# 网络

https://www.electronjs.org/docs/api/net

进程： 主进程
`net` 模块是一个发送 HTTP(S) 请求的客户端API。 它类似于Node.js的 **HTTP** 和 **HTTPS** 模块 ， 但它使用的是Chromium原生网络库来替代Node.js的实现，提供更好的网络代理支持。 It also supports checking network status.
下面是一个非详尽的列表, 用于说明为什么使用 `net` 模块而不是原生Node. js 模块:

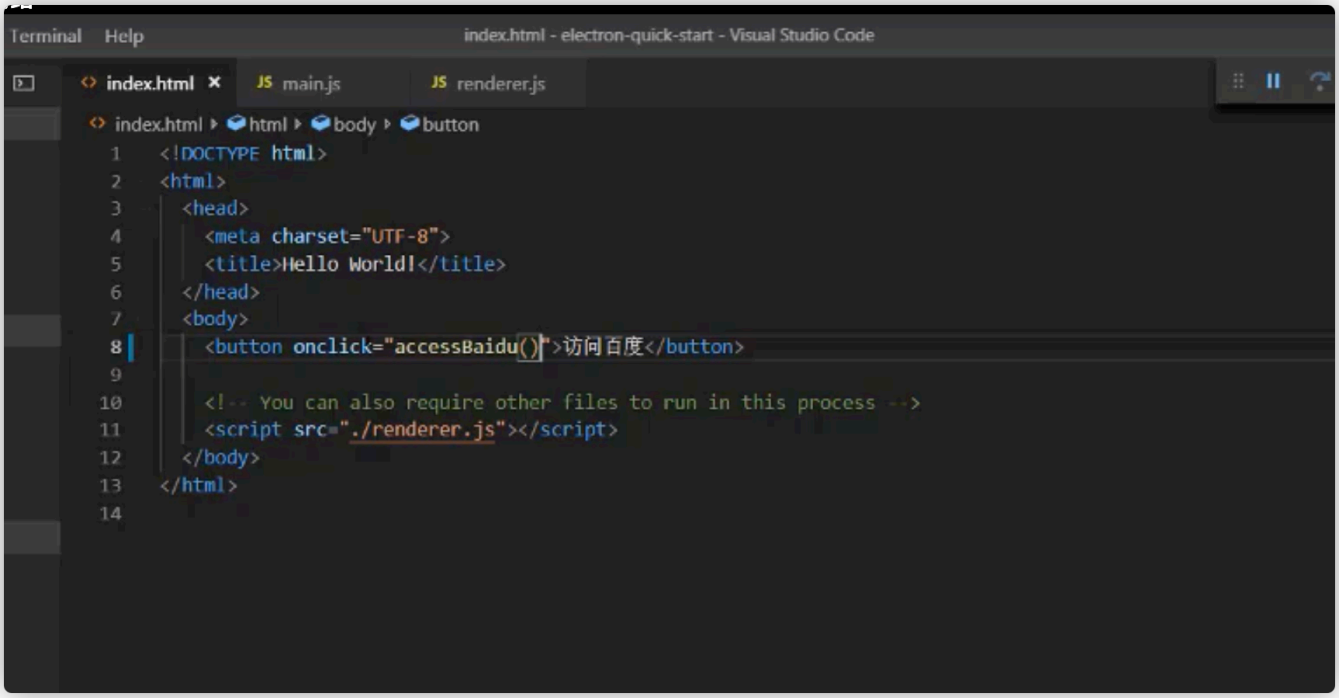
- 系统代理配置的自动管理, 支持 wpad 协议和代理 pac 配置文件。
- HTTPS 请求的自动隧道。
- 支持使用basic、digest、NTLM、Kerberos 或协商身份验证方案对代理进行身份验证。
- 支持传输监控代理: 类似于Fiddler代理，用于访问控制和监视。

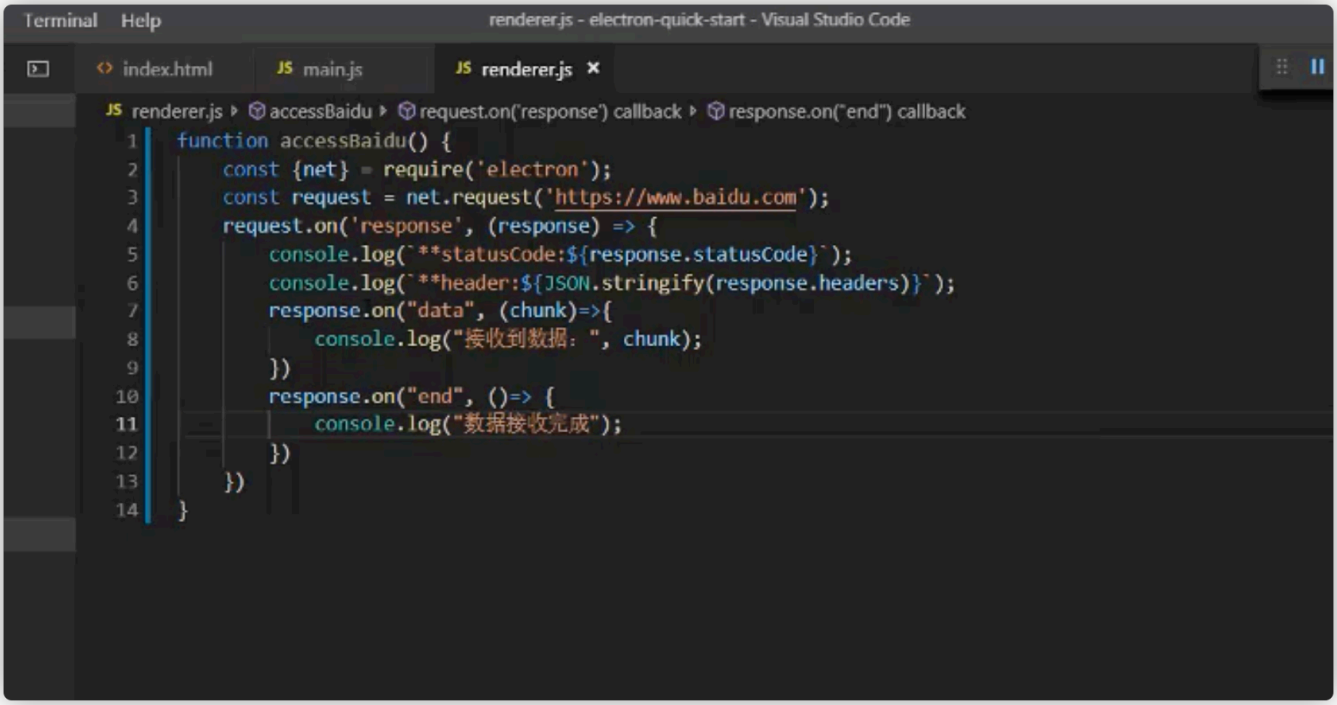
The API components (including classes, methods, properties and event names) are similar to those used in Node.js.
Example usage:

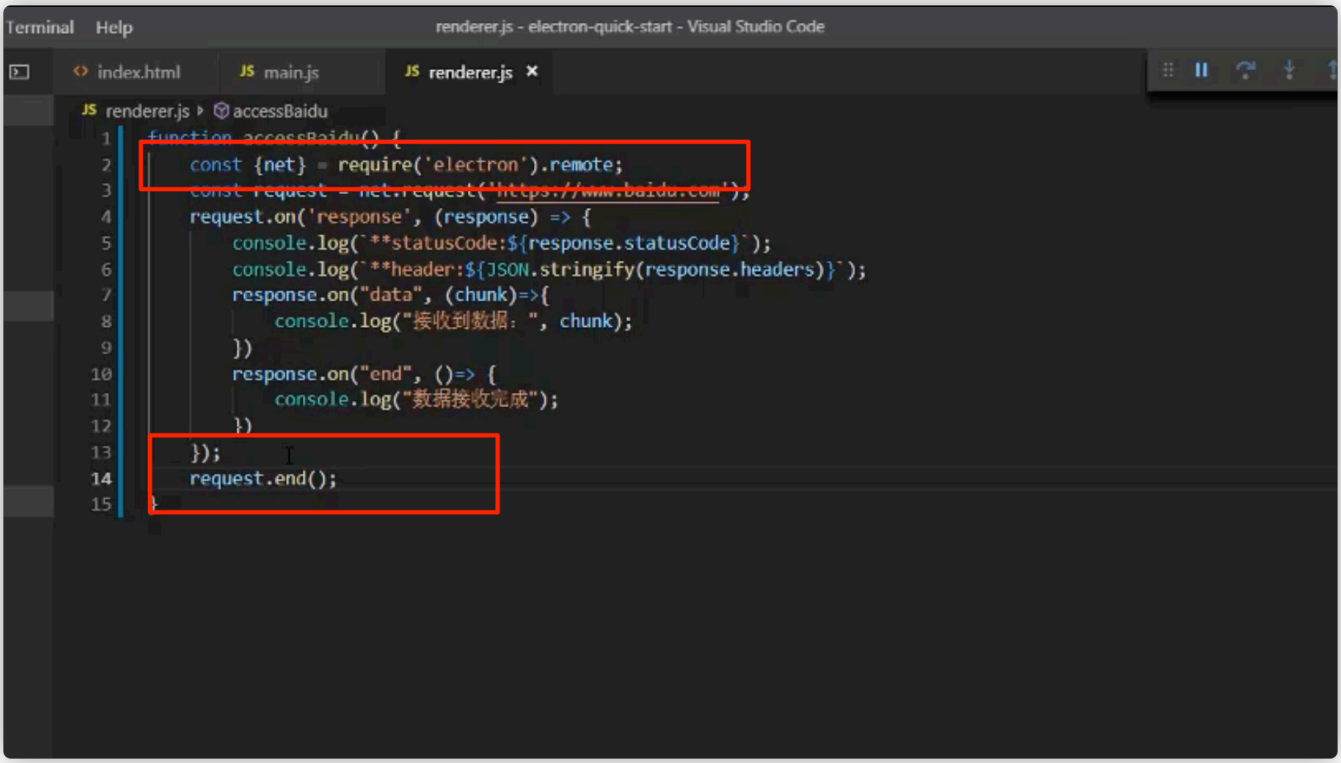
```
const { app } = require('electron')
app.whenReady().then(() => {
  const { net } = require('electron')
  const request = net.request('https://github.com')
  request.on('response', (response) => {
    console.log(`STATUS: ${response.statusCode}`)
    console.log(`HEADERS: ${JSON.stringify(response.headers)}`)
    response.on('data', (chunk) => {
      console.log(`BODY: ${chunk}`)
    })
    response.on('end', () => {
      console.log('No more data in response.')
    })
  })
  request.end()
})

Copy
```

The `net` API can be used only after the application emits the `ready` event. Trying to use the module before the `ready` event will throw an error.


Terminal Help
index.html - electron-quick-start - Visual Studio Code
index.html
main.js
renderer.js
index.html ▸ html ▸ body ▸ button
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Hello World!</title>
</head>
<body>
<button onclick="accessBaidu()">访问百度</button>
<!-- You can also require other files to run in this process -->
<script src="./renderer.js"></script>
</body>
</html>


Terminal Help
renderer.js - electron-quick-start - Visual Studio Code
index.html
main.js
renderer.js
renderer.js ▸ accessBaidu ▸ request.on('response') callback ▸ response.on("end") callback
function accessBaidu() {
const {net} = require('electron');
const request = net.request('https://www.baidu.com');
request.on('response', (response) => {
console.log(`**statusCode:${response.statusCode}`);
console.log(`**header:${JSON.stringify(response.headers)}`);
response.on("data", (chunk)=>{
console.log("接收到数据：", chunk);
})
response.on("end", ()=> {
console.log("数据接收完成");
})
})
}


Terminal Help
renderer.js - electron-quick-start - Visual Studio Code
index.html
main.js
renderer.js
renderer.js accessBaidu
function accessBaidu() {
const {net} = require('electron').remote;
const request = net.request('https://www.baidu.com');
request.on('response', (response) => {
console.log(`**statusCode:${response.statusCode}`);
console.log(`**header:${JSON.stringify(response.headers)}`);
response.on("data", (chunk)=>{
console.log("接收到数据：", chunk);
})
response.on("end", ()=> {
console.log("数据接收完成");
})
});
request.end();
}