資料來源與爬取策略

原則：API 優先 > RSS/站點地圖 > 靜態頁解析 >（在合規前提下）動態渲染截取。嚴守 robots.txt 與網站 TOS，附加識別 UA、節流、指數退避重試。

法遵：不繞過付費牆；不抓取受限內容；不複製大段全文；提供來源鏈接；清楚標示商標/版權。

資料來源 × 方法 × 頻率

| 類別 | 來源 | 方法 | 主要欄位 | 頻率 | 備註 |
| --- | --- | --- | --- | --- | --- |
| Repo | GitHub API | 官方 REST/GraphQL API | stars, forks, topics, lang, readme, releases | 6h | ETag/If-None-Match 減少流量；尊重 rate limit |
| Repo | GitLab API | 官方 API | 同上 | 12h | 公開專案為主 |
| Repo | Hugging Face API | 官方 API | likes, downloads, tags, model card | 6h | 任務型/框架型標籤 |
| Repo | SourceForge | RSS/API/專案頁 | 名稱、描述、下載量 | 24h | 優先 RSS |
| IT 新聞 | Tom’s Hardware / Wired / CNET / 掘金 等 | RSS/站點地圖/靜態頁 | 標題、摘要、作者、時間、分類、連結 | 1–3h | 僅抓公開摘要與連結，不鏡像全文 |
| 學習 | Udemy / Coursera | 公開 Catalog API/合作 Feed | 課程名、講師、評分、連結、價格區間 | 24h | 使用聯盟參數（如允許） |
| 硬體排名 | PassMark | 公開頁/表格 | CPU/GPU 分數、型號 | 24–72h | 僅引用官方可用摘要與連結 |
| DB 排名 | DB-Engines | 公開頁 | Rank, Trend | 24h | 僅摘要與趨勢 |
| LLM 榜 | HF Open LLM Leaderboard | 官方 API/頁 | 模型、分數、日期 | 6h | 僅引用公開指標與連結 |
| 媒體 | YouTube | Data API v3 | 影片 ID、標題、頻道、縮圖 | 6h | 僅存連結與中繼資料 |

# 爬取策略：

1. **API 優先原則**：對於提供 API 的來源（GitHub, Hugging Face, GitLab, Udemy, Coursera 等），應**優先使用官方 API**。這比直接爬取網頁更穩定、更高效，且不易因網站改版而失效。需注意各平台的 API Rate Limit 並妥善處理。
2. **動態網頁爬取**：對於大量使用 JavaScript 動態渲染內容的網站（如 juejin.cn, passmark.com, huggingface.co 的排行榜），使用 Playwright 或 Puppeteer 是合適的選擇。它們能模擬真實用戶行為，獲取最終渲染的 HTML。
3. **靜態網頁爬取**：對於內容相對靜態的網站（如部分新聞網站 tomshardware.com, cnet.com），可以考慮使用更輕量的函式庫如 Cheerio (搭配 axios 或 node-fetch) 進行伺服器端解析，這通常比啟動一個完整的瀏覽器核心更快、資源消耗更少。
4. **混合策略**：建立一個可配置的爬蟲服務，針對不同的目標網站，可以選擇不同的爬取引擎（API, Playwright, Cheerio）。