ChatGPT

StackBurn MVP Integration Platforms – Free API Analysis

Platforms with Free & Accessible APIs (MVP Candidates)

- **Google Drive:** Offers a free API with extremely high quotas (up to **1 billion** requests/day) – effectively no cost barriers for normal use [1]. Using Google's Drive API, StackBurn can list files/folders, fetch metadata (size, last modified), read file contents (via download or export), and even delete files (move to Trash or permanently remove) [2]. **Data access:** full read/write to the user's Drive contents (with OAuth consent), including detecting duplicates and large or stale files. **Blockers/Limits:** Standard Google API rate limit ~**1000 requests/100 seconds per user** by default (very generous) and no fees for moderate usage [3]. In short, Google Drive provides free firepower to scan and purge cloud file clutter without hitting a wall.

- **GitHub:** The GitHub REST/GraphQL API is free to use (no paid plan needed) for any user with a token. **Data access:** StackBurn can connect via OAuth token to list a user's repositories, retrieve repo metadata (last push date, size, stars), list files or search for duplicates in repos, and even automate repo archival or deletion if authorized. **Blockers/Limits:** GitHub imposes a **rate limit of 5,000 requests/hour** for authenticated requests [4] (60/hour if unauthed), which is plenty for scanning dozens of repos. Very large repo trees may require efficient querying, but overall the API is robust. No direct cost – just careful pacing to avoid hitting the 5k/hour throttle. In survival terms, GitHub's free API gives us a sharp axe to chop dead code repos with minimal friction.

- **Supabase:** Supabase provides a free and open API layer (built on Postgres) with **unlimited API calls on the free tier** [5]. **Data access:** With a user-provided service key or connection string, StackBurn can query the **database schema** (tables, row counts) and the **object storage** buckets in a Supabase project. This means we can identify abandoned tables or unused data blobs. We could fetch table metadata, last modified times (if available), or run custom SQL to find tables with no recent updates. **Blockers/Limits:** No hard rate limits on API calls (calls count toward the user's Supabase bandwidth/storage limits, which are generous on free plan) [6]. The challenge is more analytical – e.g. Supabase free tier has limited log retention (no built-in query history beyond a few days), so identifying "no recent read/writes" might require heuristics or user input [7]. Still, Supabase's API is free and accessible; it offers potent cleanup targets (stale tables, orphaned files) for technical users, with the only "cost" being the effort to interpret usage patterns. No paywall, just brainpower.

- **Dropbox:** Dropbox's API is freely available to developers and **designed to handle normal use without fees**. There are rate limits, but they're **"very generous" and only intended to prevent abuse** [8] – in practice, a typical user won't hit them. **Data access:** StackBurn can use the Dropbox API to enumerate all files and folders in the user's Dropbox, get file metadata (size, modified date), download file content for analysis (e.g. to hash for duplicates), and delete or move files. Essentially anything the user can do, the API can do. **Blockers/Limits:** Dropbox doesn't publish exact numbers, but forum guidance suggests limits are per-user and high enough that **"you generally don't need to worry about hitting it in normal use"** [9]. In short, Dropbox integration is a free and viable option – a classic clutter haven (especially for creators' old assets) that we can torch through with bulk file ops. No cost, no nonsense.

- **OneDrive:** OneDrive (including OneDrive for Business via Microsoft Graph API) has free API access with any Microsoft account – **no extra charges** beyond the user's existing subscription (free accounts and O365 accounts both allow API use) [10]. **Data access:** The Graph API lets StackBurn list files in OneDrive, read their content, move or delete them, and get metadata like last modified timestamps. We can scan a user's "Documents" or other OneDrive folders for dupes and stale files just as with Drive. **Blockers/Limits:** Microsoft imposes throttling to prevent abuse, but limits are **generous and handled via back-off** [11]. For example, clients may get a 429 error if pushing too fast, but the solution is simple: pause and retry. No fixed small cap – apps can usually sustain a few hundred requests/minute without issues. The only consideration is OAuth setup (registering a StackBurn app in Azure AD), but that's a one-time hurdle. Overall, OneDrive gives us free API access to another major file silo with minimal rate drama.

- **Local Files (Windows PC):** The user's Windows filesystem itself can be a **target of integration (no external API needed)**. StackBurn can run as a local agent or use an Electron app to scan the Windows file system directly [2]. **Data access:** Full read access to user-designated folders (e.g. the Documents, Downloads, or project directories) and the ability to delete or move files via standard OS calls. Since this is on the user's machine, there are no rate limits or third-party quotas – the only limit is disk scan speed and respecting file permissions. We **exclude macOS** in the MVP to avoid Apple-specific hurdles and because Mac users already have tools like CleanMyMac. Focusing on Windows means covering a huge population of creators/founders who often accumulate "Downloads dump" and "FINAL_v3" files on their PCs. **Blockers:** Ensure we don't scan OS/system folders to avoid messing up the machine; otherwise it's free reign. This integration is essentially free and under our full control – a brutal **local purge** with no gatekeepers. (We just need the user's consent to rummage through their drive, of course.)

- **GitLab (optional alternative):** GitLab's API is also free for users on GitLab.com or self-hosted instances. It provides similar data access to GitHub (listing projects, repository files, commit history). **Limits:** GitLab.com enforces about **10 requests per second per IP** by default [12] (roughly 600 requests/minute), which is quite high. Normal repository scans won't trigger this if optimized. No fees or plan requirements – even free-tier GitLab users can use the API. We might not prioritize it in MVP if our audience skews toward GitHub, but it's a viable integration to keep in the toolkit (no additional survival cost to add when ready).

- **Slack (workspace file cleanup):** Slack offers a free API for bots and apps. **Data access:** StackBurn could connect a Slack app to list files uploaded in a workspace, inspect their timestamps and sizes, and delete files (with appropriate scopes). This could help remove old clutter like outdated PDFs or images lurking in Slack channels. **Blockers/Limits:** Slack's main limitation is data retention on free workspaces – it only **keeps 90 days of history, and deletes files >1 year old** on free plans [13]. That means many "long-term" clutter files might already be gone in free Slack. Paid workspaces keep data indefinitely, and our app could then be useful. Rate limits on Slack's API are moderate (tiered per method, ~**1 request/sec** in many cases), so we'd need to space calls. Also, building a Slack app and having users install it is additional overhead. Slack integration is **free** in terms of cost, but not as immediately impactful for a solo creator's personal clutter (it's more team-focused). It's a candidate to consider down the road once we handle bigger pain points.

- **Trello (project boards):** Trello has a free API with generous limits (**300 requests/10 sec per API key**, and **100 requests/10 sec per user token**) [14]. StackBurn could use it to fetch all boards/cards, looking for boards that haven't been updated in ages or file attachments on cards that are old. **Data access:** Full read/write of boards and cards (we could auto-archive stale cards or even entire boards if user approves). **Blockers:** Trello clutter is more about task overflow than

duplicate files, so it's a slightly different beast. The API is free and open, but this integration might be "nice-to-have" unless many users specifically request Trello cleanup. Technically, it's viable: no cost, high call allowance, and straightforward REST calls.

- **Evernote:** Evernote's API exists and is free for developers to use, but the platform itself has put up **paywalls that make it less accessible**. Evernote recently limited free accounts to **only 50 notes and 1 notebook** max [15] – meaning any user with significant notes has to pay for Personal/Premium to even keep their data. **Data access:** For a paid user, the API allows reading all notes, notebooks, tags, and deleting or updating notes. It's rich: you can pull note content (to detect duplicates or stale notes) and note metadata (created and updated timestamps). **Blockers/Limits:** Evernote imposes an hourly call limit per user (the exact number isn't publicly stated; developers have noted around ~**60 calls/hour** for normal keys, with an initial sync boost for new connections) [16] [17]. That rate limit is relatively low – scanning thousands of notes would require careful pacing (or multiple hours). Combined with the fact that free users are now severely constrained or leaving the platform, Evernote integration is **low priority**. It's technically free to integrate (no API charges), but the ecosystem is shrinking and heavy API limits make it a slow burn. We'll keep it on watch, but not stoke this fire in the MVP.

- **Airtable (and similar cloud databases):** Airtable provides a free REST API for each base; any user (even on the free plan) can use it. **Data access:** StackBurn could list all tables in a base, fetch records (with created/modified timestamps if available), and delete records or even whole tables. This could surface "zombie" tables or unused records in a founder's no-code backends. **Blockers/Limits:** The API is capped at **5 requests per second per base** [18], which is fine but means large bases (thousands of records) take some time to scan. Also, identifying "stale" data might require the base to have a Last Modified Time field or similar; otherwise we only know if a table hasn't changed schema. While **free to use**, Airtable clutter is a niche concern – likely a smaller subset of users. It's an integration we can undertake without paying Airtable, but not mission-critical for MVP.

(The above platforms all provide free API access with no pay-per-use fees. They differ in complexity and relevance, but each could feed StackBurn's cleanup engine with valuable targets.)

Top 3 Integration Picks (Survival Priority)

1. **Google Drive** – *The biggest "digital hoarding" zone for docs and files.* It's a **free API battleground** where our target users accumulate massive clutter. Burning through Drive delivers immediate relief (duplicates, old docs) to overwhelmed creators [19] [20]. This is high-impact, hitting a pain point most users scream about – a critical strike for digital survival.

2. **Windows Local (PC)** – *Where clutter silently breeds for everyone.* Local drives (especially Windows) are **littered with forgotten files** – downloads, "final_final" docs, old screenshots – that sap focus. Direct disk integration is free and gives StackBurn a no-mercy cleanup on the user's home turf. For our users, cleaning their personal machine is a visceral win (no more toxic "Projects" folder) – pure burnination at ground zero.

3. **GitHub** – *The code graveyard of indie hackers.* Free API access, and our audience of solo founders often has dozens of **abandoned repos** [21]. Nuking stale repositories (no commits in X months, trivial toy projects taking up mental space) aligns with StackBurn's mission to cut dead weight. It's a surgical strike: clearing git clutter frees a founder's headspace to focus on live projects.

GitHub integration lets us flex our AI cleanup on code – a differentiator that speaks to the indie hacker survival instinct.

(Each of the above three hits a major clutter source with zero API cost and maximum pain relief – exactly where an MVP should focus its limited ammo.)

Other Platforms to Consider (Future Clutter Hotspots)

Even beyond the MVP's scope, there are other **digital clutter domains** our users struggle with. These are potential integration targets as StackBurn grows – to broaden the "burn radius" and maintain a survival advantage:

- **Dropbox & OneDrive:** Many users have secondary file stashes here (especially creators with media files). Both have free APIs and would be logical next integrations after Google Drive. Old photos, videos, and half-finished docs pile up in these drives too [22]. Supporting them in the future ensures we cover the cloud storage waterfront.

- **Notion (if accessible):** Notion was originally a top target (huge note-taking clutter) [23]. It remains a **known mess zone** of duplicate pages and "idea graveyards" for our audience. *However, Notion's API is gated behind a paid plan – see below.* If that ever changes (or if a chunk of our users are Business-plan Notion power-users), we'd revisit it. The demand is there; we're just waiting for the gate to open to unleash StackBurn on those wikis.

- **Slack:** Team communication can turn into a clutter archive of files and links shared over months. Slack integration (especially for paid workspaces) could help teams **auto-purge old files** eating up space. It's not as personal as other tools, but many indie teams live in Slack and will appreciate a cleanup (once we support more business-oriented use cases in a Pro tier) [22].

- **Trello/Asana (Task Managers):** Overloaded kanban boards and task lists are a form of clutter – old cards, redundant to-dos, projects that died on the vine. In the future, StackBurn might integrate with Trello (and similar PM tools) to identify boards with no activity in X months or cards that can be archived. It's a more specialized cleanup, but valuable for the "productivity junkyard" that busy founders accumulate. Definitely an area to monitor.

- **Email (Gmail)**: Inboxes become landfills of attachments and unopened newsletters. An **email clutter** integration (e.g. scanning Gmail for big attachments or years-old threads) could be a game-changer for some users. This veers outside our current focus (files/notes), but it's on the radar [24]. Many founders would kill for a bonfire of unnecessary emails. Gmail's API is free, so this could be a high-impact future addition if user feedback demands it.

- **Cloud Photos (Google Photos, etc.):** Creators often have bloated photo libraries with duplicates and pointless snaps. While more consumer-oriented, cleaning up photo storage (identifying duplicate images or videos not accessed in years) aligns with our "free space, free mind" ethos. Google Photos has an API for listing and deleting media. This might be a **future frontier** to explore once we nail the core (and if we include more AI for image deduplication).

- **Others (Watchlist):** There are always more tools in a founder's stack – e.g. **Box** (for those with corporate roots), **Confluence/Jira** (if we target teams), or even dev package registries (npm, etc. – code bloat). Our Pro plan can expand to "full multi-platform coverage" [22], adding integrations

wherever digital clutter hides. The above list represents likely culprits to prioritize based on what indie hackers and creators use daily.

Recommended MVP Stack (The First 3 Platforms)

Taking into account impact and feasibility, **StackBurn's MVP should integrate with**: **Google Drive, Windows local files, and GitHub.** These three cover the most critical clutter hotbeds for our target users:

- **Google Drive:** High chaos-to-value ratio (lots of low-hanging clutter) and wide usage among creators/founders. Free API access makes this a no-brainer first target.
- **Windows Local:** Everyone has local bloat, and cleaning a PC provides an immediate "felt" benefit. It's our home turf integration – zero external costs, high psychological payoff.
- **GitHub:** Key for the indie hacker segment – tackles code/projects clutter that general cleaners ignore. It distinguishes StackBurn as the coder's cleanup ally from day one.

By focusing on these platforms, the MVP delivers **brutal clarity and cleanup firepower** where it's needed most, without getting bogged down by paywalls or niche services. This tight trio maximizes value to our users under MVP constraints. (Notion, our original pick, is out – so we replace it with equally valuable targets that won't hamstring us.) These integrations keep StackBurn lean and lethal for v1.0.

Platforms to Exclude for Now (Keep on Radar)

In the spirit of ruthless prioritization, we are **deliberately excluding** a few platforms from the MVP – either due to external constraints or strategic focus. We'll monitor these, but we're not integrating them until they prove worth the effort:

- **Notion – *Excluded due to API paywall*:** Notion's API is only available to Business plan customers [25]. This was a show-stopper for MVP. Our target users often use the free or Plus plans; asking them to upgrade to connect StackBurn is a non-starter. Notion's clutter is very real, but until there's free API access (or a critical mass of our users on paid plans), it stays on the sidelines. We'll keep an eye on Notion in case they change their policy or we decide to offer a premium integration for those who have access. For now, we conserve our resources and **don't fight a battle we can't freely win** on Notion.
- **Evernote – *Excluded due to ecosystem decline*:** Evernote technically has an API, but their recent free tier restrictions (50-note limit) [15] make it a shrinking target. Only legacy or paying Evernote users have significant clutter to clean. Moreover, the API's low rate limits would slow our MVP down. Given the brand's uncertain future and diminishing presence among modern indie creatives, we're not investing MVP effort here. We'll reevaluate if Evernote makes a comeback or if user research finds a surprising number of Premium Evernote hoarders begging for StackBurn.
- **Slack – *Excluded for MVP*:** While Slack was mentioned as a nice integration (and is on our Pro roadmap) [22], it's not core to the **solo creator** use-case right now. Plus, Slack's free plan auto-deletes old data [13], reducing long-term clutter. Integrating Slack would also require building a Slack app and dealing with team data – a complexity better suited for a later version when we

maybe target teams. We won't burn time on Slack in MVP, but we're aware that **team digital junk** (in Slack, etc.) is a frontier for the future.

- **Trello (and similar PM tools) – *On hold*:** Task management clutter is a real thing, but it's not as painful as file/code clutter for an MVP solution. Trello's API is free, yet the value of cleaning up project boards is slightly more abstract (no storage gained, just sanity). We'll hold off on this until we've nailed the bigger wins. Trello, Asana, etc., will remain in our periphery vision – ready to strike if user feedback shows project boards are a significant stress point.

- **Supabase – *Delay for complexity*:** We included Supabase analysis since it aligns with our indie hacker crowd, and **technically it's free to integrate**. However, doing it right (identifying unused tables, unreferenced storage objects) is fairly advanced [7]. It may require deeper database analytics or risk false positives. For MVP, we prefer to avoid half-baked integrations that could misfire. Supabase is flagged for a later iteration (perhaps as a Pro feature) when we can devote more time to safely analyze DB usage. It's a valuable target, just not MVP-easy.

- **macOS Local Files – *Excluded due to focus:*** We're deliberately not doing Mac local scanning in MVP. Mac users already have tools like CleanMyMac, and macOS has its quirks (permissions, different filesystem) – it's a different terrain. Also, many indie devs use Mac, but we chose Windows first as it's **more underserved** in this regard. We'll keep Mac on the roadmap; once we prove the concept on Windows, expanding to macOS (with the lessons learned) will be our next move. It's an ecosystem risk to try to do both from the start – better to dominate one OS now than do a half-job on both.

In summary: StackBurn's MVP will strike at the heart of digital bloat with a trio of free API integrations (Google Drive, Windows PC, GitHub). These give us maximum clutter-clearing ROI for our overwhelmed creators and hackers. Other integrations – from Dropbox to Slack to Notion – are on our radar, but we're smartly reserving those battles for later versions when we have more resources (or when external conditions improve). This survival-driven focus ensures our MVP comes out swinging, delivering **brutal clarity and cleanup firepower** where it matters most, without any platform gatekeepers slowing our burn. [2] [26]

---

[1] [3] Where can I find the price list for Google Drive API? - Stack Overflow
https://stackoverflow.com/questions/14156781/where-can-i-find-the-price-list-for-google-drive-api

[2] [7] [19] [20] [21] [22] [23] [24] Deep Dive_ Designing StackBurn – The AI-Powered Digital Declutter MVP.pdf
file://file-9vp2gqsSmkqfynsKb4aKiD

[4] A Developer's Guide: Managing Rate Limits for the GitHub API
https://www.lunar.dev/post/a-developers-guide-managing-rate-limits-for-the-github-api

[5] [6] Pricing & Fees | Supabase
https://supabase.com/pricing

[8] [9] Limits on Dropbox API when calling from a single account - Stack Overflow
https://stackoverflow.com/questions/41105196/limits-on-dropbox-api-when-calling-from-a-single-account

[10] [11] Does using the OneDrive REST API come with any costs? - API - Latenode Official Community
https://community.latenode.com/t/does-using-the-onedrive-rest-api-come-with-any-costs/11250

12 Does gitlab API have a rate limit - Stack Overflow
https://stackoverflow.com/questions/38478806/does-gitlab-api-have-a-rate-limit

13 Feature limitations on the free version of Slack | Slack
https://slack.com/help/articles/27204752526611-Feature-limitations-on-the-free-version-of-Slack

14 Rate Limits - Trello - Atlassian Developers
https://developer.atlassian.com/cloud/trello/guides/rest-api/rate-limits/

15 I'm a Free User. What will happen to my existing notes / notebooks? - Evernote General Discussions - Evernote User Forum
https://discussion.evernote.com/forums/topic/149829-im-a-free-user-what-will-happen-to-my-existing-notes-notebooks/

16 Evernote Rules – an update - Spoken Like a Geek
https://www.spokenlikeageek.com/2025/05/14/evernote-rules-an-update/

17 Rate Limits - Evernote Developers
https://dev.evernote.com/doc/articles/rate_limits.php

18 Rate limits - Airtable Web API
https://www.airtable.com/developers/web/api/rate-limits

25 26 Notion's API - A detailed introduction
https://letscountapi.com/blog/notions-api-a-detailed-introduction