# window.open()

这个函数中的url必须是get请求，不能是post请求

# POST提交数据时常见的数据格式

HTTP 协议是以 ASCII 码传输，建立在 TCP/IP 协议之上的应用层规范。规范把 HTTP 请求分为三个部分：状态行、请求头、消息主体。类似于下面这样：

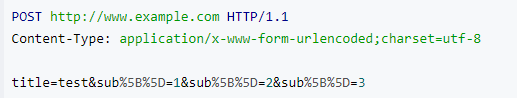


协议规定 POST 提交的数据必须放在消息主体（entity-body）中, 但协议并没有规定数据必须使用什么编码方式。实际上，开发者完全可以自己决定消息主体的格式，只要最后发送的 HTTP 请求满足上面的格式就可以。

但是，数据发送出去还要服务端解析成功才有意义。一般服务端语言如 php、python 等，以及它们的 framework，都内置了自动解析常见数据格式的功能。服务端通常是根据请求头（headers）中的 Content-Type 字段来获知请求中的消息主体是用何种方式编码，再对主体进行解析。所以说到 POST 提交数据方案，包含了 Content-Type 和消息主体编码方式两部分。

## （1）application/x-www-form-urlencoded

这应该是最常见的 POST 提交数据的方式了。浏览器的原生表单，如果不设置 enctype 属性，那么最终就会以 application/x-www-form-urlencoded方式提交数据。请求类似于下面这样（无关的请求头在本文中都省略掉了）：



首先，Content-Type 被指定为 application/x-www-form-urlencoded，其次，提交的数据按照 key1=val1&key2=val2 的方式进行编码，key 和 val 都进行了 URL 转码。大部分服务端语言都对这种方式有很好的支持。例如 PHP 中，$\_POST['title'] 可以获取到 title 的值，$\_POST['sub'] 可以得到 sub 数组。

很多时候，我们用 Ajax 提交数据时也是使用这种方式。例如 JQuery 和 QWrap 的 Ajax，Content-Type 默认值都是「application/x-www-form-urlencoded;charset=utf-8」。

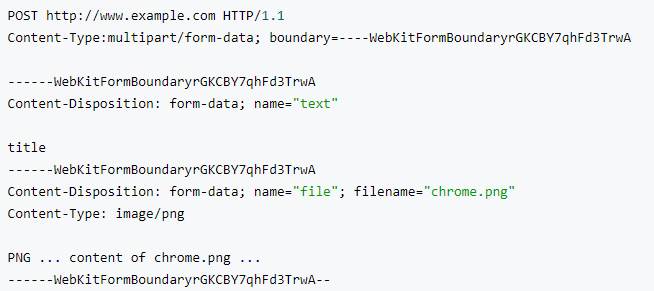
我这里用soupui调用选的xml格式Media Type为 application/xml ,这里是不对的。我改成 application/x-www-form-urlencoded这种格式后，xml报文也改成key=value形式。

requestMessage=<?xml version=...><itxEnvelope>...</itxEnvelope>

再调用就可以了，就是仿form表单提交方式完成请求的。

## （2）multipart/form-data

这又是一个常见的 POST 数据提交的方式。我们使用表单上传文件时，必须让表单的 enctype 等于 multipart/form-data。直接来看一个请求示例：



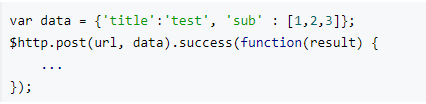
这个例子稍微复杂点。首先生成了一个 boundary 用于分割不同的字段，为了避免与正文内容重复，boundary 很长很复杂。然后 Content-Type 里指明了数据是以 multipart/form-data 来编码，本次请求的 boundary 是什么内容。消息主体里按照字段个数又分为多个结构类似的部分，每部分都是以 –boundary 开始，紧接着是内容描述信息，然后是回车，最后是字段具体内容（文本或二进制）。如果传输的是文件，还要包含文件名和文件类型信息。消息主体最后以 –boundary– 标示结束。

这种方式一般用来上传文件，各大服务端语言对它也有着良好的支持。   
上面提到的这两种 POST 数据的方式，都是浏览器原生支持的，而且现阶段标准中原生表单也只支持这两种方式（通过元素的 enctype 属性指定，默认为 application/x-www-form-urlencoded。其实 enctype 还支持 text/plain，不过用得非常少）。 随着越来越多的 Web 站点，尤其是 WebApp，全部使用 Ajax 进行数据交互之后，我们完全可以定义新的数据提交方式，给开发带来更多便利。

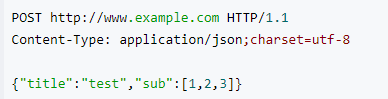
## （3）application/json

application/json 这个 Content-Type 作为响应头大家肯定不陌生。实际上，现在越来越多的人把它作为请求头，用来告诉服务端消息主体是序列化后的 JSON 字符串。由于 JSON 规范的流行，除了低版本 IE 之外的各大浏览器都原生支持 JSON.stringify，服务端语言也都有处理 JSON 的函数，使用 JSON 不会遇上什么麻烦。

JSON 格式支持比键值对复杂得多的结构化数据，这一点也很有用。Google 的 AngularJS 中的 Ajax 功能，默认就是提交 JSON 字符串。例如下面这段代码：



最终发送的请求是：



这种方案可以方便的提交复杂的结构化数据，特别适合 RESTful 的接口。各大抓包工具如 Chrome 自带的开发者工具、Firebug、Fiddler，都会以树形结构展示 JSON 数据，非常友好。但也有些服务端语言还没有支持这种方式。

## （4）text/xml

XML-RPC（XML Remote Procedure Call），它是一种使用 HTTP 作为传输协议，XML 作为编码方式的远程调用规范。典型的 XML-RPC 请求是这样的：



XML-RPC 协议简单、功能够用，各种语言的实现都有。它的使用也很广泛，如 WordPress 的 XML-RPC Api，搜索引擎的 ping 服务等等。JavaScript 中，也有现成的库支持以这种方式进行数据交互，能很好的支持已有的 XML-RPC 服务。不过我个人觉得 XML 结构还是过于臃肿，一般场景用 JSON 会更灵活方便。