**搭建项目框架**

1.创建项目OASystem(设置全局编码格式为utf-8)

2.创建包和文件夹

com.xxx.dao.impl

com.xxx.entity

com.xxx.filter

com.xxx.service.impl

com.xxx.servlet

com.xxx.util

创建专门用于测试的文件夹

jsp按照模块划分为 system人事, expense报销, inpay收支, duty考勤

3.导入jar包

数据库连接有关 :

c3p0-0.9.2-pre1

c3p0-oracle-thin-extras-0.9.2-pre1

4.加入工具类

(待完善)

5.加入过滤器

(暂时不配置)

6.导入界面原型

7.部署项目并进行测试

**部门管理-添加部门-完成后台代码并进行JUnit测试**

1)创建Department实体类

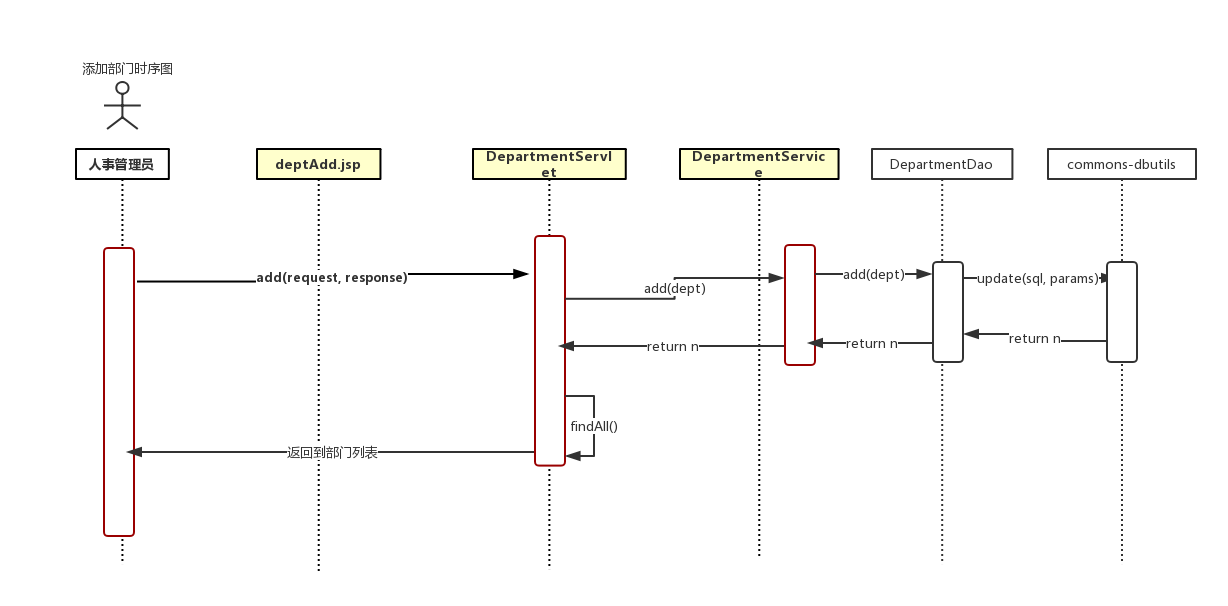
2)创建DepartmentDao接口, 并给出add抽象方法

3)创建DepartmentDaoImpl类, 实现add方法

4)创建DepartmentService接口, 并给出add方法

5)创建DepartmentServiceImpl类,实现add方法

**部门管理-添加部门-完成前台代码并成功测试**



1)创建DepartmentServlet(继承BaseServlet)

    -接收视图层的表单数据

    -调用业务层完成添加操作

    -将结果跳转到不同页面

2)创建一个deptAdd.jsp放在system文件夹下

把前台写好的html代码复制到jsp中, 注意修改好<html>的一段开头和编码设置

3)写表单

4)左边的菜单栏把html页面地址改为jsp的地址

**遇到的问题 :**

1.没有为项目配置资源路径, 从jsp请求servlet的时候, 服务器报500错误, stackover...异常

    配置资源路径等web项目前期框架搭建, 参考<https://www.cnblogs.com/yangyquin/p/5285272.html>

2.搞清楚jsp中绝对路径的第一个 " / "代表的是绝对路径, servlet中如果写重定向的绝对路径 " / "代表的是服务器根目录, 转发中写绝对路径 " / " 代表的是项目根目录

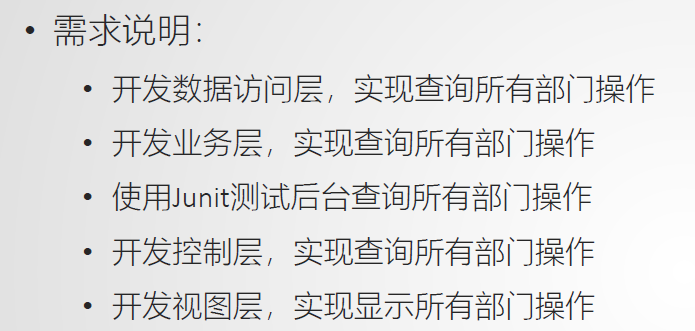
3.c3p0的配置文件要放在资源路径classes文件中, 否则会使用itcast的jar包下的c3p0配置文件

**本功能待完善部分 :**

要给表单添加校验, 如果要三项全部填写了才能提交表单

**部门管理-查询所有的部门**

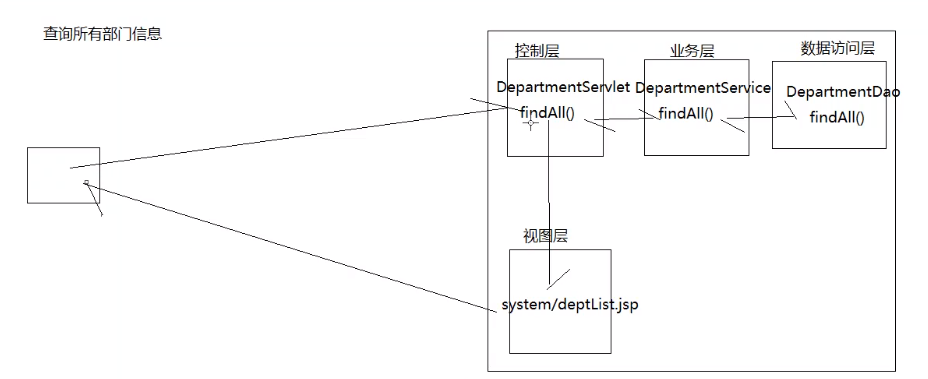
开发总体流程



具体 :

1.把html页面转换成jsp, 创建deptList.jsp

2.左边菜单栏的部门管理连接应该先访问控制层(servlet层)->service->dao 返回后才在视图层显示



3.编写servlet -> service -> dao -> 完成测试

**遇到的问题**

1.idea导入标签库jstl

先导入jstl用到的jar包

在settings中搜索DTD, 按+号添加, url写"<http://java.sun.com/jsp/jstl/core>", 文件选择jstl jar包中的c.tld文件

文件选择到jstl官网下载的jar包中的c.tld文件

参照<https://www.cnblogs.com/junhuawang/p/6953177.html>

**删除部门**

步骤 :

1)删除的时候先到Servlet, 除了要传递method参数, 还要传递部门的编号

2)dao -> service -> 测试 -> servlet

3)注意Servlet最后跳转的时候不能直接跳转到deptList.jsp, 应该跳转到servlet中的findAll方法

4)修改add操作, 添加完应该跳转到findAll方法

5)添加删除时候的确定判断

    改动删除按键的超链接 : 绑定一个javascript方法, 方法要传入参数deptno

var flag = window.confirm("您确定要删除该部门吗?");

if(flag){

    location.href = "请求servlet的地址" + deptno

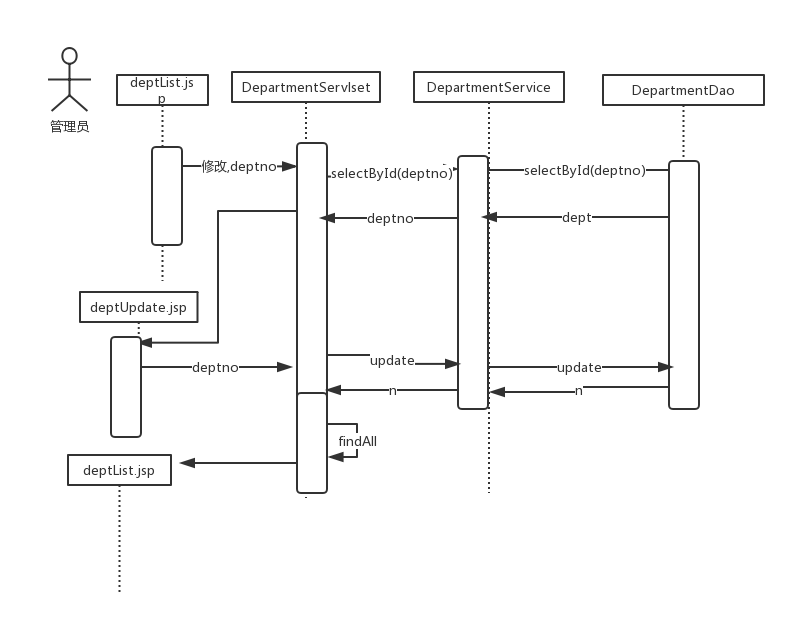
}

原本提供了更美观的删除确认窗口, 但是比较绕, 老师没有讲解,  把class 属性后面的click去掉即可

遗留下的业务逻辑问题 : 如果删除了部门, 该部门的员工怎么办?

**部门管理-更新指定部门信息**

分两步 : 先查询, 把查询的结果显示在更改的页面上, 再进行修改



在修改部门信息时, 要在部门标号里面设置readonly = "readonly";

**部门管理-绘制更新部门时序图**

使用PowerDesigner

工具箱不见了,选择视图palette

**自己完成岗位管理部分**

和部门管理类似

**员工管理-添加员工-创建实体类和搭建框架  --- 一对多关系**

思路 :

1)根据数据库表创建员工类    ---Date用util包下的Date

--问题 : **如何表示员工所属一个部门, 一个岗位, 和他的上下级关系呢?** 你不能只给出一个deptno, 员工的部门信息应该时包括整个部门的信息, 对这个问题

在数据库中通过外键来实现, deptno, posid, mgrid

在java类中通过属性关联实现

    /\*    我们不能这样写:

    private int deptno;

    private int posId;

    private String mgrId;

    \*/

           应该这样写:

    private Department dept; //员工所属部门  不仅包含部门的编号，还包含其他信息

    private Position position;

    private Employee mgr;//上级领导的信息

    private List<Employee> empList = new ArrayList<Employee>();//下级的信息，可能多个

2)然后搭建MVC框架, 接口->类

3)把员工类有关的html页面转换为jsp的

**员工管理-添加员工-实现后台开发**

1)写Dao层的时候注意员工类中日期是util类的, 我们要把它转化为sql类的Date, 利用他们之间共同的时间属性

new java.sql.Date(emp.getBirthday().getTime());    //sql的Date只有年月日, 没有时分秒

带来的新问题 : **添加员工的时候不会写离职时间, 那么向数据库添加的时候会出现空指针异常**, 待解决..

2)service -> servlet 就和部门和岗位添加差不多了

Servlet层在接收数据时要注意的地方 :

//得到日期参数

String sbirthday = request.getParameter("birthDay");

Date birthDay = null;

DateFormat format = new SimpleDateFormat("yyyy-MM-dd");

birthDay = format.parse(sbirthday);

//对象型信息得处理

int deptno = Integer.parseInt(request.getParameter("deptno"));

Department dept = new Department();

dept.setDeptno(deptno);

**员工管理-添加员工-实现前台开发**

1)完善添加员工的表单

2)到EmployeeDaoImpl中添加对离职日期的判断

java.sql.Date leavedate2 = null;

Date leavedate = emp.getLeaveDate();

if(leavedate != null){//离职日期不为空才创建新的sql Date对象,(为空也创建的话会报空指针异常), 否则传个空值给数据库

    leavedate2 = new java.sql.Date(leavedate.getTime());

}

部门, 岗位, 上级领导, 注意外键约束, 数据库表中要有值才行

关于主外键介绍的文章, 添加主键<https://blog.csdn.net/cuixiaoming1994/article/details/79525513>

3)能成功添加到数据库表

**员工管理-完善添加员工-使用日历插件和富文本编辑器**

日历插件:

1)把My97DatePicker文件夹复制到web目录下, 设置文件的编码为gbk

2)在empAdd.jsp中引入文件WdatePicker.js,

3)在出生日期的input标签中设置属性onfocus="WdatePicker()"    //WdatePicker()是WdatePicker.js中声明的函数

4)要改变皮肤怎么办?

    搜索My97DatePicker用法\*\*\*

使用json格式onfocus="WdatePicker({skin:'whyGreen', lang:'en'})    变英文和绿色

富文本编辑器:

1)框架中已经存在,editor文件夹就是

2)直接在页面中引入kindeditor.js

3)另写一个script标签,

</script>

<script type="text/javascript">

    KE.show({id:"ecp",width:"800px",height:"300px"});    //里面写json格式, 添加富文本标签的id

</script>

4)我们可以搜索富文本编辑器, 然后到响应编辑器的官网下载, 并查看文档学习使用

**员工管理-完善添加员工-动态显示部门列表和员工上级**

两种思路 :

1.在导航栏先跳转到Serlvet, 获取到部门和员工等信息后再跳转到jsp

2.直接跳转到jsp, 用三个ajax请求完成获取信息

两种都实现

1-1)获取上级员工的时候使用方法findEmpByType(), 1自己设定1表示上级, 2表示下级(这样方法的可重用性高点)

**员工管理-查询所有员工-后台实现**

也许会想 : 逻辑不是一样的吗? 直接 select \* from employee不就行了吗?

要注意 : employee里面如部门, 岗位, 上级领导, 只是保存了一个id号, 查询所有员工是一个多表查询的功能

**关键点在于sql语句**

1)老师使用的是原生的dbutil, 通过while循环获取表中所有字段的值, 然后按需要封装成对象使用

2)我是用的是common-dbutils, 里面封装好了许多种查询结果返回集(详情请查看笔记)

当多表查询的时候使用 :

// 把结果集转换成List<Map>

List<Map<String,Object>> list = qr.query(sql, new MapListHandler() );

然后同样把查找得到的逐一字段封装成对象, 再进行返回.

比如这里查询所有员工的案例 :



1)这是我最终显示的页面, 只需要用到e.empid, e.realname, e.dept.deptname, e.pos.pname, e.hireDate, e.phone

2)在数据库图形界面中先写好能查出这些字段的sql语句,

3)把sql语句复制到dao中, 通过查询返回结果List<Map>

4)通过for循环拿到每一个Map, 并在每一个Map中先封装好一个dept, pos, 最后再封装好emp

    -(用于外键的原因, emp并不是持有deptid, posid引用, 而是直接持有dept, pos引用)

5)最后把每个emp都添加到List中, 最后返回

SELECT e.empid,e.realname,d.deptname,p.pname,e.hiredate,e.phone

FROM employee e

JOIN dept d

ON e.deptno=d.deptno

JOIN POSITION p

ON e.posid = p.posid;

//如果还加上上级表的话, 最大的总裁会查不出来, 加个left就行

SELECT e.empid,e.realname,d.deptname,p.pname,e.hiredate,e.phone

FROM employee e

JOIN dept d

ON e.deptno=d.deptno

JOIN POSITION p

ON e.posid = p.posid;

left JOIN employee m

ON e.mrgid = m.empid

**员工管理-多条件查询员工-接收和记忆查询条件**

1)Servlet中创建findEmp方法, 复制findAll的代码片段

2)在findAll方法中添加查询所有部门信息的代码, 用于以下下拉框显示



3)把empList.jsp中按条件查询的html代码部分用表单封装起来

4)接收表单传来的查询条件, 并保存起来, 以后用于回显

注意下拉窗口和多选框的代码逻辑

**遇到的问题:**

1)在接收查询条件的时候, 雇佣日期以字符串的形式传来, 老师回显是把日期字符串参数直接保存到request域中,

但我我日期字符串按格式保存为Date对象, 把Date对象给emp实例, 再把emp保存到requset域中, 结果输出emp.hireDate的时候没有按格式输出, 尚未解决

**员工管理-多条件查询员工-完整的查询操作**

1)目前findEmp方法中查询员工是查询全部, 要带上条件,创建find方法, 按MVC流程把封装好的emp传给dao层

2)复制findAll的代码, 把sql改为StringBuilder类型, 便于拼接查询条件(**拼接查询条件前加个 "where 1=1"** ,技巧)

3)拼接条件的时候如果出错, 记得加上**表名**

    -日期的比较

and to\_char(e.hiredate,'YYYY-MM-DD') >= '2013-12-13'

//把Date对象变成字符串

DateFormat = df = new SimpleDateFormat("yyyy-MM-dd");    //java中,格式严格遵循大小写

String shireDate = df.format(hireDate);

//然后再去比较

完成查找后, 改动findAll方法, findAll只是findEmp的一种特殊情况, 处理一下后可以把findAll删除

1)修改left.html, 员工管理指向findEmp,

**遇到问题:**

1)mysql中日期转字符串, 参考<https://www.cnblogs.com/wangyongwen/p/6265126.html>

2)拼接sql字符串的时候, 记得在每行前面**加空格**

3)注意sql的语法, 该用单引号括起来就要括起来

--引申 : 很多地方都要加引号包住参数, 我们尽量要想办法看到拼接完的代码

        --比如可以用谷歌的F12看jsp中的代码, 还可以在谷歌设置断电调试

SELECT e.empid,e.realname,d.deptname,p.pname,e.hiredate,e.phone

FROM employee e

JOIN dept d

ON e.deptno=d.deptno

JOIN POSITION p

ON e.posid = p.posid

WHERE 1=1

AND e.empid LIKE '%a%'

AND e.onduty=1

AND DATE\_FORMAT(e.hiredate, '%Y-%m-%d') >= **'2018-10-23'**;    ----有单引号括起来的

java中代码:

sql.append(" AND DATE\_FORMAT(e.hiredate, '%Y-%m-%d') >= '"+ shireDate +"'");

**注: 如果不是用QueryRunner中填写占位符的方法, 拼接sql语句的时候字符串字段一定要注意加上单引号**

4)相同过findEmp方法代替findAll方法, 但是注意findEmp方法中获取参数的代码需要调整

        ---findEmp方法中要对接收的属性做出判断, 因为有些空的属性, 比如日期, 和deptno字符串, 我们要把他们转化为uitl.Date和Integer.parseInt(....), 如果不对这些属性作是否为空的判断, 就会出现**NumberFormatException**, 类型转换异常, 空的字符串当然不能转换了

//接收查询表单数据

String empid  = request.getParameter("empid");

String sDeptno = request.getParameter("deptno");//拿到员工的部门编号, 如果为空字符串或者没有值, 就不进行转换了

int deptno = 0;

if(sDeptno != null && !sDeptno.equals("")){

    deptno = Integer.parseInt(sDeptno);

}

String sOnDuty = request.getParameter("onDuty");//拿到员工的部门编号, 如果为空字符串或者没有值, 就不进行转换了

int onDuty = 1;//这里可能欠完善, 目前默认是只查询在职的员工

if(sOnDuty != null && !sOnDuty.trim().equals("")){

    onDuty = Integer.parseInt(sOnDuty);

}

//如果没有填写日期的话, 就不进行日期格式转换了

String shireDate = request.getParameter("hireDate");

Date hireDate = null;

    DateFormat df = new SimpleDateFormat("yyyy-MM-dd");

if(null != shireDate && !shireDate.trim().equals("")){

    try {

        hireDate = df.parse(shireDate);//转化的是字符串

    } catch (ParseException e) {

        e.printStackTrace();

    }

}

5)我的代码中, 默认设置了onDuty为1, 也就是默认只查找在职的员工, 这里以后看能不能调整一下

**Servlet层的代码:**

String sOnDuty = request.getParameter("onDuty");//拿到员工的部门编号, 如果为空字符串或者没有值, 就不进行转换了

int onDuty = 1;//这里可能欠完善, 目前默认是只查询在职的员工

if(sOnDuty != null && !sOnDuty.trim().equals("")){

    onDuty = Integer.parseInt(sOnDuty);

}

**Dao层的代码:**

//这里可能要进行完善, 目前在Servlet控制层, findEmp方法中封装的emp的onduty默认是1, 也就是在职

sql.append(" AND e.onduty=" + emp.getOnDuty());

6)字符串判断 是否为空的时候, 如果想加上使用trim()判断字符串不为空字符串 "" ,

    -- 一定要把判断不为空字符串写在 && 后面

    -- && 是前面如果为否就直接退出了, 所以我们要先判断字符串是否为空, 然后在使用trim()会报空指针异常

**员工管理-删除员工以及删除策略考核**

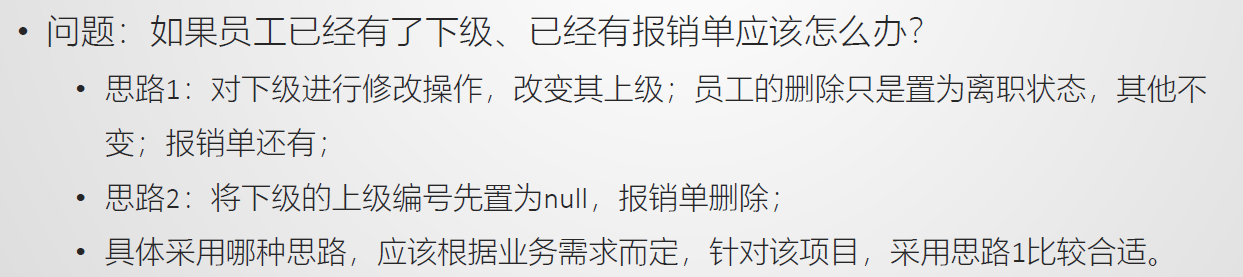
1)先补完上一步中删除findAll转为findEmp的操作

2)像删除部门, 岗位一样完成员工的删除操作, 要给删除按钮添加删除确认

    --注意 : 删除时要传递empid, 删除完转发到findEnp也要参数empid, 要注意区分

        --解决办法, 1)参数名称不同    2)该转发为重定向

3)删除操作实现容易 : 难在要思考删除后, 与该员工相关的子记录怎么办?



**遇到的问题:**

<td>

    <a href="empInfo.html" class="tablelink">查看</a>

    <a href="empUpdate.html" class="tablelink">修改</a>

    <a href="javascript:delete\_confirm**('${emp.empid}')**" class="tablelink "> 删除</a>

    <a href="#" class="tablelink"> 重置密码</a>

</td>

1)开始时没有给EL表达式加上单引号'', 导致出现numberFormatException

到目前为止系统遗留的问题 :

~~1)添加员工后, 不能跳转到显示页面~~

2)员工添加失败后没有显示错误信息

3)添加员工的时候会报空指针异常

4)全部添加的表单页面都没有验证窗口

**员工管理-更新员工-预更新操作的后台代码**

1)在servlet中添加方法toUpdate,预更新操作

2)注意在toUpdate方法中, 除了要获得对应员工的信息, 还要获得部门信息,上级和岗位信息, 以便显示在下拉框中

    -其中获得员工信息调用service层的findById(), 以后这个方法还能用作查看员工信息

3)Dao层中, 查询一个员工的信息虽然是单表查询, 但是由于最后封装一个员工对象, 结果集要用List<Map<String, Object>>

**员工管理-更新员工-预更新操作前台回显**

1)把empUpdate.html改为jsp页面

2)把empUpdate上的静态数据改为动态数据, 并且写好表单各项的名称

3)逐个测试, 完成前台回显

**员工管理-更新员工-完成更新操作**

1)设置empUpdate.jsp中empid为只读, 不能更改

2)Servlet中如果更新员工失败, 就把错误信息显示回更新页面

3)在Dao层只要复制添加员工的代码, 但是要修改sql语句, 并且在参数数组中, 把empid放到最后面

**员工管理细节完善**

1)完善员工列表中的查看功能

2)员工修改密码的功能暂时不上线

待解决问题 : 要把不在职的员工也显示出来