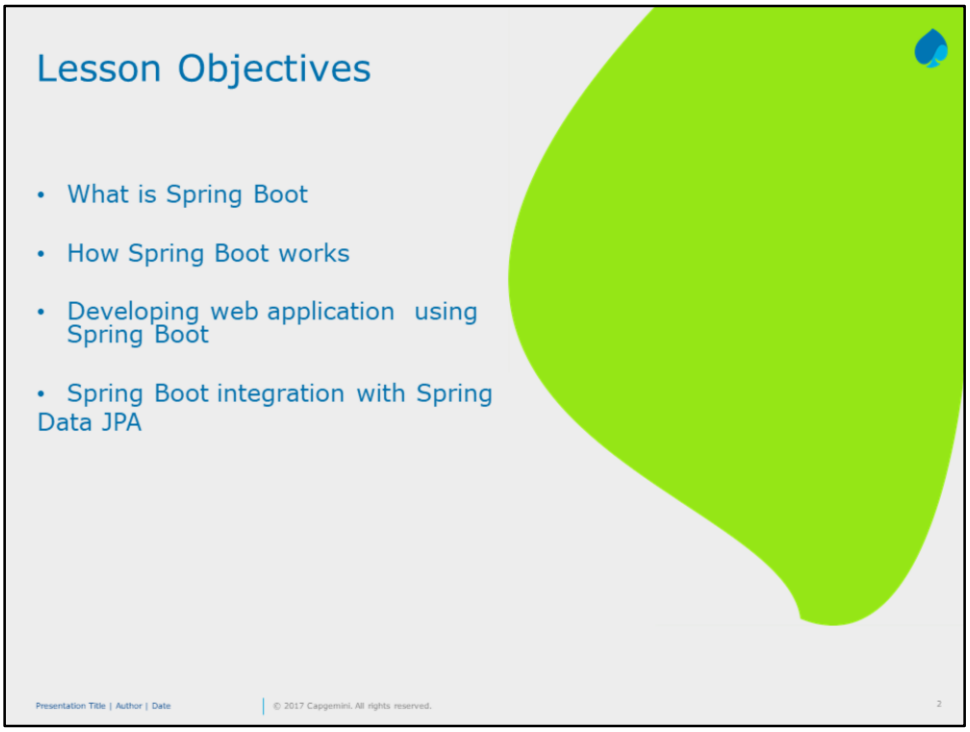
Instructor Notes:

Add instructor notes here.

Instructor Notes:

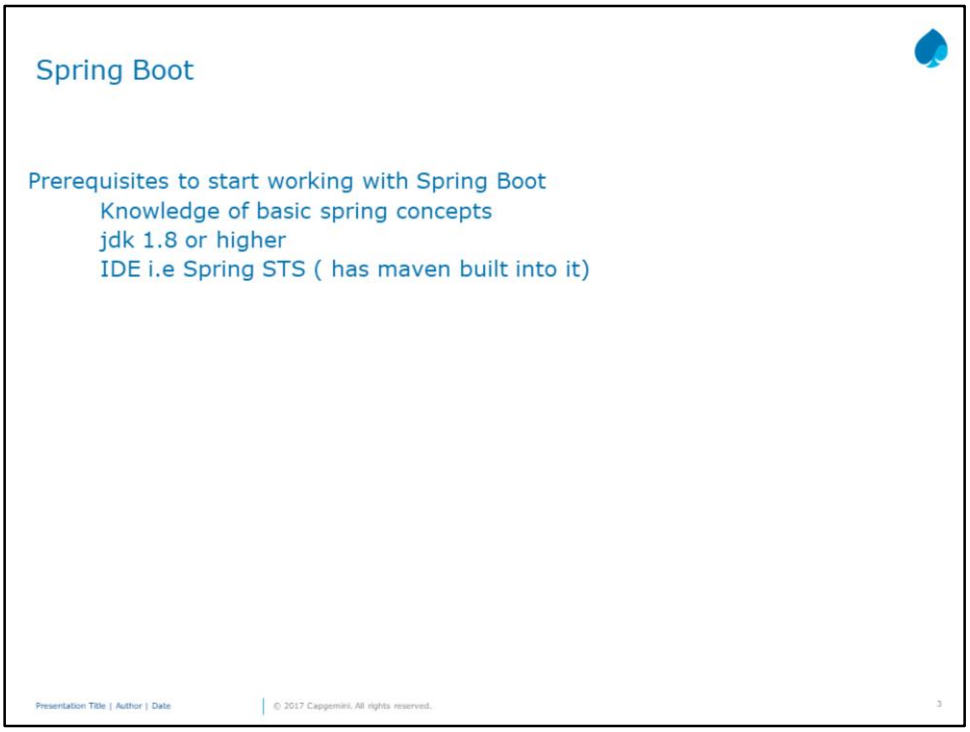
Add instructor notes here.



Following contents would be covered:

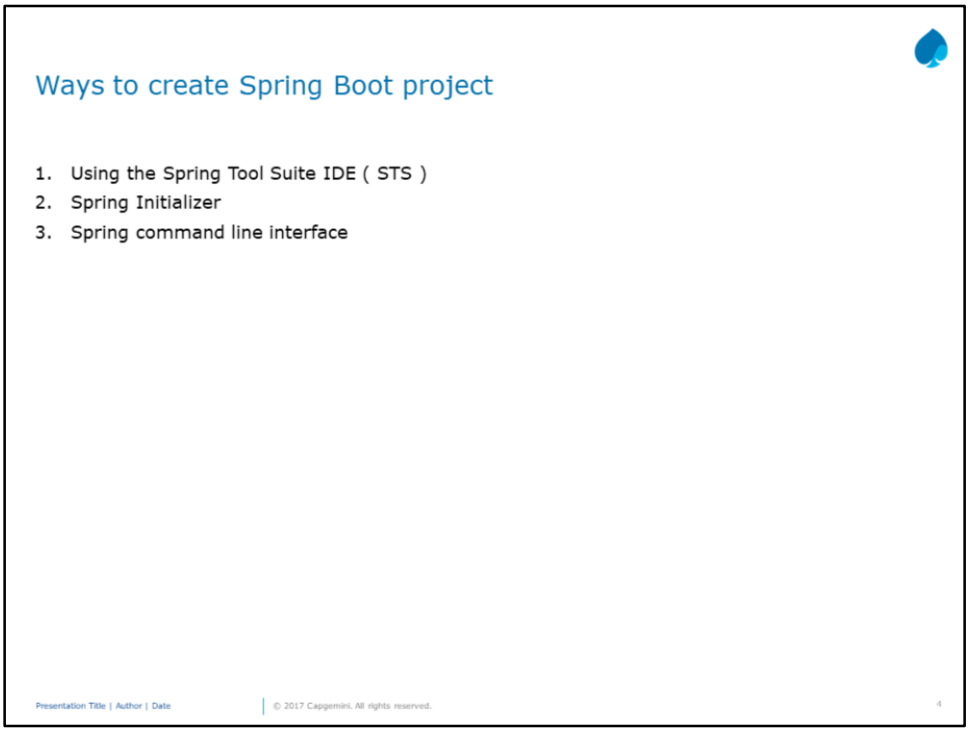
- What is Spring Boot
- How Spring Boot works
- Developing web application using Spring Boot
- Spring Boot integration with Spring Data JPA

Instructor Notes:

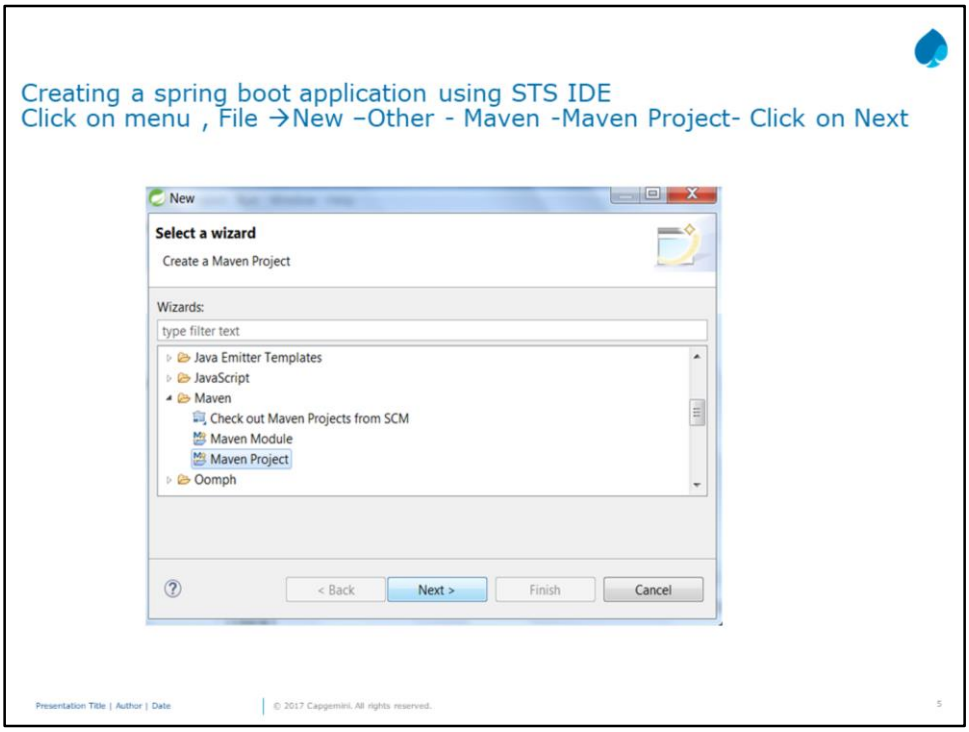


Spring STS can be downloaded from spring.io/tools

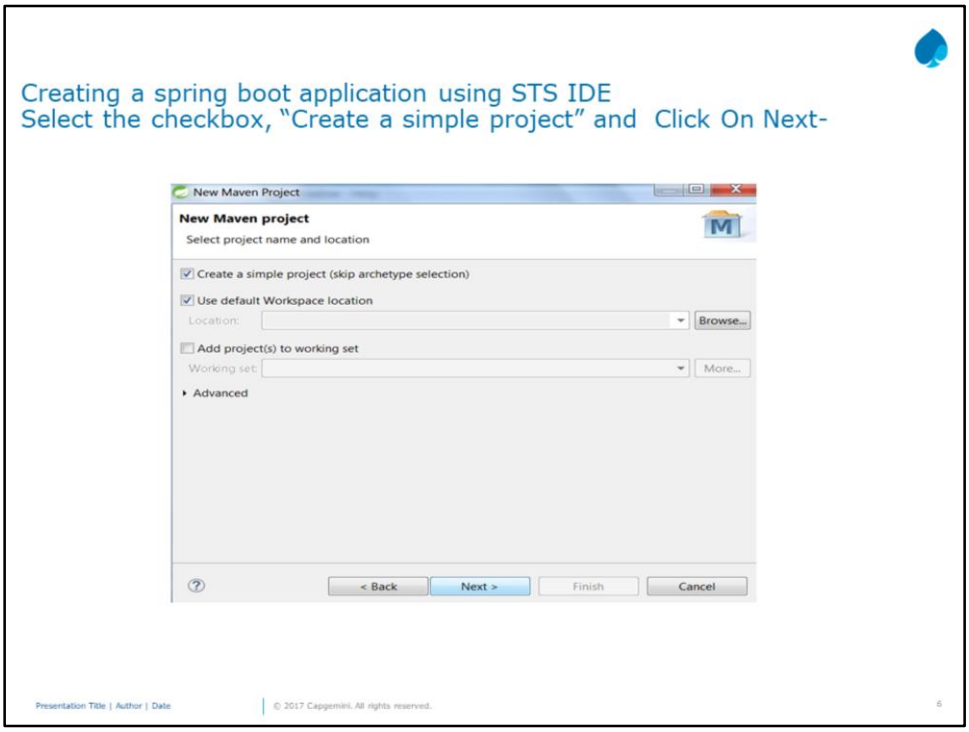
Instructor Notes:



Instructor Notes:

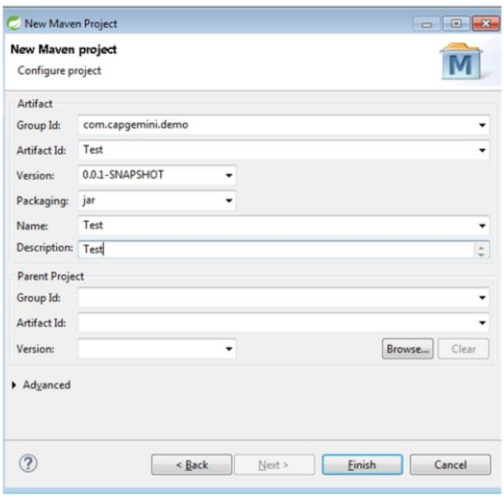


Instructor Notes:

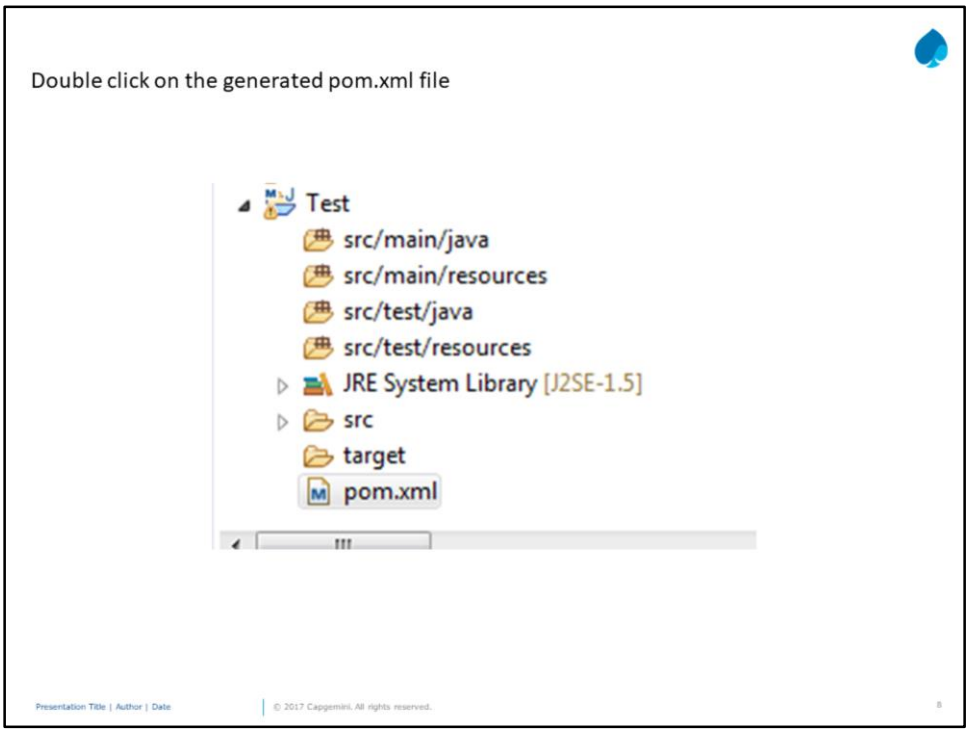


Instructor Notes:

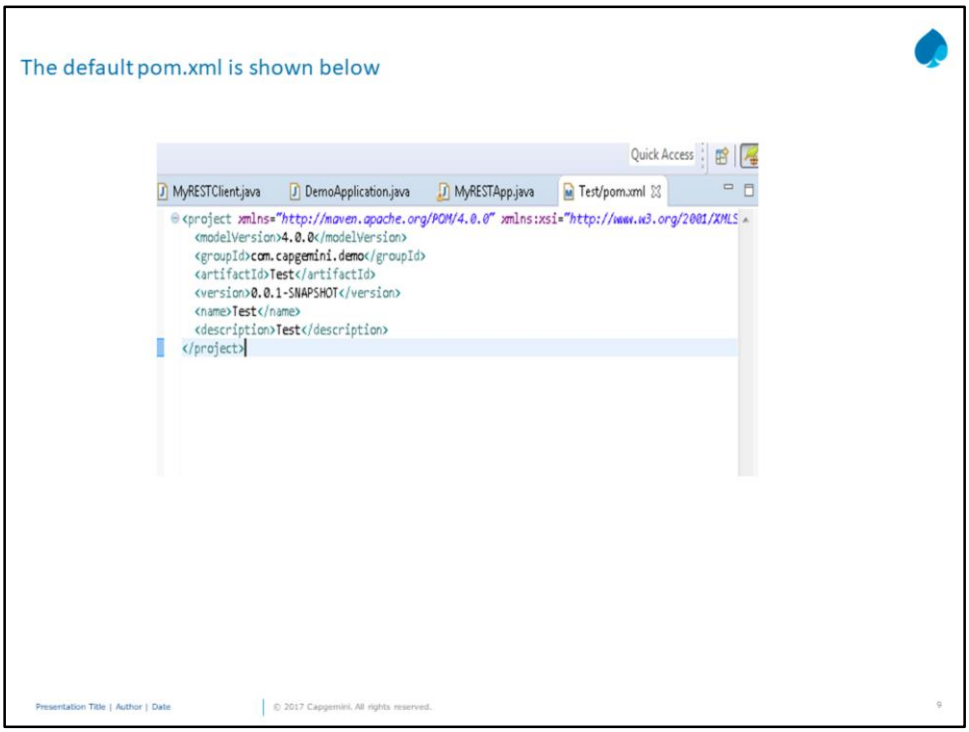
Specify the group id, artifact Id, name and description
Click On Finish. Observe the folder structure of the newly created project



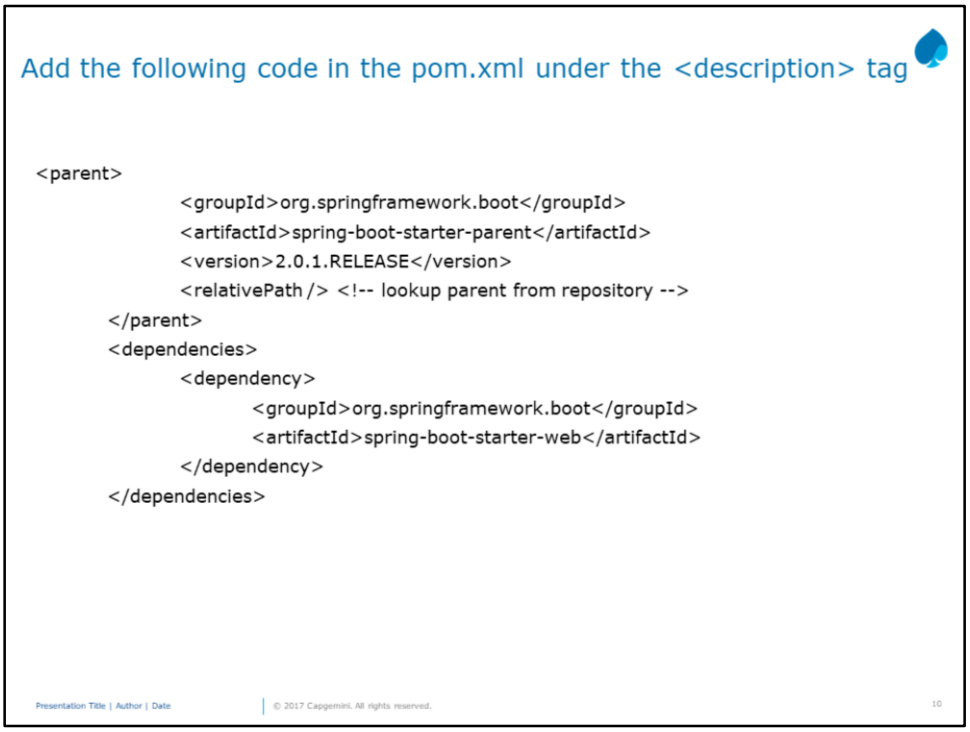
Instructor Notes:



Instructor Notes:



Instructor Notes:



<parent>

<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.0.1.RELEASE</version>
<relativePath /> <!-- lookup parent from repository -->
</parent>

Above entry will bring in all the dependency management features of Spring boot .

There is no need to declare all the dependencies one by one in pom.xml

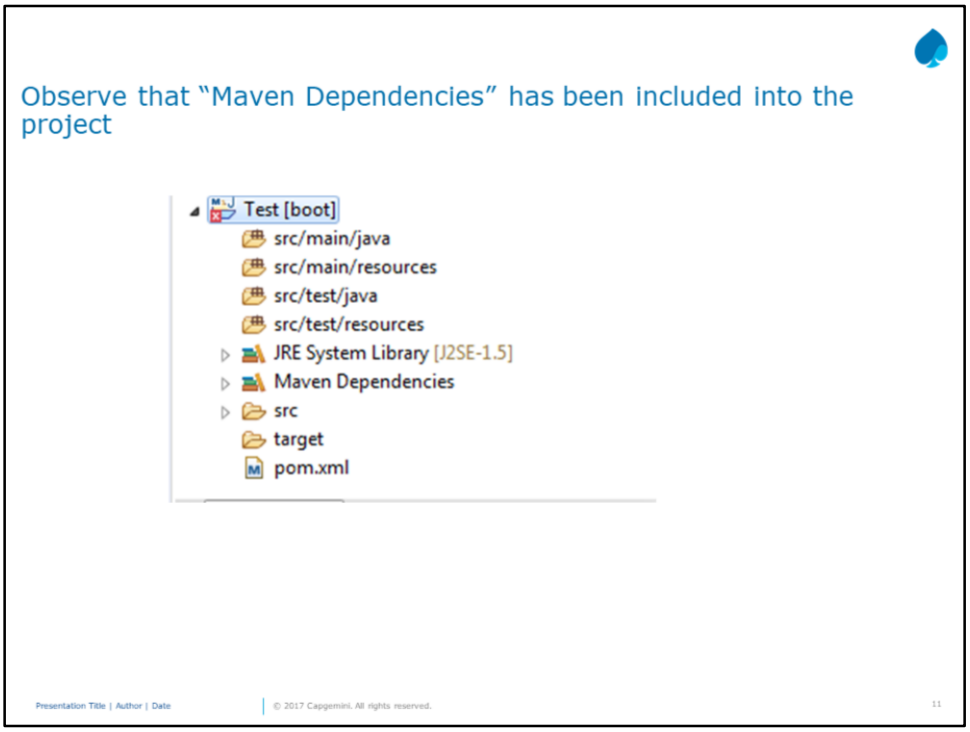
<dependencies>
<dependency>

<groupId>org.springframework.boot</groupId>

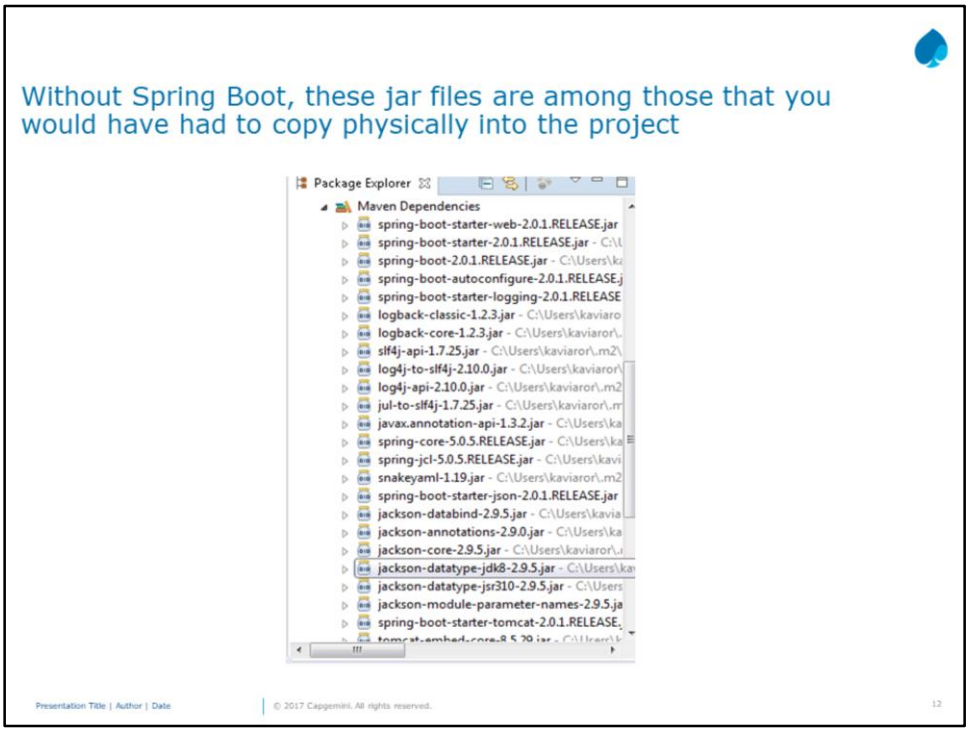
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
</dependencies>

Above will integrate Spring MVC and autoconfigure the project for us.
When u add the Spring boot starter web dependency in pom.xml, this brings in the Spring MVC sub framework dependency into the application.

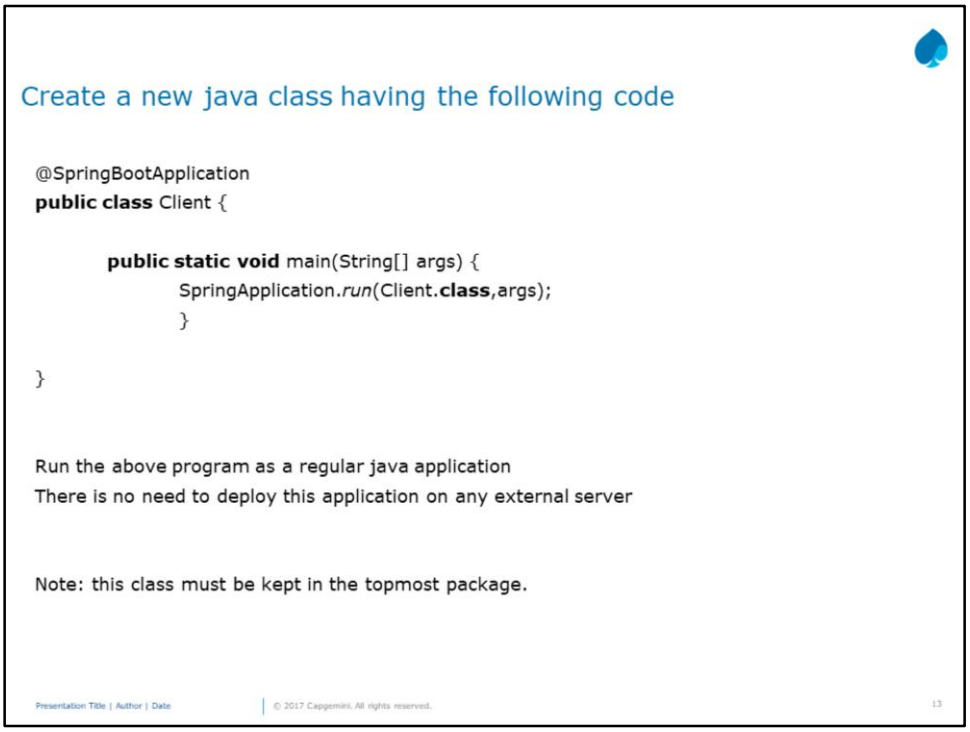
Instructor Notes:



Instructor Notes:



Instructor Notes:



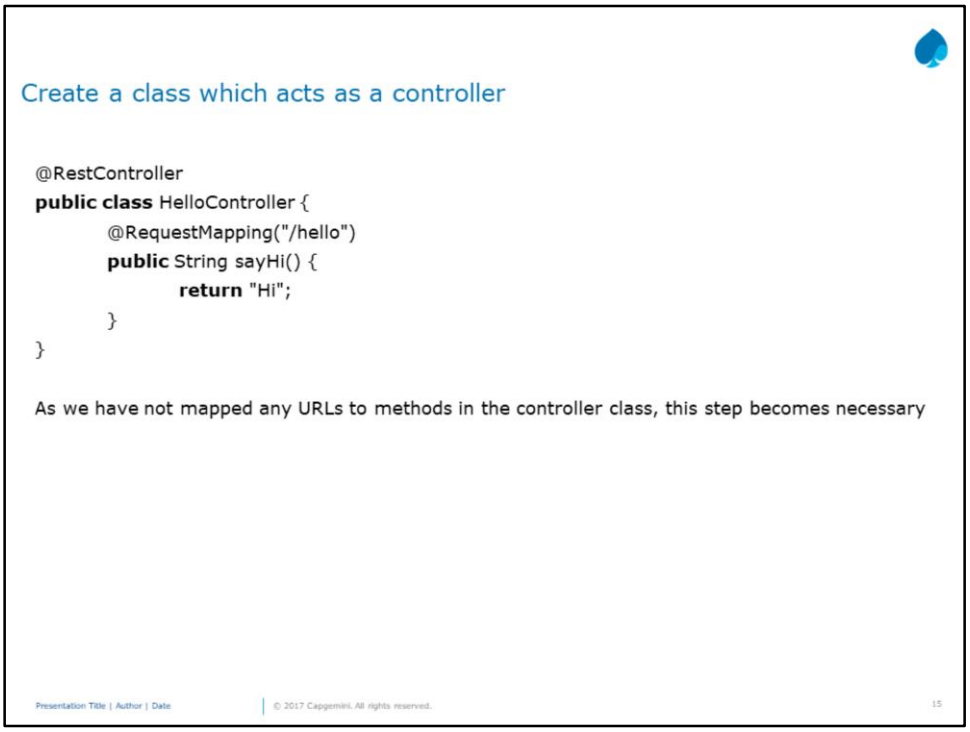
SpringApplication.run(): Starts Spring, creates Spring context , applies annotations and sets up embedded container

Instructor Notes:

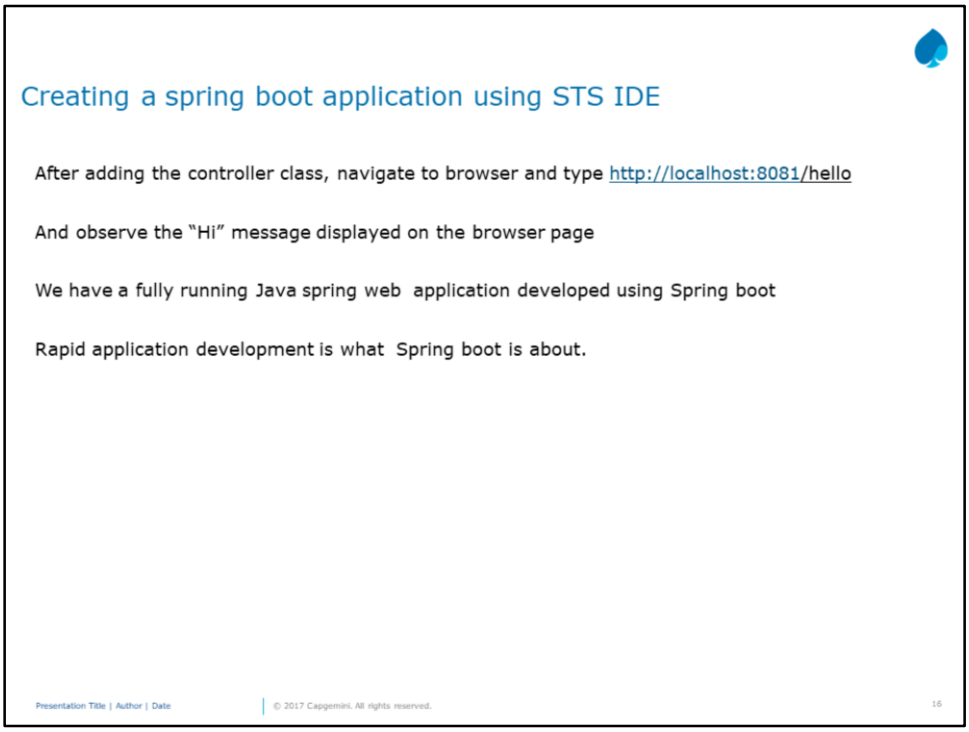
Run the application as a java application and observe the console as shown below

```
.Client                : Starting Client on LINNB267 with PID 11808 (C:\spring_boot\Test\target\classes started by kaviaror in C:\spr
.Client                : No active profile set, falling back to default profiles: default
verApplicationContext : Refreshing org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext@1
omcat.TomcatWebServer  : Tomcat initialized with port(s): 8081 (http)
core.StandardService   : Starting service [Tomcat]
a.core.StandardEngine  : Starting Servlet Engine: Apache Tomcat/8.5.29
AprLifecycleListener   : The APR based Apache Tomcat Native library which allows optimal performance in production environments was n
.[localhost].[/]       : Initializing Spring embedded WebApplicationContext
ntextLoader            : Root WebApplicationContext: initialization completed in 2480 ms
rvletRegistrationBean  : Servlet dispatcherServlet mapped to [/]
lterRegistrationBean   : Mapping filter: 'characterEncodingFilter' to: [/*]
lterRegistrationBean   : Mapping filter: 'hiddenHttpMethodFilter' to: [/*]
lterRegistrationBean   : Mapping filter: 'httpPutFormContentFilter' to: [/*]
lterRegistrationBean   : Mapping filter: 'requestContextFilter' to: [/*]
mpleUrlHandlerMapping  : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource.Resou
tMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServer
tMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.http.ResponseEntity<java.util.Map<java.lang.String, java
tMappingHandlerMapping : Mapped "{[/error],produces=[text/html]}" onto public org.springframework.web.servlet.ModelAndView org.spring
mpleUrlHandlerMapping  : Mapped URL path [/webjars/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceH
mpleUrlHandlerMapping  : Mapped URL path [/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpReque
onMBeanExporter        : Registering beans for JMX exposure on startup
omcat.TomcatWebServer  : Tomcat started on port(s): 8081 (http) with context path ''
.Client                : Started Client in 4.632 seconds (JVM running for 5.691)
```

Instructor Notes:



Instructor Notes:



Use the following to change the default port 8080 on which Tomcat listens

application.properties file
src ->main ->resources .. keep file here
server.port=8081

Instructor Notes:

How Spring Boot works

1. The applicaton is started from the Java main class

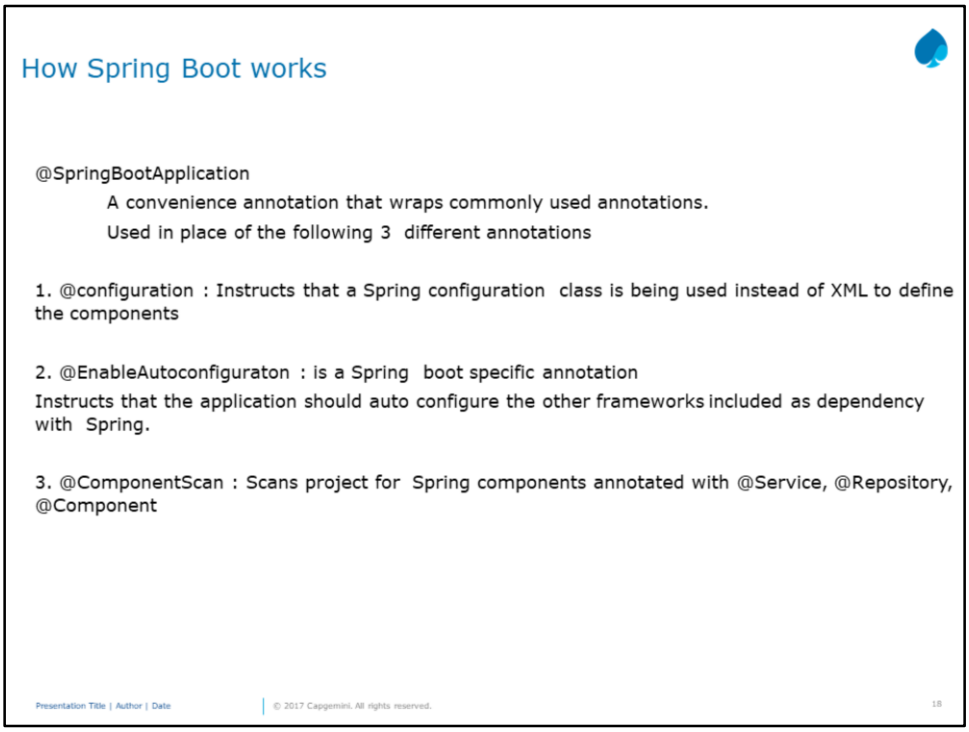
2. Spring boot initialises Spring context that comprises the Spring app and honours autoconfig initialisers, configuration and annotations which direct how to initialise and startup the spring context

3. Embedded server container is started and autoconfigured

This removes the need for web.xml

Spring has chosen "Tomcat" as the default container

Instructor Notes:

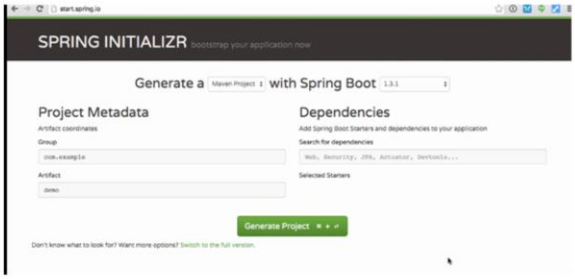


@EnableAutoconfiguration : This annotation told Spring boot to automatically set up so that we can use Spring controllers wihout doing any other integration work with MVC framework

Instructor Notes:

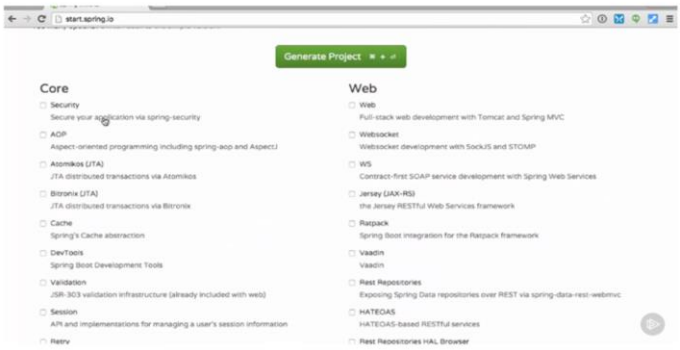
Spring Initializer

Navigate to the following URL
start.spring.io



Click on "switch to full version" link

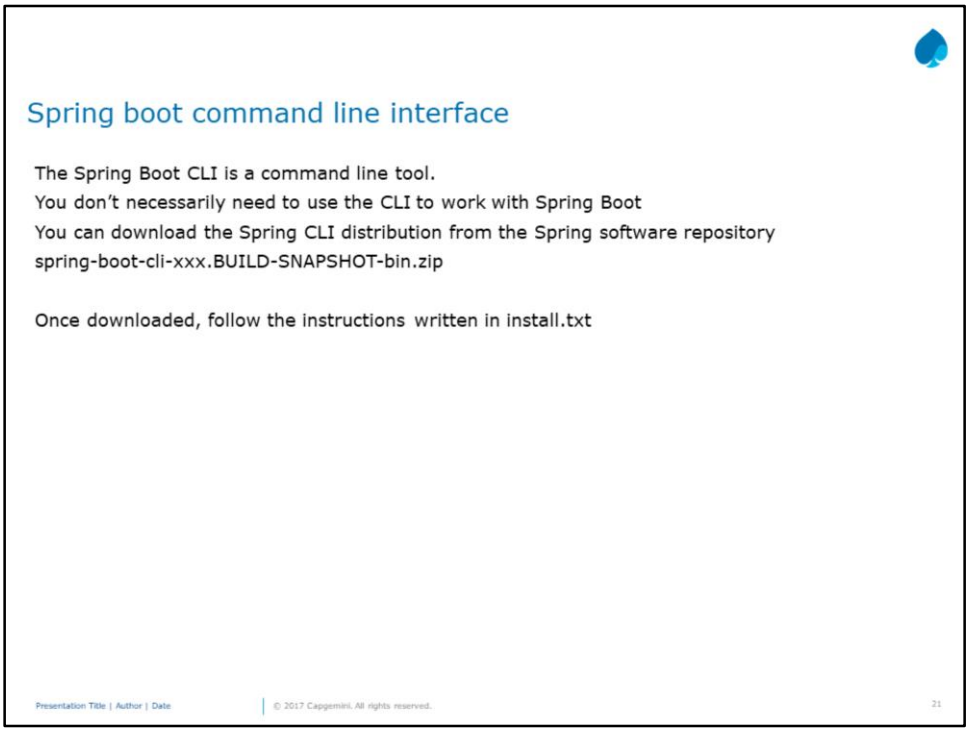
Instructor Notes:



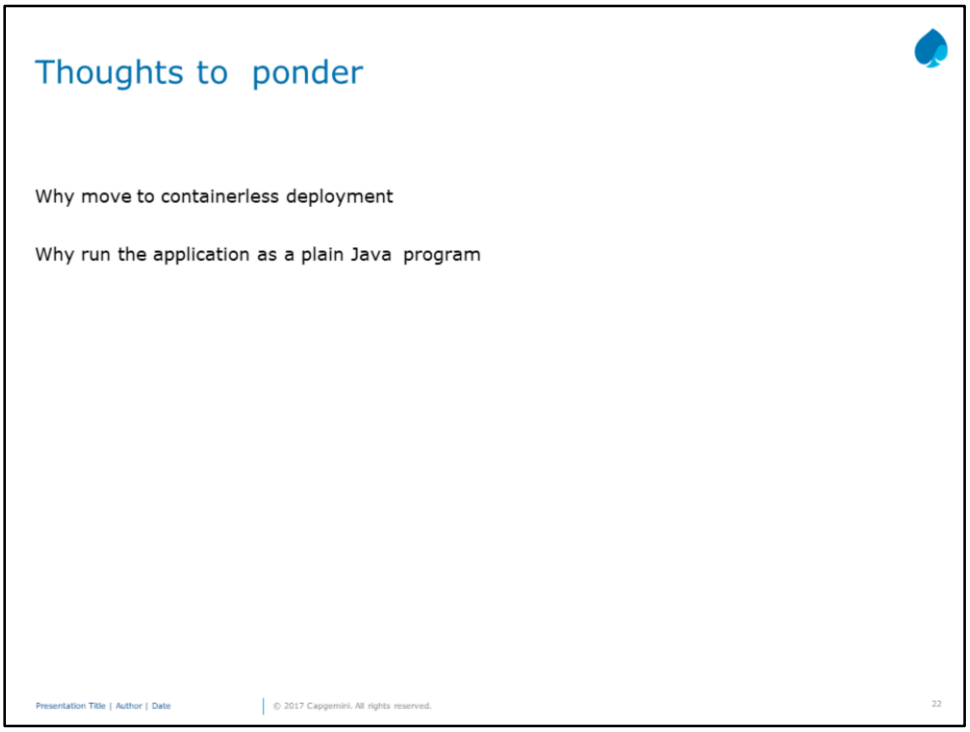
Select appropriate checkboxes which represent the different dependencies you want to include in the project and then click on "Generate Project"

Observe the zip file created for you.
This contains the folder structure of the project

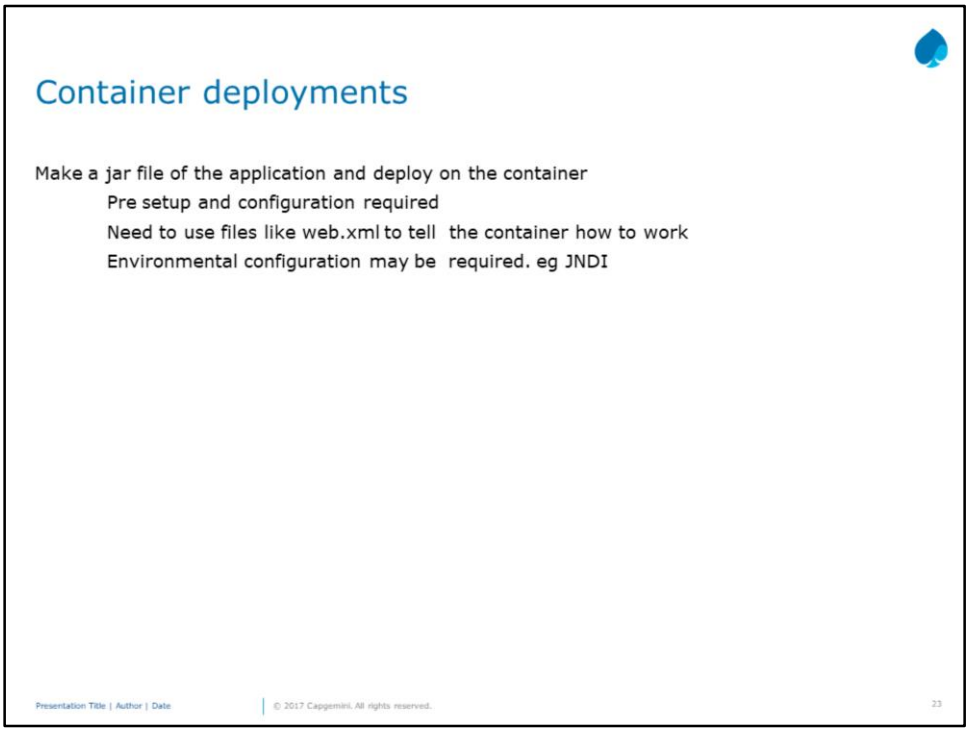
Instructor Notes:



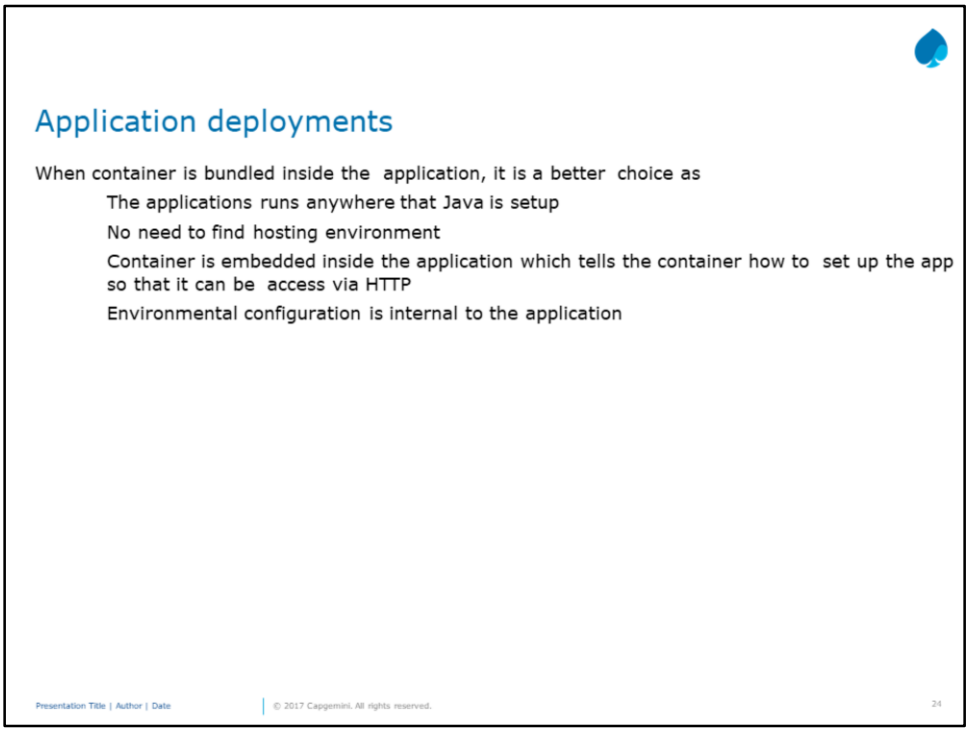
Instructor Notes:



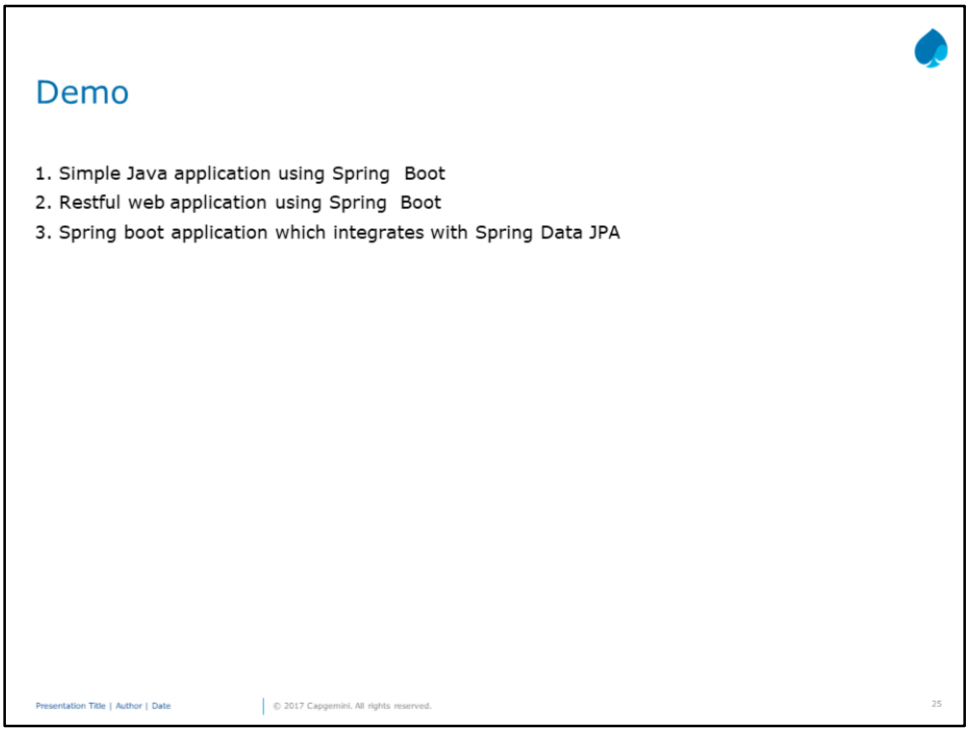
Instructor Notes:



Instructor Notes:

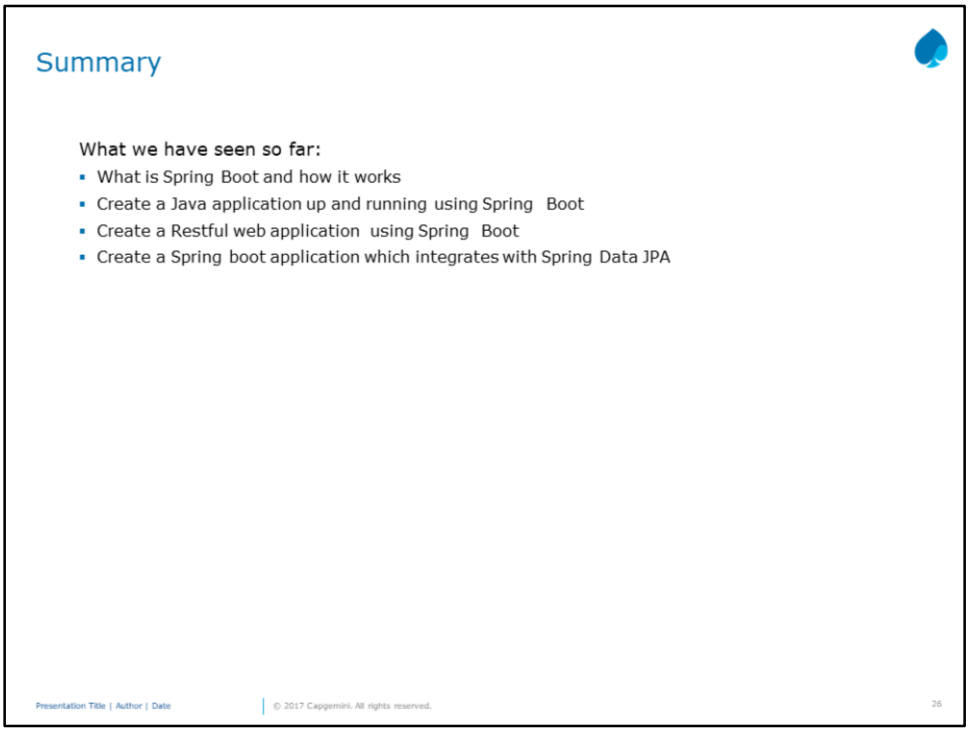


Instructor Notes:



Instructor Notes:

Trainer can summarize the points



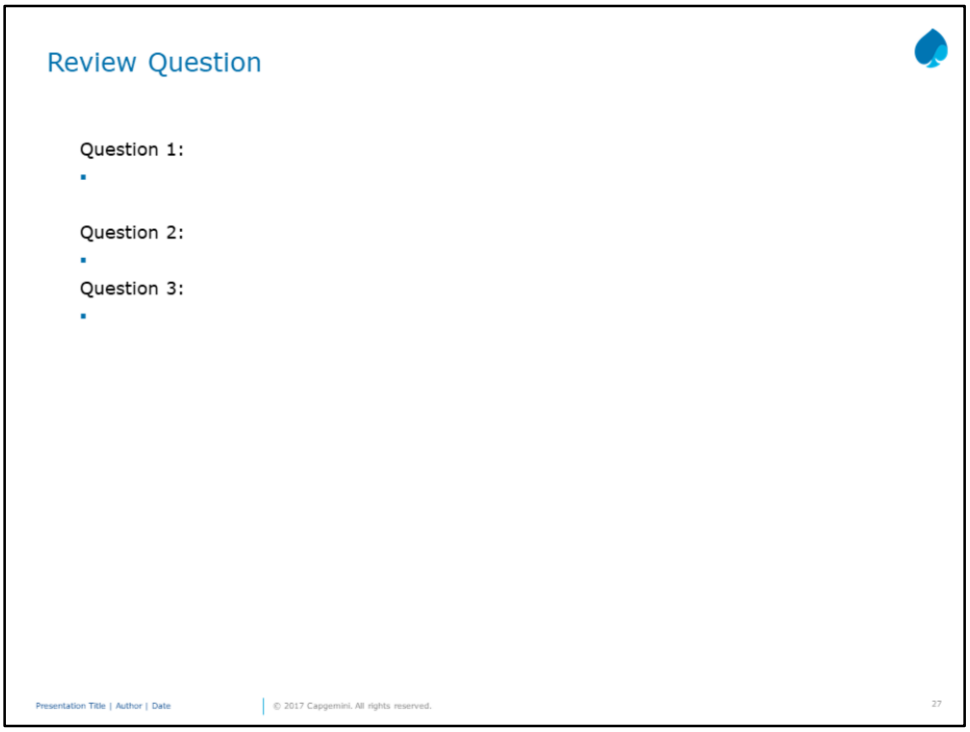
Add the notes here

Instructor Notes:

Question 1: Option 2

Question 2: True

Question 3: SOAP messages



Add the notes here.