

## Kiigo-dao 持久层封装使用说明

文档版本	版本说明
V.0.1	初始化版本

### 1. 配置

Kiigo-dao

使用 `c3p0` 连接池，需要的请自行更换配置

需要在 classPath 配置上

`jdbc.properties`

<code>jdbc.driver=com.mysql.jdbc.Driver</code>	驱动
<code>jdbc.url=jdbc:mysql://127.0.0.1:3306/wchat?characterEncoding=UTF-8&amp;autoReconnect=true&amp;failOverReadOnly=false</code>	连接字符串
<code>jdbc.username=root</code>	用户名
<code>jdbc.password=</code>	密码
<code>jdbc.initialSize=10</code>	连接池最小
<code>jdbc.idleConnectionTestPeriod=900</code>	空闲多久的连接进行测试
<code>jdbc.maxIdleTime=7200</code>	最大空闲时间
<code>jdbc.testConnectionOnCheckin=true</code>	是否测试连接
<code>jdbc.preferredTestQuery=select 1</code>	连接测试语句
<code>jdbc.checkoutTimeout=1000</code>	获取连接时最长多久时间获取连接。
<code>jdbc.maxPoolSize=75</code>	连接池最大连接数
<code>sqlSessionFactory.mapperLocations=classpath*:com/**/*.sql.xml</code>	扫描 mybatis sql 配置文件

### 2. 使用

如果想吧 持久层使用的 `applicationContext` 和 web 应用的 `applicationContext` 合并成同一个时 需要自行继承

`net.kiigo.core.ApplicationManager` 侦听器 并配置在 web.xml 上

```
package com.ccep.wchat.web;

import javax.servlet.ServletContext;
```

```

import net.kiigo.core.ApplicationManager;

import org.springframework.context.ApplicationContext;
import org.springframework.web.context.support.WebApplicationContextUtils;

public class ApplicationManagerImpl extends ApplicationManager{

    @Override
    protected ApplicationContext getSpringApplicationContext(
        ServletContext context) {
        return WebApplicationContextUtils.getWebApplicationContext(context);
    }
}

```

然后可以使用

```
KGApplicationContextManger.getApplicationContext()
```

获取 dao 使用的 spring Application 容器，无论是通过使用 web spring 容器还是 自动启动，此接口都是获取 kiigo-dao 所在的 容器。

持久层使用

```

FreeDao freeDao = DaoFactory.createFreeDao(); //使用默认 SqlSession 的 dao

DaoFactory.createFreeDao(sessionName) //使用 指定 名称 SqlSession 的 dao

```

### 3. 事物

本持久层建议编程事物，支持注解（但是不建议使用）

```

KGTransactionManager.execute(new KGTransactionWork<Object>() {

    @Override
    public Object doWork(TransactionStatus status) throws Exception {
        //在此处写的方法均在同一事物中

        return null;
    }
});

```

### 4. 注意问题

1.现阶段 批量插入仍为实验性功能