后勤管理模块

**1 Dao层**

**1.1定义从**GenericDao**继承的接口Ensure**Dao

**public** **interface** EnsureDao **extends** GenericDao<Ensure, Long> {

Ensure findByName (String name);

}

**1.3 DAO层Test类**

代码说明：

EnsureDaoTest 继承基类的GenericDaoTestCase对src中的类 testFindByNo进行测试。

@RunWith(SpringJUnit4ClassRunner.**class**)

@ContextConfiguration(classes = DaoConfig.**class**)

**class** EnsureDaoTest {

@Autowired

EnsureDao ensureDao;

@Test

**public** **void** testFindByName() {

ensureDao.findByName("Name");

}

}

**1.4 DAO层各类注解的含义**

@Entity：标志一个实体，对应数据库中的一张表

@Test：测试方法，在这里可以测试期望异常和超时时间

@Autowired：*spring*可以自动帮你把*bean*里面引用的对象的*setter/getter*方法省略，它会自动帮你*set/get*。表示被修饰的类需要注入对象,spring会扫描所有被@Autowired标注的类,然后根据类型在ioc容器中找到匹配的类注入

**2 Service层：**

**2.1** ScService**的代码说明**

定义从EnsureService继承的接口ScService的代码如下

**public** **interface** EnsureManager **extends** GenericManager<Ensure, Long>

{

/\*\*

\*

\* Check All Ensure

\* Check the Ensure by the ensure's name

\* **@param** name Ensure name

\* **@return**

\*/

List<Ensure> findAll();

Ensure findbyName(String name);

}

**2.2** ScService**Impl的代码说明**

把DAO层实现类注入到service实现类中，具体代码如下

/\*\*

\* ensure managerimpl class

\* EnsureManagerImpl extends GenericManagerImpl and it's GroupManager interface

\* **@param** EnsureDao Dao

\* **@author** lsz

\*

\*/

@Component

**public** **class** EnsureManagerImpl **extends** GenericManagerImpl<Ensure, Long> **implements** EnsureManager {

EnsureDao ensureDao;

@Autowired

**public** **void** setEnsureDao(EnsureDao ensureDao) {

**this**.ensureDao = ensureDao;

**this**.dao = **this**.ensureDao;

}

/\*\*

\* ensure managerimpl findByname method

\* **@return** object for the ensure's name

\* **@author** lsz

\*/

@Override

**public** Ensure findbyName(String name) {

// **TODO** Auto-generated method stub

**return** **null**;

}

}

**Service的Test类**

ScServiceTest继承基类的EnsureServiceTestCase，对src中的类test进行测试。

**public** **class** EnsureEntityGenerator **extends** GenericGenerator {

/\*\*

\* EnsureManager test class

\* Ensure Database mapping test

\* This class can test the availability of EnsureManager

\* **@author** lsz

\*

\*/

@Autowired

EnsureManager ensureManager;

@Test

**public** **void** gen\_ensure() {

**for** (**int** i = 0; i < 10; i++) {

Ensure g = **new** Ensure();

g.setName("ensure\_" + i);

**this**.ensureManager.save(g);

**for** (**int** j = 0; j < 10; j++)

{

Ensure ensure = **new** Ensure();

ensure.setName("ensure\_" + i + "\_" + j);

ensure.setParent(g);

g = **this**.ensureManager.save(ensure) ;

}

}

}

**2.4 service层中相关注解**

@Service用于标注业务层组件，负责注册一个bean到spring上下文中，bean的ID默认为类名称开头字母小写

就是将属性value的值UserlxgService注入到service中

@Test：测试方法，在这里可以测试期望异常和超时时间

@Autowired：spring可以自动帮你把bean里面引用的对象的setter/getter方法省略，它会自动帮你set/get。表示被修饰的类需要注入对象,spring会扫描所有被@Autowired标注的类,然后根据类型在ioc容器中找到匹配的类注入

**3 Web层**

**3.1**实现后勤管理模块

具体代码如下：

@RequestMapping("/ensure/ensure")

**public** **class** EnsureController **extends** GenericController<Ensure, Long, EnsureManager> {

EnsureManager ensureManager;

@RequestMapping(method = RequestMethod.***GET***, value = "/index.html")

**public** String index() {

**return** "ensure/index";

}

@Autowired

**public** **void** setEnsureManager(EnsureManager ensureManager) {

**this**.ensureManager = ensureManager;

**this**.manager = **this**.ensureManager;

}

}

**3.2 WEB层代码中注解的含义**

@RequestMapping是一个用来处理请求地址映射的注解，可用于类或者方法上。用于类上，表示类中的所有响应请求的方法都是以该地址作为父路径。