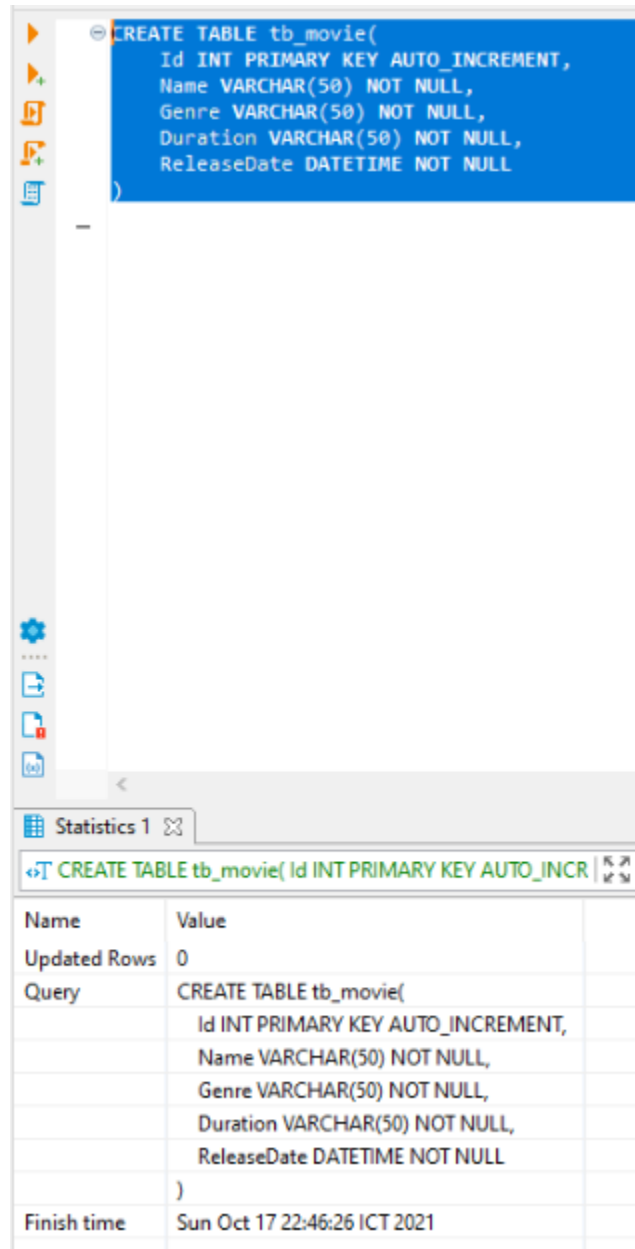


Nama Peserta : Bernard

Kode Peserta : FSDO001ONL028

Tugas : Assignment 3

Siapkan sebuah Database MySQL tempat penyimpanan data, dimana akan dilakukan proses GET, POST, PUT, DAN DELETE



The screenshot displays the MySQL Workbench interface. The top pane shows a SQL query to create a table named `tb_movie` with the following schema:

```
CREATE TABLE tb_movie(  
  Id INT PRIMARY KEY AUTO_INCREMENT,  
  Name VARCHAR(50) NOT NULL,  
  Genre VARCHAR(50) NOT NULL,  
  Duration VARCHAR(50) NOT NULL,  
  ReleaseDate DATETIME NOT NULL  
)
```

The bottom pane shows the 'Statistics 1' tab, which provides a summary of the query execution:

Name	Value
Updated Rows	0
Query	CREATE TABLE tb_movie( Id INT PRIMARY KEY AUTO_INCREMENT, Name VARCHAR(50) NOT NULL, Genre VARCHAR(50) NOT NULL, Duration VARCHAR(50) NOT NULL, ReleaseDate DATETIME NOT NULL )
Finish time	Sun Oct 17 22:46:26 ICT 2021

**Pembuatan Database**

https://localhost:5001/api/movies

POST https://localhost:5001/api/movies

Params Authorization Headers (9) Body Pre-request Script Tests Settings

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

```
1 {
2   ... "Name": "Bad Boys for Life",
3   ... "Genre": "Action/Comedy",
4   ... "Duration": "1h 41 mins",
5   ... "ReleaseDate": "2020-01-23T00:00:00"
6 }
```

Body Cookies Headers (5) Test Results

Status: 201 Created Time: 685 ms Size: 319 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "Bad Boys for Life",
4   "genre": "Action/Comedy",
5   "duration": "1h 41 mins",
6   "releaseDate": "2020-01-23T00:00:00"
7 }
```

Melakukan method POST data pertama dengan url <https://localhost:5001/api/movies>

https://localhost:5001/api/movies

POST https://localhost:5001/api/movies

Params Authorization Headers (9) Body Pre-request Script Tests Settings

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

```
1 {
2   ... "Name": "John Wick",
3   ... "Genre": "Action/Thriller",
4   ... "Duration": "1h 41 mins",
5   ... "ReleaseDate": "2014-10-24T00:00:00"
6 }
7
```

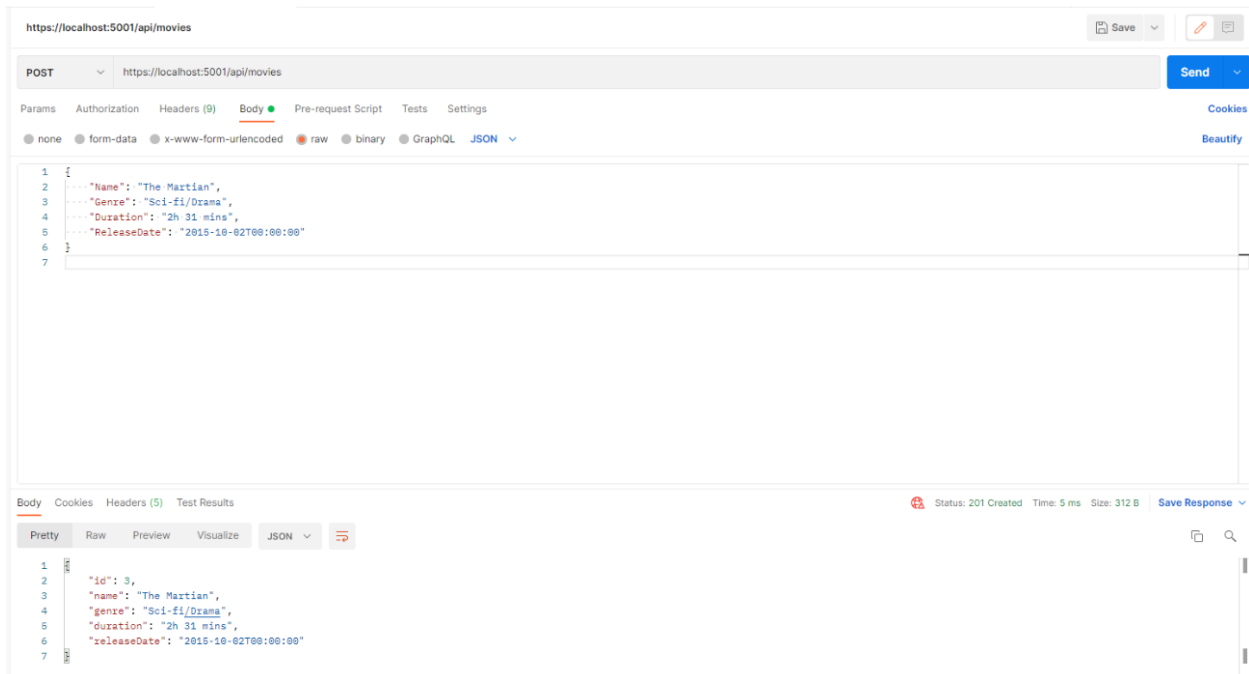
Body Cookies Headers (5) Test Results

Status: 201 Created Time: 9 ms Size: 313 B Save Response

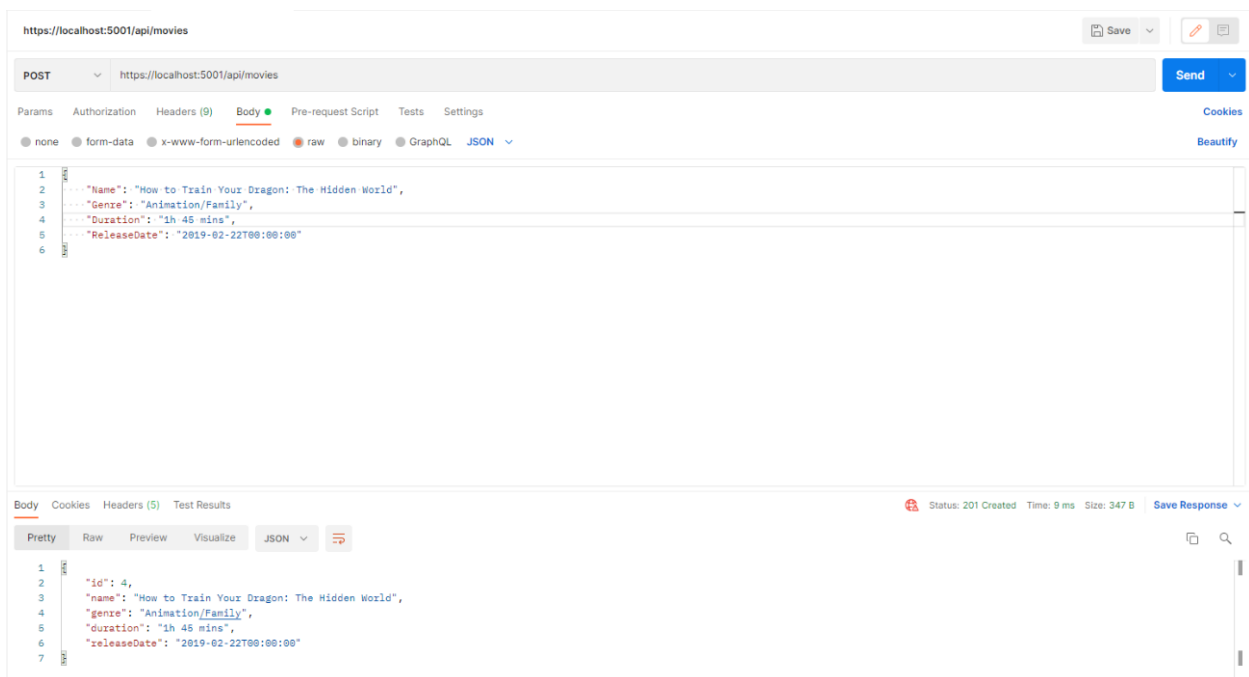
Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "name": "John Wick",
4   "genre": "Action/Thriller",
5   "duration": "1h 41 mins",
6   "releaseDate": "2014-10-24T00:00:00"
7 }
```

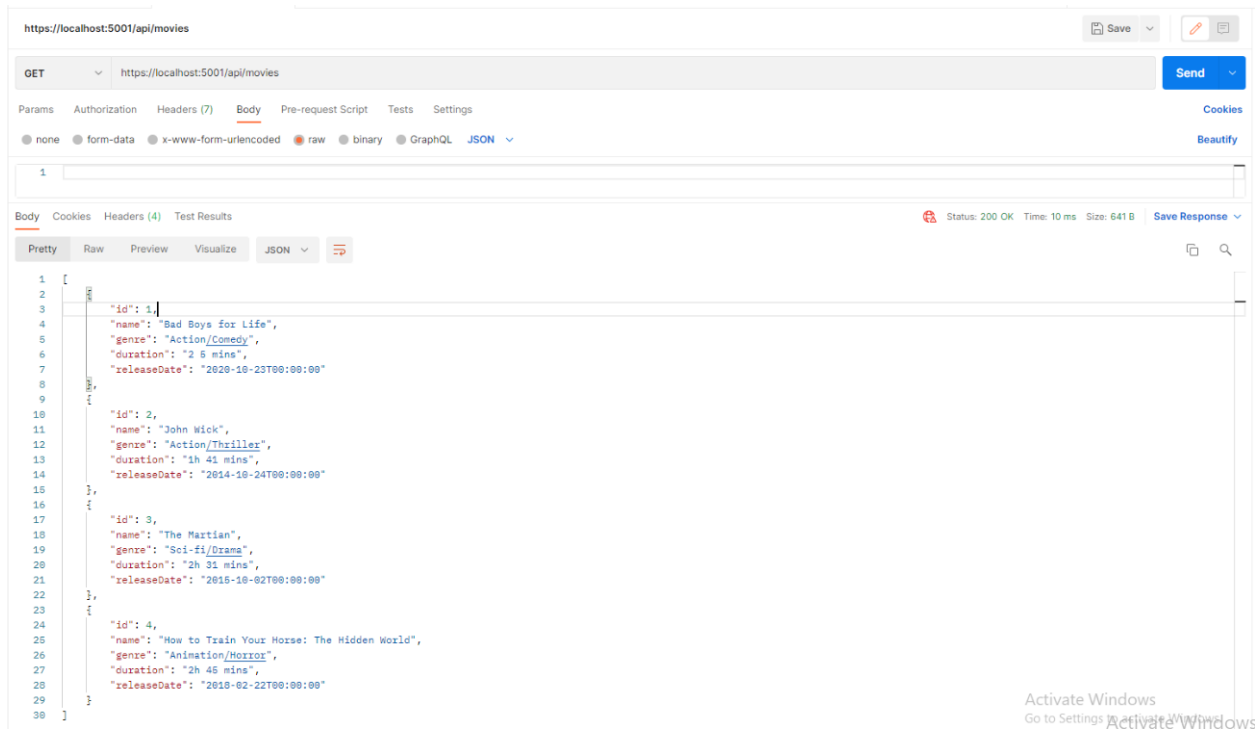
Melakukan method POST data kedua dengan url <https://localhost:5001/api/movies>



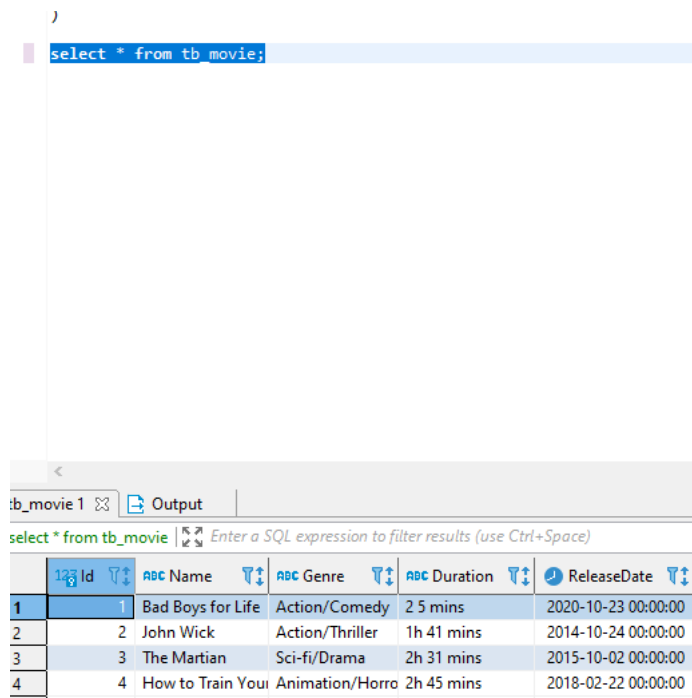
Melakukan method POST data ketiga dengan url <https://localhost:5001/api/movies>



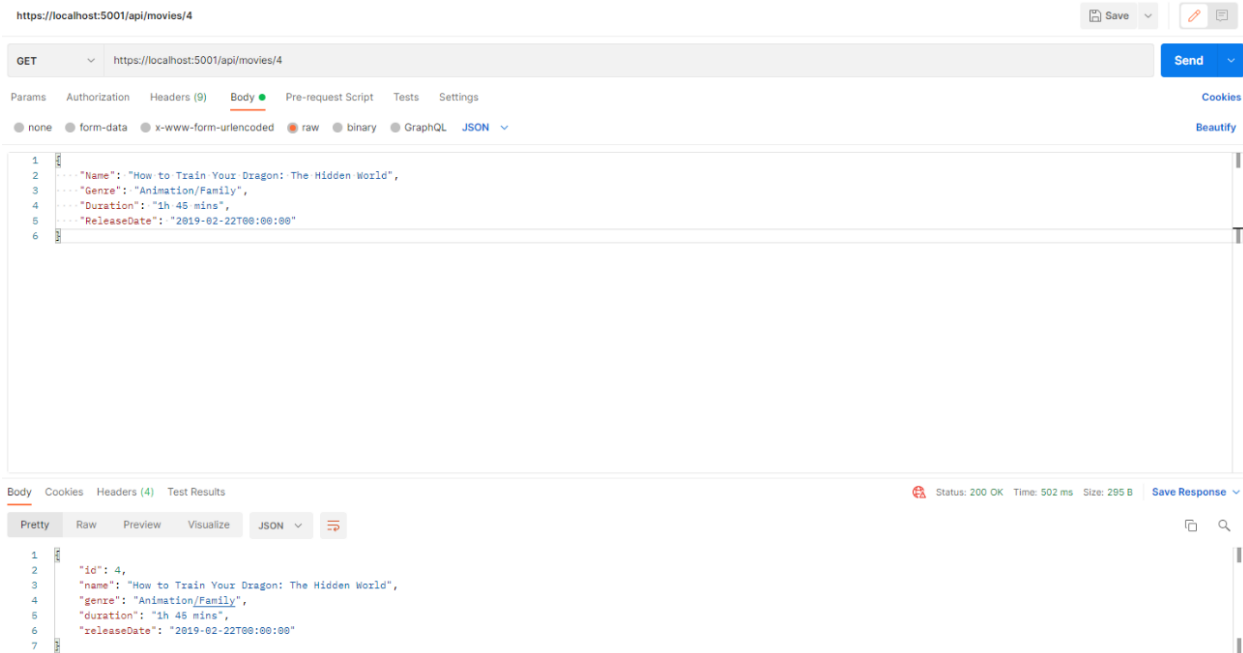
Melakukan method POST data keempat dengan url <https://localhost:5001/api/movies>



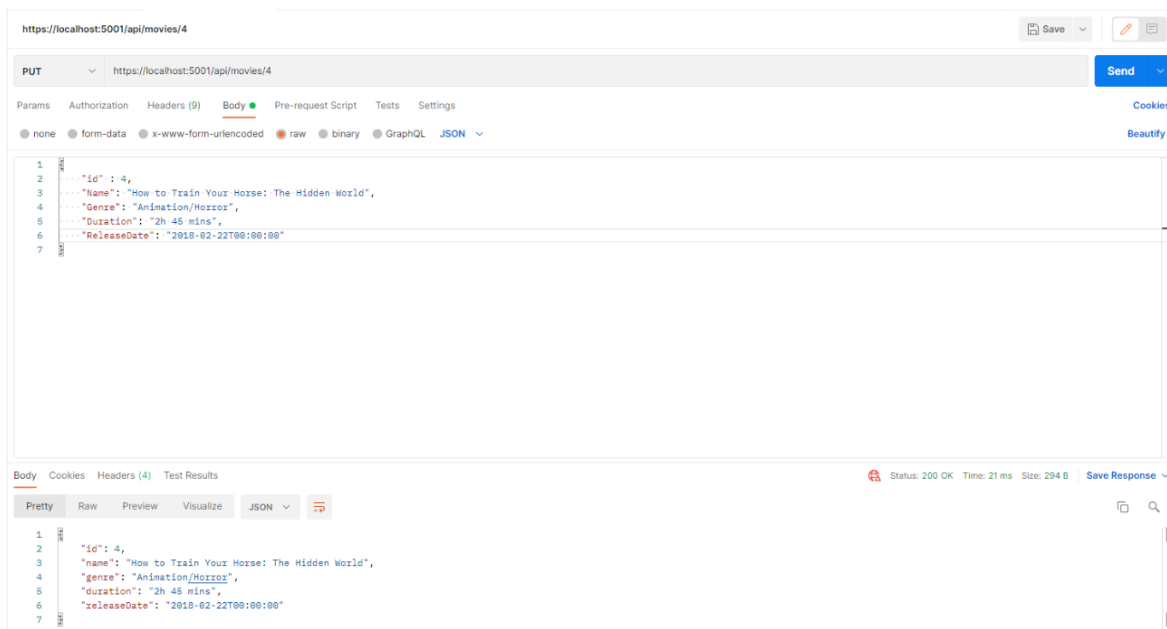
Melakukan method GET untuk menampilkan keempat data yang telah diPOST dan telah masuk ke database melalui url <https://localhost:5001/api/movies>



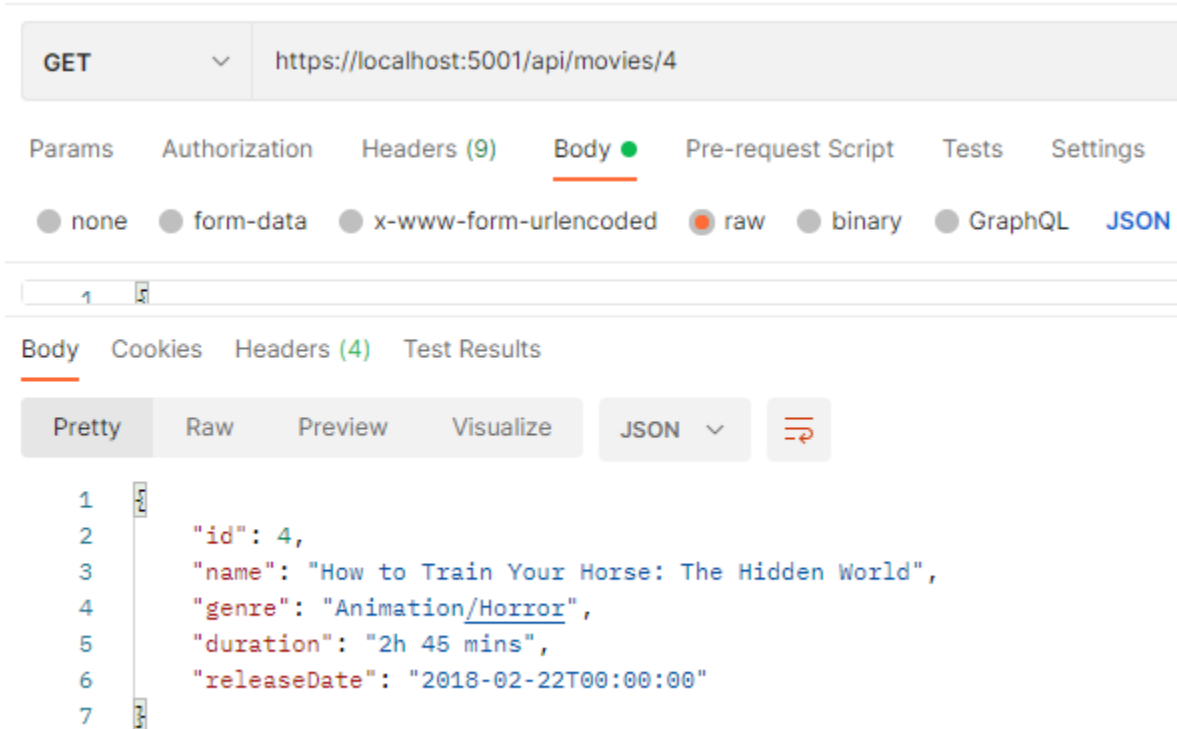
Tampilan data pada database setelah keempat data telah diPOST melalui postman



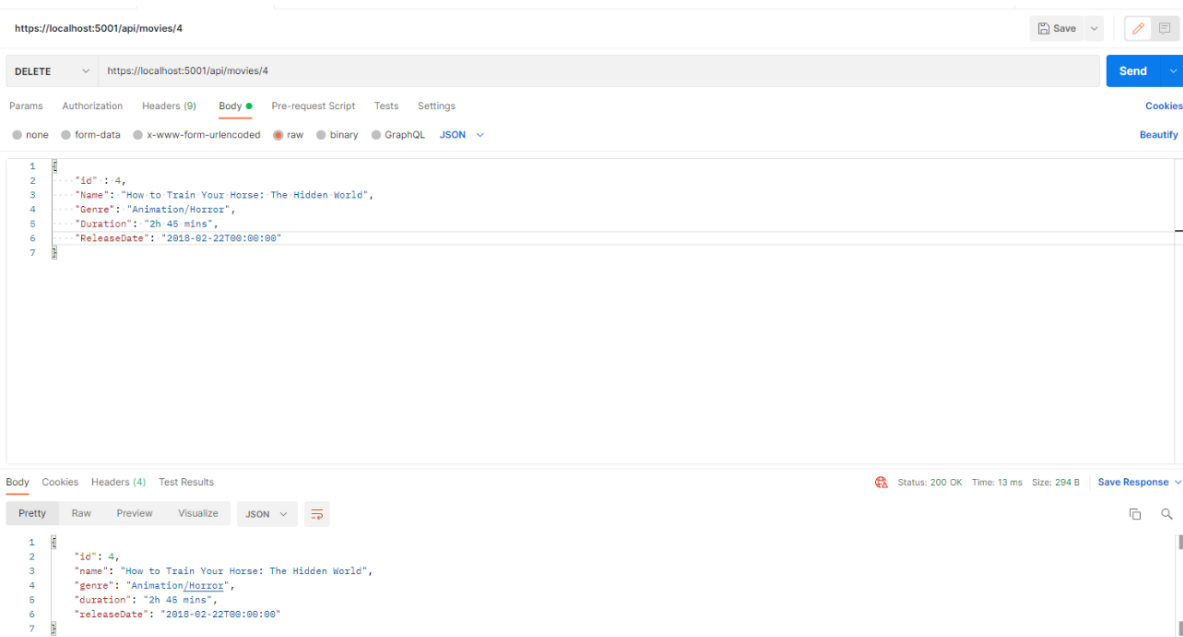
Melakukan method GET untuk menampilkan salah satu data(dengan id 4) dari database melalui url <https://localhost:5001/api/movies/{}> dimana {} akan diisi dengan id dari data yang akan diGET



Melakukan method PUT untuk mengubah data(dengan id 4) dari database melalui url <https://localhost:5001/api/movies/{}> dimana {} akan diisi dengan id dari data yang akan diubah dengan cara mengisi data yang ingin diubah melalui Body dari postman dan menekan send maka akan ditampilkan data yang telah berubah di body response



Tampilan data dengan id 4 yang ditampilkan menggunakan method GET, setelah diubah menggunakan method PUT pada Postman



Melakukan method DELETE untuk menghapus data(dengan id 4) dari database melalui url <https://localhost:5001/api/movies/{id}> dimana {id} akan diisi dengan id dari data yang akan dihapus

select \* from tb\_movie;

	Id	Name	Genre	Duration	ReleaseDate
1	1	Bad Boys for Life	Action/Comedy	2 5 mins	2020-10-23 00:00:00
2	2	John Wick	Action/Thriller	1h 41 mins	2014-10-24 00:00:00
3	3	The Martian	Sci-fi/Drama	2h 31 mins	2015-10-02 00:00:00

Tampilan data pada database setelah data dengan id 4 dihapus menggunakan method DELETE pada Postman

https://localhost:5001/api/movies

GET https://localhost:5001/api/movies

Params Authorization Headers (9) Body Pre-request Script Tests Settings

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

```

1  {
2    "id": 4,
3    "name": "How to Train Your Horse: The Hidden World",
4    "genre": "Animation/Horror",
5    "duration": "2h 45 mins",
6    "releaseDate": "2018-02-22T00:00:00"
7  }

```

Body Cookies Headers (4) Test Results


Pretty Raw Preview Visualize JSON

```

1  {
2    "id": 1,
3    "name": "Bad Boys for Life",
4    "genre": "Action/Comedy",
5    "duration": "2 5 mins",
6    "releaseDate": "2020-10-23T00:00:00"
7  },
8  {
9    "id": 2,
10   "name": "John Wick",
11   "genre": "Action/Thriller",
12   "duration": "1h 41 mins",
13   "releaseDate": "2014-10-24T00:00:00"
14  },
15  {
16    "id": 3,
17    "name": "The Martian",
18    "genre": "Sci-fi/Drama",
19    "duration": "2h 31 mins",
20    "releaseDate": "2015-10-02T00:00:00"
21  }
22  }
23

```

Tampilan data pada postman yang diGET dengan method GET, setelah data dengan id 4 dihapus menggunakan method DELETE pada Postman, dapat terlihat hanya tersisa 3 data dengan id 1, 2, dan 3

 Swagger  
powered by SMARTBEAR

Select a definition Movies v1

# Movies v1 OAS3

/swagger/v1/swagger.json

## Movies

GET

/api/Movies

POST

/api/Movies

GET

/api/Movies/{id}

PUT

/api/Movies/{id}

DELETE

/api/Movies/{id}

## Schemas

MovieData

id

name

genre

duration

releaseDate

integer(\$int32)

string

string

string

string

nullable: true

nullable: true

nullable: true

nullable: true

string(\$date-time)

Activate Window:  
Go to Settings to activate

## Tampilan Swagger