

## 使用说明

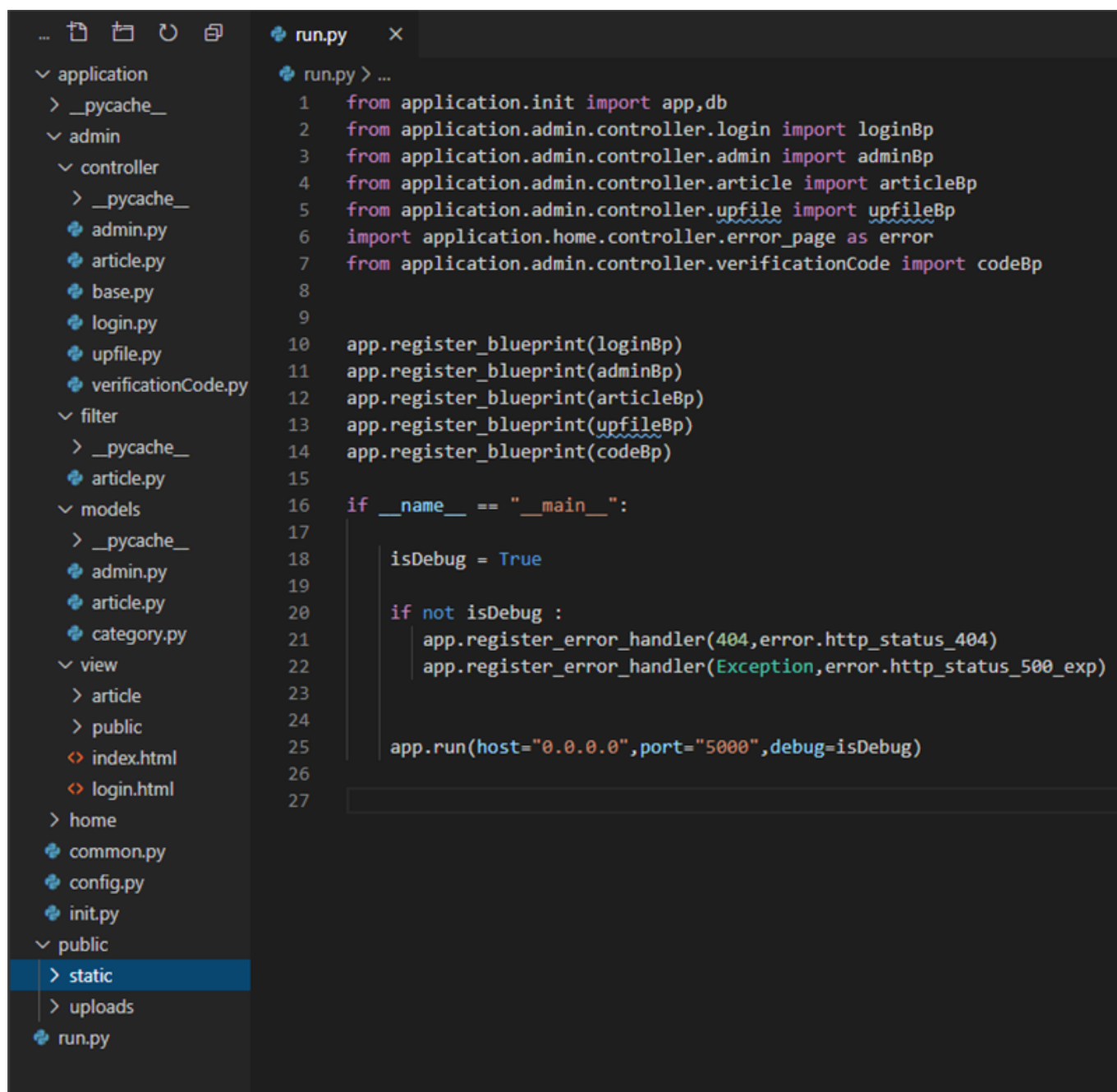
使用Flask制作的普通企业网站前台+后台 包括管理员登录（主要是后台文章发布的功能的实现），注意：使用对象是 flask的初学者，说明文档有详细的教程，包括包的引用

，程序目录如何安排关键的文件如何存放等，在此基础上可以 轻松快速构建自己想要的项目，目录布局文件存放主要参考了ThinkPHP5的思路。实例中有许多bug未处理，不适合商用,如果很介意，请勿下载!

## 运行环境

Python版本 3.8.2 + flask 1.1.2 + Flask-SQLAlchemy2.4.1 + python第三方库 + xenon admin（管理后台界面）

### 1-1.项目结构



The screenshot displays a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the project structure, including folders like application, admin, filter, models, view, and public, and files like \_\_pycache\_\_, admin.py, article.py, base.py, login.py, upfile.py, verificationCode.py, category.py, common.py, config.py, init.py, static, and uploads. The code editor shows the run.py file, which contains the following code:

```
1 from application.init import app,db
2 from application.admin.controller.login import loginBp
3 from application.admin.controller.admin import adminBp
4 from application.admin.controller.article import articleBp
5 from application.admin.controller.upfile import upfileBp
6 import application.home.controller.error_page as error
7 from application.admin.controller.verificationCode import codeBp
8
9
10 app.register_blueprint(loginBp)
11 app.register_blueprint(adminBp)
12 app.register_blueprint(articleBp)
13 app.register_blueprint(upfileBp)
14 app.register_blueprint(codeBp)
15
16 if __name__ == "__main__":
17     isDebug = True
18
19     if not isDebug :
20         app.register_error_handler(404,error.http_status_404)
21         app.register_error_handler(Exception,error.http_status_500_exp)
22
23
24
25 app.run(host="0.0.0.0",port="5000",debug=isDebug)
26
27
```

站点目录mysite下有application和public文件夹及run.py文件

1.application : 项目主程序存在目录

2.public :

static文件夹包含前端需要引用的css,js,图片,

uploads文件夹存放后台上传的文件及图片，以日期命名的文件夹

3.run.py 主程序入口：

## 1-2.application文件夹项目结构

1.admin : 网站管理后台

2.home : 网站前台，页面展示

3.init.py : 初始化 flask 和 flask\_sqlalchemy，定义 template\_folder 模板文件夹位置 和 定义 static\_folder 静态资源引用位置

4.config.py : 配置系统常量，包括数据库的连接、session的参数配置以及 自定义常量配置

5.common.py : 定义常用的处理方法函数，如 def md5(str)，供 系统其它文件调用

## 1-3.admin文件夹项目结构

1.controller : 蓝图控制器,每个控制器都以相关的功能模块来命名，如articleBp,然后在run.py入口文件注册才可以使用

2.filter : 自定义过滤器,可以在视图view的模板页面html中使用

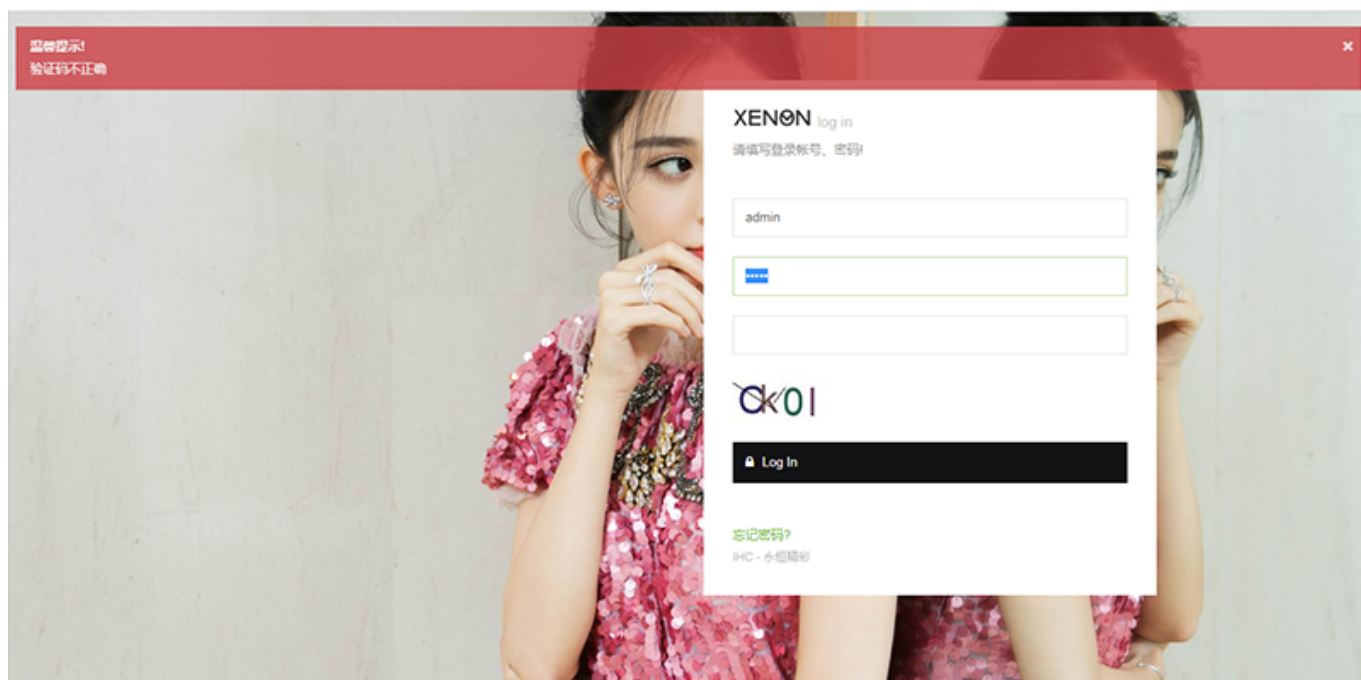
3.models : 定义表的模型，flask\_sqlalchemy 通过 models操作数据库的CURD

4.view : 模板页面，需要在 controller控制器中指定才能使用，展示数据

## 1-4.home文件夹项目结构

1.目录结构与admin类似，不同的地方在于多了一个common文件夹，里面存放404,500错误状态的页面,在run.py入口文件注册使用,该控制器存在在home\controller文件夹下

## 2-1.程序讲解(登录页面:http://localhost:5000/login/)



知识点：

- 1.动态验证码的插件使用：
- 2.AJAX数据提交：
- 3.后台登录验证：

### 动态验证码的插件使用

动态验证码插件使用到的文件是application/admin/controller/verificationCode.py,蓝图名称为codeBp，定义路由：@codeBp.route("/imgCode")，访问的路径：http://localhost:5000/code/imgCode,在入口文件run.py中注册使用：app.register\_blueprint(codeBp).注意：字体的引用位置一定要正确，否则无法使用：font = ImageFont.truetype(app.static\_folder + r"/public/static/font/Arial.ttf", 40)

### AJAX数据提交

登录访问：http://localhost:5000/login/，用户名称

密码或验证码不正确时均会在本页面顶部状态栏有相关提示,如上图所示.登录的用户名称密码均设置为admin，方便测试.

在登录页面模板文件mysite\application\admin\view\login.html里验证码的引用:

### 后台登录验证：

前端ajax提交登录信息：

```
$.ajax({
  url: "{{url_for('login.check')}}",
  method: 'POST',
  dataType: 'json',
  data: {
    do_login: true,
    username: $(form).find('#username').val(),
    password: $(form).find('#password').val(),
    captcha : $(form).find('#captcha').val(),
  },
  success: function(resp)
  {
    show_loading_bar({
      delay: .5,
      pct: 100,
      finish: function(){
        // Redirect after successful login page (when progress bar reaches 100%)
        if(resp.status){
          window.location.href = "{{url_for('admin.index')}}";
        }
        else
        {
          $("#mycaptcha").attr("src","{{url_for('code.imgCode')}}?rand="+Math.random());
          $("#captcha").val("");
          toastr.error(resp.msg, "温馨提示!", opts);
          $("#password").select();
        }
      }
    });
  }
});
```

后台登录验证：

```
loginBp = flask.Blueprint(name="login",import_name=__name__,url_prefix="/login")

@loginBp.route("/",methods=["GET"])
def index():
    return render_template("admin/view/login.html")

@loginBp.route("/check",methods=["POST"])
def check():
    if request.method == "POST":
        username = request.form.get("username")
        password = request.form.get("password")
        captcha = request.form.get("captcha").lower()

        admin = Admin.query.filter(Admin.username==username).first()

        data = {}

        if captcha==session['imageCode'].lower(): ...
        else: ...

        if admin is None: ...

        if admin.password.strip() != md5(password.strip()) : ...

        if admin.islock :
            data["status"] = 0
            data["msg"] = "该帐号已被锁定 "
            return jsonify(data)

        session["islogin"] = True
        session["adminid"] = admin.id
        session["username"] = admin.username
        session["img"] = admin.img
        session["issuper"] = admin.issuper

        data["status"] = 1
        data["msg"] = "验证通过 "
        return jsonify(data)
```

mysite\application\admin\controller\login.py为控制器(蓝图),蓝图名称: loginBp, 在run.py入口文件注册使用, 访问路径:http://localhost:5000/login/,模板文件为mysite\application\admin\view\login.html。

通过login.html的ajax提交登录信息到http://localhost:5000/login/check/, 通过控制器login.py与数据库信息对比实现用户认证,认证的时候用到了mysite\application\common.py里面自定义的md5函数。

认证过程中使用到了mysite\application\admin\models\admin.py文件, 里面定义了表ihc\_site\_admin的模型,通过flask\_sqlalchemy操作该模块实现数据的CURD基本操作,使用方法如: admin = Admin.query.filter(Admin.username==username).first()

信息无误登录成功, 认证信息存放在session中, 方便其它模块使用认证信息,如 session["islogin"] = True 。

不管登录成功与否, 都会返回json数据告诉前端登录页面该如何处理事件:data["status"] = 1 ,data["msg"] = "验证通过 ",return jsonify(data)。

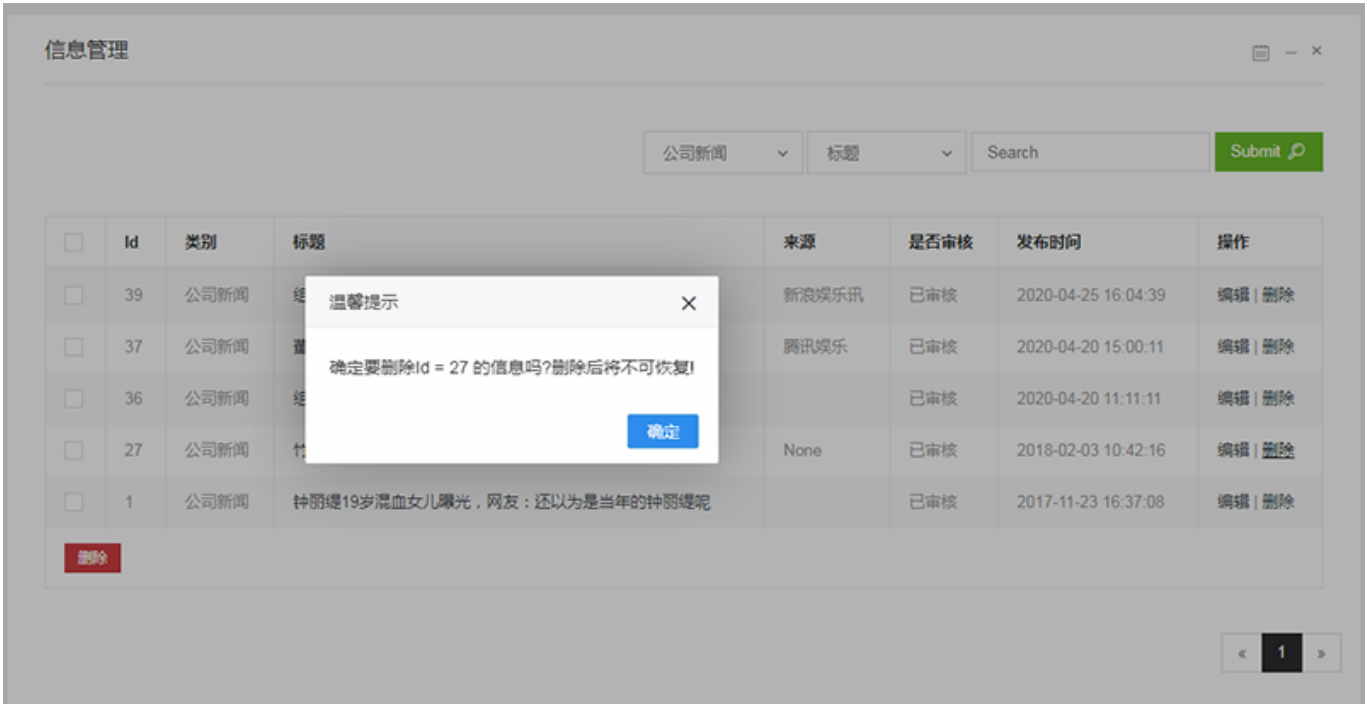
login\_require : 在后台其它模块如文章发布模块在操作前都要加上该装饰器, 判断是否已经登录, 是否有权进行下一步的操作,如:在文章发布页面:@articleBp.route("/add",methods=["GET"]),@login\_require,def add():

### 3-1.文章发布(文章列表:hhttp://localhost:5000/admin/article/lists/ca

## tegoryid/6)

效果图如下

文章列表:



新增文章:

信息发布

基本信息

图片&附件

详细介绍

属性设置

所属分类

公司新闻

标题

组图：刘诗诗穿粉色上衣元气又减龄 对镜甜笑少女感满满

#ff0000

☒ 粗体 ☒ 斜体 ☐ 删除线

副标题

刘诗诗身穿粉色上衣，对镜甜笑宛若少女

关键字

刘诗诗

内容简介

新浪娱乐讯 近日，刘诗诗在微博分享了一组照片，照片中刘诗诗身穿粉色上衣，皮肤保养得很好，对镜甜笑宛若少女

作者

未知

来源

新浪娱乐讯

保存信息

信息发布

基本信息

图片&附件

详细介绍

属性设置

多图上传

插入图片

本地上传

在线管理


图片搜索


图片浮动方式：


共2张（155.96K），2张上传成功

继续添加

开始上传







确认

取消

选择图片

选择附件

© 2018 - 2024 IHC 更多精彩案例 恒彩网络



The image shows a web editor interface. At the top, there are four tabs: '基本信息' (Basic Information), '图片&附件' (Image & Attachment), '详细介绍' (Detailed Introduction), and '属性设置' (Attribute Settings). Below the tabs is a rich text editor toolbar with various icons for text formatting (bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, etc.), alignment, and other functions. The toolbar also includes a font size dropdown set to '14px' and a language dropdown set to '代码语言' (Code Language). Below the toolbar, there is a text input area containing the sentence: '新浪娱乐讯 近日,刘诗诗在微博分享了一组照片,照片中刘诗诗身穿粉色上衣,皮肤保养得很好,对镜甜美宛若少女。' (Sina Entertainment News: Recently, Liu Shishi shared a group of photos on Weibo. In the photos, Liu Shishi is wearing a pink top, her skin is well-maintained, and she looks sweet and like a young girl in front of the mirror.) Below the text is a large image placeholder showing a close-up photo of a young woman with long brown hair and bangs, wearing a pink top, smiling and holding her hand to her face. The photo is decorated with white heart-shaped bokeh effects.



信息发布

基本信息

图片&附件

详细介绍

属性设置

定时发布

2020-04-30

2020-05-09

审核

推荐

置顶

保存信息

© 2018 - 2024 IHC 更多精彩案例 恒彩网络

信息管理

公司新闻

标题

Search

Submit

温馨提示: 文章【组图：刘诗诗穿粉色上衣元气又减龄 对镜甜笑少女感满满】保存成功!

	Id	类别	标题	来源	是否审核	发布时间	操作
<input type="checkbox"/>	39	公司新闻	组图：刘诗诗穿粉色上衣元气又减龄 对镜甜笑少女感满满	新浪娱乐讯	已审核	2020-04-25 16:04:39	编辑   删除
<input type="checkbox"/>	37	公司新闻	董璇带女儿骑车，小雀斑穿牛仔装又甜又飒	腾讯娱乐	已审核	2020-04-20 15:00:11	编辑   删除
<input type="checkbox"/>	36	公司新闻	组图：Hebe与Ella相约爬山心情大好 渔夫帽登山鞋装备齐全		已审核	2020-04-20 11:11:11	编辑   删除
<input type="checkbox"/>	27	公司新闻	竹山湖景区60栋木屋开始建设	None	已审核	2018-02-03 10:42:16	编辑   删除
<input type="checkbox"/>	1	公司新闻	钟丽缇19岁混血女儿曝光，网友：还以为是当年的钟丽缇呢		已审核	2017-11-23 16:37:08	编辑   删除

删除

« 1 »

文章已经实现新增，编辑，删除，批量删除，多条件搜索，分页的功能，控制器，模型，视图均与登录类似，不再解析，重点讲解编辑器的使用

### 4-1.编辑器的使用

资源位置：mysite\public\static\ueditor

文件位置：mysite\application\admin\view\public\editor.html

文件引用：mysite\application\admin\view\article\add.html中引用：{% include

"admin/view/public/editor.html" %}

编辑器控制器：mysite\application\admin\controller\upfile.py

编辑器上传类：mysite\public\static\ueditor\python\uploader.py

编辑器配置文件：mysite\public\static\ueditor\python\config.json

须要注意的地方:mysite\application\admin\controller\upfile.py文件中21行的位置，config.json配置文件路径一定要正确，否则无法使用文件上传功能。

mysite\application\admin\view\public\editor.html 10行 中的 serverUrl 与  
mysite\public\static\ueditor\ueditor.config.js中36行中的serverUrl均设置为 /admin/upfile/  
才可以正常使用上传功能。

上传的路径默认为：mysite\public\uploads，可以在mysite\public\static\ueditor\python\config.json 12  
行中imagePathFormat修改上传路径