

Creamos una cuenta nueva

```
root@debian-ivan:/# curl -H "Content-Type: application/json" -d '{"cantidad":100,"nombre":"ivan"}' "http://localhost:3000/accounts"
{"id":0}root@debian-ivan:/#
```

Se realiza el Post correctamente

```
entorno servidor despliegue aplicaciones web (despliegue de aplicaciones instantanea) [Corriendo] - O...
root@debian-ivan:/home/ivan/node/A13/rest-api# node server.js
POST /accounts 201 3.213 ms - 8
```

Consultamos la cuenta

```
root@debian-ivan:/# curl "http://localhost:3000/accounts"
[{"cantidad":100,"nombre":"ivan"}]root@debian-ivan:/#
```

Se realiza la petición Get

```
entorno servidor despliegue aplicaciones web (despliegue de aplicaciones instantanea) [Corriendo] - O...
root@debian-ivan:/home/ivan/node/A13/rest-api# node server.js
POST /accounts 201 3.213 ms - 8
GET /accounts 200 0.513 ms - 34
```

Actualizamos la cantidad

```
root@debian-ivan:/# curl -H 'Content-Type: application/json' -X PUT -d '{"cantidad":1000,"name":"ivan"}' "http://localhost:3000/accounts/0"
```

Se realiza el PUT

```
entorno servidor despliegue aplicaciones web (despliegue de aplicaciones instantanea) [Corriendo] - O...
root@debian-ivan:/home/ivan/node/A13/rest-api# node server.js
POST /accounts 201 3.213 ms - 8
GET /accounts 200 0.513 ms - 34
PUT /accounts/0 200 0.542 ms - 31
```

Comprobamos que se actualizo correctamente

```
root@debian-ivan:/# curl "http://localhost:3000/accounts"
[{"cantidad":1000,"name":"ivan"}]root@debian-ivan:/#
```

Se realiza un GET cuando comprobamos si se actualizo.

```
entorno servidor despliegue aplicaciones web (despliegue de aplicaciones instantanea) [Corriendo] - O...
root@debian-ivan:/home/ivan/node/A13/rest-api# node server.js
POST /accounts 201 3.213 ms - 8
GET /accounts 200 0.513 ms - 34
PUT /accounts/0 200 0.542 ms - 31
GET /accounts 200 0.173 ms - 33
```

Eliminamos la cuenta

```
root@debian-ivan:/# curl -X DELETE "http://localhost:3000/accounts/0"
root@debian-ivan:/# curl "http://localhost:3000/accounts"
[]root@debian-ivan:/# _
```

Se realiza la petición DELETE

```
root@debian-ivan:/home/ivan/node/A13/rest-api# node server.js
POST /accounts 201 3.213 ms - 8
GET /accounts 200 0.513 ms - 34
PUT /accounts/0 200 0.542 ms - 31
GET /accounts 200 0.173 ms - 33
DELETE /accounts/0 204 1.020 ms - -
GET /accounts 200 0.168 ms - 2
```

3. ¿Para qué se usa la opción -E en npm?

Para que las dependencias guardadas se configuren con una versión exacta.

4. ¿Recuerdas para qué se usa let en lugar de var a la hora de declarar una variable?

A la hora de declarar la variable con var esta se puede acceder a nivel global en cambio si se declara con let solo se puede acceder a nivel de bloque.

5. En prácticas anteriores vimos que una aplicación express se suele estructurar en una serie de pasos (imports, instanciación, configuración, uso de middlewares, etc.). Añade comentarios al código donde se vean estas partes (no tienen por qué estar todas).

6. En el código se devuelven distintos códigos de estado de HTTP. ¿Cuáles son? Explica su significado.

Código 201: La petición se ha completado y ha resultado en la creación de un recurso nuevo.

Código 200: Las peticiones han sido correctas.

Código 204: La petición se ha completado con éxito, pero la respuesta no tiene contenido, el navegador inicializa la página desde la que se realizó la petición.

7. Añade un método que permita consultar un registro concreto mediante GET.

```
22
23   app.get('/accounts/:id', (req, res) => {
24
25     res.status(200).send(store.accounts[req.params.id])
26
27   })
28
```

Obtenemos la cuenta con el id 1

entorno servidor despliegue aplicaciones web (despliegue de aplicaciones instantanea) [Corriendo] - O...

```
root@debian-ivan:/# curl "http://localhost:3000/accounts/1"
{"cantidad":100,"nombre":"jesus"}root@debian-ivan:/# _
```

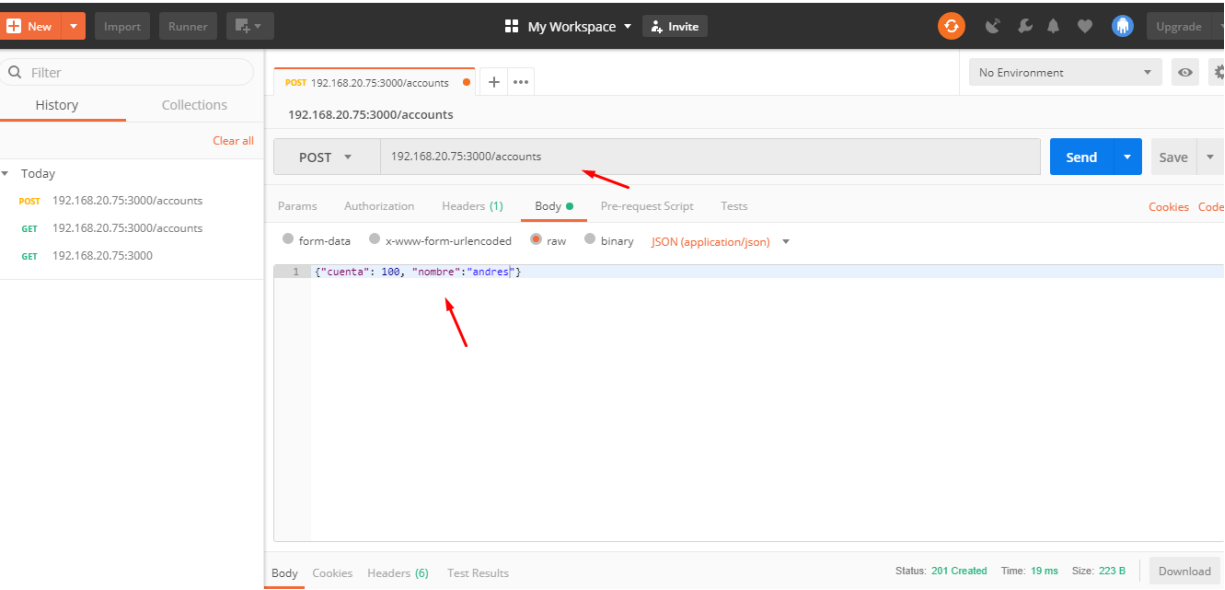
Ahora obtenemos la cuenta con el id 0

entorno servidor despliegue aplicaciones web (despliegue de aplicaciones instantanea) [Corriendo] - O...

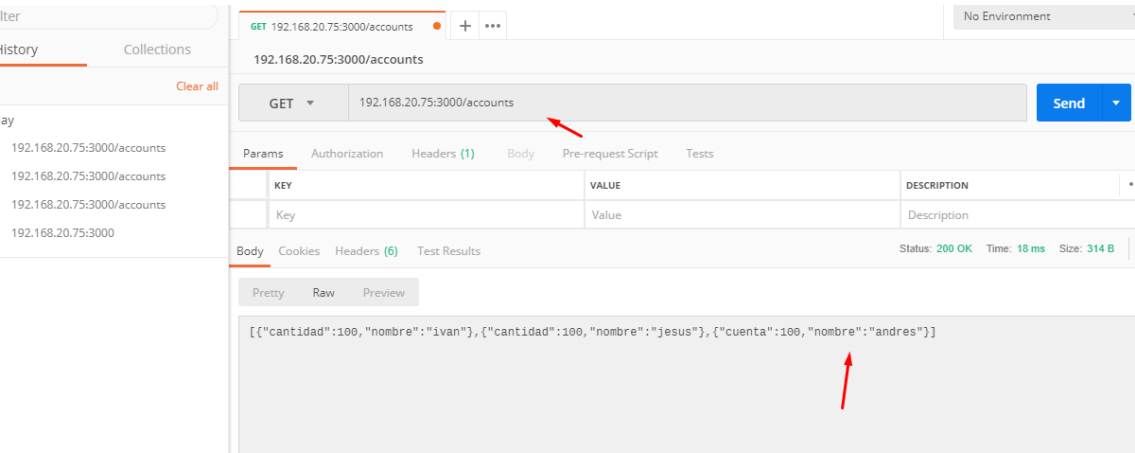
```
root@debian-ivan:/# curl "http://localhost:3000/accounts/0"
{"cantidad":100,"nombre":"ivan"}root@debian-ivan:/# _
```

8. Utiliza ahora postman para comprobar todos los endpoints que has creado. Puedes descargar postman del siguiente enlace: <https://www.getpostman.com/>. Cuando utilices Postman para la solicitud POST, asegúrate de seleccionar la configuración de body, raw y JSON (application/json) para evitar errores por no proporcionar el formato de solicitud correcto.

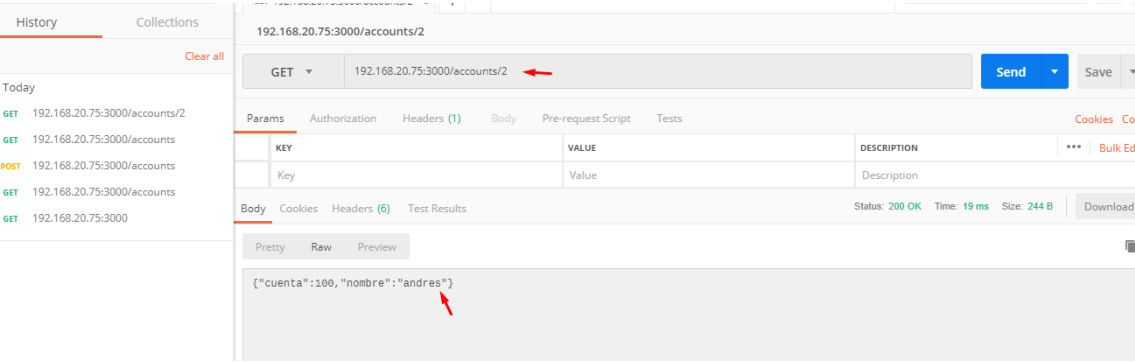
Insertando



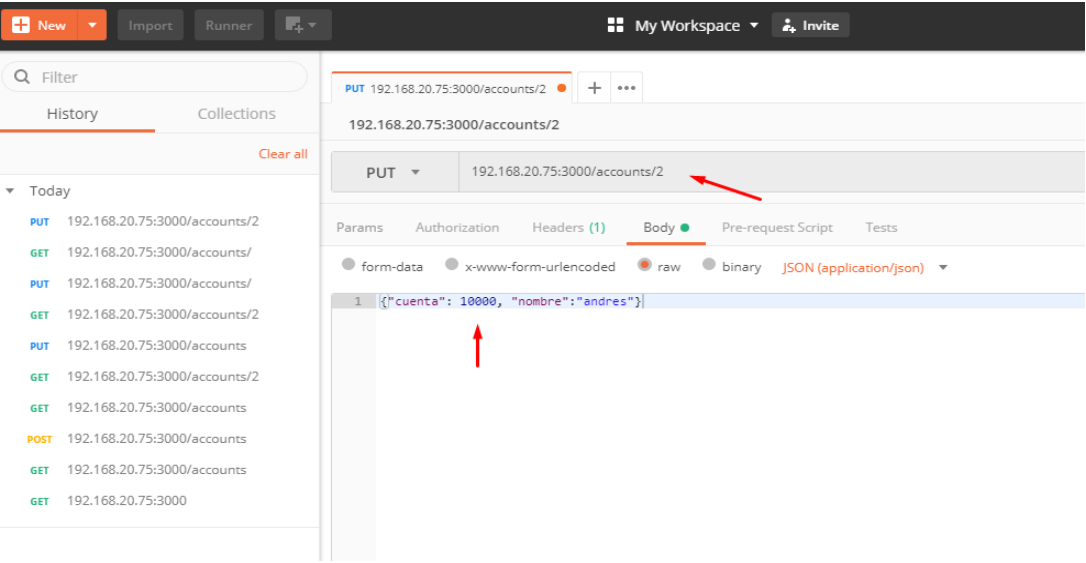
Hacemos un GET



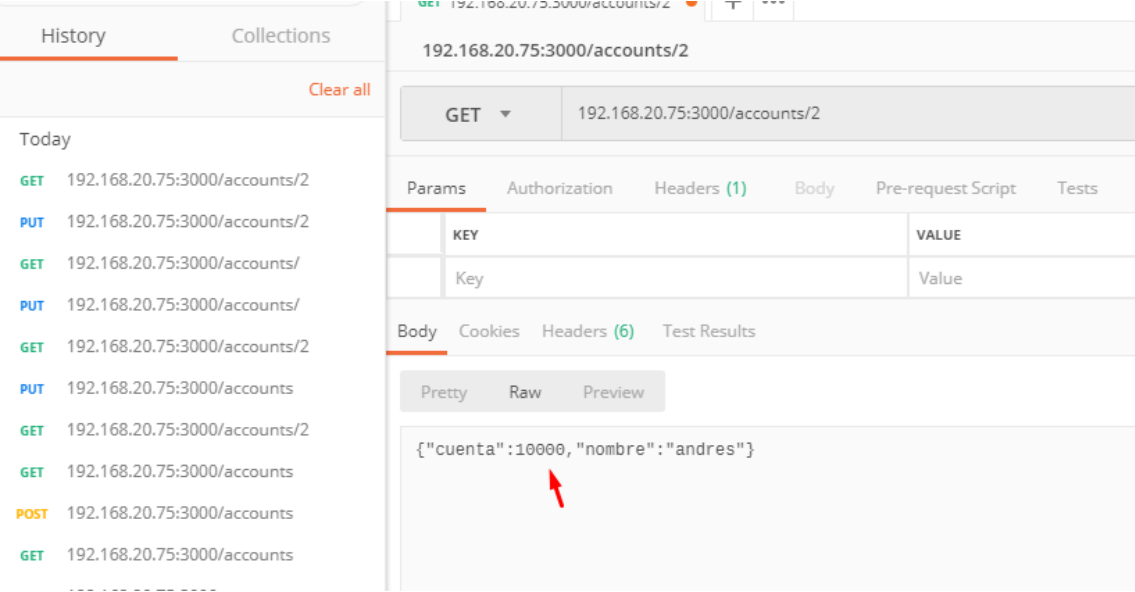
Hacemos un GET Introduciendo el Id



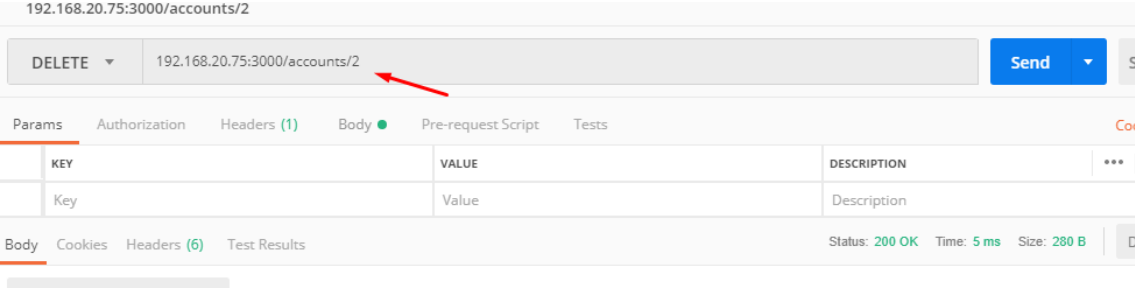
Hacemos un PUT



Hacemos un GET y los cambios se han realizado.



Hacemos un DELETE



Hacemos un GET y vemos que ya no existe

GET

192.168.20.75:3000/accounts

Params

Authorization

Headers (1)

Body

Pre-request Script

Tests

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body

Cookies

Headers (6)

Test Results

Status: 200 OK

Pretty

Raw

Preview

```
[{"cantidad":100,"nombre":"ivan"}, {"cantidad":100,"nombre":"jesus"}]
```