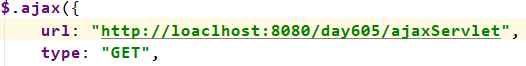
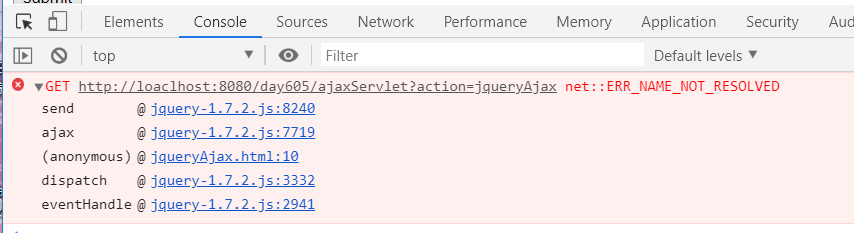
# Java

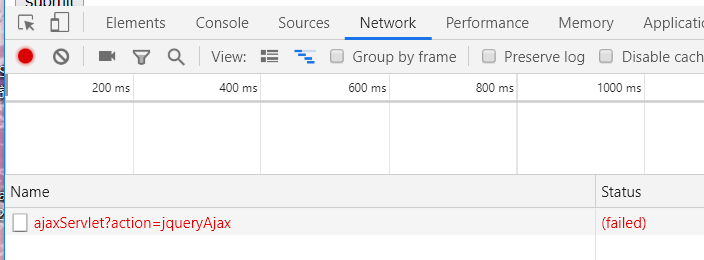
# JavaScript

## localhost拼写错误



应该拼写为localhost，此时不会在idea控制台显示错误信息，但是会在浏览器端的控制台显示错误信息。





# SpringMVC

## ClassNotFoundException



这个bug是由于Spring框架的jar没有导入完全产生的，只要导入相关jar就可以解决这个问题。

# SpringCloud

## ZuulException

com.netflix.zuul.exception.ZuulException: Hystrix Readed time out

这个错误是zuul的默认超时时间比较小，配置下zuul的超时时间，因zuul启用了ribbon的负载均衡，还需要设置ribbon的超时时间，注意ribbon的超时时间要小于zuul超时时间 。

因此在网关的项目的application.yml配置文件中做如下配置：

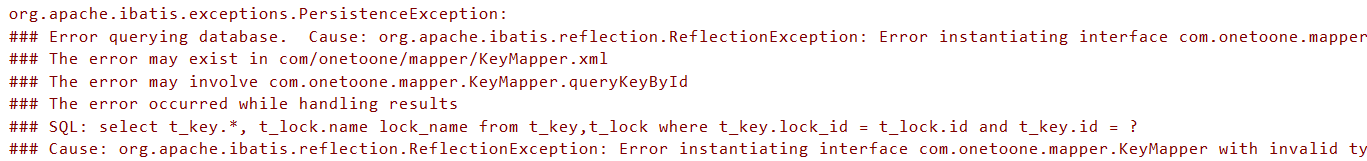
|  |
| --- |
| zuul:  host:  connect-timeout-millis: 15000 *#HTTP连接超时要比Hystrix的大* socket-timeout-millis: 60000 *#socket超时* ribbon:  ReadTimeout: 10000  ConnectTimeout: 10000 |

# Mybatis

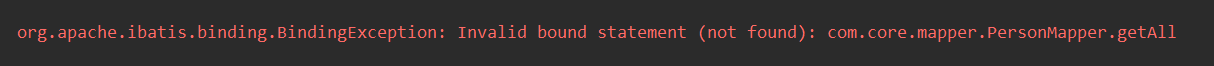
## PresistenceException



resultMap标签中的type属性应该是JavaBean的全类名，不要写成Mapper接口的全类名，否则会抛出如下异常：



## BindingException



原因是mapper.xml配置文件没有和mapper接口绑定，绑定方式是在mapper.xml中的mapper的namespace输入mapper接口的全类名。

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?> <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd" > <!--namespace中绑定的是对应JavaBean的mapper接口，否则会抛BindingException异常--> <mapper namespace="com.core.pojo.Person" >  <select id="getAll" resultType="com.core.pojo.Person">  select id from t\_person;  </select> </mapper> |

# Dubbo

## BindException

[DUBBO] qos-server can not bind localhost:22222, dubbo version: 2.6.0, current host: 127.0.0.1

在启动项目时显示dubbo的qos-server无法绑定端口号，针对dubbo的版本不同，解决方式也不同。

dubbo-2.6.0之前

在SpringBoot的src/main/resources目录下创建dubbo.properties文件并添加如下配置

|  |
| --- |
| dubbo.qos.accept.foreign.ip=true #给qos-server绑定不同于22222的端口号 dubbo.qos.port=22223 dubbo.qos.enable=false |

dubbo-2.6.0之前

在SpringBoot的src/main/resources目录下创建dubbo.properties文件并添加如下配置

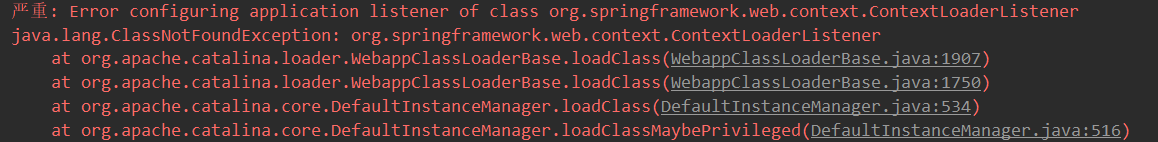
|  |
| --- |
| dubbo.application.qos.accept.foreign.ip=true #给qos-server绑定不同于22222的端口号 dubbo.application.qos.port=22223 dubbo.application.qos.enable=false |

# Maven

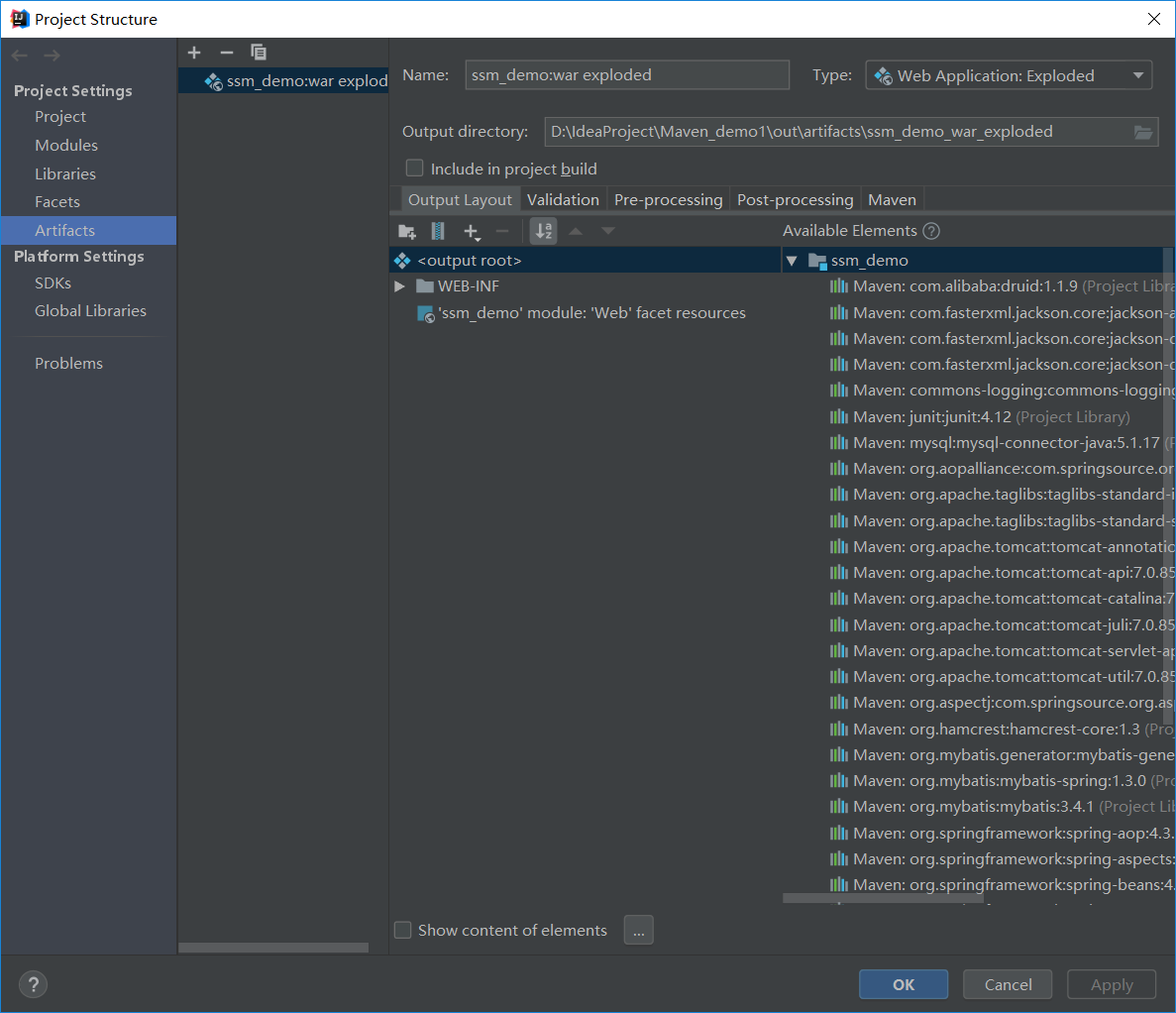
## ClassNotFoundException

严重: Error configuring application listener of class org.springframework.web.context. ContextLoaderListener

java.lang.ClassNotFoundException: org.springframework.web.context.ContextLoaderListener



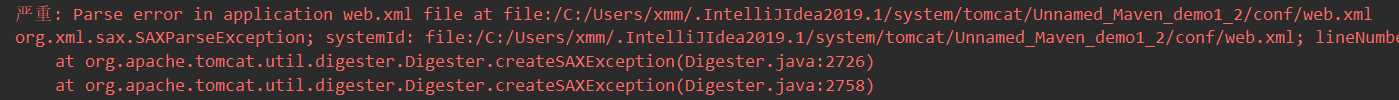
idea中出现这种异常是因为没有在module中导入相关的jar包，解决方式如下：在Project Structure中的Artifacts中选择对应的module，然后选择Available Elements中的jar包右键Put into /WEB-INF/lib/即可解决这个BUG。



## SAXParseException

严重: Parse error in application web.xml file at file:/C:/Users/xmm/.IntelliJIdea2019.1/system/ tomcat/Unnamed\_Maven\_demo1\_2/conf/web.xml

org.xml.sax.SAXParseException; systemId: file:/C:/Users/xmm/.IntelliJIdea2019.1/system/tomcat/ Unnamed\_Maven\_demo1\_2/conf/web.xml; lineNumber: 14; columnNumber: 13;

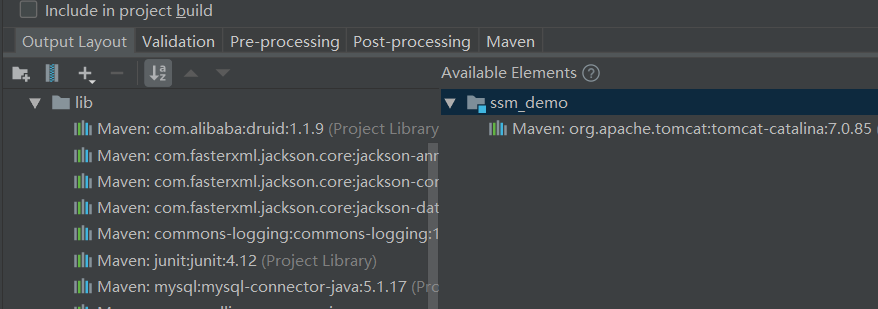


出现这个异常的原因是因为tomcat中的catalina.jar，tomcat-coyote.jar包冲突，所以有如下俩种种方式解决这个异常：

1、将该module中的org.apache.tomcat:tomcat-catalina.jar右键remove移除即可

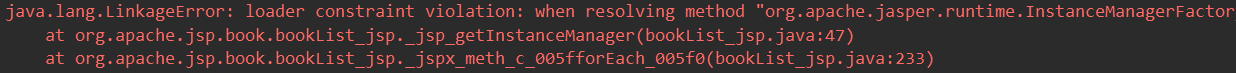
2、在tomacat的context.xml里加上<Loader delegate="true" />

这三种方式应该对应不同的项目来使用的，第二种方式最简单，最容易实现。但对于第二种方式由于是修改tomcat的配置文件，不知道以后会引起什么异常。所以如果能使用第一种或者第三种方式解决，最好不要使用第二种。或者在真正部署程序的时候，将tomcat中修改的文件修改回来。

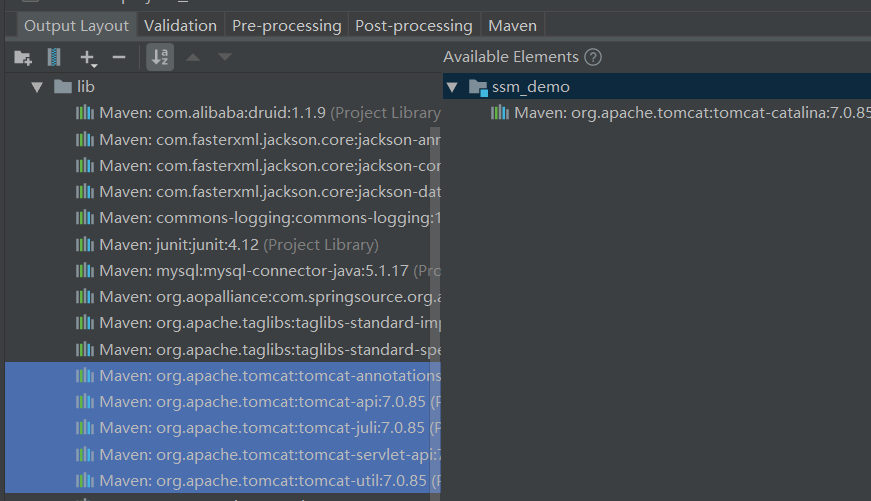


## LinkageError

java.lang.LinkageError: loader constraint violation: when resolving method "org.apache.jasper. runtime.InstanceManagerFactory.getInstanceManager(Ljavax/servlet/ServletConfig;)Lorg/apache/tomcat/InstanceManager;" …



出现这种BUG是因为module应用中不应该包含WEB服务相关（即和tomcat相关）的jar包，只要将module中的和tomcat相关的jar移除即可。



## BeanDefinitionStoreException

严重: Exception sending context initialized event to listener instance of class org.springframework.web.context.ContextLoaderListener

org.springframework.beans.factory.BeanDefinitionStoreException: IOException parsing XML document from class path resource [spring/applicationContext.xml]; nested exception is java.io.FileNotFoundException



出现该异常的原因可能如下：

1、该maven项目中的pom.xml文件中将该module配置成了父maven项目，其packaging方式设置成了pom， 此异常会导致resources目录中的配置文件无法被编译读取，从而导致FileNotFoundException。

|  |
| --- |
| <modelVersion>4.0.0</modelVersion>  <packaging>pom</packaging> <modules>  <module>../ssm\_demo2</module> </modules> <groupId>com.xmm</groupId> <artifactId>Maven\_demo1</artifactId> <version>1.0-SNAPSHOT</version> |

此时解决该异常的方式只要删除相关的配置标签即可。

|  |
| --- |
| <modelVersion>4.0.0</modelVersion> <groupId>com.xmm</groupId> <artifactId>Maven\_demo1</artifactId> <version>1.0-SNAPSHOT</version> |

# Thymeleaf

## TemplateInputException

org.thymeleaf.exceptions.TemplateInputException: Error resolving template "xxx/xxx", template might not exist or might not be accessible by any of the configured Template Resolvers

1. 在controller层请求处理完了返回时，没有使用@RestController或@ResponseBody而返回了非json格式，这种情况下返回的数据thymeleaf模板无法解析，直接报错，本人正式因为这个原因才报错。

解决方案：可以将@Controller换成@RestController，不过需要注意有没有其他的方法返回了html页面，会导致返回的不是页面而是字符串；最好的方法就是在你所请求的方法上面加一个@ResponseBody即可。

2. 在你的controller层对应的方法返回html路径及名称时，在前面多加了一个/ 。

例如return "/index"，正式这个/导致报错的，解决：去掉返回前面的/即可，例如return "/index" 。

3. 在使用springboot的过程中，如果使用thymeleaf作为模板文件，则要求HTML格式必须为严格的html5格式，必须有结束标签，否则会报错。此时可以通过添加松校验的依赖和配置来解决这个问题。

|  |
| --- |
| spring.thymeleaf.mode=LEGACYHTML5 spring.thymeleaf.cache=false |

|  |
| --- |
| <dependency>  <groupId>net.sourceforge.nekohtml</groupId>  <artifactId>nekohtml</artifactId>  <version>1.9.20</version> </dependency> |

4. 资源文件的路径被修改，如果你其他的请求都正常返回则可忽略这条。解决：在pom.xml文件的<build></build>中加入如下配置：

|  |
| --- |
| <build>  <resources>  <resource>  <directory>src/main/resources</directory>  </resource>  </resources> </build> |

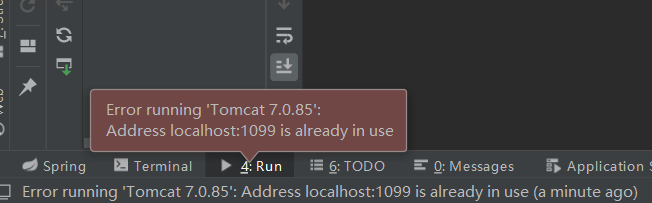
# Tomcat

## 端口号占用

### 1099占用

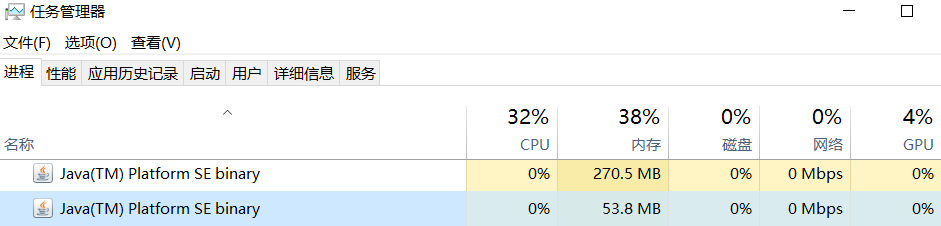
#### bug描述

1099 is already in use，目前发现出现这种bug伴随着IDEA闪退。



#### 解决方法

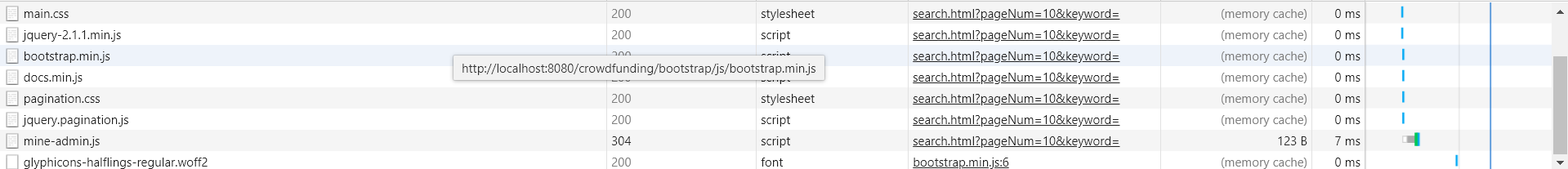
在任务管理器中结束这两个进程。



# Browser（chrome）

## 缓存问题

有时一些静态资源修改以后浏览器中却没有改变，可以在开发者工具中的Network查看是否是memory cache造成的改变无法生效。



可以在chrome浏览器缓存文件位置处右键调出菜单选择Clear browser cache选项来清除缓存。

或者使用**CTRL+F5**强制刷新。