# redis

自己在windows下开发，连的redis没有任何问题，部署到linux机器上就开始提示报错Cannot open Redis connection due invalid URI

解决问题：需要指定类型：看下面

<bean id="jedisPool" class="redis.clients.jedis.JedisPool">

<constructor-arg index="0" ref="poolConfig" />

<constructor-arg index="1" value="${redis.host}" type="java.lang.String" />

<constructor-arg index="2" value="${redis.port}" type="int"/>

</bean>

如果constructor-arg 未指定type 初始化的时候JedisPool有两个相同参数个数的构造方法会选择错误，造成类似于这样的错误。

JedisPool(GenericObjectPoolConfig, String, int)

JedisPool(GenericObjectPoolConfig, URI, int)

# spring

nested exception is java.lang.IllegalArgumentException:

出现如下问题：

Could not resolve placeholder 'host' in value "${host}"; nested exception is java.lang.IllegalArgumentException: Could not resolve placeholder 'host' in value "${host}"

这意思是找不到properties文件或key，之前我是照老师写通用的写法：

<context:property-placeholder location="classpath:\*.properties"/>这样好像版本问题，

spring要求只能写一个此标签<context:property-placeholde>

如果多个properties文件需加载通用写法：

<context:property-placeholder location="classpath:a.properties, lasspath:b.properties"/>

ava.lang.NoClassDefFoundError

缺少 commons-logging.jar，也可以添加commons-logging-1.0.4.jar，建议使用后者或者更新版本

java.lang.IllegalStateException

# 数据源

## c3po

后台无限出现debug打印日志

无法销毁资源：[com.mchange.v2.c3p0.impl.NewPooledConnection@20ce4de8] on IDLE CHECK has SUCCEEDED【在空闲时检查成功】

Testing PooledConnection [com.mchange.v2.c3p0.impl.NewPooledConnection@61f690d6] on IDLE CHECK.【在空闲状态检查。】

出现这种debug是没有任何问题的

我看到那个DEBUG,我说，是调试信息，修改一下LOG4J的等级就行了。

这个群友很不解的问，既然成功了，干嘛还要丢异常出来？

这里就不得不说到两个商业开发的原则问题了。

第一，对上家传入数据严加过滤，对传出给下家的数据仔细检查。

第二，合理使用异常。

第一点其实很简单的。也就是模块化开发的一个思想问题。对自己的行为负责。前端返回的数据究竟是什么，需要进行校验。不合格的剔除或者是修正。合格的处理完后，在传出之前也要加以校验，是否合格。

# mybatis

org.apache.ibatis.builder.IncompleteElementException: Could not find result map

一般都是发生在：<select id="queryXXXCount" resultMap="java.util.Map" >标红地方写错了，说找不到类型

Cause: org.apache.ibatis.builder.BuilderException: Error parsing SQL Mapper Configuration. Cause: java.io.IOException: Could not find resource cn.itcast.mybatis.mapper/UserMapper.xml

出现这个问题大多数都是找不到映射文件，这和没有遵循mybatis的mapper代理配置规范有关，对于我这个问题仔细看java.io.IOException:Could not find resource cn.itcast.mybatis.mapper/UserMapper.xml,

就是文件读写出现问题，系统找不到这个文件，需要检查，mapper接口与映射的mapper.xml 的命名是否一致，是否在同一目录下。

如果还没解决就在看SqlMapConfig.xml配置文件加载mapper接口文件的路径有没有错误，一般有两种方试，第一扫描类<mapper class=:"类路径和类名"/>第二种批量扫描加载：选择扫描包<package name="包名"/>经过不懈努力问题终于解决。我们在开发mybatis时要注意四点开发规范：

1、在mapper.xml中namespace等于mapper接口地址

2、mapper.java接口中的方法名和mapper.xml中statement的id一致

3、mapper.java接口中的方法输入参数类型和mapper.xml中statement的parameterType指定的类型一致。

4、mapper.java接口中的方法返回值类型和mapper.xml中statement的resultType指定的类型一致。

报错：Cannot enable lazy loading because CGLIB is not available.

- 今天在整合mybatis+spring的懒加载+缓存时报了个 的错误，很奇怪的是我已经把CGLIB的包导入到项目了。没办法，先不用懒加载和缓存了吧。

后来上网查了一下，发现CGLIB还依赖了asm包，但是它报错只报到了CGLIB这一层，所以让我蛋疼了好一段时间。

二话不说，直接把asm的jar包也加上，搞定

宝空指针异常：Request processing failed; nested exception is java.lang.NullPointerException，java.lang.NullPointerException

这个异常是对象是null，但是你又进行了get取值属性，所以就报这样的异常

javax.el.ELException: Error reading 'employee' on type wwl.pojo.User$$EnhancerByCGLIB$$b829920f

:这个已经很明确说：User包的属性employee类型（传入或传出）不正确，

我这次出现的异常是在延迟加载的地方，是要出入属性中类型的，我在引入的select标签的传入参数使用的是包类型，延迟加载的话要引出参数类型要和引入参数类型一致，包类型是不行的：<select id="selectEmployeeById" parameterType="int（延迟加载的话这里要和传入的参数一致才行的，一般都是主键id或者外键id）" resultType="wwl.pojo.Employee">

Error querying database. Cause: java.sql.SQLException: Parameter index out of range (1 > number of parameters, which is 0)

查询原因 mybatis的like语句运用错误,或多引用了双引号一些原因，导致传入的参数，没法找到需要补位的地方

# 缺包缺类

java.lang.ClassNotFoundException 类没有找到的异常

ClassNotFoundException是一个运行时异常。从类继承层次上来看，ClassNotFoundException是从Exception继承的，所以ClassNotFoundException是一个检查异常

当应用程序运行的过程中尝试使用类加载器去加载Class文件的时候，如果没有在classpath中查找到指定的类，就会抛出ClassNotFoundException。一般情况下，当我们使用Class.forName()或者ClassLoader.loadClass以及使用ClassLoader.findSystemClass()在运行时加载类的时候，如果类没有被找到，那么就会导致JVM抛出ClassNotFoundException。

org.springframework.dao.support.DaoSupport

缺少 spring-tx包

org/springframework/jdbc/datasource/TransactionAwareDataSourceProxy

缺少spring-jdbc包

java.lang.NoClassDefFoundError 类没有找到的错误

NoClassDefFoundError异常，看命名后缀是一个Error。从类继承层次上看，NoClassDefFoundError是从Error继承的。和ClassNotFoundException相比，明显的一个区别是，NoClassDefFoundError并不需要应用程序去关心catch的问题。

当JVM在加载一个类的时候，如果这个类在编译时是可用的，但是在运行时找不到这个类的定义的时候，JVM就会抛出一个NoClassDefFoundError错误。比如当我们在new一个类的实例的时候，如果在运行是类找不到，则会抛出一个NoClassDefFoundError的错误。

org/mybatis/spring/mapper/MapperFactoryBean

1. 确认maven有没添加mybatis-spring.jar 整合包

2. 确认maven有没添加spring-jdbc