HTTPCORE源码分析

2016K8009929009

秦 宏

2018

目录

[一、HTTP初步了解 2](#_Toc531694428)

[（一）概念 2](#_Toc531694429)

[（二）目的 2](#_Toc531694430)

[（三）详细内容 2](#_Toc531694431)

[（四）八种方法 2](#_Toc531694432)

[1.GET 2](#_Toc531694433)

[2.HEAD 2](#_Toc531694434)

[3.POST 2](#_Toc531694435)

[4.PUT 3](#_Toc531694436)

[5.DELETE 3](#_Toc531694437)

[6.TRACE 3](#_Toc531694438)

[7.OPTIONS 3](#_Toc531694439)

[8.CONNECT 3](#_Toc531694440)

[（五）安全方法 3](#_Toc531694441)

# 一、HTTP初步了解

## （一）概念

超文本传输协议（英语：HyperText Transfer Protocol，缩写：HTTP）是一种用于分布式、协作式和超媒体信息系统的应用层协议[1]。HTTP是万维网的数据通信的基础。

## （二）目的

设计HTTP最初的目的是为了提供一种发布和接收HTML页面的方法。通过HTTP或者HTTPS协议请求的资源由统一资源标识符（Uniform Resource Identifiers，URI）来标识。

## （三）详细内容

HTTP是一个客户端终端（用户）和服务器端（网站）请求和应答的标准（TCP）。通过使用网页浏览器、网络爬虫或者其它的工具，客户端发起一个HTTP请求到服务器上指定端口（默认端口为80）。我们称这个客户端为用户代理程序（user agent）。应答的服务器上存储着一些资源，比如HTML文件和图像。我们称这个应答服务器为源服务器（origin server）。在用户代理和源服务器中间可能存在多个“中间层”，比如代理服务器、网关或者隧道（tunnel）。

## （四）八种方法

### 1.GET

向指定的资源发出“显示”请求。使用GET方法应该只用在读取数据，而不应当被用于产生“副作用”的操作中，例如在Web Application中。

### 2.HEAD

与GET方法一样，都是向服务器发出指定资源的请求。只不过服务器将不传回资源的本文部分。它的好处在于，使用这个方法可以在不必传输全部内容的情况下，就可以获取其中“关于该资源的信息”。

### 3.POST

向指定资源提交数据，请求服务器进行处理（例如提交表单或者上传文件）。数据被包含在请求本文中。这个请求可能会创建新的资源或修改现有资源，或二者皆有。

### 4.PUT

向指定资源位置上传其最新内容。

### 5.DELETE

### 6.TRACE

### 7.OPTIONS

### 8.CONNECT

## （五）安全方法

对于GET和HEAD方法而言，除了进行获取资源信息外，这些请求不应当再有其他意义。也就是说，这些方法应当被认为是“安全的”。 客户端可能会使用其他“非安全”方法，例如POST，PUT及DELETE，应该以特殊的方式（通常是按钮而不是超链接）告知客户可能的后果（例如一个按钮控制的资金交易），或请求的操作可能是不安全的（例如某个文件将被上传或删除）。

但是，不能想当然地认为服务器在处理某个GET请求时不会产生任何副作用。事实上，很多动态资源会把这作为其特性。这里重要的区别在于用户并没有请求这一副作用，因此不应由用户为这些副作用承担责任。

## （六）关于HTTPcore

本次分析选取的版本为HTTPcore中4.4.10版本的源代码，具体包含了HTTP方法中的GET等通信方法。还未具体确定要选择的功能分析，代码结构仍然在学习阶段。