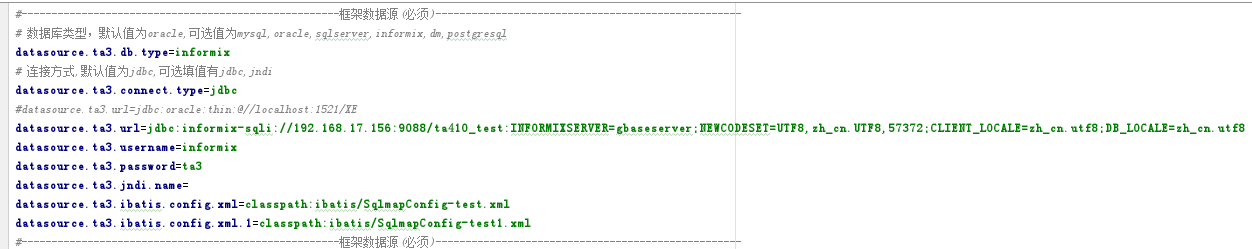
Ta+3CloudV4.1-数据源配置文档

-技术支持：钟志博

## 一.框架数据源配置详解

框架提供\*datasource-dev.properties\*文件进行基础数据源配置，对文件详细解释如下

### A.基础配置



1. 其中\*datasource.ta3.connect.type\*属性很重要，决定项目的数据源连接方式为jdbc还是jndi,框架默认值是jdbc,如果采用默认设置只需要框架数据源如上所示。

2. 其中\*datasource.ta3.db.type\*为数据库类型，框架默认是oracle,可选填值为mysql,oracle,sqlserver,informix,dm,self,特别注意拼写错误。指定数据库类型之后,会自动设置druid驱动,hibernate方言和ibatis方言.如果需要额外的数据库支持,请配置self类型自定义.

3. 其中\*datasource.ta3.url\*为数据库url，注意框架的默认的是oracle的连接url，请填写与数据库适配的url，文档中有常用url的参考格式。

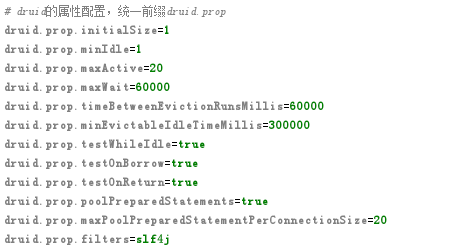
4. 其中\*datasource.ta3.username\*,\*datasource.ta3.password\*数据库用户名和密码。

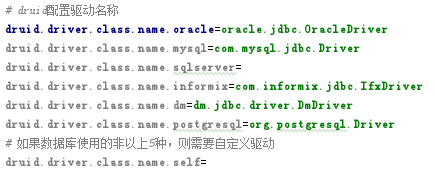
5. 其中\*datasource.ta3.ibatis.config.xml\*配置ibatis的引入位置，框架默认引入位置如如上诉代码所示。

6. 其中\*datasource.ta3.jndi.name\*如果\*connect.type\*为jdbc的时，这个配置可以如上诉代码所示，不填任何内容。

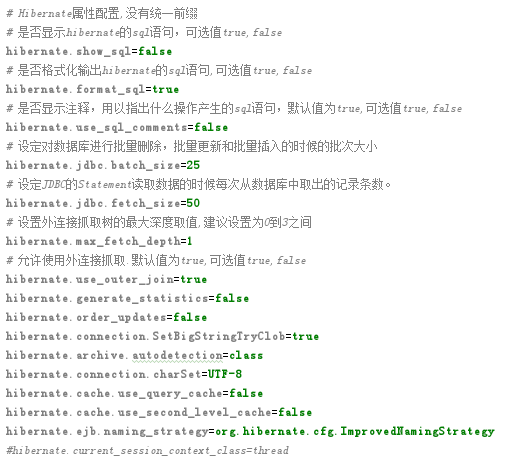
### B.相关默认配置

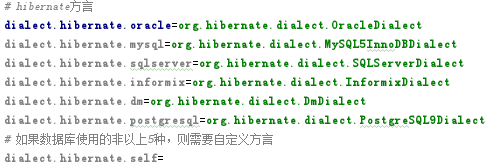
#### 1. druid配置



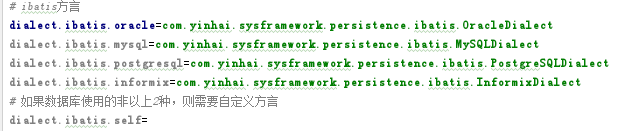


#### 2.hibernate配置

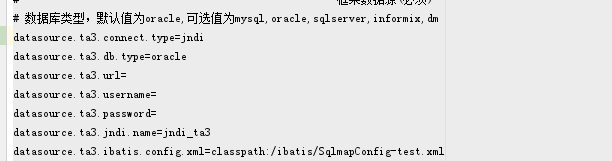




#### 3.ibatis配置



### C.jndi配置示例



### D.数据源使用

#### 1.service配置

<bean id="templateBlo" parent="transactionProxy">  
 <property name="target">  
 <bean class="com.yinhai.modules.template.domain.blo.impl.TemplateBlo">  
 </bean>  
 </property>  
</bean>

注意:事务管理器是transactionProxy

#### 2. dao的获取

import com.yinhai.sysframework.persistence.ibatis.IDao;  
IDao dao = (IDao) ServiceLocator.getService("dao");

## 二.多数据源配置

### A.额外数据源配置

如果项目要支持多数据源配置，其中配置方式，拷贝复制框架数据源的基础配置选项;

将***\*****datasource.ta3.db.type****\****，***\*****datasource.ta3.url****\****等中的ta3替换成要新建数据源的名字，例如我要新建的数据源名称为deepblue，代码如下所示



### B.额外数据源使用

#### 1.service配置

<bean id="templateBlo" parent="deepblueTransactionProxy">  
 <property name="target">  
 <bean class="com.yinhai.modules.template.domain.blo.impl.TemplateBlo">

<property name="dao" ref="deepblueDao" />  
 </bean>  
 </property>  
</bean>

注意:事务管理器为数据源名称+TransactionProxy,例: deepblueTransactionProxy

#### 2. dao的获取

import com.yinhai.sysframework.persistence.ibatis.IDao;  
IDao deepblueDao = (IDao) ServiceLocator.getService("deepblueDao");

注意:dao的名称为数据源名称+Dao,例: deepblueDao

## 三.生产和开发切换

在项目的开发过程中，数据源这块的配置在不不同环境下有不同的配置，框架默认提供了***\*****datasource-dev.properties****\****配置文件来配置开发过程中的配置，如果存在测试环境和生产环境则只需要建立相应的配置文件***\*****datasource-test.properties****\****和\**datasource-product.properties****\****其中文件的配置项和***\*****datasource-dev.properties****\****完全一致，只是改动配置项的值即可。

如何在项目中切换dev,test,product等环境配置，只需要改动***\*****env-config.properties****\****如下代码所示：   
datasource.properties.file.name=datasource-dev.properties

备注:新建的datasource-test.properties和datasource-product.properties文件应该和datasource-dev.properties在同一目录下

## 四.数据库读写分离

需要使用到数据库读写分离的需要在app-context.xml中引入spring-dynamicDataSource.xml和spring-dynamicTransaction.xml文件，  
并分别配置读写库的数据源的配置即可。

## 五.扩展

### durid属性扩展

1. datasource-dev.properties，前缀druid.prop下添加druid属性或者修改druid属性

### hibernate属性扩展

1. datasource-dev.properties，前缀hibernate下添加hibernate属性或者修改hibernate属性

### 驱动扩展

1. 框架提供了5种驱动，分别为“oracle”，“mysql”，“dm”，“sqlserver”，“informix”，如果项目上所使用的驱动不是这5种，则需要配置druid.driver.class.name.self属性，表示自定义驱动

### hibernate方言扩展

1. 框架提供了5种hibernate方言，分别为“oracle”，“mysql”，“dm”，“sqlserver”，“informix”，如果项目上所使用的方言不是这5种，则需要配置dialect.hibernate.self属性，表示自定义方言

### ibatis方言扩展

1. 框架只提供了2种ibatis方言，分别为“oracle”，“mysql”，如果项目上所使用的方言不是这2种，则需要配置dialect.ibatis.self属性，并自己实现com.yinhai.sysframework.persistence.ibatis.Dialect接口。