# 实训相关内容

1、ERP(SAP,EBS)

财务审批系统，流程审批

2、改造财务审批系统为微信App

3、微信商城改造

4、简历编写，面试问题

# 第一天

**学习XML解析与应用技术**

## XML解析技术

XML是一种通用的数据交换格式,它的平台无关性、语言无关性、系统无关性、给数据集成与交互带来了极大的方便。XML在不同的语言环境中解析方式都是一样的,只不过实现的语法不同而已。

XML的解析方式分为四种：1、DOM解析；2、SAX解析；3、JDOM解析；4、DOM4J解析。其中前两种属于基础方法，是官方提供的平台无关的解析方式；后两种属于扩展方法，它们是在基础的方法上扩展出来的，只适用于java平台。

一、DOM解析

DOM的全称是Document Object Model，也即文档对象模型。在应用程序中，基于DOM的XML分析器将一个XML文档转换成一个对象模型的集合（通常称DOM树），应用程序正是通过对这个对象模型的操作，来实现对XML文档数据的操作。通过DOM接口，应用程序可以在任何时候访问XML文档中的任何一部分数据，因此，这种利用DOM接口的机制也被称作随机访问机制。

DOM接口提供了一种通过分层对象模型来访问XML文档信息的方式，这些分层对象模型依据XML的文档结构形成了一棵节点树。无论XML文档中所描述的是什么类型的信息，即便是制表数据、项目列表或一个文档，利用DOM所生成的模型都是节点树的形式。也就是说，DOM强制使用树模型来访问XML文档中的信息。由于XML本质上就是一种分层结构，所以这种描述方法是相当有效的。

DOM树所提供的随机访问方式给应用程序的开发带来了很大的灵活性，它可以任意地控制整个XML文档中的内容。然而，由于DOM分析器把整个XML文档转化成DOM树放在了内存中，因此，当文档比较大或者结构比较复杂时，对内存的需求就比较高。而且，对于结构复杂的树的遍历也是一项耗时的操作。所以，DOM分析器对机器性能的要求比较高，实现效率不十分理想。不过，由于DOM分析器所采用的树结构的思想与XML文档的结构相吻合，同时鉴于随机访问所带来的方便，因此，DOM分析器还是有很广泛的使用价值的。

优点：

1、形成了树结构，有助于更好的理解、掌握，且代码容易编写。

2、解析过程中，树结构保存在内存中，方便修改。

缺点：

1、由于文件是一次性读取，所以对内存的耗费比较大。

2、如果XML文件比较大，容易影响解析性能且可能会造成内存溢出。

二：SAX方法

Java原生的XML解析方法之二——SAX方法，原理：通过parse(file,listener)函数用一个listener对xml文件进行查找，按顺序读取文档，遍历每个标签，当发现目标标签

时，读取标签的属性、结点值等信息并返回。

优点：

1、无需将整个xml文档载入内存，因此消耗内存少

2、可以继承ContentHandler创建多个执行不同查询的listener进行解析操作

缺点：

1、不能随机的访问xml中的节点

2、不能修改文档

## Dom4J解析XML

### POM.xml



### XML文件

<?xml version=*"1.0"* encoding=*"UTF8"*?>

<data id=*"myprop"*>

<emp>

<empno>Java</empno>

<address>Oracle</address>

<tel>010</tel>

<fax>010</fax>

<email>java@sun.com</email>

</emp>

<emp>

<empno>Oracle</empno>

<address>Oracle</address>

<tel>020</tel>

<fax>020</fax>

<email>oracle@oracle.com</email>

</emp>

</data>

### 解析代码

**package** com.test.dom4j;

**import** java.io.File;

**import** java.io.InputStream;

**import** java.util.ArrayList;

**import** java.util.Iterator;

**import** java.util.List;

**import** org.dom4j.Document;

**import** org.dom4j.Element;

**import** org.dom4j.io.SAXReader;

**public** **class** XmlParse {

**public** List<EmpInfo> parseInputStream()

{

List<EmpInfo> emps = **new** ArrayList<EmpInfo>();

//通过类加载器从类目录中获取文件流对象

InputStream is = **this**.getClass().getClassLoader().getResourceAsStream("data.xml");

**try**

{

Document doc = **new** SAXReader().read(is);

Element root = doc.getRootElement();

List lst = root.elements("emp");

**for**(Iterator it=lst.iterator();it.hasNext();)

{

Element empEle = (Element)it.next();

Element empnoEle = empEle.element("empno");

String empno = empnoEle.getText();

Element addressEle = empEle.element("address");

String address = addressEle.getText();

Element telEle = empEle.element("tel");

String tel = telEle.getText();

Element faxEle = empEle.element("fax");

String fax = faxEle.getText();

Element emailEle = empEle.element("email");

String email = emailEle.getText();

EmpInfo emi = **new** EmpInfo();

emi.setEmpno(empno);

emi.setAddress(address);

emi.setTel(tel);

emi.setFax(fax);

emi.setEmail(email);

emps.add(emi);

}

}

**catch**(Exception e)

{

e.printStackTrace();

}

**return** emps;

}

**public** List<EmpInfo> parse(String file)

{

List<EmpInfo> emps = **new** ArrayList<EmpInfo>();

**try**

{

//通过文件绝对目录加载文件对象

File f = **new** File(file);

Document doc = **new** SAXReader().read(f);

//获取XML根节点，每个XML中只有一个根节点

Element root = doc.getRootElement();

//获取根节点下emp节点集合列表对象

List lst = root.elements("emp");

//循环遍历集合对象

**for**(Iterator it=lst.iterator();it.hasNext();)

{

//迭代器中每个元素是Dom4J的Element对象类型

Element empEle = (Element)it.next();

//根据节点名称获取节点对象Element

Element empnoEle = empEle.element("empno");

String empno = empnoEle.getText();

Element addressEle = empEle.element("address");

String address = addressEle.getText();

Element telEle = empEle.element("tel");

String tel = telEle.getText();

Element faxEle = empEle.element("fax");

String fax = faxEle.getText();

Element emailEle = empEle.element("email");

String email = emailEle.getText();

//定义Java对象装载XML中的数据

EmpInfo emi = **new** EmpInfo();

emi.setEmpno(empno);

emi.setAddress(address);

emi.setTel(tel);

emi.setFax(fax);

emi.setEmail(email);

//将新生成对象放到集合中

emps.add(emi);

}

}

**catch**(Exception e)

{

e.printStackTrace();

}

**return** emps;

}

**public** **static** **void** main(String[] args) {

XmlParse xp = **new** XmlParse();

String file = "D:/workspace2/EC5\_Proj\_Blank/src/main/resources/data.xml";

List<EmpInfo> emps = xp.parse(file);

**for**(EmpInfo ei:emps)

{

System.***out***.println(ei);

}

System.***out***.println("#######################");

emps = xp.parseInputStream();

**for**(EmpInfo ei:emps)

{

System.***out***.println(ei);

}

}

}

### 生成XML代码

**package** com.test.dom4j;

**import** java.io.ByteArrayOutputStream;

**import** java.io.File;

**import** java.io.FileOutputStream;

**import** java.io.InputStream;

**import** java.util.ArrayList;

**import** java.util.Iterator;

**import** java.util.List;

**import** org.dom4j.CDATA;

**import** org.dom4j.Document;

**import** org.dom4j.DocumentHelper;

**import** org.dom4j.Element;

**import** org.dom4j.io.OutputFormat;

**import** org.dom4j.io.SAXReader;

**import** org.dom4j.io.XMLWriter;

**public** **class** XmlParse {

**public** List<EmpInfo> parseInputStream()

{

List<EmpInfo> emps = **new** ArrayList<EmpInfo>();

//通过类加载器从类目录中获取文件流对象

InputStream is = **this**.getClass().getClassLoader().getResourceAsStream("data.xml");

**try**

{

Document doc = **new** SAXReader().read(is);

Element root = doc.getRootElement();

List lst = root.elements("emp");

**for**(Iterator it=lst.iterator();it.hasNext();)

{

Element empEle = (Element)it.next();

Element empnoEle = empEle.element("empno");

String empno = empnoEle.getText();

Element addressEle = empEle.element("address");

String address = addressEle.getText();

Element telEle = empEle.element("tel");

String tel = telEle.getText();

Element faxEle = empEle.element("fax");

String fax = faxEle.getText();

Element emailEle = empEle.element("email");

String email = emailEle.getText();

EmpInfo emi = **new** EmpInfo();

emi.setEmpno(empno);

emi.setAddress(address);

emi.setTel(tel);

emi.setFax(fax);

emi.setEmail(email);

emps.add(emi);

}

}

**catch**(Exception e)

{

e.printStackTrace();

}

**return** emps;

}

**public** List<EmpInfo> parse(String file)

{

List<EmpInfo> emps = **new** ArrayList<EmpInfo>();

**try**

{

//通过文件绝对目录加载文件对象

File f = **new** File(file);

Document doc = **new** SAXReader().read(f);

//获取XML根节点，每个XML中只有一个根节点

Element root = doc.getRootElement();

//获取根节点下emp节点集合列表对象

List lst = root.elements("emp");

//循环遍历集合对象

**for**(Iterator it=lst.iterator();it.hasNext();)

{

//迭代器中每个元素是Dom4J的Element对象类型

Element empEle = (Element)it.next();

//根据节点名称获取节点对象Element

Element empnoEle = empEle.element("empno");

String empno = empnoEle.getText();

Element addressEle = empEle.element("address");

String address = addressEle.getText();

Element telEle = empEle.element("tel");

String tel = telEle.getText();

Element faxEle = empEle.element("fax");

String fax = faxEle.getText();

Element emailEle = empEle.element("email");

String email = emailEle.getText();

//定义Java对象装载XML中的数据

EmpInfo emi = **new** EmpInfo();

emi.setEmpno(empno);

emi.setAddress(address);

emi.setTel(tel);

emi.setFax(fax);

emi.setEmail(email);

//将新生成对象放到集合中

emps.add(emi);

}

}

**catch**(Exception e)

{

e.printStackTrace();

}

**return** emps;

}

**public** **void** writeXml(List<EmpInfo> emps)

{

**try**

{

Document doc = DocumentHelper.*createDocument*();

Element root = DocumentHelper.*createElement*("data");

doc.setRootElement(root);

**for**(EmpInfo ei:emps)

{

Element empEle = DocumentHelper.*createElement*("emp");

root.add(empEle);

Element empnoEle = DocumentHelper.*createElement*("empno");

empEle.add(empnoEle);

CDATA cdata = DocumentHelper.*createCDATA*(ei.getEmpno());

empnoEle.add(cdata);

Element addrEle = DocumentHelper.*createElement*("address");

empEle.add(addrEle);

addrEle.setText(ei.getAddress());

Element telEle = DocumentHelper.*createElement*("tel");

empEle.add(telEle);

telEle.setText(ei.getTel());

Element faxEle = DocumentHelper.*createElement*("fax");

empEle.add(faxEle);

faxEle.setText(ei.getFax());

Element emailEle = DocumentHelper.*createElement*("email");

empEle.add(emailEle);

emailEle.setText(ei.getEmail());

}

OutputFormat of = **new** OutputFormat("\t",**true**);

FileOutputStream fos = **new** FileOutputStream("d:/my.xml");

XMLWriter xmlw = **new** XMLWriter(fos,of);

xmlw.write(doc);

fos.close();

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

**public** **byte**[] writeXmlBytes(List<EmpInfo> emps)

{

**try**

{

Document doc = DocumentHelper.*createDocument*();

Element root = DocumentHelper.*createElement*("data");

doc.setRootElement(root);

**for**(EmpInfo ei:emps)

{

Element empEle = DocumentHelper.*createElement*("emp");

root.add(empEle);

Element empnoEle = DocumentHelper.*createElement*("empno");

empEle.add(empnoEle);

CDATA cdata = DocumentHelper.*createCDATA*(ei.getEmpno());

empnoEle.add(cdata);

Element addrEle = DocumentHelper.*createElement*("address");

empEle.add(addrEle);

addrEle.setText(ei.getAddress());

Element telEle = DocumentHelper.*createElement*("tel");

empEle.add(telEle);

telEle.setText(ei.getTel());

Element faxEle = DocumentHelper.*createElement*("fax");

empEle.add(faxEle);

faxEle.setText(ei.getFax());

Element emailEle = DocumentHelper.*createElement*("email");

empEle.add(emailEle);

emailEle.setText(ei.getEmail());

}

OutputFormat of = **new** OutputFormat("\t",**true**);

ByteArrayOutputStream baos = **new** ByteArrayOutputStream();

XMLWriter xmlw = **new** XMLWriter(baos,of);

xmlw.write(doc);

**return** baos.toByteArray();

}

**catch**(Exception e)

{

e.printStackTrace();

}

**return** **null**;

}

**public** **static** **void** main(String[] args) {

XmlParse xp = **new** XmlParse();

String file = "D:/workspace2/EC5\_Proj\_Blank/src/main/resources/data.xml";

List<EmpInfo> emps = xp.parse(file);

**for**(EmpInfo ei:emps)

{

System.***out***.println(ei);

}

System.***out***.println("#######################");

emps = xp.parseInputStream();

**for**(EmpInfo ei:emps)

{

System.***out***.println(ei);

}

xp.writeXml(emps);

**byte**[] data = xp.writeXmlBytes(emps);

System.***out***.println(**new** String(data));

}

}

### XML属性解析

package com.test.dom4j;

import java.io.ByteArrayOutputStream;

import java.io.File;

import java.io.FileOutputStream;

import java.io.InputStream;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import org.dom4j.Attribute;

import org.dom4j.CDATA;

import org.dom4j.Document;

import org.dom4j.DocumentHelper;

import org.dom4j.Element;

import org.dom4j.io.OutputFormat;

import org.dom4j.io.SAXReader;

import org.dom4j.io.XMLWriter;

public class XmlAttrParse {

public List<EmpInfo> parseInputStream()

{

List<EmpInfo> emps = new ArrayList<EmpInfo>();

//通过类加载器从类目录中获取文件流对象

InputStream is = this.getClass().getClassLoader().getResourceAsStream("data.xml");

try

{

Document doc = new SAXReader().read(is);

Element root = doc.getRootElement();

List lst = root.elements("emp");

for(Iterator it=lst.iterator();it.hasNext();)

{

Element empEle = (Element)it.next();

Attribute idAtt = empEle.attribute("id");

String id = null;

if(idAtt != null)

{

id = idAtt.getValue();

}

Attribute nameAtt = empEle.attribute("name");

String name = null;

if(nameAtt != null)

{

name = nameAtt.getValue();

}

Element empnoEle = empEle.element("empno");

String empno = empnoEle.getText();

Element addressEle = empEle.element("address");

String address = addressEle.getText();

Element telEle = empEle.element("tel");

String tel = telEle.getText();

Element faxEle = empEle.element("fax");

String fax = faxEle.getText();

Element emailEle = empEle.element("email");

String email = emailEle.getText();

EmpInfo emi = new EmpInfo();

emi.setId(id);

emi.setName(name);

emi.setEmpno(empno);

emi.setAddress(address);

emi.setTel(tel);

emi.setFax(fax);

emi.setEmail(email);

emps.add(emi);

}

}

catch(Exception e)

{

e.printStackTrace();

}

return emps;

}

public List<EmpInfo> parse(String file)

{

List<EmpInfo> emps = new ArrayList<EmpInfo>();

try

{

//通过文件绝对目录加载文件对象

File f = new File(file);

Document doc = new SAXReader().read(f);

//获取XML根节点，每个XML中只有一个根节点

Element root = doc.getRootElement();

//获取根节点下emp节点集合列表对象

List lst = root.elements("emp");

//循环遍历集合对象

for(Iterator it=lst.iterator();it.hasNext();)

{

//迭代器中每个元素是Dom4J的Element对象类型

Element empEle = (Element)it.next();

Attribute idAtt = empEle.attribute("id");

String id = null;

if(idAtt != null)

{

id = idAtt.getValue();

}

Attribute nameAtt = empEle.attribute("name");

String name = null;

if(nameAtt != null)

{

name = nameAtt.getValue();

}

//根据节点名称获取节点对象Element

Element empnoEle = empEle.element("empno");

String empno = empnoEle.getText();

Element addressEle = empEle.element("address");

String address = addressEle.getText();

Element telEle = empEle.element("tel");

String tel = telEle.getText();

Element faxEle = empEle.element("fax");

String fax = faxEle.getText();

Element emailEle = empEle.element("email");

String email = emailEle.getText();

//定义Java对象装载XML中的数据

EmpInfo emi = new EmpInfo();

emi.setId(id);

emi.setName(name);

emi.setEmpno(empno);

emi.setAddress(address);

emi.setTel(tel);

emi.setFax(fax);

emi.setEmail(email);

//将新生成对象放到集合中

emps.add(emi);

}

}

catch(Exception e)

{

e.printStackTrace();

}

return emps;

}

public void writeXml(List<EmpInfo> emps)

{

try

{

Document doc = DocumentHelper.createDocument();

Element root = DocumentHelper.createElement("data");

doc.setRootElement(root);

for(EmpInfo ei:emps)

{

Element empEle = DocumentHelper.createElement("emp");

Attribute idAtt = DocumentHelper.createAttribute(empEle, "id", ei.getId());

empEle.add(idAtt);

Attribute nameAtt = DocumentHelper.createAttribute(empEle, "name", ei.getName());

empEle.add(nameAtt);

root.add(empEle);

Element empnoEle = DocumentHelper.createElement("empno");

empEle.add(empnoEle);

CDATA cdata = DocumentHelper.createCDATA(ei.getEmpno());

empnoEle.add(cdata);

Element addrEle = DocumentHelper.createElement("address");

empEle.add(addrEle);

addrEle.setText(ei.getAddress());

Element telEle = DocumentHelper.createElement("tel");

empEle.add(telEle);

telEle.setText(ei.getTel());

Element faxEle = DocumentHelper.createElement("fax");

empEle.add(faxEle);

faxEle.setText(ei.getFax());

Element emailEle = DocumentHelper.createElement("email");

empEle.add(emailEle);

emailEle.setText(ei.getEmail());

}

OutputFormat of = new OutputFormat("\t",true);

FileOutputStream fos = new FileOutputStream("d:/my.xml");

XMLWriter xmlw = new XMLWriter(fos,of);

xmlw.write(doc);

fos.close();

}

catch(Exception e)

{

e.printStackTrace();

}

}

public byte[] writeXmlBytes(List<EmpInfo> emps)

{

try

{

Document doc = DocumentHelper.createDocument();

Element root = DocumentHelper.createElement("data");

doc.setRootElement(root);

for(EmpInfo ei:emps)

{

Element empEle = DocumentHelper.createElement("emp");

Attribute idAtt = DocumentHelper.createAttribute(empEle, "id", ei.getId());

empEle.add(idAtt);

Attribute nameAtt = DocumentHelper.createAttribute(empEle, "name", ei.getName());

empEle.add(nameAtt);

root.add(empEle);

Element empnoEle = DocumentHelper.createElement("empno");

empEle.add(empnoEle);

CDATA cdata = DocumentHelper.createCDATA(ei.getEmpno());

empnoEle.add(cdata);

Element addrEle = DocumentHelper.createElement("address");

empEle.add(addrEle);

addrEle.setText(ei.getAddress());

Element telEle = DocumentHelper.createElement("tel");

empEle.add(telEle);

telEle.setText(ei.getTel());

Element faxEle = DocumentHelper.createElement("fax");

empEle.add(faxEle);

faxEle.setText(ei.getFax());

Element emailEle = DocumentHelper.createElement("email");

empEle.add(emailEle);

emailEle.setText(ei.getEmail());

}

OutputFormat of = new OutputFormat("\t",true);

ByteArrayOutputStream baos = new ByteArrayOutputStream();

XMLWriter xmlw = new XMLWriter(baos,of);

xmlw.write(doc);

return baos.toByteArray();

}

catch(Exception e)

{

e.printStackTrace();

}

return null;

}

public static void main(String[] args) {

XmlAttrParse xp = new XmlAttrParse();

String file = "D:/workspace2/EC5\_Proj\_Blank/src/main/resources/data.xml";

List<EmpInfo> emps = xp.parse(file);

byte[] data = xp.writeXmlBytes(emps);

System.out.println(new String(data));

}

}

## 使用XML实现数据表迁移数据

### 数据库连接工厂

**package** com.test.dom4j;

**import** java.sql.Connection;

**public** **interface** IDbFactory {

/\*\*

\* 使用JDBC创建数据库连接

\* 使用Mysql数据库，自行下载Mysql数据库驱动连接数据库

\* JDBC Mysql连接配置

\* url=jdbc:mysql://localhost:3306/w1?useUnicode=true&characterEncoding=utf8

\* username=root

\* password=root

\* **@param** url 数据库连接URL

\* **@param** user 数据库用户

\* **@param** pwd 数据库密码

\* **@return**

\*/

**public** Connection getCon(String url,String user,String pwd);

}

### 工厂实现类

**package** com.test.dom4j;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**public** **class** DbFactory **implements** IDbFactory{

**public** Connection getCon(String url, String user, String pwd) {

**try**

{

Class.*forName*("com.mysql.jdbc.Driver");//java.lang.Class

Connection con = DriverManager.*getConnection*(url, user, pwd);

**return** con;

}

**catch**(Exception e)

{

e.printStackTrace();

}

**return** **null**;

}

}

### 数据迁移接口

**package** com.test.dom4j;

**import** java.sql.Connection;

**public** **interface** ITableMagrate {

/\*\*

\* 此方法实现读取数据库表中数据并保存到文件系统，以XML形式存储

\* XML存储格式为,根据节点为table,属性name为表名,每行生成row节点,row节点中每列生成column节点,属性name为列名

\* <table name="">

\* <row>

\* <column name=""></column>

\* <column name=""></column>

\* ...

\* <row>

\* <row>

\* <column name=""></column>

\* <column name=""></column>

\* ...

\* <row>

\* </table>

\* **@param** con 数据库连接

\* **@param** tableName 表名

\* **@param** file 文件绝对目录

\*/

**public** **void** db2xml(Connection con,String tableName,String file);

/\*\*

\* 从XML数据文件中解析数据并保存到数据库表tableName中,数据库表可以预先定义

\* XML存储格式为,根据节点为table,属性name为表名,每行生成row节点,row节点中每列生成column节点,属性name为列名

\* <table name="">

\* <row>

\* <column name=""></column>

\* <column name=""></column>

\* ...

\* <row>

\* <row>

\* <column name=""></column>

\* <column name=""></column>

\* ...

\* <row>

\* </table>

\* **@param** con 数据库连接

\* **@param** tableName 表名

\* **@param** file 文件绝对目录

\*/

**public** **void** xml2db(Connection con,String tableName,String file);

}

### 迁移实现类

根据XML解析动态拼接SQL，针对类不同类型还需要细化业务逻辑，本例只针对字符类型字段处理。

**package** com.test.dom4j;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.ResultSetMetaData;

**import** java.sql.Statement;

**import** java.util.Iterator;

**import** java.util.List;

**import** org.dom4j.Attribute;

**import** org.dom4j.CDATA;

**import** org.dom4j.Document;

**import** org.dom4j.DocumentHelper;

**import** org.dom4j.Element;

**import** org.dom4j.io.OutputFormat;

**import** org.dom4j.io.SAXReader;

**import** org.dom4j.io.XMLWriter;

**public** **class** TableMagrate **implements** ITableMagrate{

@Override

**public** **void** db2xml(Connection con, String tableName, String file) {

**try**

{

Document doc = DocumentHelper.*createDocument*();

Element root = DocumentHelper.*createElement*("table");

Attribute attTab = DocumentHelper.*createAttribute*(root, "name", tableName);

root.add(attTab);

doc.setRootElement(root);

String sql = "select \* from "+tableName;

Statement stat = con.createStatement();

ResultSet rs = stat.executeQuery(sql);

ResultSetMetaData rsmd = rs.getMetaData();

**int** colCnt = rsmd.getColumnCount();

**while**(rs.next())

{

Element rowEle = DocumentHelper.*createElement*("row");

root.add(rowEle);

**for**(**int** i=1;i<=colCnt;i++)

{

String colName = rsmd.getColumnName(i);

String colVal = rs.getString(colName);

**if**(colVal == **null**)

colVal = "";

Element columnEle = DocumentHelper.*createElement*("column");

rowEle.add(columnEle);

Attribute att = DocumentHelper.*createAttribute*(columnEle, "name", colName);

columnEle.add(att);

CDATA cdata = DocumentHelper.*createCDATA*(colVal);

columnEle.add(cdata);

}

}

OutputFormat of = **new** OutputFormat("\t",**true**);

FileOutputStream fos = **new** FileOutputStream(file);

XMLWriter xmlw = **new** XMLWriter(fos,of);

xmlw.write(doc);

fos.close();

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

@Override

**public** **void** xml2db(Connection con, String tableName, String file) {

**try**

{

FileInputStream fis = **new** FileInputStream(file);

SAXReader saxr = **new** SAXReader();

Document doc = saxr.read(fis);

Element root = doc.getRootElement();

Attribute attTab = root.attribute("name");

String tabName = attTab.getValue();

List rows = root.elements("row");

**for**(Iterator it=rows.iterator();it.hasNext();)

{

//insert into table (id,name) values (1,2)

StringBuffer sb = **new** StringBuffer();

sb.append("insert into ");

sb.append(tabName);

sb.append(" (");

StringBuffer sb2 = **new** StringBuffer();

sb2.append(" values (");

Element rowEle = (Element)it.next();

List cols = rowEle.elements("column");

**for**(Iterator it2=cols.iterator();it2.hasNext();)

{

Element colEle = (Element)it2.next();

Attribute colAtt = colEle.attribute("name");

String colName = colAtt.getValue();

**if**("id".equals(colName))

**continue**;

sb.append(colName);

sb.append(",");

String colVal = colEle.getText();

**if**("".equals(colVal))

{

sb2.append("null");

sb2.append(",");

}

**else**

{

sb2.append("'");

sb2.append(colVal);

sb2.append("',");

}

}

String sql1 = sb.toString();

**if**(sql1.endsWith(","))

sql1 = sql1.substring(0,sql1.length()-1);

sql1 = sql1 + ")";

String sql2 = sb2.toString();

**if**(sql2.endsWith(","))

sql2 = sql2.substring(0,sql2.length()-1);

sql2 = sql2 + ")";

String sql = sql1 + sql2;

Statement stat = con.createStatement();

stat.executeUpdate(sql);

stat.close();

}

fis.close();

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

}

### 测试类

**package** com.test.dom4j;

**import** java.sql.Connection;

**public** **class** TestDb {

**public** **static** **void** main(String[] args) {

IDbFactory dbf = **new** DbFactory();

ITableMagrate itm = **new** TableMagrate();

String url = "jdbc:mysql://localhost:3306/w1?useUnicode=true&characterEncoding=utf8";

String user = "root";

String pwd = "root";

Connection con = dbf.getCon(url, user, pwd);

//itm.db2xml(con, "t\_student", "d:\\my.xml");

itm.xml2db(con, "t\_student", "d:\\my.xml");

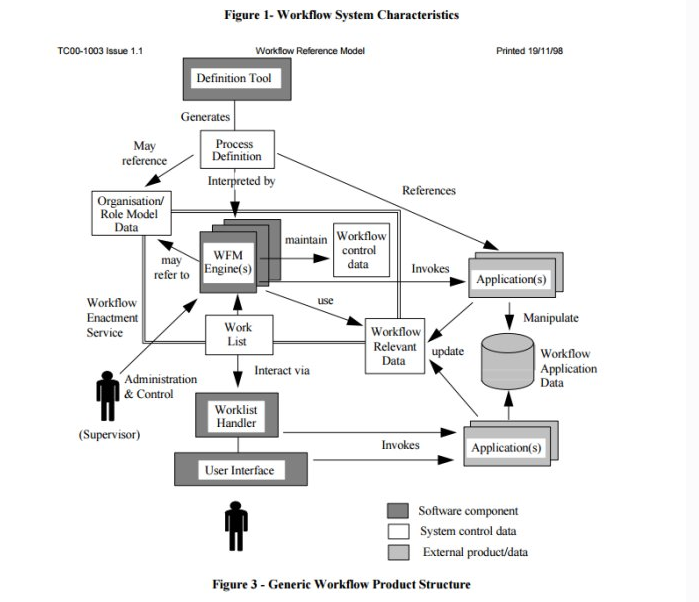
}

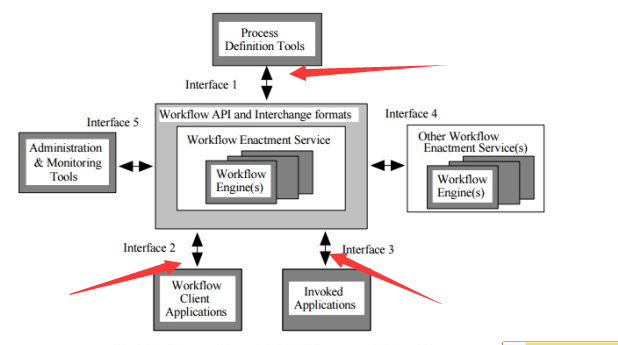
}

# 第二天

**学习工作流技术与应用，掌握主要应用接口**

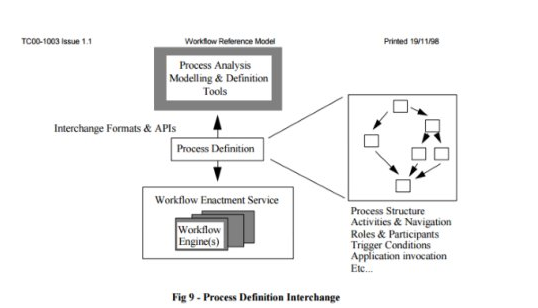
工作流管理系统有很多，企业信息系统都需要工作流技术实现业务流程重组与系统解耦，每个企业化系统都会用到工作流技术，或者自己实现，或者采购成型的产品。国外商业化工作流，最好的是IBM Domino note工作流，还有一些开源工作流技术OFBiz，OBE，Shark等。开发工作流产品都会参照WfMC组织定义的规范。如下是WfMC规范定义的工作流模型和工作流引擎的五个接口





## 编程时调用工作流接口

### 工作流定义工具接口



### 创建工作流实例接口

startWorkflow(processName,HashMap)

## 提交工作流接口

submitWorkfow(instProcessId,instActivityId,instWorkitemId, HashMap)

### 获取某用户待办任务列表

getWorkitemList(userId)

## 工作流相关Jar包



开发人员常用工作流接口

1. 在第一个节点创建流程
2. 在其他节点提交流程
3. 获取某用户待办件列表，办结件列表

@Autowired

**private** StatusFlowWAPI wapi;

@Test

**public** **void** testMethod()

{

**try**

{

String defProcessId = "testwf";//工作流模板名称

String startUserId = "1";//流程实例启动用户

String userId = "1";//操作工作流实例当前用户

String instanceId = "order1";//业务模块ID

String type = "请假单";//流程类型，通常取流程模板名称

String title = "UserA的请假单";//流程实例标题

String nexttask = "account";//下一办理环节

String url = "/holidayapply?id="+instanceId;//业务模块应用链接

HashMap hmap = **new** HashMap();//工作流实例相关数据

hmap.put("days", 5);

String flag = "gf";//区分客户标识，每一个用户分配统一的标识

String testMode = "no";//是否进入测试模式

//wapi.startWorkflow(defProcessId, startUserId, userId, instanceId, type, title, nexttask, url, hmap, flag, testMode);

nexttask = "draft";

String instProcessId = "proc5bfe5bb88e25";//流程实例ID

String instActivityId = "acti5bfea9d88e25";//流程Activity实例ID

String instWorkitemId= "work5bfea9d98e25";//待办件实例ID

List<String> userIdLst = **new** ArrayList<String>();

userIdLst.add("2");//下一办理人

//wapi.submitWorkflow(instProcessId, instActivityId, instWorkitemId, userIdLst, instanceId, type, title, nexttask, url, hmap, flag, testMode);

//获取某用户办结件列表

List lst = wapi.getFinishList("1", testMode);

System.***out***.println("list="+lst);

**for**(Iterator it=lst.iterator();it.hasNext();)

{

DefWorkItem dwi = (DefWorkItem)it.next();

System.***out***.println("instProcessId="+dwi.getInstProcessId());

System.***out***.println("instActivityId="+dwi.getInstActivityId());

System.***out***.println("instWorkitem="+dwi.getId());

System.***out***.println("title="+dwi.getTitle());

System.***out***.println("url="+dwi.getUrl());

}

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

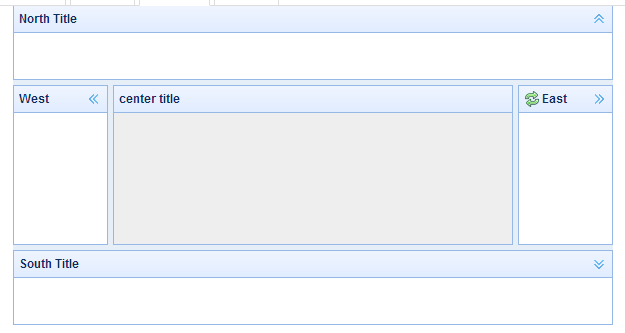
# 第三天

**学习使用前端框架Easyui搭建系统基本页面框架**

## 使用Easyui后台管理框架搭建

参考Easyui Layout控件，绘制主工作页面

<https://blog.csdn.net/qixiang_chen/article/details/83145145>



### 页面原型

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%

String menu = (String)request.getAttribute("menu");

%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<script type=*"text/javascript"* src=*"/easyui/jquery.min.js"*></script>

<script type=*"text/javascript"* src=*"/easyui/jquery.easyui.min.js"*></script>

<script type=*"text/javascript"* src=*"/easyui/easyui-lang-zh\_CN.js"*></script>

<link rel=*"stylesheet"* href=*"/easyui/themes/default/easyui.css"*/>

<link rel=*"stylesheet"* href=*"/easyui/themes/icon.css"*/>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>财务报销系统</title>

<script>

//当点击导航树的节点时，在办公区域显示页面

//此方法检查是否存在打开页面

**var** h = $(window).height() - 125;

**function** addTabCheckExist(title, url,icon){

**if** ($('#tabui').tabs('exists', title)){

$('#tabui').tabs('select', title);

} **else** {

**var** content = '<iframe scrolling="auto" frameborder="0" src="'+url+'" style="width:100%;height:'+h+'px;"></iframe>';

$('#tabui').tabs('add',{

title:title,

content:content,

closable:**true**,

iconCls:icon

});

}

}

</script>

</head>

<body class=*"easyui-layout"*>

<div data-options=*"region:'north'"* style="height:*82px*;">

<!-- 定义面板 -->

<div class=*"easyui-panel"* style="height:*50px*;padding-bottom:*0px*;background:*#B3DFDA*;" data-options=*"border:false"*>

<!-- 显示Logo Div -->

<div style="width:*38%*;float:*left*;margin-left:*2%*">

<img height=*"50px"* src=*"/static/images/logo.png"*/>

</div>

<!-- 显示系统名称 Div -->

<div style="width:*60%*;float:*right*;margin:*0*;padding:*0*">

<p style="font-size:*18pt*;font-weight:*bold*;color:*black*;margin:*0*;padding:*0*">古方红糖创客平台</font></p>

</div>

<!-- 显示登录用户 Div -->

<div id=*"userdiv"* style="position:*absolute*;bottom:*30px*;right:*20px*;">

<font color=*"black"*><b>当前用户:${user.name}</b></font>

</div>

</div>

<!-- 菜单 -->

<%=menu %>

</div>

<div data-options=*"region:'west',title:'自定义导航'"* style="width:*150px*;">

<div class=*"easyui-panel"* style="padding:*5px*;width:*100%*;heigth:*100%*;">

<ul class=*"easyui-tree"*>

<li>

<span>My Documents</span>

<ul>

<li data-options=*"state:'closed'"*>

<span>Photos</span>

<ul>

<li>

<span>Friend</span>

</li>

<li>

<span>Wife</span>

</li>

<li>

<span>Company</span>

</li>

</ul>

</li>

<li>

<span>Program Files</span>

<ul>

<li>Intel</li>

<li>Java</li>

<li>Microsoft Office</li>

<li>Games</li>

</ul>

</li>

<li>index.html</li>

<li>about.html</li>

<li>welcome.html</li>

</ul>

</li>

</ul>

</div>

</div>

<div data-options=*"region:'center',title:'center title'"* style="padding:*5px*;background:*#eee*;">

<div id=*"tabui"* class=*"easyui-tabs"* style="width:*100%*;height:*100%*;margin:*0px*;margin-right:*0px*;">

</div>

</div>

</body>

</html>

### 后台Controller

@RequestMapping("/main")

**public** String main(HttpServletRequest req)

{

String userId = "";

String menu = serv.generateMenuBar(userId);

req.setAttribute("menu", menu);

**return** "main";

}

服务类的方法generateMenuBar负责生成动态菜单控件，菜单是树状数据结构，需要使用递归遍历生成，如下分析Easyui菜单控件代码生成规则。

### 递归生成菜单方法

#### 数据准备：

create table gf\_function(id varchar(40),name varchar(100),

url varchar(100),priority int,icon varchar(50),parentid varchar(40));

insert into gf\_function(id,name,url,priority,icon,parentid)

values('root','功能树','',0,'',null);

insert into gf\_function(id,name,url,priority,icon,parentid)

values('sysadmin','系统管理','',1,'','root');

insert into gf\_function(id,name,url,priority,icon,parentid)

values('corp','协同办公','',2,'','root');

insert into gf\_function(id,name,url,priority,icon,parentid)

values('org','组织管理','',1,'','sysadmin');

insert into gf\_function(id,name,url,priority,icon,parentid)

values('dept','部门管理','/org',0,'','org');

insert into gf\_function(id,name,url,priority,icon,parentid)

values('user','用户管理','/user',1,'','org');

insert into gf\_function(id,name,url,priority,icon,parentid)

values('changepwd','修改密码','',1,'','user');

insert into gf\_function(id,name,url,priority,icon,parentid)

values('adduser','添加用户','/user',2,'','user');

insert into gf\_function(id,name,url,priority,icon,parentid)

values('permission','权限分配','',2,'','sysadmin');

参考Easyui菜单控件规范

<div class="easyui-panel" style="padding:5px;">

<a href="#" class="easyui-linkbutton" data-options="plain:true">Home</a>

<a href="#" class="easyui-menubutton" data-options="menu:'#mm1',iconCls:'icon-edit'">Edit</a>

<a href="#" class="easyui-menubutton" data-options="menu:'#mm2',iconCls:'icon-help'">Help</a>

<a href="#" class="easyui-menubutton" data-options="menu:'#mm3'">About</a>

</div>

<div id="mm1" style="width:150px;">

<div data-options="iconCls:'icon-undo'">

<span>菜单1</span>

<div>

<div>Address</div>

<div>Link</div>

<div>Navigation Toolbar</div>

<div>Bookmark Toolbar</div>

<div class="menu-sep"></div>

<div>

<span>菜单11</span>

<div>

<div>Address</div>

<div>Link</div>

<div>Navigation Toolbar</div>

<div>Bookmark Toolbar</div>

<div class="menu-sep"></div>

<div>

</div>

</div>

</div>

</div>

</div>

<div data-options="iconCls:'icon-redo'">Redo</div>

<div class="menu-sep"></div>

<div>Cut</div>

<div>Copy</div>

<div>Paste</div>

<div class="menu-sep"></div>

<div>

<span>Toolbar</span>

<div>

<div>Address</div>

<div>Link</div>

<div>Navigation Toolbar</div>

<div>Bookmark Toolbar</div>

<div class="menu-sep"></div>

<div>New Toolbar...</div>

</div>

</div>

<div data-options="iconCls:'icon-remove'">Delete</div>

<div>Select All</div>

</div>

<div id="mm2" style="width:100px;">

<div>Help</div>

<div>Update</div>

<div>About</div>

</div>

<div id="mm3" class="menu-content" style="background:#f0f0f0;padding:10px;text-align:left">

<img src="http://www.jeasyui.com/images/logo1.png" style="width:150px;height:50px">

<p style="font-size:14px;color:#444;">Try jQuery EasyUI to build your modern, interactive, javascript applications.</p>

</div>

#### 第一级菜单无规律，必须使用单独方法生成

/\*\*

<div class="easyui-panel" style="padding:5px;">

<a href="#" class="easyui-linkbutton" data-options="plain:true">Home</a>

<a href="#" class="easyui-menubutton" data-options="menu:'#mm1',iconCls:'icon-edit'">Edit</a>

<a href="#" class="easyui-menubutton" data-options="menu:'#mm2',iconCls:'icon-help'">Help</a>

<a href="#" class="easyui-menubutton" data-options="menu:'#mm3'">About</a>

</div>

\*/

@Override

**public** String generateMenuBar(String userId) {

FunctionInfo root = getRoot();

**if**(root != **null**)

{

List<FunctionInfo> child = getMenu(root.getId());

StringBuffer sb = **new** StringBuffer();

sb.append("<div class=\"easyui-panel\" style=\"height:30px;padding-bottom:0px;background:#B3DFDA;\">\r\n");

StringBuffer sb2 = **new** StringBuffer();

**for**(FunctionInfo fi:child)

{

String id = fi.getId();

String icon = fi.getIcon();

String name = fi.getName();

sb.append("<a class=\"easyui-menubutton\" data-options=\"menu:'#"+id+"',iconCls:'"+icon+"'\">"+name+"</a>\r\n");

sb2.append("<div id=\""+id+"\" style=\"width:150px;\">\r\n");

recurseFunction(userId,id,sb2);

sb2.append("</div>\r\n");

}

sb.append("</div>\r\n");

String menu = sb.toString()+sb2.toString();

**return** menu;

}

**return** "";

}

#### 从第二级菜单存在规律，可以使用递归生成

/\*\*

<div id="sysadmin" style="width:150px;">

<div>

<span>组织管理</span>

<div>

<div>

<span>用户管理</span>

<div>

<div>修改密码</div>

</div>

</div>

</div>

</div>

</div>

<div id="corp" style="width:150px;">

</div>

\*/

@Override

**public** **void** recurseFunction(String userId,String id, StringBuffer sb) {

List<FunctionInfo> child = getMenu(id);

**for**(FunctionInfo fi:child)

{

sb.append("<div onclick='javascript:addTabCheckExist(\""+fi.getName()+"\",\""+fi.getUrl()+"\",\""+fi.getIcon()+"\");'>\r\n");

List<FunctionInfo> child2 = getMenu(fi.getId());

**if**(child2 != **null** && child2.size()>0)

{

sb.append("<span>"+fi.getName()+"</span>\r\n");

sb.append("<div>\r\n");

recurseFunction(userId,fi.getId(),sb);

sb.append("</div>\r\n");

}

**else**

{

sb.append(fi.getName()+"\r\n");

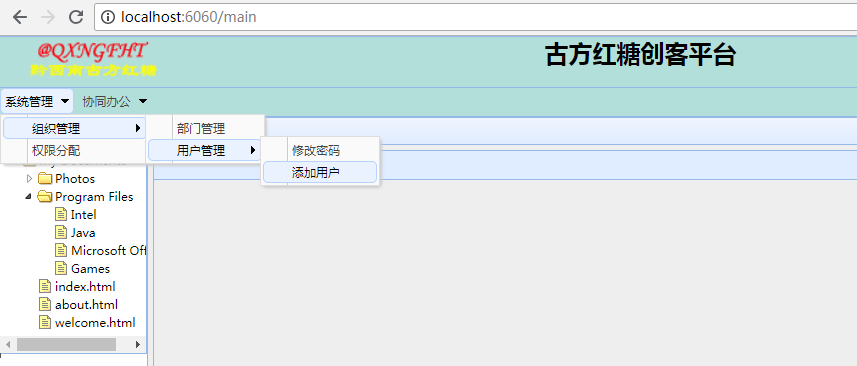
}

sb.append("</div>\r\n");

}

}

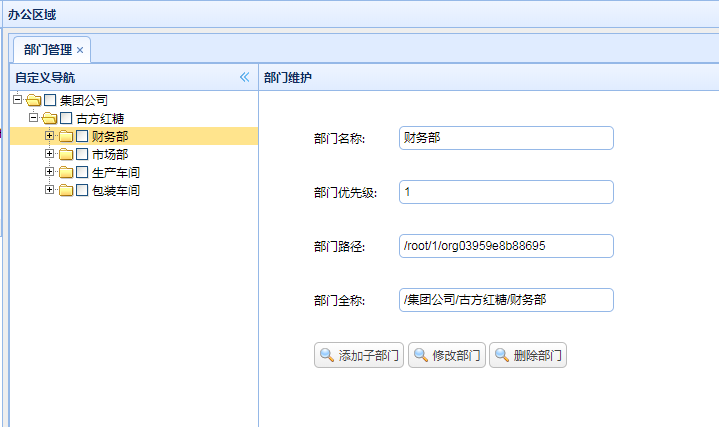
菜单的功能授权后续章节完成，所以在菜单生成方法里传入了用户ID。



# 第四天

**学习企业应用中组织机构模型，完成部门实体管理**

组织机构模块，组织机构模块是信息化系统中核心模块，工作流与授权都与组织机构模块关联，组织机构涉及部门管理，用户管理，角色管理及职位管理等。工作流模块中定义了一套组织机构接口，我们需要实现这一套接口，



## 实现部门接口IOrg

**package** com.gf.statusflow;

**import** java.util.List;

**public** **interface** IOrg

{

**public** **void** setId(String id);

**public** String getId();

**public** **void** setName(String name);

**public** String getName();

**public** **void** setParentId(String parentId);

**public** String getParentId();

**public** **void** setOrgPath(String orgPath);

**public** String getOrgPath();

**public** **void** setFullName(String fullName);

**public** String getFullName();

**public** **void** setSeqno(Integer seqno);

**public** Integer getSeqno();

**public** **void** setFlag(String flag);

**public** String getFlag();

}

## 部门实体类

**package** com.gf.statusflow.def;

**import** com.gf.statusflow.IOrg;

**public** **class** DefaultOrg **implements** IOrg{

**private** String id = **null**;

**private** String name = **null**;

**private** String parentId = **null**;

**private** String orgPath = **null**;// /1/2

**private** String fullName = **null**;// /集团公司/古方红糖厂家/财务部

**private** Integer seqno = **null**;

**private** String flag = "gf";

@Override

**public** **void** setId(String id) {

**this**.id = id;

}

@Override

**public** String getId() {

**return** id;

}

@Override

**public** **void** setName(String name) {

**this**.name = name;

}

@Override

**public** String getName() {

**return** name;

}

@Override

**public** **void** setParentId(String parentId) {

**this**.parentId = parentId;

}

@Override

**public** String getParentId() {

**return** parentId;

}

@Override

**public** **void** setOrgPath(String orgPath) {

**this**.orgPath = orgPath;

}

@Override

**public** String getOrgPath() {

**return** orgPath;

}

@Override

**public** **void** setFullName(String fullName) {

**this**.fullName = fullName;

}

@Override

**public** String getFullName() {

**return** fullName;

}

@Override

**public** **void** setSeqno(Integer seqno) {

**this**.seqno = seqno;

}

@Override

**public** Integer getSeqno() {

**return** seqno;

}

@Override

**public** **void** setFlag(String flag) {

**this**.flag = flag;

}

@Override

**public** String getFlag() {

**return** flag;

}

}

## SQL

create table gf\_org(id varchar(32),

name varchar(100),

parentid varchar(32),

orgpath varchar(500),

fullname varchar(500),

seqno int,

flag varchar(20));

insert into gf\_org(id,name,parentid,orgpath,fullname,seqno,flag)

values('root','集团公司',null,'/root','/集团公司',1,'gf');

insert into gf\_org(id,name,parentid,orgpath,fullname,seqno,flag)

values('1','古方红糖','root','/root/1','/集团公司/古方红糖',1,'gf');

## Mapper层部门相关方法定义

/\*\*

\* 部门相关Mybatis方法

\* **@return**

\*/

**public** List<DefaultOrg> getAllOrg();

**public** DefaultOrg getRootOrg();

**public** **void** saveOrg(DefaultOrg org);

**public** **void** deleteOrg(@Param("id") String id);

**public** **void** updateOrg(DefaultOrg org);

**public** List<DefaultOrg> getOrgList(@Param("id") String id);

**public** DefaultOrg getOrgById(@Param("id") String id);

## 组织机构接口IOrgModel部门相关方法定义

/\*\*

\* 部门相关接口

\* **@return**

\*/

**public** List<DefaultOrg> getAllOrg();

**public** DefaultOrg getRootOrg();

**public** **void** saveOrg(DefaultOrg org);

**public** **void** deleteOrg(String id);

**public** **void** updateOrg(DefaultOrg org);

**public** List<DefaultOrg> getOrgList(String id);

**public** DefaultOrg getOrgById(String orgId);

## 组织机构服务类DefaultOrgModel相关部门方法

@Override

**public** List<DefaultOrg> getAllOrg()

{

**return** mapper.getAllOrg();

}

@Override

**public** DefaultOrg getRootOrg() {

**try**

{

**return** mapper.getRootOrg();

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

**return** **null**;

}

@Override

**public** **void** saveOrg(DefaultOrg org) {

log.debug("saveOrg org="+org.getId()+","+org.getName()+","+org.getParentId());

**try**

{

IOrg parentOrg = **this**.getOrgById(org.getParentId());

log.debug("saveOrg parentOrg="+parentOrg);

String id = UUID.*create*("org");

org.setId(id);

String orgPath = parentOrg.getOrgPath();

orgPath = orgPath + "/"+id;

org.setOrgPath(orgPath);

String fullName = parentOrg.getFullName();

fullName = fullName + "/" + org.getName();

org.setFullName(fullName);

mapper.saveOrg(org);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

}

@Override

**public** **void** updateOrg(DefaultOrg org) {

**try**

{

mapper.updateOrg(org);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

}

@Override

**public** **void** deleteOrg(String id) {

**try**

{

mapper.deleteOrg(id);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

}

@Override

**public** List<DefaultOrg> getOrgList(String id) {

**try**

{

**return** mapper.getOrgList(id);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

**return** **null**;

}

@Override

**public** DefaultOrg getOrgById(String orgId) {

**try**

{

**return** mapper.getOrgById(orgId);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

**return** **null**;

}

## 组织机构Controller

@RequestMapping("/org")

**public** String org()

{

**return** "/orgmodel/org";

}

@RequestMapping("/orgroot")

@ResponseBody

**public** List<TreeNode> orgroot()

{

List<TreeNode> roots = **new** ArrayList<TreeNode>();

IOrg org = orgmodel.getRootOrg();

TreeNode tn = fromOrg(org);

roots.add(tn);

**return** roots;

}

@RequestMapping("/orgchild")

@ResponseBody

**public** List<TreeNode> orgchild(String id)

{

List<TreeNode> roots = **new** ArrayList<TreeNode>();

List<DefaultOrg> orgs = orgmodel.getOrgList(id);

**for**(IOrg org:orgs)

{

roots.add(fromOrg(org));

}

**return** roots;

}

@RequestMapping("/orgsubsave")

@ResponseBody

**public** Boolean orgsubsave(DefaultOrg org)

{

orgmodel.saveOrg(org);

**return** **true**;

}

@RequestMapping("/orgsave")

@ResponseBody

**public** Boolean orgsave(DefaultOrg org)

{

orgmodel.updateOrg(org);

**return** **true**;

}

@RequestMapping("/orgload")

@ResponseBody

**public** List<DefaultOrg> orgload()

{

**return** orgmodel.getAllOrg();

}

@RequestMapping("/orgdelete")

@ResponseBody

**public** String orgdelete(String id)

{

String successRtn = "";

String failureRtn = "";

log.debug("orgdelete id="+id);

String[] dim = id.split(",");

**for**(String orgId:dim)

{

DefaultOrg org = orgmodel.getOrgById(orgId);

log.debug("orgdelete orgId="+orgId);

List<DefaultOrg> child = orgmodel.getOrgList(orgId);

**if**(child != **null** && child.size()>0)

{

failureRtn = failureRtn + org.getName();

}

**else**

{

orgmodel.deleteOrg(orgId);

successRtn = successRtn + org.getName();

}

}

String rtn = "删除成功:"+successRtn+"\r\n";

rtn = rtn + "删除失败:"+failureRtn;

**return** rtn;

}

**public** TreeNode fromOrg(IOrg org)

{

TreeNode tn = **new** TreeNode();

tn.setId(org.getId());

tn.setText(org.getName());

tn.addAttribute("isload", "false");

tn.addAttribute("parentId", org.getParentId());

tn.addAttribute("orgPath", org.getOrgPath());

tn.addAttribute("fullName", org.getFullName());

tn.addAttribute("seqno", org.getSeqno());

tn.addAttribute("flag", org.getFlag());

**return** tn;

}

## 部门JSP页面

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%

String menu = (String)request.getAttribute("menu");

%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<script type=*"text/javascript"* src=*"/easyui/jquery.min.js"*></script>

<script type=*"text/javascript"* src=*"/easyui/jquery.easyui.min.js"*></script>

<script type=*"text/javascript"* src=*"/easyui/easyui-lang-zh\_CN.js"*></script>

<link rel=*"stylesheet"* href=*"/easyui/themes/default/easyui.css"*/>

<link rel=*"stylesheet"* href=*"/easyui/themes/icon.css"*/>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>财务报销系统</title>

<script>

**function** addsuborg()

{

**var** objs = $('#tree').tree('getChecked');

**if**(objs.length != 1)

{

$.messager.alert('错误','请选中一个父节点');

**return**;

}

**var** id = objs[0].id;

$('#parentId').val(id);

$('#frm').form('submit', {

url:'/orgsubsave',

onSubmit: **function**(){

},

success:**function**(data){

reloadtree();

}

});

}

**function** updatesuborg()

{

$('#frm').form('submit', {

url:'/orgsave',

onSubmit: **function**(){

},

success:**function**(data){

reloadtree();

}

});

}

**function** deleteorg()

{

**var** objs = $('#tree').tree('getChecked');

**if**(objs.length == 0)

{

$.messager.alert('错误','请选中一个需要删除部门');

**return**;

}

**var** id = '';

**for**(**var** i=0;i<objs.length;i++)

{

id = id + objs[i].id + ',';

}

console.log('deleteorg()='+objs.length+','+id);

$.ajax({

url:'/orgdelete?id='+id,

success:**function**(data)

{

$.messager.alert('信息',data);

reloadtree();

}

})

}

**function** reloadtree()

{

//$('#tree').tree('loadData',[{"id":"1","text":"Root","iconCls":null,"url":null,"state":"closed","attributes":{"type":"system","isload":"false","path":"/","parentId":null}}]);

$.get('/orgroot',**function**(data)

{

$('#tree').tree({

data: data

});

}

)

}

$(document).ready(

**function**()

{

reloadtree();

$('#tree').tree({

onBeforeExpand:**function**(node,param)

{

**if**(node.attributes.isload=='false')

{

$.ajaxSettings.async = **false**;

**var** url = '/orgchild?id='+node.id;

$.get(url,**function**(data)

{

$('#tree').tree('append', {

parent: node.target,

data: data

});

}

)

$.ajaxSettings.async = **true**;

node.attributes.isload = 'true';

}

},

onClick: **function**(node){

$('#id').val(node.id);

$('#name').textbox('setValue',node.text);

$('#seqno').textbox('setValue',node.attributes.seqno);

$('#parentId').val(node.attributes.parentId);

$('#orgPath').textbox('setValue',node.attributes.orgPath);

$('#fullName').textbox('setValue',node.attributes.fullName);

}

});

}

);

</script>

</head>

<body class=*"easyui-layout"*>

<div data-options=*"region:'west',title:'自定义导航'"* style="width:*250px*;">

<ul id=*"tree"* class=*"easyui-tree"* data-options=*"lines:true,checkbox:true,cascadeCheck:false"*>

</ul>

</div>

<div data-options=*"region:'center',title:'部门维护'"* style="padding:*5px*;">

<form id=*"frm"* method=*"post"*>

<input type=*"hidden"* id=*"id"* name=*"id"*/>

<input type=*"hidden"* id=*"parentId"* name=*"parentId"*/>

<div style="margin-left:*50px*;margin-top:*30px*">

<input class=*"easyui-textbox"* type=*"text"* id=*"name"* name=*"name"* data-options=*"width:300,label:'部门名称:'"* />

</div>

<div style="margin-left:*50px*;margin-top:*30px*">

<input class=*"easyui-textbox"* type=*"text"* id=*"seqno"* name=*"seqno"* data-options=*"width:300,label:'部门优先级:'"* />

</div>

<div style="margin-left:*50px*;margin-top:*30px*">

<input class=*"easyui-textbox"* type=*"text"* id=*"orgPath"* name=*"orgPath"* data-options=*"readonly:true,width:300,label:'部门路径:'"* />

</div>

<div style="margin-left:*50px*;margin-top:*30px*">

<input class=*"easyui-textbox"* type=*"text"* id=*"fullName"* name=*"fullName"* data-options=*"readonly:true,width:300,label:'部门全称:'"* />

</div>

<div style="margin-left:*50px*;margin-top:*30px*">

<a id=*"btn"* onclick="addsuborg()" class=*"easyui-linkbutton"* data-options=*"iconCls:'icon-search'"*>添加子部门</a>

<a id=*"btn"* onclick="updatesuborg()" href=*"#"* class=*"easyui-linkbutton"* data-options=*"iconCls:'icon-search'"*>修改部门</a>

<a id=*"btn"* onclick="deleteorg()" href=*"#"* class=*"easyui-linkbutton"* data-options=*"iconCls:'icon-search'"*>删除部门</a>

</div>

</form>

</div>

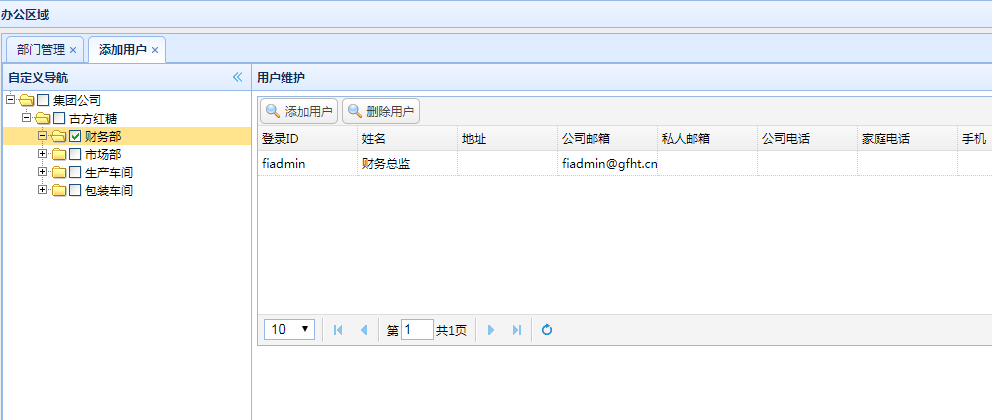
</body>

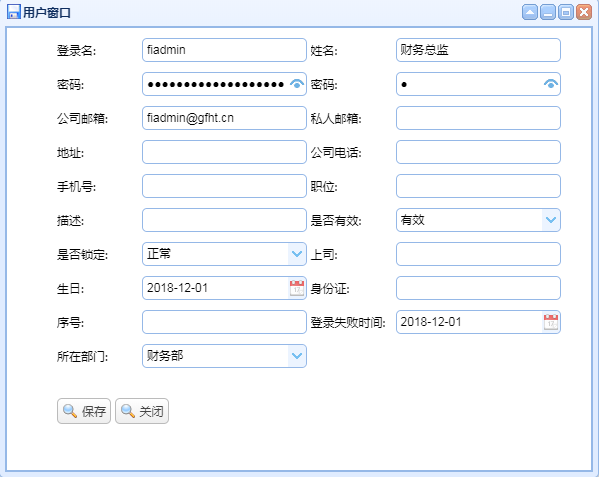
</html>

# 第五天

**学习企业应用中组织机构模型，完成用户实体管理**

**代码下载：**git clone –b day5 <https://github.com/qixiangchen/gufang_fisys.git>





## 实现用户接口IUser

package com.gf.statusflow;

import java.sql.Date;

import java.util.List;

public interface IUser extends java.io.Serializable

{

public void setId(String id);

public String getId();

public void setLoginId(String loginId);

public String getLoginId();

public void setPassword(String password);

public String getPassword();

public void setName(String name);

public String getName();

public void setAddress(String address);

public String getAddress();

public void setCompanyMail(String companyMail);

public String getCompanyMail();

public void setPrivateMail(String privateMail);

public String getPrivateMail();

public void setCompanyTeleNo(String companyTeleNo);

public String getCompanyTeleNo();

public void setHomeTeleNo(String homeTeleNo);

public String getHomeTeleNo();

public void setMobile(String mobile);

public String getMobile();

public void setTitle(String title);

public String getTitle();

public void setDesc(String desc);

public String getDesc();

public void setOrgId(String orgId);

public String getOrgId();

public void setOrgPath(String orgPath);

public String getOrgPath();

public void setEnabled(boolean enabled);

public boolean getEnabled();

public void setLocked(String locked);

public String getLocked();

public void setManagerId(String managerId);

public String getManagerId();

public Date getBirthday();

public void setBirthday(Date birthday);

public String getCardId();

public void setCardId(String cardId);

public Date getFailureDate();

public void setFailureDate(Date failureDate);

public Integer getFailureCount();

public void setFailureCount(Integer failureCount);

public void setOpenid(String openid);

public String getOpenid();

public void setSeqno(Integer seqno);

public Integer getSeqno();

public void setFlag(String flag);

public String getFlag();

public void setRoleList(List<String> roleList);

public List<String> getRoleList();

public void setTestMode(String testMode);

public String getTestMode();

}

## 用户实现类

package com.gf.statusflow.def;

import java.util.ArrayList;

import java.sql.Date;

import java.util.List;

import com.gf.statusflow.IUser;

public class DefaultUser implements IUser{

private String id = null;

private String loginId = null;

private String password = null;

private String name = null;// /1/2

private String address = null;// /集团公司/古方红糖厂家/财务部

private String companyMail = null;

private String privateMail = null;

private String companyTeleNo = null;

private String homeTeleNo = null;

private String mobile = null;

private String title = null;

private String desc = null;

private String orgId = null;

private String orgPath = null;

private boolean enabled = true;

private String locked = null;

private String managerId = null;

private Date birthday = null;

private String cardId = null;

private Date failureDate = null;

private Integer failureCount= null;

private String openid = null;

private Integer seqno = null;

private String flag = "gf";

private String testMode = null;

private List<String> roleList = new ArrayList<String>();

@Override

public String getId() {

return id;

}

@Override

public void setId(String id) {

this.id = id;

}

@Override

public String getLoginId() {

return loginId;

}

@Override

public void setLoginId(String loginId) {

this.loginId = loginId;

}

@Override

public String getPassword() {

return password;

}

@Override

public void setPassword(String password) {

this.password = password;

}

@Override

public String getName() {

return name;

}

@Override

public void setName(String name) {

this.name = name;

}

@Override

public String getAddress() {

return address;

}

@Override

public void setAddress(String address) {

this.address = address;

}

@Override

public String getCompanyMail() {

return companyMail;

}

@Override

public void setCompanyMail(String companyMail) {

this.companyMail = companyMail;

}

@Override

public String getPrivateMail() {

return privateMail;

}

@Override

public void setPrivateMail(String privateMail) {

this.privateMail = privateMail;

}

@Override

public String getCompanyTeleNo() {

return companyTeleNo;

}

@Override

public void setCompanyTeleNo(String companyTeleNo) {

this.companyTeleNo = companyTeleNo;

}

@Override

public String getHomeTeleNo() {

return homeTeleNo;

}

@Override

public void setHomeTeleNo(String homeTeleNo) {

this.homeTeleNo = homeTeleNo;

}

@Override

public String getMobile() {

return mobile;

}

@Override

public void setMobile(String mobile) {

this.mobile = mobile;

}

@Override

public String getTitle() {

return title;

}

@Override

public void setTitle(String title) {

this.title = title;

}

@Override

public String getDesc() {

return desc;

}

@Override

public void setDesc(String desc) {

this.desc = desc;

}

@Override

public String getOrgId() {

return orgId;

}

@Override

public void setOrgId(String orgId) {

this.orgId = orgId;

}

@Override

public String getOrgPath() {

return orgPath;

}

@Override

public void setOrgPath(String orgPath) {

this.orgPath = orgPath;

}

@Override

public boolean getEnabled() {

return enabled;

}

@Override

public void setEnabled(boolean enabled) {

this.enabled = enabled;

}

@Override

public String getLocked() {

return locked;

}

@Override

public void setLocked(String locked) {

this.locked = locked;

}

@Override

public String getManagerId() {

return managerId;

}

@Override

public void setManagerId(String managerId) {

this.managerId = managerId;

}

@Override

public Date getBirthday() {

return birthday;

}

@Override

public void setBirthday(Date birthday) {

this.birthday = birthday;

}

@Override

public String getCardId() {

return cardId;

}

@Override

public void setCardId(String cardId) {

this.cardId = cardId;

}

@Override

public Date getFailureDate() {

return failureDate;

}

@Override

public void setFailureDate(Date failureDate) {

this.failureDate = failureDate;

}

@Override

public Integer getFailureCount() {

return failureCount;

}

@Override

public void setFailureCount(Integer failureCount) {

this.failureCount = failureCount;

}

@Override

public String getOpenid() {

return openid;

}

@Override

public void setOpenid(String openid) {

this.openid = openid;

}

@Override

public Integer getSeqno() {

return seqno;

}

@Override

public void setSeqno(Integer seqno) {

this.seqno = seqno;

}

@Override

public String getFlag() {

return flag;

}

@Override

public void setFlag(String flag) {

this.flag = flag;

}

@Override

public List<String> getRoleList() {

return roleList;

}

@Override

public void setRoleList(List<String> roleList) {

this.roleList = roleList;

}

@Override

public String getTestMode() {

return testMode;

}

@Override

public void setTestMode(String testMode) {

this.testMode = testMode;

}

public String toString()

{

return "DefaultUser[id="+id+",loginId="+loginId+",password="+password+",name="+name+",address="+address+",companyMail="+companyMail

+",privateMail="+privateMail+",companyTeleNo="+companyTeleNo+",homeTeleNo="+homeTeleNo+",mobile="+mobile

+",title="+title+",desc="+desc+",orgId="+orgId+",orgPath="+orgPath+",enabled="+enabled+",locked="+locked

+",managerId="+managerId+",birthday="+birthday+",cardId="+cardId+",failureDate="+failureDate+",failureCount="+failureCount

+",openid="+openid+",seqno="+seqno+",flag="+flag+",testMode="+testMode+"]";

}

}

## SQL

create table gf\_user(id varchar(32),

loginid varchar(100),

password varchar(32),

name varchar(100),

address varchar(200),

companymail varchar(100),

privatemail varchar(100),

companyteleno varchar(100),

hometeleno varchar(100),

mobile varchar(100),

title varchar(32),

description varchar(200),

orgid varchar(32),

orgpath varchar(200),

enabled int,

locked varchar(20),

managerid varchar(32),

birthday date,

cardid varchar(20),

failuredate datetime,

failurecount int,

openid varchar(50),

seqno int,

flag varchar(50),

testmode varchar(10)

);

## Mapper层用户相关方法定义

/\*\*

\* 用户相关Mybatis方法

\* **@param** dwi

\*/

**public** **void** saveUser(DefaultUser user);

**public** **void** updateUser(DefaultUser user);

**public** **void** deleteUser(@Param("id") String id);

**public** List<DefaultUser> getUserByOrgId(@Param("orgId") String orgId);

**public** DefaultUser getManagerById(@Param("userId") String userId);

**public** List<DefaultUser> getUserByManagerId(@Param("managerId") String managerId);

**public** DefaultUser checkLogin(@Param("loginId") String loginId,

@Param("password") String password);

**public** DefaultUser getUserByLoginId(@Param("loginId") String loginId);

## 组织机构接口IOrgModel用户相关方法

/\*\*

\* 用户相关接口

\* **@param** userId

\* **@return**

\*/

**public** **void** saveUser(DefaultUser user);

**public** **void** updateUser(DefaultUser user);

**public** **void** deleteUser(String id);

**public** List<DefaultUser> getUserByOrgId(String orgId);

**public** DefaultUser getManagerById(String userId);

**public** List<DefaultUser> getUserByManagerId(String managerId);

**public** DefaultUser checkLogin(String loginId,String password);

**public** DefaultUser getUserByLoginId(String loginId);

## 组织机构服务类DefaultOrgModel

@Override

**public** **void** saveUser(DefaultUser user) {

**try**

{

mapper.saveUser(user);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

}

@Override

**public** **void** updateUser(DefaultUser user) {

**try**

{

mapper.updateUser(user);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

}

@Override

**public** **void** deleteUser(String id) {

**try**

{

mapper.deleteUser(id);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

}

@Override

**public** List<DefaultUser> getUserByOrgId(String orgId) {

**try**

{

**return** mapper.getUserByOrgId(orgId);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

**return** **null**;

}

@Override

**public** DefaultUser getManagerById(String userId) {

**try**

{

**return** mapper.getManagerById(userId);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

**return** **null**;

}

@Override

**public** List<DefaultUser> getUserByManagerId(String managerId) {

**try**

{

**return** mapper.getUserByManagerId(managerId);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

**return** **null**;

}

@Override

**public** DefaultUser checkLogin(String loginId, String password) {

**try**

{

**return** mapper.checkLogin(loginId,password);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

**return** **null**;

}

**public** DefaultUser getUserByLoginId(String loginId)

{

**try**

{

**return** mapper.getUserByLoginId(loginId);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

**return** **null**;

}

## 组织机构Controller

@RequestMapping("/user")

**public** String user()

{

**return** "/orgmodel/user";

}

@RequestMapping("/userload")

@ResponseBody

**public** Map userload(Integer page,Integer rows,String orgId)

{

**if**(page == **null**)

page = 1;

**if**(rows == **null**)

rows = 10;

Map m = **new** HashMap();

**if**("".equals(Util.*fmtStr*(orgId)))

{

m.put("total", 0);

m.put("rows", **new** ArrayList<DefaultUser>());

**return** m;

}

PageHelper.*startPage*(page,rows);

List<DefaultUser> list = orgmodel.getUserByOrgId(orgId);

PageInfo pi = **new** PageInfo(list);

Long total = pi.getTotal();

List<DefaultUser> users = pi.getList();

m.put("total", total);

m.put("rows", users);

**return** m;

}

@RequestMapping("/usersave")

@ResponseBody

**public** Boolean usersave(DefaultUser duser)

{

log.debug("usersave duser="+duser);

String id = duser.getId();

String pwd = duser.getPassword();

String md5 = Util.*getMD5*(pwd);

log.debug("usersave pwd="+pwd+",md5="+md5);

duser.setPassword(md5);

String loginId = duser.getLoginId();

DefaultUser loginUser = orgmodel.getUserByLoginId(loginId);

log.debug("usersave loginUser="+loginUser);

**if**("".equals(Util.*fmtStr*(id)))

{

**if**(loginUser != **null**)

{

**return** **false**;

}

duser.setId(UUID.*create*("user"));

orgmodel.saveUser(duser);

}

**else**

{

log.debug("usersave id="+id+",loginUser.getId()="+loginUser);

//说明当前用户对应的LoginId存在两个

**if**(loginUser != **null** && !id.equals(loginUser.getId()))

{

**return** **false**;

}

orgmodel.updateUser(duser);

}

**return** **true**;

}

## 用户JSP页面

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%

String menu = (String)request.getAttribute("menu");

%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<script type=*"text/javascript"* src=*"/easyui/jquery.min.js"*></script>

<script type=*"text/javascript"* src=*"/easyui/jquery.easyui.min.js"*></script>

<script type=*"text/javascript"* src=*"/easyui/easyui-lang-zh\_CN.js"*></script>

<link rel=*"stylesheet"* href=*"/easyui/themes/default/easyui.css"*/>

<link rel=*"stylesheet"* href=*"/easyui/themes/icon.css"*/>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>财务报销系统</title>

<script>

**function** addsuborg()

{

**var** objs = $('#tree').tree('getChecked');

**if**(objs.length != 1)

{

$.messager.alert('错误','请选中一个父节点');

**return**;

}

**var** id = objs[0].id;

$('#parentId').val(id);

$('#frm').form('submit', {

url:'/orgsubsave',

onSubmit: **function**(){

},

success:**function**(data){

reloadtree();

}

});

}

**function** updatesuborg()

{

$('#frm').form('submit', {

url:'/orgsave',

onSubmit: **function**(){

},

success:**function**(data){

reloadtree();

}

});

}

**function** deleteorg()

{

**var** objs = $('#tree').tree('getChecked');

**if**(objs.length == 0)

{

$.messager.alert('错误','请选中一个需要删除部门');

**return**;

}

**var** id = '';

**for**(**var** i=0;i<objs.length;i++)

{

id = id + objs[i].id + ',';

}

console.log('deleteorg()='+objs.length+','+id);

$.ajax({

url:'/orgdelete?id='+id,

success:**function**(data)

{

$.messager.alert('信息',data);

reloadtree();

}

})

}

**function** reloadtree()

{

//$('#tree').tree('loadData',[{"id":"1","text":"Root","iconCls":null,"url":null,"state":"closed","attributes":{"type":"system","isload":"false","path":"/","parentId":null}}]);

$.get('/orgroot',**function**(data)

{

$('#tree').tree({

data: data

});

}

)

}

**function** loaddatagrid(orgId)

{

$('#userdg').datagrid({

url:'/userload?orgId='+orgId

})

}

**function** chkpwd()

{

**var** pwd = $('#password').val();

**var** repwd = $('#repwd').val();

**if**(pwd != repwd)

{

$.messager.alert('信息','两次密码不一致');

}

}

**function** adduser()

{

$('#win').window('open');

}

**function** closeuser()

{

$('#win').window('close');

}

**function** saveuser()

{

**var** rtn = $('#frm').form('validate');

**if**(rtn)

{

orgId = $('#orgId').val();

$('#frm').form('submit', {

url:'/usersave',

onSubmit: **function**(){

},

success:**function**(data){

loaddatagrid(orgId);

closeuser();

}

});

}

}

**function** openuser(index,rowObj)

{

$('#win').window('open');

$('#id').val(rowObj.id);

$('#orgPath').val(rowObj.orgPath);

$('#openid').val(rowObj.openid);

$('#flag').val(rowObj.flag);

$('#testMode').val(rowObj.testMode);

$('#loginId').textbox('setValue',rowObj.loginId);

$('#name').textbox('setValue',rowObj.name);

$('#password').passwordbox('setValue',rowObj.password);

$('#companyMail').textbox('setValue',rowObj.companyMail);

$('#privateMail').textbox('setValue',rowObj.privateMail);

$('#address').textbox('setValue',rowObj.address);

$('#companyTeleNo').textbox('setValue',rowObj.companyTeleNo);

$('#mobile').textbox('setValue',rowObj.mobile);

$('#title').textbox('setValue',rowObj.title);

$('#desc').textbox('setValue',rowObj.desc);

$('#enabled').combobox('setValue',rowObj.enabled);

$('#managerId').textbox('setValue',rowObj.managerId);

$('#birthday').datebox('setValue',rowObj.birthday);

$('#cardId').textbox('setValue',rowObj.cardId);

$('#seqno').textbox('setValue',rowObj.seqno);

$('#failureDate').datebox('setValue',rowObj.failureDate);

$('#cardId').textbox('setValue',rowObj.cardId);

$('#orgId').combobox('setValue',rowObj.orgId);

}

$(document).ready(

**function**()

{

reloadtree();

$('#tree').tree({

onBeforeExpand:**function**(node,param)

{

**if**(node.attributes.isload=='false')

{

$.ajaxSettings.async = **false**;

**var** url = '/orgchild?id='+node.id;

$.get(url,**function**(data)

{

$('#tree').tree('append', {

parent: node.target,

data: data

});

}

)

$.ajaxSettings.async = **true**;

node.attributes.isload = 'true';

}

},

onClick: **function**(node){

**var** orgId = node.id;

loaddatagrid(orgId);

}

});

$('#repwd').textbox({

'onChange':chkpwd

});

$('#win').window('close');

$('#userdg').datagrid({'onClickRow':openuser})

}

);

</script>

</head>

<body class=*"easyui-layout"*>

<div data-options=*"region:'west',title:'自定义导航'"* style="width:*250px*;">

<ul id=*"tree"* class=*"easyui-tree"* data-options=*"lines:true,checkbox:true,cascadeCheck:false"*>

</ul>

</div>

<div data-options=*"region:'center',title:'用户维护'"* style="padding:*5px*;">

<table id=*"userdg"* class=*"easyui-datagrid"* style="width:*100%*;height:*250px*"

data-options=*"url:'/userload',fitColumns:true,pagination:true,*

*pageSize:10,pageList:[10,50,100,200],toolbar:'#tb'"*>

<thead>

<tr>

<th data-options=*"field:'id',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'loginId',width:100"*>登录ID</th>

<th data-options=*"field:'name',width:100"*>姓名</th>

<th data-options=*"field:'address',width:100"*>地址</th>

<th data-options=*"field:'companyMail',width:100"*>公司邮箱</th>

<th data-options=*"field:'privateMail',width:100"*>私人邮箱</th>

<th data-options=*"field:'companyTeleNo',width:100"*>公司电话</th>

<th data-options=*"field:'homeTeleNo',width:100"*>家庭电话</th>

<th data-options=*"field:'mobile',width:100"*>手机</th>

<th data-options=*"field:'title',width:100"*>职位</th>

<th data-options=*"field:'desc',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'orgId',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'orgPath',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'enabled',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'locked',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'managerId',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'birthday',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'cardId',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'failureDate',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'failureCount',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'openid',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'seqno',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'flag',width:100,hidden:true"*>ID</th>

<th data-options=*"field:'testMode',width:100,hidden:true"*>ID</th>

</tr>

</thead>

</table>

<div id=*"tb"*>

<a id=*"btn"* onclick="adduser()" class=*"easyui-linkbutton"* data-options=*"iconCls:'icon-search'"*>添加用户</a>

<a id=*"btn"* onclick="deleteuser()" class=*"easyui-linkbutton"* data-options=*"iconCls:'icon-search'"*>删除用户</a>

</div>

<div id=*"win"* class=*"easyui-window"* title=*"用户窗口"* style="width:*600px*;height:*480px*"

data-options=*"iconCls:'icon-save',modal:true"*>

<div class=*"easyui-layout"* data-options=*"fit:true"*>

<div data-options=*"region:'center'"*>

<form id=*"frm"* method=*"post"*>

<input type=*"hidden"* id=*"id"* name=*"id"*/>

<input type=*"hidden"* id=*"orgPath"* name=*"orgPath"*/>

<input type=*"hidden"* id=*"openid"* name=*"openid"*/>

<input type=*"hidden"* id=*"flag"* name=*"flag"*/>

<input type=*"hidden"* id=*"testMode"* name=*"testMode"*/>

<div style="margin-left:*50px*;margin-top:*10px*">

<input class=*"easyui-textbox"* type=*"text"* id=*"loginId"* name=*"loginId"* data-options=*"width:250,label:'登录名:',required:true"* />

<input class=*"easyui-textbox"* type=*"text"* id=*"name"* name=*"name"* data-options=*"width:250,label:'姓名:'"* />

</div>

<div style="margin-left:*50px*;margin-top:*10px*">

<input class=*"easyui-passwordbox"* type=*"text"* id=*"password"* name=*"password"* data-options=*"width:250,label:'密码:'"* />

<input class=*"easyui-passwordbox"* type=*"text"* id=*"repwd"* name=*"repwd"* onclick="chkpwd()" data-options=*"width:250,label:'密码:'"* />

</div>

<div style="margin-left:*50px*;margin-top:*10px*">

<input class=*"easyui-textbox"* type=*"text"* id=*"companyMail"* name=*"companyMail"* data-options=*"width:250,label:'公司邮箱:',required:true,validType:'email'"* />

<input class=*"easyui-textbox"* type=*"text"* id=*"privateMail"* name=*"privateMail"* data-options=*"width:250,label:'私人邮箱:'"* />

</div>

<div style="margin-left:*50px*;margin-top:*10px*">

<input class=*"easyui-textbox"* type=*"text"* id=*"address"* name=*"address"* data-options=*"width:250,label:'地址:'"* />

<input class=*"easyui-textbox"* type=*"text"* id=*"companyTeleNo"* name=*"companyTeleNo"* data-options=*"width:250,label:'公司电话:'"* />

</div>

<div style="margin-left:*50px*;margin-top:*10px*">

<input class=*"easyui-textbox"* type=*"text"* id=*"mobile"* name=*"mobile"* data-options=*"width:250,label:'手机号:'"* />

<input class=*"easyui-textbox"* type=*"text"* id=*"title"* name=*"title"* data-options=*"width:250,label:'职位:'"* />

</div>

<div style="margin-left:*50px*;margin-top:*10px*">

<input class=*"easyui-textbox"* type=*"text"* id=*"desc"* name=*"desc"* data-options=*"width:250,label:'描述:'"* />

<select id=*"enabled"* name=*"enabled"* class=*"easyui-combobox"* data-options=*"width:250,label:'是否有效:'"* >

<option value=*"0"*>有效</option>

<option value=*"1"*>无效</option>

</select>

</div>

<div style="margin-left:*50px*;margin-top:*10px*">

<select id=*"enabled"* name=*"enabled"* class=*"easyui-combobox"* data-options=*"width:250,label:'是否锁定:'"* >

<option value=*"no"*>正常</option>

<option value=*"yes"*>已锁</option>

</select>

<input class=*"easyui-textbox"* type=*"text"* id=*"managerId"* name=*"managerId"* data-options=*"width:250,label:'上司:'"* />

</div>

<div style="margin-left:*50px*;margin-top:*10px*">

<input class=*"easyui-datebox"* type=*"text"* id=*"birthday"* name=*"birthday"* data-options=*"width:250,label:'生日:'"* />

<input class=*"easyui-textbox"* type=*"text"* id=*"cardId"* name=*"cardId"* data-options=*"width:250,label:'身份证:'"* />

</div>

<div style="margin-left:*50px*;margin-top:*10px*">

<input class=*"easyui-textbox"* type=*"text"* id=*"seqno"* name=*"seqno"* data-options=*"width:250,label:'序号:'"* />

<input class=*"easyui-datebox"* type=*"text"* id=*"failureDate"* name=*"failureDate"* data-options=*"width:250,label:'登录失败时间:'"* />

</div>

<div style="margin-left:*50px*;margin-top:*10px*">

<select id=*"orgId"* name=*"orgId"* class=*"easyui-combobox"*

data-options=*"width:250,label:'所在部门:',valueField:'id',textField:'name',url:'/orgload'"* >

</select>

</div>

<div style="margin-left:*50px*;margin-top:*30px*">

<a id=*"btn"* onclick="saveuser()" class=*"easyui-linkbutton"* data-options=*"iconCls:'icon-search'"*>保存</a>

<a id=*"btn"* onclick="closeuser()" href=*"#"* class=*"easyui-linkbutton"* data-options=*"iconCls:'icon-search'"*>关闭</a>

</div>

</form>

</div>

</div>

</div>

</div>

</body>

</html>

# 第六天

**实现角色管理，角色与用户之间多对多分配**

**代码下载：**git clone –b day6 <https://github.com/qixiangchen/gufang_fisys.git>

## 角色接口IRole

**package** com.gf.statusflow;

**public** **interface** IRole

{

**public** **void** setId(String id);

**public** String getId();

**public** **void** setName(String name);

**public** String getName();

**public** **void** setDescription(String description);

**public** String getDescription();

**public** **void** setFlag(String flag);

**public** String getFlag();

}

## 角色实现类DefaultRole

package com.gf.statusflow.def;

import com.gf.statusflow.IOrg;

import com.gf.statusflow.IRole;

public class DefaultRole implements IRole{

private String id = null;

private String name = null;

private String description = null;

private String flag = "gf";

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public String getFlag() {

return flag;

}

public void setFlag(String flag) {

this.flag = flag;

}

}

## 用户角色关联接口

package com.gf.statusflow;

import java.util.Date;

import java.util.List;

public interface IOrgUserRole extends java.io.Serializable

{

public String USER = "user";

public String ORG = "org";

public String getId();

public void setId(String id);

public String getEntityId();

public void setEntityId(String entityId);

public String getEntityType();

public void setEntityType(String entityType);//entityType="user",entityType="org"

public String getRoleId();

public void setRoleId(String roleId);

public String getFlag();

public void setFlag(String flag);

public String getEntityName();

public void setEntityName(String entityName);

public String getRoleName();

public void setRoleName(String roleName);

}

## 用户角色实体类

package com.gf.statusflow.def;

import com.gf.statusflow.IOrg;

import com.gf.statusflow.IOrgUserRole;

public class DefaultOrgUserRole implements IOrgUserRole{

private String id = null;

private String entityId = null;

private String entityType = USER;

private String roleId = null;

private String flag = null;

//冗余字段

private String entityName = null;

private String roleName = null;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getEntityId() {

return entityId;

}

public void setEntityId(String entityId) {

this.entityId = entityId;

}

public String getEntityType() {

return entityType;

}

public void setEntityType(String entityType) {

this.entityType = entityType;

}

public String getRoleId() {

return roleId;

}

public void setRoleId(String roleId) {

this.roleId = roleId;

}

public String getFlag() {

return flag;

}

public void setFlag(String flag) {

this.flag = flag;

}

public String getEntityName() {

return entityName;

}

public void setEntityName(String entityName) {

this.entityName = entityName;

}

public String getRoleName() {

return roleName;

}

public void setRoleName(String roleName) {

this.roleName = roleName;

}

}

## SQL

create table gf\_role(id varchar(32),

name varchar(100),

description varchar(200),

flag varchar(50));

create table gf\_orguserrole(id varchar(32),

entityid varchar(32),

entitytype varchar(10),

roleid varchar(32),

flag varchar(50));

## Mapper层设计

<!-- 角色相关SQL -->

<insert id=*"saveRole"* parameterType=*"com.gf.statusflow.def.DefaultRole"*>

insert into gf\_role(id,name,description,flag)

values(#{id},#{name},#{description},#{flag})

</insert>

<update id=*"updateRole"* parameterType=*"com.gf.statusflow.def.DefaultRole"*>

update gf\_role set name=#{name},description=#{description},flag=#{flag}

where id=#{id}

</update>

<delete id=*"deleteRole"*>

delete from gf\_role where id=#{id}

</delete>

<select id=*"getAllRole"* resultType=*"com.gf.statusflow.def.DefaultRole"*>

select \* from gf\_role

</select>

<select id=*"getRoleById"* resultType=*"com.gf.statusflow.def.DefaultRole"*>

select \* from gf\_role where id=#{id}

</select>

<!-- 部门与角色，用户与角色相关SQL -->

<insert id=*"saveOrgUserRole"* parameterType=*"com.gf.statusflow.def.DefaultOrgUserRole"*>

insert into gf\_orguserrole(id,entityid,entitytype,roleid,flag)

values(#{id},#{entityId},#{entityType},#{roleId},#{flag})

</insert>

<delete id=*"deleteOrgUserRoleByEntityId"*>

delete from gf\_orguserrole where entityid=#{entityId}

</delete>

<select id=*"getOrgUserRoleByEntityId"* resultType=*"com.gf.statusflow.def.DefaultOrgUserRole"*>

select \* from gf\_orguserrole where entityid=#{entityId}

</select>

## 组织机构接口IOrgModel修改

/\*\*

\* 角色相关Mybatis方法

\*/

**public** **void** saveRole(DefaultRole role);

**public** **void** updateRole(DefaultRole role);

**public** **void** deleteRole(String id);

**public** List<DefaultRole> getAllRole();

**public** DefaultRole getRoleById(String id);

/\*\*

\* 部门与角色，用户与角色相关SQ

\*/

**public** **void** saveOrgUserRole(DefaultOrgUserRole our);

**public** **void** deleteOrgUserRoleByEntityId(String entityId);

**public** List<DefaultOrgUserRole> getOrgUserRoleByEntityId(String entityId);

## 组织机构实现类DefaultOrgModel

/\*\*

\* 角色相关Mybatis方法

\*/

@Override

**public** **void** saveRole(DefaultRole role) {

**try**

{

mapper.saveRole(role);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

}

@Override

**public** **void** updateRole(DefaultRole role) {

**try**

{

mapper.updateRole(role);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

}

@Override

**public** **void** deleteRole(String id) {

**try**

{

mapper.deleteRole(id);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

}

@Override

**public** List<DefaultRole> getAllRole() {

**try**

{

**return** mapper.getAllRole();

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

**return** **null**;

}

@Override

**public** DefaultRole getRoleById(String id) {

**try**

{

**return** mapper.getRoleById(id);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

**return** **null**;

}

/\*\*

\* 部门与角色，用户与角色相关SQ

\*/

@Override

**public** **void** saveOrgUserRole(DefaultOrgUserRole our) {

**try**

{

mapper.saveOrgUserRole(our);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

}

@Override

**public** **void** deleteOrgUserRoleByEntityId(String entityId) {

**try**

{

mapper.deleteOrgUserRoleByEntityId(entityId);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

}

@Override

**public** List<DefaultOrgUserRole> getOrgUserRoleByEntityId(String entityId) {

**try**

{

**return** mapper.getOrgUserRoleByEntityId(entityId);

}

**catch**(Exception e)

{

log.error(e.getMessage());

}

**return** **null**;

}

## Controller层设计

@RequestMapping("/userquery")

@ResponseBody

**public** Map userquery(Integer page,Integer rows,String qname)

{

**if**(page == **null**)

page = 1;

**if**(rows == **null**)

rows = 10;

Map m = **new** HashMap();

PageHelper.*startPage*(page,rows);

List<DefaultUser> list = orgmodel.findUserByName(qname);

PageInfo pi = **new** PageInfo(list);

Long total = pi.getTotal();

List<DefaultUser> users = pi.getList();

m.put("total", total);

m.put("rows", users);

**return** m;

}

@RequestMapping("/role")

**public** String role()

{

**return** "/orgmodel/role";

}

@RequestMapping("/roleload")

@ResponseBody

**public** Map roleload(Integer page,Integer rows)

{

**if**(page == **null**)

page = 1;

**if**(rows == **null**)

rows = 10;

Map m = **new** HashMap();

PageHelper.*startPage*(page,rows);

List<DefaultRole> list = orgmodel.getAllRole();

PageInfo pi = **new** PageInfo(list);

Long total = pi.getTotal();

List<DefaultRole> users = pi.getList();

m.put("total", total);

m.put("rows", users);

**return** m;

}

@RequestMapping("/rolesave")

@ResponseBody

**public** Boolean rolesave(DefaultRole role)

{

log.debug("rolesave role="+role);

String id = role.getId();

**if**("".equals(Util.*fmtStr*(id)))

{

id = UUID.*create*("role");

role.setId(id);

orgmodel.saveRole(role);

}

**else**

{

orgmodel.updateRole(role);

}

**return** **true**;

}

@RequestMapping("/roledelete")

@ResponseBody

**public** Boolean roledelete(String[] id)

{

log.debug("roledelete id="+id);

**if**(id != **null**)

{

**for**(String id2:id)

orgmodel.deleteRole(id2);

}

**return** **true**;

}

@RequestMapping("/roleuser")

**public** String orguserrole()

{

**return** "/orgmodel/orguserrole";

}

@RequestMapping("/roleroot")

@ResponseBody

**public** List<TreeNode> roleroot()

{

List<TreeNode> rtn = **new** ArrayList<TreeNode>();

TreeNode root = **new** TreeNode();

root.setId("root");

root.setText("角色树");

rtn.add(root);

List<DefaultRole> roles = orgmodel.getAllRole();

**for**(DefaultRole role:roles)

{

TreeNode tn = **new** TreeNode();

tn.setId(role.getId());

tn.setText(role.getName());

root.addChild(tn);

}

**return** rtn;

}

## 页面orguserrole.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%

String menu = (String)request.getAttribute("menu");

%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<script type=*"text/javascript"* src=*"/easyui/jquery.min.js"*></script>

<script type=*"text/javascript"* src=*"/easyui/jquery.easyui.min.js"*></script>

<script type=*"text/javascript"* src=*"/easyui/easyui-lang-zh\_CN.js"*></script>

<link rel=*"stylesheet"* href=*"/easyui/themes/default/easyui.css"*/>

<link rel=*"stylesheet"* href=*"/easyui/themes/icon.css"*/>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>财务报销系统</title>

<script>

**function** reloadtree()

{

//$('#tree').tree('loadData',[{"id":"1","text":"Root","iconCls":null,"url":null,"state":"closed","attributes":{"type":"system","isload":"false","path":"/","parentId":null}}]);

$.get('/roleroot',**function**(data)

{

$('#tree').tree({

data: data

});

}

)

}

**function** queryuser()

{

**var** qname = $('#qname').val();

$('#userdg').datagrid({

url:'/userquery?qname='+qname

})

}

**function** saveroleuser()

{

**var** nodes = $('#tree').tree('getChecked');

**var** roleid = '';

**for**(**var** i=0;i<nodes.length;i++)

{

**if**('root' != nodes[i].id)

roleid = roleid + nodes[i].id + ',';

}

**var** users = $('#userdg').datagrid('getChecked');

**var** userid = '';

**for**(**var** i=0;i<users.length;i++)

userid = userid + users[i].id;

$.ajax({

url:'/roleassign?roleId='+roleid+"&userId="+userid,

success:**function**(data)

{

queryuser();

}

})

}

**function** openuser(index, row)

{

$('#win').window('open');

**var** userId = row.id;

$('#userroledg').datagrid({

url:'/userrolequery?userId='+userId

})

}

**function** deluserrole()

{

**var** objs = $('#userroledg').datagrid('getChecked');

**var** id = '';

**for**(i=0;i<objs.length;i++)

{

id = id + objs[i].id + ',';

userId = objs[i].userId;

}

$.ajax({

url:'/userroledel?id='+id,

success:**function**(data)

{

$('#userroledg').datagrid({

url:'/userrolequery?userId='+userId

})

}

})

}

$(document).ready(

**function**()

{

reloadtree();

$('#win').window('close');

$('#userdg').datagrid({

'onClickRow':openuser

})

}

);

</script>

</head>

<body class=*"easyui-layout"*>

<div data-options=*"region:'west',title:'角色树'"* style="width:*250px*;">

<ul id=*"tree"* class=*"easyui-tree"* data-options=*"lines:true,checkbox:true,cascadeCheck:false"*>

</ul>

</div>

<div data-options=*"region:'center',title:'用户查询'"* style="padding:*5px*;">

<table id=*"userdg"* class=*"easyui-datagrid"* style="width:*100%*;height:*250px*"

data-options=*"url:'/userquery',fitColumns:true,pagination:true,*

*pageSize:10,pageList:[10,50,100,200],toolbar:'#tb'"*>

<thead>

<tr>

<th data-options=*"field:'id',width:100,checkbox:true"*>ID</th>

<th data-options=*"field:'loginId',width:100"*>登录ID</th>

<th data-options=*"field:'name',width:100"*>姓名</th>

<th data-options=*"field:'mobile',width:100"*>手机</th>

<th data-options=*"field:'roleName',width:100"*>已分配角色</th>

</tr>

</thead>

</table>

<div id=*"tb"*>

<input class=*"easyui-textbox"* type=*"text"* id=*"qname"* name=*"qname"* data-options=*"width:250,label:'输入用户名称:'"* />

<a id=*"btn"* onclick="queryuser()" class=*"easyui-linkbutton"* data-options=*"iconCls:'icon-search'"*>查询</a>

<a id=*"btn"* onclick="saveroleuser()" class=*"easyui-linkbutton"* data-options=*"iconCls:'icon-search'"*>保存</a>

</div>

<div id=*"win"* class=*"easyui-window"* title=*"用户角色授权窗口"* style="width:*600px*;height:*320px*"

data-options=*"iconCls:'icon-save',modal:true"*>

<div class=*"easyui-layout"* data-options=*"fit:true"*>

<table id=*"userroledg"* class=*"easyui-datagrid"* style="width:*100%*;height:*250px*"

data-options=*"url:'/userrolequery',fitColumns:true,pagination:true,*

*pageSize:10,pageList:[10,50,100,200]"*>

<thead>

<tr>

<th data-options=*"field:'id',width:100,checkbox:true"*>ID</th>

<th data-options=*"field:'name',width:100"*>姓名</th>

<th data-options=*"field:'userId',width:100"*>姓名</th>

<th data-options=*"field:'roleName',width:100"*>角色</th>

</tr>

</thead>

</table>

<a id=*"btn"* onclick="deluserrole()" class=*"easyui-linkbutton"* data-options=*"iconCls:'icon-search'"*>删除</a>

<a id=*"btn"* onclick="closeuserrole()" class=*"easyui-linkbutton"* data-options=*"iconCls:'icon-search'"*>关闭</a>

</div>

</div>

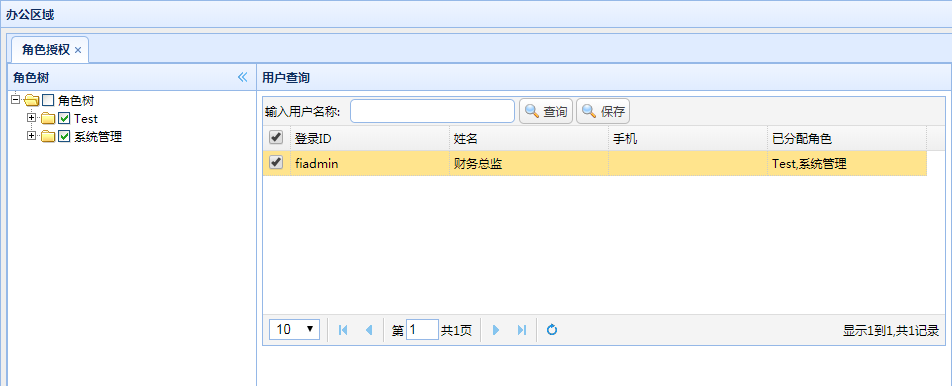
</div>

</body>

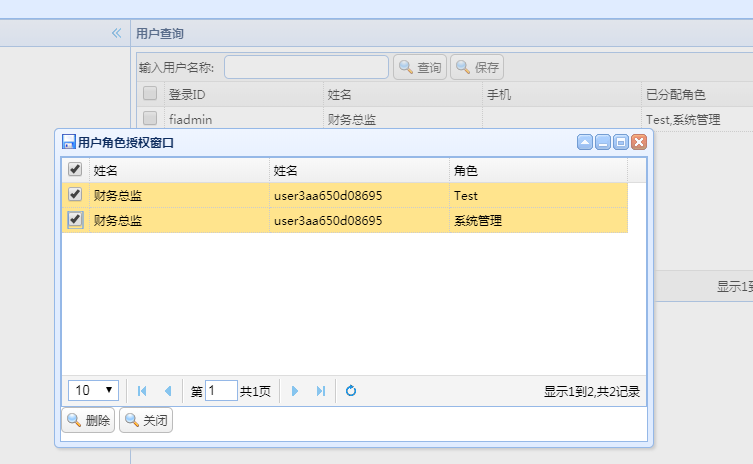
</html>

## 批量授权页面

管理员可以选择多个角色和多个用户，批量进行授权。



删除某用户已分配角色



# 第七天

## SpringMVC日期类型处理

SpringMVC处理日期类型有两种方式，SpingMVC将Form表单中的数据域封装为Controller中的实体对象，我们需要在封装之前增加转换代码，过滤日期类型为空的转换，避免转换错误。

1. 在具体的Controller上添加日期类型处理

@InitBinder

**protected** **void** initBinder(WebDataBinder binder) {

SimpleDateFormat dateFormat = **new** SimpleDateFormat("yyyy-MM-dd");

binder.registerCustomEditor(Date.**class**, **new** CustomDateEditor(dateFormat, **true**));

}

1. 添加全局的日期类型处理

添加类BaseDateAdvice，对所有com包下的Controller都有效。

package com.gf.ctrl;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.springframework.beans.propertyeditors.CustomDateEditor;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.ControllerAdvice;

import org.springframework.web.bind.annotation.InitBinder;

@ControllerAdvice("com")

public class BaseDateAdvice {

@InitBinder

protected void initBinder(WebDataBinder binder) {

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

binder.registerCustomEditor(Date.class, new CustomDateEditor(dateFormat, true));

}

}

## EasyUI树控件

Easyui树控件需要按照Easyui树控件规范生成对应的JSON对象。

Easyui树控件数据格式如下:

1. [{
2. "id":1,
3. "text":"Folder1",
4. "iconCls":"icon-save",
5. "children":[{
6. "text":"File1",
7. "checked":**true**
8. },{
9. "text":"Books",
10. "state":"open",
11. "attributes":{
12. "url":"/demo/book/abc",
13. "price":100
14. },
15. "children":[{
16. "text":"PhotoShop",
17. "checked":**true**
18. },{
19. "id": 8,
20. "text":"Sub Bookds",
21. "state":"closed"
22. }]
23. }]
24. },{
25. "text":"Languages",
26. "state":"closed",
27. "children":[{
28. "text":"Java"
29. },{
30. "text":"C#"
31. }]
32. }]

我们编写Java后台Controller返回的JSON必须按照此格式，可以使用Map承载数据，也可以自己设计类承载数据，自定义属性必须定义在attributes属性中，我们定义装载Tree数据的类是TreeNode

package com.gf.statusflow.def;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

public class TreeNode {

private String id = null;

private String text = null;

private String iconCls = null;

private String url = null;

private String state = "closed";

private Map<String,Object> attributes = new HashMap<String,Object>();

private List<TreeNode> children = new ArrayList<TreeNode>();

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getText() {

return text;

}

public void setText(String text) {

this.text = text;

}

public String getIconCls() {

return iconCls;

}

public void setIconCls(String iconCls) {

this.iconCls = iconCls;

}

public String getUrl() {

return url;

}

public void setUrl(String url) {

this.url = url;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

public Map<String, Object> getAttributes() {

return attributes;

}

public void setAttributes(Map<String, Object> attributes) {

this.attributes = attributes;

}

public void addAttribute(String key,Object value)

{

if(attributes == null)

attributes = new HashMap<String,Object>();

attributes.put(key, value);

}

public List<TreeNode> getChildren() {

return children;

}

public void setChildren(List<TreeNode> children) {

this.children = children;

}

public void addChild(TreeNode tn)

{

children.add(tn);

}

}

加载EasyUI树控件的数据有两种方式，一是使用递归一次遍历表中全部数据，二是使用异步加载数据的方式，第二种方式编程比较简单，执行效率也比较高，我们介绍如何使用第二种方式遍历树。

1. 首先加载树控件的根节点

定义JS方法

**function** reloadtree()

{

$.get('/orgroot',**function**(data)

{

$('#tree').tree({

data: data

});

}

)

}

页面就绪事件

$(document).ready(

**function**()

{

//加载树控件根节点

reloadtree();

//设计树控件事件响应规则，当点击树的某节点时执行这些方法

$('#tree').tree({

onBeforeExpand:**function**(node,param)

{

**if**(node.attributes.isload=='false')

{

$.ajaxSettings.async = **false**;

**var** url = '/orgchild?id='+node.id;

$.get(url,**function**(data)

{

$('#tree').tree('append', {

parent: node.target,

data: data

});

}

)

$.ajaxSettings.async = **true**;

node.attributes.isload = 'true';

}

},

onClick: **function**(node){

$('#id').val(node.id);

$('#name').textbox('setValue',node.text);

$('#seqno').textbox('setValue',node.attributes.seqno);

$('#parentId').val(node.attributes.parentId);

$('#orgPath').textbox('setValue',node.attributes.orgPath);

$('#fullName').textbox('setValue',node.attributes.fullName);

}

});

}

);

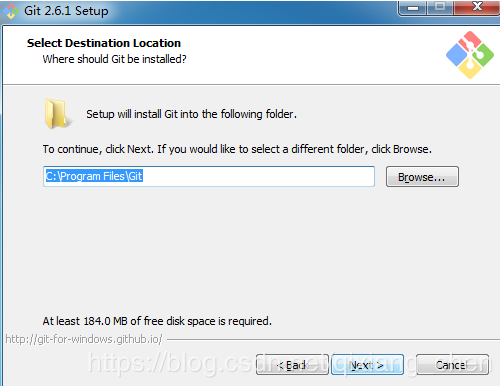
# Git版本控制

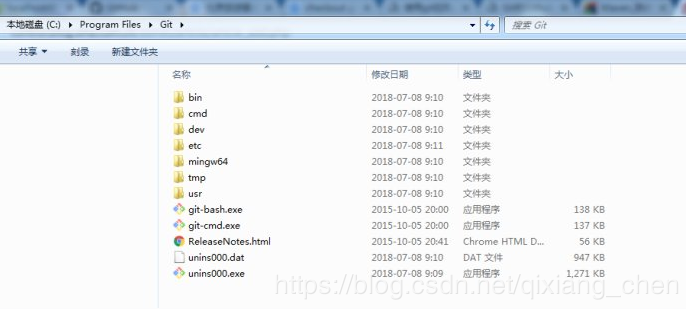
Git是目前世界上最先进的分布式版本控制系统，下面介绍如何使用Git管理项目。

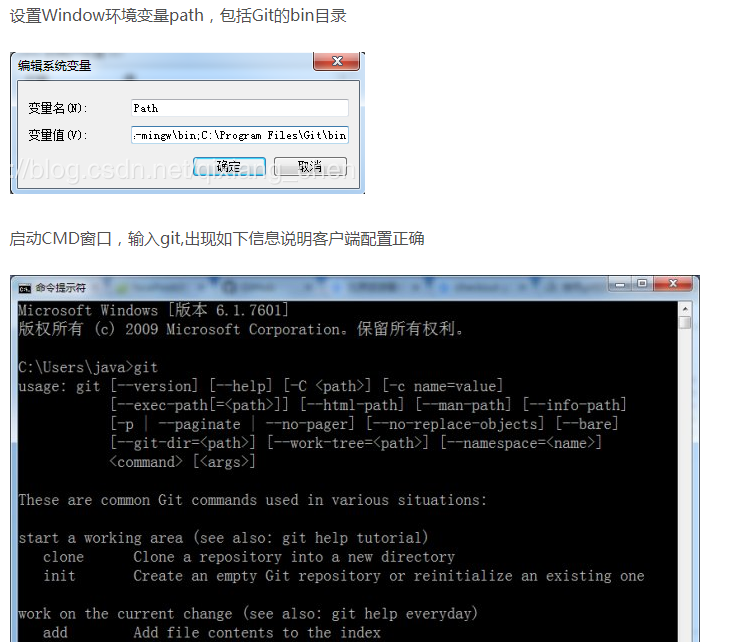
一、下载Window Git工具

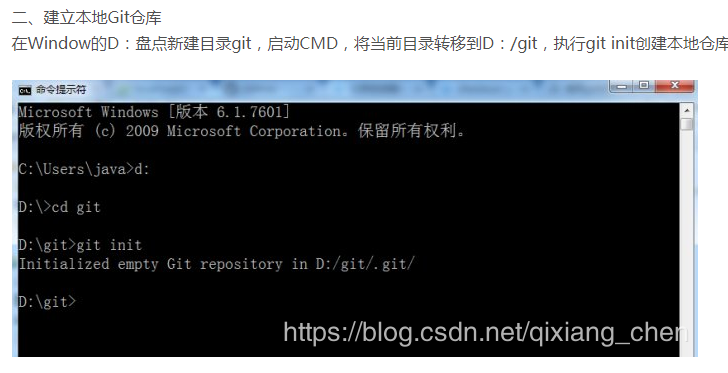
<https://pan.baidu.com/s/1EF_KybTb32VACzXniFBDdw>

下载安装









## 下载Git Master版本

Git clone <https://github.com/qixiangchen/gufang_fisys.git>

## 下载Git 分支版本

Git clone –b day5 <https://github.com/qixiangchen/gufang_fisys.git>

Git clone –b day6 <https://github.com/qixiangchen/gufang_fisys.git>

Git clone –b day7 <https://github.com/qixiangchen/gufang_fisys.git>

## 上传Git版本

参考https://blog.csdn.net/qixiang\_chen/article/details/83474193

# 理论知识

## Spring核心原理