参考：<http://thinking80s.iteye.com/blog/677797>

## 概念

同步：提交请求->等待服务器处理->处理完毕返回 这个期间客户端浏览器不能干任何事   
  
异步: 请求通过事件触发->服务器处理（这是浏览器仍然可以作其他事情）->处理完毕   
  
  
同步就是你叫我去吃饭，我听到了就和你去吃饭；如果没有听到，你就不停的叫，直到我告诉你听到了，才一起去吃饭。   
  
异步就是你叫我，然后自己去吃饭，我得到消息后可能立即走，也可能等到下班才去吃饭。   
  
所以，要我请你吃饭就用同步的方法，要请我吃饭就用异步的方法，这样你可以省钱。

请求方式，分为GET与POST：   
  
GET   
  
最为常见的HTTP请求，普通上网浏览页面就是GET。GET方式的参数请求直接跟在URL后，以问号开始。（JS中用 window.location.search获得）。参数可以用encodeURIComponent进行编码，使用方式:   
  
var EnParam = encodeURIComponent(param);   
  
URL只支持大约2K的长度，即2048字符数；使用GET进行AJAX请求时候会缓存导致出现的页面不是正确的，一般方法加random参数值；ajax.send(null)。

## GET 方法

请注意，查询字符串（名称/值对）是在 GET 请求的 URL 中发送的：

/test/demo\_form.asp?name1=value1&name2=value2

有关 GET 请求的其他一些注释：

* GET 请求可被缓存
* GET 请求保留在浏览器历史记录中
* GET 请求可被收藏为书签
* GET 请求不应在处理敏感数据时使用
* GET 请求有长度限制
* GET 请求只应当用于取回数据

POST   
  
向服务器提交数据用到。   
  
需要将form表单中的值先取出转换成字符串，用&符号连接，（同GET传参数一样）；提交数据量2GB ；使用ajax.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded')，处理提交的字符串;ajax.send(strings)，这个 strings表示form中需要提交的内容，例如a=1&b=2类似这样的字符串。

举个例子：普通B/S模式（同步）AJAX技术（异步）   
同步：提交请求->等待服务器处理->处理完毕返回 这个期间客户端浏览器不能干任何事   
异步: 请求通过事件触发->服务器处理（这是浏览器仍然可以作其他事情）->处理完毕

## POST 方法

请注意，查询字符串（名称/值对）是在 POST 请求的 HTTP 消息主体中发送的：

POST /test/demo\_form.asp HTTP/1.1

Host: w3schools.com

name1=value1&name2=value2

有关 POST 请求的其他一些注释：

* POST 请求不会被缓存
* POST 请求不会保留在浏览器历史记录中
* POST 不能被收藏为书签
* POST 请求对数据长度没有要求

同步执行模式:   
所谓同步执行模式，是指语句在同步执行模式下，将始终保持对程序流的控制，直至 程序结束。如查询操作，客户机上的应用程序在向服务器发出查询操作的指令后，将 一直等待服务器将查询结果返回客户机端，然后才继续进行下一步操作。   
众所周知，应用程序要从一个大表中删除所有的记录将是非常耗时的，如果应用程序 采用的是单线程(thread)同步执行方式，某次删除工作很可能耽误其他重要工作的完 成。如果应用程序等待的是远程任务，那么远程服务器失败或网络故障或一些无法预 知的情况都可能使应用程序无限期地等下去，这是同步执行最大的缺陷。   
但是同步执行模式可以简化程序编制的复杂性。程序员可以不用过多地了解比较复杂 的ODBC 2.0 API 的使用，而只需使用 ODBC 的同步执行模式或使用数据控制项和数据库对象变量来编写应用程序，可以提高开发效率，但程序运行速度比不上异步执行 模式的速度。

异步执行模式:   
所谓异步执行模式，是指语句在异步执行模式下，各语句执行结 束的顺序与语句执行开始的顺序并不一定相同。例如 查询操作，客户机上的应用程序在向服务器发出了查 询操作的指令后，将立刻执行查询语句指令的下一条 语句，而不需要等到服务器将查询结果返回客户机端。 异步执行方式使应用程序能摆 脱单个任务的牵制，提高了灵活性和应用程序的执行 效率。但异步执行模式也存在一些问题，如它增加了 编程的复杂性，特别是编写互用性(interoperable)要求较高 的程序。   
在负荷很重的客户／服务器系 统中，适宜采用异步执行模式。在这种环境下，时间 延迟频繁且漫长，相比之下异步执行的开销微不足 道。但是，如果应用运行的环境比较复杂，则必须建 立一套完整的机制，周期性地检查函数执行的状态， 以决定下一步执行方案。进行周期的检查可以有多种方法，如在 应用中设置计时器并处理WM\_TIMER信息等。   
虽然使用异步执行模式在编程 序时十分复杂，但可以实现多任务并行执行，使执行的效率大大提高。   
选择并设置执行模式 在应用程序开发中选择同步模式还是异步模式，是一个比 较复杂的层次。当查询或对数据库的修改相对简单时，同步执行模式是一种 很好的选择，它能够在几秒 或更少的时间内返回结果数据。另外，在应用程序获 得结果集前不能继续执行时，根本不必要使用异步执 行模式。在复杂查询情况下，特别是复杂的多行数据 库的UPDATE 或DELETE 操作，可能需要很长的时间才能完成， 需采用异步执行模式，让用户可以同时对程序的其他 部分进行操作。