**Web development using spring MVC**

**Sample web app:**

1. Controllers and Views
2. Annotations
   1. @Controller
   2. @RequestMapping

|  |  |  |
| --- | --- | --- |
| **Row number** | **Controller** | **Views** |
| 1 | dogHome() | dogHome.html |
| 2 | dogSchool() | dogSchool.html |
| 3 | dogPlay() | dogPlay.html |
| 4 | dogIndex() | dogIndex.html |
| 5 | dogTreatment() | dogTreatment using JSP (need Tomcat Jasper dependency) |
| 6 | dogFood() | dogFood.html using Thymeleaf  <html xmlns:th= <http://www.thymeleaf.org>> |

**Web Application for Dog Management System**

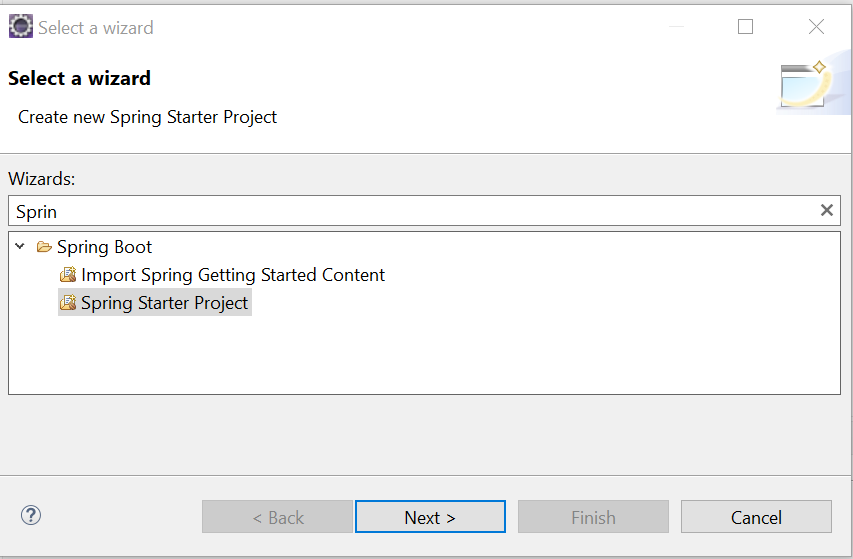
**Step-1:** Create a Spring boot Starter Project named “**SampleWebAppDemo**” project which requires the following three packages:

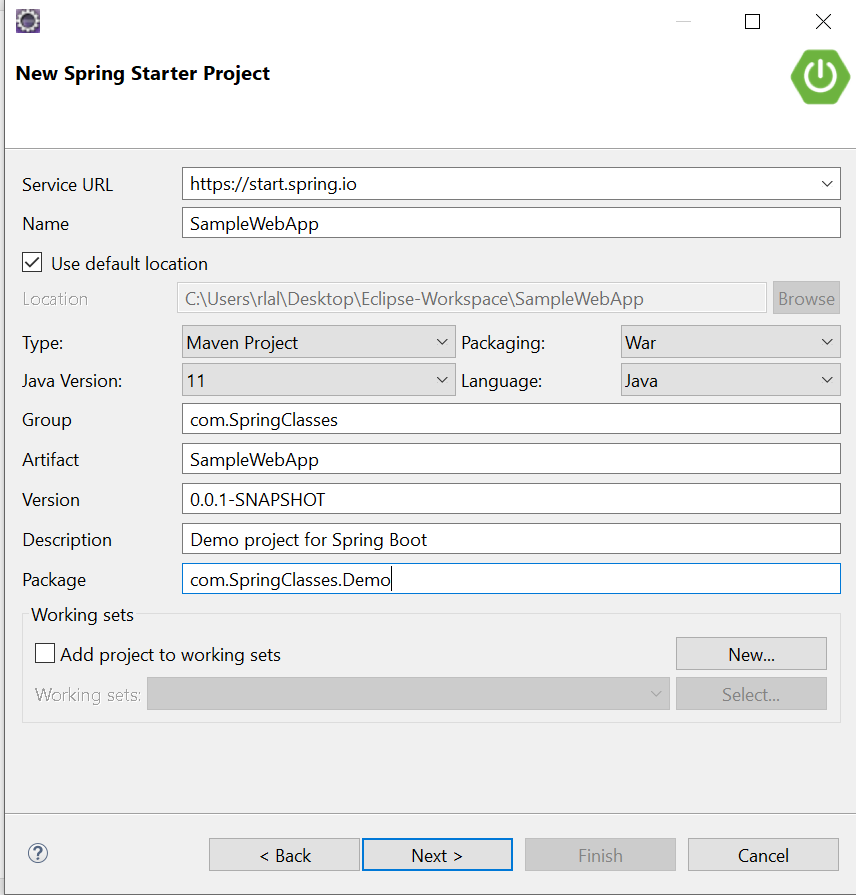
1. Spring Web
2. Spring Boot DevTools
3. Thymeleaf

**Note:** You should have installed enterprise and web applications version of eclipse. (When you open eclipse exe file, you should select the second option for the support of enterprise and web applications).

**Steps to follow for creating a spring boot starter project for the web application:**

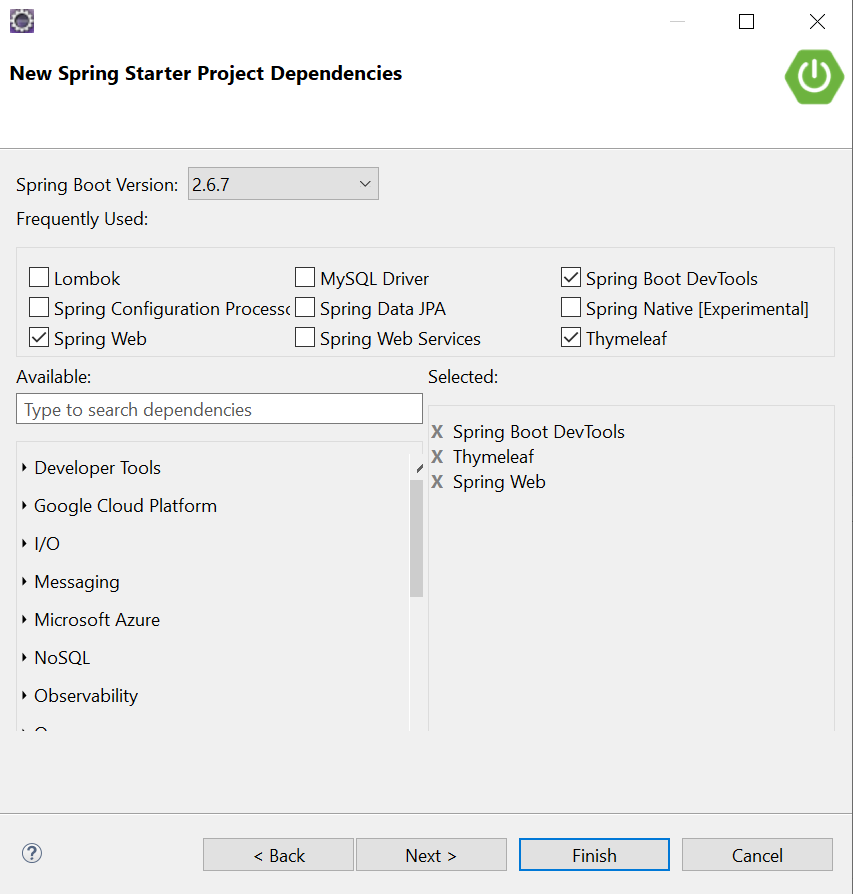
1. Click on Other… via File -> New -> Other…; you should see the following:



1. Click on the next in the above, you should see the following:

In the above image, note that Package is not the same as the project name that you entered in the Name. If you also have the same, then please ensure that you have the same group and project name in the **Package** box. In addition, for the web development, please make sure that you choose **War** option in the Packaging.

1. Click on the next in the above. You should see the following:



Please select three basic dependencies, namely, Spring Web, Spring Boot DevTools and Thymeleaf which are required for the development of a web application.

**Step-2:** Create a controller class named “**DogController**”, in which you develop all the controllers related to the dog.

1. Add the class level annotation **@Controller** above the **DogController** class
2. Define the first controller method named “**dogHome**” that returns String “**dogHome.html**”.

@RequestMapping ("dogHome")

**public** String dogHome() {

**return** "dogHome.html";

}

1. Run your application
2. Open a web browser and enter the url: **“localhost:8080/dogHome”**
3. You may see **Thymeleaf** error on the webpage
4. Please comment Thymeleaf dependency in pom.xml file
5. Run your application and refresh the url: **“localhost:8080/dogHome”**
6. Since dogHome.html is not defined, you may see not found error
7. Create dog.html file within src/main/webapp folder
8. Change Title with “Dog Home Page” and body with “This is a dog home page!”
9. Similarly, please create **dogSchool** and **dogPlay** controller methods and **dogSchool.html** and **dogPlay.html** view pages with your preferred message
10. Run your application and refresh the url: **“localhost:8080/dogHome”**
11. Try **“localhost:8080/dogSchool”** and **“localhost:8080/dogPlay”**
12. Please create dogTreatment controller method and **dogTreatment.jsp** page with your preferred message
13. To render JSP page, we need to add **tomcat-embed-jasper** dependency in pom.xml file

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

<version>9.0.60</version>

</dependency>

**Note:** the version should match with tomcat version that you can find in the **“Maven Dependencies”** folder

1. Run your application and refresh the url: **“localhost:8080/dogTreatment”**
2. Please create dogFood controller method and **dogFood.html** view page in the **src/main/templates** folder with your preferred message
3. Add **Thymeleaf** in **dogFood.html** file within html tag

<html xmlns:th=*"http://www.thymeleaf.org"*>

1. Run your application and refresh the url: **“localhost:8080/dogFood”**
2. You may get an error because additional setting is required for running both **jsp** and **thymeleaf** pages
3. At this point, please comment **thymeleaf** dependency in **pom.xml** file and remove xmlns:th=*"http://www.thymeleaf.org"* part from dogFood.html

You have seen the interaction between controller and view pages. Next, we will introduce models.

**WebApp using MVC**

**Parameters:**

* HttpServletRequest
* HttpSession

**ModelAndView class**

**Step-1:** Create a Spring boot Starter Project named “**ParametersDemo**” project which requires the following three packages:

1. Spring Web
2. Spring Boot DevTools
3. Thymeleaf

**Step-2:** Create a sub-package named **Controller** under **com.SpringClasses.ParametersDemo**.Then, create a **dogController** class under “**com.SpringClasses.ParametersDemo.Controller**” package

**Step-3:** Please create **display** controller method and **display.html** view page using Thymeleaf template

@RequestMapping("display")

**public** String display(HttpServletRequest request)

{

HttpSession session = request.getSession();

String dogNameIn = request.getParameter("dogName");

session.setAttribute("dogNameDisplay", dogNameIn);

**return** "display.html";

}

In display.html, please add the following line in the body:

<p th:text="Hello+' '+${session.dogNameDisplay}"></p>

**Step-4:** Run your application and refresh the url: **“localhost:8080/display?dogName=Puppy”**

**Step-5:** Modify display controller method as given below:

@RequestMapping("display")

**public** String display(String dogName, HttpSession session) {

session.setAttribute("dogNameDisplay", dogName);

**return** "display.html";

}

**Step-6:** Run your application and refresh the url: **“localhost:8080/display?dogName=Puppy”**

**Step-7:** Modify display controller method as given below:

@RequestMapping("display")

**public** String display(@RequestParam("maskedName") String dogName, HttpSession session) {

session.setAttribute("dogNameDisplay", dogName);

**return** "display.html";

}

**Step-8:** Run your application and refresh the url: **“localhost:8080/display?maskedName=Puppy”**

**Step-9:** Modify display controller method as given below:

@RequestMapping("display")

**public** ModelAndView display(String dogName) {

ModelAndView mv = **new** ModelAndView();

mv.setViewName("display.html");

mv.addObject("dogNameDisplay", dogName);

**return** mv;

}

**Step-9:** Modify display.html as given below:

<p th:text=*"Hello+'-----'+${dogNameDisplay}"*></p>

**Note:** If you are copying the above code, then replace single quote by your own single quote.

**Step-10:** Run your application and refresh the url: **“localhost:8080/display?dogName=Puppy”**

**Step-11:** Create a sub-package named **Model** under **com.SpringClasses.ParametersDemo**. Then, create a class named **Dog** having three attributes, named dogId, dogName and dogBreed and their getter and setter method in the package **com.SpringClasses.ParametersDemo.Model**

**Step-12:** Modify display controller method as given below:

@RequestMapping("display")

**public** ModelAndView display(Dog d)

{

ModelAndView mv = **new** ModelAndView("display.html");

mv.addObject("dogNameDisplay", d);

**return** mv;

}

**Step-13:** Modify display.html as given below:

<p th:text=*"Hello+'-----'+${dogNameDisplay.dogId}+' '+${dogNameDisplay.dogName}+’ ’ +${dogNameDisplay.dogBreed}"*></p>

**Note:** If you are copying the above code, then replace single quote by your own single quote.

**Step-14:** Run your application and refresh **“localhost:8080/display?dogId=12&dogName=Puppy&dogBreed=Husky”**