创建Maven项目时，更改了%MAVEN\_HOME%/conf/settings.xml

1. 创建本地库 <localRepository>H:/Application/maven/repository</localRepository>
2. 再标签<mirrors>内创建阿里镜像

<mirror>

<id>alimaven</id>

<name>aliyun maven</name>

<url>http://maven.aliyun.com/nexus/content/groups/public/</url>

<mirrorOf>central</mirrorOf>

</mirror>

1. 更改项目jdb版本为jdk1.8，项目编译版本1.8

<properties>

<java.version>1.8</java.version>

</properties>

<parent>

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

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.4.4.RELEASE</version>

</parent>

1. 添加SpringBoot依赖

<dependencies>

<dependency>

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

<artifactId>spring-boot-starter-web</artifactId>

<version>1.4.4.RELEASE</version>

</dependency>

<dependency>

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

<artifactId>spring-boot-starter-web</artifactId>

<version>1.4.4.RELEASE</version>

</dependency>

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<version>1.4.4.RELEASE</version>

</dependency>

<dependency>

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

<artifactId>spring-boot-starter-data-jpa</artifactId>

<version>1.4.4.RELEASE</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<version>5.1.6</version>

</dependency>

<!--JavaServer Pages Standard Tag Library，JSP标准标签库-->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

</dependency>

<!--内置tomcat对Jsp支持的依赖，用于编译Jsp-->

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

<scope>provided</scope>

</dependency>

</dependencies>

1. 书写app.java启动类

注意：controller/service等需要在app.java的同包或者子包

页面出现“Whitelabel Error Page”说明启动类没有加扫描注解

@ComponentScan(basePackages = {"com.dbp.app.\*"})

1. 发现没有webapp这个文件，这时候我们需要自己创建webapp文件夹

首先在src/main/resource下创建一个application.properties名称是固定的

随后在文件中添加访问jsp文件目录前后缀

server.port=80

spring.mvc.view.prefix=/WEB-INF/jsp/

spring.mvc.view.suffix=.jsp

随后我们在src/main下创建webapp文件夹，webapp下创建WEB-INF/jsp文件夹

这时候 我们视图解析器就会自动在返回的字符串中添加前后缀，寻找这个文件夹下的jsp文件

1. SpringBoot默认静态文件访问顺序:
2. src/main/resources/META-INF/resources
3. src/main/resources/resources
4. src/main/resources/static
5. src/main/resources/public

更改默认访问静态文件路径在properties文件中加：

spring.resources.static-locations：classpath(src/main/resources):\*\*\*/\*\*\*/、