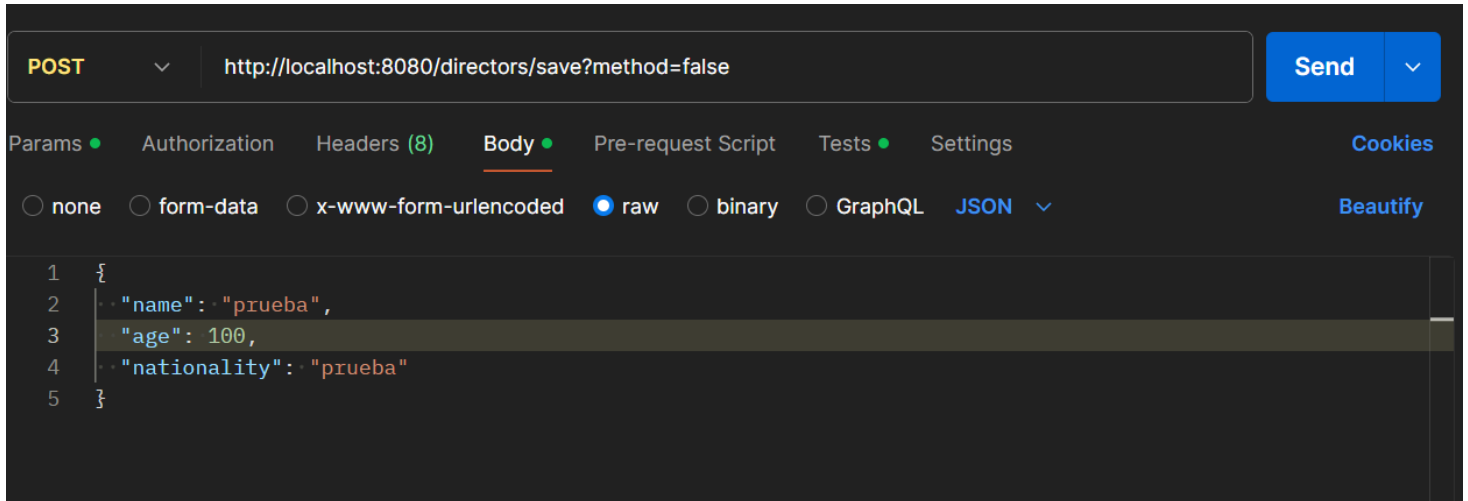


Ejecución Automática de Caso de Test	
Dominio	Proyecto
Aplicación	Ámbito Ejecución
NOMBRE DE LA APLICACIÓN Films	Offline
Descripción	Pre-requisitos
Prueba unitaria para Proyecto Spring Films - Directors	
Tester Ejecución	Fecha Ejecución
	09/05/2024

# Paso:	1
Descripción	Create Director 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 POST tipo la siguiente URL:

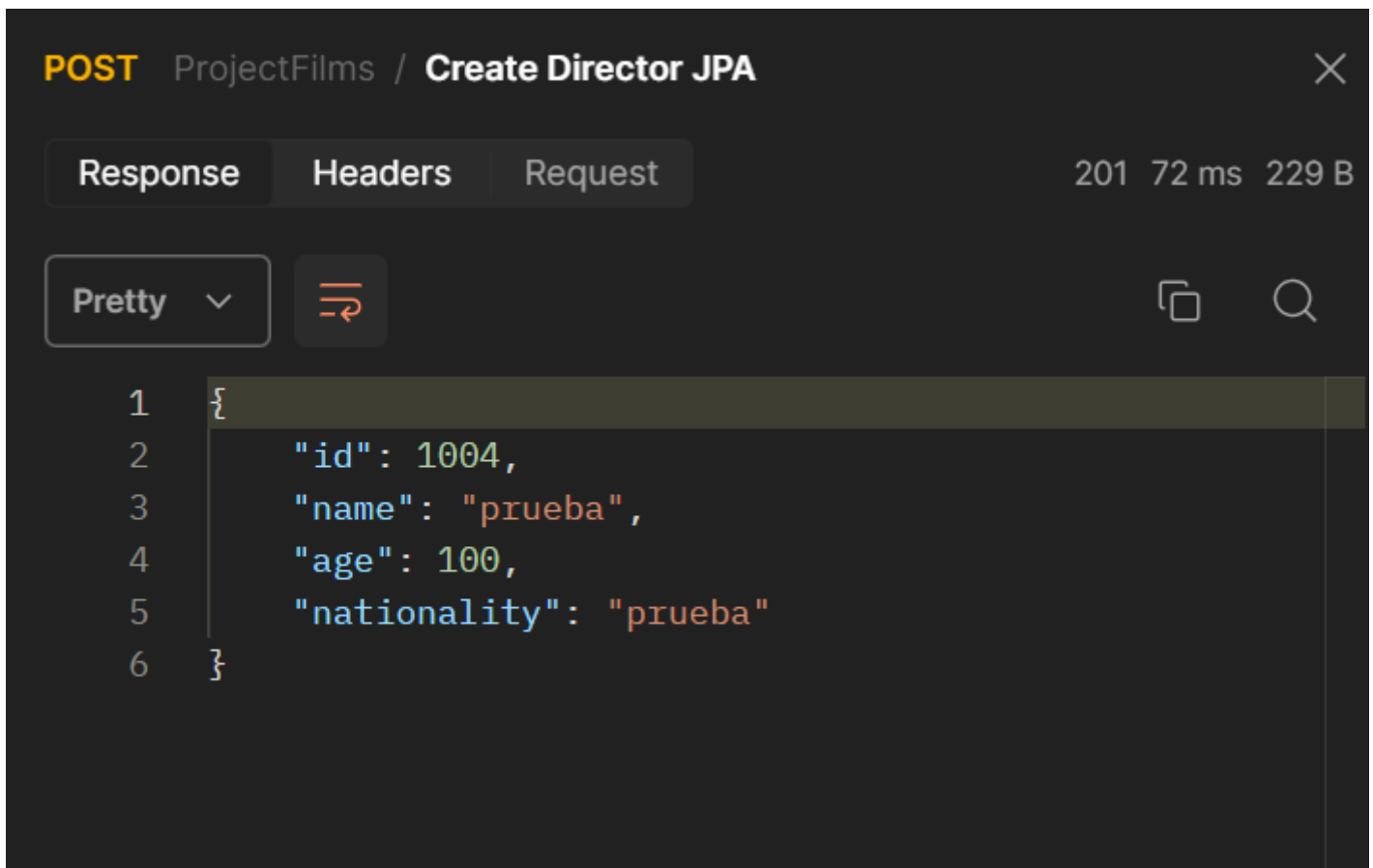
```
http://localhost:8080/directors/save?method=false
```



Con el siguiente body

```
{  
  "name": "prueba",  
  "age": 100,  
  "nationality": "prueba"  
}
```

Revisamos la salida:



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

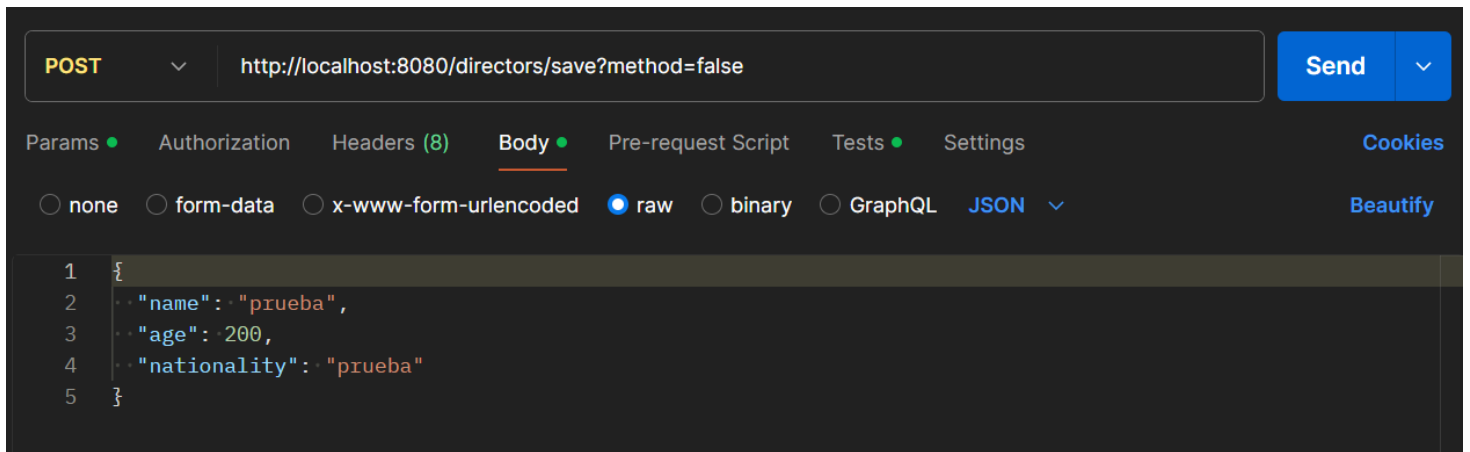
- Method:** POST
- Path:** ProjectFilms / Create Director JPA
- Response Tab:** Selected, showing a status of 201, 72 ms, and 229 B.
- Format:** Pretty (indicated by a dropdown menu).
- Response Body:**

```
1 {  
2   "id": 1004,  
3   "name": "prueba",  
4   "age": 100,  
5   "nationality": "prueba"  
6 }
```

# Paso:	2
Descripción	Create Director 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:

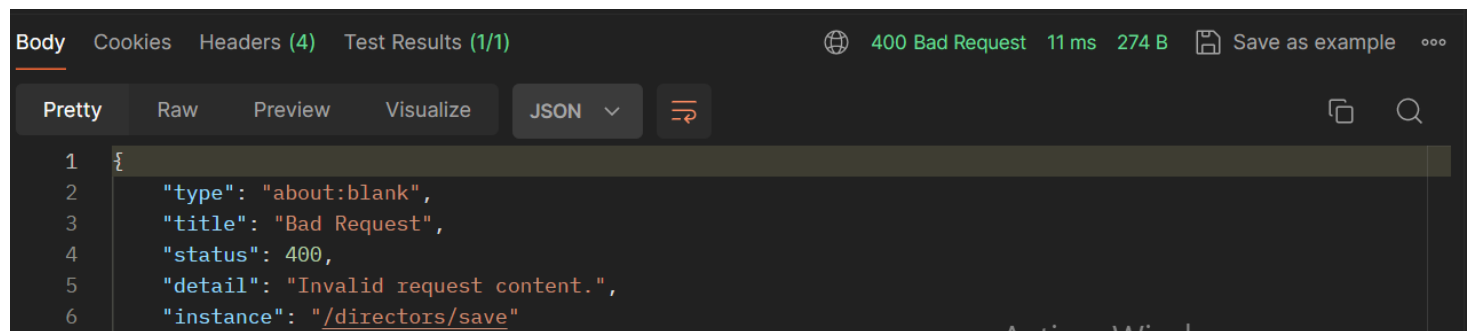
`http://localhost:8080/directors/save?method=false`



Con el siguiente body

```
{  
  "name": "prueba",  
  "age": 200,  
  "nationality": "prueba"  
}
```

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: Save as example, Copy, Search.
- Format: JSON.
- Response body (Pretty):

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

# Paso:	3
Descripción	Update Director 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/directors/update?method=false

PUT

http://localhost:8080/directors/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

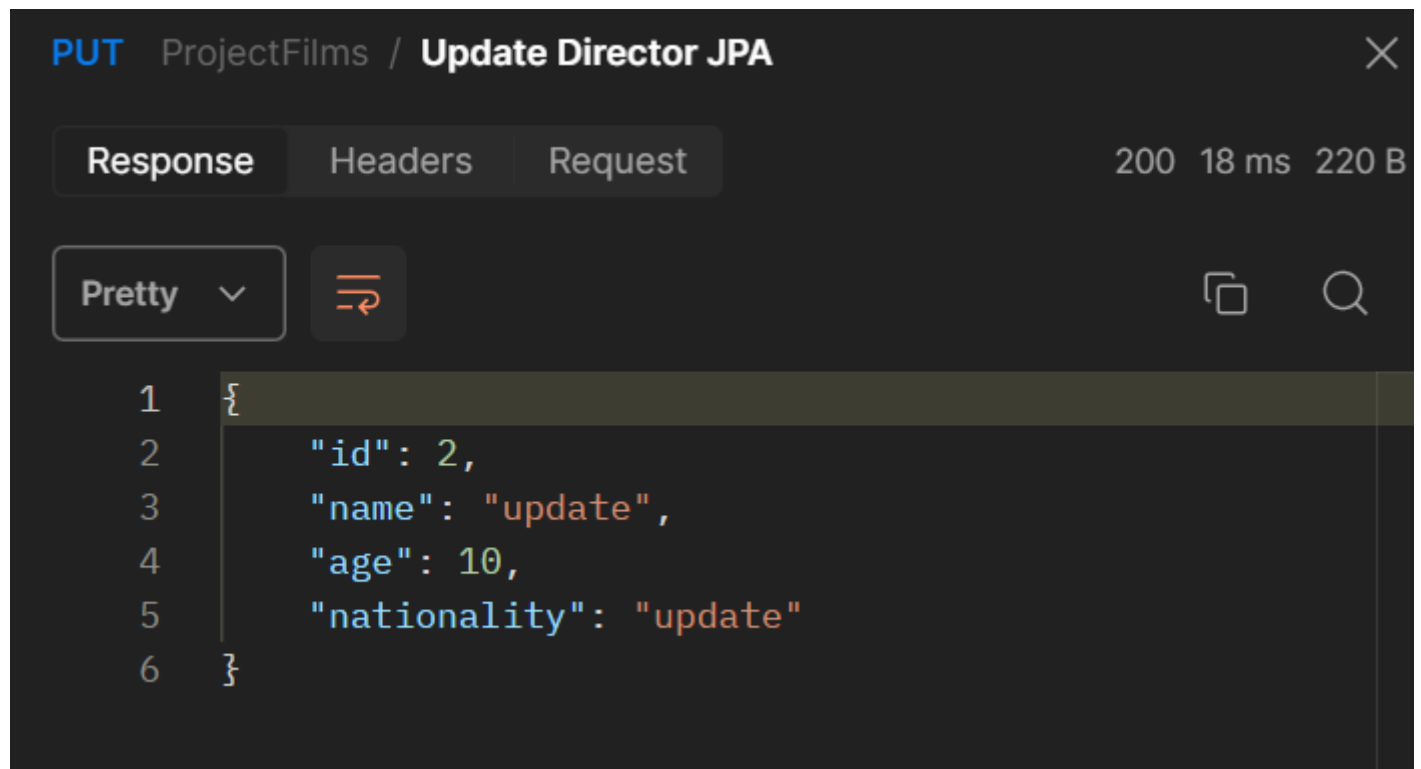
```

1 {
2   ... "id": 2,
3   ... "name": "update",
4   ... "age": 10,
5   ... "nationality": "update"
6 }
```

Con el siguiente body

```
{  
  "id": 2,  
  "name": "update",  
  "age": 10,  
  "nationality": "update"  
}
```

Revisamos la salida:



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

- Method:** PUT
- Path:** ProjectFilms / Update Director JPA
- Response Tab:** Selected, showing status 200, time 18 ms, and size 220 B.
- Format:** Pretty (indicated by a dropdown menu).
- Response Body:**

```
1 {  
2   "id": 2,  
3   "name": "update",  
4   "age": 10,  
5   "nationality": "update"  
6 }
```

# Paso:	4
Descripción	Update Director 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/directors/update?method=false

PUT

http://localhost:8080/directors/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

```

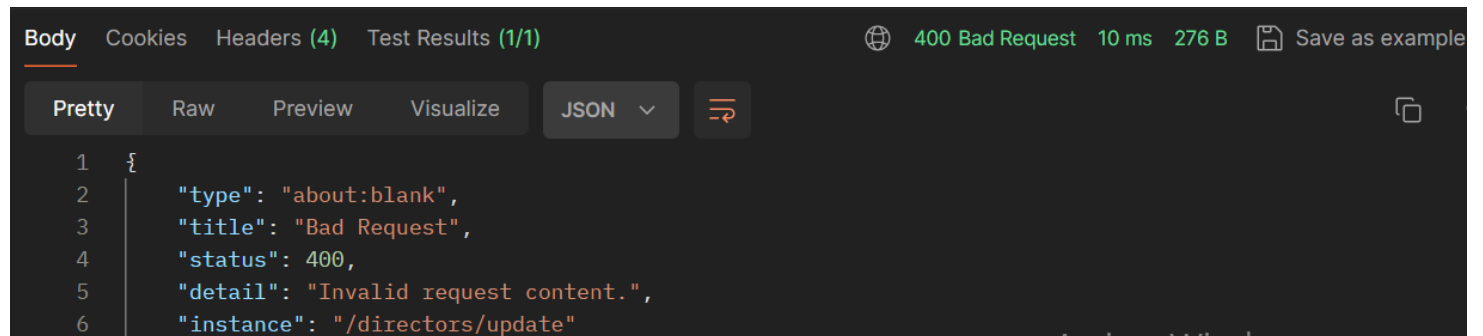
1  {
2    ... "id": 2,
3    "name": "update",
4    ... "age": 200,
5    ... "nationality": "update"
6  }

```


Con el siguiente body

```
{  
  "id": 2,  
  "name": "update",  
  "age": 200,  
  "nationality": "update"  
}
```

Revisamos la salida:



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

- Body** tab selected.
- Headers (4)** and **Test Results (1/1)** tabs are visible.
- Status:** 400 Bad Request, 10 ms, 276 B.
- Save as example** button.
- JSON** format selected.
- Response Body:**

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

# Paso:	5
Descripción	Create Director 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/directors/save?method=true

POST

▼

http://localhost:8080/directors/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

· "age": 100,

4

· "nationality": "prueba"

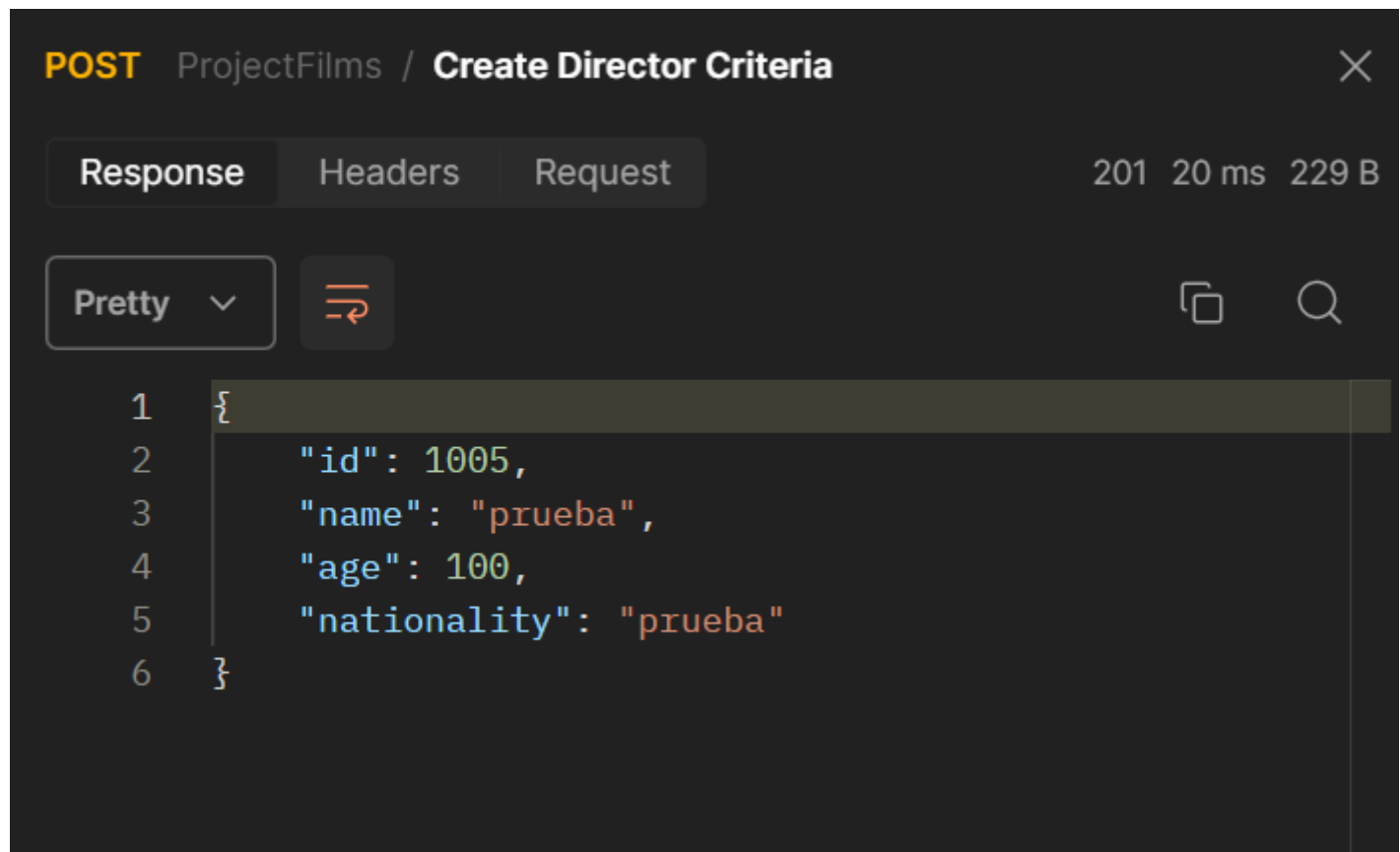
5

}

Con el siguiente body

```
{  
  "name": "prueba",  
  "age": 100,  
  "nationality": "prueba"  
}
```

Revisamos la salida:



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

- Method:** POST
- Path:** ProjectFilms / Create Director Criteria
- Response Tab:** Selected, showing status 201, time 20 ms, and size 229 B.
- Format:** Pretty (JSON)
- Response Body:**

```
1 {  
2   "id": 1005,  
3   "name": "prueba",  
4   "age": 100,  
5   "nationality": "prueba"  
6 }
```

# Paso:	6
Descripción	Create Director 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/directors/save?method=true

POST

▼

http://localhost:8080/directors/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 " age": 200,

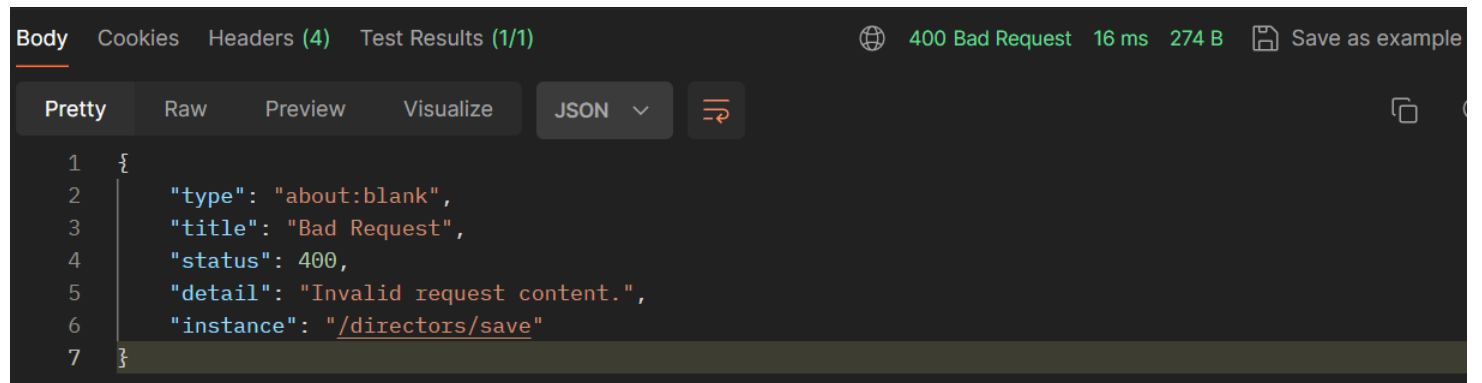
4 " nationality": "prueba"

5 }

Con el siguiente body

```
{  
  "name": "prueba",  
  "age": 200,  
  "nationality": "prueba"  
}
```

Revisamos la salida:



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

- Body** tab selected.
- Response status: **400 Bad Request** (16 ms, 274 B).
- Buttons: **Pretty**, **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": "/directors/save"  
7 }
```

# Paso:	7
Descripción	Update Director 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/directors/update?method=true

PUT

http://localhost:8080/directors/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

```

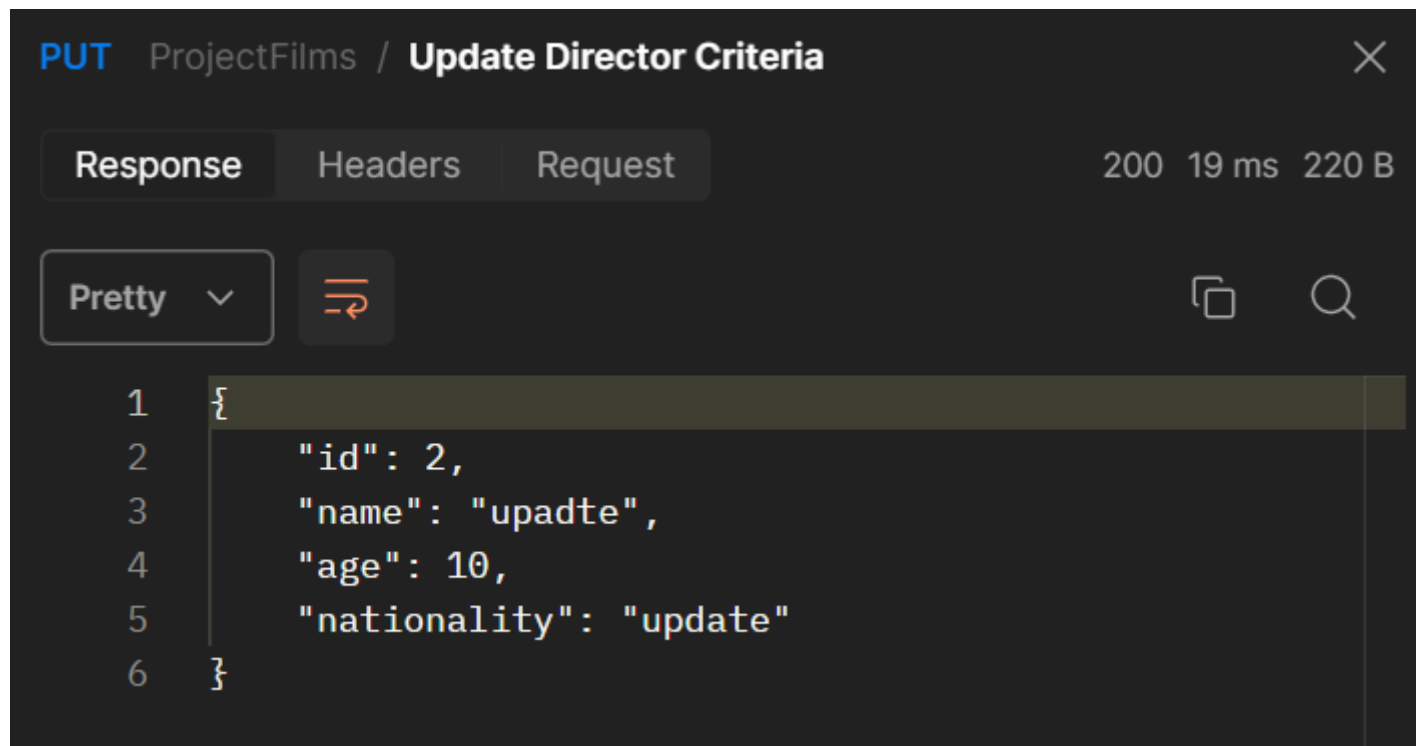
1 {
2   ... "id": 2,
3   ... "name": "upadte",
4   ... "age": 10,
5   ... "nationality": "update"
6 }

```

Con el siguiente body

```
{  
  "id": 2,  
  "name": "upadte",  
  "age": 10,  
  "nationality": "update"  
}
```

Revisamos la salida:



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

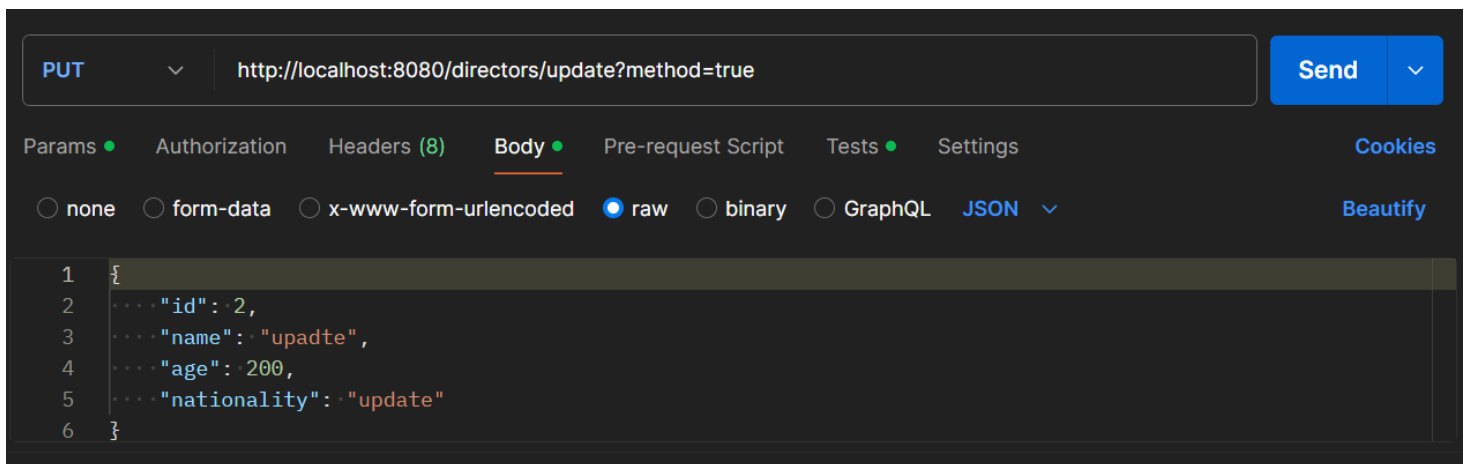
- Method:** PUT
- Path:** ProjectFilms / Update Director Criteria
- Response Tab:** Selected, showing status 200, time 19 ms, and size 220 B.
- Format:** Pretty (indicated by a dropdown menu).
- Response Body:**

```
1 {  
2   "id": 2,  
3   "name": "upadte",  
4   "age": 10,  
5   "nationality": "update"  
6 }
```

# Paso:	8
Descripción	Update Director 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/directors/update?method=true`



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

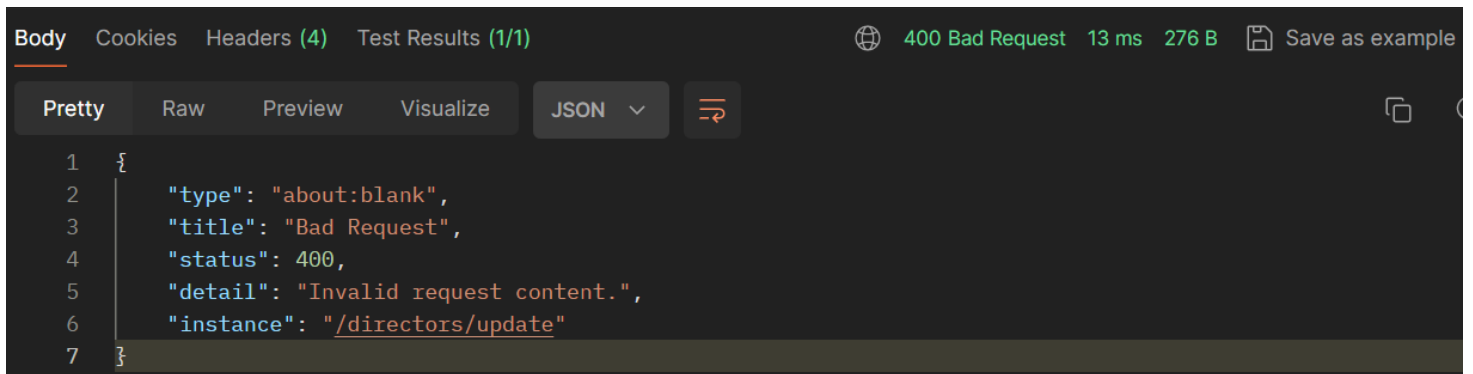
- Method:** PUT
- URL:** `http://localhost:8080/directors/update?method=true`
- Body Type:** raw (selected)
- Body Content:**

```
1 {  
2   "id": 2,  
3   "name": "upadte",  
4   "age": 200,  
5   "nationality": "update"  
6 }
```


Con el siguiente body

```
{
  "id": 2,
  "name": "upadte",
  "age": 200,
  "nationality": "update"
}
```

Revisamos la salida:



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

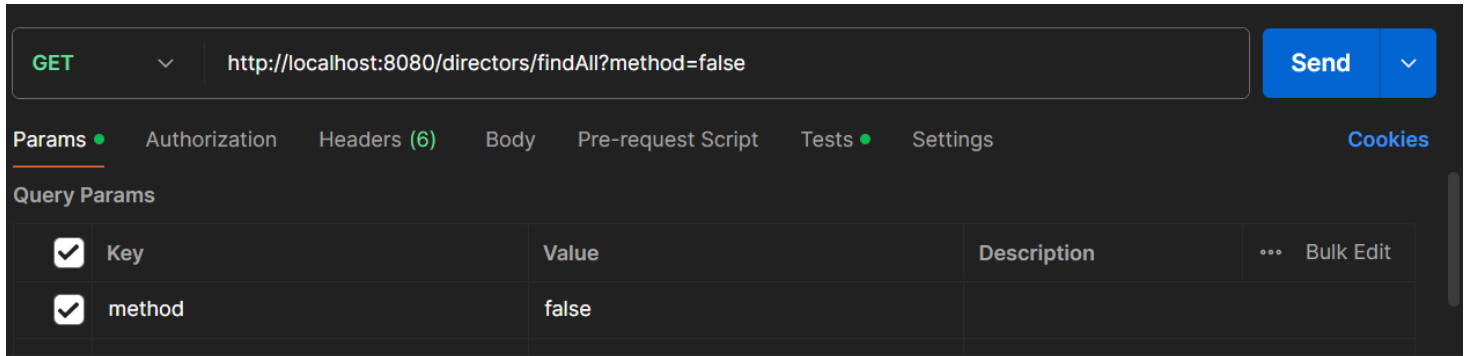
- Body** tab selected.
- Response status: **400 Bad Request** (13 ms, 276 B).
- Buttons: **Pretty**, **Raw**, **Preview**, **Visualize**, **JSON** (dropdown), and a refresh icon.
- Response body (JSON):

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

# Paso:	9
Descripción	Get All Directors 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/directors/findAll?method=false`



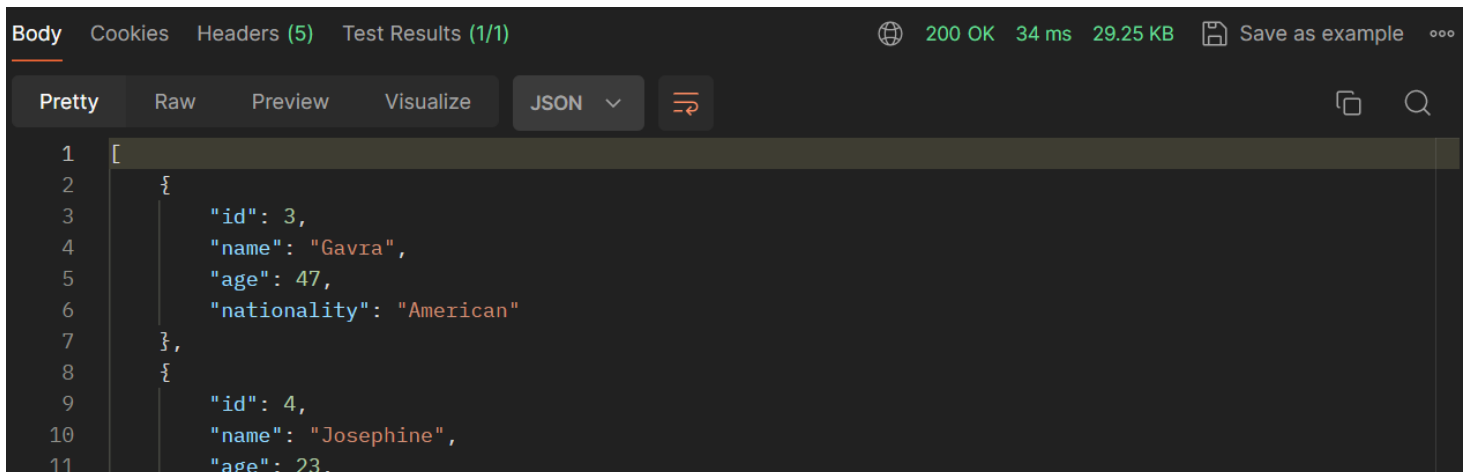
GET `http://localhost:8080/directors/findAll?method=false` Send

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			

Revisamos la salida:



Body Cookies Headers (5) Test Results (1/1) 200 OK 34 ms 29.25 KB Save as example

Pretty Raw Preview Visualize JSON

```

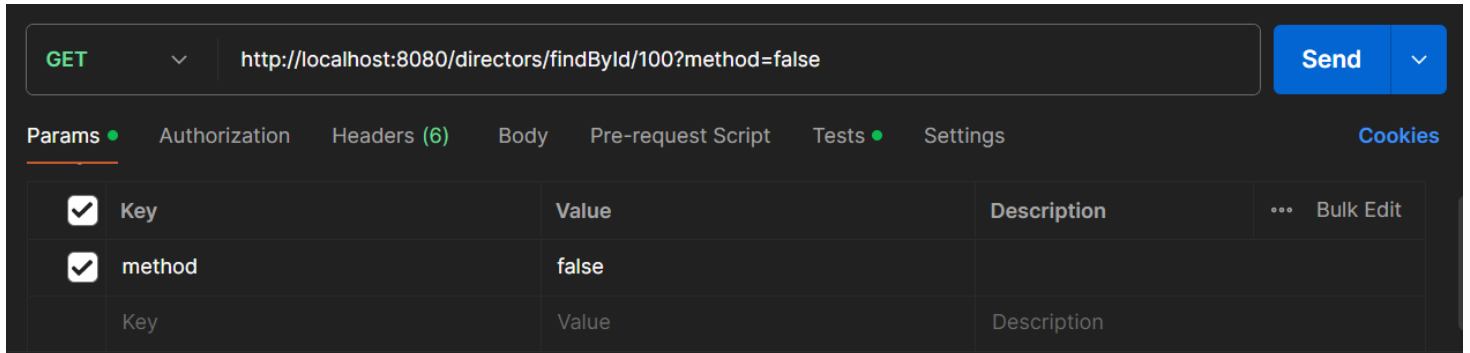
1  [
2    {
3      "id": 3,
4      "name": "Gavra",
5      "age": 47,
6      "nationality": "American"
7    },
8    {
9      "id": 4,
10     "name": "Josephine",
11     "age": 23,

```

# Paso:	10
Descripción	Get A Director 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 la GET siguiente URL:

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

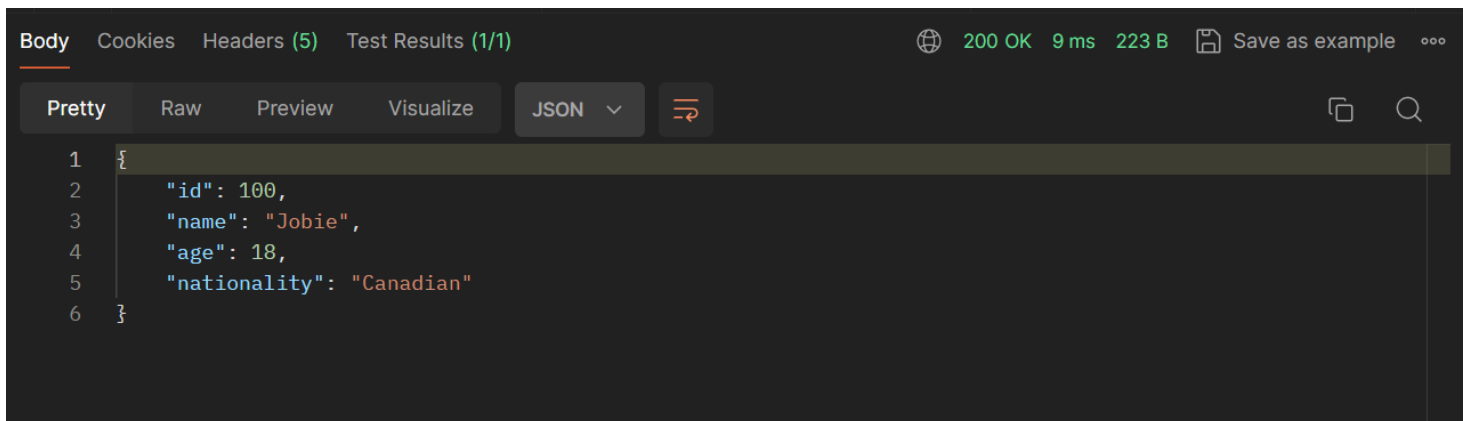


The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:8080/directors/findById/100?method=false
- Params:**

Key	Value	Description
method	false	

Revisamos la salida:



The screenshot shows the response body of the GET request, which is a JSON object:

```
{
  "id": 100,
  "name": "Jobie",
  "age": 18,
  "nationality": "Canadian"
}
```

# Paso:	11
Descripción	Get A Director 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/directors/findById/2000?method=false
```

GET

http://localhost:8080/directors/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

8 ms

230 B

Save as example

Pretty

Raw

Preview

Visualize

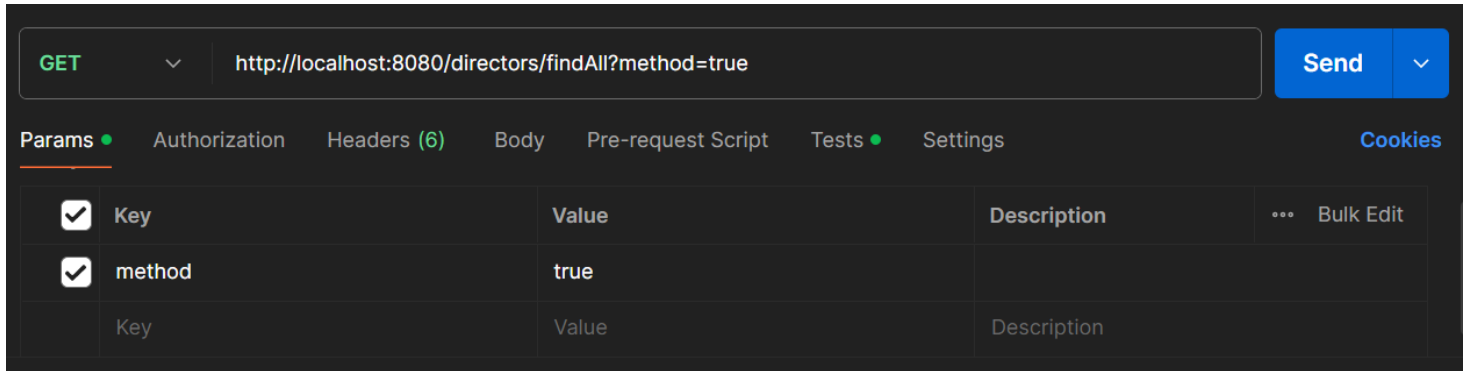
JSON

```
1 {
2   "date": "09/05/2024 15:41:45",
3   "message": "Person not found"
4 }
```

# Paso:	12
Descripción	Get All Directors 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/directors/findAll?method=true`

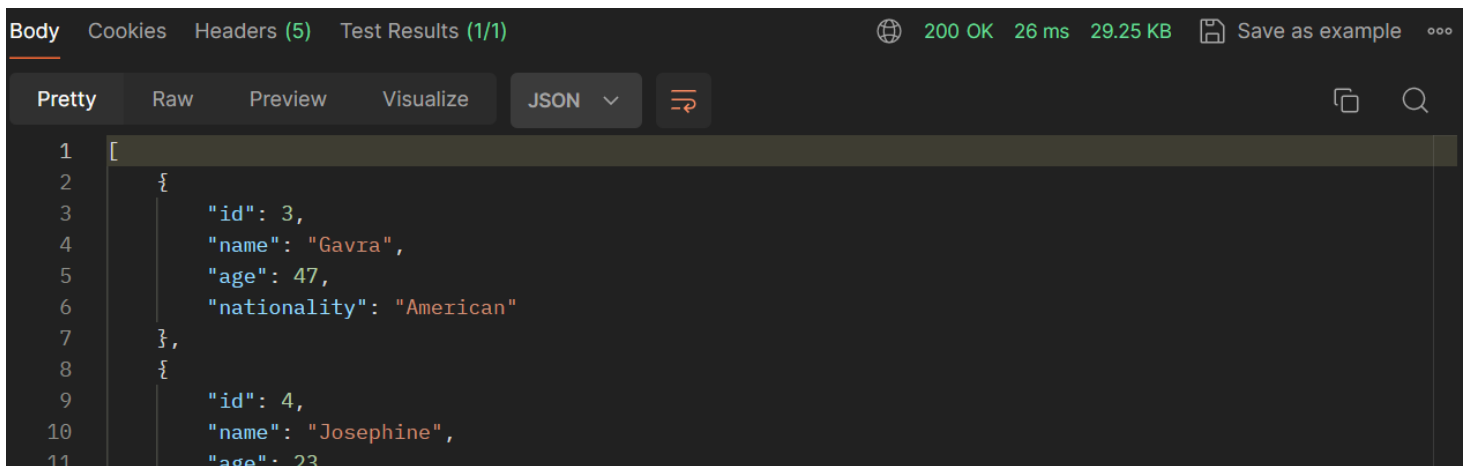


GET `http://localhost:8080/directors/findAll?method=true` Send

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

Key	Value	Description
method	true	

Revisamos la salida:



Body Cookies Headers (5) Test Results (1/1) 200 OK 26 ms 29.25 KB Save as example

Pretty Raw Preview Visualize JSON

```

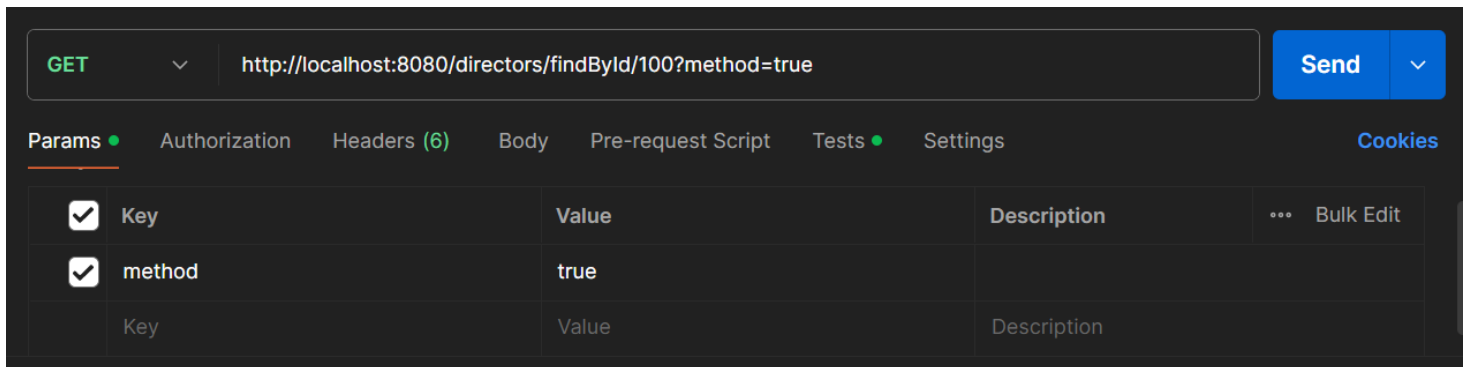
1 [
2   {
3     "id": 3,
4     "name": "Gavra",
5     "age": 47,
6     "nationality": "American"
7   },
8   {
9     "id": 4,
10    "name": "Josephine",
11    "age": 23

```

# Paso:	13
Descripción	Get A Director 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/directors/findById/100?method=true
```

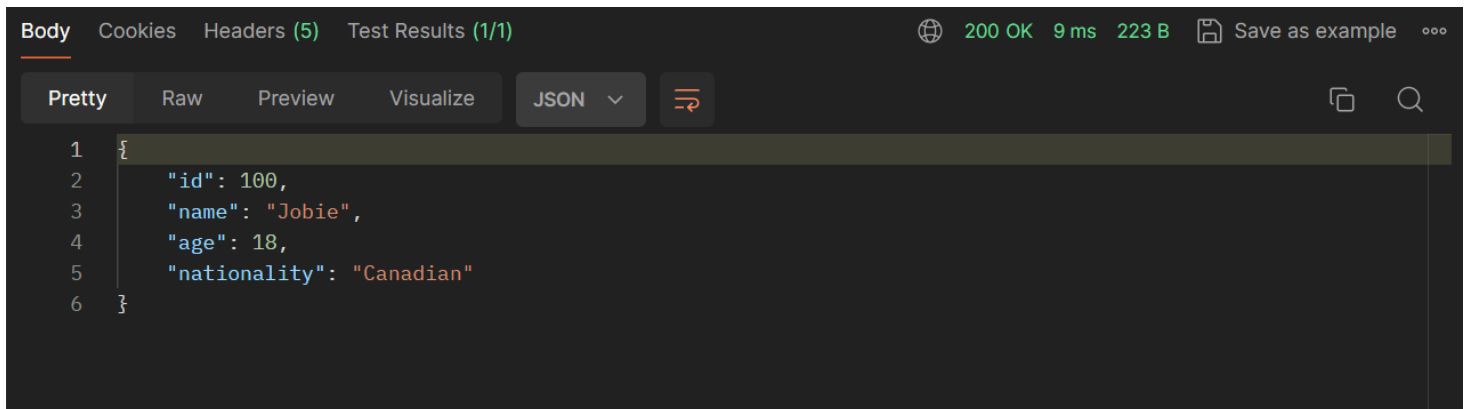


The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:8080/directors/findById/100?method=true
- Params:**

Key	Value	Description
method	true	

Revisamos la salida:



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

- Status:** 200 OK
- Time:** 9 ms
- Size:** 223 B
- Body:**

```
{
  "id": 100,
  "name": "Jobie",
  "age": 18,
  "nationality": "Canadian"
}
```

# Paso:	14
Descripción	Get A Director 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/directors/findById/2000?method=true

GET

http://localhost:8080/directors/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

11 ms

230 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "date": "09/05/2024 15:44:34",
3   "message": "Person not found"
4 }
```

# Paso:	15
Descripción	Delete A Director 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/directors/deleteById?method=false&id=1
```

DELETE

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

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:

1

DELETE

ProjectFilms / Delete A Director JPA

×

Response

Headers

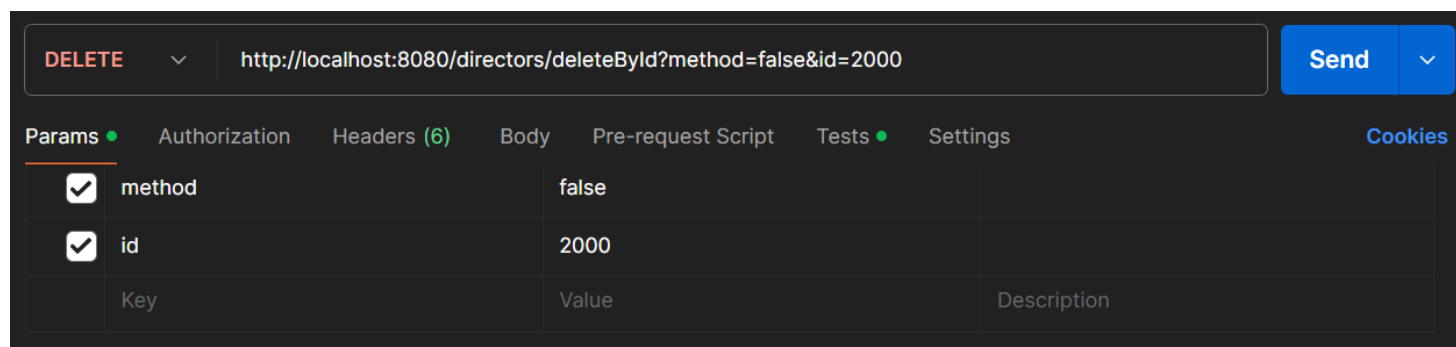
Request

204 12 ms 144 B

# Paso:	16
Descripción	Delete A Director 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:

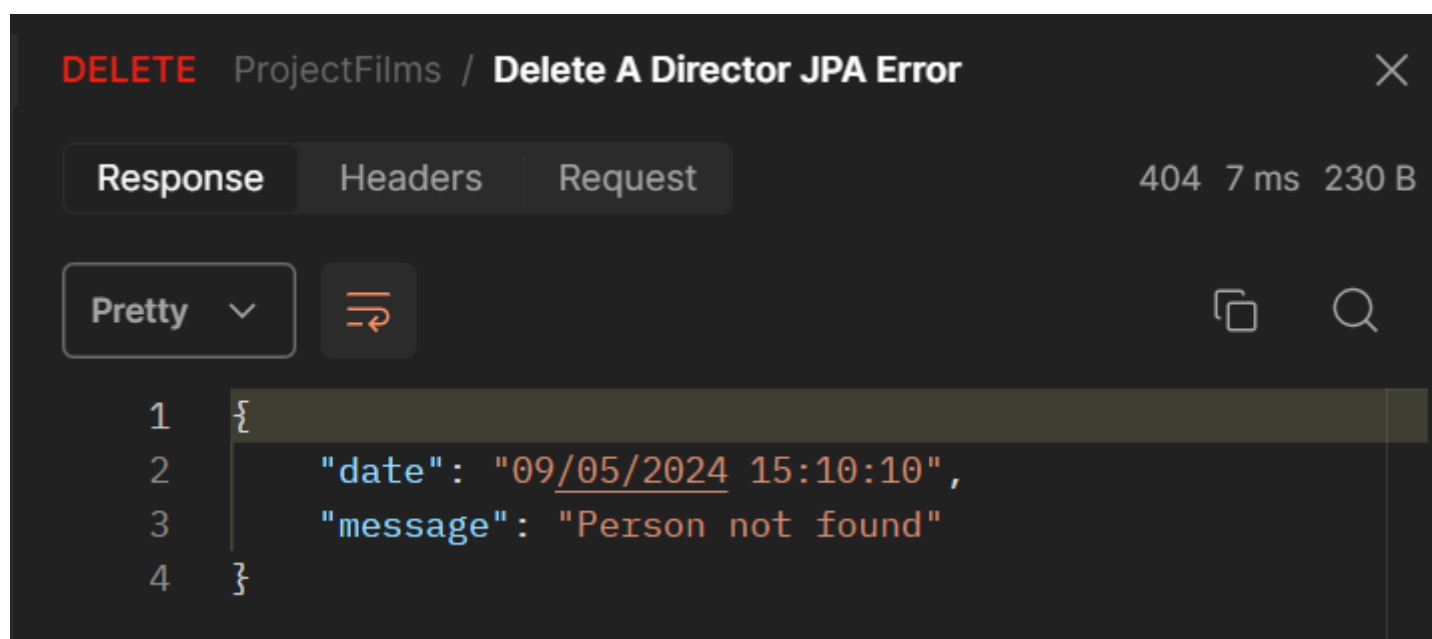
```
http://localhost:8080/directors/deleteById?method=false&id=2000
```



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

- Method:** DELETE
- URL:** http://localhost:8080/directors/deleteById?method=false&id=2000
- Params:**
 - method: false
 - id: 2000
- Headers:** 6 headers are listed, but none are visible in the screenshot.
- Body:** Empty
- Pre-request Script:** Empty
- Tests:** 0 tests are listed.
- Settings:** Default settings are shown.
- Cookies:** No cookies are present.

Revisamos la salida:



The screenshot shows the response of a DELETE request in a REST client interface. The response is as follows:

```
{
  "date": "09/05/2024 15:10:10",
  "message": "Person not found"
}
```

The response status is 404, with a response time of 7 ms and a response size of 230 B.

# Paso:	17
Descripción	Delete A Director 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/directors/deleteById?method=true&id=2`

DELETE

▼

http://localhost:8080/directors/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 Director Criteria

×

Response

Headers

Request

204 28 ms 144 B

# Paso:	18
Descripción	Delete A Director 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/directors/deleteById?method=true&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/directors/deleteById?method=true&id=2000'. 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 'true' and 'id' with value '2000'. A 'Send' button is located to the right of the URL bar.

Key	Value	Description
method	true	
id	2000	

Revisamos la salida:

The screenshot shows the response of the DELETE request in a REST client interface. The title bar indicates the method is 'DELETE' and the path is 'ProjectFilms / Delete A Director Criteria Error'. Below the title bar, there are tabs for 'Response', 'Headers', and 'Request'. The 'Response' tab is active, showing the response status '404', time '10 ms', and size '230 B'. The response body is displayed in a code editor with a 'Pretty' button and a search icon. The response is a JSON object with a 'date' field and a 'message' field.

```
1 {  
2   "date": "09/05/2024 15:10:10",  
3   "message": "Person not found"  
4 }
```