


















实习生Javaweb 开发规范V1

命名规范









1.1 包及文件结构

```
dao---->数据持久层，放置数据存取相关代码  
service ---->业务逻辑层，放置逻辑处理以及实体类  
utils---->工具类，放置一些共用的工具类  
action---->访问控制层，放置 servlet、spring controller 相关代码
```

包名统一使用小写，并且包以com.epoint.+项目，后续以action,domain,service等功能类型分类，最后才是业务模块分类。点分隔符之间有且仅有一个自然语义的英语单词。action和html的文件结构必须一致。例如学生管理系统中有学生基本信息与成绩信息模块，则包名结构如下

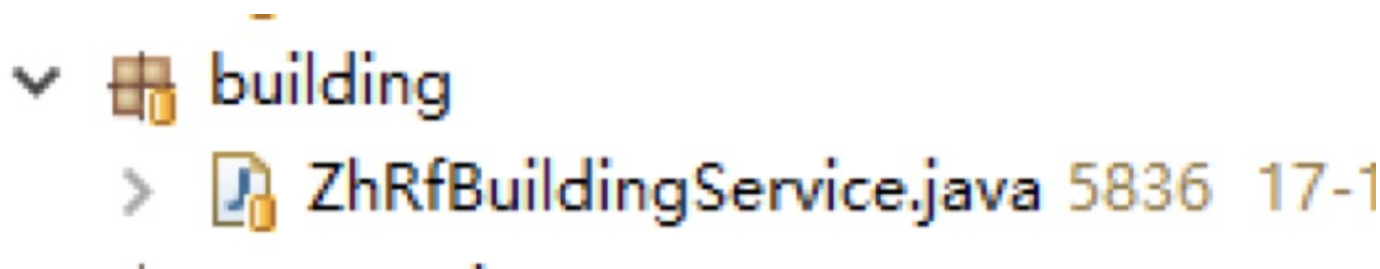
- ▼  com.epoint.studentmanage
 - ▼  action
 - ▼  markinfo
 - ▶  MarkInfoListAction.java
 - ▼  studentinfo
 - ▶  StudentInfoListAction.java
 - ▼  bizlogic
 - ▼  markinfo
 - ▼  domain
 - ▶  MarkInfo.java
 - ▶  MarkInfoService.java
 - ▼  studentinfo
 - ▼  domain
 - ▶  StudentInfo.java
 - ▶  StudentInfoService.java
 - ▼  utils
 - ▶  DBUtils.java

html 页面及文件结构必须与包名一致

- | | |
|--|--|
| ▼  category <ul style="list-style-type: none"> >  ZhSqBookCategoryAddAction.java 5944 17-2-4 上午 >  ZhSqBookCategoryDetailAction.java 5944 17-2-4 上午 >  ZhSqBookCategoryEditAction.java 5944 17-2-4 上午 >  ZhSqBookCategoryListAction.java 5944 17-2-4 上午 | ✓  category <ul style="list-style-type: none">  zhsqbookcategoryadd.html 58  zhsqbookcategorydetail.html  zhsqbookcategoryedit.html 58  zhsqbookcategorylist.html 58 |
|--|--|

ACTION的名称和页面名称一致

service一般以实体名称+Service组成



除 web.xml 文件以外其他配置文件统一放入 resources 目录，不允许将 jdbc.properties 文件放入 src 目录

webapp 目录结构与包名结构可类似，格式：项目名/模块/xxxlist.html 引用的资源文件例如 css、js、images 等，统一放入根目录，index.html 放入 webapp 根目录

1.2 类命名

类名以英文单词取名，首字母必须大写，多个英文单词以大写字母间隔，避免使用单词的缩写，除非它的缩写已经广为人知，如 HTTP。类名中不允许 '_'、'-' 等符号。1.3 属性、变量命名 a. 定义属性的位置，在类定义的开始，按照 `public, protected, package, private` 顺序放置。

b. 尽量用相同含义英文单词表示，不允许 '_'、'-' 等符号，如：custName。第一个字母小写，中间单词的第一个字母大写。不要用 _ 或 & 等符号作为第一个字母。单字符的变量名一般只用于生命期非常短暂的变量。如：i, j, k, m, n 一般用于 int。如果变量是集合，则变量名应用复数，即以小写 s 结尾。1.4 常量命名

所有常量名均全部大写，单词间以 '_' 隔开。例如：MAX_NUM 1.5 方法命名

方法命名采用“动作+属性”的方法。并且，动作以小写字母开始，属性以大写字母开始。常用的动作有：get、set、save、add、del 等。

编码规范

2.1 类及方法注释 2.1.1 类的注释 目的：主要标注该类的作用，对类进行简单描述；记录类的作者和类的创建日期

```
/**
 * @Description 客户信息实体类
 * @author 作者
 * @Date 2017年5月18日 下午4:58:07
 * @version 1.0.0
 */
public class Customer {}
```

2.1.2 方法的注释 目的：描述该方法的主要意义；对方法的参数进行说明

```
/**
 *
 * @Description 根据客户id, 查询客户信息
 * @param customerId 客户id
 * @return 客户信息
 * @throws Exception
 */
public Customer getCustomerInfo(String customerId) throws Exception {}
```

2.1.3 变量、属性的注释 目的：描述该属性的意义；对于属性有特殊含义的需要进行特别书面

```
/**
 * 用户姓名
 */
private String name;

或者

//用户姓名
private String name;
```

页面规范

3.1、通用

1. 标签必须以完整的标签对出现，例如

```
<input id="x" />
必须写成
<input id="x"></input>
```

可以规避非自闭合标签引起的html错误。（不强制）

2. 做隐藏控件时，一般做到元素初始状态是隐藏的。
3. 列表页面设置合理的显示条数，避免出现滚动条
4. 列表区域与toolbar 区域宽度应该保持一致
5. 列表及弹出也不允许拖拽调节表格尺寸

3.2 列表页

1. 页面必须至少有一个查询条件，查询输入域不能有数据校验。
2. 若表格的展示的列非常多，需要删去不重要的信息，只保留必要的信息。若展示的列很少，则需要尽量多显示一些信息，使列表看起来比较饱满。
3. 表格中，重要的信息放置于左侧，次要信息放置于右侧，操作按钮放置于末尾。
4. 表格中，非固定长度内容的列，须靠左对齐，固定按钮一律居中，多个按钮且数量不定须靠左。
5. 表格中，所有固定的列均需要列宽，但须保证至少有一列设置宽度为100%，其余宽度为合适的大小（无%符号）。列的设置原则是使表格看起来饱满而非拥挤。miniui列宽的分配原则，优先满足固定长度列，再根据剩余列按百分比占总值的比例再次分配。下表是一些常见的优先使用的列宽规范（部分情形下不适应，自行评估）
6. 所有控件必须使用miniUI 提供的方式，禁止采用 dom 方式完成

3.3 弹出页

1. 必填信息必须要有必填验证。
2. 必要信息放置在表单前面，多行输入框放到表单最后
3. 特殊类型必须要有校验，例如数字，身份证号码。
4. 非必选的下拉框，必须要有“请选择”
5. 所有的同类的弹出页面（一般情况是增改查）大小，布局（字段位置等），效果必须一致
6. 正常4列布局宽度设置为600，高度根据表单元素设置，确保最下方按钮与页边有一定的距离

3.3 js

1. 确保 js 代码中无不需要代码
2. function 命名确保与当前业务有意义
3. ajax请求url 中不允许中文传输，如需传输中文必须使用 post 方式提交

打包要求

1. maven 结构的项目在打包的时候仅需要 pom.xml、src 目录，其他 eclipse 生成的配置文件在打包工程的时候一律删除

其他要求

1. 数据库操作对象使用完毕以后必须关闭
2. 必须使用数据库分页，禁止一次读取内存分页
3. 如果表结构存在主表、子表，即使需求中没有提到在删除主表时应删除子表相关记录