

文件上传笔记:

1. 在模版中, form表单中, 需要指定`encotype='multipart/form-data'`才能上传文件。
2. 在后台如果想要获取上传的文件, 那么应该使用`request.files.get('avatar')`来获取。
3. 保存文件之前, 先要使用`werkzeug.utils.secure_filename`来对上传上来的文件名进行一个过滤。这样才能保证不会有安全问题。
4. 获取到上传上来的文件后, 使用`avatar.save(路径)`方法来保存文件。
5. 从服务器上读取文件, 应该定义一个url与视图函数, 来获取指定的文件。在这个视图函数中, 使用`send_from_directory(文件的目录,文件名)`来获取。

示例代码如下:

```
@app.route('/upload/', methods=['GET', 'POST'])
def upload():
    if request.method == 'GET':
        return render_template('upload.html')
    else:
        # 获取描述信息
        desc = request.form.get("desc")
        avatar = request.files.get("avatar")
        filename = secure_filename(avatar.filename)
        avatar.save(os.path.join(UPLOAD_PATH, filename))
        print(desc)
        return '文件上传成功'

@app.route('/images/<filename>/')
def get_image(filename):
    return send_from_directory(UPLOAD_PATH, filename)
```

对上传文件使用表单验证:

1. 定义表单的时候, 对文件的字段, 需要采用`FileField`这个类型。
2. 验证器应该从`flask_wtf.file`中导入。`flask_wtf.file.FileRequired`是用来验证文件上传是否为空。`flask_wtf.file.FileAllowed`用来验证上传的文件的后缀名。
3. 如果表单中既有文件, 也有输入数据, 使用`from werkzeug.datastructures import CombinedMultiDict`来把`request.form`与`request.files`来进行合并。再传给表单来验证。

示例代码如下:

python

```
from werkzeug.datastructures import CombinedMultiDict
form = UploadForm(CombinedMultiDict([request.form,request.files]))
```