Express中间件设计思想：

利用use方式来进行无差别代码修饰，最后通过内置的next将修饰好的代码交给下方的程序员代码区

express、koa

中间件：

1.写、用在哪儿

2.原理

--------------------------------------------------------------------------------

设计：

server.use(function (req, res, next){

处理...

...

...

next();

});

server.get(xxx);

--------------------------------------------------------------------------------

server.post('/user', next)

server.post('/reg')

--------------------------------------------------------------------------------

实现？

--------------------------------------------------------------------------------

响应方法：

res.send

res.sendFile 控制

server.use(express.static) 自动

res.sendStatus(code) 404

res.redirect(url) 重定向

原生Node：

res.writeHeader(404);

res.write('not found');

express：

res.sendStatus(404);

--------------------------------------------------------------------------------

文件上传-multer

//1.引入、配置

const multer=require('multer');

let multerObj=multer({dest: './xxx'});

//2.加给中间件

server.use(multerObj);

//3.用files

req.files

文件重命名 fs.rename(原地址，新地址，cb)

--------------------------------------------------------------------------------

cookie、session：

1.cookie——浏览器存数据，在请求服务器的时候，会被带到请求里面

缺点：容量有限(4K)、不安全(用户随意篡改)

2.session——只存在服务器端

优点：容量不限、安全(用户跟本碰不到)

session基于cookie

风险：如果sess\_ID泄露了——session劫持

1.提醒用户不要在F12的console里面乱输入

2.session\_ID做的非常复杂、session\_ID定期更换

--------------------------------------------------------------------------------

cookie(cookie-parser)：

1.设置 res.cookie(name,value,options)

2.读取 req.cookies

\*3.安全(防篡改)

为啥不自动签名所有cookie：

1.不是所有cookie都敏感

2.签名会导致cookie体积扩大很多

123546

s:123456.DKnjqyiL2vFPHENuKDBlHWXN/F9JFTGw3SgzipPAuFM

--------------------------------------------------------------------------------

session(cookie-session)：

1.设置 req.session['cash']=xxx

2.读取 req.session['cash']-=100

\*3.安全(防篡改)

session sess\_ID

session.sig sess\_ID签名

--------------------------------------------------------------------------------

路由

express.router

--------------------------------------------------------------------------------

服务端渲染

ejs/pug/consolidate

--------------------------------------------------------------------------------

设计数据字典

设计接口

token

--------------------------------------------------------------------------------

管理端

前端

--------------------------------------------------------------------------------

KOA