

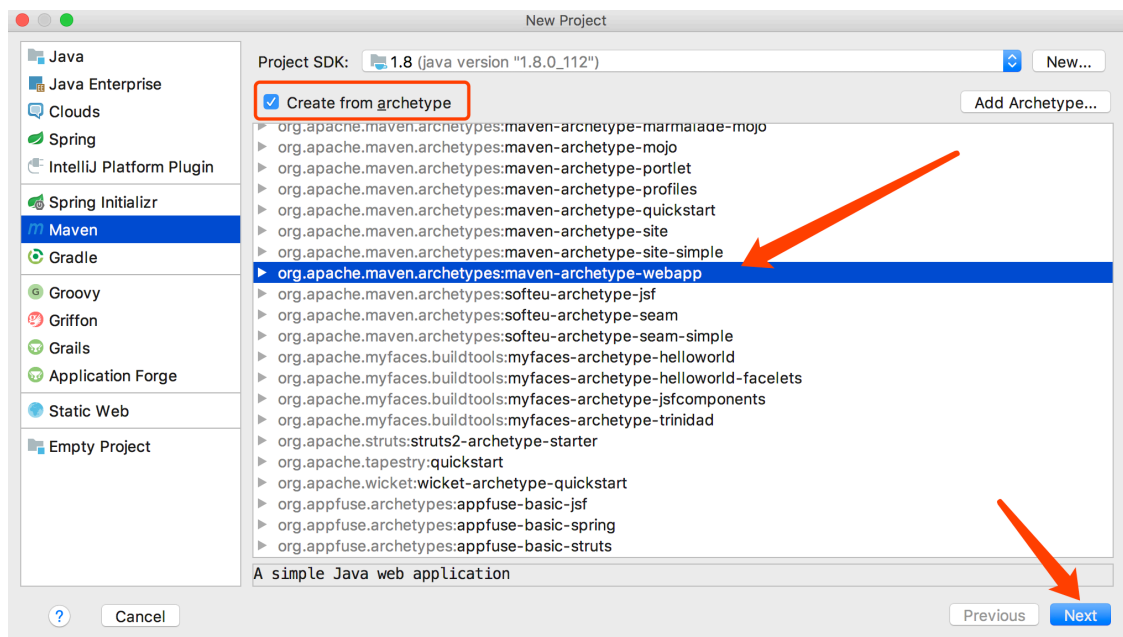
搭建SSM工程

Spring+SpringMVC+Mybatis

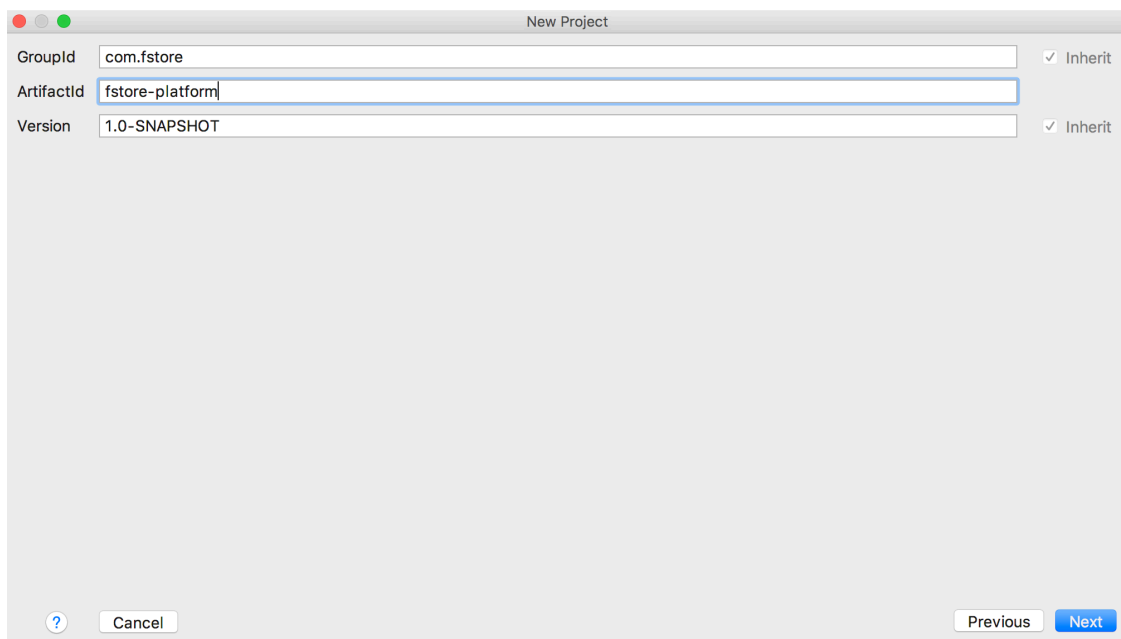
使用IntelliJ IDEA创建spring + maven + SpringMVC + MyBatis项目

1、创建maven项目

【1.1】 File -> New Module，进入创建项目窗口。

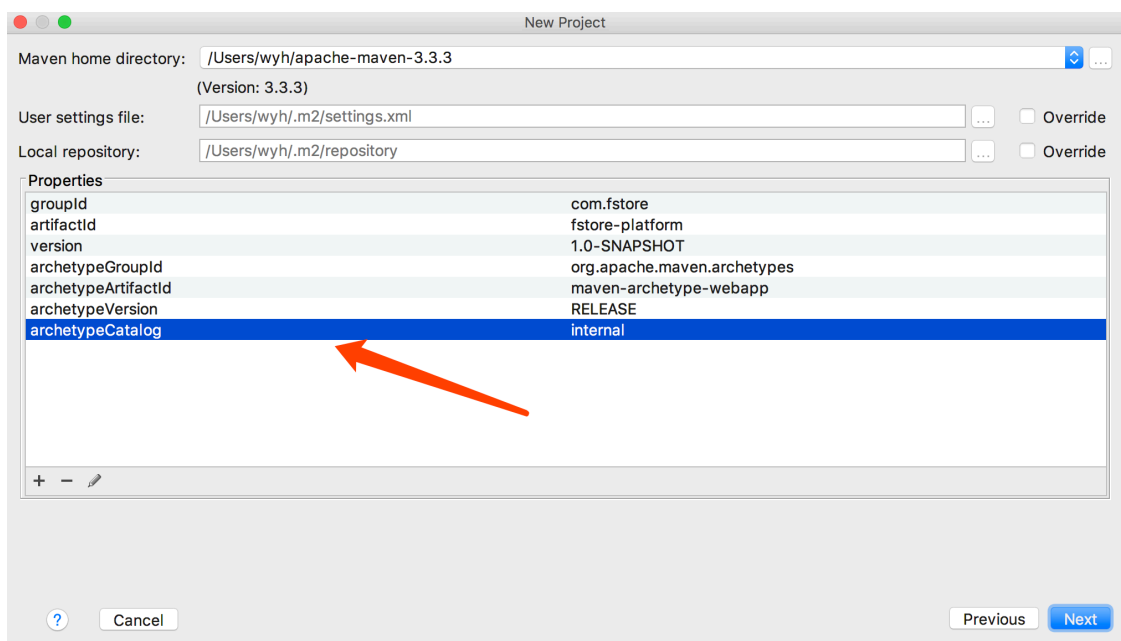


【1.2】 点击Next，填写GroupId、ArtifactId和Version

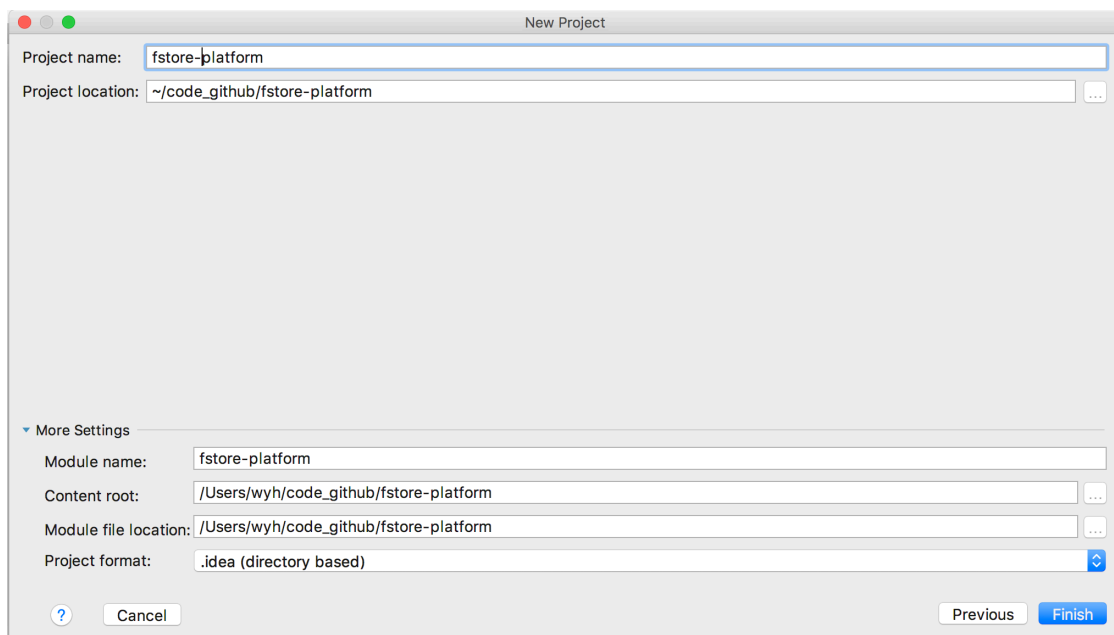


【1.3】 接着下一步，这里需要注在Properties中添加一个参数 `archetypeCatalog=internal`，不加这个参数，在maven生成骨架的时候将会非常慢，有时候直接卡住。来自网上的解释：

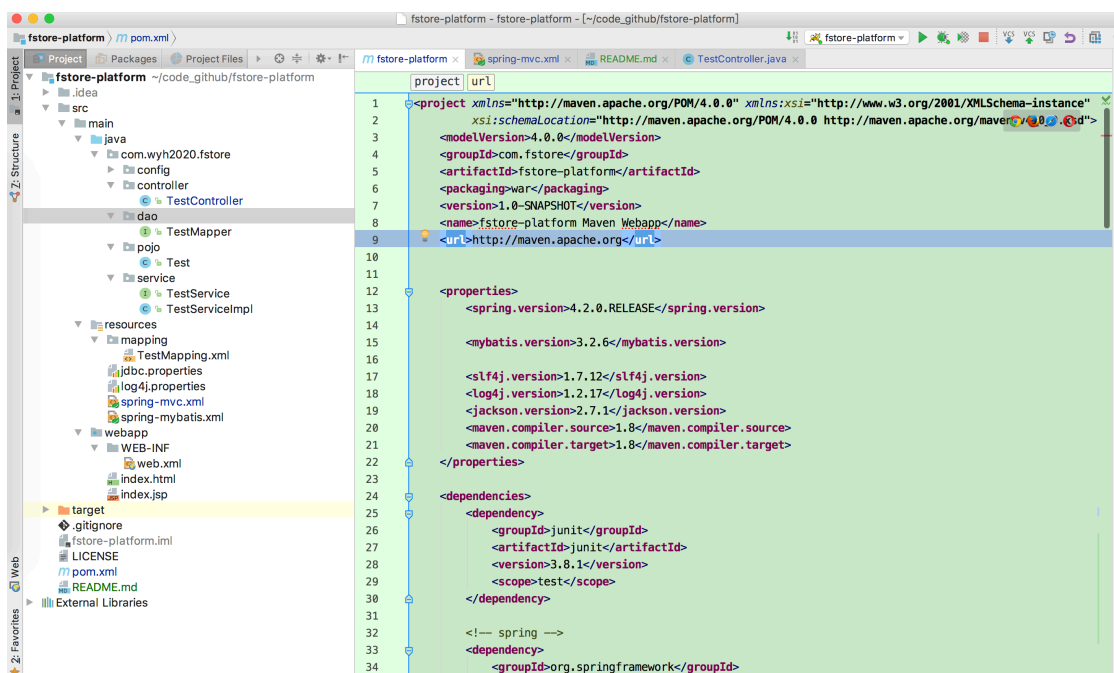
`archetypeCatalog`表示插件使用的archetype元数据，不加这个参数时默认为remote，local，即中央仓库archetype元数据，由于中央仓库的archetype太多了所以导致很慢，指定internal来表示仅使用内部元数据。



【1.4】填写Module name



【1.5】 生成maven的项目骨架之后，我们还需要手动在 src/main 下创建 java目录。现在可以直接编写了，我把项目所需要的文件都编写完成之后，项目的工程结构如图。



到此为止，项目的框架基本搭建完成，下面重点说一下项目的配置文件

2、项目配置文件

【2.1】 pom.xml

这里使用maven来引入项目所需要的jar包，所以也就不需要手动来管理jar包了。

```
<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
http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.fstore</groupId>
    <artifactId>fstore-platform</artifactId>
    <packaging>war</packaging>
    <version>1.0-SNAPSHOT</version>
    <name>fstore-platform Maven Webapp</name>
    <url>http://maven.apache.org</url>

    <properties>
        <spring.version>4.2.0.RELEASE</spring.version>

        <mybatis.version>3.2.6</mybatis.version>

        <slf4j.version>1.7.12</slf4j.version>
        <log4j.version>1.2.17</log4j.version>
        <jackson.version>2.7.1</jackson.version>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
```

```

        <scope>test</scope>
    </dependency>

    <!-- spring -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-core</artifactId>
        <version>${spring.version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-web</artifactId>
        <version>${spring.version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
        <version>${spring.version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-aop</artifactId>
        <version>${spring.version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context-support</artifactId>
        <version>${spring.version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-jdbc</artifactId>
        <version>${spring.version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-test</artifactId>
        <version>${spring.version}</version>
    </dependency>

    <!-- 日志 -->
    <dependency>
        <groupId>log4j</groupId>
        <artifactId>log4j</artifactId>
        <version>${log4j.version}</version>
    </dependency>
    <dependency>
        <groupId>org.slf4j</groupId>

```

```
        <artifactId>slf4j-api</artifactId>
        <version>${slf4j.version}</version>
    </dependency>
</dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>${slf4j.version}</version>
</dependency>

<!-- mybatis -->
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>${mybatis.version}</version>
</dependency>

<!-- mybatis/spring-->
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>1.2.2</version>
</dependency>

<!-- mysql数据库-->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.39</version>
</dependency>

<!-- 导入dbcp的jar包, 用来在applicationContext.xml配置数据库-->
<dependency>
    <groupId>commons-dbcp</groupId>
    <artifactId>commons-dbcp</artifactId>
    <version>1.2.2</version>
</dependency>

<!-- fastjson -->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.17</version>
</dependency>

</dependencies>
<build>
    <finalName>fstore-platform</finalName>
</build>
```

```
</project>
```

【2.2】 jdbc.properties

```
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3307/fstoredb?
useUnicode=true&characterEncoding=utf-8&zeroDateTimeBehavior=convertToNull
jdbc.username=root
jdbc.password=root

#定义初始连接数
jdbc.initialSize=0
#定义最大连接数
jdbc.maxActive=20
#定义最大空闲
jdbc.maxIdle=20
#定义最小空闲
jdbc.minIdle=0
#定义最长等待时间
jdbc.maxWait=60000
```

【2.3】 log4j.properties

```
log4j.rootLogger=DEBUG,Console,File

#控制台日志
log4j.appender.Console=org.apache.log4j.ConsoleAppender
log4j.appender.Console.Target=System.out
log4j.appender.Console.layout=org.apache.log4j.PatternLayout
log4j.appender.Console.layout.ConversionPattern=%-d{yyyy-MM-dd HH:mm:ss}
[ %t:%r ] - [ %p ] %m%n

#普通文件日志
log4j.appender.File=org.apache.log4j.RollingFileAppender
log4j.appender.File.File=logs/fstore.log
log4j.appender.File.MaxFileSize=10M
log4j.appender.File.Threshold=DEBUG
log4j.appender.File.layout=org.apache.log4j.PatternLayout
log4j.appender.File.layout.ConversionPattern=%-d{yyyy-MM-dd HH:mm:ss} [
 %t:%r ] - [ %p ] %m%n
```

【2.4】 spring-mvc.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: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-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/mvc
                           http://www.springframework.org/schema/mvc/spring-
mvc.xsd">

    <context:component-scan base-package="com.wyh2020"/>

    <!-- 启动SpringMVC的注解功能，完成请求和注解POJO的映射 -->
    <mvc:annotation-driven/>

    <!-- 定义跳转的文件的前后缀，视图模式配置 -->
    <bean id="defaultViewResolver"
        class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="viewClass"
            value="org.springframework.web.servlet.view.JstlView"/>
        <property name="prefix" value="/WEB-INF/" />
        <property name="suffix" value=".html"/>
    </bean>

    <import resource="classpath:spring-mybatis.xml"/>

</beans>
```

【2.5】 spring-mybatis.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"
       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">

<!-- 加载配置文件 -->
<bean id="propertyConfigurer"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigu
rer">
    <property name="location" value="classpath:jdbc.properties"/>
</bean>

<!-- 配置数据源 -->
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
destroy-method="close">
    <property name="driverClassName" value="${jdbc.dirver}"/>
    <property name="url" value="${jdbc.url}"/>
    <property name="username" value="${jdbc.username}"/>
    <property name="password" value="${jdbc.password}"/>
    <property name="initialSize" value="${jdbc.initialSize}"/>
    <property name="maxActive" value="${jdbc.maxActive}"/>
    <property name="maxIdle" value="${jdbc.maxIdle}"/>
    <property name="minIdle" value="${jdbc.minIdle}"/>
    <property name="maxWait" value="${jdbc.maxWait}"/>
</bean>

<!-- mybatis和spring完美整合, 不需要mybatis的配置映射文件 -->
<bean id="sqlSessionFactory"
class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource"/>
    <!-- 自动扫描mapping.xml文件 -->
    <property name="mapperLocations" value="classpath:mapping/*.xml"/>
</bean>

<!-- DAO接口所在包名, Spring会自动查找其下的类 -->
<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
    <property name="basePackage" value="com.wyh2020.*.dao"/>
    <property name="sqlSessionFactoryBeanName"
value="sqlSessionFactory"/>
</bean>

<!-- 事务管理 -->
<bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    <property name="dataSource" ref="dataSource"/>
</bean>

</beans>
```

【2.6】 web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0">
  <display-name>Fstore Platform Web Application</display-name>

  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>

  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:spring-mvc.xml</param-value>
  </context-param>

  <!--配置springmvc DispatcherServlet-->
  <servlet>
    <servlet-name>springMVC</servlet-name>
    <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>classpath:spring-mvc.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
    <async-supported>true</async-supported>
  </servlet>
  <servlet-mapping>
    <servlet-name>springMVC</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>

  <!--Log4j监听-->
  <context-param>
    <param-name>log4jConfigLocation</param-name>
    <param-value>classpath:log4j.properties</param-value>
  </context-param>
  <context-param>
    <param-name>log4jRefreshInterval</param-name>
    <param-value>60000</param-value>
  </context-param>
```

```
<listener>
  <listener-
class>org.springframework.web.util.Log4jConfigListener</listener-class>
</listener>

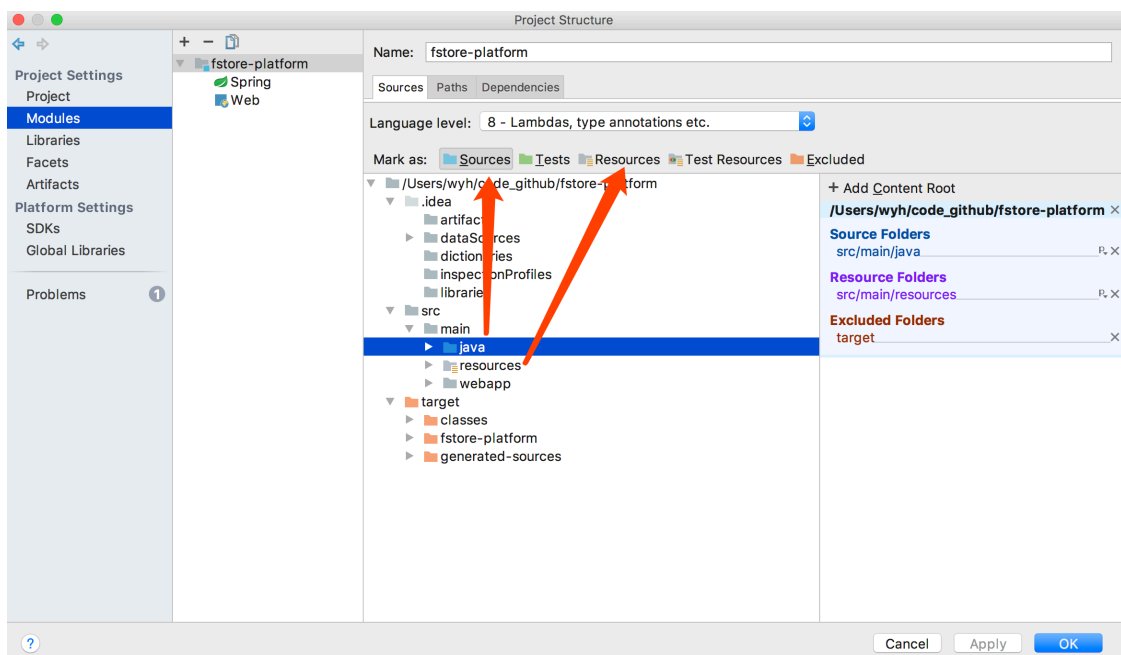
<listener>
  <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-
class>
</listener>

</web-app>
```

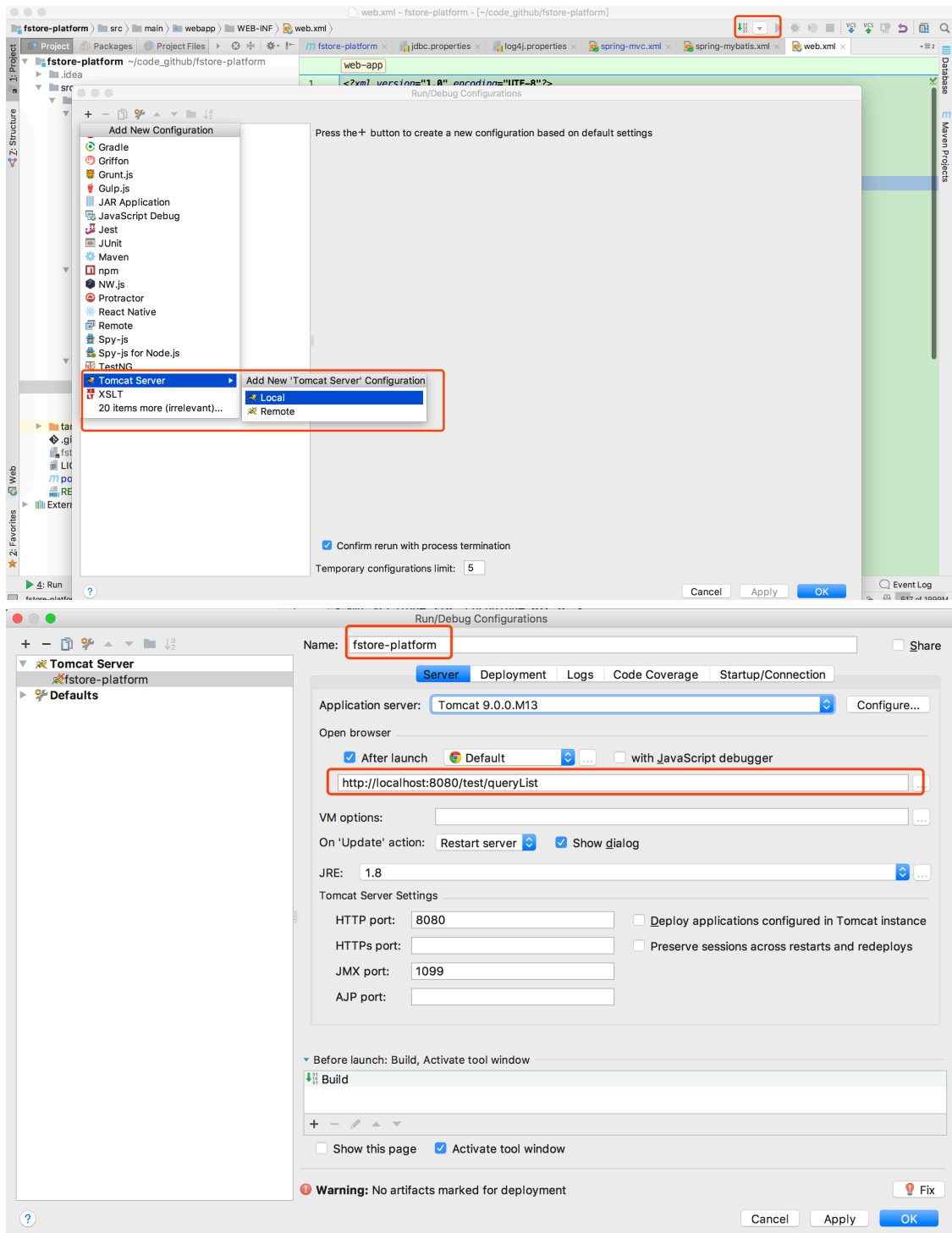
3、项目配置和部署

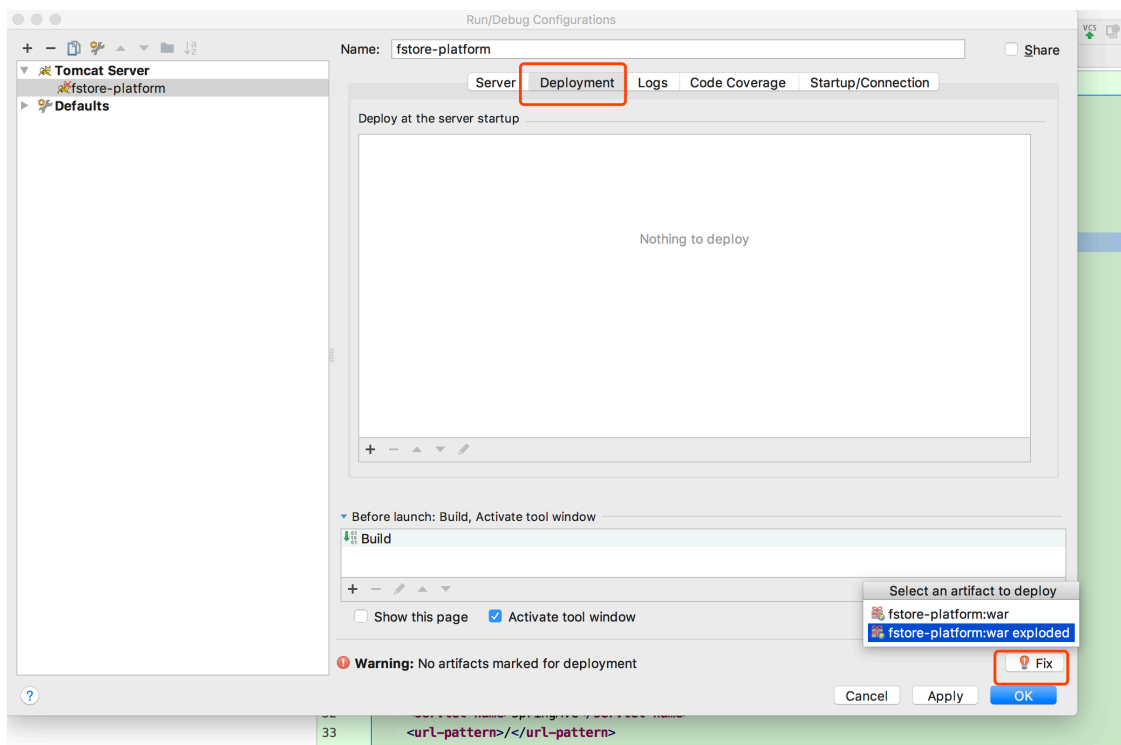
【3.1】 File -> Project Structure, 进入创建项目配置窗口。

【3.2】 创建一个Tomcat容器实例, 并把项目部署进去

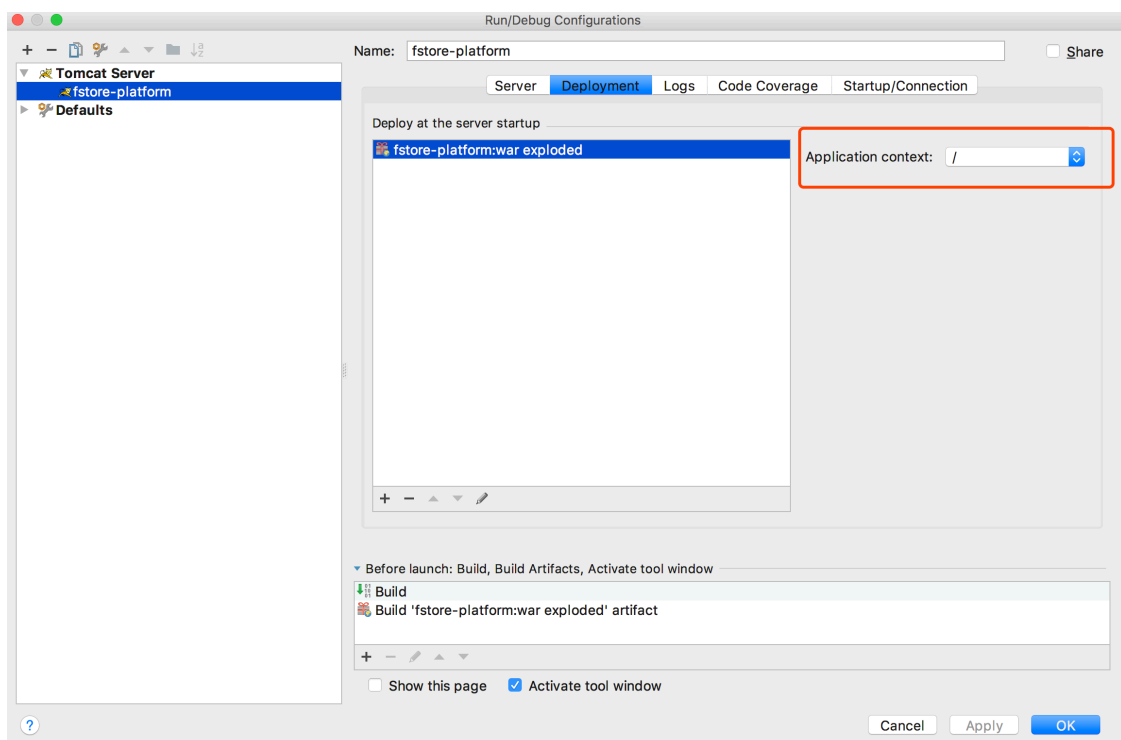


【3.2】创建一个Tomcat容器实例，并把项目部署进去

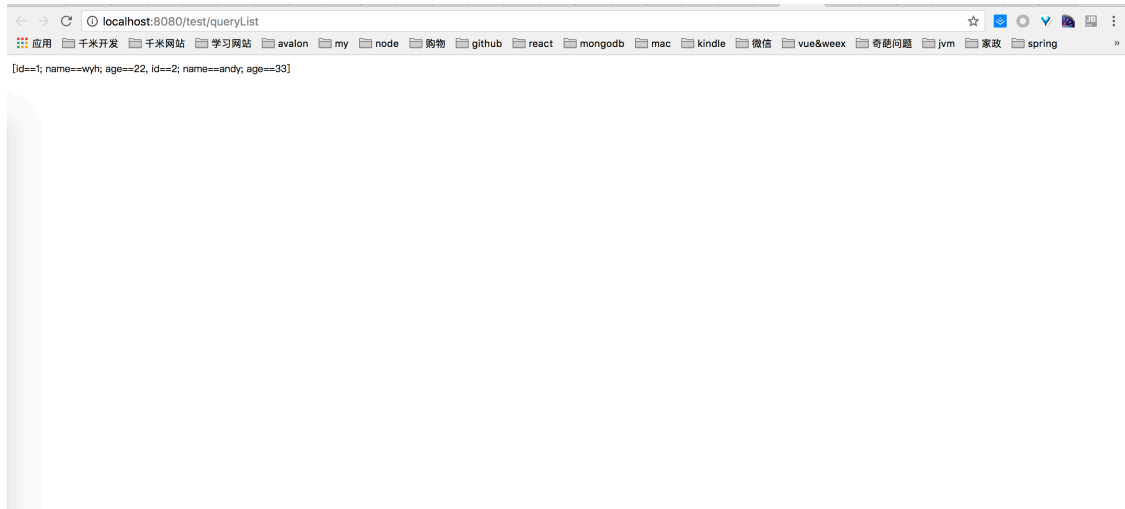




【3.3】项目所需配置好项目访问的根路径，然后启动Tomcat。



【3.4】在浏览器地址栏中输入：<http://localhost:8080/test/queryList>



项目所需看到图中显示效果，则表示项目搭建成功。
