

OpenLibrary

By Iliass Bensalah

Previously we had

Local Java project

Use of a Database

No use of Apache Tomcat Server



Now we need

Web based Project

Use of Database

Use of Apache Tomcat Server



Difficulties

So we just used a Tomcat server, right?

No:

Tomcat isn't compatible with the previous JDBC driver

A good part of the project had to be rewritten

Solution: A new Project from scratch



So Why a Library?

Simple

Comprehensive



Challenges

User Roles (user / admin)

Optimal DB queries

Use JSON Format for the API (only in search Function)



The Homepage

[Home](#)

[login](#) [register](#)

Open Library

all



Find

Search Function

Open Library

all



Find

Title: cats

Author: nyan nyan

ISBN: 1231231234

Available

Title: good cats

Author: nyan

ISBN: 01230123

Available

User View

[Home](#)

[items](#) [logout](#)

Open Library

all



Find

Users borrowed Items

Open Library

all



Find

Title: cats

Author: nyan nyan

ISBN: 1231231234

BIID: 27

Borrow Date: 2019-08-04 01:16:58

Title: cats

Author: nyan nyan

ISBN: 1231231234

Admin View

[Home](#)

[borrow](#)

[return](#)

[manage](#)

[items](#)

[logout](#)

Open Library

all



Find

Borrow And Return

Borrow

e.g. mail@example.com

bookitem id

Borrow

Return Book

bookitem

Return

Borrow and Return

Are only available to Admin because:

- A Library worker does it
- Self Borrow/Return operations are done at terminals with admin privileges



Search Function 1

Splitting the keywords into single keywords for LIKE command

```
String[] splitKeyword = keyword.split(" ", 0);  
keyword = "";  
for (String a : splitKeyword)  
    keyword += "%" + a;  
ArrayList<Book> list = BookDAO.searchByAuthor(keyword + "%");
```



Search Function 2

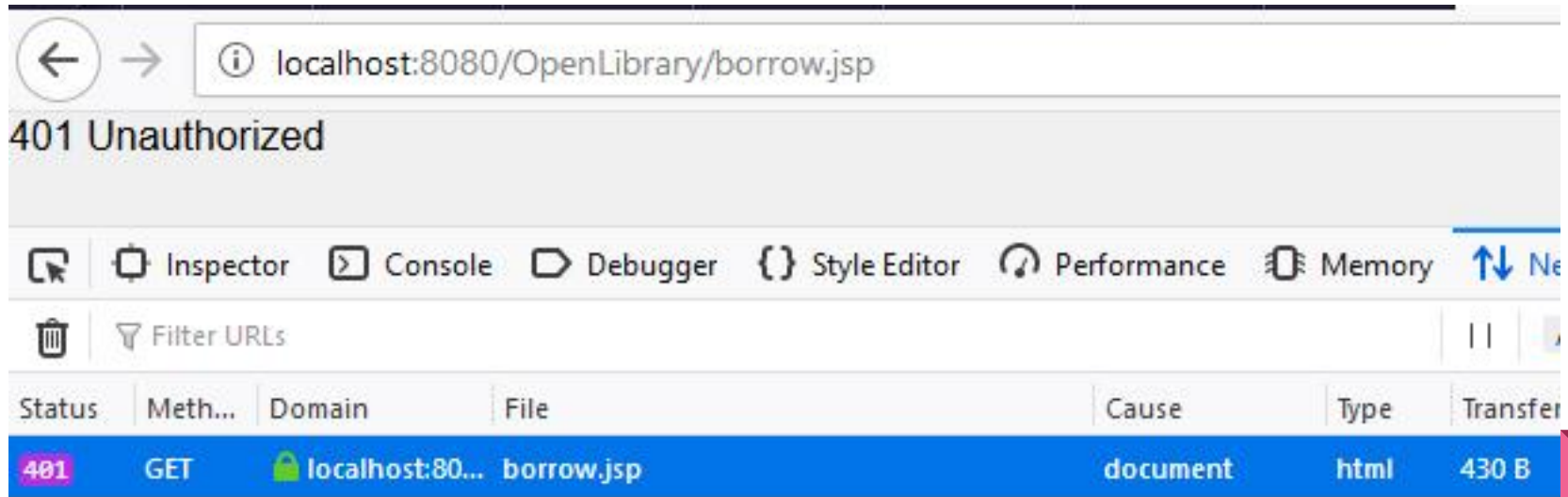
Getting the availability in the same search query with MAX command

```
SELECT * FROM `books` JOIN  
( SELECT bookitems.*, MAX(bookitems.available)  
FROM bookitems GROUP BY bookitems.book_id )  
as t ON `books`.id = t.book_id  
where books.ISBN LIKE 'keyword'  
OR books.title LIKE 'keyword'  
OR books.author LIKE 'keyword'
```



Security Checks

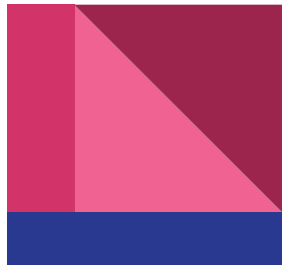
Manually entering when not logged in as admin /borrow.jsp



REST + JSON

GET /api/search.jsp?keyword={{keyword}}&search_by=all

```
[{"id":27,"title":"cats","author":"nyan nyan","ISBN":"1231231234","available":true},  
{"id":30,"title":"good cats","author":"nyan","ISBN":"01230123","available":true},  
{"id":31,"title":"bad cats","author":"nyan","ISBN":"31233123","available":true},  
{"id":32,"title":"cats and dogs","author":"woof  
woof","ISBN":"987654321","available":true}]
```



REST + JSON

Lesson learned:

- Drag .jar into WEB-INF/lib
- Don't install the usual way



Conclusion

We spent more time trying to install drivers/libraries as necessary

Next time may be use Gradle/Maven

After the project structure is complete, it's only copy and paste

Gson is nice... if you know how to install it

