商 学 院

**课 程 设 计**

#### **设计题目：** 我我音乐

## 学 院：

#### 商学院

**班 级：** 16软件工程2班

## 学 号： 201604040222

**姓 名：**

**指 导 教 师：**

### 方志月

徐艺武

2 0 1 9 年 01 月 03 日

**目 录**

一级标题，黑体，三号，居中

1. [需求分析 1](#_bookmark0)
2. [总体设计 2](#_bookmark1)
   1. [开发环境介绍 2](#_bookmark2)
   2. [系统功能描述 2](#_bookmark2)
   3. [E-R模型 3](#_bookmark3)
   4. [系统功能模块划分 3](#_bookmark3)
   5. [系统流程图 3](#_bookmark3)
3. [数据库设计 4](#_bookmark4)
4. [详细设计与实现 4](#_bookmark4)
   1. [主要源代码 5](#_bookmark2)
   2. [系统运行界面 5](#_bookmark3)
5. [总结 6](#_bookmark4)

[参考文献 7](#_bookmark5)

第一章 需求分析

* 1. 功能需求

本系统的目的是实现在线音乐的播放以及实现收藏进行热门歌曲的推送

1. 热门歌曲

主页可以查看最近的热门歌曲,对热门的歌曲进行细分，歌曲的分类包括华语，欧美，日本，韩国等常用国家的歌曲，以及歌曲的信息

1. 热么艺人

查询歌手的相关信息，以及对应歌手发布的音乐条目，当然通过点击对应的条目还可以直接播放

1. 热门专辑

查询最近的热门的专辑的信息，点击进入热门专辑可以查看专辑的相关信息，作者，以及专辑内部的歌曲的相关信息

1. 登录收藏

实现登录对歌曲的收藏。

(5)推荐歌曲，推荐歌手，推荐歌单

系统通过你点击收藏的音乐，自动为你推荐相似曲风的歌曲

* 1. 性能需求

(1)Tomcat可以免安装，体积小，运行效率高，能很方便的在Windows和linux系统下发布。同时使用mysql数据库，安装方便且免费使用，mysql在Windows和linux下都能被良好的支持。而JSP不但可以使用eclipse，也可使用dreamweaver编辑，甚至记事本等文本编辑工具即可编写，因此对系统运行环境非常低，小型服务器和家用电脑均作为本系统运行环境。

1.3 界面设计要求

(1)界面设计友好美观、数据要准确安全可靠；实现对新闻信息的添加、修改、删除，便于更新网站内容；系统最大限度地实现易维护性和易操作性；对于一个新闻发布系统而言新闻信息是很多的，而且使用的人数较多，所以对系统的安全性有着比较高的要求，对于数据库，要设置不同用户的权限，数据的修改必须由合法用户操作。

(2)网站页面是用户直接访问的页面，所以必须要让用户访问就能看出来这个网站的主题思想是什么，一下就能体现出这个网站的中心内容。并且首页的模块之间要联系紧密，更好的让用户理解明白。

1.4 安全需求

由于使用JSP编写的程序代码隐蔽，在客户端仅可以看到由JSP输出的动态HTML文件，因此只要编写程序时注意严谨性，防止SQL注入和cookies欺骗，整个系统的安全就能得到保障，同时也能防止别人抄袭源代码。本系统的管理员可以有多名，为防止暴力猜解密码，在设计后台登录模块时需考虑验证码的设计。

第二章 总体设计

高效的音乐软件平台可简便维护工作，使用我我音乐平台，用户可以听各种音乐，以及实现对音乐的智能推送。而对于前台界面，为适应多个用户的需求，必须达到友好、方便、简洁和易用的特点。本文将根据前期的需求分析，对我我音乐进行总体设计。

2.1 系统功能描述

1．用户管理：

(1)用户的注册；

(2)用户的登录；

(3)用户的收藏歌曲；

2．新歌收藏：

(1)对歌曲进行收藏；

(2)对专辑进行收藏；

(3)对歌手进行收藏(关注)；

4．最新信息推送：

(1)推送最新歌曲；

(2)推送最新歌单；

(3)推送热门歌手。

2.2 E-R模型

根据分析可划分出新闻信息、用户信息、评论信息以及新闻信息类别和几个实体，具体E-R模型如下。

歌手信息

出身地

作品

名称

其他

工作

描述

英文名

国家

图片

图2-1 歌单实体的E-R图

发布时间

歌单信息

名称

描述

图片

分类类型

公司

作者

类型

语言

播放量

图2-1 用户实体的E-R图

用户信息

密码

用户名

图2-1 用户实体的E-R图

收藏信息表

用户名

音乐id

2.3 系统功能模块划分

从功能的内容可以看到，该平台可以实现5个完整的功能。我们根据这些功能，设计出系统的功能模块。

我我音乐

热门推荐

收藏

注册

登录

热门艺人

热门专辑

热门歌曲

图2-5 系统功能模块图

新闻发布及管理系统的功能模块之间的关系：

前台系统

后台系统

音乐平台的操作

基本歌曲分类发布

各模块的数据显示

热门推送

音乐系统管理

数据库

2.6 开发环境

操作系统：Linux；

开发工具：Eclipse；

开发语言：Java；

开发技术：JSP+Servlet+ajax+javaScript+jquery;

数据库：MySQL；

第三章 数据库设计

3.1概念设计

我我平台中，主要需要以下数据来保持相应的信息：

1．歌手信息表：用于保存歌手的基本信息

2．分类信息表：用于保存歌曲的分类信息

3．专辑信息表：用于保存最新的专辑信息

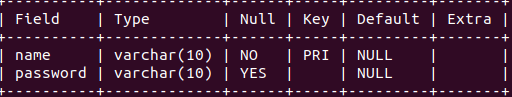
4. 用户信息表：保存用户的相关信息

5. 收藏表:用户收藏的相关信息

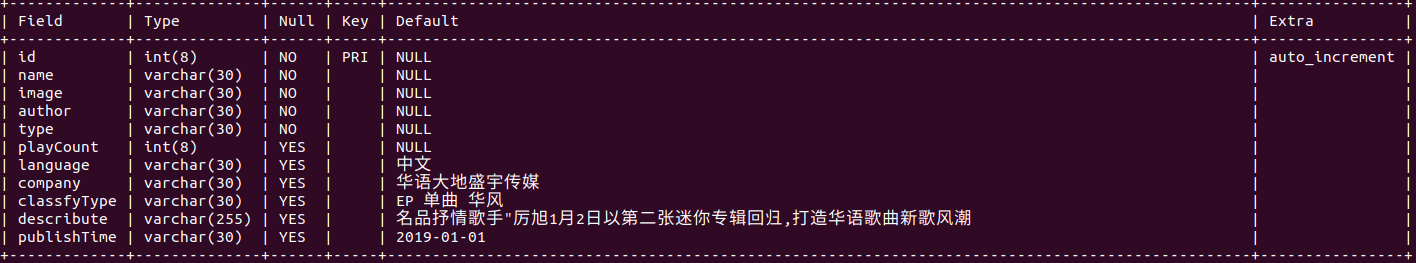
3.3物理设计

本系统主要涉及六张表。我们采用了MySql数据库,在数据库中使用数据表来储存现有的信息。

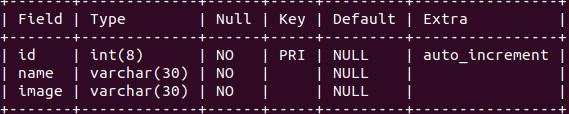
1. 用户表



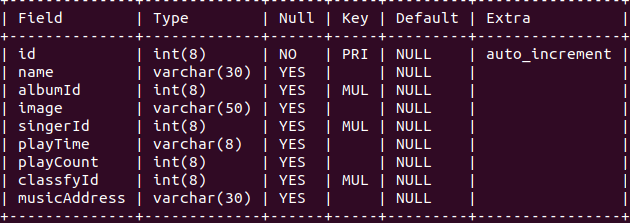
1. 专辑表



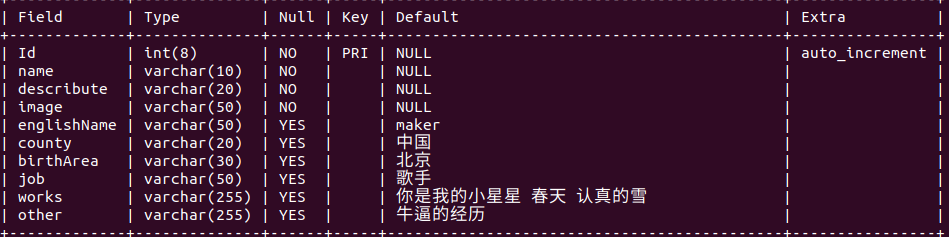
1. 分类表



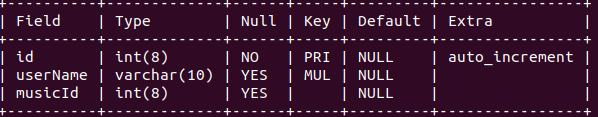
1. 音乐表



5.歌手表



1. 收藏表



第四章 详细设计与实现

4.1前台界面和功能实现

**4.1.1主页功能介绍**

主页采用的技术是ajax技术，http://localhost:8080/music/MusicProject/html/index.html

热门歌曲，热门艺人，新歌分类等相关信息都是通过ajax技术实现的数据传输，传输的格式是json，通过

Javascript的 JSON类实现string类型json格式字符串的转换。以及通过javascript实现内容的填充，考虑源码过多，下面展示主要的代码

**var** onLoadUserData = **function** () {// 登录的检测

**var** url1 = "?loginStatus=false";

sendAjax("http://localhost:8080/music/UserLoginAction" + url1,

**function** (response) {

**var** user = JSON.parse(response);//传递的是user对象

**if** (user == **null** || user.name == **null**) {

} **else** {

setLoginStatus(**true**, user.name);

}

});

};

**function** sendAjax(url, response) {//ajax源码

**var** xhr;

**if** (window.XMLHttpRequest) {

xhr = **new** XMLHttpRequest();

} **else** {

xhr = **new** ActiveXObject("Microsoft.XMLHTTP");

}

xhr.onreadystatechange = **function** () {

**if** (xhr.readyState == 4 && xhr.status == 200) {

response(xhr.responseText);

}

}

xhr.open("GET", url, **true**);

xhr.send();

}

<!--创建导航条-->

<div style="overflow: *hidden*" id=*"navigateBarOut"*></div>

<script src=*"../template/mode/model.js"*></script>

<script src=*"../template/navigateBar.js"*></script>

<link rel=*"stylesheet"* href=*"../template/css/navigateBar.css"*>

<script>

**var** navigateBarOut = document.getElementById("navigateBarOut")

navigateBarOut.style.marginBottom = "20px"

navigateBarOut.innerHTML = createView;

createNavigate(**null**, [

**new** ClickItem("爱看", "#"), **new** ClickItem("我的音乐", "myMusic.html")

], "login.html", "register.html")

</script>

<!--banner-->

<div style="overflow: *hidden*" id=*"banner"*></div>

<script src=*"../template/banner.js"*></script>

<link rel=*"stylesheet"* href=*"../template/css/banner.css"*>

<script>

document.getElementById("banner").innerHTML = containerView;

createBanner();

</script>

//上面是导航条和banner的实现，通过将具体的html模块封装实现代码的简化，例如上面的banner，通过引入相应的js，css，在ajax请求结束后，将数据传递过去，会自动生成html代码

<!--推荐歌单-->

<div style="overflow: *hidden*" id=*"tuijian"*></div>

<!--推荐的数据-->

<script src=*"../template/tuijian.js"*></script>

<link rel=*"stylesheet"* href=*"../template/css/tuijian.css"*>

<script>

</script>

<!--新歌-->

<div class=*"newMusicContainer"* style="width: *970px*;">

<div class=*"newMusicHead"*>

<span class=*"title"*>新歌</span>

<ul class=*"newMusicHeadUl"*>

<li>华语</li>

<li>欧美</li>

<li>日本</li>

<li>韩国</li>

</ul>

</div>

<div>

<div class=*"leftImage"*>

<img src=*"../images/album.jpg"* class=*"albumImg"*/>

</div>

<div class=*"rightMusicList"*>

<div class=*"musicListHead"* style="">

<img src=*"../images/播放.svg"* class=*"leftimg"* onclick="changeLocation('play.html')"/>

<span style="position: *relative*;top: *-4px*;font-size: *14px*">播放全部</span>

<div class=*"rightImg"*>

<img src=*"../images/downstep2.svg"* onclick="changeLocation('allSinger.html')">

</div>

</div>

<div class=*"musicListUlContainer"* style="overflow: *hidden*">

</div>

</div>

</div>

</div>

<!--热门艺人-->

<div style="overflow: *hidden*" id=*"star"*></div>

<script>

</script>

<!--版权模块-->

<div class=*"copyright"*>

<ul>

<li>关于虾米</li>

<li>关于我们</li>

<li>虾米招聘</li>

<li>联系我们</li>

<li>知识产权声明</li>

<li>联系客服</li>

<li>意见反馈</li>

</ul>

<ul>

<li>入职虾米</li>

<li>音乐人</li>

<li>音乐专栏</li>

</ul>

</div>

<!--这个是播放的列表-->

<div style="width: *100%*;

height: *50px*;background-image: *url('../images/bg.jpg')*;

position:*fixed*;z-index: *15*;bottom:*0px*; left:*0px*;">

<div style="width: *1000px*;overflow: *hidden*;margin: *0 auto*;position: *relative*;top: *5px*;

background: *#111111*;color: *#ffffff*;">

<audio controls style="width: *500px*;" id=*"audioResource"*>

</audio>

<img src=*"../images/上一页.svg"*/>

<img src=*"../images/下一页.svg"* style="position: *relative*;top: *-2px*;"/>

<div style="display: *inline-block*">

<img src=*""* id=*"playingMusic"* height=*"30px"*/>

<ul>

<li><span id=*"singerName"*></span></li>

<li><span id=*"musicName"*></span></li>

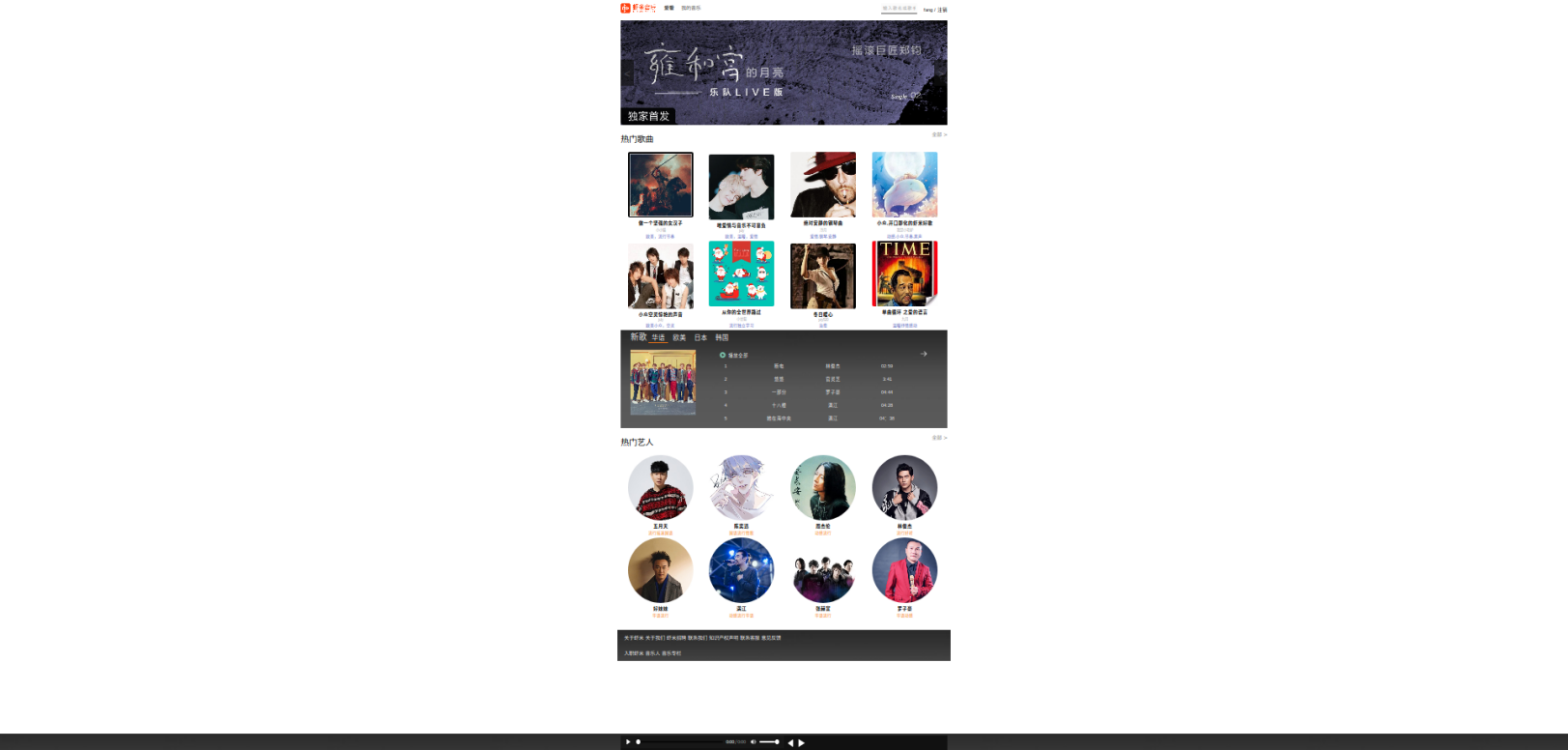
</ul>

</div>

</div>

</div>

<!--这个是具体的网页布局代码 下面是界面显示 有点像手机软件界面 下面黑的是播放条-->



点击我的音乐会进入我的音乐，当然按照逻辑需要实现session维持，我们的用户登录的信息使用的是session维持的，下面到login.html界面去看看

**function** change() {

**var** submit = document.getElementById("submit");

**var** user = document.getElementById("user")

**var** password = document.getElementById("password");

submit.onclick = **function** (ev) {

**if** (user.value.length == 0 || password.value.length == 0) {

alert("请输入有效数据")

**return**;

}

**var** param = "?user=" + user.value + "&password=" + password.value;

sendAjax("http://localhost:8080/music/UserLoginAction" + param, **function** (response) {

**var** responseObj = JSON.parse(response);

**if** (responseObj.responseCode == 0) {

location.href = "http://localhost:8080/music/MusicProject/html/index.html";

} **else** {

alert("登录名或密码错误");

}

});

}

上面的代码实现的是登录的功能，为了实现异步登录，也就是不是那种登录跳来跳去的效果，我们首推ajax技术实现

<div>

<h1>登录界面</h1>

<form method=*"get"*>

<table>

<tr>

<td>用户名</td>

<td><input type=*"text"* name=*"user"* placeholder=*"输入用户名"* id=*"user"*></td>

</tr>

<tr>

<td>密码</td>

<td><input type=*"text"* name=*"password"* placeholder=*"输入密码"* id=*"password"*></td>

</tr>

<tr>

<td></td>

<td><input type=*"button"* value=*"提交"* id=*"submit"*></td>

</tr>

</table>

</form>

</div>

<!--上面的代码实现的就是登录界面的显示，时间很赶，界面直接用了table实现 看到了吗 method使用的是get

原因还是懒，只实现了ajax的get方法

-->

就是这么实现的，没有任何东西 logo都没有

下面是我的音乐的显示界面

**var** server="http://localhost:8080/music"

**function** loadDataReal(text) {

**var** url1 = url + "?theme=" + text;

sendAjax(url1, **function** (response) {

**var** responseObj = JSON.parse(response);

**if** (responseObj.responseCode != **null**) {

location.href = "login.html";

} **else** {

//代表已经登录请求的是对应的数据

**var** musicList = responseObj;

// alert(response)

**var** musicCollection = document.getElementById("musicCollection");

**var** musicSingle = musicList[0];

**if** ("author" **in** musicSingle) {

musicCollection.innerHTML = albumTableHead;

} **else** **if** ("describute" **in** musicSingle) {//代表加载的是singer数据

musicCollection.innerHTML = singleTableHead;

} **else** {

musicCollection.innerHTML = tableHead;

}

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

**var** musicSingle = musicList[i];

**var** inner;

**if** ("author" **in** musicSingle) {//加载的是歌单的数据

inner = " <tr class=\"singleMusicData\">\n" +

" <td>\n" +

" <span style=\"position: relative;top: -2px;margin-right: 10px\">1</span>\n" +

" <img src=\"../images/play.svg\" width=\"20px\" style=\"position: relative;top: 2px;\"/>\n" +

" </td>\n" +

" <td class=\"one\">" + musicSingle.name + "</td>\n" +

" <td class=\"one\">" + musicSingle.author + "</td>\n" +

" <td class=\"one\">" + musicSingle.type + "</td>\n" +

" <td class=\"one\">" + musicSingle.playCount + "</td>\n" +

" </tr>";

} **else** **if** ("describute" **in** musicSingle) {//代表加载的是singer数据

inner = " <tr class=\"singleMusicData\">\n" +

" <td>\n" +

" <span style=\"position: relative;top: -2px;margin-right: 10px\">1</span>\n" +

" <img src=\"../images/play.svg\" width=\"20px\" style=\"position: relative;top: 2px;\"/>\n" +

" </td>\n" +

" <td class=\"one\">" + musicSingle.name + "</td>\n" +

" <td class=\"one\">" + musicSingle.describute + "</td>\n" +

" </tr>";

} **else** {

inner = " <tr class=\"singleMusicData\">\n" +

" <td>\n" +

" <span style=\"position: relative;top: -2px;margin-right: 10px\">1</span>\n" +

" <img src=\"../images/play.svg\" width=\"20px\" style=\"position: relative;top: 2px;\"/>\n" +

" </td>\n" +

" <td class=\"one\">" + musicSingle.musicName + "</td>\n" +

" <td class=\"one\">" + musicSingle.playTime + "</td>\n" +

" <td class=\"one\">" + musicSingle.singerName + "</td>\n" +

" <td class=\"one\">" + musicSingle.albumName + "</td>\n" +

" <td class=\"one\" style='display:none'>" + musicSingle.musicAddress + "</td>\n" +

" </tr>";

}

musicCollection.innerHTML += inner;

}

**var** trSet = document.getElementsByClassName("singleMusicData");

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

trSet[i].onclick = **function** () {

//在这里去查询歌曲的地址，然后直接赋值给播放的audio进行播放

**var** tdSet=**this**.getElementsByTagName("td");

**if**(tdSet.length==6){//代表加载的是歌曲或者是推荐的歌曲

**var** mn=tdSet[1].innerText;

**var** sn=tdSet[3].innerText;

**var** musicAddress=tdSet[5].innerText;

**var** musicName = document.getElementById("musicName");

**var** singerName = document.getElementById("singerName");

**var** source=document.getElementById("audioResource");

musicName.innerText = mn;

singerName.innerText =sn;

source.src=server+musicAddress;

source.play();

}

}

trSet[i].onmouseover = **function** (ev) {

**this**.style.backgroundColor = "#ccc";

}

trSet[i].onmouseout = **function** (ev) {

**this**.style.backgroundColor = "";

}

}

}

});

}

上面的代码实现的就是加载数据的代码，同样需要ajax，当然前面已经写过了，不在重复

<div class=*"container"*>

<!--左边的div，用于显示操作列表-->

<div class=*"leftMenu"*>

<ul>

<li>收藏的歌曲</li>

<li>热门推荐</li>

<li>推荐歌手</li>

<li>推荐歌单</li>

</ul>

</div>

<!--右边的div，用于显示最新的推荐-->

<div class=*"rightList"*>

<div class=*"rightListTop"* style="margin-left: *100px*">

<img src=*"../images/like.png"* style="float: *left*;margin-right: *40px*"/>

<div style="float: *left*">

<img src=*"../images/tag.svg"*/>

<span style="font-size: *26px*">我喜欢的音乐</span>

<div>

<img src=*"../images/user.jpg"* height=*"30px"*/>

<span>方方方</span>

</div>

<div>

<button>播放</button>

<button>收藏</button>

<button>分享</button>

<button>下载</button>

<button>评论</button>

</div>

</div>

</div>

<!---下面是菜单的显示-->

<div style="clear: *both*;margin-left: *50px*;padding-top: *40px*">

<span style="font-size: *23px*">歌曲列表</span>

<span style="color: *#9a9a9c*;font-size: *14px*;">345首歌曲</span>

<span style="float: *right*;color: *red*;font-size: *14px*">播放量：5757</span>

<hr style="background-color: *red*"/>

<table style="width: *600px*;margin-top: *20px*" id=*"musicCollection"*>

<tr>

<td></td>

<td>歌曲标题</td>

<td>时长</td>

<td>歌手</td>

<td>专辑</td>

</tr>

</table>

</div>

</div>

</div>

就这几行代码实现的，主要在css上面，这里主要讲解业务逻辑，不在对具体的修饰进行说明



下面开始讲解专辑界面这个界面是albumDetail.jsp是个jsp界面

<sql:setDataSource driver=*"com.mysql.jdbc.Driver"* user=*"root"* password=*"123"* url=*"jdbc:mysql://localhost:3306/music"*/>

<sql:query var=*"albums"* >

select music.id, music.name musicName, s.name singerName, music.playTime,music.musicAddress, a.\*

from music

join album a on music.albumId = a.id

join singer s on music.singerId = s.Id

where albumId = ?

<sql:param value=*"*${param.albumId}*"*></sql:param>

</sql:query>

通过数据库标签查询数据，之所以不在里面编写 脚本，原因是不好维护，前端显示应该与具体的业务逻辑分离，所以我们没有使用该技术实现，而是使用sql语句，一般来说，没有比较复杂的业务逻辑，就可以使用该技术实现，对于复杂的业务逻辑，一般使用servlet来进行转发，通过session维持http无状态的状况，(仅供参考)

<div style="width: *1000px*;margin: *0 auto*;">

<div class=*"leftContainer"* style="float: *left*;">

<img src=*"http://localhost:8080/music*${albums.rows[0].image}*"* width=*"240px"* style="display: *block*" class=*"albumImage"*/>

<ul>

<li>专辑语种 <span>${albums.rows[0].language}</span></li>

<li>唱片公司 <span>${albums.rows[0].company}</span></li>

<li>专辑类别 <span>${albums.rows[0].classfyType}</span></li>

</ul>

<h3 style="margin-top: *20px*;margin-bottom: *20px*">专辑介绍</h3>

<p style="margin-bottom: *20px*">${albums.rows[0].describute}</p>

<p><a href=*"#"*>更多</a></p>

</div>

<div class=*"rightContainer"* style="float: *left*;">

<img src=*"http://localhost:8080/music*${ albums.rows[0].image}*"* width=*"40px"* style="border-radius: *3px*"/>

<span style="margin-left: *10px*;font-size: *13px*;color: *#9a9a9c*;">${albums.rows[0].publishTime} 发行</span>

<h2 style="margin-top: *10px*;margin-bottom: *10px*">${albums.rows[0].name}</h2>

<h6 style="color: *#6f6f71*;font-size: *14px*;margin-bottom: *10px*">${albums.rows[0].type}</h6>

<div class=*"oprateSet"*>

<button>全部播放</button>

<button>收藏<span>3245</span></button>

<button>分享</button>

<button>手机试听</button>

<button>更多&gt;</button>

</div>

<ul class=*"albumDes"*>

<li><span style="">${albums.rowCount}</span> 首歌</li>

<li>歌曲</li>

<li>歌手</li>

<li>时长</li>

</ul>

<h3 style="margin-top: *40px*;margin-bottom: *30px*">Disc 1</h3>

<table style="width: *600px*;">

<c:forEach items=*"*${albums.rows}*"* var=*"album"*>

<tr style="height: *40px*;" class=*"albumListMusic"*>

<td >${album.musicName}</td>

<td>${album.singerName}</td>

<td>${album.playTime}</td>

<td style="width:*0px*;display: *none*;">${album.musicAddress}</td>

</tr>

</c:forEach>

</table>

</div>

</div>

上面就是具体的html代码，使用了el表达式实现，以及jstl标签库 下面贴下照骗





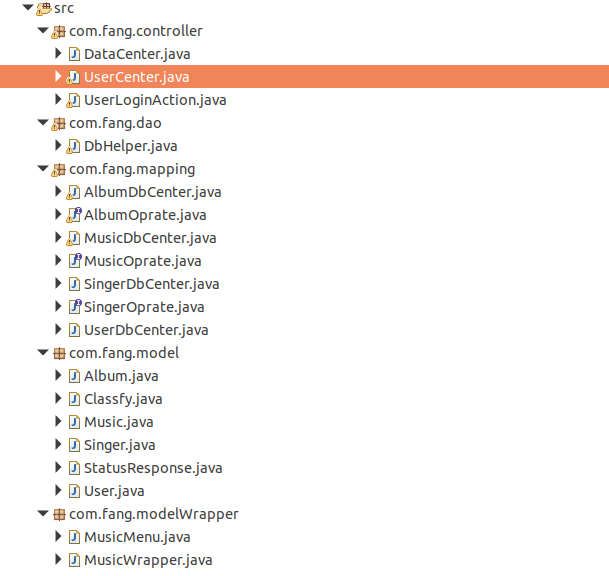
下面是歌手的主页的显示界面.singerDetail.jsp,其实实现的技术和专辑的实现是一样的，就是显示不太一样

，就贴下界面显示效果,里面的数据是从数据库拿的，由于时间问题，就录的默认值，并非死的html实现



下面由界面显示进入业务逻辑层的设计介绍

先看工程结构



1.DataCenter数据的集中处理类servlet(index.html获得信息类)

2.UserCenter用户收藏以及智能推送的类

3.UserLoginAction用户处理类 (登录,检查登录状态)

1. DbHelper:数据库的底层的连接类
2. AlbumDbCenter:专辑的数据处理类(AlbumOprate是接口，其他同样，是将操作封装的接口)
3. MusicDbCenter:音乐的数据处理类

....其他同样

Model下面都是一些javabean,不在讲解

modelWrapper:bean的封装类，解决数据库联结查询的bean问题

总结

经过一个星期的课程设计我我音乐基本功能完成，但是功能不是很齐，界面也不是很好看。那是因为，一个星期的时间有点不够，时间显得十分紧凑，但我们一直努力对待，争取使平台的功能更加完善。本平台基本能够完成听歌收藏，智能推送，热门专辑，艺人 ，歌曲分类。

我们开发的这个平台还存在许多漏洞和不足，由于时间和技术问题，还得以后继续改进，希望以后能把这个系统完善。

经过此次课程设计，我们发现要用到许多我们上课讲到的知识，还有些上课没讲到的，要通过自己上网搜索或请教别人学习的知识点。把之前学的知识用了一遍后不紧活学活用，而且增加了自信，加强了我们解决问题的能力。特别是要实现一个功能时不是一下子就能完成的，期间会遇到许多困难，会感到困惑，但是不要紧，只要一心想实现那个功能，通过自己的努力实现了此功能，会感到无比舒畅与自信。这对我们以后面对各种困难打好了应对基础。因此，这次课程设计对我们的意义很大。

致谢

此次课设我们非常感谢我们的指导老师徐艺武，是有了老师严格要求我们必须认真完成此次课程设计，我们有了压力和学习的动力去完成此次课程设计。

徐老师在专业上能力很强，最主要的是懂得如何让我们学到东西。他常说不管做什么，目的只是为了让你们学到东西。我们上过徐老师课程，每次都觉得会学到很多知识，并且学到许多学习方法。比如说，徐老师经常会给我们拿一段代码来讲，告诉我们他是如何一点一点的分析和看懂代码的。我们以后也会按照徐老师的方法继续学习新的知识，让自己的专业能力提升。