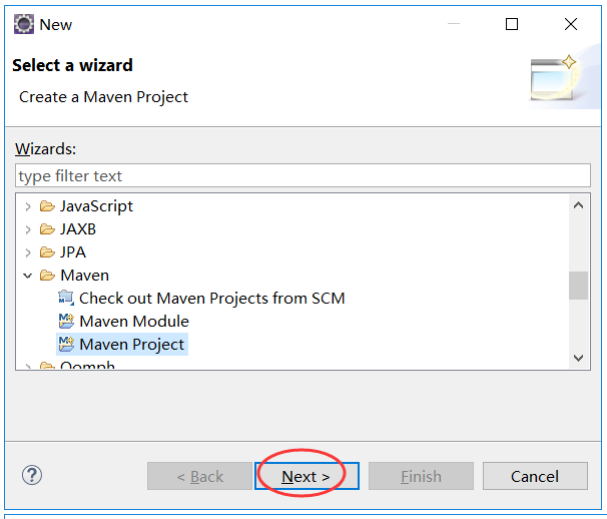
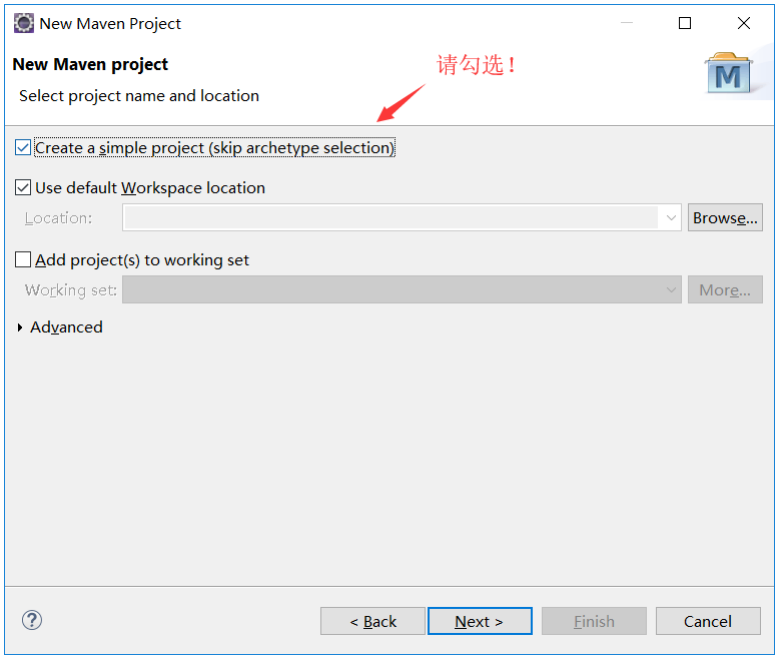
用SpringBoot搭建简单电商项目 01

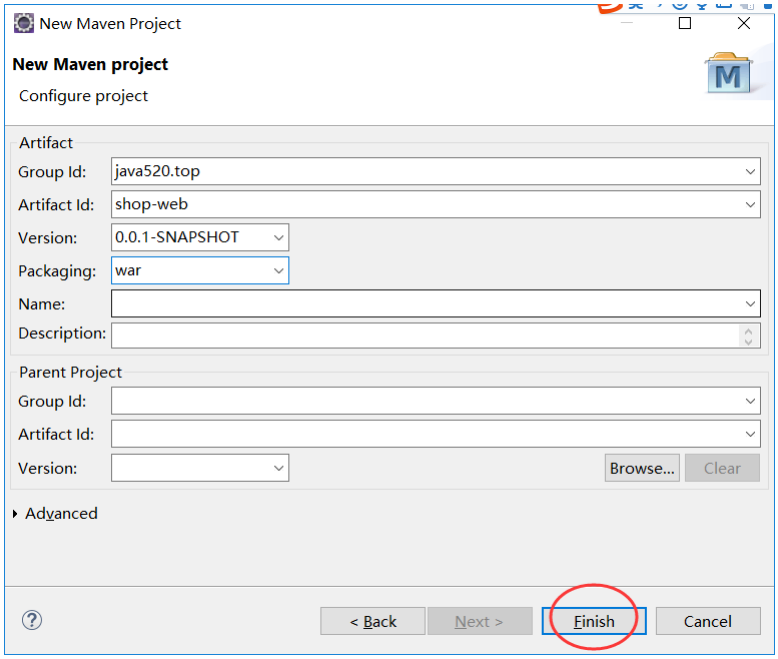
前几节呢，我们已经简单介绍了SpringBoot框架的使用，从这一节开始，我们尝试着使用SpringBoot框架来一步一步搭建一个简单电商项目。当然了，这不是真正的电商项目，你可以看成是一个CRUD案例，只是应用到了SpringBoot框架而已。

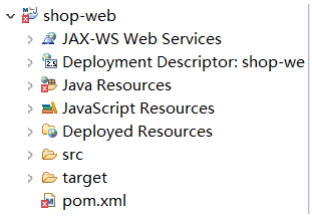
开发工具：eclipse
数据库：MySQL

1.新建项目

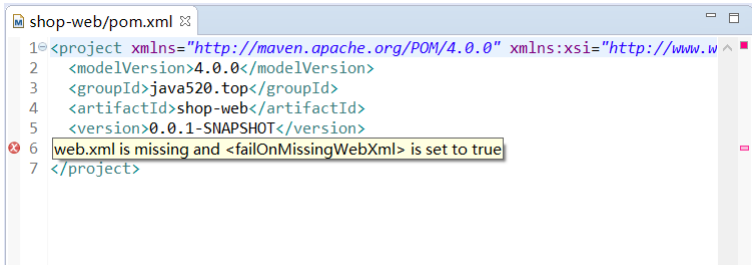




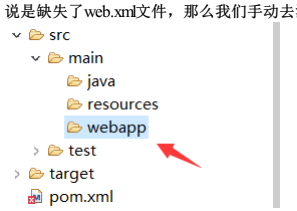




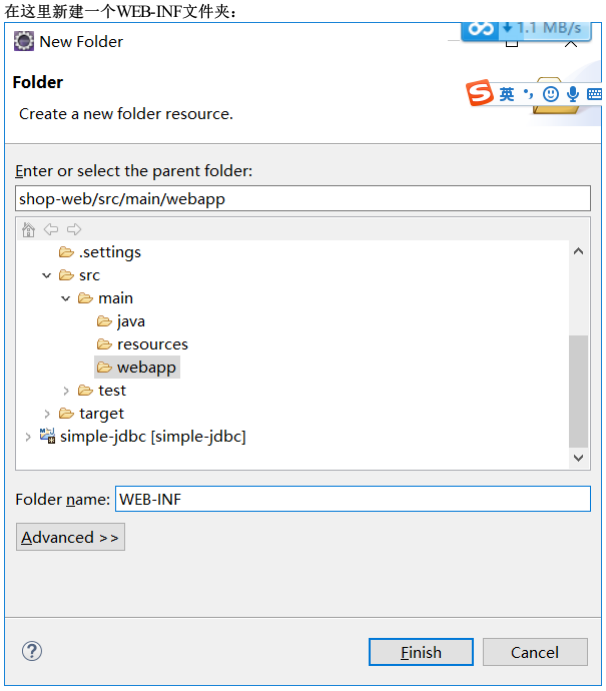
你可以看到选择的pom文件是报错的，让我们打开pom文件一探究竟。



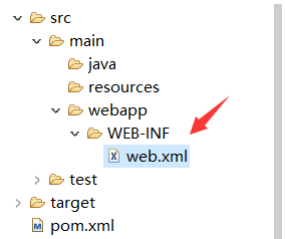
说是缺失了web.xml文件，那么我们手动去添加一下。



在这里新建一个WEB-INF文件夹：



再新建一个web.xml文件：



噢耶，pom文件不报错啦！

web.xml

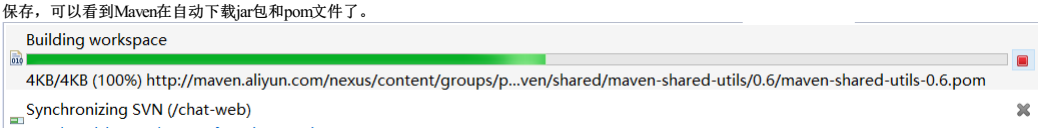
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com
  <display-name>java520.top</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

在pom文件中添加上SpringBoot的父依赖：

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

```
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.10.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
</parent>
```

保存，可以看到Maven在自动下载jar包和pom文件了。



再添加上编码和jdk的配置：

```
<properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <java.version>1.7</java.version>
    </properties>
```

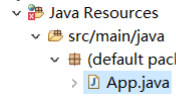
还有jar包依赖：

```
<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>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
        </dependency>


    </dependencies>
```

插件依赖：

```
   <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
```

在java源文件夹中创建一个启动类：



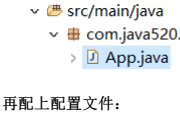
代码：

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class App {

    public static void main(String[] args) {
        SpringApplication.run(App.class, args);
    }
}
```

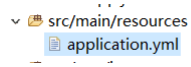
运行，发现报错：

```
rg.springframework.beans.factory.BeanDefinitionStoreException: Failed to parse configuration class [App]; nested exception is java.io.FileNotFoundException: class path resource [
    at org.springframework.context.annotation.ConfigurationClassParser.parse(ConfigurationClassParser.java:181) ~[spring-context-4.3.14.RELEASE.jar:4.3.14.RELEASE]
    at org.springframework.context.annotation.ConfigurationClassPostProcessor.processConfigBeanDefinitions(ConfigurationClassPostProcessor.java:308) ~[spring-context-4.3.14.RELEA
    at org.springframework.context.annotation.ConfigurationClassPostProcessor.postProcessBeanDefinitionRegistry(ConfigurationClassPostProcessor.java:228) ~[spring-context-4.3.14.
    at org.springframework.context.support.PostProcessorRegistrationDelegate.invokeBeanDefinitionRegistryPostProcessors(PostProcessorRegistrationDelegate.java:272) ~[spring-conte
    at org.springframework.context.support.PostProcessorRegistrationDelegate.invokeBeanFactoryPostProcessors(PostProcessorRegistrationDelegate.java:92) ~[spring-context-4.3.14.RE
    at org.springframework.context.support.AbstractApplicationContext.invokeBeanFactoryPostProcessors(AbstractApplicationContext.java:687) ~[spring-context-4.3.14.RELEASE.jar:4.3
    at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:525) ~[spring-context-4.3.14.RELEASE.jar:4.3.14.RELEASE]
    at org.springframework.boot.context.embedded.EmbeddedWebApplicationContext.refresh(EmbeddedWebApplicationContext.java:122) ~[spring-boot-1.5.10.RELEASE.jar:1.5.10.RELEASE]
    at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:693) [spring-boot-1.5.10.RELEASE.jar:1.5.10.RELEASE]
    at org.springframework.boot.SpringApplication.refreshContext(SpringApplication.java:360) [spring-boot-1.5.10.RELEASE.jar:1.5.10.RELEASE]
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:303) [spring-boot-1.5.10.RELEASE.jar:1.5.10.RELEASE]
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:1118) [spring-boot-1.5.10.RELEASE.jar:1.5.10.RELEASE]
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:1107) [spring-boot-1.5.10.RELEASE.jar:1.5.10.RELEASE]
    at App.main(App.java:8) [classes/:na]
Caused by: java.io.FileNotFoundException: class path resource [org/springframework/social/config/annotation/SocialConfigurerAdapter.class] cannot be opened because it does not ex
    at org.springframework.core.io.ClassPathResource.getInputStream(ClassPathResource.java:172) ~[spring-core-4.3.14.RELEASE.jar:4.3.14.RELEASE]
    at org.springframework.core.type.classreading.SimpleMetadataReader.<init>(SimpleMetadataReader.java:50) ~[spring-core-4.3.14.RELEASE.jar:4.3.14.RELEASE]
    at org.springframework.core.type.classreading.SimpleMetadataReaderFactory.getMetadataReader(SimpleMetadataReaderFactory.java:102) ~[spring-core-4.3.14.RELEASE.jar:4.3.14.RELE
    at org.springframework.boot.type.classreading.ConcurrentReferenceCachingMetadataReaderFactory.createMetadataReader(ConcurrentReferenceCachingMetadataReaderFactory.java:89) ~[
    at org.springframework.boot.type.classreading.ConcurrentReferenceCachingMetadataReaderFactory.getMetadataReader(ConcurrentReferenceCachingMetadataReaderFactory.java:76) ~[spr
    at org.springframework.core.type.classreading.SimpleMetadataReaderFactory.getMetadataReader(SimpleMetadataReaderFactory.java:80) ~[spring-core-4.3.14.RELEASE.jar:4.3.14.RELEA
    at org.springframework.context.annotation.ConfigurationClassParser.asSourceClass(ConfigurationClassParser.java:693) ~[spring-context-4.3.14.RELEASE.jar:4.3.14.RELEASE]
    at org.springframework.context.annotation.ConfigurationClassParser$SourceClass.getSuperClass(ConfigurationClassParser.java:857) ~[spring-context-4.3.14.RELEASE.jar:4.3.14.RELE
    at org.springframework.context.annotation.ConfigurationClassParser.doProcessConfigurationClass(ConfigurationClassParser.java:328) ~[spring-context-4.3.14.RELEASE.jar:4.3.14.RE
    at org.springframework.context.annotation.ConfigurationClassParser.processConfigurationClass(ConfigurationClassParser.java:245) ~[spring-context-4.3.14.RELEASE.jar:4.3.14.RELE
    at org.springframework.context.annotation.ConfigurationClassParser.parse(ConfigurationClassParser.java:190) ~[spring-context-4.3.14.RELEASE.jar:4.3.14.RELEASE]
    at org.springframework.context.annotation.ConfigurationClassParser.doProcessConfigurationClass(ConfigurationClassParser.java:292) ~[spring-context-4.3.14.RELEASE.jar:4.3.14.RE
    at org.springframework.context.annotation.ConfigurationClassParser.processConfigurationClass(ConfigurationClassParser.java:245) ~[spring-context-4.3.14.RELEASE.jar:4.3.14.RELE
    at org.springframework.context.annotation.ConfigurationClassParser.parse(ConfigurationClassParser.java:198) ~[spring-context-4.3.14.RELEASE.jar:4.3.14.RELEASE]
    at org.springframework.context.annotation.ConfigurationClassParser.parse(ConfigurationClassParser.java:167) ~[spring-context-4.3.14.RELEASE.jar:4.3.14.RELEASE]
    ... 13 common frames omitted
```

百度了一下，大概意思好像是不能在默认的包里面启动，得有一个具体的package。
ok，那我们就给他一个package。



再配上配置文件：




```
server:
  port: 80
  context-path: /
```

运行后报错：

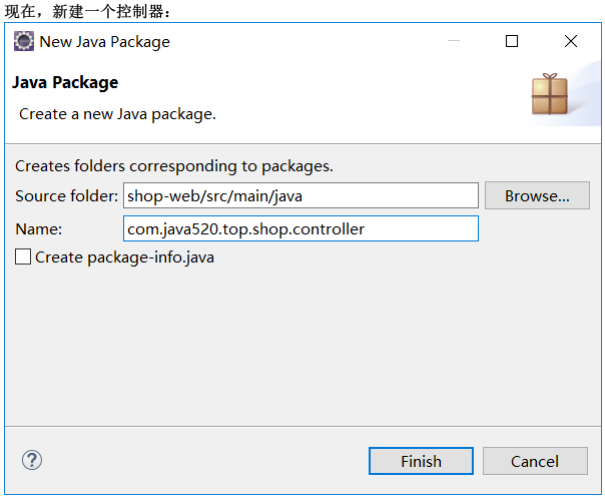
```
***************************
APPLICATION FAILED TO START
***************************

Description:

Cannot determine embedded database driver class for database type NONE

Action:

If you want an embedded database please put a supported one on the classpath. If you have database settings to be loaded from a particular profile you may need to active it (no p
```

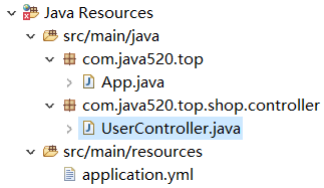
大概的意思呢，就是说springboot自动帮我们注入DataSource了，而我们在yml文件里面还并没有配置数据库的连接信息，所以就报错了。

现在，我们完善一下yml文件：

```
server:
  port: 80
  context-path: /

spring:
  datasource:
    driver-class-name: com.mysql.jdbc.Driver
    url: jdbc:mysql://127.0.0.1:3306/db_shop
    username: root
    password: 123456
  jpa:
    hibernate:
      ddl-auto: update
      show-sql: true
```

数据库采用我本地的MySQL
再次启动，终于不报错了。
现在，新建一个控制器：





测试代码：

```
package com.java520.top.shop.controller;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class UserController {

    @RequestMapping("/getUserById")
    public String getUserById(){
        return "剽悍一小兔";
    }

}
```

启动项目，访问：http://localhost/getUserById
得：

剽悍一小兔

成功了，至此，我们的项目搭建完毕。

源码下载地址：http://java520.top/article/2be4eda6-c749-44ad-8f77-fce641b09a08.html

PS:

更正一个错误，已经是SpringBoot项目了，那么打包方式就不需要打成war包了，因为以前做项目都是这样做的，习惯一下子没改过来。。。