**Ajax(Asynchronous JavaScript and XML)**

**ajax** 异步的 JavaScript 和 XML。Ajax是一种用于创建快速动态网页的技术。前端通过ajax与后台服务器进行少量的数据交互实现动态的创建web应用程序。

传统的web应用程序在开发的过程中，前端所需的数据都是通过浏览器向后台服务器发送HTTP请求，后台服务器接收到请求后从数据库中检索得到数据，重新组织成为响应报文(此时响应报文的实体是一个新的html页面)，发送给客户端展示。因此每次数据交互都需要建立一次HTTP请求。有时，客户端微小的一次交互有可能就需要重载刷新一次页面。服务器在处理客户端提交的请求的过程中，客户端只能停下等待，只有在服务器处理完客户端请求，完成响应之后客户端才能完成功能展示。这种做法浪费了许多网络带宽，不仅会重复发送大量重复数据，而且使整个应用程序的响应时间依赖与服务器的响应时间，服务器荷载过大的时候，容易出现页面得不到响应的情况。

使用了Ajax技术的web应用，可以仅向服务器发送并取回必要的数据它使用SOAP或其它一些基于XML的Web Service接口，在客户端使用JavaScript处理来自服务器的响应。因为在服务器和浏览器之间交换的数据大量减少，结果我们就能看到响应更快的应用。同时很多的处理工作可以在发出请求的客户端机器上完成，所以Web服务器的处理时间也减少了。

Ajax工作原理：

Ajax其核心有JavaScript、XMLHTTPRequest、DOM对象组成，通过XmlHttpRequest对象来向服务器发异步请求，从服务器获得数据，然后用JavaScript来操作DOM而更新页面。

Ajax的工作原理相当于在用户和服务器之间加了—个中间层(AJAX引擎),使用户操作与服务器响应异步化。并不是所有的用户请求都提交给服务器,像—些数据验证和数据处理等都交给Ajax引擎自己来做, 只有确定需要从服务器读取新数据时再由Ajax引擎代为向服务器提交请求。

Ajax引擎，实际上是一个比较复杂的JavaScript应用程序，用来处理用户请求，读写服务器和更改DOM内容。JavaScript的Ajax引擎读取信息，并且互动地重写DOM，这使网页能无缝化重构，也就是在页面已经下载完毕后改变页面内容，这是我们一直在通过JavaScript和DOM在广泛使用的方法，但要使网页真正动态起来，不仅要内部的互动，还需要从外部获取数据，在以前，我们是让用户来输入数据并通过DOM来改变网页内容，但现在XMLHTTPRequest，可以让我们在不重载页面的情况下读写服务器上的数据，使用户的输入达到最少。

Ajax使WEB中的界面与应用分离（也可以说是数据与呈现分离），而在以前两者是没有清晰的界限，数据与呈现的分离，有利于分工合作、减少非技术人员对页面的修改造成的WEB应用程序错误、提高效率。也可以把以前的一些服务器负担的工作转嫁到客户端，利于客户端闲置的处理能力来处理。





4.AJAX的优缺点

(1).AJAX的优点

<1>.无刷新更新数据。

AJAX最大优点就是能在不刷新整个页面的前提下与服务器通信维护数据。这使得Web应用程序更为迅捷地响应用户交互，并避免了在网络上发送那些没有改变的信息，减少用户等待时间，带来非常好的用户体验。

<2>.异步与服务器通信。

AJAX使用异步方式与服务器通信，不需要打断用户的操作，具有更加迅速的响应能力。优化了Browser和Server之间的沟通，减少不必要的数据传输、时间及降低网络上数据流量。

<3>.前端和后端负载平衡。

AJAX可以把以前一些服务器负担的工作转嫁到客户端，利用客户端闲置的能力来处理，减轻服务器和带宽的负担，节约空间和宽带租用成本。并且减轻服务器的负担，AJAX的原则是“按需取数据”，可以最大程度的减少冗余请求和响应对服务器造成的负担，提升站点性能。

<4>.基于标准被广泛支持。

AJAX基于标准化的并被广泛支持的技术，不需要下载浏览器插件或者小程序，但需要客户允许JavaScript在浏览器上执行。随着Ajax的成熟，一些简化Ajax使用方法的程序库也相继问世。同样，也出现了另一种辅助程序设计的技术，为那些不支持JavaScript的用户提供替代功能。

<5>.界面与应用分离。

Ajax使WEB中的界面与应用分离（也可以说是数据与呈现分离），有利于分工合作、减少非技术人员对页面的修改造成的WEB应用程序错误、提高效率、也更加适用于现在的发布系统。

(2).AJAX的缺点

<1>.AJAX干掉了Back和History功能，即对浏览器机制的破坏。

在动态更新页面的情况下，用户无法回到前一个页面状态，因为浏览器仅能记忆历史记录中的静态页面。一个被完整读入的页面与一个已经被动态修改过的页面之间的差别非常微妙；用户通常会希望单击后退按钮能够取消他们的前一次操作，但是在Ajax应用程序中，这将无法实现。

后退按钮是一个标准的web站点的重要功能，但是它没法和js进行很好的合作。这是Ajax所带来的一个比较严重的问题，因为用户往往是希望能够通过后退来取消前一次操作的。那么对于这个问题有没有办法？答案是肯定的，用过Gmail的知道，Gmail下面采用的Ajax技术解决了这个问题，在Gmail下面是可以后退的，但是，它也并不能改变Ajax的机制，它只是采用的一个比较笨但是有效的办法，即用户单击后退按钮访问历史记录时，通过创建或使用一个隐藏的IFRAME来重现页面上的变更。（例如，当用户在Google Maps中单击后退时，它在一个隐藏的IFRAME中进行搜索，然后将搜索结果反映到Ajax元素上，以便将应用程序状态恢复到当时的状态。）

但是，虽然说这个问题是可以解决的，但是它所带来的开发成本是非常高的，并与Ajax框架所要求的快速开发是相背离的。这是Ajax所带来的一个非常严重的问题。

一个相关的观点认为，使用动态页面更新使得用户难于将某个特定的状态保存到收藏夹中。该问题的解决方案也已出现，大部分都使用URL片断标识符（通常被称为锚点，即URL中#后面的部分）来保持跟踪，允许用户回到指定的某个应用程序状态。（许多浏览器允许JavaScript动态更新锚点，这使得Ajax应用程序能够在更新显示内容的同时更新锚点。）这些解决方案也同时解决了许多关于不支持后退按钮的争论。

<2>.AJAX的安全问题。

AJAX技术给用户带来很好的用户体验的同时也对IT企业带来了新的安全威胁，Ajax技术就如同对企业数据建立了一个直接通道。这使得开发者在不经意间会暴露比以前更多的数据和服务器逻辑。Ajax的逻辑可以对客户端的安全扫描技术隐藏起来，允许黑客从远端服务器上建立新的攻击。还有Ajax也难以避免一些已知的安全弱点，诸如跨站点脚步攻击、SQL注入攻击和基于Credentials的安全漏洞等等。

<3>.对搜索引擎支持较弱。

对搜索引擎的支持比较弱。如果使用不当，AJAX会增大网络数据的流量，从而降低整个系统的性能。

<4>.破坏程序的异常处理机制。

至少从目前看来，像Ajax.dll，Ajaxpro.dll这些Ajax框架是会破坏程序的异常机制的。关于这个问题，曾在开发过程中遇到过，但是查了一下网上几乎没有相关的介绍。后来做了一次试验，分别采用Ajax和传统的form提交的模式来删除一条数据……给我们的调试带来了很大的困难。

<5>.违背URL和资源定位的初衷。

例如，我给你一个URL地址，如果采用了Ajax技术，也许你在该URL地址下面看到的和我在这个URL地址下看到的内容是不同的。这个和资源定位的初衷是相背离的。

<6>.AJAX不能很好支持移动设备。

一些手持设备（如手机、PDA等）现在还不能很好的支持Ajax，比如说我们在手机的浏览器上打开采用Ajax技术的网站时，它目前是不支持的。

<7>.客户端过肥，太多客户端代码造成开发上的成本。

编写复杂、容易出错 ；冗余代码比较多（层层包含js文件是AJAX的通病，再加上以往的很多服务端代码现在放到了客户端）；破坏了Web的原有标准。

5.AJAX注意点及适用和不适用场景

(1).注意点

Ajax开发时，网络延迟——即用户发出请求到服务器发出响应之间的间隔——需要慎重考虑。不给予用户明确的回应，没有恰当的预读数据，或者对XMLHttpRequest的不恰当处理，都会使用户感到延迟，这是用户不希望看到的，也是他们无法理解的。通常的解决方案是，使用一个可视化的组件来告诉用户系统正在进行后台操作并且正在读取数据和内容。

(2).Ajax适用场景

<1>.表单驱动的交互

<2>.深层次的树的导航

<3>.快速的用户与用户间的交流响应

<4>.类似投票、yes/no等无关痛痒的场景

<5>.对数据进行过滤和操纵相关数据的场景

<6>.普通的文本输入提示和自动完成的场景

(3).Ajax不适用场景

<1>.部分简单的表单

<2>.搜索

<3>.基本的导航

<4>.替换大量的文本

<5>.对呈现的操纵