# HTTPServlet源码分析

## 简单介绍

* HTTPServlet是为了HTTP协议准备的，比GenericServlet更加适合HTTP协议下的开发。
* HTTPServlet的位置：jakarta.http.HttpServlet

到目前为止我们**接触过的接口**：

jakarta.servlet.Servlet;

jakarta.servlet.GenericServlet;

jakarta.servlet.ServletContext;

jakarta.servlet.ServeltConfig;

jakarta.servlet.ServletReuest;

jakarta.servlet.ServletResponse;

jakarta.servlet.ServletException;

**http包下都有哪些接口**？

jakarta.servlet.http.HttpServlet; （HTTP协议专用的Servlet类，抽象类）

jakarta.servlet.http.HttpServletRequest; （HTTP专用的请求对象）

jakarta.servlet.http.HttpServetResponse; （HTTP专用的响应对象）

注意：**HttpServletRequest中封装了哪些信息**？

HttpServletRequest对象，简称request对象，其中封装了请求协议的全部内容。

tomcat服务器（web服务器）将“请求协议”中的数据全部解析出来，然后将其封装到request对象中了。所以，只要面向HttpServletRequest，就可以获得请求协议中的数据。

而，HttpServletResponse是专门用来响应HTTP协议的，其中应该包含的是全部相应内容。

## 源码分析

*考虑到代码在WPS比较丑，而这一段的代码比较简单，所以就不在这里贴源码了*。

HttpServlet继承了GenericServlet，并且包含了两个service函数。

1. 首先重写了父类的service函数。
2. 然后让重写的service函数调用了自己的重载过的service函数。（这个函数就是模板方法）

*HttpServlet没有重写父类中的init函数，所以调用的时候还是会调用GenericServlet中的init()函数。*

*在上述1.中的service主要负责将获取到的ServletRequest和ServletResponse向下转型为HttpServletRequest和HttpServletResponse类，然后用这两个Http开头的类调用自己的2.标记的service函数。*

我们编写的Servlet在继承HttpServlet之后，可以重写HttpServlet类的service函数，但是这样将会无法享受405错误。（405错误表示请求类型和处理类型不一致，比如对于get请求调用了doPos()函数）

405错误是**前端错误**，表示前端**发送的请求类型和后端要求的请求类型不一致**。

**为什么会报405**？

如果继承HttpServlet之后，没有重写某一个函数（比如doPost），当收到一个post请求的时候，tomcat会自动调用doPost函数，但是这个时候因为编写的Servlet没有重写doPost()，所以就会调用父类HttpServlet中的doPost()函数，而HttpServlet中的doPost会直接抛出405错误（因为程序员写的程序没有编写这个函数）。

**如何避免405错误**？

后端要求发什么方式，前端就发什么类型的请求。后端重写了doGet()，那么前端就发送get请求；后端重写了doPost()，前端就发送post请求。

*有些时候会有人为了避免405，把doGet和doPost全部重写，这样是不对的，因为405存在是有自己的意义的，强行避免反而不是很好，如果要是同时重写doGet和doPost，那还不如直接重写service函数。*

# Servlet类的开发步骤（最终版）

到这里位置，我们就得到了最终的Servlet类的开发步骤：

1. 编写一个Servlet类，继承HttpServlet类。
2. 重写doGet或者doPost函数，具体哪个根据实际情况判断。
3. 将Servlet类配置到web.xml中。
4. 准备前端页面（form表单）发送请求，指定请求路径即可。