**Node综合blog案例**

1. 案例介绍

本案例是基于nodejs和MongoDB等实现的一个多人社区聊天工具，具有用户登入、注册、注销、发表话题、发表评论、修改信息等功能。

1. 目录结构

/

app.js 项目的入口文件

models 存储使用mongoose设计的数据模型

node\_modules 第三方包

package.json 包描述文件

package-lock.json 第三方包版本锁定文件

public 公共的静态资源

routers 如果业务较多代码量大 最好把路由按照业务的分类存储到该目录中

router.js 简单一点把所有路由都放到这个文件

views 存储视图目录

1. 项目运行需要事先加载的第三方包

express

art- template

express-art-template

bootstrap

body-parser

china-time

jquery…等

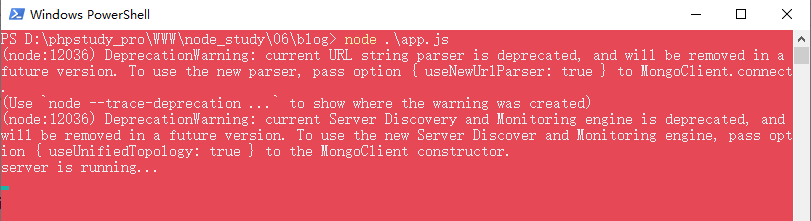
1. 路由设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 路径 | 方法 | get参数 | post参数 | 是否需要登入 | 备注 |
| / | GET |  |  | 不需要 | 渲染首页 |
| / | POST |  |  | 需要 | 处理用户注销请求 |
| /register | GET |  |  | 不需要 | 渲染注册页面 |
| /register | POST |  | email nickname password | 不需要 | 处理注册请求 |
| /login | GET |  |  | 不需要 | 渲染登入页面 |
| /login | POST |  | email password | 不需要 | 处理登入请求 |
| /logout | GET |  |  | 需要 | 处理退出请求 |
| /topic/new | GET |  | block motif title txt code | 需要 | 发表新的话题 |
| /setting/profile | GET |  |  | 需要 | 渲染修改用户信息页面 |
| /setting/profile | POST |  | nickname bio gender  last\_modified\_time  birthday | 需要 | 将修改的信息提交 |
| /setting/admin | GET |  |  | 需要 | 渲染账户设置页面 |
| /setting/admin | POST |  | pwd\_now  pwd\_change1  pwd\_change2 | 需要 | 处理修改密码的请求 |
| /setting/show | GET | comment |  | 需要 | 处理提交评论的请求 |
| /setting/show | POST |  | mytitle myemail | 需要 | 处理点击话题后的请求 |

1. 使用方法

在运行之前，将所需要的第三方包加载好，并在本地打开MongoDB数据库。接着运行入口文件app.js，就可以在局域网中根据开放的端口访问到了。

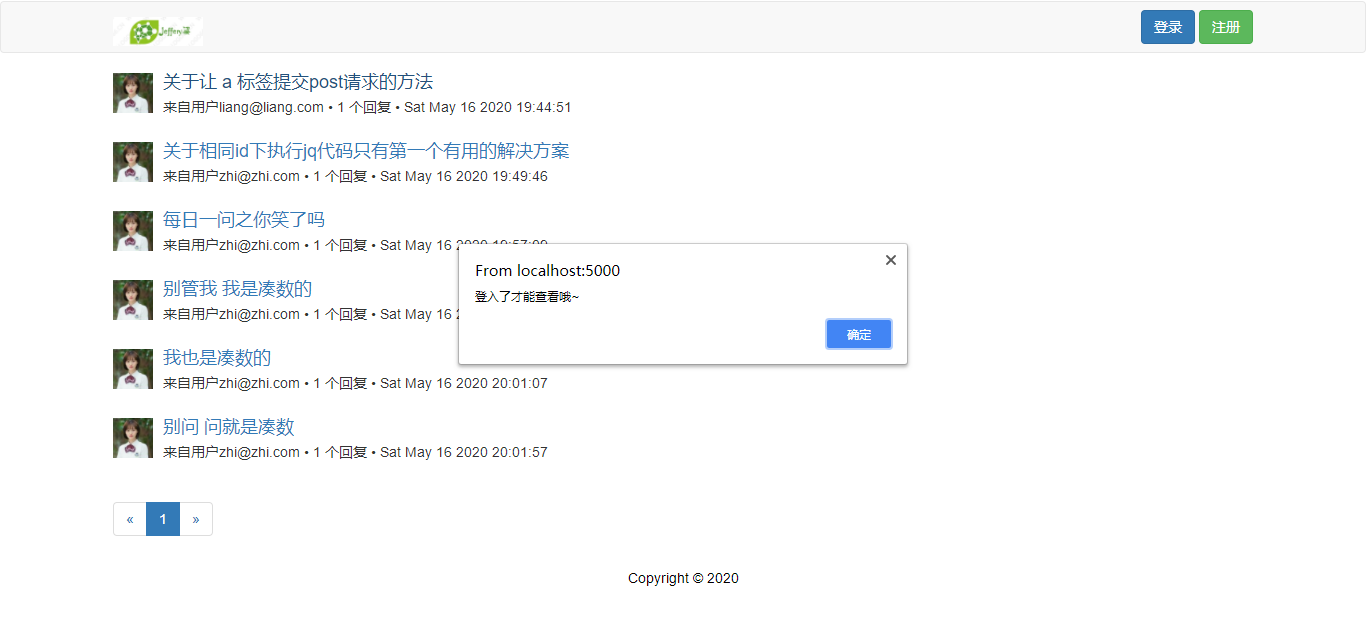
1. 实例截图



开放服务器 开放端口为5000 接下来以本机访问为例展示：



未登入时不能查看话题的全部内容：

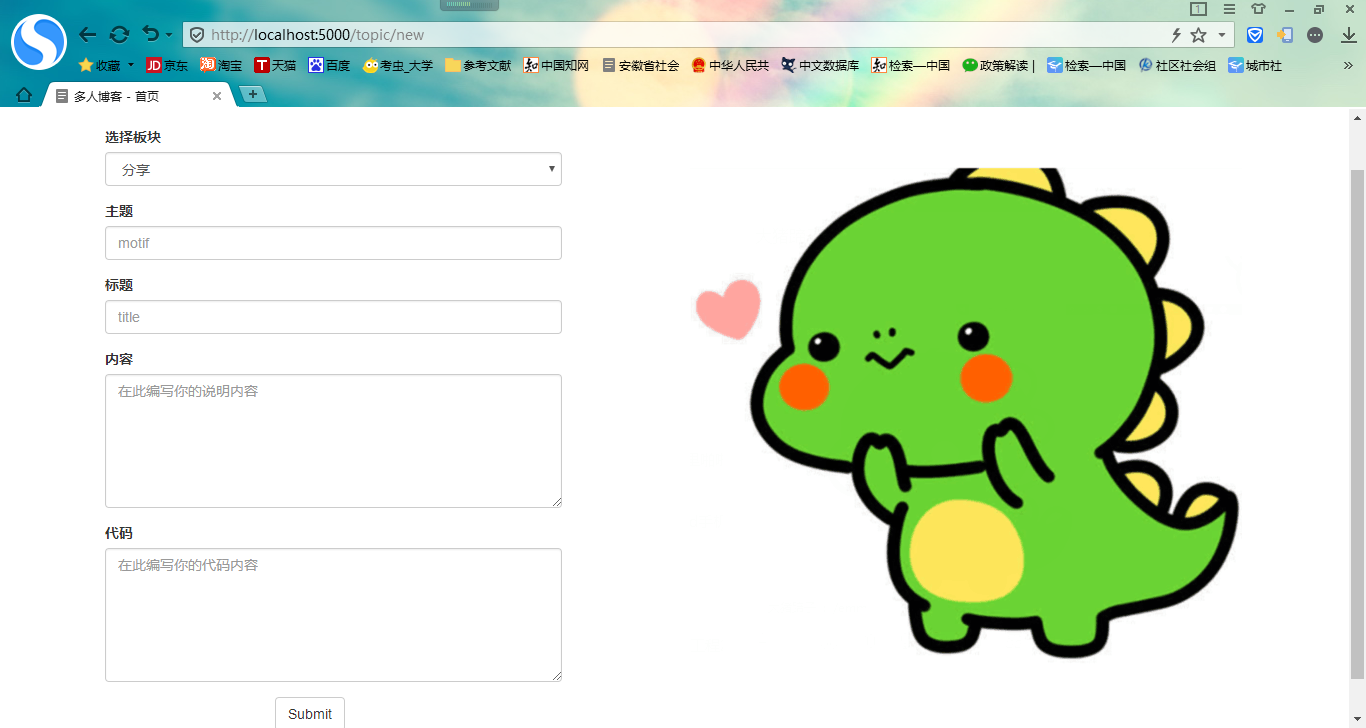


登入之后可以查看某个话题的主要内容和发表评论：





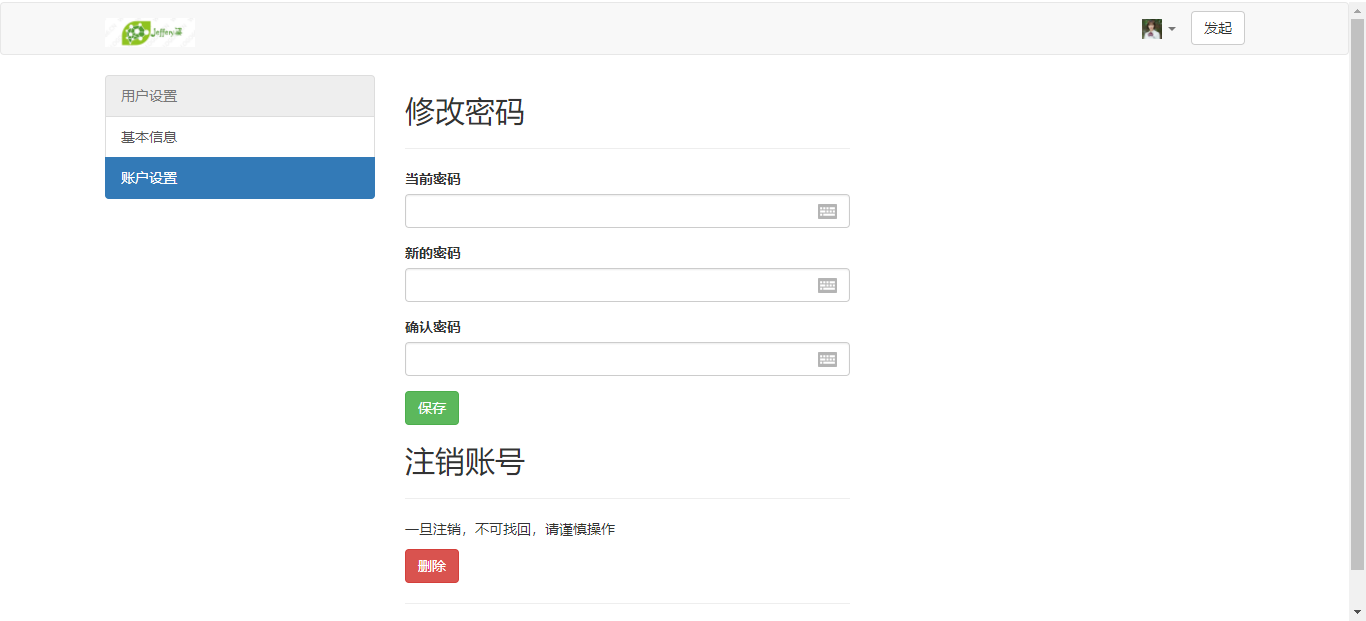
也可以自己发起一个话题：



修改信息页面：



修改账户密码或注销页面：



以及用户的登入 注册页面：



1. 项目总结

在这个项目中，学习到了许多新的知识，比如如何利用模板引擎在HTML中使用模板继承和子模板，node中接收get和post请求的方法，html中如何让 a 标签也能发送post请求，mongoose操纵数据库的各种方法等等。总体来说收获还是很大，但是还是有很多的不足，项目中有些自己想实现的功能但是基于自己目前的能力也没有完全实现，希望自己能在未来不断补充和完善。