**利用flask-wtf验证上传的文件：**

**关键点：**

1. 定义验证表单类的时候，对文件类型的字段，需要采用`FileField`这个类型，即wtforms.FileField。

2. 验证器需要从`flask\_wtf.file`中导入。`flask\_wtf.file.FileRequired`和`flask\_wtf.file.FileAllowed`

3.`flask\_wtf.file.FileRequired`是用来验证文件上传不能为空。

4.`flask\_wtf.file.FileAllowed`用来验证上传的文件的后缀名， 如常见图片后缀 .jpg 和.png以及.gif等。

5. 在视图函数中，需要使用`from werkzeug.datastructures import CombinedMultiDict`来把`request.form`与`request.files`来进行合并。

6.最后使用 表单验证对象.validate()进行验证。

**代码如下：**

upload.html页面

|  |
| --- |
| <!DOCTYPE **html**>  <**html lang="en"**>  <**head**>  <**meta charset="UTF-8"**>  <**title**>上传文件</**title**>  </**head**>  <**body**>  <**form action="" method="post" enctype="multipart/form-data"**>  <**table**>  <**tr**>  <**td**>头像：</**td**>  <**td**><**input type="file" name="pichead"**></**td**>  </**tr**>  <**tr**>  <**td**>描述：</**td**>  <**td**><**input type="text" name="desc"**></**td**>  </**tr**>  <**tr**>  <**td**></**td**>  <**td**><**input type="submit" value="提交"**></**td**>  </**tr**>  </**table**>  </**form**>  </**body**>  </**html**> |

formscheck.py文件

|  |
| --- |
| **from** wtforms **import** Form,FileField,StringField  **from** wtforms.validators **import** InputRequired  *# flask\_wtf*  **from** flask\_wtf.file **import** FileRequired,FileAllowed  **class** UploadForm(Form):  pichead = FileField(validators=[FileRequired(),FileAllowed([**'jpg'**,**'png'**,**'gif'**])])  desc = StringField(validators=[InputRequired()]) |

app.py文件

|  |
| --- |
| **from** flask **import** Flask,request,render\_template  **import** os  **from** werkzeug.utils **import** secure\_filename  **from** formscheck **import** UploadForm  **from** werkzeug.datastructures **import** CombinedMultiDict  app = Flask(\_\_name\_\_)  UPLOAD\_PATH = os.path.join(os.path.dirname(\_\_file\_\_),**'images'**)  *#利用flask-wtf验证上传的文件*  @app.route(**'/upload/'**,methods=[**'GET'**,**'POST'**])  **def** upload():  **if** request.method == **'GET'**:  **return** render\_template(**'upload.html'**)  **else**:  form = UploadForm(CombinedMultiDict([request.form,request.files]))  **if** form.validate():  *# desc = request.form.get("desc")*  *# pichead = request.files.get("pichead")*  desc = form.desc.data  pichead = form.pichead.data  filename = secure\_filename(pichead.filename)  pichead.save(os.path.join(UPLOAD\_PATH,filename))  print(desc)  **return '文件上传成功'**  **else**:  print(form.errors)  **return "文件上传失败"**  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |