

# 有关包的命名

---

## 意义

将类定义的进行分包，在一定意义上使得该类具有全球的唯一性，不与其他类进行冲突

## 分包

包可分为四到五层

- 第一层：甲方公司域名的倒序。如：com.jr
- 第二层：项目名称。
- 第三层：项目模块信息
- 第四层：功能顶层包。如：service, dao, beans
- 第五层：实现类层。如：imp。有些功能顶层包是没有实现类层的，所以有些项目包名知到四层

## 数据封装层的分类

### 1. 根据数据库表字段映射

- beans
- entity
- po (Persistence Object) 持久化对象

上面这些包中存放的类，一般具有id属性。因为它们在数据库中有对应的表

### 2. 根据实际业务需求数据映射

- vo (Value Object) 值对象。一般用于在类与页面间传值
- dto (Data Transfer Object) 数据传输对象。一般用于在类间传值

上面这些包中存放的类，一般是没有id属性的。因为他们不需要持久化到数据库

# XML选择本地DTD不走网络

---

设置XML不提示问题

## Log4j

---

### 配置文件详解

- log4j.rootLogger=[level], appenderName1, appenderName2.....

level: 是日志记录的优先级, 分为OFF, FATAL, ERROR, WARN, DEBUG

通过在这里定义的级别, 你可以控制到应用程序中相应级别的日志信息的开关。比如定义了INFO级别, 则应用程序中所有DEBUG级别的日志信息将不被打印出来。

appenderName: 就是制定日志信息输出到哪个地方。你可以同时指定多个输出目的地。例如:

log4j.rootLogger=info,A,B,C配置了三个输出地方, 这个名字可以随意 (如上面的A), 但必须与我们在后面进行的设置名字对应

- log4j.appender.A=[输出类型]

定义为A的输出端是哪种输出类型, 可以是

- log4j.appender.A.layout=[布局格式]

定义为A的输出端的layout是哪种类型, 可以是

- log4j.appender.A.layout,ConversionPattern=自定义模板

打印参数如下:

- log4j.appender.A.File=path

指定文件输出路径

- log4j.appender.A.Append=true | false

是否追加输入文件

- log4j.appender.A.Threshold=[level]

指定到该输出端的log日志级别, 假如为ERROR,那么当出现ERROR级别的日志就会被该定义的输出端输出到指定位置

## Web应用中的路径问题

---

### 访问路径与资源名称

通常的URL资源访问路径由两部分构成: **访问路径与资源名称**。

资源名称: 指的是访问资源的直接名称, 如show.jsp, 或与要访问资源存在映射关系的间接名称, 如show.do。

访问路径: 是通过该路径则可以定位到指定的资源, 即在URL资源访问路径中除了资源名称以为的其他部分。

访问路径	资源名称
<a href="http://127.0.0.1:8080/primary">http://127.0.0.1:8080/primary</a>	<a href="#">show.do</a>

根据访问路径是否可以独立完成资源准确定位的判断标准，可以将访问路径分为**绝对路径**与**相对路径**。

## 绝对路径

绝对路径，是指根据给出的访问路径可以准确定位到资源的路径。例如，你要告诉对方你现在的位置，你说：我现在在北京故宫游客A入口处。这就是个绝对路径，听者根据你所描述地址，可以准确的找到你。

而对于计算机中Web应用的绝对路径，则是指带访问协议的路径。例如下面的路径就是一个带有Http访问协议的绝对路径。

<http://127.0.0.1:8080/primary/index.jsp>

## 相对路径

相对路径，是指仅根据访问路径无法准确定位资源的路径。相对路径必须要结合其参照路径才可以组成可以准确定位的绝对路径。参照路径的不同，所形成的可以准确定位资源的绝对路径也是不同的。在进行资源访问时，必须要将相对路径转为绝对路径才可以完成资源的准确定位。

在Web应用中，相对路径的写法有两种：

1. 以斜杆开头的相对路径，
2. 不以斜杆开头的相对路径

根据相对路径是否以斜杆开头，且出现的文件所处位置的不同，其默认的参照路径是不同的。

### 以斜杆开头的相对路径

以斜杆开头的相对路径，根据路径所在文件所处位置的不同，分为两种：**前台路径**与**后台路径**。

#### 前台路径

所谓前台路径是指，由浏览器解析执行的代码中所包含的路径。例如：htm、css、js中的路径，及[jsp中静态部分的路径](#)。像html及jsp中的静态部分中的个、<a href=""/>、<form action=""/>等；像css中的个background:img(""/>等；想js中的window.location.href=""/>等，都属于前台路径。

前台路径的参照路径是**Web服务器的根路径**，即<http://127.0.0.1:8080/>

#### 后台路径

所谓后台路径是指，由服务器解析执行的代码及文件所包含的路径。例如，java代码中的路径、jsp文件的[动态部分](#)中的路径、xml文件中的路径等。

后台路径的参照路径是**Web应用的根路径**。如：<http://127.0.0.1:8080/primary/>

#### • 后台路径特例

对于后台路径的参照路径有一个特例：当代码中使用response的sendRedirect()方法进行重定向时，其参照路径不是Web应用的根路径，而是Web服务器的根路径。

## 不以斜杆开头的相对路径

不以斜杆开头的相对路径，无论是前台路径，还是后台路径，其参照路径都是当前资源的访问路径。