node.js day2

Apach功能实现

- return的作用
  - 阻止代码继续往下运行
  - 方法返回值
-

```
// 1.引入http模块
var http=require('http')
var fs=require('fs')
// 2.创建server
var server=http.createServer()
// 3.监听server事件

//index.html和a.txt需要在www文件夹内
server.on('request',function(req,res){
    var url = req.url
    if(url=='/'){
        fs.readFile('./index.html',function(err,data){
        if(err){
            return res.end('404 not fond')
        }
    })
    }else if(url==='/index.html'){
        fs.readFile('./index.html',function(err,data){
            if(err){
                return res.end('404 not fond')
            }
            res.end(data)
        })
    }else if(url==='/a.txt'){
        fs.readFile('./a.txt',function(err,data){
            if(err){
                return res.end('404 not fond')
            }
            res.end(data)
        })
    }
    else{
        res.end('wrong')
    }
})
// 4.绑定端口号，启动服务器
server.listen(3000,function(){
    console.log('running.....')
})
```

```
// 方法二 使用filePath可以更简单的完成第三步
server.on('request',function(req,res){
var url = req.url
var path='./'
var filePath='/index.html'
if(url !== '/'){
    filePath = url
}
fs.readFile(path+filePath,function(err,data){
    if(err){
        return res.end('404 not fond')
    }
    res.end(data)
})
})
```

- 在浏览器中文件目录列表的实现
    - 获得 www目录下的文件名
        - Fs.readdir 参数：error files
    - 将获得的文件名替换到template.html中
    - 模板引擎 art-template
        - 安装 npm install art-template --save
        - 默认下载到node_modules 目录中
        - 模板引擎不关心你的字符串内容，只关心能识别的模板语法
        - 在node中使用时需要引入,用require
        - 将模板源代码编译成函数并立刻执行 template.render(source, data, options);
- 服务端渲染和客户端渲染的不同
    - 客户端渲染不利于seo搜索引擎优化
    - 服务端渲染可以被爬虫抓取的，客户端异步渲染是很难被爬虫抓取的
    - 所以一般网站及不是纯异步也不是纯服务端渲染，由两者结合完成
    - 京东商城商品列表是服务端渲染，为了搜索引擎优化，而商品评论为了用户体验，就采用客户端渲染

留言本案例

- 页面构建
    - app.js 服务器页面
    - index.html 留言本首页，标题，发表按钮，评论按钮
    - post.html 发表评论页面 标题，姓名文本框，内容文本框，提交按钮
- app.js
    - 引入模块 http fs等
    - 自定义部分数据到comments，方便初期观察
    - 创建server
    - 路径判断

- 定义一个comment接收post页面提交的信息，将comment用unshift插入到评论列表开头
- 让用户重定向跳转到首页 /
  *

```
res.statusCode = 302 //302表示状态码  301是永久重定向 302是临时重定向
res.setHeader('Location','/') //表示要重定向跳转到位置是 / 即首页
```

  - 302是一个普通的重定向代码。直观的看来是，请求者（浏览器或者模拟http请求）发起一个请求，然后服务端重定向到另一个地址。而事实上，服务端仅仅是增加一条属性到header，location=重定向地址。而一般的，浏览器会自动的再去请求这个location，重新获取资源。也就是说，这个会使得浏览器发起两次请求。
- 绑定端口号，启动服务器

- index.html
  - 获取app页的数据

```
<ul class="list-group">
  {{each comments}}
  <li class="list-group-item">{{ $value.name }}说: {{ $value.message }} <span
class="pull-right">{{ $value.dateTime }}</span></li>
  {{/each}}
</ul>
```

- post.html

  - 表单中需要提交的表单控件元素必须具有 name 属性 表单提交分为：

    1. 默认的提交行为
    2. 表单异步提交

    action 就是表单提交的地址，说白了就是请求的 url 地址 method 请求方法 get post

  - 区别

    - GET把参数包含在URL中，POST通过request body传递参数
    - GET请求参数会被完整保留在浏览器历史记录里，而POST中的参数不会被保留
    - GET比POST更不安全，因为参数直接暴露在URL上，所以不能用来传递敏感信息
    - 对于GET方式的请求，浏览器会把http header和data一并发送出去，服务器响应200（返回数据）； 而对于POST，浏览器先发送header，服务器响应100 continue，浏览器再发送data，服务器响应200 ok（返回数据）。