

Master 1 Informatique

Parcours MLDS

Rapport de projet

Programmation Web

Programmation Distribuée

Mohammed Abdelhadi BOUDJEMAI

N° étudiant : 21912904 (MLDS)

Professeur : Mr CHAROUX Benoit

Année universitaire : 

Table Des Matières

- Introduction
- Fonctionnalité général de l'application
- Les outils utilisés
- Architecture général de l'application
- diagramme classes UML et Schéma des tables MySQL
- Les réponse http du Web service REST
- Docker
- code source lien GitHub

Introduction :

Le but de ce projet est de faire une application web qui permet de consulter des livres de 4 catégories différents : langage de programmation, développement web, Administration et de base de données.

Le but de ce projet était aussi d'appliquer les technologies du web qu'on a appris en cours notamment : les micro services, docker , Kubernetes ect .

Fonctionnalité général de l'application :

Cette application permet d'afficher ces livres selon leur catégorie : par exemple :

Web development : Python for SAS user, deep learning with JavaScript ect.

Programming : introduction JaKarta EE CDI, Modern C ect.

Databases : Pro T-SQL 2019, Practical Oracle Cloud Infrastructure ect.

Administration : Professional Outlook 2007 Programming ect.

Online BookStore

Search

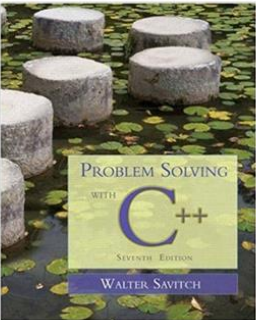
Home About Services Contact

Web Development

Programming

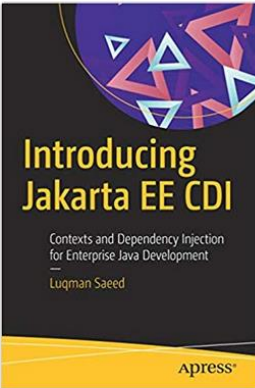
Databases

Administration



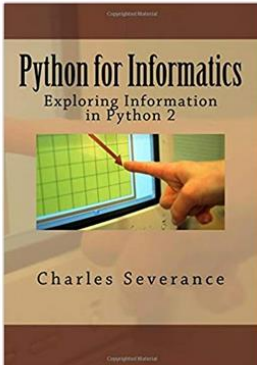
Problem Solving with C++
€400.00
Walter Savitch's Problem Sol...

Add Card



Introducing Jakarta EE CDI
€500.00
Discover the Jakarta EE Cont...

Add Card



Python for Informatics
€600.00
This book is designed to intr...

Add Card

Recherche des livres par catégorie :

Online BookStore

Search books

Search

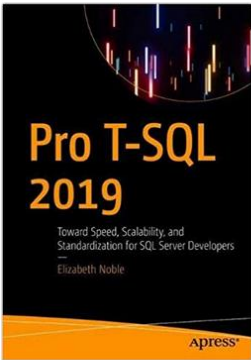
Home About Services Contact

Web Development

Programming

Databases

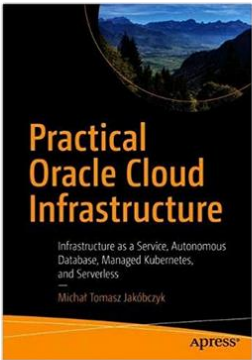
Administration



Pro T-SQL 2019
Toward Speed, Scalability, and Standardization for SQL Server Developers
— Elizabeth Noble
Apress

Pro T-SQL 2019
€400.00
Design and write simple and...

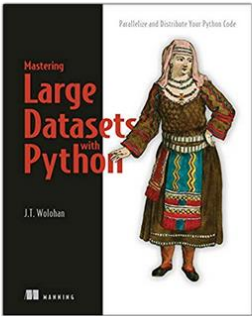
Add Card



Practical Oracle Cloud Infrastructure
Infrastructure as a Service, Autonomous Database, Managed Kubernetes, and Serverless
— Michał Tomasz Jakóbczyk
Apress

Practical Oracle Cloud Infrastructure
€500.00
Use this fast-paced and com...

Add Card



Mastering Large Datasets with Python
Parallelize and Distribute Your Python Code
— J.I. Wehlan
No Starch Press

Mastering Large Datasets with Python
€600.00
Modern data science solutio...

Add Card

En cliquant sur le livre pour le consulter :

Online BookStore

Search books

Search

Home About Services Contact

Web Development

Programming

Databases

Administration



Pro T-SQL 2019
Toward Speed, Scalability, and Standardization for SQL Server Developers
— Elizabeth Noble
Apress

Pro T-SQL 2019

Design and write simple and efficient T-SQL code in SQL Server 2019 and beyond. Writing T-SQL that pulls back correct results can be challenging.

Price **€400.00**

Qty. Available : 100

[← Back](#) [Add to Cart](#)

Recherche des livre par mot clé exemple java :

Online BookStore

java

Search

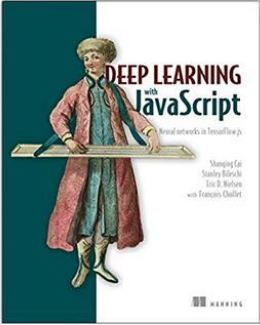
HomeAboutServicesContact

Web Development

Programming

Databases

Administration

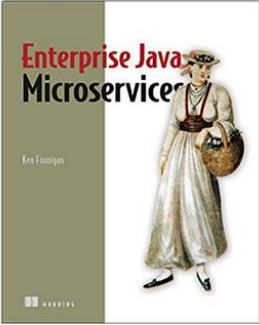


Deep Learning with JavaScript

€700.00

Deep learning has transform...

Add Card



Enterprise Java Microservices

€600.00

Enterprise Java Microservice...

Add Card

Pro Java Clustering and Scalability

€600.00

Build clustered and scalable ...

Add Card

Si le livre n'existe pas :

localhost:4200/search/javyyy

Online BookStore

javyyy

Search

Hor

Web Development

Programming

Databases

Administration

No books found

Si le lien http est erroné :

localhost:4200/searchz

Online BookStore

Search

Home

404

The page you are looking for was not found.

[Back to Home](#)

Les outils utilisé dans cette application :

Coté backend : Spring boot, java 8, maven, JPA, Hibernate (Spring tool suits 3)

Base de donnée : MySql 8

Coté frontend : Angular 9, Bootstrap (Visual code)

L'approche microservice et des conteurs sur docker

Architecture du projet :

On a choisi l'architecture micro service, on a 1 micro service Spring boot que je l'ai embarqué dans un container

et le service frontend sur un container aussi et la base de donnée mySql sur un autre container :

tous les container partagent la même network bookapp :

image explique l'architecture :

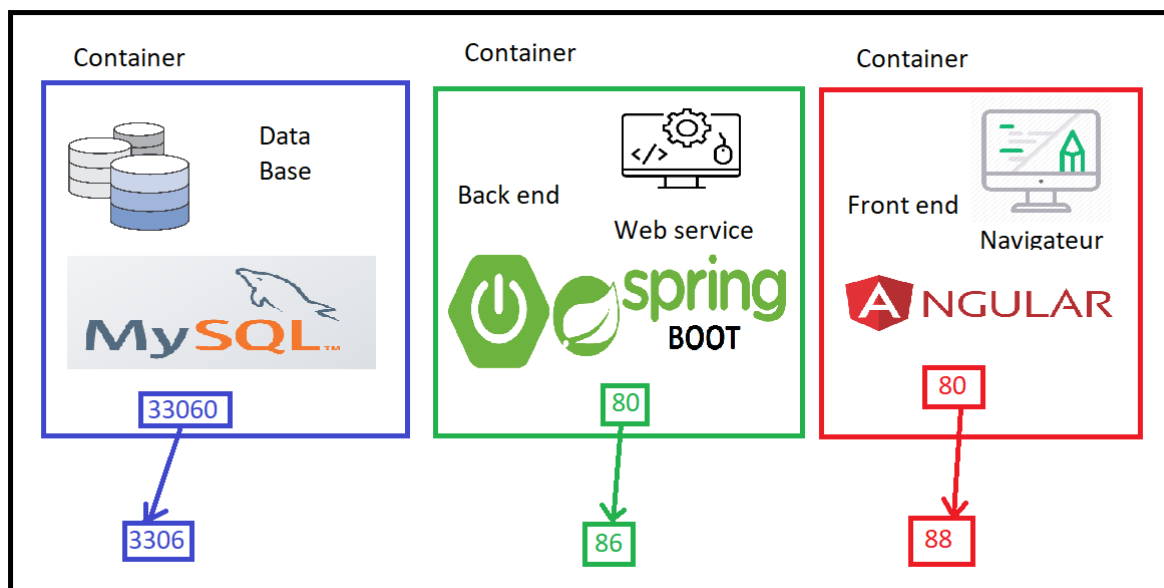
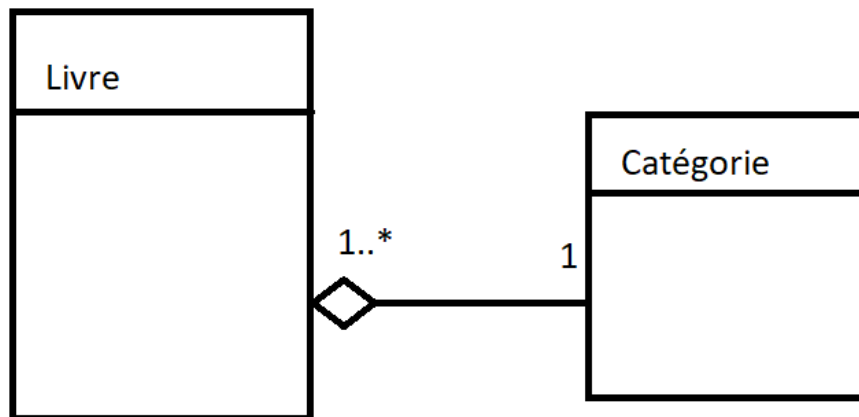


Diagramme classes UML et les tables MySQL :



Les tables de la base de donnée :

```
Database changed
mysql> show tables;
+-----+
| Tables_in_book-store-dev |
+-----+
| tbl_book                  |
| tbl_category              |
+-----+
2 rows in set (0.00 sec)

mysql>
```



```
mysql> describe tbl_book;
```

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
sku	varchar(255)	YES		NULL	
name	varchar(255)	YES		NULL	
description	varchar(255)	YES		NULL	
unit_price	decimal(13,2)	YES		NULL	
image_url	varchar(255)	YES		NULL	
active	bit(1)	YES		b'1'	
units_in_stock	int(11)	YES		NULL	
date_created	datetime	YES		NULL	
last_updated	datetime	YES		NULL	
category_id	bigint(20)	NO	MUL	NULL	

```
11 rows in set (0.00 sec)
```

```
mysql> describe tbl_category;
```

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
category_name	varchar(255)	YES		NULL	

```
2 rows in set (0.00 sec)
```

Chaque catégorie contient plusieurs livres et chaque livre a une seule catégorie.

Les réponse http du Web service REST : (Postman) :

Récupération des catégories :

The screenshot shows the Postman interface with a GET request to `http://localhost:8082/api/v1/book-category`. The response is displayed in JSON format, showing two book categories: "Web Development" and "Programming".

```
1 {
2   "_embedded": {
3     "bookCategory": [
4       {
5         "id": 1,
6         "categoryName": "Web Development",
7         "_links": {
8           "self": {
9             "href": "http://localhost:8082/api/v1/book-category/1"
10          },
11         "bookCategory": {
12           "href": "http://localhost:8082/api/v1/book-category/1"
13         },
14         "book": {
15           "href": "http://localhost:8082/api/v1/book-category/1/book"
16         }
17       },
18     ],
19     {
20       "id": 2,
21       "categoryName": "Programming",
22       "_links": {
23         "self": {
24           "href": "http://localhost:8082/api/v1/book-category/2"
25         },
26       },
27       "bookCategory": {
28         "href": "http://localhost:8082/api/v1/book-category/2"
29       },
30       "book": {
31         "href": "http://localhost:8082/api/v1/book-category/2/book"
32       }
33     }
34   }
35 }
```

R cup ration des livres :

GET

http://localhost:8082/api/v1/books

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body

Cookies

Headers (8)

Test Results

Status: 2

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

```
{
  "_embedded": {
    "books": [
      {
        "id": 1,
        "sku": "webdevelopment-100",
        "name": "Python for SAS Users",
        "description": "Business users familiar with Base SAS programming can now learn Python by example. You will learn via examples that map SAS programming constructs and coc
        "unitPrice": 600.00,
        "imageUrl": "assets/images/webdevelopment/webdevelopment-100.jpg",
        "active": true,
        "unitsInStock": 100,
        "createdOn": "2020-04-16T02:08:24.000+0000",
        "updatedOn": null,
        "_links": {
          "self": {
            "href": "http://localhost:8082/api/v1/books/1"
          },
          "book": {
            "href": "http://localhost:8082/api/v1/books/1"
          },
          "category": {
            "href": "http://localhost:8082/api/v1/books/1/category"
          }
        }
      },
      {
        "id": 2,
        "sku": "webdevelopment-101",
        "name": "Deep Learning with JavaScript",
        "description": "Deep learning has transformed the fields of computer vision, image processing, and natural language applications.",
        "unitPrice": 700.00,
        "imageUrl": "assets/images/webdevelopment/webdevelopment-101.jpg",
```

LaunchpadGET http://localhost:8082/api/v1GET http://localhost:8082/restfull/w...DEL http://localhost:8082/rest

Untitled Request

GEThttp://localhost:8082/api/v1

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettings

Query Params

KEY	VALUE
Key	Value

BodyCookiesHeaders (8)Test Results

PrettyRawPreviewVisualizeJSON

```
1 {
2   "_links": {
3     "books": {
4       "href": "http://localhost:8082/api/v1/books{?page,size,sort}",
5       "templated": true
6     },
7     "bookCateogry": {
8       "href": "http://localhost:8082/api/v1/book-category{?page,size,sort}",
9       "templated": true
10    },
11    "profile": {
12      "href": "http://localhost:8082/api/v1/profile"
13    }
14  }
15 }
```

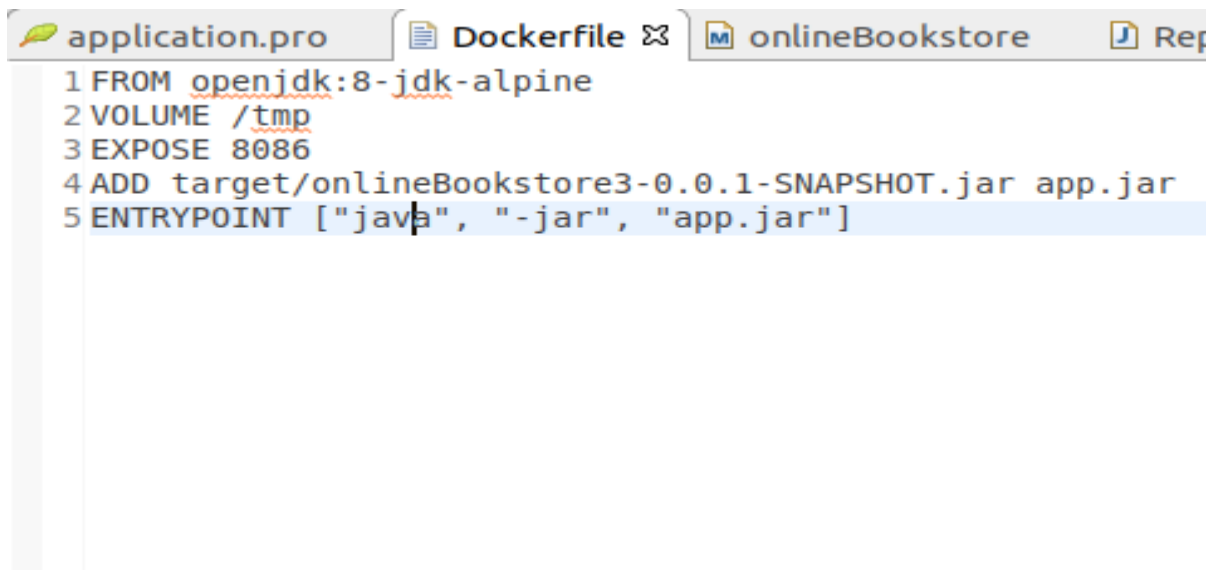
Docker :

Création des images :

- Pour Spring boot :

1/ creation du dockerfile :

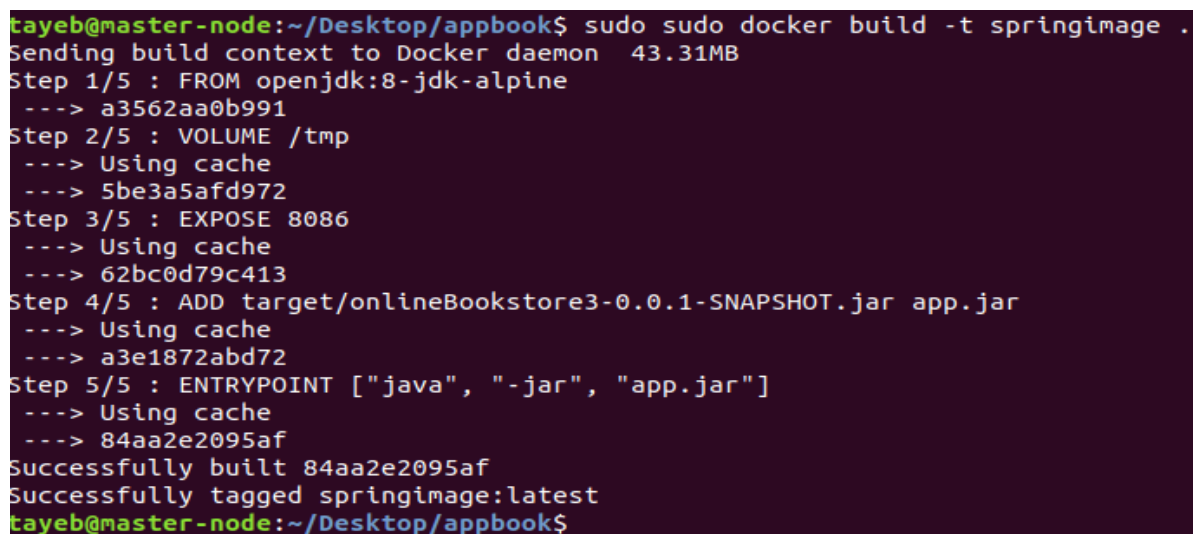
Les Dockerfiles sont des fichiers qui permettent de construire une image Docker adaptée à nos besoins



```
1 FROM openjdk:8-jdk-alpine
2 VOLUME /tmp
3 EXPOSE 8086
4 ADD target/onlineBookstore3-0.0.1-SNAPSHOT.jar app.jar
5 ENTRYPOINT ["java", "-jar", "app.jar"]
```

2/ generation du jar : mvn clean install

3/ generation de l'image :



```
tayeb@master-node:~/Desktop/appbook$ sudo docker build -t springimage .
Sending build context to Docker daemon 43.31MB
Step 1/5 : FROM openjdk:8-jdk-alpine
--> a3562aa0b991
Step 2/5 : VOLUME /tmp
--> Using cache
--> 5be3a5afd972
Step 3/5 : EXPOSE 8086
--> Using cache
--> 62bc0d79c413
Step 4/5 : ADD target/onlineBookstore3-0.0.1-SNAPSHOT.jar app.jar
--> Using cache
--> a3e1872abd72
Step 5/5 : ENTRYPOINT ["java", "-jar", "app.jar"]
--> Using cache
--> 84aa2e2095af
Successfully built 84aa2e2095af
Successfully tagged springimage:latest
tayeb@master-node:~/Desktop/appbook$
```

- Pour mysql :

1/ On recupère une image mysql : docker pull mysql

2/ création de la base :

```
tayeb@master-node:~/Desktop/appbook$ sudo docker run --name mybddapps3 -e MYSQL_ROOT_PASSWORD=password -e MYSQL_DATABASE=dev-book-store -e MYSQL_USER=root -e MYSQL_PASSWORD=password -d mysql:5.7
479bf8f55ea437d8fd1925bf7d88a0794350200897f2e621ed65e91ec136167d
tayeb@master-node:~/Desktop/appbook$
```

3/ la lisaion entre spring et mysql :

```
tayeb@master-node:~/Desktop/appbook$ sudo docker run --network bookapp -p 8086:8086 --name linkwithspring --link mybddapps3:mysql -d springimage
d312abac74b20a0f6f74f8709cd1fa4c2e2a82db9918966fd186b282444b7302
tayeb@master-node:~/Desktop/appbook$
```

- Pour Angular :

Dockerfile :

```
Dockerfile X
Dockerfile > ...
1  # stage 1 build image
2  FROM node:latest as node
3  WORKDIR /app
4  COPY . .
5  RUN npm install
6  RUN npm run build --prod
7  #RUN chmod 775 app/node_modules/.bin/ng
8
9  #RUN chmod 775 app/node_modules/.bin/ng.cmd
10 # stage 2 run the app nginx optimize the runing
11 FROM nginx:alpine
12 COPY --from=node /app/dist/angular-bookstore /usr/share/nginx/html
```

Création de l'image de Angular :

```
tayeb@master-node:~/Downloads/angular-bookstore$ sudo docker build --rm -f "Dockerfile" -t angularbookstore:v2 "."
[sudo] password for tayeb:
Sending build context to Docker daemon 448.6MB
Step 1/7 : FROM node:latest as node
--> a5a6a9c32877
Step 2/7 : WORKDIR /app
--> Using cache
--> ad2ec25ed1a5
Step 3/7 : COPY . .
--> Using cache
--> d98172648150
Step 4/7 : RUN npm install
--> Using cache
--> 8af8600c9a65
Step 5/7 : RUN npm run build --prod
--> Using cache
--> 543d818df376
Step 6/7 : FROM nginx:alpine
--> 89ec9da68213
Step 7/7 : COPY --from=node /app/dist/angular-bookstore /usr/share/nginx/html
--> Using cache
--> fb69e655c6c2
Successfully built fb69e655c6c2
Successfully tagged angularbookstore:v2
tayeb@master-node:~/Downloads/angular-bookstore$
```

On va maintenant faire tourner les containers créés :

sudo docker start -a nom_container :

mysql :

```
tayeb@master-node:~/Desktop/appbook$ sudo docker start -a mybddapps2
2020-05-01 20:55:00+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.29-1debian10 started.
2020-05-01 20:55:00+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2020-05-01 20:55:00+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.29-1debian10 started.
2020-05-01T20:55:00.450614Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option.
2020-05-01T20:55:00.451554Z 0 [Note] mysqld (mysqld 5.7.29) starting as process 1 ...
2020-05-01T20:55:00.453937Z 0 [Note] InnoDB: PUNCH HOLE support available
2020-05-01T20:55:00.454160Z 0 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
2020-05-01T20:55:00.454251Z 0 [Note] InnoDB: Uses event mutexes
2020-05-01T20:55:00.454306Z 0 [Note] InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
2020-05-01T20:55:00.454401Z 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
2020-05-01T20:55:00.454457Z 0 [Note] InnoDB: Using Linux native AIO
2020-05-01T20:55:00.454641Z 0 [Note] InnoDB: Number of pools: 1
2020-05-01T20:55:00.454767Z 0 [Note] InnoDB: Using CPU crc32 instructions
2020-05-01T20:55:00.457583Z 0 [Note] InnoDB: Initializing buffer pool, total size = 128M, instances = 1, chunk size = 128M
2020-05-01T20:55:00.462673Z 0 [Note] InnoDB: Completed initialization of buffer pool
2020-05-01T20:55:00.464049Z 0 [Note] InnoDB: If the mysqld execution user is authorized, page cleaner thread priority can be changed. See the
2020-05-01T20:55:00.475204Z 0 [Note] InnoDB: Highest supported file format is Barracuda.
2020-05-01T20:55:00.491577Z 0 [Note] InnoDB: Creating shared tablespace for temporary tables
2020-05-01T20:55:00.493529Z 0 [Note] InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file full; Please wait ...
2020-05-01T20:55:00.523746Z 0 [Note] InnoDB: File './ibtmp1' size is now 12 MB.
2020-05-01T20:55:00.524234Z 0 [Note] InnoDB: 96 redo rollback segment(s) found. 96 redo rollback segment(s) are active.
2020-05-01T20:55:00.524242Z 0 [Note] InnoDB: 32 non-redo rollback segment(s) are active.
2020-05-01T20:55:00.524429Z 0 [Note] InnoDB: Waiting for purge to start
2020-05-01T20:55:00.574833Z 0 [Note] InnoDB: 5.7.29 started; log sequence number 13420303
2020-05-01T20:55:00.575922Z 0 [Note] Plugin 'FEDERATED' is disabled.
2020-05-01T20:55:00.577859Z 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
2020-05-01T20:55:00.578143Z 0 [Note] InnoDB: Buffer pool(s) load completed at 200501 20:55:00
2020-05-01T20:55:00.579503Z 0 [Note] Found ca.pem, server-cert.pem and server-key.pem in data directory. Trying to enable SSL support using them
2020-05-01T20:55:00.580623Z 0 [Note] Skipping generation of SSL certificates as certificate files are present in data directory.
2020-05-01T20:55:00.581069Z 0 [Warning] CA certificate ca.pem is self signed.
2020-05-01T20:55:00.581598Z 0 [Note] Skipping generation of RSA key pair as key files are present in data directory.
2020-05-01T20:55:00.581993Z 0 [Note] Server hostname (bind-address): '*'; port: 3306
2020-05-01T20:55:00.583608Z 0 [Note] IPv6 is available.
2020-05-01T20:55:00.583677Z 0 [Note] - '::' resolves to '::';
2020-05-01T20:55:00.583829Z 0 [Note] Server socket created on IP: '::'.
2020-05-01T20:55:00.587216Z 0 [Warning] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS
2020-05-01T20:55:00.598218Z 0 [Note] Event Scheduler: Loaded 0 events
2020-05-01T20:55:00.598477Z 0 [Note] mysqld: ready for connections.
Version: '5.7.29' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server (GPL)
2020-05-01T21:00:17.188000Z 2 [Note] Aborted connection 2 to db: 'book-store-dev' user: 'tayeb' host: '172.17.0.4' (Got an error reading comm
```

Spring :

```
BOOT-INF/lib/jakarta.xml.bind-api-2.3.3.jar/, jar:file:/app.jar!/BOOT-INF/lib/spring-core-5.2.5.RELEASE.jar!/, jar:file:/app.jar!/BOOT-INF/lib/spring-jcl-5.2.5.RELEASE.jar!/, jar:file:/app.jar!/BOOT-INF/lib/spring-boot-2.2.6.RELEASE.jar!/, jar:file:/app.jar!/BOOT-INF/lib/spring-boot-autoconfigure-2.2.6.RELEASE.jar!/]

=====
:: Spring Boot :: (v2.2.6.RELEASE)

2020-05-01 22:13:34.454 INFO 1 --- [           main] e.b.o.OnlineBookstoreApplication : Starting OnlineBookstoreApplication v0.0.1-SNAPSHOT on 2d712736cb77
/)
2020-05-01 22:13:34.463 INFO 1 --- [           main] e.b.o.OnlineBookstoreApplication : No active profile set, falling back to default profiles: default
2020-05-01 22:13:34.464 DEBUG 1 --- [           main] o.s.boot.SpringApplication : Loading source class ecole.bookStore.onlineBookStore.OnlineBookStore
2020-05-01 22:13:34.715 DEBUG 1 --- [           main] o.s.b.c.c.ConfigFileApplicationListener : Loaded config file 'jar:file:/app.jar!/BOOT-INF/classes!/applicatio
n.properties'
2020-05-01 22:13:34.717 DEBUG 1 --- [           main] ConfigServletWebServerApplicationContext : Refreshing org.springframework.boot.web.servlet.context.AnnotationC
onfigServletWebServerApplicationContext
2020-05-01 22:13:36.760 INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2020-05-01 22:13:36.876 DEBUG 1 --- [           main] o.s.b.a.AutoConfigurationPackages : @EnableAutoConfiguration was declared on a class in the package 'ecole
.bookStore'
2020-05-01 22:13:37.144 INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 330ms. Found 2 JPA repo
sitories.
2020-05-01 22:13:40.260 DEBUG 1 --- [           main] .s.b.w.e.t.TomcatServletWebServerFactory : Code archive: /app.jar
2020-05-01 22:13:40.264 DEBUG 1 --- [           main] .s.b.w.e.t.TomcatServletWebServerFactory : Code archive: /app.jar
2020-05-01 22:13:40.265 DEBUG 1 --- [           main] .s.b.w.e.t.TomcatServletWebServerFactory : None of the document roots [src/main/webapp, public, static] point
s to a location on disk
2020-05-01 22:13:40.461 INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8086 (http)
2020-05-01 22:13:40.532 INFO 1 --- [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2020-05-01 22:13:40.538 INFO 1 --- [           main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.33]
2020-05-01 22:13:40.768 INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2020-05-01 22:13:40.772 DEBUG 1 --- [           main] o.s.web.context.ContextLoader : Published root WebApplicationContext as ServletContext attribute with name
'org.springframework.web.context.request.context'
2020-05-01 22:13:40.773 INFO 1 --- [           main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 6056 ms
2020-05-01 22:13:41.363 DEBUG 1 --- [           main] o.s.b.w.s.ServletContextInitializerBeans : Mapping filters: characterEncodingFilter urls=[/*] order=-214748364
8, requestContextFilter urls=[/*] order=-105
2020-05-01 22:13:41.365 DEBUG 1 --- [           main] o.s.b.w.s.ServletContextInitializerBeans : Mapping servlets: dispatcherServlet urls=[/]
2020-05-01 22:13:41.470 DEBUG 1 --- [           main] o.s.b.w.s.f.OrderedRequestContextFilter : Filter 'requestContextFilter' configured for use
2020-05-01 22:13:41.470 DEBUG 1 --- [           main] s.b.w.s.f.OrderedCharacterEncodingFilter : Filter 'characterEncodingFilter' configured for use
```

Angular:

```
tayeb@master-node:~/git/distributed-project-web/appbook$ sudo docker run --rm -d -p 88:80 angularbookstore:v
5f7bcaf82d03f8c9a14dc839bb746c2d980cc0770e6f064967ca2e236b8e34
tayeb@master-node:~/git/distributed-project-web/appbook$
```


Check running containers :

```
ST9f8d02f4bc3eed430bbccca9002f9a37f305243b73f8f080103b941900b2c
taye@master-node:~/Desktop/appbook$ sudo docker container ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
3f9fedd2f4bc   springappv2   "java -jar app.jar"      4 seconds ago Up 4 seconds  0.0.0.0:8086->8086/tcp             mylink
9a1f47c510a4   mysql:5.7     "docker-entrypoint.s..." 7 hours ago   Up 50 seconds  3306/tcp, 33060/tcp               mybddappsv2
ac226badbca0   angularbookstore:v "nginx -g 'daemon of..." 8 hours ago   Up 8 hours    0.0.0.0:88->80/tcp                 condescending_yonath
490759dbd80f   mysql:5.7     "docker-entrypoint.s..." 15 hours ago  Up 5 minutes  3306/tcp, 33060/tcp               mybddapps
```

Le lien GitHub du projet :

<https://github.com/mohadi21/distributed-project-web>