

SIMPLE MICORSERVICIO API CON NODEJS Y EXPRESS

METODO POST (REGISTRAR IDOMA DEL USUARIO)

Se recibe en los parámetros de la URI los campos "name" y "Lang". Se guardan en un diccionario donde la clave es el nombre y el valor el idioma.

```
router.post("/demo/register", (req, res) => {  
  users[req.query.name] = req.query.Lang;  
  res.json({ status: "ok" });  
})
```

```
27  
28 router.post("/demo/register", (req, res) => {  
29   console.log("[POST] : ", req.query.name, req.query.Lang)  
30   users[req.query.name] = req.query.Lang;  
31   console.log(users);  
32   res.json({ status: "ok" });  
33 })  
34  
35 app.use(router);  
36  
37 app.listen(3000, () => {  
38   console.log("INICIADO.....")  
39 })
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

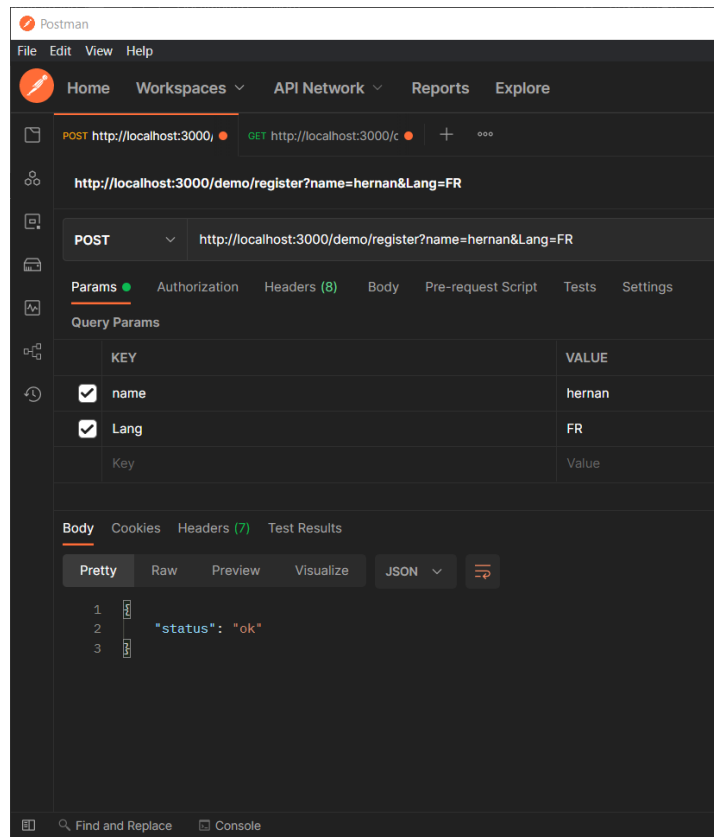
node + - □ □ ^ ×

Windows PowerShell

Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/pscore6>

```
PS C:\Users\Mannulus\Documents\UIS\2022-1\TrabajosSoftwareIII> cd .\simpleAPI\  
PS C:\Users\Mannulus\Documents\UIS\2022-1\TrabajosSoftwareIII\simpleAPI> node .\index.js  
INICIADO.....  
[POST] : felipe ES  
{ felipe: 'ES' }  
[POST] : juan EN  
{ felipe: 'ES', juan: 'EN' }  
[POST] : hernan FR  
{ felipe: 'ES', juan: 'EN', hernan: 'FR' }  
□
```



METODO GET (SALUDO AL USUARIO)

Recibe un parámetro en la URI “name” el cual usa para buscar el índice en el diccionario y el valor que sería el Lang, Mediante un switch decide el saludo dependiendo de su idioma y luego envía el mensaje.

```
routes.get("/demo/greeting", (req, res) => {
  let id = Object.keys(users).findIndex(val => val == req.query.name);
  var saludo = "";
  switch(users[req.query.name]){
    case "ES":
      saludo = "Hola";
      break;
    case "FR":
      saludo = "Salut"
      break;
    default:
      saludo = "Hello";
  }
  res.json({ id: id, content: `${saludo}, ${req.query.name}!` })
})
```

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

POST http://localhost:3000/ GET http://localhost:3000/c + ...

http://localhost:3000/demo/greeting?name=hernan

GET http://localhost:3000/demo/greeting?name=hernan

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
<input checked="" type="checkbox"/> name	hernan
Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "content": "Salut, hernan!"
4 }
```

Find and Replace Console

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

POST http://localhost:3000/ GET http://localhost:3000/c + ...

http://localhost:3000/demo/greeting?name=juan

GET http://localhost:3000/demo/greeting?name=juan

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
<input checked="" type="checkbox"/> name	juan
Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "content": "Hello, juan!"
4 }
```

Find and Replace Console



POST http://localhost:3000/ ●

GET http://localhost:3000/c ●



http://localhost:3000/demo/greeting?name=felipe



GET ▾

http://localhost:3000/demo/greeting?name=felipe



Params ●

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings



Query Params



	KEY	VALUE
<input checked="" type="checkbox"/>	name	felipe
	Key	Value



Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON ▾



```
1 {
2   "id": 0,
3   "content": "Hola, felipe!"
4 }
```



Find and Replace



Console