

## 1.文件系统 fs

```
fs.readFile(filename, [encoding], [callback(err, data)])
//是最简单的读取文件的函数。第二个参数可选，表示文件的字符编码.callback 是回调函数，用于接收文件的内容。
fs.readFileSync(filename, [encoding])
//是fs.readFile同步的版本。而读取到的文件内容会以函数返回值的形式返回
fs.open(path, flags, [mode], [callback(err, fd)])
//path 为文件的路径, flags 可以是以下值`r`,`r+`,`w`,`w+`,`a`,`a+`
```

## 2.HTTP 服务器与客户端 2.1获取 GET 请求内容

由于GET请求直接被嵌入在路径中，URL是完整的请求路径，包括了后面的部分，因此你可以手动解析后面的内容作为GET请求的参数。Node.js 的`url`模块中的`parse`函数提供了这个功能。

### 2.2获取 POST 请求内容

POST 请求的内容全部都在请求体中。http.ServerRequest并没有一个属性内容为请求体，原因是等待请求体传输可能是一件耗时的工作，譬如上传文件。

2.3HTTP 客户端 http 模块提供了两个函数 http.request 和 http.get，功能是作为客户端向 HTTP服务器发起请求。

#### 2.3.1.http.request(options, callback)

发起HTTP请求。接受两个参数，option是一个类似关联数组的对象，表示请求的参数,callback是请求的回调函数。option常用的参数：

```
host(请求网站的域名或 IP 地址),`port`(端口, 默认 80),
method(请求方法, 默认是 GET。),`path`(请求的相对于根的路径, 默认是"/")
http.request 返回一个 实例,不要忘了通过 req.end() 结束请求, 否则服务器将不会收到信息。
```

#### 2.3.2.http.get(options, callback)

它是 http.request 的简化版，区别在于http.get自动将请求方法设为了GET请求，同时不需要手动调用 req.end()。

#### 2.3.3.http.ClientRequest

http.ClientRequest是由http.request或http.get返回产生的对象，表示一个已经产生而且正在进行中的HTTP请求。它提供一个 response 事件，即 http.request或 http.get {}；像 http.ServerResponse 一样也提供了 write 函数，用于向服务器发送请求体，通常用于 POST、PUT等操作。所有写结束以后必须调用end函数以通知服务器，否则请求无效。http.ClientRequest还提供了以下函数。

2.3.3.1 request.abort(): 终止正在发送的请求。

2.3.3.2 request.setTimeout(timeout, [callback]): 设置请求超时时间,timeout为毫秒数。

#### 2.3.4.http.ClientResponse

http.ServerRequest相似，提供了三个事件data、end和close，分别在数据到达、传输结束和连接结束时触发，其中 data 事件传递一个参数chunk，表示接收到的数据。

2.3.4.1response.setEncoding([encoding]): 设置默认的编码，当 data 事件被触发时，数据将会以 encoding 编码。默认值是 null，即不编码，以 Buffer 的形式存储。常用编码为 utf8。

2.3.4.2response.pause(): 暂停接收数据和发送事件，方便实现下载功能。

2.3.4.3response.resume(): 从暂停的状态中恢复。