中国科学 : 信息科学

SCIENTIA SINICA Informationis

论文

NMail

徐杨鑫[1]

1. 南京大学仙林校区, 中国-江苏-南京210023
E-mail: poker-svg@foxmail.com

摘要 此文为《网络应用开发技术》的课程设计项目的说明报告，主要介绍在开发一个简易的e-mail系统（NMail）的过程中所展现出的灵感来源、设计理念、具体框架、详细代码、优化处理等，此文的研究目的是从实际的项目开发中熟悉、掌握、理解基础的Web开发技术中所涉及的前后端语言、工具等，例如HTML、CSS、JavaScript、Node.js、npm等。

关键词 Web开发技术, E-Mail , HTML , CSS , JavaScript , Node.js , npm

1 灵感来源

刚开始上这门课的时候感觉知识很多很复杂，而且课程设计的项目没有一个明确的方向，想做很多东西但又怕能力跟不上。后来老师建议我们先搞起来再说，任何项目开发过程中遇到的问题都可以见招拆招，一点点地解决，就怕拖而不决。

恰巧这个时候我在Steam上买了一款游戏——《》，然后我的学生邮箱中就收到了这样一封邮件：

于是我就萌生了自己写一个邮箱系统的想法，一个具备e-mail 基础功能的网页。但是这个想法尚未落地，我又从未开发过类似的项目，自然也不知道最终的项目工作量大小，但无论怎样，先开始，总是好的。

2 技术栈

《网络应用开发技术》这门课主要教学目标是掌握基本的前后端开发技术，而如今随着互联网的兴起，以及浏览器这一杀手级的应用不断更新迭代，开发网络应用所需的前后端技术也飞速发展。好的一方面是面对开发网络应用我有了非常多的选择，坏的一方面是前后端技术的选择太多导致我不知道什么样的技术适合我的项目开发。

我也是消耗了一部分时间去了解和选择如今较为流行的几种前后端开发技术栈链。而我选择的标准是：

1.复杂度低，易于快速入手。由于我是入门新手，复杂的技术栈链所需的学习周期长，不利于快速投入开发。

2.轻量级。这是因为我的开发目标（E-Mail邮箱）是一个玩具性质的小型项目，轻量级技术栈更符合需求。

3.开源免费且有丰富第三方库。这是为了加快开发进度，丰富项目功能。

通常我们会将技术栈分为前端、后端、中间件、数据库和工具，下面我会一一介绍此项目中所使用的技术栈链：

2.1 前端

2.1.1 html

2.1.2 css

2.1.3 JavaScript

2.2 后端

2.2.1 Node.js

2.2.2 express

express是一个简洁高效的运行在Node.js上的Web服务器框架，可以用于快速部署常见的Web服务器。

它的主要特点有：

1.健壮的路由

2.专注于高性能

3.超高的测试覆盖率

4.HTTP帮助(重定向，缓存等)

5.视图系统支持14+模板引擎

6.内容协商

7.用于快速生成应用程序的可执行文件

2.2.3 mysql

由于后端服务器所连接的数据库是mysql，所以需要一个第三方库来辅助服务器连接和操作对应的数据库，这里我选取的是非常流行的mysql第三方模块。它是mysql的node.js驱动程序，使用JavaScript编写。

2.2.4 bcryptjs

由于为了安全性考虑，存储在数据库中的用户信息，尤其是用户密码不应以明文形式存储。因此需要对密码进行加密处理后再存储到数据库中，目前有很多加密算法和第三方工具包，这里我选择的是bcryptjs，该库与CommonJS和AMD加载器兼容，并以dcodeIO的形式在全球公开。它的特点包括：

1.加密后的密码无法被逆向破解。

2.同一明文密码多次加密得到的加密结果不同。

2.2.5 nodemailer

用于实现发送邮件功能的第三方工具库

2.2.6 imap 和mailparser

用于实现接收邮件功能的第三方工具库

2.3 中间件

2.3.1 CORS中间件

CORS是一个node.js包，用于提供一个Connect/Express中间件，该中间件可用于使用各种选项启用CORS。

2.4 数据库

2.4.1 MySQL

2.5 工具

Node.js具有丰富的第三方库工具，可以使用npm进行包的下载。这里我简要介绍我所使用的工具：

2.5.1 apiDoc

apiDoc是一个非常流行的全局包，可使用npm install -g apidoc进行下载。它允许我们在源代码中编写注释，然后自动生成响应的api 文档，而且支持Go, Dart, Java, JavaScript, PHP, Scala等多种语言。这可以大大减少后端人员编写api文档所需时间，也便于前端人员阅读。

2.5.2

3 需求分析

3.1 用户信息

不同的用户会有不同的用户信息，而这些信息的存储、认证、修改、删除等操作都是N-Mail项目的基本要求。具体包括：

3.1.1 用户信息存储

3.1.2 用户信息格式

3.1.3 用户身份认证

这具体包括：

1.用户必须登陆成功才可以进入自己的后台

2.用户的账号信息在前后端交互过程中需要进行加密传输，以防止爬虫导致的信息泄露

3.用户退出后需要删除存储在浏览器中的有关信息痕迹

4 相关难点

《网络应用开发技术》这门课的一大特点就是没有理论考试，只有课程设计项目，这就意味着这门课的所有收获都来自项目开发过程中所遇到的一个个难点，持续不断地“发现困难、解决困难”是这门课的正确打开方式。因此接下来我会把在项目开发过程中遇到的点点滴滴的难点记录下来：

4.1 CORS

首先先来了解一下浏览器的同源问题：当一个请求在浏览器端发送出去后，服务端是会收到的并且也会处理和响应，只不过浏览器在解析这个请求的响应之后，发现不属于浏览器的同源策略（地址里面的协议、域名和端口号均相同）也没有包含正确的CORS 响应头，返回结果就会被浏览器拦截。

而跨域资源共享(Cross-origin Resource Sharing, CORS)就是为了解决在浏览器发出只能在同源的情况下向服务器获取数据的限制而产生的。

4.2 用户身份认证

根据上面的需求分析我们得知，用户身份认证包括

1.证明你是你自己

2.认证过程需要加密

3.退出需要删除登陆痕迹

4.非登录用户无法进入后台

所以这里我使用了token技术：

首先，用户在注册过程中会将自己的密码hash为一个token，将token存储在浏览器的Local Storage中，随后取出浏览器本地的token传输给后端，后端会将这个token看作对应用户的密码存储在数据库中。实现了“认证过程需要加密”。

然后，用户在登陆时，会将自己的密码hash为对应的token并发给后端，后端验证成功后会跳转到邮箱后台的主界面index.html。实现了“证明你是你自己”。

再然后，用户在退出邮箱后台时会自动删除浏览器本地的Token，防止用户信息泄露。实现了“退出需要删除登陆痕迹”。

最后，如果用户强制进入后台（例如在浏览器地址栏中输入后台网页地址并跳转），前端的js代码会自动取出浏览器本地的token传输给后台进行身份认证。由于此时浏览器本地token为空，所以导致身份认证失败，从而强制自动跳转到登陆界面，保证后台邮箱的安全性。实现了非登录用户无法进入后台。

4.3 中间件

express的一大特点就是中间件的使用。那什么是中间件呢？中间件的作用又是什么？

实际上中间件就是一种特殊的封装，在C语言的学习中我们了解到很多重复的类似的，仅是参数不同而功能相同的代码可以依靠函数进行过程式封装，从而实现代码的集中和简洁。

而在服务器的搭建中，我们可以将服务器对客户端发来的数据的处理过程看作一条处理链。在许多服务器模块处理不同数据之前，可能需要进行相同的对数据的“预处理”。因此为了实现预处理的集中封装，中间件便被引入了express框架中。

我们可以在服务器的全局或者局部模块挂载相应的中间件处理链，这样数据在进入某个实际的功能结点之前，就会沿着预先设定的中间件处理链进行一系列“预处理”，从而简化功能结点的代码量。下面我们不妨来举个项目中的例子：

4.3.1 响应数据中间件

4.3.2 数据验证中间件

一般而言，在前后端的开发过程中会遵循“后端底线原则”，即后端不相信前端所做的任何验证，因此后端需要对前端发来的数据进行全面的验证，以确保数据的合法性。但是如果后端完全不借助任何第三方库手搓验证数据的话，会使得代码非常臃肿而且效率低下，所以对于某些较为基础和普遍的数据验证，后端人员会借助第三方库来实现。

这里我选择的第三方库是@escook/express-joi，并且我将它挂载到全局中间件中，使得数据会先经过验证再得到处理。从而保证了后端服务器的安全性。

4.3.3 发送邮件

发送邮件是N-Mail项目的核心功能，但是它的实现却并没有非常困难，因为我们有非常丰富的第三方工具库可以辅助实现。

唯一的难点是我并没有真正的email服务器，因此我并不能真正实现某个自定义邮箱后缀的发送，所以我必须借助其它的邮箱服务器(这里我选择的是qq邮箱的服务器)来实现邮件代发，这种代发的缺点是发送方邮箱地址必须和经过验证的邮箱地址相同，也就是说发送方的邮件地址只能是我的qq邮箱地址。这种代发机制也是为了防止伪装诈骗邮件的发送。

因此N-Mail项目的发送邮件功能具有天然的限制性，所有用户都只能通过我的邮箱地址代发，不过我们可以通过在邮件信息中对发生用户的身份进行标注

4.3.4 接收邮件

接收邮件也是N-Mail项目的核心功能，同样由于第三方工具库的存在我们可以顺利实现此功能需求。

但同样的，由于代发机制的限制，其它用户向我们N-Mail的用户发送邮件时只能向代发服务器发送邮件，而且需要在邮件正文中按特定格式指定目标用户，这样接收邮件的模块才会从代理服务器中监听邮件并进行分发。

这实际上就是一种简陋的NET机制(用代理服务器去弥补资源不足)

还有一点值得注意的是，由于接收邮件需要持续监听（和主服务器的工作线程独立），因此这里还涉及一点简单的多线程。

5 一级标题

正文开始. 下载测试

定义1 (定义名,可省略)　这是一个定义.

图 1　图题
Figure 1　Caption

表 1　表题
Table 1　Caption

Title a	Title b	Title c	Title d
Aaa	Bbb	Ccc	Ddd
Aaa	Bbb	Ccc	Ddd
Aaa	Bbb	Ccc	Ddd

图片如1所示.

5.1　二级标题

表格如表1所示.

5.1.1　三级标题

算法如算法1所示.

算法 1 算法标题

输入: $n \geqslant 0 \vee x \neq 0$;
主迭代: $y = x^n$;
1: $y \Leftarrow 1$;
2: **if** $n < 0$ **then**
3: $X \Leftarrow 1/x$;
4: $N \Leftarrow -n$;
5: **else**
6: $X \Leftarrow x$;
7: $N \Leftarrow n$;
8: **end if**
9: **while** $N \neq 0$ **do**
10: **if** N is even **then**
11: $X \Leftarrow X \times X$;
12: $N \Leftarrow N/2$;
13: **else** $\{N$ is odd$\}$
14: $y \Leftarrow y \times X$;
15: $N \Leftarrow N - 1$;
16: **end if**
17: **end while**
输出:

参考文献

1　Author A, Author B, Author C. Reference title. Journal, Year, Vol: Number or pages
2　Author A, Author B, Author C, et al. Reference title. In: Proceedings of Conference, Place, Year. Number or pages

Xu YangXin[1]

1. *Xianlin Campus of Nanjing University, Nanjing-JiangSu* 210023*, China*
E-mail: poker-svg@foxmail.com