基本类和接口

一、javax.servlet.Servlet接口

servlet抽象集是javax.servlet.Servlet接口，它规定了必须由Servlet类实现由servlet引擎识别和管理的方法集。

Servlet接口的基本目标是提供生命期方法init()、service()和destroy()方法。

servlet接口中的方法

void init(ServletConfit config)throws ServletException 在servlet被载入后和实施服务前由servlet引擎进行?次性调用。如果init()产生溢出UnavailableException，则 servle退出服务。

ServletConfig getServletConfig() 返回传递到servlet的init()方法的ServletConfig对象

void service(ServletRequest request, ServletResponse response)throws ServletException,IOException 处理request对象中描述的请求，使用response对象返回请求结果

String getServletInfo() 返回描述servlet的一个字符串

void destory() 当servlet将要卸载时由servlet引擎调用

二、javax.servlet.GenericServlet类（协议无关版本）

GenericServlet是一种与协议无关的servlet，是一种跟本不对请求提供服务的servlet，而是简单地从init()方法启动后台线程并在destory()中杀死。它可以用于模拟操作系统的端口监控进程。

servlet API提供了Servlet接口的直接实现，称为GenericServlet。此类提供除了service()方法外所有接口中方法的缺省实现。这意味着通过简单地扩展GenericServlte可以编写一个基本的servlet。

除了Servlet接口外，GenericServlet也实现了ServletConfig接口，处理初始化参数和servlet上下文，提供对授权传递到init()方法中的ServletConfig对象的方法。

GenericServlet类中的方法

void destory() 编飞组成单词“destory”的一个注册入口

String getInitParameter(String name) 返回具有指定名称的初始化参数值。通过凋用config.getInitParameter(name)实现。

Enumeration getInitParameterNames() 返回此serv]et已编码的所有初始化参数的?个枚举类型值。调用config.getInitParameterNames()获得列表。如果未提供初始化参数，则返回?个空的枚举类型值（但不是null）

ServletConfig getServletConfig() 返回传递到init()方法的ServletConfig对象

ServletContext getServletContext() 返回在config对象中引用的ServletContext

String getServletInfo() 返回空字符串

void init(ServletConfig config)throws ServletException 在一实例变量中保存config对象。编写组成单词“init”的注册入口,然后调用方法init()

void init()throws ServletException 可以被跳过以处理servlet初始化.在config对象被保存后init(ServletConfig config)的结尾处自动被调用.servlet作者经常会忘记调用super.init(config)

void log(String msg) 编写注册servlet的入口.为此调用servlet上下文的log()方法.servlet的名字被加到消息文本的开头。

void log(String msg,Throwable t) 编写一个入口和servlet注册的栈轨迹。此方法也是ServletContext中相应方法的一个副本。

abstract void service(Request request,Response response)throws ServletException,IOException 由servlet引擎调用为请求对象描述的请求提供服务。这是GenericServlet中唯一的抽象方法。因此它也是唯一必须被子类所覆盖的方法。

String getServletName() 返回在Web应用发布描述器（web.xml）中指定的servlet的名字。

三、javax.servlet.http.HttpServlet类（HTTP版本）

虽然servlet API允许扩展到其它协议，但最终所有的servlet均在Web环境下实施操作，只有几种servlet直接扩展了GenericServlet。对servlet更一般的是扩展其HTTP子类HttpServlet。

HttpServlet 类通过调用指定到HTTP请求方法的方法实现service()，亦即对DELETE、HEAD、GET、OPTIONS、POST、PUT和 TRACE，分别调用doDelete()、doHead()、doGet()、doOptions()、doPost()、doPut()和 doTrace()方法，将请求和响应对象置入其HTTP指定子类。

HttpServlet类中的方法

Void doGet(HttpServletRequest request,HttpServletResponse response)throws ServletException,IOException 由servlet引擎调用用处理一个HTTP GET请求。输入参数、HTTP头标和输入流可从request对象、response头标和response对象的输出流中获得。

Void doPost(HttpServletRequest request,HttpServletResponse response)throws ServletException,IOException 由servlet引擎调用用处理一个HTTP POST请求。输入参数、HTTP头标和输入流可从request对象、response头标和response对象的输出流中获得。

Void doPut(HttpServletRequest request,HttpServletResponse response)throws ServletException,IOException 由servlet引擎调用用处理一个HTTP PUT请求。本方法中请求URI指出被载入的文件位置。

Void doDelete(HttpServletRequest request,HttpServletResponse response)throws ServletException,IOException 由servlet引擎调用用处理一个HTTP DELETE请求。请求URI指出资源被删除。

Void doOptions(HttpServletRequest request,HttpServletResponse response)throws ServletException,IOException 由servlet引擎调用用处理一个HTTP OPTIONS请求。返回一个Allow响应头标表明此servlet支持的HTTP方法。一个servlet不需要覆盖此方法，因为 HttpServlet方法已经实现规范所需的功能。

Void doTrace(HttpServletRequest request,HttpServletResponse response)throws ServletException,IOException 由servlet引擎调用用处理一个HTTP TRACE请求。使得请求头标被反馈成响应关标。一个servlet不需要覆盖此方法，因为HttpServlet方法已经实现HTTP规范所需的功能。

Void service(HttpServletRequest request,HttpServletResponse response)throws ServletException,IOException Service(Request request,Response response)调用的一个立即方法，带有指定HTTP请求和响应。此方法实际上将请求导向doGet()、doPost()等等。不应该覆盖此方法。

Void service(Request request,Response response)throws ServletException,IOException 将请求和响应对象置入其指定的HTTP子类，并调用指定HTTP的service()方法。

四、javax.servlet.ServletRequest类（协议无关版本）

ServletRequest接口封装了客户端请求的细节。它与协议无关，并有一个指定HTTP的子接口。

ServletRequest主要处理：

1.找到客户端的主机名和IP地址

2.检索请求参数

3.取得和设置属性

4.取得输入和输出流

ServletRequest类中的方法

Object getAttribute(String name) 返回具有指定名字的请求属性，如果不存在则返回null。属性可由servlet引擎设置或使用setAttribute()显式加入。

Enumeration getAttributeName() 返回请求中所有属性名的枚举。如果不存在属性，则返回一个空的枚举。

String getCharacteEncoding() 返回请求所用的字符编码。

Int getContentLength() 指定输入流的长度，如果未知则返回-1。

ServletInputStream getInputStream()throws IOException 返回与请求相关的（二进制）输入流。可以调用getInputStream()或getReader()方法之一。

String getParameter(String name) 返回指定输入参数，如果不存在，返回null。

Enumeration getParameterName() 返回请求中所有参数名的一个可能为空的枚举。

String[] getParameterValues(String name) 返回指定输入参数名的取值数组，如果取值不存在则返回null。它在参数具有多个取值的情况下十分有用。

String get Protocol() 返回请求使用协议的名称和版本。

String getScheme() 返回请求URI的子串，但不包含第一个冒号前的内容。

String getServerName() 返回处理请求的服务器的主机名。

String getServerPort() 返回接收主机正在侦听的端口号。

BufferedReader getReader()throws IOException 返回与请求相关输入数据的一个字符解读器。此方法与getInputStream()只可分别调用，不能同时使用。

String getRemoteAddr() 返回客户端主机的数字型IP地址。

String getRemoteHost() 如果知道，返回客户端主机名。

void setAttribute(String name,Object obj) 以指定名称保存请求中指定对象的引用。

void removeAttribute(String name) 从请求中删除指定属性

Locale getLocale() 如果已知，返回客户端的第一现场或者为null。

Enumeration getLocales() 如果已知，返回客户端的第一现场的一个枚举，否则返回服务器第一现场。

boolean isSecure() 如果请求使用了如HTTPS安全隧道，返回true

RequestDispatcher getRequestDispatcher(String name) 返回指定源名称的RequsetDispatcher对象。

本文来自CSDN博客，转载请标明出处：http://blog.csdn.net/andymu077/archive/2008/05/15/2448849.aspx