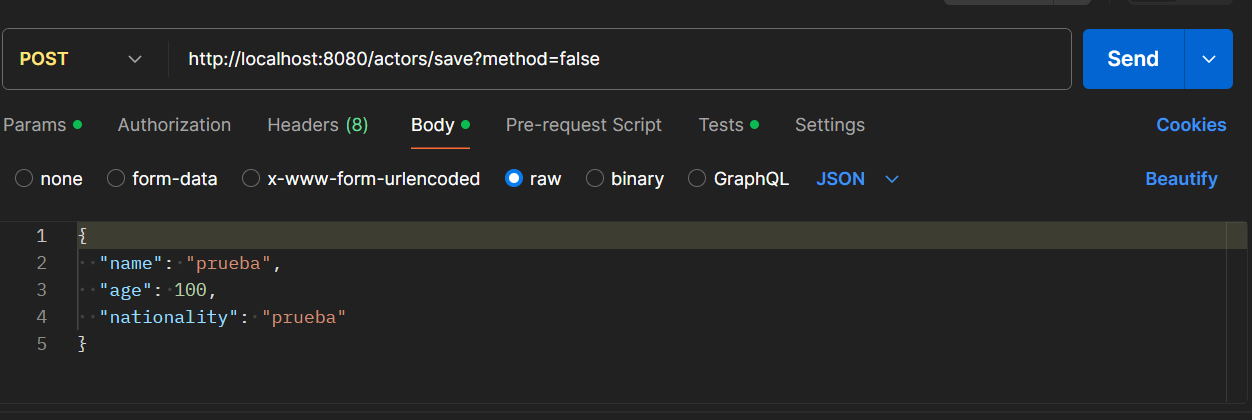
|  |  |
| --- | --- |
| Ejecución Automática de Caso de Test | |
| Dominio | **Proyecto** |
|  |  |
| Aplicación | **Ámbito Ejecución** |
| NOMBRE DE LA APLICACIÓN Films | Offline |
| Descripción | **Pre-requisitos** |
| Prueba unitaria para Proyecto Spring Films - Actors |  |
| Tester Ejecución | **Fecha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

|  |  |
| --- | --- |
| # Paso: | 1 |
| Descripción | Create Actor 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/actors/save?method=false |



Con el siguiente body

{

    "id": 1002,

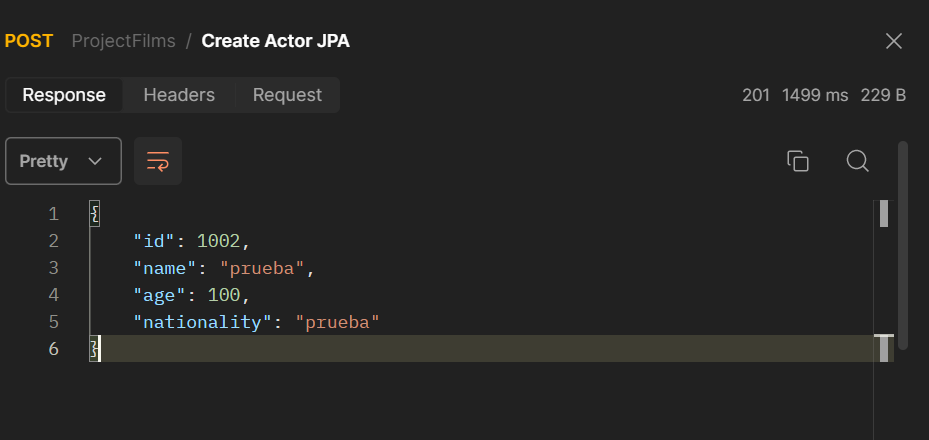
    "name": "prueba",

    "age": 100,

    "nationality": "prueba"

}

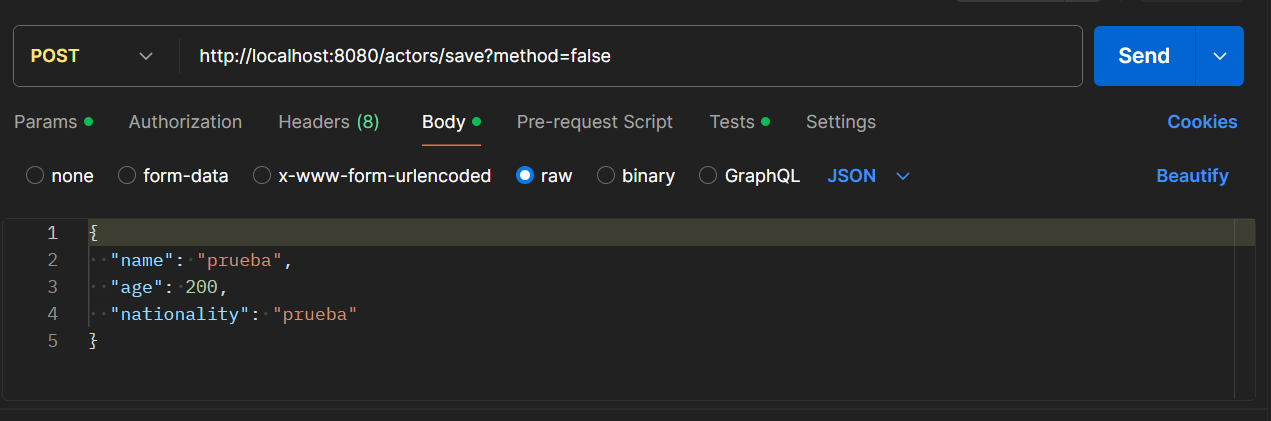
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 2 |
| Descripción | Create Actor 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/actors/save?method=false |



Con el siguiente body

{

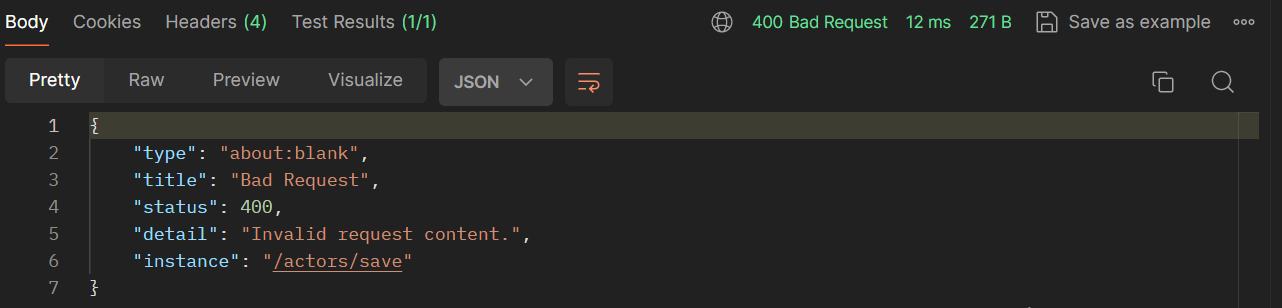
  "name": "prueba",

  "age": 200,

  "nationality": "prueba"

}

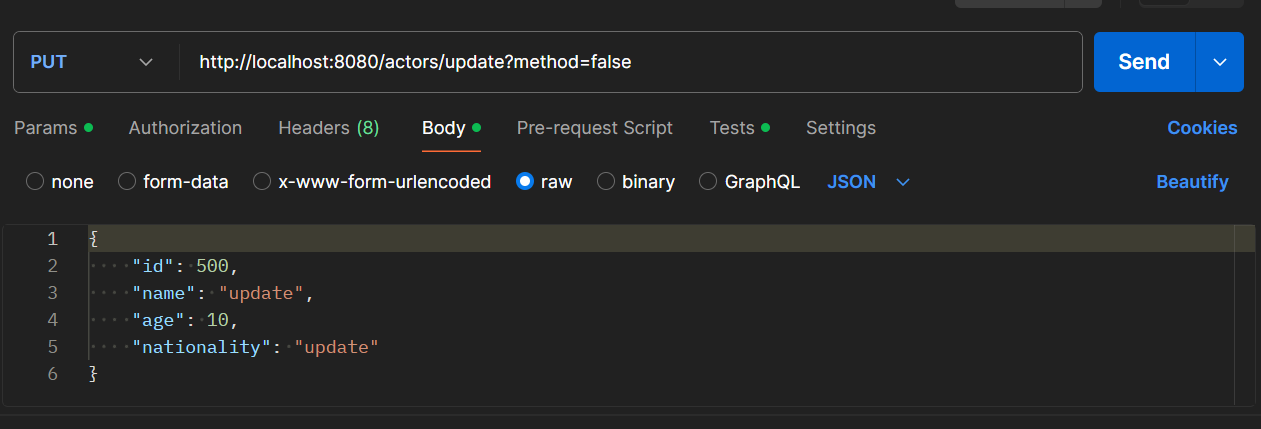
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 3 |
| Descripción | Update Actor 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/actors/update?method=false |



Con el siguiente body

{

    "id": 500,

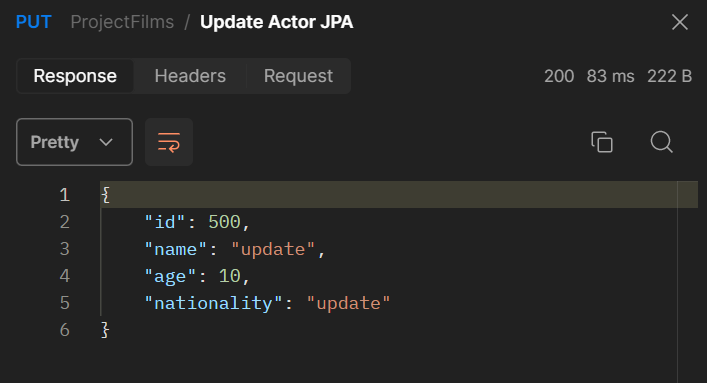
    "name": "update",

    "age": 10,

    "nationality": "update"

}

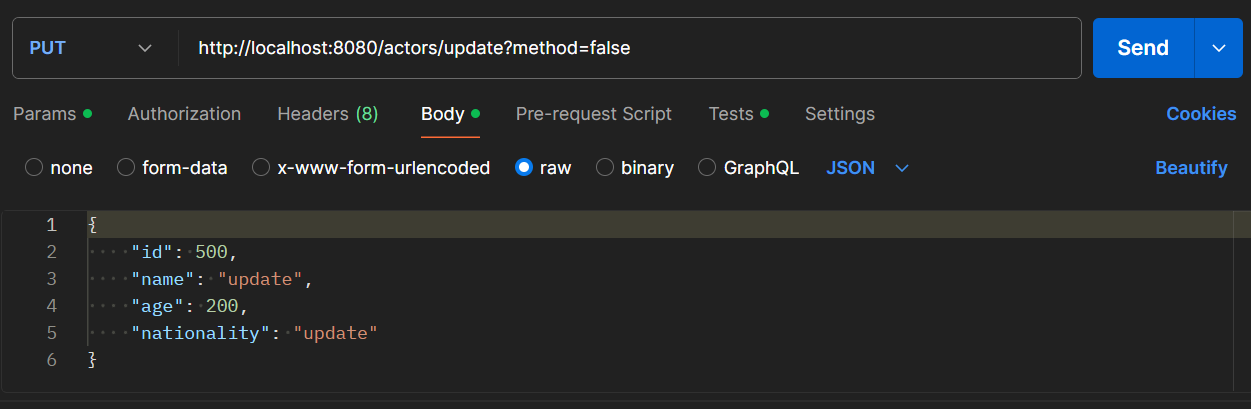
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 4 |
| Descripción | Update Actor 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/actors/update?method=false |



Con el siguiente body

{

    "id": 500,

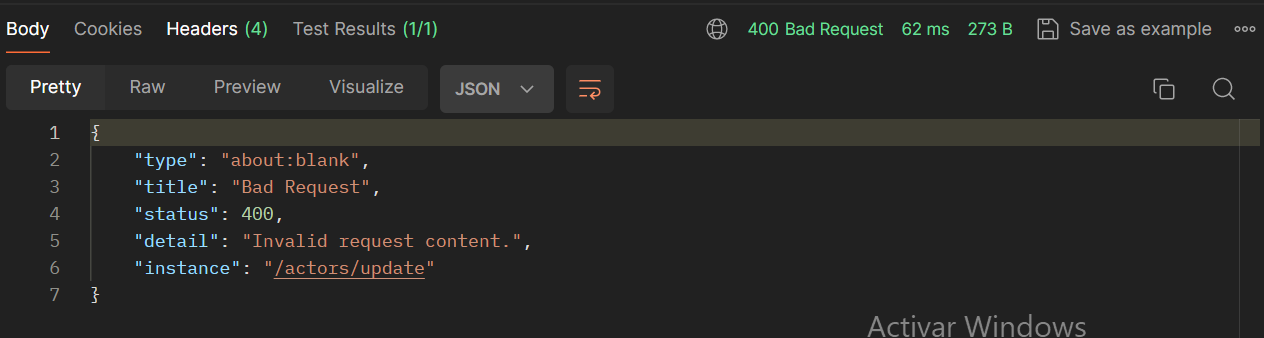
    "name": "update",

    "age": 200,

    "nationality": "update"

}

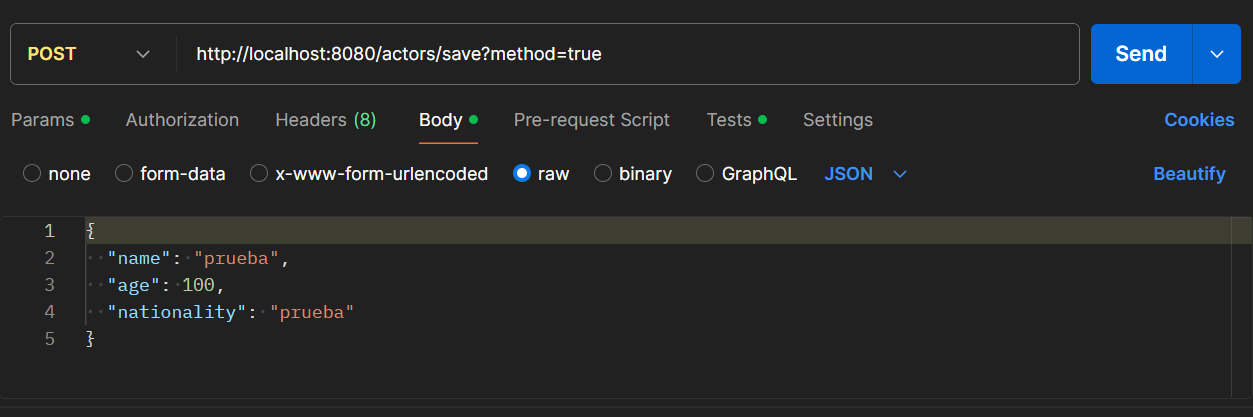
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 5 |
| Descripción | Create Actor 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/actors/save?method=true |



Con el siguiente body

{

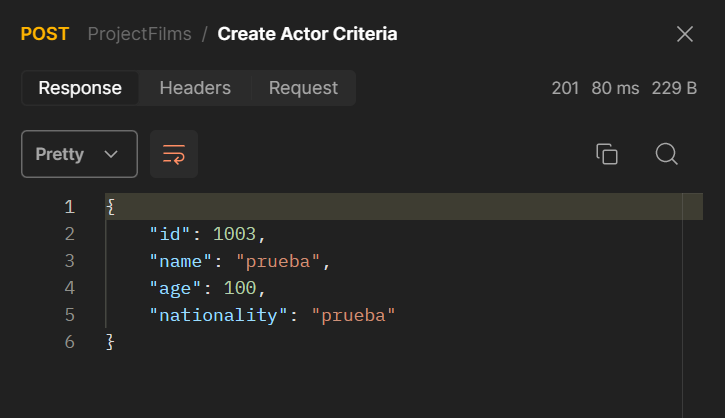
  "name": "prueba",

  "age": 100,

  "nationality": "prueba"

}

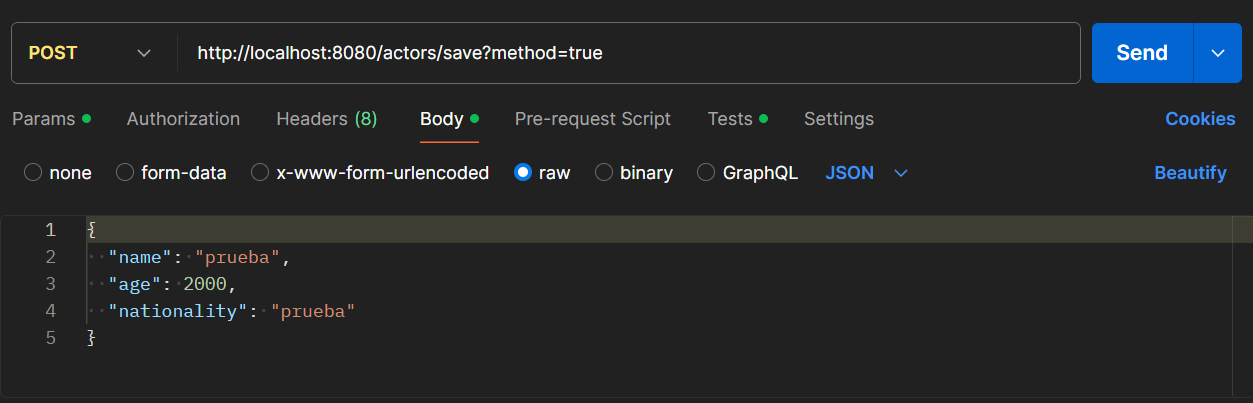
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 6 |
| Descripción | Create Actor 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/actors/save?method=true |



Con el siguiente body

{

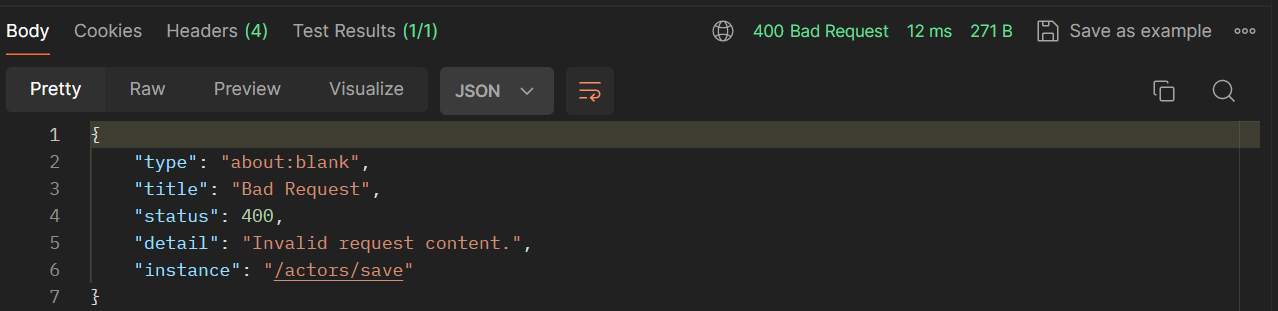
  "name": "prueba",

  "age": 2000,

  "nationality": "prueba"

}

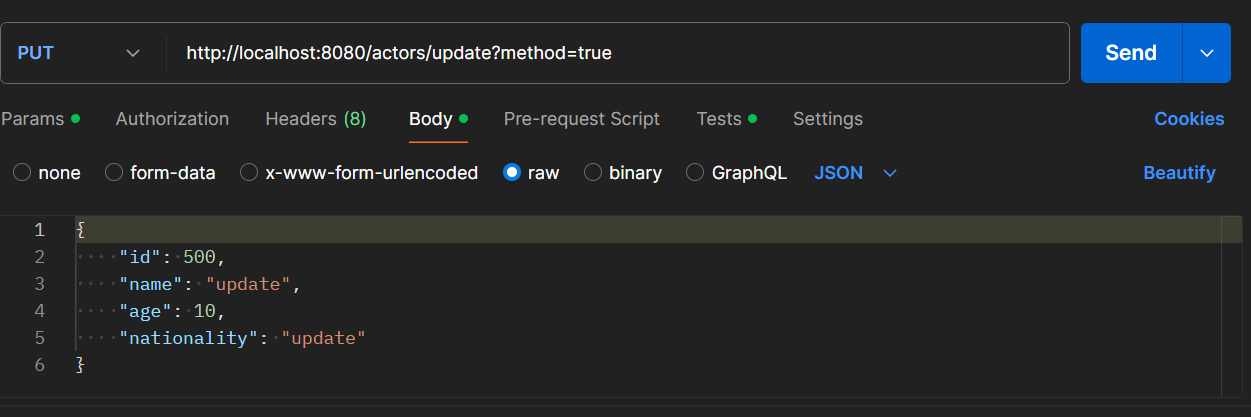
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 7 |
| Descripción | Update Actor 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/actors/update?method=true |



Con el siguiente body

{

    "id": 500,

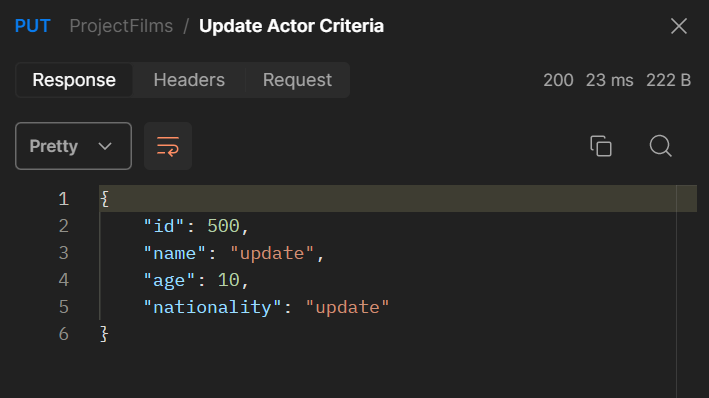
    "name": "update",

    "age": 10,

    "nationality": "update"

}

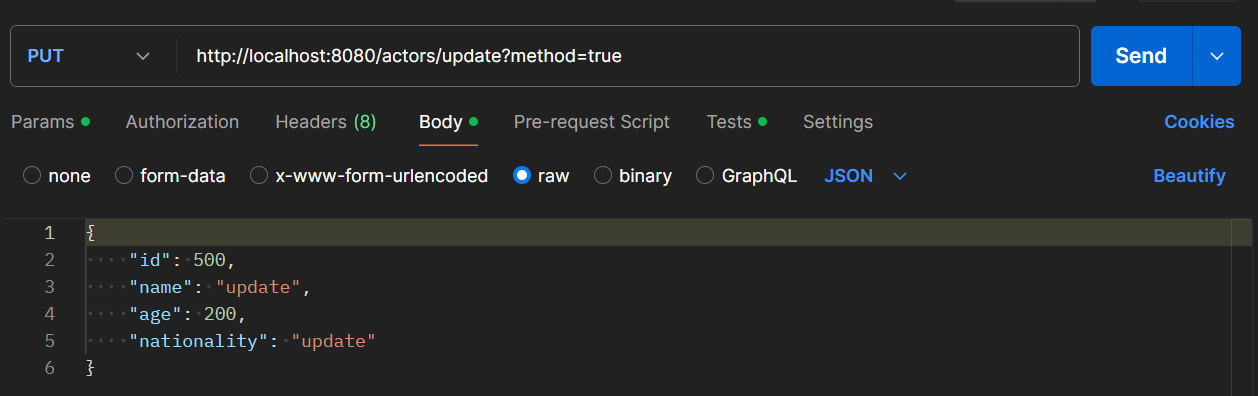
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 8 |
| Descripción | Update Actor 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/actors/update?method=true |



Con el siguiente body

{

    "id": 500,

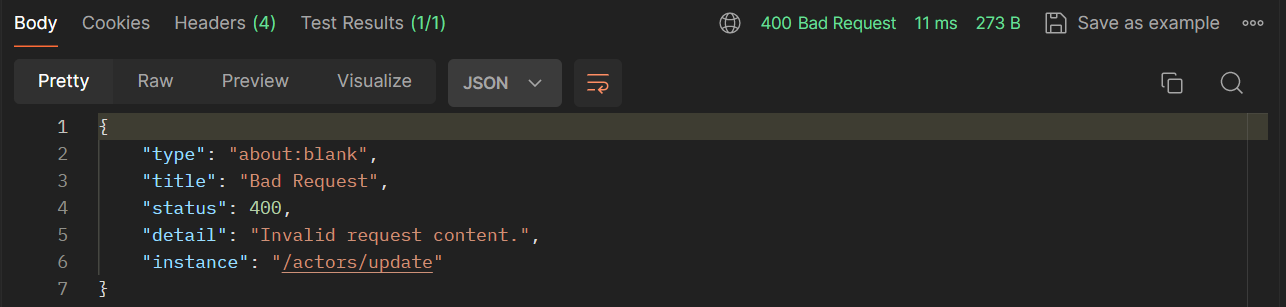
    "name": "update",

    "age": 200,

    "nationality": "update"

}

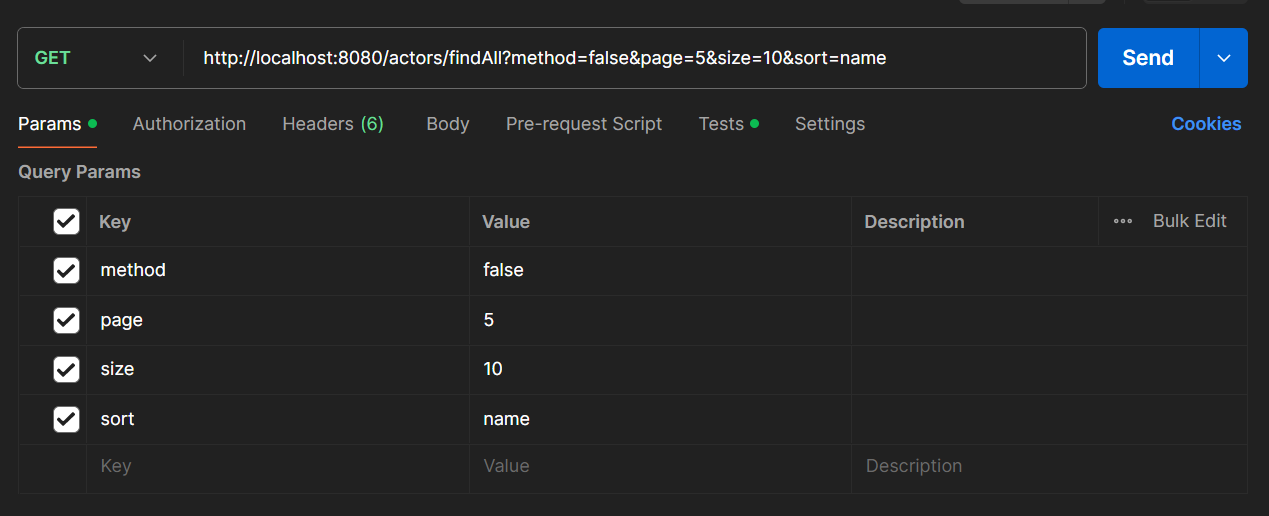
Revisamos la salida:



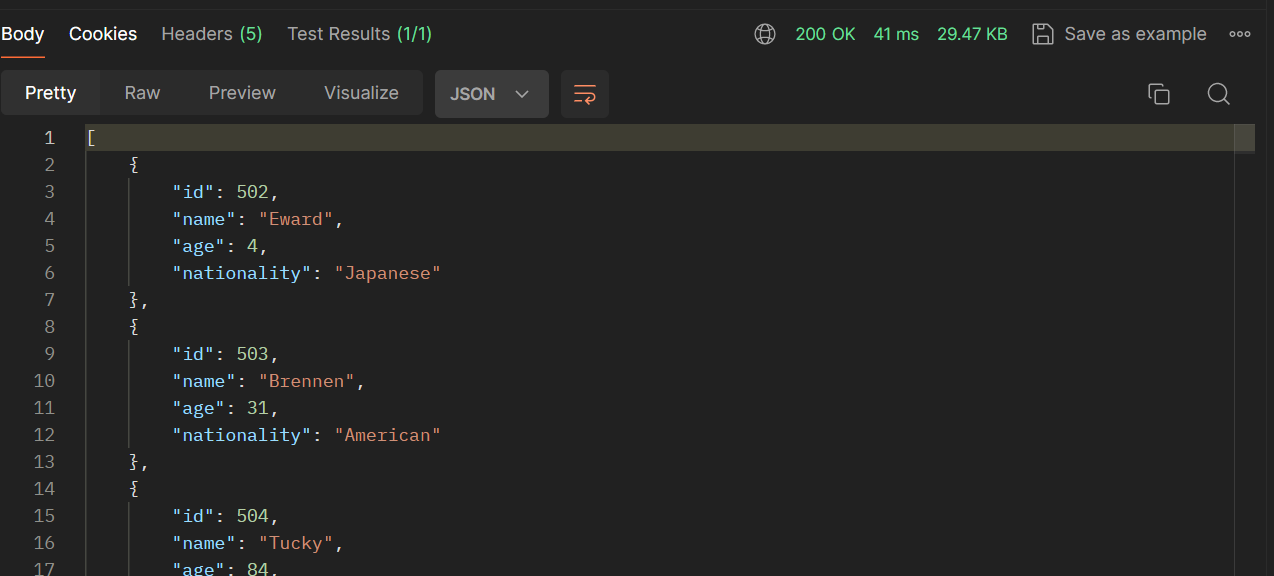
|  |  |
| --- | --- |
| # Paso: | 9 |
| Descripción | Get All Actors JPA |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo la GET siguiente URL:

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



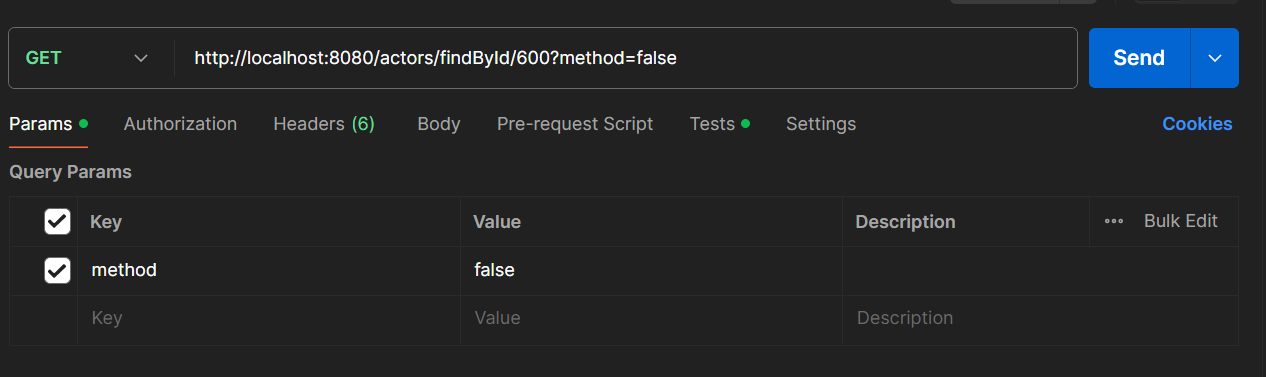
Revisamos la salida:



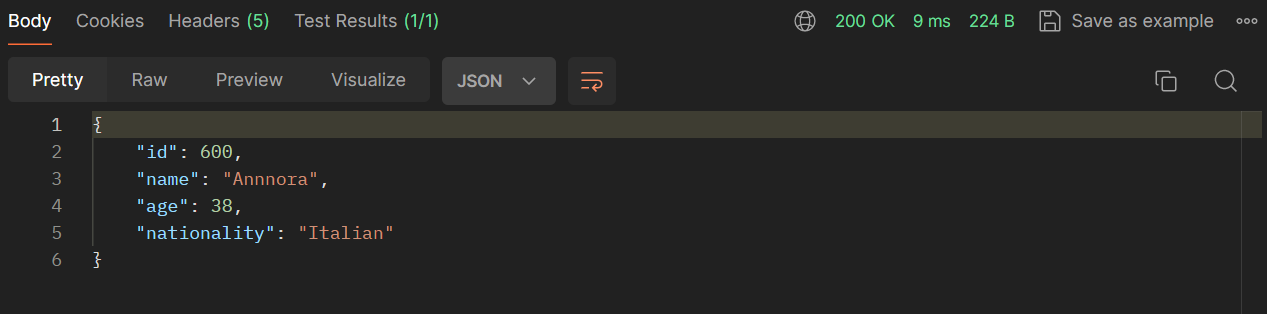
|  |  |
| --- | --- |
| # Paso: | 10 |
| Descripción | Get An Actor JPA |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

|  |
| --- |
| http://localhost:8080/actors/findById/600?method=false |



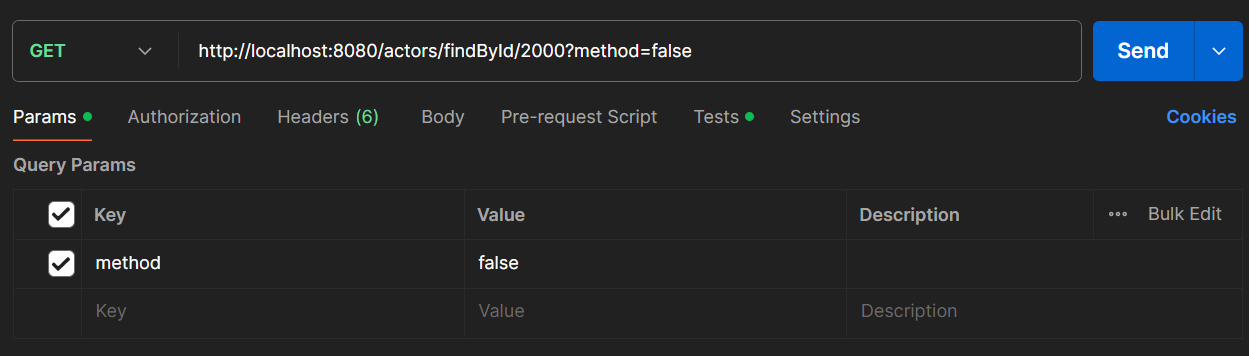
Revisamos la salida:



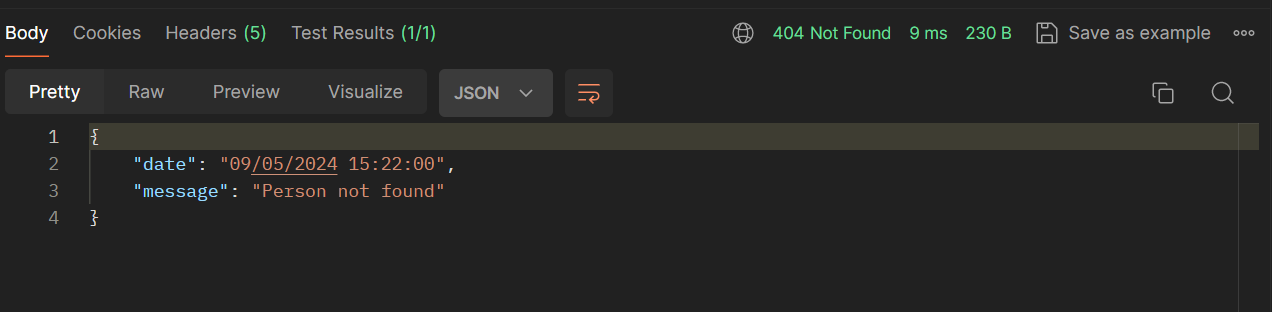
|  |  |
| --- | --- |
| # Paso: | 11 |
| Descripción | Get An Actor JPA Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 404 Not Found. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo la GET siguiente URL:

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



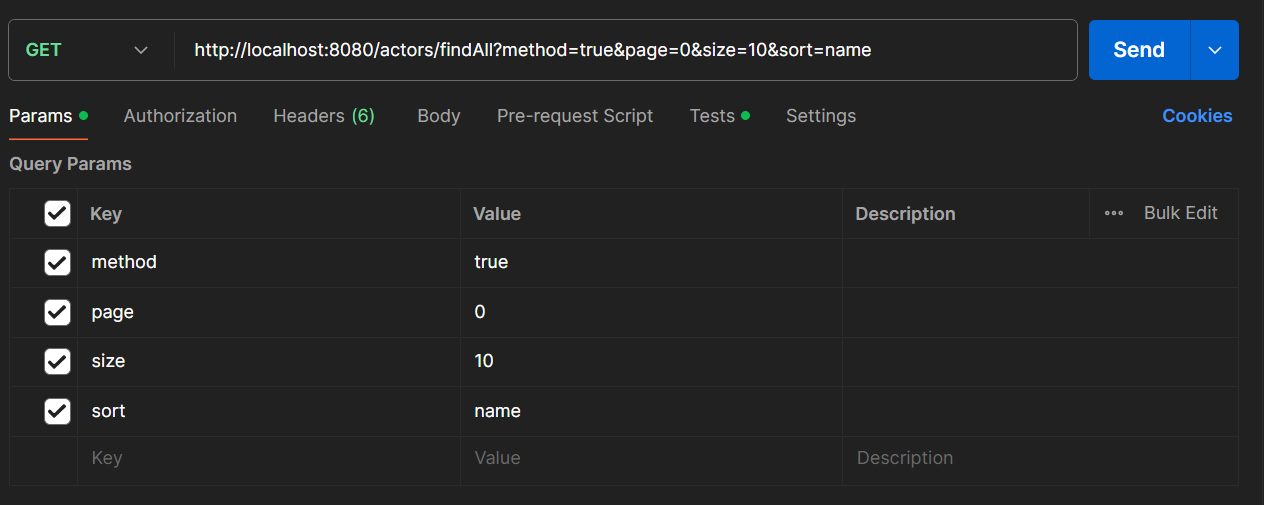
Revisamos la salida:



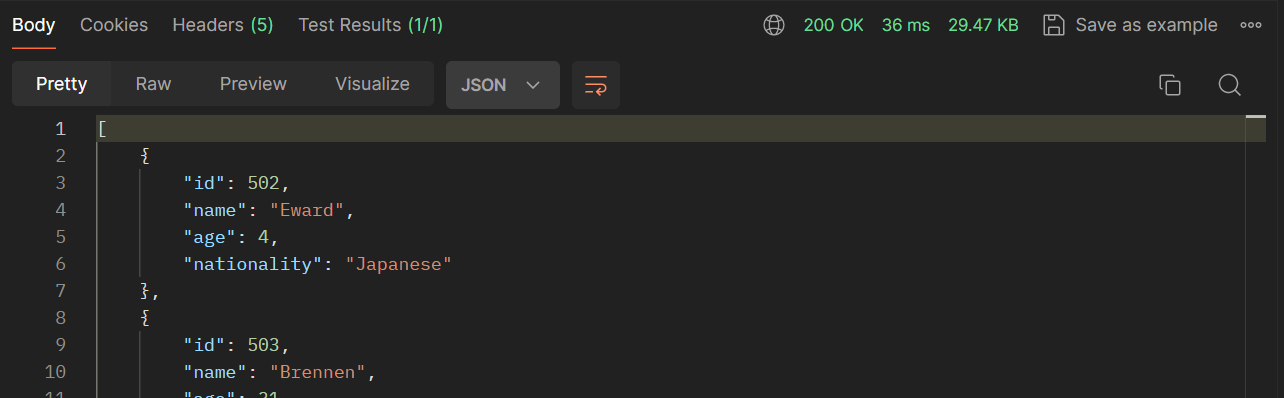
|  |  |
| --- | --- |
| # Paso: | 12 |
| Descripción | Get All Actors Criteria |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

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



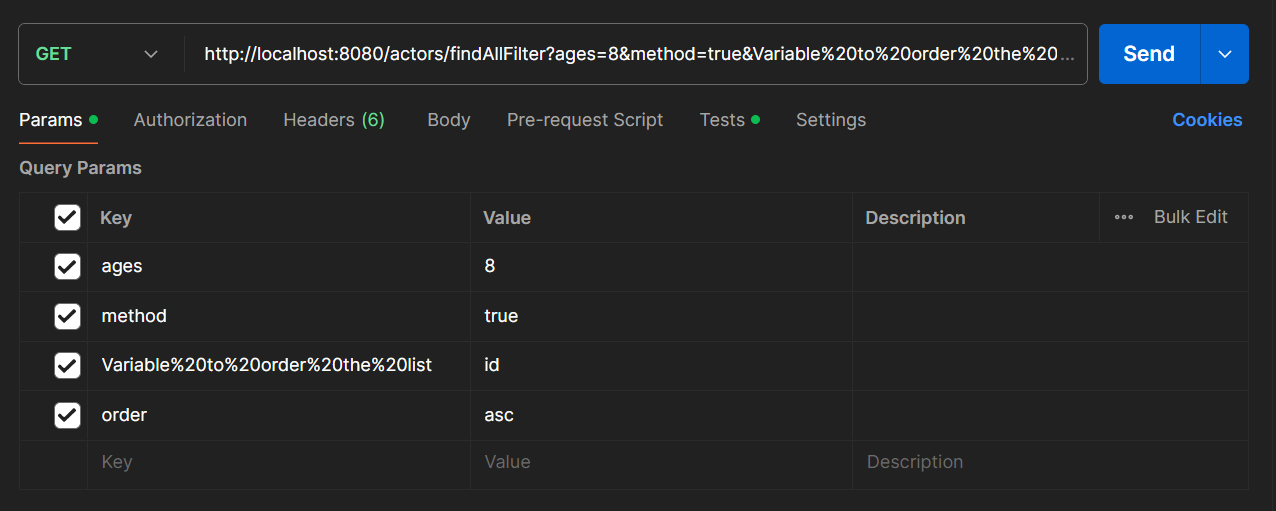
Revisamos la salida:



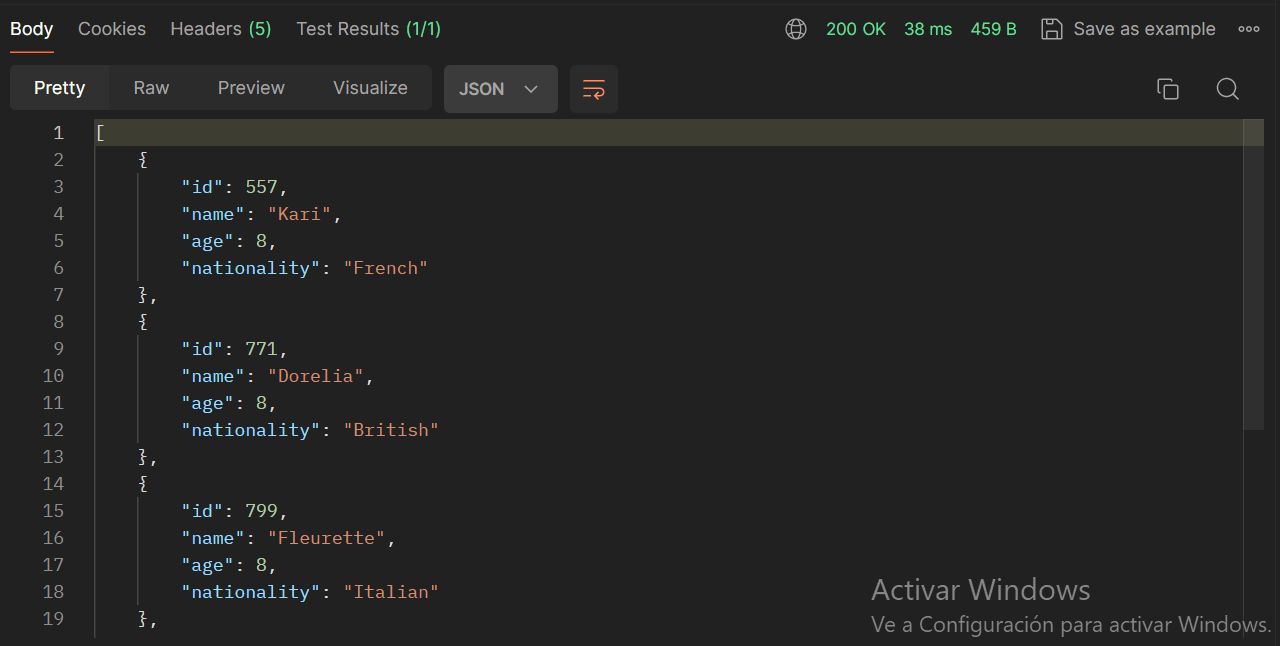
|  |  |
| --- | --- |
| # Paso: | 13 |
| Descripción | Get All Actors Criteria Filter |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 Ok Found. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

|  |
| --- |
| http://localhost:8080/actors/findAllFilter?ages=8&method=true&Variable%20to%20order%20the%20list=id&order=asc |



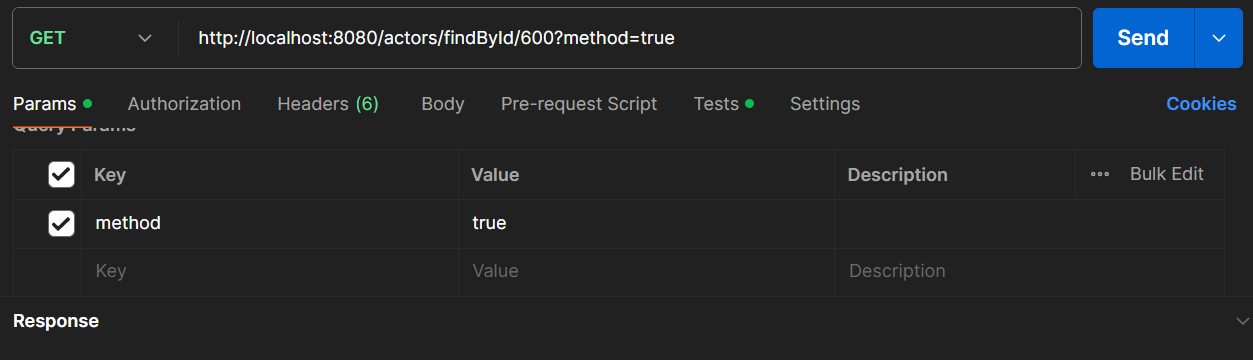
Revisamos la salida:



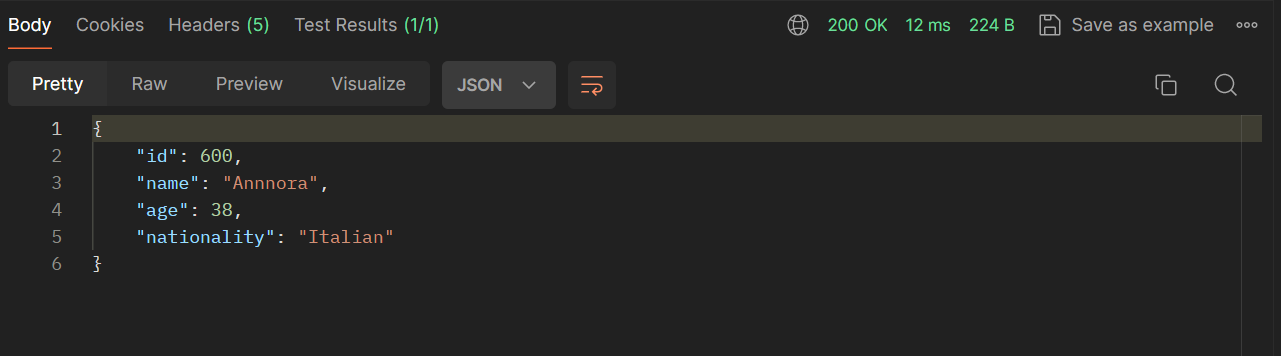
|  |  |
| --- | --- |
| # Paso: | 14 |
| Descripción | Get An Actor Criteria |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 200 OK. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

|  |
| --- |
| http://localhost:8080/actors/findById/600?method=true |



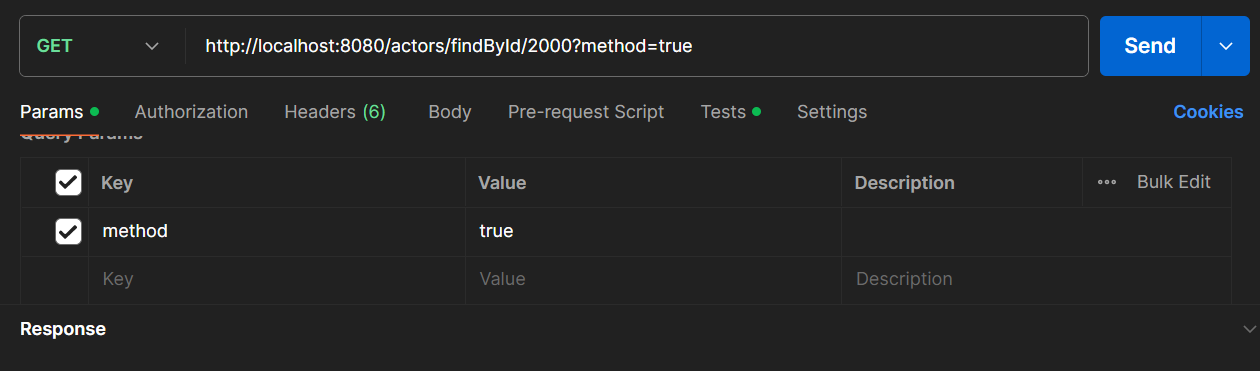
Revisamos la salida:



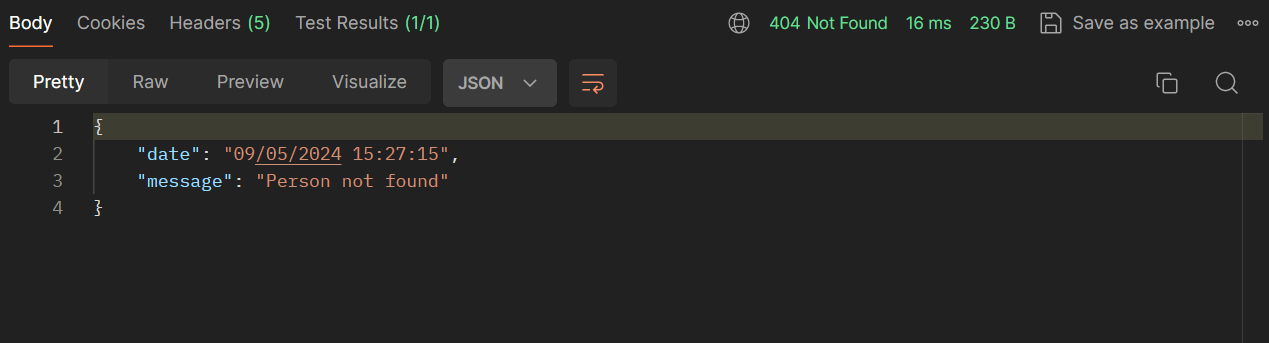
|  |  |
| --- | --- |
| # Paso: | 15 |
| Descripción | Get An Actor Criteria Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 404 Not Found. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo GET la siguiente URL:

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



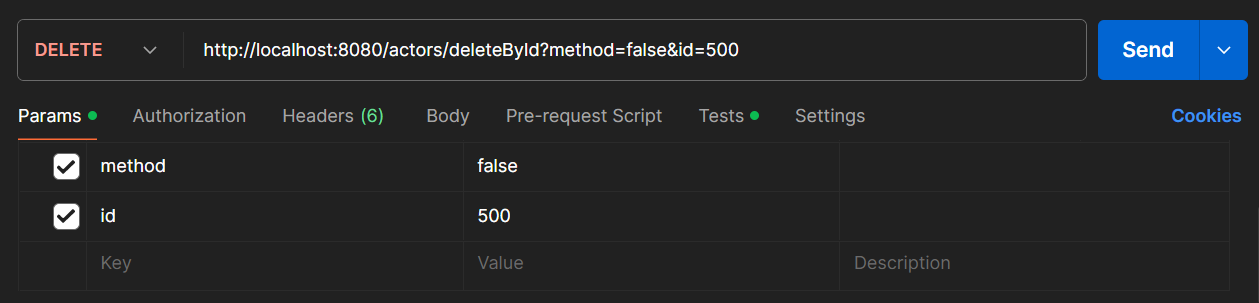
Revisamos la salida:



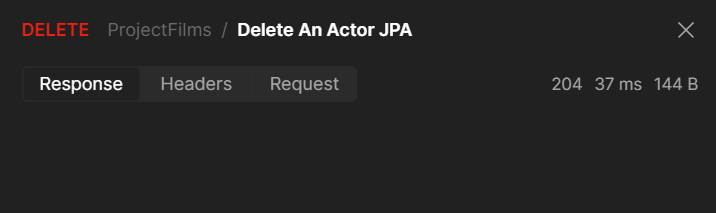
|  |  |
| --- | --- |
| # Paso: | 16 |
| Descripción | Delete An Actor JPA |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 204 No Content. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo DELETE la siguiente URL:

|  |
| --- |
| http://localhost:8080/actors/deleteById?method=false&id=500 |



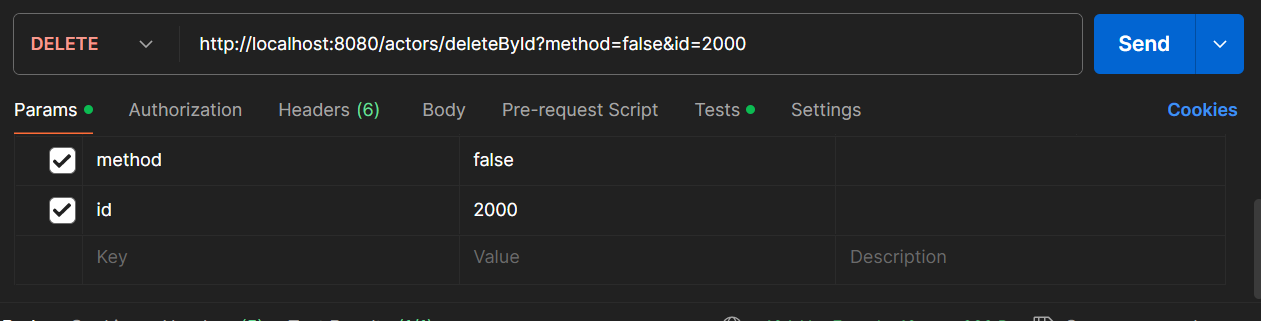
Revisamos la salida:



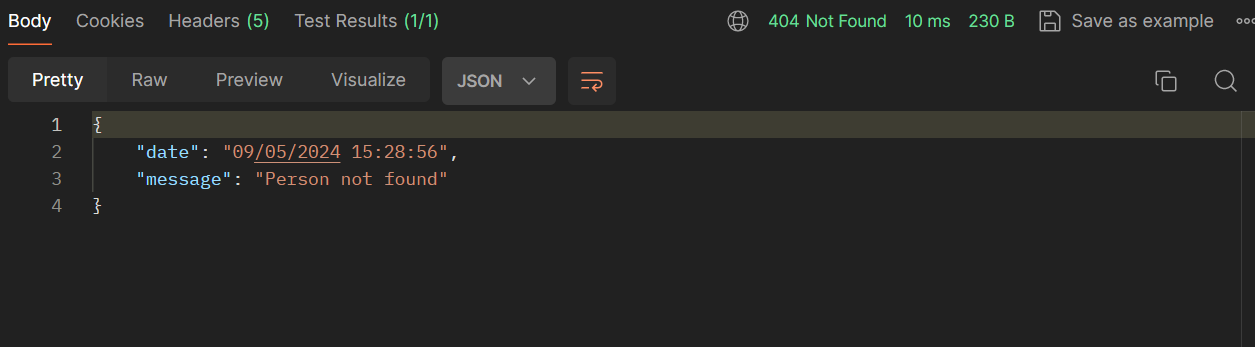
|  |  |
| --- | --- |
| # Paso: | 17 |
| Descripción | Delete An Actor JPA Error |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 404 Not Found. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo DELETE la siguiente URL:

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



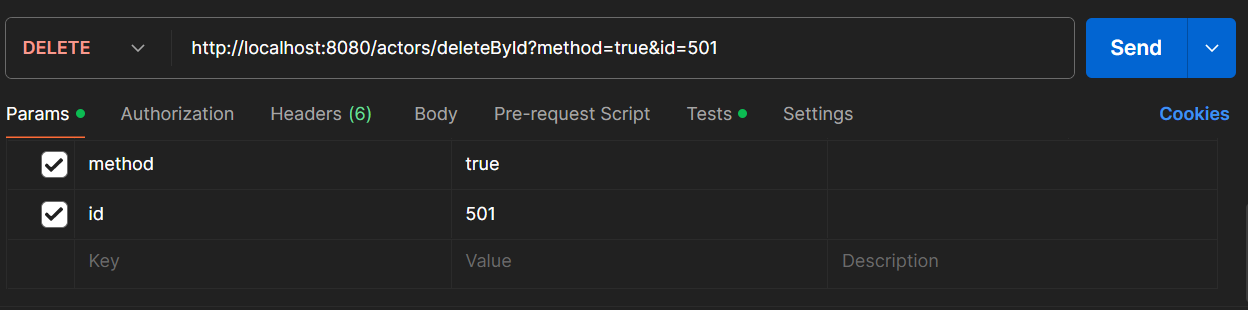
Revisamos la salida:



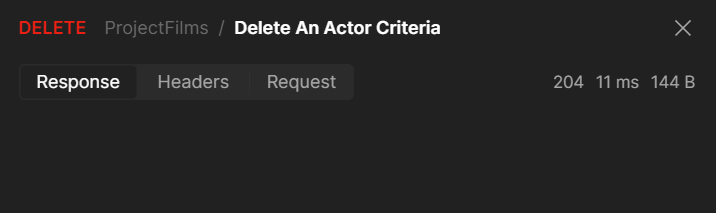
|  |  |
| --- | --- |
| # Paso: | 18 |
| Descripción | Delete An Actor Criteria |
| Resultado Esperado | La operación se completa correctamente obteniendo respuesta HTTP 204 No Content. La operación en la base de datos se completa correctamente |

Realizamos una petición REST de tipo DELETE la siguiente URL:

|  |
| --- |
| http://localhost:8080/actors/deleteById?method=true&id=501 |



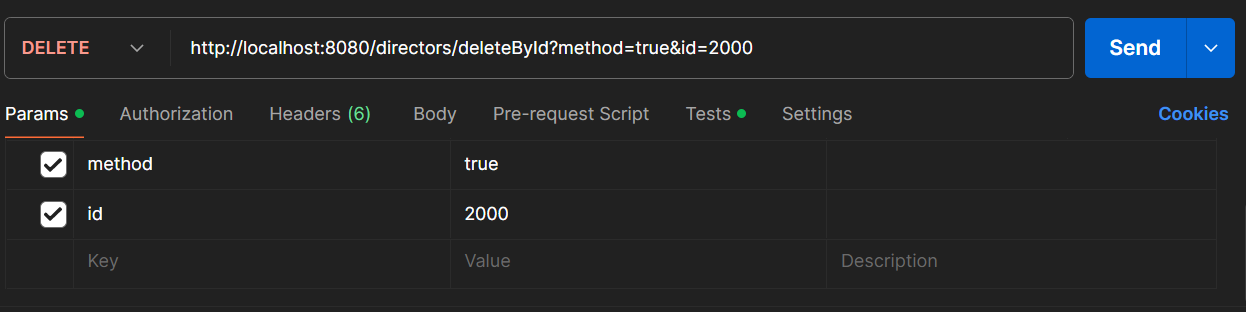
Revisamos la salida:



|  |  |
| --- | --- |
| # Paso: | 19 |
| Descripción | Delete An Actor 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/actors/deleteById?method=true&id=2000 |



Revisamos la salida:

