

eLibrary - My First Spring Boot Web Application

To create your first Spring Boot Webapp (named, eLibrary), do the following steps:

1. If you do not already have one, Create a new empty folder (directory) to serve as your project workspace (e.g. name it, workspace-eclipse).
2. Launch-open your Eclipse IDE (or some other preferred IDE tool of your choice, such as STS (Spring Tool Suite) on to the workspace.
3. If using Eclipse and if your Eclipse IDE does not already have it, install the **Spring Tools 4** plugin for Eclipse, by doing the following:
 - 3.1. In Eclipse, Select the Help -> Eclipse Marketplace menu item
 - 3.2. In the "Find" textfield under the Search tab, enter the text - "Spring Tools 4" and click on the "Go" button, to search for the plugin.
 - 3.3. Click on the "Install" button for the Spring Tools 4 plugin item found, in the search result. And when the plugin is successfully installed, restart Eclipse when prompted.
4. In Eclipse or STS, click on File -> New menu and select, "Spring Starter Project".
5. Fill-in the project details as shown below and click "Next":

The screenshot shows the 'New Spring Starter Project' dialog box in Eclipse. The fields are filled as follows:

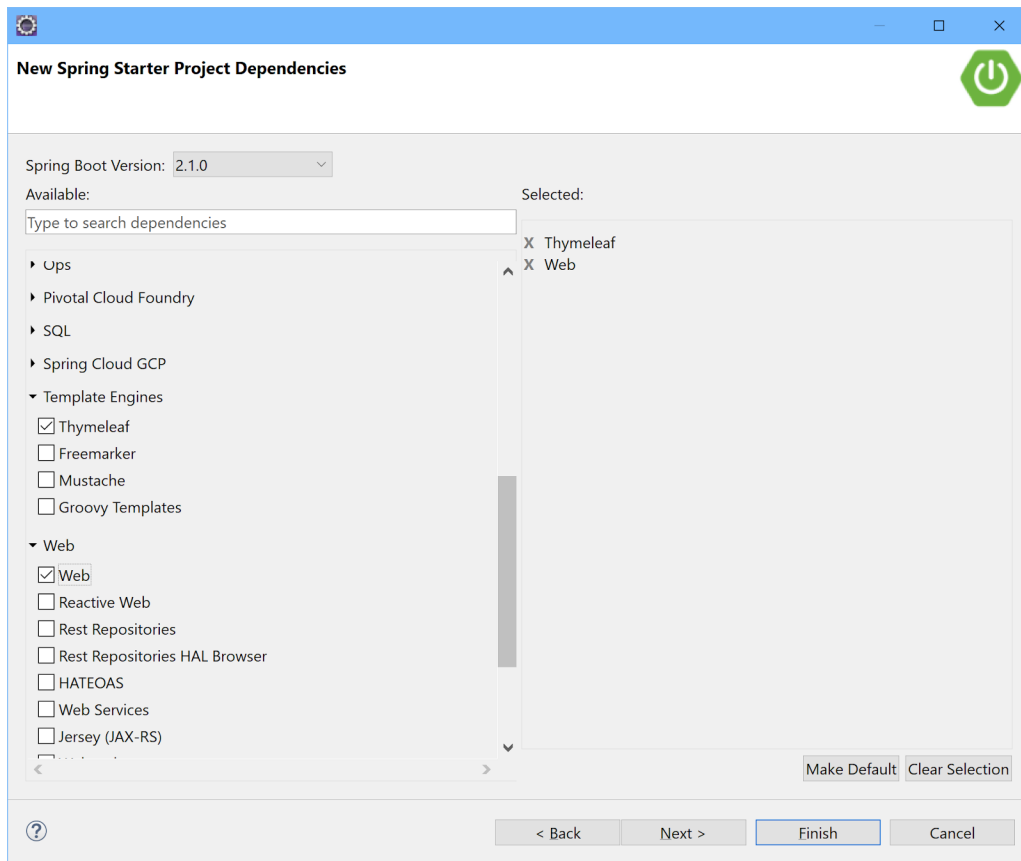
- Service URL: <https://start.spring.io>
- Name: elibrary
- Use default location: ☒
- Location: C:\oak\MyMUM\teaching\courses\CS425 - SWE\201811\workspace-e [Browse]
- Type: Maven
- Packaging: Jar
- Java Version: 8
- Language: Java
- Group: edu.mum.cs.cs425.demowebapps.elibrary
- Artifact: elibrary
- Version: 0.0.1-SNAPSHOT
- Description: My First demo Spring Boot webapp for CS425-SWE
- Package: edu.mum.cs.cs425.demowebapps.elibrary

Working sets section:

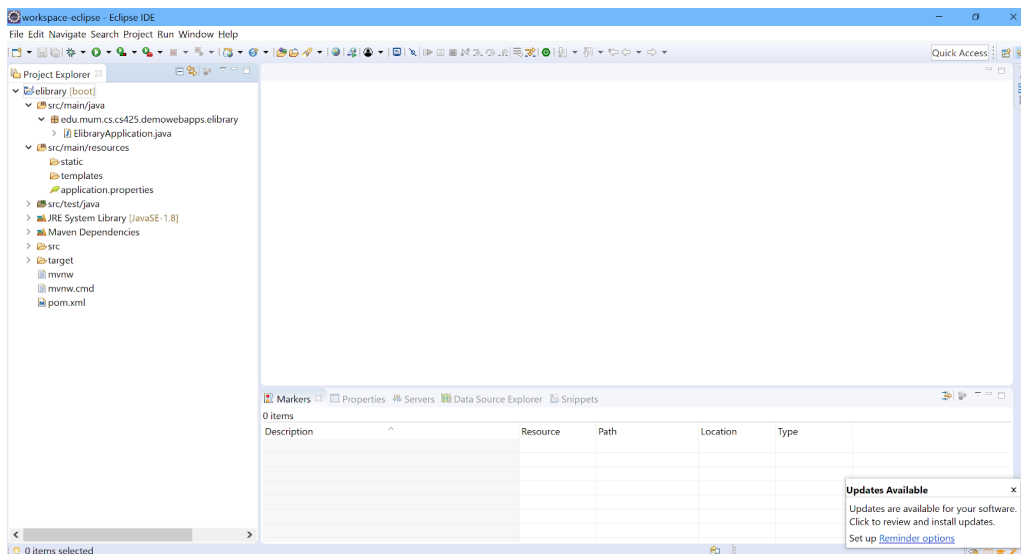
- ☐ Add project to working sets
- Working sets: []

Buttons at the bottom: ? < Back Next > Finish Cancel

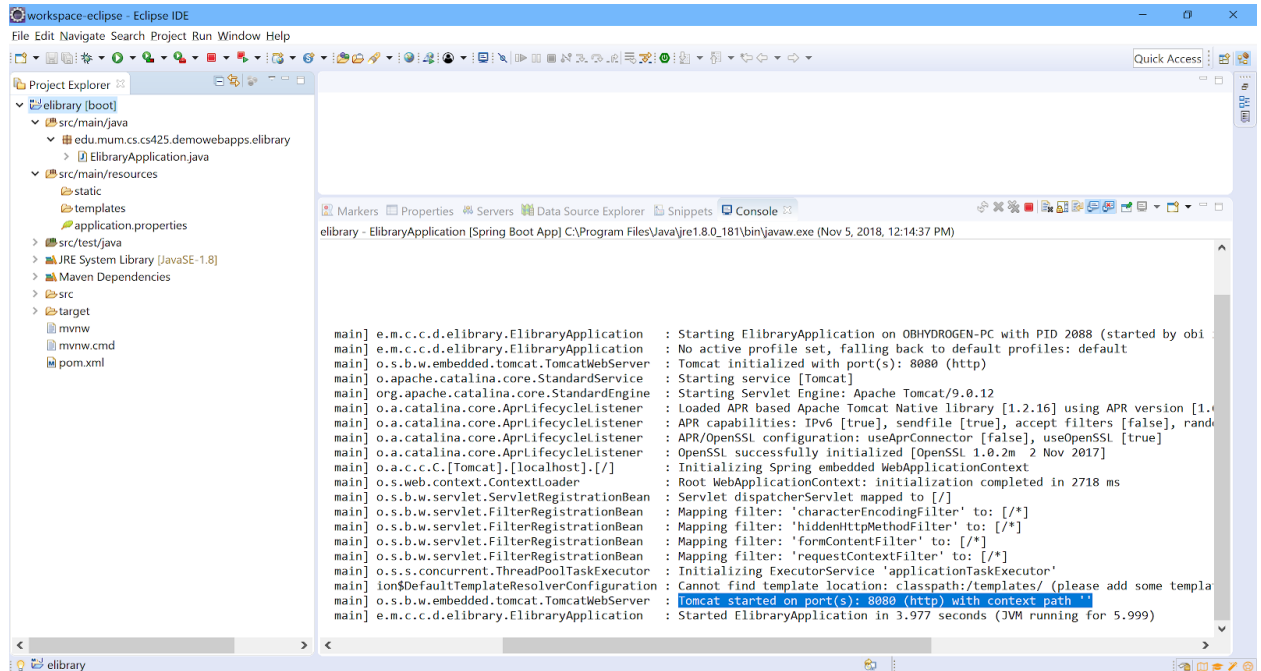
6. For the project Dependencies, select the 2 items (Web and Thymeleaf), as shown below:



7. On clicking "Finish", the generated Spring Boot webapp project should show-up within your Eclipse as can be seen below:



8. To run (i.e. execute) the project in Eclipse, simply right-click on the project, go to "Run As" and select "Spring Boot App".
9. The project is then executed and the embedded tomcat engine is started on the default port number, 8080, as shown below:



10. However, opening a web browser and navigating to the url - <http://localhost:8080/>, produces a Whitelabel Error page with status code, 404. This is because there is no resource (e.g. a view template resource, like a webpage) available at that url, yet. Therefore, to enable a webpage (e.g. our eLibrary web application's homepage) to be displayed, do the following:
 - 10.1. In Eclipse IDE, go to the Project Explorer, expand/open the "src/main/resources" folder.
 - 10.2. Right-click on the "templates" sub-folder and select "New -> Folder" and add a new folder named, say, home.
 - 10.3. Right-click on this new "home" folder and select "New -> Other" and expand the Web folder and select "HTML File" to create/add a new HTML file. Click "Next" and name the file, index.html. Click "Next" again, select the "HTML 5" template and click "Finish".
 - 10.4. Select and delete the entire generated HTML markup content of the file and replace it with the html markup content provided below and save the file:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <meta name="description" content="">
  <meta name="author" content="O. Kalu">
  <link rel="icon" href="favicon.ico">
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/cerulean/bootstrap.min.css" rel="stylesheet"
integrity="sha384-EmRcsPP774S9MOi4yWB+JXUWJYnBdyInMzG2XsC5siZEUpexqOMJW358dCYEjPu"
crossorigin="anonymous">
  <link rel="stylesheet" type="text/css" th:href="@{/css/master.css}">
</head>
<title>eLibrary - a demo spring boot webapp</title>
```

```

</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
    <a style="padding-right: 2em;" class="navbar-brand" href="/elibrary/home">eLibrary : : : a basic demo
webapp using Spring Boot</a>
    <!--<span style="color: #ffffff;"> &nbsp; | &nbsp;</span>-->
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarColor01"
aria-controls="navbarColor01" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarColor01">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item" style="border-left: white 1px solid; padding-left: 2em; padding-right: 2em;">
          <a class="nav-link" href="/elibrary/book/list">Browse Books</a>
        </li>
      </ul>
      <form class="form-inline my-2 my-lg-0">
        <!--<input class="form-control mr-sm-2" type="text" placeholder="Search">-->
        <!--<a style="color: #ffffff; margin-right: 0em;" class="nav-link" href="#">John Doe</a>-->
        <!--<a href="#" class="btn btn-secondary">Sign In</a>-->
      </form>
    </div>
  </nav>

  <p>&nbsp;</p>
  <div class="container">
    <h3>Welcome to the eLibrary Management system (eLMS)</h3>
    <hr/>
    <div style="font-size: 1em">
      To browse and manage the <a href="/elibrary/book/list">catalogue of Books</a>, please first make sure
to read the
      instructions and key points provided below. And then click the
      appropriate menu item, located on the menu-bar above.<br/><br/>

      Lorem Ipsum is simply dummy text of the printing and typesetting
      industry. Lorem Ipsum has been the industry's standard dummy text
      ever since the 1500s, when an unknown printer took a galley of type
      and scrambled it to make a type specimen book. It has survived not
      only five centuries, but also the leap into electronic typesetting,
      remaining essentially unchanged. It was popularised in the 1960s with
      the release of Letraset sheets containing Lorem Ipsum passages, and
      more recently with desktop publishing software like Aldus PageMaker
      including versions of Lorem Ipsum.
    </div>
  </div>

  <!-- <footer class="footer">
    <div class="container">
      <span class="text-muted">O. Kalu : : : CS425-SWE</span><span style="float:right;"
class="text-muted">&copy; November 2018</span>
    </div>
  </footer> -->

  <!-- Bootstrap core JavaScript
===== -->
  <!-- Placed at the end of the document so the pages load faster -->

```

```

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
integrity="sha384-ZMP7rVo3mlykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPiPm49"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"
integrity="sha384-smHYKdLADwkXOn1EmN1qk/HfnUcbVRZyYmZ4qpPea6sjB/pTJ0euyQp0Mk8ck+5T"
crossorigin="anonymous"></script>
</body>
</html>

```

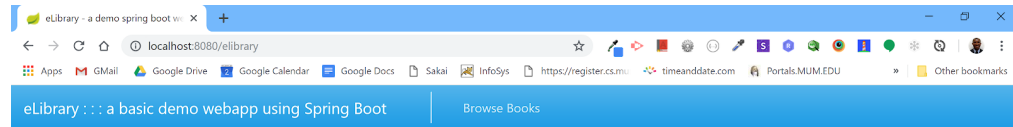
- 10.5. In the Project Explorer, expand/open the "src/main/java" folder; right-click on the package named, "edu.mum.cs.cs425.demowebapps.elibrary" and select "New -> Package" and create a new (sub)package named, edu.mum.cs.cs425.demowebapps.elibrary.controller.
- 10.6. Right-click on the "controller" (sub)package, select "New -> Class" and add a Java class named, HomePageController and click "Finish".
- 10.7. Add the couple of import statements, annotations and the one method (named, displayHomePage) into the HomePageController class, as shown below and save the file:

```

1 package edu.mum.cs.cs425.demowebapps.elibrary.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5
6 @Controller
7 public class HomePageController {
8
9     @GetMapping(value = {"/", "/elibrary", "/elibrary/home"})
10    public String displayHomePage() {
11        return "home/index";
12    }
13
14 }

```

- 10.8. Now, re-run the project and when the embedded tomcat engine has started, simply access the elibrary webapp by opening your browser to the url, <http://localhost:8080/> (or <http://localhost:8080/elibrary> or <http://localhost:8080/elibrary/home>). The elibrary web application's home page should be displayed as shown below:



11. If you got this home page displayed, then Congratulations! You've successfully created a web application using Spring Framework 5.x.x, with Spring Boot 2.x.x and Thymeleaf 3.x.x.
12. The End!!!