

文章编号: 2096-1472(2017)-01-09-03

# 简单HTTP Server的设计与实现

周 畅, 王 贇

(天津工业大学计算机科学与软件学院, 天津 300387)

**摘 要:** 超文本传输协议(HTTP)是分布式、协作和超媒体信息系统的协议。HTTP Server是一个监听特定端口TCP连接, 对客户端请求进行处理的一个应用。得益于高级语言对HTTP协议的内置支持, 我们也可以使用简单的方式来实现一个较完整功能的HTTP Server。本文主要是研究一个HTTP Server的简单实现, 在实现的过程中, 能够体现出POST、GET方法, 现已将要求完成, 本文一共分为三个章节, 分别对于此次设计进行阐述。

**关键词:** 超文本传输协议服务; 万维网; 外部应用程序

**中图分类号:** TP393.0 **文献标识码:** A

## Design and Implementation of the Simple HTTP Server

ZHOU Chang, WANG Ze

(School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin 300387, China)

**Abstract:** The Hypertext Transfer Protocol(HTTP) is an application protocol for distributed, collaborative, and hypermedia information systems. HTTP Server is an application to monitor the TCP connection of a specific port and handle requests from the client. Thanks to the built-in support for the high-level language HTTP protocol, we can also use a simple method to implement a HTTP Server with complete functionality. This paper focuses on a simple implementation process of a HTTP Server, during which, the POST and GET methods has been completed. The paper is divided into three chapters, elaborating on the design and implementation.

**Keywords:** HTTP server; world wide web; external applications

### 1 引言(Introduction)

CGI是WWW技术中最重要的技术之一, 有着不可替代的重要地位。CGI是外部应用程序(CGI程序)与Web服务器之间的接口标准, 是在CGI程序和Web服务器之间传递信息的规程<sup>[1]</sup>。CGI规范允许Web服务器执行外部程序, 并将它们的输出发送给Web浏览器, CGI将Web的一组简单的静态超媒体文档变成一个完整的新的交互式媒体。使在网络服务器下运行外部分应用程序(或网关)成为可能。CGI-BIN目录是存放CGI脚本的地方。这些脚本使Web服务器和浏览器能运行外部程序, 而无需启动另一个程序<sup>[2]</sup>。

### 2 项目背景(The project background)

在HTTP中定义了很多和服务器之间进行交互的方法, 例如平时我们所看到的GET、POST、PUT、DELETE<sup>[3,4]</sup>。其中资源描述符是URL, 在这里我们可以这样理解, 一个URL地址可以描述一个网络上的资源, 而前面所提到的HTTP中的, GET、POST、PUT、DELETE所对应的分别是这个资源的查、改、增、删四个操作, 而这其中的POST一般适用于资

源信息的更新, GET用于信息的获取/查询, 因为早期的系统对于DELETE是不支持的, 所以说PUT和DELETE用的比较少。

POST方法是HTTP中的一个重要的组成部分, 这种方法一般是用来项目的服务器发出更新请求, 在请求中附有请求实体<sup>[5]</sup>。

在客户机和服务器之间进行请求-响应时, 这两种方法是最常被用到的。

POST: 向指定的资源提交要被处理的数据, 采用这种方法, 查询字符串(名称/值对)是在POST请求的HTTP消息主体中发送的。

GET: 从指定的资源请求数据, 采用这种方法查询字符串(名称/值对)是在GET请求的URL中发送的<sup>[6,7]</sup>。

除以上介绍的内容之外, 这两种方法在安全性等方面也有一定的区别, 表1列出了两种方法在具体项目上的区别。

### 3 具体实现(Implementation)

实现HTTP Server的简单设计有多种语言可以选择, 但

是本文采取的是Python语言进行设计。Python是一种代表简单主义思想的语言<sup>[8]</sup>。它的底层是用C语言写的，很多标准库和第三方库也都是用C写的，运行速度非常快，无需考虑诸如如何管理你的程序使用的内存一类的底层细节<sup>[9]</sup>。

表1 POST和GET的比较

Tab.1 The comparison of POST and GET

项目	GET	POST
后退按钮/刷新	无害	数据会被重新提交(浏览器应该告知用户数据会被重新提交)
书签	可收藏为书签	不可收藏为书签
缓存	能被缓存	不能缓存
编码类型	application/x-www-form-urlencoded	application/x-www-form-urlencoded或multipart/form-data，为二进制数据使用多重编码
历史	参数保留在浏览器历史中	参数不会保存在浏览器历史中
对数据长度的限制	当发送数据时，GET方法向URL添加数据；URL的长度是受限制的(URL的最大长度是2048个字符)	无限制
对数据类型的限制	只允许ASCII字符	没有限制，也允许二进制数据
可见性	数据在URL中对所有人都是可见的	数据不会显示在URL中

### 3.1 代码编写

程序的代码编写主要包括post、get方法实现，以及服务器的建立，其中在代码编写的过程中，post和get的体系基本相同，所以这里只介绍get的方法，post就不再进行赘述，其中get的代码思路主要是首先从cgi中将存储的数据调到form中，接着取出字段名是name的字段，取出字段名是comment的字段，最后打印html，并且加入name comment。除了get、post的编写之外还有服务器的建立，大体思路是重写http server类，然后获得cgi句柄，接着设置cgi脚本路径，最后是监听端口的死循环。

以上是简单的介绍一下程序代码的思路，为了代码编写过程中能够更加的简洁一些，才会采用Python这种编程语言，下面附上代码的截图以及代码的注释，以下两个图分别是get和服务器创建的代码及注释，post的不再进行重复。

```
#!/usr/bin/env python
import cgi #
import cgiib # Traceback manger for CGI scripts

form = cgi.FieldStorage() # 把cgi中存储的数据读到form中

name = form.getvalue('Name') # 取出字段名是name的字段
comment = form.getvalue('comment') # 取出字段名是comment的字段

# 打印html 并加入name comment
print 'Content-type:text/html\r\n\r\n'
print '<html>'
print '<head>'
print '<title>Post method Example</title>'
print '</head>'
print '<body>'
print '<h2> %s sends a comment: %s</h2>' %(name, comment)
print '</body>'
print '</html>'
```

图1 get创建的代码及注释

Fig.1 The code and comments created through GET

```
import os
import cgi
import argparse
import BaseHTTPServer
import CGIHTTPServer
import cgiib
cgiib.enable()

def web_server(port):
    server = BaseHTTPServer.HTTPServer # 重写httpserver类
    handler = CGIHTTPServer.CGIHTTPRequestHandler # 获得cgi句柄
    server_address = ('', port)
    handler.cgi_directories = ['/cgi-bin',] #设置cgi脚本路径
    httpd = BaseHTTPServer.HTTPServer(server_address, handler)
    print 'Starting web server with CGI support on port: %s ...' %port
    httpd.serve_forever() # 监听端口的死循环

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description = 'CGI Server')
    parser.add_argument('-port', action='store', dest='port', type=int, required=True)
    given_args = parser.parse_args()
    web_server(given_args.port)
```

图2 服务器创建的代码及注释

Fig.2 The code and comments created through the server

### 3.2 软件实现

建立在理解以上概念的基础之上，接下来对于此项目进行具体的实现，在创建项目的时候采用的Python语言，首先介绍一下程序实现的流程，程序运行开始之后，要建立一个服务器，然后创建监听端口，接下来用户会发送一个请求，在接收到用户的请求之后，要对用户的请求类型进行判断，因为在HTTP中定义了很多和服务器之间进行交互的方法，如前面提到的GET、POST、PUT、DELETE，但是由于此次项目只是针对GET、POST方法的实现进行设计，所以说PUT、DELETE在这里不做考虑。

判断用户的类型之后，要对用户的要求进行处理，例如，用户的请求类型是POST，那么接下来要在CGI-BIN中找到相对应的脚本，找到脚本之后，对请求进行处理，并且将数据返回到监听端口，GET类型的流程和POST是一样的，因为没有涉及DELETE和PUT方法，所以如果用户的要求是属于这两种类型的话，那么将不做任何处理，直接返回到监听端口<sup>[10]</sup>。

如图3所示是程序实现过程中的流程图，程序的流程图主要分为两个部分，一个是主程序的流程图，另外一个是在进行判断是后的流程图。

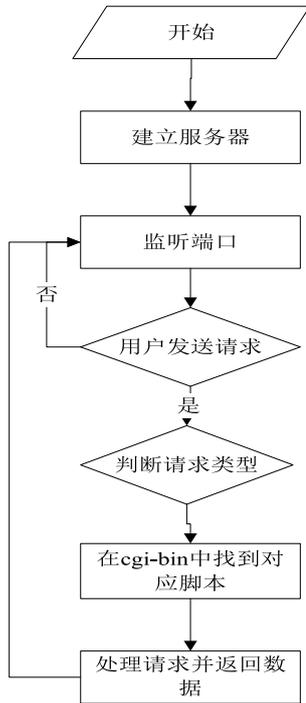


图3 主程序流程图

Fig.3 Main program flow chart

图3所显示的是该程序的主程序流程图，表现了不管采用的是什么样的方法，都是按照这样的流程进行，但是正如前面所说的那样，这个程序在处理用户请求的时候认可的是POST和GET两种方法，所以说当用户的请求是DELETE或者是PUT的时候，程序是不会进行数据处理的，如图4所示请求类型判断时候的具体方法。

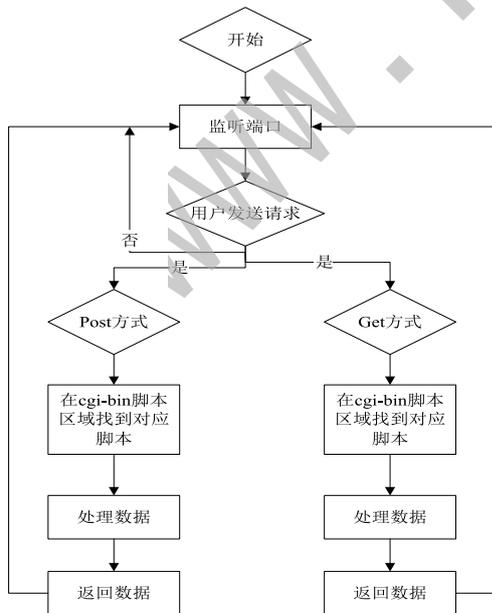


图4 请求方式判断流程图

Fig.4 Request flow chart

### 4 结论(Conclusion)

在这篇文章中，我们讨论实现HTTP Server的一些方法，并分析了其优缺点。经过对比，我们针对GET和POST方法进行了设计。并使用Python语言编写脚本，对程序进行测试，最终实现简单HTTP Server的设计。

### 参考文献(References)

- [1] Huang B,Lin H.A Java/CGI Approach to Developing a Geographic Virtual Reality Toolkit on the Internet[J]. Computers & Geosciences,2002,28(1):13-19.
- [2] Min B,Nahm E,Kim H.An analytic Model of an HTTP Server[J].Journal of the Electrochemical Society,2002,140(140):1080-1085.
- [3] Laliberte D,Hamilton M.Experimental HTTP Methods to Support Indexing and Searching[J].Journal of Biomolecular Structure & Dynamics,1996,1(6):1543-1546.
- [4] Demiryurek U,Shahabi C,Banaei-Kashani F.Efficient K-nearest Neighbor Search in Time-Dependent Spatial Networks:US8566030[P].2013.
- [5] 刘一峰,刘冰,郭良动.抓取HTTP协议POST文本数据方法设计[J].中国科技信息,2016(18):26-27.
- [6] 杜雷,辛阳.基于规则库和网络爬虫的漏洞检测技术研究与应用[J].信息安全,2014(10):38-43.
- [7] 齐红心.PHP开发过程中页面参数传递方法分析[J].软件导刊,2016,15(1):6-9.
- [8] Salvatier J,Wiecki T,Fonnesbeck C.Probabilistic Programming in Python using PyMC[J].Statistics,2015.
- [9] Perkel J M.Programming:Pick up Python[J].Nature,2015, 518(7537):125-126.
- [10] 狄刚.HTTP实现代理服务器及缓存替换算法的研究[D].吉林大学,2010.

### 作者简介:

周 畅(1993-),女,硕士生.研究领域:网络安全.

王 曠(1976-),男,博士,副教授.研究领域:网络安全.