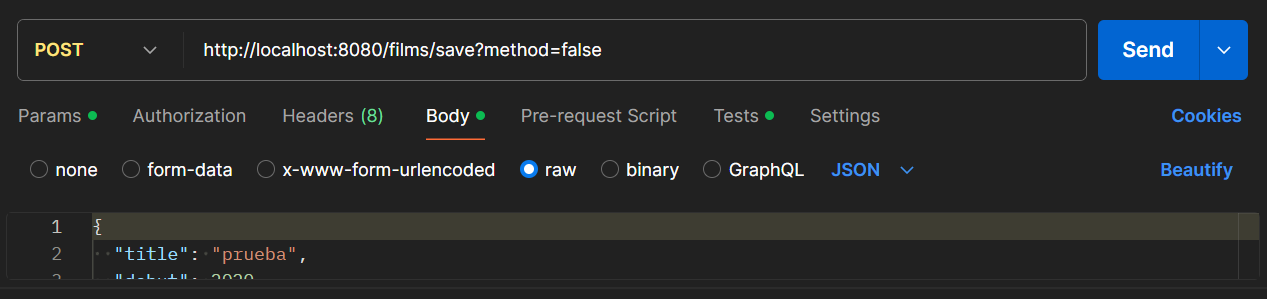
|  |  |
| --- | --- |
| Ejecución Automática de Caso de Test | |
| Dominio | **Proyecto** |
|  |  |
| Aplicación | **Ámbito Ejecución** |
| NOMBRE DE LA APLICACIÓN Films | Offline |
| Descripción | **Pre-requisitos** |
| Prueba unitaria para Proyecto Spring Films - Films |  |
| Tester Ejecución | **Fecha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

|  |  |
| --- | --- |
| # Paso: | 1 |
| Descripción | Create Film 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/films/save?method=false |



Con el siguiente body

{

  "title": "prueba",

  "debut": 2020,

  "director": {

    "id": 2,

    "name": "string",

    "age": 100,

    "nationality": "string"

  },

  "producer": {

    "id": 2,

    "name": "string",

    "debut": 2024

  },

  "actors": [

    {

      "id": 500,

      "name": "string",

      "age": 100,

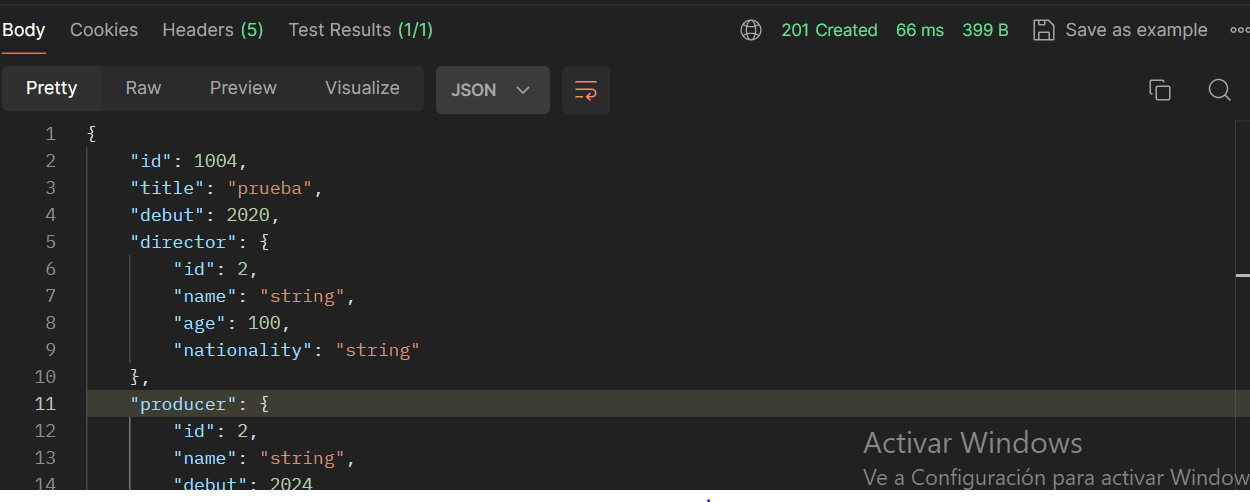
      "nationality": "string"

    }

  ]

}

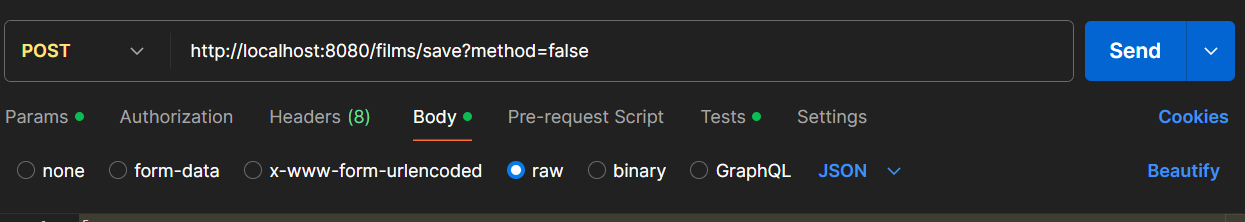
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 2 |
| Descripción | Create Film 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/films/save?method=false |



Con el siguiente body

{

  "title": "prueba",

  "debut": 3000,

  "director": {

    "id": 2,

    "name": "string",

    "age": 100,

    "nationality": "string"

  },

  "producer": {

    "id": 2,

    "name": "string",

    "debut": 2024

  },

  "actors": [

    {

      "id": 500,

      "name": "string",

      "age": 100,

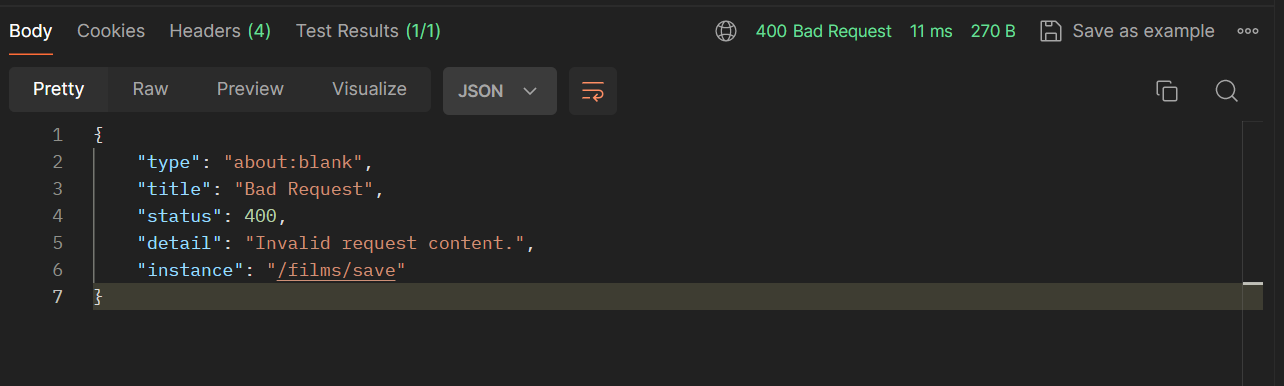
      "nationality": "string"

    }

  ]

}

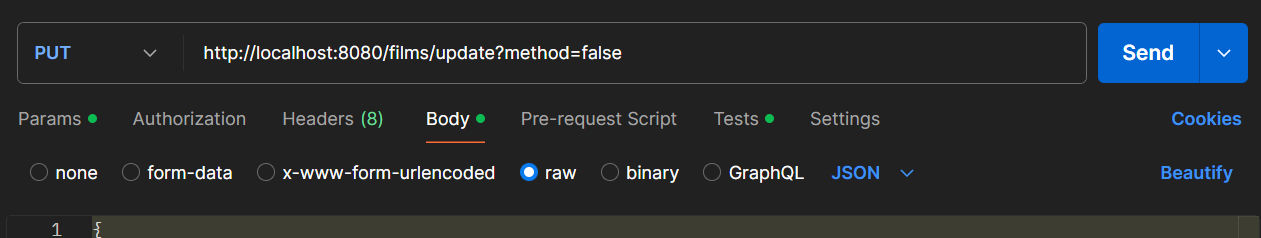
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 3 |
| Descripción | Update Film 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/films/update?method=false |



Con el siguiente body

{

    "id": 3,

    "title": "update",

    "debut": 2020,

    "director": {

        "id": 2,

        "name": "string",

        "age": 100,

        "nationality": "string"

    },

    "producer": {

        "id": 2,

        "name": "string",

        "debut": 2024

    },

    "actors": [

        {

            "id": 500,

            "name": "string",

            "age": 100,

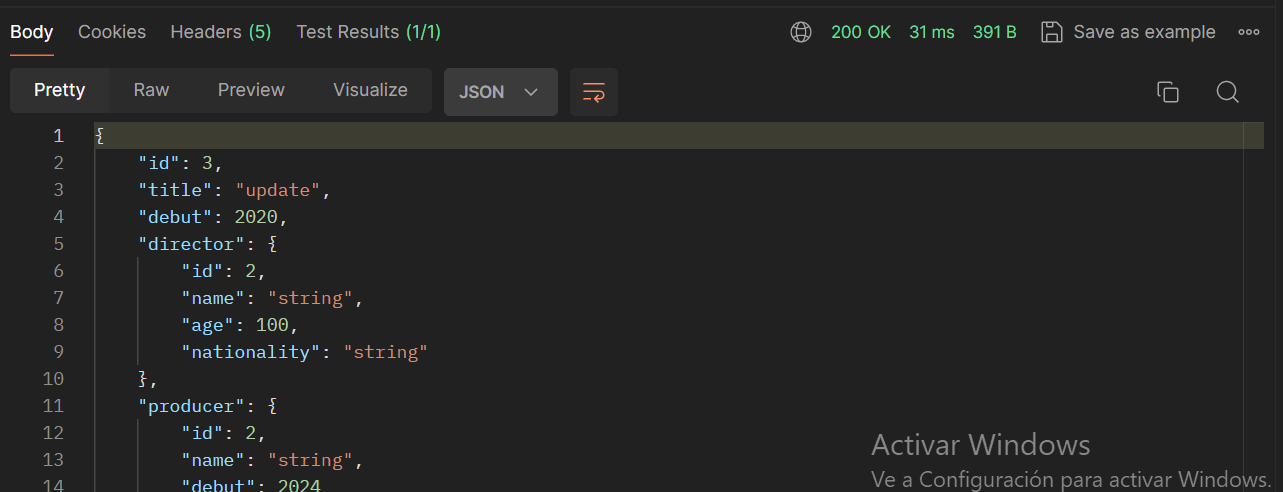
            "nationality": "string"

        }

    ]

}

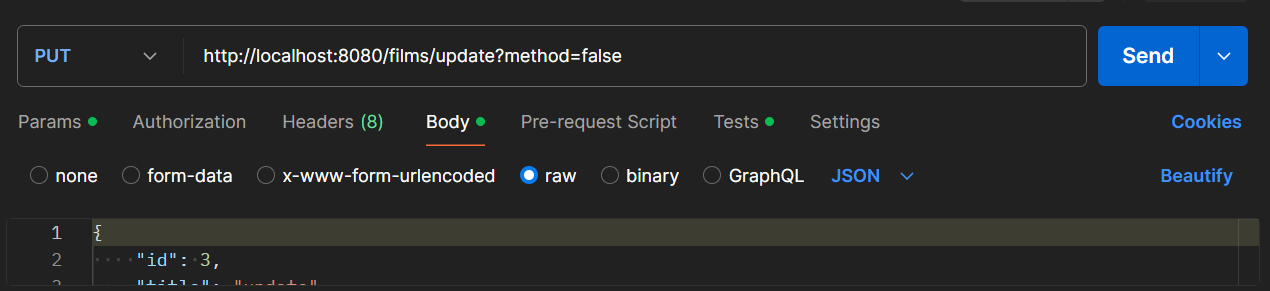
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 4 |
| Descripción | Update Film 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/films/update?method=false |



Con el siguiente body

{

    "id": 3,

    "title": "update",

    "debut": 3000,

    "director": {

        "id": 2,

        "name": "string",

        "age": 100,

        "nationality": "string"

    },

    "producer": {

        "id": 2,

        "name": "string",

        "debut": 2024

    },

    "actors": [

        {

            "id": 500,

            "name": "string",

            "age": 100,

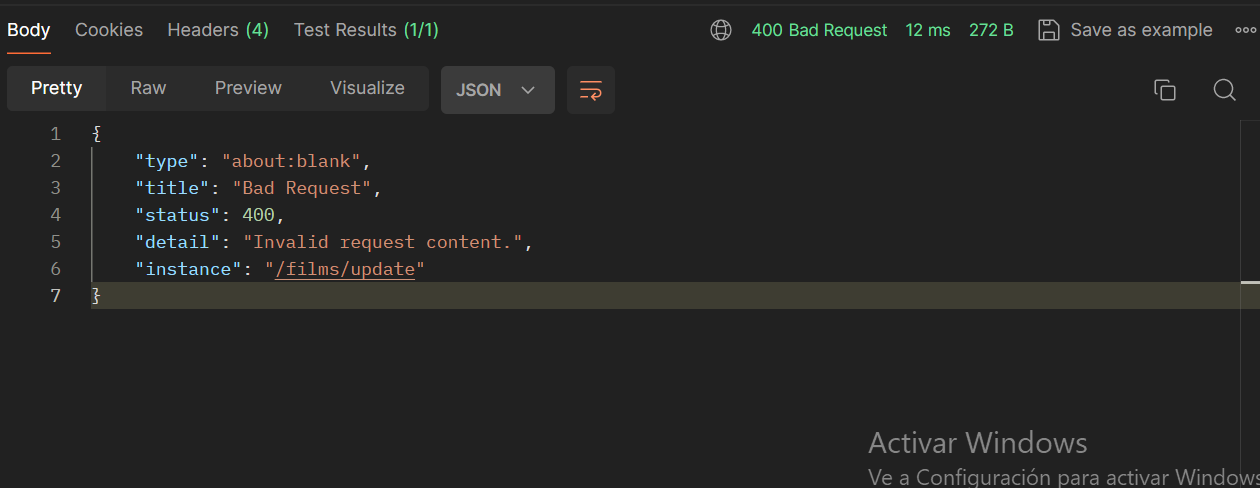
            "nationality": "string"

        }

    ]

}

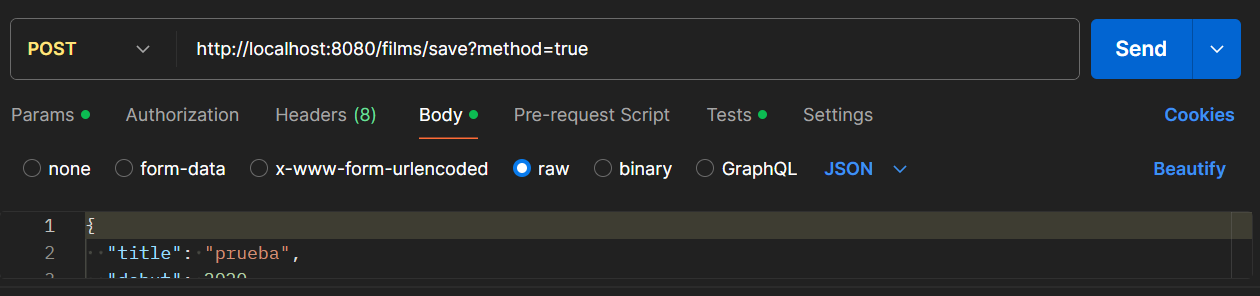
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 5 |
| Descripción | Create Film 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/films/save?method=true |



Con el siguiente body

{

  "title": "prueba",

  "debut": 2020,

  "director": {

    "id": 2,

    "name": "string",

    "age": 100,

    "nationality": "string"

  },

  "producer": {

    "id": 2,

    "name": "string",

    "debut": 2024

  },

  "actors": [

    {

      "id": 500,

      "name": "string",

      "age": 100,

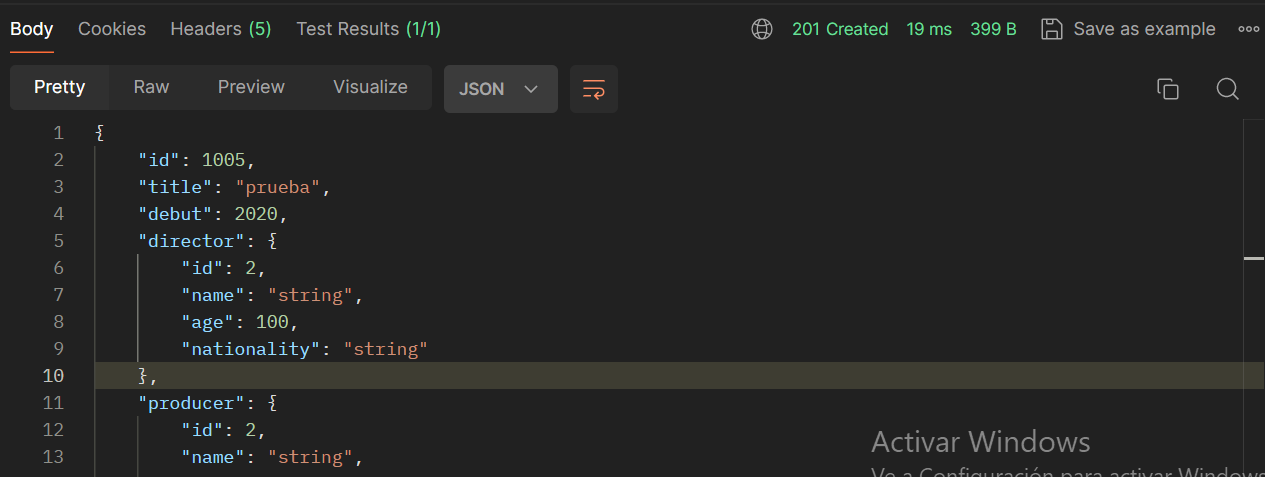
      "nationality": "string"

    }

  ]

}

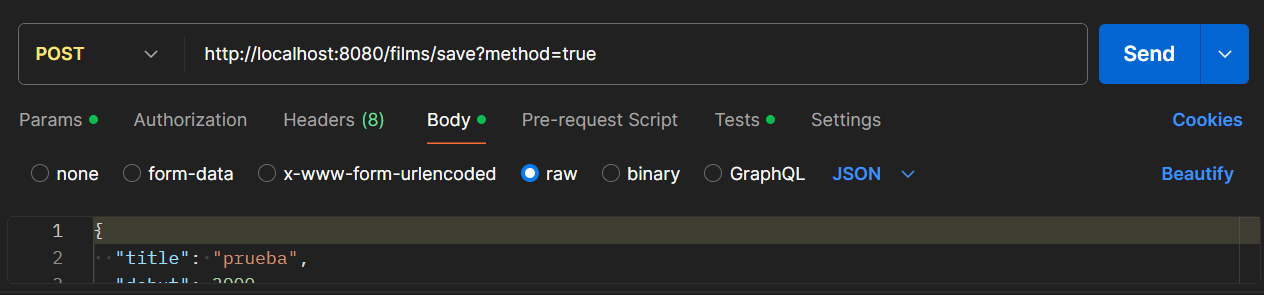
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 6 |
| Descripción | Create Film 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/films/save?method=true |



Con el siguiente body

{

  "title": "prueba",

  "debut": 3000,

  "director": {

    "id": 2,

    "name": "string",

    "age": 100,

    "nationality": "string"

  },

  "producer": {

    "id": 2,

    "name": "string",

    "debut": 2024

  },

  "actors": [

    {

      "id": 500,

      "name": "string",

      "age": 100,

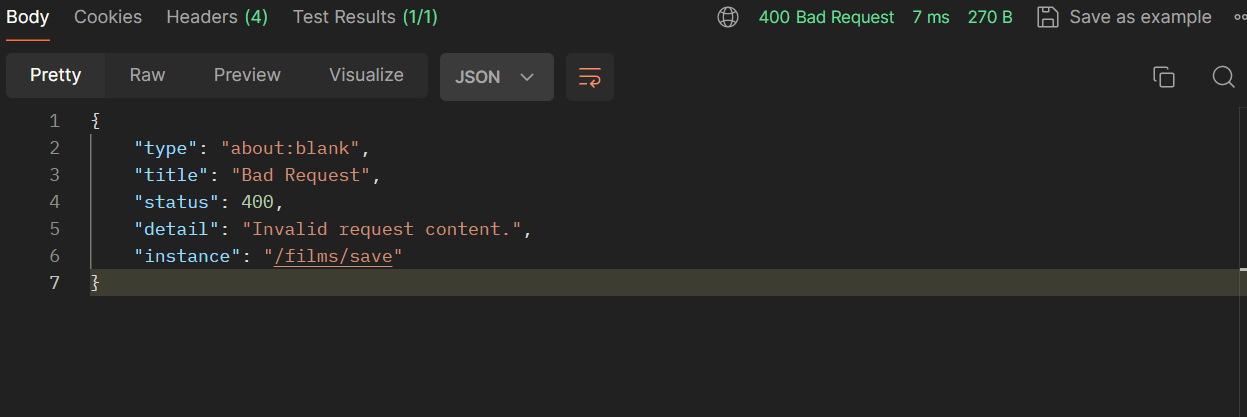
      "nationality": "string"

    }

  ]

}

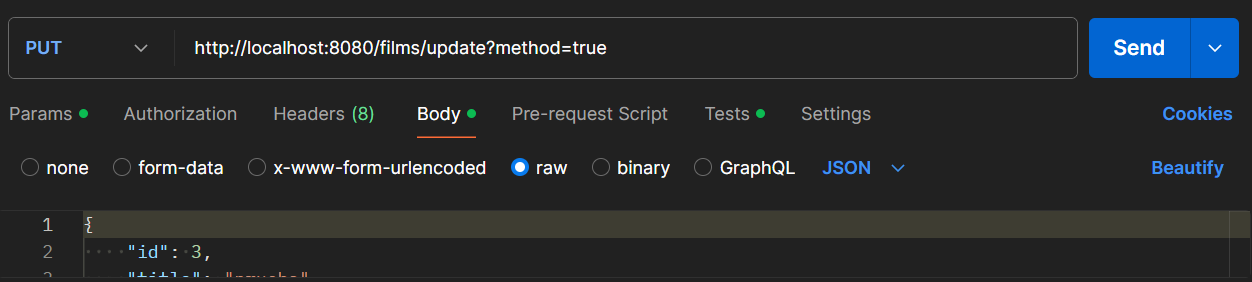
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 7 |
| Descripción | Update Film 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/films/update?method=true |



Con el siguiente body

{

    "id": 3,

    "title": "prueba",

    "debut": 2020,

    "director": {

        "id": 2,

        "name": "string",

        "age": 100,

        "nationality": "string"

    },

    "producer": {

        "id": 2,

        "name": "string",

        "debut": 2024

    },

    "actors": [

        {

            "id": 500,

            "name": "string",

            "age": 100,

            "nationality": "string"

        }

    ]

}

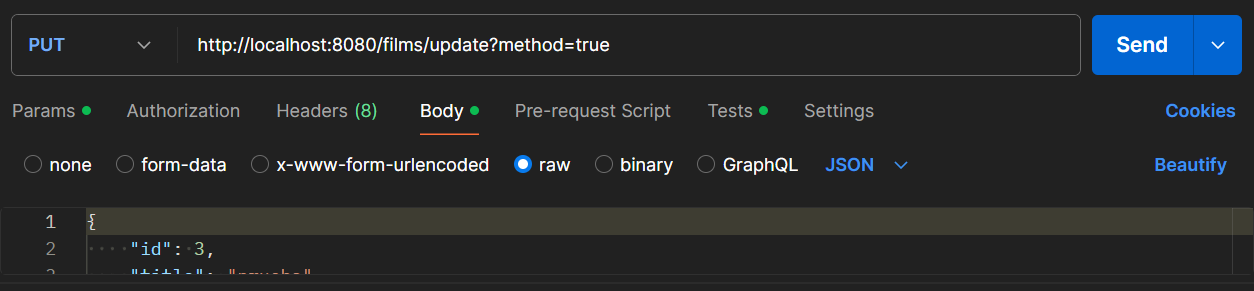
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 8 |
| Descripción | Update Film 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/films/update?method=true |



Con el siguiente body

{

    "id": 3,

    "title": "prueba",

    "debut": 3000,

    "director": {

        "id": 2,

        "name": "string",

        "age": 100,

        "nationality": "string"

    },

    "producer": {

        "id": 2,

        "name": "string",

        "debut": 2024

    },

    "actors": [

        {

            "id": 500,

            "name": "string",

            "age": 100,

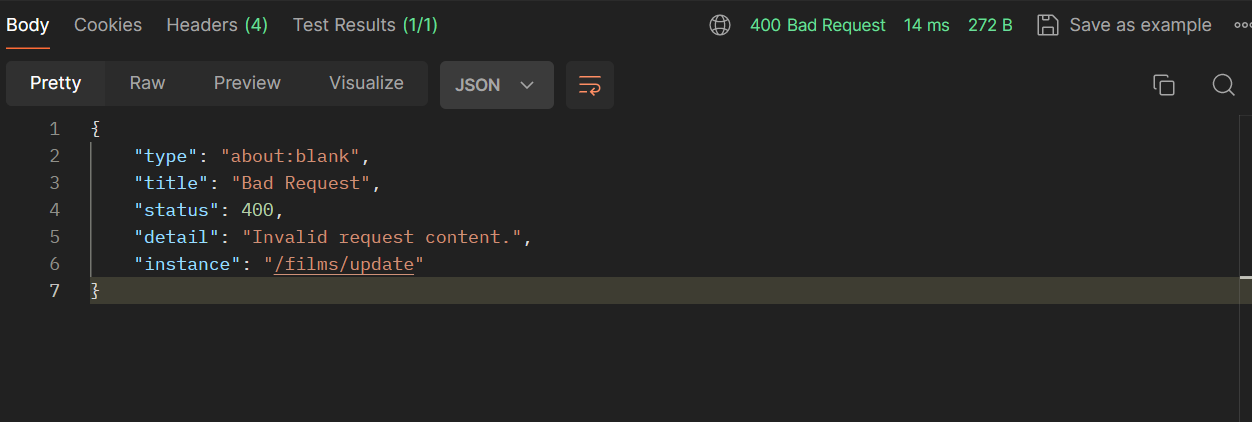
            "nationality": "string"

        }

    ]

}

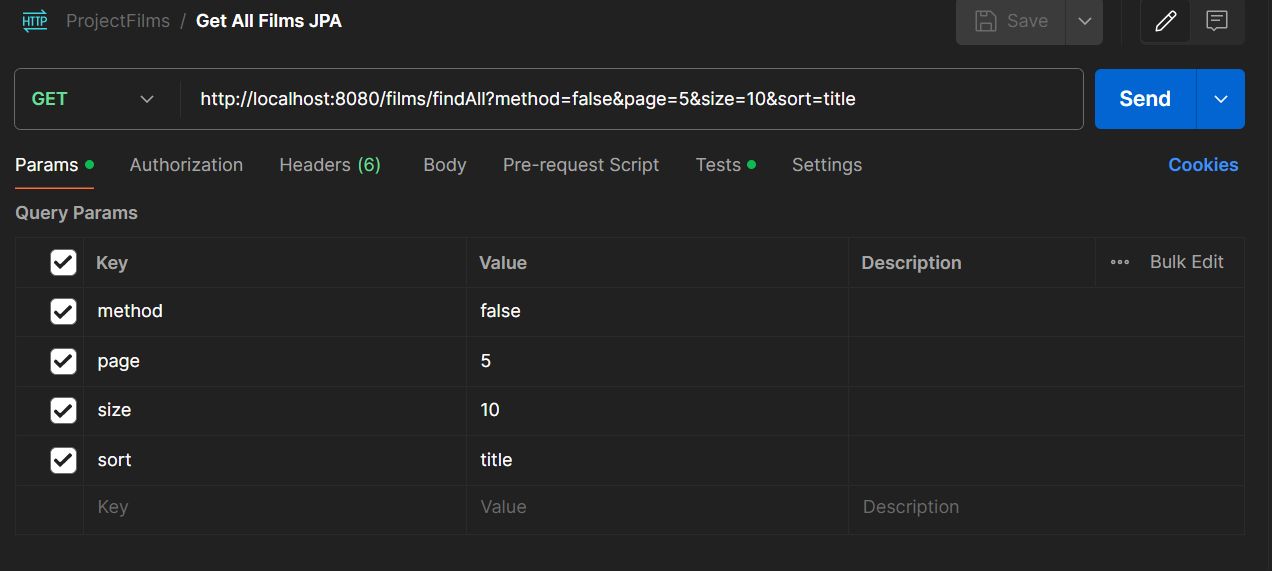
Revisamos la salida:



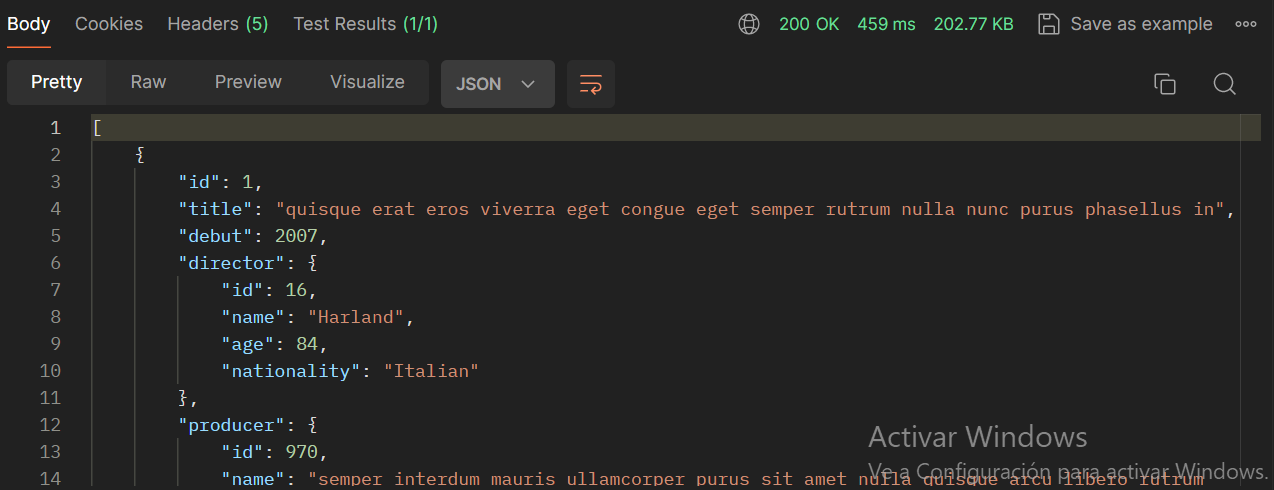
|  |  |
| --- | --- |
| # Paso: | 9 |
| Descripción | Get All Films JPA |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

|  |
| --- |
| http://localhost:8080/films/findAll?method=false&page=5&size=10&sort=title |



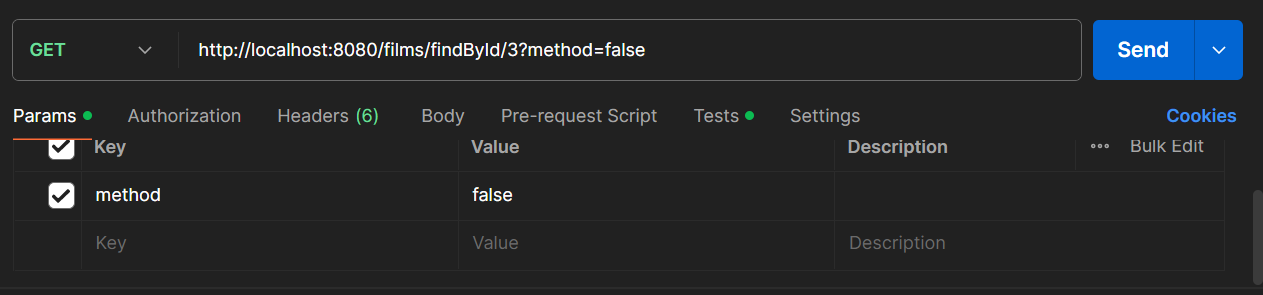
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 10 |
| Descripción | Get A Film 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/films/findById/3?method=false |



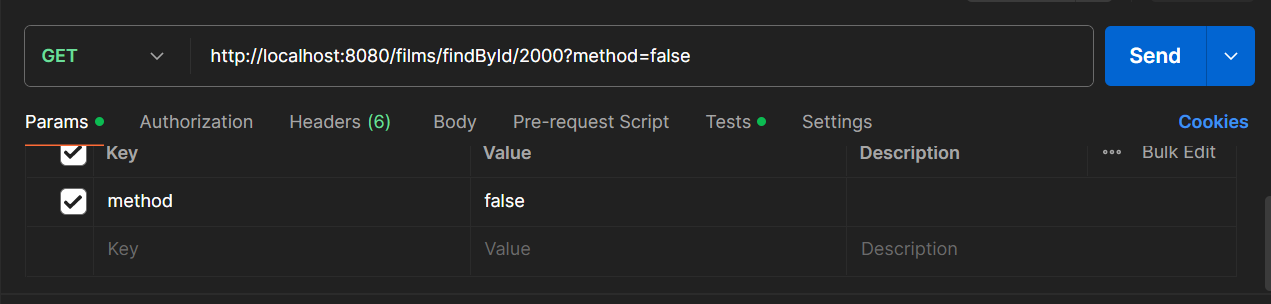
Revisamos la salida:



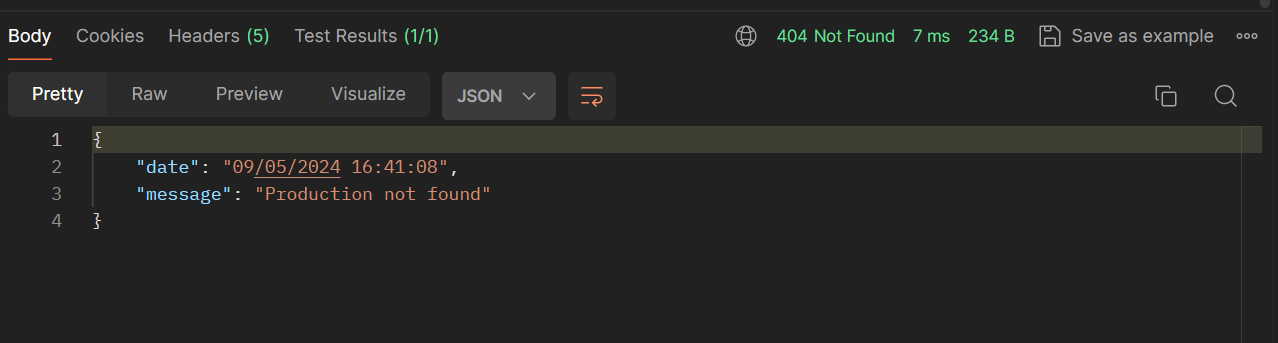
|  |  |
| --- | --- |
| # Paso: | 11 |
| Descripción | Get A Film JPA Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 404 Not Found. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

|  |
| --- |
| http://localhost:8080/films/findById/2000?method=false |



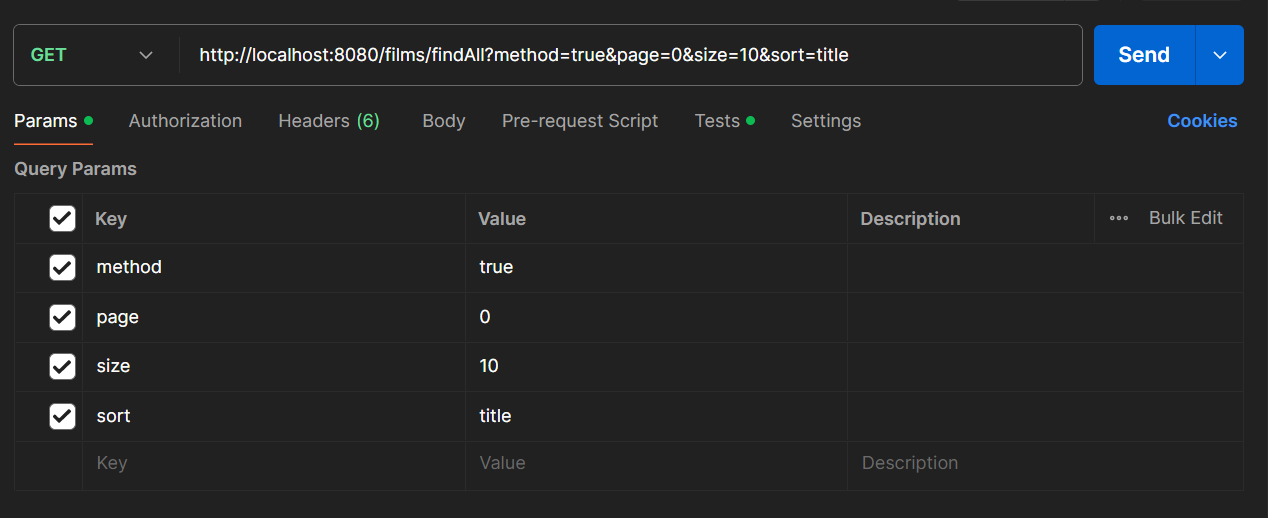
Revisamos la salida:



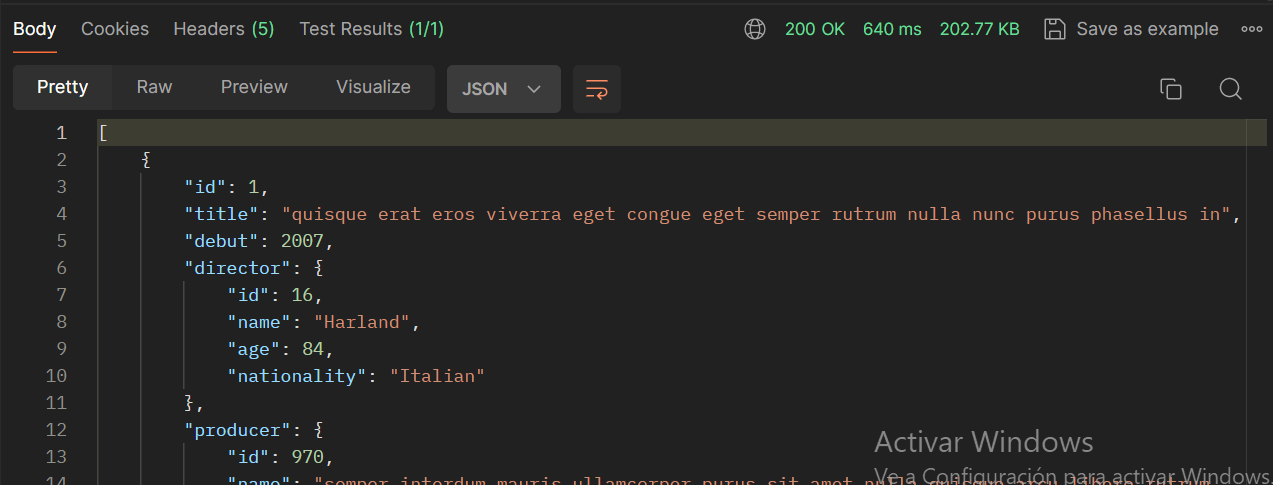
|  |  |
| --- | --- |
| # Paso: | 12 |
| Descripción | Get All Films Criteria |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

|  |
| --- |
| http://localhost:8080/films/findAll?method=true&page=5&size=10&sort=title |



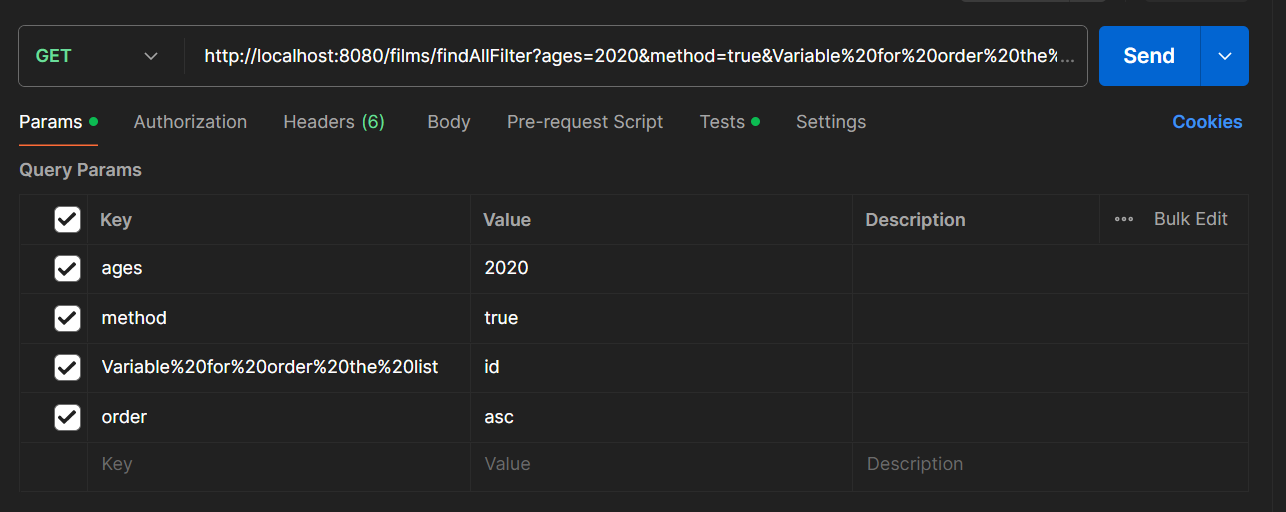
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 13 |
| Descripción | Get All Films Criteria Filter |
| 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/films/findAllFilter?ages=2020&method=true&Variable%20for%20order%20the%20list=id&order=asc |



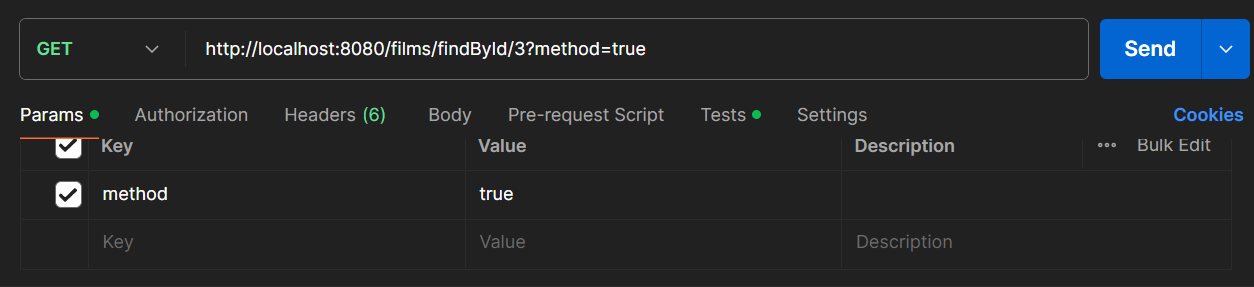
Revisamos la salida:



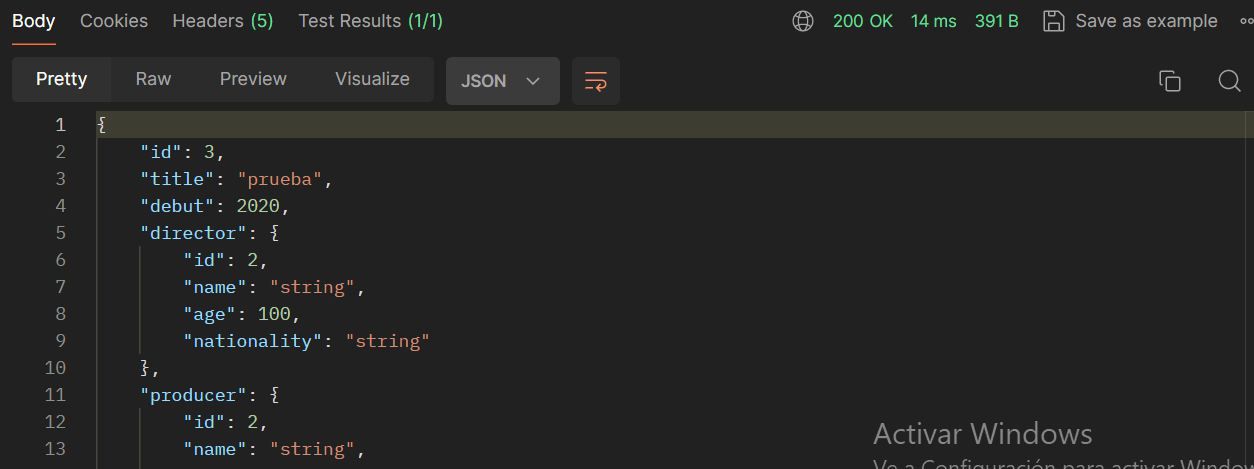
|  |  |
| --- | --- |
| # Paso: | 14 |
| Descripción | Get A Films Criteria |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

|  |
| --- |
| http://localhost:8080/films/findById/3?method=true |



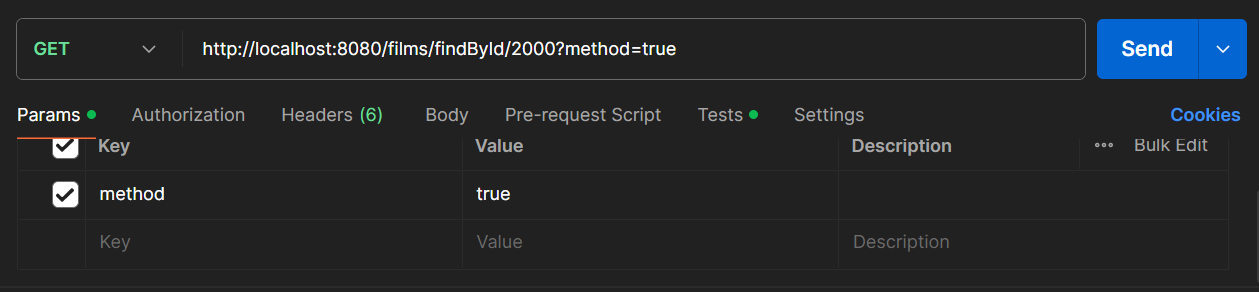
Revisamos la salida:



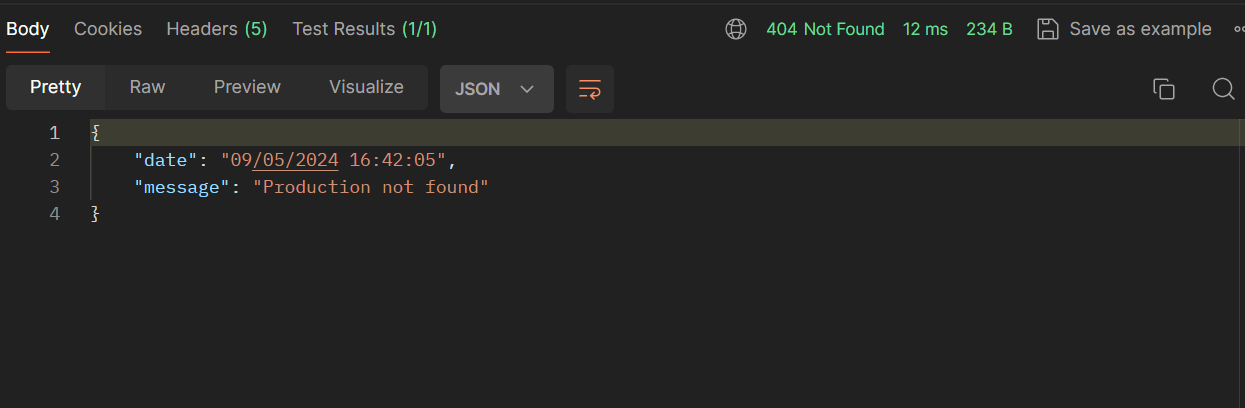
|  |  |
| --- | --- |
| # Paso: | 15 |
| Descripción | Get A Films Criteria Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 404 Not Found. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

|  |
| --- |
| http://localhost:8080/films/findById/2000?method=true |



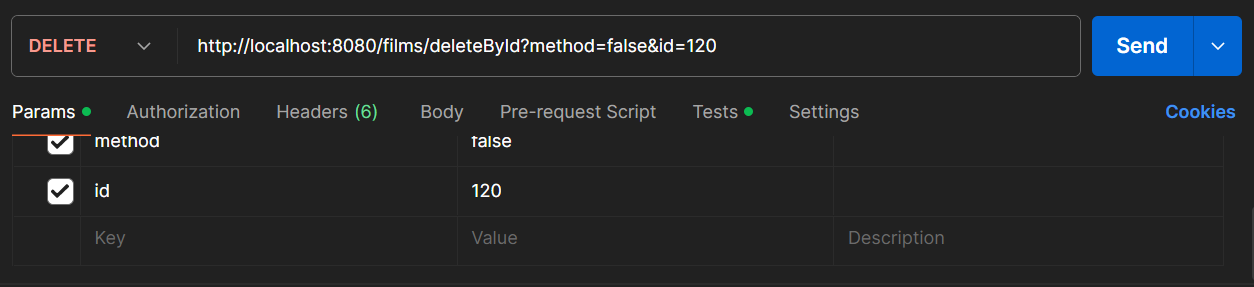
Revisamos la salida:



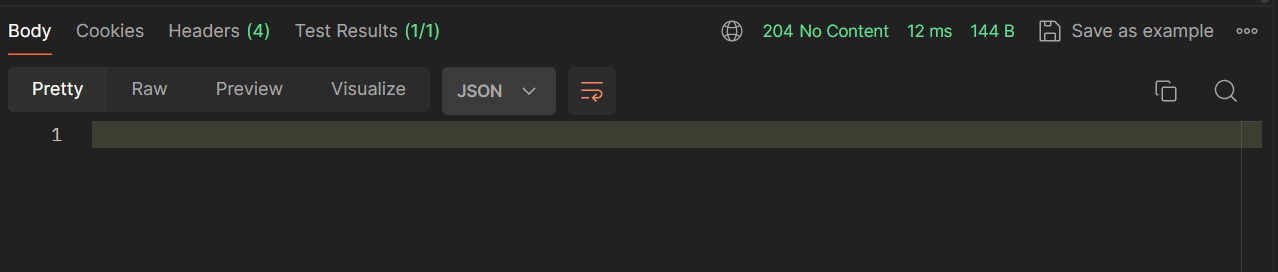
|  |  |
| --- | --- |
| # Paso: | 16 |
| Descripción | Delete A Film JPA |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 204 No Content. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo DELETE la siguiente URL:

|  |
| --- |
| http://localhost:8080/films/deleteById?method=false&id=120 |



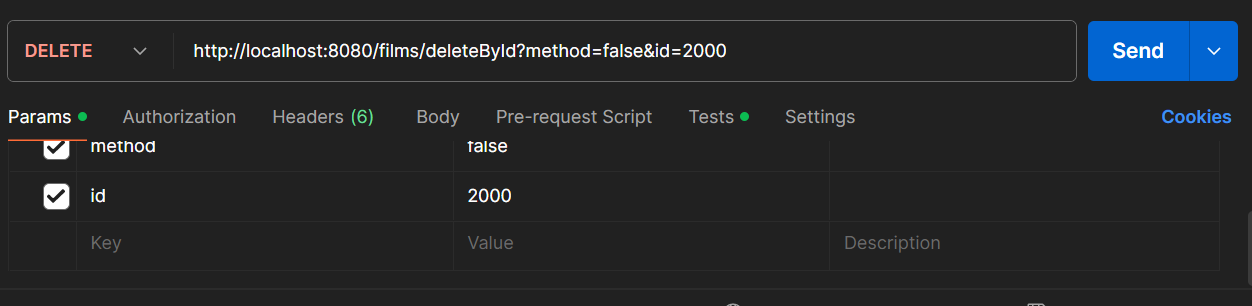
Revisamos la salida:



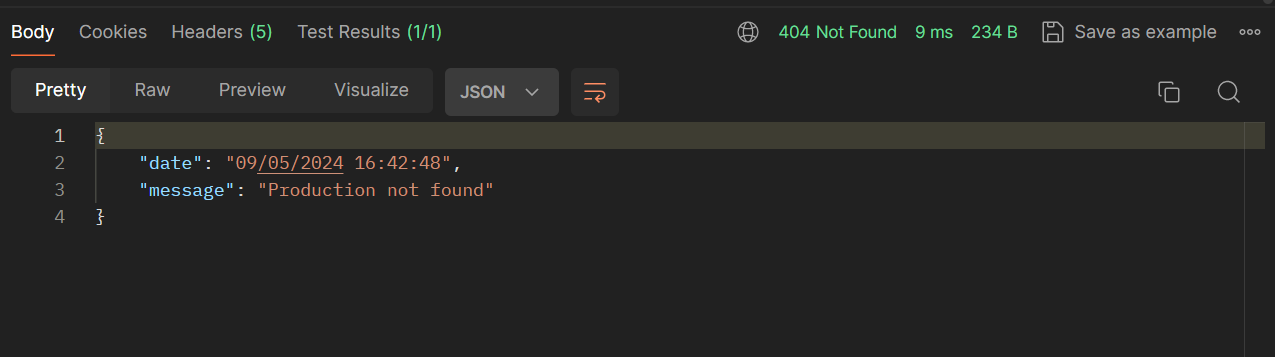
|  |  |
| --- | --- |
| # Paso: | 17 |
| Descripción | Delete A Film JPA Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 404 Not Found. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo DELETE la siguiente URL:

|  |
| --- |
| http://localhost:8080/films/deleteById?method=false&id=2000 |



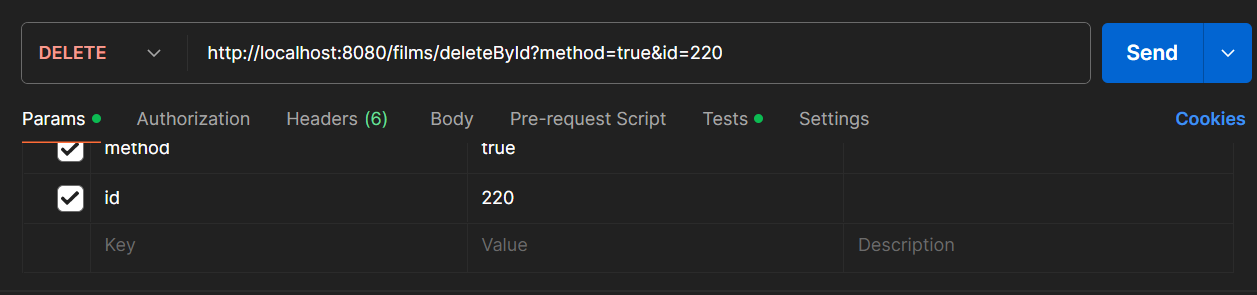
Revisamos la salida:



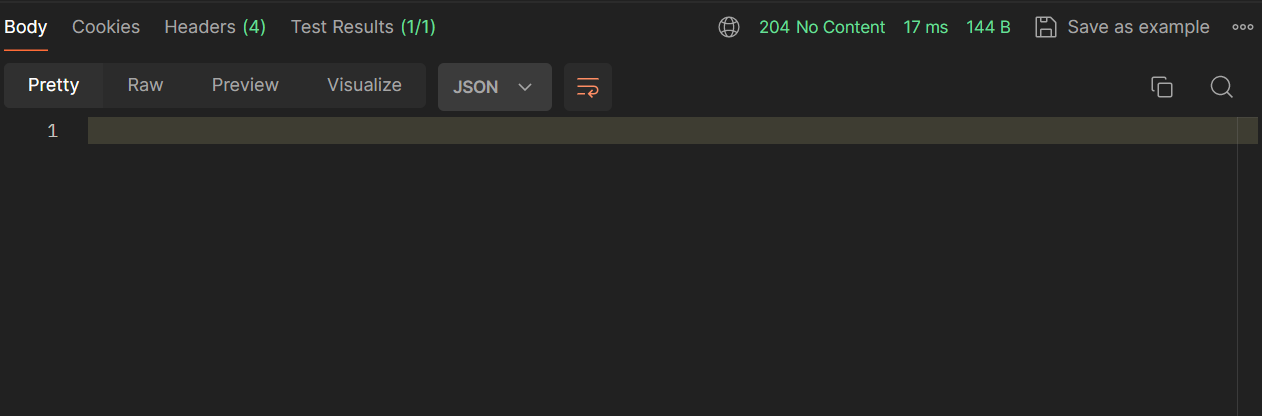
|  |  |
| --- | --- |
| # Paso: | 18 |
| Descripción | Delete A Film Criteria |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 204 No Content. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo DELETE la siguiente URL:

|  |
| --- |
| http://localhost:8080/films/deleteById?method=true&id=220 |



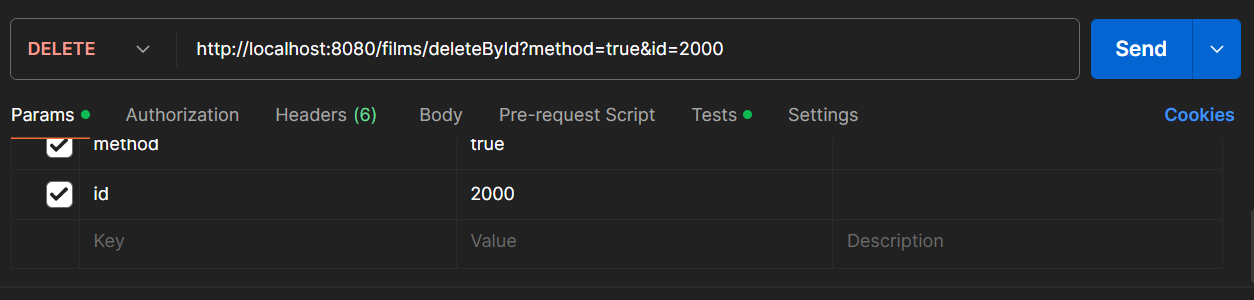
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 19 |
| Descripción | Delete A Film 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/films/deleteById?method=true&id=2000 |



Revisamos la salida:

