



HTTP 押题

GET 和 POST 的区别有哪些？

HTTP 缓存有哪些方案？

HTTP 和 HTTPS 的区别有哪些？

HTTP/1.1 和 HTTP/2 的区别有哪些？

TCP 三次握手和四次挥手是什么？

说说同源策略和跨域

Session、Cookie、LocalStorage、SessionStorage 的区别



📱 扫码购买 [《前端押题》视频课程](#)

😄 让您面试无忧

😊 绝对物超所值

GET 和 POST 的区别有哪些？

区别一：幂等性

1. 由于 GET 是读，POST 是写，所以 GET 是幂等的，POST 不是幂等的。


2. 由于 GET 是读，POST 是写，所以用浏览器打开网页会发送 GET 请求，想要 POST 打开网页要用 form 标签。
3. 由于 GET 是读，POST 是写，所以 GET 打开的页面刷新是无害的，POST 打开的页面刷新需要确认。
4. 由于 GET 是读，POST 是写，所以 GET 结果会被缓存，POST 结果不会被缓存。
5. 由于 GET 是读，POST 是写，所以 GET 打开的页面可被书签收藏，POST 打开的不行。

区别二：请求参数

1. 通常，GET 请求参数放在 url 里，POST 请求数据放在 body（消息体）里。（这里注意老师的讲解）
2. GET 比 POST 更不安全，因为参数直接暴露在 URL 上，所以不能用来传递敏感信息。（xjb扯）
3. GET 请求参数放在 url 里是有长度限制的，而 POST 放在 body 里没有长度限制。（xjb扯）

区别三：TCP packet

1. GET 产生一个 TCP 数据包；POST 产生两个或以上 TCP 数据包。

 根据技术规格文档，GET 和 POST 最大的区别是语义；但面试官一般问的是实践过程中二者的区别，因此你需要了解服务器和浏览器对 GET 和 POST 的常见实现方法。

HTTP 缓存有哪些方案？

	缓存（强缓存）	内容协商（弱缓存）
HTTP 1.1	Cache-Control: max-age=3600 <u>Etag</u> : ABC	If-None-Match: ABC 响应状态码：304 或 200
HTTP 1.0	Expires: Wed, 21 Oct 2015 02:30:00 GMT Last-Modified: Wed, 21 Oct 2015 01:00:00 GMT	If-Modified-Since: Wed, 21 Oct 2015 01:00:00 GMT 响应状态码：304 或 200

面试官可能还会提到 `Pragma`，但 MDN 已经明确不推荐使用它。

更详细的内容可以看我的课程《[全面攻克 Web 性能优化](#)》中的《[缓存与内容协商](#)》视频。

HTTP 和 HTTPS 的区别有哪些？

HTTPS = HTTP + SSL/TLS (安全层)

区别列表

1. HTTP 是明文传输的，不安全；HTTPS 是加密传输的，非常安全。
2. HTTP 使用 80 端口，HTTPS 使用 443 端口。
3. HTTP 较快，HTTPS 较慢。
4. HTTPS 的证书一般需要购买（但也有免费的），HTTP 不需要证书。

HTTPS 的细节可以看网上的博客，比较复杂，难以记忆，建议写博客总结一下。

- [图解SSL/TLS协议 - 阮一峰的网络日志 \(ruanyifeng.com\)](https://ruanyf.com/blog/ssl-tls/)
- [HTTPS原理以及握手阶段](#)

HTTP/1.1 和 HTTP/2 的区别有哪些？

区别列表

1. HTTP/2 使用了**二进制传输**，而且将 head 和 body 分成**帧**来传输；HTTP/1.1 是字符串传输。
2. HTTP/2 支持**多路复用**，HTTP/1.1 不支持。多路复用简单来说就是一个 TCP 连接从单车道（不是单行道）变成了几百个双向通行的车道。
3. HTTP/2 可以**压缩 head**，但是 HTTP/1.1 不行。
4. HTTP/2 支持**服务器推送**，但 HTTP/1.1 不支持。（实际上没多少人用）

更详细的内容可以看我的课程《[全面攻克 Web 性能优化](#)》中的《[什么是多路复用](#)》视频。

TCP 三次握手和四次挥手是什么？

建立 TCP 连接时 server 与 client 会经历三次握手

1. **浏览器**向**服务器**发送 TCP 数据：SYN(seq=x)
2. **服务器**向**浏览器**发送 TCP 数据：ACK(seq=x+1) SYN(y)

3. 浏览器向服务器发送 TCP 数据：ACK(seq=y+1)

关闭 TCP 连接时 server 与 client 会经历四次挥手

1. 浏览器向服务器发送 TCP 数据：FIN(seq=x)
2. 服务器向浏览器发送 TCP 数据：ACK(seq=x+1)
3. 服务器向浏览器发送 TCP 数据：FIN(seq=y)
4. 浏览器向服务器发送 TCP 数据：ACK(seq=y+1)

为什么 2、3 步骤不合并起来呢？看起来是脱裤子放屁。

答案：2、3 中间服务器很可能还有数据要发送，不能提前发送 FIN。

TCP 的细节我作为一个十年的前端，也不太想去参透。

说说同源策略和跨域

同源策略是什么？

如果两个 URL 的协议、端口和域名都完全一致的话，则这两个 URL 是同源的。

```
http://www.baidu.com/s
http://www.baidu.com:80/ssdasdsadad
```

同源策略怎么做？

只要在浏览器里打开页面，就默认遵守同源策略。

优点

保证用户的隐私安全和数据安全。

缺点

很多时候，前端需要访问另一个域名的后端接口，会被浏览器阻止其获取响应。

比如甲站点通过 AJAX 访问乙站点的 /money 查询余额接口，请求会发出，但是响应会被浏览器屏蔽。

怎么解决缺点

使用跨域手段。

1. JSONP (前端体系课有完整且详细的介绍)

- 甲站点利用 script 标签可以跨域的特性, 向乙站点发送 get 请求。
- 乙站点**后端改造** JS 文件的内容, 将数据传进回调函数。
- 甲站点通过回调函数拿到乙站点的数据。

2. CORS (前端体系课有完整且详细的介绍)

- 对于简单请求, 乙站点在响应头里添加 `Access-Control-Allow-Origin: http://甲站点` 即可。

- 对于复杂请求, 如 PATCH, 乙站点需要:

- 响应 OPTIONS 请求, 在响应中添加如下的响应头

```
Access-Control-Allow-Origin: https://甲站点
Access-Control-Allow-Methods: POST, GET, OPTIONS, PATCH
Access-Control-Allow-Headers: Content-Type
```

- 响应 POST 请求, 在响应中添加 `Access-Control-Allow-Origin` 头。

- 如果需要附带身份信息, JS 中需要在 AJAX 里设置 `xhr.withCredentials = true`。

3. Nginx 代理 / Node.js 代理

- 前端 ⇒ 后端 ⇒ 另一个域名的后端

详情参考 [MDN CORS 文档](#)。

Session、Cookie、LocalStorage、SessionStorage 的区别

• Cookie V.S. LocalStorage

- 主要区别是 Cookie 会被发送到服务器, 而 LocalStorage 不会
- Cookie 一般最大 4k, LocalStorage 可以用 5Mb 甚至 10Mb (各浏览器不同)

• LocalStorage V.S. SessionStorage

- LocalStorage 一般不会自动过期 (除非用户手动清除)
- SessionStorage 在会话结束时过期 (如关闭浏览器之后, 具体由浏览器自行决定)

- Cookie V.S. Session

1. Cookie 存在浏览器的文件里，Session 存在服务器的文件里

2. Session 是基于 Cookie 实现的，具体做法就是把 SessionID 存在 Cookie 里

其他区别请在网上找高票答案看看，自己写文章总结一下。