Web Technology

Student Name: Bako Asaad

Student Number: 510944

Second attempt

Date: 1/8/2023

Table of Contents

[1. Introduction 3](#_Toc69399127)

[2. GET requests 4](#_Toc69399128)

[3. POST requests 5](#_Toc69399129)

[4. PUT requests 6](#_Toc69399130)

[5. DELETE requests 7](#_Toc69399131)

# Introduction

The diagram below is about a book store website that called Happy Reader and a user can register and bid on the book he or she want.

*Diagram

Description automatically generated with medium confidence*

# GET requests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GET** | **http://localhost:5000/user/{id}** | | | |
| get one user by id | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *\* required* | | Id | path | User id |
|
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 200  400  404 | OK the given id must be numeric  User with that id {} does not exist | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GET** | **http://localhost:5000/books/id** | | | |
| get one book by id | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *\* required* | | id \* | path | id of book to find |
|
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 200 | OK | |
| 404 | Book with that id {} does not exist | |
|  | | 400 | the given id must be numeric | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GET** | **http://localhost:5000/users** | | | |
| Gets all users that are registers | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *\* required* | | Id | query | Id of the bids |
| Name | query | Name of the bids |
| Start date | query | Start date of the bids |
| End date | query | End date of the bids |
|
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 200 | list of all the bids | |

# POST requests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **POST** | **htttp://localhost:5000/bids** | | | |
| Create a bid, the id of the bid is added automated. | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *\* required* | | Id | body | Id of the bid is automatically added. |
| UserId | Body | Id of the user who do the bidding |
| Username | Body | Name of the user who do the bid |
| BookId | Body | Id of the book that bids on it |
| Amount | Body | Amount of the bid the user do |
|
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 201 | OK | |
| 400 | All fields need to be filled to place a bid.(Bad request) | |
| 403 | your bid is lower than the start price for this auction(Forbidden). | |
| 403 | your bid is lower than the highest bid for this auction(Forbidden). | |
| 400 | you can not place a bid with a non exisiting ID.(Bad request) | |
| 400 | you can not place a bid on a book that does not exist.(Bad request) | |
|  |  | |
|

# PATCH requests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PATCH** | **htttp://localhost:5000/books/{id}** | | | |
| Use a specific id to update any information based on the id. | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *\* required* | | Starting price | Body | The price of the book to be modified terrace |
|
|
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 200 | The book has been updated, (OK) | |
|

# DELETE requests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DELETE** | **htttp://localhost:5000/books/{id}** | | | |
| Delete a specific book based on id | | | | |
|  | | | | |
| **Parameters:** | | **Name** | **Type** | **Description** |
| *\* required* | | Id \* | path | Id of book to delete |
|
| **Responses:** | | **Code** | **Description / example if successful** | |
|  | | 404 | No books were found with that id | |
| 200 | Book with id {} has been deleted successfully.(OK) | |
|  | | 404 | Id must be numeric | |