1. 测试用例分类

* 单元测试；
* 集成测试（接口测试）。

1. 测试用例存放位置

src/test/java主目录下。

1. 测试用例命名

* 包名与被测试代码包名一致；
* 测试用例命名：单元测试用例名称以被测试的类名后加Test结尾，接口测试用例以ITTest结尾。如：UserServiceTest、UserServiceITTest。

1. 测试框架

DBunit、EasyMock、Unitils、Spring-Test

1. 代码示例(详细代码参见：https://192.168.50.210/svn/ProductDevDepartment/pera/pera-demo)

* 单元测试示例：

package com.peraglobal.demo.tree.web;

import java.util.ArrayList;

import java.util.List;

import org.unitils.UnitilsJUnit4;

import org.unitils.easymock.EasyMockUnitils;

import org.unitils.easymock.annotation.Mock;

import org.unitils.inject.annotation.InjectIntoByType;

import org.unitils.inject.annotation.TestedObject;

import org.unitils.reflectionassert.ReflectionAssert;

import org.easymock.EasyMock;

import org.hamcrest.CoreMatchers;

import org.junit.Assert;

import org.junit.Before;

import org.junit.Test;

import com.peraglobal.demo.tree.exception.TreeNodeExistException;

import com.peraglobal.demo.tree.model.TreeNode;

import com.peraglobal.demo.tree.service.TreeService;

import com.peraglobal.demo.tree.web.TreeController;

import com.peraglobal.framework.exception.UserException;

public class TreeControllerTest extends UnitilsJUnit4 {

@TestedObject

private TreeController treeController;

@Mock

@InjectIntoByType

private TreeService treeService;

private List<TreeNode> girdList;

@Before

public void setUp() {

girdList = new ArrayList<TreeNode>();

girdList.add(new TreeNode());

}

@Test

public void testShowWhenLftIsNull(){

UserException exception = new UserException("树节点左值不能为空。", null);

try {

treeController.show(null, 18L);

} catch (Exception e) {

ReflectionAssert.assertLenientEquals(exception, e);

}

}

@Test

public void testShowWhenRgtIsNull(){

UserException exception = new UserException("树节点右值不能为空。", null);

try {

treeController.show(1L, null);

} catch (Exception e) {

ReflectionAssert.assertLenientEquals(exception, e);

}

}

@Test

public void testShow(){

EasyMock.expect(treeService.getFullTree((TreeNode) EasyMock.anyObject())).andReturn(girdList);

EasyMockUnitils.replay();

Assert.assertThat(treeController.show(1L, 18L).getRows().size(), CoreMatchers.is(1));

}

@Test

public void testGetTreeNodeById(){

TreeNode treeNode = new TreeNode();

treeNode.setId(1L);

EasyMock.expect(treeService.getTreeNodeById(EasyMock.anyLong())).andReturn(treeNode);

EasyMockUnitils.replay();

ReflectionAssert.assertLenientEquals(treeNode, treeController.getTreeNodeById(1L));

}

@Test

public void testAddTreeNodeWhenException() throws TreeNodeExistException {

EasyMock.expect(treeService.addTreeNode((TreeNode) EasyMock.anyObject()))

.andThrow(new TreeNodeExistException());

EasyMockUnitils.replay();

Assert.assertNull(treeController.addTreeNode(null));

}

@Test

public void testAddTreeNode() throws TreeNodeExistException{

EasyMock.expect(treeService.addTreeNode((TreeNode) EasyMock.anyObject())).andReturn(1L);

EasyMockUnitils.replay();

ReflectionAssert.assertLenientEquals(1L, treeController.addTreeNode(null));

}

@Test

public void testEditTreeNode(){

treeService.editTreeNode((TreeNode) EasyMock.anyObject());

EasyMock.expectLastCall();

EasyMockUnitils.replay();

treeController.editTreeNode(null);

}

@Test

public void testRemoveTreeNode(){

treeService.removeTreeNode(EasyMock.anyLong());

EasyMock.expectLastCall();

EasyMockUnitils.replay();

treeController.removeTreeNode(null);

}

}

* 接口测试示例：

**package** com.peraglobal.demo.tree.client;

**import** org.junit.Test;

**import** org.unitils.spring.annotation.SpringBeanByType;

**import** com.peraglobal.demo.DemoSpringTestCase;

**import** com.peraglobal.demo.tree.client.RESTClient;

**import** com.peraglobal.demo.tree.model.TreeNode;

**import** com.peraglobal.framework.common.GridData;

**import** com.peraglobal.framework.test.ConcurrentTestUtil;

**import** com.peraglobal.framework.test.LatchWorker;

**public** **class** RESTClientITTest **extends** DemoSpringTestCase {

@SpringBeanByType

**private** RESTClient client;

@Test

**public** **void** testShow() {

**final** TreeNode treeNode = **new** TreeNode();

treeNode.setLft(1L);

treeNode.setRgt(18L);

ConcurrentTestUtil.*test*(**new** LatchWorker() {

@Override

**public** **void** doWork() {

GridData<TreeNode> gridData = client.show(treeNode);

System.*out*.println(gridData.getRows().size());

}

}, 300);

}

@Test

**public** **void** getTreeNodeById(){

TreeNode treeNode = client.getTreeNodeById(1L);

System.*out*.println(treeNode.getName());

}

@Test

**public** **void** addTreeNode(){

TreeNode treeNode = **new** TreeNode();

treeNode.setName("test");

client.addTreeNode(treeNode);

}

}

* DAO测试示例：

package com.peraglobal.demo.tree.dao;

import java.util.List;

import javax.sql.DataSource;

import org.dbunit.DatabaseUnitException;

import org.dbunit.database.IDatabaseConnection;

import org.dbunit.ext.mysql.MySqlConnection;

import org.junit.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.jdbc.CannotGetJdbcConnectionException;

import org.springframework.jdbc.datasource.DataSourceUtils;

import org.springframework.test.context.transaction.TransactionConfiguration;

import org.springframework.transaction.annotation.Transactional;

import com.peraglobal.demo.DaoTestBase;

import com.peraglobal.demo.tree.model.TreeNode;

import com.peraglobal.framework.orm.BaseDao;

@TransactionConfiguration(transactionManager = "txManager", defaultRollback = true)

@Transactional

public class TreeDaoTest extends DaoTestBase{

@Autowired

private DataSource dataSource;

@Autowired

private BaseDao<TreeNode> baseDao;

protected TreeDaoTest() {

super();

}

protected TreeDaoTest(final boolean useTempData) {

super(useTempData);

}

protected TreeDaoTest(final String testDataFile) {

super(testDataFile);

}

@Override

protected IDatabaseConnection createConnection() {

try {

return new MySqlConnection(DataSourceUtils.getConnection(dataSource), "pera");

} catch (final CannotGetJdbcConnectionException e) {

throw new RuntimeException(e);

} catch (final DatabaseUnitException e) {

throw new RuntimeException(e);

}

}

@Test

public void getFullTreeTest(){

TreeNode treeNode = new TreeNode();

treeNode.setLft(1L);

treeNode.setRgt(18L);

List<TreeNode> list = baseDao.queryForList("TreeNode.showFullTree", treeNode);

System.out.println(list);

}

@Test

public void addTreeNodeTest(){

TreeNode treeNode = new TreeNode();

treeNode.setParentId(1L);

treeNode.setName("test");

baseDao.insertEntity("TreeNode.insertTreeNode", treeNode);

}

}