

MiniDao 集成 Spring 文档

JEECG 开源社区

www.jeecg.org

2017/08/23

目 录

一、 MiniDao-PE 简介	1
1. MiniDao 与 spring 集成.....	1
(1) 新建 MiniDao 的 spring 配置文件	1
(2) MiniDao 配置文件.....	1
(3) MiniDao 依赖 maven 引入	2
2. Minidao 代码案例讲解	3
(4) 接口定义[EmployeeDao.java]	3
(5) SQL 文件[EmployeeDao_getAllEmployees.sql]	3
(6) minidao 接口定义和 SQL 文件对应目录.....	4
(7) MiniDao 的 spring 配置	4
(8) 测试代码.....	4
二、 技术交流.....	5



一、MiniDao-PE 简介

MiniDao 是一款超级轻量的 JAVA 持久层框架，基于 SpringJdbc + freemarker 实现，具备 Mybatis 一样的 SQL 分离灵活性和标签逻辑。最大优点：可无缝集成 Hibernate 项目，支持事务统一管理，有效解决 Hibernate 项目，实现灵活的 SQL 分离问题。

具有以下特征：

- O/R mapping 不用设置 xml，零配置便于维护
- 不需要了解 JDBC 的知识
- SQL 语句和 java 代码的分离
- 接口和实现分离，不用写持久层代码，用户只需写接口，以及某些接口方法对应的 sql 它会通过 AOP 自动生成实现类
- 支持自动事务处理和手动事务处理
- 支持与 hibernate 轻量级无缝集成
- SQL 支持脚本语言

1. MiniDao 与 spring 集成

(1) 新建 MiniDao 的 spring 配置文件

文件名：spring-minidao.xml（可以自定义），只要让 spring 扫描到这个文件即可。

扫描方法一：

在 web.xml 中的 spring 监听器中扫描规则中包含 spring-minidao.xml。

```
<context-param>

    <param-name>contextConfigLocation</param-name>

    <param-value>classpath:spring-*.xml</param-value>

</context-param>
```

扫描方法二：

在 spring 的配置文件中引入 spring-minidao.xml。

```
<import resource="classpath*:spring-minidao.xml" />
```

(2) MiniDao 配置文件

```
<!-- MiniDao动态代理类 -->

<bean id="miniDaoHandler" class="org.jeecgframework.minidao.factory.MinidaoBeanScannerConfigurer">

    <!-- 是使用什么字母做关键字Map的关键字 默认值origin 即和sql保持一致,lower小写(推荐),upper 大写 -->
```

```
<property name="keyType" value="lower"></property>
<!-- 格式化sql -->
<property name="formatSql" value="false"></property>
<!-- 输出sql -->
<property name="showSql" value="false"></property>
<!-- 数据库类型 -->
<property name="dbType" value="mysql"></property>
<!-- dao地址,配置符合spring方式 -->
<property name="basePackage" value="org.jeecgframework.web.com.jeecg"></property>
<!-- 使用的注解,默认是Minidao,推荐 Repository-->
<property name="annotation" value="org.springframework.stereotype.Repository"></property>

</bean>
```

(3) MiniDao 依赖 maven 引入

```
<dependency>
  <groupId>org.jeecgframework</groupId>
  <artifactId>minidao-pe</artifactId>
  <version>1.6-SNAPSHOT</version>
  <exclusions>
  </exclusions>
</dependency>
```

2. Minidao 代码案例讲解

(4) 接口定义[EmployeeDao.java]

```
@MiniDao
public interface EmployeeDao {

    /**
     * 返回List<Map>类型，全部数据
     * @param employee
     * @return
     */
    @Arguments({ "employee"})
    List<Map<String,Object>> getAll(Employee employee);

    /**
     * 查询返回Java对象
     * @param empno
     * @return
     */
    @Arguments("empno")
    Employee getEmployee(String empno);

    /**
     * 支持多个参数，查看返回Map
     * @param empno
     * @param name
     * @return
     */
    @Arguments({ "empno", "name"})
    Map<String,Object> getMap(String empno, String name);

    Map<String,Object> getMap2(@Param("empno") String empno,@Param("name")String name);

    /**
     * 修改数据
     * @param employee
     * @return
     */
}
```

(5) SQL 文件[EmployeeDao_getAllEmployees.sql]

```
SELECT * FROM employee where 1=1
```

```
<#if employee.age ?exists>
```

```
and age = :employee.age
```

```
</#if>
```

```
<#if employee.name ?exists>
```

```
and name = :employee.name
```

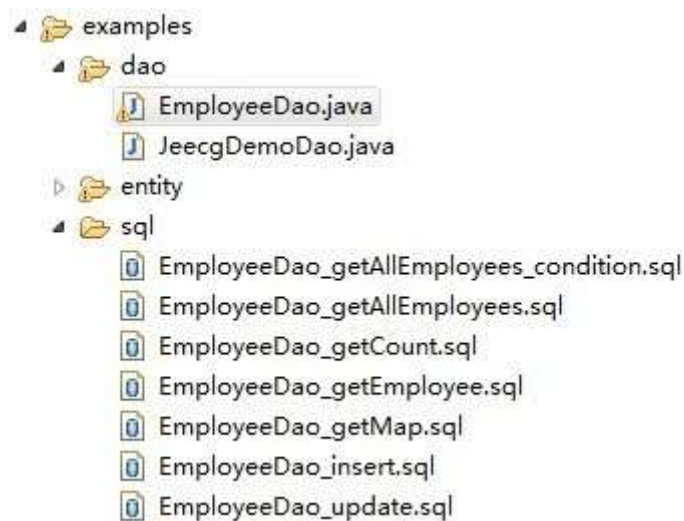
```
</#if>
```

```
<#if employee.empno ?exists>
```

```
and empno = :employee.empno
```

```
</#if>
```

(6) minidao 接口定义和 SQL 文件对应目录



(7) MiniDao 的 spring 配置

```
<!-- 注册 MiniDao 接口 -->

<!-- MiniDao动态代理类 -->
<bean id="miniDaoHandler"
class="org.jeecgframework.minidao.factory.MinidaoBeanScannerConfigurer">
    <!-- 是使用什么字母做关键字Map的关键字 默认值origin 即和sql保持一致,lower小写(推荐),upper 大
    写 -->
    <property name="keyType" value="lower"></property>
    <!-- 格式化sql -->
    <property name="formatSql" value="false"></property>
    <!-- 输出sql -->
    <property name="showSql" value="false"></property>
    <!-- 数据库类型 -->
    <property name="dbType" value="mysql"></property>
    <!-- dao地址,配置符合spring方式 -->
    <property name="basePackage" value="org.jeecgframework.web,com.jeecg"></property>
    <!-- 使用的注解,默认是Minidao,推荐 Repository-->
    <property name="annotation"
value="org.springframework.stereotype.Repository"></property>
</bean>
```

(8) 测试代码

```
public class Client {
```

```
public static void main(String args[]) {  
  
    BeanFactory factory = new ClassPathXmlApplicationContext(  
        "applicationContext.xml");  
  
    EmployeeDao employeeDao = (EmployeeDao) factory.getBean("employeeDao");  
  
    Employee employee = new Employee();  
  
    List<Map> list = employeeDao.getAllEmployees(employee);  
  
    for(Map mp:list){  
        System.out.println(mp.get("id"));  
        System.out.println(mp.get("name"));  
        System.out.println(mp.get("empno"));  
        System.out.println(mp.get("age"));  
        System.out.println(mp.get("birthday"));  
        System.out.println(mp.get("salary"));  
    }  
}  
  
}
```

二、技术交流

- 作者： 张代浩
- 技术论坛： www.jeecg.org
- 邮箱： jeecg@sina.com
- 交流群： 325978980, 143858350
- 源码下载： <http://git.oschina.net/jeecg/minidao>