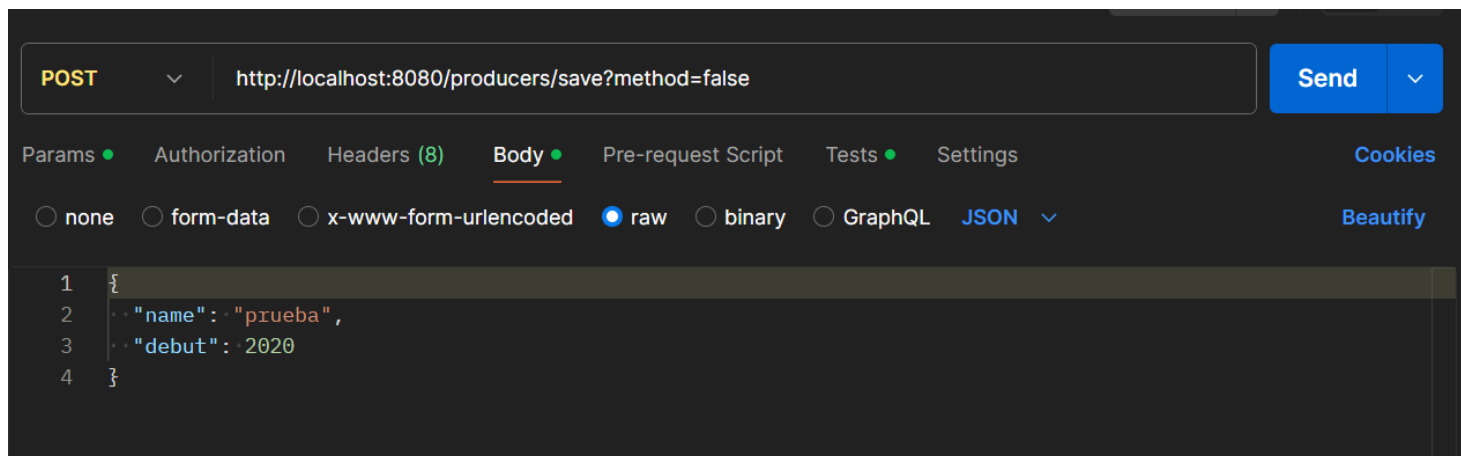


| Ejecución Automática de Caso de Test | |
|--|------------------|
| Dominio | Proyecto |
| | |
| Aplicación | Ámbito Ejecución |
| NOMBRE DE LA APLICACIÓN Films | Offline |
| Descripción | Pre-requisitos |
| Prueba unitaria para Proyecto Spring Films - Producers | |
| Tester Ejecución | Fecha Ejecución |
| | 09/05/2024 |

| | |
|---------------------------|--|
| # Paso: | 1 |
| Descripción | Create Producer JPA |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 201 Created. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo POST la siguiente URL:

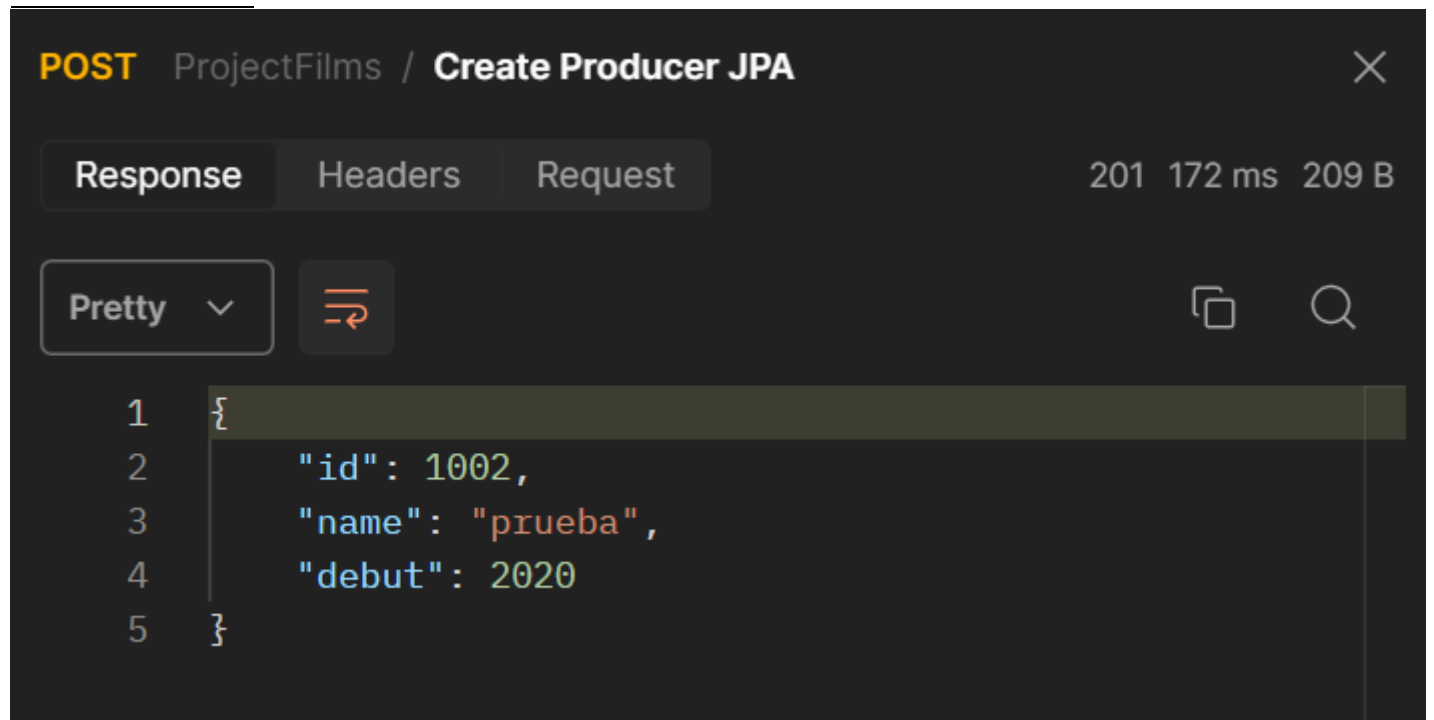
http://localhost:8080/producers/save?method=false



Con el siguiente body

```
{  
  "name": "prueba",  
  "debut": 2020  
}
```

Revisamos la salida:



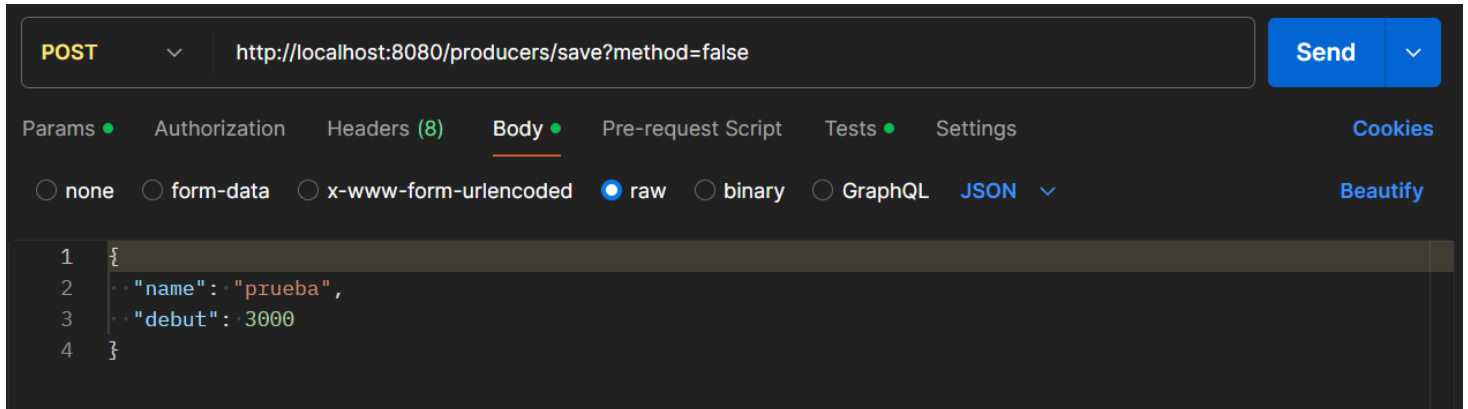
The screenshot shows a REST client interface with a dark theme. The top bar indicates a **POST** request to `ProjectFilms / Create Producer JPA`. Below this, there are tabs for **Response**, **Headers**, and **Request**. The **Response** tab is active, showing a status of **201**, a time of **172 ms**, and a size of **209 B**. The response body is displayed in a code editor with a 'Pretty' button and a search icon. The JSON response is as follows:

```
1 {  
2   "id": 1002,  
3   "name": "prueba",  
4   "debut": 2020  
5 }
```

| | |
|--------------------|--|
| # Paso: | 2 |
| Descripción | Create Producer JPA Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 400 Bad Request. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo POST la siguiente URL:

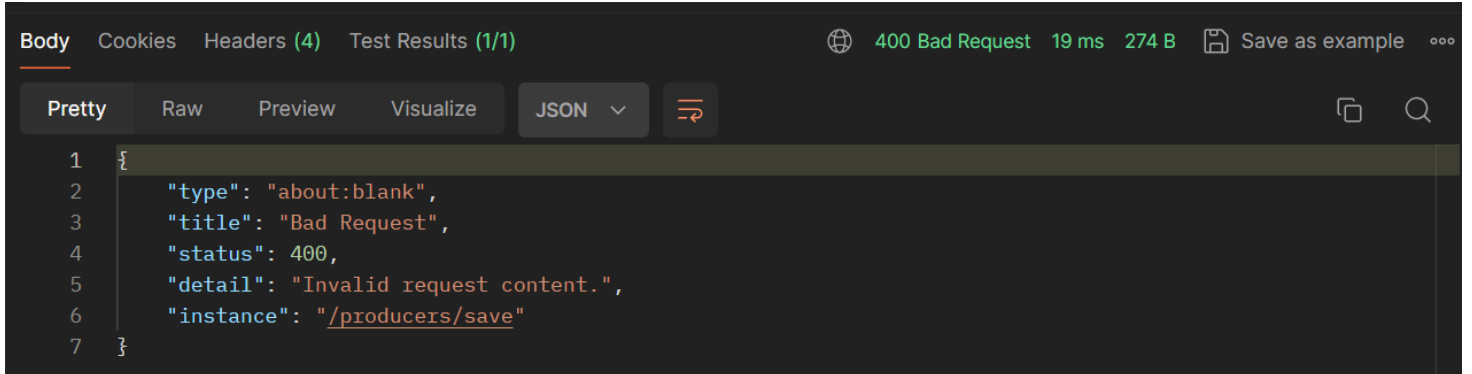
`http://localhost:8080/producers/save?method=false`



Con el siguiente body

```
{  
  "name": "prueba",  
  "debut": 3000  
}
```

Revisamos la salida:



The screenshot shows a web browser's developer console with the 'Body' tab selected. The console displays a 400 Bad Request error. The JSON response body is as follows:

```
1 {  
2   "type": "about:blank",  
3   "title": "Bad Request",  
4   "status": 400,  
5   "detail": "Invalid request content.",  
6   "instance": "/producers/save"  
7 }
```

| | |
|---------------------------|---|
| # Paso: | 3 |
| Descripción | Update Producer JPA |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo PUT la siguiente URL:

http://localhost:8080/producers/update?method=false

PUT

http://localhost:8080/producers/update?method=false

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```

1 {
2   ... "id": 2,
3   ... "name": "update",
4   ... "debut": 2000
5 }
```

Con el siguiente body

```
{
  "id": 2,
  "name": "update",
  "debut": 2000
}
```

Revisamos la salida:



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

- Method:** PUT
- Path:** ProjectFilms / Update Producer JPA
- Response Status:** 200
- Response Time:** 40 ms
- Response Size:** 201 B
- Response Body:** A JSON object with the following structure:

```
{
  "id": 2,
  "name": "update",
  "debut": 2000
}
```

| | |
|---------------------------|--|
| # Paso: | 4 |
| Descripción | Update Producer JPA Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 400 Bad Request. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo PUT la siguiente URL:

http://localhost:8080/producers/update?method=false

PUT

⌵

http://localhost:8080/producers/update?method=false

Send

⌵

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

⬢ none

⬢ form-data

⬢ x-www-form-urlencoded

⬢ raw

⬢ binary

⬢ GraphQL

JSON

⌵

Beautify

1

{

2

"id": 2,

3

"name": "update",

4

"debut": 3000

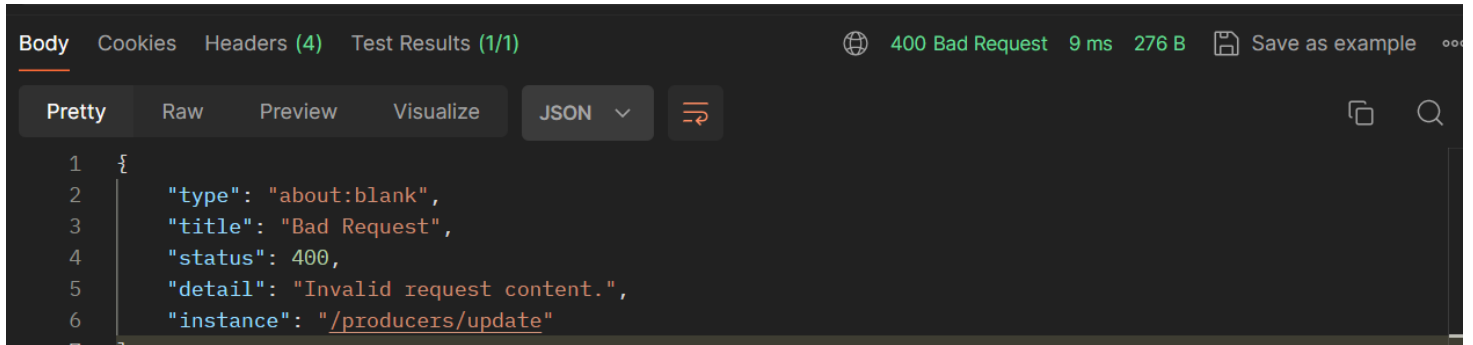
5

}

Con el siguiente body

```
{
  "id": 2,
  "name": "update",
  "debut": 3000
}
```

Revisamos la salida:



The screenshot shows a web browser's developer console with the 'Body' tab selected. The console displays a 400 Bad Request error. The error details are as follows:

```
{
  "type": "about:blank",
  "title": "Bad Request",
  "status": 400,
  "detail": "Invalid request content.",
  "instance": "/producers/update"
}
```

The console also shows the status '400 Bad Request', a response time of '9 ms', and a size of '276 B'. There are buttons for 'Save as example' and a search icon.

| | |
|---------------------------|--|
| # Paso: | 5 |
| Descripción | Create Producer Criteria |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 201 Created. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo POST la siguiente URL:

http://localhost:8080/producers/save?method=true

POST

▼

http://localhost:8080/producers/save?method=true

Send ▼

Params ●

Authorization

Headers (8)

Body ●

Pre-request Script

Tests ●

Settings

Cookies

○ none

○ form-data

○ x-www-form-urlencoded

● raw

○ binary

○ GraphQL

JSON ▼

Beautify

```

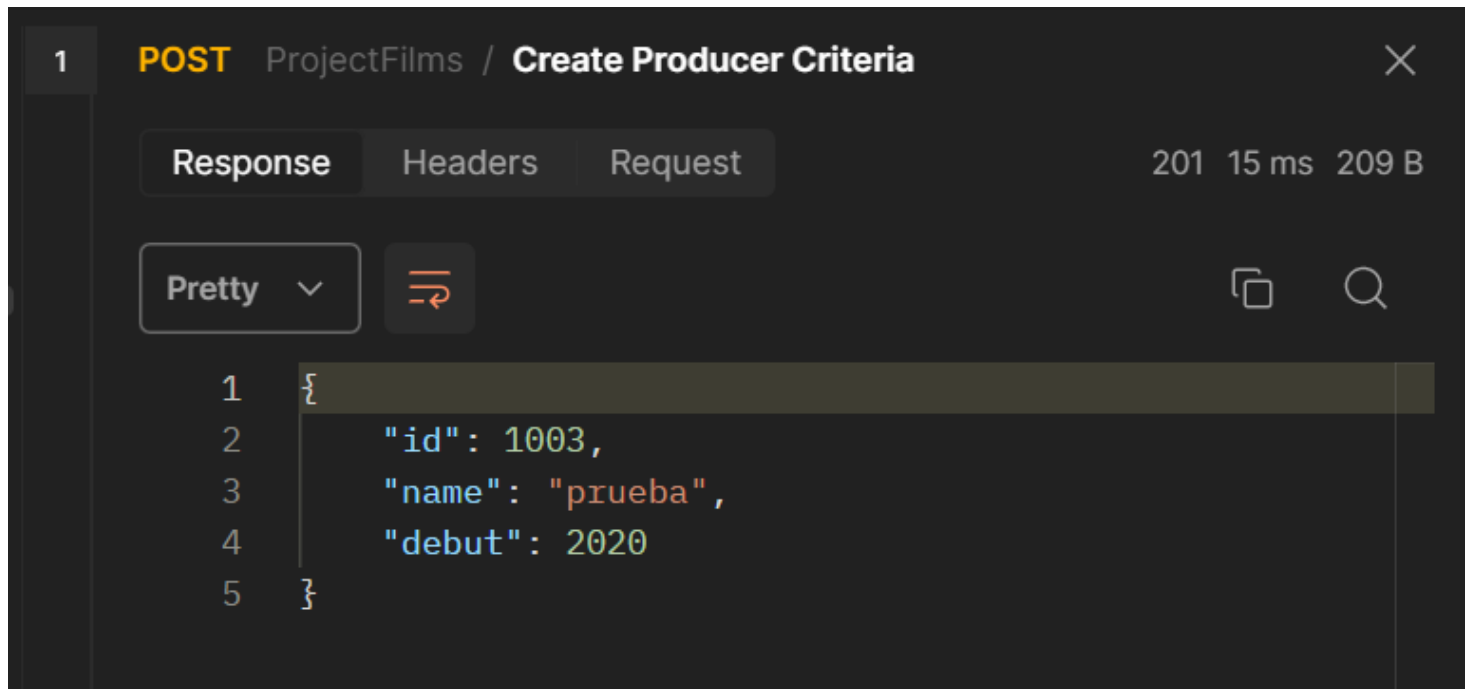
1  {
2    · "name": "prueba",
3    · "debut": 2020
4  }

```

Con el siguiente body

```
{  
  "name": "prueba",  
  "debut": 2020  
}
```

Revisamos la salida:



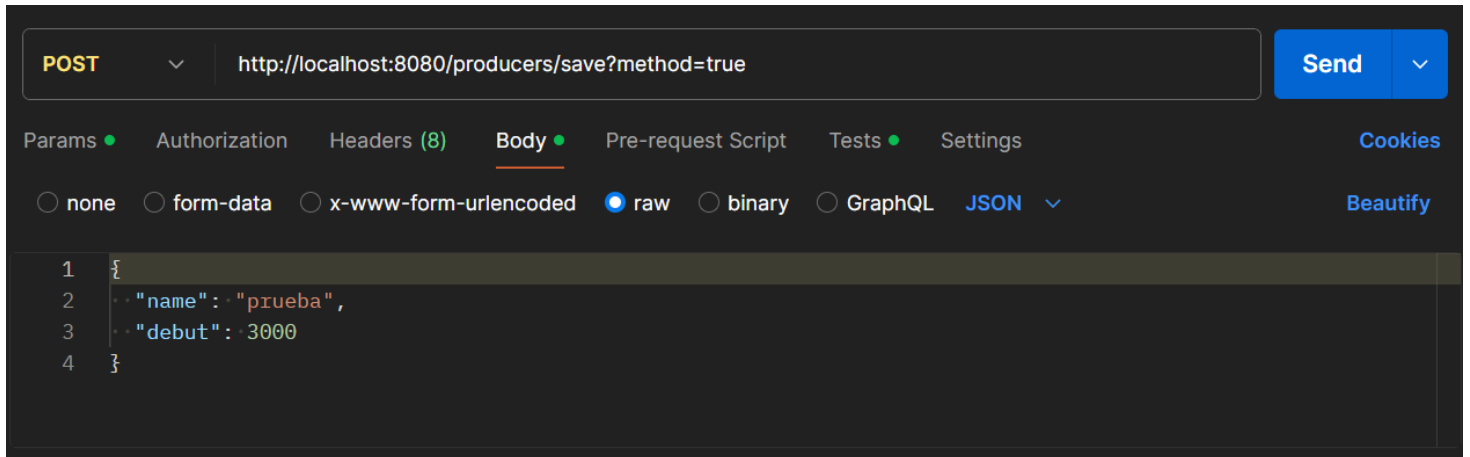
The screenshot shows a REST client interface with a dark theme. The top bar indicates a **POST** request to `ProjectFilms / Create Producer Criteria`. Below the bar, there are tabs for **Response**, **Headers**, and **Request**. The **Response** tab is selected, showing a status of **201**, a time of **15 ms**, and a size of **209 B**. A **Pretty** button with a dropdown arrow is visible. The response body is displayed in a code editor with line numbers 1 through 5. The JSON response is:

```
1 {  
2   "id": 1003,  
3   "name": "prueba",  
4   "debut": 2020  
5 }
```

| | |
|---------------------------|--|
| # Paso: | 6 |
| Descripción | Create Producer Criteria Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 400 Bad Request. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo POST la siguiente URL:

`http://localhost:8080/producers/save?method=true`



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

- Method:** POST
- URL:** http://localhost:8080/producers/save?method=true
- Body Type:** raw (selected)
- Body Content:**

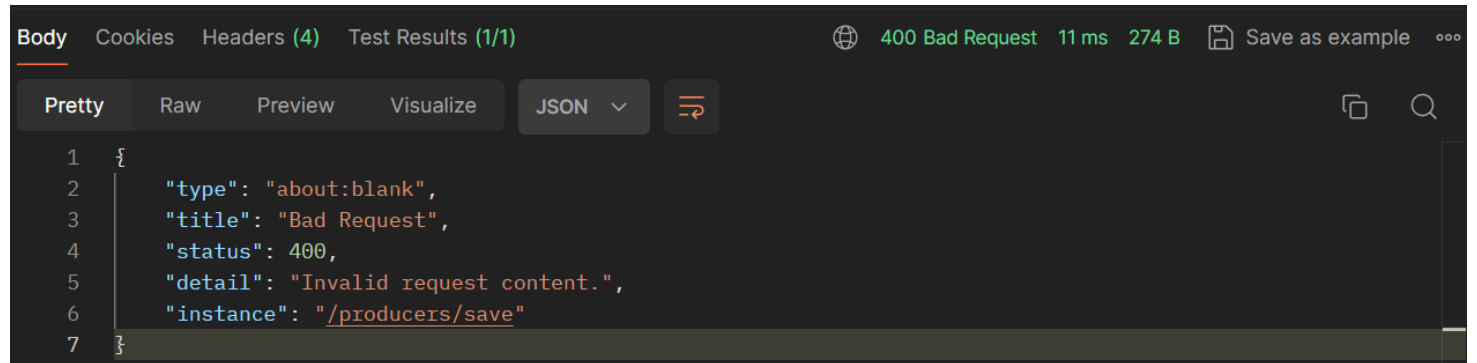
```

1 {
2   "name": "prueba",
3   "debut": 3000
4 }
```
- Buttons:** Send, Beautify
- Tabs:** Params, Authorization, Headers (8), Body (selected), Pre-request Script, Tests, Settings, Cookies

Con el siguiente body

```
{  
  "name": "prueba",  
  "debut": 3000  
}
```

Revisamos la salida:



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

- Body** tab selected.
- Response status: **400 Bad Request** (11 ms, 274 B).
- Buttons: **Pretty** (selected), **Raw**, **Preview**, **Visualize**, **JSON** (dropdown), and a refresh icon.
- Response body (JSON):

```
1 {  
2   "type": "about:blank",  
3   "title": "Bad Request",  
4   "status": 400,  
5   "detail": "Invalid request content.",  
6   "instance": "/producers/save"  
7 }
```

| | |
|---------------------------|---|
| # Paso: | 7 |
| Descripción | Update Producer Criteria |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo PUT la siguiente URL:

http://localhost:8080/producers/update?method=true

PUT

⌵

http://localhost:8080/producers/update?method=true

Send

⌵

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

⌵

Beautify

1

{

2

··· "id": 2,

3

··· "name": "update",

4

··· "debut": 2000

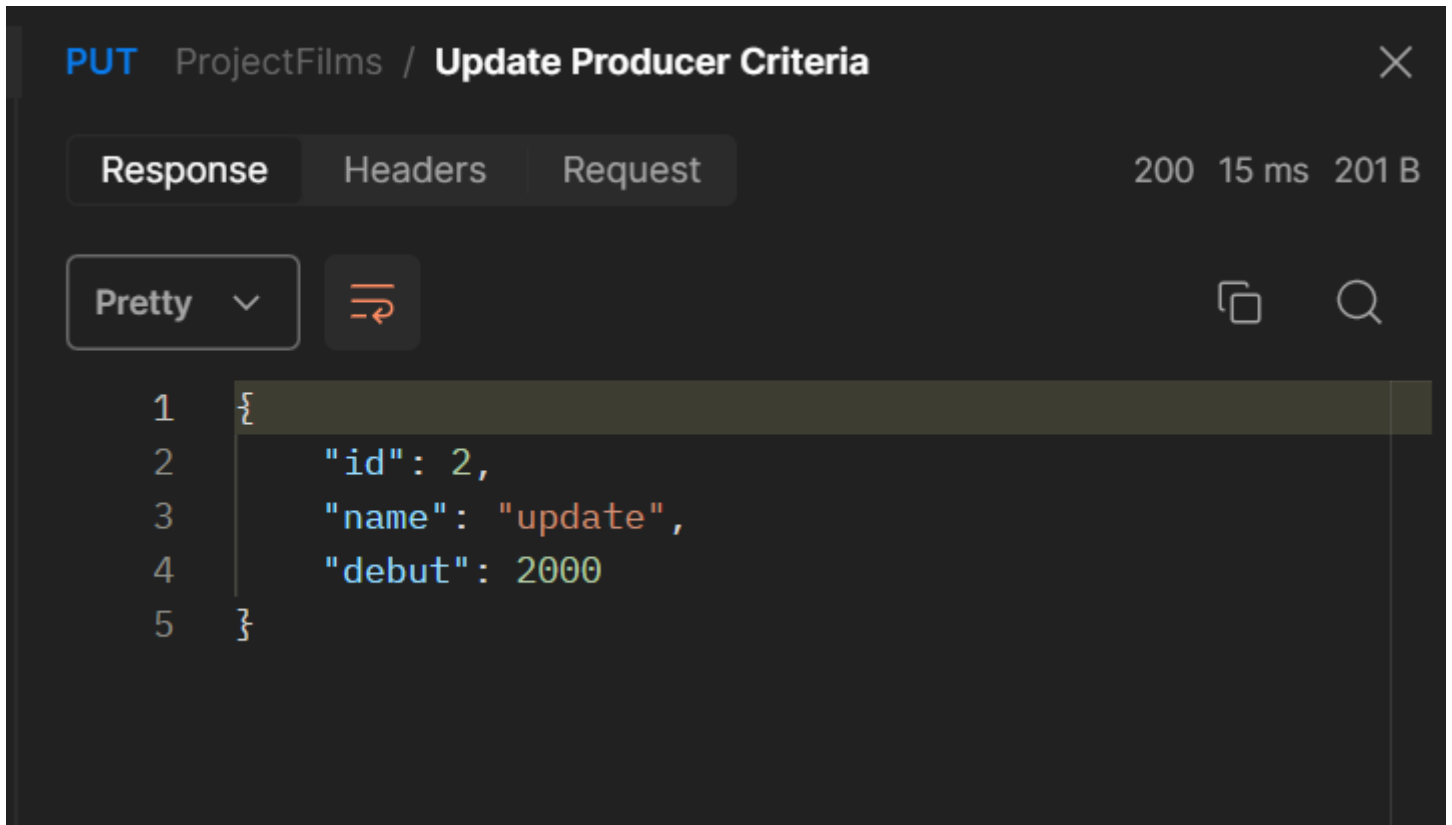
5

}

Con el siguiente body

```
{  
  "id": 2,  
  "name": "update",  
  "debut": 2000  
}
```

Revisamos la salida:



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

- Method:** PUT
- Path:** ProjectFilms / Update Producer Criteria
- Response Status:** 200
- Response Time:** 15 ms
- Response Size:** 201 B
- Response Body:** A JSON object with the following structure:

```
{  
  "id": 2,  
  "name": "update",  
  "debut": 2000  
}
```

| | |
|---------------------------|--|
| # Paso: | 8 |
| Descripción | Update Producer Criteria Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 400 Bad Request. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo PUT la siguiente URL:

http://localhost:8080/producers/update?method=true

PUT

http://localhost:8080/producers/update?method=true

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```

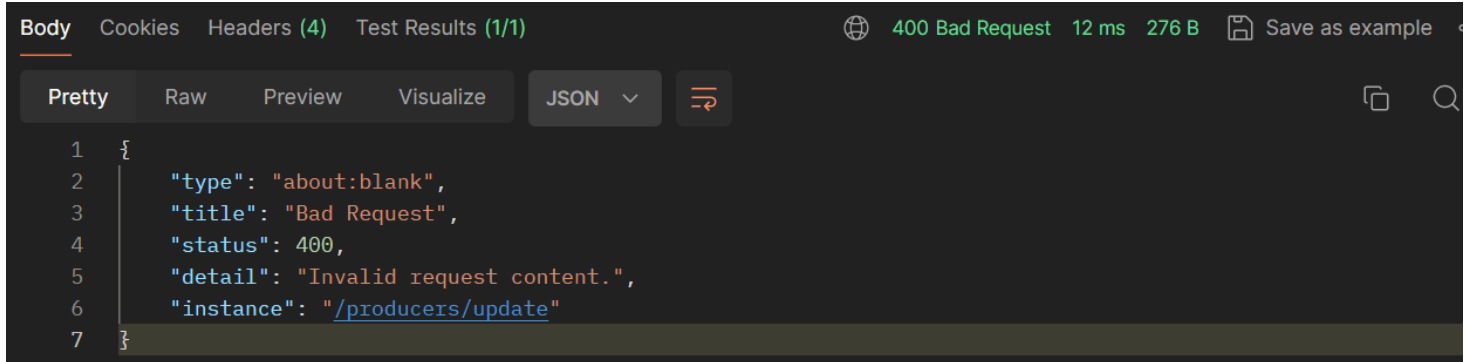
1  {
2    ... "id": 2,
3    ... "name": "update",
4    ... "debut": 3000
5  }

```


Con el siguiente body

```
{
  "id": 2,
  "name": "update",
  "debut": 3000
}
```

Revisamos la salida:



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

- Body** tab selected.
- Response status: **400 Bad Request** (12 ms, 276 B).
- Buttons: **Pretty** (selected), Raw, Preview, Visualize, JSON (dropdown), and a refresh icon.
- Response body (JSON):

```
1 {
2   "type": "about:blank",
3   "title": "Bad Request",
4   "status": 400,
5   "detail": "Invalid request content.",
6   "instance": "/producers/update"
7 }
```

| | |
|--------------------|---|
| # Paso: | 9 |
| Descripción | Get All Producers JPA |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

http://localhost:8080/producers/findAll?method=false

GET

http://localhost:8080/producers/findAll?method=false

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

| <input checked="" type="checkbox"/> Key | Value | Description | ... | Bulk Edit |
|--|-------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> method | false | | | |
| Key | Value | Description | | |

Revisamos la salida:

Body

Cookies

Headers (5)

Test Results (1/1)

200 OK

77 ms

127.91 KB

Save as example

Pretty

Raw

Preview

Visualize

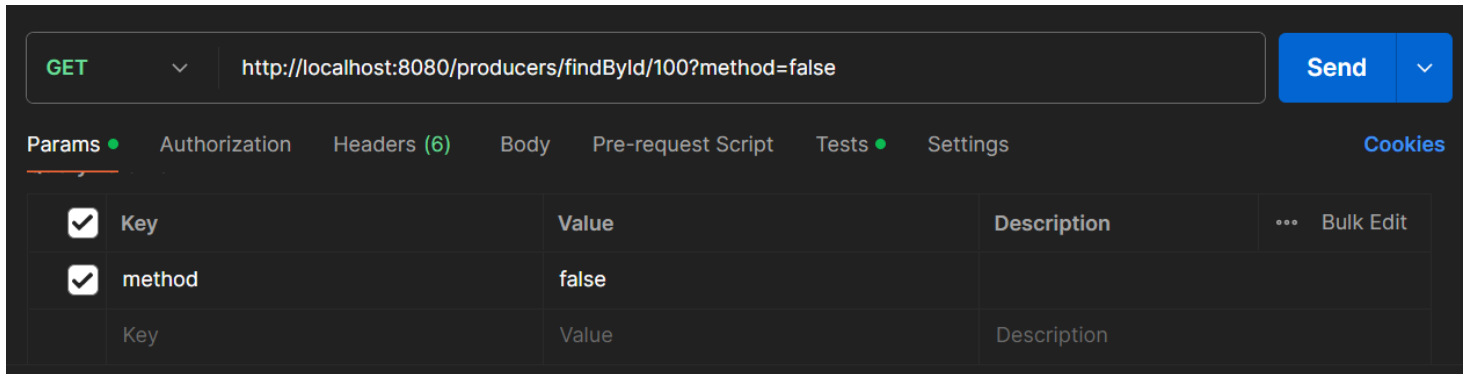
JSON

```
[
  {
    "id": 3,
    "name": "bibendum imperdiet nullam orci pede venenatis non sodales sed tincidunt eu felis fusce posuere felis sed lacus",
    "debut": 2023
  },
  {
    "id": 4,
    "name": "pharetra magna ac consequat metus sapien ut nunc vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere",
    "debut": 2023
  }
]
```

| | |
|---------------------------|---|
| # Paso: | 10 |
| Descripción | Get A Producer JPA |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

```
http://localhost:8080/producers/findById/100?method=false
```



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

- Method:** GET
- URL:** http://localhost:8080/producers/findById/100?method=false
- Buttons:** Send, Bulk Edit
- Tabs:** Params, Authorization, Headers (6), Body, Pre-request Script, Tests, Settings, Cookies
- Params Table:**

| Key | Value | Description |
|--------|-------|-------------|
| method | false | |

Revisamos la salida:



The screenshot shows the response body in a REST client interface with the following details:

- Tabs:** Body, Cookies, Headers (5), Test Results (1/1)
- Status:** 200 OK, 9 ms, 324 B
- Buttons:** Save as example
- Body Format:** JSON
- Response Body (Pretty):**

```
{
  "id": 100,
  "name": "tincidunt ante vel ipsum praesent blandit lacinia erat vestibulum sed magna at nunc commodo placerat praesent blandit nam nulla",
  "debut": 2023
}
```

| | |
|--------------------|--|
| # Paso: | 11 |
| Descripción | Get A Producer JPA Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 404 Not Found. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

```
http://localhost:8080/producers/findById/2000?method=false
```

GET

http://localhost:8080/producers/findById/2000?method=false

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

| <input checked="" type="checkbox"/> | Key | Value | Description | ... | Bulk Edit |
|-------------------------------------|--------|-------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | method | false | | | |
| | Key | Value | Description | | |

Revisamos la salida:

Body

Cookies

Headers (5)

Test Results (1/1)

404 Not Found

6 ms

232 B

Save as example

Pretty

Raw

Preview

Visualize

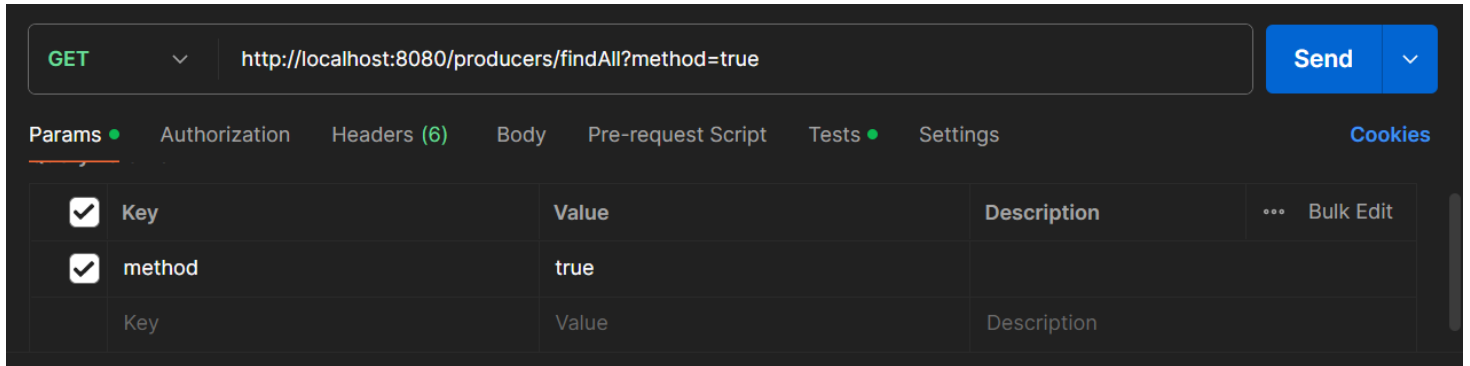
JSON

```
1 {
2   "date": "09/05/2024 15:56:23",
3   "message": "Producer not found"
4 }
```

| | |
|---------------------------|---|
| # Paso: | 12 |
| Descripción | Get All Producers Criteria |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

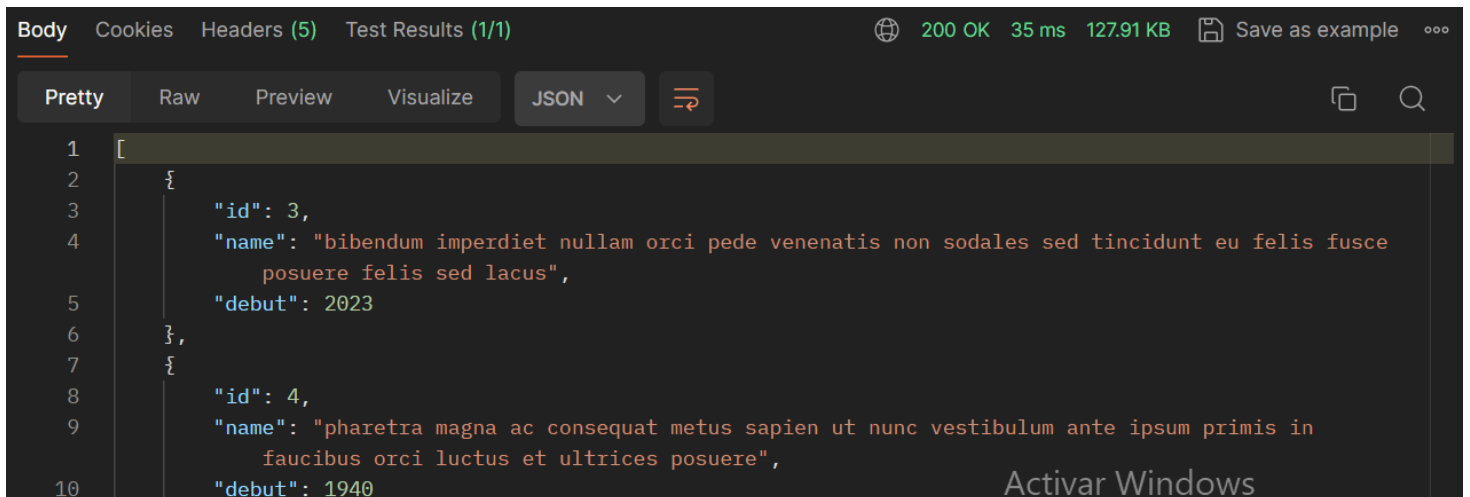
Realizamos una petición REST de tipo GET la siguiente URL:

http://localhost:8080/producers/findAll?method=true



The screenshot shows the Postman interface with a GET request to `http://localhost:8080/producers/findAll?method=true`. The 'Params' tab is active, showing a single parameter `method` with the value `true`.

Revisamos la salida:



The screenshot shows the Postman interface with the 'Body' tab active, displaying the JSON response in 'Pretty' format. The response is a JSON array with two objects, each containing 'id', 'name', and 'debut' fields.

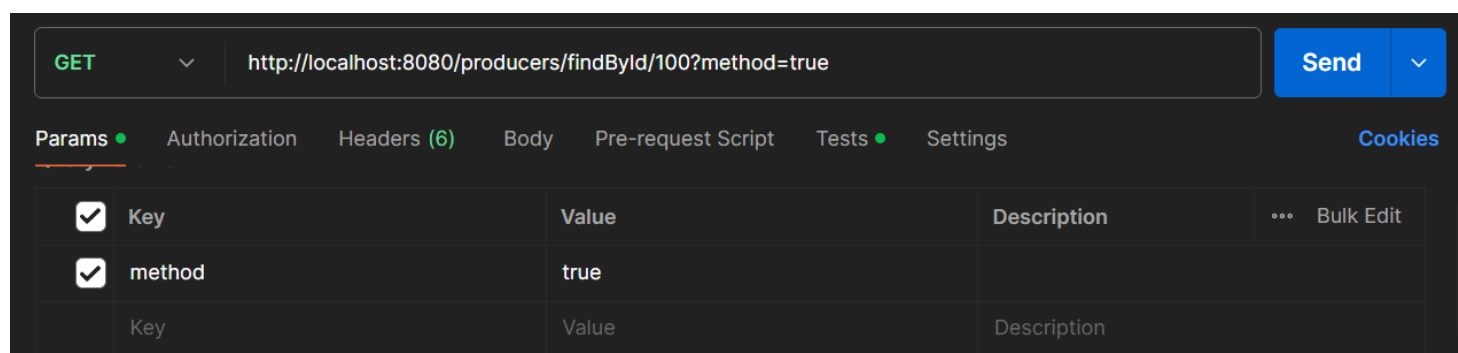
```

1 [
2   {
3     "id": 3,
4     "name": "bibendum imperdiet nullam orci pede venenatis non sodales sed tincidunt eu felis fusce
        posuere felis sed lacus",
5     "debut": 2023
6   },
7   {
8     "id": 4,
9     "name": "pharetra magna ac consequat metus sapien ut nunc vestibulum ante ipsum primis in
        faucibus orci luctus et ultrices posuere",
10    "debut": 1940
  }
]
```

| | |
|---------------------------|---|
| # Paso: | 13 |
| Descripción | Get A Producer Criteria |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

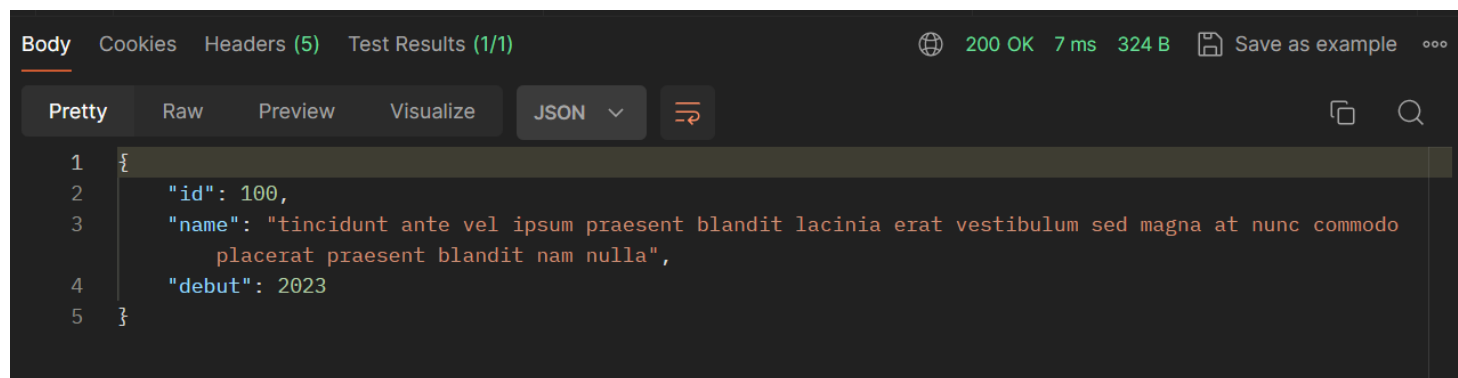
Realizamos una petición REST de tipo GET la siguiente URL:

`http://localhost:8080/producers/findById/100?method=true`



| Key | Value | Description |
|--|-------|-------------|
| <input checked="" type="checkbox"/> method | true | |
| <input type="checkbox"/> Key | Value | Description |

Revisamos la salida:



```

1 {
2   "id": 100,
3   "name": "tincidunt ante vel ipsum praesent blandit lacinia erat vestibulum sed magna at nunc commodo
         placerat praesent blandit nam nulla",
4   "debut": 2023
5 }
```

| | |
|--------------------|--|
| # Paso: | 14 |
| Descripción | Get A Producer Criteria Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 404 Not Found. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

```
http://localhost:8080/producers/findById/2000?method=true
```

GET

http://localhost:8080/producers/findById/2000?method=true

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

| <input checked="" type="checkbox"/> | Key | Value | Description | ... | Bulk Edit |
|-------------------------------------|--------|-------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | method | true | | | |
| | Key | Value | Description | | |

Revisamos la salida:

Body

Cookies

Headers (5)

Test Results (1/1)

404 Not Found

6 ms

232 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "date": "09/05/2024 15:57:33",
3   "message": "Producer not found"
4 }
```

| | |
|---------------------------|---|
| # Paso: | 15 |
| Descripción | Delete A Producer JPA |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 204 No Content. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo DELETE la siguiente URL:

```
http://localhost:8080/producers/deleteById?method=false&id=1
```

DELETE

Send

Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies

| | | | |
|-------------------------------------|--------|-------|-------------|
| <input checked="" type="checkbox"/> | method | false | |
| <input checked="" type="checkbox"/> | id | 1 | |
| | Key | Value | Description |

Revisamos la salida:

DELETE
ProjectFilms / Delete A Film JPA
×

Response
Headers
Request

204 15 ms 144 B

| | |
|--------------------|--|
| # Paso: | 16 |
| Descripción | Delete A Producer JPA Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 404 Not Found. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo DELETE la siguiente URL:

```
http://localhost:8080/producers/deleteById?method=false&id=2000
```

The screenshot shows a REST client interface with a dark theme. At the top, the method is set to 'DELETE' and the URL is 'http://localhost:8080/producers/deleteById?method=false&id=2000'. A 'Send' button is on the right. Below the URL bar, there are tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', 'Settings', and 'Cookies'. The 'Params' tab is active, showing a table with two parameters: 'method' with value 'false' and 'id' with value '2000'. Below the table is a header with 'Key', 'Value', and 'Description' columns. At the bottom, there is a 'Response' section with a downward arrow.

Revisamos la salida:

The screenshot shows the response body of the REST client. The status is '404 Not Found' with a response time of '16 ms' and a size of '232 B'. There is a 'Save as example' button. The response is displayed in 'Pretty' format as JSON. The JSON content is:

```
{  "date": "09/05/2024 15:58:22",  "message": "Producer not found"}
```

| | |
|---------------------------|---|
| # Paso: | 17 |
| Descripción | Delete A Producer Criteria |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 204 No Content. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo DELETE la siguiente URL:

http://localhost:8080/producers/deleteById?method=true&id=2

DELETE

http://localhost:8080/producers/deleteById?method=true&id=2

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

| | | | |
|-------------------------------------|--------|-------|-------------|
| <input checked="" type="checkbox"/> | method | true | |
| <input checked="" type="checkbox"/> | id | 2 | |
| | Key | Value | Description |

Revisamos la salida:

DELETE

ProjectFilms / Delete A Film Criteria

×

Response

Headers

Request

204 21 ms 144 B

| | |
|--------------------|--|
| # Paso: | 18 |
| Descripción | Delete A Producer Criteria Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 404 Not Found. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo DELETE la siguiente URL:

```
http://localhost:8080/producers/deleteById?method=true&id=2000
```

DELETE

http://localhost:8080/producers/deleteById?method=true&id=2000

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

| | | | |
|-------------------------------------|--------|-------|-------------|
| <input checked="" type="checkbox"/> | method | true | |
| <input checked="" type="checkbox"/> | id | 2000 | |
| | Key | Value | Description |

Revisamos la salida:

Body

Cookies

Headers (5)

Test Results (1/1)

404 Not Found

8 ms

232 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "date": "09/05/2024 15:58:57",
3   "message": "Producer not found"
4 }
```