

A large, light gray play button icon is positioned on the left side of the slide. It consists of a white right-pointing triangle centered within a series of concentric circles in shades of gray.

# Creating a Spring Boot Web App

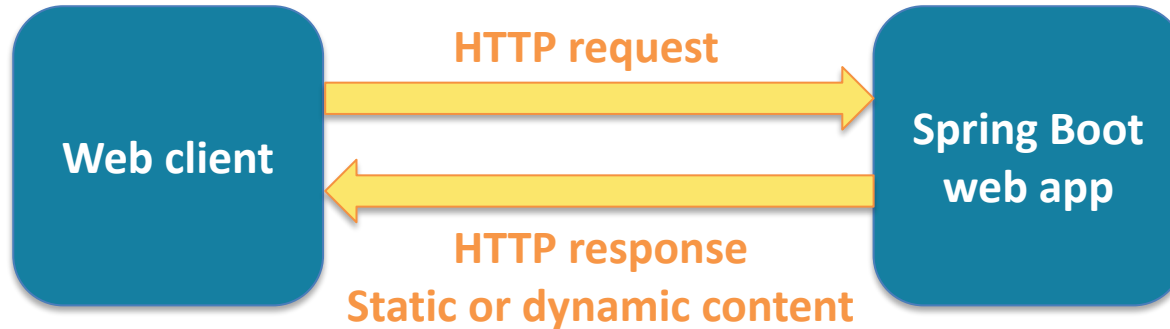
1. Creating a web app project in IntelliJ
2. Understanding the web app
3. Defining application properties

# 1. Creating a Web App Project in IntelliJ

- Overview
- Creating a Spring Boot web project
- Specifying project dependencies

# Overview

- Spring Boot applications are typically "web apps"
  - Listen for HTTP requests from web client (e.g. a browser)
  - Return static or dynamic content



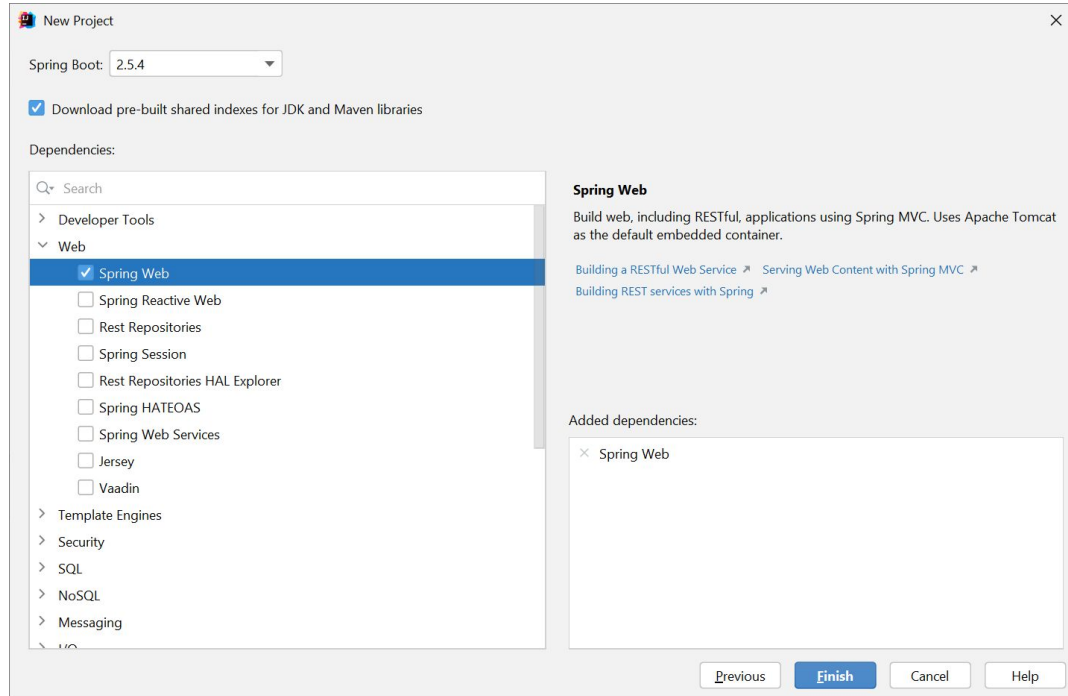
- We'll see how to return **static content** for now
  - Later we'll see how to return dynamic content, via REST services

# Creating a Spring Boot Web Project

- Start IntelliJ, click New Project, and select Spring Initializr
- Specify project info as follows:
  - Enter a suitable project name and location
  - Language - Java
  - Enter a suitable group ID, artifact ID, and package name
  - Project SDK - Java 11
  - Java version - 11
  - Packaging - JAR
  - Then click Next

# Specifying Project Dependencies

- In the next window you can add dependencies to your project
  - Expand **Web** and select **Spring Web**

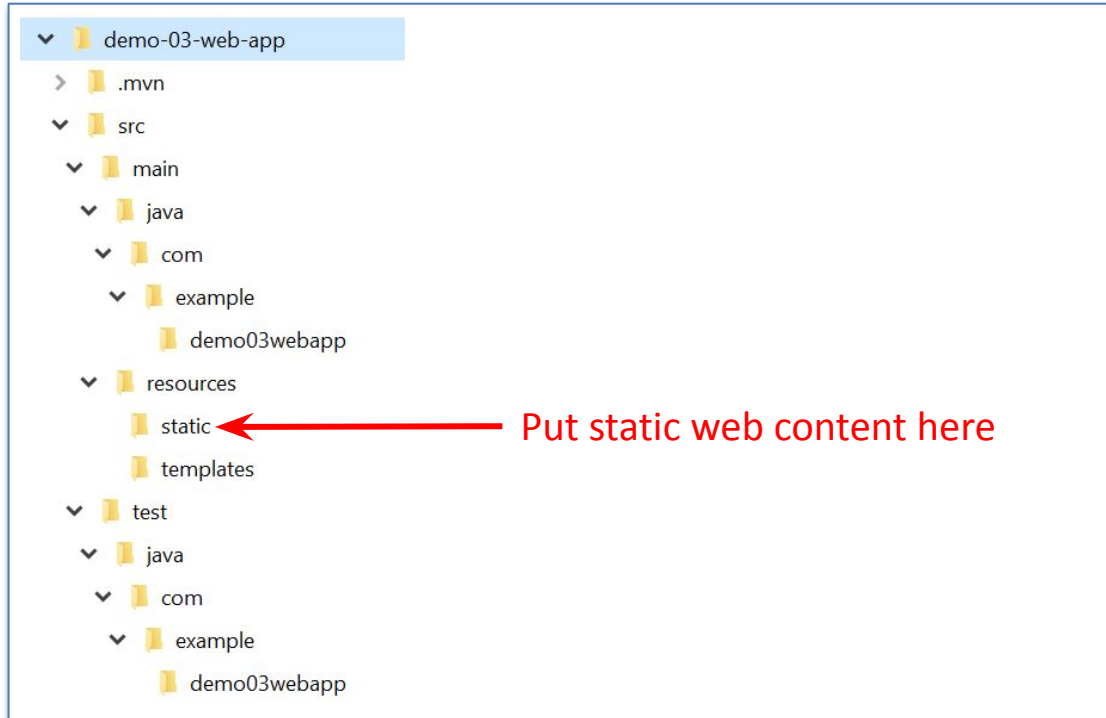


## 2. Understanding the Web Application

- Understanding the web application structure
- Understanding the Maven POM file
- Adding web content
- Running the application
- Pinging the application

# Understanding the Web Application Structure

- The generated application is a regular Maven **web** project:



# Understanding the Maven POM File

- Here's the relevant section in the Maven POM file:

```
<project ... >
...

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
...
```

Spring Boot web dependency

pom.xml



# Adding Web Content

- You can add static web content in the following directory:
  - `src\main\resources\static`
- For example, add an `index.html` file:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Home</title>
</head>
<body>
  Hello world!
</body>
</html>
```

`index.html`

# Running the Application

- To run the application:
  - Right-click the Java app file, then click Run
  - Compiles the code, bundles into a JAR, then runs the JAR
  - The application has an embedded Tomcat web server

```
Actuator
"C:\Program Files\Java\jdk-11.0.9\bin\java.exe" ...

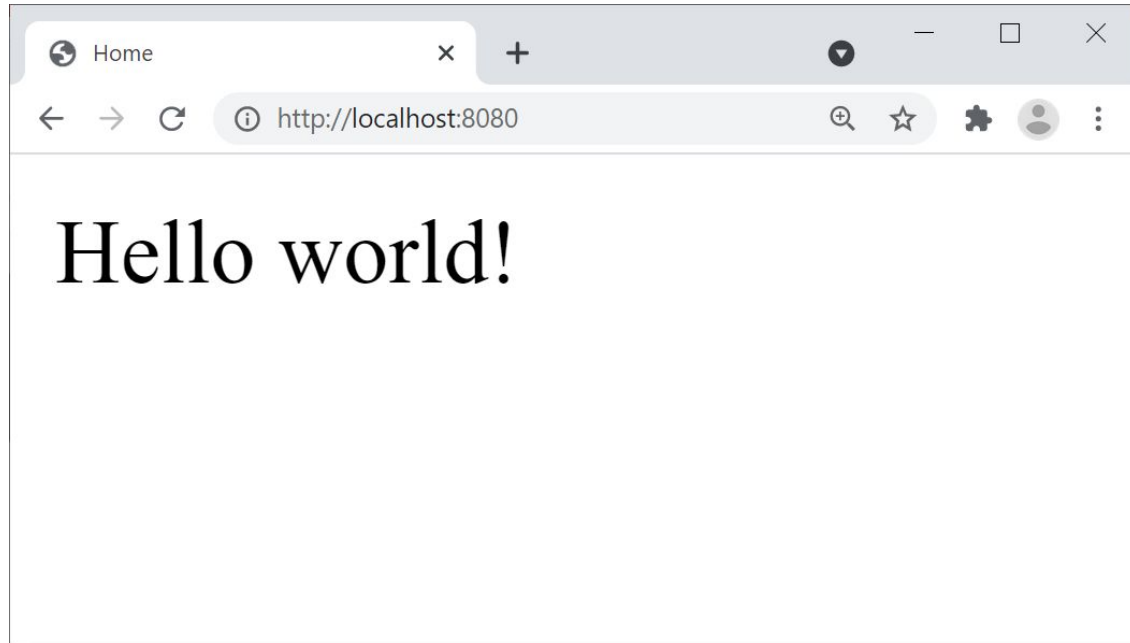
      _--_
     /___ \   _--_   _--_   _--_   _--_
    ( ) ___ | '._| '._| '._| '._| '._|
     \|___||_|_|_|_|_|_|_|_|_|_|_|_|
      '_____|_|_|_|_|_|_|_|_|_|_|_|_|
=====|_|=====|_|_|_|_|_|_|_|_|_|_|
:: Spring Boot ::                (v2.5.4)

2021-09-15 14:14:52.355 INFO 16632 --- [
2021-09-15 14:14:52.358 INFO 16632 --- [
2021-09-15 14:14:53.110 INFO 16632 --- [
2021-09-15 14:14:53.117 INFO 16632 --- [
2021-09-15 14:14:53.117 INFO 16632 --- [
2021-09-15 14:14:53.183 INFO 16632 --- [
2021-09-15 14:14:53.183 INFO 16632 --- [
2021-09-15 14:14:53.375 INFO 16632 --- [
2021-09-15 14:14:53.464 INFO 16632 --- [
2021-09-15 14:14:53.475 INFO 16632 --- [

main] c.e.d.Demo03WebAppApplication : Starting Demo03WebAppApplication using Java 11.0.9 on Andy0-Laptop with PID 16632
main] c.e.d.Demo03WebAppApplication : No active profile set, falling back to default profiles: default
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.52]
main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 756 ms
main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page: class path resource [static/index.html]
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
main] c.e.d.Demo03WebAppApplication : Started Demo03WebAppApplication in 1.508 seconds (JVM running for 2.699)
```

# Pinging the Application

- Open a browser and go to `http://localhost:8080`
  - Renders `index.html` (a standard welcome page in Java web)



A large, light gray play button icon is positioned on the left side of the slide. It consists of a white right-pointing triangle centered within a series of concentric circles in shades of gray.

## 3. Defining Application Properties

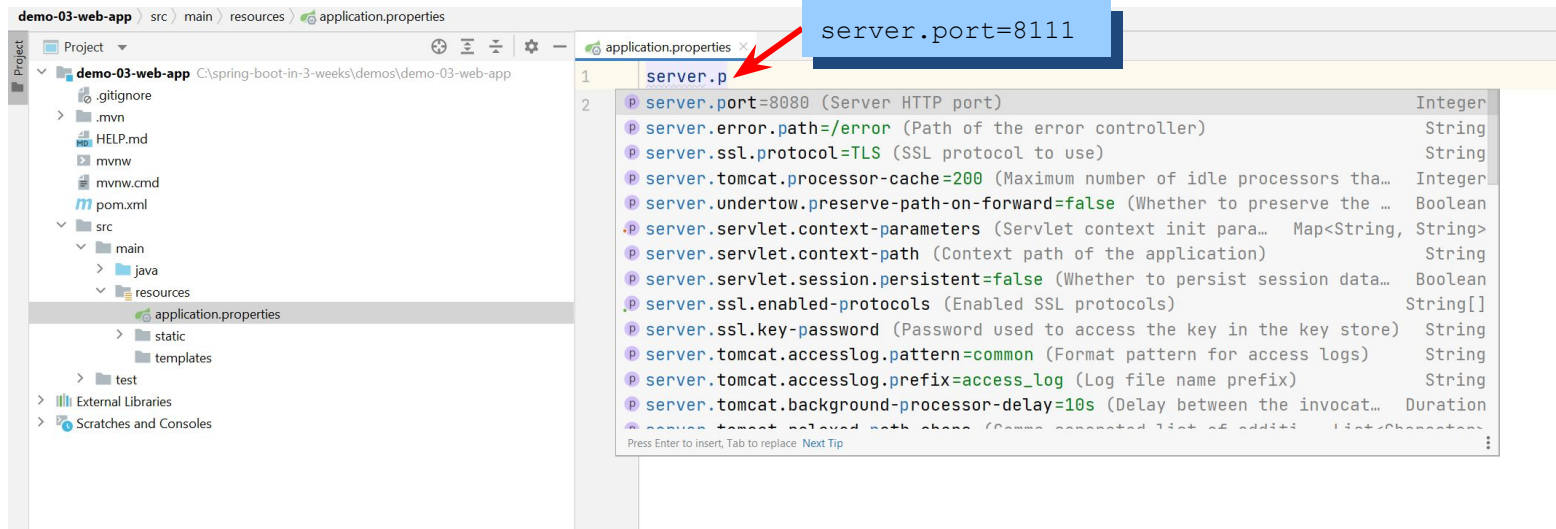
- Overview of application properties
- Editing application properties
- Restarting the application

# Overview of Application Properties

- Spring Boot applications have a standard text file named `application.properties`
  - Very important!
  - The recommended place to set application properties
  - i.e. name=value pairs
- You can also use YAML if you like
  - YAML = "YAML Ain't Markup Language"
  - More on YAML files later...

# Editing Application Properties

- To help you edit `application.properties`, IntelliJ provides a Spring Properties Editor tool
  - Provides nice content assistance and error checking

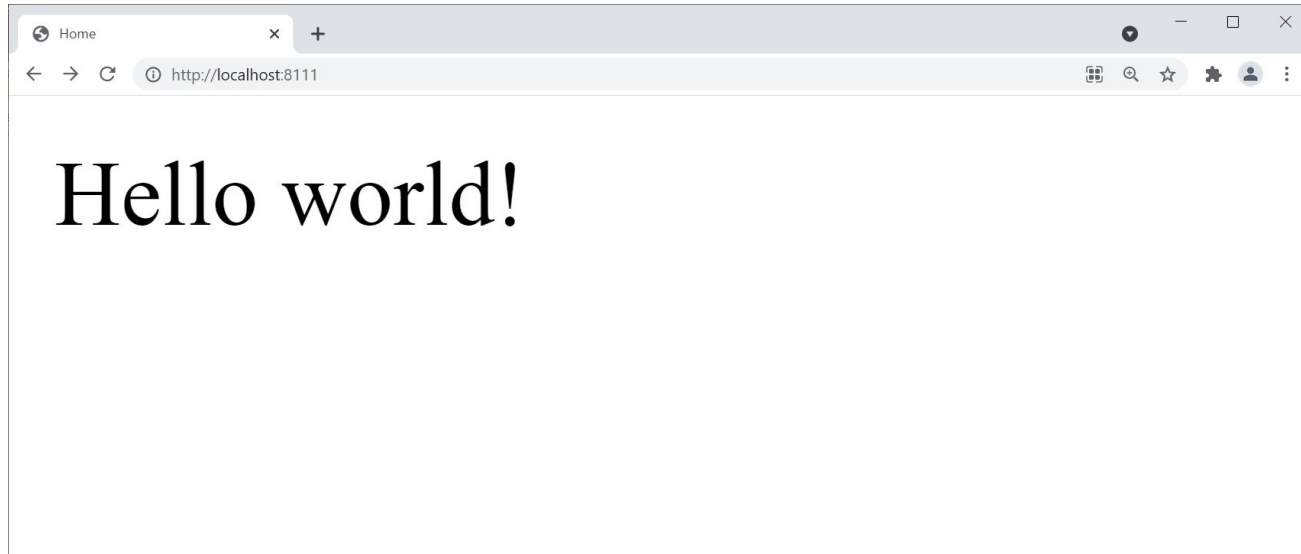


# Restarting the Application

- Restart the application, and verify Tomcat starts on the new port number, 8111

```
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8111 (http) with context path ''
```

- Ping the Web server using the new port number, 8111



A large, light gray play button icon is positioned on the left side of the slide. It consists of a white right-pointing triangle centered within a series of concentric circles in varying shades of gray.

# Summary

- Creating a web app project in IntelliJ
- Understanding the web app
- Defining application properties