

```
1 | <center> <font size="6" color="blue">Spring的数据源配置</font></center>
```

1、场景：引入database.properties配置文件时报错：

- 重点：

```
1 | Caused by: java.lang.IllegalArgumentException: Could not resolve placeholder  
  | 'users' in string value "${users}"
```

- 报错详情：

```
1 | - (369 ms) - 2019-10-9 16:34:56[ WARN](AbstractApplicationContext.java:486)  
  | Exception encountered during context initialization - cancelling refresh  
  | attempt  
2 | org.springframework.beans.factory.BeanDefinitionStoreException: Invalid  
  | bean definition with name 'dataSource' defined in class path resource  
  | [applicationContext.xml]: Could not resolve placeholder 'users' in string  
  | value "${users}"; nested exception is java.lang.IllegalArgumentException:  
  | Could not resolve placeholder 'users' in string value "${users}"  
3 |     at  
  | org.springframework.beans.factory.config.PlaceholderConfigurerSupport.doPro  
  | cessProperties(PlaceholderConfigurerSupport.java:209)  
4 |     at  
  | org.springframework.beans.factory.config.PropertyPlaceholderConfigurer.proc  
  | essProperties(PropertyPlaceholderConfigurer.java:223)  
5 |     at  
  | org.springframework.beans.factory.config.PropertyResourceConfigurer.postPro  
  | cessBeanFactory(PropertyResourceConfigurer.java:84)  
6 |     at  
  | org.springframework.context.support.AbstractApplicationContext.invokeBeanFa  
  | ctoryPostProcessors(AbstractApplicationContext.java:696)  
7 |     at  
  | org.springframework.context.support.AbstractApplicationContext.invokeBeanFa  
  | ctoryPostProcessors(AbstractApplicationContext.java:671)  
8 |     at  
  | org.springframework.context.support.AbstractApplicationContext.refresh(Abst  
  | ractApplicationContext.java:461)  
9 |     at org.springframework.context.support.ClassPathXmlApplicationContext.  
  | <init>(ClassPathXmlApplicationContext.java:139)  
10 |    at org.springframework.context.support.ClassPathXmlApplicationContext.  
  | <init>(ClassPathXmlApplicationContext.java:83)  
11 |    at com.javaxyz.test.SpringTest.test9(SpringTest.java:48)  
12 |    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
13 |    at  
  | sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:5  
  | 7)  
14 |    at  
  | sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImp  
  | l.java:43)  
15 |    at java.lang.reflect.Method.invoke(Method.java:606)
```

```
16      at
org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod
.java:44)
17      at
org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.
java:15)
18      at
org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.j
ava:41)
19      at
org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.ja
va:20)
20      at
org.junit.runners.BlockJUnit4ClassRunner.runNotIgnored(BlockJUnit4ClassRunn
er.java:79)
21      at
org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.ja
va:71)
22      at
org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.ja
va:49)
23      at org.junit.runners.ParentRunner$3.run(ParentRunner.java:193)
24      at org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:52)
25      at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:191)
26      at org.junit.runners.ParentRunner.access$000(ParentRunner.java:42)
27      at org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:184)
28      at org.junit.runners.ParentRunner.run(ParentRunner.java:236)
29      at
org.eclipse.jdt.internal.junit4.runner.JUnit4TestReference.run(JUnit4TestRe
ference.java:50)
30      at
org.eclipse.jdt.internal.junit.runner.TestExecution.run(TestExecution.java:
38)
31      at
org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.runTests(RemoteTestR
unner.java:467)
32      at
org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.runTests(RemoteTestR
unner.java:683)
33      at
org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.run(RemoteTestRunner
.java:390)
34      at
org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.main(RemoteTestRunne
r.java:197)
35      Caused by: java.lang.IllegalArgumentException: Could not resolve
placeholder 'users' in string value "${users}"
36      at
org.springframework.util.PropertyPlaceholderHelper.parseStringValue(Property
PlaceholderHelper.java:173)
37      at
org.springframework.util.PropertyPlaceholderHelper.replacePlaceholders(Prop
ertyPlaceholderHelper.java:125)
38      at
org.springframework.beans.factory.config.PropertyPlaceholderConfigurer$Plac
eholderResolvingStringValueResolver.resolveStringValue(PropertyPlaceholderC
onfigurer.java:258)
```

```

39      at
org.springframework.beans.factory.config.BeanDefinitionVisitor.resolveStrin
gValue(BeanDefinitionVisitor.java:282)
40      at
org.springframework.beans.factory.config.BeanDefinitionVisitor.resolveValue
(BeanDefinitionVisitor.java:204)
41      at
org.springframework.beans.factory.config.BeanDefinitionVisitor.visitPropert
yValues(BeanDefinitionVisitor.java:141)
42      at
org.springframework.beans.factory.config.BeanDefinitionVisitor.visitBeanDef
inition(BeanDefinitionVisitor.java:82)
43      at
org.springframework.beans.factory.config.PlaceholderConfigurersSupport.doPro
cessProperties(PlaceholderConfigurersSupport.java:206)
44      ... 31 more
45  - (371 ms) - 2019-10-9 16:34:56[ INFO]
(DefaultSingletonBeanRegistry.java:444) Destroying singletons in
org.springframework.beans.factory.support.DefaultListableBeanFactory@6a109a
c: defining beans
[org.springframework.beans.factory.config.PropertyPlaceholderConfigurer#0,d
ataSource,sqlSessionFactory,org.mybatis.spring.mapper.MapperScannerConfigur
er#0,userService,org.springframework.context.annotation.internalConfigurati
onAnnotationProcessor,org.springframework.context.annotation.internalAutowi
redAnnotationProcessor,org.springframework.context.annotation.internalRequi
redAnnotationProcessor,org.springframework.context.annotation.internalCommo
nAnnotationProcessor,transactionManager,org.springframework.aop.config.inte
rnalAutoProxyCreator,org.springframework.transaction.annotation.AnnotationT
ransactionAttributeSource#0,org.springframework.transaction.interceptor.Tra
nsactionInterceptor#0,org.springframework.transaction.config.internalTransa
ctionAdvisor,org.springframework.context.annotation.ConfigurationClassPostP
rocessor.importAwareProcessor,userMapper]; root of factory hierarchy
46

```

2、配置文件applicationContext.xml：

引入配置文件的方式：

```

1      <!-- 快捷方式 Ctrl+Shift+H 直接拿到对象完全限定名-->
2      <bean
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigur
er">
3          <property name="location">
4              <value>classpath:database.properties</value>
5          </property>
6
7      </bean>
8      <!-- 配置数据源对象DataSource -->
9      <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">
10         <!-- 连接信息 -->
11         <property name="driverClassName" value="${driver}"></property>
12         <property name="url" value="${url}"></property>
13         <property name="username" value="${user}"></property>
14         <property name="password" value="${password}"></property>
15     </bean>

```

3、使用Jndi数据源的方式applicationContext.xml：

```
1 <!--配置数据源Jndi-->
2     <bean id="dataSource"
3         class="org.springframework.jndi.JndiObjectFactoryBean">
4         <property name="jndiName">
5             <value>java:comp/env/jdbc/java</value>
6         </property>
7     </bean>
```

4、注释applicationContext.xml：

```
1 <!-- <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">
2     连接信息
3     <property name="driverClassName" value="${driver}"></property>
4     <property name="url" value="${url}"></property>
5     <property name="username" value="${user}"></property>
6     <property name="password" value="${password}"></property>
7 </bean> -->
```

5、修改Tomcat目录conf下的配置文件context.xml

```
1 <Resource name="jdbc/java"
2     type="javax.sql.DataSource"
3     auth="Container"
4     driverClassName="com.mysql.jdbc.Driver"
5     url="jdbc:mysql:///p19_java7_mybatis"
6     username="root"
7     password="aaa"
8     maxActive="100"
9     maxIdle="50"
10    maxWait="1000"
11 />
```

6、新建java文件 在包com.javaxyz.servlet下创建UserServlet.java

```
1 /**
2  * @Author: DongGaoYun
3  * @Description:
4  * @Date 2019-10-9 下午5:11:10
5  * @Version 1.0
6  * @Company: www.springhome.org
7  */
8 package com.javaxyz.servlet;
9
10 import java.io.IOException;
11 import java.io.PrintWriter;
12 import java.text.ParseException;
13 import java.text.SimpleDateFormat;
14 import java.util.ArrayList;
15 import java.util.Date;
16 import java.util.List;
17
18 import javax.servlet.ServletException;
```

```

19 import javax.servlet.http.HttpServlet;
20 import javax.servlet.http.HttpServletRequest;
21 import javax.servlet.http.HttpServletResponse;
22
23 import org.apache.log4j.Logger;
24 import org.junit.Test;
25 import org.springframework.context.ApplicationContext;
26 import org.springframework.context.support.ClassPathXmlApplicationContext;
27
28 import com.javaxyz.entity.User;
29 import com.javaxyz.service.UserService;
30 import com.javaxyz.test.SpringTest;
31
32 /**
33  * @ClassName: UserServlet.java
34  * @Description: 描述信息
35  * @Author: DongGaoYun
36  * @URL: www.javaxyz.com 或 www.gyun.org
37  * @Email: DongGaoYun@qq.com
38  * @QQ: 1050968899
39  * @Weixin: QingYunJiao
40  * @WeixinGongZhongHao: JavaForum
41  * @Date: 2019-10-9 下午5:11:10
42  * @Version: 1.0
43  */
44 public class UserServlet extends HttpServlet {
45     private static Logger logger = Logger.getLogger(UserServlet.class);
46
47     /**
48      * The doGet method of the servlet. <br>
49      *
50      * This method is called when a form has its tag value method equals
51      * to get.
52      *
53      * @param request
54      *         the request send by the client to the server
55      * @param response
56      *         the response send by the server to the client
57      * @throws ServletException
58      *         if an error occurred
59      * @throws IOException
60      *         if an error occurred
61      */
62     public void doGet(HttpServletRequest request, HttpServletResponse
63     response)
64         throws ServletException, IOException {
65         doPost(request, response);
66     }
67
68     /**
69      * The doPost method of the servlet. <br>
70      *
71      * This method is called when a form has its tag value method equals
72      * to
73      *
74      * post.
75      *
76      * @param request

```

```

74      *           the request send by the client to the server
75      * @param response
76      *           the response send by the server to the client
77      * @throws ServletException
78      *           if an error occurred
79      * @throws IOException
80      *           if an error occurred
81      */
82      public void doPost(HttpServletRequest request, HttpServletResponse
response)
83          throws ServletException, IOException {
84
85          /*
86           * add用户信息 UserService调用 配置声明式事务
87           */
88          ApplicationContext context = new ClassPathXmlApplicationContext(
89              "applicationContext.xml");
90          // 原来是通过映射接口去调用
91          // UserMapper userMapper = (UserMapper)
context.getBean("userMapper");
92          //
93          String[] bean = context.getBeanDefinitionNames();
94          for (String string : bean) {
95              System.out.println(string);
96          }
97          // 现在是通过业务接口去调用
98          UserService service = (UserService)
context.getBean("userService");
99          User user = new User();
100         user.setUserCode("zhangxiulian");
101         user.setUserName("张秀连666");
102         /**
103         * userCode, userName, userPassword, gender, birthday, phone,
address,
104         * userRole, createdBy, creationDate
105         */
106         user.setUserPassword("123999");
107         user.setGender(0);
108         SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
109         try {
110             user.setBirthday(sdf.parse("2010-9-17"));
111         } catch (ParseException e) {
112             e.printStackTrace();
113         }
114         user.setPhone("13608880888");
115         user.setAddress("养育巷49");
116         user.setUserRole(2);
117         user.setCreatedBy(1);
118         user.setCreationDate(new Date());
119         Integer userRole = 2;
120         // int num = userMapper.addUser(user);
121         List<User> listUsers = new ArrayList<User>();
122         listUsers.add(user);
123         listUsers.add(user);
124         int num = service.saveUser(listUsers);
125         if (num > 0) {
126             logger.info("增加成功! ");
127         } else {

```

```

128         logger.error("增加失败! ");
129     }
130 }
131 }
132

```

注意：使用jndi数据源，显示结果需要启动Tomcat服务器，执行web路径: <http://localhost:9999/jjj/USERServlet>

7、Myeclipse控制台显示：

```

1  - (1061 ms) - 2019-10-9 17:15:13[DEBUG]
    (PropertySourcesPropertyResolver.java:103) Could not find key
    'spring.liveBeansView.mbeanDomain' in any property source. Returning [null]
2  org.springframework.beans.factory.config.PropertyPlaceholderConfigurer#0
3  dataSource
4  sqlSessionFactory
5  org.mybatis.spring.mapper.MapperScannerConfigurer#0
6  userService
7  org.springframework.context.annotation.internalConfigurationAnnotationProcessor
8  org.springframework.context.annotation.internalAutowiredAnnotationProcessor
9  org.springframework.context.annotation.internalRequiredAnnotationProcessor
10 org.springframework.context.annotation.internalCommonAnnotationProcessor
11 transactionManager
12 org.springframework.aop.config.internalAutoProxyCreator
13 org.springframework.transaction.annotation.AnnotationTransactionAttributeSource#0
14 org.springframework.transaction.interceptor.TransactionInterceptor#0
15 org.springframework.transaction.config.internalTransactionAdvisor
16 org.springframework.context.annotation.ConfigurationClassPostProcessor.importAwareProcessor
17 userMapper
18 - (1061 ms) - 2019-10-9 17:15:13[DEBUG](AbstractBeanFactory.java:243)
    Returning cached instance of singleton bean 'userService'
19 - (1063 ms) - 2019-10-9 17:15:13[DEBUG](AbstractBeanFactory.java:243)
    Returning cached instance of singleton bean
    'org.springframework.transaction.interceptor.TransactionInterceptor#0'
20 - (1066 ms) - 2019-10-9 17:15:13[DEBUG]
    (AbstractFallbackTransactionAttributeSource.java:107) Adding transactional
    method 'UserServiceImpl.saveUser' with attribute:
    PROPAGATION_REQUIRED,ISOLATION_DEFAULT; ''
21 - (1069 ms) - 2019-10-9 17:15:13[DEBUG](AbstractBeanFactory.java:243)
    Returning cached instance of singleton bean 'transactionManager'
22 - (1076 ms) - 2019-10-9 17:15:13[DEBUG]
    (AbstractPlatformTransactionManager.java:366) Creating new transaction with
    name [com.javaxyz.service.impl.UserServiceImpl.saveUser]:
    PROPAGATION_REQUIRED,ISOLATION_DEFAULT; ''
23 - (1078 ms) - 2019-10-9 17:15:13[DEBUG]
    (DataSourceTransactionManager.java:204) Acquired Connection
    [jdbc:mysql:///p19_java7_mybatis, Username=root@localhost, MySQL-AB JDBC
    Driver] for JDBC transaction
24 - (1087 ms) - 2019-10-9 17:15:13[DEBUG]
    (DataSourceTransactionManager.java:221) Switching JDBC Connection
    [jdbc:mysql:///p19_java7_mybatis, Username=root@localhost, MySQL-AB JDBC
    Driver] to manual commit

```

```

25 - (1093 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    Creating a new SqlSession
26 - (1099 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    Registering transaction synchronization for SqlSession
    [org.apache.ibatis.session.defaults.DefaultSqlSession@1cd1130a]
27 - (1138 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    JDBC Connection [jdbc:mysql:///p19_java7_mybatis, Username=root@localhost,
    MySQL-AB JDBC Driver] will be managed by Spring
28 - (1141 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    ooo Using Connection [jdbc:mysql:///p19_java7_mybatis,
    Username=root@localhost, MySQL-AB JDBC Driver]
29 - (1146 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    ==> Preparing: INSERT INTO smbms_user (userCode, userName, userPassword,
    gender, birthday, phone, address, userRole, createdBy, creationDate) VALUES
    (?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
30 - (1172 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    ==> Parameters: zhangxiulian(String), 张秀连666(String), 123999(String),
    0(Integer), 2010-09-17 00:00:00.0(Timestamp), 13608880888(String), 养育巷
    49(String), 2(Integer), 1(Integer), 2019-10-09 17:15:13.769(Timestamp)
31 - (1175 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    Releasing transactional SqlSession
    [org.apache.ibatis.session.defaults.DefaultSqlSession@1cd1130a]
32 - (1176 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    Fetched SqlSession
    [org.apache.ibatis.session.defaults.DefaultSqlSession@1cd1130a] from
    current transaction
33 - (1177 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    ooo Using Connection [jdbc:mysql:///p19_java7_mybatis,
    Username=root@localhost, MySQL-AB JDBC Driver]
34 - (1178 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    ==> Preparing: INSERT INTO smbms_user (userCode, userName, userPassword,
    gender, birthday, phone, address, userRole, createdBy, creationDate) VALUES
    (?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
35 - (1178 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    ==> Parameters: zhangxiulian(String), 张秀连666(String), 123999(String),
    0(Integer), 2010-09-17 00:00:00.0(Timestamp), 13608880888(String), 养育巷
    49(String), 2(Integer), 1(Integer), 2019-10-09 17:15:13.769(Timestamp)
36 - (1180 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    Releasing transactional SqlSession
    [org.apache.ibatis.session.defaults.DefaultSqlSession@1cd1130a]
37 - (1180 ms) - 2019-10-9 17:15:13[DEBUG]
    (AbstractPlatformTransactionManager.java:753) Initiating transaction commit
38 - (1181 ms) - 2019-10-9 17:15:13[DEBUG]
    (DataSourceTransactionManager.java:267) Committing JDBC transaction on
    Connection [jdbc:mysql:///p19_java7_mybatis, Username=root@localhost,
    MySQL-AB JDBC Driver]
39 - (1188 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    Transaction synchronization committing SqlSession
    [org.apache.ibatis.session.defaults.DefaultSqlSession@1cd1130a]
40 - (1188 ms) - 2019-10-9 17:15:13[DEBUG](JakartaCommonsLoggingImpl.java:46)
    Transaction synchronization closing SqlSession
    [org.apache.ibatis.session.defaults.DefaultSqlSession@1cd1130a]
41 - (1191 ms) - 2019-10-9 17:15:13[DEBUG]
    (DataSourceTransactionManager.java:325) Releasing JDBC Connection
    [jdbc:mysql:///p19_java7_mybatis, Username=root@localhost, MySQL-AB JDBC
    Driver] after transaction
42 - (1191 ms) - 2019-10-9 17:15:13[DEBUG](DataSourceUtils.java:327) Returning
    JDBC Connection to DataSource

```


43 | - (1192 ms) - 2019-10-9 17:15:13[INFO](UserServlet.java:126) 增加成功!