# 笔记

Express本身比较简单，借助第三方模块以及各种中间件实现强大的功能；

**分成五层**：

实体层：如果你想上网 网卡（硬件，无线和有线） 光纤：双绞线

链接层:局域网传输 MACs

网络层：广域网传输 IP

传输层：

应用层：HTTP

第三方模块并不是express专用的；

常用的第三方模块：

body-parser:接收post数据；

mysql:数据库操作模块；

ejs：模板引擎，可以把数据和模板文件(html文件)结合起来；

express.static:静态资源托管；

express.Router:实现子路由；

项目：

项目下面只有一个js文件作为入口文件；

其他相关资源应该放到对应的文件夹下；

路由：

Get：轻便，快速，大小有限制( 浏览器限制)，通过地址传值 ? 后面跟值 ；

安全性较低；

POST请求：显示的指定是post请求的时候才是post请求

表单的method指定为post；

Ajax的method是post的时候；

安全性较高；

All：满足get和post；

Ejs模板引擎四大技能：

输出数据：<%= %>

输出html数据：<%- %>

执行Js代码：<% 写所有的原生的ECMAScript代码 %>

公用文件引用：<% include 文件路径 %>

## 文件上传：

1. 需要一个表单，method必须是POST，enctype定义成 “multipart/form-data”
2. 需要一个input，type是file，有个属性叫accept可以定义你选择的文件类型；
3. 服务器端需要接收文件：multer，路由级的中间件；

上传的文件应该有后缀，文件应该分目录存储；

保证文件名的唯一性：防止被覆盖；

时间 + 随机数 + 文件后缀

## 富文本编辑器：

类似我们的ofice word；

Textarea 富文本编辑器

Txt office word

uploads\\2018\\09\\1537168718255\_17745738.jpg

uploads 18\09阿\_17745738.jpg

uploads/2018/09/1537168718255\_17745738.jpg

## Cookie和Session

**HTTP协议**：无状态

1,建立连接

2,客户端发起请求；

3,服务器响应；

4,断开连接：客户端和服务器端没有任何关系了，所以说http是无状态的；

下单存在的问题：

登录完成以后—》断开连接—》没状态-》服务器端不知道我是谁了；

当我下单的时候，服务器端怎么知道是谁下的单？

**实现会话控制：**

服务器端要知道当前客户是谁，需要依赖于cookie；

住过旅社：cookie相当于房卡

进房间：房卡—》旅社给你的住房的证明；

服务器端给客户端的cookie，对用户进行跟踪，实现会话控制；

Cookie是存在客户端的，是服务器响应的时候存在客户端的标识；

有大小限制：4K；

当客户端发起请求的时候，请求头里面会自动加上cookie信息；

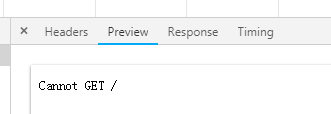
然后服务器端可以收到这个cookie，

通过cookie可以判断当前用户是谁以及他以前做过什么事情；

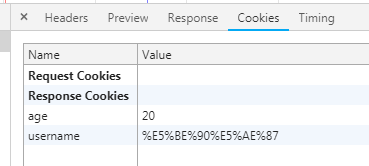
安全性比较低：用户可以修改和删除cookie；

签名可以提高cookie的安全性：齐缝章

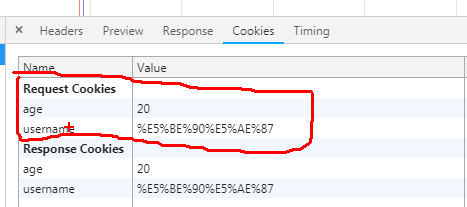
金额 银行卡信息



第一次，刚进来：此时服务器端响应的有cookie，客户端还没cookie



第二次：当用户再次访问时，请求头就会自动加上cookie信息，不需要程序设置



Multer Cookie cookie-parser FormData

Res.cookie(‘name’, ‘value’);

Req.cookies

Session 验证码