

## Summary test report

**Author:** Igor Marchuk

**Data:** 23.12.2023

### Description:

In this document are described some defects that were found in the process of creating test automation framework for test task from SCB.

Software solution can be downloaded for authorized users from GitHub by link:

<https://github.com/atelit/SCBTask>

Detailed test report could be found in directory /report or by link

<https://github.com/atelit/SCBTask/tree/main/report>

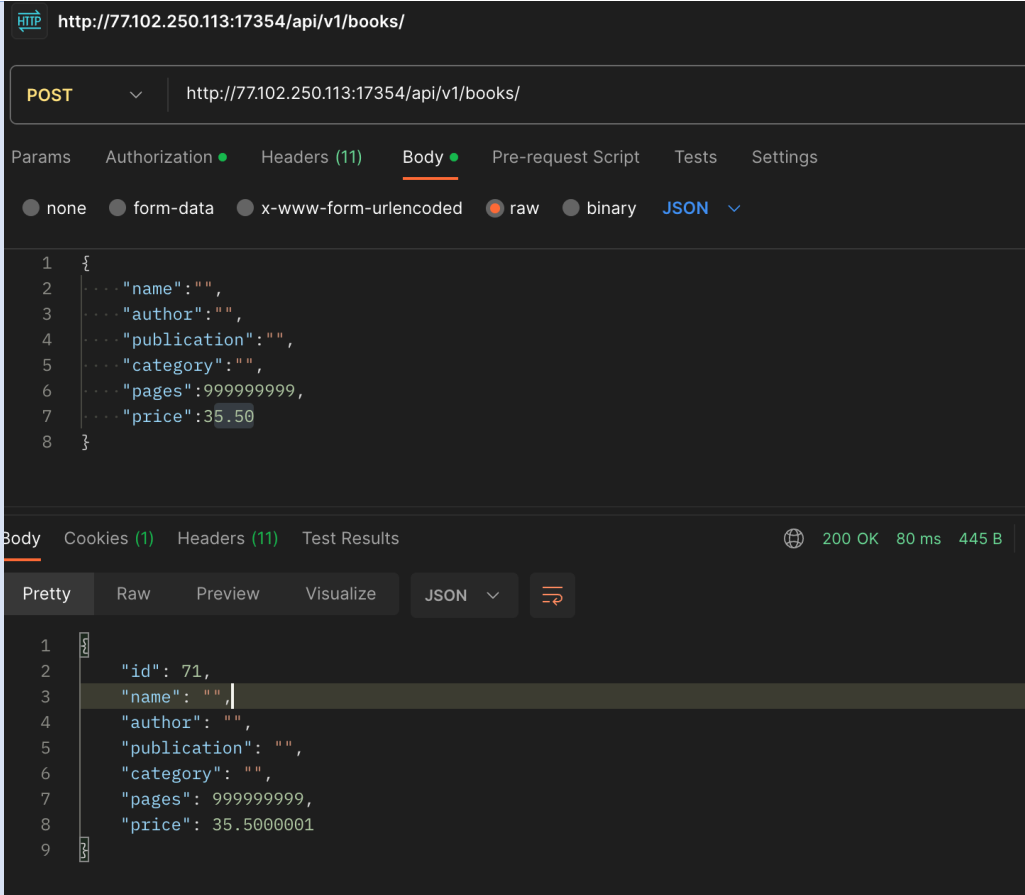
### Recommendations to dev team:

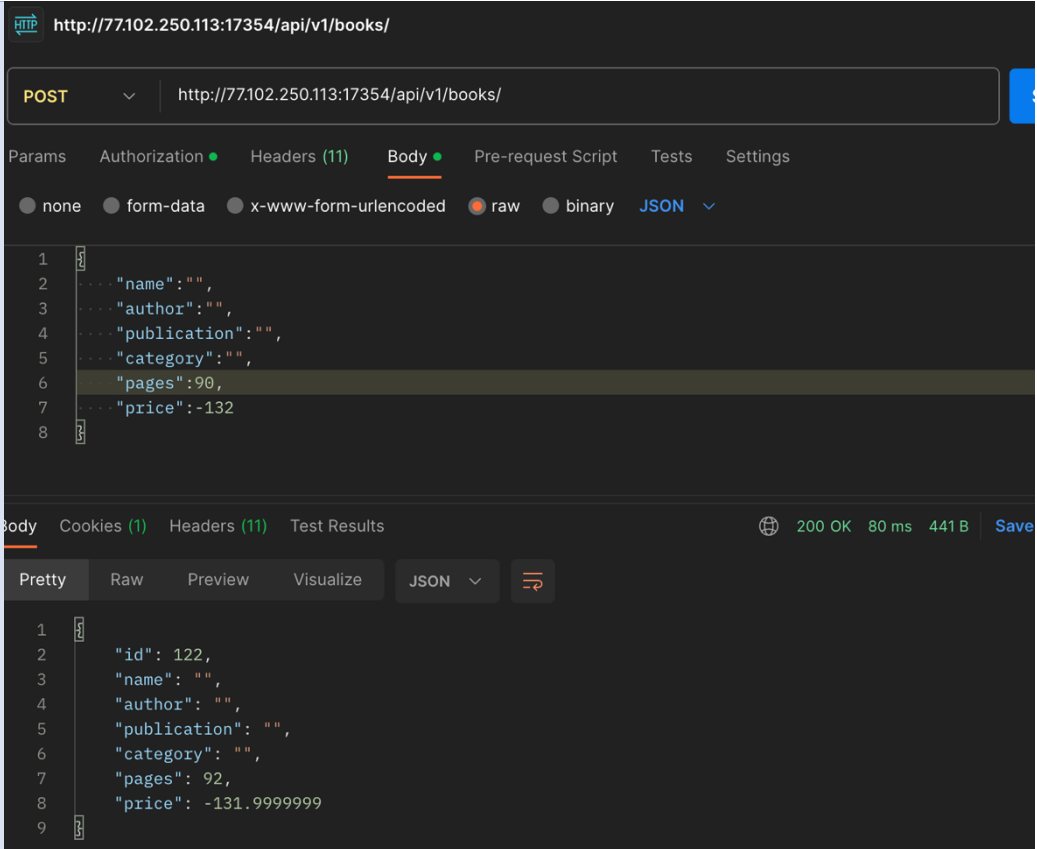
1. Please carefully read requirements before push update to server
2. Make system more user-friendly: tell user why and what he can't do in some period. Example: when user trying to delete non-exist record, he receives 500 error, but not error message that record is already deleted or non-exist
3. Create user-friendly documentation to see what fields are mandated for some type of request. For example: in Json-object for POST all fields are mandated or for PUT method only id is mandated.

### Recommendation to author of the task:

Next time please add some sample of reports that you expect to see from the Candidate.

Best regard,  
Igor Marchuk

|                          |   |
|--------------------------|---|
| Defect ID                | DEF01   |
| Defect Title             | System added 0.0000001 to price of book in random period of time  |
| Defect Description       | <p><b>Preconditions:</b></p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p><b>Steps to reproduce:</b></p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to http://77.102.250.113:17354/api/v1/books/</p> <ol style="list-style-type: none"> <li>2. Body of request is:</li> </ol> <pre>curl --request POST \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"name":"Refactoring: Improving the Design of Existing Code","author":"Martin Fowler","publication":"Addison-Wesley Professional","category":"Programming","pages":448,"price":35.50}' \   http://77.102.250.113:17354/api/v1/books</pre> <ol style="list-style-type: none"> <li>3. Repeat steps 2-3 until price will be 35.5000001</li> </ol> |
| Expected result          | User update book and price saved as expected  |
| Actual result            | System added 0.0000001 to price of book in random period of time  |
| Evidence/attachm<br>ents |  <p>The screenshot shows a Postman interface for a POST request to the URL <code>http://77.102.250.113:17354/api/v1/books/</code>. The request body is a JSON object with the following fields: <code>"name": "", "author": "", "publication": "", "category": "", "pages": 999999999, "price": 35.50</code>. The response status is <code>200 OK</code> with a response time of <code>80 ms</code> and a body size of <code>445 B</code>. The response body is a JSON object with the following fields: <code>"id": 71, "name": "", "author": "", "publication": "", "category": "", "pages": 999999999, "price": 35.5000001</code>.</p>  |
| Severity                 | High  |
| Priority                 | High  |
| Module affected          | backend   |
| Environment              | Test env  |
| Reported by              | Igor Marchuk  |
| Reported on              | 23.12.2023  |
| Status                   | New   |
| Assigned to              | Developer ABC   |

|                          |  |
|--------------------------|--|
| Defect ID                | DEF02  |
| Defect Title             | System added 2 to pages of book in random period of time   |
| Defect Description       | <p><b>Preconditions:</b></p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p><b>Steps to reproduce:</b></p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to <code>http://77.102.250.113:17354/api/v1/books/</code></p> <ol style="list-style-type: none"> <li>2. Send POST request. Body of request is:</li> </ol> <pre>curl --request POST \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"name":"Refactoring: Improving the Design of Existing Code","author":"Martin Fowler","publication":"Addison-Wesley Professional","category":"Programming","pages":448,"price":35.50}' \   http://77.102.250.113:17354/api/v1/books</pre> <ol style="list-style-type: none"> <li>3. Repeat steps 2-3 until pages will be 450</li> </ol> |
| Expected result          | User update book and page saved as expected  |
| Actual result            | System added 2 to pages of book in random period of time   |
| Evidence/attachm<br>ents |  <p>The screenshot shows a Postman interface for a POST request to <code>http://77.102.250.113:17354/api/v1/books/</code>. The request body is a JSON object: <code>{ "name": "", "author": "", "publication": "", "category": "", "pages": 90, "price": -132 }</code>. The response is a 200 OK status with a JSON body: <code>{ "id": 122, "name": "", "author": "", "publication": "", "category": "", "pages": 92, "price": -131.9999999 }</code>.</p>  |
| Severity                 | High   |
| Priority                 | High   |
| Module affected          | backend  |
| Environment              | Test env   |
| Reported by              | Igor Marchuk   |
| Reported on              | 23.12.2023   |
| Status                   | New  |
| Assigned to              | Developer ABC  |

|                                  |  |
|----------------------------------|--|
| <b>Defect ID</b>                 | DEF03  |
| <b>Defect Title</b>              | User can't update book's pages with PUT method   |
| <b>Defect Description</b>        | <p>Preconditions:</p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p>Steps to reproduce:</p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to http://77.102.250.113:17354/api/v1/books/</p> <ol style="list-style-type: none"> <li>2. Send POST request. Body of request is:</li> </ol> <pre>curl --request POST \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"name":"Refactoring: Improving the Design of Existing Code","author":"Martin Fowler","publication":"Addison-Wesley Professional","category":"Programming","pages":448,"price":35.50}' \   http://77.102.250.113:17354/api/v1/books</pre> <ol style="list-style-type: none"> <li>3. Send PUT request with new number of pages to update book that was created on step 2</li> <li>4. Request to Update book, example:</li> </ol> <pre>curl --request PUT \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"id":2,"name":"Test Driven Development: By Example","author":"Kent Beck","publication":"Addison-Wesley Professional","category":"Programming","pages":240,"price":29.26}' \   http://77.102.250.113:17354/api/v1/books/2</pre> |
| <b>Expected result</b>           | User update book and page saved as expected  |
| <b>Actual result</b>             | User can't update book's pages with PUT method   |
| <b>Evidence/attachm<br/>ents</b> | none   |
| <b>Severity</b>                  | High   |
| <b>Priority</b>                  | Normal (Standard)  |
| <b>Module affected</b>           | backend  |
| <b>Environment</b>               | Test env   |
| <b>Reported by</b>               | Igor Marchuk   |
| <b>Reported on</b>               | 23.12.2023   |
| <b>Status</b>                    | New  |
| <b>Assigned to</b>               | Developer ABC  |

|                             |   |
|-----------------------------|---|
| <b>Defect ID</b>            | <b>DEF04</b>  |
| <b>Defect Title</b>         | Sometimes book's price is saves as 1.0E-7   |
| <b>Defect Description</b>   | <p>Preconditions:</p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p>Steps to reproduce:</p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to<br/>http://77.102.250.113:17354/api/v1/books/</p> <ol style="list-style-type: none"> <li>2. Send POST request. Body of request is:</li> </ol> <pre>curl --request POST \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"name":"Refactoring: Improving the Design of Existing Code","author":"Martin Fowler","publication":"Addison-Wesley Professional","category":"Programming","pages":448,"price":35.50}' \   http://77.102.250.113:17354/api/v1/books</pre> <ol style="list-style-type: none"> <li>3. Repeat steps 2-3 until pages will be price will be saved as 1.0E-7</li> </ol> |
| <b>Expected result</b>      | Recently created book will be deleted every time  |
| <b>Actual result</b>        | Sometimes book's price is saves as 1.0E-7   |
| <b>Evidence/attachments</b> |   |
| <b>Severity</b>             | High  |
| <b>Priority</b>             | High  |
| <b>Module affected</b>      | backend   |
| <b>Environment</b>          | Test env  |
| <b>Reported by</b>          | Igor Marchuk  |
| <b>Reported on</b>          | 23.12.2023  |
| <b>Status</b>               | New   |
| <b>Assigned to</b>          | Developer ABC   |

|                             |  |
|-----------------------------|--|
| <b>Defect ID</b>            | <b>DEF05</b>   |
| <b>Defect Title</b>         | Method DELETE /books sometimes returns 500   |
| <b>Defect Description</b>   | <p>Preconditions:</p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p>Steps to reproduce:</p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to <a href="http://77.102.250.113:17354/api/v1/books/">http://77.102.250.113:17354/api/v1/books/</a></p> <ol style="list-style-type: none"> <li>2. Send POST request. Body of request is:</li> </ol> <pre>curl --request POST \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"name":"Refactoring: Improving the Design of Existing Code","author":"Martin Fowler","publication":"Addison-Wesley Professional","category":"Programming","pages":448,"price":35.50}' \   http://77.102.250.113:17354/api/v1/books</pre> <ol style="list-style-type: none"> <li>3. In response copy id</li> <li>4. Create Delete method to delete recently created book.</li> <li>5. Request is:</li> </ol> <pre>curl -X DELETE -u &lt;user&gt;:&lt;password&gt; <a href="http://77.102.250.113:17354/api/v1/books/1">http://77.102.250.113:17354/api/v1/books/1</a></pre> <ol style="list-style-type: none"> <li>6. Repeat steps 2-4 until response will be 500</li> </ol> |
| <b>Expected result</b>      | Recently created book will be deleted every time   |
| <b>Actual result</b>        | Method DELETE /books sometimes returns 500   |
| <b>Evidence/attachments</b> |  |
| <b>Severity</b>             | High   |
| <b>Priority</b>             | High   |
| <b>Module affected</b>      | backend  |
| <b>Environment</b>          | Test env   |
| <b>Reported by</b>          | Igor Marchuk   |
| <b>Reported on</b>          | 23.12.2023   |
| <b>Status</b>               | New  |
| <b>Assigned to</b>          | Developer ABC  |

|                             |  |
|-----------------------------|--|
| <b>Defect ID</b>            | <b>DEF06</b>   |
| <b>Defect Title</b>         | User can save book with empty fields   |
| <b>Defect Description</b>   | <p>Preconditions:</p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p>Steps to reproduce:</p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to http://77.102.250.113:17354/api/v1/books/</p> <ol style="list-style-type: none"> <li>2. Send POST request. Body of request is:</li> </ol> <pre>curl --request POST \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"name":"","author":"","publication":"","category":"","pages":0,"price":0}' \   http://77.102.250.113:17354/api/v1/books</pre> |
| <b>Expected result</b>      | System will return error message “System can’t save the empty record” or something like that   |
| <b>Actual result</b>        | Books saved successfully   |
| <b>Evidence/attachments</b> |  |
| <b>Severity</b>             | High   |
| <b>Priority</b>             | High   |
| <b>Module affected</b>      | backend  |
| <b>Environment</b>          | Test env   |
| <b>Reported by</b>          | Igor Marchuk   |
| <b>Reported on</b>          | 23.12.2023   |
| <b>Status</b>               | New  |
| <b>Assigned to</b>          | Developer ABC  |

|                             |   |
|-----------------------------|---|
| <b>Defect ID</b>            | <b>DEF07</b>  |
| <b>Defect Title</b>         | User can't update book's ID with PUT method   |
| <b>Defect Description</b>   | <p>Preconditions:</p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p>Steps to reproduce:</p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to http://77.102.250.113:17354/api/v1/books/</p> <ol style="list-style-type: none"> <li>2. Send POST request. Body of request is:</li> </ol> <pre>curl --request POST \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"name":"Refactoring: Improving the Design of Existing Code","author":"Martin Fowler","publication":"Addison-Wesley Professional","category":"Programming","pages":448,"price":35.50}' \   http://77.102.250.113:17354/api/v1/books</pre> <ol style="list-style-type: none"> <li>3. Send PUT request with ID of book to update book that was created on step 2</li> <li>4. Request to Update book, example:</li> </ol> <pre>curl --request PUT \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"id":2,"name":"Test Driven Development: By Example","author":"Kent Beck","publication":"Addison-Wesley Professional","category":"Programming","pages":240,"price":29.26}' \   http://77.102.250.113:17354/api/v1/books/2</pre> |
| <b>Expected result</b>      | User updated book's ID successfully   |
| <b>Actual result</b>        | User can't update book's ID with PUT method   |
| <b>Evidence/attachments</b> | none  |
| <b>Severity</b>             | High  |
| <b>Priority</b>             | Normal (Standard)   |
| <b>Module affected</b>      | backend   |
| <b>Environment</b>          | Test env  |
| <b>Reported by</b>          | Igor Marchuk  |
| <b>Reported on</b>          | 23.12.2023  |
| <b>Status</b>               | New   |
| <b>Assigned to</b>          | Developer ABC   |



|                             |  |
|-----------------------------|--|
| <b>Defect ID</b>            | <b>DEF08</b>   |
| <b>Defect Title</b>         | User see 500 error when trying to update non-exist book  |
| <b>Defect Description</b>   | <p>Preconditions:</p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p>Steps to reproduce:</p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to <a href="http://77.102.250.113:17354/api/v1/books/">http://77.102.250.113:17354/api/v1/books/</a></p><br><ol style="list-style-type: none"> <li>2. Send PUT request with ID 999999 to update book that was created on step 2</li> <li>3. Request to Update book, example:</li> </ol> <pre>curl --request PUT \ --user &lt;user&gt;:&lt;password&gt; \ --header "Content-Type: application/json" \ --data '{"id":999999,"name":"Test Driven Development: By Example","author":"Kent Beck","publication":"Addison-Wesley Professional","category":"Programming","pages":240,"price":29.26}' \ http://77.102.250.113:17354/api/v1/books/2</pre> |
| <b>Expected result</b>      | User see 404 error   |
| <b>Actual result</b>        | User see 500 error   |
| <b>Evidence/attachments</b> | none   |
| <b>Severity</b>             | Normal   |
| <b>Priority</b>             | Normal (Standard)  |
| <b>Module affected</b>      | backend  |
| <b>Environment</b>          | Test env   |
| <b>Reported by</b>          | Igor Marchuk   |
| <b>Reported on</b>          | 23.12.2023   |
| <b>Status</b>               | New  |
| <b>Assigned to</b>          | Developer ABC  |

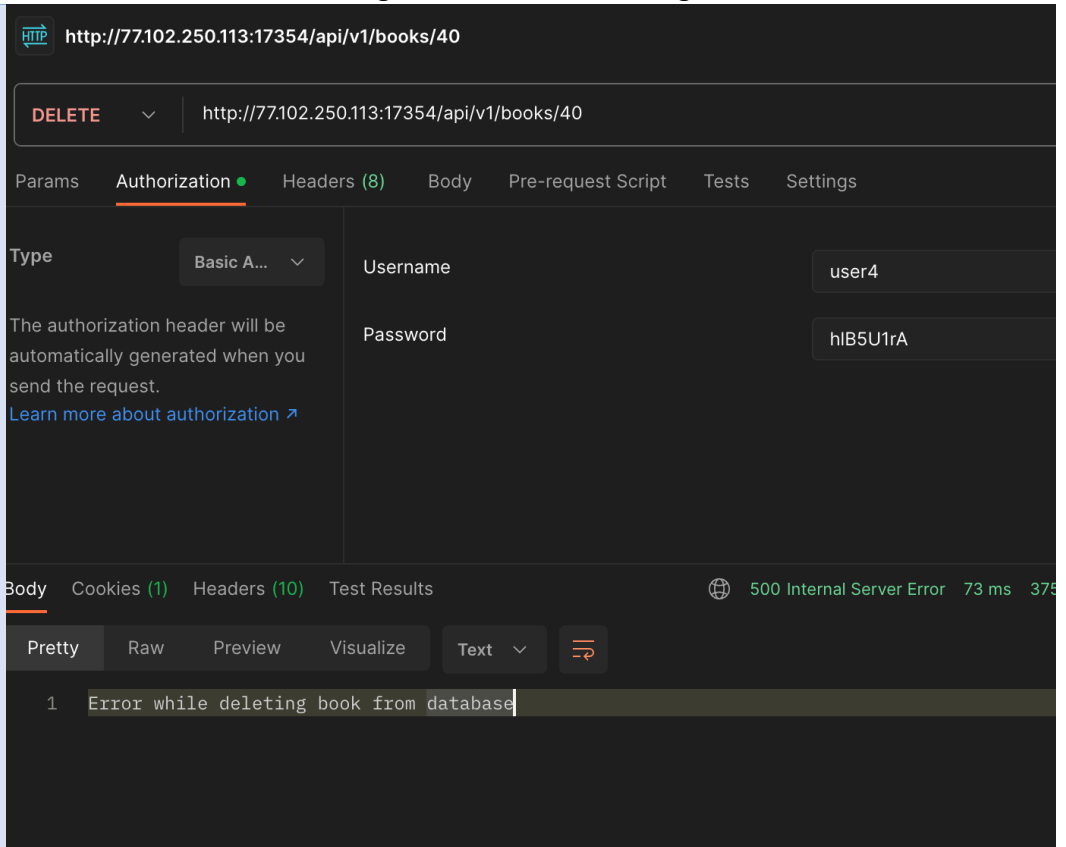
|                             |   |
|-----------------------------|---|
| <b>Defect ID</b>            | <b>DEF09</b>  |
| <b>Defect Title</b>         | User see 500 error when trying to delete non-exist book   |
| <b>Defect Description</b>   | <p>Preconditions:</p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p>Steps to reproduce:</p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to <a href="http://77.102.250.113:17354/api/v1/books/">http://77.102.250.113:17354/api/v1/books/</a></p><br><ol style="list-style-type: none"> <li>2. Send DELETE request with ID 99999 to update book that was created on step 2</li> <li>3. Request to Delete book, example:<br/> <pre>curl -x DELETE -u &lt;user&gt;:&lt;password&gt; <a href="http://77.102.250.113:17354/api/v1/books/99999">http://77.102.250.113:17354/api/v1/books/99999</a></pre> </li> </ol> |
| <b>Expected result</b>      | User see 404 error  |
| <b>Actual result</b>        | User see 500 error  |
| <b>Evidence/attachments</b> | none  |
| <b>Severity</b>             | Normal  |
| <b>Priority</b>             | Normal (Standard)   |
| <b>Module affected</b>      | backend   |
| <b>Environment</b>          | Test env  |
| <b>Reported by</b>          | Igor Marchuk  |
| <b>Reported on</b>          | 23.12.2023  |
| <b>Status</b>               | New   |
| <b>Assigned to</b>          | Developer ABC   |

|                             |   |
|-----------------------------|---|
| <b>Defect ID</b>            | <b>DEF10</b>  |
| <b>Defect Title</b>         | User can save negative value for price and pages  |
| <b>Defect Description</b>   | <p>Preconditions:</p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p>Steps to reproduce:</p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to http://77.102.250.113:17354/api/v1/books/</p> <ol style="list-style-type: none"> <li>2. Send POST request. Body of request is:</li> </ol> <pre>curl --request POST \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"name":"Refactoring: Improving the Design of Existing Code","author":"Martin Fowler","publication":"Addison-Wesley Professional","category":"Programming","pages":-448,"price":-35.50}' \   http://77.102.250.113:17354/api/v1/books</pre> |
| <b>Expected result</b>      | System will return an error message that price and pages can't be negative  |
| <b>Actual result</b>        | User see response code 200 and book is saved successfully   |
| <b>Evidence/attachments</b> | none  |
| <b>Severity</b>             | High  |
| <b>Priority</b>             | Normal (Standard)   |
| <b>Module affected</b>      | backend   |
| <b>Environment</b>          | Test env  |
| <b>Reported by</b>          | Igor Marchuk  |
| <b>Reported on</b>          | 23.12.2023  |
| <b>Status</b>               | New   |
| <b>Assigned to</b>          | Developer ABC   |

|                             |   |
|-----------------------------|---|
| <b>Defect ID</b>            | <b>DEF11</b>  |
| <b>Defect Title</b>         | User can save negative value for price and pages  |
| <b>Defect Description</b>   | <p>Preconditions:</p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p>Steps to reproduce:</p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to http://77.102.250.113:17354/api/v1/books/</p> <ol style="list-style-type: none"> <li>2. Send POST request. Body of request is:</li> </ol> <pre>curl --request POST \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"name":"Refactoring: Improving the Design of Existing Code","author":"Martin Fowler","publication":"Addison-Wesley Professional","category":"Programming","pages":448,"price":35.50}' \   http://77.102.250.113:17354/api/v1/books</pre> <ol style="list-style-type: none"> <li>3. Send PUT request with ID of book to update book that was created on step 2</li> <li>4. Request to Update book, example:</li> </ol> <pre>curl --request PUT \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"id":2,"name":"Test Driven Development: By Example","author":"Kent Beck","publication":"Addison-Wesley Professional","category":"Programming","pages":-240,"price":-29.26}' \   http://77.102.250.113:17354/api/v1/books/2</pre> |
| <b>Expected result</b>      | System will return an error message that price and pages can't be negative  |
| <b>Actual result</b>        | User see response code 200 and book is updated successfully   |
| <b>Evidence/attachments</b> | none  |
| <b>Severity</b>             | High  |
| <b>Priority</b>             | Normal (Standard)   |
| <b>Module affected</b>      | backend   |
| <b>Environment</b>          | Test env  |
| <b>Reported by</b>          | Igor Marchuk  |
| <b>Reported on</b>          | 23.12.2023  |
| <b>Status</b>               | New   |
| <b>Assigned to</b>          | Developer ABC   |

|                                  |  |
|----------------------------------|--|
| <b>Defect ID</b>                 | <b>DEF12</b>   |
| <b>Defect Title</b>              | User see 500 error when trying to save the book from Guinness World Record book  |
| <b>Defect Description</b>        | <p>Preconditions:</p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p>Steps to reproduce:</p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to <code>http://77.102.250.113:17354/api/v1/books/</code></p> <ol style="list-style-type: none"> <li>2. Generate random String with 26000 chars and replace the name of the book in request from step 3</li> <li>3. Send POST request. Body of request is:</li> </ol> <pre>curl --request POST \ --user &lt;user&gt;:&lt;password&gt; \ --header "Content-Type: application/json" \ --data '{"name":"Refactoring: Improving the Design of Existing Code","author":"Martin Fowler","publication":"Addison-Wesley Professional","category":"Programming","pages":448,"price":35.50}' \ http://77.102.250.113:17354/api/v1/books</pre> |
| <b>Expected result</b>           | System will return an error message name is too long or book will be saved without issues  |
| <b>Actual result</b>             | User see 500 error   |
| <b>Evidence/attachm<br/>ents</b> | <p>The <b>Guinness world record for the longest book title</b> belongs to Vityala Yethir book “The Historical Development of the Heart from Its Formation From ...” In contains 3,777 words and over 26,000 characters!</p>  |
| <b>Severity</b>                  | Low  |
| <b>Priority</b>                  | Low  |
| <b>Module affected</b>           | backend  |
| <b>Environment</b>               | Test env   |
| <b>Reported by</b>               | Igor Marchuk   |
| <b>Reported on</b>               | 23.12.2023   |
| <b>Status</b>                    | New  |
| <b>Assigned to</b>               | Developer ABC  |

|                             |   |
|-----------------------------|---|
| <b>Defect ID</b>            | <b>DEF13</b>  |
| <b>Defect Title</b>         | User see message "Book has been deleted successfully" instead of book's info  |
| <b>Defect Description</b>   | <p>Preconditions:</p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p>Steps to reproduce:</p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to <a href="http://77.102.250.113:17354/api/v1/books/">http://77.102.250.113:17354/api/v1/books/</a></p> <ol style="list-style-type: none"> <li>2. Send POST request. Body of request is:</li> </ol> <pre>curl --request POST \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"name":"Refactoring: Improving the Design of Existing Code","author":"Martin Fowler","publication":"Addison-Wesley Professional","category":"Programming","pages":448,"price":35.50}' \   http://77.102.250.113:17354/api/v1/books</pre> <ol style="list-style-type: none"> <li>3. In response copy id</li> <li>4. Create Delete method to delete recently created book.</li> <li>5. Request is:</li> </ol> <pre>curl -x DELETE -u &lt;user&gt;:&lt;password&gt; <a href="http://77.102.250.113:17354/api/v1/books/1">http://77.102.250.113:17354/api/v1/books/1</a></pre> |
| <b>Expected result</b>      | <ol style="list-style-type: none"> <li>1. Server response is 200</li> <li>2. User will book's info. Example:</li> </ol> <pre>{   "id": 2,    "name": "Test Driven Development: By Example",   "author": "Kent Beck",   "publication": "Addison-Wesley Professional",   "category": "Programming",   "pages": 240,   "price": 29.26 }</pre>  |
| <b>Actual result</b>        | <ol style="list-style-type: none"> <li>1. Server response is 200</li> <li>2. Book has been deleted successfully</li> </ol>  |
| <b>Evidence/attachments</b> | none  |
| <b>Severity</b>             | High  |
| <b>Priority</b>             | High  |
| <b>Module affected</b>      | backend   |
| <b>Environment</b>          | Test env  |
| <b>Reported by</b>          | Igor Marchuk  |
| <b>Reported on</b>          | 23.12.2023  |
| <b>Status</b>               | New   |
| <b>Assigned to</b>          | Developer ABC   |

|                       |  |
|-----------------------|--|
| Defect ID             | DEF14  |
| Defect Title          | Method DELETE /books sometimes returns message: "Error while deleting book from database"  |
| Defect Description    | <p>Preconditions:</p> <ul style="list-style-type: none"> <li>- User has valid credentials.</li> <li>- User has installed Curl or Postman</li> </ul> <p>Steps to reproduce:</p> <ol style="list-style-type: none"> <li>1. Open Postman or Curl</li> </ol> <p>Execute Post request to <a href="http://77.102.250.113:17354/api/v1/books/">http://77.102.250.113:17354/api/v1/books/</a></p> <ol style="list-style-type: none"> <li>2. Send POST request. Body of request is:</li> </ol> <pre>curl --request POST \   --user &lt;user&gt;:&lt;password&gt; \   --header "Content-Type: application/json" \   --data '{"name":"Refactoring: Improving the Design of Existing Code","author":"Martin Fowler","publication":"Addison-Wesley Professional","category":"Programming","pages":448,"price":35.50}' \   http://77.102.250.113:17354/api/v1/books</pre> <ol style="list-style-type: none"> <li>3. In response copy id</li> <li>4. Create Delete method to delete recently created book.</li> <li>5. Request is:</li> </ol> <pre>curl -X DELETE -u &lt;user&gt;:&lt;password&gt; <a href="http://77.102.250.113:17354/api/v1/books/1">http://77.102.250.113:17354/api/v1/books/1</a></pre> <ol style="list-style-type: none"> <li>6. Repeat steps 2-4 until response will be 500</li> </ol> |
| Expected result       | Recently created book will be deleted every time without errors  |
| Actual result         | <ol style="list-style-type: none"> <li>1. Method DELETE /books returns 500</li> <li>2. User see error message: "Error while deleting book from database"</li> </ol>  |
| Evidence/attach ments |  <p>The screenshot shows a Postman interface for a DELETE request to <a href="http://77.102.250.113:17354/api/v1/books/40">http://77.102.250.113:17354/api/v1/books/40</a>. The request is configured with Basic Authentication using the username 'user4' and password 'h1B5U1rA'. The response tab is selected, showing a '500 Internal Server Error' with the message 'Error while deleting book from database'.</p>  |
| Severity              | High   |
| Priority              | High   |

|                        |               |
|------------------------|---------------|
| <b>Module affected</b> | backend       |
| <b>Environment</b>     | Test env      |
| <b>Reported by</b>     | Igor Marchuk  |
| <b>Reported on</b>     | 23.12.2023    |
| <b>Status</b>          | New           |
| <b>Assigned to</b>     | Developer ABC |