

API REST

JOSE MANUEL RAMOS SIERRA

JUAN SEBASTIAN HURTADO ROMERO

CENTRO DE ELECTRICIDAD, ELECTRONICA Y TELECOMUNICACIONES

(CEET)

SERVICIO NACIONAL DE APRENDIZAJE (SENA)

EDWIN MARIN

08/11/2024

Contenido

METODO GET.....	3
METODO GET CON ID.....	4
METODO POST.....	5
METODO PUT.....	6
METODO DELETE	7
REPOSITORIO GITHUB.....	8

METODO GET

The screenshot displays a REST client interface with a GET request to `http://localhost:8000/api/students` and its corresponding JSON response. The interface includes tabs for Query, Headers, Auth, Body, Tests, and Pre Run. The Query Parameters section is empty. The Response tab shows the status 200 OK, size 506 Bytes, and time 338 ms. The response body is a JSON array of three student objects.

Query Parameters

parameter	value
-----------	-------

Status: 200 OK **Size:** 506 Bytes **Time:** 338 ms

Response

```
{
  "id": 1,
  "name": "juan",
  "email": "juani1234@gmail.com",
  "job": "developer software",
  "created_at": "2024-11-08T21:41:00.000000Z",
  "updated_at": "2024-11-09T01:52:30.000000Z"
},
{
  "id": 3,
  "name": "jose",
  "email": "vegetalesjose@gmail.com",
  "job": "developer software",
  "created_at": "2024-11-09T02:27:37.000000Z",
  "updated_at": "2024-11-09T02:27:37.000000Z"
},
{
  "id": 4,
  "name": "juan jose",
  "email": "juanrex@gmail.com",
  "job": "developer software",
  "created_at": "2024-11-09T02:28:19.000000Z",
  "updated_at": "2024-11-09T02:28:19.000000Z"
}
```

METODO GET CON ID

The screenshot displays a REST client interface with a dark theme. At the top, a dropdown menu is set to 'GET' and the URL 'http://localhost:8000/api/students/4' is entered. A blue 'Send' button is to the right. Below the URL bar, tabs for 'Query', 'Headers', 'Auth', 'Body', 'Tests', and 'Pre Run' are visible, with 'Query' being the active tab. The 'Query Parameters' section shows a table with one row: 'parameter' and 'value'. To the right, the 'Response' tab is active, showing the status '200 OK', size '193 Bytes', and time '485 ms'. The response body is a JSON object:

```
{  "student": {    "id": 4,    "name": "juan jose",    "email": "juanrex@gmail.com",    "job": "developer software",    "created_at": "2024-11-09T02:28:19.000000Z",    "updated_at": "2024-11-09T02:28:19.000000Z"  },  "status": 200}
```

 At the bottom right, there are buttons for 'Response' and 'Chart'.

GET Send

Query Headers² Auth Body¹ Tests Pre Run

Query Parameters

<input type="checkbox"/>	parameter	value
--------------------------	-----------	-------

Status: 200 OK Size: 193 Bytes Time: 485 ms

Response Headers⁷ Cookies Results Docs {} ≡

```
1 {
2   "student": {
3     "id": 4,
4     "name": "juan jose",
5     "email": "juanrex@gmail.com",
6     "job": "developer software",
7     "created_at": "2024-11-09T02:28:19.000000Z",
8     "updated_at": "2024-11-09T02:28:19.000000Z"
9   },
10  "status": 200
11 }
```

Response Chart ↗

METODO POST

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

- Method:** POST
- URL:** http://localhost:8000/api/students
- Send Button:** A blue button labeled "Send".
- Response Status:** 201 Created
- Response Size:** 193 Bytes
- Response Time:** 286 ms
- Request Body (JSON):**

```
1 {
2
3   "name": "juan jose",
4   "email": "juanrex@gmail.com",
5   "job": "developer software"
6 }
7
8
9
```
- Response Body (JSON):**

```
1 {
2   "student": {
3     "name": "juan jose",
4     "email": "juanrex@gmail.com",
5     "job": "developer software",
6     "updated_at": "2024-11-09T02:28:19.000000Z",
7     "created_at": "2024-11-09T02:28:19.000000Z",
8     "id": 4
9   },
10  "status": 201
11 }
```

METODO PUT

The screenshot displays a REST client interface with a PUT request to `http://localhost:8000/api/students/4`. The request body is a JSON object with student details. The response is a 200 OK status with a JSON body containing a success message and the updated student data.

Request:

```
PUT http://localhost:8000/api/students/4
```

Request Body (JSON):

```
1 {
2   "name": "andres camilo",
3   "email": "andres1453@gmail.com",
4   "job": "ceo"
5 }
```

Response:

```
1 {
2   "message": "Estudiante actualizado correctamente",
3   "status": 200,
4   "student": {
5     "id": 4,
6     "name": "andres camilo",
7     "email": "andres1453@gmail.com",
8     "job": "ceo",
9     "created_at": "2024-11-09T02:28:19.000000Z",
10    "updated_at": "2024-11-09T03:07:06.000000Z"
11  }
12 }
```

METODO DELETE

The screenshot displays a REST client interface with a dark theme. The top bar shows the method 'DELETE', a dropdown arrow, the URL 'http://localhost8000/api/students/4', and a 'Send' button. Below this, a tabbed interface includes 'Query', 'Headers 2', 'Auth', 'Body 1' (selected), 'Tests', and 'Pre-Run'. The 'Body 1' tab shows a JSON editor with the following content:

```
1 {  
2  
3   "name" : "andres camilo",  
4   "email": "andres1453@gmail.com",  
5   "job": "ceo"  
6  
7 }  
8  
9
```

Below the editor is a 'Format' button. The right-hand pane shows the 'Response' tab, displaying the status and response body:

Status: 200 OK Size: 61 Bytes Time: 200 ms

```
1 {  
2   "message": "Estudiante eliminado correctamente",  
3   "status": 200  
4 }
```

At the bottom right of the response pane, there are buttons for 'Response' and 'Chart', along with a small icon.

REPOSITORIO GITHUB

<https://github.com/jose3284/api-rest.git>