

Documentatie Enterprise Web Development - Joris Van Duyse

- [TL;DR](#)
- [Requirements](#)
- [Installation](#)
- [Running in development versus running in production](#)
- [Screenshots](#)
 - [The navigation bar](#)
- [REST](#)
 - [Get request voor health van de server](#)
 - [Get request for author](#)
 - [Get request for a book](#)

TL;DR

1. This application runs online on isbin.qwict.com
2. Requirements
 1. java openjdk 17.0.5 or later
 2. Apache Maven 3.9.1 or later
 3. A MySQL server with an empty schema (with SSL for production)
3. To install it:
 1. clone
 2. create and fill in application.properties to `/src/main/resources`
 3. from the root build to jar with `maven install`
 4. from the root run the jar with `java -jar target/IsBin.jar`
 5. the application should start; go to localhost:9091

Requirements

1. java openjdk 17.0.5 or later
2. Apache Maven 3.9.1 or later
3. A MySQL server with an empty schema (with SSL for production)

Installation

1. Clone the repository to your device: ``

```
git clone https://github.com/JorisVanDuyseHogent/IsBin.git language-sh
```

2. Create the required `application.properties` file in `/src/main/resources/application.properties`, remember to replace all `{text}` with your information curly brackets included :)

```
# ----- Development settings ----- language-sh
spring.datasource.url=jdbc:mysql://localhost:3306/{your_database_name}?
serverTimezone=UTC
spring.datasource.username={your_database_username}
spring.datasource.password={your_database_password}
application.port={the_port_for_your_application}
application.env=development
spring.jpa.generate-ddl=true
spring.jpa.hibernate.ddl-auto=create-drop
spring.messages.basename=i18n/messages

# ----- Production settings -----
# ----- Uncomment these to run in production! -----
# ----- And remove the developer settings! -----
#spring.datasource.url=jdbc:mysql://{example.com}/{your_production_database_name}
#?useSSL=true&requireSSL=true&serverTimezone=UTC
#spring.datasource.username={your_production_database_username}
#spring.datasource.password={your_production_database_password}
#application.port={the_port_for_your_production_application}
#application.env=production
#spring.jpa.hibernate.ddl-auto=none
#spring.messages.basename=i18n/messages
```

3. Run the application from your favorite IDE. I will not explain this, because it is very different for each developer and operation system.
4. Build the application to jar with from the root of the git repository (this is where the `pom.xml` is located):

```
mvn install language-sh
```

5. Run the application also from the root with (If the database schema was already used, you will get foreign key constraint errors, but you can ignore these, will only happen with development settings):

```
java -jar target/IsBin.jar language-sh
```

Running in development versus running in production

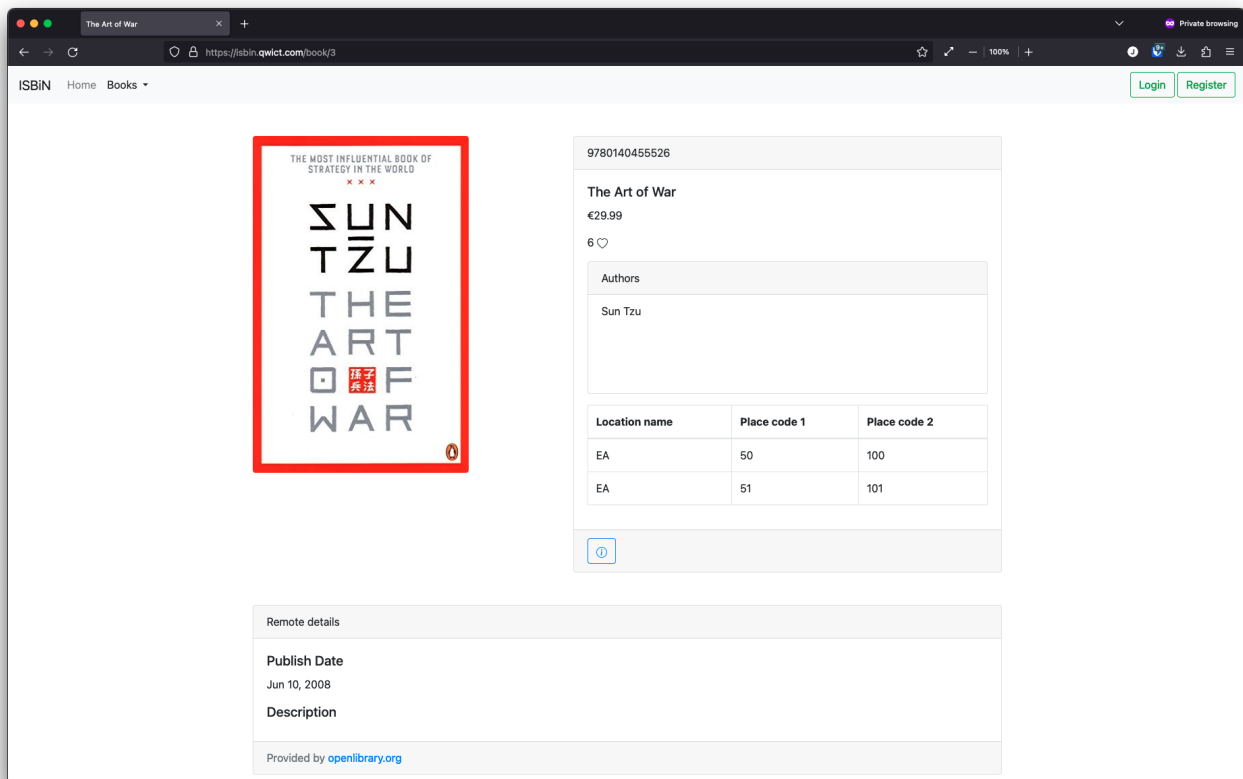
There is no big difference between running with the in development and running in production except for that in development, the database schema will be dropped after every restart and the database will be seeded with books and users.

Recommended for production: start the application one time in development; this will seed the database with some books and users. After that enable all the production settings in the application.properties file and remove the developer settings. Reinstall the application with maven to jar and run the jar again with java.

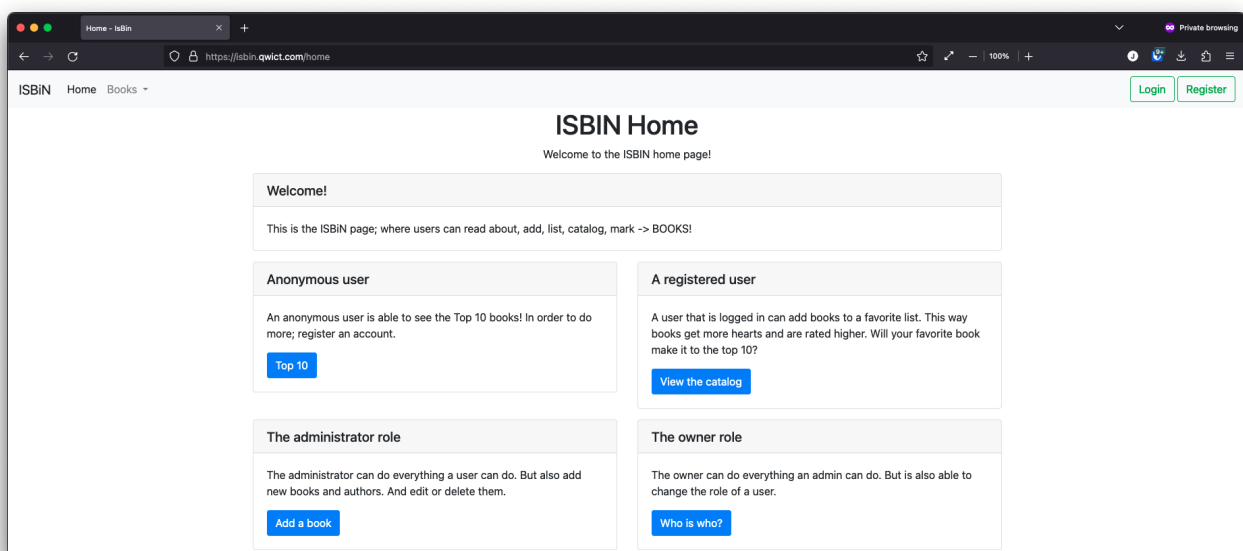
This way you will have a production server with some books, authors and users already in the database.

Screenshots

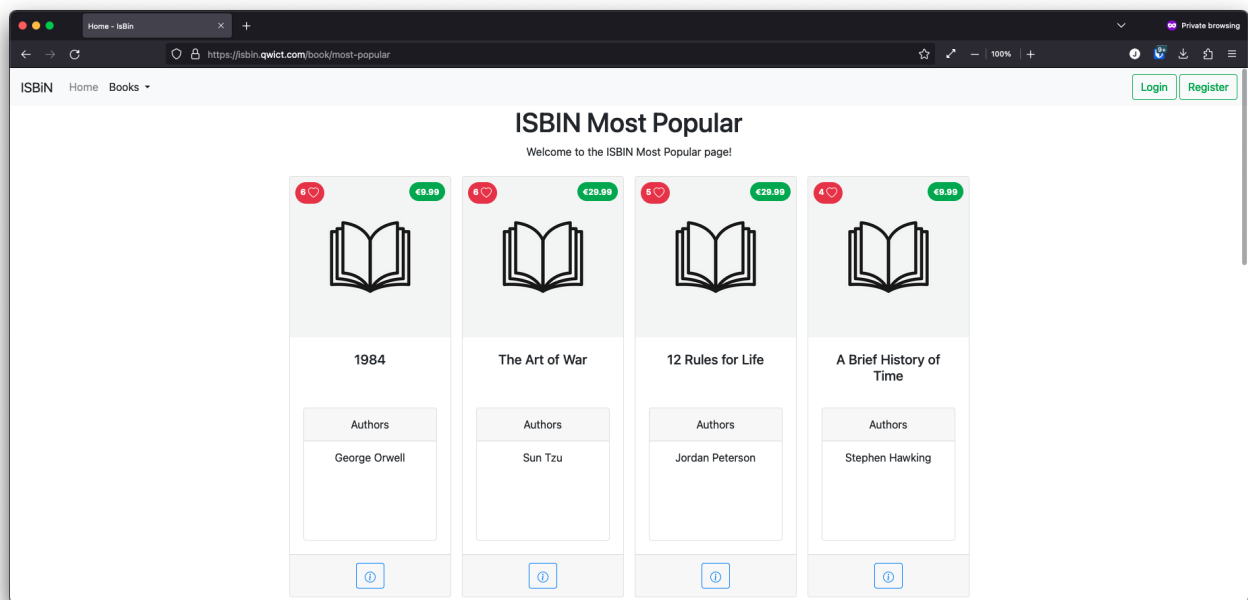
An example of the detailpage for a book (The Art of War).



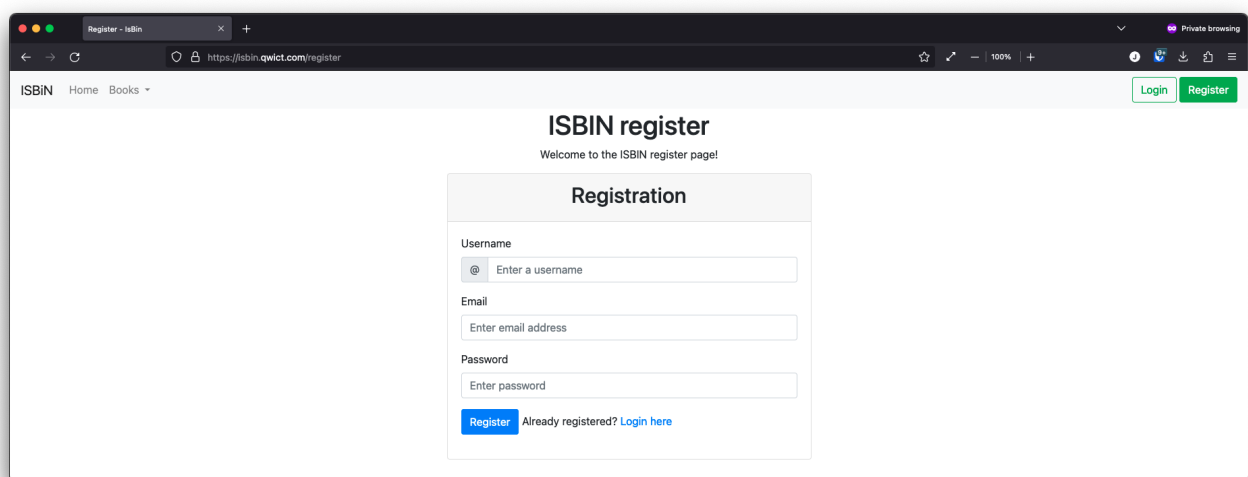
The IsBin home page with info about different roles.



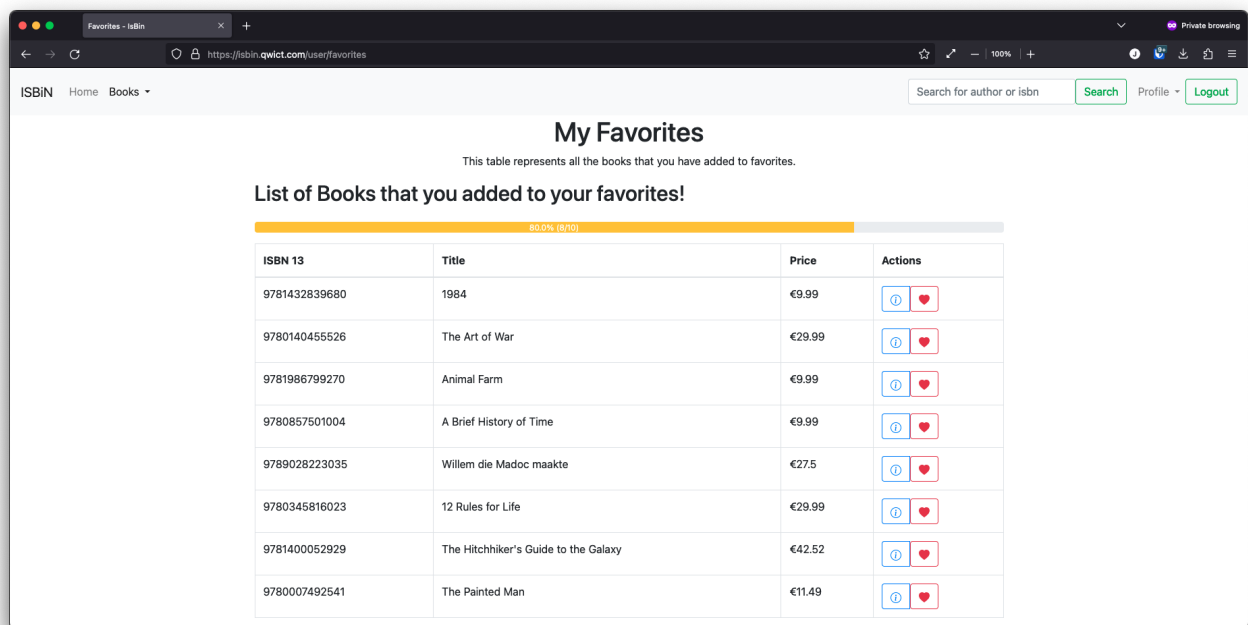
Most popular books ordered by number of "hearts".



The registration page; registration is required to view the catalog



The favorites page available for users



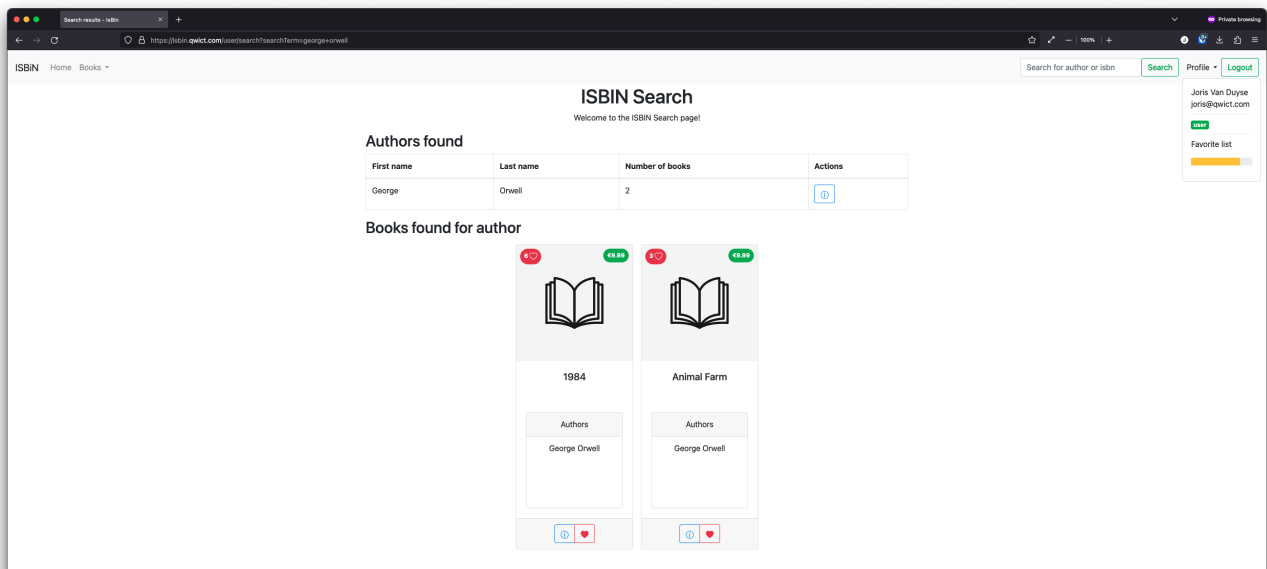
My Favorites

This table represents all the books that you have added to favorites.

List of Books that you added to your favorites!

ISBN 13	Title	Price	Actions
9781432839680	1984	€9.99	i ♥
9780140455526	The Art of War	€29.99	i ♥
9781986799270	Animal Farm	€9.99	i ♥
9780857501004	A Brief History of Time	€9.99	i ♥
9789028223035	Willem die Madoc maakte	€27.5	i ♥
9780345816023	12 Rules for Life	€29.99	i ♥
9781400052929	The Hitchhiker's Guide to the Galaxy	€42.52	i ♥
9780007492541	The Painted Man	€11.49	i ♥

The search page that supports author lookup by first and lastname, but also isbn lookup




ISBIN Search

Welcome to the ISBIN Search page!

Authors found

First name	Last name	Number of books	Actions
George	Orwell	2	i

Books found for author




1984

Authors

George Orwell

[i](#) [♥](#)



Animal Farm

Authors

George Orwell

[i](#) [♥](#)

The book form that allows administrators to add books to the catalog; in this case the administrator tried to submit a book without filling in any information.

ISBiN Home Books

Search for author or isbn Search Profile Logout

admin
admin@qwict.com
admin user
Favorite list

Book Form

Title Enter the book's title
Title should not be empty

ISBN13 Enter the book's ISBN13
ISBN cannot be empty

price

Author 1: First name Last name
The primary author must have a first and last name.

Author 2: First name Last name

Author 3: First name Last name

Location 1: Name 0 0
A book needs at least one location.

Location 2: Name 0 0

Location 3: Name 0 0

Submit

The navigation bar

The right side of the navigation bar with a normal user logged in.

Search for author or isbn Search Profile Logout

Joris Van Duyse
joris@qwict.com

user

Favorite list

The left side of the navigation bar with the owner logged in.

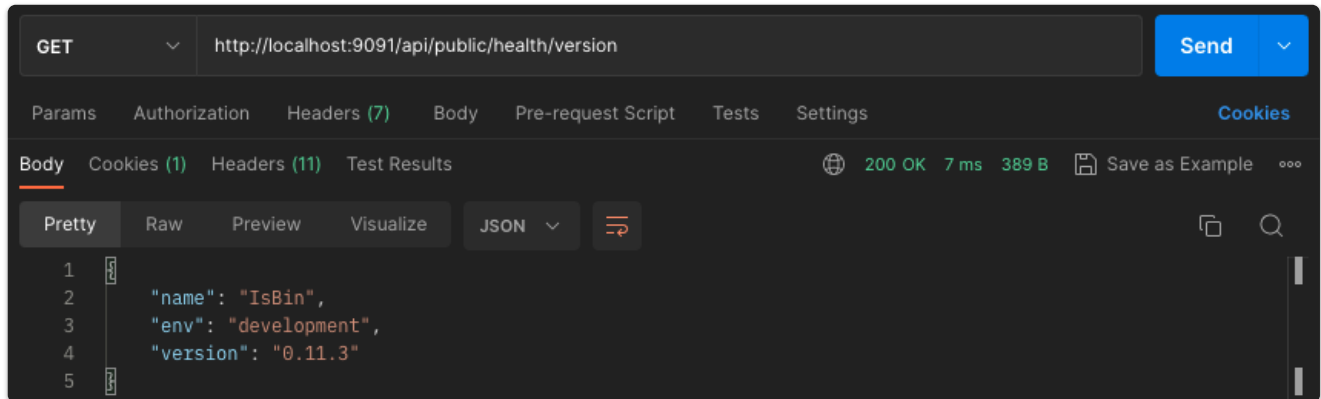
ISBiN Home Books Owner functions

- ☆ Most Popular
- ♥ My Favorites
- 📖 Catalog
- 📖 Add book

REST

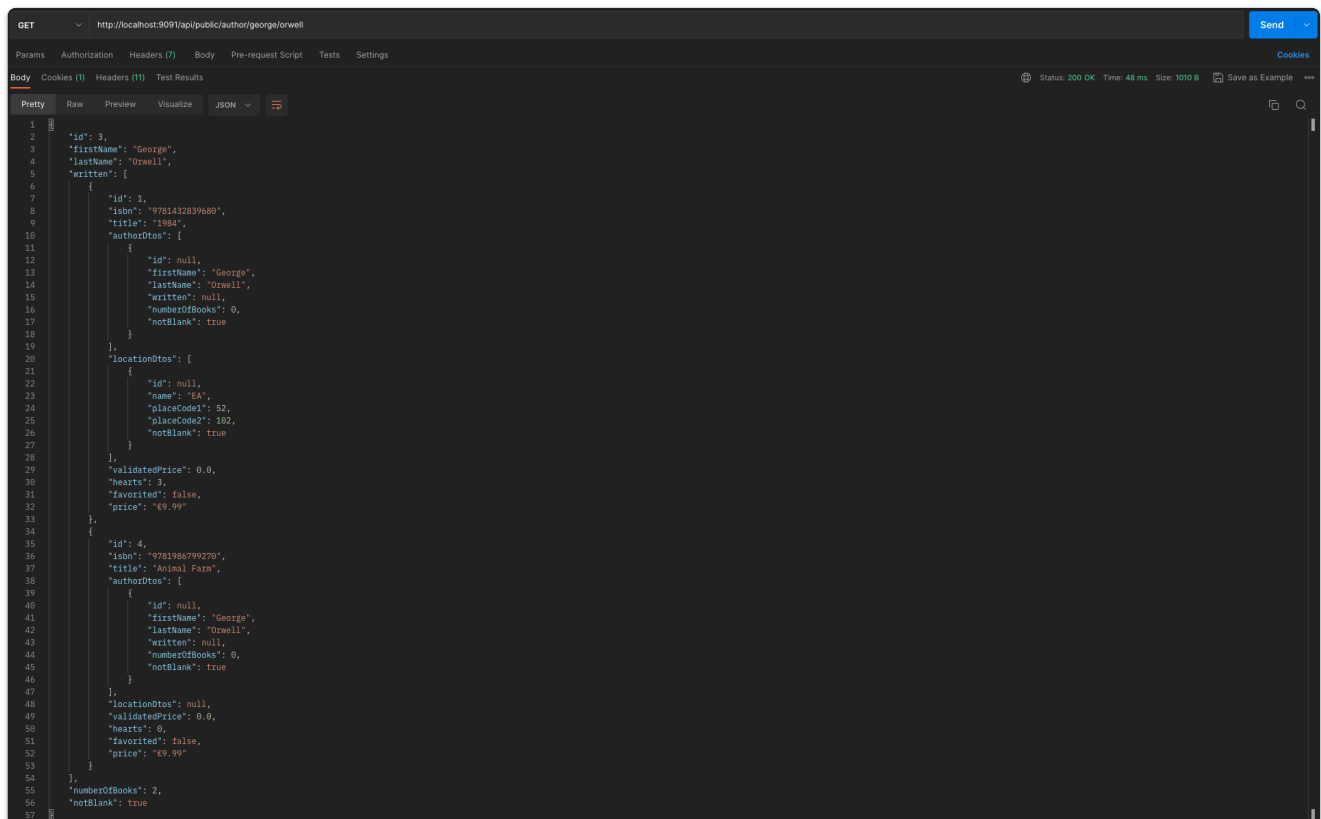
Get request voor health van de server

To get information about the spring boot application you can GET `http://localhost:9091/api/public/health/version`



Get request for author

Call GET `http://localhost:9091/api/public/author/george/orwell` to look up a specific author by entering their first and last name; This will return a JSON



Get request for a book

By calling GET `http://localhost:9091/api/public/book/{isbn}` it is possible to look up a book by its isbn number.

The screenshot shows a REST client interface with a GET request to `http://localhost:9091/api/public/book/9780140455526`. The response is a JSON object with the following structure:

```
1 {
2   "id": 3,
3   "isbn": "9780140455526",
4   "title": "The Art of War",
5   "authorDtos": [
6     {
7       "id": null,
8       "firstName": "Sun",
9       "lastName": "Tzu",
10      "written": null,
11      "numberOfBooks": 0,
12      "notBlank": true
13    }
14  ],
15  "locationDtos": [
16    {
17      "id": null,
18      "name": "EA",
19      "placeCode1": 50,
20      "placeCode2": 100,
21      "notBlank": true
22    },
23    {
24      "id": null,
25      "name": "EA",
26      "placeCode1": 51,
27      "placeCode2": 101,
28      "notBlank": true
29    }
30  ],
31  "validatedPrice": 0.0,
32  "hearts": 3,
33  "favorited": false,
34  "price": "€29.99"
35 }
```