

Realizamos un POST

POST 192.168.20.75:3000/posts/ Send Save

Params Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 [{"name": "joaquin", "url": "google.es", "text": "soy joaquin", "comments": [{"text": "este es mi primer comentario de joaquin"}, {"text": "este es el segundo comentario de joaquin"}]}, {"name": "jesus", "url": "google.es", "text": "hola a todos", "comments": [{"text": "esto es un comentario de jesus"}, {"text": "esto es otro comentario de jesus"}]}]
```

Realizamos un GET a todos los posts

GET 192.168.20.75:3000/posts/ Pretty Raw Preview JSON

```
1 [
2   {
3     "name": "joaquin",
4     "url": "google.es",
5     "text": "soy joaquin",
6     "comments": [
7       {
8         "text": "este es mi primer comentario de joaquin"
9       },
10      {
11        "text": "este es el segundo comentario de joaquin"
12      }
13    ]
14  },
15  {
16    "name": "jesus",
17    "url": "google.es",
18    "text": "hola a todos",
19    "comments": [
20      {
21        "text": "esto es un comentario de jesus"
22      },
23      {
24        "text": "esto es otro comentario de jesus"
25      }
26    ]
27  }
28 ]
29 ]
30 ]
```

Realizamos un PUT a Jesús

192.168.20.75:3000/posts/1 PUT 192.168.20.75:3000/posts/1 Send Save

Params Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {"name": "jesus arribi", "url": "google.es", "text": "hola a todos", "comments": [{"text": "esto es un comentario de jesus"}, {"text": "esto es otro comentario de jesus"}]}
2
```

Realizamos un Get y vemos como se han modificado los archivos

Postman interface showing a GET request to `192.168.20.75:3000/posts/`. The response is a JSON array with two posts. The first post is for user 'joaquin' and the second for 'jesus arribi'.

```
1 [
2   {
3     "name": "joaquin",
4     "url": "google.es",
5     "text": "soy joaquin",
6     "comments": [
7       "este es mi primer comentario de joaquin",
8       "este es el segundo comentario de joaquin"
9     ]
10  },
11  {
12    "name": "jesus arribi",
13    "url": "google.es",
14    "text": "hola a todos",
15    "comments": [
16      "esto es un comentario de jesus"
17    ],
18    "text": "esto es otro comentario de jesus"
19  }
20 ]
```

Realizamos un Delete al usuario joaquin(id 0)

Postman interface showing a DELETE request to `192.168.20.75:3000/posts/0`. The status is 200 OK.

KEY	VALUE	DESCRIPTION
Key	Value	Description

Realizamos un Get y vemos como ya no aparece

Postman interface showing a GET request to `192.168.20.75:3000/posts/`. The response is a JSON array with only one post for user 'jesus arribi'.

```
1 [
2   {
3     "name": "jesus arribi",
4     "url": "google.es",
5     "text": "hola a todos",
6     "comments": [
7       "esto es un comentario de jesus"
8     ],
9     "text": "esto es otro comentario de jesus"
10  }
11 ]
```

Volvemos añadir los mismos usuarios y probamos los get,post,put,delete de los comments

Realizamos un get de los comments según el post id en este caso 1(jesus)

The screenshot shows the Postman interface with a GET request to `192.168.20.75:3000/posts/1/comments`. The response is a JSON array with two comment objects. Red arrows point to the URL, the GET method, and the response body.

```
1 [
2   {
3     "text": "esto es un comentario de jesus"
4   },
5   {
6     "text": "esto es otro comentario de jesus"
7   }
8 ]
```

Realizamos un post

The screenshot shows the Postman interface with a POST request to `192.168.20.75:3000/posts/1/comments`. The request body is a JSON object with a text field. Red arrows point to the URL, the POST method, and the request body.

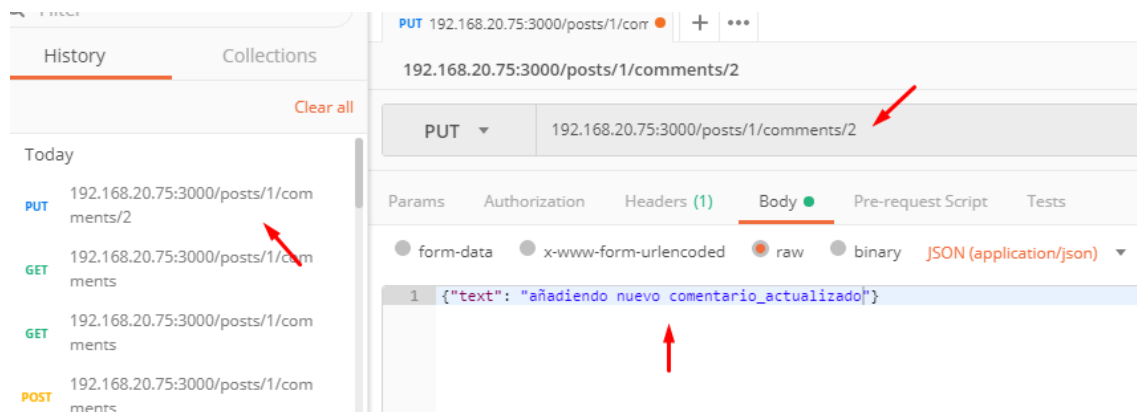
```
1 { "text": "añadiendo nuevo comentario" }
```

Realizamos un Get y se añade el nuevo comentario

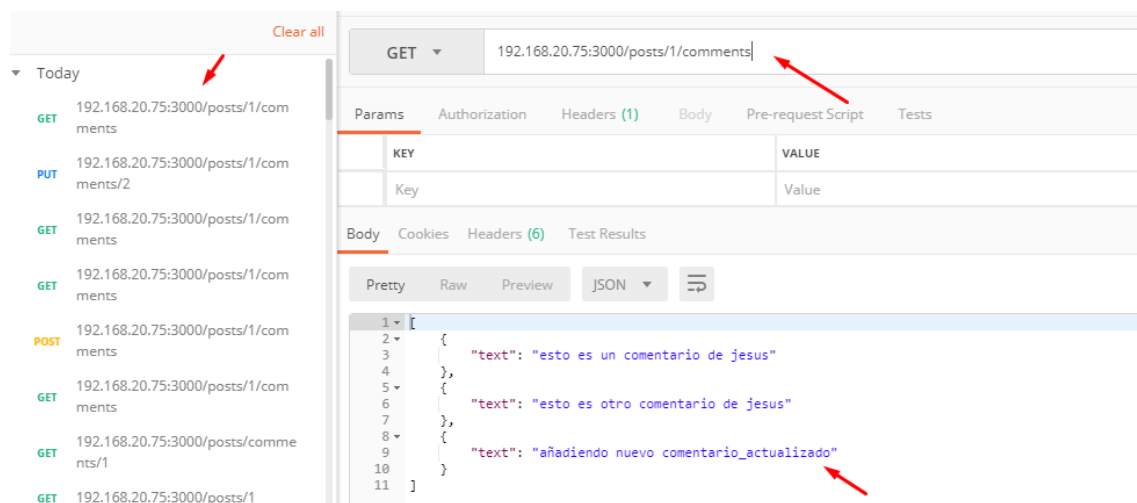
The screenshot shows the Postman interface with a GET request to `192.168.20.75:3000/posts/1/comments`. The response is a JSON array with three comment objects, including the newly added one. Red arrows point to the URL, the GET method, and the response body.

```
1 [
2   {
3     "text": "esto es un comentario de jesus"
4   },
5   {
6     "text": "esto es otro comentario de jesus"
7   },
8   {
9     "text": "añadiendo nuevo comentario"
10  }
11 ]
```

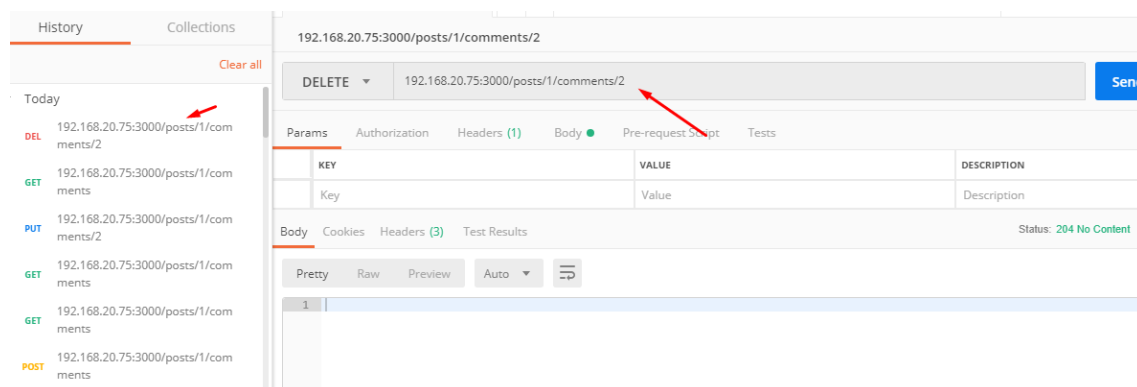
Realizamos un PUT para modificar el comentario que acabamos de añadir



Realizamos la petición GET y vemos como se ha modificado correctamente.



Por último, realizamos un DELETE del comentario.



Realizamos un GET y vemos que ya no esta el comentario

Clear all

Today

GET

192.168.20.75:3000/posts/1/comments/

DEL

192.168.20.75:3000/posts/1/comments/2

GET

192.168.20.75:3000/posts/1/comments

PUT

192.168.20.75:3000/posts/1/comments/2

GET

192.168.20.75:3000/posts/1/comments

GET

192.168.20.75:3000/posts/1/comments

POST

192.168.20.75:3000/posts/1/comments

GET

192.168.20.75:3000/posts/1/comments/

Params

Authorization

Headers (1)

Body

Pre-request Script

Tests

KEY

VALUE

Key

Value

Body

Cookies

Headers (6)

Test Results

Pretty

Raw

Preview

JSON

1

2

3

4

5

6

7

8

[

{

"text": "esto es un comentario de jesus"

,

{

"text": "esto es otro comentario de jesus"

]