

构建文件上传服务器

★ by calidion

处理POST请求

处理最基本的POST请求

```
if (req.method === "POST") {  
  var data = [];  
  var length = 0;  
  req.on('data', function (chunk) {  
    length += chunk.length;  
    data.push(chunk);  
  });  
  req.on('end', function () {  
    data = Buffer.concat(data, length);  
    // process(data)  
  });  
}
```

测试POST处理：

```
curl -X POST -d "a=1010&b=100&c=中文" http://127.0.0.1:8080
```

处理文件的提交

1. 获取multi-part属性

处理文件时需要确定提交的内容属性是

`multipart/form-data`

```
var contentType = req.headers['content-type'];  
  
if (contentType.indexOf('multipart/form-data;') !== -1) {  
  // 有上传文件内容  
}
```

获取分割符号(boundary)

分割符是放在***content-type***里面的。

所以我们获取 `boundary` 时需要按下面的步骤来实现。

1. 取出content-type的内容，并由"; "(注意;后有空格)进行切分。

```
var splitors = contentType.split("; ");
```

获取分割符号(boundary)

2. 遍历分割的字符数组，找出包含 `boundary=` 的一个字符串，并取出对应的值。

```
for(var i = 0; i < splitors.length; i++) {  
    var temp = splitors[i];  
    if (!temp) {  
        continue;  
    }  
    if (temp.indexOf("boundary=") !== -1) {  
        splitor = temp.split("=")[1];  
        break;  
    }  
}
```

分割分段内容

由于上传的内容是通过 `--` + `boundary` 实现的，所以我们可以这样切割分段内容。其中 `data` 是所有的 POST 数据。

```
// 注意前面多两个--  
var parts = String(data).split("--" + splitor);
```

提取分段内容

```
var parts = String(data).split("--" + splitor);
for(var i = 0; i < parts.length; i++) {
    var contents = parts[i].split("\r\n\r\n");
    for (var j = 0; j < contents.length; j++) {
        // 消除分段前后的换行信息
        contents[j] = contents[j].replace(/\r\n$/, '');
        contents[j] = contents[j].replace(/^\\r\\n/, '');
    }
    parts[i] = contents;
}
```


分解分段内容

提取分段头

分段头的内容形式：

1. 普通表单字段

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

2. 文件字段

```
Content-disposition: attachment; filename="file1.txt"
```

提取分段头

提取代码：

```
var header = contents[0];
var headers = header.split("\r\n");
for (var i = 0; i < headers.length; i++) {
    var subHead = headers[i];
    var subHeaders = subHead.split(": ");
    if (subHeaders[0] === 'Content-Disposition') {
        if (subHeaders[1].indexOf('filename') !== -1) {
            extract(subHeaders, 'filename', body);
        } else {
            extract(subHeaders, 'name', body);
        }
    }
}
```

分解分段内容

2. 提取分段体

```
var filenames = subHeaders[1].split("; ");
for(var j = 0; j < filenames.length; j++) {
    if (filenames[j].indexOf(name) !== -1) {
        var filename = filenames[j].split("=");
        req.files[filename[1]] = body;
        break;
    }
}
```