COMP 3095

Spring Overview – Creating Projects

[redacted]

Agenda

- What is Spring?
- What is Spring Boot?
- What is Spring Initializr?
- Spring Boot Starters?
- Create a web project / Review
- @SpringBootApplication, @SpringBootTest annotations
- JPA Model /ORM Model – Creating Entities
- Spring Data Repositories
- Initializing Data with Spring
- Overview of H2 Database console
- Introduction into Spring MVC
- Configuring Spring MVC Controllers
- Introduction to Thymeleaf Templates

What is Spring?

- One of the most widely used Java EE frameworks used for building applications.
- Provides an elaborate programming and configuration model, providing flexibility.
- Aims to simplify the Java EE development and help developers be more productive
- It can be used on any platform
- One of its major feature is dependency injection, which make development simpler by allowing developers to create loosely coupled applications.
- Allows developers to spend more time , developing business solutions and less time on an applications convention and/or configuration.

What is Spring Boot?

- Spring boot aims to shorten the length of time (and code in general) to provide a viable Spring application
- Uses special annotation, to help create web/stand alone applications , quicker, with almost zero-configuration.
- Does not require the deployment of war files
- It helps embed, Tomcat, Jetty, or Undertow directly
- It offer production ready features
- Easier to launch
- Easier customization and management

What is Spring Boot? Continued…

- Spring boot is a Spring-based production ready project initializer. With feature like auto-configuration, it saves you from writing lengthy code and helps avoid unnecessary configuration.
    - Spring boot used the Spring Framework, as a foundation and improvises on it. It simplifies Spring Dependencies and runs applications straight for the command line if desired.
- Spring Framework on the other hand, offers you the underlying features like dependency injection, IoC, and handles transactions.
    - Spring Framework is highly most effective when it is utilized alongside Spring Boot.
    - The added advantages that come with Spring Boot are great value as they offer completion of project with less effort..

What is Initializr?

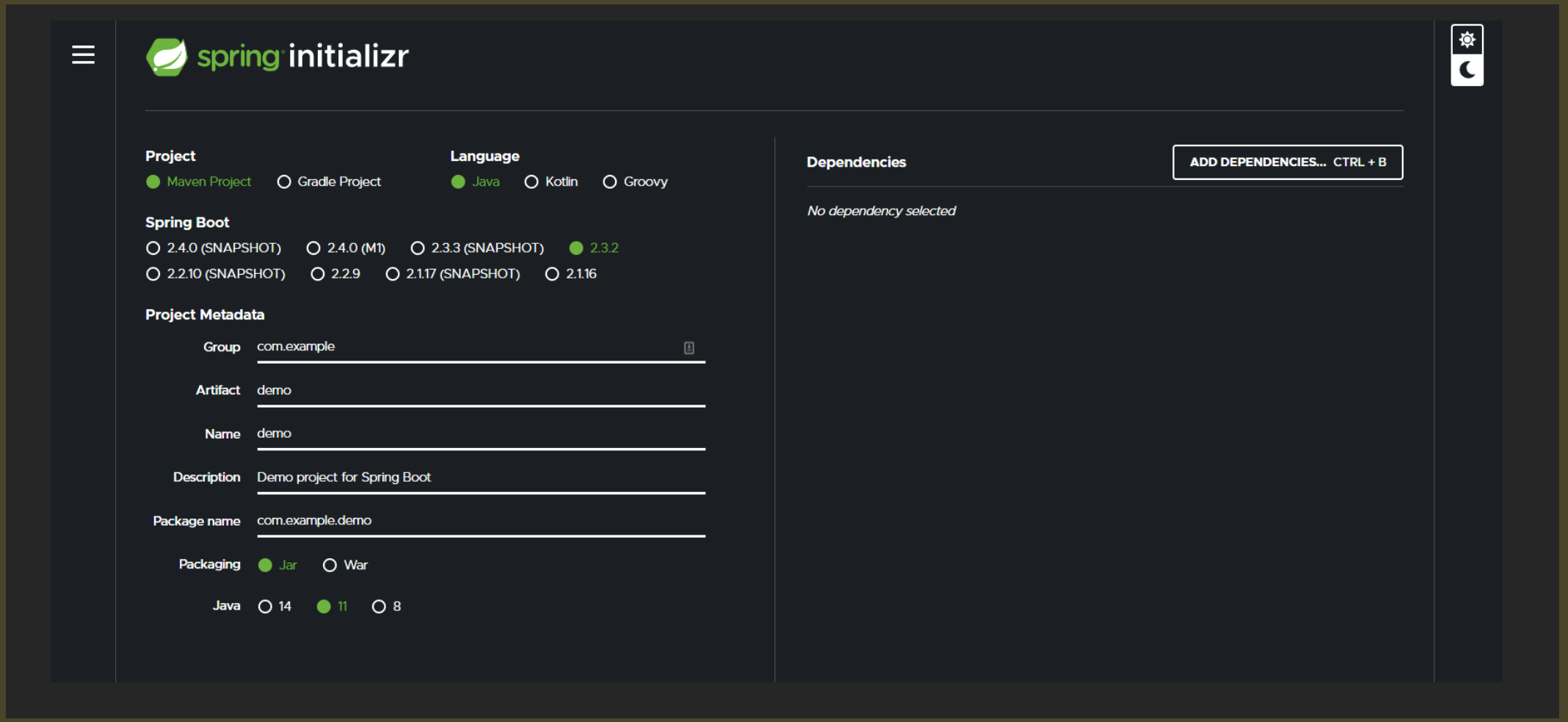
- Spring Initializr is a **web-based tool** provided by the Pivotal Web Service.
- Spring Initializr, allow you to easily generate the structure of the **Spring Boot Project**.
- It offers extensible API for creating JVM-based projects.
- It also provides various options for the project that are expressed in a metadata model. The metadata model allows us to configure the list of dependencies supported by JVM and platform versions, etc.

Spring Boot Starters

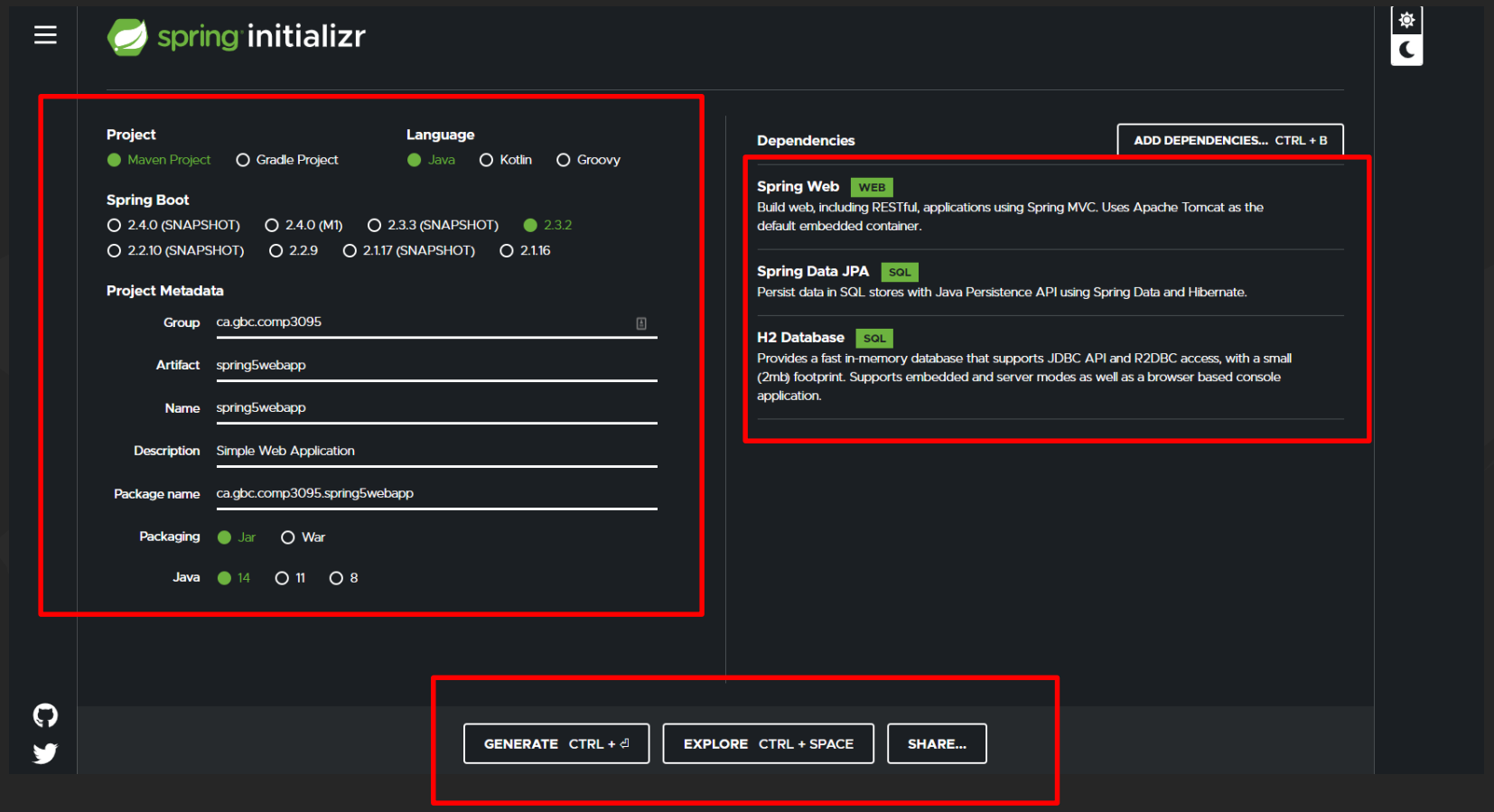
- Spring Boot Starters are a set of convenient dependency descriptors that you can include in your application.
- You get a one-stop-shop for all the Spring related technologies that you may need without having to hunt through sample code and copy paste loads of dependency descriptors.
- For example, if you want to get started using Spring JPA for database access, include the **spring-boot-starter-data-jpa** dependency in your project, and you are good to go.

https://github.com/spring-projects/spring-boot/tree/master/spring-boot-project/spring-boot-starters

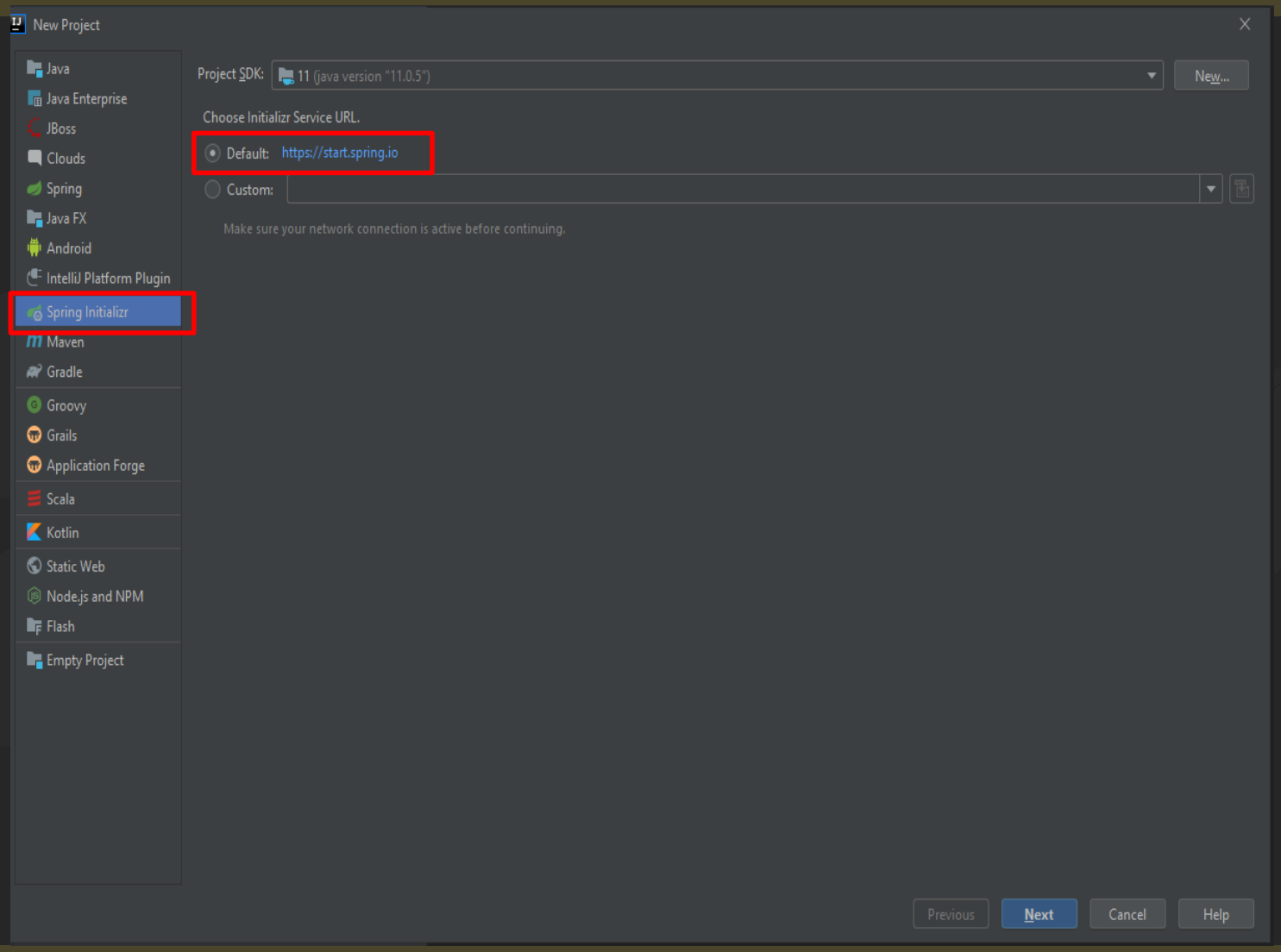
https://start.spring.io



Create a Web Project (start.spring.io)



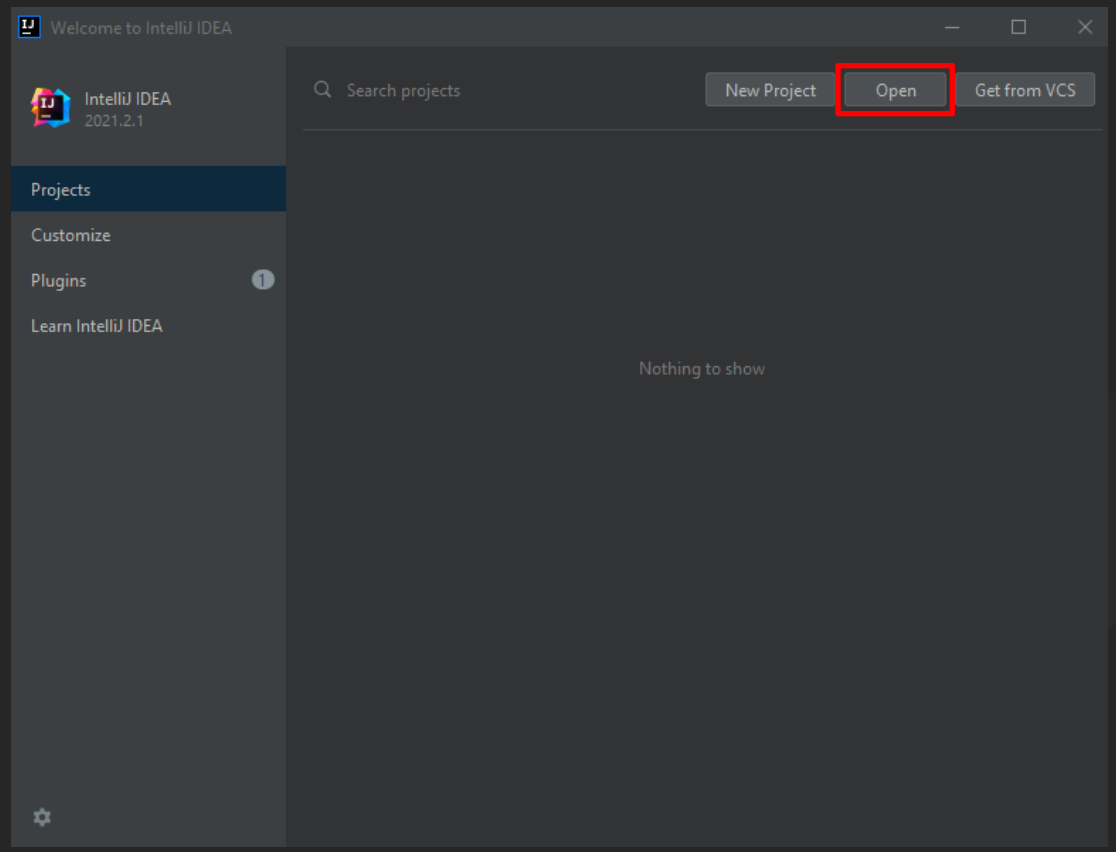
Create a New Project (Inside Intellij)

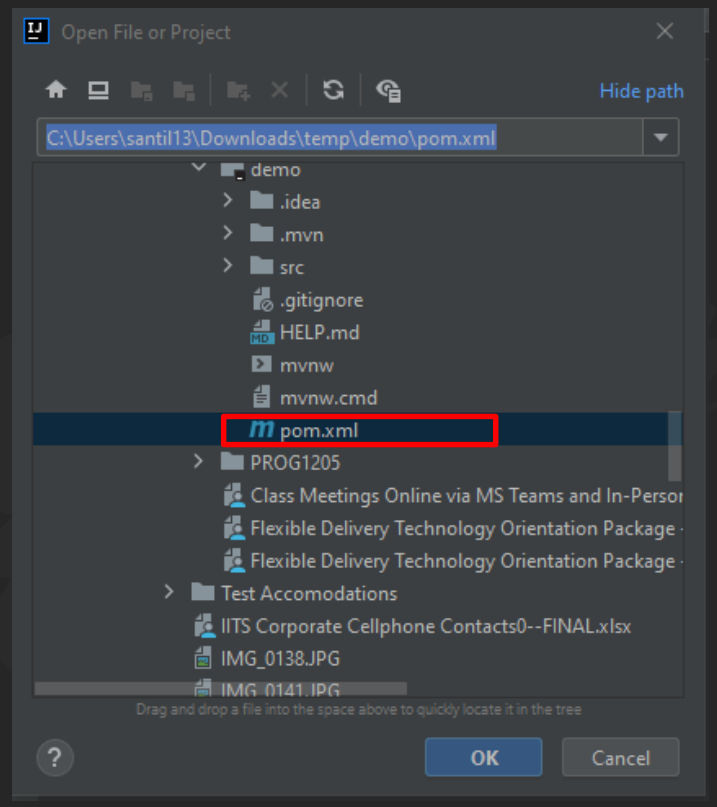


Create a Web Project continued…

Name	Date modified	Type	Size
.mvn	8/11/2020 6:13 PM	File folder	
src	8/11/2020 6:13 PM	File folder	
	8/11/2020 6:13 PM	Text Document	1 KB
HELP.md	8/11/2020 6:13 PM	MD File	2 KB
mvnw	8/11/2020 6:13 PM	File	10 KB
mvnw	8/11/2020 6:13 PM	Windows Comma...	7 KB
pom	8/11/2020 6:13 PM	XML Document	2 KB

Open/Import Project - IntelliJ





Open/Import Project - IntelliJ

spring5webapp › pom.xml

Project

spring5webapp C:\MyProjects\java\comp3095\spring5webapp
- .idea
- .mvn
- src
- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- spring5webapp.iml

External Libraries

Scratches and Consoles

pom.xml

```
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.3.2.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>ca.gbc.comp3095</groupId>
    <artifactId>spring5webapp</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>spring5webapp</name>
    <description>Simple Web Application</description>

    <properties>
        <java.version>14</java.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>com.h2database</groupId>
            <artifactId>h2</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <exclusions>
                <exclusion>
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
    </dependencies>

    <build>
        <plugins>
```

project › artifactId

Terminal | Build | Java Enterprise | Spring | 6: TODO

Maven Tab

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help spring5webapp [C:\MyProjects\java\comp3095\spring5webapp] - ...\pom.xml - IntelliJ IDEA

spring5webapp > pom.xml

Spring5webappApplication

Project

- spring5webapp C:\MyProjects\java\
  - .idea
  - .mvn
  - src
  - .gitignore
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
  - spring5webapp.iml
- External Libraries
- Scratches and Consoles

pom.xml

```
15      <description>Simple Web Application</description>
16
17      <properties>
18          <java.version>14</java.version>
19      </properties>
20
21      <dependencies>
22          <dependency>
23              <groupId>org.springframework.boot</groupId>
24              <artifactId>spring-boot-starter-data-jpa</artifactId>
25          </dependency>
26          <dependency>
27              <groupId>org.springframework.boot</groupId>
28              <artifactId>spring-boot-starter-web</artifactId>
29          </dependency>
30
31          <dependency>
32              <groupId>com.h2database</groupId>
33              <artifactId>h2</artifactId>
34              <scope>runtime</scope>
35          </dependency>
36          <dependency>
37              <groupId>org.springframework.boot</groupId>
38              <artifactId>spring-boot-starter-test</artifactId>
39              <scope>test</scope>
40              <exclusions>
41                  <exclusion>
42                      <groupId>org.junit.vintage</groupId>
43                      <artifactId>junit-vintage-engine</artifactId>
44                  </exclusion>
45              </exclusions>
46          </dependency>
47      </dependencies>
48
49      <build>
50          <plugins>
51              <plugin>
52                  <groupId>org.springframework.boot</groupId>
53                  <artifactId>spring-boot-maven-plugin</artifactId>
54              </plugin>
55          </plugins>
56      </build>
```

project > parent > artifactId

Maven

- spring5webapp

Terminal Build Java Enterprise Spring 6: TODO Event Log

7:45 LF UTF-8 Tab*

@SpringBootApplication, @SpringBootTest, Help.md, .gitignore

@SpringBootApplication
serves two purposes in a Spring Boot application: **configuration** and **bootstrapping**. First, it signifies the main Spring boot configuration class and second, it enables the auto-configuration feature of a Spring Boot application

@SpringBootTest
can be used when we need to bootstrap the entire container. The annotation works by creating the *ApplicationContext* (central api within Spring) that will be utilized in our tests.

Help.md
Markdown document. Generates helpful links to documentation to the resources that were included in the starter project (Spring Web, Spring Data etc..)

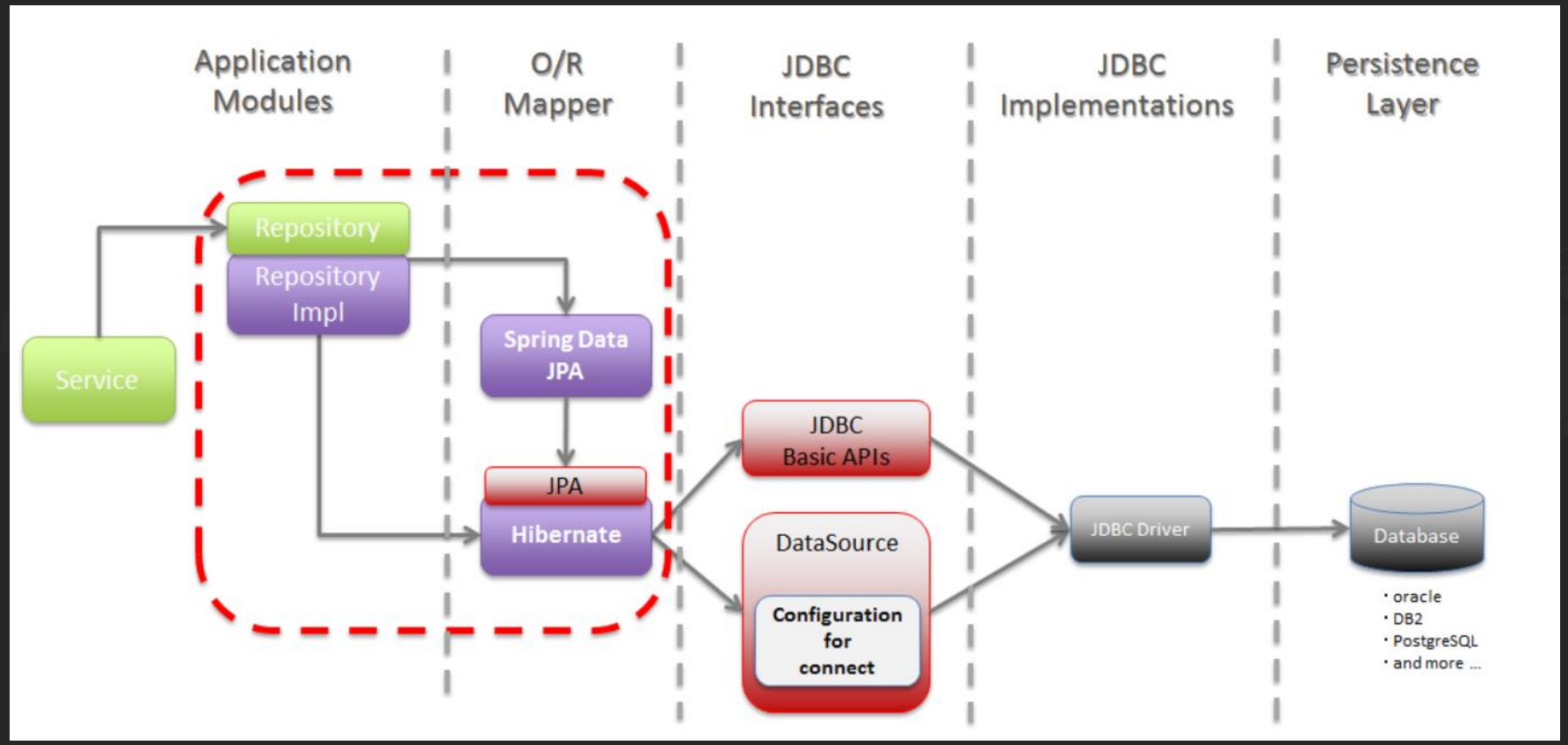
.gitignore
File is usually placed in the root directory of a project. You can also create a global .gitignore file and any entries in that file will be ignored in all of your Git repositories.

JPA / ORM Model - Creating Entities

JPA (Java Persistence API) Model

- JPA is an object relational mapping tool (ORM) that makes writing code easier.
- Object Relational Mapping is the idea of being able to write SQL queries, using the object oriented paradigm of your preferred language.
- In Short, we are trying to interact with our applications database, using our language of choice (Java) instead of raw SQL.
- An ORM (object relational mapper) specifically refers to a library that implements this technique and makes it available our application.
- The ORM we will use in this course is **Hibernate**.
- Note: JPA is the specification we will use in this course, while hibernate is the actual JPA provider (Hibernate implements the various specification mentions in the JPA contract).

Accessing a database using JPA

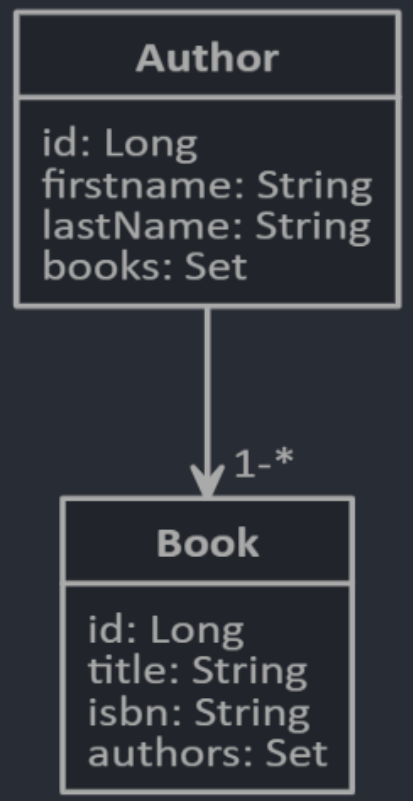


- In this course, Hibernate is the JPA provider, Sprionmg Data JPA as the JPA wrapper

Example: Entity Relationship Mapping

JDL-Studio

```
entity Author {
    id Long
    firstname String
    lastName String
    books Set
}

entity Book {
    id Long
    title String
    isbn String
    authors Set
}


relationship ManyToMany {
    Author{book} to Book{author}
}
```



Creating Plain Old Java Objects (POJOs)

1. **Create Book POJO**
    - **data members title, isbn, and authors**
    - **include two constructors**
        - **Zero argument constructor is generally mandatory**
    - **Setters and Getters**

2. **Create Author POJO**
    - **data members firstname, lastname, books**
    - **Include two constructors**
        - **Zero argument constructor is generally mandatory**
    - **Setters and Getters**

Convert POJOs into JPA Entities

@Entity

- **Part of the javax.persistence package**
- **Tells hibernate this annotated class is an entity**

@Id

- **Marks the field as a primary key**
- **@GeneratedValue(Strategy = GenerationType.AUTO)**
  - **Marks the field as automatically generated.**
  - **Underlying database provides the generation of this value**

Annotations are used to provide supplemental information about a program

- Annotations start with @
- Help to associate metadata (information) to the program.
- Classes, variables, constructors methods, can all be annotated.

JPA Entity Relationships

@ManyToMany (mappedBy = “some key”)

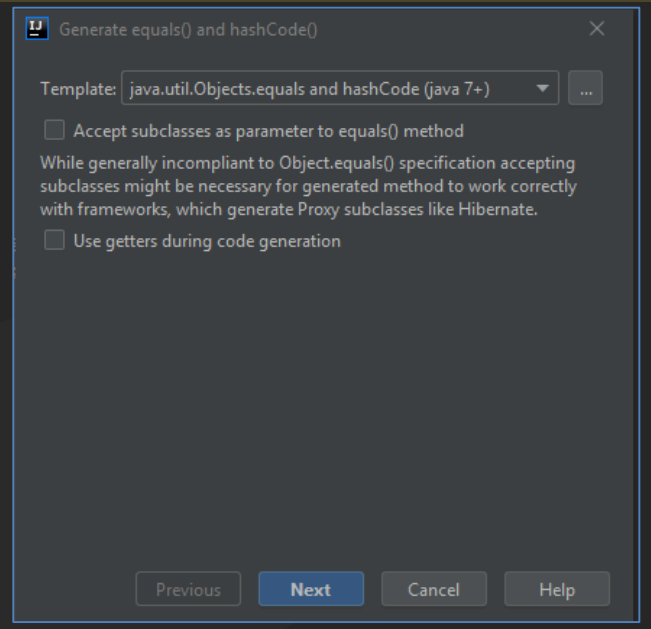
- **Part of the javax.persistence package**
- **Informs hibernate of a many to many relationship**
- **mappedBy signals to hibernate that the key for the relationship is on the other side**

@JoinTable

- **Part of the javax.persistence package**
- **Informs hibernate of a Join table (think many-to-many)**

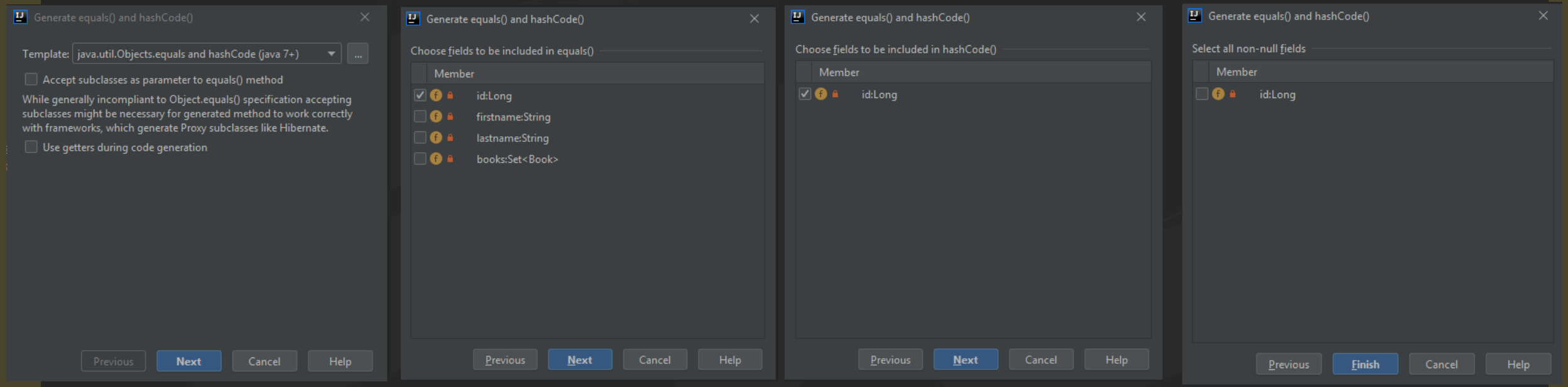
Equals and HashCode

- Standard equals method we inherit from Java (default) will not suffice for our needs.
- We need to base equality on the Id (database records) of the objects.
- The Id/primary key (in our example) is what identifies objects around the entities they represent.
- We need to incorporate logic that utilizes the Ids, such that, if two objects contain the same Id value, then they are equal (same object).
- The purpose of the hashCode() method is to provide a numeric representation of an objects contents, so as to provide an alternate mechanism to loosely identify it.
- This needs to be done for both our example classes, Authors and Books.



alt + I → equals and hashCode()

Generate equals() and hashCode()



equals() and hashCode() continued…

```java
@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    Author author = (Author) o;
    return Objects.equals(id, author.id);
}

@Override
public int hashCode() {
    return Objects.hash(id);
}
```

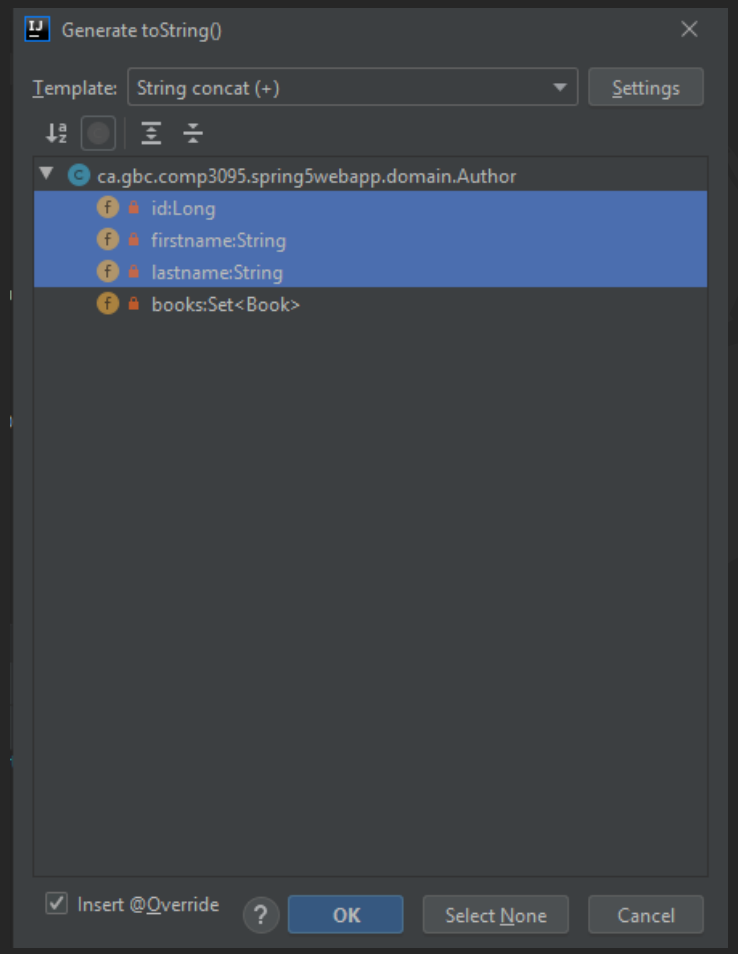
equals() and hashCode() for Author class

equals() and hashCode() continued…

```java
@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    Book book = (Book) o;
    return Objects.equals(id, book.id);
}

@Override
public int hashCode() {
    return Objects.hash(id);
}
```

equals() and hashCode() for Book class

ToString()




```
@Override
public String toString() {
    return "Author{" +
            "id=" + id +
            ", firstname='" + firstname + '\'' +
            ", lastname='" + lastname + '\'' +
            '}';
}
```

Spring Data Repositories

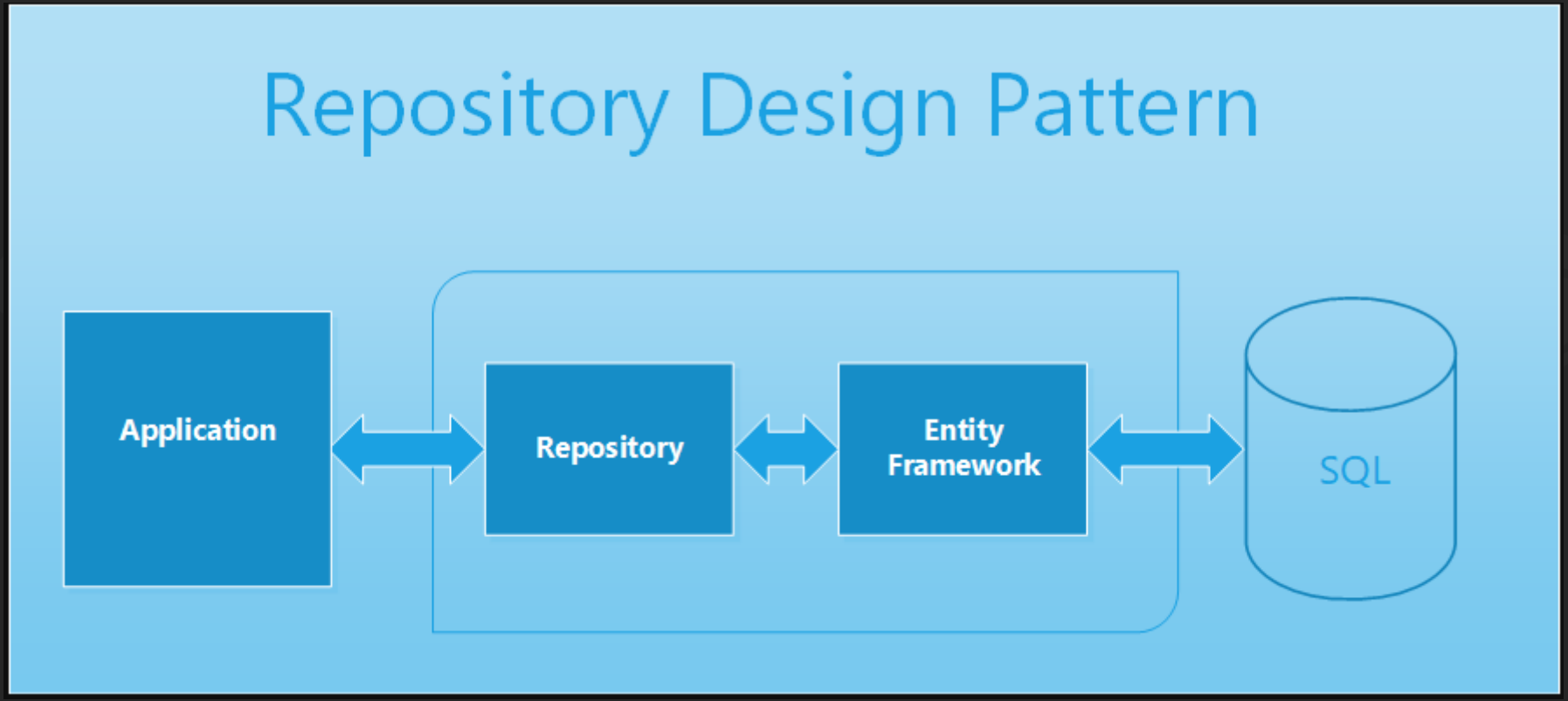
[redacted]

Spring Data JPA / Repositories

- Spring Data is a family of projects
- Spring Data JPA is made to work with hibernate
  - Hibernate is bundled within the Spring Data JPA Dependency
- Alleviates a lot of coding required to work with database operations
- Implements repository design pattern (competing pattern to dao – data access objects)

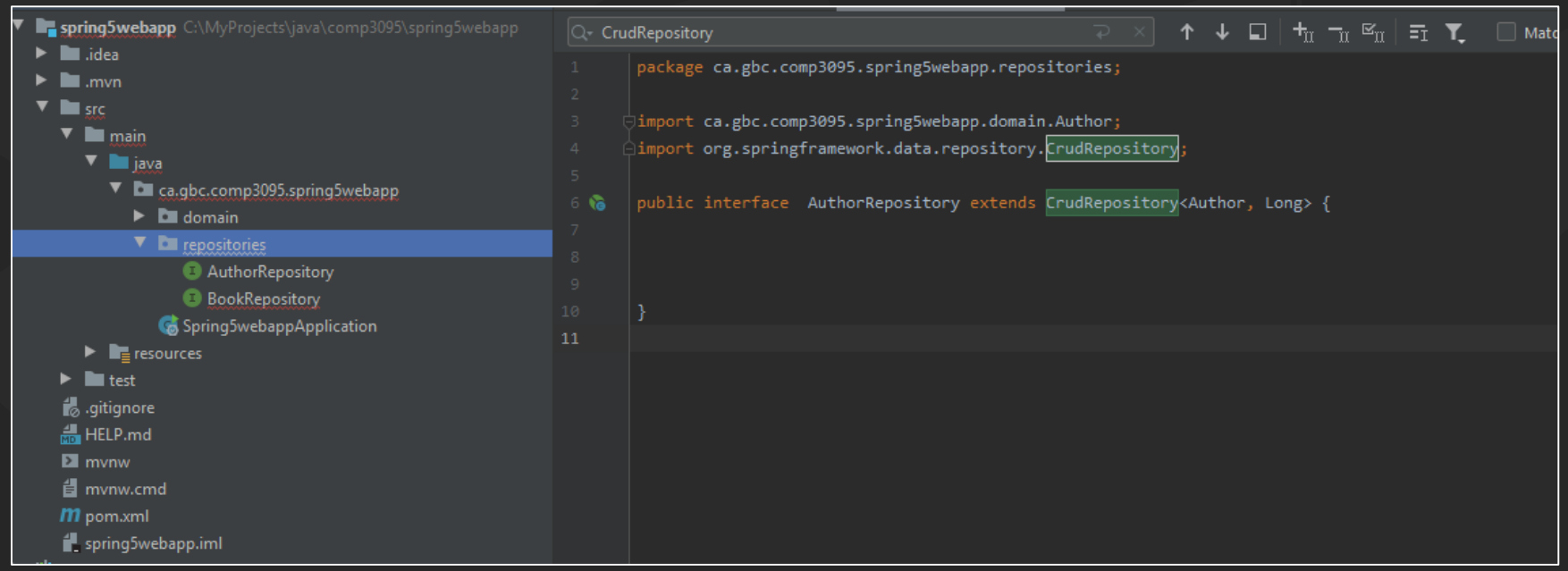
- Repository pattern
  - Utilizes a repository object that is responsible for persistence and query operations.
  - Simple design pattern to implement
  - Spring Data JPA underneath the covers is taking care of all the hibernate commands, transactional commands and other ceremonial jdbc code etc…
  - Allows developers the ability to focus on business functionality

Repository Design Pattern



Implement Spring Data JPA Repositories

- Create AuthorRepository
  - extends CrudRepository → CrudRepository is part of the Spring Data API
- Create BookRepository
  - extends CrudRepository



Initializing Data with Spring

[redacted]

Initializing Data with Spring

To populate data and further our example, we will …

- Create new package bootstrap

- Create new class BootStrapData
  - Implements CommandLineRunner
    - this is used to indicate a bean should run when it is contained within a Spring Application.
    - must implement variable argument method run(String… args)
    - only run() method is required for CommandLineRunner

  - Mark BootStrapData as **@Component** → this tells SpringFramework to detect this as a Spring Managed Component Bean

Spring Data JPA

```java
public class BootStrapData implements CommandLineRunner {

    private final AuthorRepository authorRepository;
    private final BookRepository bookRepository;

    //dependency injection for both AuthorRepository and BookRepository
    public BootStrapData(AuthorRepository authorRepository, BookRepository bookRepository) {
        this.authorRepository = authorRepository;
        this.bookRepository = bookRepository;
    }

    @Override
    public void run(String... args) throws Exception {

        //create an author and book
        Author eric = new Author("eric", "Evans");
        Book ddd = new Book( title: "Domain Driven Design",  isbn: "123123");

        //add book to author
        eric.getBooks().add(ddd);
        //add author to book
        ddd.getAuthors().add(eric);

        //save both objects (eric and ddd) to H2 database;
        authorRepository.save(eric);
        bookRepository.save(ddd);

        //create an author and book
        Author rod = new Author("Rod", "Johnson");
        Book noEJB = new Book( title: "J2EE Developement without EJB",  isbn: "3939459459");

        //add book to author
        rod.getBooks().add(noEJB);
        //add author to book
        noEJB.getAuthors().add(rod);

        System.out.println("Started in Bootstrap");
        System.out.println("Number of Books" + bookRepository.count());
```

Run the Application

```java
package ca.gbc.comp3095.spring5webapp;

import ...

@SpringBootApplication
public class Spring5webappApplication {

    public static void main(String[] args) { SpringApplication.run(Spring5webappApplication.class, args); }

}
```

```
2020-08-12 16:35:51.899  INFO 20908 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2020-08-12 16:35:51.900  INFO 20908 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2030 ms
2020-08-12 16:35:52.077  INFO 20908 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2020-08-12 16:35:52.086  INFO 20908 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2020-08-12 16:35:52.216  INFO 20908 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2020-08-12 16:35:52.267  INFO 20908 --- [         task-1] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2020-08-12 16:35:52.299  WARN 20908 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explic
2020-08-12 16:35:52.316  INFO 20908 --- [         task-1] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 5.4.18.Final
2020-08-12 16:35:52.439  INFO 20908 --- [         task-1] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.1.0.Final}
2020-08-12 16:35:52.559  INFO 20908 --- [         task-1] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.H2Dialect
2020-08-12 16:35:52.677  INFO 20908 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
2020-08-12 16:35:52.680  INFO 20908 --- [           main] DeferredRepositoryInitializationListener : Triggering deferred initialization of Spring Data repositories…
2020-08-12 16:35:53.218  INFO 20908 --- [         task-1] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2020-08-12 16:35:53.225  INFO 20908 --- [         task-1] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2020-08-12 16:35:53.380  INFO 20908 --- [           main] DeferredRepositoryInitializationListener : Spring Data repositories initialized!
2020-08-12 16:35:53.390  INFO 20908 --- [           main] c.g.c.s.Spring5webappApplication         : Started Spring5webappApplication in 4.494 seconds (JVM running for 6.649)
Started in Bootstrap
Number of Books 2
```

H2 Database Console

[redacted]

Tomcat Service

Console Endpoints

```
 :: Spring Boot ::        (v2.3.2.RELEASE)

2020-08-13 12:26:42.695  INFO 14036 --- [           main] c.g.c.s.Spring5webappApplication         : Starting Spring5webappApplication on ITS-B4CX1Z2 with PID 14036 (C:\MyProjects\java\comp3095\Lecture\spring5webapp\target
2020-08-13 12:26:42.720  INFO 14036 --- [           main] c.g.c.s.Spring5webappApplication         : No active profile set, falling back to default profiles: default
2020-08-13 12:26:44.620  INFO 14036 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFERRED mode.
2020-08-13 12:26:44.753  INFO 14036 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 119ms. Found 3 JPA repository interfaces.
2020-08-13 12:26:45.839  INFO 14036 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2020-08-13 12:26:45.857  INFO 14036 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2020-08-13 12:26:45.858  INFO 14036 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.37]
2020-08-13 12:26:46.071  INFO 14036 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2020-08-13 12:26:46.071  INFO 14036 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 3197 ms
2020-08-13 12:26:46.292  INFO 14036 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2020-08-13 12:26:46.304  INFO 14036 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2020-08-13 12:26:46.473  INFO 14036 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2020-08-13 12:26:46.540  INFO 14036 --- [         task-1] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2020-08-13 12:26:46.589  WARN 14036 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explic
2020-08-13 12:26:46.603  INFO 14036 --- [         task-1] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 5.4.18.Final
2020-08-13 12:26:46.778  INFO 14036 --- [         task-1] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.1.0.Final}
2020-08-13 12:26:46.962  INFO 14036 --- [         task-1] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.H2Dialect
2020-08-13 12:26:47.133  INFO 14036 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
2020-08-13 12:26:47.136  INFO 14036 --- [           main] DeferredRepositoryInitializationListener : Triggering deferred initialization of Spring Data repositories…
2020-08-13 12:26:48.201  INFO 14036 --- [         task-1] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatf
2020-08-13 12:26:48.213  INFO 14036 --- [         task-1] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityMan
2020-08-13 12:26:48.530  INFO 14036 --- [           main] DeferredRepositoryInitializationListener : Spring Data repositories
2020-08-13 12:26:48.551  INFO 14036 --- [           main] c.g.c.s.Spring5webappApplication         : Started Spring5webappAppl
Started in Bootstrap...
```

localhost:8080/h2-console/

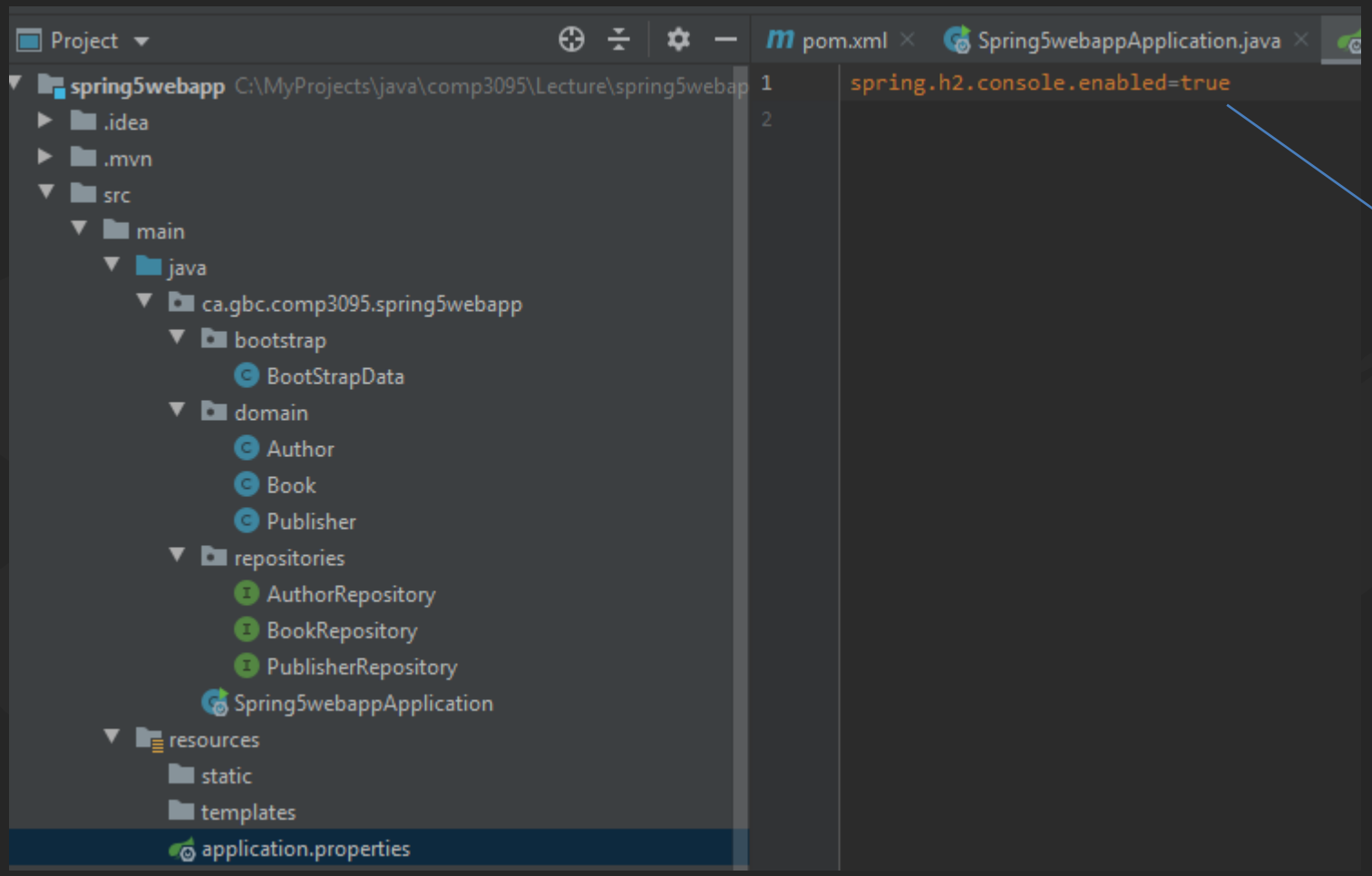
Bookmarks UTSC Other News

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Thu Aug 13 12:30:22 EDT 2020
There was an unexpected error (type=Not Found, status=404).

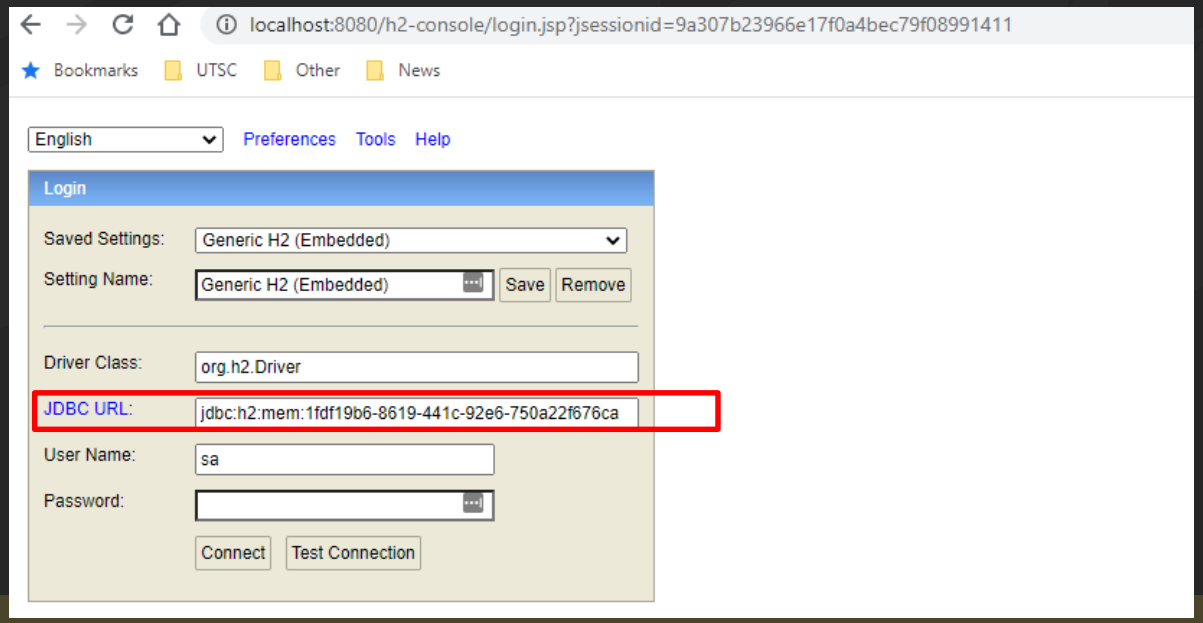
H2-Console



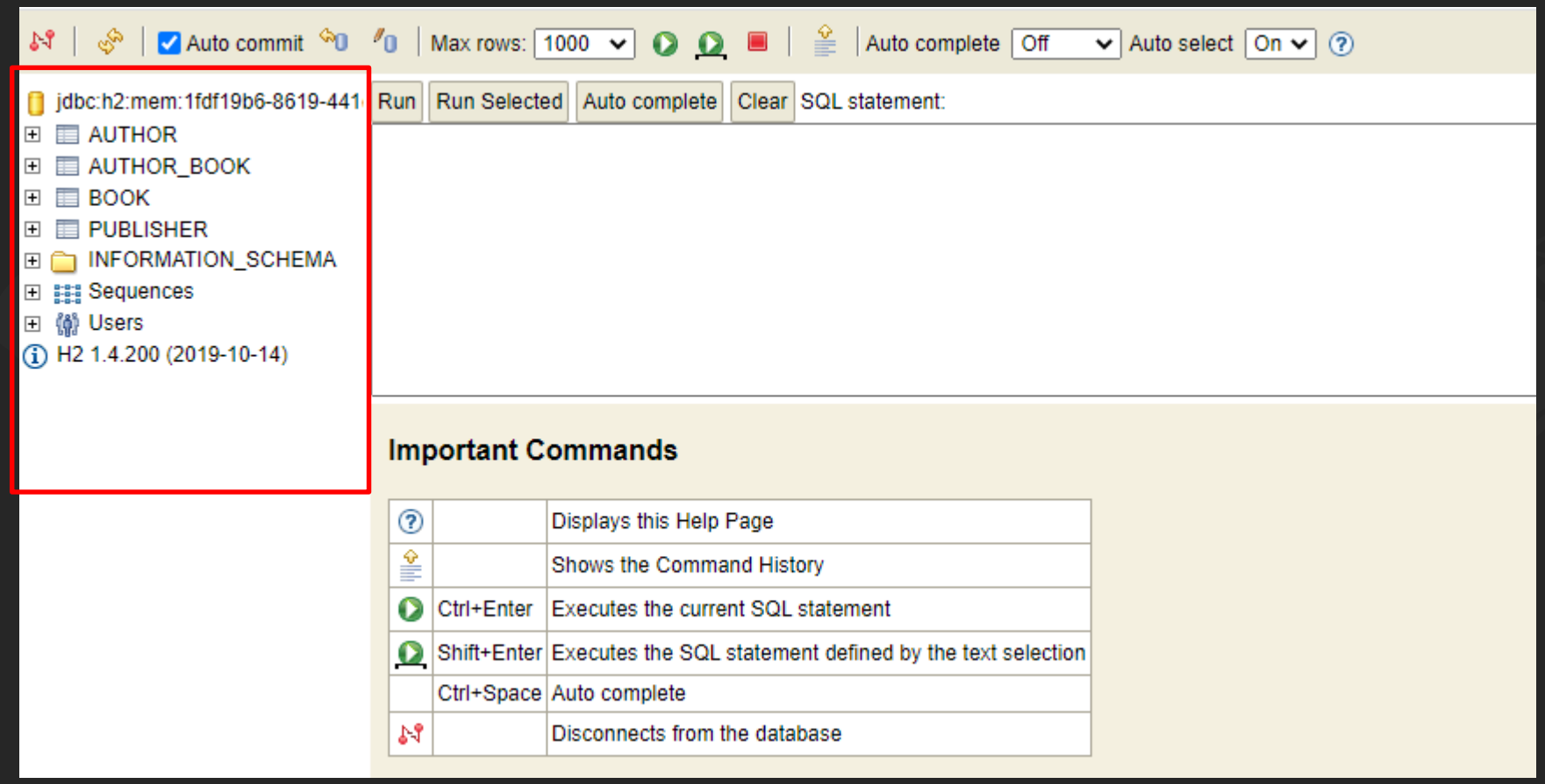
enable H2 console

H2-Console

```
Spring5webappApplication
Console   Endpoints
:: Spring Boot ::        (v2.3.2.RELEASE)

2020-08-13 12:35:02.315  INFO 14176 --- [           main] c.g.c.s.Spring5webappApplication         : Starting Spring5webappApplication on ITS-B4CX1Z2 with PID 14176 (C:\MyProjects\java\comp3095\Lecture\spring5webapp\ta
2020-08-13 12:35:02.320  INFO 14176 --- [           main] c.g.c.s.Spring5webappApplication         : No active profile set, falling back to default profiles: default
2020-08-13 12:35:03.390  INFO 14176 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFERRED mode.
2020-08-13 12:35:03.495  INFO 14176 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 91ms. Found 3 JPA repository interfaces.
2020-08-13 12:35:04.185  INFO 14176 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2020-08-13 12:35:04.195  INFO 14176 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2020-08-13 12:35:04.195  INFO 14176 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.37]
2020-08-13 12:35:04.334  INFO 14176 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2020-08-13 12:35:04.334  INFO 14176 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1935 ms
2020-08-13 12:35:04.382  INFO 14176 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2020-08-13 12:35:04.510  INFO 14176 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2020-08-13 12:35:04.519  INFO 14176 --- [           main] o.s.b.a.h2.H2ConsoleAutoConfiguration    : H2 console available at '/h2-console'. Database available at 'jdbc:h2:mem:1fdf19b6-8619-441c-92e6-750a22f676ca'
2020-08-13 12:35:04.650  INFO 14176 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2020-08-13 12:35:04.696  INFO 14176 --- [        task-1] o.hibernate.jpa.internal.util.LogHelper   : HHH000204: Processing PersistenceUnitInfo [name: default]
```



H2-Console


Auto commit
Max rows: 1000
Auto complete Off
Auto select On
jdbc:h2:mem:1fdf19b6-8619-441
Run
Run Selected
Auto complete
Clear
SQL statement:
AUTHOR
AUTHOR_BOOK
BOOK
PUBLISHER
INFORMATION_SCHEMA
Sequences
Users
H2 1.4.200 (2019-10-14)
Important Commands
Displays this Help Page
Shows the Command History
Ctrl+Enter
Executes the current SQL statement
Shift+Enter
Executes the SQL statement defined by the text selection
Ctrl+Space
Auto complete
Disconnects from the database

Introduction to Spring MVC

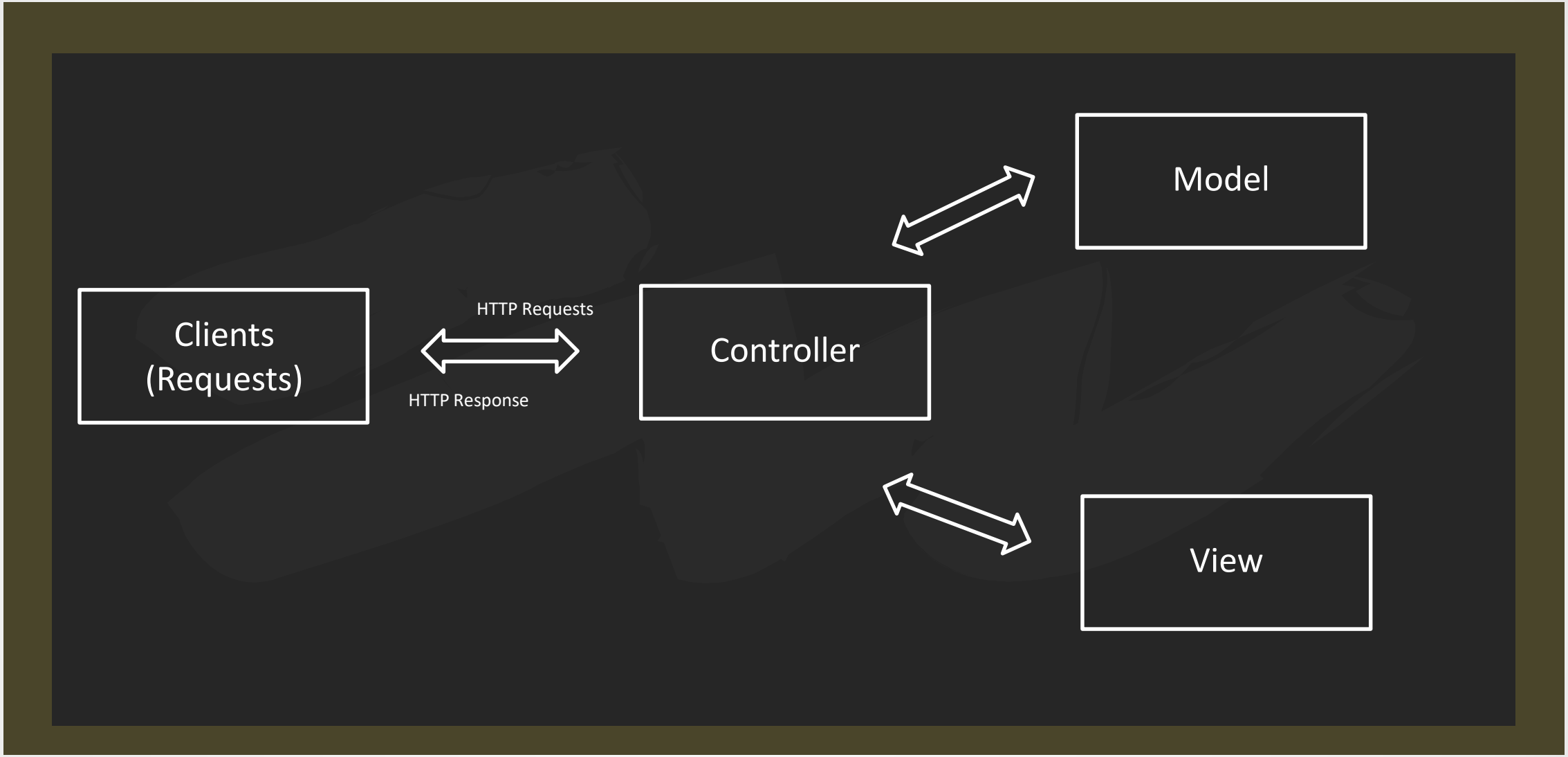
(Web Module)

[redacted]

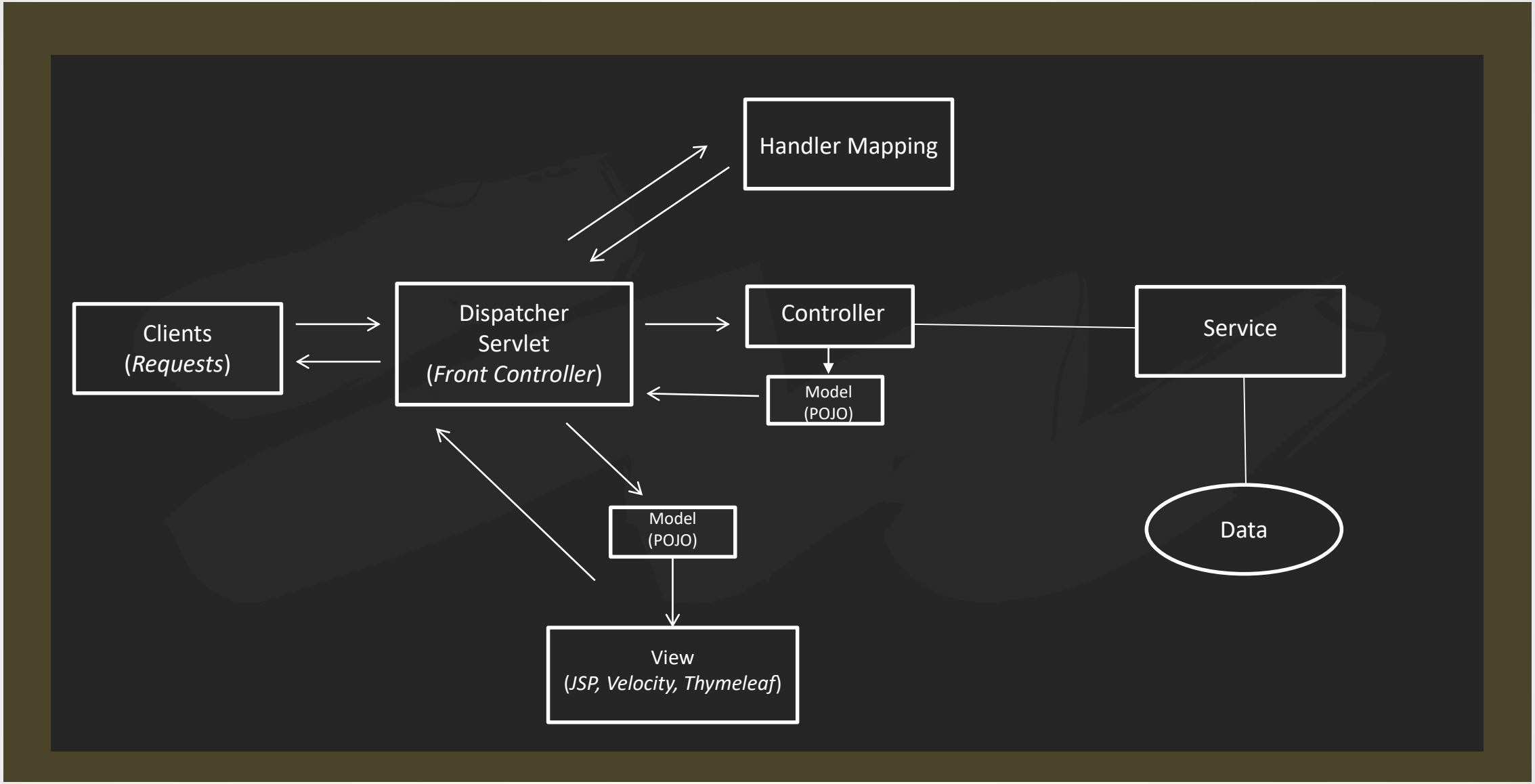
What is MVC?

- Spring Web uses a design paradigm called MVC
- MVC is a common design pattern for GUI and Web applications
  - M = Model
  - V = View
  - C = Controller
- Widely adopted because mvc does an excellent job of separating implementation concerns.

MVC


Model
Clients
(Requests)
HTTP Requests
HTTP Response
Controller
View

Spring MVC


Handler Mapping
Clients
(*Requests*)
Dispatcher
Servlet
(*Front Controller*)
Controller
Service
Model
(POJO)
Model
(POJO)
Data
View
(*JSP, Velocity, Thymeleaf*)

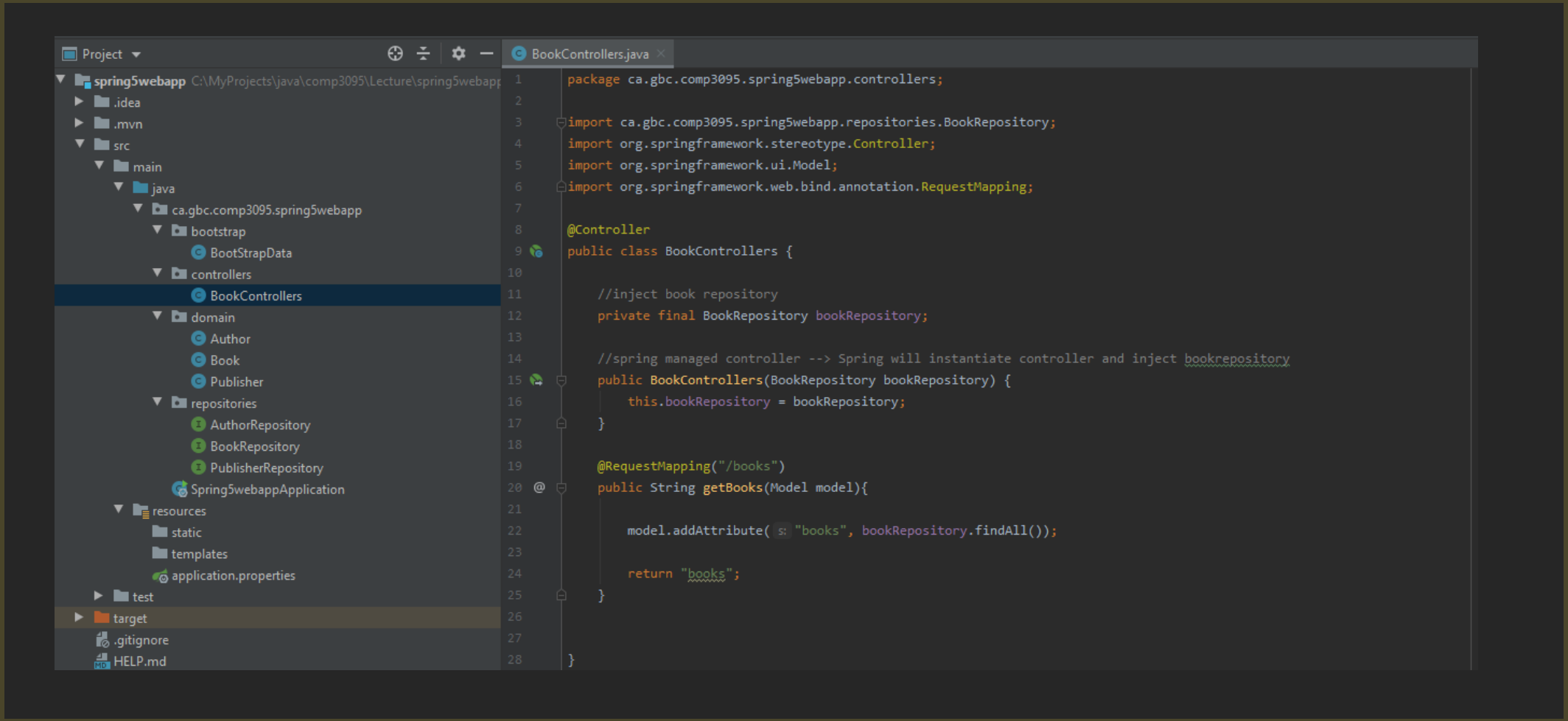
Configuring Spring MVC Controllers

[redacted]

Configuring Spring Controllers

- Annotate Controller class with **@Controller**
  - This is registers the class as a Spring Bean and as a Controller
- Map methods to HTTP request paths using **@RequestMapping**

Creating a BookController



Thymeleaf Templates

(“time-leaf”)

[redacted]

Thymeleaf Templates

- Several templating technologies to choose from
  - Thymeleaf
  - FreeMarker
  - ~~Velocity~~
  - JSF
  - JSPs …
- Templates are a technology used to generate html
  - Example:
    - pulling info from the database to render it onto the browser.

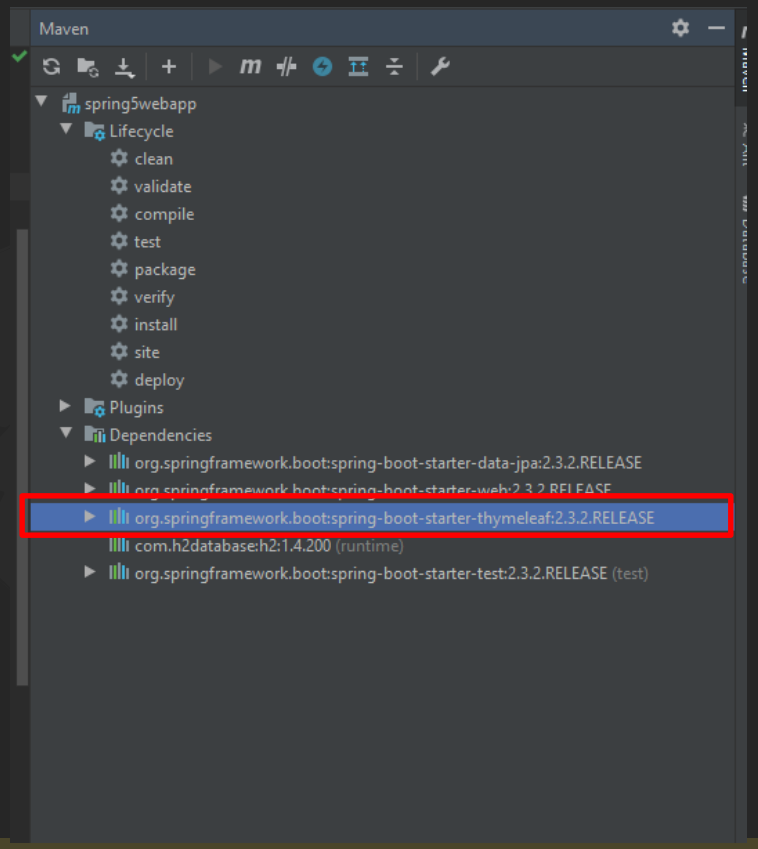
Introducing Thymeleaf

- Thymeleaf is a Java template engine
  - recommended alternative to JSPs
- First release in July 2011
- Rapidly gaining popularity in the Spring community
- Thymeleaf is a natural template engine
  - “Natural” meaning you can view templates in your browser.

Including Thymeleaf

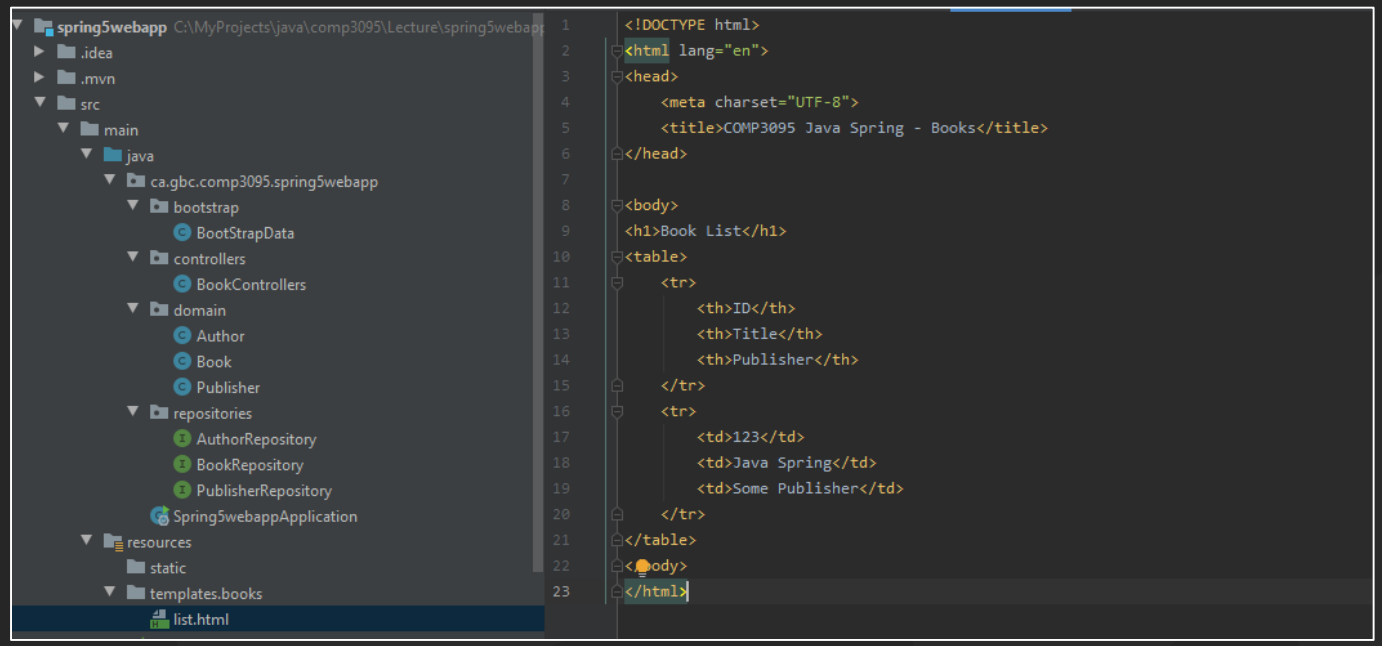
- must include spring-boot-starter-thymeleaf in applications pom.xml

```
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>

    <dependency>
        <groupId>com.h2database</groupId>
        <artifactId>h2</artifactId>
        <scope>runtime</scope>
    </dependency>
```



Thymeleaf Resources

- By default spring will look for templates under “resources”
- Update BookController, getBooks() to return books/list

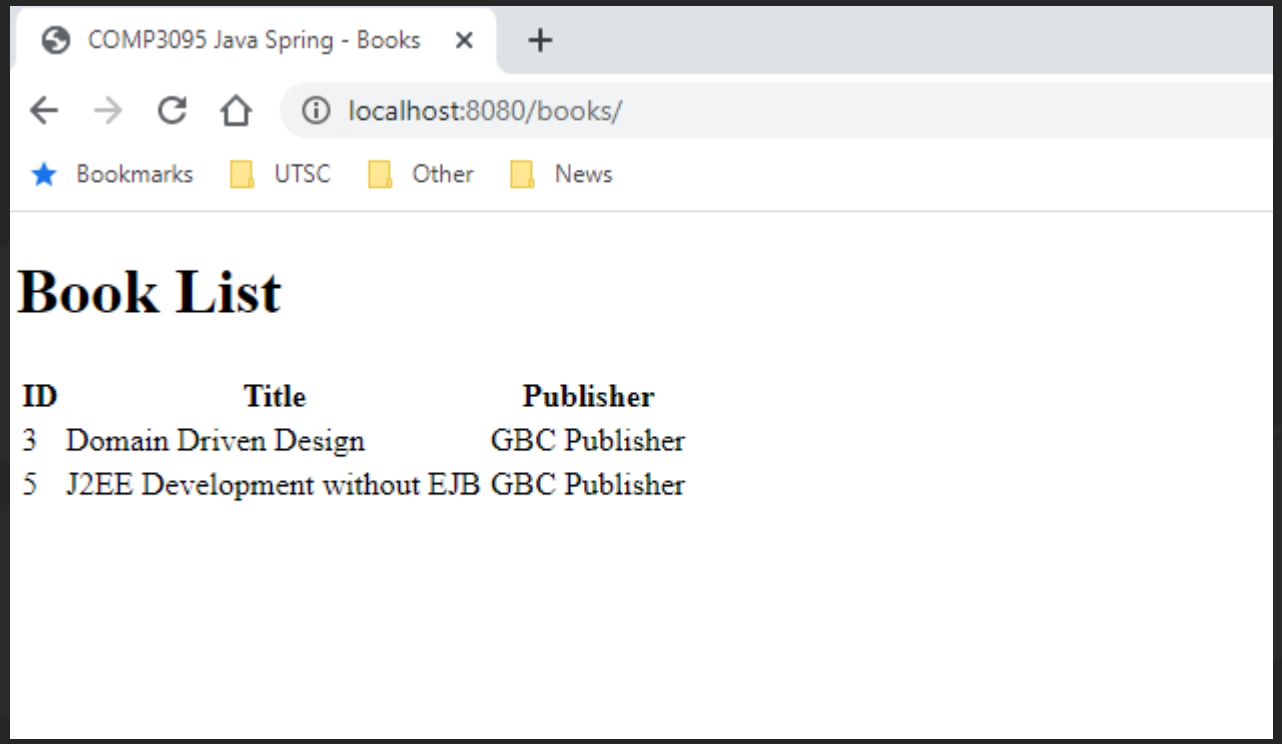



```
@RequestMapping("/books")
public String getBooks(Model model){

    model.addAttribute( s: "books", bookRepository.findAll());

    //look for list template in book directory under resources (resources/templates/books/list)
    return "books/list";
}
```

Add Thymeleaf Syntax

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>COMP3095 Java Spring - Books</title>
</head>

<body>
<h1>Book List</h1>
<table>
    <tr>
        <th>ID</th>
        <th>Title</th>
        <th>Publisher</th>
    </tr>
    <tr th:each="book : ${books}">
        <td th:text="${book.id}">123</td>
        <td th:text="${book.title}">Java Spring</td>
        <td th:text="${book.publisher.name}">Some Publisher</td>
    </tr>
</table>
</body>
</html>
```

View Listing



View Page Source

view-source:localhost:8080/books/

Bookmarks UTSC Other News

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>COMP3095 Java Spring - Books</title>
</head>

<body>
<h1>Book List</h1>
<table>
    <tr>
        <th>ID</th>
        <th>Title</th>
        <th>Publisher</th>
    </tr>
    <tr>
        <td>3</td>
        <td>Domain Driven Design</td>
        <td>GBC Publisher</td>
    </tr>
    <tr>
        <td>5</td>
        <td>J2EE Development without EJB</td>
        <td>GBC Publisher</td>
    </tr>
</table>
</body>
</html>
```

Iteration

Questions?

[redacted]