



微信和小程序开发

HTTP协议和Node.js

HTTP协议

- HTTP是HyperText Transfer Protocol的缩写，被称为‘超文本传输协议’。
- HTTP是应用层协议。
- HTML就是超文本。但是，现在HTTP传输的内容早已不限定于HTML的范围。
- HTTP和HTML本来没有什么必然的联系，蒂姆·伯纳斯·李最早结合它们实现了最初的Web体系（Web是World Wide Web的简称）。

HTTP协议

- 从HTTP/0.9到HTTP/2都基于TCP协议。所以HTTP保证了可靠传输。
- HTTP/3基于QUIC， QUIC基于UDP。 UDP不保证可靠传输， 而QUIC基于UDP实现了可靠传输。
- 目前HTTP/3没有发布正式标准， 还在测试阶段。预计需要很长的路要走。

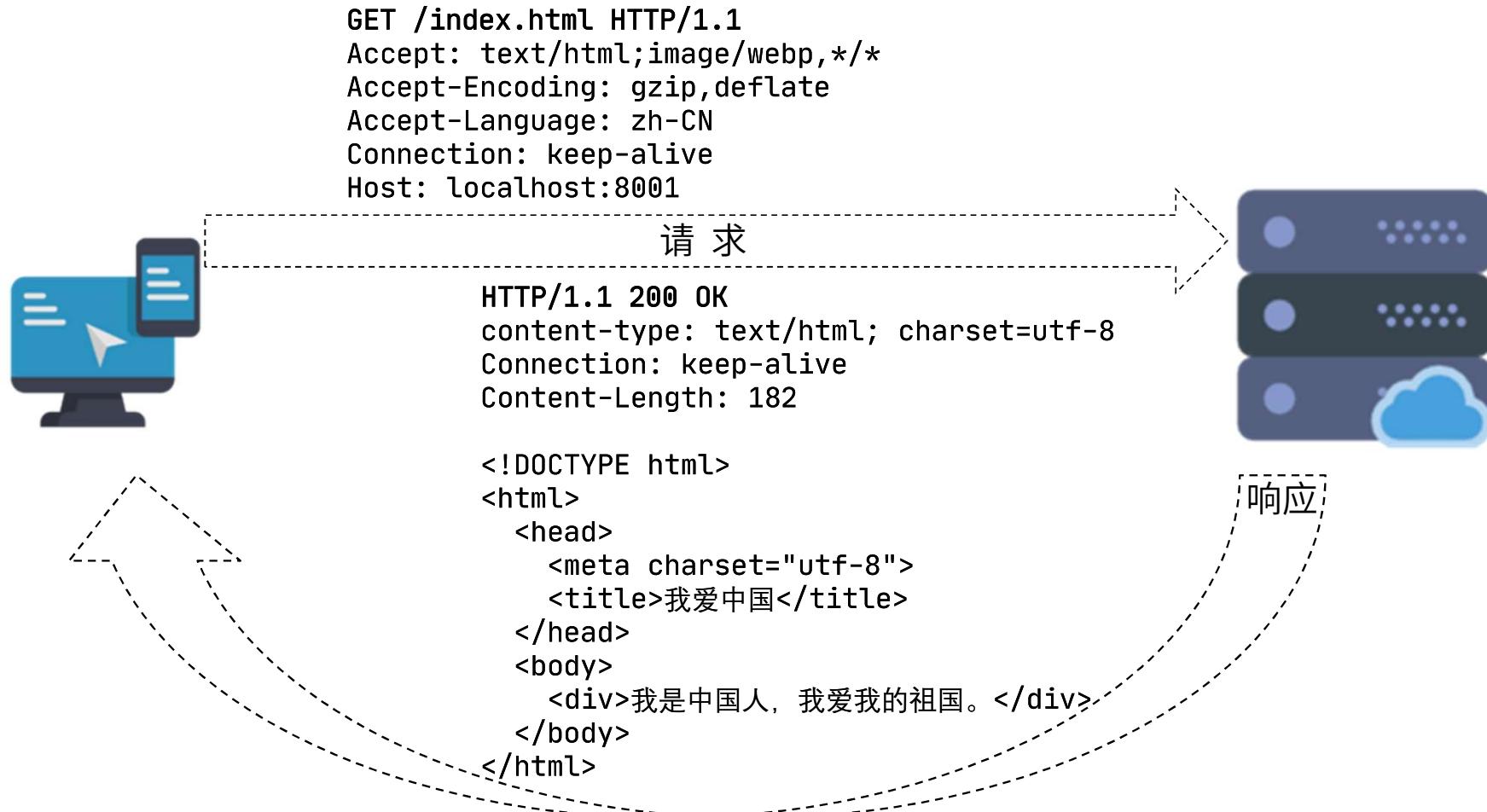
HTTP协议

- **HTTP/1.1**以前的版本传输数据的起始行和消息头采用**ASCII**文本，包装后的协议数据可以直接查看。
- **HTTP/2**采用了二进制传输，需要程序解析后才可读。
- 从**HTTP/2**开始，不会再有小版本号，所以现在最新的版本是**HTTP/3**，下一个就是**HTTP/4**。当然这个是标准协会规定的，谁也无法保证以后会不会再改回小版本号。

HTTP协议和通信软件

- 基于HTTP通信，需要客户端和服务端都实现HTTP协议。
- 使用最多的客户端就是浏览器。
- HTTP的服务端有很多，古老的有Apache，目前使用比较多的是Nginx，也有lighttpd。
- Node.js自带高性能的HTTP服务，支持HTTP、HTTPS、HTTP/2，和Nginx同样都基于异步IO，在Linux上底层都基于epoll函数族。

HTTP协议通信示例



请求的格式

- 最开始的一行是首行，也被称为起始行。起始行描述了：请求类型 URL 协议版本。
- 起始行之后，要通过\r\n结束。早期一些不标准的程序实现仅仅使用\n进行换行。
- 紧接着是首部，首部是用:分割的key-value格式。每行结束使用\r\n。
- 首部结束后，有一个\r\n的空行。
- 之后是消息主体。一般被称为Body，不过不是所有的消息都需要携带Body数据。

响应的格式

- 响应数据的起始行描述了：协议版本 状态码 状态码对应的描述信息。
- 首部同样是以分割的**key-value**值，描述了返回数据的相关信息。
- 紧接着一个\r\n空行后是返回的主体数据。

请求类型

- HTTP目前提供了8个请求类型：
 - GET 表示要获取资源，不携带请求体数据。
 - POST 表示要创建新的资源，但是早期应用存在很多误用，只要提交数据全部使用POST。
 - PUT 表示更新资源，要携带请求体数据，和POST不同的是，资源已经创建了。
 - DELETE 表示删除资源，请求体数据是可选的。
 - OPTIONS 表示发起一个预检请求，浏览器通常会自动发起，比如DELETE操作之前。
 - PATCH 表示更新资源的部分数据。
 - HEAD 只请求头部信息，包括起始行，但是没有响应主体，和GET类似，有时也用GET实现。
 - TRACE 主要用于诊断，主要是用于debug场景。
- 其中POST、PUT、PATCH需要携带请求体数据，DELETE请求可选。其他请求都不能带请求体数据。

响应状态码

- 响应状态码描述了通信返回的状态信息。状态码用3位数字表示。
 - 1xx 信息描述状态码，但是使用相对较少，除了使用update字段进行websocket协议切换。
 - 2xx 表示请求成功，但是不同的数字代表不同的请求状态，最多使用200表示数据成功返回。
 - 3xx 重定向，通常要提供一个Location头部字段描述资源的新地址，浏览器会自动跳转。
 - 4xx 客户端出错，比如请求的资源不存在返回404，请求体太大返回413。
 - 5xx 服务端出错，比如程序执行错误返回500，502表示网关错误，503表示服务暂时不可用。