springBoot学习笔记

记录时间：2019.09.19

公司名称： 火鹰科技有限公司

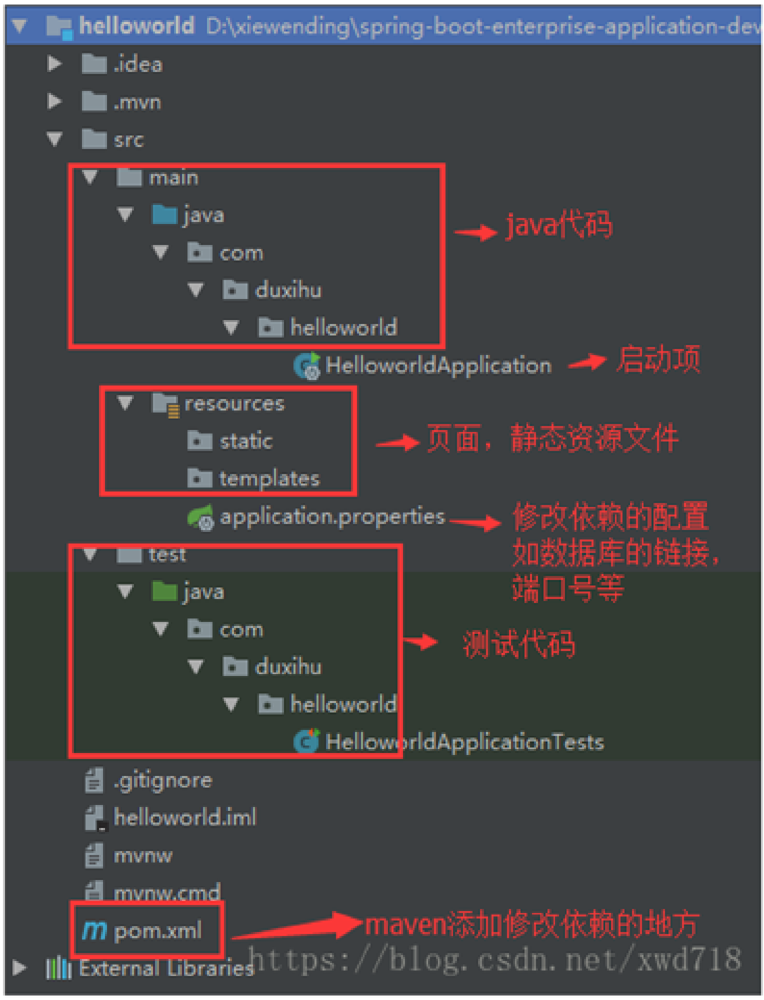
学生姓名： 曾伟涛

记录类型： 学习笔记

记录阶段如下：

2019.09.19-- 2019.09.22 自我研究springBoot

（1）第一个springBoot项目目录结构



（2）应用程序入口

```
package com.example.demo
import org.springframework.boot.SpingApplication
import org.springframework.boot.autofigure.EnableAutoConfiguration
import org.springframework.boot.autoconfigure.jdbc.DataSourceConfiguration
import org.springframeword.context.annotation.ComponentScan

@SpringBootApplication //springBootc项目核心注释，目的开启自动配置
@ComponentScan("com.ecample.demo")//扫描com.example.demo包下所有文件
```

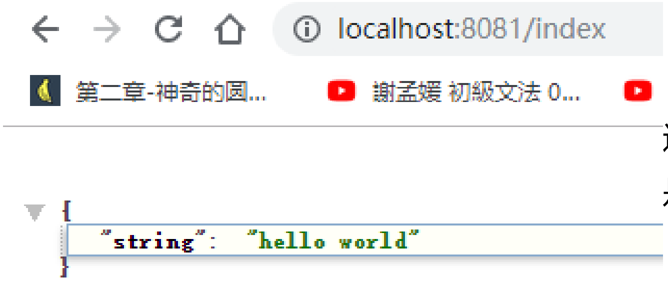
```
@EnableAutoConfiguration //将所有符合自动配置条件的bean定义加载到IoC容器
 public class DemoApplication{
  public static void main(String[] args){
  SpringApplication.run(DemoApplication.class,args);
  }
 }
```

(3）控制类(controllee) 下 indexController.java

```
/*
 此处省略导入的包
*/

@RestController // @Controller+@ResponseBody两个注解的结合，返回json数据
public class indexController{
 RequsetMapping("/index") //映射url
 public String index(Map<String, String> map){
 map.put("string","Hello world!");
  return map;
  }
 }
```

(4）测试结果

在浏览器地址栏输入： localhost:8081/index,出现 以json格式显示; "string" : "Hello world!";



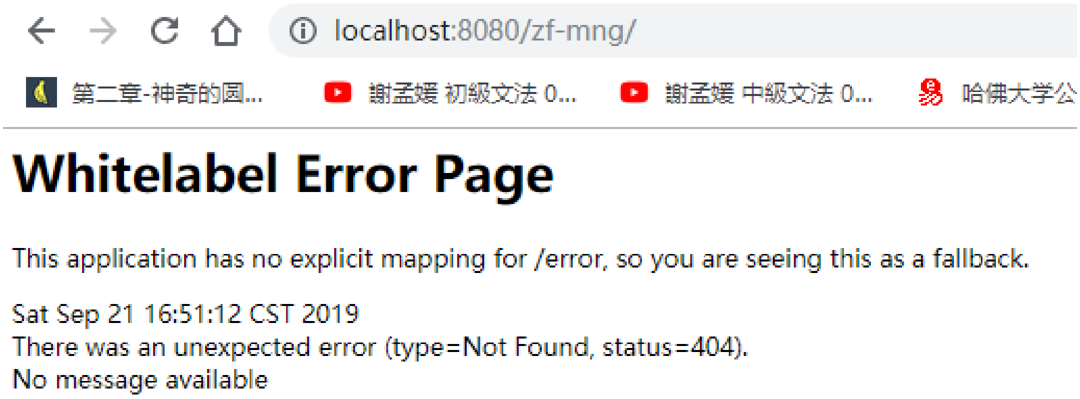
```
{
  "string": "hello world"
}
```

这里的地址端口是我手动更改设置，默认端口是8080

更改配置在resource文件夹下application.properties里配置，在里面添加：

```
server.port = 8081
```

若出现如下情况：（莫慌莫慌）



Whitelabel Error Page

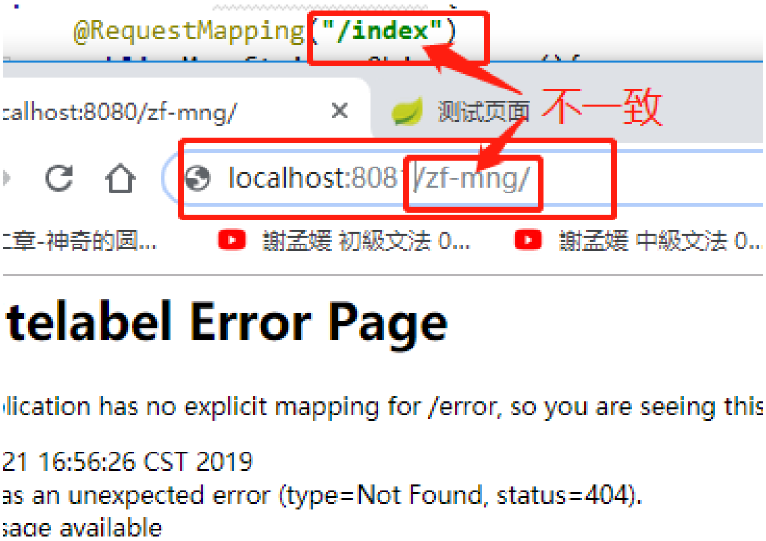
This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sat Sep 21 16:51:12 CST 2019
There was an unexpected error (type=Not Found, status=404).
No message available

莫慌原因： 表明springBoot已正常启动，只是未找到相应文件，故此回到相应代码区查看代码

解决方案：

1.观察代码区某个类中的@RequrtMapping("/xxx")与地址栏输入是否一致



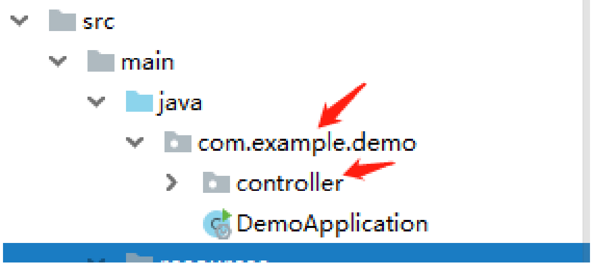
```
@RequestMapping("/index")
```

telabel Error Page

lication has no explicit mapping for /error, so you are seeing this

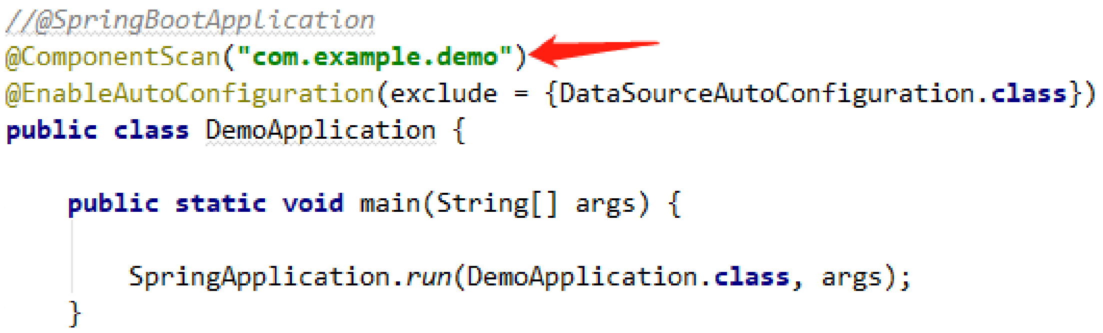
21 16:56:26 CST 2019
as an unexpected error (type=Not Found, status=404).
sage available

2.看是否所写的Java类是否在java包的子包内



controller包是在com.example.demo下

3.应用主程序内是否有注解@Compenonscan("包名");



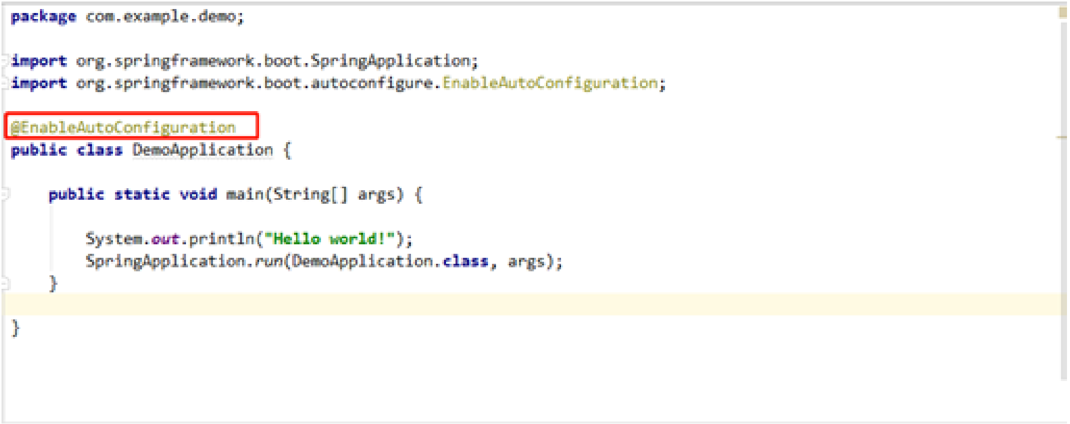
```
//@SpringBootApplication
@ComponentScan("com.example.demo")
@EnableAutoConfiguration(exclude = {DataSourceAutoConfiguration.class})
public class DemoApplication {

    public static void main(String[] args) {

        SpringApplication.run(DemoApplication.class, args);
    }
```

附录：学习springBoot遇到的坑

1.Unable to start web server; nested exception is org.springframework.context.ApplicationContextException: Unable to start ServletWebServerApplicationContext due to missing ServletWebServerFactory bean.

解决方案:

在类上方添加@EnableAutoConfiguration注释

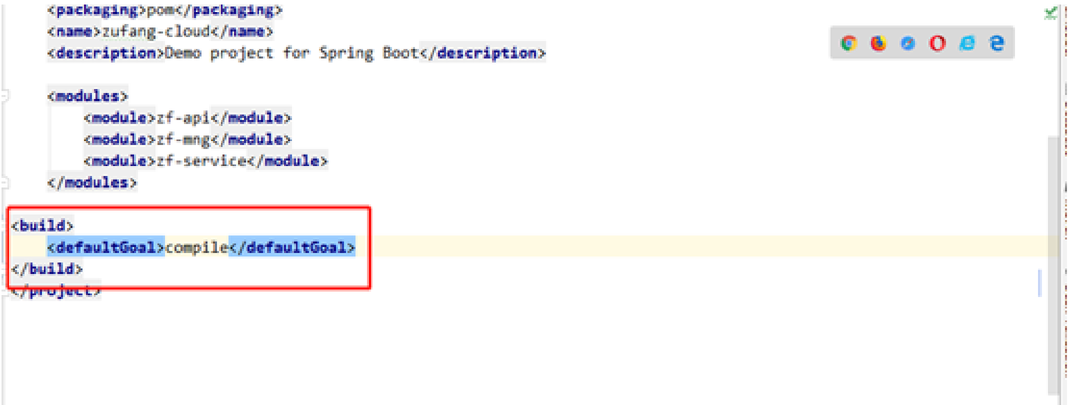


```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

@EnableAutoConfiguration
public class DemoApplication {

    public static void main(String[] args) {

        System.out.println("Hello world!");
        SpringApplication.run(DemoApplication.class, args);
    }

}
```

2.[ERROR] No goals have been specified for this build. You must specify a valid lifecycle phase or a goal in the format <plugin-prefix>:<goal> or <plugin-group-id>:<plugin-artifact-id>[:<plugin-version>]:<goal>. Available lifecycle phases are: validate, ini.......

解决方案：

在主要的pom.xml文件中在<bulid></bulid>中添加<defaultGoal>compile</defaultGoal>即可

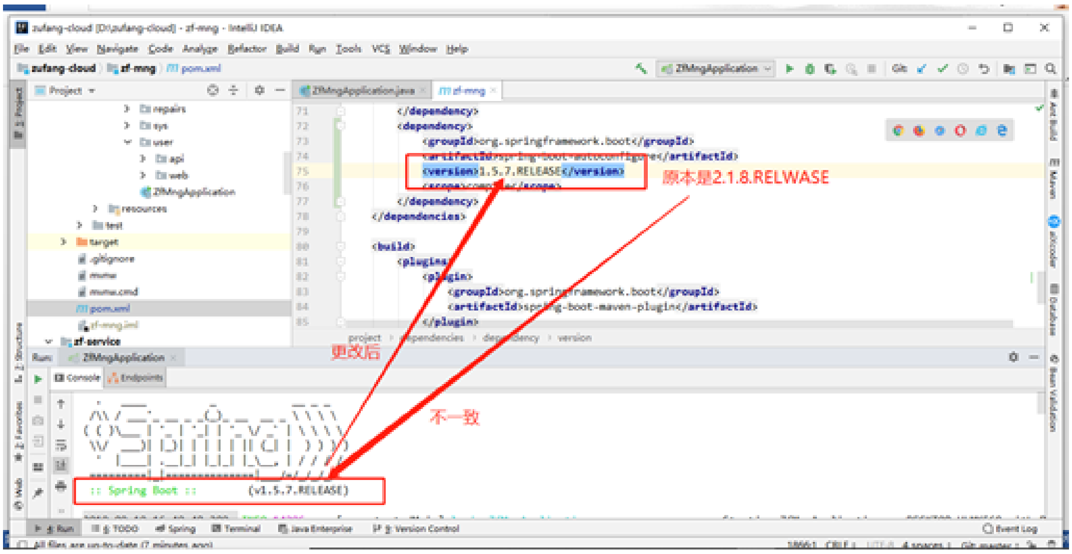


```
    <packaging>pom</packaging>
    <name>zufang-cloud</name>
    <description>Demo project for Spring Boot</description>

    <modules>
        <module>zf-api</module>
        <module>zf-mng</module>
        <module>zf-service</module>
    </modules>

<build>
    <defaultGoal>compile</defaultGoal>
</build>
</project>
```

3.java.lang.IllegalStateException: LifecycleProcessor not initialized - call '....或者 java.lang.NoSuchMethodError: org.springframework.beans.factory.support.BeanDefinitionBuilder.genericBeanDefinition

根源：一般是spring boot 版本与maven 依赖的jar 冲突

解决方案：在pom.xml文件中找到spring –boot版本是否与运行的spring boot版本一致，若不一致更改为一致版本

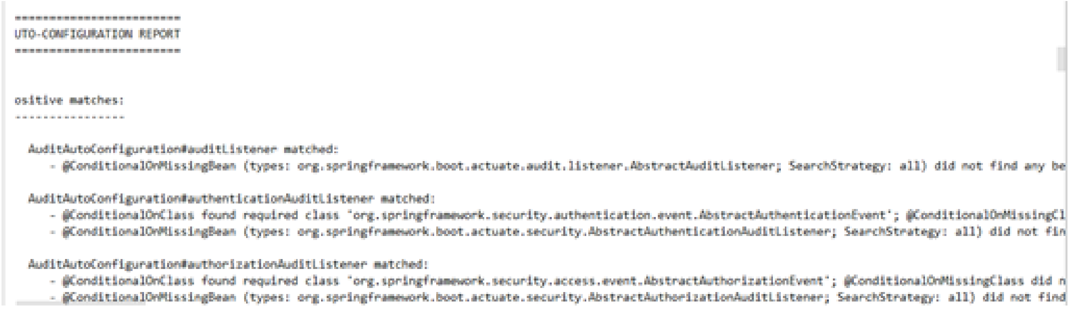


```
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-autoconfigure</artifactId>
                <version>1.5.7.RELEASE</version>
                <scope>compile</scope>
            </dependency>
        </dependencies>

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

```
 :: Spring Boot ::        (v1.5.7.RELEASE)
```

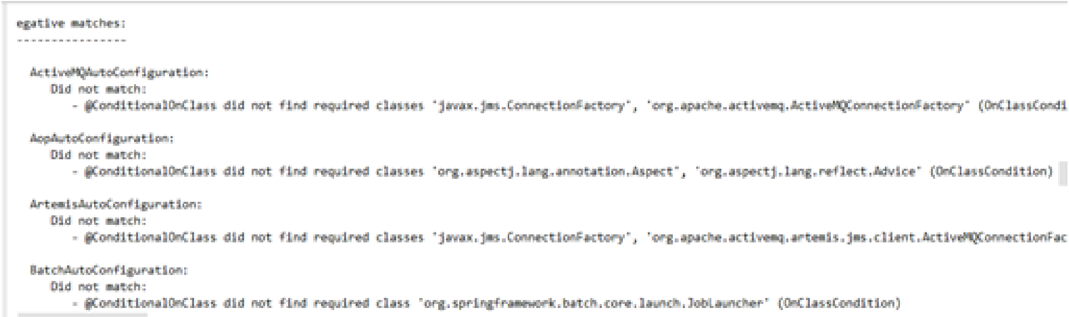
4. 以下情况非bug情况

控制台输出，已启用的自动配置：



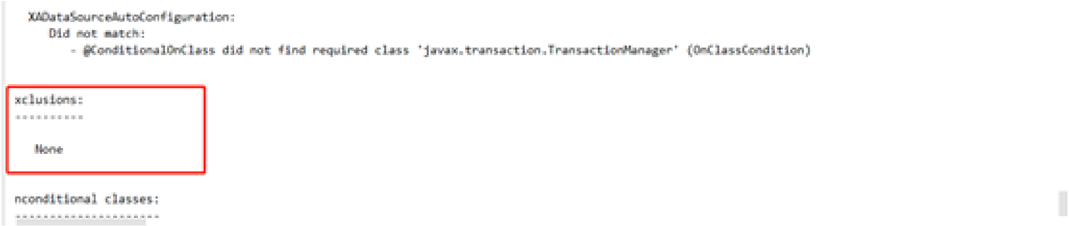
```
=========================
UTO-CONFIGURATION REPORT
=========================


ositive matches:
-----------------

   AuditAutoConfiguration#auditListener matched:
      - @ConditionalOnMissingBean (types: org.springframework.boot.actuate.audit.listener.AbstractAuditListener; SearchStrategy: all) did not find any be

   AuditAutoConfiguration#authenticationAuditListener matched:
      - @ConditionalOnClass found required class 'org.springframework.security.authentication.event.AbstractAuthenticationEvent'; @ConditionalOnMissingCl
      - @ConditionalOnMissingBean (types: org.springframework.boot.actuate.security.AbstractAuthenticationAuditListener; SearchStrategy: all) did not fin

   AuditAutoConfiguration#authorizationAuditListener matched:
      - @ConditionalOnClass found required class 'org.springframework.security.access.event.AbstractAuthorizationEvent'; @ConditionalOnMissingClass did n
      - @ConditionalOnMissingBean (types: org.springframework.boot.actuate.security.AbstractAuthorizationAuditListener; SearchStrategy: all) did not find
```

未启用的自动配置：



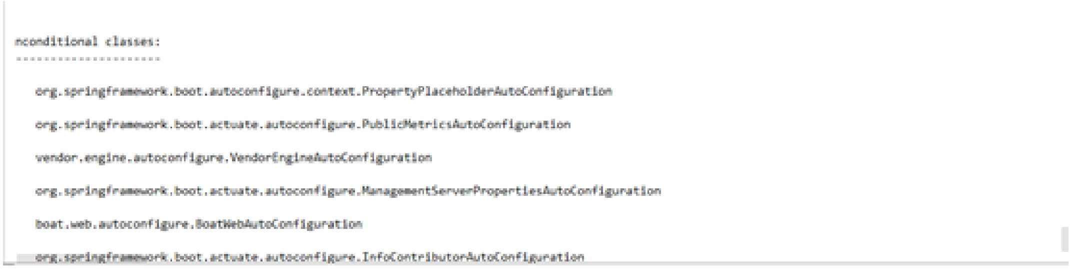
```
egative matches:
-----------------

  ActiveMQAutoConfiguration:
     Did not match:
        - @ConditionalOnClass did not find required classes 'javax.jms.ConnectionFactory', 'org.apache.activemq.ActiveMQConnectionFactory' (OnClassCondi

  AopAutoConfiguration:
     Did not match:
        - @ConditionalOnClass did not find required classes 'org.aspectj.lang.annotation.Aspect', 'org.aspectj.lang.reflect.Advice' (OnClassCondition)

  ArtemisAutoConfiguration:
     Did not match:
        - @ConditionalOnClass did not find required classes 'javax.jms.ConnectionFactory', 'org.apache.activemq.artemis.jms.client.ActiveMQConnectionFac

  BatchAutoConfiguration:
     Did not match:
        - @ConditionalOnClass did not find required class 'org.springframework.batch.core.launch.JobLauncher' (OnClassCondition)
```

已排除的自动配置:



```
  XADataSourceAutoConfiguration:
     Did not match:
        - @ConditionalOnClass did not find required class 'javax.transaction.TransactionManager' (OnClassCondition)

xclusions:
----------

   None

nconditional classes:
---------------------
```

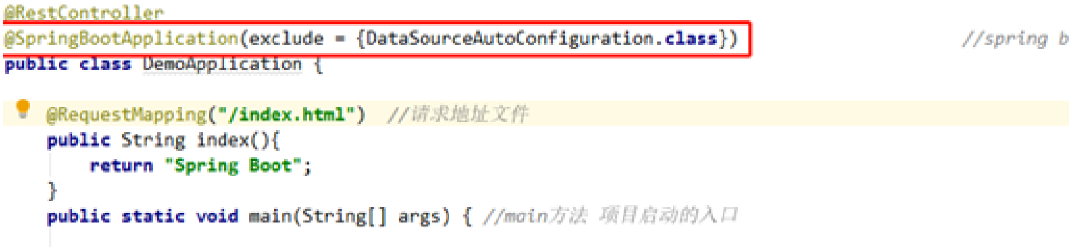
无条件类:



```
nconditional classes:
---------------------

   org.springframework.boot.autoconfigure.context.PropertyPlaceholderAutoConfiguration

   org.springframework.boot.actuate.autoconfigure.PublicMetricsAutoConfiguration

   vendor.engine.autoconfigure.VendorEngineAutoConfiguration

   org.springframework.boot.actuate.autoconfigure.ManagementServerPropertiesAutoConfiguration

   boat.web.autoconfigure.BoatWebAutoConfiguration

   org.springframework.boot.actuate.autoconfigure.InfoContributorAutoConfiguration
```

5.ERROR 13052 --- [main] o.s.b.d.LoggingFailureAnalysisReporter   :

解决方案 ： 在@SpringBootApplication注解后面加上(exclude {DataSourceAutoConfiguration.class})



```
@RestController
@SpringBootApplication(exclude = {DataSourceAutoConfiguration.class})           //spring b
public class DemoApplication {

    @RequestMapping("/index.html")  //请求地址文件
    public String index(){
        return "Spring Boot";
    }
    public static void main(String[] args) { //main方法 项目启动的入口
```