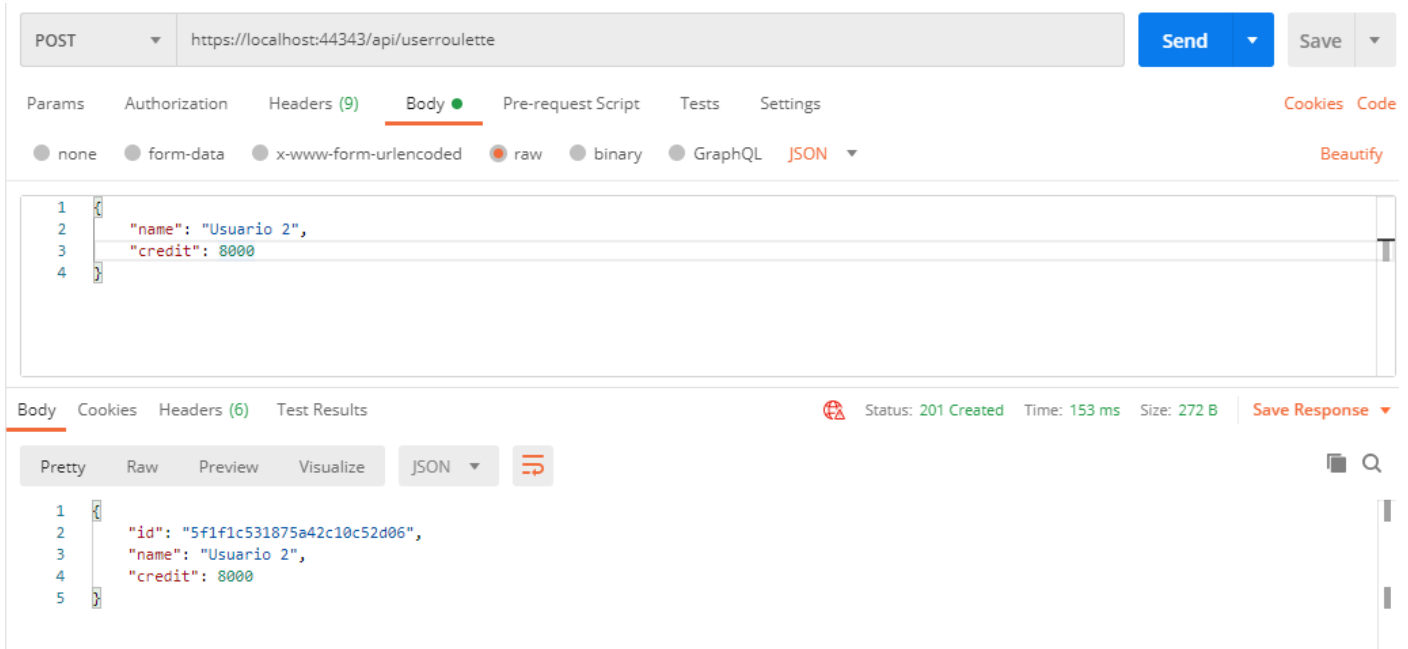


API Roulette

A continuación, se da una breve explicación de los métodos del API Roulette que se hizo para la prueba, se usa Postman para ver las respuestas que devuelve el API.

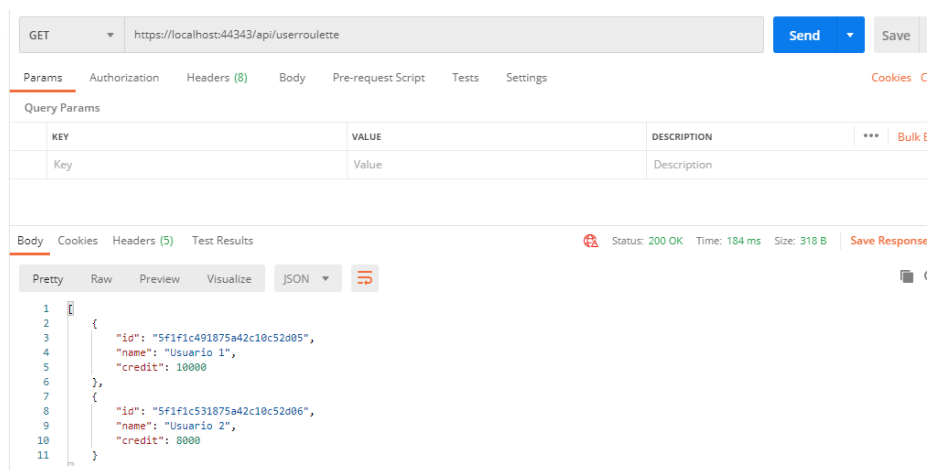
Creación de usuarios:

Es un método POST en la ruta <https://localhost:44343/api/userroulette>, en el cual se le debe pasar como parámetro un objeto con un nombre (name -> texto) y un crédito (credit -> numerico) , como se muestra a continuación:



Visualizar todos los usuarios

Para poder ver todos los usuarios se debe usar el método GET en la ruta <https://localhost:44343/api/userroulette>, donde como respuesta nos da la lista completa de los usuarios existentes.



Creación de Ruleta:

Para la creación de una ruleta se debe usar el método GET en la ruta <https://localhost:44343/api/roulette>, donde se deberá pasar un objeto con un nombre (name -> texto) y un estado (state -> texto) donde para un estado inicial debe ser "open"). Como se muestra a continuación:

The screenshot shows a REST client interface with a POST request to `https://localhost:44343/api/roulette`. The request body is a JSON object: `{ "name": "Ruleta3", "state": "open" }`. The response status is 201 Created, with a time of 258 ms and a size of 210 B. The response body is `true`.

```
POST https://localhost:44343/api/roulette

{
  "name": "Ruleta3",
  "state": "open"
}
```

Status: 201 Created Time: 258 ms Size: 210 B

```
1 true
```

Consultar las ruletas:

Para poder ver todas las ruletas se debe usar el método GET en la ruta <https://localhost:44343/api/roulette>, donde como respuesta nos da la lista completa de las ruletas existentes.

The screenshot shows a REST client interface with a GET request to `https://localhost:44343/api/roulette`. The response status is 200 OK, with a time of 268 ms and a size of 382 B. The response body is a JSON array of two ruleta objects.

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 268 ms Size: 382 B

```
1 [
2   {
3     "id": "5f1f20a51875a42c10c52d08",
4     "name": "Ruleta1",
5     "state": "open"
6   },
7   {
8     "id": "5f1f20b61875a42c10c52d09",
9     "name": "Ruleta2",
10    "state": "close"
11  },
12 ]
```

Crear apuesta:

Para la creación de una apuesta se debe usar un método POST pasando un objeto con los siguientes atributos:

idUser: id del usuario que apuesta.

idRoulette: id de la ruleta en la que se realiza la apuesta.

moneyValue: valor de la apuesta realizada.

betNumber: número en la ruleta.

betColor: color por el que se apuesta.

The screenshot shows a REST client interface with the following details:

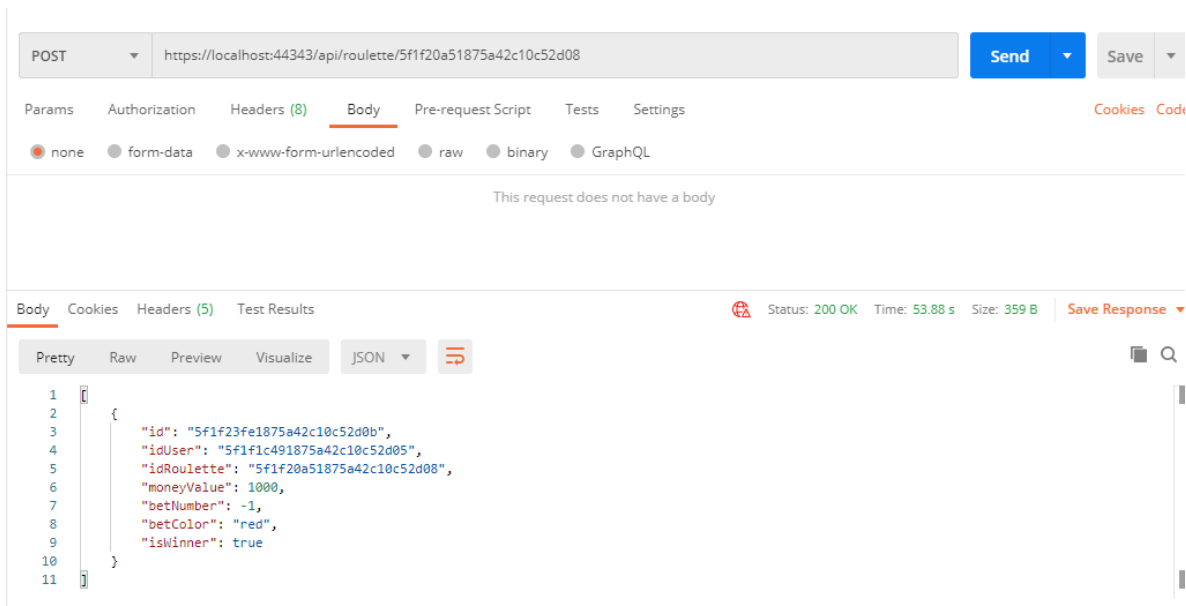
- Method:** POST
- URL:** https://localhost:44343/api/bet
- Buttons:** Send, Save
- Tabs:** Params, Authorization, Headers (9), Body (selected), Pre-request Script, Tests, Settings, Cookies, Code
- Body Type:** JSON
- Body Content:**

```
1 {  
2   "idUser": "5f1f1c491875a42c10c52d05",  
3   "idRoulette": "5f1f20a51875a42c10c52d08",  
4   "moneyValue": 1000,  
5   "betNumber": -1,  
6   "betColor": "red"  
7 }
```
- Footer:** Body, Cookies, Headers (5), Test Results, Status: 400 Bad Request, Time: 413 ms, Size: 226 B, Save Response

Para este caso se realizan varias validaciones como que los idUser y idRoulette existan, que el estado de la ruleta sea “open”, que la cantidad apostada no supere el crédito del usuario, que la apuesta sea válida (un número entre 0 y 36 o un color).

Cerrar apuestas

Para cerrar apuestas se debe usar el método POST en la ruta <https://localhost:44343/api/roulette/> concatenado con el id de la ruleta que se desea cerrar. Como se muestra a continuación:



Para este caso se buscan todas apuestas realizadas en esa Ruleta, y se le agrega al usuario la cantidad apostada multiplicado por dos, también se indica si la apuesta fue ganadora o no en la variable “isWinner”, y finalmente la ruleta cambia de estado a “close”.