01 服务器是如何服务前端 把前端变成一个动态网页的

```

const express = require('express')

const app = express ()

app.get('/video',(req,res)=>{
    console.log('有人访问我了/video  索要视频')
    res.send({
        name:'我是视频',
        url:'https://www.baidu.com/',

    })
})



app.get('/car',(req,res)=>{
    console.log('有人访问我了/car 汽车资讯')
    res.send({
        name:'汽车资讯',
        content:'最新新车资讯 。。。。。。。。。。。。。。。。。。。。'
    })
})

app.get('/car/benz',(req,res)=>{
    console.log('有人访问我了/car/benz 汽车资讯 之benz')
    res.send({
        name:'汽车资讯之奔驰最新讯息',
        content:'梅赛德斯-奔驰（Mercedes-Benz），德国豪华汽车品牌，汽车的发明者，被认为是世界上最成功的高档汽车品牌之一，其完美的技术水平、过硬的质量标准、推陈出新的创新能力，以及一系列经典轿跑车款式令人称道。奔驰三叉星已成为世界上最著名的汽车及品牌标志之一'
    })
})

app.get('/car/bmw',(req,res)=>{
```

```
35      console.log('有人访问我了/car/bmw 汽车资讯 之bmw')
36      res.send({
37          name:'汽车资讯之奔驰最新讯息',
38          content:' 宝马汽车，是指宝马汽车公司（Bayerische Motoren Werke AG，简称BMW）生产的汽车，主要的系列车型有1、2、3、4、5、6、7、8等系列。以生产豪华轿车、摩托车和高性能发动机而闻名于世。宝马汽车公司是世界著名的轿车公司'
39      })
40  })
41
42
43
```

02后端路由繁琐 需要配置路由表

原因：如果路由表和业务逻辑写在一起会大量冗余的面条型代码 所以必须有路由表的映射

```
1  const express = require('express')
2
3  const app = express()
4
5  const router = express.Router()
6
7  // 这是第一个路由表
8  router.get('/a',(req,res)=>{
9      res.send('我是get请求的/a')
10 })
11 //这是第二个路由表
12 router.get('/b',(req,res)=>{
13     res.send('我是get请求的/a')
14 })
15
16
17 //路由表只是类似于 wps excel那种文档 想要使用 要明确声明要用
18 app.use(router)
```

```
19
20
21
22 app.listen(8001,()=>{
23     console.log('服务器8001端口启动了')
24 })
25
26
27
28
29
```

03 express.router配置路由表的真正意义用处所在

```
1  //a.js
2  const express = require('express')
3  const goodsRouter = require('./route/goods')
4  const usersRouter = require('./route/users')
5  const app = express()
6
7  app.use( usersRouter)
8  app.use( goodsRouter)
9
10 app.listen(8000,()=>{
11     console.log('服务器启动到8000端口了')
12 })
13
14
15
16
17
18 //routes/goods.js
19 //route/goods    路由表
20
21 // 任务: 配置一个商品相关的路由表
22 const express = require('express')
23
24 //建立表
```

```
const router = express.Router()

// 进行路由表的配置
router.get('/goodsList', (req, res) => {
    res.send('获取商品信息')
  })

router.post('/addGoods', (req, res) => {
    res.send('添加商品')
  })


module.exports = router



//users.js
const express = require('express')
//建立表
const router = express.Router()

router.get('/a',(req,res)=>{
    res.send('我是get /a请求')
})

router.post('/user',(req,res)=>{
    res.send('获取用户信息')
})



module.exports = router


```

04 整合路由表出现的小小的问题

```
核心代码


// 告诉 app 使用这个路由表
// 只有以 /users 开头的请求，会使用这个表
app.use('/users', userRouter)
// 告诉 app 这个表也要使用
app.use('/goods', goodsRouter)
```

05 express 配置静态资源 向服务器请求css的服务器加载

```
1.js

const express = require('express')
const viewsRouter = require('./route/view')
const app = express()



// 挂载静态资源
 // 所有的 /public 开头的都会去到 public 文件夹下寻找内容
 // 按照你请求路径后面的内容去寻找
 app.use('/public', express.static('./public'))
 app.use('/views',viewsRouter)
```

```
14
15 app.listen(8080,()=>{
16     console.log('8080启动了')
17 })
18
19
20
21
22
23
24 /view/index.html
25 <!DOCTYPE html>
26 <html lang="en">
27 <head>
28     <meta charset="UTF-8">
29     <meta name="viewport" content="width=device-width, initial-scale=1.
0">
30     <link rel="stylesheet" href="/public/css/index.css">
31     <title>Document</title>
32 </head>
33 <body>
34     <div>
35         <h1>我是view文件脸的index.html</h1>
36         <img src="/public/1.jpg" alt="行情">
37     </div>
38 </body>
39 </html>
40
41
42
43 /route/view.js
44 const router = require('express').Router()
45 const fs = require('fs')
46
47 router.get('/index.html', (req, res) => {
48     fs.readFile('./view/index.html', 'utf8', (err, data) => {
49       if (err) return console.log(err)
50
51       // res.send() 方法如果返回的 buffer 的数据格式, 会自动下载
52       res.send(data)
53     })
```

```
  })



module.exports = router



/public/
下面一个1.jpg 一个index.css 自行配置
```