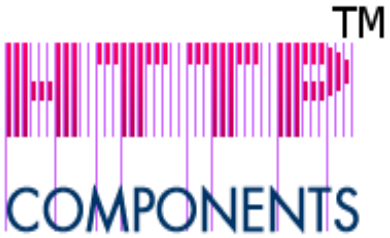
Report on Httpcomponents

张宇轩, 2017K8009908041



Contents

1 Preface

由于先前的课程中并没有讲过关于网络与信息方面的知识，笔者之也并没有接触过。而这次的选题是HttpComponents，涉及到大量的网络方面的知识，所以在准备这篇报告的第一阶段笔者的主要精力是放在研究一些基本的网络方面的知识。要讲到HttpComponents，就先要讲一下关于http的各种知识。

2 The First Part

第一阶段解读的重点放在了连接http周边知识上。

2.1 Related Introduction

2.1.1 http

从万维网（World Wide Web ）服务器传输超文本到本地浏览器，基于请求与响应，无状态的，应用层的协议。 从本地角度去看http协议，就相当于浏览器访问一个url，然后就得到相应的web页面 从服务器角度去看http协议，就相当于客户端（浏览器）链接远程http服务器，服务器返回数据，浏览器接收、解析数据之后显示出来。

2.1.2 https

利用SSL（此处：基于应用层的访问控制）进行传输层加密的http传输协议，更加安全。

https = http + SSL

2.1.3 Proxy Server

代理服务器：代理个人网络从互联网服务商那里获取信息（通过在二者间建立非直接的连接），可以保证安全。

2.1.4 Cookies

网站保存在用户端的含有用户信息以及行为的文本，用以弥补http协议的无状态性。举两个例子:

1.网站上的记住密码。网站将用户的身份和密码经过加密cookies存在用户硬盘中，下次登录的时候就不用手动输密码了。
2.上下文相关网址。如果上一个页面会对下一个页面产生影响，那次是就必须使用cookies。
比如一个购物网站，上一个页面选好商品，接下来的页面进行支付，支付页面就必须知道上一个页面中购买了什么东西，此时就需要cookies。

2.1.5 Request Format

客户端发送一个请求到服务器的请求消息格式如 Figure 1 所示（后面还会说到），请求头部的最后会有一个空行，表示请求头部结束，接下来为请求数据。

Figure 1: Request Format

2.1.6 URL

URL(Uniform Resource Locator，统一资源定位符)：一种资源位置的抽象唯一识别方法。

URL组成：<协议>://<主机>:<端口>/<路径> （端口和路径可以省略，端口默认80），如 Figure 2 所示：

另外需要注意URI(Uniform Resource Identifier，统一资源标识符)，用来表示web上每一种可用的资源，与URL不一样。

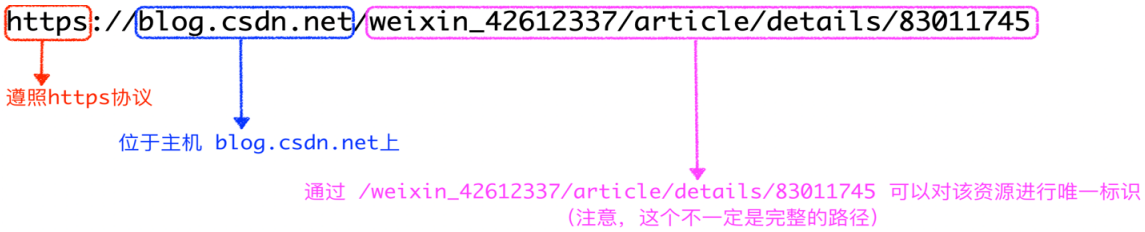


Figure 2: Request Format

2.1.7 Http Request Methods

Http 协议共有9种请求方法，如 Figure 3 所示，其中最常用的两种方法是 GET 和 POST，其对比如 Figure 4 所示。

GET - 从指定的资源请求数据。

POST - 向指定的资源提交要被处理的数据。

方法	描述
GET*	向特定资源发出请求 请求指定的页面信息，并返回实体主体。
HEAD	类似于 GET 请求，响应体不会返回，获取包含在小消息头中的原信息 返回的响应中没有具体的内容，用于获取报头。
POST	向指定资源提交数据进行处理请求（例如提交表单或者上传文件），数据被包含在请求体中。 可能会导致新的资源的建立或已有资源的修改。
PUT*	向指定资源位置上传最新内容 从客户端向服务器传送的数据取代指定文档的内容
PATCH	是对 PUT 方法的补充，用来对已知资源进行局部更新 。
DELETE	请求服务器删除指定的页面（requestURL对应资源）。
CONNECT	HTTP/1.1 协议中预留给能够将连接改为管道方式的代理服务器。
OPTIONS	返回服务器针对特定资源所支持的HTML请求方法，或向web服务器发送测试服务器功能 允许客户端查看服务器的性能。
TRACE	回显服务器收到的请求，主要用于测试或诊断。

Figure 3: Request Format

	get	post
cache	请求可被缓存（主动cache）	请求不会被缓存
刷新	无影响	重新提交请求（浏览器提醒）
编码	只有url编码 application/x-www-form-urlencoded	application/x-www-form-urlencoded 或 multipart/form-data
	请求只应当用于取回数据	
数据长度	请求有长度限制	请求对数据长度没有要求
历史记录	请求保留在浏览器历史记录中	请求不会保留在浏览器历史记录中

Figure 4: Get and Post

本文要介绍的HttpComponents就是用于提供对于http服务器的访问功能。

2.2 About HttpComponents

HttpComponents：用于提供对于http服务器的访问功能的超文本传输协议，其对HTTP底层协议进行了很好的封装。

在构建HTTP客户端或者服务器端应用中很常见，比如WEB浏览器、爬虫、HTTP代理、WEB服务库、基于调整或扩展HTTP协议的分布式通信系统 etc.

2.2.1 Major Function

实现http方法：GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH

对https协议的支持（https = http + SSL）

对代理服务器的支持

对cookies的支持

支持在特定的执行上下文（HTTP上下文）中执行HTTP消息——http不行（ 无状态、面向响应请求 ）

2.2.2 Component Structure

HttpComponents 的组件结构如 Figure 5 所示，包含 HttpComponents Core 和 HttpComponents AsyncClient，本次分析的是 HttpComponents Core。

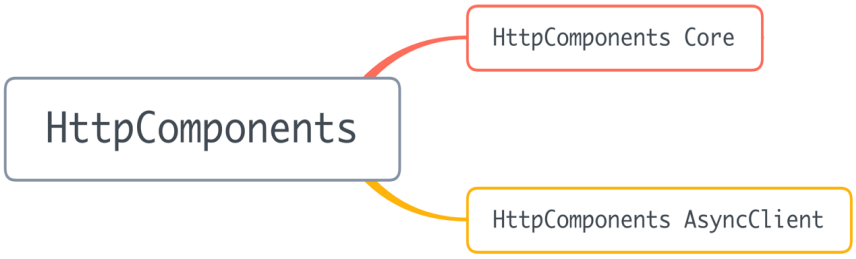


Figure 5: Component Structure of HttpsComponents

2.3 HttpComponents Core

HttpComponents Core 简称 HttpCore，其实现基本http协议的组件，用于搭建客户端和服务器端的http服务。HttpCore 希望在实现基本的http功能的基础上提高性能与平衡性，如 Figure 6 所示：

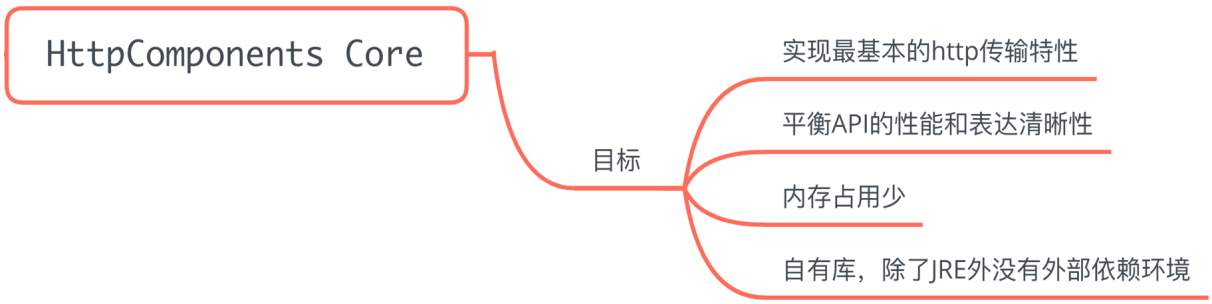


Figure 6: Goal of HttpCore

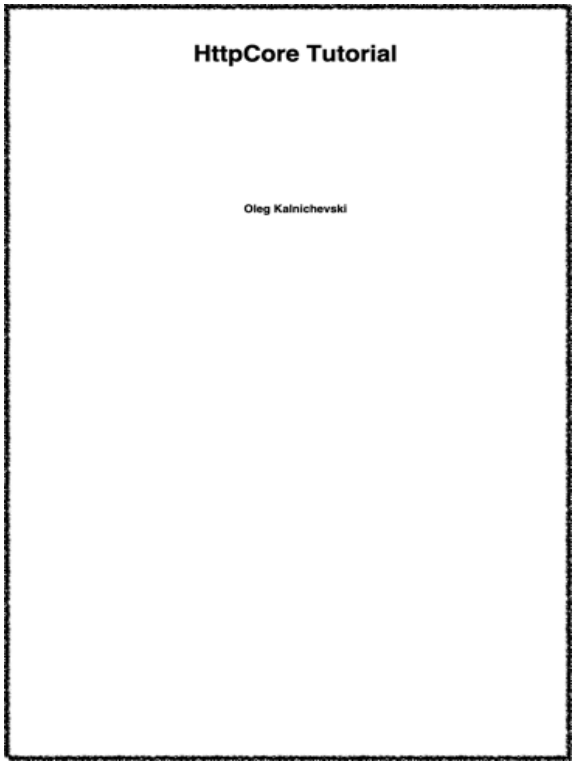


Figure 7: Guide Book

官方文档： httpcore-tutorial http://hc.apache.org/httpcomponents-core-ga/tutorial/pdf/httpcore-tutorial.pdf

2.3.1 Scope of HttpCore

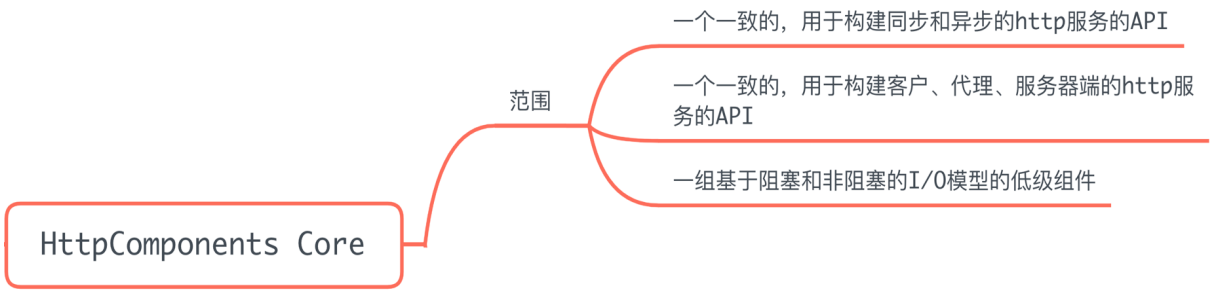


Figure 8: Scope of HttpCore

2.3.2 HttpCore API

HttpCore 中共包含了17个包，其功能如 Figure 9 所示:

Packages	
org.apache.http	Core HTTP component APIs and primitives.
org.apache.http.annotation	Provides annotations for public interface definitions
org.apache.http.concurrent	Core concurrency APIs.
org.apache.http.config	Core configuration APIs.
org.apache.http.entity	Core HTTP entity implementations.
org.apache.http.impl	Default implementations of HTTP connections for synchronous, blocking communication.
org.apache.http.impl.bootstrap	Embedded server and server bootstrap.
org.apache.http.impl.entity	Default implementations of entity content strategies.
org.apache.http.impl.io	Default implementations of message parses and writers for synchronous, blocking communication.
org.apache.http.impl.pool	Default implementations of client side connection pools for synchronous, blocking communication.
org.apache.http.io	HTTP message parser and writer APIs for synchronous, blocking communication.
org.apache.http.message	Core HTTP message components, message element parser and writer APIs and their default implementations.
org.apache.http.params	Deprecated.
org.apache.http.pool	Client side connection pools APIs for synchronous, blocking communication.
org.apache.http.protocol	Core HTTP protocol execution framework and HTTP protocol handlers for synchronous, blocking communication.
org.apache.http.ssl	Utility classes for trust and key material management and TLS/SSL context initialization.
org.apache.http.util	Core utility classes.

Figure 9: Goal of HttpCore

其中所有的类如图所示：

AbstractConnPool
AbstractHttpClientConnection
AbstractHttpEntity
AbstractHttpMessage
AbstractHttpParams
AbstractHttpServerConnection
AbstractMessageParser
AbstractMessageWriter
AbstractSessionInputBuffer
AbstractSessionOutputBuffer
Args
Asserts
BasicConnFactory
BasicConnPool
BasicFuture
BasicHeader
BasicHeaderElement
BasicHeaderElementIterator
BasicHeaderIterator
BasicHeaderValueFormatter

AbstractConnPool
AbstractHttpClientConnection
AbstractHttpEntity
AbstractHttpMessage
AbstractHttpParams
AbstractHttpServerConnection
AbstractMessageParser
AbstractMessageWriter
AbstractSessionInputBuffer
AbstractSessionOutputBuffer
Args
Asserts
BasicConnFactory
BasicConnPool
BasicFuture
BasicHeader
BasicHeaderElement
BasicHeaderElementIterator
BasicHeaderIterator
BasicHeaderValueFormatter

(a)

BasicHeaderValueParser
BasicHttpContext
BasicHttpEntity
BasicHttpEntityEnclosingRequest
BasicHttpParams
BasicHttpProcessor
BasicHttpRequest
BasicHttpResponse
BasicLineFormatter
BasicLineParser
BasicListHeaderIterator
BasicNameValuePair
BasicPoolEntry
BasicRequestLine
BasicStatusLine
BasicTokenIterator
BHttpConnectionBase
BufferedHeader
BufferedHttpEntity
BufferInfo

(b)

3 The Second Part

4 The Third Part