

# API Validaciones

HTTP <http://localhost:8080/ejemplares/ejemplar> Save Share

POST <http://localhost:8080/ejemplares/ejemplar> Send

Params Authorization Headers (8) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "libroByIsbn": {
3     "isbn": "9781234567890",
4     "titulo": "El Quijote",
5     "autor": "Miguel de Cervantes"
6   }
7 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 16 ms 322 B

{ } JSON Preview Visualize

```
1 {
2   "timestamp": "2025-01-31T07:47:36.147+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='ejemplar'. Error count: 2",
6   "path": "/ejemplares/ejemplar"
7 }
```

```
AltMessageSourceResolvable: codes [ejemplar.estado,estado]; arguments []; default message [estado]; default message [Error en el estado, ingresa un valor que no sea nulo.]] ]
```

POST <http://localhost:8080/ejemplares/ejemplar> Save Share

POST <http://localhost:8080/ejemplares/ejemplar> Send

Params Authorization Headers (8) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "estado": "",
3   "libroByIsbn": {
4     "isbn": "9781234567890",
5     "titulo": "El Quijote",
6     "autor": "Miguel de Cervantes"
7   }
8 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 12 ms 322 B

{ } JSON Preview Visualize

```
1 {
2   "timestamp": "2025-01-31T07:48:30.961+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='ejemplar'. Error count: 2",
6   "path": "/ejemplares/ejemplar"
7 }
```

```
straints.Pattern$Flag;@5a6602a5,^(Prestado|Disponible|Dañado)$]; default message [El estado solo puede ser: Disponible, Prestado o Dañado (con la primera letra mayúscula).]] ]
```

http://localhost:8080/ejemplares/ejemplar

POST http://localhost:8080/ejemplares/ejemplar

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   |   "estado": "Disponible"
3   | }

```

Body Cookies Headers (4) Test Results 400 Bad Request 10 ms 322 B

JSON Preview Visualize

```
1 {
2   |   "timestamp": "2025-01-31T07:50:42.329+00:00",
3   |   "status": 400,
4   |   "error": "Bad Request",
5   |   "message": "Validation failed for object='ejemplar'. Error count: 1",
6   |   "path": "/ejemplares/ejemplar"
7   | }

```

```
s [ejemplar.libroByIsbn,libroByIsbn]; arguments []; default message [libroByIsbn]; default message [Error en el libro, ingresa un valor que no sea nulo.]] ]
```

http://localhost:8080/libros/libro

POST http://localhost:8080/libros/libro

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   |   "isbn": "rauw",
3   |   "titulo": "El Quijote",
4   |   "autor": "Miguel de Cervantes"
5   | }

```

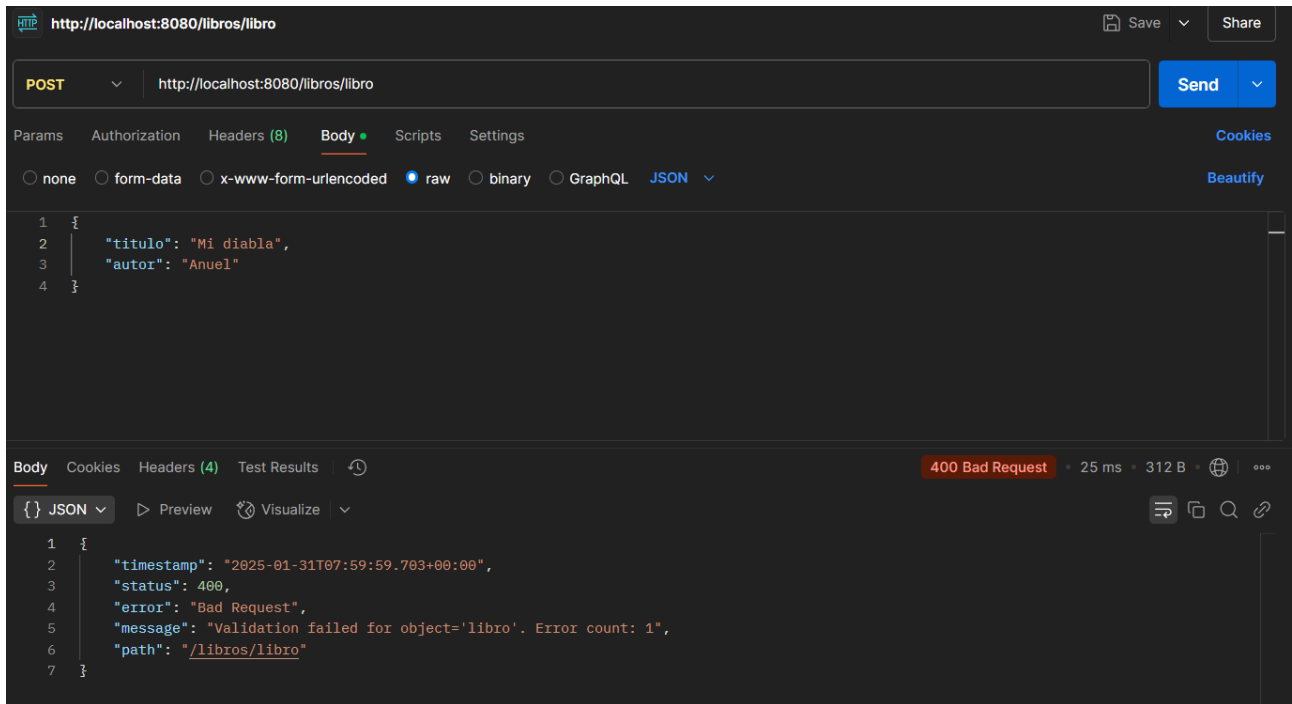
Body Cookies Headers (4) Test Results 400 Bad Request 13 ms 312 B

JSON Preview Visualize

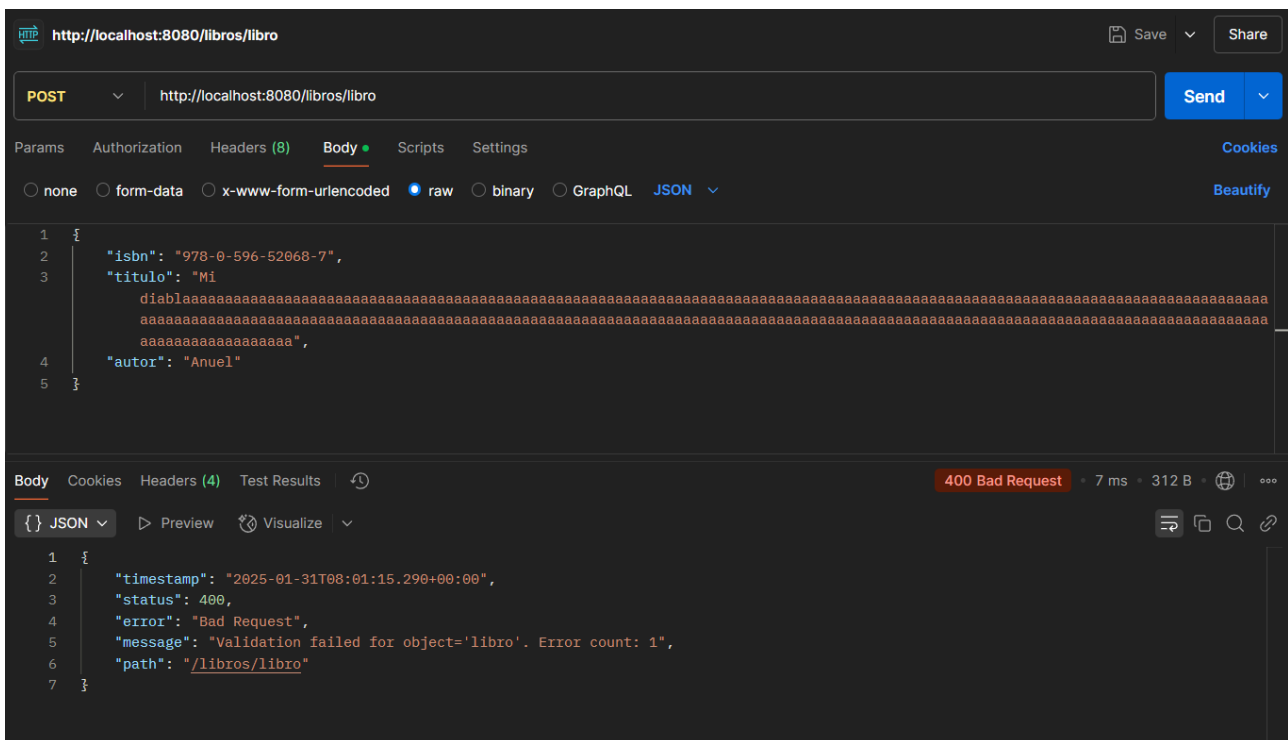
```
1 {
2   |   "timestamp": "2025-01-31T07:54:19.757+00:00",
3   |   "status": 400,
4   |   "error": "Bad Request",
5   |   "message": "Validation failed for object='libro'. Error count: 3",
6   |   "path": "/libros/libro"
7   | }

```

```
?pattern$Flag;@658fc65c,^97[89]-\d{-}\d{2,5}-\d{2,7}-\d; default message [Error en el ISBN, ingresa un valor con el patrón del siguiente ejemplo: ISBN-13: 978-0-596-52068-7.]] ]
```



```
support.DefaultMessageSourceResolvable: codes [libro.isbn,isbn]; arguments []; default message [isbn]; default message [Error en el ISBN, ingresa un valor que no sea nulo.] ]
```



```
table: codes [libro.titulo,titulo]; arguments []; default message [titulo],100,0]; default message [Error en el título, ingresa un valor que no tenga más de 100 caracteres]] ]
```

HTTP <http://localhost:8080/libros/libro> Save Share

**POST** <http://localhost:8080/libros/libro> Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   "isbn": "978-0-596-52068-7",
3   "titulo": "$",
4   "autor": "Anuel"
5 }
```

**Body** Cookies Headers (4) Test Results 400 Bad Request 6 ms 312 B 🌐 ⋮

**{}** **JSON** Preview Visualize 🔍 📄 🔍 🔗

```
1 {
2   "timestamp": "2025-01-31T08:02:26.586+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='libro'. Error count: 1",
6   "path": "/libros/libro"
7 }
```

```
akarta.validation.constraints.Pattern$Flag;@609966ae,[A-Za-z0-9ΆΈΙΌÚάέιόύñ ]+);$]; default message [Error en el titulo, ingresa un titulo que solo tenga letras y números.]] ]
```

HTTP <http://localhost:8080/libros/libro> Save Share

**POST** <http://localhost:8080/libros/libro> Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   "isbn": "978-0-596-52068-7",
3   "autor": "Anuel"
4 }
```

**Body** Cookies Headers (4) Test Results 400 Bad Request 7 ms 312 B 🌐 ⋮

**{}** **JSON** Preview Visualize 🔍 📄 🔍 🔗

```
1 {
2   "timestamp": "2025-01-31T08:03:34.620+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='libro'. Error count: 2",
6   "path": "/libros/libro"
7 }
```

```
essageSourceResolvable: codes [libro.titulo,titulo]; arguments []; default message [titulo]]; default message [Error en el titulo, ingresa un valor que no sea nulo.]] ]
```

[illegible]

The screenshot shows the Postman application interface. At the top, the URL bar displays `http://localhost:8080/libros/libro`. The request method is set to **POST**. The request body is a JSON object: 

```
{  "isbn": "978-0-596-52068-7",  "titulo": "Mi diablo",  "autor": "$"}
```

. The request is configured with the **raw** body type and **JSON** format. The response status is **400 Bad Request** with a response time of 7 ms and a size of 312 B. The response body is a JSON object: 

```
{  "timestamp": "2025-01-31T00:04:50.443+00:00",  "status": 400,  "error": "Bad Request",  "message": "Validation failed for object='libro'. Error count: 1",  "path": "/libros/libro"}
```

http://localhost:8080/libros/libro

SaveShare

POST

http://localhost:8080/libros/libro

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Beautifulify

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "isbn": "978-0-596-52068-7",
3   "titulo": "Mi diablo"
4 }
```

Body

Cookies

Headers (4)

Test Results

400 Bad Request

7 ms

312 B

Visualize

JSON

Preview

Visualize

```
1 {
2   "timestamp": "2025-01-31T08:05:44.200+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='libro'. Error count: 2",
6   "path": "/libros/libro"
7 }
```

t.DefaultMessageSourceResolvable: codes [libro.autor,autor]; arguments []; default message [autor]; default message [Error en el autor, ingresa un valor que no sea nulo.] ]

http://localhost:8080/prestamos/prestamo

SaveShare

POST

http://localhost:8080/prestamos/prestamo

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Beautifulify

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "fechaDevolucion": null,
3   "usuarioByUsuarioId": {
4     "id": 1,
5     "dni": "12345678A",
6     "nombre": "Juan Pérez",
7     "email": "juan.perez@example.com",
8     "password": "password123",
9     "tipo": "normal",
10    "penalizacionHasta": null
11  },
12 }
```

Body

Cookies

Headers (4)

Test Results

400 Bad Request

63 ms

321 B

Visualize

JSON

Preview

Visualize

```
1 {
2   "timestamp": "2025-01-31T08:32:54.015+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='prestamo'. Error count: 2",
6   "path": "/prestamos/prestamo"
7 }
```

.amo.fechaDevolucion,fechaDevolucion]; arguments []; default message [fechaDevolucion]; default message [Error en la fecha de devolución, ingresa un valor que no sea nulo]] ]

http://localhost:8080/prestamos/prestamo

POST

http://localhost:8080/prestamos/prestamo

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {
2   "fechaInicio": "2024-11-01",
3   "fechaDevolucion": null,
4   "ejemplarByEjemplarId": {
5     "id": 3,
6     "estado": "Prestado",
7     "libroByIsbn": {
8       "isbn": "9781234567890",
9       "titulo": "El Quijote",
10      "autor": "Miguel de Cervantes"
11    }
12  }
13 }
```

Body

Cookies

Headers (4)

Test Results

400 Bad Request

20 ms

321 B

JSON

Preview

Visualize

```
1 {
2   "timestamp": "2025-01-31T08:35:08.325+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='prestamo'. Error count: 1",
6   "path": "/prestamos/prestamo"
7 }
```

```
estamo.usuarioByUsuarioId,usuarioByUsuarioId]; arguments []; default message [usuarioByUsuarioId]; default message [Error en el usuario, ingresa un valor que no sea nulo]] ]
```

http://localhost:8080/prestamos/prestamo

POST

http://localhost:8080/prestamos/prestamo

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
3   "fechaDevolucion": null,
4   "usuarioByUsuarioId": {
5     "id": 1,
6     "dni": "12345678A",
7     "nombre": "Juan Pérez",
8     "email": "juan.perez@example.com",
9     "password": "password123",
10    "tipo": "normal",
11    "penalizacionHasta": null
12  }
13 }
```

Body

Cookies

Headers (4)

Test Results

400 Bad Request

7 ms

321 B

JSON

Preview

Visualize

```
1 {
2   "timestamp": "2025-01-31T08:35:51.107+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='prestamo'. Error count: 1",
6   "path": "/prestamos/prestamo"
7 }
```

```
ejemplarByEjemplarId,ejemplarByEjemplarId]; arguments []; default message [ejemplarByEjemplarId]; default message [Error en el ejemplar, ingresa un valor que no sea nulo]] ]
```

http://localhost:8080/prestamos/prestamo

POST http://localhost:8080/prestamos/prestamo

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "fechaInicio": "2024-11-01",
3   "usuarioByUsuarioId": {
4     "id": 1,
5     "dni": "12345678A",
6     "nombre": "Juan Pérez",
7     "email": "juan.perez@example.com",
8     "password": "password123",
9     "tipo": "normal",
10    "penalizacionHasta": null
11  },
12 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 176 ms 321 B

{ JSON Preview Visualize

```
1 {
2   "timestamp": "2025-01-31T08:37:27.985+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='prestamo'. Error count: 1",
6   "path": "/prestamos/prestamo"
7 }
```

```
.amo.fechaDevolucion,fechaDevolucion]; arguments []; default message [fechaDevolucion]]; default message [Error en la fecha de devolución, ingresa un valor que no sea nulo]] ]
```

http://localhost:8080/usuarios/usuario

POST http://localhost:8080/usuarios/usuario

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "dni": "12345678",
3   "nombre": "Juan Pérez",
4   "email": "juan.perez@example.com",
5   "password": "password123",
6   "tipo": "normal",
7   "penalizacionHasta": null
8 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 168 ms 318 B

{ JSON Preview Visualize

```
1 {
2   "timestamp": "2025-01-31T08:39:15.652+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='usuario'. Error count: 4",
6   "path": "/usuarios/usuario"
7 }
```



HTTP <http://localhost:8080/usuarios/usuario> Save Share

**POST** <http://localhost:8080/usuarios/usuario> Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON Beautify

```
1 {
2   "dni": "12345678",
3   "nombre": "Juan Pérez",
4   "email": "juanperez@gmail.com",
5   "password": "pass12",
6   "tipo": "normal",
7   "penalizacionHasta": null
8 }
```

**Body** Cookies Headers (4) Test Results 400 Bad Request • 8 ms • 318 B • 🌐 ⋮

{} **JSON** Preview Visualize ⌵ 🔍 🔗

```
1 {
2   "timestamp": "2025-01-31T08:42:03.630+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='usuario'. Error count: 2",
6   "path": "/usuarios/usuario"
7 }
```

```
rints.Pattern$Flag;86945e689,^(\\d{8}[A-HJ-NP-TV-Z]|[XYZ]\\d{7}[A-HJ-NP-TV-Z])$}; default message [Error en el DNI o NIE, ingresa un DNI o NIE que tenga un patrón correcto.]]
```

HTTP <http://localhost:8080/usuarios/usuario> Save Share

**POST** <http://localhost:8080/usuarios/usuario> Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON Beautify

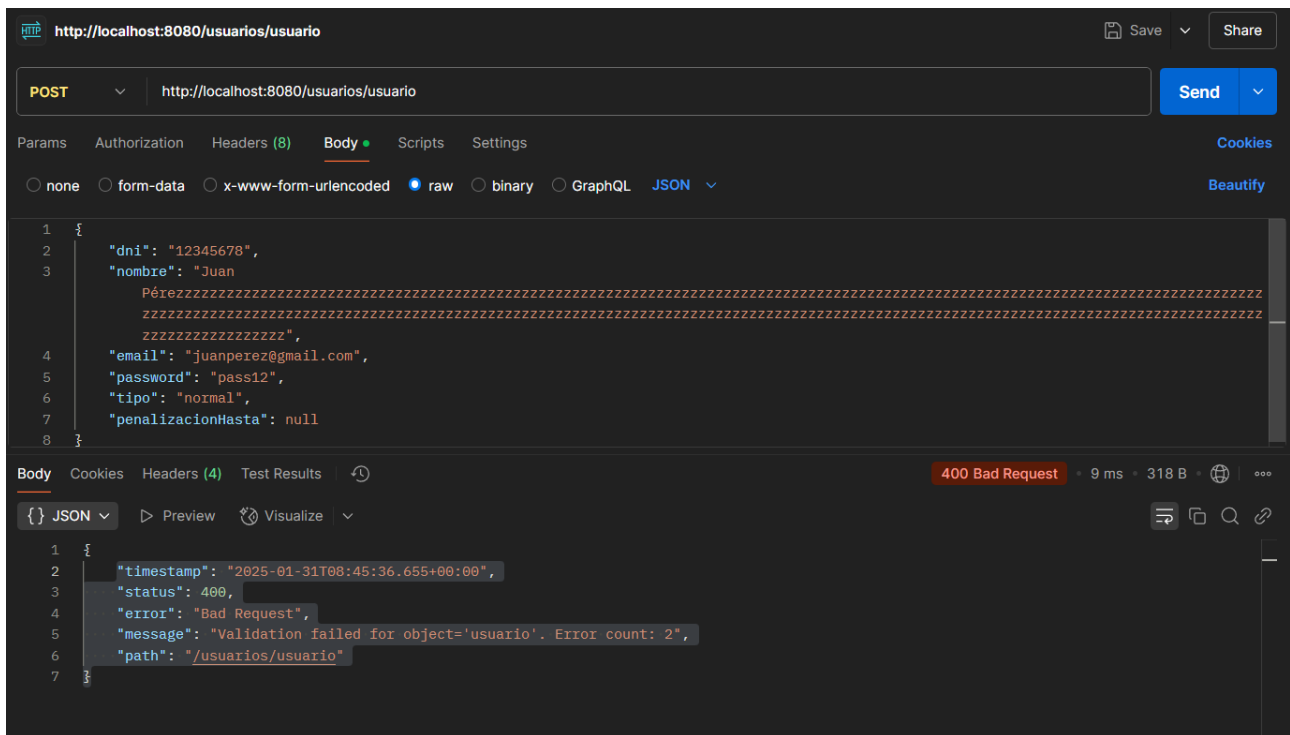
```
1 {
2   "nombre": "Juan Pérez",
3   "email": "juanperez@gmail.com",
4   "password": "pass12",
5   "tipo": "normal",
6   "penalizacionHasta": null
7 }
```

**Body** Cookies Headers (4) Test Results 400 Bad Request • 173 ms • 318 B • 🌐 ⋮

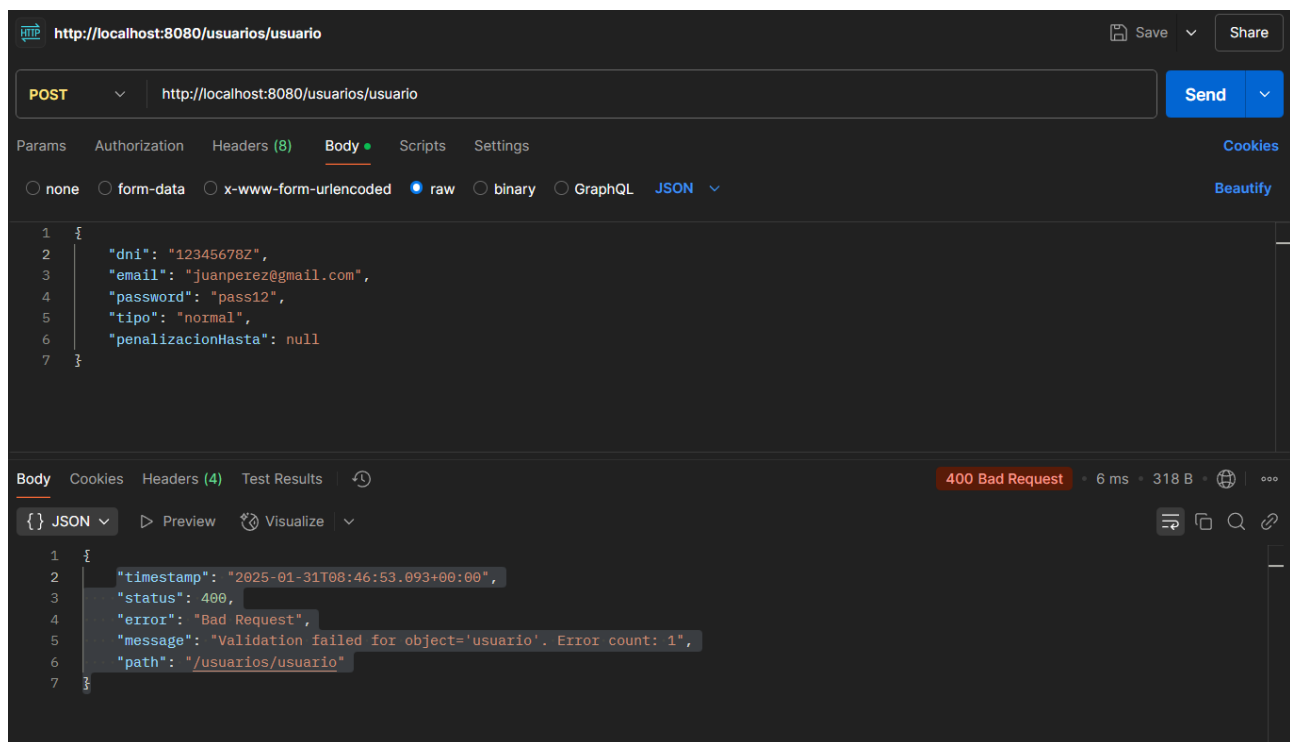
{} **JSON** Preview Visualize ⌵ 🔍 🔗

```
1 {
2   "timestamp": "2025-01-31T08:44:39.885+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='usuario'. Error count: 2",
6   "path": "/usuarios/usuario"
7 }
```

```
DefaultMessageSourceResolvable: codes [usuario.dni,dni]; arguments []; default message [dni]; default message [Error en el DNI o NIE, ingresa un valor que no esté vacío.]] ]
```



```
pl: codes [usuario.nombre,nombre]; arguments []; default message [nombre],100,0]; default message [Error en el nombre, ingresa un valor que no tenga más de 100 caracteres]] ]
```



```
tMessageSourceResolvable: codes [usuario.nombre,nombre]; arguments []; default message [nombre]]; default message [Error en el nombre, ingresa un valor que no sea nulo.] ]
```

http://localhost:8080/usuarios/usuario

POST http://localhost:8080/usuarios/usuario

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "dni": "12345678Z",
3   "nombre": "Juan Pérez",
4   "email": "juanperez@example.com",
5   "password": "pass12",
6   "tipo": "normal",
7   "penalizacionHasta": null
8 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 7 ms 318 B

{ } JSON Preview Visualize

```
1 {
2   "timestamp": "2025-01-31T08:47:53.505+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='usuario'. Error count: 1",
6   "path": "/usuarios/usuario"
7 }
```

```
rt.validation.constraints.Pattern$Flag;019fdec1f,([A-Za-z0-9]{1,50}@gmail.com)); default message [Error en el email, ingresa un email que sea del proveedor de Gmail.]] ]
```

http://localhost:8080/usuarios/usuario

POST http://localhost:8080/usuarios/usuario

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "dni": "12345678Z",
3   "nombre": "Juan Pérez",
4   "password": "pass12",
5   "tipo": "normal",
6   "penalizacionHasta": null
7 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 8 ms 318 B

{ } JSON Preview Visualize

```
1 {
2   "timestamp": "2025-01-31T08:48:48.218+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='usuario'. Error count: 2",
6   "path": "/usuarios/usuario"
7 }
```

```
rt.DefaultMessageSourceResolvable: codes [usuario.email,email]; arguments []; default message [email]; default message [Error en el email, ingresa un valor que no sea nulo.]]
```

POST http://localhost:8080/usuarios/usuario

Send

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "dni": "12345678Z",
3   "nombre": "Juan Pérez",
4   "email": "juanperez@gmail.com",
5   "password": "pass12asdasdasdads",
6   "tipo": "normal",
7   "penalizacionHasta": null
8 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 6 ms 318 B

{ JSON Preview Visualize

```
1 {
2   "timestamp": "2025-01-31T08:49:50.471+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='usuario'. Error count: 1",
6   "path": "/usuarios/usuario"
7 }
```

Pattern\$Flag:@19fdec1f,^[A-Za-z0-9]{4,8}+);\$]; default message [Error en la contraseña, ingresa una contraseña que tenga de 4 a 12 caracteres que tenga letras y números.]] ]

POST http://localhost:8080/usuarios/usuario

Send

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "dni": "12345678Z",
3   "nombre": "Juan Pérez",
4   "email": "juanperez@gmail.com",
5   "password": "$",
6   "tipo": "normal",
7   "penalizacionHasta": null
8 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 7 ms 318 B

{ JSON Preview Visualize

```
1 {
2   "timestamp": "2025-01-31T08:50:32.080+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='usuario'. Error count: 1",
6   "path": "/usuarios/usuario"
7 }
```

s.Pattern\$Flag:@19fdec1f,^[A-Za-z0-9]{4,8}+);\$]; default message [Error en la contraseña, ingresa una contraseña que tenga de 4 a 12 caracteres que tenga letras y números.]] ]

HTTP <http://localhost:8080/usuarios/usuario> Save Share

POST <http://localhost:8080/usuarios/usuario> Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "dni": "12345678Z",
3   "nombre": "Juan Pérez",
4   "email": "juanperez@gmail.com",
5   "password": "pass12",
6   "tipo": "Normal",
7   "penalizacionHasta": null
8 }
```

**Body** Cookies Headers (4) Test Results 400 Bad Request • 6 ms • 318 B

{ JSON Preview Visualize

```
1 {
2   "timestamp": "2025-01-31T08:51:26.076+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='usuario'. Error count: 1",
6   "path": "/usuarios/usuario"
7 }
```

```
ta.validation.constraints.Pattern$Flag;@19fdec1f,^(normal|administrador)$}); default message [El usuario solo puede ser: normal o administrador (todo en minúsculas.)]
```

HTTP <http://localhost:8080/usuarios/usuario> Save Share

POST <http://localhost:8080/usuarios/usuario> Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "dni": "12345678Z",
3   "nombre": "Juan Pérez",
4   "email": "juanperez@gmail.com",
5   "password": "pass12",
6   "penalizacionHasta": null
7 }
```

**Body** Cookies Headers (5) Test Results 200 OK • 85 ms • 317 B

{ JSON Preview Visualize

```
1 {
2   "id": 9,
3   "dni": "12345678Z",
4   "nombre": "Juan García García",
5   "email": "juangg@gmail.com",
6   "password": "asahbi",
7   "tipo": "administrador",
8   "penalizacionHasta": null
9 }
```

```
le: codes [usuario.tipo,tipo]; arguments []; default message [tipo]]; default message [Error en el tipo de usuario, ingresa un valor que no sea nulo.]]
```