**自动化消息发送工具文档**

**1. 程序概述**

本程序是一款基于 **Python** 的桌面自动化工具，用于 **读取 Excel 账单表格** 并通过 **模拟按键操作** 自动向客户经理发送欠费通知。

**核心特点：**

* 自动读取账单信息（企业名称、欠费金额、客户经理姓名、电话号码）
* 自动打开通讯录并定位客户经理
* 构造消息内容并发送
* 支持中文消息发送（通过剪贴板方式）
* 可打包为单个可执行文件，便于分发和使用

**2. 环境要求**

**2.1 软件环境**

* Windows 10 或以上
* Python 3.9 及以上

**2.2 Python 依赖库**

| **库** | **作用** | **安装方式** |
| --- | --- | --- |
| pandas | 读取 Excel 文件 | pip install pandas |
| pyautogui | 模拟鼠标和键盘操作 | pip install pyautogui |
| pyperclip | 处理剪贴板操作，用于中文输入 | pip install pyperclip |
| openpyxl | Excel 文件支持（pandas 依赖） | pip install openpyxl |
| time, string | 内置库 | 无需安装 |

**3. 程序功能说明**

**3.1 核心功能**

1. **读取账单 Excel**
   * Excel 表格必须包含列：
     + 企业名称
     + 欠费金额
     + 客户经理
     + 电话号码
2. **自动定位客户经理**
   * 使用 pyautogui 模拟：
     + Ctrl+F 打开搜索框
     + 输入客户经理姓名或电话号码
     + Enter 键进入聊天窗口
3. **自动发送消息**
   * 构造消息模板：
   * 你好，{公司名称} 客户已产生欠费，欠费金额为：¥{欠费金额}。请及时联系客户处理。谢谢！
   * 中文消息通过剪贴板 + Ctrl+V 输入，避免乱码
   * 按 Enter 键发送
4. **遍历 Excel 所有行**
   * 对每个客户经理重复执行发送操作
   * 发送后等待 1~2 秒，保证操作稳定

**3.2 运行前要求**

* 通讯录窗口必须处于 **激活状态**
* 鼠标停留在屏幕中心，保证 pyautogui 点击准确
* Excel 文件路径正确，建议与程序在同一文件夹

**4. 使用方法**

**4.1 直接运行 Python 脚本**

python auto\_send.py

* 运行后程序会提示：
* 程序将在5秒后开始，请将鼠标移动到通讯录窗口...
* 5 秒倒计时后开始遍历 Excel 并发送消息

**4.2 打包为可执行文件（可选）**

* 使用 **Auto-py-to-exe** 或 **PyInstaller**：

pyinstaller --onefile --noconsole auto\_send.py

* 可选设置：
  + Icon (-i)：自定义 exe 图标
  + Additional Files (-add-data)：添加 Excel 文件
  + Advanced Settings：
    - 排除无用模块：matplotlib, scipy, cv2
    - 使用 UPX 压缩（减小文件体积）

**5. 当前使用情况**

* 程序会在所在文件夹中读取 Excel 表格
* 启动时需要手动启动移动办公并置于最前端
* 通过模拟按键完成消息发送
* 对于已建立对话的客户经理，程序发送稳定
* 中文消息可以正确发送，通过剪贴板输入避免乱码

**6. 使用中存在的问题**

1. **通讯录搜索对未对话人员不生效**
   * 输入电话号码并按 Enter，无法直接进入新聊天页面
   * 程序会返回到上一个聊天窗口，导致消息无法发送
   * 原因：纯模拟按键依赖客户端 UI，无法直接建立新对话
2. **屏幕依赖性高**
   * 需要固定窗口位置和分辨率
   * 弹窗或窗口遮挡可能导致点击失效
3. **性能与稳定性问题**
   * 大量消息发送可能出现延迟、输入错误或界面刷新不及时

**7. 当前可行的使用建议**

* 确保通讯录窗口置顶，在操作过程中不要进行额外操作
* 对未建立过对话的员工，先手动建立聊天
* 消息发送间隔保持 1~2 秒
* Excel 文件路径固定，建议与程序同目录

**8. 未来优化方向**

1. **使用官方 API 或 SDK**
   * 替代模拟按键，直接建立聊天并发送消息
   * 优点：稳定性高，不依赖屏幕位置
2. **增加 GUI 界面**
   * 可选择 Excel 文件并显示发送状态
   * 增加失败重试和日志功能
3. **智能识别未对话员工**
   * 通过 OCR 或读取聊天列表判断是否存在对话
   * 对未对话员工提示手动处理

**9. 程序结构**

auto\_send.py

├── import string, pandas, pyautogui, pyperclip, time

├── send\_message\_via\_ui() # 核心发送函数

├── main() # 读取 Excel 并遍历每一行

└── if \_\_name\_\_ == "\_\_main\_\_"

* send\_message\_via\_ui：模拟键盘鼠标操作，构造消息并发送
* main：读取 Excel 数据，逐行调用发送函数
* \_\_main\_\_：程序入口，提示用户准备通讯录窗口