

# 本篇文章是对超级无敌服务的用法说明

## 项目解释

本项目是运行在 windows 上的一款可以让局域网用户

- 上传文件（入口由密码保护**注意，这个是假保护，上传文件的链接是固定的，如果用户知道了链接，那么下一次不需要输入密码就可以进入，所以请注意**）
- 下载文件
- 向服务端发送消息
- 管理文件（入口由密码保护**这个是真保护**）

的应用，可搭配 Radmin LAN<sup>1</sup> 使用。

### ⚠ Caution

写代码的环境为 python3.11 的 windows11 中，如无法运行，请试一下这个

## 准备工作

需要在 cmd 或 powershell 中输入：

```
1 pip install flask werkzeug plyer pillow rich termcolor pyperclip
```

## 代码放置方式

以下是该项目的文件放置方式

```
1 消息（此名可改）
2  |   app.py
3  |
4  |─mima
5  |   mima.txt
6  |
7  |─shangchuan
8  |
9  |─templates
10      download.html
11      filemanagement.html
12      index.html
13      indexapp.html
14      xz.html
```

## 详细代码

### 📌 Important

请自行修改 mima.txt，如果复制我的 mima.txt，那么密码就是 114514

密码仅为一行

### 📌 Note

此部分内容可不看

直接跳转到[使用方式](#)即可

## 主程序(app.py)

主程序由python编写

```
1  from flask import Flask, render_template, request, redirect, url_for,
    flash, send_from_directory, render_template_string
2  from werkzeug.utils import secure_filename
3  from plyer import notification
4  from datetime import datetime
5  import os
6  import subprocess
7  import time
8  import re
9  from PIL import ImageGrab
10 from rich.progress import Progress
11 from tkinter import Tk, simpledialog
12 from termcolor import colored
13 import socket
14 import pyperclip
15
16 app = Flask(__name__)
17 app.secret_key = 'your_secret_key'
18
19 # 上传文件存储的目录
20 UPLOAD_FOLDER = 'shangchuan'
21 if not os.path.exists(UPLOAD_FOLDER):
22     os.makedirs(UPLOAD_FOLDER)
23
24 app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
25
26 # 密码文件的路径
27 PASSWORD_FILE = 'mima/mima.txt'
28
29 # 用于记录上传的文件信息
30 upload_records = []
31
32 # 清理非法字符以用于文件名
33 def sanitize_filename(filename):
34     return re.sub(r'[\\"/:*?"<>|]', '', filename)
35
36 # 执行命令后延迟截屏并保存
37 def execute_and_capture(command):
38     subprocess.Popen(f'start cmd /k {command}', shell=True)
39     time.sleep(sj)
40     timestamp = datetime.now().strftime('%Y%m%d_%H%M%S')
41     sanitized_command = sanitize_filename(command)
42     screenshot_filename = f'{sanitized_command}_{timestamp}.png'
43     screenshot_dir = 'shangchuan'
44     os.makedirs(screenshot_dir, exist_ok=True)
45     screenshot_path = os.path.join(screenshot_dir, screenshot_filename)
46     screenshot = ImageGrab.grab()
```

```
47     screenshot.save(screenshot_path)
48
49 # 获取当天日期, 格式为 yyyy-mm-dd
50 def get_today_date():
51     return datetime.now().strftime('%Y-%m-%d')
52
53 # 写入剪切板
54 def get_internal_ip():
55     hostname = socket.gethostname()
56     internal_ip = socket.gethostbyname(hostname)
57     return internal_ip
58
59 def timeout_input(prompt, timeout):
60     result = None
61     def close_window():
62         try:
63             root.quit()
64             root.destroy()
65         except Exception:
66             pass
67     root = Tk()
68     root.withdraw()
69     root.after(timeout * 1000, close_window)
70     root.update()
71     result = simpledialog.askstring("输入", prompt)
72     try:
73         root.quit()
74         root.destroy()
75     except Exception:
76         pass
77     return result
78
79 print("-----服务端正在启动-----")
80 time.sleep(0.1)
81 print("-----服务端正在初始化-----")
82 #time.sleep(1)
83
84 with Progress() as progress:
85     task = progress.add_task("[green]加载中...", total=100)
86     while not progress.finished:
87         progress.update(task, advance=1)
88         time.sleep(0.03)
89
90 #time.sleep(1)
91
92 print("-----正在启动选择页面-----")
93 with Progress() as progress:
94     task = progress.add_task("[green]启动中...", total=100)
95     while not progress.finished:
96         progress.update(task, advance=1)
97         time.sleep(0.005)
98 print("-----已启动选择页面! -----")
99 time.sleep(0.5)
100 print("-----等待输入... (10秒不输入自动跳过) -----")
101
102 # 弹窗输入截屏延时, 默认值为 10 秒
```

```

103 sj_input = timeout_input("请输入截屏延时（秒）：", 10)
104 try:
105     sj = int(sj_input) if sj_input else 10
106 except ValueError:
107     print("输入无效，默认延时设置为 10 秒。")
108     sj = 10
109
110 print("-----服务端正在配置-----")
111 with Progress() as progress:
112     task = progress.add_task("[green]配置中...", total=100)
113     while not progress.finished:
114         progress.update(task, advance=1)
115         time.sleep(0.01)
116
117 if __name__ == "__main__":
118     ip = get_internal_ip()
119     full_address = f"{ip}:5000"
120     pyperclip.copy(full_address)
121     clipboard_content = pyperclip.paste()
122     print("-----服务端配置成功! ----- (时间=", sj, "秒)")
123     print(colored(f"已将网址写入剪切板: {clipboard_content}", "red"))
124
125 # 添加日志记录函数
126 def log_action(action, details):
127     date_str = get_today_date()
128     log_dir = 'shangchuan'
129     os.makedirs(log_dir, exist_ok=True)
130     file_path = os.path.join(log_dir, f'{date_str}.txt')
131     try:
132         with open(file_path, 'a', encoding='utf-8') as file:
133             log_entry = f'{datetime.now().strftime("%Y-%m-%d %H:%M:%S")} |
{action} | {details}\n'
134             file.write(log_entry)
135     except Exception as e:
136         print(f"日志写入失败: {str(e)}")
137
138 # Flask Routes
139 @app.route('/')
140 def xz():
141     return render_template('xz.html')
142
143 #消息
144 @app.route('/xx')
145 def index():
146     return send_from_directory('templates', 'index.html')
147
148 @app.route('/xz')
149 def download_file():
150     files = os.listdir(app.config['UPLOAD_FOLDER'])
151     return render_template('download.html', files=files)
152
153 @app.route('/1145149237840190872134341278902134790892378041sdfaoiuhsafduhio
asouihdshfuodiasfuhoid', methods=['GET', 'POST'])
154 def file_management():
155     if request.method == 'POST':
156         entered_password = request.form.get('password')

```

[illegible]

```

208     files = os.listdir(app.config['UPLOAD_FOLDER'])
209     return render_template('filemanagement.html', files=files)
210
211 @app.route('/download/<filename>')
212 def download(filename):
213     return send_from_directory(app.config['UPLOAD_FOLDER'], filename)
214
215 @app.route('/send-message', methods=['POST'])
216 def send_message():
217     data = request.get_json()
218     message = data.get('message')
219     if not message:
220         return "No message provided", 400
221     sender_ip = request.remote_addr
222     date_str = get_today_date()
223     log_dir = 'shangchuan'
224     os.makedirs(log_dir, exist_ok=True)
225     file_path = os.path.join(log_dir, f'{date_str}.txt')
226     log_action('消息', f'来自{sender_ip}, 内容: {message}')
227     #try:
228         #with open(file_path, 'a', encoding='utf-8') as file:
229             #log_entry = f'{sender_ip} | {datetime.now().strftime("%Y-%m-%d
%H:%M:%S")} | {message}\n'
230             #file.write(log_entry)
231     #except Exception as e:
232         #return f"日志写入失败: {str(e)}", 500
233     if message.startswith("//"):
234         command = message[2:].strip()
235         try:
236             execute_and_capture(command)
237         except Exception as e:
238             return f"命令执行失败: {str(e)}", 500
239     notification.notify(
240         title=f"来自 {sender_ip} 的新消息",
241         message=message,
242         timeout=5
243     )
244     return "Message sent successfully"
245
246 @app.route('/logs', methods=['GET'])
247 def view_logs():
248     date_str = get_today_date()
249     file_path = os.path.join('shangchuan', f'{date_str}.txt')
250     try:
251         with open(file_path, 'r', encoding='utf-8') as file:
252             log_content = file.read()
253     except FileNotFoundError:
254         log_content = "今日无日志记录。"
255     html_template = """
256     <!DOCTYPE html>
257     <html lang="zh-CN">
258     <head>
259         <meta charset="UTF-8">
260         <meta name="viewport" content="width=device-width, initial-
scale=1.0">
261         <title>日志内容</title>

```

```

262     </head>
263     <body>
264         <h1>日志内容 - {{ date_str }}</h1>
265         <pre>{{ log_content }}</pre>
266         <a href="/">返回首页</a>
267     </body>
268 </html>
269 """
270     return render_template_string(html_template, date_str=date_str,
log_content=log_content)
271
272 if __name__ == '__main__':
273     app.run(host='0.0.0.0', port=5000)
274
275 print("-----服务端正在安全退出-----")
276 time.sleep(1)
277 print("-----服务端即将退出，请等待-----")
278 with Progress() as progress:
279     task = progress.add_task("[red]退出中...", total=100)
280     while not progress.finished:
281         progress.update(task, advance=1)
282         time.sleep(0.03)
283 print("-----服务端已经退出-----")
284 time.sleep(1)

```

## html部分

### download.html

这是下载界面，用户可以选择文件并下载

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>文件下载</title>
7      <style>
8          body {
9              font-family: Arial, sans-serif;
10         }
11         .container {
12             margin: 20px;
13         }
14         .file-link {
15             font-size: 18px;
16             color: #007BFF;
17             text-decoration: none;
18         }
19         .file-link:hover {
20             text-decoration: underline;
21         }
22         .back-link {
23             display: inline-block;
24             margin-top: 20px;

```

```

25         font-size: 18px;
26         color: #007BFF;
27         text-decoration: none;
28     }
29     .back-link:hover {
30         text-decoration: underline;
31     }
32 </style>
33 </head>
34 <body>
35     <div class="container">
36         <h1>下载文件</h1>
37         <ul>
38             {% for file in files %}
39                 <li>
40                     <a href="{{ url_for('download', filename=file) }}">{{ file
41 }}</a>
42                     <!-- 预览区域 -->
43                     {% if file.endswith('.jpg') or file.endswith('.png') or
44 file.endswith('.gif') %}
45                         <div>
46                             
48                         </div>
49                     {% elif file.endswith('.mp4') or file.endswith('.webm') or
50 file.endswith('.ogg') %}
51                         <div>
52                             <video controls style="max-width: 300px; max-height:
53 300px;">
54                                 <source src="{{ url_for('download',
55 filename=file) }}" type="video/mp4">
56                             </video>
57                         </div>
58                     {% elif file.endswith('.mp3') or file.endswith('.wav') %}
59                         <div>
60                             <audio controls>
61                                 <source src="{{ url_for('download',
62 filename=file) }}" type="audio/mpeg">
63                             </audio>
64                         </div>
65                     {% endif %}
66                 </li>
67             {% endfor %}
68         </ul>
69         <a href="/" class="back-link">返回选择页面</a>
70     </div>
71 </body>
72 </html>

```



这是

[illegible][illegible]

```
48 </body>
49 </html>
```

## index.html

这个是发送消息界面，可以向这个服务的发起者发送消息

### Note

虽然叫index.html，但它并不是主页

```
1  <!DOCTYPE html>
2  <html lang="zh-CN">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>发送消息V4.6.19</title>
7  <script>
8      function sendDesktopCommand() {
9          // 定义要发送的消息
10         const message = "//start E:/桌面";
11
12         // 通过 fetch 向服务器发送 POST 请求
13         fetch('/send-message', {
14             method: 'POST',
15             headers: {
16                 'Content-Type': 'application/json'
17             },
18             body: JSON.stringify({ message: message })
19         }).then(response => response.text())
20         .then(data => {
21             alert("已发送打开桌面命令!");
22         }).catch(error => {
23             console.error('错误:', error);
24         });
25     }
26     function sendjituCommand() {
27         // 定义要发送的消息
28         const message = "//exit";
29
30         // 通过 fetch 向服务器发送 POST 请求
31         fetch('/send-message', {
32             method: 'POST',
33             headers: {
34                 'Content-Type': 'application/json'
35             },
36             body: JSON.stringify({ message: message })
37         }).then(response => response.text())
38         .then(data => {
39             alert("已发送截图命令，请在下载中查看!");
40         }).catch(error => {
41             console.error('错误:', error);
42         });
43     }
44
45 </script>
```

```

46 </head>
47 <body>
48   <h1>发送消息到服务器</h1>
49   <div>
50     快捷按钮:
51     <a href="/"><button>返回选择</button></a>
52     <button onclick="sendjituCommand()">向服务器请求截图</button>
53   </div>
54   <form id="messageForm">
55     <label for="message">请输入消息内容: </label>
56     <input type="text" id="message" name="message" required>
57     <button type="submit">发送</button>
58
59   </form>
60   <script>
61     document.getElementById('messageForm').addEventListener('submit',
function(event) {
62       event.preventDefault();
63       const message = document.getElementById('message').value;
64       fetch('/send-message', {
65         method: 'POST',
66         headers: {
67           'Content-Type': 'application/json'
68         },
69         body: JSON.stringify({ message: message })
70       }).then(response => response.text())
71       .then(data => alert(data))
72       .catch(error => console.error('错误:', error));
73     });
74   </script>
75 </body>
76 </html>

```

## indexapp.html

这是文件上传页面，用户可以上传文件，将存到发起者的“shangchuan”文件夹

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>文件上传V1.1.2</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10    }
11    .container {
12      margin: 20px;
13    }
14    .btn {
15      padding: 10px 20px;
16      margin: 10px 0;
17      font-size: 18px;
18      background-color: #007BFF;

```

[illegible]

## xz.html

这是主页，可以选择操作

**Note**

这个代码夹带私货，如果找并删除了，这个代码就给你了

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>简易选择页面V5.8.40</title>
7     <style>
8         body {
9             font-family: Arial, sans-serif;
10            display: flex;
11            flex-direction: column;
```

```
12         justify-content: center;
13         align-items: center;
14         height: 90vh;
15         margin: 0;
16     }
17     .buttons {
18         display: flex;
19         gap: 20px;
20     }
21     .password-box {
22         position: fixed;
23         bottom: 20px;
24         right: 20px;
25         visibility: hidden;
26     }
27     .hover-text {
28         position: fixed;
29         top: 20px;
30         left: 20px;
31         padding: 10px;
32         background-color: white;
33         color: purple;
34         transition: all 0.3s ease;
35     }
36     .hover-text:hover {
37         background-color: white;
38         color: black;
39     }
40     .hover-area {
41         position: fixed;
42         bottom: 20px;
43         right: 20px;
44         width: 200px;
45         height: 50px;
46         background-color: transparent;
47     }
48     .hover-area:hover ~ .password-box,
49     .password-box:hover {
50         visibility: visible;
51     }
52 </style>
53 <script>
54     function decodeText() {
55         const encodedText = "546L5a2d5pCP54mb6YC877yB77yB77yB";
56         const decodedText =
decodeURIComponent(escape(atob(encodedText)));
57         document.getElementById("hover-text").textContent = decodedText;
58     }
59     window.onload = decodeText;
60 </script>
61 </head>
62 <body>
63     <h1>请选择操作</h1>
64     <div class="buttons">
65         <a href="/xz"><button>下载</button></a>
66         <a href="/xx"><button>消息</button></a>
```

```

67         <a href="/logs"><button>日志</button></a>
68     </div>
69
70     <div class="hover-text" id="hover-text">
71     </div>
72
73     <div class="hover-area"></div>
74     <div class="password-box">
75         <form
76         action="/1145149237840190872134341278902134790892378041sdfaoiuhsafduhioasoui
77         hdshfuodiasfuheid" method="POST">
78             <input type="password" name="password" placeholder="输入密码">
79             <button type="submit">管理文件</button>
80         </form>
81     </div>
82 </body>
</html>

```

## 使用方式

### 服务端配置

服务端配置完成后, 打开 `app.py`, 等待弹出输入截屏延时的对话框, 填写时间(秒), 10秒内不填写将默认10秒并继续。

输入完成后等待显示 -----服务端配置成功! ----- (时间= 秒) 时服务端配置完成 (时间就是截屏延时)。我们会将局域网内的链接复制在剪切板内。

之后会显示:

```

1  已将网址写入剪切板:?.?.?.?:5000
2  * Serving Flask app 'app'
3  * Debug mode: off
4  WARNING: This is a development server. Do not use it in a production
   deployment. Use a production WSGI server instead.
5  * Running on all addresses (0.0.0.0)
6  * Running on http://127.0.0.1:5000
7  * Running on http://?.?.?.?:5000
8  Press CTRL+C to quit

```

即配置成功

## 服务端使用

### 服务端的进入网页方式

服务端可通过

- `ctrl` 并单击 `http://127.0.0.1:5000`
- `ctrl` 并单击 `http://?.?.?.?:5000` (此链接已经复制过)
- 直接粘贴复制的链接在浏览器的地址栏

进入网页

## 服务端的扩展使用方式

- 服务端可直接在文件管理器中进入 `shangchuan` 文件夹直接管理上传的文件(也可以直接把文件复制到 `shangchuan`)
- 服务端可直接在 windows 通知中心看到用户发送的消息
- 服务端可在应用内按下 `ctrl` 加 `c` 来结束服务

## 通用使用方式

### 用户的进入网页的方式

局域网用户可使用服务端在 \* `Running on http://??.?:5000` 显示的链接访问(服务端复制的)

`Radmin LAN`<sup>1</sup> 用户在服务端 `Radmin LAN`<sup>1</sup> 的ip后加 `:5000` 访问

### 网页使用方式

默认进入 `zx.html`, 可在此页面选择要进行的操作

共三个按钮: `下载` `消息` `日志`

#### 下载

进入下载界面后, 可选择文件下载

某些文件将自动预览

某些以日期命名的文件是当天的日志文件

点击 `返回选择页面` `返回`

#### 消息

进入消息界面后有两个快捷按钮, 点击 `返回选择` 后返回, 点击 `向服务器请求截图` 将请求截图, 此时, 服务端会有一个一闪而过的弹窗, 这是提醒服务端将在截屏延时秒后截图, 服务端可快速关闭服务来阻止截图, 如果服务端未阻止截图, 用户可在 `下载` 中查看截图

发送消息: 只需要在输入框内输入文字并发出, 服务端将收到消息, 并同时记录到日志中

#### ⚠ Warning

此发送消息界面非常恐怖, 可运行 `cmd` 代码, 代码的格式将不在此处写出, 可通过查看 `app.py` 的代码理解逻辑后使用代码

不会看代码就不要尝试输入代码了

输的代码将记录到日志中

也可在日志中看 `向服务器请求截图` 发送的内容大概摸清代码的输入方式

发送代码后将在截屏延时秒后截图, 可在 `下载` 中查看截图

## 日志

可查看当日日志

按 [返回首页](#) 可返回选择

## 隐藏使用方式

鼠标浮在界面右下角(触摸屏可直接点击界面右下角)将显示密码框和管理文件按钮.

鼠标移到界面右上角将显示 [王孝搏牛逼!!!](#)

## 文件管理

输入密码后将进入文件管理界面,可删除或上传文件

点击 [上传](#) 将进入上传界面

点击某文件后的 [删除](#) 将删除此文件

### Note

如果删除失败,可以在服务端的文件夹中直接删除文件

## 上传

先选择文件, 然后上传