

基于Go语言的博客web系统

——彭云

|  |  |
| --- | --- |
| 学 院 | 计算机学院 |
| 专 业 | 软件技术 |
| 班 级 | 1414班 |
| 学 号 | 2014030962 |
| 姓 名 | 彭云 |
| 指导教师 | 赵丙秀 |

目录

[1.1 摘要 3](#_Toc22202)

[1.2 关键字 4](#_Toc26254)

[1.3 前言 4](#_Toc19563)

[1.4 设计思路 6](#_Toc16381)

[1.5 检测方法 7](#_Toc1595)

[1.5.1 通过LiteIDE的快捷键编译 8](#_Toc24405)

[1.5.2 使用git客户端编译 8](#_Toc25722)

[1.6 结论 10](#_Toc10209)

[2.1 系统简介 12](#_Toc6298)

[2.2 软件系统环境及工具配置 12](#_Toc15188)

[2.2.1 Go环境搭建 13](#_Toc17250)

[2.2.2 安装LiteIDE 14](#_Toc11704)

[2.2.3 配置Mysql 17](#_Toc11688)

[2.3 技术可行性 22](#_Toc23982)

[2.3.1 技术可行性 22](#_Toc14348)

[2.3.2 个人可行性 22](#_Toc15316)

[2.3.3 需求分析 22](#_Toc16707)

[3.1 系统需要解决的主要问题 23](#_Toc25173)

[3.2 系统具备的基本功能 23](#_Toc32750)

[3.2.1 结构设计 24](#_Toc9704)

[4.1博客系统部分功能展示及实现原理分析 25](#_Toc29725)

[4.1.1 注册页面 25](#_Toc21445)

[4.1.2 登录页面 29](#_Toc30183)

[4.1.3 登录成功后 32](#_Toc11362)

[4.1.4 文章分类管理 33](#_Toc20927)

[4.1.5 文章发布管理 34](#_Toc26866)

[4.1.6 博客留言管理 35](#_Toc10346)

[5.1 概述  35](#_Toc18938)

[6.1 测试的任务及目标 36](#_Toc14285)

[6.1.1 测试的任务  36](#_Toc24446)

[6.1.2 测试的目标  36](#_Toc6879)

[6.2 测试方案  36](#_Toc32651)

[6.2.1 模块测试： 37](#_Toc17035)

[6.2.1 模块测试：  37](#_Toc17968)

[6.2.2 集成测试Goconvey工具包： 38](#_Toc13894)

[6.2.3 验收测试 39](#_Toc4642)

[7.0 结论总结  40](#_Toc25248)

# 1.1 摘要

研究的内容：基于Go语言的web开发

目的：在熟悉go语言的情况下巩固框架基础,学习如何从底层封装golang代码，加快开发效率,更深刻的了解web开发中前后台之间的交互流程

实验方法：使用html5+bootstrap+angular.js开发web前端页面，golang以及beego开发web后台逻辑

主要成果：通过go语言以及beego框架实现多张数据表的增删改

特色:简洁明了,代码明晰

# 1.2 关键字

**MVC 轻量级 简明 ORM 自动 视图分离**

# 1.3 前言

博客英文为Blog是Web Log的混成词，意指log on the web即在网络上记录，是可以由个人管理;张贴新的文章图片或者视频的网站,主要可以用来记录抒发情感和分享信息，如今博客已成为社会媒体网络的一部分。世界比较有名的博客网站有最早提出设计博客系统的Open Diary以及后来著名博客做着Brad Fitzpatrick创立的LiveJournal，而后[Evan Williams](https://zh.wikipedia.org/w/index.php?title=Evan_Williams&action=edit&redlink=1" \o "Evan Williams（页面不存在）)和[Meg Hourihan](https://zh.wikipedia.org/w/index.php?title=Meg_Hourihan&action=edit&redlink=1" \o "Meg Hourihan（页面不存在）)推出了blogger.com,此站点于2003年2月被[Google](https://zh.wikipedia.org/wiki/Google" \o "Google)公司收购。国内比较有名的博客有[新浪](https://zh.wikipedia.org/wiki/%E6%96%B0%E6%B5%AA" \o "新浪)提供的博客服务，以[名人](https://zh.wikipedia.org/wiki/%E5%90%8D%E4%BA%BA" \o "名人)博客为特色;[搜狐](https://zh.wikipedia.org/wiki/%E6%90%9C%E7%8B%90" \o "搜狐)提供的博客以及[Blogbus](http://www.blogbus.com/)：中国大陆第一家Blog托管服务商（Blog Service Provider）,是商业运作、提供收费服务的中文Blog网站;我个人也比较倾向于国内拥有最多程序员的CSDN还有博客园。说到为何要建立这个博客系统，其中很大的一个原因便是我个人的性格喜好。我现在的工作虽然是一名刚实习即将转正的软件开发工程师，但是我比较喜欢在网上搜阅一些技术文章以及他人分享过的生活日志，我发现每个对生活有梦想有追求的技术爱好者他们一般都会拥有自己的博客又或者说是自己的空间，用来记录他们在发现问题并且解决问题后的过程及感受给后人一定的提示，很高效的解决了我们生活中及工作中的一部分共同面对的问题。关于这个博客系统我更想做到的是寄居第三方服务的个人博客，就像新浪博客和网易博客这类的，但是我会提供永久免费的使用权以及后期的维护与升级。因为我认为个人博客无论选择是否去盈利，是否想去搏出名，我相信所有搞个人博客的第一个原因就是为了能记录与展示，这点与玩朋友圈、空间、微博等目的是一样的。而区别在于个人博客可以更加充分的记录与展示自己，我想写多少都可以，文字、图片、视频、特效等等我想怎么弄就怎么弄。而朋友圈等则因为其规则限制和格局限制，基本都属于碎片化展示。另一方面个人博客第二个价值，是有效并且节省时间、精力来扩展人脉。这种通过个人博客扩展的人脉是与微博、QQ空间等结识的朋友圈完全不同的（微信朋友圈基本就是建立在从小到大现实中认识的朋友之间的，例如亲人、同学、朋友，不存在还需要相互了解的问题）。另一个价值我觉得是往事全面的回忆。如前所说，个人博客基本都是详细记录自己的生活，而朋友圈等是碎片化记录。假如你已经是个老人，年轻时的很多事肯定都记不清了，而翻阅自己以前的博客肯定能通过详细的文字勾起很多的回忆。而翻阅自己的微博、朋友圈，上面大部分都是几张图片，然后配上几行文字描述。这两者的差距应该不多比较了吧。无论什么时候，博客的价值在于通过写来提升自己。作为开发者，我有兴趣也有义务去做好这个系统，其中不仅仅是我对于展示自己记录生活的热爱，更是我对技术的追求以及喜好！

# 1.4 设计思路

由于本人今年暑假已出来参加实习工作，而且公司使用的主要编程语言是golang加上本人对web编程的喜爱，于是便设计出了这款基于go语言的web系统。主要思路还是mvc的设计模式以及双向的数据绑定,以下简洁介绍下我的设计思路:

核心：网页前端ajax的绑定来实现前后台对接

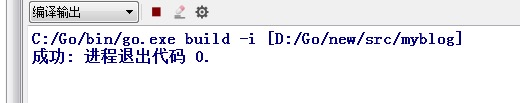
设计流程主要是先设计出网页的前端页面以及搭建好框架，按照mvc的设计模式来做好开发，首先肯定开发的是web的网页前端，包括界面设计以及表单输入框的关键索引,由于之前对前端的框架稍有了解，所以采用了略为臃肿功能完善的bootstrap框架来开发网页的前端界面，主要界面包括网页前端显示的博客文章以及博客分类，还有留言板以及关于界面，其中关于界面采用markdown的语法来解析将.md的文件解析为HTML来供用户查看,后续开发的界面还有网页的前端登录以及注册界面，管理员界面包括文章的发布管理以及分类管理，留言管理，文件附件上传等，其中首先开发view层是由于做好视图分离后，后端开发不用依赖于前端，所以按照view设计到model层和数据交互的方面开发然后最后到控制层的开发，整个开发过程中也存在不少问题，如之前所遇到的beego中使用orm操作数据库出现数据库连接问题以及浏览器的缓存问题，总体来说收获也是不少的。

# 1.5 检测方法

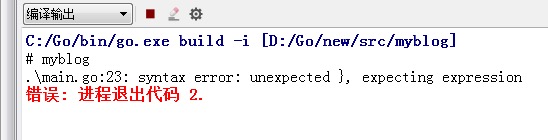
由于本系统采用的是go语言以及beego搭建，为测试能够使用可通过命令(使用git bash的情况下) cd $GOPATH/src(进入到你配置好的GOPATH下) 然后cd myblog下, 执行Go语言的编译命令 go build main.go ,系统将会和C语言以及C++一样生成一个可执行的exe文件，双击运行这个exe文件,将会打开一个WebServer服务器,在这里你可以看到启动的端口号并且你可以通过浏览器访问这个对应的端口访问对应的页面，你将可以访问本系统对应的Web页面;如果出现错误控制台将会提示对应的错误并且这个Web服务器将无法正常启动起来,具体详情请查看以下截图:

# 1.5.1 通过LiteIDE的快捷键编译

以下为控制台编译成功截图:



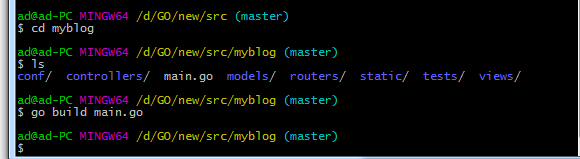
如编译失败则:



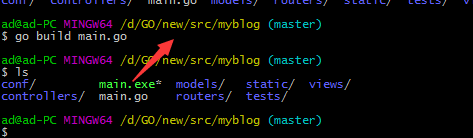
如图上,程序将无法正常运行。

# 1.5.2 使用git客户端编译

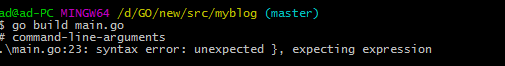
编译成功截图:



如图git bash没有提示错误,通过ls显示当前文件夹命令可以发现系统生成了可运行的main.exe文件如下图:



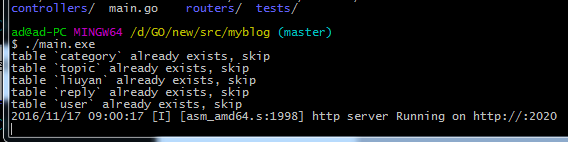
编译失败截图:



和java一样，程序对应的会提示出相应的错误信息并且精确到具体文件的行数，你可以进入相应的文件内找到错误并且修改它

程序启动截图:

通过执行main.exe 文件可以看到控制台对应的信息:



此时本地已成功启动一个webserver服务器,通过在浏览器中输入localhost:2020或者127.0.0.1:2020便可进入对应的web页面:



如图所示,便成功打开了之前写好的web页面。

# 1.6 结论

本系统的搭建成功说明golang的程序远比java以及php的web程序要简易，而且执行效率及性能要更好,经过测试golang只需要不到30行的代码便可实现一个简易的Web服务器，如图下所示:



　　执行上面的程序之后，打开 Web 浏览器，在地址栏输入：127.0.0.1:9090 就可以访问这个服务器了。它只实现了最简单的单一页面的输出。

之前早在大一我便开始接触java以及php的web开发课程，在使用了go语言后我深深地被这门语言所吸引,采用beego框架后web应用程序的搭建便更加容易实现，beego和java的ssh以及ssm框架一样，拥有orm的数据映射功能,能够自动连接到数据库以及生成对应的数据表,orm的功能同样能够在你增加数据表的字段后生成对应的数据表，从而能够实现Web程序和数据库进行交互，通过表单提交来传递数据使得web的开发更加容易!

# 2.1 系统简介

本系统中主要实现的功能有管理员的登录以及用户的登录注册，其中管理员的账号密码默认为admin及123456,通过对应的账号可以登录到对应的系统中区，如使用admin+123456可登录到对应的管理员界面，使用用户的账号以及密码可登录到对应的用户界面,其他的包含了4个功能模块：文章发布管理，文章分类管理，文章统计管理，系统留言管理。

本系统使用html5进行网页界面的设计，前端框架采用bootstrap以及angular.js,使用**MVC**设计模式，后台采用了基于go语言的框架beego,它采用了当今软件设计的最新技术，具有开发效率高、设计灵活、生成的软件界面友好美观等特点。本系统中通过orm进行事务管理和数据库进行无缝连接，后端的数据库是mysql，也是一个开源的数据库系统，该数据库具有较高的完整性，一致性和安全性。

# 2.2 软件系统环境及工具配置

Windows+Go语言环境+LiteIDE+Mysql

在编译beego程序时,如已经成功安装beego框架,你便可以通过在GOPATH的src目录下通过bee new创建新的web项目或者API，通过在创建程序下的根目录里通过bee run来运行你的beego项目,如编译成功后无法运行请先在你的电脑环境下安装好mysql的环境，并且通过配置mysql的账号密码（账号我在model中使用是root,密码我设置的是123456），并且web程序的成功开启必须要在mysql中创建一个名为default的数据库，从而和一般的orm数据映射框架一样实现数据库数据表字段的自动添加。

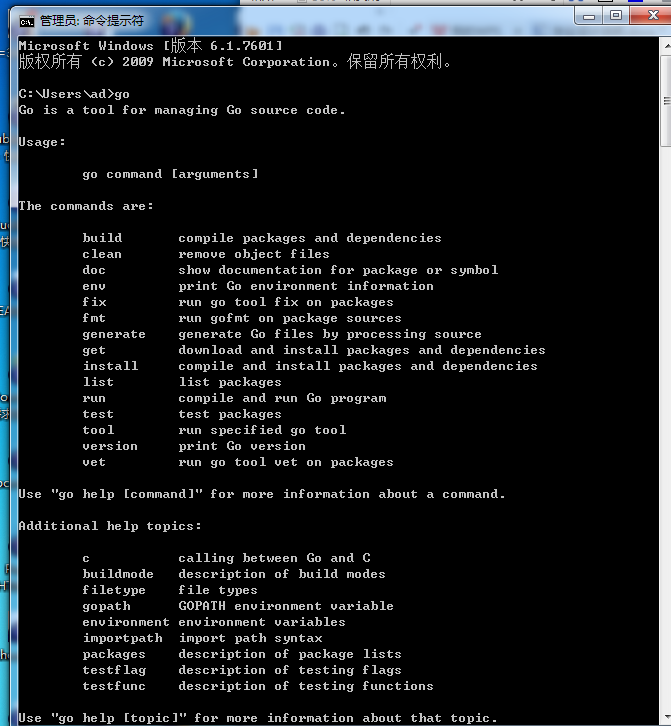
## 2.2.1 Go环境搭建

· 官方说明   
[http://golang.org/doc/install#windows](http://golang.org/doc/install" \l "windows" \t "http://studygolang.com/articles/_blank)

· msi下载   
[https://golang.org/dl/](https://golang.org/dl/" \t "http://studygolang.com/articles/_blank)

· 傻瓜式一路向下安装

· 在系统变量中设置GOROOT为Go的安装目录，GOPATH为golang项目的工作目录

· golang环境测试，cmd或powershell下输入go，显示如下表示安装成功:

## 2.2.2 安装LiteIDE

（1）下载LiteIDE  
       下载地址：[https://code.google.com/p/golangide/downloads/list](https://code.google.com/p/golangide/downloads/list" \t "http://wlb.wlb.blog.163.com/blog/static/4674132015127947893/_blank)  
（2）安装  
       解压缩即可。避免出现中文目录。  
配置LiteIDE  
（1）设置编译环境   
       【查看】-【选项】，双击win64.env，修改如下

# native compiler windows amd64

GOROOT=d:\go#GOBIN=

GOARCH=amd64

GOOS=windows

CGO\_ENABLED=1  
  
PATH=c:\mingw32\bin;%GOROOT%\bin;%PATH%

LITEIDE\_GDB=gdb64

LITEIDE\_MAKE=mingw32-make

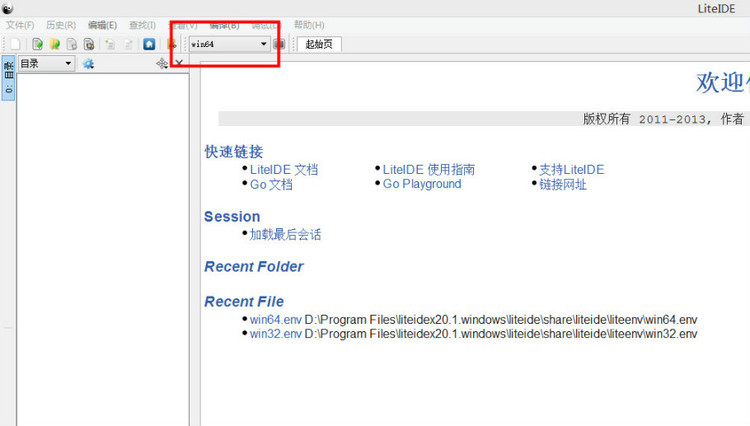
LITEIDE\_TERM=%COMSPEC%

LITEIDE\_TERMARGS=

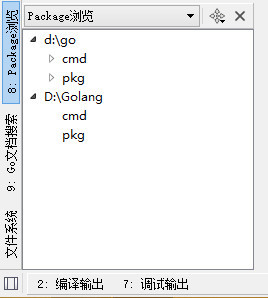
LITEIDE\_EXEC=%COMSPEC%

LITEIDE\_EXECOPT=/C

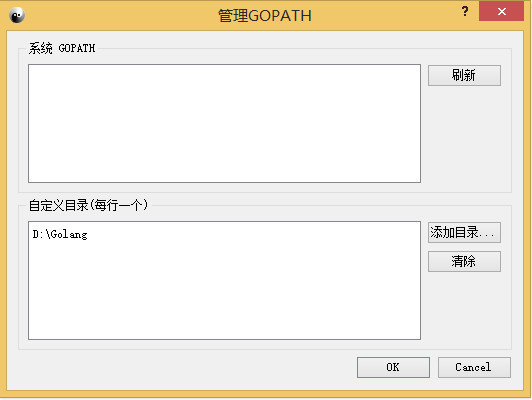
  然后如下图所示：



注意：如果能在Package浏览中看到下图所示，证明配置是成功的,如下图所示:



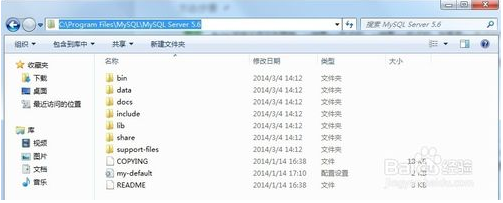
(2)自定义管理GoPath  
【查看】-【管理GoPath】，创建新目录如d:\golang，并添加到自定义目录，如下所示：



## 2.2.3 配置Mysql

MySQL安装文件分为两种，一种是msi格式的，一种是zip格式的。如果是msi格式的可以直接点击安装，按照它给出的安装提示进行安装（相信大家的英文可以看懂英文提示），一般MySQL将会安装在C:\Program Files\MySQL\MySQL Server 5.6 该目录中；zip格式是自己解压，解压缩之后其实MySQL就可以使用了，但是要进行配置。

解压之后可以将该文件夹改名，放到合适的位置，个人建议把文件夹改名为MySQL Server 5.6，放到C:\Program Files\MySQL路径中。当然你也可以放到自己想放的任意位置



完成上述步骤之后，很多用户开始使用MySQL，但会出现图示的错误。这是因为没有配置环境变量所致。配置环境变量很简单：

我的电脑->属性->高级->环境变量

选择PATH,在其后面添加: 你的mysql bin文件夹的路径 (如:C:\Program Files\MySQL\MySQL Server 5.6\bin )

PATH=.......;C:\Program Files\MySQL\MySQL Server 5.6\bin (注意是追加,不是覆盖)





配置完环境变量之后先别忙着启动mysql，我们还需要修改一下配置文件（如果没有配置，之后启动的时候就会出现图中的错误哦！:错误2 系统找不到文件），mysql-5.6.1X默认的配置文件是在C:\Program Files\MySQL\MySQL Server 5.6\my-default.ini，或者自己建立一个my.ini文件，

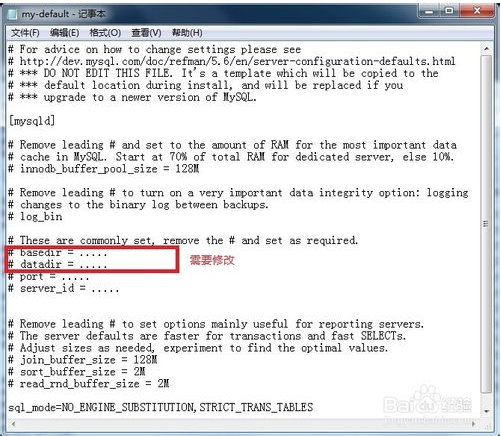
在其中修改或添加配置（如图）：

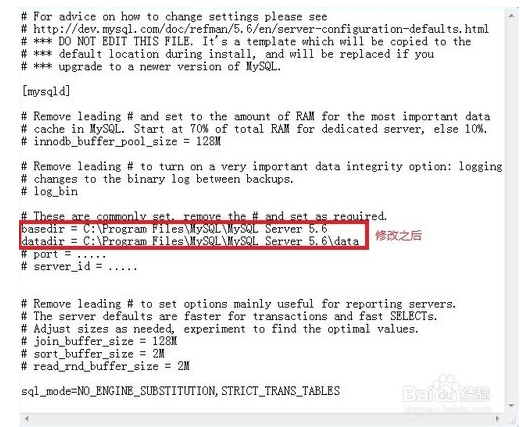
[mysqld]

basedir=C:\Program Files\MySQL\MySQL Server 5.6（mysql所在目录）

datadir=C:\Program Files\MySQL\MySQL Server 5.6\data （mysql所在目录\data）





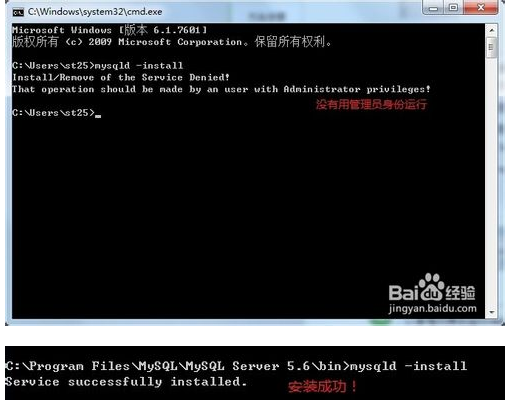


以管理员身份运行cmd（一定要用管理员身份运行，不然权限不够），

输入：cd C:\Program Files\MySQL\MySQL Server 5.6\bin 进入mysql的bin文件夹(不管有没有配置过环境变量，也要进入bin文件夹，否则之后启动服务仍然会报错误2)

输入mysqld -install(如果不用管理员身份运行，将会因为权限不够而出现错误：Install/Remove of the Service Denied!)

安装成功



安装成功后就要启动服务了，继续在cmd中输入:net start mysql（如图）,服务启动成功！

此时很多人会出现错误，请看注意：

注意：这个时候经常会出现错误2和错误1067。

如果出现“错误2 系统找不到文件”，检查一下是否修改过配置文件或者是否进入在bin目录下操作，如果配置文件修改正确并且进入了bin文件夹，需要先删除mysql（输入 mysqld -remove）再重新安装（输入 mysqld -install）；

如果出现错误1067，那就是配置文件修改错误，确认一下配置文件是否正确。



服务启动成功之后，就可以登录了，如图，输入mysql -u root -p（第一次登录没有密码，直接按回车过）,登录成功！接下来使用mysql命令create database default创建default数据库，以便于我们通过beego的ORM来添加数据表字段等。

# 2.3 技术可行性

## 2.3.1 技术可行性

系统的性质为管理博客文章的web软件，因而数据库的设计与操纵是系统设计的核心。这里从两方面分析技术上的可行性。

首先，对于Go编程部分，本人暑假实习的公司便采用是Go语言来开发智能网关，本人通过暑假的实习时间基本上掌握了对Go语言的编程技巧。加上暑假运用了大部分时间来对web界面编程设计的学习。因此在本毕业设计中，对web界面设计方面的问题都能迎韧而解。其次，对于数据库的有关知识，本人运用了大量的时间对相关书籍的阅读以及公司上机实验，再加上网上视频带的学习，基本上掌握了运用MySQL数据库技术，具备一定的系统分析与设计能力，熟悉数据库的设计与操纵；因而该系统的实现在技术上是可行的。

## 2.3.2 个人可行性

本人对该毕业设计有足够的技术上的准备，并且时间上也足够充裕。所以在进行该毕业设计中，不需要人员分工，整个设计过程只由本人完成，并且已经较好的完成了该目的开发工作。所以，在个人可行性方面是可行的。

## 2.3.3 需求分析

# 3.1 系统需要解决的主要问题

本系统要实现对文章的统计以及留言，可以提供给在校的热爱编程以及技术的同学来学习交流，从而解决大家在校学习知识有限的缺憾，并且如果后期有时间，我将一直维护本博客系统，让技术趣味充斥整个校园。

# 3.2 系统具备的基本功能

博客web管理系统主要包括用户部分以及管理员部分，其中用户将只能对文章分类以及文章进行浏览,对博客留言界面可以进行留言，管理员可以对用户进行管理以及对文章留言进行管理，是本系统的最高权限，用户模块的开发主要包括应用程序系统和后台数据库，应用程序系统要求界面友好，功能完备，易使用等特点。而后台数据库的数据要具有一致性、完整性、安全性。本设计根据博客系统分析，以便于方便管理考虑，将博客系统的功能总结起来，共需要实现以下几个方面功能：

①管理所有用户信息

管理所有用户的基本信息，对用户信息的操作包括添加、修改、删除等；比如修改，通过对需要修改项进行修改并保存修改后的结果存入数据库的用户表中。

②管理文章信息

管理所有文章的基本信息，包括对文章信息的添加、修改、删除等操作；对相应的信息进行操作。比如修改和删除，对相应的文章分类进行修改、删除操作。

③管理留言信息

根据前端提交过来的留言信息，可以在后台的管理员界面进行留言信息的管理。

通过以上功能的设计与实现，并实现博客管理系统的基本功能。当然在实际的博客系统中，其功能要比本设计的多的多，也复杂的多，本设计说明仅以比较简单易懂的方式介绍给读者一个博客管理系统的设计与实现的流程，，从而让读者明白一个项目开发中需要做的工作，并在今后的工作中能遇到相关问题能知道如何解决。

## 3.2.1 结构设计

根据对系统进行需求分析，本系统将分为4个模块：

● 文章管理

管理文章基本信息，包括文章的添加、修改、删除。

● 文章分类管理

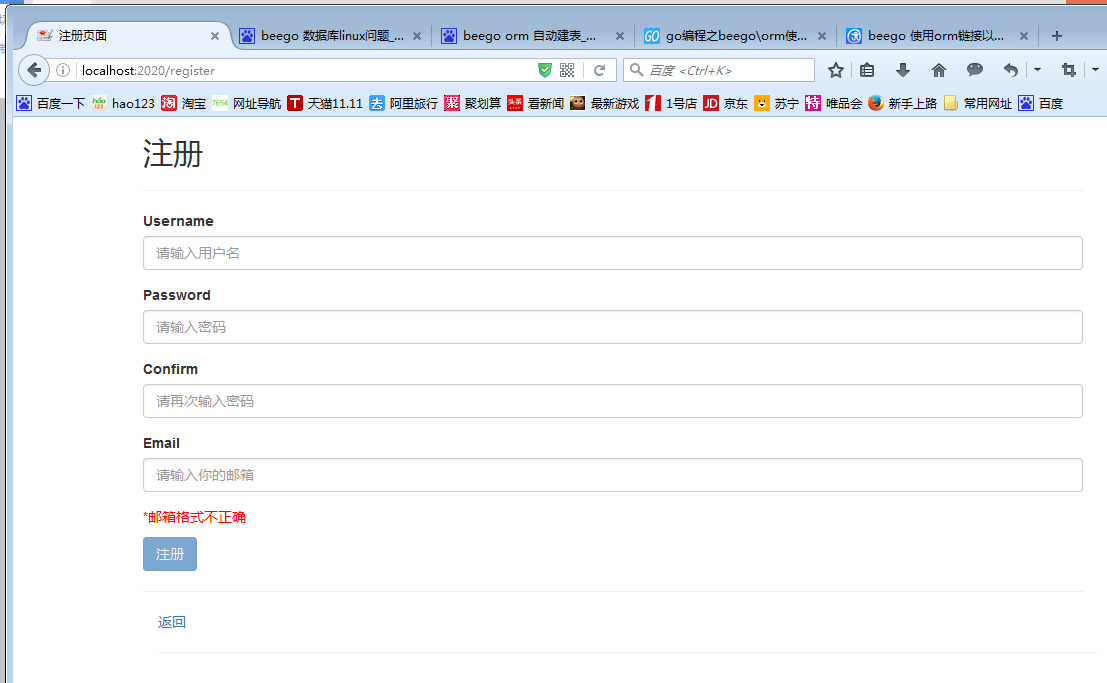
管理文章对应的分类，包括分类的添加、修改和删除。

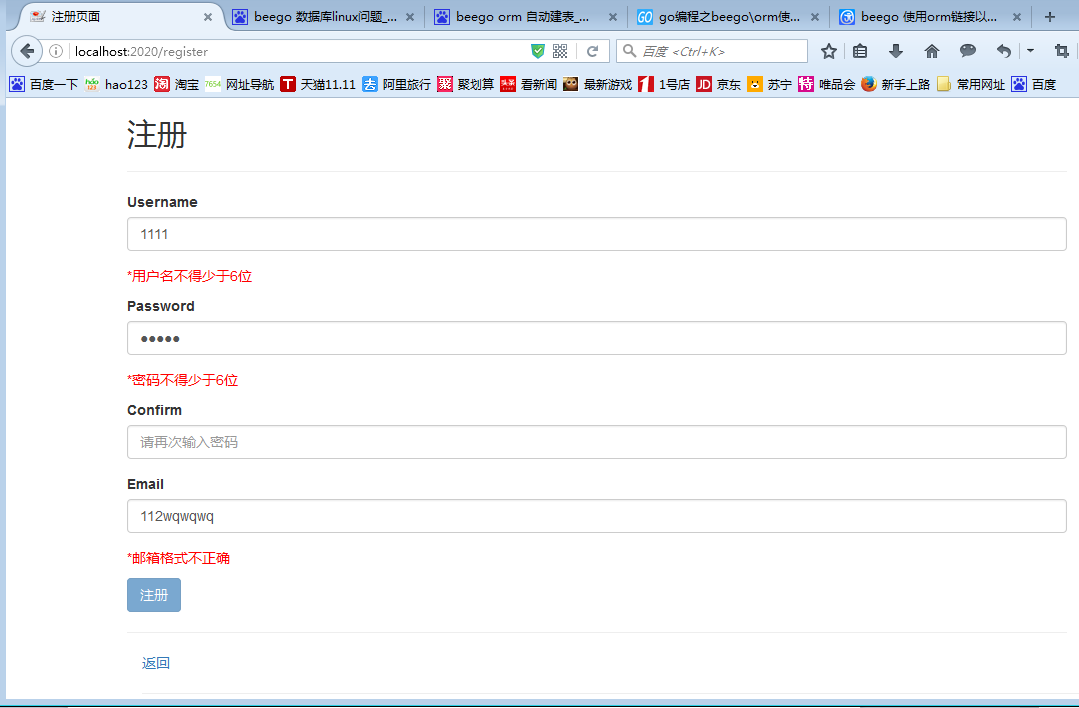
● 留言管理

管理web前端的留言，能够进行留言的删除及添加功能。

# 4.1博客系统部分功能展示及实现原理分析

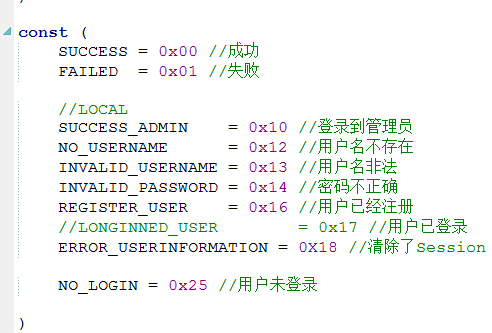
## 4.1.1 注册页面





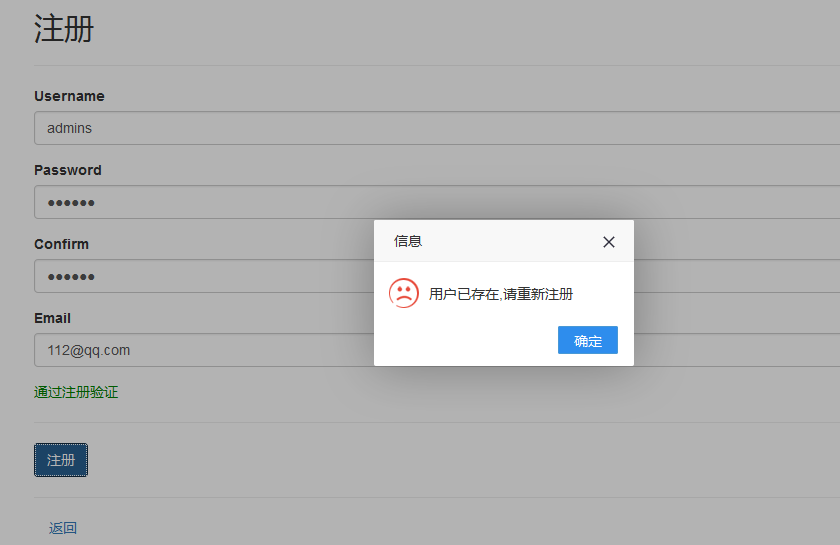


注册基本采取angular.js的表单验证功能，用户名以及密码的注册规则都在6到16个字段，并且两次输入密码要求一致，邮箱也需验证通过注册按钮才会激活出来，在所有表单验证通过后会给予通过提示，由于注册采取的数据交互模式是ajax，所以前端post到后台的数据 如果出现问题，后台将会发送给前端对应的error值，error的定义规则在我之前写好的基类中做好了规则，如下图所示:

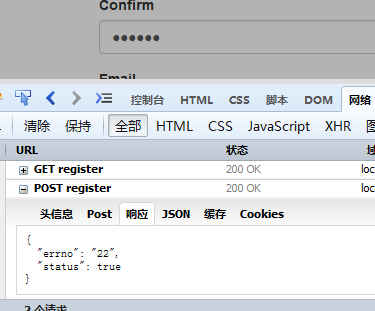


**由于采用的是ajax，所以前端约束的规则便可更加丰富，如已经注册过的用户无法继续注册，如下图所示**

此时点击注册，由于之前数据库中我已注册过admins这个用户，所以用户将无法继续注册，因此系统将会给予对应的窗口提示,具体如下图:

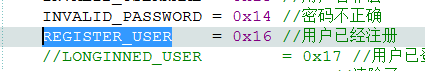


使用firebug调试可以看到响应的json对应的error值

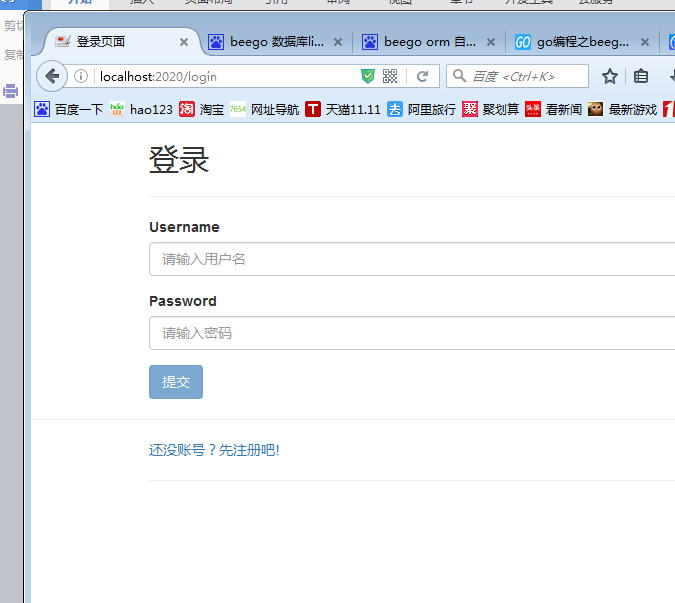


可以看到errno为22进行进制转换后便是对应的0x16 REGISTER\_USER

用户已经注册！



## 4.1.2 登录页面



同样的登录页面也和注册页面一样 采取的一样是ajax的方式，后端控制器中已经写好了对应模板中的方法，下面具体来看下前端对应的具体的angular.js代码:

<script src="/static/js/jquery.min.js"></script>

<script src="/static/js/aj.js"></script>

<script src="/static/js/layer-2.4/layer.js"></script>

<script type="text/javascript">

var app=angular.module('myblog',[])

app.config(['$interpolateProvider', function($interpolateProvider) {

$interpolateProvider.startSymbol('[[');

$interpolateProvider.endSymbol(']]');

}]);

app.controller('addtopic', function($scope) {

$scope.alert=function(){

layer.open({

type:2,

title: '添加文章',

shadeClose: true,

shade: 0.5,

area: ['600px', '75%'],

content: 'http://localhost:2020/admin/addtopic' //content的url

});

}

});

app.controller('login',['$scope','$http',function($scope,$http){

$scope.asave=function(){

$http({

method:'POST',

url:'/login',

data:$.param($scope.formData),

headers:{'Content-Type':'application/x-www-form-urlencoded; charset=UTF-8'}

}).success(function (data){

if(data.errno==0x00){

layer.alert("登录成功!",{icon:3});

location.href="/";

}else if(data.errno==0x12){

layer.alert("用户名不存在!",{icon:5});

}else if(data.errno==0x14){

layer.alert("密码不正确!",{icon:5});

}else if(data.errno==0x10){

layer.alert("管理员你好!");

location.href="/admin/index";

}

});

};

}]);

app.controller('register',['$scope','$http',function($scope,$http){

$scope.asave= function(){

$http({

method: 'POST',

url:'/register',

data:$.param($scope.formData), // pass in data as strings

headers: {'Content-Type':'application/x-www-form-urlencoded; charset=UTF-8'}

}).success(function (data) {

if(data.errno==0x00){

layer.msg('注册成功', {

offset: 0,

shift: 6

});

location.href="/login";

}else if(data.errno==0x16){

layer.alert("用户已存在,请重新注册",{icon:5});

}

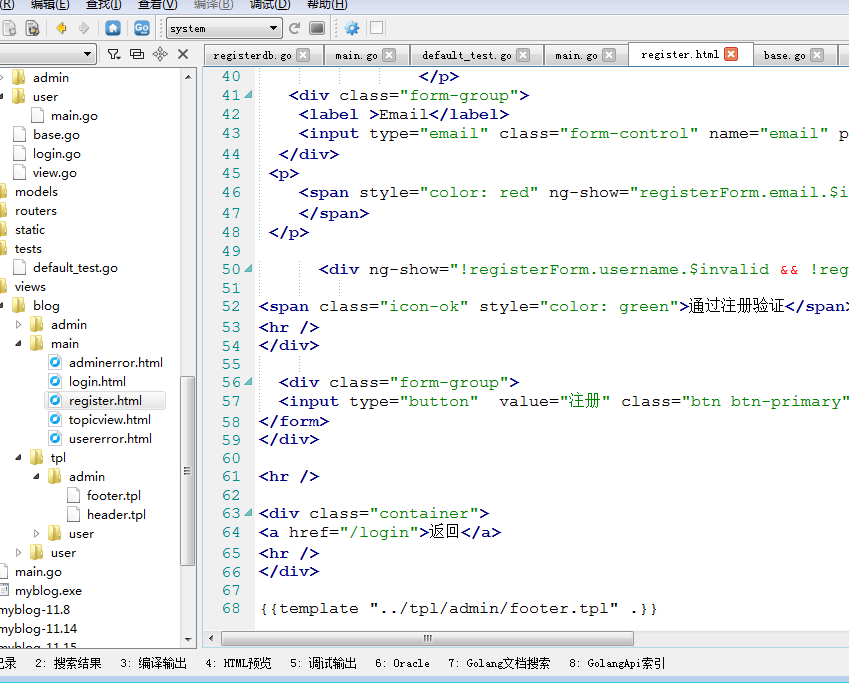
});

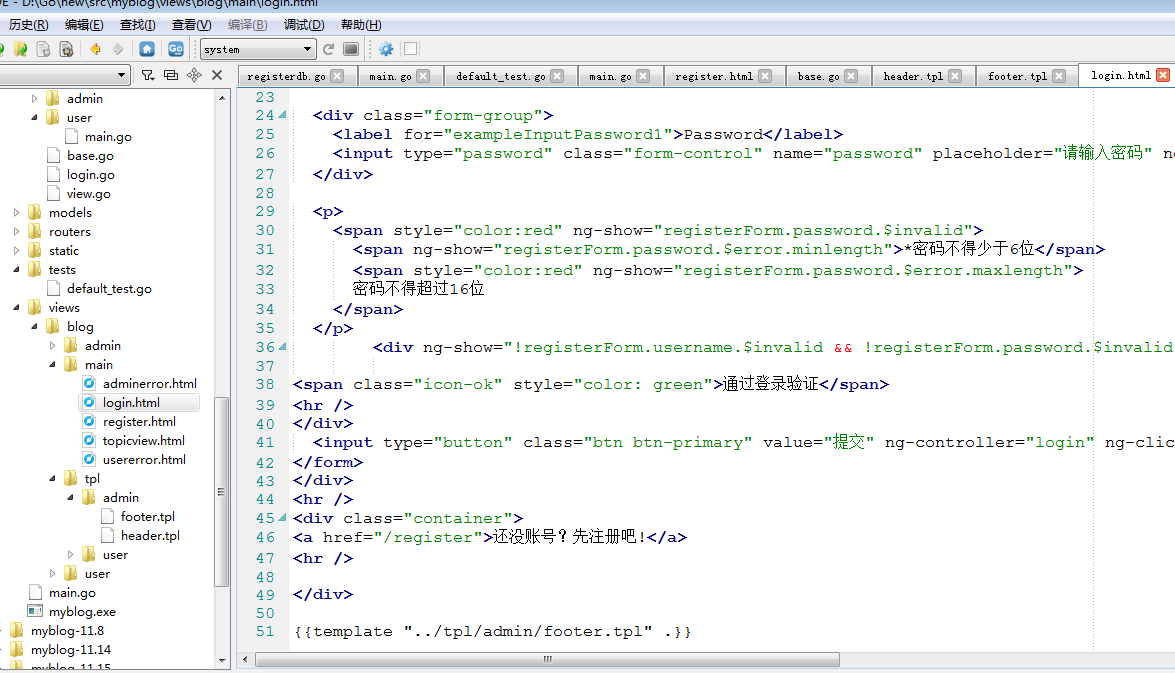
};

}]);

</script>

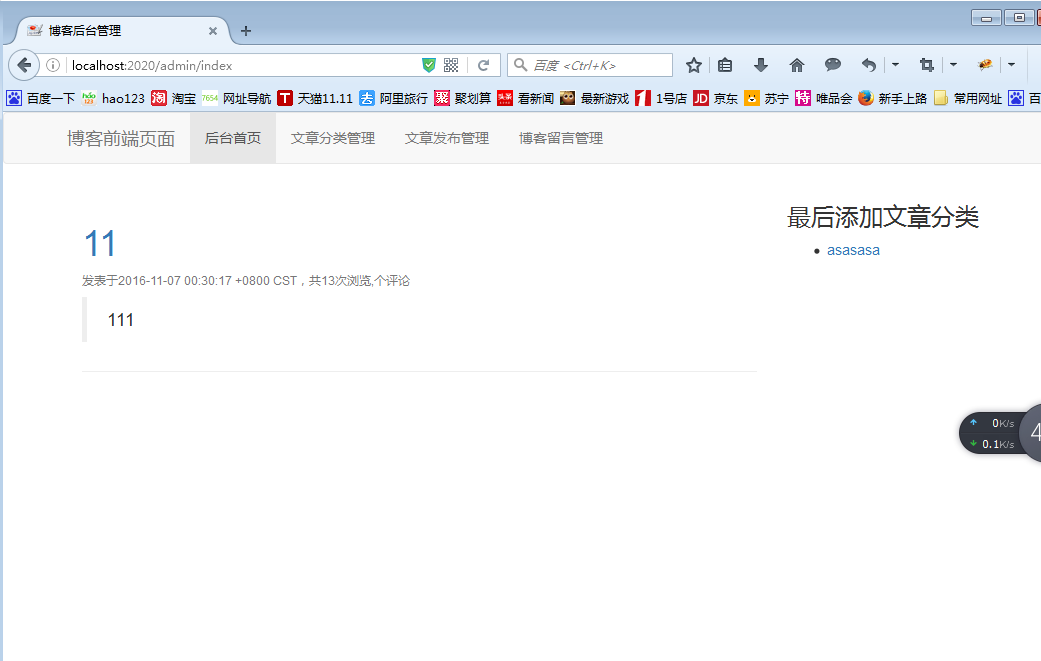
为了程序的耦合性，我采用了头尾分离的设计，网页的头部以及尾部都被我封装了出来，尾部</body>标签后加载js会使页面的加载更加顺畅，因此我们可以一次封装多次使用，如下图所示:





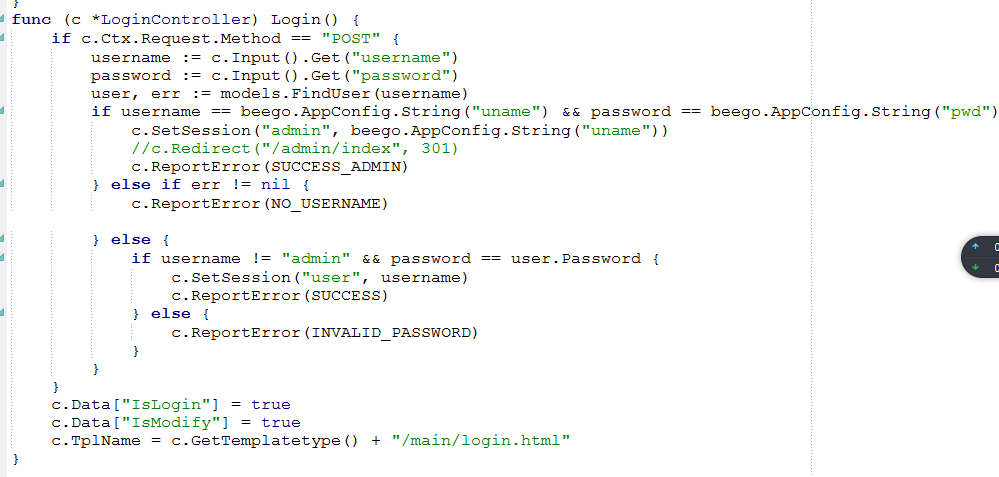
这样，登录和注册页面就共享了同一个尾部的js，降低了代码量的同时是程序更加简洁!

## 4.1.3 登录成功后



以下页面是在用户使用管理员的账户登录进入的页面，只有管理员才能对文章及分类进行管理!用户只能进行浏览及留言,这里可以顺带牵扯下我在登录控制器中写好的逻辑

如下图:



首先抓取到表单中的name属性，如果用户名及密码匹配我们在app.conf中的值，我们将会给予登录到admin的命令并且发送一段json数据给前端，前端通过ajax来接收参数并且通过js来实现对应的页面跳转，前端处理如下图所示:



## 4.1.4 文章分类管理

## **4.1.5 文章发布管理**



## 4.1.6 博客留言管理



# 5.1 概述

详细设计阶段的根本目标是确定应该怎样具体的实现所要求的系统，也就是说，经过这个阶段的设计工作，应该得出目标系统的精确描述，从而在编码阶段可以把这个描述直接翻译成用某种程序设计语言书写的程序。详细设计的目标不仅仅是逻辑上正确地实现每个模块的功能，更重要的是设计的处理过程应该尽可能简明易懂。

6.1 测试的任务及目标

## 6.1.1 测试的任务

在软件投入生产性运行之前，尽可能多地发现软件中的错误。

## 6.1.2 测试的目标

测试的目的是为了发现程序中的错误而执行程序的过程。 好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案。 成功的测试是发现了到今为止尚未发现的错误的测试。

# 6.2 测试方案

设计测试方案是测试阶段的关键技术问题。所谓测试方案包括预定要测试的功能。应该输入的测试数据和预期的结果，目标是设计一组可能发现错误的数据。测试有两种方法：功能测试和结构测试。功能测试，在程序接口进行，只检查程序功能是否能够按照规格说明书的规定正确使用，程序是否能适当地接收输入数据并发生正确的输出信息，而且要能够保持外部信息的完整性。构测试，完全了解程序的结构和处理过程，这种方法按照程序内部的逻辑测试程序，检验程序中每条通路是否都能按照预定要求正确工作。 本系统开发过程进行的测试步骤如下：

6.2.1 模块测试：

也叫单元测试，目的是保证每一个模块作为一个单元能正确运行，本测试所发现的往往是编码和详细设计的错误，主要评价模块的下述五个特点：

1. 模块接口；
2. 局部数据结构;
3. 重要的执行路径;
4. 出错处理通路;
5. 影响上述各方面特性的边界条件.

模块测试主要由代码审查和软件测试两部分组成。

## 6.2.1 模块测试：

也叫单元测试，目的是保证每一个模块作为一个单元能正确运行，本测试所发现的往往是编码和详细设计的错误，主要评价模块的下述五个特点：

1. 模块接口；

2. 局部数据结构;

3. 重要的执行路径;

4. 出错处理通路;

5. 影响上述各方面特性的边界条件.

模块测试主要由代码审查和软件测试两部分组成。

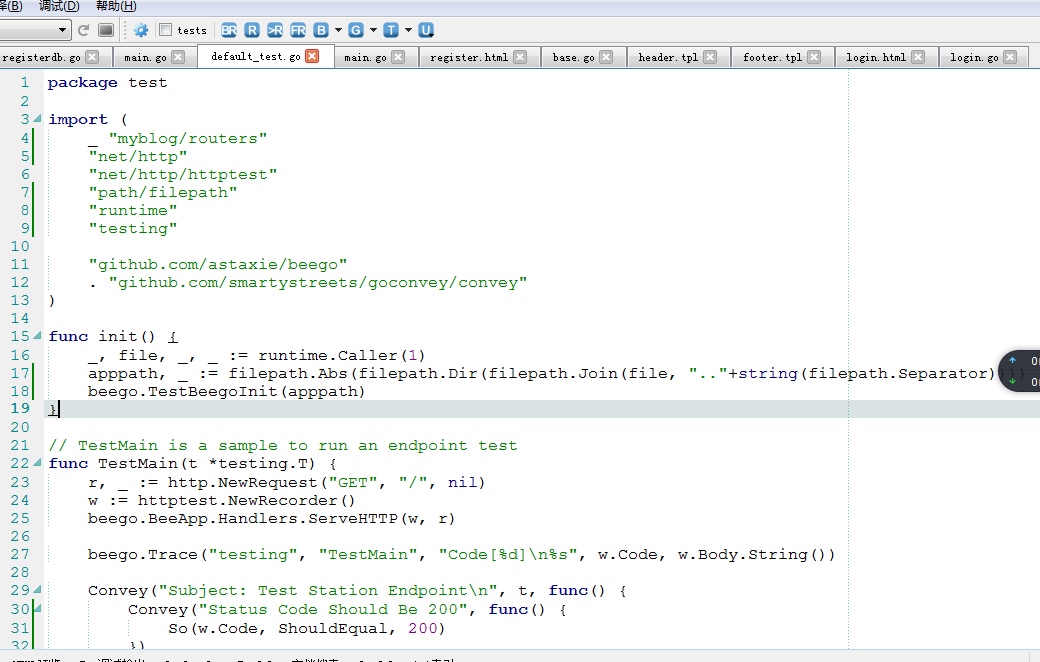
6.2.2 集成测试Goconvey工具包：

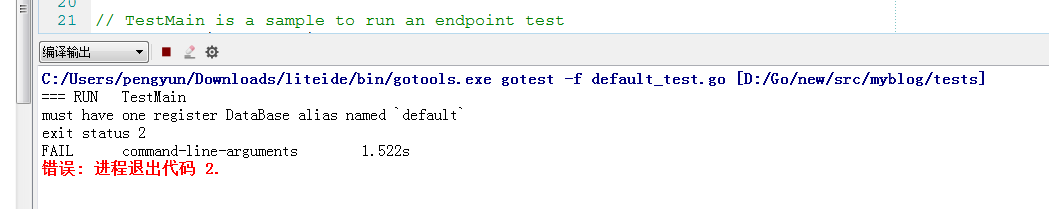
包括系统测试和子系统测试。集成测试是组装软件的系统技术，主要目标是发现与接口有关的问题。集成测试有两种方法：非渐增式和渐增式，但比较而言，渐增式方法比较好，因为：编写的测试软件较少，开销较小。较早地发现模块间的接口错误。错误位置容易判断。测试更彻底。

因此，本系统集成测试采用渐增式测试中的混合法，即对软件结构中较上层使用自顶而下的测试方法，而对软件结构中较下层，使用的是自底向上方法，两者结合，这是对模块较多时测试的一种折衷方法。

发现与接口有关的问题。集成测试有两种方法：非渐增式和渐增式，但比较而言，渐增式方法比较好，因为：编写的测试软件较少，开销较小.较早地发现模块间的接口错误。错误位置容易判断。测试更彻底。

因此，本系统集成测试采用渐增式测试中的混合法，即对软件结构中较上层使用自顶而下的测试方法，而对软件结构中较下层，使用的是自底向上方法，两者结合，这是对模块较多时测试的一种折衷方法。

生成表结构及添加数据测试代码例图：



由于采用了beego的orm所以必须要注册default数据库后才可进行页面操作，当然这也是所有orm框架的一个弊端所在。

## 6.2.3 验收测试

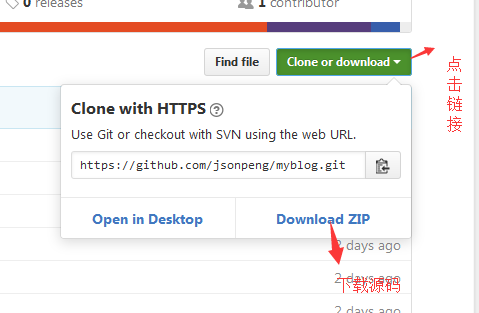
这一步是验证软件的有效性。目的是向未来的用户表明系统能够象预定的那样工作，验收测试一般使用黑盒测试法，验收测试有两种可能的结果。 功能和特性与用户的要求有差距。

功能和性能与用户要求一致，软件是可以接受的。这个阶段发现的问题往往和需求分析阶段的差距有关。

# 7.0 结论总结

通过完成这次毕业设计，使我对Go语言有了更为深刻的了解，也更加熟悉了MVC的设计模式以及基于Go的web编程技巧，对网页如何和数据库交互有了更加明晰的认识

在这次设计过程中，培养了我的自学能力，在遇到技术困难的时候，能够认真的对待，永不放弃，无论是查看各种书籍还是到网上论坛查找资料，想尽一切办法将这个问题解决。我总结了很多WEB的编程经验，这将在以后的工作生活中有很大的帮助，从而也算是积累了我的项目经验,由于本人对开源软件的热爱以及习惯使用github保存我的代码，所以本次设计的所有源码都将在github中展示，本人github地址是 [https://github.com/jsonpeng](http://github.com/jsonpeng) （彭云），博客源码 <https://github.com/jsonpeng/myblog> ，详细如下图:



项目使用方法:双击myblog.exe，会启动webserver本地服务器,使用前提:必须安装配置好mysql,mysql数据库中必须创建好default数据库，从而符合orm的自动建表要求标准。