**Servlet Study Notes**

**@author Count Monte Cristo**

**@create 2015/07/09**

**@version 2016/05/31 14:56**

**2016/05/26 15:08**

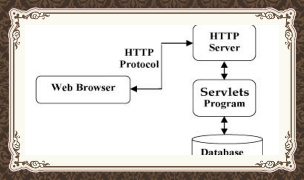
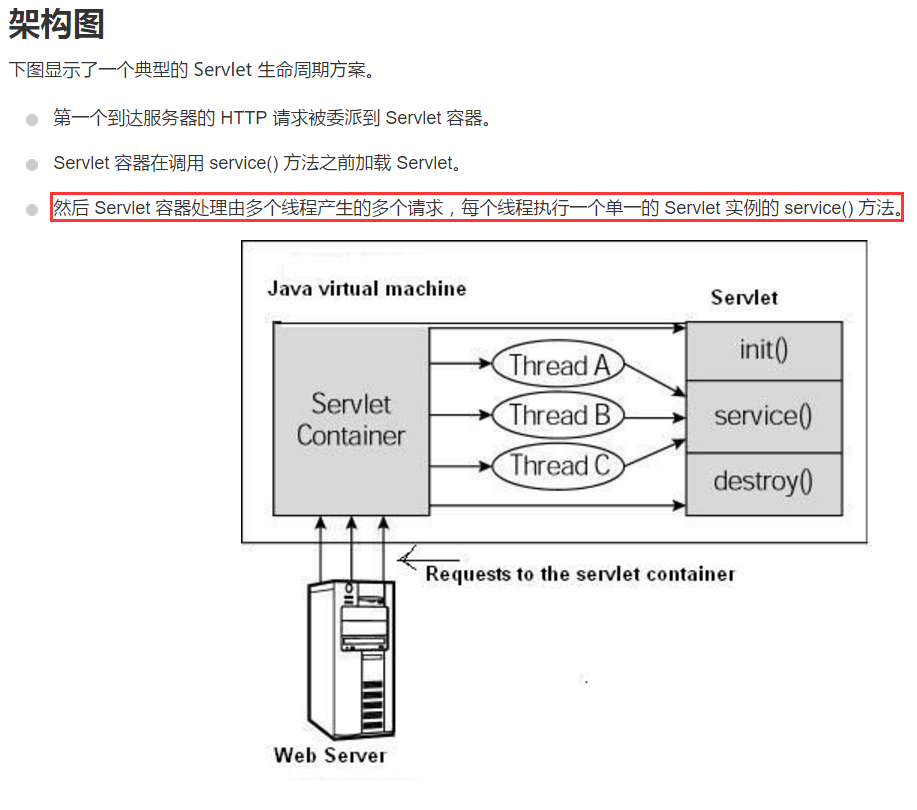
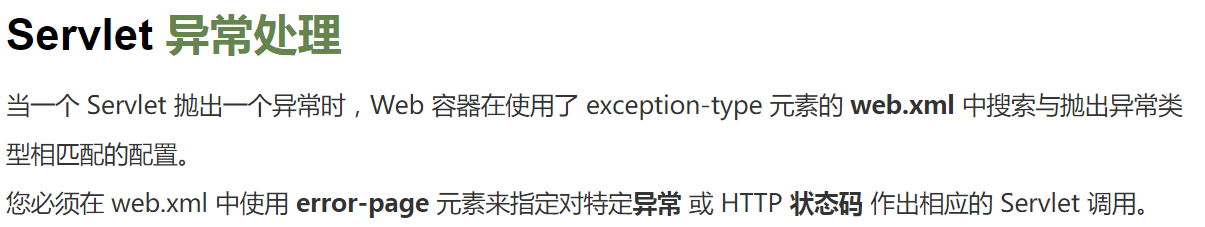
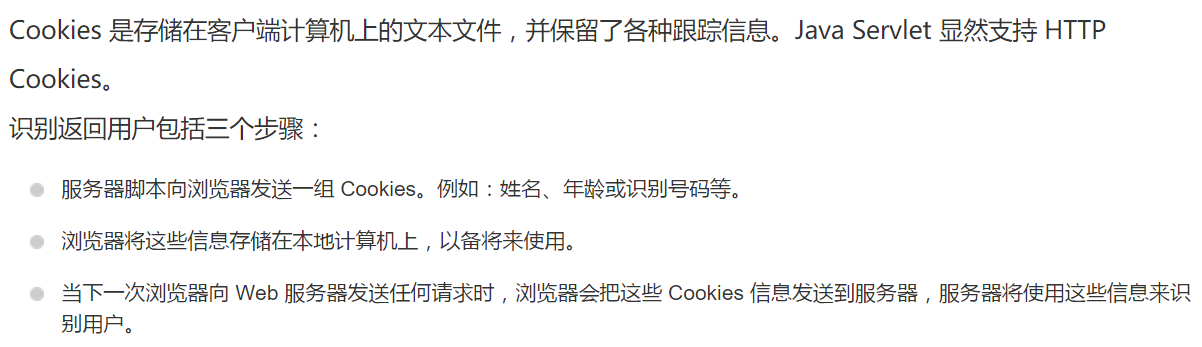
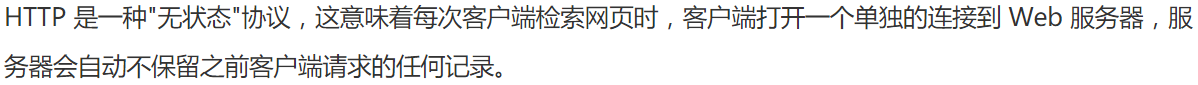
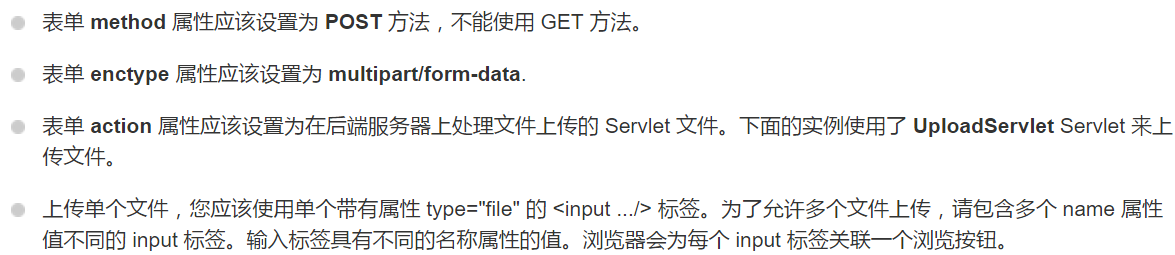
**2016/05/26 15:08**

**2015/07/09**

**@description 描述Servlet的重要知识**

**@motto VICTORY LOVES PREPARATION**

**@criteria 凯旋基诺**

1. **@reference** 
   1. **菜鸟教程**
   2. **百科百科**
2. **Servlet**
   1. Servlet（Server Applet），全称Java Servlet，未有中文译文。是用**Java编写的服务器端程序**。其主要功能在于**交互式地浏览和修改数据，生成动态Web内容**。狭义的Servlet是指Java语言实现的一个接口，广义的Servlet是指**任何实现了这个Servlet接口的类**，一般情况下，人们将Servlet理解为后者。
   2. **Servlet运行于支持Java的应用服务器中（运行在带有支持Java Servlet规范的解释器的web服务器上，所以一般来说支持Java Servlet的服务器都有Servlet的jar包：servelt-api.jar）**。从原理上讲，Servlet可以响应任何类型的请求，但绝大多数情况下Servlet只用来扩展**基于HTTP协议的Web服务器**。
   3. 最早支持Servlet标准的是JavaSoft的Java Web Server，此后，一些其它的基于Java的Web服务器开始支持标准的Servlet
   4. 它是作为来自 **Web 浏览器或其他 HTTP 客户端的请求**和 **HTTP 服务器上的数据库或应用程序**之间的**中间层**
   5. **由来【因为速度更快，替代了CGI（实现客户端和服务器之间的交互）】**
      1. **服务器上需要一些程序，常常是根据用户输入访问数据库的程序**。这些通常是使用公共网关接口（Common Gateway Interface，CGI）应用程序完成的。然而，在服务器上运行Java，这种程序可使用Java编程语言实现。在通信量大的服务器上，Java Servlet 的优点在于它们的执行速度更快于CGI 程序。**各个用户请求被激活成单个程序中的一个线程，而无需创建单独的进程，这意味着服务器端处理请求的系统开销将明显降低（因为线程的开销比进程少很多，@reference 百度百科）。**
   6. **架构**
      1. 
3. **Servlet的生命周期**
   1. 三个过程：init() -> service()[根据Http的请求类型调用doGet()、doPost()、doPut()、doDelete()等] -> destory()（**一般当 Server 关闭时被调用，或者卸载Web应用的时候**）
   2. 架构图
      1. 
4. **Web.xml**
5. **HTTP请求和响应**
   1. HTTP请求
      1. 浏览器请求网页的时候，不仅向浏览器传递了一些像姓名、年龄等属性数据，还提供了封装在HTTP请求头中的一般数据，如发送请求主机地址等信息，这些数据也可以通过HttpServletRequest对象获取
      2. 实例
         1. 
         2. 注意
            1. 上面是HTTP请求头的信息，HTTP请求还有用户传递的特殊数据
   2. HTTP响应
      1. 一般情况下，有请求就有与之对应的响应，**HTTP响应通常包含：一个状态行、一些响应报头、一个空行和文档**
      2. 实例
         1. 
         2. **@reference Java具体设置HTTP响应**
      3. 注意
         1. 可以通过设置MIME的类型来实现显示HTML、下载文件等等，也就是说返回内容的类型不只是”text/html”的，也可以是其他的，完全可以定义
         2. Http的响应完全封装在HttpServletResponse中，所有与页面响应有关的参数都可以在其中定义
6. **过滤器**
   1. 
7. **Servelt异常处理**
   1. 
   2. **注意**
      1. **具体情况 @reference 菜鸟教程-Servlet异常处理**
8. **Servelt Cookies处理**
   1. 
   2. **注意**
      1. **具体情况 @reference 菜鸟教程-Servlet Cookies处理**
9. **Servlet Session跟踪**
   1. 
   2. Session的跟踪可以使用Cookies、隐藏的表单元素、URL重写和HttpSession接口（该接口提供一种跨多个页面请求的存储用户信息：session的方式）
   3. **注意：面向对象编程，优点之一就是如果想操作某个对象就可以调用该对象的方法就行。例如，操作session，那就可以调用HttpSession对象的方法**
10. **Servlet文件上传**
    1. 
    2. **注意**
       1. **具体情况 @reference 菜鸟教程-Servlet 文件上传处理**
11. **Servlet处理日期**
    1. **具体情况 @reference 菜鸟教程-Servlet Cookies处理**