

# GotLucky 开发协作完整对话记录

## ## 项目背景

\*\*GotLucky\*\* 是一个利用 Lucky 服务器、FRP 穿透配置和 Cloudflare DNS 信息的自动化内网服务导航系统。

---

## 阶段一：可视化增强与细节打磨 (Step 1 - 70)

### ### 1. 动态染色标签与过滤器

- \*\*目标\*\*: 为服务备注 (Comment) 标签和对应的过滤器按钮添加动态颜色，使界面更美观。
- \*\*方案\*\*: 在 `template.html` 中引入 `commentPalette` 颜色组，并实现基于哈希的 `getColorForComment` 函数，确保相同的备注始终显示相同颜色。
- \*\*成果\*\*: 实现了 Google 风格的多彩标签系统。

### ### 2. FRP 路径识别增强

- \*\*目标\*\*: 准确识别并显示 "Pre-Lucky FRP" (Entry → FRPS → Lucky) 的映射关系，特别是当 Lucky 位于内网时。
- \*\*方案\*\*:
  - 改进 `main.py` 匹配逻辑，同时检查 FRP 的 `remote\_port` 和 `local\_port`。
  - 增加 FRP 服务端地址 (server\_addr) 的自动 DNS 解析。
  - 识别内网穿透场景并标注 `(内网Lucky)`。
- \*\*成果\*\*: 解决了 `sp.nat.nasnas.site` 等服务的路径溯源问题。

### ### 3. IP 别名系统优化

- \*\*目标\*\*: 在路径展示中，优先使用 `config.json` 中的 `ip\_aliases`，不再直接显示 Lucky 服务器名称作为入口 IP 的标注。
- \*\*成果\*\*: 路径显示更加直观（如：`144.x.x.x (搬瓦工) → ...`）。

---

## 阶段二：功能微调与 Bug 修复 (Step 71 - 82)

### ### 4. 端口映射逻辑自动修正

- \*\*用户反馈\*\*: 当流量通过 FRP 进入内网 Lucky 时，导航链接应使用 FRP 的 `remote\_port` 而不是 Lucky 的 `local\_port`。
- \*\*改进\*\*: 在逻辑中增加 `display\_port = p\_remote` 的自动修正。如果匹配到前置 FRP，系统会自动切换对外访问端口，确保点击跳转成功。

---

## 阶段三：数据脱敏与 demo 发布 (Step 83 - 192)

### ### 5. 深度去隐私化处理

- \*\*需求\*\*: 生成一个完全不含敏感信息的 `index.html` 用于项目演示。
- \*\*开发过程\*\*:
  - 编写 `anonymize\_data.py` 脱敏工具。
  - \*\*域名脱敏\*\*: 保留层级结构 (如 `a.b.com` → `xabc.ydef.zghi`)，但混淆实际文字。
  - \*\*IP 脱敏\*\*: 所有 IP 映射为 `172.16.x.x` 随机段。
  - \*\*端口混淆\*\*: 所有敏感端口随机化为 `10000-60000` 之间的端口。
  - \*\*内容清洗\*\*: 自动识别并替换备注、备注标签、FRP 配置文件名中的敏感信息。
- \*\*成果\*\*: 成功在 `nopersonalinfo/` 目录下生成了安全的 demo 页面。

---

## 阶段四：GitHub 仓库与在线演示 (Step 193 - 280)

### ### 6. 仓库忽略规则优化

- \*\*改进\*\*: 修改 `.gitignore` 规则，将 `index.html` 等忽略规则锁定在根目录 (`/index.html`)，从而允许 `demo/index.html` 被 Git 跟踪并提交。

### ### 7. 部署 Live Demo

- \*\*操作\*\*: 协助用户利用 GitHub Pages 功能，配合自定义域名 `git.nasnas.site` 部署了在线演示页面。
- \*\*README 更新\*\*: 增加了醒目的 \*\*Live Demo\*\* 链接入口。

----

## ## 协作总结

通过本次协作，GotLucky 项目在代码健壮性、路径分析深度、UI 美观度以及项目开源标准（脱敏与演示）上均得到了显著提升。

### \*\*当前状态\*\*:

- 项目代码已同步至远程仓库。
- 自动化脱敏流程已建立。
- 在线演示页面已通过自定义域名上线。