平时开发过程中，你们常用的 HTTP 请求方法都有哪些啊？

**我**：是这样的，HTTP 1.1 协议中共定义了八种方法，有时也叫动作，来表明 Request-URL 指定的资源不同的操作方式：

* 在 HTTP 1.0 中，定义了三种请求方法：GET、POST 和 HEAD 方法。
* 在 HTTP 1.1 中，新增了五种请求方法：OPTIONS、PUT、DELETE、TRACE 和 CONNECT 方法。但我们常用的一般就是 GET 和 POST 请求。

**面试官**：嗯，那你说说 GET 和 POST 请求都有哪些区别呢？（果然进套了，看你小子有几斤几两）

**我**：（我假装被难住，然后思考了一会儿说）这个……没有特别去关注过，但是按照我的理解，大概有这么几种区别吧。

1. GET 请求在 URL 中传送的参数是有长度限制的，而 POST 没有；
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。而 POST 数据不会显示在 URL 中。是放在 Request body 中；
3. 对参数的数据类型，GET 只接受 ASCII 字符，而 POST 没有限制；
4. GET 请求参数会被完整保留在浏览器历史记录里。相反，POST 请求参数也不会被浏览器保留；
5. GET 请求只能进行 URL 编码（ application/x-www-form-urlencoded），而 POST 支持多种编码方式；
6. GET 请求会被浏览器主动缓存，而 POST 不会，除非手动设置；
7. GET 在浏览器回退时是无害的，而 POST 会再次提交请求。

面试官：没关注过，那你还背的一条不差。跟我搁这儿装杯呢？看我怎么教育你。

**面试官**：那 GET 请求有 Request body 吗？如果有的话参数可以像 POST 请求一样放在里面吗？

**我**：吼吼，看来有机会把我昨天精心准备的东西给他扯半小时了[手动挠头] 。让开，我要开始放大招了

其实吧，GET 和 POST 在本质上没有区别，都是 HTTP 协议中的两种发送请求的方法。而 HTTP 呢，是基于 TCP/IP 关于数据如何在万维网中如何通信的协议。

万维网：简称 WWW，是 World Wide Web 的简称，也称为 Web、3W 等。

HTTP 的底层是 TCP/IP，所以 GET 和 POST 的底层也是 TCP/IP。也就是说， GET/POST 都是 TCP 链接。

GET 和 POST 能做的事情是一样一样的。你要给 GET 加上 Request body，给 POST 带上 URL 参数，技术上是完全行的通的。

举个栗子吧：（嗯？栗子？好饿，待会儿面试完去买点）

TCP 就像汽车，我们用 TCP 来运输数据很可靠，从来不会发生丢件少件的现象。

但是如果路上跑的全是看起来一模一样的汽车，那这个世界看起来是一团混乱。送急件的汽车可能被前面满载货物的汽车拦堵在路上，整个交通系统一定会瘫痪。

为了避免这种情况发生，交通规则HTTP诞生了。HTTP 给汽车运输设定了好几个服务类别，包括 GET、POST、PUT 等等。

HTTP 规定，当执行 GET 请求的时候，要给汽车贴上 GET 的标签（设置 method为 GET），而且要求把传送的数据放在车顶上（URL 中）以方便记录。

如果是 POST 请求，就要在车上贴上 POST 的标签，并把货物放在车厢里（Request body 中）。

当然，你也可以在用 GET 时往车厢内偷偷藏点货物，但这并不不光彩。也可以在POST的时候在车顶上也放一些数据，也会让人觉得傻乎乎的。

HTTP 只是个行为准则，而 GET 和 POST 本质上就是 TCP 连接，并无差别。但是由于 HTTP 的规定和浏览器/服务器的限制，导致他们在应用过程中体现出一些不同。

面试官：哎呀，这小子还真的了解这块儿啊。看来是我误会他了，难道遇到了大佬？

**面试官**：你说的不错，那你刚才说的 URL 中传送参数的长度限制在 GET 和 POST 中都是怎么样的呢？

**我**：其实在 Web 中啊，还有另一个重要的角色——运输公司。

不同的浏览器 Client 端（发起 HTTP 请求）和服务器 Server 端（接受 HTTP 请求）就是不同的运输公司。

虽然理论上，你可以在车顶上无限的堆货物（URL 中无限加参数）。但是运输公司可不傻，装货和卸货也是有很大成本的。他们会限制单次运输量来控制风险，数据量太大对浏览器和服务器都是很大负担。

业界不成文的规定是，（大多数）浏览器通常都会限制 URL 长度在 2K 个字节，而（大多数）服务器最多处理 64K 大小的 URL。超过的部分恕不处理。

如果你用 GET 服务，在 Request body 偷偷藏了数据，不同服务器的处理方式也是不同的：有些服务器会帮你卸货读出数据，有些服务器直接忽略。

所以，虽然 GET 可以带 Request body，却不能保证一定能被接收到。

**面试官**：（看来理论确实是掌握的不错，让我考考他实际应用）那 GET 方法参数写法是固定的吗？

**我**：在约定中，我们的参数是写在 **?** 后面，用 **&** 分割。就像下面这样：

http:*//ip:port/test/getHelloWorld?username=langwang&age=26&sex=2*

我们知道，解析报文的过程是通过获取 TCP 数据，用正则等工具从数据中获取 Header 和 Body，从而提取参数。比如 **header** 请求头中添加 token，来验证用户是否登录等权限问题。

也就是说，我们可以自己约定参数的写法，只要服务端能够解释出来就行，万变不离其宗。

**面试官**：那么说来，是不是 POST 方法比 GET 方法更安全呢？

**我**：有人说 POST 比 GET 安全，因为数据在地址栏上不可见。

然而，**从传输的角度来说，他们都是不安全的**。因为 HTTP 在网络上是明文传输的，只要在网络节点上捉包，就能完整地获取数据报文。

其实，要想安全传输，就只有加密，也就是 HTTPS。

**面试官**：嗯，不错不错，看来你对 HTTP 协议这块儿还是有一定了解的，那么你知道GET、POST 请求发送的数据包有什么不同吗？

**我**：（看来这面试官是非要是把我问倒才满意啊？可惜可惜。看我套路他一波）实不相瞒，我上家公司加班比较多，最近刚签完离职，贵公司是我面的第一家公司，所以准备的不充分。这样吧，我大致谈谈我的理解吧，不对的地方您见谅。

**面**试官：没关系，按你的理解聊聊就行。（这小子没准备就说成这样，我的好好考虑一下，别错失了先机，失去一个能为公司加班的人才）

**我**：嗯嗯。是这样的，GET 请求时产生一个 TCP 数据包，POST 请求时产生两个 TCP 数据包。

* **GET**：浏览器会把 http header 和 data 一并发送出去，服务器响应 200（返回数据）；

**POST**：（这里存疑，POST不一定都是分两次发送，只有特殊需求才会使用100 Continue进行两次发送）浏览器先发送 header，服务器响应 100 continue，浏览器再发送 data，服务器响应 200 OK（返回数据）。

就像是 GET 只需要汽车跑一趟就把货送到了，而 POST 得跑两趟。

第一趟，先去和服务器打个招呼：老铁，我等下要送一批货来，你们准备接收一下哈。然后再回头把货送过去。

因为 POST 需要两步，理论上时间上消耗的要多一点。看起来 GET 比 POST 更有效。但并不是，后来发现原来是个坑。

在我看来：

* GET 与 POST 都有自己的语义，不能随便混用；
* 据研究，在网络环境好的情况下，发一次包的时间和发两次包的时间差别基本可以无视。而在网络环境差的情况下，两次包的 TCP 在验证数据包完整性上，有非常大的优点；
* 并不是所有浏览器都会在 POST 中发送两次包，Firefox 就只发送一次。我去年用 Chrome 浏览器测试发现也是只发送一次，所以我认为 GET、POST 性能差可以人为忽略。

**面试官**：嗯，你说的很不错，那你稍微等一下啊。

**我**：（难道还有变故？这波操作难道被发现了我在套路面试官？不应该啊）五分钟后，面试官拿着一杯饮料走了进来……

**面试官**：你尝尝这咖啡味道如何？（自古套路得人心，学好了）

我喝了一口，竖起大拇指说道：“面试官，这是正宗的卡布奇诺啊，我平时就喜欢喝，可惜太贵了”。

咦？我进来之前没有看到你们这有咖啡店呀，这是？

面试官嘴角上翘，微微一笑，露出了洁白的两颗大门牙说道：

我看你简历说你爱喝咖啡。巧了，我们公司有专门的水吧，福利之一就是是每天免费一杯咖啡或饮料。以后你可以经常喝了。

**我**：那您的意思是，我面试过了了吗？太好了！对了，水吧每天几点下班呐？跟开发一个点儿么？（不会是因为经常加班才免费喝咖啡吧，我得试探试探）

**面试官**：哈哈是啊，水吧一直都是按点下班，咱们公司如果有人加班的话，可以自己去水吧做着喝就行，加班时间都是免费的。你这儿没问题的话，我看下周一就入职吧。

**我**：好的，面试官，我回去考虑下，到时候和您联系，再见！

至此，本次关于 HTTP 请求方式，以及 GET 和 POST 的区别相关的面试就结束啦，感谢大家的观看！