基于HTTP的文件上传的规范

by calidion

基于HTTP协议的文件上传的规范

HTTP文件上传所基于的通用协议是：rfc1867
https://tools.ietf.org/html/rfc1867

这个协议里主要有两个方面的修改

1. 对HTML元素的增强(具体就是INPUT元素)
2. 对表单媒体类型的支持

HTML元素的增强

主要体现在两点：

1. 添加FILE选项到INPUT元素

```
<input type="file"/>
```

3. 添加了ACCEPT属性到INPUT元素

```
<input type="file" accept="image/gif,image/jpg"/>
```

定义了新的媒体类型

1. 表单提交的默认的媒体类型是：

`application/x-www-form-urlencoded`

```
<form enctype="application/x-www-form-urlencoded">
```

2. 定义的一个新的MIME媒体类型：

`multipart/form-data`

```
<form enctype="multipart/form-data">
```

URLEncoded

URL是通用资源定位器（Uniform Resource Locator），它是由可见的ASCII编码的字符组成的字符串。

在表达中文，不可见字符，二进制等情况时需要做相应的编码转换，这种转换规则就是URL的Encode的规则。

比如，对于URL上的目录`/file/`，浏览器是可以直接识别的。
但是对于目录`/中国/`，浏览器是无法直接识别的。
所以经过编码后得到对应的URL是：
`/%E4%B8%AD%E5%9B%BD/`。这样浏览器就可以识别了。

> 由于`中国`可能有GBK，UTF-8等编码格式，不同的平台转换的结果是不同的，这个时候识别需要服务器的支持与规范。

表单提交

那么在表单提交时，我们采用了name=data的格式，表区分不同的表单名称对应的值，并使用`&`符号进行分割.

例如:

```
a=10&b=10&c=aaa
```

`location=中国`可以表示如下：

```
location=%E4%B8%AD%E5%9B%BD
```

表单提交文件的问题

考虑这种场景：

1. 文件内容

```
a=b&c=d&e=g
```

2. 如果上传文件采用表单提交的格式。如：

```
file=[文件内容]
```

3. 考察文件的url编码：

```
encodeURI("a=b&c=d&e=g")
```

的结果还是：

```
a=b&c=d&e=g
```

4. 所以使用表单提交的话就会出现这样的提交情况：

```
file=a=b&c=d&e=g
```

5. 提交的内容发生了错乱

这就是无法使用表单方法提交的原因。

boundary

Multipart的协议提供了一个boundary分割符，用于表示提交内容中的唯一标识。

目前的浏览器都是通过提供一串hash的值来表示唯一性的。

因为通常哈希算法有比较好的唯一标识能力。

Multipart

```
Content-type: multipart/form-data, boundary=AaB03x

--AaB03x
content-disposition: form-data; name="field1"

Joe Blow
--AaB03x
content-disposition: form-data; name="pics"; filename="file
Content-Type: text/plain

... contents of file1.txt ...
--AaB03x--
```