




Reddit hotpost

README Reddit 热帖自动化 · n8n → 飞书多维表 → LLM 洞察指南

能力概览

-  定时抓取指定 Subreddit 的热门帖子，自动过滤 NSFW、置顶或低分内容。
-  将帖子关键信息结构化写入飞书多维表，便于追踪与二次加工。
-  调用 DeepSeek LLM 产出爆火分析、选题建议与改写文案，并推送飞书群机器人。

Reddit → n8n (三套工作流) → Feishu Bitable & Webhook

↘ DeepSeek LLM ↗

本项目提供三份 n8n 工作流，请按顺序导入和配置：

工作流	作用	触发方式
Reddit · Initialize Feishu Bitable Fields (Run Once)	初始化飞书表字段（含 LLM 解析字段）	手动运行一次
Reddit Hotposts to Feishu Bitable (Batch)v3	周期性抓取 + 写入飞书表	每 6 小时 + 手动测试
Reddit - High Score Rewrite Broadcast v3	调用 LLM 生成洞察并推送飞书群	每日一次 + 手动测试

准备工作

1. 环境要求

- 已部署的 n8n 实例（建议 v1.40+）。
- Reddit OAuth2 应用（只读权限）。
- 飞书自建应用，已开通 Bitable 与机器人 Webhook 能力。
- DeepSeek API Key（或兼容 OpenAI 接口的模型 API）。

2. n8n Credentials

- **Reddit OAuth2**：用于拉取帖子。
- **DeepSeek API**：HTTP 节点调用 `chat/completions`。
- 飞书接口通过 HTTP 节点动态获取 `tenant_access_token`，无需额外 Credential。

3. 环境变量

变量	示例	说明
<code>REDDIT_CLIENT_ID</code>	<code>abc123xyz</code>	Reddit OAuth2 Client ID
<code>REDDIT_CLIENT_SECRET</code>	<code>def456uvw</code>	Reddit OAuth2 Client Secret
<code>REDDIT_USER_AGENT</code>	<code>hotpost-n8n/0.1 (by u/yourname)</code>	Reddit API User-Agent
<code>FEISHU_APP_ID</code>	<code>cli_XXXXXX</code>	飞书应用 ID
<code>FEISHU_APP_SECRET</code>	<code>XXXXXX</code>	飞书应用 Secret
<code>FEISHU_BITABLE_REDDIT_APP_TOKEN</code>	<code>HGeEb...</code>	Reddit 帖子表 <code>app_token</code>
<code>FEISHU_BITABLE_REDDIT_TABLE_ID</code>	<code>tbl0a9...</code>	Reddit 帖子表 <code>table_id</code>
<code>FEISHU_WEBHOOK_URL</code>	<code>https://open.feishu.cn/...</code>	飞书群机器人地址
<code>DEEPSEEK_MODEL</code>	<code>deepseek-chat</code>	LLM 模型名

| `DEEPSEEK_BASE_URL` (可选) | `https://api.deepseek.com/v1/chat/completions` | LLM 接口地址 |

| `LIMIT_PER_SUB` (可选) | `30` | 每个 Subreddit 默认抓取数量 |

| `SCORE_THRESHOLD` (可选) | `50` | 过滤分数阈值 |

💡 如果未设置 `SCORE_THRESHOLD`，默认使用 50；未设置 `LIMIT_PER_SUB`，默认使用 30。

预留变量 `FEISHU_BITABLE_REDDIT2_APP_TOKEN` / `FEISHU_BITABLE_REDDIT2_TABLE_ID``
暂未使用，可留空或用于后续扩展。

🔧 导入与初始化

1. 导入 workflow JSON

在 n8n 中依次导入三份 JSON，检查工作流名称与结构是否正确。

2. 配置 Reddit OAuth2 Credential

- Redirect URL 固定为 `http://临时公网/rest/oauth2-credential/callback`（如：<https://job-partnerships-anatomy-wrote.trycloudflare.com/rest/oauth2-credential/callback> 详见下面教程部分）。
- Scope 选择 `read`。

3. 配置 DeepSeek Credential

- 在 Credentials 中创建 `DeepSeek API`，填写 API Key。若更换模型服务商，请同步更新 `DEEPSEEK_BASE_URL`。

4. 初始化飞书表字段

- 打开 `Reddit · Initialize Feishu Bitable Fields (Run Once)`，点击 *Execute Workflow*。

- 工作流会自动创建包括 LLM 输出在内的全部字段：

`subreddit`, `post_id`, `title`, `author`, `content`, `score`, `num_comments`,
`created_utc`, `permalink`, `url`, `thumbnail`, `over_18`, `stickied`, `created_at`,
热点趋势, 爆火原因, 选题建议, 文案建议 (含推荐标签), 相关性。

- 若字段已存在，工作流会跳过并在总结节点里提示。

5. 校验环境变量

- `Reddit Hotposts to Feishu Bitable (Batch)v3` 中的 `Config` 节点可直观查看变量解析结果。
- 修改 `SUBREDDIT_LIST` 为目标社区（例如 `["MachineLearning", "LocalLLaMA", "Automation"]`）。

- 想额外通过关键词补充热点时，可设置 `KEYWORD_LIST`（如 `["n8n workflow", "automation growth"]`）。

workflow说明

1. Reddit · Initialize Feishu Bitable Fields (Run Once)

- 自动创建飞书多维表字段，支持重复执行。
- 关键节点： `Configuration` → `Get Feishu Token` → `Prepare Field List` → `Create Field` → `Summarize`。
- 运行结果会说明新增 / 已存在 / 失败的字段数量。

2. Reddit Hotposts to Feishu Bitable (Batch)v3

- **触发**：内置 Cron 每 6 小时，可改成任意频率。

- **主要步骤**：

- 拆分 Subreddit → Reddit API 获取热帖 → 过滤 NSFW/置顶/低分 → 字段映射 → 批内去重。
- 与飞书表对比 `post_id`，跳过已存在记录。
- 分批写入飞书表（默认每批 10 条）。
- 如设置 `KEYWORD_LIST`，还会额外搜索全站关键词结果。

- **Config 节点可自定义的关键参数**：

- `SUBREDDIT_LIST`：数组格式的社区列表，示例 `["MachineLearning", "LocalLLaMA"]`。可直接在节点里编辑，也可以改为从环境变量读取。
- `LIMIT_PER_SUB`：每个社区抓取的帖子数，默认取环境变量 `LIMIT_PER_SUB`（无则为 30）。
- `SCORE_THRESHOLD`：保留的最低得分，默认取环境变量 `SCORE_THRESHOLD`（无则为 5）。
- `KEYWORD_LIST`：可选的关键词数组，用于触发“Search for a post”分支进行全站搜索；留空或删除即不执行关键词检索。
- `REDDIT_USER_AGENT`：Reddit API 的 UA 字符串，若需区分环境可分别设置。

- **可调参数**：Cron 周期、抓取数量、分数阈值、关键词列表等。

3. Reddit - High Score Rewrite Broadcast v3

- **触发**：默认每日执行，可自行调整。

- **主要步骤**：

- 通过视图 ID（默认 `vewbLDXZ8j`）读取飞书表记录。
- 初筛包含 `n8n` 等关键词的帖子。
- 每批随机 10 条调用 DeepSeek LLM，生成热点趋势、爆火原因、选题、文案及相关性评分。
- 将结果写回表中对应记录，同时筛出相关性 ≥ 8 的帖子推送飞书群卡片。

- **配置要点**：确保飞书群机器人 URL、DeepSeek Credential 和相关字段类型正确（建议多行文本 / 数字）。

- **视图配置说明**：

- 默认使用的视图 ID `vewbLDXZ8j` 来自飞书多维表中的一个自定义视图：过滤条件为 `score ≥ 500` 且 `created_utc` 晚于 2025/01/01，并按 `subreddit` 分组。
- 若需替换，可在飞书表中创建自己的视图（设置好过滤、排序、分组），然后从视图链接（例如 `https://...&view=XXXXXXXX`）中复制 `view=` 后面的 ID，替换工作流内 `Build Query` 节点中的值。

✅ 验证流程

1. 手动执行 **初始化** 工作流，确认 Summary 输出。
 2. 测试执行 **抓取工作流**：确保 Reddit 节点返回 200，飞书写入返回 ``code: 0``。
 3. 测试执行 **洞察工作流**：检查 DeepSeek 返回的 JSON 是否完整，飞书表字段与卡片推送同步更新。
1. 在 n8n Execution 日志中查看详细输入输出，排查任何异常。

🐛 常见问题排查

- **字段重复错误（code 1254044）**

飞书字段已存在，忽略即可或手动确认字段名。

- **LLM 输出格式异常**

若解析失败，可降低 `temperature`，或检查源帖子正文是否超长。

- 抓取数量少

Reddit 热门接口受社区活跃度限制，可增加更多 Subreddit 或开启关键词搜索。

🚀 后续扩展建议

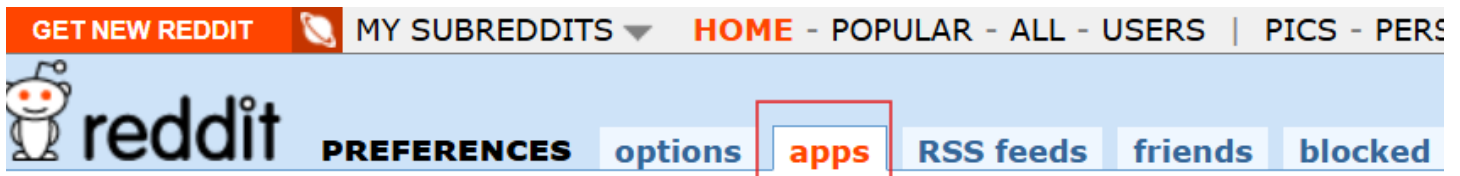
- 将 `KEYWORD_LIST` 从飞书表中读取，实现动态配置。
- 在抓取流程添加自动翻译、情绪分析等节点。
- 使用预留的第二张表变量，将 LLM 结果同步至独立洞察表。

Happy automating! ✨

🚀 教程

1. Reddit 新建应用与 n8n 回调设置

在 Reddit 新建应用 (<https://old.reddit.com/prefs/apps>)：



authorized applications

developed applications



重点填写：**name**、**redirect uri**（其他都可以留空，不影响）

Reddit 应用回调 URL 不能使用 `localhost:5678`（使用 docker+n8n 的 n8n 地址），必须是一个可访问的公网。（推荐使用 cloudflare）

下载并安装 Cloudflare Tunnel

代码块

```
1 # Windows
2 # 从 https://github.com/cloudflare/cloudflared/releases 下载 对应版本
```

🧩 临时公网 (快速调试)

cloudflare 每次在终端如下启用会生成一个可用的公网URL，但是终端关闭/电脑重启这个 URL 就会变化），使用它的公开网址，把自托管的 [localhost:5678](#)，替换成 cloudflared 输出的公开网址，获取方式是：下载完 cloudflared，在终端（Windows+R）打开，输入

代码块

```
1 cd C:\cloudflared #你下载cloudflare的路径
2 cloudflared tunnel --url http://localhost:5678
```

```
C:\Users\ >cloudflared tunnel --url http://localhost:5678
2025-11-06T03:54:01Z INF Thank you for trying Cloudflare Tunnel. Doing so, without a Cloudflare account, is a quick way
to experiment and try it out. However, be aware that these account-less Tunnels have no uptime guarantee, are subject to
the Cloudflare Online Services Terms of Use (https://www.cloudflare.com/website-terms/), and Cloudflare reserves the ri
ght to investigate your use of Tunnels for violations of such terms. If you intend to use Tunnels in production you shou
ld use a pre-created named tunnel by following: https://developers.cloudflare.com/cloudflare-one/connections/connect-app
s
2025-11-06T03:54:01Z INF Requesting new quick Tunnel on trycloudflare.com...
2025-11-06T03:54:06Z INF +-----+
2025-11-06T03:54:06Z INF | Your quick Tunnel has been created! Visit it at (it may take some time to be reachable): |
2025-11-06T03:54:06Z INF | https://job-partnerships-anatomy-wrote.trycloudflare.com |
2025-11-06T03:54:06Z INF +-----+
```

然后会看到终端输出：一个新的公开网址如 <https://job-partnerships-anatomy-wrote.trycloudflare.com>

🧱 固定公网 (长期使用)

代码块

```
1 #cloudflared.exe service install <your_token>
2 cloudflared service install <token>
3 #启动服务:
4 sc start Cloudflared
```

如果你有自己的域名，可以借助 cloudflare 生成一个固定的 tunnel 公网，这样就不会因为每次重启更换 URL 了。公网只要不在 cloudflare 控制台刷新 token，tunnel 就不需要每次启用终端命令，只需要偶尔检查运行状态即可：

代码块

```
1 #查看状态:
2 sc query Cloudflared #显示 STATE: 4 RUNNING 即表示隧道成功连接
3 cloudflared.exe tunnel info n8n-zero
4 #回调地址填写为:
5 https://固定公网/rest/oauth2-credential/callback
```

把这个公网设置为 docker-compose.yml 的环境变量：

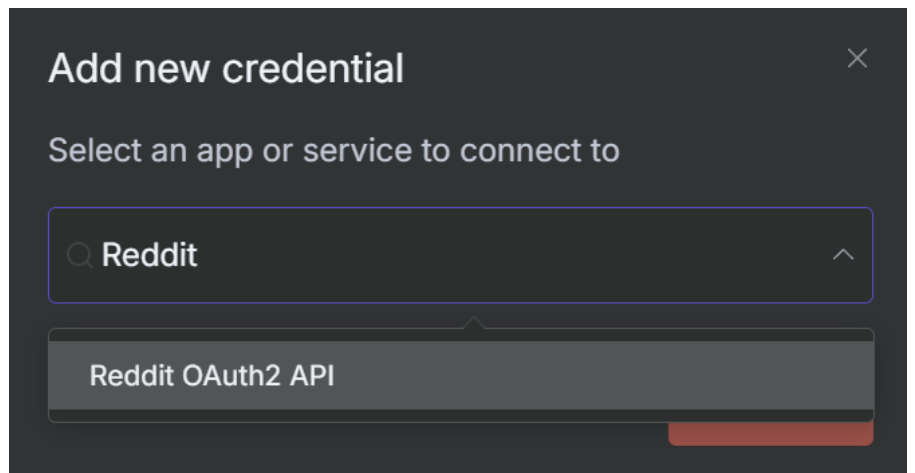
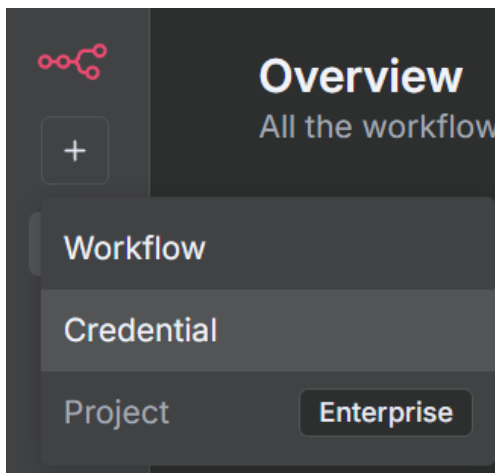
```
docker-compose.yml
services:
  n8n:
    environment:
      # 本地测试
      # cloudflared配置,用于内网穿透
      - WEBHOOK_URL=https://[REDACTED].com/
      - N8N_EDITOR_BASE_URL=https://[REDACTED].com
      #- WEBHOOK_URL=http://localhost:5678
```

然后重启 n8n（复制粘贴下面的命令到 docker terminal）：

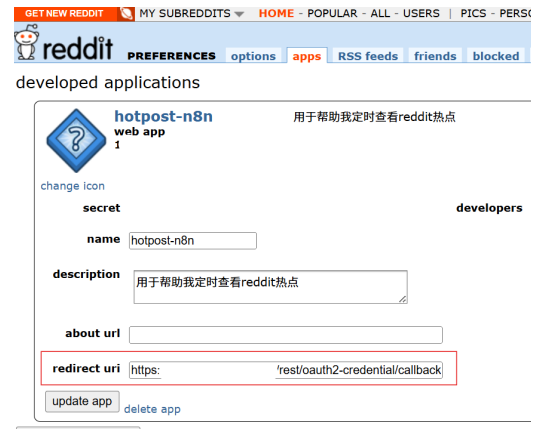
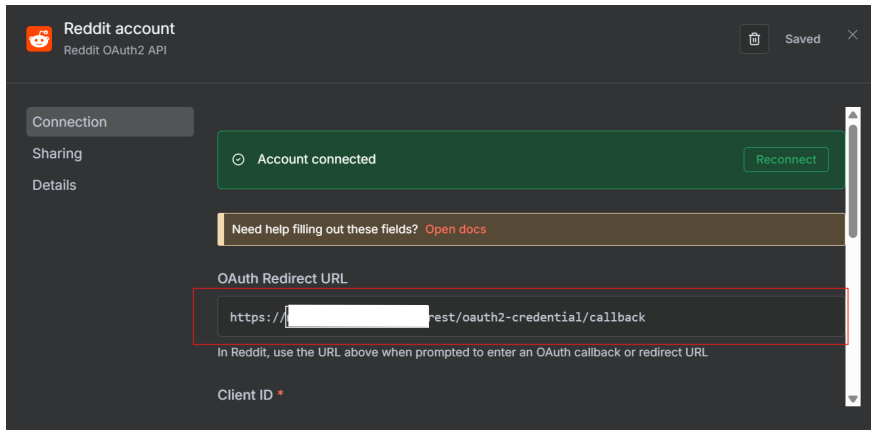
代码块

- 1 docker compose down
- 2 docker compose up -d

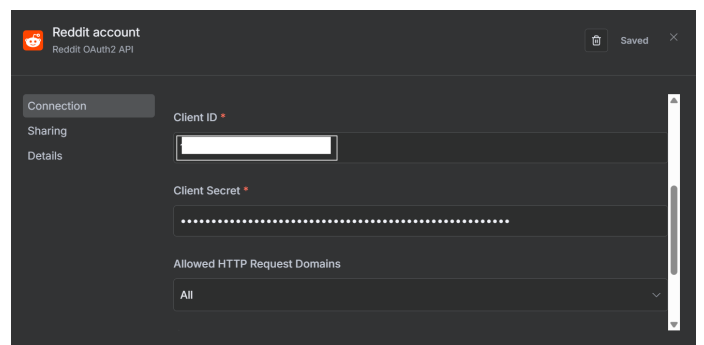
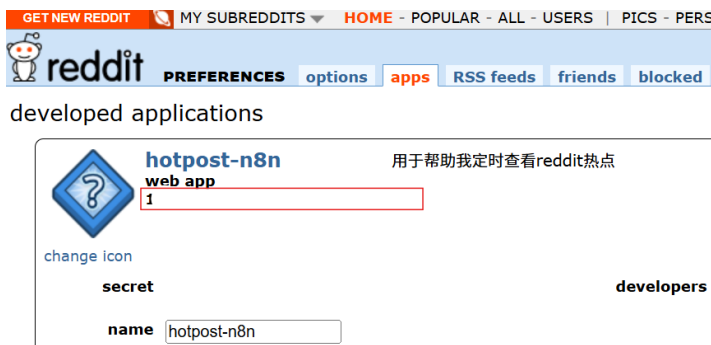
用上面 cloudflare 给出的公网登录 n8n，打开 credential，新建 Reddit account：



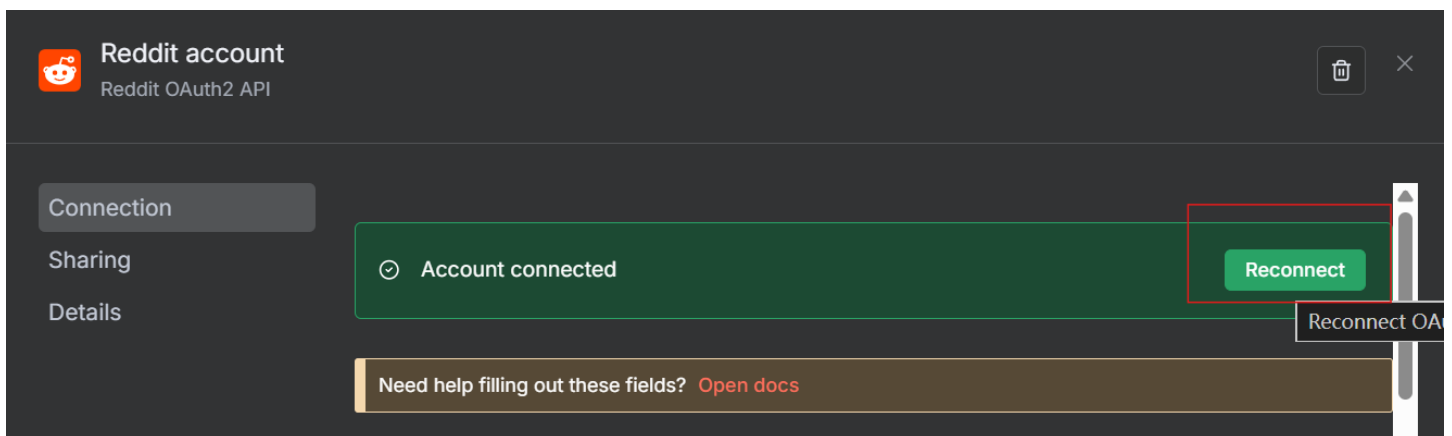
点击新建 Reddit OAuth2 API 后，复制 n8n 里 Reddit credential 配置中 OAuth Redirect URL 粘贴到 Reddit应用的redirect uri：（环境变量配置好后这个URL会自动出现，直接复制粘贴进 Reddit 应用的 uri 就行）



Reddit 应用的 web app id 即 n8n Reddit credential 的 Client ID，Reddit 应用的 web app secret 即 n8n Reddit credential 的 Client Secret：

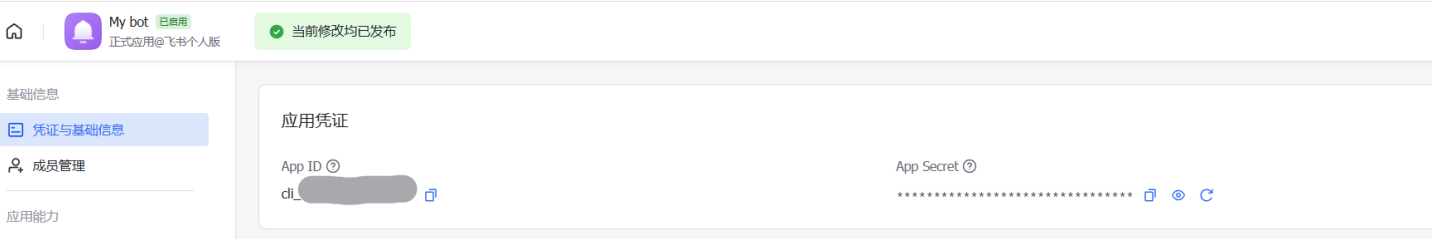


n8n Reddit account 创建完点击 connect，正常情况下会弹出 Reddit 应用的授权页面，点击 allow，n8n 页面显示account connected 之后，Reddit 节点就可以正常在工作流中使用了。（Reddit account 只需用公网 connect 成功一次，后面用 [localhost:5678](#) 打开也可以使用 Reddit 节点。）

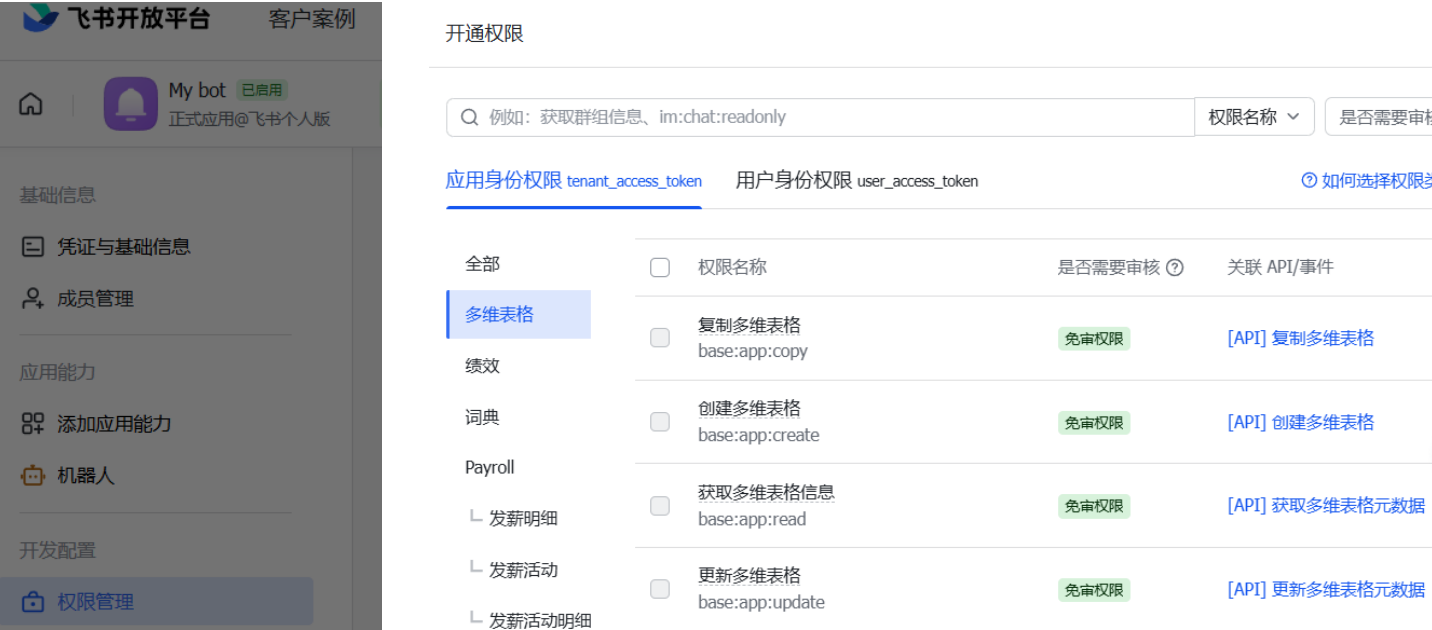


2. 创建飞书应用

在飞书开放平台-开发者后台 <https://open.feishu.cn/app?lang=zh-CN> 创建飞书应用，获取飞书应用的 APP ID 和 APP Secret：



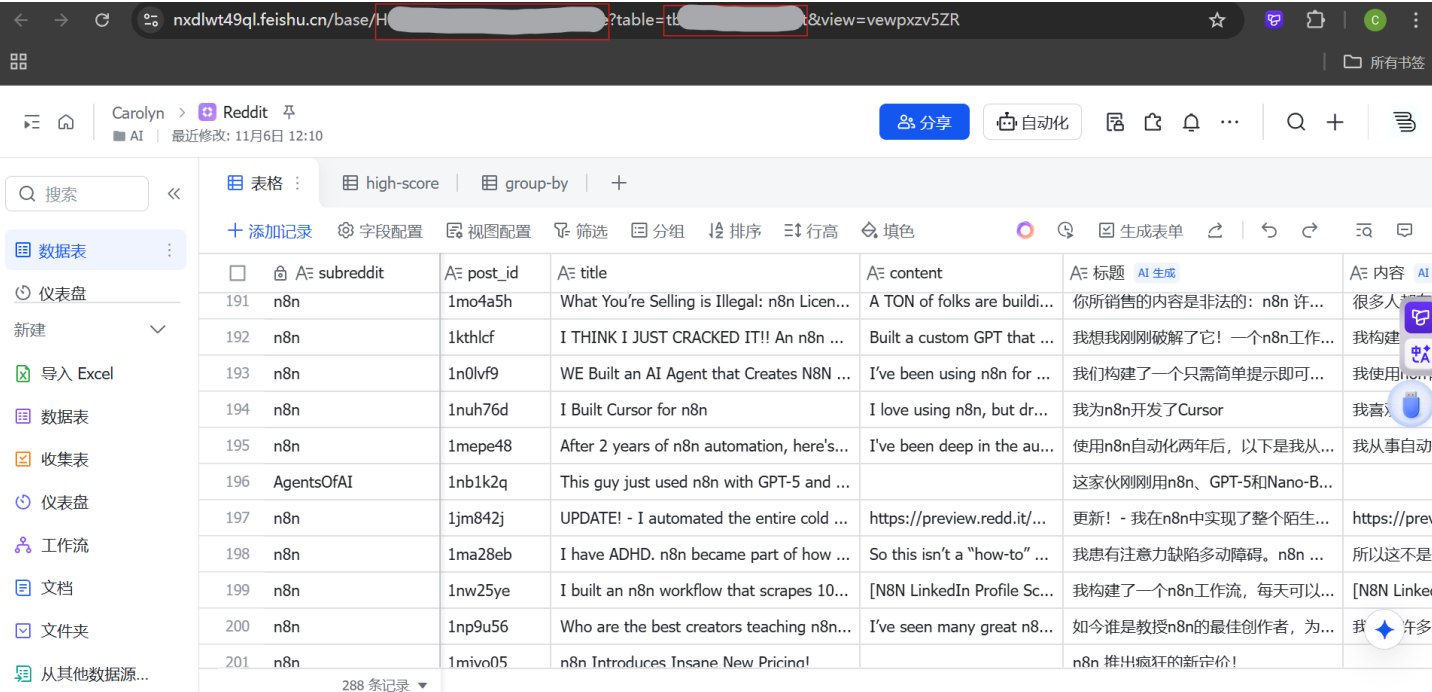
在权限管理中开通飞书多维表的相关权限：



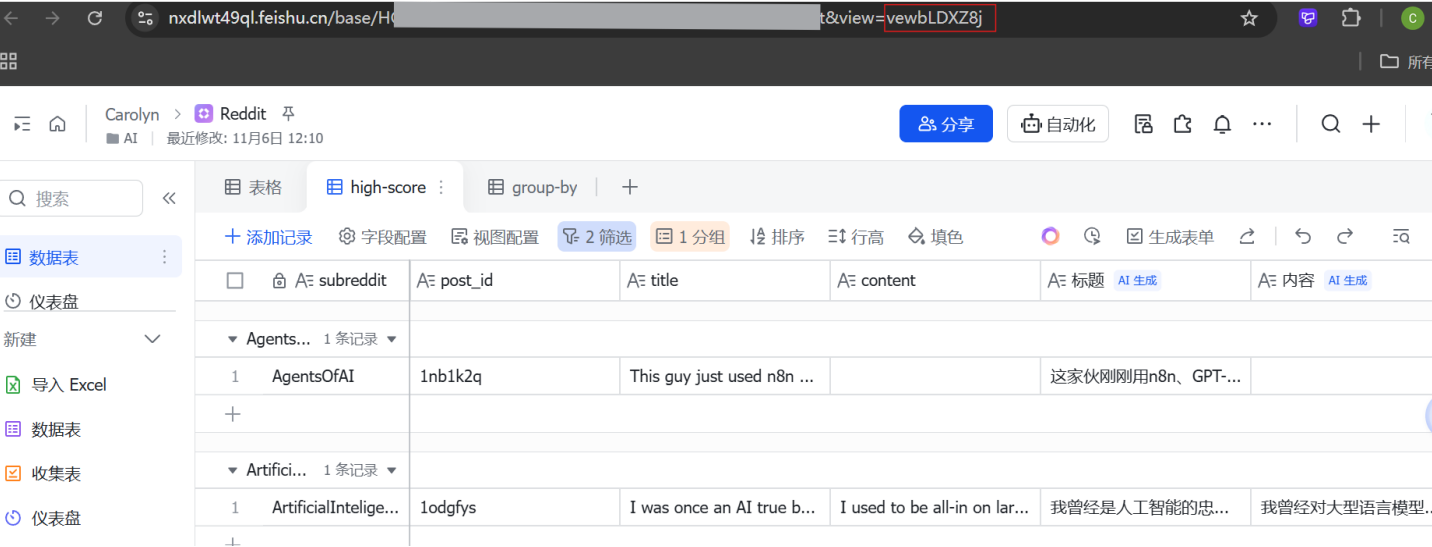
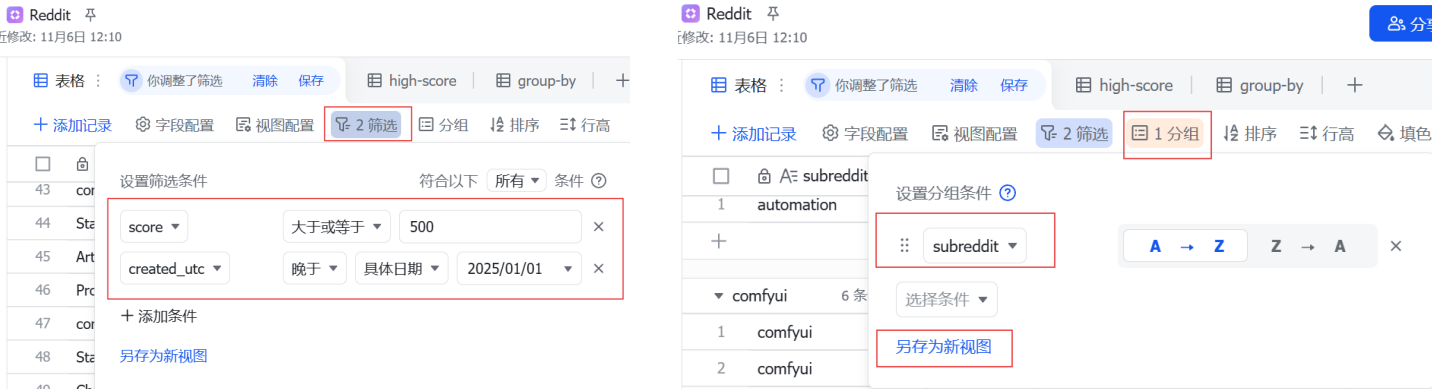
3. 在飞书云文档--云盘新建飞书多维表 命名为 Reddit

注意：是在飞书云盘，不是知识库，知识库建立飞书多维表的 APP Token 和 Table ID 的获取方式不一样。

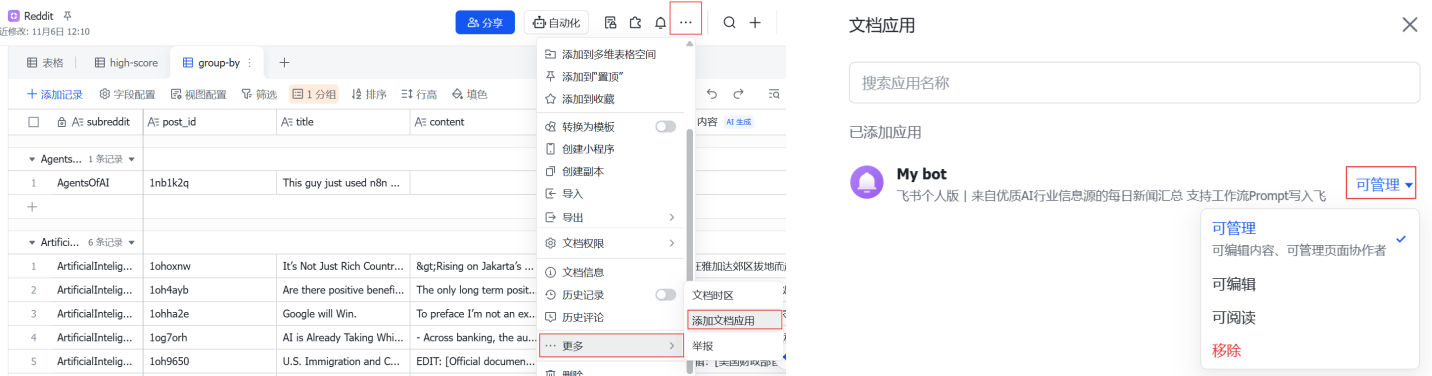
用浏览器打开 Reddit 飞书多维表格，获取 Reddit 飞书多维表的 **APP Token**（如图 H...）和 **Table ID**（如图 tb...）填入 env.example 文件对应的变量中：



把原始表格按照 score大于等于500 和 created_utc 晚于 2025/01/01 筛选，并根据 subreddit 分组后，另存为新视图 high-score，获取 high-score 视图的 **view ID**，填入 env.example 文件对应的变量中：（工作流中使用的默认 view ID 是 high-score 视图的 view ID）

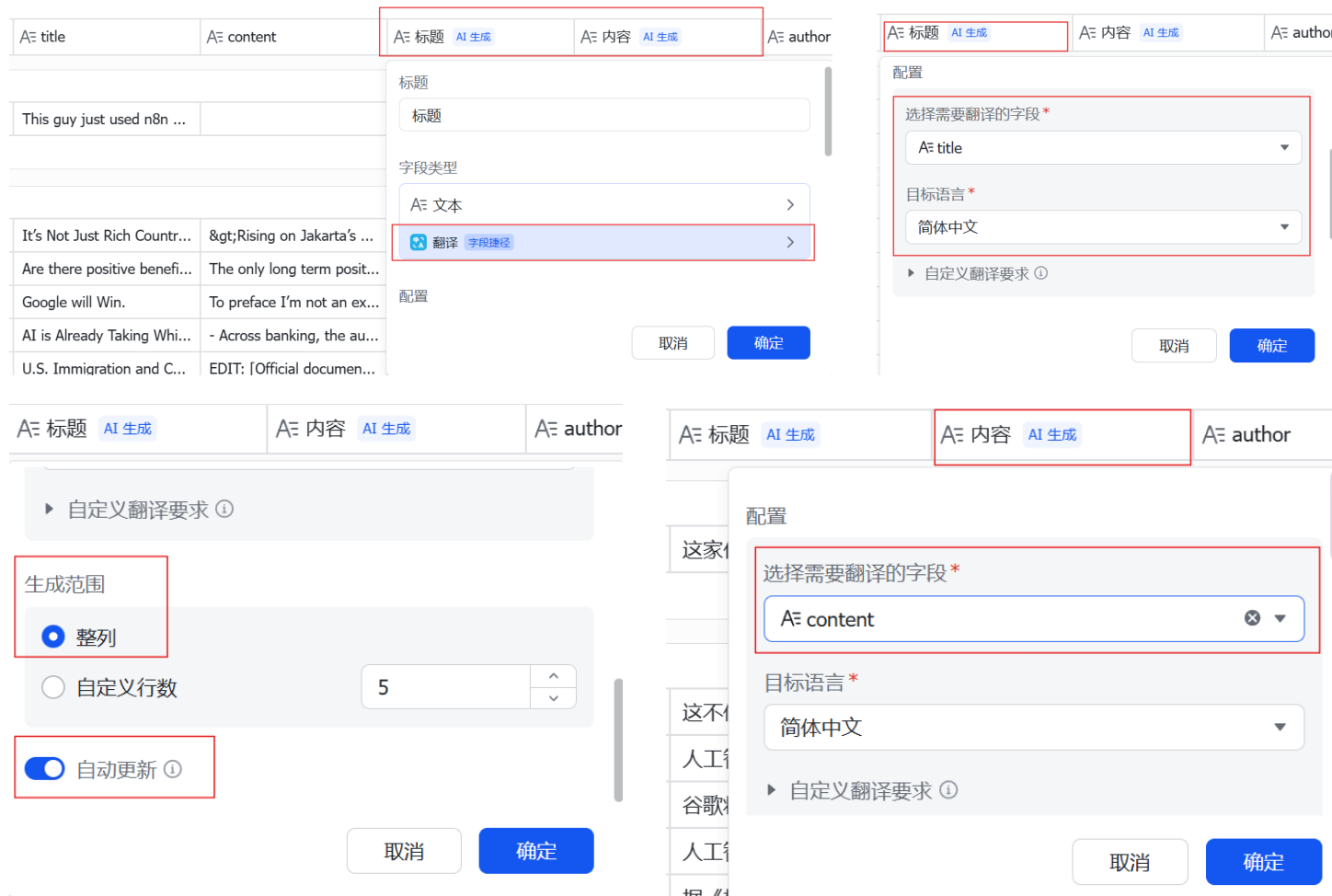


为保证第一个 initial 工作流运行成功，需要给飞书多维表格添加文档应用，在文档应用里搜索你刚刚创建的飞书应用名称（如 My bot），并设置应用的权限为可管理：



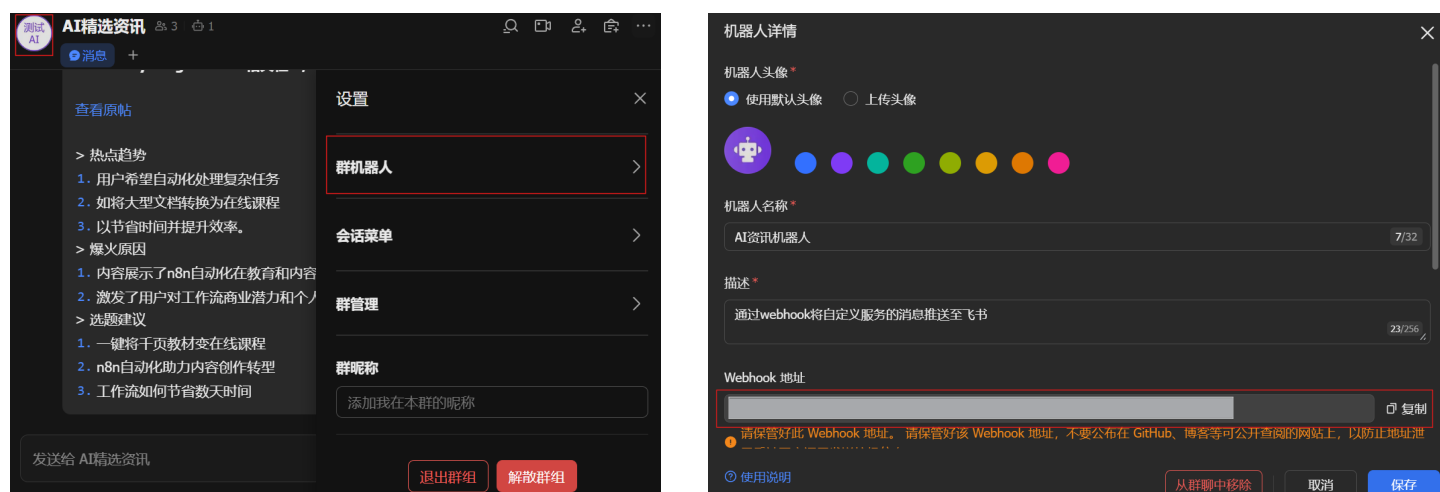
工作流2 写入飞书多维表的都是英文，可以用飞书多维表的AI翻译添加两列，自动翻译标题和帖子内容：

新建“标题”列和“内容”列，在**字段类型**下选择 **字段捷径--AI--翻译**，“标题”配置下选择 翻译 title 字段、简体中文，生成范围--翻译整列、自动更新（“内容”配置下选择 翻译 content 字段，其他设置同“标题”列）



4. 添加群机器人 获取 Webhook URL

为保证工作流3 输出的洞察能发送到飞书群聊中，在飞书群聊中点击头像，点击群机器人，添加群机器人，复制 Webhook URL 并粘贴到 env.example 文件的对应变量中：



5. 配置 DeepSeek API Key

在 DeepSeek 开放平台 https://platform.deepseek.com/api_keys 获取 API key 并充值（¥ 10可以用很久）复制 DeepSeek API Key，粘贴到 n8n 的 DeepSeek credential account 可以在 n8n 中 Authentication 选择 Predefined credential Type，直接使用 DeepSeek account（或者添加到环境变量 .env 文件中，使用时引用环境变量 {{ env.DEEPSEEK_API_KEY }}）：

DeepSeek account

DeepSeek

Connection

Sharing

Details

Connection tested successfully

Retry

Need help filling out these fields?

Open docs

API Key *

Fixed Expression

.....

Parameter: ""

Allowed HTTP Request Domains

All

DeepSeek Rewrite

Execute step

Parameters

Settings

Docs

Method

POST

URL

js

{{ \$env.DEEPSEEK_BASE_URL || "https://api.deepseek.com/v1/chat/completions" }}

https://api.deepseek.com/v1/chat/completions

Authentication

Predefined Credential Type

Credential Type

DeepSeek

DeepSeek

DeepSeek account

Send Query Parameters