HTTP 版本 1.1

Content-Type: text/x-rst

Created: 2019-10-10

Author: Kris(New Name) 以前的名字不对，特意改了一下，不要见怪哈哈哈 ⭐♥⭐

强烈建议阅读HTTP相关知识，对于爬虫来说是非常重要的一个基础知识

* 合抱之木，生于毫末：九层之台，起于垒土；千里之行，始于足下。——老子

如果文中有错误的地方，还请指正或者email联系我!

HTTP知识较多，一次没做全面，后续补上，还请谅解！

介绍

概念：HTTP(Hypertext Transfer Protocol) 中文超文本传输协议，是一种分布式、协作式和超媒体信息系统的应用层协议。HTTP是万维网的数据通信基础。

HTTP请求的内容通称为“资源”，“资源“的概念非常的广泛，它可以是一个文档，一幅图片，音乐等等所有你能想到的格式。

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。

--- 本段内容摘录自wiki百科 (https://zh.wikipedia.org/wiki/超文本传输协议)

争议（拓展）

自HTTP被引进过来我们一直都对HTTP翻译的名称有着较大的争议即超文本传输协议，也就是英文Transfer的解释有关，下面让我们看一下 transfer 的标准释义：

v. 转让；转接；移交；转移（地方）；（使）换乘；转存，转录；调动（工作）；

这里我们就会发现一个很大的问题，Transfer 并没有传输的含义，按照这里的翻译我们其实可以叫它超文本转移协议，也就是说wiki百科的中文翻译是完全错误的，什么？What fuck? 对的，其实HTTP协议一直都被我们理解错了，这才是真正HTTP的含义，不信我们可以再来看一下Fielding博士的论文《Architectural Styles and the Design of Network-based Software Architectures》里面是怎么描述的HTTP：

>>> 6.5.3 HTTP is not a Transport Protocol

翻译：6.5.3 HTTP 不是传输协议

>>> HTTP is not designed to be a transport protocol. It is a transfer protocol in which the messages reflect the semantics of the Web architecture by performing actions on resources through the transfer and manipulation of representations of those resources. It is possible to achieve a wide range of functionality using this very simple interface, but following the interface is required in order for HTTP semantics to remain visible to intermediaries.

翻译：HTTP并非设计为传输协议。它是一种转移协议，其中消息通过传输和操纵那些资源的表示对资源执行操作来反映Web架构的语义。使用此非常简单的界面可以实现广泛的功能，但是为了使HTTP语义对中间人保持可见，必须遵循该界面。

好了，对于HTTP的争议问题，就先说这么多，想要拓展的伙伴可以自行去网上找更多相关此争议的内容！

HTTP版本问题：

HTTP协议已经演化了很对个版本,它们中的大部分都是向下兼容的。在 RFC 2145 中描述了HTTP版本号的用法。客户端在请求的开始告诉服务器它采用的协议版本号，而后者则在响应中采用相同或者更早的协议版本。 - 此段均来自维基百科

HTTP 0.9

已过时。只接受GET一种请求方法，没有在通讯中指定版本号，且不支持请求头。由于该版本不支持POST方法，因此客户端无法向服务器传递太多信息。

HTTP 1.0

HTTP 1.0 最早在网页中使用实在1996年，这是第一个在通讯中指定版本号的HTTP协议版本，至今仍被广泛采用，特别是在代理服务器中。

HTTP 1.1

持久连接被默认采用，并能很好地配合代理服务器工作。还支持以管道方式在同时发送多个请求，以便降低线路负载，提高传输速度。HTTP/1.1相较于HTTP/1.0协议的区别主要体现在：

1. 缓存处理
2. 带宽优化以及网络连接的使用
3. 错误通知的管理
4. 消息在网络中发送
5. 互联网地址维护
6. 安全性以及完整性

HTTP 2

当前的版本，于2015年5月作为互联网标准正式发布，HTTP/2 的主要目标是通过支持完整的请求与响应复用来减少延迟，通过有效压缩 HTTP 标头字段将协议开销降至最低，同时增加对请求优先级和服务器推送的支持。 为达成这些目标，HTTP/2 还给我们带来了大量其他协议层面的辅助实现，例如新的流控制、错误处理和升级机制。上述几种机制虽然不是全部，但却是最重要的，每一位网络开发者都应该理解并在自己的应用中加以利用。HTTP/2 没有改动 HTTP 的应用语义。 HTTP 方法、状态代码、URI 和标头字段等核心概念一如往常。 不过，HTTP/2 修改了数据格式化（分帧）以及在客户端与服务器间传输的方式。这两点统帅全局，通过新的分帧层向我们的应用隐藏了所有复杂性。 因此，所有现有的应用都可以不必修改而在新协议下运行。

* HTTP 2 相关内容摘录自Google Developers(<https://www.developers.google.com>)

HTTP请求方法

HTTP/1.1协议中共定义了八种方法（也叫“动作”）来以不同方式操作指定的资源：

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，例如在网络应用程序中。其中一个原因是GET可能会被网络蜘蛛等随意访问。参见安全方法

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”（元信息或称元数据）。

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。

PUT

向指定资源位置上传其最新内容。

DELETE

请求服务器删除Request-URI所标识的资源。

TRACE

回显服务器收到的请求，主要用于测试或诊断。

OPTIONS

这个方法可使服务器传回该资源所支持的所有HTTP请求方法。用'\*'来代替资源名称，向Web服务器发送OPTIONS请求，可以测试服务器功能是否正常运作。

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。通常用于SSL加密服务器的链接（经由非加密的HTTP代理服务器）。

方法名称是区分大小写的。当某个请求所针对的资源不支持对应的请求方法的时候，服务器应当返回状态码405（Method Not Allowed），当服务器不认识或者不支持对应的请求方法的时候，应当返回状态码501（Not Implemented）。

HTTP服务器至少应该实现GET和HEAD方法，其他方法都是可选的。当然，所有的方法支持的实现都应当符合下述的方法各自的语义定义。此外，除了上述方法，特定的HTTP服务器还能够扩展自定义的方法。例如：

PATCH（由 RFC 5789 指定的方法）

用于将局部修改应用到资源。

For everybody

本文档编写自Kris 2019-10-10文中大量引用wiki内容，文字较多，阅读可能会引起大量不适，不过还是强烈建议阅读完整，最起码读一遍，对HTTP基础有个认识，下节为更加详细的HTTP知识，有的读哈哈！

托更这么久准备重新来过，认真写文档，所以使用了word，以后有代码的部分还是会用Markdown专门引用，请放心♥，感谢支持啦~