

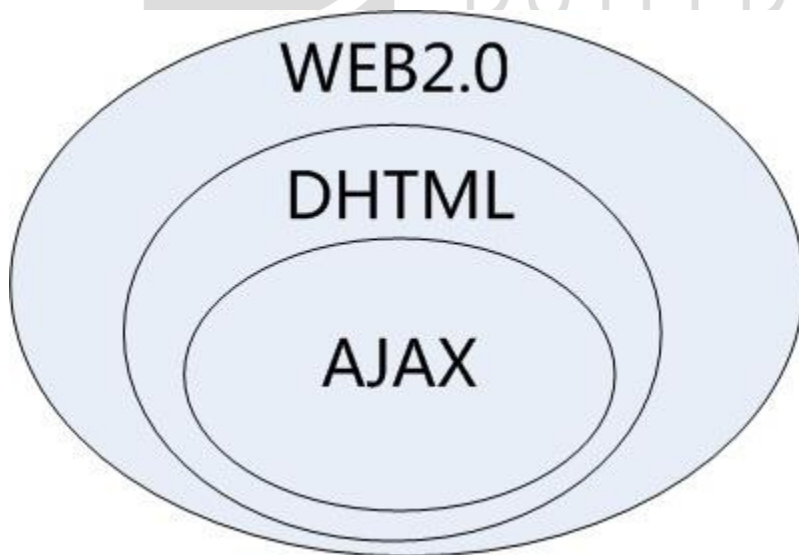
你了解 Ajax 么？常用的 ajax 优化有哪些？

AJAX 简介

AJAX 全称 Asynchronous JavaScript and XML（异步的 JavaScript 和 XML）。将 AJAX 中处理的一些异步信息或叫“即时”信息，保存在缓存中，不要每次去异步处理时，都去调用 DHTML 的元素信息。

正常情况下填好表单信息并提交后，整个表单信息会发送到服务器，服务器会将它转发给处理表单的脚本，通常是后台的 PHP 或 JAVA，后台脚本执行完成后服务器会发送回全新的页面信息。AJAX 正常会把 JavaScript 技术和 XMLHttpRequest 对象放在 Web 表单和服务端之间，当填好表单信息并提交后，会先使用 JavaScript 代码执行而不是直接发送给服务器，也就是说 JavaScript 代码会在后台发送请求到服务器，并且可以异步处理，即 JavaScript 代码在发送信息时，不用等待服务器的响应，可以继续发送其它的信息。

AJAX 与 WEB 和 DHTML 的关系如图所示。



DHTML 是一种使 HTML 页面具有动态特性的艺术，DHTML 是一种创建动态和交互 WEB 站点的技术集。对大多数人来说，DHTML 意味着 HTML、样式表和 JavaScript 的组合。

当发起主动 AJAX 请求时，用户可能仍须等待，所以必须优化请求，优化请求最主要的方式就是使响应可以缓存。

即当 AJAX 发送数据成功后，会把请求的 URL 和返回的响应结果信息保存在缓存中，这样下次调用 AJAX 发送相同请求时，会直接从缓存中把数据取出来，这样可以提高请求的响应速度。

通过 Ajax 缓存来提升页面性能

AJAX 缓存可以让请求对一些静态页面内容的信息处理的更迅速，比如图片、CSS 文件、JS 脚本等。可以让 AJAX 可缓存的响应头包括：Expires、Last-Modified 和 Cache-Control。

1) Expires

Expires 是通过判断内容是否被修改来确定是否使用浏览器缓存中的内容，如果我们知道内容何时修改，那么可以使用 Expires 响应头来处理。

http/1.0 中定义的 header，是最基础的浏览器缓存处理，表示资源在一定时间内从浏览器的缓存中获取资源，不需要请求服务器获取资源，从而达到快速获取资源，缓解服务器压力的目的。

在 response 的 header 中的格式为：Expires: Thu, 01 Dec 1994 16:00:00 GMT （必须是 GMT 格式）

应用：

1、可以在 html 页面中添加<meta http-equiv="Expires" content="Thu, 01 Dec 1994 16:00:00"/> 来给页面设置缓存时间。

2、对于图片、css 等文件则需要在 IIS 或者 apache 等运行容器中进行规则配置来让容器在请求资源的时候添加在 response 的 header 中。

2) Last-Modified

设置这个标记会通知浏览器使用 if-modified-since 头，通过 GET 请求来检查其本地缓存相关信息，如果数据不需要更新，服务器将使用 HTTP 304 状态码来响应此请求，如果需要更新则服务器返回 200 的状态码。

在浏览器第一次请求某一个 URL 时，服务器端的返回状态会是 200，内容是你请求的资源，同时有一个 Last-Modified 的属性标记 (HttpReponse Header) 此文件在服务期端最后被修改的时间，格式类似这样：

Last-Modified:Thu, 28 Feb 2019 08:01:04 GMT

客户端第二次请求此 URL 时，根据 HTTP 协议的规定，浏览器会向服务器传送 If-Modified-Since 报头 (HttpRequest Header)，询问该时间之后文件是否有被修改过：

If-Modified-Since:Thu, 28 Feb 2019 08:01:04 GMT

如果服务器端的资源没有变化，则自动返回 HTTP304 (NotChanged.) 状态码，内容为空，这样就节省了传输数据量。

当服务器端代码发生改变或者重启服务器时，则重新发出资源，返回和第一次请求时类似。从而保证不向客户端重复发出资源，也保证当服务器有变化时，客户端能够得到最新的资源。

3) Cache-Control

Cache-Control 指定请求和响应遵循的缓存机制。在请求消息或响应消息中设置 Cache-Control 并不会修改另一个消息处理过程中的缓存处理过程。

请求时的缓存指令包括 no-cache、no-store、max-age、max-stale、min-fresh、only-if-cached，响应消息中的指令包括 public、private、no-cache、no-store、no-transform、must-revalidate、proxy-revalidate、max-age。

如果允许应该被设置为“public”，使其它用户可以在中间代理和缓存服务器上存储和共享数据。Public 指示响应可被任何缓存区缓存。

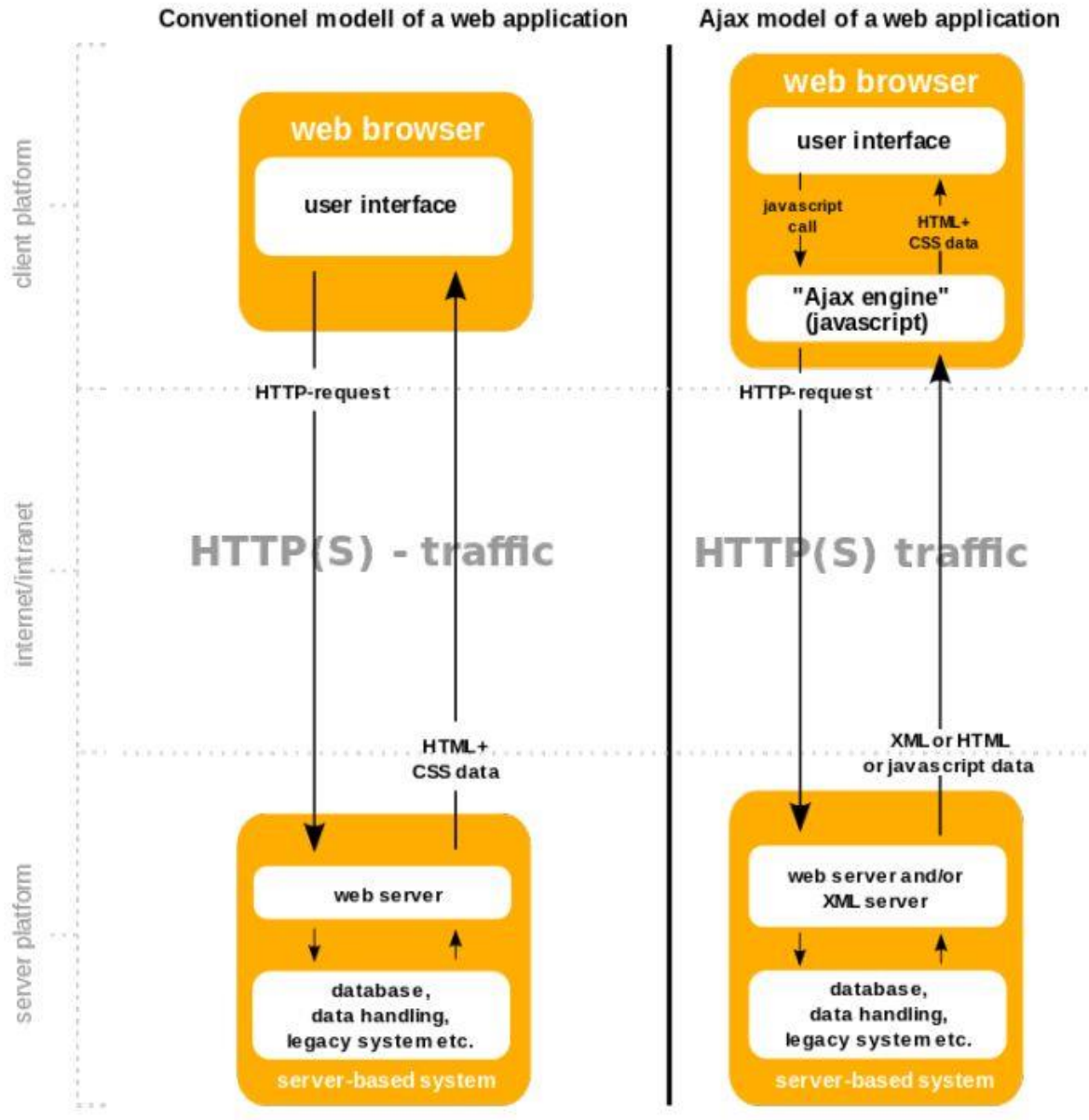
jQuery 提供了一些 API，可以很轻松的创建 AJAX 请求，通过 jQuery AJAX 方法，您能够使用 HTTP Get 和 HTTP Post 从远程服务器上请求文本、HTML、XML 或 JSON 同时您能够把这些外部数据直接载入网页的被选元素中。

jQuery 是一个 JavaScript 库，jQuery 极大地简化了 JavaScript 编程。jQuery 提供了 load()、get() 和 post() 方法，使用处理 AJAX 请求。

AJAX 缓存带来很多优点，但是 AJAX 缓存也存在不足，如果 AJAX 对一些后台数据进行更改时，虽然数据在服务器端修改了，但是浏览器缓存中的结果并没有以改变，浏览器只是简单的从缓存中读取数据并返回到客户端。当然要解决 AJAX 缓存的问题可以禁止页面缓存。

什么时候应该用 Ajax?

先看看 Ajax 和传统 Server Centri Web 架构的区别。



Ajax 本质上是一种浏览器端技术，从图中可以看出，传统 Server Centri Web 架构的最大区别是将大量业务逻辑从服务器端移到浏览器。

好处是当数据发生变更时，只需要重新渲染相关的 HTML，而不需要加载整个页面。坏处是使用大量 Ajax 的项目与其说是 B/S 架构，不如说是 C/S 架构，自然也具备 C/S 架构的特点，在浏览器环境下，某些特点会成为缺陷。

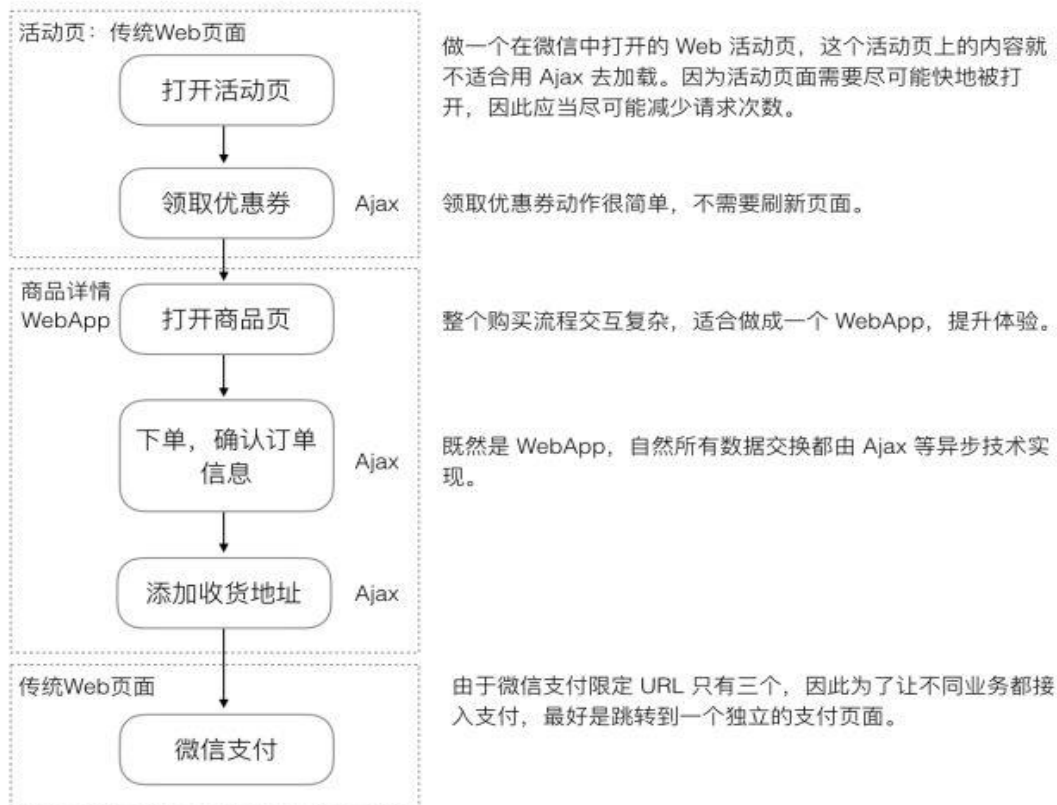
比如说 Ajax 要求业务逻辑必须先于数据加载，浏览器必须加载完相关 js 文件后才能开始加载数据，因此第一次页面 ready 的时间会晚于传统 Web 页面。不过这样的问题也有很多解决方案，例如 Application Cache 可以将文件保存在浏览器里，避免反复加载相同的脚本和资源文件。

所以结论是：**是否应该使用 Ajax 取决于业务流程和技术栈**

那怎么判定一个业务流程是否适合 Ajax 呢？

举个栗子：

假如你需要在微信里接入一个卖水果的活动，流程可以如下设计：



关于使用 Ajax 建议

- 页面上首次显示的内容尽量不要用 Ajax，显示更多（或换页）的时候再用
比如说知乎首页的第一页 Timeline 上的问题是直接从服务器读取的，并写入在当前的页面源代码里面的，而随后 “显示更多” 得到的内容才是用 Ajax 读取。
- 切换页面的时候不用要 Ajax
不要因为网页的 header 或者 side 部分没有变化而用 Ajax。
因为这是不同的页面，换句话说，是有完全不同的内容或者完全不同的页面逻辑。他们的 URL 就应该不同。
补充：使用了 HTML5 History API 例外，但是同时要考虑兼容早期版本的浏览器。
- 提交、修改、删除的部分一般用 Ajax
- 消息提醒的部分一般用 Ajax
因为很有可能会用到 Comet 长连接来保证消息的实时推送，那么除了 WebSocket 之外最好的选择只有 Ajax。
- 编码的时候模块分工明确
比如使用了 jQuery，那么所有的 Ajax 请求都最好做一层包装，然后再转移给 \$.post 或者 \$.ajax 之类的方法。