直接使用 Spring 构建Web项目

# 概述

本文档基于spring 4.3.18.RELEASE 来展示 Spring Web 项目的构建过程，使用的开发工具是 IntelliJ IDEA 14，示例包含的功能有：

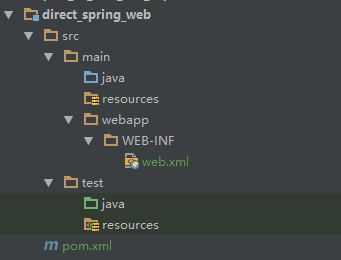
1. RESTFul API
2. 日志框架集成

参考文档：

<https://docs.spring.io/spring/docs/4.3.14.RELEASE/spring-framework-reference/htmlsingle/#Usage%20scenarios>

# 创建项目

创建项目可以很容易，你可以直接创建相应的文件夹就可以了，当然，也可以直接用你熟悉的IDE创建一个Maven项目, 创建后的目录结构如下：



web.xml 的内容如下：

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="  http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0">  </web-app> |

# 添加相关依赖

编辑 pom.xml:

|  |
| --- |
| <properties>  <spring.version>4.3.18.RELEASE</spring.version>  <log4j.version>2.7</log4j.version>  <slf4j.version>1.7.25</slf4j.version> </properties>  <dependencies>  *<!-- Spring MVC 依赖包 开始 -->* <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-webmvc</artifactId>  <exclusions>  <exclusion>  <groupId>commons-logging</groupId>  <artifactId>commons-logging</artifactId>  </exclusion>  </exclusions>  </dependency>  *<!-- Spring MVC 依赖包 结束 -->   <!-- 日志包 开始 -->* <dependency>  <groupId>org.apache.logging.log4j</groupId>  <artifactId>log4j-slf4j-impl</artifactId>  <version>${log4j.version}</version>  </dependency>  <dependency>  <groupId>org.apache.logging.log4j</groupId>  <artifactId>log4j-api</artifactId>  <version>${log4j.version}</version>  </dependency>  <dependency>  <groupId>org.apache.logging.log4j</groupId>  <artifactId>log4j-core</artifactId>  <version>${log4j.version}</version>  </dependency>  <dependency>  <groupId>org.slf4j</groupId>  <artifactId>jcl-over-slf4j</artifactId>  <version>${slf4j.version}</version>  </dependency>  <dependency>  <groupId>org.slf4j</groupId>  <artifactId>jul-to-slf4j</artifactId>  <version>${slf4j.version}</version>  </dependency>  *<!-- 日志包 结束 -->* </dependencies>  <dependencyManagement>  <dependencies>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>spring-framework-bom</artifactId>  <version>${spring.version}</version>  <type>pom</type>  <scope>import</scope>  </dependency>  </dependencies> </dependencyManagement>  <repositories>  <repository>  <id>releases</id>  <url>http://maven.aliyun.com/nexus/content/groups/public/</url>  </repository>   <repository>  <id>snapshots</id>  <url>http://maven.aliyun.com/nexus/content/repositories/snapshots/</url>  </repository> </repositories> |

# 添加日志配置文件

本示例使用的是 log4j2，在resources目录下添加 log4j2.xml 文件：

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <Configuration status="WARN">  <Appenders>  <Console name="Console" target="SYSTEM\_OUT">  <PatternLayout pattern="%d %p [%X,%t] - [%clr{%-40.40c{1.}}] - &lt;%m&gt;%n"/>  </Console>  </Appenders>  <Loggers>  <Logger name="org.springframework.beans.factory" level="DEBUG"/>  <Root level="INFO">  <AppenderRef ref="Console"/>  </Root>  </Loggers> </Configuration> |

# 添加 Spring 监听器和分发Servlet

在 web.xml 中添加 如下代码：

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="  http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0">   *<!-- 非 WEB 容器的spring XML -->* <context-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:applicationContext.xml</param-value>  </context-param>   *<!-- Spring Web Listener -->* <listener>  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  </listener>   <servlet>  <servlet-name>dispatcher</servlet-name>  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  <init-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:applicationContext-web.xml</param-value>  </init-param>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>dispatcher</servlet-name>  <url-pattern>/\*</url-pattern>  </servlet-mapping>  </web-app> |

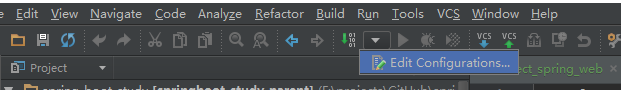
# applicationContext.xml

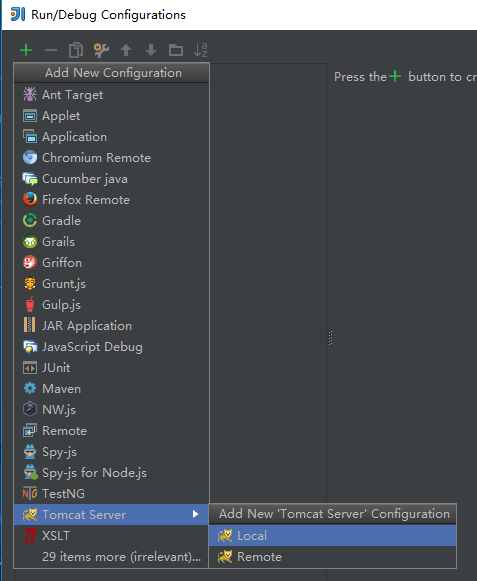
|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:p="http://www.springframework.org/schema/p"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="  http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd">   <context:component-scan base-package="com.github.example.service.impl"/>  </beans> |

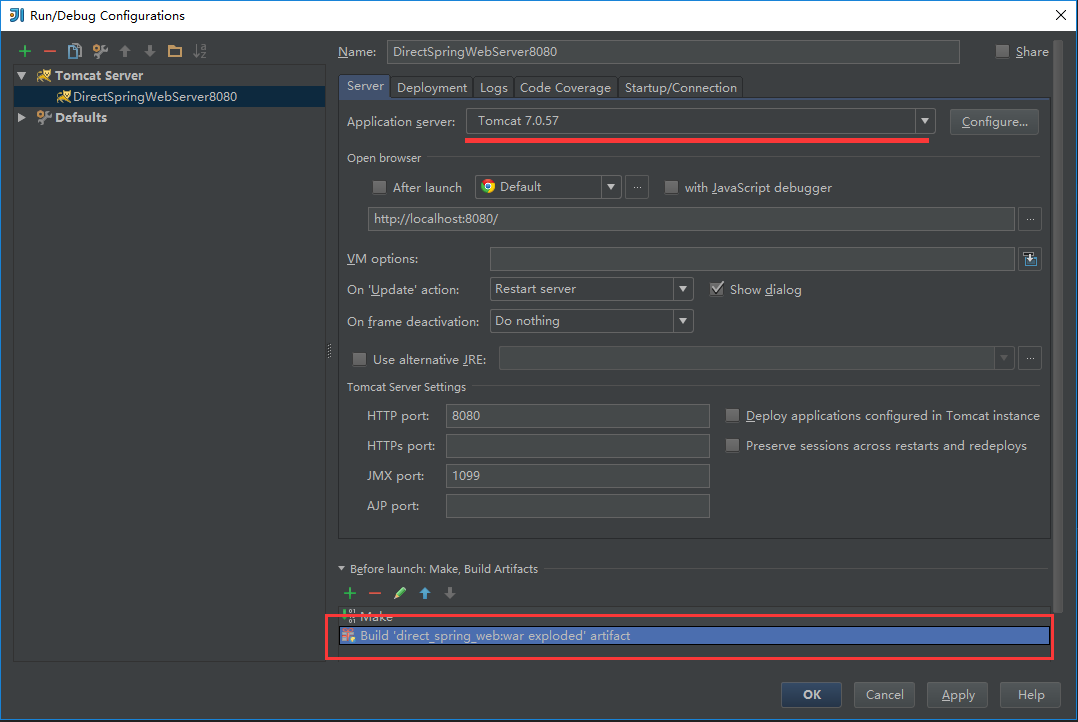
# applicationContext-web.xml

|  |
| --- |
| <?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:context="http://www.springframework.org/schema/context"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xsi:schemaLocation="  http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/context/spring-mvc.xsd">   <context:component-scan base-package="com.github.example.controller"/> </beans> |

# 添加Tomcat启动







# 启动Tomcat服务器

