一、启动报如下错误：

|  |
| --- |
| java.lang.NoClassDefFoundError: ch/qos/logback/core/joran/spi/JoranException  at org.slf4j.LoggerFactory.bind(LoggerFactory.java:150)  at org.slf4j.LoggerFactory.performInitialization(LoggerFactory.java:124)  at org.slf4j.LoggerFactory.getILoggerFactory(LoggerFactory.java:412)  at org.slf4j.LoggerFactory.getLogger(LoggerFactory.java:357)  at org.apache.commons.logging.impl.SLF4JLogFactory.getInstance(SLF4JLogFactory.java:155)  at org.apache.commons.logging.impl.SLF4JLogFactory.getInstance(SLF4JLogFactory.java:132)  at org.apache.commons.logging.LogFactory.getLog(LogFactory.java:273)  at org.springframework.boot.SpringApplication.<clinit>(SpringApplication.java:179)  at com.imooc.security.demo.DemoApplication.main(DemoApplication.java:15)  Caused by: java.lang.ClassNotFoundException: ch.qos.logback.core.joran.spi.JoranException  at java.net.URLClassLoader.findClass(URLClassLoader.java:381)  at java.lang.ClassLoader.loadClass(ClassLoader.java:424)  at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:331)  at java.lang.ClassLoader.loadClass(ClassLoader.java:357)  ... 9 more  Exception in thread "main" java.lang.NoClassDefFoundError: ch/qos/logback/core/joran/spi/JoranException  at org.slf4j.LoggerFactory.bind(LoggerFactory.java:150)  at org.slf4j.LoggerFactory.performInitialization(LoggerFactory.java:124)  at org.slf4j.LoggerFactory.getILoggerFactory(LoggerFactory.java:412)  at org.slf4j.LoggerFactory.getLogger(LoggerFactory.java:357)  at org.apache.commons.logging.impl.SLF4JLogFactory.getInstance(SLF4JLogFactory.java:155)  at org.apache.commons.logging.impl.SLF4JLogFactory.getInstance(SLF4JLogFactory.java:132)  at org.apache.commons.logging.LogFactory.getLog(LogFactory.java:273)  at org.springframework.boot.SpringApplication.<clinit>(SpringApplication.java:179)  at com.imooc.security.demo.DemoApplication.main(DemoApplication.java:15)  Caused by: java.lang.ClassNotFoundException: ch.qos.logback.core.joran.spi.JoranException  at java.net.URLClassLoader.findClass(URLClassLoader.java:381)  at java.lang.ClassLoader.loadClass(ClassLoader.java:424)  at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:331)  at java.lang.ClassLoader.loadClass(ClassLoader.java:357)  ... 9 more |

原因分析：缺少了**logback.jar**

**解决方案：添加如下依赖**

|  |
| --- |
| **<dependency>**  **<groupId>ch.qos.logback</groupId>**  **<artifactId>logback-classic</artifactId>**  **<version>1.2.3</version>**  **</dependency>**  **<dependency>**  **<groupId>ch.qos.logback</groupId>**  **<artifactId>logback-classic</artifactId>**  **<version>1.2.3</version>**  **<exclusions>**  **<exclusion>**  **<groupId>org.slf4j</groupId>**  **<artifactId>slf4j-log4j12</artifactId>**  **</exclusion>**  **</exclusions>**  **</dependency>**  **<dependency>**  **<groupId>ch.qos.logback</groupId>**  **<artifactId>logback-core</artifactId>**  **<version>1.2.3</version>**  **</dependency>** |

# **二、启动报如下错误**

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  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 profiles are currently active). |

原因分析:maven中添加了如下依赖，但却没有配置数据源信息

|  |
| --- |
| <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-jdbc</**artifactId**> </**dependency**> <**dependency**>  <**groupId**>mysql</**groupId**>  <**artifactId**>mysql-connector-java</**artifactId**> </**dependency**> |

解决方案：

在application.properties文件中增加如下的配置信息：

|  |
| --- |
| *## 数据库连接配置* **spring.datasource.driver-class-name**: **com.mysql.jdbc.Driver spring.datasource.url**: **jdbc:mysql://127.0.0.1:3306/zhangshenglu?useUnicode=true&characterEncoding=UTF-8&allowMultiQueries=true spring.datasource.username**: **root spring.datasource.password**: **123** |

三、spring Session忘记配置，报错信息如下：

|  |  |
| --- | --- |
| org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'org.springframework.boot.autoconfigure.session.SessionAutoConfiguration$SessionRepositoryValidator': Invocation of init method failed; nested exception is java.lang.IllegalArgumentException: No Spring Session store is configured: set the 'spring.session.store-type' property |  |

原因分析：未配置spring session的存储,Spring session是用来做集群session管理的。如果想暂时不用，可以在application.properties属性文件中配置如下：

**spring.session.store-type**=*none*

四、对其中的demo项目进行打包，添加以下的打包插件

|  |
| --- |
| <**build**>  <**plugins**>  <**plugin**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-maven-plugin</**artifactId**>   <**executions**>  <**execution**>  <**goals**>  <**goal**>repackage</**goal**>  </**goals**>  </**execution**>  </**executions**>  </**plugin**>  </**plugins**>  <**finalName**>demo</**finalName**> </**build**> |

然后在打好的demo.jar文件夹中执行：

java –jar demo.jar

五、在pom文件中引入Spring-test测试框架

|  |
| --- |
| <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-test</**artifactId**>  <**scope**>test</**scope**>  </**dependency**> |

编写一个测试类：

|  |
| --- |
| **package** com.imooc.security.demo;  **import** com.imooc.security.demo.DemoApplication; **import** org.junit.Before; **import** org.junit.Test; **import** org.junit.runner.RunWith; **import** org.springframework.beans.factory.annotation.Autowired; **import** org.springframework.boot.test.context.SpringBootTest; **import** org.springframework.http.MediaType; **import** org.springframework.test.context.junit4.SpringRunner; **import** org.springframework.test.web.servlet.MockMvc; **import** org.springframework.test.web.servlet.request.MockMvcRequestBuilders;  **import** org.springframework.test.web.servlet.result.MockMvcResultMatchers; **import** org.springframework.test.web.servlet.setup.MockMvcBuilders; **import** org.springframework.web.context.WebApplicationContext; */\*\*  \* zhshl 2018/9/9  \*/* @RunWith(SpringRunner.**class**) @SpringBootTest **public class** UserControllerTest {  @Autowired  **private** WebApplicationContext **wac**;  **private** MockMvc **mockMvc**;  @Before  **public void** setUp(){  **mockMvc** = MockMvcBuilders.*webAppContextSetup*(**wac**).build();  }  @Test  **public void** whenQuerySuccess() **throws** Exception {  **mockMvc**.perform(MockMvcRequestBuilders.*get*(**"/user"**).  contentType(MediaType.***APPLICATION\_JSON\_UTF8***))  .andExpect(MockMvcResultMatchers.*status*().isOk())  .andExpect(MockMvcResultMatchers.*jsonPath*(**"$.length()"**)  .value(3));  } } |

然后右键方法whenQuerySuccess,出现如下报错：

|  |
| --- |
| Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'requestMappingHandlerAdapter' defined in class path resource [org/springframework/boot/autoconfigure/web/WebMvcAutoConfiguration$EnableWebMvcConfiguration.class]: Bean instantiation via factory method failed; nested exception is org.springframework.beans.BeanInstantiationException: Failed to instantiate [org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter]: Factory method 'requestMappingHandlerAdapter' threw exception; nested exception is java.lang.NoClassDefFoundError: net/minidev/asm/FieldFilter |

根据红色的报错信息，提示说缺少net.minidev.asm.FieldFilter类没有找到，在网上搜索net.minidev项目，然后在maven依赖中添加依赖

|  |
| --- |
| <**dependency**>  <**groupId**>net.minidev</**groupId**>  <**artifactId**>asm</**artifactId**>  <**version**>1.0</**version**> </**dependency**> |

然后再次运行项目，发现问题得以解决。

六、[hibernate validation HV000030: No validator could be found for constraint](https://www.cnblogs.com/xlh91118/p/8125820.html)

使用hibernate validator出现上面的错误， 需要 注意

@NotNull 和 @NotEmpty  和@NotBlank 区别

@NotEmpty 用在集合类上面  
@NotBlank 用在String上面  
@NotNull    用在基本类型上