

下面是我为你的脚本打造的**完整图形化界面（GUI）实现，不修改任何核心逻辑，而是直接调用你现有的 `book()` 与 `timer_run()`**。我把它封装成一个独立的 `piano_booking_gui.py` 文件，并把所有参数持久化到 `config.yaml`，满足你提出的交互与样式细节（滚轮式时间设定、双圈时间轮盘、Cookie 单独弹窗、设置面板、可添加多组预定请求、自动 2 小时切片、结果判定/弹窗/200ms 重试等）。核心函数来源并保持不变：`(main.txt)`

下载：

[下载 `piano_booking_gui.py`](#)

你将获得什么

- 一个**美观、易用、零命令行**的界面程序（基于 **PySide6/Qt**）。
- 支持**设置弹窗**（`proxies`，`user_id`，`user_name`，`user_email`，`user_phone`，`theme`，其中前四项为**必填**，红色星号标注）。
- **Cookie** 通过单独弹窗粘贴，**主界面不显示 Cookie 内容**，只显示**“上次更新时间”**。
- **目标启动时间使用滚轮式组件**（日期 + 时分秒）且有**“立即启动”勾选框**；**默认值为当天 21:00:00**。
- **预定请求**可按**“+”**添加一组**“无限扩展”**；每组包含：
 - `place`：**可编辑下拉**（内置所有可选琴房，支持自动补全，也可直接输入）。
 - `start_time` / `end_time`：**双圈时间轮盘**（两个“滚轮”，小时与分钟），范围**06:00–23:00**，并强制**开始 ≤ 结束**。
 - **显示预览格式**严格为 `YY-MM-DD HH:MM - HH:MM`，如 `25-09-17 13:00 - 14:30`，默认日期为**明天**。
- **每条请求时长 > 2 小时自动切片**为若干段（每段 ≤ 120 分钟）并分别调用 `book()`，以**最大化成功率**。
- **故障处理与弹窗**严格遵循你的规则：
当且仅当返回字符串出现以下**四类子串之一**才弹窗并停止该段重试；否则**每 200ms 自动重试**：

- Cookie 过期
 - 保存成功
 - 手速太慢，该时间段已经被预订啦
 - 请求失败，检查网络、代理服务器或 VPN
 - **并行提交**：多段/多组会并行执行（多线程），以**卡点提交**。
 - 运行日志实时显示在界面底部；**所有参数写入 config.yaml**，下次直接复用。
-

运行步骤

1. 准备依赖（一次性）

```
pip install PySide6 pyyaml
```

2. 将你的核心脚本 `main.txt` 与 `piano_booking_gui.py` 放在同一目录（若你把 `main.txt` 改名为 `main.py` 也可，程序会自动优先导入 `main.py`）。

3. 运行：

```
python piano_booking_gui.py
```

4. **第一次使用**：右上角点击“**设置**”填入必填项（红色星号）；点击“**设置 Cookie**”粘贴 Cookie（主界面只显示**上次更新时间**）。

- `proxies` 文本框**直接粘贴 JSON 或 YAML**，例如：

```
{"http": "10.101.28.225:9000", "https": "10.101.28.225
```



也可简写为一行 `host:port`（程序会自动映射到 `http/https`）。

5. 在主界面点击“+ **添加一组预定请求**”，完善 `place`、日期、开始/结束时间（**06:00–23:00**）。

6. 设置**启动时间**（滚轮设定），或勾选**立即启动**。

7. 点击**开始**（按钮就在**启动时间**右侧）。

- 若未勾选“立即启动”，程序将使用你脚本的 `timer_run()` 在设定的**绝对时间**启动所有并发请求（界面保持开启即可）。

设计与实现要点（逐条对照你的要求）

设置界面（右上角按钮进入）

- **字段：** `proxies, user_id, user_name, user_email, user_phone, theme`。
其中 `proxies, user_id, user_name, user_email` **必填**，名称旁以**红色星号**标示；文本框内有**灰色示例**（如 `example@link.cuhk.edu.cn`），一旦输入便消失。
- **保存到 config.yaml：**所有字段（包含 Cookie 和更新时间）在每次保存或主要操作时**自动写入**同目录下的 `config.yaml`，下次打开自动填充。

主界面

- **target_time：**滚轮式（Wheel）日期/时间选择器，放在“开始”按钮旁边；默认今日 21:00:00；提供**“立即启动”**勾选框（勾选=马上跑；不勾选=准点跑）。
- **Cookie：**主界面**不显示内容**，仅在点击按钮后弹出编辑框粘贴；关闭后主界面仅显示**“Cookie 已设置，上次更新时间：YYYY-MM-DD HH:MM:SS”**（或“未设置”）。
- **预定请求分组：**
 - 通过 “+ 添加一组预定请求” 动态添加。
 - 每组包含：
 - **place：**可编辑下拉（内置全部琴房名，带**自动补全**），也可直接输入。
 - **start_time / end_time：**双圈时间轮盘（两个滚轮：小时与分钟），限制在 06:00–23:00，并强制**开始 ≤ 结束**。
 - **显示预览：**即时显示为 YY-MM-DD HH:MM - HH:MM（如 25-09-17 13:00 - 14:30）。
日期默认**明天**，以提升命中率。

- **超过 2 小时自动切片**：每个请求若超过 120 分钟，会被**任意分割**成若干段（每段 ≤ 120 分钟）并分别调用 `book()`。（保证满足系统限制）

故障处理（重试/弹窗）

- 每次 `book()` 返回可能包含杂乱字符，但**只要未命中**以下任一关键子串：
Cookie 过期、保存成功、手速太慢，该时间段已经被预订啦、请求失败，
检查网络、代理服务器或 VPN
—— **就继续每 200ms 重试**；
一旦命中上述四种中的任意一种，**立刻弹窗并停止该段**的重试。
- 结果类型与弹窗级别：
 - 保存成功 → 信息提示（成功）
 - 手速太慢... → 警告提示（被抢）
 - Cookie 过期 / 请求失败... → 错误提示（需要处理）

并发与准点

- **并行**对所有切片与分组**启动线程**请求，降低等待时间，提升“卡点”成功率。
 - **准点启动**：当未勾选“立即启动”时，严格使用你的 `timer_run(target_time, func)` **按绝对时间**（到秒）触发，在触发时瞬时启动所有并发请求。
-

文件结构与持久化

- `piano_booking_gui.py`：GUI 主程序（你现在就可以下载并运行）。
 - `main.txt`（或 `main.py`）：**你的核心函数文件**，被 GUI 动态导入，仅**调用不修改**。（`main.txt`）
 - `config.yaml`：所有参数的持久化文件（含 `requests` 分组、`target_time`、`cookie` 与 `cookie_updated_at` 等），程序会在每次设置/操作时自动更新。
-

关键实现片段（说明用，已集成在文件中）

- **加载核心，不做改动：**

```
# 优先 import main.py; 否则从 main.txt 动态加载
from importlib.machinery import SourceFileLoader
core = importlib.import_module("main") # 若存在
# 否则:
core = SourceFileLoader("cuhk_booking_core", "main.txt").load_module()
book = core.book
timer_run = core.timer_run
```

- **2 小时自动切片：**

```
def split_to_slots(start_hm, end_hm, max_minutes=120):
    # 返回若干 (HH:MM, HH:MM) 段
```

- **重试逻辑 (200ms) + 四类结果弹窗：**

Worker 线程对每段调用 `book()`，直到命中四类关键词之一才停止该段；否则 `time.sleep(0.2)` 重试。

交互细节与可用性

- **滚轮体验：**我自定义了 `WheelCombo + DateWheel + TimeWheel`，放大字号、增大下拉高度与滚轮响应，**在桌面环境中近似 iPhone 样式的纵向滚轮**（不依赖外部复杂组件，跨平台稳健）。
- **场地选择：**内置了与你核心函数中 `FID_MAP` 键一致的完整场地列表（带自动补全）；也可直接输入（若输入与系统不匹配，后端会返回“地点错误”——这不是四类结果之一，**按你的规则仍将 200ms 重试**。为避免误操作，界面已经尽可能引导选择列表项）。
- **即时保存：**任何重要变更（请求项、目标时间、立即启动勾选）都会即时保存到 `config.yaml`，避免意外退出带来丢失。

小提示

- **proxies 填写**：建议粘贴 JSON（或 YAML），例如：

```
{"http": "10.101.28.225:9000", "https": "10.101.28.225:9000"}
```



也可直接 `10.101.28.225:9000` 单行（程序会映射为 http/https）。

- **Cookie** 必须是能够访问预约系统的有效 Cookie。如果过期，界面会弹窗提示“**Cookie 过期**”，请重新在右上角设置 Cookie。
- **时间边界**：为了避免越界，时间轮盘强制在 **06:00–23:00**。若结束时间设得比开始早，界面会自动纠正。

接下来你可以做什么

1. 点击上方链接下载 `piano_booking_gui.py`。
2. 与你的 `main.txt` 放在一起并运行。
3. 如需打包为可执行文件（你说这不是本次任务，这里仅一行提示）：可用 `pyinstaller -F -w piano_booking_gui.py`。

如需我把按钮布局、视觉主题或“滚轮”动画**进一步拟真 iOS 风格**（QML/动画/毛玻璃效果等），告诉我你的偏好，我会在不触碰核心逻辑的前提下继续精细化。