

# Post character

The screenshot shows a REST client interface with a POST request to `http://localhost:3000/addperson`. The request body is a JSON object representing a character. The response status is 200 OK, and the response body is a JSON object representing the created character.

**Request:**

```
POST http://localhost:3000/addperson
{
  "name": "Paco",
  "age": 21,
  "born": "mty",
  "timeline": "asd",
  "alliegance": ["hola", "como"],
  "playedBy": "asd",
  "titles": ["uno", "dos"],
  "father": "str",
  "mother": "string",
  "spouse": "string"
}
```

**Response:**

```
{
  "alliegance": [
    "hola",
    "como"
  ],
  "titles": [
    "uno",
    "dos"
  ],
  "_id": "5cc77675631e8817f02998a9",
  "name": "Paco",
  "age": 21,
  "born": "mty",
  "timeline": "asd",
  "playedBy": "asd",
  "father": "str",
  "mother": "string",
  "spouse": "string",
  "__v": 0
}
```

The screenshot shows a query results interface with two JSON documents. The first document is the same as the one in the previous screenshot. The second document is a similar object with different values.

**Query Results 1-2 OF 2**

**Document 1:**

```
{
  "_id": ObjectId("5cc77675631e8817f02998a9"),
  "alliegance": Array,
  "titles": Array,
  "name": "Paco",
  "age": 21,
  "born": "mty",
  "timeline": "asd",
  "playedBy": "asd",
  "father": "str",
  "mother": "string",
  "spouse": "string",
  "__v": 0
}
```

**Document 2:**

```
{
  "_id": ObjectId("5cc772d287d1811ad9ec0051"),
  "alliegance": Array,
  "titles": Array,
  "name": "asd",
  "age": 12,
  "born": "asd",
  "timeline": "asd",
  "playedBy": "asd",
  "father": "str",
  "mother": "string",
  "spouse": "string",
  "__v": 0
}
```

# Get all

GET http://localhost:3000/allpersons

Status: 200 OK Time: 63 ms Size: 646 B

Body Cookies Headers (6) Test Results

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
5      "como"
6    ],
7    "titles": [
8      "uno",
9      "dos"
10   ],
11   "_id": "5cc772d287d1811ad8ec0051",
12   "name": "asd",
13   "age": 12,
14   "born": "asd",
15   "timeline": "asd",
16   "playedBy": "asd",
17   "father": "str",
18   "mother": "string",
19   "spouse": "string",
20   "__v": 0
21 },
22 {
23   "alliance": [
24     "hola",
25     "como"
26   ],
```

# Get by id

GET http://localhost:3000/getperson/5cc772d287d1811ad8ec0051

Status: 200 OK Time: 53 ms Size: 427 B

Body Cookies Headers (6) Test Results

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {
2   "alliance": [
3     "hola",
4     "como"
5   ],
6   "titles": [
7     "uno",
8     "dos"
9   ],
10  "_id": "5cc772d287d1811ad8ec0051",
11  "name": "asd",
12  "age": 12,
13  "born": "asd",
14  "timeline": "asd",
15  "playedBy": "asd",
16  "father": "str",
17  "mother": "string",
18  "spouse": "string",
19  "__v": 0
20 }
```

# Update

http://localhost:3000/modifyperson/ Examples

PATCH http://localhost:3000/modifyperson/5cc772d287d1811ad8ec0051 Send Save

Params Auth Headers (9) **Body** Pre-req. Tests Cookies Code (0)

none form-data x-www-form-urlencoded **raw** binary

JSON (application/json) Beautify

```
1 {
2   "name": "Paco",
3   "father": "papa",
4   "mother": "mama"
5 }
```

Status: 200 OK Time: 54 ms Size: 427 B Save Down

Body Cookies Headers (6) Test Results

Pretty Raw Preview JSON ≡

```
1 {
2   "alliegance": [
3     "hola",
4     "como"
5   ],
6   "titles": [
7     "uno",
8     "dos"
9   ],
10  "_id": "5cc772d287d1811ad8ec0051",
11  "name": "asd",
12  "age": 12,
13  "born": "asd",
14  "timeline": "asd",
15  "playedBy": "asd",
16  "father": "str",
17  "mother": "string",
18  "spouse": "string",
19  "__v": 0
20 }
```

## QUERY RESULTS 1-2 OF 2

```
_id: ObjectId("5cc772d287d1811ad8ec0051")
> alliegance: Array
> titles: Array
  name: "Paco"
  age: 12
  born: "asd"
  timeline: "asd"
  playedBy: "asd"
  father: "papa"
  mother: "mama"
  spouse: "string"
  __v: 0
```

# Delete

The screenshot shows a REST client interface with a DELETE request to `http://localhost:3000/deleteperson/5cc772d287d1811ad8ec0051`. The status is 200 OK, time is 57 ms, and size is 427 B. The response body is a JSON object:

```
1 {
2   "alliance": [
3     "hola",
4     "como"
5   ],
6   "titles": [
7     "uno",
8     "dos"
9   ],
10  "_id": "5cc772d287d1811ad8ec0051",
11  "name": "Paco",
12  "age": 12,
13  "born": "asd",
14  "timeline": "asd",
15  "playedBy": "asd",
16  "father": "papa",
17  "mother": "mama",
18  "spouse": "string",
19  "__v": 0
20 }
```

Ojo: en los otros screenshots tenía 2 documents después del delete ya queda 1

The screenshot shows a MongoDB query results interface with a filter `{ "filter": "example" }`. The results show 1-1 OF 1 document:

```
> _id: ObjectId("5cc77675631e8817f02998a9")
> alliance: Array
> titles: Array
  name: "Paco"
  age: 21
  born: "mty"
  timeline: "asd"
  playedBy: "asd"
  father: "str"
  mother: "string"
  spouse: "string"
  __v: 0
```