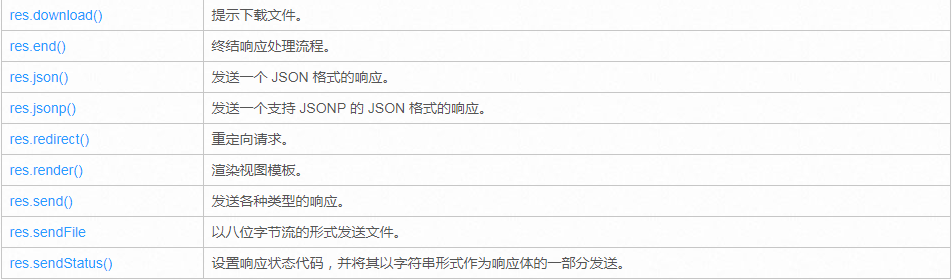
EXPRESS

1、路由：响应对象（res）的方法向客户端返回响应，终结请求响应的循环。



## app.route()

可使用 app.route() 创建路由路径的链式路由句柄。由于路径在一个地方指定，这样做有助于创建模块化的路由，而且减少了代码冗余和拼写错误



1. express.Router

可使用 express.Router 类创建模块化、可挂载的路由句柄。Router 实例是一个完整的中间件和路由系统，因此常称其为一个 “mini-app”。

下面的实例程序创建了一个路由模块，并加载了一个中间件，定义了一些路由，并且将它们挂载至应用的路径上。

Express 中间件

Express 是一个自身功能极简，完全是由路由和中间件构成一个的 web 开发框架：从本质上来说，一个 Express 应用就是在调用各种中间件。

中间件（Middleware） 是一个函数，它可以访问请求对象（[request object](http://www.expressjs.com.cn/4x/api.html" \l "req) (req)）, 响应对象（[response object](http://www.expressjs.com.cn/4x/api.html" \l "res) (res)）, 和 web 应用中处于请求-响应循环流程中的中间件，一般被命名为 next 的变量。

## 1、应用级中间件

应用级中间件绑定到 [app 对象](http://www.expressjs.com.cn/4x/api.html" \l "app) 使用 app.use() 和 app.METHOD()， 其中， METHOD 是需要处理的 HTTP 请求的方法，例如 GET, PUT, POST 等等，全部小写。

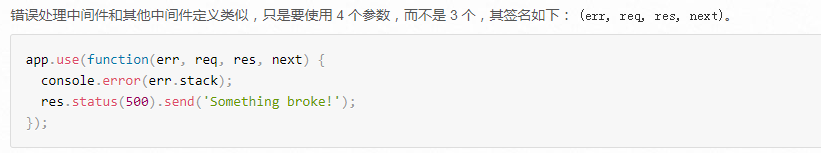
## 路由级中间件

## 路由级中间件和应用级中间件一样，只是它绑定的对象为 express.Router()



## 错误处理中间件

错误处理中间件有 **4** 个参数，定义错误处理中间件时必须使用这 4 个参数。即使不需要 next 对象，也必须在签名中声明它，否则中间件会被识别为一个常规中间件，不能处理错误



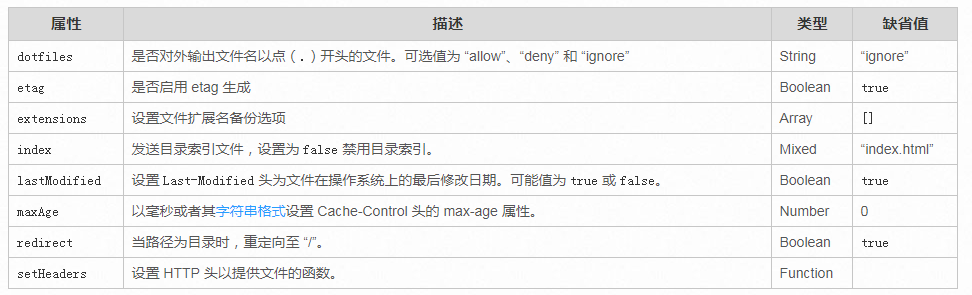
## 内置中间件

#### express.static(root, [options])

express.static 是 Express 唯一内置的中间件。它基于 [serve-static](https://github.com/expressjs/serve-static)，负责在 Express 应用中提托管静态资源。

参数 root 指提供静态资源的根目录。

可选的 options 参数拥有如下属性。



第三方中间件

Cookie-parse



**模板引擎（jade）**