**WEB API Documentation**

**Contents**

Introduction ……………..………………………………………………………………………………………………………. 2

HTTPS Status Codes….……………………………………………………………………………………………………….. 2

Address……………………………………………………………………………………………………………………………… 2

Services……………………………………………………………………………………………………………………………… 2

Ping…………………………………………………………………………………………………………………………………... 2

Request……………………………………………………………………………………………………………………………. 2

Response…………………………………………………………………………………………………………………………. 2

Get all Books……………………………………………………………………………………………………………………… 3

Request……………………………………………………………………………………………………………………………. 3

Response…………………………………………………………………………………………………………………………. 3

Get a Book by ID………………………………………………………………………………………………………………… 4

Request……………………………………………………………………………………………………………………………. 4

Response…………………………………………………………………………………………………………………………. 4

Create a new Book…….………………………………………………………………………………………………………. 5

Request……………………………………………………………………………………………………………………………. 5

Response…………………………………………………………………………………………………………………………. 5

Update an existing Book……………………….………………..…………………………………………………………. 6

Request……………………………………………………………………………………………………………………………. 6

Response…………………………………………………………………………………………………………………………. 6

Delete an Existing Book by ID…………………………………………………………………………………………….. 7

Request……………………………………………………………………………………………………………………………. 2

Response…………………………………………………………………………………………………………………………. 7

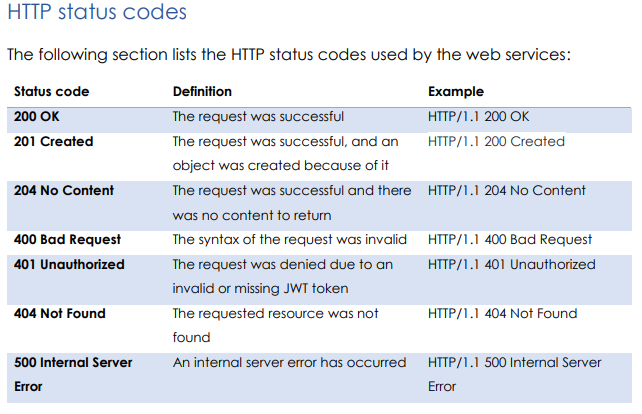
**Introduction**

The following document documents the web API exposed by the course project

application for COS310b Web API Security, Fall, 2018.

**HTTPS Status codes**

The following section lists the HTTP status codes used by the web services:



**Address**

The following is the path for accessing the web API:

<https://localhost:8445/api/>

**Services**

**Ping**

Pings the server to check whether it is available

**Request**

GET /ping HTTP/1.1

Authorization: Bearer

**Response**

PONG

The response indicates that the server is running

**Get all Books**

Retrieves all Books

**Request**

GET /v1/books HTTP/1.1

Authorization: Bearer <token>

**Response**

200 OK

Content-type: application/json

{

"title": String,

"author": String,

"genre": String,

"price": Double,

"id": Number

}

The response includes an array of book objects listed under a key called “books.” Each book has title, author, genre and price.

**Get a Book by ID**

Retrieves a Book by its ID

**Request**

GET /v1/books/{id} HTTP/1.1

Authorization: Bearer <token>

Where ID is a valid ID that exists in the Books repository. The ID must be greater than 0 and in the case such an item doesn’t exits a 400 Bad Request status will be returned.

**Response**

200 OK

Content-type: application/json

{

"title": String,

"author": String,

"genre": String,

"price": Double,

"id": Number

}

The response includes a single book object. Each book has title, author, genre and price.

**Create a new Book**

Creates a new Book

**Request**

POST /v1/books HTTP/1.1

Authorization: Bearer <token>

Content-type: application/json

{

"title": String,

"author": String,

"genre": String,

"price": Double

}

The payload must NOT contain ID or a 400 Bad Request status will be returned.

**Response**

201 Created

Content-type: application/json

{

"title": String,

"author": String,

"genre": String,

"price": Double,

"id": Number

}

The response includes a single book object. Each book has title, author, genre and price and an auto-generated ID.

**Update an existing Book**

Updates an existing Book

**Request**

PUT /v1/books/{id} HTTP/1.1

Authorization: Bearer <token>

Content-type: application/json

{

"title": String,

"author": String,

"genre": String,

"price": Double

}

Where ID is a valid ID that exists in the Books repository. The payload must NOT contain ID or a 400 Bad Request status will be returned.

**Response**

200 OK

Content-type: application/json

{

"title": String,

"author": String,

"genre": String,

"price": Double,

"id": Number

}

The response includes a single book object with updated fields.

**Delete a Book by ID**

Deletes an existing Book by ID

**Request**

DELETE /v1/books/{id} HTTP/1.1

Authorization: Bearer <token>

Where ID is a valid id of an existing book.

**Response**

204 No Content