5 Documentation

This chapter is continued from chapter 4. If you have gone through chapter 4 it will be very easy to follow.

DIRECTORS CASES

Case 1

Method: POST

Endpoint: <http://127.0.0.1:5000/api/directors/>

Auth credentials: No authentication case in this chapter!

Example 1:

Request:

{

"first\_name": "chris",

"last\_name": "columbus"

}

Response:

{

"code": "success",

"director": {

"created": "2021-05-20 06:43:19",

"first\_name": "chris",

"id": 1.0,

"last\_name": "columbus",

"movies": []

}

}

Example 2:

Request:

{

"first\_name": "Christopher",

"last\_name": "Nolan"

}

Response:

{

"code": "success",

"director": {

"created": "2021-05-20 06:43:49",

"first\_name": "Christopher",

"id": 2.0,

"last\_name": "Nolan",

"movies": []

}

}

Case 2:

Method: GET

Endpoint: <http://127.0.0.1:5000/api/directors/>

Response:

{

"code": "success",

"directors": [

{

"first\_name": "chris",

"id": 1.0,

"last\_name": "columbus"

},

{

"first\_name": "Christopher",

"id": 2.0,

"last\_name": "Nolan"

}

]

}

Case 3:

Method: GET

Endpoint: <http://127.0.0.1:5000/api/directors/1>

{

"code": "success",

"director": {

"created": "2021-05-20 06:43:19",

"first\_name": "chris",

"id": 1.0,

"last\_name": "columbus",

"movies": []

}

}

Case 4:

Method: PUT

Endpoint: <http://127.0.0.1:5000/api/directors/1>

Request:

{

"first\_name": "chris1",

"last\_name": "columbus1"

}

Response:

{

"code": "success",

"director": {

"created": "2021-05-20 06:43:19",

"first\_name": "chris1",

"id": 1.0,

"last\_name": "columbus1",

"movies": []

}

Case 5:

Method: PATCH

Endpoint: <http://127.0.0.1:5000/api/directors/1>

Request:

{

"first\_name": "chris2"

}

Response:

{

"code": "success",

"director": {

"created": "2021-05-20 06:43:19",

"first\_name": "chris2",

"id": 1.0,

"last\_name": "columbus1",

"movies": []

}

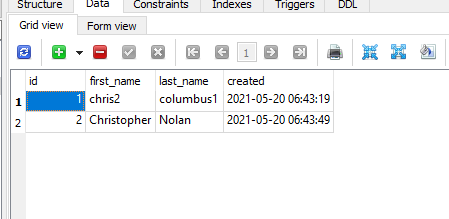
}

Case 6:

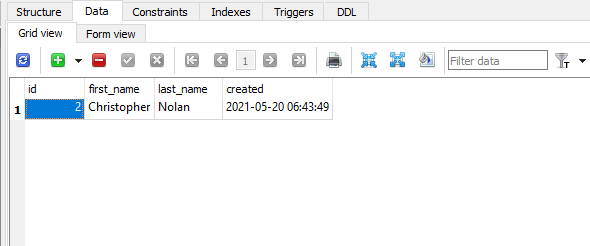
Method: DELETE

Endpoint: <http://127.0.0.1:5000/api/directors/1>

DB before deleting



DB after deleting:



Next cases:

In this chapter all get, post, put, delete and patch methods are used used. In the upcoming chapters authentication will be added!

MOVIES cases

Case 1:

Method: POST

Endpoint: <http://127.0.0.1:5000//api/movies/>

Request:

{

"title" : "inception",

"year" : 2010,

"director\_id": 2 }

Response:

{

"code": "success",

"movie": {

"director\_id": 2,

"id": 1.0,

"title": "inception",

"year": 2016

}

}

Case 2:

Method: GET

Endpoint: <http://127.0.0.1:5000//api/movies/>

{

"code": "success",

"movies": [

{

"director\_id": 2,

"title": "inception",

"year": 2016

}

]

}

Next cases:

Likewise get with id, put, post, delete and patch are written for movies endpoint

Directors get after adding movies to it:

Method: get

Endpoint:<http://127.0.0.1:5000/api/directors/2>

{

"code": "success",

"director": {

"created": "2021-05-20 06:43:49",

"first\_name": "Christopher",

"id": 2.0,

"last\_name": "Nolan",

"movies": [

{

"id": 1.0,

"title": "inception",

"year": 2016

}

]

}

}