



链滴

node.js 初学笔记

作者: [CM](#)

原文链接: <https://ld246.com/article/1486375452505>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

学习网址: [node.js菜鸟教程](#)

引入模块等不作记录。从Buffer章节开始记录笔记。

Buffer 支持的编码格式: "ascii", "utf8", "utf16le", "ucs2", "base64" 和 "hex"。(utf-8 是默认的编码方式)

写入缓冲区的方法: write (string,offset,length,encoding);

● 参数解释:

string:写入的字符串值

offset:写入的位置, 默认为0, 可以不填

length:写入长度, 默认为str.length,可以不填

encoding: 编码格式, 默认为utf-8可以不填

从缓冲区读取数据方法: buf.toString(encoding,start,end);

参数均可不填, 有默认值, 与write保持一致

转换成JSON

buf.toJSON();

Node.js Express 框架

Express 框架核心特性:

- 可以设置中间件来响应 HTTP 请求。
- 定义了路由表用于执行不同的 HTTP 请求动作。
- 可以通过向模板传递参数来动态渲染 HTML 页面。

请求和响应:

```
//get请求
app.get('/', function (req, res) {
  // --
})
//post请求
app.post('/', function (req, res) {
  // --
})
```

Request 对象

常见属性:

req.app: 当callback为外部文件时, 用req.app访问express的实例

req.baseUrl: 获取路由当前安装的URL路径

req.body / req.cookies: 获得「请求主体」/ Cookies

req.fresh / req.stale: 判断请求是否还「新鲜」

req.hostname / req.ip: 获取主机名和IP地址

req.originalUrl: 获取原始请求URL

req.params: 获取路由的parameters

req.path: 获取请求路径

req.protocol: 获取协议类型
req.query: 获取URL的查询参数串
req.route: 获取当前匹配的路由
req.subdomains: 获取子域名
req.accepts () : 检查请求的Accept头的请求类型
req.acceptsCharsets / req.acceptsEncodings / req.acceptsLanguages
req.get () : 获取指定的HTTP请求头
req.is () : 判断请求头Content-Type的MIME类型

Response 对象

res.app: 同req.app一样
res.append () : 追加指定HTTP头
res.set () 在res.append () 后将重置之前设置的头
res.cookie (name, value [, option]) : 设置Cookie
option: domain / expires / httpOnly / maxAge / path / secure / signed
res.clearCookie () : 清除Cookie
res.download () : 传送指定路径的文件
res.get () : 返回指定的HTTP头
res.json () : 传送JSON响应
res.jsonp () : 传送JSONP响应
res.location () : 只设置响应的Location HTTP头, 不设置状态码或者close response
res.redirect () : 设置响应的Location HTTP头, 并且设置状态码302
res.send () : 传送HTTP响应
res.sendFile (path [, options] [, fn]) : 传送指定路径的文件 -会自动根据文件extension设定Content-Type
res.set () : 设置HTTP头, 传入object可以一次设置多个头
res.status () : 设置HTTP状态码
res.type () : 设置Content-Type的MIME类型