## 1 线程安全与不安全的理解

可以这样理解，加锁的就是安全的，不加锁的就是不安全的（感觉比较生动，不知道对不对，望指出）

线程安全 就是多线程访问时，采用了加锁机制，当一个线程访问该类的某个数据时，进行保护，其他线程不能进行访问直到该线程读取完，其他线程才可使用。不会出现数据不一致或者数据污染。

线程不安全 就是不提供数据访问保护，有可能出现多个线程先后更改数据造成所得到的数据是脏数据

## 2 Struts配置

### 2.1 配置web.xml（主要是核心过滤器）

<!-- 配置 Struts2框架核心Filter -->

<filter>

<!-- 过滤器名 -->

<filter-name>struts2</filter-name>

<!-- 配置Struts2核心Filter实现类 -->

<filter-class>org.apache.struts2.dispatcher.filter.StrutsPrepareAndExecuteFilter</filter-class>

</filter>

<!-- 让Struts2的核心Filter拦截所有请求 -->

<filter-mapping>

<!-- 过滤名 -->

<filter-name>struts2</filter-name>

<!-- 匹配所有请求 -->

<url-pattern>/\*</url-pattern>

</filter-mapping>

注意：任何MVC框架需要与Web应用整合时都要借助web.xml配置文件。通常MVC框架只需要在Web应用里加载一个核心控制器即可，对于Struts2来说就是加载StrutsPrepareAndExecuteFilter。一个Web应用只要加载StrutsPrepareAndExecuteFilter后，就具有Struts2的基本功能。

### 2.2 由控制器向页面传递消息：

<1>在控制器中添加属性，并且为属性添加get方法；

<2>在控制器方法中向属性赋值；

<3>在JSP页面上使用EL表达式读取属性值；EL表达式实际上执行属性的get方法，如果没有get方法就无法获取属性值。

### 2.3 编码

response.setCharacterEncoding("utf-8");设置服务器端编码方案，response.setContentType("text/html; charset=UTF-8");设置客户端解码方案。

### 2.4 action处理结果

（1）dispatcher：转发；

（2）redirectAction：重定向到其他Action；

（3）stream：用于处理图片或者下载；

（4）json：用于处理ajax请求。

  转发类型是在Struts的主配置文件中定义struts-default.xml；json：在struts2-json-plugin插件中定义。

 dispatcher：是默认的Result，也是最常用的类型。

 redirectAction：重定向到其他Action，用于处理重定向结果。org.apache.struts2.dispatcher.ServletActionRedirectResult

 stream：用于处理图片或者下载。org.apache.struts2.dispatcher.StreamResult

 json：用于处理ajax请求。json不是struts默认的result类型，需要使用一个插件来支持这个类型。org.apache.struts2.json.JSONResult

 有两种使用方式：

（1）将Action的全部属性发送到客户端；

     （2）将Action的一个属性发送到客户端。

 json返回结果必须使用struts2-json-plugin插件支持，必须先导入json插件。

### 2.5 值栈ValueStack

      （1）从控制器向页面传值的缓冲区；

      （2）ValueStack包含当前环境相关的全部对象： request、session、application等；

      （3）ValueStack包含两个区域：一个区域是root，是一个List，存储值，一个区域是context，是一个Map，存储环境相关的全部对象；

      在Struts2中ValueStack是软件的消息中心，软件的数据和相关环境信息都缓存在这个对象中，其作用与JSP中的pageContext类似，pageContext是JSP中的消息中心。

      JSP页面中使用<s: debug/>标签可以展示ValueStack

### 摘抄：

### <https://blog.csdn.net/gongxifacai_believe/article/details/72778097>

### 2.6 po类的字段类型的选择

**属性的数据类型都选用了对象类型，而非基本数据类型，原因是属性和字段对应，只有对象才能表示空值的概念**

### 2.7 主键生成策略

主键生成策略：

assigned：用户定义

increment:自动递增（long,integer,short）

uuid.hex：利用uuid算法

identity：由数据库根据identity生成主键

native：系统自动选择相应算法生成主键

sequence:由数据库根据序列生成主键

hilo:根据Hibernate的hilo生成主键

### 2.8 关于hibernate的表对应关系的配置文件的写法

<https://www.cnblogs.com/winniewu/p/4968316.html>（太多了，不复制了，不懂再去看）