

Movie add localhost:2010

The screenshot shows the Postman application with a workspace named 'ujianbackend'. On the left sidebar, under the 'Collections' tab, the 'addmovie' request is selected. The main panel displays the details of this POST request. The URL is set to `{{url}}/add/movie`. The request body is in raw JSON format, containing the following data:

```
1 {
2   "nama": "donald duck",
3   "tahun": "1998",
4   "deskripsi": "lucu banget"
5 }
```

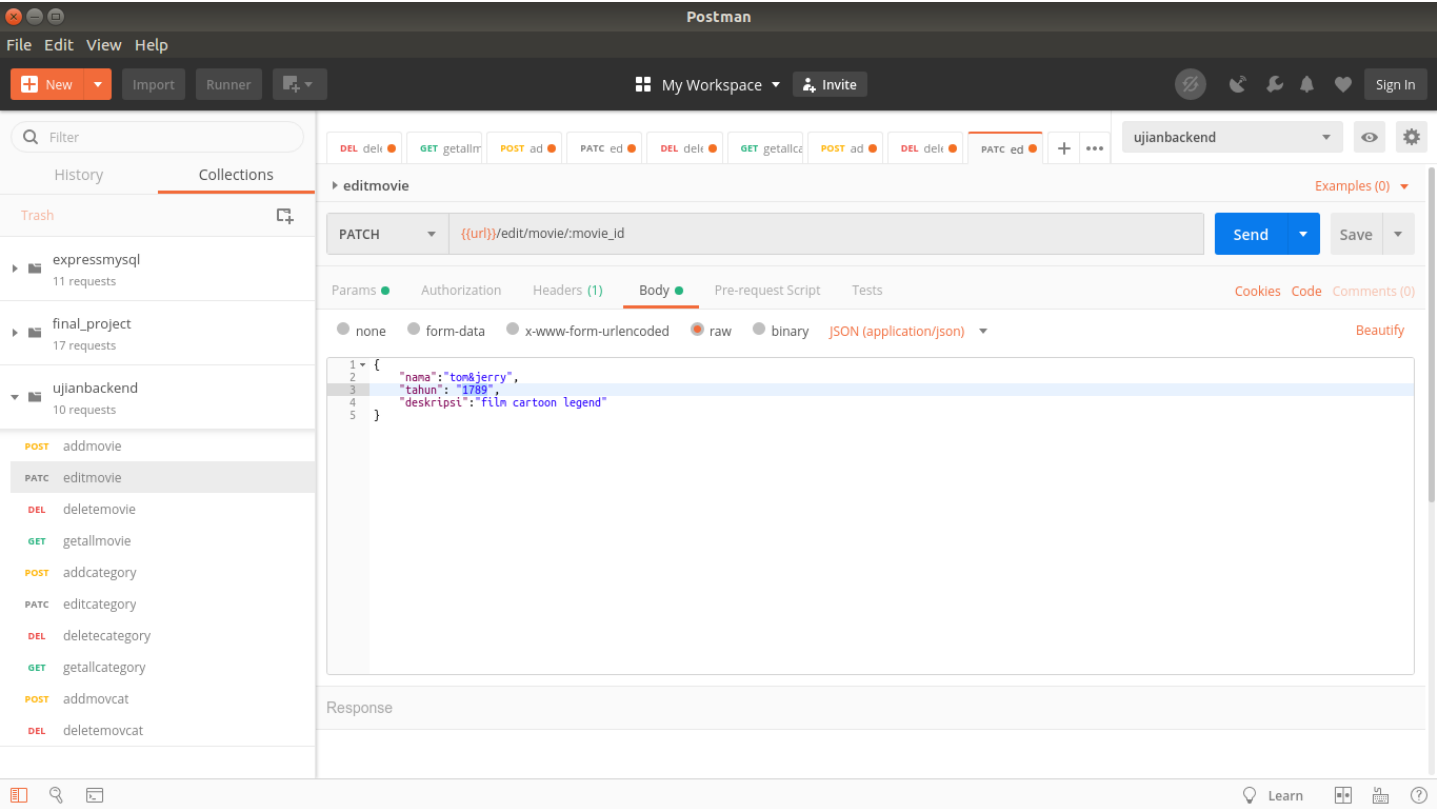
The status bar at the bottom indicates a successful response with a status of 200 OK, a time of 31 ms, and a size of 340 B. The response body is shown in the bottom panel in JSON format.

Movie delete

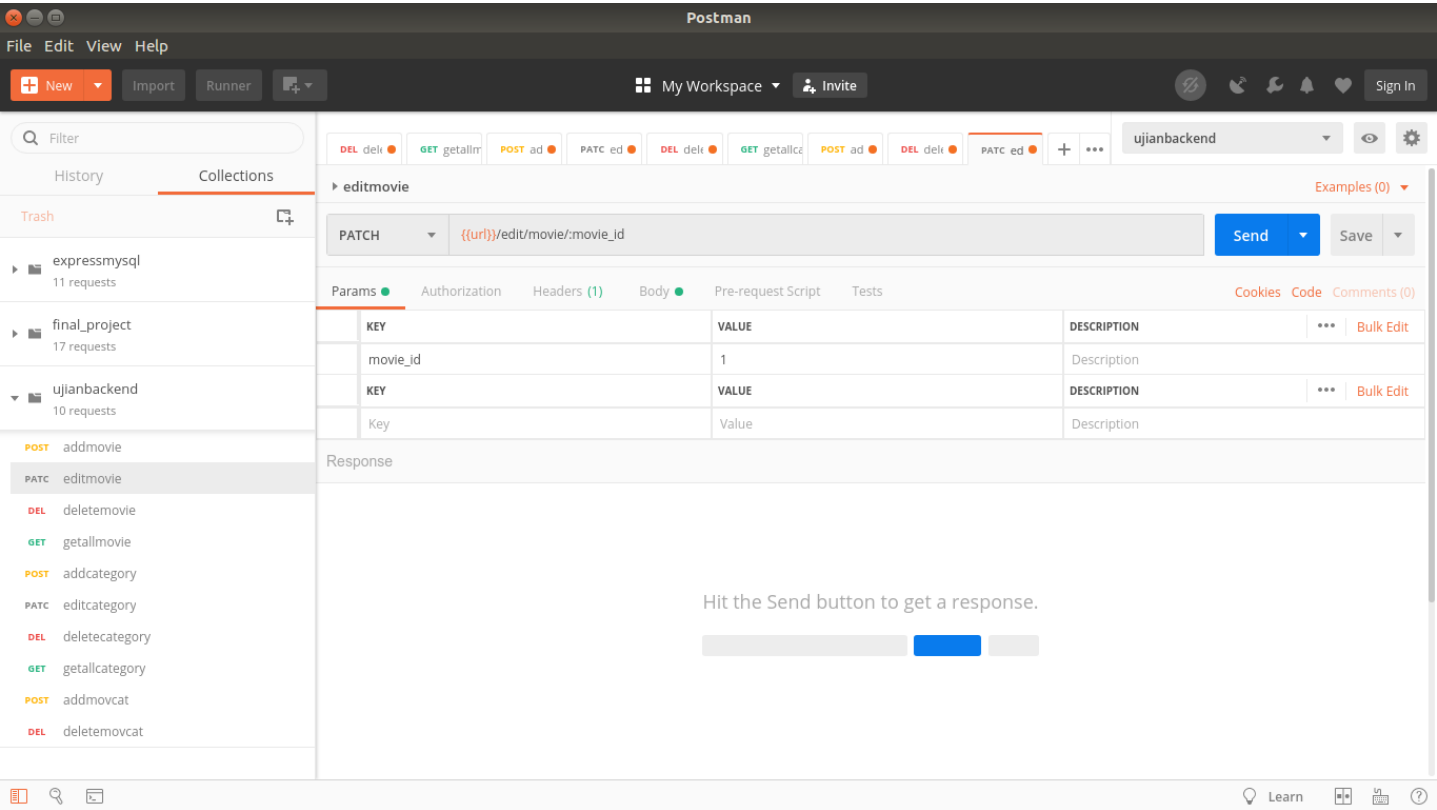
The screenshot shows the Postman application with the same 'ujianbackend' workspace. The 'deletemovie' request is selected in the left sidebar. The main panel shows the details of this DELETE request. The URL is set to `{{url}}/delete/movie/:movie_id`. The 'Params' tab is active, showing a parameter named 'movie_id' with a value of '6'. The status bar at the bottom indicates a successful response with a status of 200 OK, a time of 19 ms, and a size of 340 B. The response body is shown in the bottom panel in JSON format:

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 0,
5   "serverStatus": 2,
6   "warningCount": 0,
7   "message": "",
8   "protocol41": true,
9   "changedRows": 0
10 }
```

Movie edit by body



Movie edit by params



Movie show all

Postman

File Edit View Help

New Import Runner My Workspace Invite Sign In

Filter

History Collections

Trash

expressmysql 11 requests

final_project 17 requests

ujianbackend 10 requests

POST addmovie

PATC editmovie

DEL deletemovie

GET getallmovie

POST addcategory

PATC editcategory

DEL deletecategory

GET getallcategory

POST addmovcat

DEL deletemovcat

DEL deletemovie GET getallmovie

getallmovie Examples (0)

GET {{url}}/get/movie Send Save

Params Authorization Headers Body Pre-request Script Tests Cookies Code Comments (0)

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 14 ms Size: 439 B Save Download

Pretty Raw Preview JSON

```
1 {
2   "id": 1,
3   "nama": "tom&jerry",
4   "tahun": 1789,
5   "deskripsi": "filn cartoon legend"
6 },
7
8 {
9   "id": 2,
10  "nama": "micky mouse",
11  "tahun": 1789,
12  "deskripsi": "lebih lucu dari donalduck"
13 },
14
15 {
16   "id": 5,
17   "nama": "donalduck",
18   "tahun": 1990,
19   "deskripsi": "lucu banget"
20 }
```

Learn

Category add

The screenshot shows the Postman application with a workspace named 'ujianbackend'. The 'Collections' panel on the left lists several request collections, including 'ujianbackend' which contains a 'POST addcategory' request. The main editor displays the details of this request:

- Method:** POST
- URL:** `{{url}}/add/category`
- Body:** The body is set to 'raw' with the JSON content:

```
{  "name": "sicon"}
```
- Headers:** One header is present: 'Content-Type' with value 'application/json'.
- Status:** 200 OK
- Time:** 18 ms
- Size:** 340 B

Category delete

The screenshot shows the Postman application with the same workspace. The 'Collections' panel now includes a 'DELETE deletecategory' request. The main editor displays the details of this request:

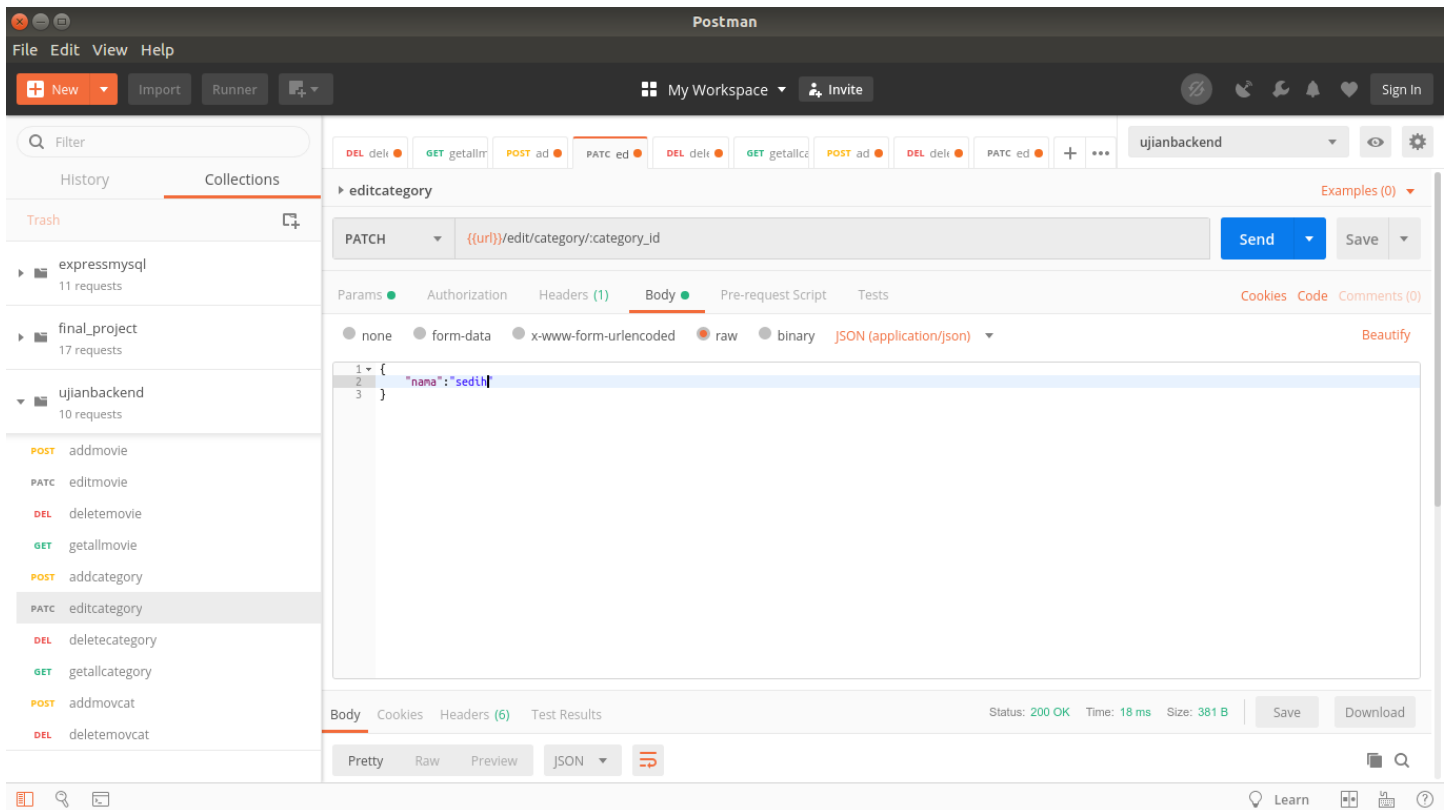
- Method:** DELETE
- URL:** `{{url}}/delete/category/:category_id`
- Params:** A table of parameters is shown:

KEY	VALUE	DESCRIPTION
category_id	5	Description
Key	Value	Description

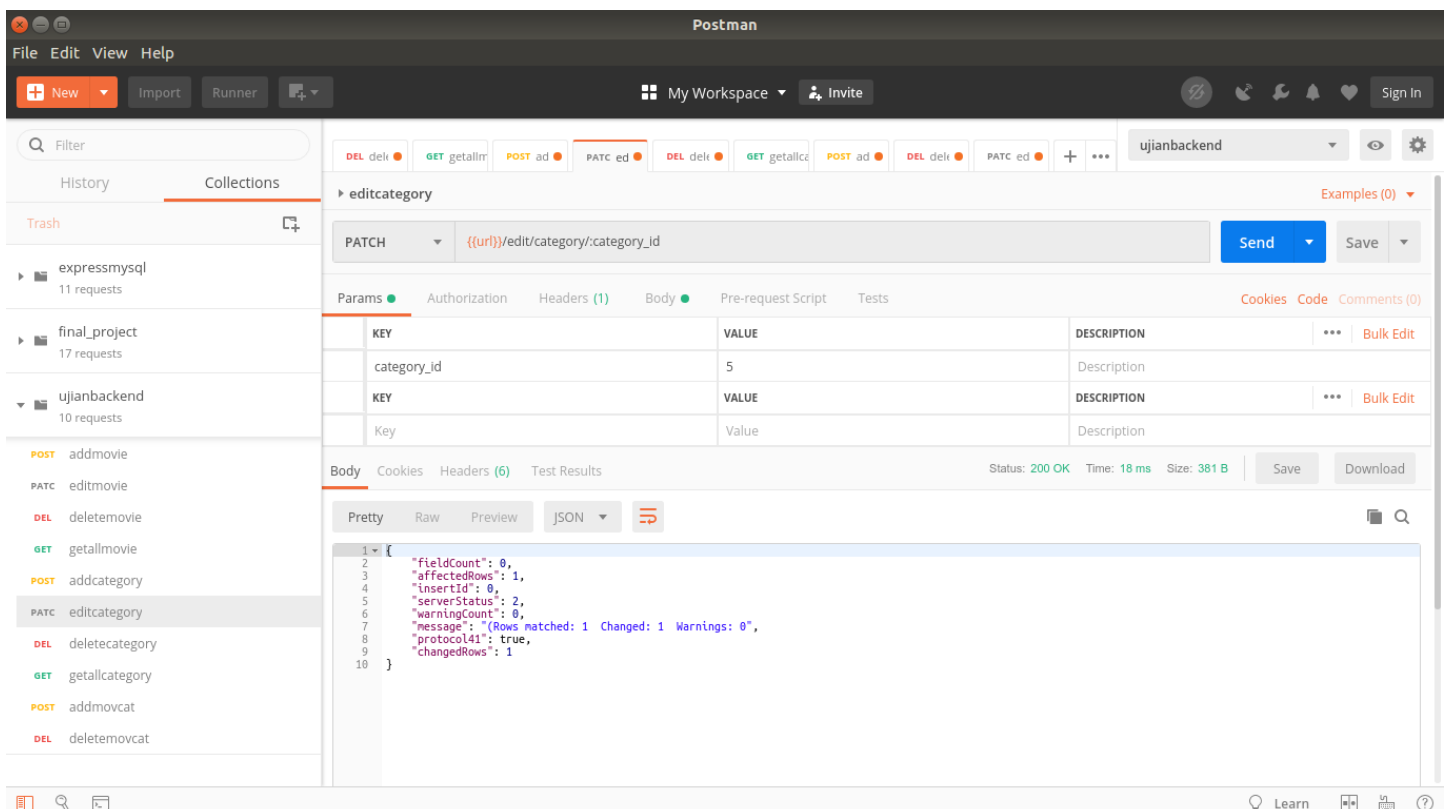
- Body:** The body is set to 'raw' with the JSON content:

```
{  "fieldCount": 0,  "affectedRows": 1,  "insertId": 0,  "serverStatus": 2,  "warningCount": 0,  "message": "",  "protocol41": true,  "changedRows": 0}
```
- Status:** 200 OK
- Time:** 15 ms
- Size:** 340 B

Category edit by body



Category edit by params



Category show all

Postman

File Edit View Help

New Import Runner My Workspace Invite Sign In

Filter

History Collections

Trash

expressmysql 11 requests

final_project 17 requests

ujianbackend 10 requests

POST addmovie

PATC editmovie

DEL deletemovie

GET getallmovie

POST addcategory

PATC editcategory

DEL deletecategory

GET getallcategory

POST addmovcat

DEL deletemovcat

getallcategory

GET {{url}}/get/category

Send Save

Params Authorization Headers Body Pre-request Script Tests Cookies Code Comments (0)

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (6) Test Results Status: 200 OK Time: 7 ms Size: 287 B Save Download

Pretty Raw Preview JSON

```
1 [
2   {
3     "id": 1,
4     "nama": "comedy"
5   },
6   {
7     "id": 2,
8     "nama": "action"
9   },
10  {
11    "id": 4,
12    "nama": "drama"
13  }
14 ]
```

Learn

Movcat add

The screenshot shows the Postman application with a workspace named 'ujianbackend'. The 'Collections' tab is active, showing a collection named 'addmovcat'. The selected request is a POST request to the endpoint `{{url}}/add/movcat/:movie_id/:category_id`. The request body is in JSON format, containing three objects representing movie categories:

```
1 {
2   "idMovcat": 2,
3   "NanaMovie": "micky mouse",
4   "NanaCategory": "comedy"
5 }
6 {
7   "idMovcat": 7,
8   "NanaMovie": "donalduck",
9   "NanaCategory": "comedy"
10 }
11 {
12   "idMovcat": 9,
13   "NanaMovie": "ton&jerry",
14   "NanaCategory": "drana"
15 }
16 }
17
```

The response status is 200 OK, with a time of 10 ms and a size of 404 B. The response body is in JSON format, containing the same three objects as the request body.

Movcat delete

The screenshot shows the Postman application with a workspace named 'ujianbackend'. The 'Collections' tab is active, showing a collection named 'deletemovcat'. The selected request is a DELETE request to the endpoint `{{url}}/delete/movcat/:id_movcat`. The request body is in JSON format, containing a single object representing a movie category:

```
1 {
2   "fieldCount": 0,
3   "affectedRows": 1,
4   "insertId": 0,
5   "serverStatus": 2,
6   "warningCount": 0,
7   "message": "",
8   "protocol": "true",
9   "changedRows": 0
10 }
```

The response status is 200 OK, with a time of 15 ms and a size of 340 B. The response body is in JSON format, containing the same object as the request body.

Movcat show all

The screenshot shows the Postman application interface. On the left, the 'Collections' sidebar is active, displaying a collection named 'ujianbackend' with 10 requests. The main workspace shows a GET request to the endpoint `{{url}}/get/movcat`. The response is a JSON array of three movie objects, displayed in 'Pretty' format. The status bar at the bottom indicates a successful response with status 200 OK, a time of 47 ms, and a size of 365 B.

Request Details:

- Method: GET
- URL: `{{url}}/get/movcat`
- Params: None
- Body: None

Response Details:

- Status: 200 OK
- Time: 47 ms
- Size: 365 B

Response Body (JSON):

```
[
  {
    "NamaMovie": "micky mouse",
    "NamaCategory": "comedy"
  },
  {
    "NamaMovie": "donald duck",
    "NamaCategory": "comedy"
  },
  {
    "NamaMovie": "ton&jerry",
    "NamaCategory": "drana"
  }
]
```