var express = require('express'); //获取express模块

var app = express(); //创建express应用

app.engine(ext, callback) 注册模板引擎的 callback 用来处理ext扩展名的文件

app.render(view, [options], callback) 渲染 view, callback 用来处理返回的渲染后的字符串

app.set("views", \_\_dirname + "/views");

app.use([path], function) //path为匹配路径，回调函数有三个参数req,res,next;next()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*重点\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

express.static(root,[options]) //处理静态资源的中间价

//root为静态资源所在的根目录

例如：express.static(’public’)

// 可接收路径:：http://location:8888/css/icon.css（如果在public下面有指定文件的话）

var app = express() //表示express应用程序

setting：可以改变express行为的设置

1. env // 运行时环境，默认为 process.env.NODE\_ENV 或者 “development”
2. trust proxy // 激活反向代理，默认是未激活状态
3. jsonp callback name //修改 ?callback= 的默认 callback 的名字
4. json replacer //JSON 替换时的回调, 默认为 null
5. json spaces //JSON 响应被格式化时的空格数量，开发环境下是2，生产环境是0
6. case sensitive routing //路由的大小写敏感, 默认是关闭状态，”/Foo” 和 “/foo” 被认为是一样的
7. strict routing //路由的严格格式, 默认情况下 “/foo” 和 “/foo/” 被路由认为是一样的
8. view cache //E模板缓存，在生产环境中是默认开启的
9. view engine //默认的模板引擎
10. views //模板的目录, 默认是 “process.cwd() + ‘/views’”

app.engine(ext, callback) //注册模板引擎的 callback 用来处理 ext 扩展名的文件。

例如：app.engine('jade', require('jade').\_\_express);

路由

var router=express.routes();