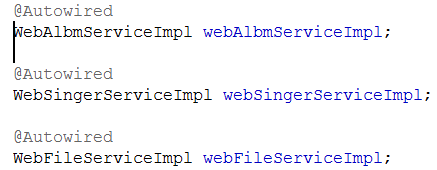
专辑管理-详细设计文档

# 一．专辑管理控制器-WebAlbmController类

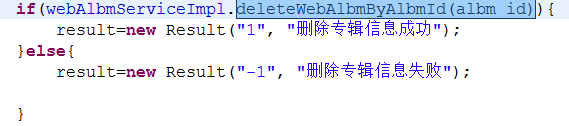
1. 使用@Controller定义一个Controller，使用@RequestMapping来映射请求
2. 定义服务，便于调用歌手管理，专辑管理、下载管理里面的方法，利用@Autowired是用在加载实现service实现类



1. 在WebAlbmController类里面书写控制方法

## （一）删除专辑-deleteWebAlbm()

1. 利用@RequestMapping 是一个用来处理请求地址映射的注解
2. @Responsebody 注解表示该方法的返回的结果直接写入HTTP响应正文（ResponseBody）中
3. 在deleteWebAlbm()方法中利用@RequestParam传递参数albm\_id
4. 方法中创建并且初始化一个result实体对象
5. 书写一个if判断语句，获取到webAlbmServiceImpl中的deleteWebAlbmByAlbmId(albm\_id)方法



获取到删除方法，若返回1，提示显示成功；若返回-1，提示显示失败。

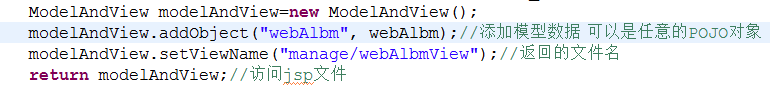
1. 最后返回结果，信息显示。

## （二）用户专辑列表显示-viewWebAlbmDetail()

1. 利用@RequestMapping({"viewWebAlbmDetail"})处理请求地址映射
2. HttpServletRequest对象代表客户端的请求，当客户端通过HTTP协议访问服务器时，HTTP请求头中的所有信息都封装在这个对象中，开发人员通过这个对象的方法，可以获得客户这些信息。
3. HttpServletResponse对象代表服务器的响应。这个对象中封装了向客户端发送数据、发送响应头，发送响应状态码的方法。
4. 在viewWebAlbmDetail()方法中，通过webFileServiceImpl方法类中的getWebFileListByPkId(albm\_id)获取到专辑数据列表信息
5. 创建且实体化一个webAlbm实体
6. 构造ModelAndView对象(ModelAndView向前台页面传值)
7. 获取到webAlbmServiceImpl类中的getWebAlbmByAlbmId(albm\_id)给webAlbm赋值
8. addObject()有几个不同参数的方法，可以默认和指定返回对象的名字
9. 通过setViewName()方法跳转到指定的页面webuser中viewAlbmDetail.Jsp

## （三）专辑详情（管理员界面）- viewWebAlbm()

1. 创建且初始化一个webAlbm实体
2. 获取到webAlbmServiceImpl方法类中的getWebAlbmByAlbmId（）中的webAlbmDaoImpl中的getWebAlbmByAlbmId的具体DAO方法，查询信息。
3. 后台利用ModelAndView向前台页面传值



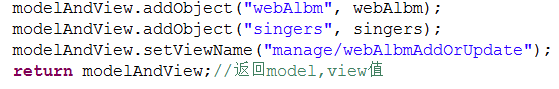
1. 构造ModelAndView对象
2. modelAndView.addObject添加模型数据
3. modelAndView.setViewName跳转到指定的页面manage文件里webAlbmView.jsp
4. 最后返回jsp页面信息

## （四）专辑添加和修改（管理员）

## -toWebAlbmAddOrUpdate(),-WebSingerAddOrUpdateSubmit()

### 〈一〉：-toWebAlbmAddOrUpdate()

1. 利用webSingerServiceImpl方法类中getAllSingers()方法中的webSingerDaoImpl方法类中getAllSingers()方法构造一个歌手实体列表singers
2. 创建且初始化一个webAlbm实体
3. getParameter它通过albm\_id或取参数的方法。把jsp文件中的数据读取到出来。然后就可以封装利用起来
4. 执行一个if判断语句albm\_id!=null，则利用DAO方法数据库操作通过albm\_id获取webAlbm对象
5. 后台利用ModelAndView向前台页面传值



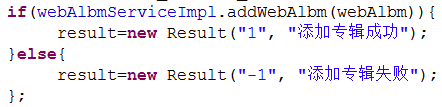
1. 构造ModelAndView对象
2. modelAndView.addObject添加模型数据
3. modelAndView.setViewName跳转到指定的页面
4. 返回视图webAlbmAddOrUpdate.jsp

### <二>：-WebSingerAddOrUpdateSubmit()

1. 创建且初始化一个result实体对象
2. 获取model类里面的webAlbm类的id
3. 利用判断语句判断添加专辑还是修改专辑
4. 如果albm\_id为空，添加专辑

利用UUID.randomUUID().toString()自动生成主键；

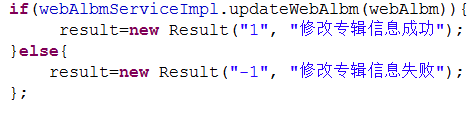
利用DAO方法中数据库操作，添加信息



返回“1”，添加专辑成功；返回“-1”，添加专辑失败。

1. 反之，albm\_id不为空，修改专缉

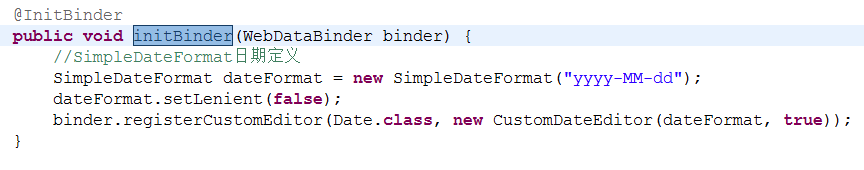
判断



返回“1”，修改专辑信息成功；返回“-1”，修改专辑信息失败。

## （五）解决前端日期类型传递

时间属性编辑器，时间类型直接传入会失败，需要利用initBinder进行类型转换



1. SimpleDateFormat日期定义，设置日期格式
2. 严格限制时间转换
3. 注册一个时间编辑器

# 二．Service-WebAlbmService

# Service- WebAlbmServiceImpl

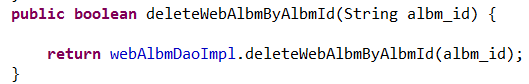
**功能： 歌曲信息管理的业务逻辑层**

**@Autowired是用在加载实现WebAlbmDaoImpl实现类**

****

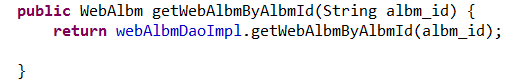
1. **删除专辑信息——deleteWebAlbmByAlbmId（）**

通过调用DAO层webAlbmDaoImpl里面的删除方法，并且使用albm\_id来判断到底删除哪一专辑的信息



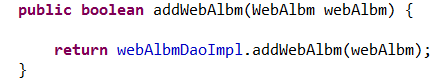
1. **得到专辑信息——getWebAlbmByAlbmId（）**

通过调用DAO层webAlbmDaoImpl里面的查找方法，并且使用albm\_id来判断到底查找哪一专辑的信息



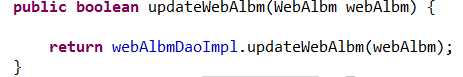
1. **添加专辑信息——addWebAlbm（）**

通过调用DAO层webAlbmDaoImpl里面的添加方法，并且使用albm\_id来判断到底是否有这专辑，再进行添加操作



1. **修改专辑信息—— updateWebAlbm（）**

通过调用DAO层webAlbmDaoImpl里面的修改方法，并且使用albm\_id来判断到底修改哪专辑的信息



# 三．DAO-WebAlbmdDao

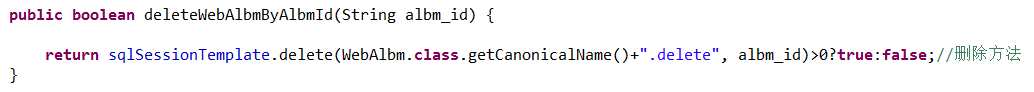
# DAO- WebAlbmDaoImpl

**功能：**通过Dao层封装实体类对象，进行数据库操作

通过SqlSessionTemplate基于模板类实现Mybatis，调用SqlSessionTemplate类访问数据库。

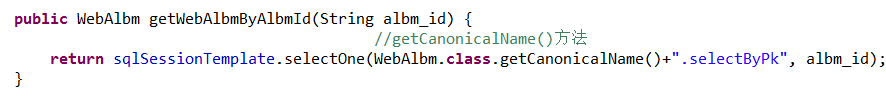
1. 删除专辑的信息——deleteWebAlbmByAlbmId（）

（1）通过albm\_id，去数据库中找到这首歌

（2）用sqlSessionTemplate去操作数据库 ，删除专辑的信息

1. 查询专辑的信息——getWebAlbmByAlbmId（）

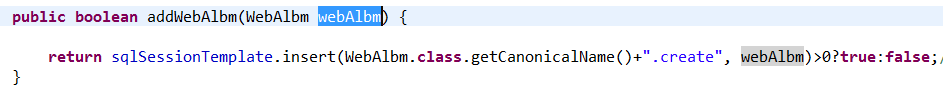
（1） 通过albm\_id，去数据库中找到这首歌

（2） 用sqlSessionTemplate去操作数据库 ，查询专辑的信息

1. 添加专辑的信息——addWebAlbm（）

（1） 通过调用WebAlbm中的方法体和参数

（2） 用sqlSessionTemplate去操作数据库 ，添加专辑的信息



1. 修改专辑的信息——updateWebAlbm（）

（1） 通过调用WebAlbm中的方法体和参数

（2） 用sqlSessionTemplate去操作数据库 ，修改专辑的信息

