

HTTP `http://localhost:8080/api/users/1`

DELETE `http://localhost:8080/api/users/1`

Params Authorization Headers (8) Body Scripts Settings

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	...

Body Cookies Headers (5) Test Results

Raw Preview Visualize

1 User successfully deleted!

200 OK • 72 ms • 190 B

Online Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash