

# HTTP使用指南 | 青训营笔记

用户835936271820   2022-08-05   👁 42   ⌚ 阅读6分钟

关注



📄 复制

🔄 重新生成

这篇文章是关于 HTTP 的使用指南，包括初识 HTTP 及其特点，协议分析（如报文结构、发展历程、状态码等），场景分析（静态资源、登录），实际应用（浏览器与 node 中使用，如 AJAX 等），还提到了解更多的扩展内容（如 WebSocket、QUIC 等）。

关联问题: [HTTP有哪些优化方法](#) [如何处理跨域问题](#) [怎样利用CDN缓存](#)

基于该文章内容继续向AI提问

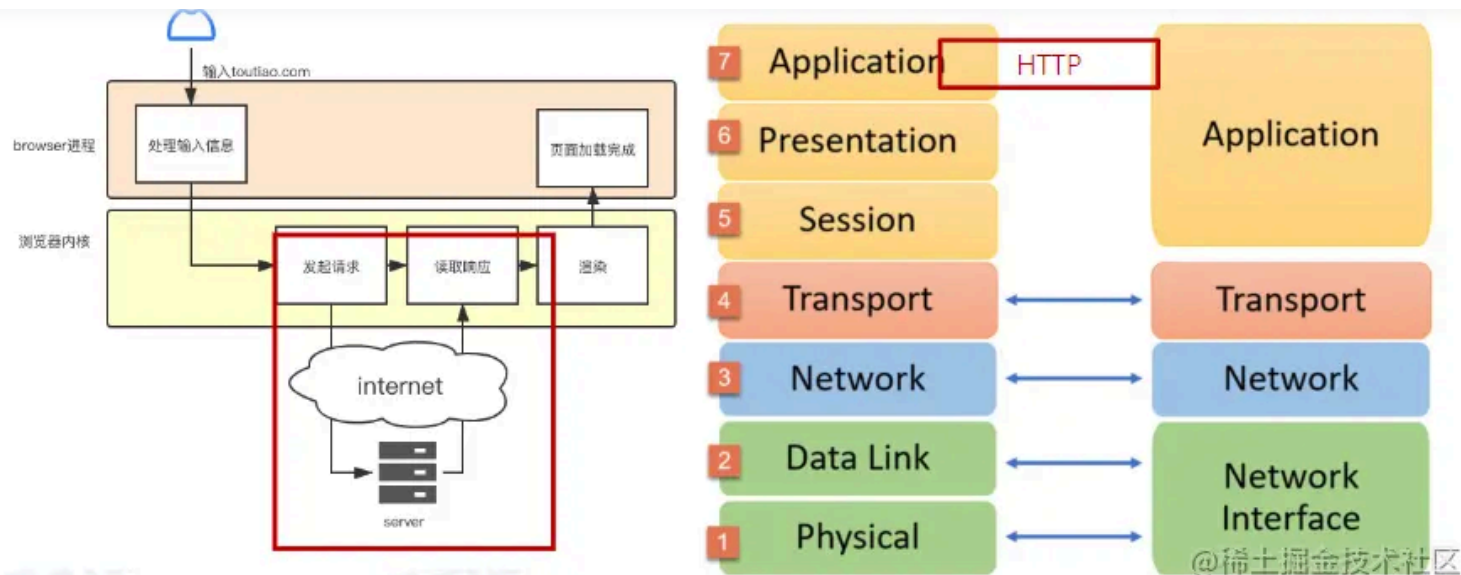
## HTTP使用指南 | 青训营笔记

这是我参加【第四届青训营】笔记创作活动的第11天



背景知识：什么是HTTP，其基本特点

浏览器的地址栏输入字符串到页面渲染，这个过程中发生了什么？

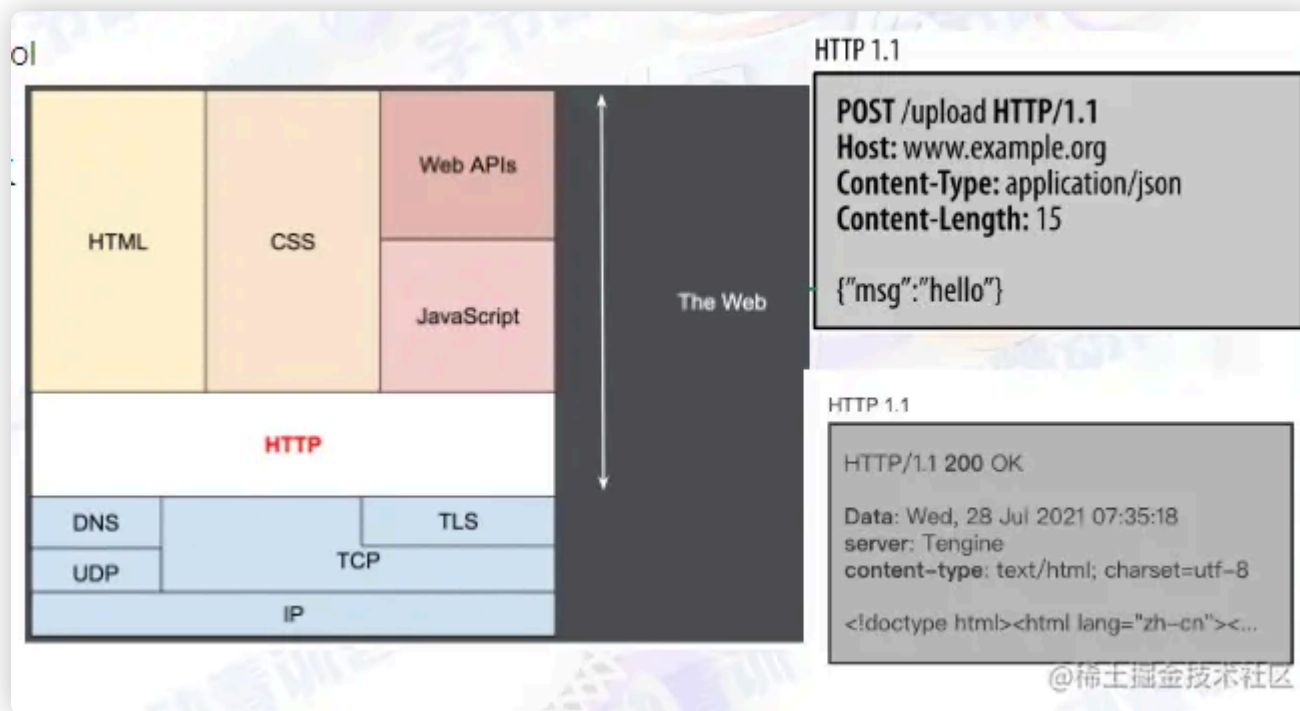


流程分析：在地址栏输入字符串，识别出意图是要访问这个地址，搜索进入头条，得到请求地址→发出真实请求，经过网络到达服务器→读取响应→渲染→页面加载完成。

初识HTTP——什么是HTTP

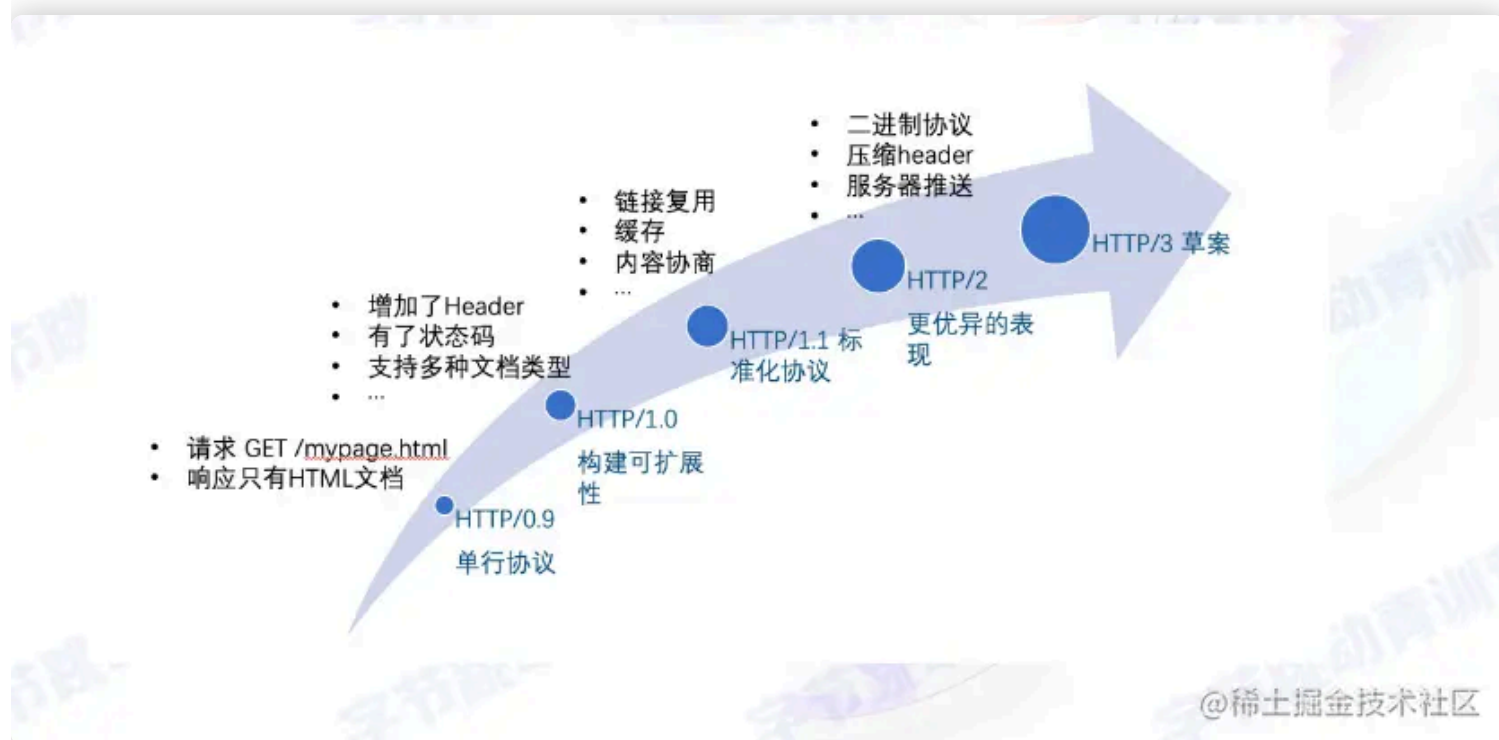
Hyper Text Transfer Protocol超文本传输协议

特点：应用层协议，基于TCP协议；请求 响应；简单可扩展；无状态。



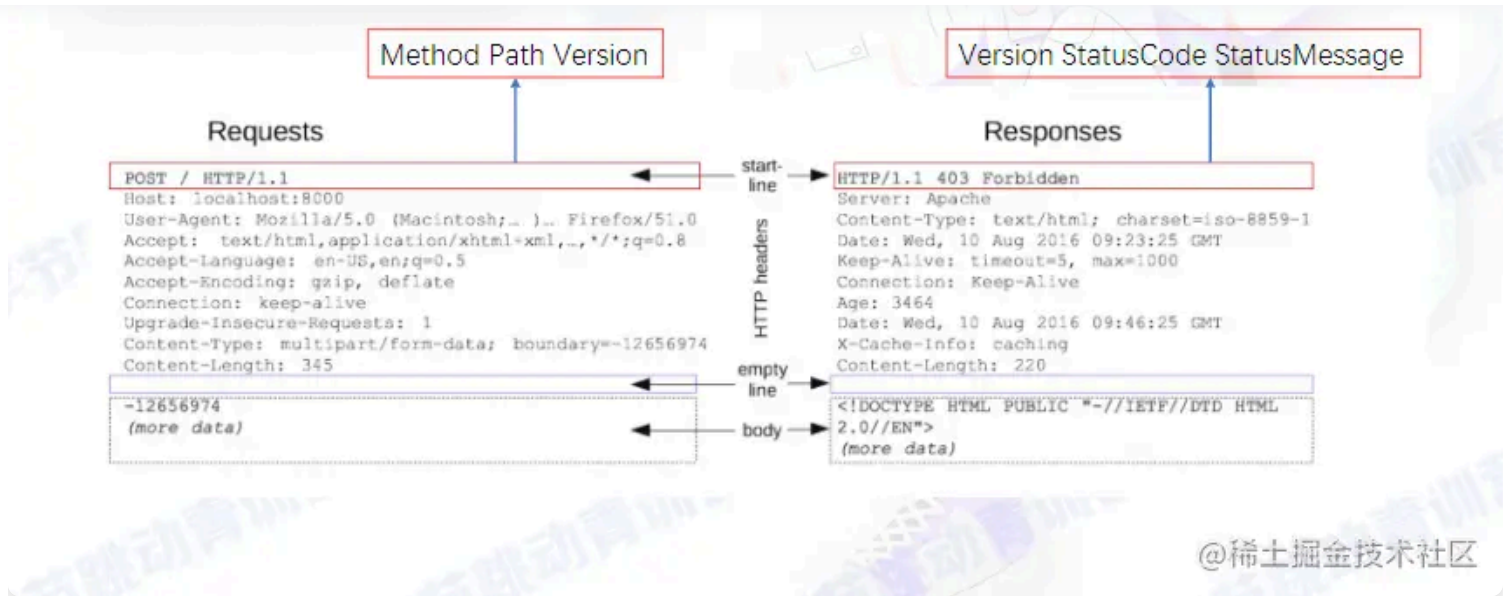
报文结构，发展历程

单行协议HTTP/0.9 → 构建可扩展HTTP/1.0 → 标准化洗衣HTTP/1.1 → 更优异的表现HTTP/2 → 草案HTTP/3



## 协议分析-报文

## HTTP/1.1



## 1.请求报文

http的请求报文由：请求行、头部、空行、主体(请求数据)四个部分组成。其中请求行由请求方法字段、URL字段和HTTP协议版本字段3个字段组成，它们用空格分隔。

常用的请求类型：

Method	
GET	请求一个指定资源的表示形式. 使用GET的请求应该只被用于获取数据.
POST	用于将实体提交到指定的资源. 通常导致在服务器上的状态变化或副作用
PUT	用请求有效载荷替换目标资源的所有当前表示
DELETE	删除指定的资源
HEAD	请求一个与GET请求的响应相同的响应, 但没有响应体
CONNECT	建立一个到由目标资源标识的服务器的隧道。
OPTIONS	用于描述目标资源的通信选项。
TRACE	沿着到目标资源的路径执行一个消息环回测试。
PATCH	用于对资源应用部分修改。

@稀土掘金技术社区

GET用来做获取资源请求；POST用来做提交的请求；PUT已有载体的替换；DELETE删除；HEAD/CONNECT用的相对较少——HEAD与GET请求基本相似，但是没有具体的响应体；CONNECT建联过程；OPTIONS前置请求；TRACE测试；PATCH修改。

一个响应由状态行、响应头部、空行和响应数据4部分组成。

### 3.请求行类型

safe(安全的): 不会修改服务器的数据的方法: GET HEAD OPTIONS

idempotent(幂等): 同样的请求被执行一次与连续执行多次的效果是一样的, 服务器的状态也是一样的

所有safe的方法都是idempotent的: GET HEAD OPTIONS PUT DELETE

### 状态码

#### RESTful API

RESTful API : 一种API设计风格; REST - Representational State Transfer

(1) 每一个URI代表一种资源;

(2) 客户端和服务端之间, 传递这种资源的某种表现层;

(3) 客户端通过HTTP method, 对服务器端资源进行操作, 实现"表现层状态转化"。

报文结构及字段的介绍

### 常用请求头

### 常用响应头

## 缓存

先看强缓存是否新鲜，如果不新鲜先校验ETag是否存在，若有发起请求，验证是不是最新的，新则返回304，用本地缓存返回，不新则返回200状态码，然后把结果返回。

## HTTP/2

概述：更快、更稳定、更简单

帧(frame)：HTTP/2通信最小单位，每个帧都包含帧头，至少也会标识出当前帧所属的数据流；二进制。

消息：与逻辑请求或响应消息对应的完整的一系列帧。

数据流：已建立的连接内的双向字节流，可以承载一条或多条消息。

交错发送，接收方重组织。

HTTP/2特性：HTTP/2连接都是永久的，而且仅需要每个来源一个连接

流控制：阻止发送方向接收方发送大量数据的机制。

服务器推送

## HTTPS概述

HTTPS：Hypertext Transfer Protocol Secure

经过TSL/SSL加密

对称加密：加密和解密都是使用同一个密钥；



静态资源、登陆

场景分析：1.打开chrome；

2.输入 [www.toutiao.com](http://www.toutiao.com)；

3.打开控制台 右键->检查F12 4.切换到network

静态资源：

状态码200一定发起了请求吗？ from disk cache从本地缓存拿到的响应

静态资源方案：缓存 + CDN + 文件名hash

CDN：Content Delivery Network

通过用户就近性和服务器负载的判断，CDN确保内容以一种极为高效的方式为用户的请求提供服务。

场景分析——-登录

业务场景：表单登录；扫码登录

技术方式：SSO

表单登录

为什么会有options的请求？跨域；cross-origin

什么是跨域

scheme/host/port任意一个区域有区别都是跨域。

最后一行没有写出端口号，默认443端口号，也是同域的

跨域请求是怎么被处理的：

跨域解决方案：

CORS；

代理服务器（同源策略是浏览器的安全策略，不是HTTP的）；

Iframe（诸多不便）。

1.向什么地址做了什么动作？

使用POST方法；目标域名：[sso.toutiao.com](https://sso.toutiao.com)；目标path/quick\_login/v2/

2.携带了哪些信息，返回了哪些信息？

携带信息：

Post body，数据格式为form；希望获取的数据格式为json；已有的cookie

返回信息：数据格式json；种cookie的信息。

鉴权





## 实战——浏览器篇

### AJAX之XHR

「**AJAX概述**」：是一种在「**无需重新加载**」整个网页的情况下，能够更新部分网页的技术，也称「**懒加载**」。通过在后台与服务器进行数据交互，使网页实现异步更新，也就是在不加载整个网页的情况下，对网页的某部分进行更新。

XHR：XMLHttpRequest 的简写，XMLHttpRequest 对象提供了对 HTTP 协议的完全的访问，包括做出 POST 和 HEAD 请求以及普通的 GET 请求的能力。XMLHttpRequest 可以同步或异步地返回 Web 服务器的响应，并且能够以文本

或者一个DOM文档的形式返回内容。

#### 「AJAX的请求流程」：

1. 创建ajax对象
2. 设置请求,发送请求地址，发送请求方式
3. 发送数据
4. 设置一个监听事件、监听后台是否返回请求
5. 处理数据

### AJAX之Fetch

Fetch是XMLHttpRequest的升级版。

#### 「Fetch特点」：

- 使用Promise。
- 模块化设计，Response、Request、Header对象。
- 通过数据流处理对象，支持分块读取。

默认模块，无需安装其他依赖；

功能有限/不是十分友好。

常用的请求库：`axios`

支持浏览器、`nodejs`环境；丰富的拦截器

## 用户体验

网络优化

稳定性



不止HTTP协议一个选择

扩展——通信方式

WebSocket

浏览器与服务器进行全双工通讯的网络技术

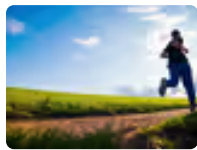
典型场景：实时性要求高，例如聊天室

URL使用`ws://`或`wss://`等开头

扩展

QUIC: Quick UDP Internet Connection

本文收录于以下专栏



青训营笔记

专栏目录

谨以此，记录一下青训营的学习。

1 订阅 · 17 篇文章

订阅

上一篇 Web开发的安全之旅 | 青训营笔记

下一篇 Typescript入门 | 青训营笔记

评论 0

登录 / 注册 即可发布评论!

暂无评论数据

目录

收起 ^

HTTP使用指南 | 青训营笔记

这是我参加【第四届青训营】笔记创作活动的第11天

- 01初识
- 02协议分析
- 03场景分析
- 04实际应用
- 05了解更多

相关推荐

[ HTTP使用指南 | 青训营笔记]

40阅读 · 0点赞

HTTP使用指南|青训营笔记

64阅读 · 0点赞

# [ HTTP使用指南 | 青训营笔记]

35阅读 · 0点赞

HTTP使用指南 | 青训营笔记

18阅读 · 0点赞

为你推荐

HTTP实用指南 | 青训营笔记

小鱼干soo

2年前

 844

 1

 评论

笔记

HTTP实用指南|青训营笔记

Victor252

2年前

 519

 1

 评论

笔记

总结：HTTP指南！以及相关的实践分析与代码分析（个人总结与课程笔记结合体） | 青训营

飘尘

1年前

 1.3k

 4

 评论

青训营笔记

HTTP 使用指南 | 青训营

无敌矿工

1年前

 484

 点赞

 评论

青训营笔记

## HTTP协议请求格式详解

磨刀石 3年前 3.7k 2 评论

HTTP

## 02-Node.js Http协议

cv开发工程师 1年前 274 点赞 1

前端 Node.js

## 【青训营】- 前端开发HTTP实用指南

吃了一天糖葫芦 3年前 578 2 评论

笔记

## 【Java Web】探索HTTP请求与响应结构

林小新 1年前 221 点赞 评论

Java 后端

## 网络协议-应用层（HTTP/HTTPS）

Potato\_土豆 2年前 1.3k 6 4

前端 HTTP HTTPS

## HTTP协议请求（request）（上）

also\_lucky 1年前 264 1 评论

后端 Java 程序员

## 揭秘 HTTP 请求过程

SeanMa 1年前 1.6k 3 评论

网络协议 HTTP HTTPS

## HTTP协议的特点、HTTP报文的组成部分

火星飞鸟 3年前 2.2k 9 2

HTTP 网络协议

## 什么是apache? 以及apache配置

菜就多练a 6月前 812 3 1

Linux 云计算 Apache

## 万字长文，掌握必备网络知识（下篇）来了解一下HTTP协议的相关知识

LuciferHuang 3年前 719 14 评论

网络协议