





spring

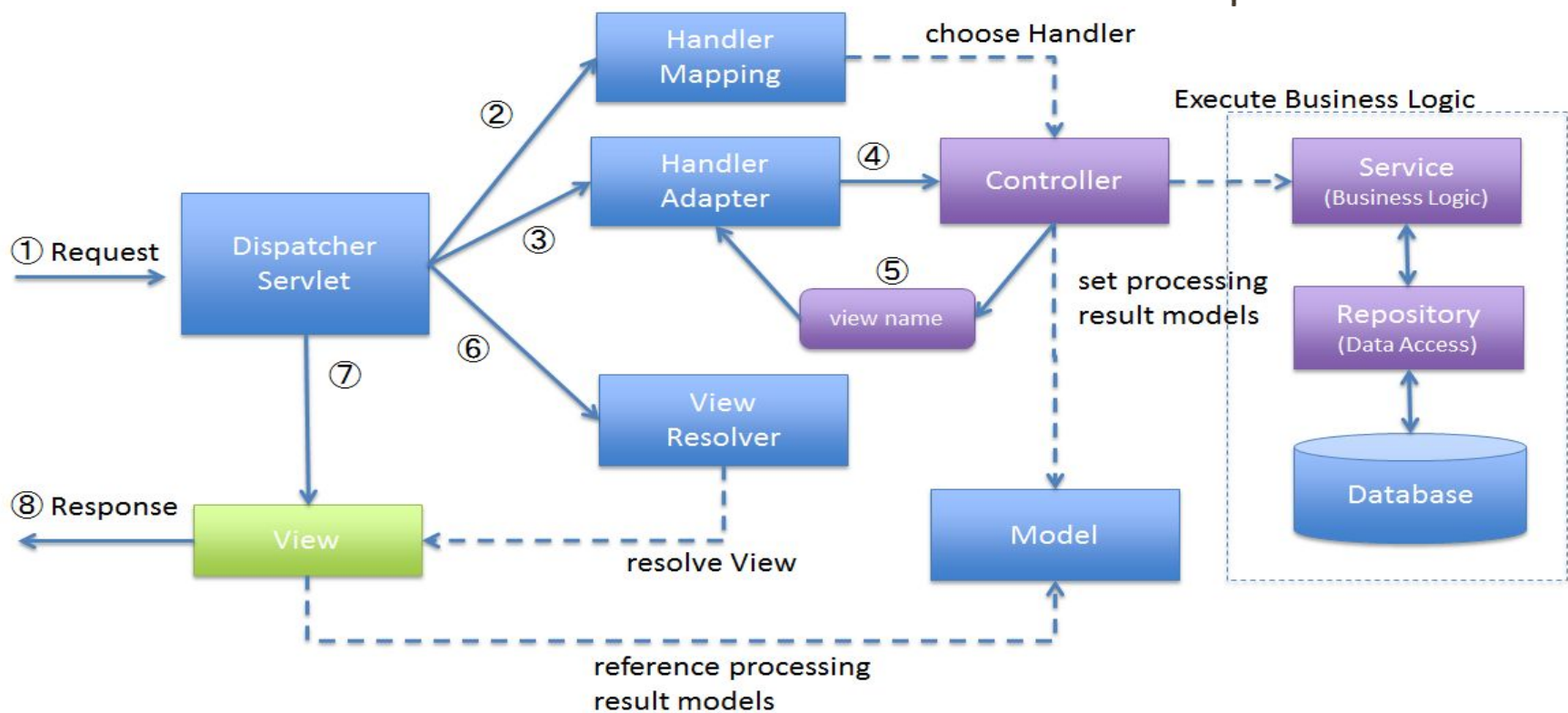
FRAMEWORK



What is Spring Framework?

- ✓ Series of technologies that make building web applications using Java simpler and more streamlined
- ✓ Manages instantiation of objects & connects them the way they are intended to work—efficiently
- ✓ Provides database connectivity through API & Services





1

Setting Up Your Environment

Installing Spring Tool Suite or Add-On



Spring Tool Suite

Go to [this site](#), then:

- ✓ Determine which installation method is most appropriate for you
 - Spring Suite?
 - IDE plug-in?



THE STS Interface

Explorer / Navigation Pane

- *File paths*

Workspace

- *Where files are edited*

Document Outline

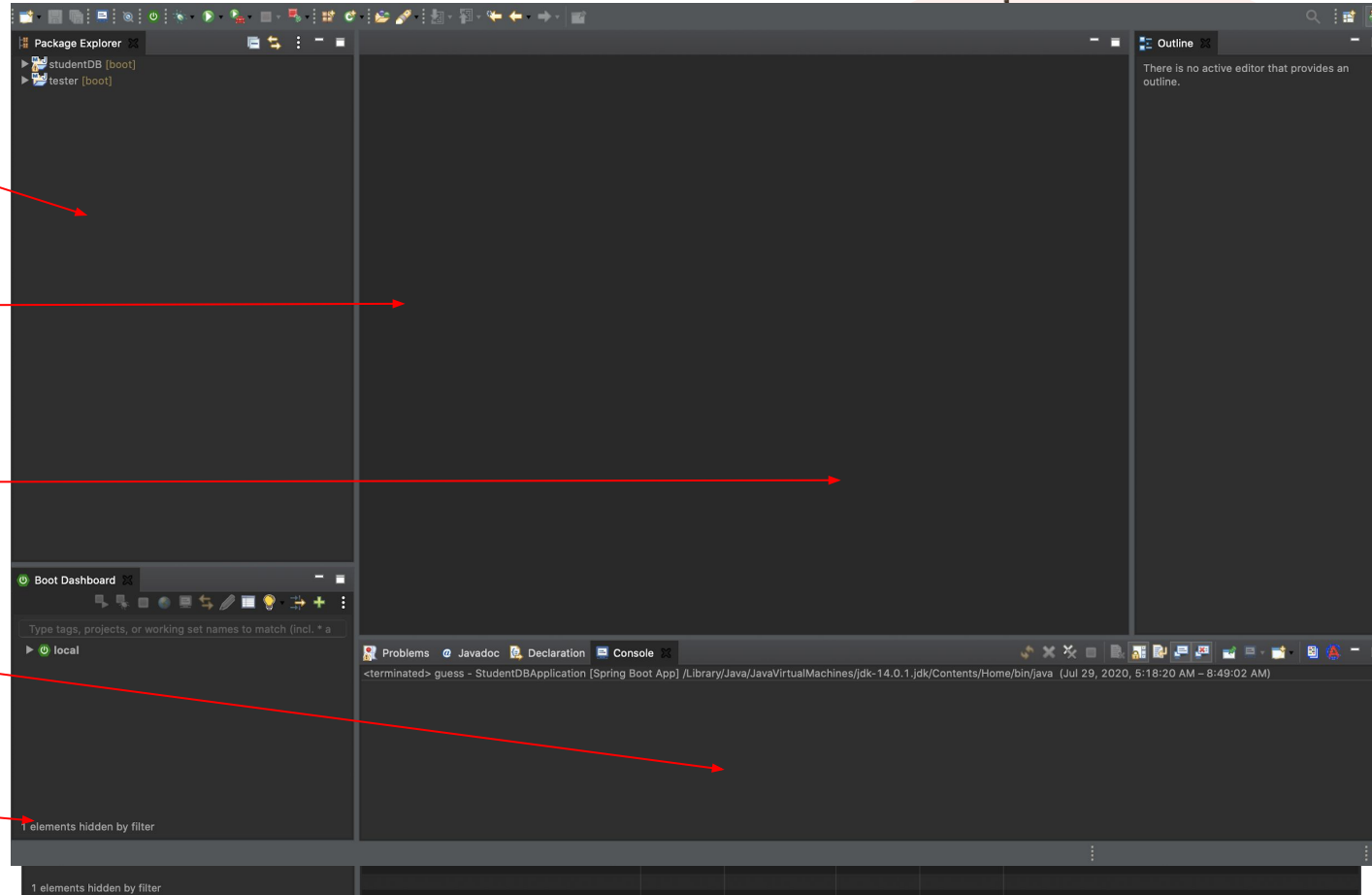
- *Specific document sections*

Console

- *Command-Line UI*

Boot Dashboard

- *find servers & other services*



THE STS Interface

Explorer / Navigation Pane

- *File paths*

Workspace

- *Where files are edited*

Document Outline

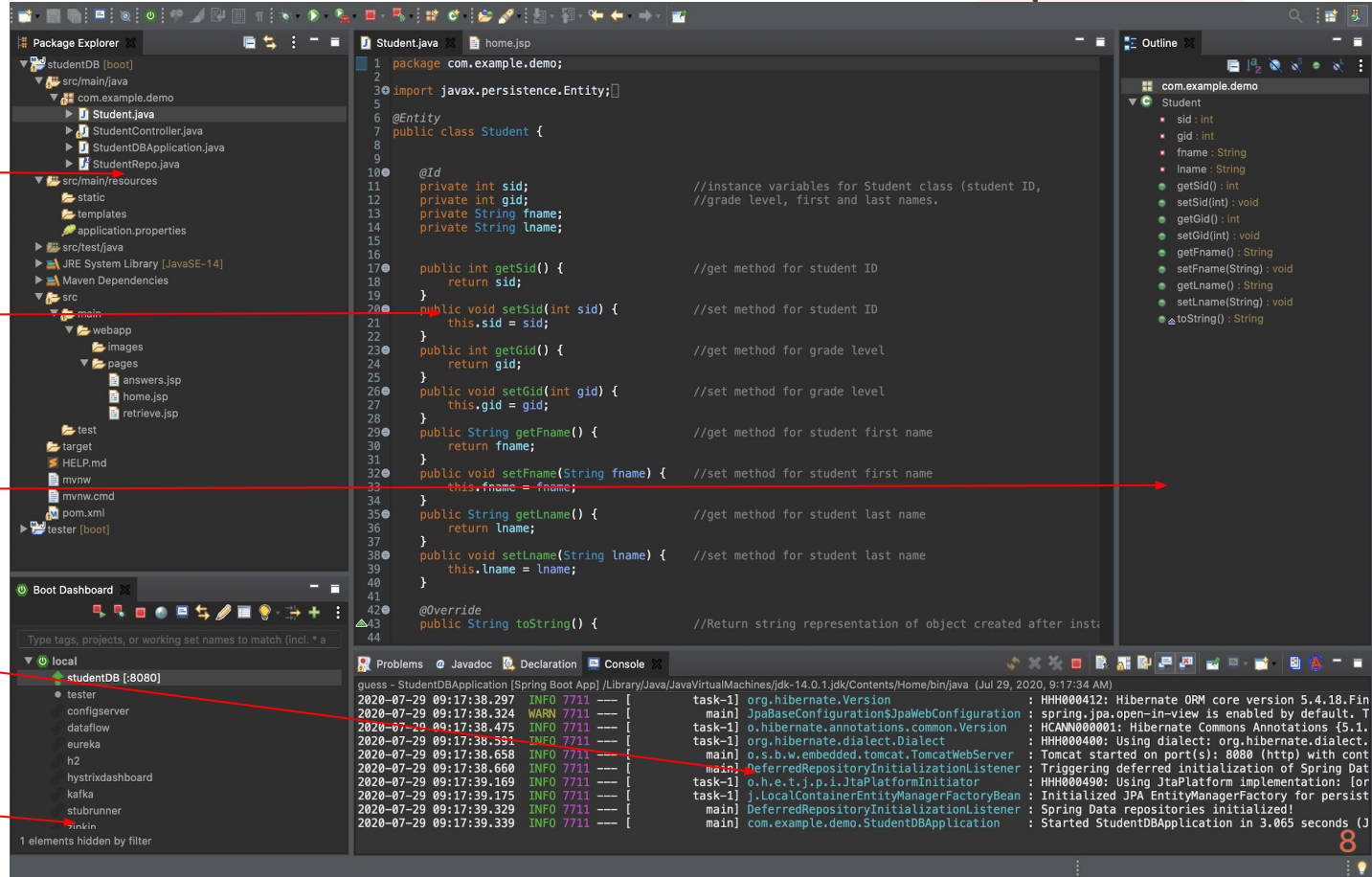
- *Specific document sections*

Console

- *Command-Line UI*

Boot Dashboard

- *find servers & other services*



2

Creating Your Project

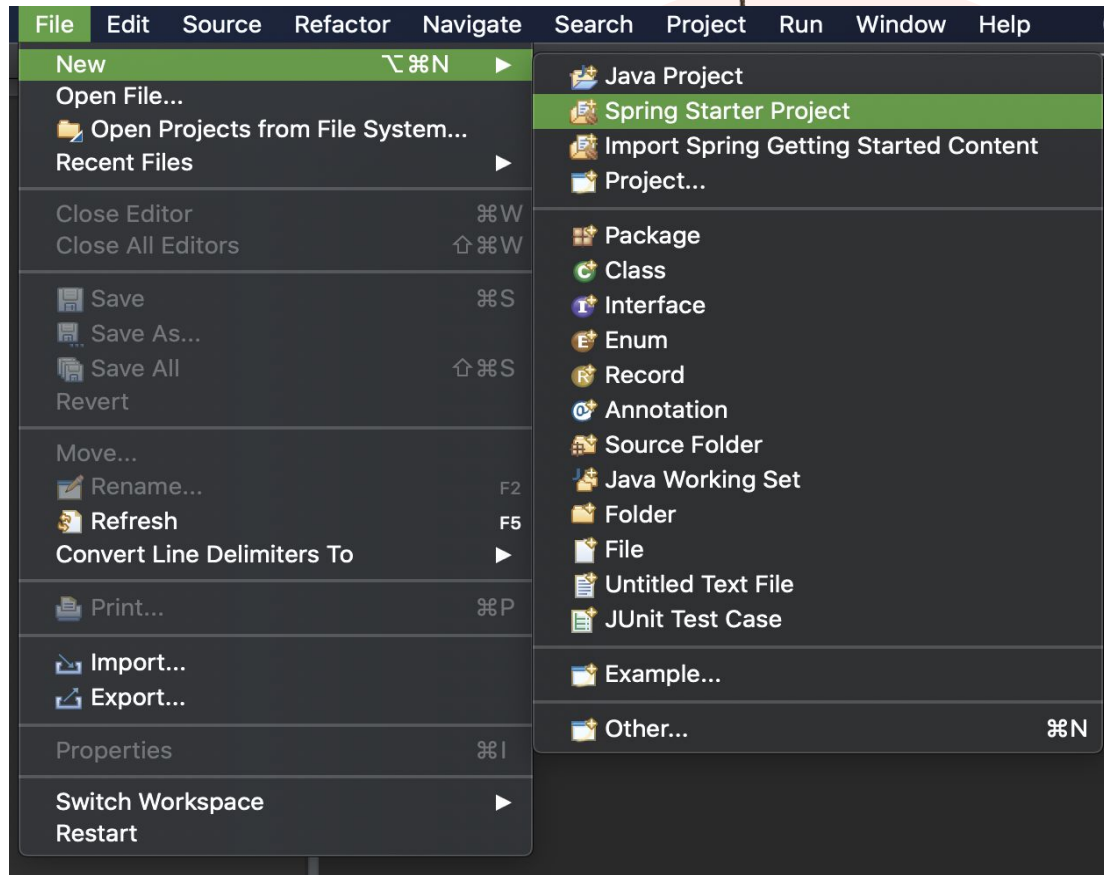
Packages with Dependencies



Start A New “Starter Project”

Steps

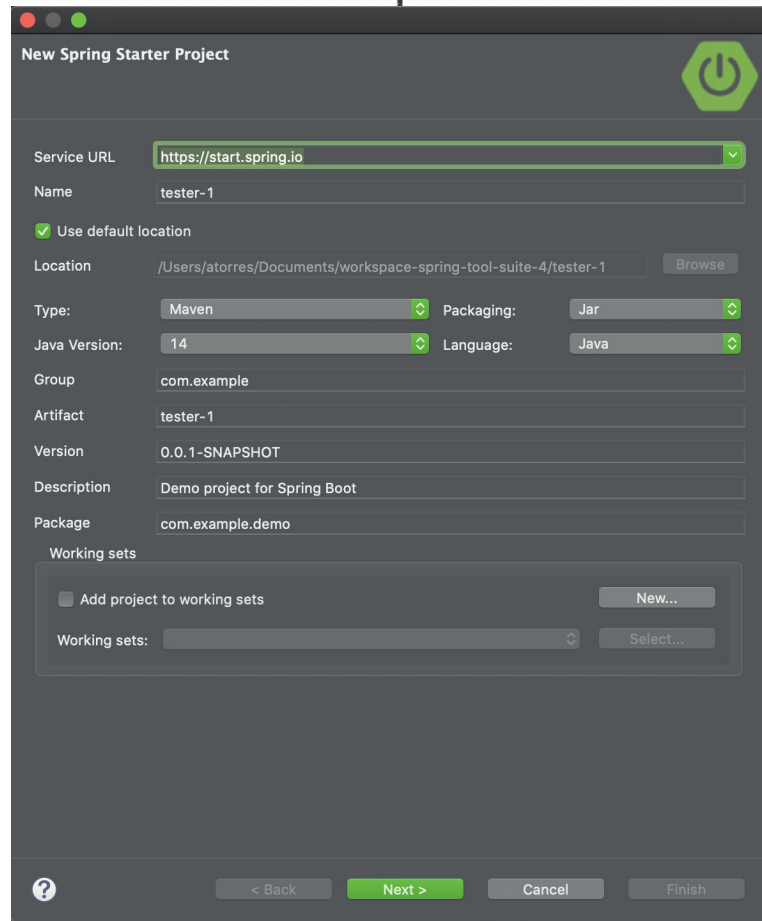
- ✓ Go to **File**
- ✓ Select **New**
- ✓ Select **Spring Starter Project**



Name Your Project

Steps

- ✓ Give your project a **name**
- ✓ Select **Java v. 14** (or whatever version you have)
- ✓ Select **Next**



The screenshot shows the 'New Spring Starter Project' dialog box in Spring Tool Suite. The dialog is titled 'New Spring Starter Project' and features a green power button icon in the top right corner. The fields are as follows:

- Service URL:** <https://start.spring.io>
- Name:** tester-1
- Use default location:** ☒
- Location:** /Users/atorres/Documents/workspace-spring-tool-suite-4/tester-1 (with a 'Browse' button)
- Type:** Maven
- Packaging:** Jar
- Java Version:** 14
- Language:** Java
- Group:** com.example
- Artifact:** tester-1
- Version:** 0.0.1-SNAPSHOT
- Description:** Demo project for Spring Boot
- Package:** com.example.demo

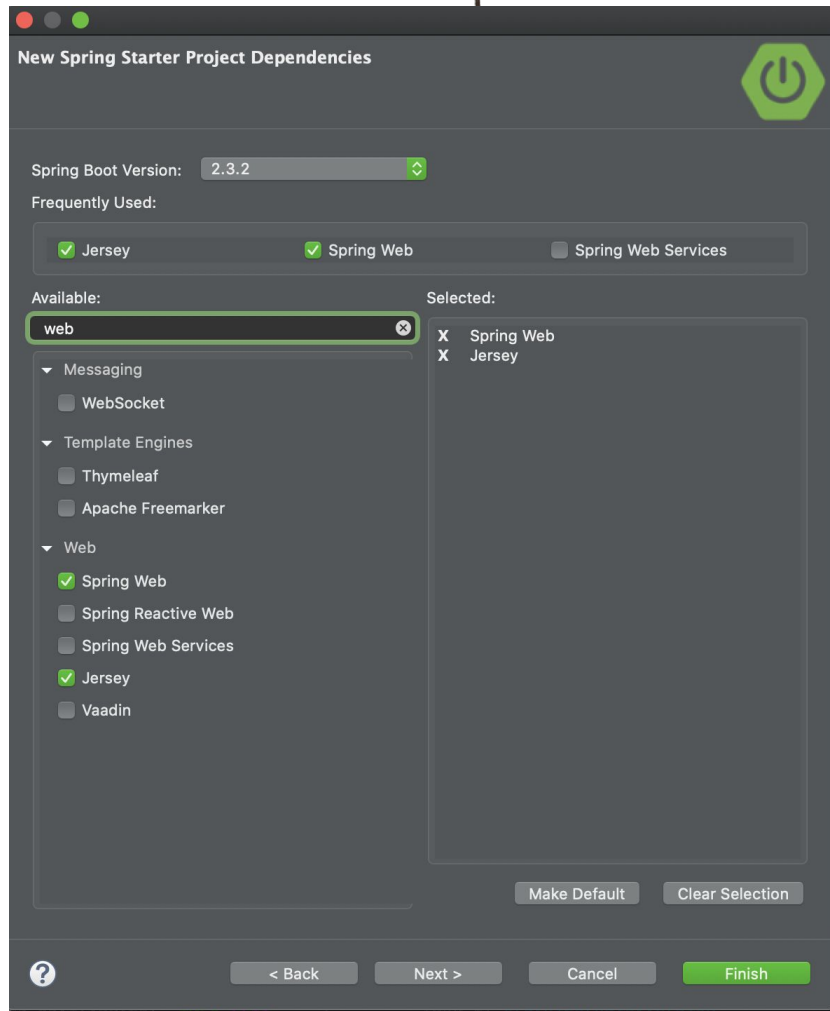
Below these fields is a 'Working sets' section with a checkbox 'Add project to working sets' (unchecked), a 'New...' button, a 'Working sets:' dropdown menu, and a 'Select...' button.

At the bottom of the dialog are four buttons: a help button (question mark), '< Back', 'Next >' (highlighted in green), 'Cancel', and 'Finish'.

Select Your “dependencies”

Steps

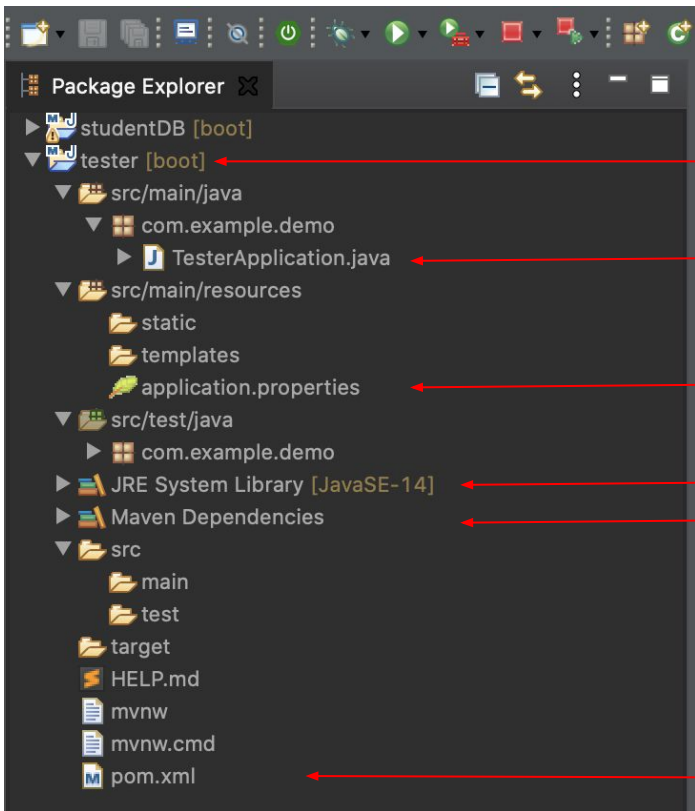
- ✓ Type in “**web**” in the “**Available:**” search bar
- ✓ Under “web” check off **Spring Web** and **Jersey**
- ✓ Select **Finish**



Navigation & Paths

Understanding the Package Explorer





Boot (root) folder for your project

Java files for project found in **src/main/java**

Properties management found in **src/main/resources**

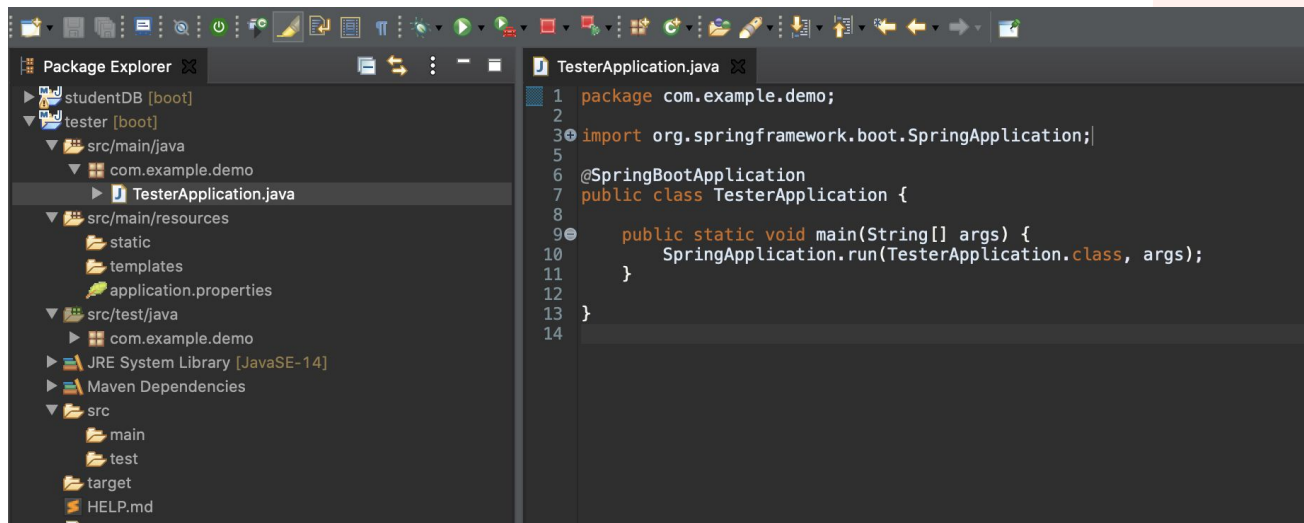
Pre-loaded JRE library & Maven Dependencies

Pom.xml is where you can update/add dependencies you may need later on

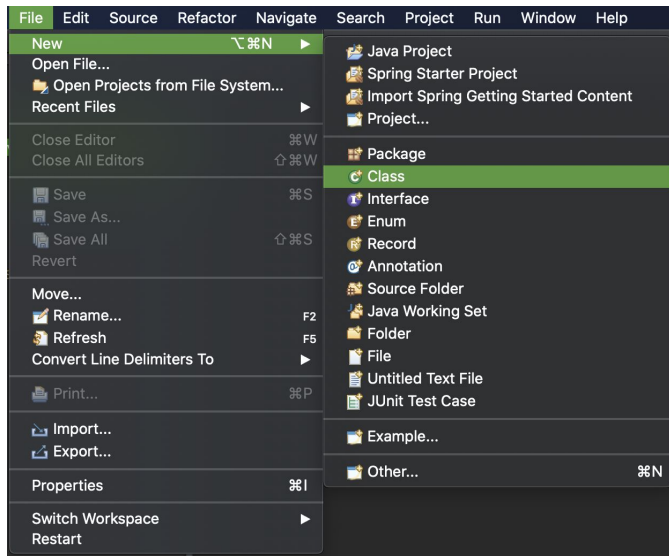


How do you read the base file?

- ✓ This file is your first Java File.
- ✓ You can run a hello world to see how it works, but more than likely it will become your controller file.
- ✓ This file “runs” your program.



How do you create new files?



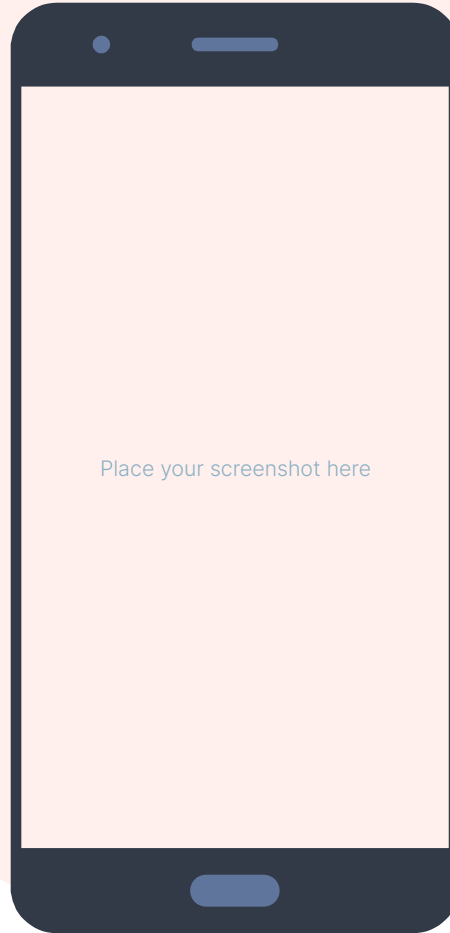
- ✓ Go to **File**
- ✓ Select **New**
- ✓ Scroll down to **Class**
- ✓ **Name** the class whatever you deem appropriate



A woman with curly hair and glasses is sitting on a light-colored sofa, leaning back against a cushion. She is using a laptop on her lap. The room has a large window in the background with a plant and a white air conditioner unit. The floor is covered with a red shag rug.

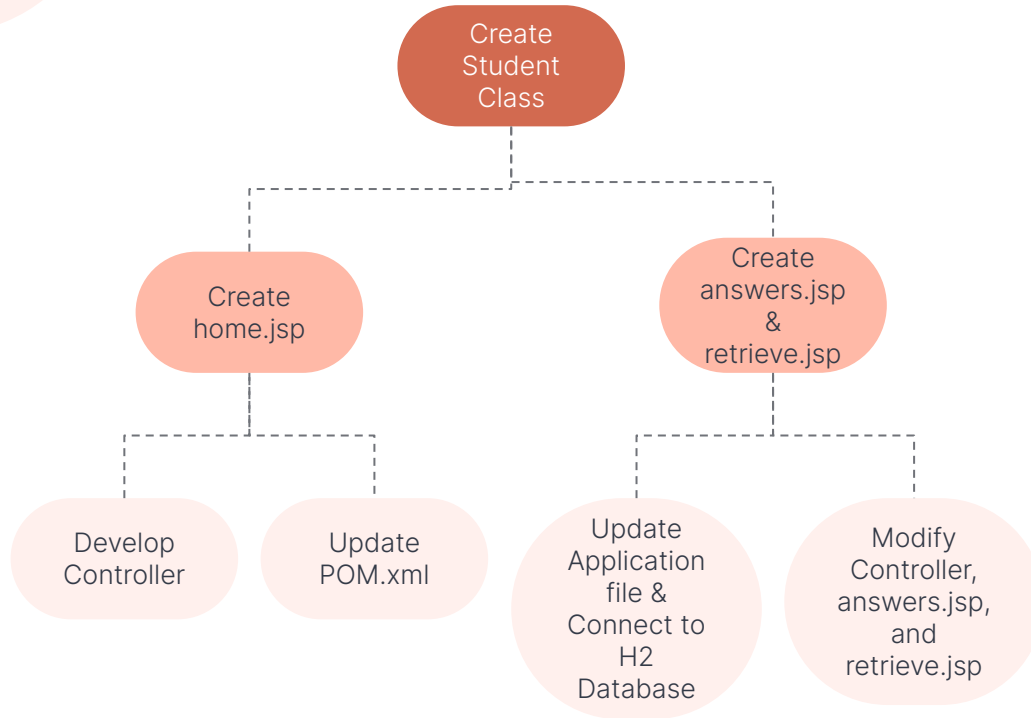
The Web App Project

Mobile App project



- ✓ Create the Skeleton Java Class
- ✓ Create the HTML Interface
- ✓ Develop the Database Connection
- ✓ Launch your first WebAPP!!!

Project Overview

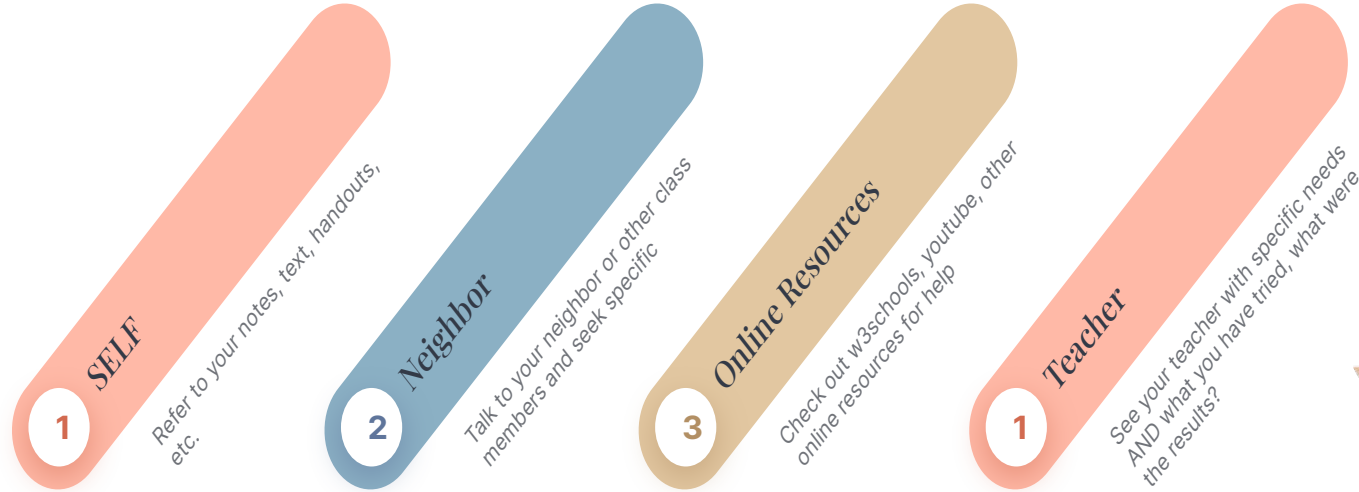


Time Expenditure Overview

TOPIC	#of Classes	Class Day
Introduction & Set Up (including "hello world" Test)	1	1
Create Student Class	<1	2
Create JSP files (including mini teach on HTML)	<1	2
Develop Controller Class (including mini teach on annotations)	1	3
Update POM XML (including mini-teach on Maven Repo & Library/Dependency version look-up)	1	4
Update Application & Connect to H2 (including mini teach on properties & UI)	1	5
Update Controller & JSP Files (including mini teach on annotations)	1	6
Individual Web App (Students ideate and create their own APP or expand the one started in class)	4	7-11



SNOT: Our Process Is Recursive



Thanks!

Any questions?

You can find me at:

- ✓ sabiustech@gmail.com
- ✓ slack

