**SpringBoot：**

**problem：**

**application.properties与application.yml的区别？**

**solution：**

1. **properties优先级更高。**
2. **server、port的配置方式不同。**

**problem：**

**使用SpringJPA时，发现如果实体类中的属性采用“驼峰命名法”，则会出现Error——“org.springframework.dao.InvalidDataAccessResourceUsageException: could not extract ResultSet; SQL [n/a]; nested exception is org.hibernate.exception.SQLGrammarException: could not extract ResultSet”**

**solution：**

**SpringJPA只是一个规范。**

**SpringJPA的默认规则是只读“小写”，在application.properies中添加——**

**“**spring.jpa.hibernate.naming.physical-strategy=**org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl**

**”**

**更改springJPA的命名规范，使其能够读取“大小写组合”，则完美解决。**

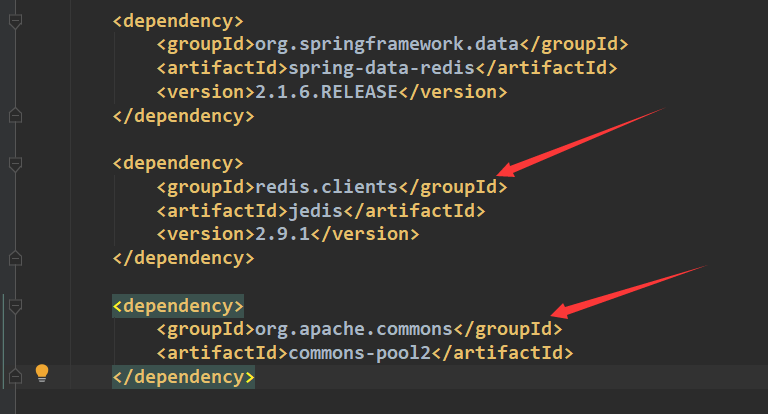
**problem：**

**集成redis时，“Caused by: java.lang.ClassNotFoundException: redis.clients.util.Pool”。**

**solution：**

**jedis的版本过高造成，降低到3.0.0以下，okay!**

**且引入redis时，三种dependency缺一不可！**



**Problem：**

**Redis插入数据插入后，key值为乱码？**

**Solution：**

**添加配置文件RedisConfig。解决！**

**Problem：**

**引入log4j2日志配置时报错！**

**“Logging system failed to initialize using configuration from 'classpath:log4j2.xml'**

**java.lang.IllegalStateException: Logback configuration error detected:**

**”**

**Solution：**

**实际上是pom中jar包冲突的问题！**

**不能仅在starter-web中除去(exclusions) logging！其他dependency中也可能默认配置了logging！**

**<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>**

**<exclusions>  
 <exclusion>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-logging</artifactId>  
 </exclusion>  
 </exclusions>  
</dependency>**

**所以，要直接在根starter中除去(exclusions)！**

**<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter</artifactId>  
 <exclusions>  
 <exclusion>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-logging</artifactId>  
 </exclusion>  
 </exclusions>  
</dependency>**

**Problem：**

**Java Mail Exception“Authentication failed; nested exception is javax.mail.AuthenticationFailedException”**

**Solution：**

**发送方的邮件要开通“POP3 / SMTP服务”，并遵守**

**邮箱配置，且要“开启服务”！**



**邮件主机(如：smtp.qq.com、smtp.163.com)必须和发件人的主机一致。**

**邮件授权码(password)：开启服务时会给，或者就是邮箱的登录密码。**



**以下为QQ邮箱开启服务方式：**

**开启服务：要开启“POP3 / SMTP服务”！**



**Problem：**

**SpringBoot集成Mybatis时：**

1. **出现的(not found)的问题。**
2. **“No enum constant org.apache.ibatis.type.JdbcType.INT”。**

**Solution：**

**1.按此博客：“**[**https://blog.csdn.net/weixin\_44440658/article/details/88194381**](https://blog.csdn.net/weixin_44440658/article/details/88194381)**”；基本解决！**

**2. <!-- jdbcType的值必须是enum中存在的类型，且必须大写 -->**

**Problem：**

**EmbeddedServletContainerCustomizer不存在**

**Solution：**

**SpringBoot2.0 以上，使用“WebServerFactoryCustomizer”接口替换“EmbeddedServletContainerCustomizer”组件完成对嵌入式Servlet容器的配置”。**

**Problem：**

**SpringBoot启动Tomcat失败：[StandardEngine[Tomcat].StandardHost[localhost]**

**Solution：**

**原因是挂了SSR，ip地址跳转到了境外的原因！**

**解决方案：**

**进入Tomcat的目录，“手动启动” 之后再 “手动关闭” Tomcat服务器。**

**之后cmd，ping localhost ，如果成功则说明SSR已彻底关闭可使用！**

**修正： 观察ActiveMQ和redis命令行的状态，如有异常，关闭重启即可！**

**Problem：**

**Spring Boot 2.x中的“management.security.enabled=false”无效问题！**

**Solution：**

**无效的原因是被弃用！**

**新的配置为：**

**management.endpoints.web.exposure.include=\***

**注：**

**同时要将SpringSecurity注释掉，或者更改其“configure”方法如下：**

**protected void configure(HttpSecurity *http*) throws Exception { // 重写configure方法  
  
// // 路由策略和访问权限的简单配置  
// http.formLogin() // 启动默认登录页面  
// .failureUrl("/login?error") // 登陆失败, 返回URL：/login?error  
// .defaultSuccessUrl("/user/findAll") // 登陆成功跳转URL, 这里跳转到用户首页  
// .permitAll(); // 登陆页面全部权限可访问  
//  
// super.configure(http);  
  
 *http*.authorizeRequests().anyRequest().permitAll().and().logout().permitAll();  
  
 }**

**尤其需要注意的是：springboot2.0后，挂载点发生变化！所有节点都改挂在/actuator上了。**

**即，如：**

[**http://localhost:8080/actuator/**](http://localhost:8080/actuator/)

**当然，也可通过更改相关的配置修改挂载点。**

**还有，application.properties内的配置为：**

**# 设置关闭安全限制  
management.endpoints.web.exposure.include=\***

**SpringBoot2.0官方迁移：**

[**http://blog.didispace.com/Spring-Boot-2.0-Migration-Guide/**](http://blog.didispace.com/Spring-Boot-2.0-Migration-Guide/)

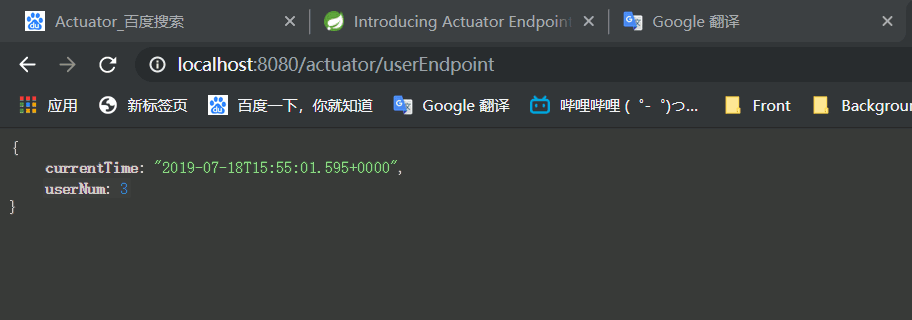
**Problem：**

**SpringBoot2.0中，Actuator自定义端点书写方式完全改变。**

**Solution：**

**见：**

**package season.spring.actuator;  
  
import org.springframework.boot.actuate.autoconfigure.endpoint.condition.ConditionalOnEnabledEndpoint;  
import org.springframework.boot.actuate.endpoint.annotation.Endpoint;  
import org.springframework.boot.actuate.endpoint.annotation.ReadOperation;  
import org.springframework.boot.autoconfigure.condition.ConditionalOnBean;  
import org.springframework.boot.autoconfigure.condition.ConditionalOnMissingBean;  
import org.springframework.boot.logging.LoggingSystem;  
import org.springframework.context.annotation.Bean;  
import org.springframework.stereotype.Component;  
import season.spring.service.UserService;  
  
import javax.annotation.Resource;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.Map;  
  
@Endpoint(id = "userEndpoint")  
@Component  
public class UserEndpoint {  
  
 @Resource  
 private UserService userService;  
  
 @ReadOperation  
 public Map<String, Object> userNum() {  
  
 Map<String, Object> *map* = new HashMap<String, Object>();  
  
 // 当前时间  
 *map*.put("currentTime", new Date());  
  
 // 用户总数  
 *map*.put("userNum", userService.findUserTotalNum());  
  
 return *map*;  
  
 }  
  
  
 @Bean  
 @ConditionalOnBean(LoggingSystem.class)  
 @ConditionalOnMissingBean  
 @ConditionalOnEnabledEndpoint  
 public UserEndpoint userEndpoint() {  
  
 return new UserEndpoint();  
  
 }  
  
}**



**Problem：**

**启动Tomcat报错“java.lang.IllegalStateException: Illegal access: this web application instance has been stopped already. Could not load []. The following stack trace is thrown for debugging purposes as well as to attempt to terminate the thread which caused the illegal access.”**

**Solution：**

**在IDE中关闭Tomcat后，有些线程还未被杀死，所以，需要手动关闭/重启Tomcat！**