

# Screenshot

1

Out of 7

Method	Description	Status Code
GET	Using the '/notes' route, retrieving all notes	200

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing collections like 'CIS 385 - Assignment 3' and 'CIS385 - Fake API'. The main area displays a 'CIS 385 - Assignment 3 / Get All Notes' collection. A specific GET request for '/api/v1/notes' is selected, with the URL set to 'http://127.0.0.1:5000/api/v1/notes'. The 'Params' tab shows a single parameter 'Key' with value 'Value'. The 'Body' tab shows a JSON response with three objects:

```
1  "1": {  
2      "title": "First Note Title",  
3      "body": "This is some note one text. This is some note one text. "  
4  },  
5  "2": {  
6      "title": "Note Two Title",  
7      "body": "Note Two Text. Note Two Text. Note Two Text. "  
8  },  
9  "3": {  
10     "title": "The Third Title",  
11     "body": "Some placeholder text for Note 3."  
12  },  
13  
14
```

The status bar at the bottom indicates a 200 OK response with 4 ms latency.

# Screenshot

2

Out of 7

Method	Description	Status Code
GET	Using the '/notes/<id>' route, retrieving note by ID	200

The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with collections like 'CIS 385 - Assignment 3' and 'CIS385 - Fake API'. The main workspace shows a 'CIS 385 - Assignment 3 / Get Note by ID' collection. A 'GET' request is selected with the URL `http://127.0.0.1:5000/api/v1/notes/1`. The 'Headers' tab shows 'Content-Type: application/json'. The 'Body' tab is set to 'Pretty' and displays the JSON response:

```
1 "title": "First Note Title",
2 "body": "This is some note one text. This is some note one text. "
```

The status bar at the bottom indicates a 200 OK response with 8 ms latency and 255 B size.

# Screenshot

3

Out of 7

Method	Description	Status Code
GET	Using the '/notes/<title>' route, retrieving note by Title	200

The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with collections like 'CIS 385 - Assignment 3' and 'CIS385 - Fake API'. The main workspace shows a 'CIS 385 - Assignment 3 / Get Note by Title' collection. A GET request is selected with the URL `http://127.0.0.1:5000/api/v1/notes/Note%20Two%20Title`. The 'Headers' tab shows 'Content-Type: application/json'. The 'Body' tab displays the response in JSON format:

```
1 "title": "Note Two Title",
2 "body": "Note Two Text. Note Two Text. Note Two Text. "
```

The status bar at the bottom indicates a 200 OK response with 6 ms latency and 241 B size.

# Screenshot

4

Out of 7

Method	Description	Status Code
PUT	Using the '/notes/<id>' route, updating note by ID	201

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main workspace is titled "CIS 385 - Assignment 3 / Update Note by ID". A PUT request is selected with the URL "http://127.0.0.1:5000/api/v1/notes/2". The "Body" tab is active, showing a JSON payload:

```
1
2   "title": "Note Two Title UPDATED",
3   "body": "Note Two Text. Note Two Text. Note Two Text. UPDATED"
4
```

Below the request, the response is displayed with a status of 201 CREATED, 5 ms duration, and 262 B size. The response body is identical to the request body.

# Screenshot

5

Out of 7

Method	Description	Status Code
PUT	Using the '/notes/<title>' route, updating note by Title	201

The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with collections like 'CIS 385 - Assignment 3' and 'CIS385 - Fake API'. The main workspace shows a 'PUT Update Note by Title' request. The 'Body' tab is selected, containing the following JSON payload:

```
1 ... "title": "UPDATED The Third Title",
2 ... "body": "UPDATED Some placeholder text for Note 3."
```

Below the request, the response pane shows a 201 CREATED status with a response body identical to the request's body.

# Screenshot

6

Out of 7

Method	Description	Status Code
POST	Using the '/notes' route, adding a new note Returning the new ID	201

The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with collections like 'CIS 385 - Assignment 3' and 'MERN Tutorial'. The main workspace shows a POST request to 'http://127.0.0.1:5000/api/v1/notes'. The 'Body' tab is selected, showing a JSON payload:

```
1 {  
2   "title": "FORTH note TITLE",  
3   "body": "This is some text for the FORTH note "  
4 }
```

The response at the bottom indicates a successful 201 CREATED status with 4 ms response time.

# Screenshot

7

Out of 7

Method	Description	Status Code
DELETE	Using the '/notes/<id>' route, deleting the note by ID Returning the deleted ID	204

The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with collections like 'CIS 385 - Assignment 3' containing various API endpoints (GET, PUT, POST) and a selected 'DEL Delete Note by ID'. Other sections include 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. The main workspace shows a DELETE request to 'http://127.0.0.1:5000/api/v1/notes/1'. The 'Params' tab is active, showing a single parameter 'Key' with value 'Value'. The 'Body' tab shows a JSON response with the number '1'. The status bar at the bottom indicates a 204 NO CONTENT response with 4 ms duration and 134 B size.