

考研倒计时邮件提醒

这是一个 Python 脚本，用于每日定时发送考研倒计时提醒邮件。你可以将它部署在自己的服务器上，实现无人值守的自动化提醒。

💎 功能特性

- **定时任务**：每日早晚自动发送邮件。
- **可定制化**：邮件内容模板可自由修改。
- **部署简单**：提供了详细的 `systemd` 后台部署指南。
- **高兼容性**：使用 Python 标准库和少量第三方库，兼容性好。

🚀 快速开始

准备工作

1. 一台安装了 **Python 3** 的 Linux 服务器。
2. 一个邮箱账号（例如 QQ 邮箱），并已开启 **SMTP** 服务、获取了**授权码**（注意不是邮箱密码）。

部署步骤

第一步：准备项目文件

首先，将项目文件上传到你的服务器。然后进入项目目录。

```
cd /path/to/your/project/kaoyan-reminder-email
```

第二步：创建并激活虚拟环境

使用虚拟环境是一个好习惯，可以隔离项目依赖，避免与系统环境冲突。

```
# 1. 如果你的系统没有 `venv` 模块，请先安装
# sudo apt install python3-venv -y

# 2. 创建虚拟环境（venv 是虚拟环境的名称，你可以自定义）
python3 -m venv venv

# 3. 激活虚拟环境
source venv/bin/activate
```

提示：激活成功后，你的命令行提示符前会显示 `(venv)` 字样。

第三步：安装依赖

在激活的虚拟环境中，安装脚本所需的第三方库。

```
pip install requests apscheduler
```

第四步：配置并测试脚本

在运行前，你需要修改 `kaoyanemail.py` 脚本中的邮箱配置。

1. **修改配置**：打开 `kaoyanemail.py` 文件，找到并修改发件人、授权码和收件人等信息。
2. **直接运行测试**：

```
python kaoyanemail.py
```

运行后，检查你的收件箱是否能收到邮件。

⚙️ 配置为后台服务 (Systemd)

为了让脚本能在服务器上长期稳定地后台运行，推荐使用 `systemd` 进行管理。

第一步：获取 Python 解释器路径

在已激活的虚拟环境中，运行以下命令获取 Python 解释器的绝对路径。

```
which python
```

你会得到一个类似 `/root/your_project/venv/bin/python` 的路径，请复制并记下它。

第二步：创建 Systemd 服务文件

创建一个新的服务文件。

```
sudo nano /etc/systemd/system/kaoyan-reminder.service
```

将以下内容粘贴到文件中。**注意**：请务必将 `ExecStart` 和 `WorkingDirectory` 的路径替换为你自己的实际路径。

```
[Unit]
Description=Kaoyan Email Reminder Service
After=network.target

[Service]
ExecStart=/root/your_project/venv/bin/python /root/your_project/kaoyanemail.py
WorkingDirectory=/root/your_project/
Restart=always
```

```
User=root
# 如果你使用非 root 用户，请修改为对应的用户名

[Install]
WantedBy=multi-user.target
```

第三步：启动并管理服务

完成配置后，使用以下命令来管理你的服务。

```
# 重新加载 systemd 配置，让新服务文件生效
sudo systemctl daemon-reload

# 启动服务
sudo systemctl start kaoyan-reminder.service

# 查看服务状态
sudo systemctl status kaoyan-reminder.service
```

如果看到 **Active: active (running)** 的绿色字样，说明服务已成功启动。

```
# (可选) 设置服务开机自启
sudo systemctl enable kaoyan-reminder.service

# (可选) 停止服务
sudo systemctl stop kaoyan-reminder.service
```

故障排查

1. 查看服务日志

如果服务启动失败或运行异常，首先应该查看日志。

```
# 查看最新的 50 条日志
journalctl -u kaoyan-reminder.service -n 50 --no-pager
```

2. 测试 SMTP 服务器连通性

很多云服务商（如腾讯云、阿里云）默认会封禁 25、465 等邮件端口，导致无法发送邮件。

```
# 使用 telnet 测试 465 端口
telnet smtp.qq.com 465
```

- 正常情况：显示 `Connected to smtp.qq.com`，并等待输入。
- 异常情况：提示 `Connection timed out` 或 `Unable to connect`，说明你的服务器网络无法访问该端口。你需要联系云服务商解封端口。

3. 手动执行邮件发送脚本

你也可以使用下面的独立脚本来排查是否是 `SMTP` 登录或邮件发送本身的问题。

```
import smtplib
from email.mime.text import MIMEText

# --- 请修改以下配置 ---
SENDER_EMAIL = "1958232837@qq.com"      # 你的发件邮箱
SENDER_AUTH_CODE = "你的授权码"         # 你的邮箱授权码
RECEIVER_EMAIL = "1958232837@qq.com"    # 收件人邮箱
# -----

msg = MIMEText("这是一封从服务器发送的测试邮件。", "plain", "utf-8")
msg["Subject"] = "SMTP 功能测试"
msg["From"] = SENDER_EMAIL
msg["To"] = RECEIVER_EMAIL

try:
    # 连接到 QQ 邮箱的 SMTP 服务器，SSL 加密，端口 465
    server = smtplib.SMTP_SSL("smtp.qq.com", 465)

    # 启用 Debug 模式，打印出详细的交互信息
    server.set_debuglevel(1)

    # 登录邮箱
    server.login(SENDER_EMAIL, SENDER_AUTH_CODE)

    # 发送邮件
    server.sendmail(SENDER_EMAIL, [RECEIVER_EMAIL], msg.as_string())

    # 关闭连接
    server.quit()
    print("☑ 邮件发送成功！")

except Exception as e:
    print(f"✗ 邮件发送失败，原因：{e}")
```

将以上代码保存为 `test_email.py`，修改配置后运行 `python test_email.py`，根据详细的错误输出来定位问题。