## 深入Servlet学习

### 基础篇：Servlet基础篇

#### 基本要求：

1、认真完成作业

2、学习过程中针对罗列的以及作业中涉及到的每一个接口方法的使用和特点形成自己简洁的总结文档

2、作业规范：

1. 项目名：servlet\_用户名全拼

例:刘长喜的作业 servlet\_liuchangxi

1. 基础包名：com.ly.learning.servlet.lession小节号

例：第一节包名：com.ly.learning.servlet.lession1

1. 每节一个总结文档

**参考资料：《servlet和jsp学习指南》 1.1 -1.7**

**学习目的：**掌握Servlet最底层的接口使用、能够结合一些开源框架，深刻体会各API的使用场景

**思考：**Servlet 是只针对WEB的开发接口吗？

##### javax.servlet.Servlet

注：接口方法的执行时机和参数使用方式

练1：基于 javax.servlet.Servlet接口自定义一个Servlet类

在该类的各方法中分别输出方法名到控制台，观察各方法的执行时机

##### javax.servlet.ServletRequest

注：接口方法的作用及参数含义 明确功能相似方法的区别

典型问题：getParameter 和getAttribute区别和使用场景

练2：

基于 javax.servlet.Servlet接口在练1的基础上实现请求参数的传递

通过getAttribute实现转发过程中的参数传递

##### javax.servlet.ServletResponse

注：接口方法的作用及参数含义

练3：基于 javax.servlet.Servlet.ServletResponse接口在练1的基础上实现向页面输出一段HTML 的功能

##### javax.servlet.ServletConfig

注：接口方法的功能，ServletConfig的使用方式和使用场景

练4：在练3的基础上通过XML配置的方式实现在 Servlet初始化时传入配置参数，并留存该配置，供每次Servlet调用时使用

思考：org.springframework.web.servlet.DispatcherServlet 作为SpringMVC的核心配置类，SpringMVC的初始化是其重要使命之一，请分析ServletConfig在该过程中的重要作用，直接在项目的web.xml 中找到配置，并简单跟踪源码分析

##### javax.servlet.ServletContext

注：接口方法的功能，使用场景不多

练5：在练4的基础上通过XML配置的方式实现在 Servlet初始化时传入配置参数到ServletContext中

思考：org.springframework.web.context.ContextLoaderListener 作为Spring的核心配置类，初始化Spring框架组件是其核心任务，请分析在Spring的初始化过程中 ServletContext是如何定位到Spring的核心配置文件位置的。，直接在项目的web.xml 中找到配置，并简单跟踪源码分析

#### 第二节： 从Servlet到HttpServlet（更加友好的WEB开发）

**学习目的：**掌握 Servlet到HttpServlet 扩展中的关键扩展点，掌握整个扩展的类图结构

**思考：**Servlet接口的实现在哪里？为什么我们只是针对接口编程确可以运行？什么是Servlet容器

参考文档：1.8~1.9节，GenericServlet, HttpServlet源码

##### 1.javax.servlet.GenericServlet

**内容1:** 了解GenericServlet对Servlet接口中的方法的实现（阅读GenericServlet源码）；

**内容2：**了解GenericServlet对ServletConfig接口的实现（阅读GenericServlet源码）

练习1：继承javax.servlet. GenericServlet类自定义一个HelloGenericServlet类,传递一个初始化参数address，值为“同程大厦”,在service方法中将此参数取出以html形式写到客户端。

**思考：** GenericServlet 本质上就是对Servlet的接口中的“一般性逻辑的方法进行了实现”，很显然Servlet对象应该提供非常直接友好的方式去获取初始化参数等Servlet对象的“基本信息”,那么思考并概括GenericServlet 对此的实现方式和运行流程。

##### 2. javax.servlet.http. HttpServlet

**内容1：**学习HttpServlet如何对http请求进行处理（阅读HttpServlet的service源码）

**内容2：**了解HttpServlet对哪些http请求方式进行支持（阅读HttpServlet的service源码）

**练习：** 继承HttpServlet实现一个HelloHttpServlet, 测试doHead, doTrace， doGet方法

**思考：**从GenericServlet到HttpServlet ，从类设计的角度概括分析HttpServlet 主要扩展了GenericServlet的什么方法，为什么要这么做？请尝试，不参考资料，自己按照Servlet ->GenericServlet->HttpServlet 的扩展演化过程，绘制出他们的基本类图，并对关键扩展写上说明。

##### 3. javax.servlet.http.HttpServletRequest

**内容：**学习HttpServletRequest对Http，cookie,请求头，请求行，请求参数，请求扩展的接口

**练习：** 编写一个Servlet，获取请求路径，请求方法，所有的请求头，所有的Cookie, 将这些信息通过response写到浏览器页面。

**思考：** HttpServletRequest扩展了ServletRequest哪些功能？

##### 4. javax.servlet.http.HttpServletResponse

内容：学习HttpServletResponse对Http cookie,响应头，响应行扩展的接口，

练习：编写一个Servlet, 往客户端设置一个cookie, 发送”我在学习HttpServlet”到客户端，内容类型设置为html, 设置过期时间为10秒。

思考： HttpServletRequest扩展了ServletRequest哪些功能？

#### 第三节：监听器

**学习目标：**学习在ServletContext, HttpSession, Request三个级别的监听器

##### javax.servlet. ServletContextListener

**内容**：学习如何配置Context listener. 触发时机

**练习**：实现ServletContextListener接口，配置一个context initial parameter, 在监听方法中输出类名和方法名，同时获取这个context initial parameter.部署观察方法触发调用的时机。

**思考**：如何给ServletContextListener传递初始化参数，注册多个ServletContextListener的话，多个listener调用顺序怎么处理？

##### javax.servlet. ServletContextAttributeListener

**内容**：学习如何配置Context attribute listener及其触发时机

**练习**：实现ServletContextAttributeListener接口， 在监听方法中输出类名和方法名，部署观察方法触发调用的时机。

**思考：**往ServletContext中添加attribute这个操作是线程安全的吗？注册多个Listener的话，多个listener调用顺序怎么处理？

##### javax.servlet. ServletRequestListener

**内容：**学习配置ServletRequestListner及其触发时机

**练习：**实现ServletRequestListener接口， 在监听方法中输出类名和方法名，部署观察方法触发调用的时机

**思考：**如果有Filter, requestInitialized， requestDestroyed方法和filter的方法调用顺序是什么样的？ 如果在请求转发或包含，ServletRequestListener会触发吗？

##### javax.servlet. ServletRequestAttributeListener

**内容：**学习配置ServletRequestAttributeListener及其触发时机

**练习：**实现ServletRequestAttributeListener接口， 在监听方法中输出类名和方法名，部署观察方法触发调用的时机

**思考：**request中parameter和attribute有什么区别， attribute在什么场景下使用较多？

##### javax.servlet.http.HttpSessionListener

**内容：**学习配置HttpSessionListener及其触发时机

**练习：**实现HttpSessionListener接口， 在监听方法中输出类名和方法名，部署观察方法触发调用的时机

##### javax.servlet.http.HttpSessionAttributeListener

**内容：**学习配置HttpSessionAttributeListener及其触发时机

**练习：**实现HttpSessionAttributeListener接口， 在监听方法中输出类名和方法名，部署观察方法触发调用的时机

思考：往session中添加attribute时线程安全的操作吗？

#### 第四节：过滤器

**学习目的：掌握责任链设计模式，及该设计模式在web Filter中的应用**

1. 熟悉Filter, FilterChain, FilterConfig接口，并编写两个过滤器拦截所有请求。过滤器1计算请求的耗时。过滤器2检查cookie中是否有access\_token， 如果没有，返回给用户未登录提示，如果有则继续执行。
2. Java web中，请求是如何在各个过滤器间传递，最终到达servlet的？编码模拟实现此功能。参考<https://www.cnblogs.com/lizo/p/7503862.html>

**思考：**监听器和过滤器有什么区别？ 如果在过滤器中执行耗时的操作对整个请求有什么影响？ 如果在监听器中执行耗时操作对整个请求有什么影响？

#### 第五节：阶段总结和任务：

需求：实现一个用户管理功能并模拟spring mvc来实现：

1. 用户包含Long id, String name两个属性。

2. 控制器类UserController， 包含addUser(User user), removeUser(Long id)两个方法。

3. 控制器类返回Response对象， 包含code, message和User。 code 10000表示成功，message自定义。

4. 服务类UserService， 包含addUser(User u), removeUser(Long id)两个方法，此方法被UserControlle调用。方法中输出信息即可。

5. 以properties文件来模拟spring的bean定义, 如果时间上足够，可以使用xml，注解等方式来模拟spring的bean定义。

userService=com.ly.user.service.UserService #key为bean name， value为类名

6. 定义一个类ApplicationContext来模拟spring的容器，该类提供一个getBean(String name)方法。 内部可以用hashmap来存储，bean name为key, bean为value.

7. 模拟spring mvc的ContextLoaderListener来初始化ApplicationContext, 就是读取bean的定义文件，来初始化ApplicationContext容器。

8. 模拟spring mvc的DispatchServlet。 url的映射可以properties文件书写。如果时间上足够，可以使用xml,注解等方式.

/user=com.ly.user.controller.UserController #url的映射有bean name +方法名的方式映射，如/user/addUser映射到UserController.addUser方法。

9. 模拟DispatchServlet的init方法来初始化上面的映射关系和控制器类UserController，可以使用hashmap等方式来存储。注意UserControlel依赖于UserService.

10. 实现service方法，根据上面定义的映射来处理请求。 控制器类返回的数据以json形式返回给客户端。

### 窥探：Servlet容器

#### 第一节：Servlet 容器的模块设计，及各模块基本职责

#### 第二节：Servlet 初始化、生命周期、Filter、Listener初始化及生命周期与Tomcat的关联

#### 里程碑：阶段总结和任务：