**数据库应用课程设计**

**实验报告**

指导老师：王宏健

学生姓名：董承明

学 号：171310524

完成时间：2020/4/23

目录

[一、 项目背景及意义 3](#_Toc38573250)

[一、 选题依据与研究意义 3](#_Toc38573251)

[二、 选题的研究背景 4](#_Toc38573252)

[二、项目目标与要求 4](#_Toc38573253)

[一、项目目标 4](#_Toc38573254)

[二、具体功能要求 4](#_Toc38573255)

[三、 项目技术 5](#_Toc38573256)

[四、 个人承担的任务 5](#_Toc38573257)

[五、 项目的设计与实现 7](#_Toc38573258)

[一、 数据库设计 7](#_Toc38573259)

[二、项目结构 9](#_Toc38573260)

[三、json文件格式 10](#_Toc38573261)

[1） bean层与dao层 11](#_Toc38573262)

[2） servlet写法 12](#_Toc38573263)

[3） 前端测试数据与接收 13](#_Toc38573264)

[四、ajax技术 15](#_Toc38573265)

[五、layui的一些方法 17](#_Toc38573266)

[1） 分页功能 18](#_Toc38573267)

[2） 加载layui模块 19](#_Toc38573268)

[3） 表格重载 19](#_Toc38573269)

[4） 动态生成下拉列表 20](#_Toc38573270)

[5） 初始界面的设置 21](#_Toc38573271)

[六、 实验中的一些细节 22](#_Toc38573272)

[1） 退房时信息的读取 22](#_Toc38573273)

[2） 动态生成下拉列表 26](#_Toc38573274)

[3） 模糊查询与精确查询 27](#_Toc38573275)

[六、使用手册 28](#_Toc38573276)

[七、 实验中遇到的问题 31](#_Toc38573277)

[八、 心得体会 34](#_Toc38573278)

# 项目背景及意义

## 选题依据与研究意义

为了提高宾馆的经济效益，对人流、物流、资金流进行科学管理和有效控制，开发一个稳定可靠的宾馆管理系统用以提高员工的工作效率与降低各种经营成本具有现实的必要性和重要意义。随着计算机技术的飞速发展，信息时代的到来，信息改变了我们这个社会。各类行业在日常经营管理各个方面也在悄悄地走向规范化和信息化。宾馆管理的信息化程度体现在将计算机及信息技术应用 于经营与管理，以现代化工具代替传统手工作业。无疑，使用信息化管理使客房管理更先进、更高效、更科学，信息交流更迅速。宾馆管理系统是宾馆经营管理中不可缺少的部分，它的内容对于经营的决策者和管理者来说都至关重要，所以宾馆管理系统应该能够为用户提供充足的信息和快捷的查询手段。用。作为一种以服务为主的无烟工业，宾馆在其运作期间，服务水平的高低，如服务安排、调度等，直接影响到宾馆的声誉和形象。以客房管理为核心，为用户提供迅速、高效的服务，减免手工处理的繁琐与误差的宾馆管理系统能提高宾馆的服务质量，再配合现代化的宾馆管理，能获得更好的经济效益。

但一直以来人们使用传统人工的方式管理文件档案，这种管理方式存在着许多弊端，如：效率低、保密性差，容易出现差错等，且对于查询空房间及已定房间等极为不方便。宾馆管理是将宾馆管理与计算机技术紧密结合起来，对宾馆的日常事务进行现代化的科学管理，以此来提高宾馆形象、档次和服务水准。这不仅使工作人员的劳动强度降低，工作效率增加，而且能快速地调度宾馆各部门的工作，实时动态地掌握宾馆的经济状况，从而为企业带来良好的经济效益和社会效益。在国内，近年来，随着我国改革开放的发展，国内的宾馆服务业得到了日新月异的发展。现代宾馆作为一体化的接待场所，相当于一个小型社会。对一个城市而言，宾馆业也起到了举足轻重的作用。

## 选题的研究背景

选题研究背景面对庞大的信息量，宾馆在正常运营中需要对客房资源、顾客信息、结算信息进行管理，为了轻松的处理宾馆庞大的信息量，减少在管理方面的物力和人力，提高宾馆的工作效率和经济效益，及时了解各个环节中信息的变更。开发一个宾馆管理系统将会为宾馆的管理带来极大的方便，并且降低宾馆的管理成本。通过系统，可以做到信息的规范管理、统计和快速的查询，从而减少在管理方面的工作量。在某种意义上，宾馆管理的信息化已成为现代化宾馆的重要标志。

# 二、项目目标与要求

## 一、项目目标

通过本课程实践，掌握开发Web应用系统的方法与技术。

通过分组形式，使用先进的Web开发技术开发一个功能相对完整、界面友好的数据库应用系统。

## 二、具体功能要求

一套功能强大而又使用简便实用的客房管理软件，适用于宾馆管理、宾馆管理、招待所管理。主要功能：团体,散客入住客人的信息登记，换房，正常退房，挂帐退房，挂单补款结帐，预定管理，贵宾卡管理，押金管理，各种强大的宾馆统计报表，异常处理。

# 项目技术

使用maven项目结构。数据库采用mysql，并用navicat可视化工具进行管理。数据传输交互采用servlet、ajax、json。前端UI采用layui框架，套用layuimini模板，使用基于jquery及javascript的jsp页面实现。使用IDEA作为开发工具，并将网页部署在tomcat上运行。

# 个人承担的任务

1. 在项目初期在网上查阅大量资料，最终由本人分析各种不同技术的利弊并最终选定项目框架以及前后端采用的技术。
2. 中期通过自学servlet、json、jquery、ajax、javascript等知识，掌握前后端信息交互的原理。在github上花大量时间寻找合适的前端模板，并最终选定layuimini。通过阅读文档熟练掌握其重要组件如：数据表格(table)、表单(form)、弹出层(layer)、分页(laypage)等等。将以上知识融合并做成一个可以实现简单前后端交互的demo，通过视频会议的方式分享给另两个组员，帮助他们两人参与到项目的开发中，使得小组不同组员的工作能够顺利开展，为当时迷茫的队友指明方向。
3. 在实际开发中，耐心解决了许多队友无法解决的问题。大到ajax参数设置、servlet调用调试方法，小到引入css的路径错误、前端jquery组件少打#导致无法获取input的value、servlet没有在web.xml中配置等等等。
4. 独立开发：
5. 项目的login界面
6. 退房界面(checkout.jsp)、servlet接口：显示房间(RoomTableServlet)、

查询房间(RoomSearchServlet)、显示当前入住表(NowTableServlet)

1. 退房子页面退房信息(checkoutinfo.jsp)、servlet接口：所有信息(InfoServlet)、退房动作(CheckoutData)
2. 退房子页面换房(changeroom.jsp)、servlet接口：换房动作(RoomChange)
3. 还款页面(returnmoney.jsp)、servet接口：还款动作(ReturnMoney)、永久信息表(PermanentTable)
4. AllInfo, Permanent, Now, Checkout, Vip, Room 等表及操作的Bean层以及Dao层，分别置于model与db文件夹下

# 项目的设计与实现

## 数据库设计

数据库共7张表：分别如下

房间表：

房间号(主键)

房型（外键--押金表）

座机号(候选码)

租房价格

状态（有人住，没人住，被预定(押金)）

押金表：

房型（主键）

押金

当前入住：

身份证号（主键）

住的房间号（外键->房间表）

团体还是散客

贵宾表：

贵宾号（主键）

身份证号(外键->永久信息表)

统计报表：

时间（主键）

入住人数

房间预定数

赊账总金额

盈利总额

贵宾增加数

入住统计表：

事件号（主键）

入住人身份证（外键->永久信息表）

房间号（外键->房间表）

时间

是否团体

永久信息表：

姓名

身份证号（主键）

是否贵宾

赊账金额（0就没有赊账）

创建数据库的文件请参照createdb.txt

## 二、项目结构

Maven文件结构的创建与个人任务相同，这里简要介绍maven项目的几个优势：优势一：相同的项目结构，使用Maven管理的Java 项目都有着相同的项目结构1. 有一个pom.xml 用于维护当前项目都用了那些jar包

2. 所有的java代码都放在 src/main/java

3. 所有的测试代码都放在src/test/java

Maven 优势二：统一维护jar包。maven风格的项目，首先把所有的jar包都放在“maven仓库”里，然后哪个项目需要用到这个jar包，只需要给出jar包的名称和版本号就行了。这样jar包就实现了共享。

Maven 优势三：不像springboot及hibernate等会因为不同的版本号，或与mysql版本不匹配导致运行错误，方便组员之间的代码重用与合作。

## 三、json文件格式

由于layui的table组件的数据使用的是url自动生成，必须学会如何将数据打包成json文件。在laymini提供的模板中，其使用的是静态的.json文件，这对于我们的动态项目显然不合适。在查阅了资料后，我发现servlet可以作为json文件的载体接口。参考链接：

<https://blog.csdn.net/hsfy2012/article/details/80251426>

首先给出jsonobject的定义：

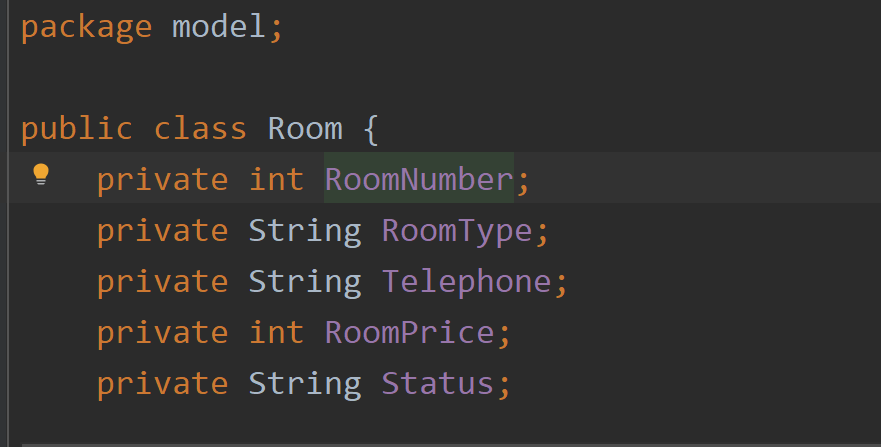
JSONObject只是一种数据结构，可以理解为JSON格式的数据结构（key-value 结构），可以使用put方法给json对象添加元素。JSONObject可以很方便的转换成字符串，也可以很方便的把其他对象转换成JSONObject对象。

而JSONArray则是一个数组。JSONArray和JSONObject再使用时候的第一个最重要的区别就是：JSONArray是一个一个加进去的，而JSONObject是一对一对加进去的。

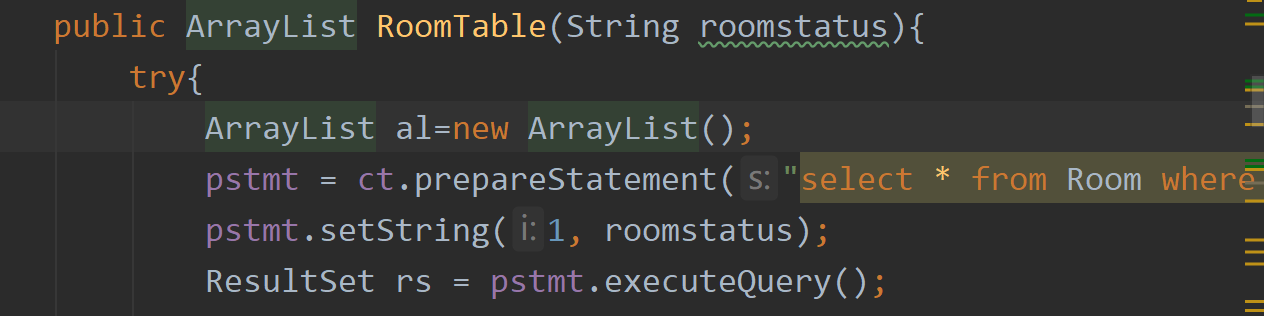
事实上，JSON就是python中的dictionary，是一种非常方便、简洁的数据储存格式，一旦掌握后，前端可以通过key获取相应的value，而layui的几乎所有组件都依赖于这种键值对的数据格式。以下以Room表为例，介绍如何将数据库中的一张table，一步步传输封装数据，在layui框架的前端中显示出来。

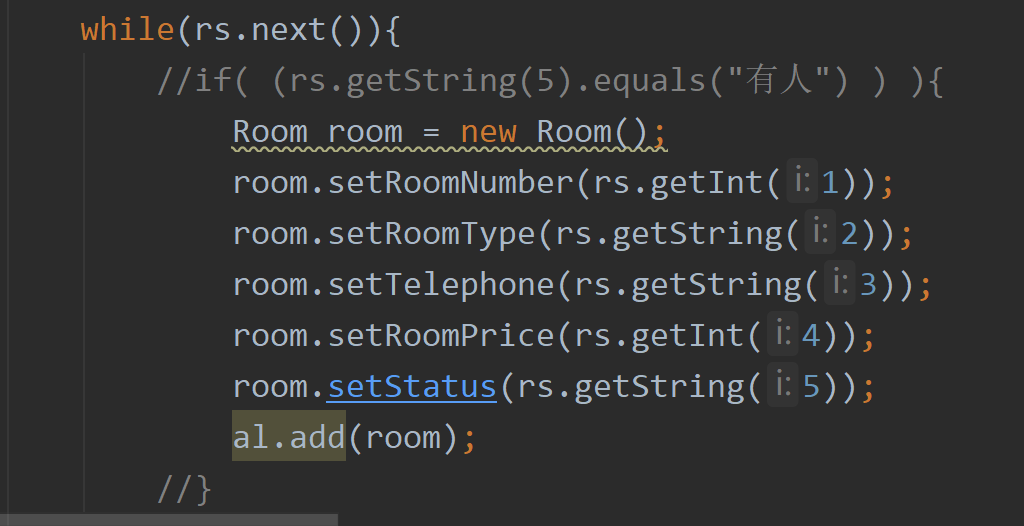
### bean层与dao层

bean层不多介绍，对应数据库的字段设置私有属性，并使用idea的generate方法自动生成每个属性的get、set方法。



Dao层以CheckoutDao中的RoomTable方法为例。

该方法返回一个ArrayList数组。首先执行sql语句，查询出room表的每一条信息。



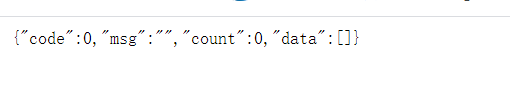
对于每一条信息，new一个Room对象，一一对应地设置其属性，完成后添加到ArrayList中，因此该方法返回的就是一个Room类型的数组。

### servlet写法

这里以RoomTableServlet的doPost为例

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws IOException {  
 String roomstatus=req.getParameter("roomstatus");//从前端jsp页面获取要查询的房间状态  
 resp.setContentType("text/json;charset=UTF-8");  
 PrintWriter out=resp.getWriter();//以输出流的方法向前端页面输出  
 JSONObject jsonObject = new JSONObject();  
 JSONArray result = null;  
 CheckOutDao co = new CheckOutDao();  
 result = JSONArray.*fromObject*( co.RoomTable(roomstatus));//RoomTable返回的是一个ArrayList，这里将其转为JsonArray  
 jsonObject.put("code",0);//Layui规定  
 jsonObject.put("msg","");//Layui规定  
 jsonObject.put("count",co.RoomTable(roomstatus).size());//数组大小  
 jsonObject.put("data",result);//把数组作为data  
 out.println(jsonObject.toString());//一定要有这条语句，才能把这个Json对象以字符串的形式传给前端  
 out.flush();//清空输出流  
 out.close();//关闭输出流  
}

如果该servlet输出的json字符串作为layui组件接口，一定要符合如下格式。code，msg，count缺一不可，值也不能设错。



如果不是接口，只是数据的话，则可以采用普通的键值对

### 前端测试数据与接收

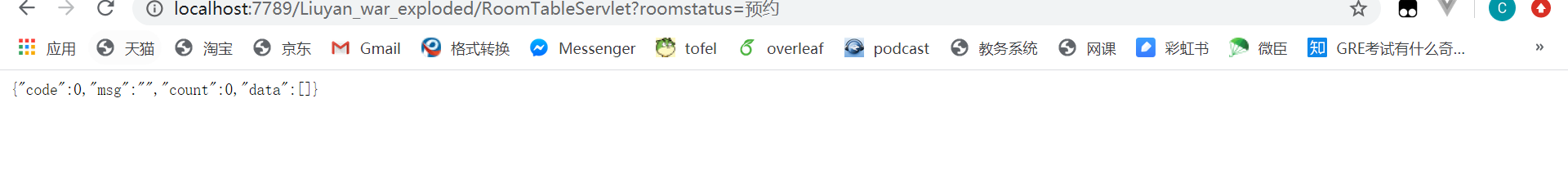
在checkout.jsp的script中，可以看到，url正式刚才写好的Servlet，由于layui已经封装好了table这个组件，我们无需处理数据。而where还可以设置参数，将roomstatus逆向传给servlet，做到条件查询。这样数据的双向传输就完成了。

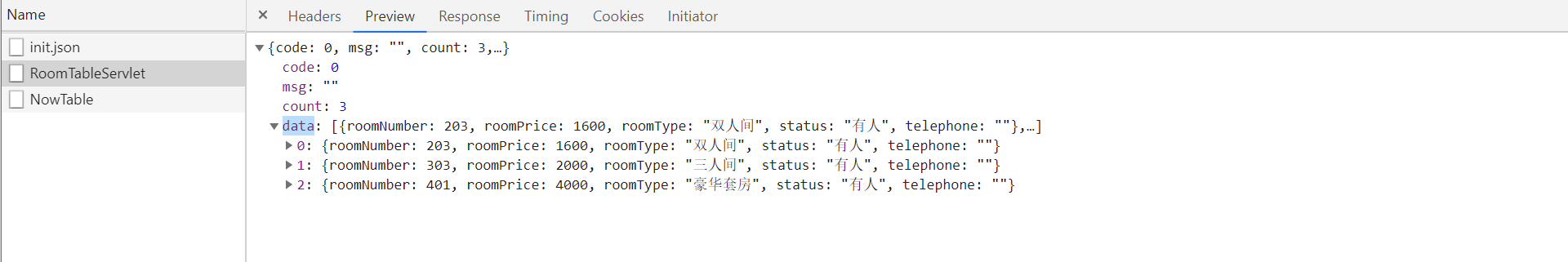


而在table.render方法中，cols参数也需要设置。这里的每列的表头名需要和传过来的json字符串data中的各个key相同，区分大小写。而顺序可以自己设置。



测试servlet向前端传送的json数据是否符合要求，可以直接在地址栏输入调用servlet的网址，如果正确结果如下。





如果出现404报错，多半是servlet相对路径错误，采用以上方法先检查servlet输出的数据是否有效，确保network中成功调用当前servlet，再进行前端的调试。

## 四、ajax技术

AJAX全称为“Asynchronous JavaScript and XML”（异步JavaScript和XML），是一种创建交互式网页应用的网页开发技术，通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。说的简单易懂一点，它可以在不刷新整个页面的情况下与服务器通信保持原有页面状态。

使用ajax的一个重要原因就是，在实验过程中，表单甚至任何数据与后端的交互都依赖于submit。如果使用form的action方法，非常受限制。比如同一个页面如果有多个嵌套的form，同一个form中拥有多个不同的按钮对应不同的servlet操作等情况。通过使用ajax技术，摆脱了form的约束，页面做到了异步刷新，在script的任何地方都可以调用servlet。可以说，ajax是jsp页面不可或缺的技术之一。这里提供一个ajax函数，注释中为每个参数的意义与作用。

$.ajax({  
 type: "post", //传数据的方式  
 url: "../../CheckoutData", //servlet地址  
 data: { room:***parent***.room, paymen:option ,ownmoney:should-actual,earn:actual }, //传的数据  
  
 success: function(data){  
 //传数据成功之后的操作 result是servlet传过来的数据 这个函数对result进行处理，让它显示在 输入框中  
 }  
});

以上ajax基本格式可以当作一个函数，在任何想要对后台进行操作的时候进行操作。

关于data参数，可以传键值对，形如： Data:{ key: value}

这样在servlet中可以通过request.getParameter(“key”)获得value的值。

如果想要获得批量数据，可以将某一部分数据放置在一个form中，通过其id属性获得其中所有的数据，形如：data: $('#test').serialize() test即为form的id。此方法与form的action方法很类似，servlet可以通过name与id获取前端对应id中的内容。

注意，由于使用了ajax，form无需提交。实验过程中遇到过ajax与form交叉重叠导致network中有一个servlet状态显示为cancelled。其内在原因可能是，layui的button监听通过form.on('submit(button的id)', function (data) {…}来实现，layui自带的form的submit方法无法与页面上的普通form重叠。

处理方法如下（将form的onsubmit值自动设置为false，这样就不会提交表单）

<form id="test" onsubmit="return false">

这个方法同样解决了调用servlet之后自动刷新页面的问题，从而使得数据可以保留在input内不被刷掉。

## 五、layui的一些方法

layui可以说是目前国内最优秀的几个前端UI之一，它是一款采用自身模块规范编写的前端 UI 框架，它遵循原生的 HTML/CSS/JS 书写方式。它虽然外在极简，但是内容丰富，里面包含众多组件从核心代码到 API 都非常适合界面的快速开发。事实上layui更多是面向于后端开发者，而且它还拥有自己的模式，更加轻量和简单。layui 部分模块依赖 jQuery（比如 layer），但是你并不用去额外加载 jQuery。

layui 已经将 jQuery 最稳定的一个版本改为 layui 的内部模块，当你去使用 layer 之类的模块时，它会首先判断你的页面是否已经引入了 jQuery，如果没有，则加载内部的 jQuery 模块，如果有，则不会加载。

Layui使用非常简单，无需搭建任何框架，只要将其完整地部署到你的项目目录（或静态资源服务器），只需要引入下述两个文件：

codelayui.code

./layui/css/layui.css

./layui/layui.js //提示：如果是采用非模块化方式（最下面有讲解），此处可换成：./layui/layui.all.js

在jsp的<head>标签中，统一加入：

<link rel="stylesheet" href="../lib/layui-v2.5.5/css/layui.css" media="all">  
<link rel="stylesheet" href="../css/public.css" media="all">

在jsp的script标签前，统一加入：

<script src="../lib/layui-v2.5.5/layui.js" charset="utf-8"></script>

### 分页功能

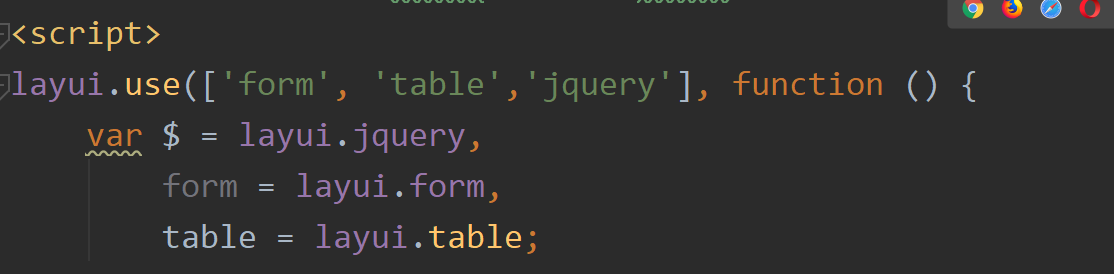
Layui的table可以自动实现分页功能。只要传送的是正确的json数据，可以在前端设置参数并加入parseData函数，即自动实现分页，无需再后端对data进行繁琐的处理。

limits: [3,6,9],  
limit: 3,  
page: true,  
parseData:function(res){ //res 即为原始返回的数据  
 if( this.page.curr) {  
 var result=res.data.slice(this.limit\*(this.page.curr-1),this.limit\*this.page.curr);  
 } else {  
 var result=res.data.slice(0,this.limit);  
 }  
 return {  
 "code": res.code,  
 "msg":"",  
 "count": res.count,  
 "data": result  
 }  
}

以上代码加在table.render()函数中，limit以及limits作为分页条数可自行设置。parseData函数作为模板，可直接复制粘贴并通用于任何table中。

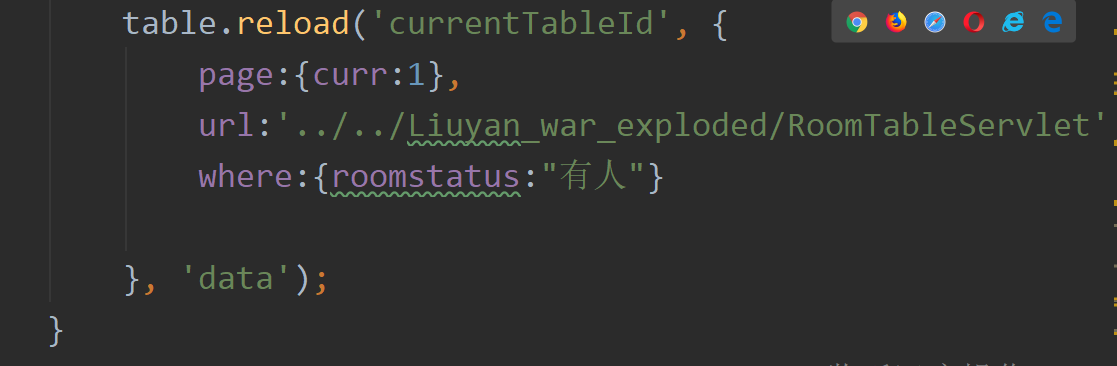
### 加载layui模块

这点特别需要注意，在实验中浪费了非常多时间。layui 的内置模块并非默认就加载的，他必须在你执行该方法后才会加载。它的参数跟上述的 define 方法完全一样。另外请注意，mods 里面必须是一个合法的模块名，不能包含目录。如果需要加载目录，建议采用 extend 建立别名。在任何使用到layui的组件中都必须加上类似layui.use的代码。



### 表格重载

Layui的reload方法非常简便。可以通过设置不同的url以及where参数，调用不同的servlet服务，传入不同的参数，而之前render方法中的各个参数则依旧保留。



Page:{curr:1}，这条语句非常重要。因为开启了分页功能之后，数据被parseData函数分割。不加这条语句，重载信息的搜索不会从第一页而会从当前选定的页进行搜索，导致该页之前的信息被忽略。如果操作的信息在该页之前，操作就会失效。

### 动态生成下拉列表

初始化下拉列表组件时不设置option的值



当ajax成功返回需要设置的数据（住户的身份证数组）时，获取列表的dom元素，使用append方法添加option，类似字符串连接，先当与在html代码里新增几行。记得使用form.render函数重新渲染select组件。

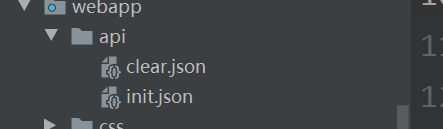
for (var index in JSobj.ids) {  
 $("#paymen").append("<option value=" + index + ">" + JSobj.ids[index] + "</option>");  
}  
form.render('select');

关于表格的其他操作，详见layui官方文档

### 初始界面的设置

由于采用了laymini的iframe模板。index.jsp是模板页。其他所有页面都是其子页面。

其左端列表采用静态json文件初始化，分别为init.json和clear.json：



通过修改init.json，我们可以自定义左侧菜单栏的内容：



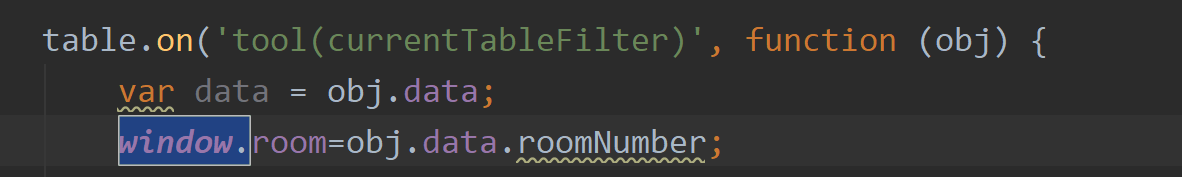
## 实验中的一些细节

### 退房时信息的读取

在退房界面中，需要通过当前的房间获取所有的信息。首先，我使用了一个layui中layer的弹窗页面，通过设置content参数，可以跳转到对应的jsp页面。

content: './table/checkoutinfo.jsp',

这里涉及到一个layui父子页面传值的技巧。在父页面监听table中的其中一行，并且用obj获得改行所有数据，用roomNumber获得房间号。然后给window（相当于当前页面父页面）设置一个该页面自己的变量room。这样子页面就可以使用parent.room获得父页面中设置的这个值，作为ajax参数进行后台操作。





通过这个房间号，我们需要获取数据库中关于当前这个房间的所有信息，包括住户身份、团体、贵宾、房间金额、入住日期等等。这势必要查询到checkin表。

然而，简单地通过房间号查找checkin表并不合适。反例比如：同一个住户a反复入住房间b，由于checkin表并不会因为退房而删除其中数据，之前入住房间b的住户以及信息将被错误地读入。因此，查询checkin表的正确也是唯一条件应该是时间。

"select YearMonthDay from checkin where roomliving=? order by YearMonthDay desc limit 0,1";

这条sql查询语句的意思是：以时间降序，获得第一条，房间号与传入的房间参数相同的记录的时间。结果将返回一行一列，一个单元格的数据。这个数据一定是当前住在房间内的住户的入住时间。

因为入住表的插入有时间先后，最后插入的记录一定是当前正住在房间里等待退房的那条记录。一旦获取了它的入住时间信息，我们就能在checkin表中获得其他拥有相同时间的记录，也就是当前住在该房间内的住户。

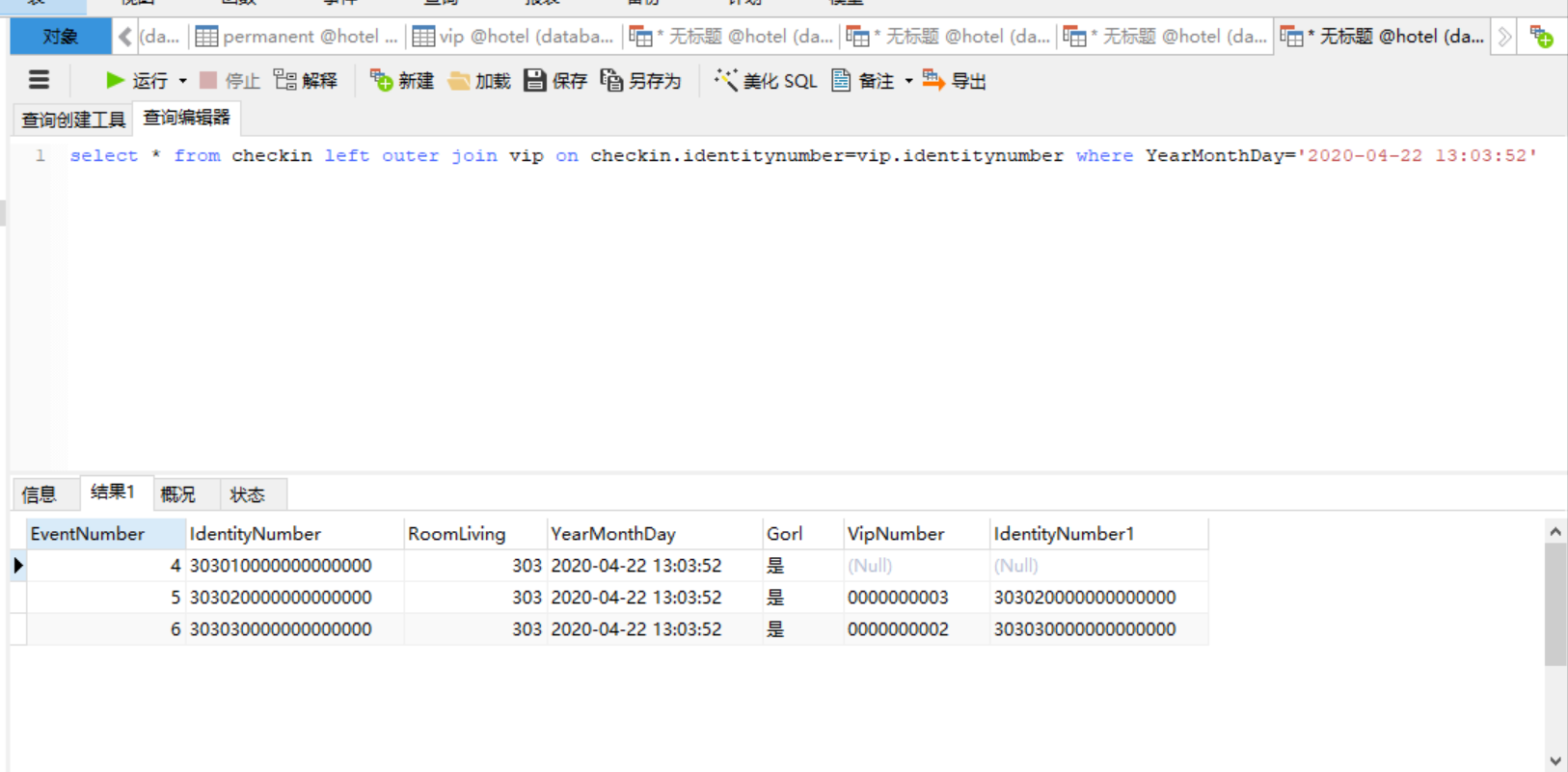
"select \* from checkin left outer join vip on checkin.identitynumber=vip.identitynumber where YearMonthDay=?"

之后我们使用这条“左联查询”语句，通过身份证联合查询checkin表以及vip表，因为贵宾信息必须通过vip表获得。

左联查询定义：显示左表T1中的所有行，并把右表T2中符合条件加到左表T1中;右表T2中不符合条件，就不用加入结果表中，并且NULL表示。

使用左联查询的原因是，它会显示左表(checkin)中的所有信息，右表(vip)即使查询不到，也会返回null。如果使用联合查询，如果当前用户不是vip，那这条记录就不会被查出来，显然与我们的逻辑不符。

示例结果如下：



可以看到，最终我们通过房间号获得了最近的时间，并且用一条sql查询语句联合3张表，一口气获得了所有需要的数据。代码得到了非常大的精简，同时提高了查询速度。

为了打包储存这些数据，我又专门定义了一个model类名为AllInfo。每一条联合查询获得的记录作为一个对象放入对象数组ArrayList中。

public class AllInfo {  
 public boolean gorl;  
 public ArrayList<String> ids=new ArrayList();  
 public String checkintime;  
 public Integer price;  
 public boolean vip;  
 public void setDate(Date date) {  
 this.checkintime = new SimpleDateFormat("yyyy-MM-dd").format(date);  
 }  
}

setDate方法将sql中读出来的Date类型数据转换成string并发送给前端。前端再解析这个字符串做时间间隔的判断。

这边需要特别注意一下mysql的sqldate以及java的utildate之间的类型转换：

Mysql 与 java 的时间类型

MySql的时间类型有 Java中与之对应的时间类型

  date                 java.sql.Date

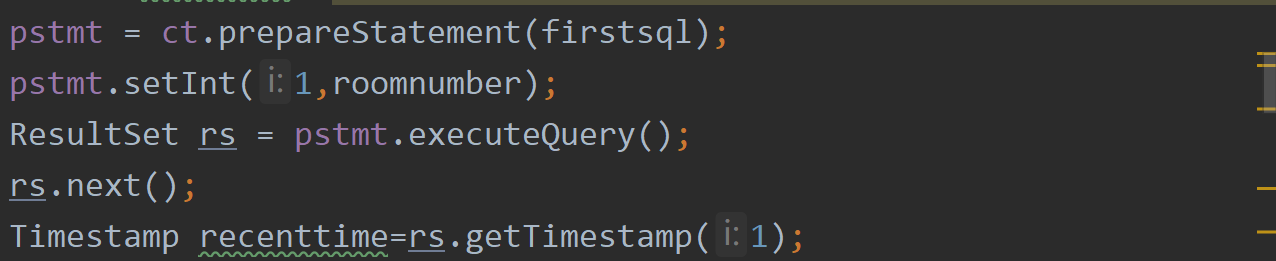
  Datetime          java.sql.Timestamp

  Timestamp       java.sql.Timestamp

  Time                 java.sql.Time

  Year                 java.sql.Date

这边，数据库中的YearMonthDay是datetime类型，所以在java程序中我们使用Timestamp类型来获取

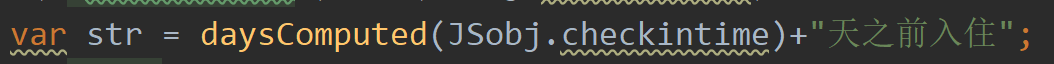


### 动态生成下拉列表

此功能在前端jsp页面中实现，具体函数如下

var daysComputed = function(time) {  
 var oldTimeFormat = new ***Date***(time.replace(/-/g, '/'));  
 var nowDate = new ***Date***();  
 if(nowDate.getTime() - oldTimeFormat.getTime() > 0) {  
 var times = nowDate.getTime() - oldTimeFormat.getTime();  
 var days = parseInt(times / (60\*60\*24\*1000));  
 return days;  
 } else {  
 throw('传入时间不能大于当前日期');  
 }  
}

此函数非常高效且通用，可处理多种不同格式日期，如'2018-08-01或2018/08/01或2018/8/1'。这样就可以动态生成入住日期距今有几天



### 模糊查询与精确查询

在退房、换房、还款界面的最上方都有一个搜索功能



其中，房间号只支持精确查询，sql语句就是最简单的select。而身份证号支持模糊查询，sql语句使用like：使得任何包含查询信息的身份证都会被显示

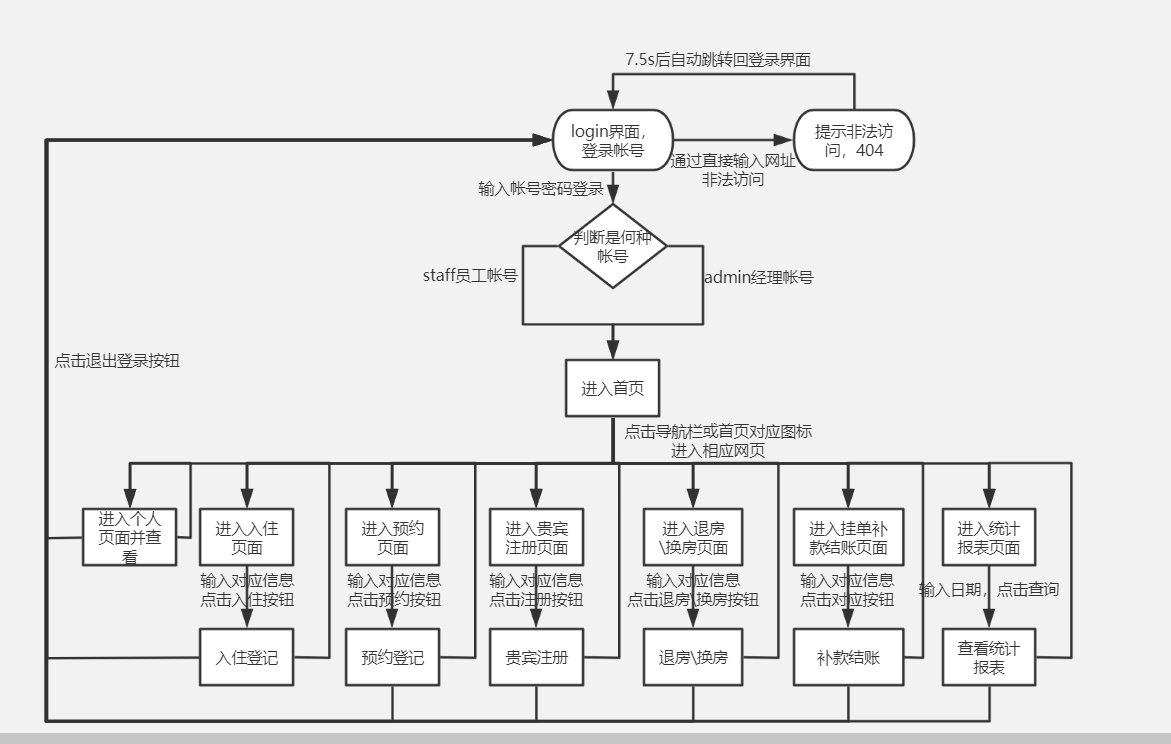
sql="select \* from Permanent where identitynumber like \'%"+id+"%\'";

在后端，同样进行了trim处理，id=id.trim();

通过这个方法，可以去除输入信息中的多余空格。

# 六、使用手册

**功能流程图如下：**

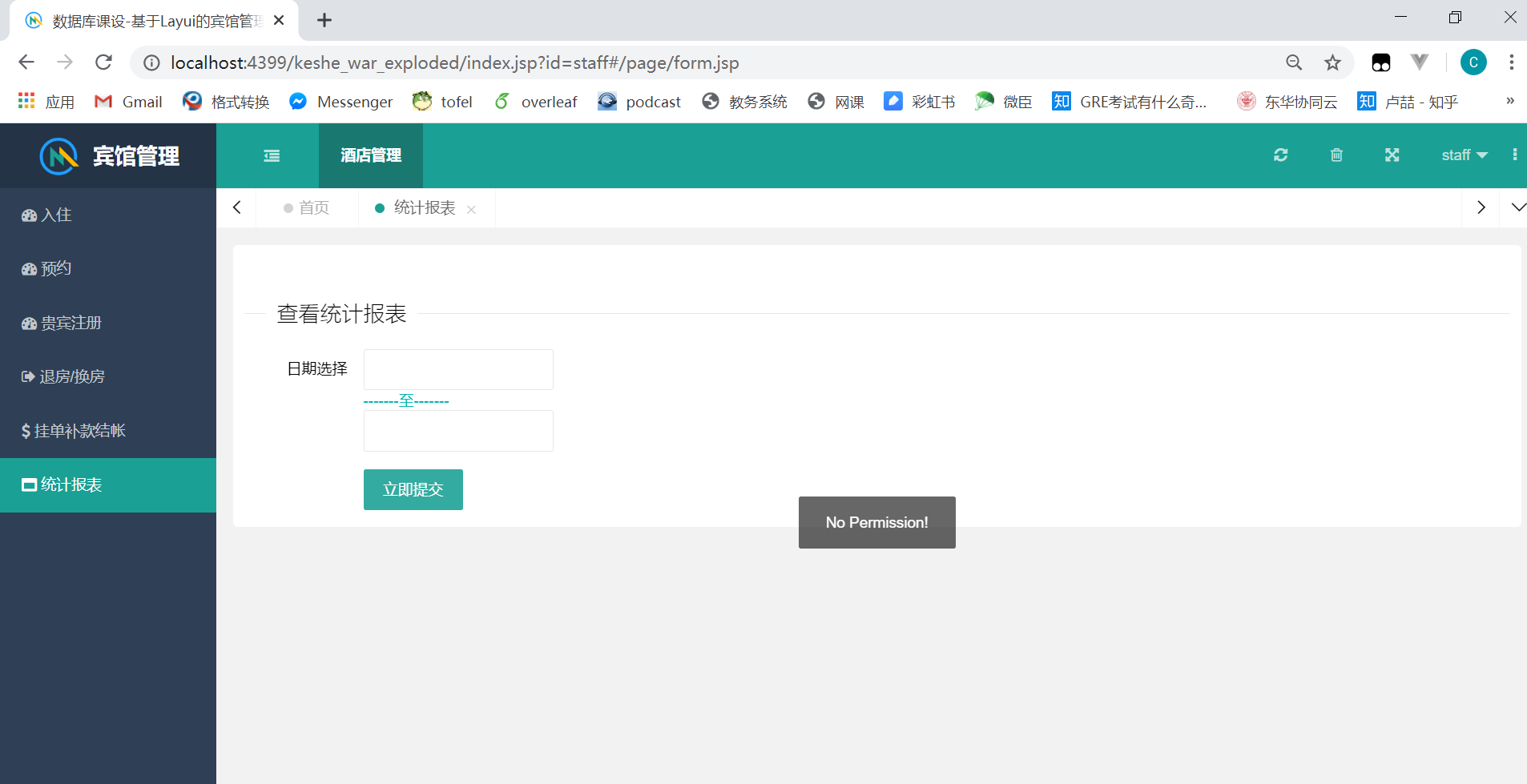


**登陆页面仅有两组密码：**

用户名：staff 密码：staff 登录身份：酒店员工

用户名：admin 密码：admin 登录身份：大堂经理

其中大堂经理可以查看统计报表，而酒店员工会提示没有权限。



**金钱结算规则：**

1. 首先获取房间原价
2. 如果是团体，享受10%折扣
3. 如果过房间中任何一名住户是vip，再享受10%折扣，可叠加
4. 折后价乘以入住日期至今的天数，即为最终应付金额

所有涉及到统计报表的操作都有以下逻辑：如果当日没有没记录，增加一条新记录。初始数值会因为不同情景各不相同，详细请参看代码。

**换房时的逻辑：**

将当前房间的住户的老房间号改成新房间号。老房间变成空闲，新房间变成有人，更新now和checkin表

**退房时的逻辑：**

1. 如果实付金额大于应付金额，退房失败。
2. 如果实付金额等于应付金额，提示正常退房，统计报表利润增加应付金额，统计报表当日的记录赊账不变。
3. 如果实付金额小于应付金额，提示赊账退房，统计报表利润增加实付金额，统计报表当日的记录赊账增加应付与实付的差值。

**还款时的逻辑：**

1. 如果还款额大于赊账金额，还款失败。
2. 如果还款额等于应付金额，提示正常退房，统计报表当日的记录利润增加还款金额，统计报表当日的记录赊账减少还款金额。

**重要提示：**由于所有操作仅能修改当日统计报表的值，当日赊账有可能为负数！ 但所有的赊账金额相加永远大于等于零。

统计报表每天的赊账金额累加的意义：目前还有多少赊账未还

统计报表每天的利润金额累加的意义：目前赚到的利润

统计报表每天的利润金额与社长金额相加再累加的意义：理论上在未来可以赚到的净利润的总和（所有赊账都还上之后）

# 实验中遇到的问题

1. 所有jsp页面的开头都要加入以下两条语句：

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

//让页面可以显示中文  
<!DOCTYPE html>

第二条非常重要，将页面设置成html5，否则许多layui的css样式无法正常显示

1. 建议在所有弹窗关闭以及操作按钮结束后加上以下模板语句，刷新所有页面，这样才能反映出数据库值得变化

var iframeIndex = ***parent***.layer.getFrameIndex(***window***.name);  
***parent***.layer.close(iframeIndex);  
***parent***.location.reload();

如果过没有弹窗，那就将parent替换成window。

1. DOM中获取元素的不同的方式:

document.getElementById();

jQuery对象转DOM对象,前端获取dom元素得方法为:

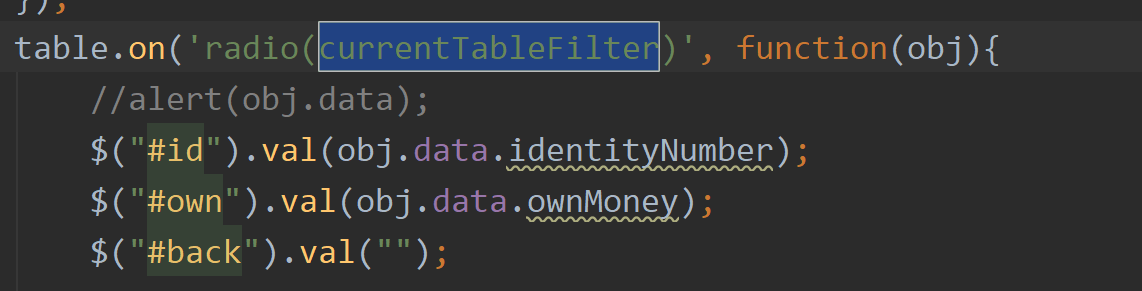
$('#btn')[0]或者: $('#btn').get(0);

btn即对象的对应id

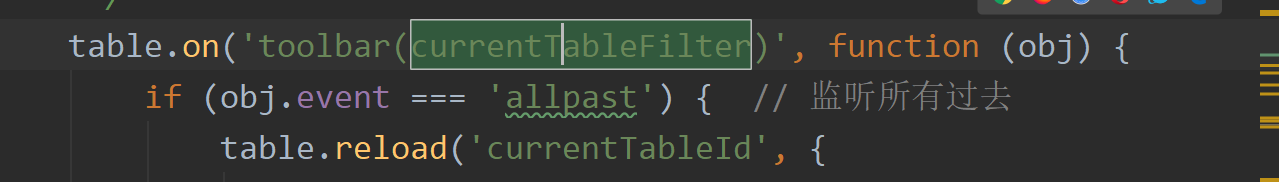
lay-filter是layui的事件过滤器。你可能会在很多地方看到他，他一般是用于监听特定的自定义事件。可以把它看作是一个ID选择器通过设置，可以监听同一组件的不同按钮，比如：

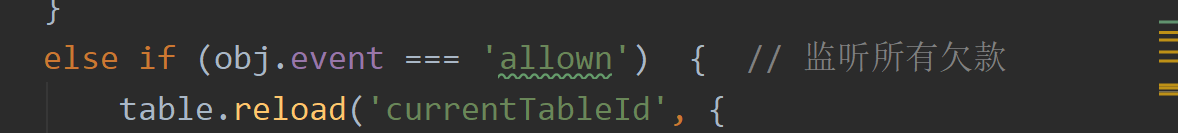
<table class="layui-hide" id="currentTableId" lay-filter="currentTableFilter"></table>

这里定义了当前table的lay-filter



这样，就可以监听这张table中每一行数据的头上的单选按钮radio，obj返回数据对象，其中包含改行的所有数据。



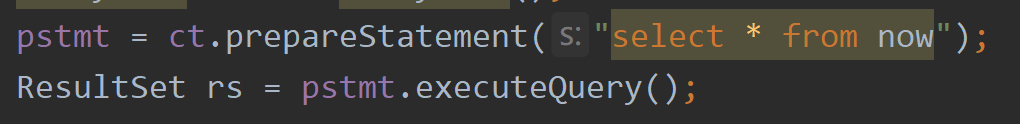


同样是这张table，我们通过lay-filter还能监听它的工具栏toolbar，根据返回obj中不同的事件类型，进行不同的事件处理。

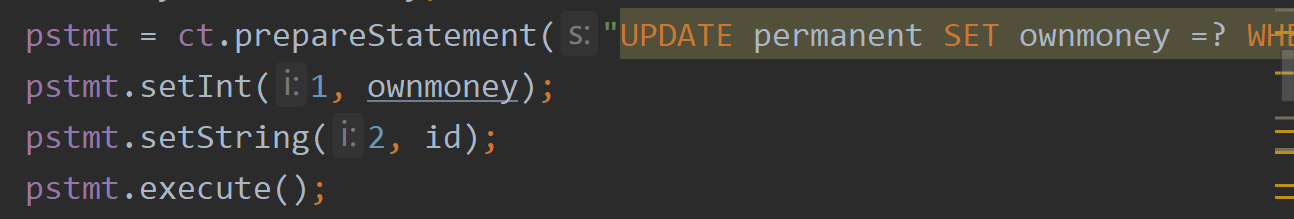
1. 前端调试多使用console.log。后端多使用System.out.println。新建的Servlet一定要在web.xml中添加配置。相对路径以及url最好的测试方法是直接在浏览器的url中输入报错网址。../代表退出一层文件夹，路径可直接在我的电脑的中输入路径并测试。
2. 由于mysql登陆密码不一样，所有的dao层代码都需要根据自己电脑上mysql的用户名密码以及创建的表进行配置。



1. Select语句使用executeQuery方法，并且有返回值



而update和insert语句使用execute方法，没有返回值



通过try catch，我们可以在sql语句执行错的情况下获取错误信息。

# 心得体会

耗时6周开发的宾馆管理系统是我大学开发过最复杂同时也是最完整的项目。一开始在网上阅读他人代码时的我简直一头雾水。遇到各种疑难杂症，花上整整一天，查遍国内外论坛都无法解决也让我焦躁烦乱，甚至一度奔溃。然而，最后所有的问题都被奇迹般地解决了，不仅实现了任务指导书中的所有功能要求，界面设计得也完美符合原先的预期。

前期的学习工作最为辛苦，需要进行大量的代码测试与debug，网上往往没有很好的解决方案。比如layui模板更改init.json的方法，我是在b站一个点击量不到500的视频中找到了解决方案。但是一旦学会整个前后端数据交互过程后，也就是做完样板网站的完整前后端后，制作其他页面就成了依葫芦画瓢，真可谓熟能生巧，开发进程也加快了很多。

通过数据库课程设计，我简单地制作了一个web应用系统，了解了前后端的全栈开发流程。但是在实际开发过程中，我也体会到了jquery与servlet体系相较时下热门前后端开发框架的劣势。比如vue，作为一个渐进式框架，vue不需要像html页面那样频繁的操作真实的dom元素，它本身都是虚拟dom跟数据进行捆绑，降低了性能消耗。同时其规模较小巧，运行速度快，开发者可以将注意力集中保持在核心库，因为有丰富的插件库。很可惜由于时间紧迫，加之我们组组员都是0 web开发基础的小白，考虑到vue的学习时间成本较高，我们最终选用了较常规的jquery。学习是终生的，今后我们必将运用本次课设获得的自学能力，进一步丰富我们的开发知识，制作更优秀的项目。

最后衷心感谢我的组员：汪杰与周欣宇同学。没有他们的帮助我无法一个人完成如此多的页面的编码。感谢他们在开发过程中与我一起查资料，商讨解决问题的方法，并最后在测试中提出了我的一些细小问题。是我们三人无数个日夜的zoom会议，积极交流，与共同努力，促成了项目最终的成功！