POST提交数据方式

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。

application/x-www-form-urlencoded.

HTTP的消息体以查询字符串的方式发送(名/值对，用&分割，名字和值用＝连接).其中非字母数字的字符用'%HH'替换。(HH是十六进制)

multipart/form-data

- 每一部份用分开的MIME消息(文件的类型)表示。
- 并且，每一部份被特定的字符边界分开(boundary=--------------------------735323031399963166993862150).这个字符边界精心挑选一遍不会和内容重复。
- 每一部份有自己的MIME头(content-type).通过这个值可以选择更有效率的方式存储值，以节省带宽。(base64或者raw binary)

```
  POST / HTTP/1.1
  [[ Less interesting headers ... ]]
  Content-Type: multipart/form-data; boundary=------------------------
---735323031399963166993862150
  Content-Length: 834

  -----------------------------735323031399963166993862150
  Content-Disposition: form-data; name="text1"

  text default
  -----------------------------735323031399963166993862150
  Content-Disposition: form-data; name="text2"

  aωb
  -----------------------------735323031399963166993862150
  Content-Disposition: form-data; name="file1"; filename="a.txt"
  Content-Type: text/plain

  Content of a.txt.

  -----------------------------735323031399963166993862150
  Content-Disposition: form-data; name="file2"; filename="a.html"
  Content-Type: text/html

  <!DOCTYPE html><title>Content of a.html.</title>

  -----------------------------735323031399963166993862150
  Content-Disposition: form-data; name="file3"; filename="binary"
  Content-Type: application/octet-stream

  aωb
  -----------------------------735323031399963166993862150--
```

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

text/xml

用的比较少了

application/x-www-form-urlencoded VS multipart/form-data两者区别

- 文件内容的上传必须用multipart/form-data
- 简单的字符串key-value对(大部分form－data的情况)，用application/x-www-form-urlencoded更简单有效。用multipart就会添加一些不必要的额外字符串了。

  stackovflow 对form编码的讨论 四种提交的方式