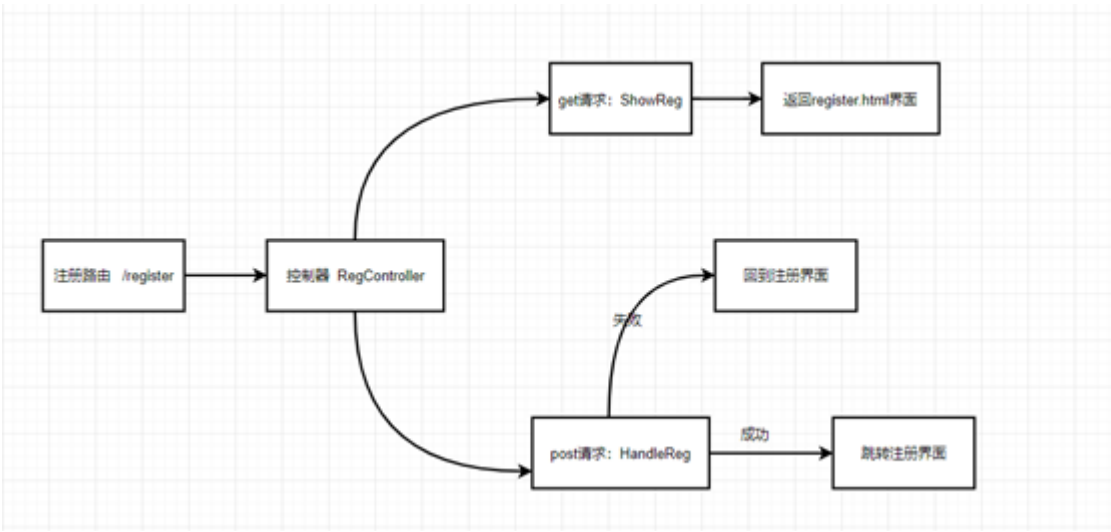
Beego案例-新闻发布系统

1.注册

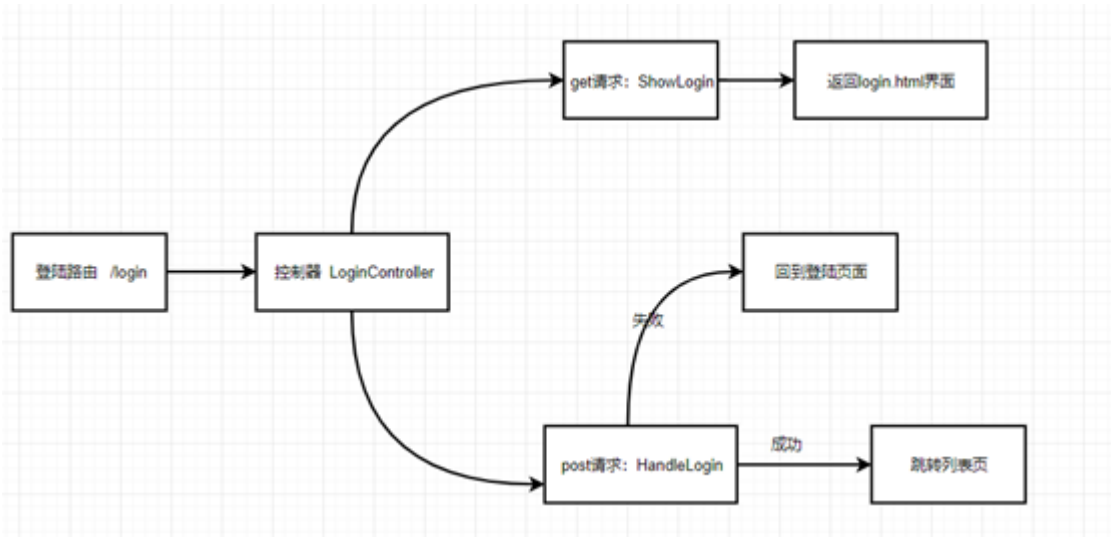
后台代码和昨天案例代码一致。,所以这里面只写一个注册的业务流程图。

业务流程图



2.登陆

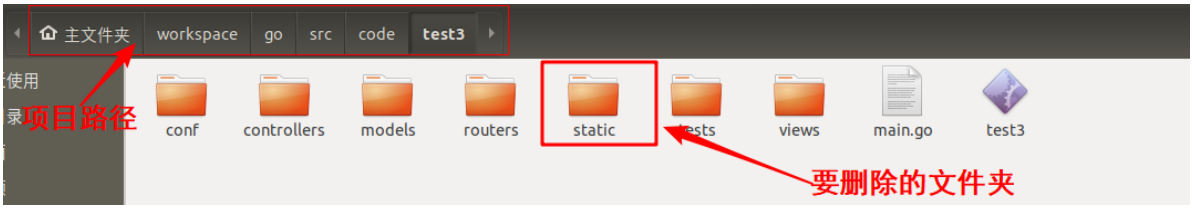
业务流程图



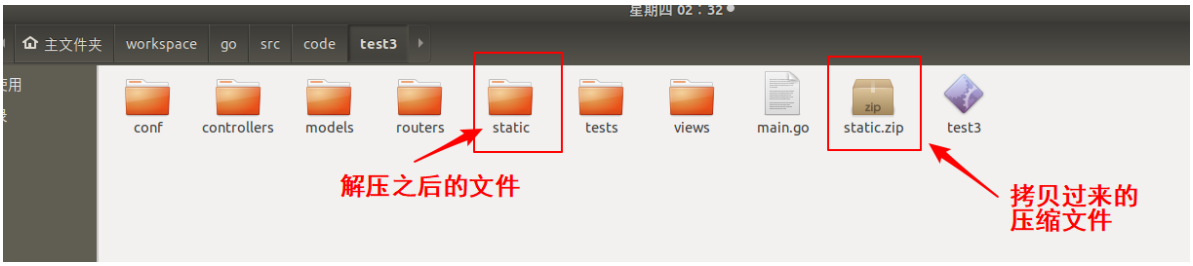
登陆和注册业务和我们昨天登陆和注册基本一样，所以就不再重复写这个代码

但是我们遇到的问题是如何做代码的迁移，把昨天的登陆和注册拿过来直接用？

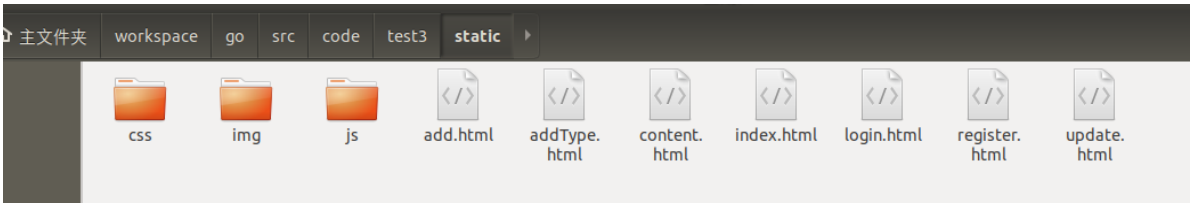
- 首先，我们需要把静态页面拷贝到我们项目目录下面。
    - 进入项目目录，删除掉原来的static文件夹



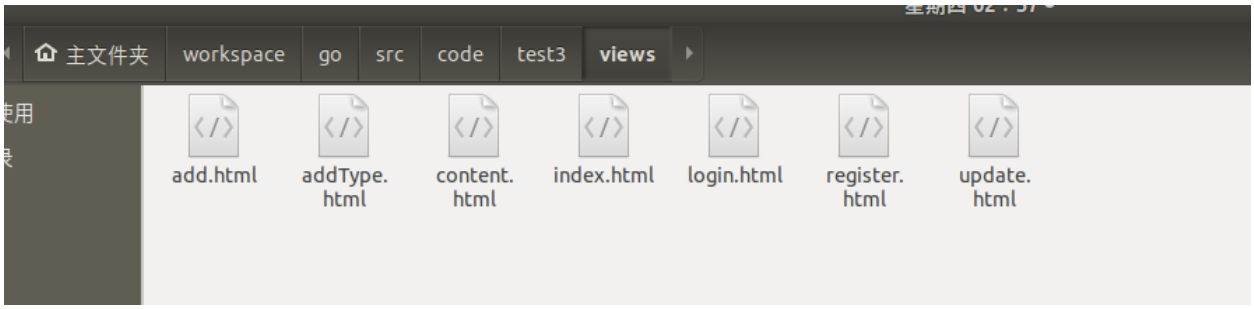
- 然后拷贝我们昨天课堂资料中的 `static.zip` 到这个目录，并解压，解压之后如下图：



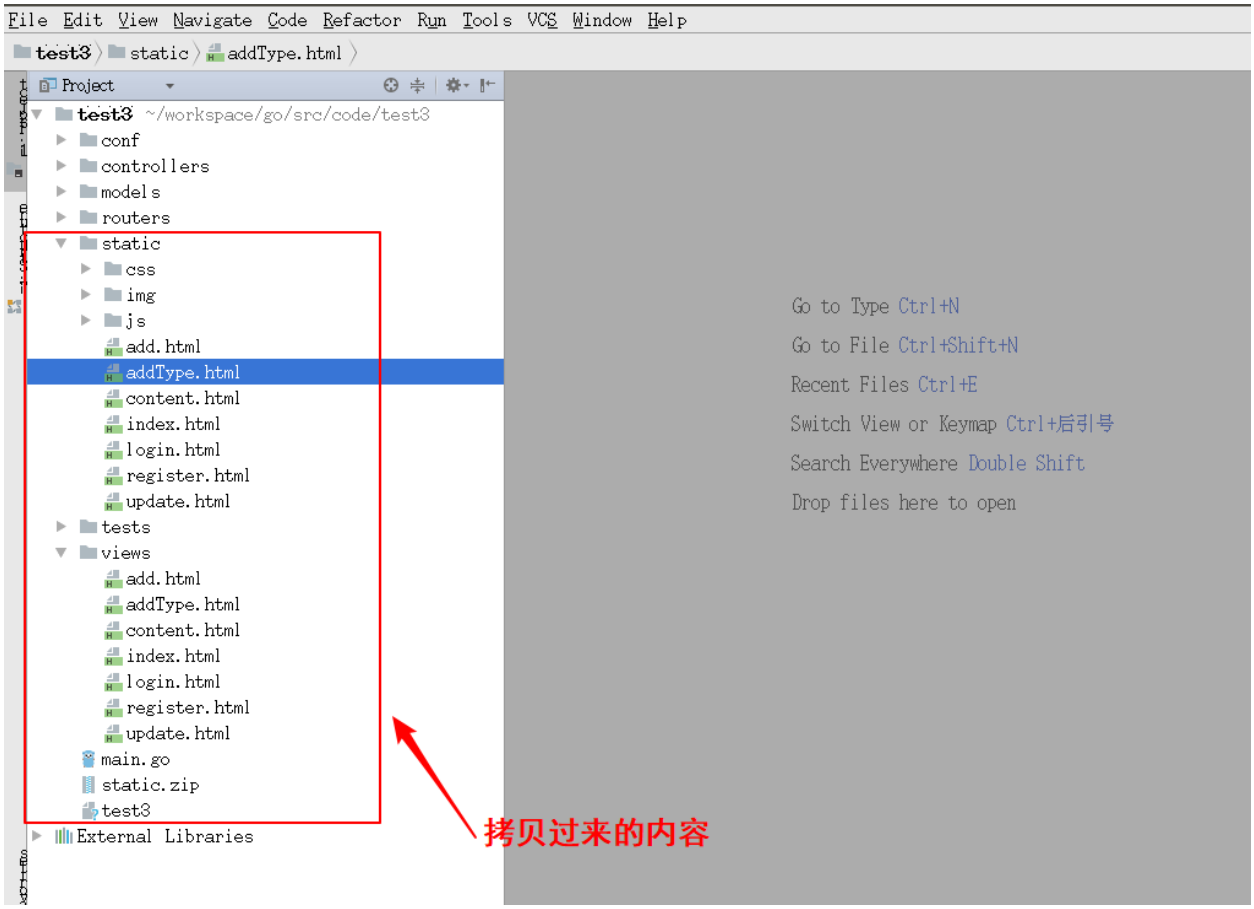
- 打开static文件夹，显示如下，则说明拷贝成功：



- 把static文件夹中所有的html文件都拷贝到views文件夹下面（昨天的几个页面已经没用了，可以删除），拷贝之后views文件显示如下：



这时候用GoLand打开我们的项目,显示如下：



- 打开`register.html`页面，修改页面中form表单的内容
  - 给`<form>`标签加上`action="/register" method="post"`属性
  - 给两个`<input>`标签的name分别改为`name="userName"`和`name = "passwd"`
  - 表单相关代码如下:

```
<form  class="login_form" name = "login" action="/register" method="post">
    <h1 class="login_title">用户注册</h1>
    <input type="text" placeholder="用户名" class="input_txt" name="userName">
    <input type="password" placeholder="密码" class="input_txt" name = "passwd">
    <input type="submit" value="注 册" class="input_sub">
</form>
```

- 打开`login.html`页面，修改form表单的内容
  - 给`<form>`标签加上`action="/login" method="post"`属性
  - 给两个`<input>`标签的name分别改为`name="userName"`和`name = "passwd"`
  - 表单相关代码如下:

```
<form  class="login_form"  name = "login" action="/login" method="post">
    <h1 class="login_title">用户登录</h1>
    <input type="text"  class="input_txt" name = "userName">
    <input type="password" name = "passwd"  class="input_txt">
    <div class="remember"><input type="checkbox" name="remember" >
        <label>记住用户名</label>
    </div>
    <input type="submit" value="登 录" class="input_sub">
</form>
```

（登陆界面多了个记住用户名标签，明天我们实现这个功能）

改完之后，运行项目，测试注册和登陆页面能够能唱显示，并且功能没有问题说明代码迁移成功。

3.创建数据库

3.1数据库表的设计

接下来我们就要实现文章相关的操作，所以这里我们要在数据库中生成一个文章表。

我们以前在数据库中创建表的时候，会给字段加很多限制属性，比如非空，长度，默认值等等，在ORM中，创建表时也可以给各个字段添加相应的限制。那如何去加限制呢？我们先看例子：

```
type Article struct {
    Id int `orm:"pk;auto"`
    ArtiName string `orm:"size(20)"`
    Atime time.Time `orm:"auto_now"`
    Acount int `orm:"default(0);null"`
    Acontent string `orm:"size(500)"`
    Aimg string  `orm:"size(100)"`
}
```

由上面的代码可以看出，要给哪个字段添加属性，需要在这个字段后面添加 `` 括起来的内容，格式为`orm:"限制条件"`。那这些限制条件都有哪些呢？我在这里给大家列了一个表格。

限制条件	作用
pk	设置该字段为主键
auto	这只该字段自增，但是要求该字段必须为整型
default(0)	设置该字段的默认值，需要注意字段类型和默认值类型一致
size(100)	设置该字段长度为100个字节，一般用来设置字符串类型
null	设置该字段允许为空，默认不允许为空
unique	设置该字段全局唯一
digits(12);decimals(4)	设置浮点数位数和精度。比如这个是说，浮点数总共12位，小数位为四位。
auto_now	针对时间类型字段，作用是保存数据的更新时间
auto_now_add	针对时间类型字段,作用是保存数据的添加时间

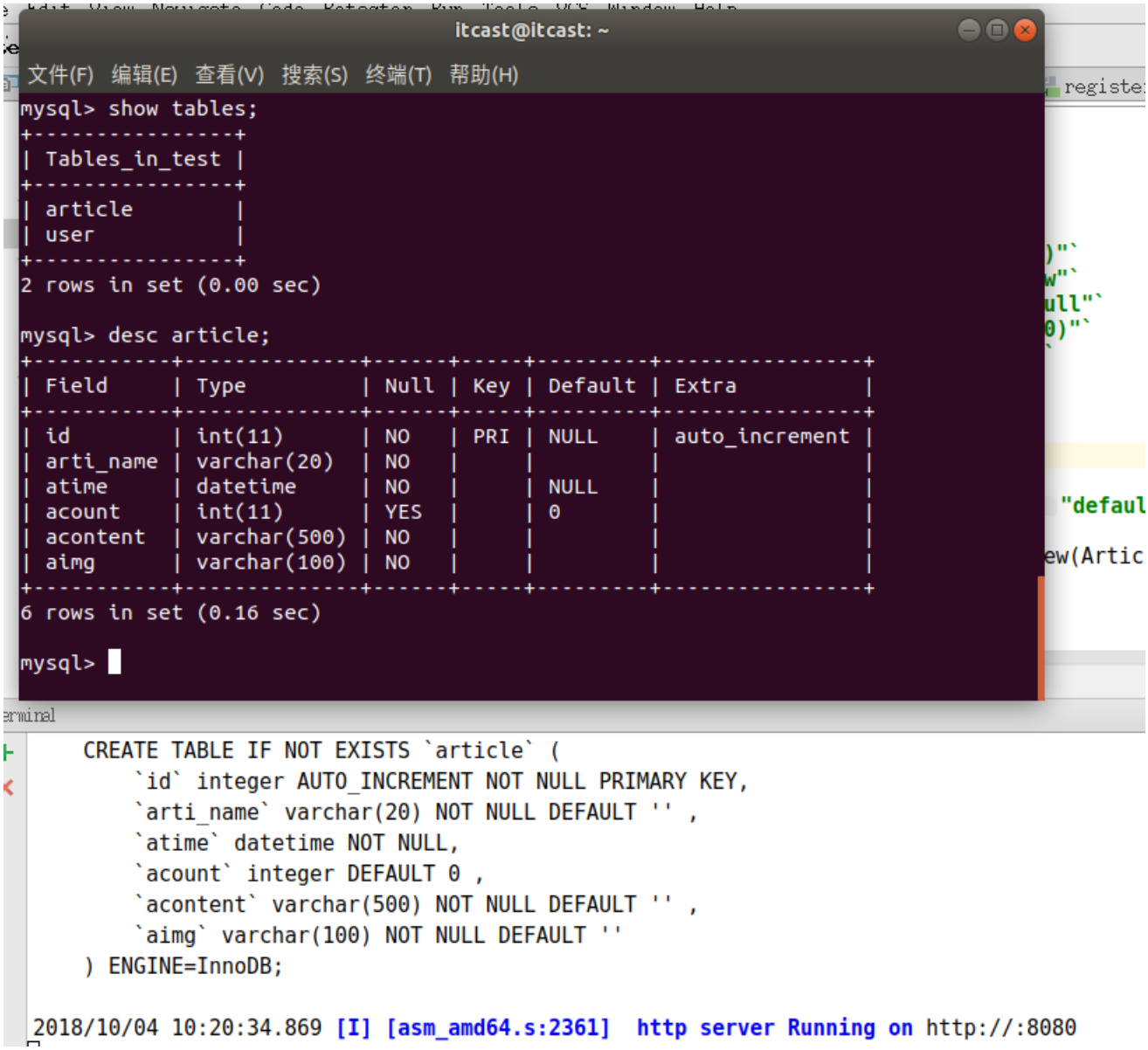
> 注意：**当模型定义里没有主键时，符合int, int32, int64, uint, uint32, uint64 类型且名称为 Id 的 Field 将被视为主键，能够自增.** "
>
> Mysql中时间类型有date和datetime两种类型，但是我们go里面只有time.time一种类型，如果项目里面要求精确的话，就需要指定类型，指定类型用的是type(date)或者type(datetime)

3.2生成表

这时候注意，我们添加了结构体对象之后，并不能直接生成表，需要注册，注册的代码就是初始化数据库三行代码中的第二行，注册表结构，把要创建的表对应的结构体对象作为函数的参数，代码如下：

```
orm.RegisterModel(new(User),new(Article))
```

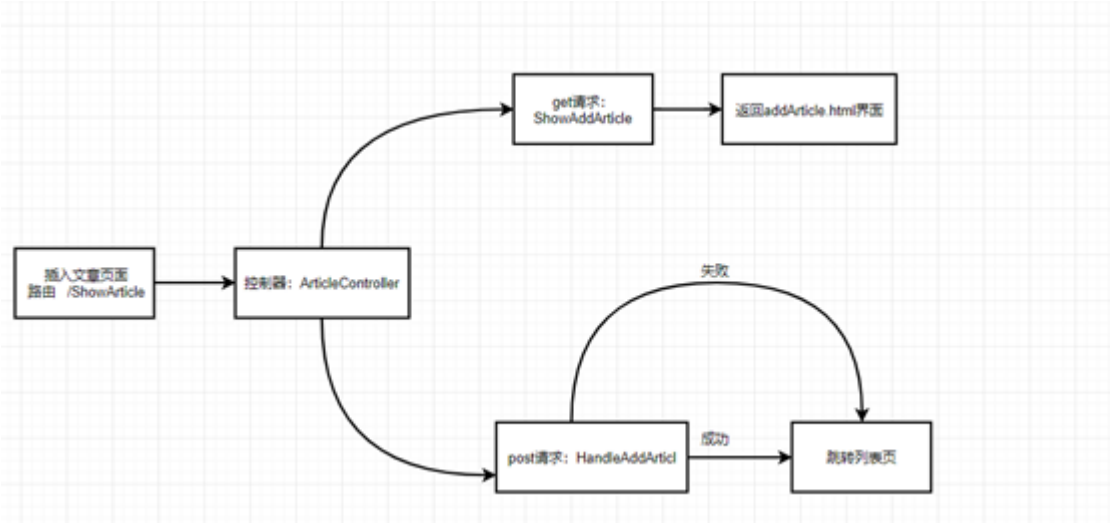
创建之后，我们可以在goland下方查看创建表过程，也可以进入数据库查看是否建表成功，成功的话，数据库显示如下:



登陆成功之后，访问新闻列表展示页面，但是我们现在还没有新闻，所以我们先实现插入文章界面。

4.插入文章

业务流程图



插入页面我们用的视图是`add.html`，这里我们规定添加文章界面的请求路径为`/addArticle`

4.1修改路由文件

在router.go文件的init函数中添加下面这一行代码

```
beego.Router("/addArticle",&controllers.ArticleControlle{},"get:ShowAddArticle")
```

4.2添加文章界面的显示

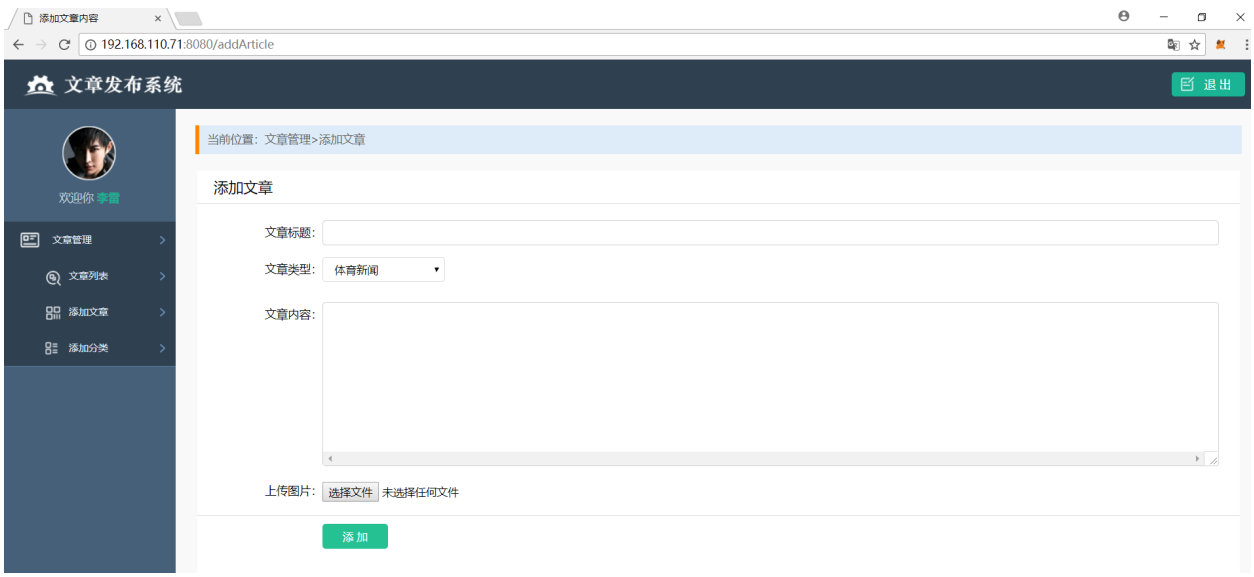
- 先创建一个article.go文件用来存放文章有关的业务代码
- 然后在article.go文件中创建一个ArticleController控制器，并定义一个ShowAddArticle函数代码如下:

```
import "github.com/astaxie/beego"

type ArticleController struct {
    beego.Controller
}

func (this*ArticleController)ShowAddArticle(){

}
```

- 接着我们来实现`ShowAddArticle`函数，这个函数只是用来展示页面的，所以我们只需要给他制定一个视图就可以，代码如下:

```
func (this*ArticleController)ShowAddArticle(){
    this.TplName = "add.html"
}
```

写完代码之后，我们从浏览器发出一个请求`http://192.168.110.71:8080/addArticle`，如果能在浏览器中看到下面这个界面，表示页面展示成功:



4.3插入文章数据处理

上面我们显示了添加文章界面，观察界面可以发现，我们需要获取**文章标题**，**文章类型**, **文章内容**，**上传图片**。其中文章类型牵涉到多表操作，我们放到明天来讲，今天只讲简单的单表操作。首先让我们来看一下，插入页面的前端部分修改。

4.3.1前端页面修改

由页面可知，我们这里面是要上传数据，所以我们这里需要一个form表单，打开前端界面`add.html`，能看到我们这里面确实有一个标签，只是没有属性，我们需要给标签添加action和method属性，这个请求还是添加文章，所以我们还可以用添加文章的请求路径，设置action属性`action="/addArticle"`。因为上传数据，我们这里用post方法，设置method属性`method="post"`。其他部分不用修改。form修改代码如下：

```
<form method="post" action="/addArticle">
```

4.3.2路由内容修改

我们在前端添加了addArticle请求的post方法，所以需要修改一下router.go，给addArticle的post请求指定一个函数，修改代码如下:

```
beego.Router("/addArticle",&controllers.ArticleController{},"get:ShowAddArticle;post:HandleAddArticle")
```

4.3.3后台代码实现

有了函数名之后，我们就需要在后台中实现这个函数。

- 首先是获取数据

  这时候我们看一下前端界面，我们需要获取**文章标题**，**文章内容**，**上传图片**数据，文章标题和文章内容都是字符串，比较简单，直接通过GetString获取，所以我们先获取这两个内容。通过查看add.html代码我们发现，文章标题对应的 标签name等于articleName，文章内容对应的 标签name等于content（注意这里用