

Iniciamos sesión con un usuario para ver que se genera el Token Postman

The screenshot shows the Postman interface for a REST client. At the top, it displays the URL bar as "localhost:8080/login". The method dropdown is set to "POST". Below the URL bar, there are tabs for Params, Authorization, Headers (9), Body, Scripts, Tests, and Settings. The "Body" tab is selected, and the format is set to "raw". The body content is a JSON object: {"username": "kevin", "password": "12345"}. On the right side, there are buttons for "Save", "Share", and "Send". Below the main editor, there's a status bar showing "200 OK", "654 ms", and "894 B". At the bottom, the "Body" tab is expanded, showing the response JSON: {"message": "Hola kevin has iniciado seccion con exito", "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJlbiIsImVudCI6IjEwODcxOTAsImF1dC..."}.

Get up-to-speed with JSON Web Tokens. [Get the JWT Handbook for free!](#)

English

JWT Debugger

JSON WEB TOKEN (JWT)

Valid JWT

Invalid Signature

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50L3R1b3QiLCJ1eWUiOiJ1b3R1b3QiLCJleHAiOjE3NjIxODcxOTAsImhhdCI6MTc2MjE4MzU5MH0.VcoUnkgqAage3dQR1LIoDlvrsdqfhfdwbPypSBkr9GI
```

DEBUGGER

CLAIMS TABLE

```
{  "alg": "HS256"}
```

DEBUGGER

CLAIMS TABLE

```
{  "sub": "kevin",  "authorities": "[{\\"authority\\":\\"ROLE_USER\\"}]",  "username": "kevin",  "exp": 1762187190,  "iat": 1762183590}
```

The screenshot shows the Postman application window. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'API Network'. Below that, a breadcrumb trail shows 'localhost:8080/login' and 'POST localhost:8080/api/tasks'. The main area is divided into tabs: 'localhost:8080/api/tasks', 'Save', 'Share', and 'Send'. The 'Send' tab is active, showing the request details. The URL is 'localhost:8080/api/tasks' and the method is 'POST'. The body is set to 'raw' and contains a JSON object: {"name": "Tarea con Seguridad", "description": "Otra Tarea"}. The status bar at the bottom indicates a '403 Forbidden' response with a status of 52 ms and a size of 300 B.

[illegible]

This screenshot shows the Spring Boot Admin web interface. At the top, there is a header with the text 'API Documentation for Task with Spring Boot 3.5.7'. Below this, the 'Servers' tab is selected, displaying a single server entry: 'http://localhost:8080 - Generated server url'. To the right of this entry is a green 'Authorize' button with a lock icon.

## Se crea una Tarea

**POST** /api/tasks

Parameters

Cancel

Reset

No parameters

Request body required

application/json

```
{
  "name": "Otra Tarea",
  "description": "Descripcion Tarea"
}
```

## Vemos que se creó correctamente

Responses

Curl

```
curl -X 'POST' \
'http://localhost:8080/api/tasks' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "name": "Otra Tarea",
  "description": "Descripcion Tarea"
}'
```

Request URL

```
http://localhost:8080/api/tasks
```

Server response

Code

Details

201

Undocumented

Response body

```
{
  "id": 2,
  "name": "Otra Tarea",
  "description": "Descripcion Tarea"
}
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Sat,01 Nov 2025 22:31:01 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses

## Listamos las Tareas

Curl

```
curl -X 'GET' \
'http://localhost:8080/api/tasks' \
-H 'accept: */*'
```

Request URL

```
http://localhost:8080/api/tasks
```

Server response

Code

Details

200

Response body

```
{
  {
    "id": 1,
    "name": "Probar",
    "description": "Probar buen funcionamiento"
  },
  {
    "id": 2,
    "name": "Programar",
    "description": "Programar sitio Web"
  },
  {
    "id": 3,
    "name": "Otra Tarea",
    "description": "Descripcion Tarea"
  }
}
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Sat,01 Nov 2025 22:31:13 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

## Buscamos Tarea Por ID

Name	Description				
<div><div>id * required</div><div>integer(\$int64)</div><div>(path)</div></div> <div><input type="text" value="1"/></div>					
<div><div>Execute</div><div>Clear</div></div>					
Responses					
<div>Curl<div>curl -X 'GET' \ 'http://localhost:8080/api/tasks/1' \ -H 'accept: */*'</div></div> <div>Request URL<div>http://localhost:8080/api/tasks/1</div></div> <div>Server response</div> <table><thead><tr><th>Code</th><th>Details</th></tr></thead><tbody><tr><td>200</td><td><div>Response body<div>{ "id": 1, "name": "Probar", "description": "Probar buen funcionamiento" }</div></div></td></tr></tbody></table>		Code	Details	200	<div>Response body<div>{ "id": 1, "name": "Probar", "description": "Probar buen funcionamiento" }</div></div>
Code	Details				
200	<div>Response body<div>{ "id": 1, "name": "Probar", "description": "Probar buen funcionamiento" }</div></div>				

## Verificamos que no se creen Tareas con nombre vacio

Responses					
<div>Curl<div>curl -X 'POST' \ 'http://localhost:8080/api/tasks' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d { "name": "", "description": "Descripcion" }</div></div> <div>Request URL<div>http://localhost:8080/api/tasks</div></div> <div>Server response</div> <table><thead><tr><th>Code</th><th>Details</th></tr></thead><tbody><tr><td>400</td><td><div>Error: response status is 400</div><div>Response body<div>{ "name": "El campo name el tamaño debe estar entre 3 y 25" }</div></div><div>Response headers<div>connection: close content-type: application/json date: Mon,03 Nov 2025 17:36:47 GMT transfer-encoding: chunked</div></div></td></tr></tbody></table>		Code	Details	400	<div>Error: response status is 400</div> <div>Response body<div>{ "name": "El campo name el tamaño debe estar entre 3 y 25" }</div></div> <div>Response headers<div>connection: close content-type: application/json date: Mon,03 Nov 2025 17:36:47 GMT transfer-encoding: chunked</div></div>
Code	Details				
400	<div>Error: response status is 400</div> <div>Response body<div>{ "name": "El campo name el tamaño debe estar entre 3 y 25" }</div></div> <div>Response headers<div>connection: close content-type: application/json date: Mon,03 Nov 2025 17:36:47 GMT transfer-encoding: chunked</div></div>				

Vemos que se eliminen por ID las tareas

Name	Description
<b>id</b> * required integer(\$int64) (path)	<input type="text" value="4"/>

Execute

Clear

### Responses

Curl

```
curl -X 'DELETE' \  
  'http://localhost:8080/api/tasks/4' \  
  -H 'accept: */*'
```

Request URL

http://localhost:8080/api/tasks/4

Server response

Code	Details
200	<p>Response body</p> <pre>{   "id": 4,   "name": "Tarea 4",   "description": "Descripcion Tarea 4" }</pre> <div><div></div>Download</div>

Actualizamos la Tarea con el número 3 ya que el nombre es erróneo

Curl

```
curl -X 'PUT' \  
  'http://localhost:8080/api/tasks/3' \  
  -H 'accept: */*' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "name": "Diseñar",  
    "description": "Diseñar Pagina Web"  
  }'
```

Request URL

http://localhost:8080/api/tasks/3

Server response

Code	Details
201 <i>Undocumented</i>	<p>Response body</p> <pre>{   "id": 3,   "name": "Diseñar",   "description": "Diseñar Pagina Web" }</pre> <div><div></div>Download</div> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Sat,01 Nov 2025 22:37:41 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>

Volvemos a Listar y verificamos que se haya actualizado la tarea 3

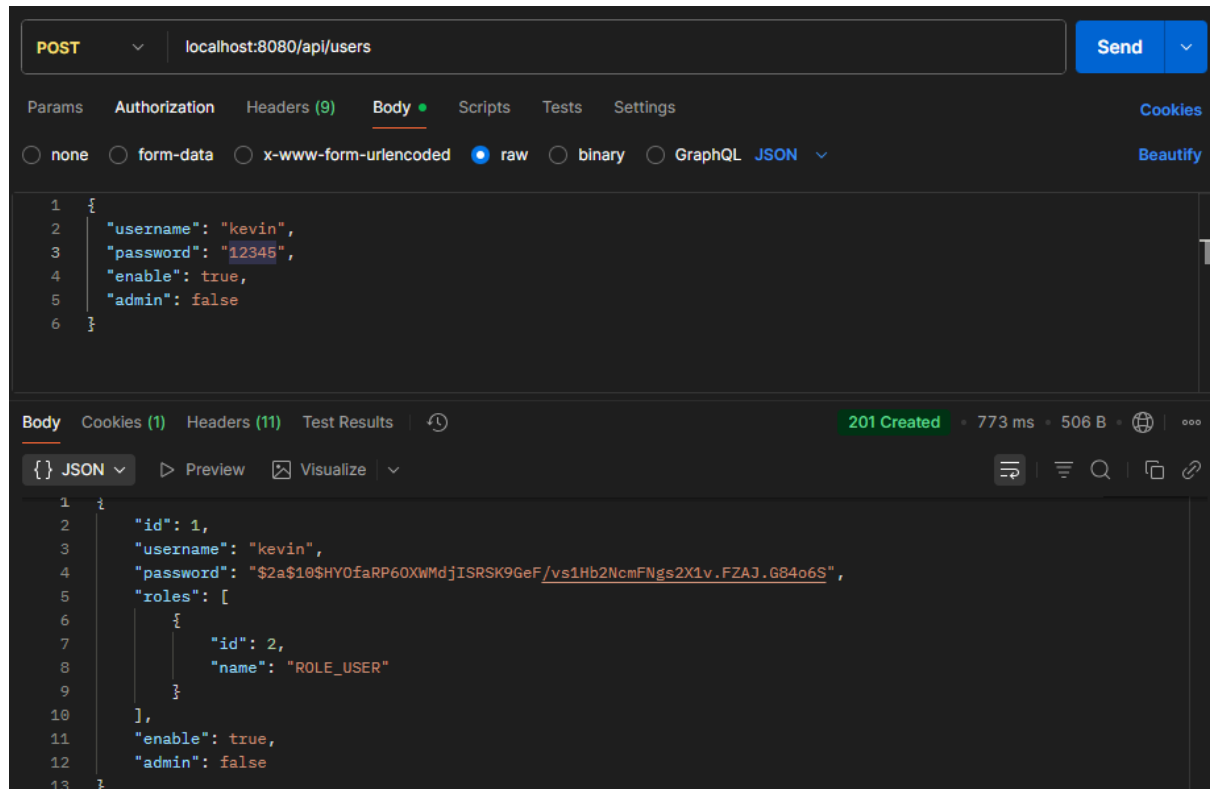
Request URL

http://localhost:8080/api/tasks

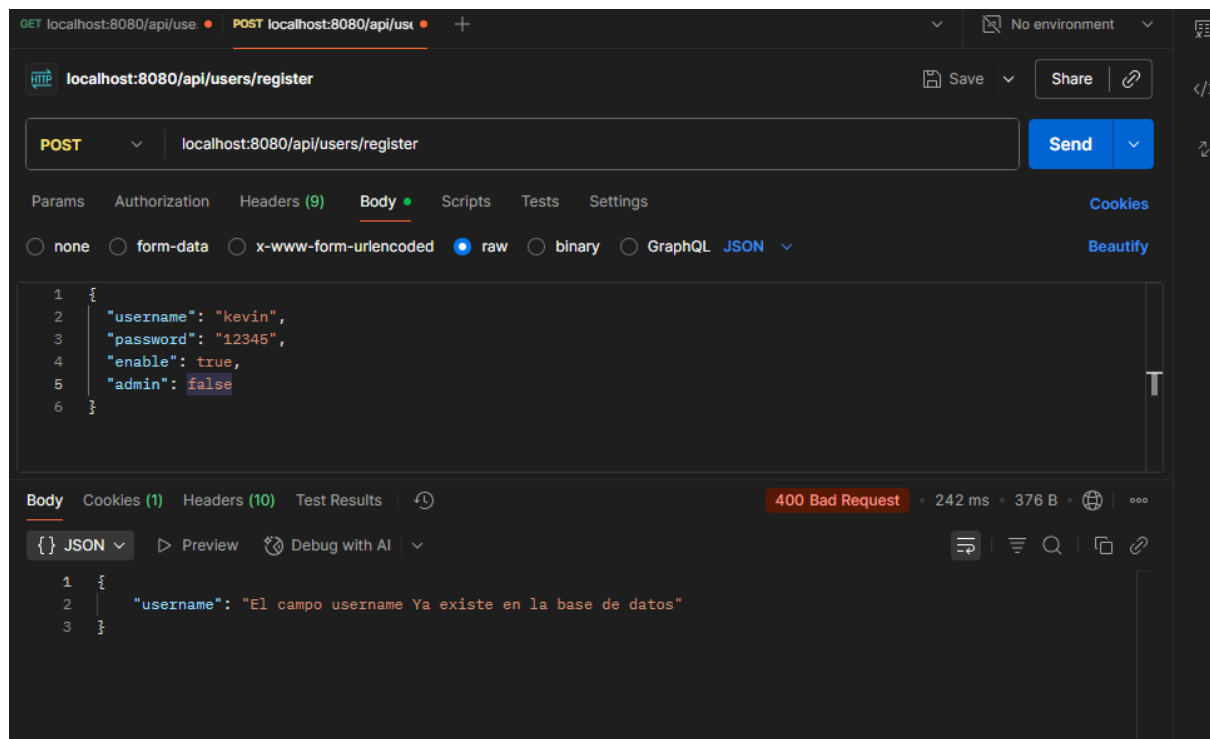
Server response

Code	Details
200	<p>Response body</p> <pre>[   {     "id": 1,     "name": "Probar",     "description": "Probar buen funcionamiento"   },   {     "id": 2,     "name": "Programar",     "description": "Programar sitio Web"   },   {     "id": 3,     "name": "Diseñar",     "description": "Diseñar Pagina Web"   } ]</pre> <div><div></div>Download</div> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Sat,01 Nov 2025 22:38:57 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>

Se crea un usuario con contraseña ya encriptada desde Postman



Se valida que no existan usuarios repetidos Postman



Se valida que se logue con un usuario existente Postman

POST localhost:8080/login

SaveShare

POSTlocalhost:8080/login

Send

ParamsAuthorizationHeaders (9)BodyScriptsTestsSettings

noneform-datax-www-form-urlencoderawbinaryGraphQLJSON

Beautify

```
1 {
2   "username": "kevin",
3   "password": "12345"
4 }
```

BodyCookies (1)Headers (12)Test Results

200 OK • 613 ms • 894 B

{ } JSONPreviewVisualize

```
1 {
2   "message": "Hola kevin has iniciado seccion con exito",
3   "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrZXZpbiIsImF1dG8iOiJ1b3R1cyIsImVudCI6IjE5OTYwMDE3IiwiaWF0IjoiMTY5OTYwMDE3In0.eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJrZXZpbiIsImF1dG8iOiJ1b3R1cyIsImVudCI6IjE5OTYwMDE3IiwiaWF0IjoiMTY5OTYwMDE3In0",
4   "username": "kevin"
5 }
```