

Spring Project

Spring Project using STS

In this article, we are going to create a Spring project by using the STS tool. The STS (Spring Tool Suite) is an official IDE provided by the Spring foundation. We can use it to create spring applications. It is similar to the eclipse, if you are familiar with the eclipse then you will find it very easy to use. Let's start by downloading the STS and then creating the application.

To download the STS, you can visit the official site of the Spring or simply use this [link\(download STS\)](#). After downloading install it to your local system and then run.

Create a Spring Project

To create a Spring project, click on the File menu and select Project that further popup several project types such as Maven Project, Dynamic Project, Java Project, etc. Choose Maven Project because we used a maven-based spring project.

After creating a project now, let's create a simple hello spring application. To create the application, we mainly created three files:

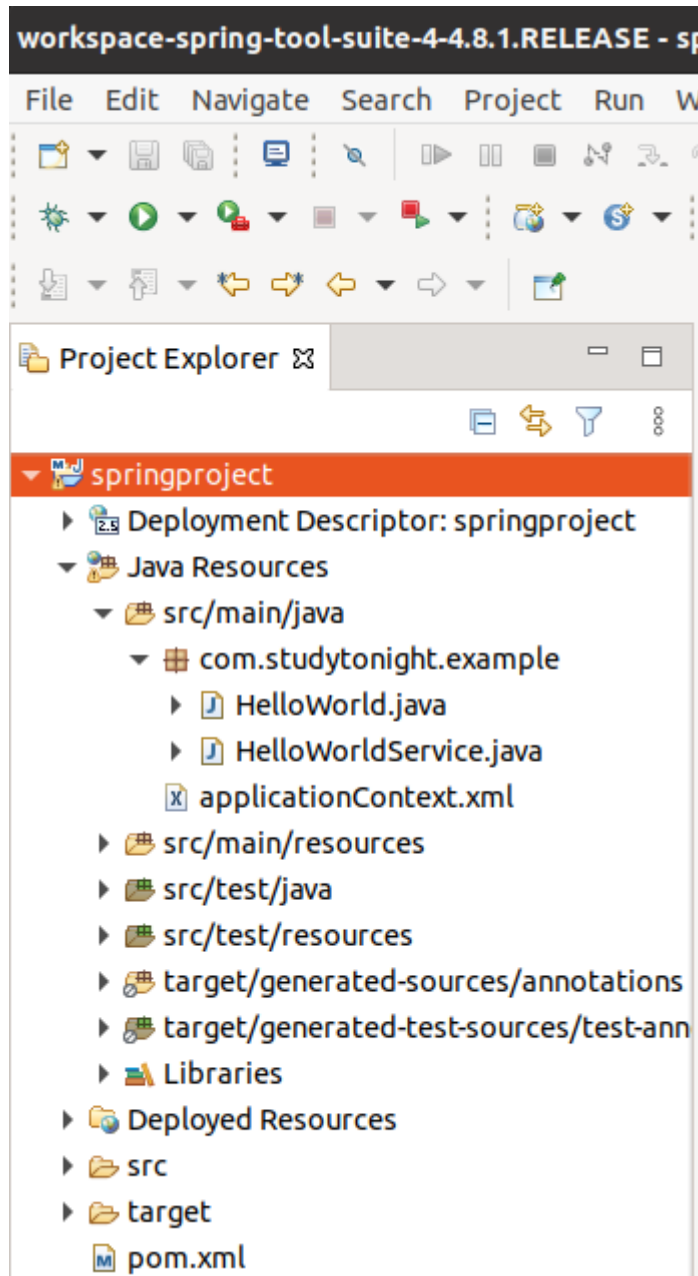
- HelloWorld.java
- HelloWorldService.java
- applicationContext.java

This project has a pom.xml file that contains configuration and project dependencies. You must add Spring Dependencies to this file to configure the application. After that, your project structure should look like below.

Spring Project

Project Structure:

Following will be the project structure of the new project created:



Now let's see the code in each file one by one.

Spring Project

HelloWorld.java

This file contains code for loading the application context file which is configured to load bean. Curious to know, what is Bean? Don't worry, we will cover it later in our tutorial. For now, just create this file.

```
package com.studytonight.example;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class HelloWorld {

    public static void main(String[] args)
    {

        // loading the Bean and XML definitions from the given XML file
        ClassPathXmlApplicationContext context = new
        ClassPathXmlApplicationContext("applicationContext.xml");

        HelloWorldService obj =
        context.getBean(HelloWorldService.class);

        obj.hello();

        context.close();

    }

}
```

Spring Project

HelloWorldService.java

This file contains the code that will print the "Hello Spring" message to the console. Create this file with the "HelloWorldService.java" name.

```
package com.studytonight.example;

public class HelloWorldService
{
    public void hello()
    {
        System.out.println("Hello Spring!");
    }
}
```

Spring Project

applicationContext.xml

This is the configuration file for the Spring project. We can name it anything, but now save it as applicationContext.xml in your maven project.

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:p="http://www.springframework.org/schema/p"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:jee="http://www.springframework.org/schema/jee"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xsi:schemaLocation="http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-3.2.xsd
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.2.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.2.xsd
http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.2.xsd
http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd
http://www.springframework.org/schema/task
http://www.springframework.org/schema/task/spring-task-3.2.xsd">

    <context:component-scan base-package="com.studytonight.examples" />
    <bean id="helloWorldService"
          class="com.studytonight.example.HelloWorldService">
    </bean>

</beans>
```

Spring Project

pom.xml

This file is a part of the maven project and used to add dependencies for our project. For our project, we added spring dependencies. This is the latest by the time of this project. Although we can get these latest dependencies from the maven repository. [Spring Dependencies](#).

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.studytonight</groupId>
  <artifactId>springproject</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
  <dependencies>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-core</artifactId>
      <version>${spring.version}</version>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>${spring.version}</version>
    </dependency>
  </dependencies>
  <properties>
    <spring.version>5.2.8.RELEASE</spring.version>
  </properties>
  <build>
    <sourceDirectory>src</sourceDirectory>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

Spring Project

Run the Application:

Run this application (project) and get the below result.
Since it is not a web application then we can run it simply as a Java application.

Hello Spring!