

Visual Studio Code interface showing a REST client request and response.

Request:

- Method: GET
- URL: http://localhost:3002
- Body (JSON):

```
1 {
2   "_id": {
3     "$oid": "6647926c0946d1cf5a20f022"
4   },
5   "name": "Alex",
6   "email": "barakiller73@gmail.com",
7   "password": "Barakiller73"
8 }
```

Response:

- Status: 200 OK
- Size: 23 Bytes
- Time: 64 ms
- Response body: 1 Estoy en el método get

Terminal:

```
El servidor está en el puerto 3002
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
Mongo connected
El servidor está en el puerto 3002
```

Visual Studio Code interface showing a REST client request and response.

Request:

- Method: POST
- URL: http://localhost:3002
- Body (JSON):

```
1 {
2   "name": "Pedro",
3   "email": "pedroarturo@gmail.com",
4   "password": "tenjo123*",
5   "_id": "6647a7edfbdd4289cb677315",
6   "createdAt": "2024-05-17T18:54:37.324Z",
7   "updatedAt": "2024-05-17T18:54:37.324Z",
8   "__v": 0
9 }
```

Response:

- Status: 200 OK
- Size: 190 Bytes
- Time: 43 ms
- Response body: 1 {
2 "name": "Pedro",
3 "email": "pedroarturo@gmail.com",
4 "password": "tenjo123*",
5 "_id": "6647a7edfbdd4289cb677315",
6 "createdAt": "2024-05-17T18:54:37.324Z",
7 "updatedAt": "2024-05-17T18:54:37.324Z",
8 "__v": 0
9 }

Terminal:

```
El servidor está en el puerto 3002
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
Mongo connected
El servidor está en el puerto 3002
```

Insertar registro con el método POST

The screenshot shows the VS Code interface with the REST client extension. The Explorer sidebar on the left shows the project structure with files like package.json, app.js, userModel.js, and config.js. The main editor area displays a REST client request configuration for a POST method to http://localhost:3002. The request body is a JSON object with fields: name (Carlos), email (bogota@gmail.com), password (Barakiller73*), createdAt (2024-05-17T18:38:56.065Z), updatedAt (2024-05-17T18:38:56.065Z), and _v (0). The response status is 200 OK, with a size of 190 Bytes and a time of 27 ms. The response body is a JSON object with the same fields as the request. The terminal at the bottom shows the output of the application, including messages from nodemon and MongoDB.

File Edit Selection View Go Run ... apikronos

EXPLORER ...

OPEN EDITORS

- package.json
- app.js
- New Request
- userModel.js
- config.js

APIKRONOS

- app.js
- config.js
- package.json
- userModel.js

POST http://localhost:3002 Send

Query Headers² Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "name": "Carlos",
3   "email": "bogota@gmail.com",
4   "password": "Barakiller73*",
5 }
6
```

Status: 200 OK Size: 190 Bytes Time: 27 ms

Response Headers⁶ Cookies Results Docs {}

```
1 {
2   "name": "Carlos",
3   "email": "bogota@gmail.com",
4   "password": "Barakiller73*",
5   "_id": "6647a44038ecd91be0e431ed",
6   "createdAt": "2024-05-17T18:38:56.065Z",
7   "updatedAt": "2024-05-17T18:38:56.065Z",
8   "_v": 0
9 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[nodemon] app crashed - waiting for file changes before starting...

[nodemon] restarting due to changes...

[nodemon] starting `node app.js`

Mongo connected

El servidor está en el puerto 3002

node + ... ^ x

Go Live

The screenshot shows the VS Code interface with the REST client extension. The Explorer sidebar on the left shows the project structure with files like package.json, app.js, userModel.js, and config.js. The main editor area displays a REST client request configuration for a GET method to http://localhost:3002/6647a44038ecd91be0e431ed. The request body is a JSON object with fields: name (Pedro), email (pedroarturo@gmail.com), password (tenjo123*), createdAt (2024-05-17T18:38:56.065Z), updatedAt (2024-05-17T18:38:56.065Z), and _v (0). The response status is 200 OK, with a size of 190 Bytes and a time of 11 ms. The response body is a JSON object with the same fields as the request. The terminal at the bottom shows the output of the application, including messages from nodemon and MongoDB.

File Edit Selection View Go Run ... apikronos

EXPLORER ...

OPEN EDITORS

- package.json
- app.js
- New Request
- userModel.js
- config.js

APIKRONOS

- app.js
- config.js
- package.json
- userModel.js

GET http://localhost:3002/6647a44038ecd91be0e431ed Send

Query Headers² Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "name": "Pedro",
3   "email": "pedroarturo@gmail.com",
4   "password": "tenjo123*",
5 }
6
```

Status: 200 OK Size: 190 Bytes Time: 11 ms

Response Headers⁶ Cookies Results Docs {}

```
1 {
2   "_id": "6647a44038ecd91be0e431ed",
3   "name": "Carlos",
4   "email": "bogota@gmail.com",
5   "password": "Barakiller73*",
6   "createdAt": "2024-05-17T18:38:56.065Z",
7   "updatedAt": "2024-05-17T18:38:56.065Z",
8   "_v": 0
9 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

El servidor está en el puerto 3002

node + ... ^ x

Go Live

Obtener registro con el método GET

The screenshot shows the VS Code interface with the REST client extension. The Explorer sidebar on the left shows the project structure with files like package.json, app.js, config.js, and userModel.js. The main editor area is split into three panes. The top pane shows a REST client request with the method PUT and the URL http://localhost:3002/6647aa2eff19fb70e2546300. The middle pane shows the request body in JSON format:

```
1 {
2   "name": "PedroNel",
3   "email": "pedroarturo2345@gmail.com",
4   "password": "tenjo12345566*"
5 }
6
```

. The bottom pane shows the response status 200 OK, size 202 Bytes, and time 6 ms. The response body is in JSON format:

```
1 {
2   "_id": "6647aa2eff19fb70e2546300",
3   "name": "PedroNel",
4   "email": "pedroarturo2345@gmail.com",
5   "password": "tenjo12345566*",
6   "createdAt": "2024-05-17T19:04:14.511Z",
7   "updatedAt": "2024-05-17T20:32:21.888Z",
8   "__v": 0
9 }
```

. The bottom status bar shows the terminal output: El servidor está en el puerto 3002.

This screenshot is identical to the one above, showing the same REST client request and response in VS Code. The request is a PUT to http://localhost:3002/6647aa2eff19fb70e2546300 with a JSON body. The response is a 200 OK status with a JSON body containing the created user data.

Actualizar registro con un parámetro id PUT http://localhost:3002/6647aa2eff19fb70e2546300

File Edit Selection View Go Run ... apikronos

EXPLORER ... {} package.json JS app.js TC New Request * X JS userModel.js JS config.js

OPEN EDITORS {} package.json JS app.js TC New Request * X JS userModel.js JS config.js

APIKRONOS JS app.js JS config.js {} package.json JS userModel.js

DELETE http://localhost:3002/6647aa2eff19fb70e2546300 Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "name": "PedroNel",
3   "email": "pedroarturo2345@gmail.com",
4   "password": "tenjo12345566*"
5 }
6
```

Status: 200 OK Size: 202 Bytes Time: 6 ms

Response Headers 6 Cookies Results Docs {}

```
1 {
2   "_id": "6647aa2eff19fb70e2546300",
3   "name": "PedroNel",
4   "email": "pedroarturo2345@gmail.com",
5   "password": "tenjo12345566*",
6   "createdAt": "2024-05-17T19:04:14.511Z",
7   "updatedAt": "2024-05-17T20:32:21.888Z",
8   "_v": 0
9 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

El servidor está en el puerto 3002

node + v ... ^ x

Go Live

File Edit Selection View Go Run ... apikronos

EXPLORER ... {} package.json JS app.js TC New Request * X JS userModel.js JS config.js

OPEN EDITORS {} package.json JS app.js TC New Request * X JS userModel.js JS config.js

APIKRONOS JS app.js JS config.js {} package.json JS userModel.js

DELETE http://localhost:3002/6647aa2eff19fb70e2546300 Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "name": "PedroNel",
3   "email": "pedroarturo2345@gmail.com",
4   "password": "tenjo12345566*"
5 }
6
```

Status: 200 OK Size: 38 Bytes Time: 36 ms

Response Headers 6 Cookies Results Docs {}

```
1 {
2   "acknowledged": true,
3   "deletedCount": 1
4 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

El servidor está en el puerto 3002

node + v ... ^ x

Go Live

Borrar un registro con parámetro DELETE <http://localhost:3002/6647aa2eff19fb70e2546300>

The screenshot displays the Apikronos web interface, which functions as a REST client. The interface is divided into several sections:

- Top Bar:** Includes standard web browser controls (back, forward, address bar) and the Apikronos logo.
- Left Sidebar:** Contains navigation icons and a file explorer showing the project structure with files like `package.json`, `app.js`, `userModel.js`, and `config.js`.
- Request Section:**
 - Method:** GET
 - URL:** `http://localhost:3002`
 - Body:** A JSON object:

```
{  "name": "PedroWe1",  "email": "pedroarturo2345@gmail.com",  "password": "tenjo12345566*"}
```
- Response Section:**
 - Status:** 200 OK
 - Size:** 571 Bytes
 - Time:** 17 ms
 - Body:** A JSON array of three user objects:

```
[  {    "_id": "6647a44038ecd91be0e431ed",    "name": "Carlos",    "email": "bogota@gmail.com",    "password": "Barakiller73*",    "createdAt": "2024-05-17T18:38:56.065Z",    "updatedAt": "2024-05-17T18:38:56.065Z",    "__v": 0  },  {    "_id": "6647a4d638ecd91be0e431f1",    "name": "Andres",    "email": "toretota@gmail.com",    "password": "cerrito*",    "createdAt": "2024-05-17T18:41:26.419Z",    "updatedAt": "2024-05-17T18:41:26.419Z",    "__v": 0  },  {    "_id": "6647a7edfbbdd4289cb677315",    "name": "Pedro",    "email": "pedroarturo@gmail.com",    "password": "tenjo123*",    "createdAt": "2024-05-17T18:54:37.324Z",    "updatedAt": "2024-05-17T18:54:37.324Z",    "__v": 0  }]
```

EJERCICO CON JAVASCRIPT Y CON BASE DE DATOS NO RELACIONAL (MONGODB)