```
node.js Web应用框架, 提供了一系列
                         强大特性帮助你创建各种 Web 应
              简介
                         用,和丰富的 HTTP 工具。
                         使用 Express 可以快速地搭建一个
                         完整功能的网站
                              sudo npm install express
              安装并运行
                              var express = require("express");
                              var app = express();//初始化
                                       get方法 — 根据请求路径来处理客户端发出的
                                       GET请求
                                       app.get(path,cb);
                                         path:为请求的路径
                                         cb:第二个参数为处理函数的回调,有两个参数
                                       request和response,代表请求信息和响应信息
                                                         <!DOCTYPE html>
                                                         <html lang="en">
                                                         <head>
                                                           <meta charset="UTF-8">
                                                           <title>Title</title>
                                                         </head>
                                                         <body>
                                       前端页面代码
                                                         <form action="/list" method="post">
                                                          用户名:<input type="text" name="user"><br>
                                                          密码: <input type="password"><br>
                                                           <input type="submit" value="发送">
                                                         </form>
                                                         </body>
                                                         </html>
                                                   //处理get方式发送的请求
                                                   app.get("/", function (req,res) {
                                                     //返回一个静态文件
                                                     res.sendFile(__dirname+"/post.html");
                                                   app.get("/list", function (req,res) {
                                                     //返回一个字符串
                                                     res.send("get"+req.url);
                                                   });
                                       服务端
                                                   app.post("/list", function (req,res) {
                                                     res.send("post"+req.url);
                            路由
                                                   //all就是处理不管是任何方式发送的请求,*为任意路径
                                                   app.all("*", function (req,res) {
                                                     res.send("Welcome to Express");
                                                   });
                                                   //设置端口号
                                                   app.listen(8080);
                                                          4.返回参数
                                                            res.send();返回数据,默认会转为字符串,编码为utf8
                                       获取返回的参数
                                                            res.sendFile();返回文件
                                                            res.sendStatus();返回状态码
                                                          获取请求参数
                                                           * req.host:返回请求头里取的主机名(不含端口号)
                                                           * req.path:返回请求的url的路径名
                                       获取请求的参数
                                                          * req.query:是一个可获取客户端get请求路径参数的对象属性,包含着被解析过
                                                          的请求参数对象,默认为{}
                                                            req.params:获取路由的parameters
                                                  //http://localhost:8080/query?user=tangcaiye
                                                  app.get("/query", function (req,res) {
                                                    res.send(req.query);
                                                  //http://localhost:8080/article/2/tangcaiye
                                                  app.get("/article/:id/:name", function (req,res) {
                                                    res.send(req.params);
                                                  });
                                       例子
                                                  app.all('/*', function(req,res){
                                                    console.log(req.path);
                                                    console.log(req.host);
                                                    res.send("没找到页面");
                                                  });
                                                  app.listen(8080);
                                                   中间件就是处理HTTP请求的函数,用来完成各种特定的任务,比如检测用户是否登录,分析数
                                                   据,以及其他在需要最终将数据发送给用户之前完成的任务
                                                   * 特点:
                                                   * 1.每个中间件都可以控制流程是否继续执行
                                         概念
                                                   * 2.每个中间件的req res都是用的同一个对象
                                                   * 3.如果出错了,转交错误处理中间件进行处理
                                                   * 4.最大的特点是,一个中间件处理完,可以把相应数据再传递给下一个中间件
express
                                                    * 5.如果调用回调函数的next参数表示将请求数据传递给下一个中间件
                                                     //中央发了100快钱
                                                     app.use(function (req,res,next) {
                                                       req.money = 100;
              核心功能
                                                       next();
                                                     });
                                                     //市政府
                                                     app.use(function (req,res,next) {
                                                       req.money -= 20;
                                                       //如果next传了一个不为null的参数,说明有错,会中止其他中间件并执行错误处理中间件
                            中间件
                                                       next("钱丢了");
                                                     });
                                                     //村
                                                     app.use(function (req,res,next) {
                                         例子:
                                                       req.money -= 50;
                                                       next();
                                                     });
                                                     //错误处理中间件
                                                     app.use(function (err,req,res,next) {
                                                       console.error(err);
                                                       res.end(err);
                                                     });
                                                     app.all('/*', function(req,res){
                                                       res.send(""+req.money);
                                                     });
                                                     app.listen(8080);
                                          模板引擎-首先需要先安装才能用
                                          * nam install ejs
                                          * 1.指定渲染模板引擎
                                              app.set("view engine","ejs");
                                          * 2.设置放模板文件的目录
                                              add.set("views",path.join(__dirname,"/"));
                                          * 3.rander函数,对网页模板进行渲染,在渲染模板时locals可谓其模板传入变量
                                          值,在模板中就可以条用所传变量了
                                              res.rander(view,[local],callback);
                                                         <!DOCTYPE html>
                                                         <html lang="en">
                                                         <head>
                                                          <meta charset="UTF-8">
                                                          <title>Title</title>
                                                         </head>
                            模板引擎
                                          前端代码
                                                         <body>
                                                        姓名:<%=name%><br>
                                                        年龄:<%=age%>
                                                         哪谁的年龄也是:<%=age%>岁
                                                         </body>
                                                         </html>
                                                            app.set("view engine","ejs");
                                                            app.set("views",__dirname);//放在当前目录下
                                                            app.get("/", function (req,res) {
                                                              //默认模板文件后缀格式是pis
                                          服务器端代码
                                                              res.render("muban",{
                                                                name:"tangcaiye",
                                                                age:18
                                                              });
                                         安装cookie-parser模块
                                        npm install cookie-parser
                                                   var express = require("express");
                                                   var cookieParser = require("cookie-parser");
                                                   var app = express();
                                                   app.use(cookieParser());
                                                   app.get("/", function (req,res) {
                                                     //如果请求中cookie存在visited,则输出cookie
                                                     //否则,设置cookie字段visited,并设置过期时间
                                        代码
                                                     if (req.cookies.visited){
                            cookie
                                                       res.send("已访问过");
                                                     }else {
                                                       res.cookie("visited","1",{maxAge: 10*60*1000});
                                                       res.send("第一次访问");
                                                   });
                                                   app.listen(8080);
                                                       expires cookie的过期时间,GMT格式。如果没有指定或者设置为
                                                       0,则产生新的cookie。
                                         参数说明
                                                       maxAge:是设置过去时间的方便选项,其为过期时间到当前时间的
                                                       毫秒值。
```

https://www.zybuluo.com/XiangZhou/note/208532

api网址:

Express 是一个简洁而灵活的