

Google News RSS Redirect Links – Reasons and Workarounds

Why Google News RSS Uses Redirect URLs (Not Direct Links)

Google News RSS feed items **do not expose direct publisher URLs**. Instead, each `<link>` is a Google redirect (e.g. `https://news.google.com/rss/articles/CBMi...`) pointing to Google's domain ¹. There are likely both **technical and policy reasons** for this design:

- **Click Tracking & Analytics:** The Google News link acts as a tracking URL (counting clicks and user engagement) before forwarding to the publisher site ². By routing through Google, they can log which stories are clicked.
- **Consistent User Experience:** The intermediate link lets Google decide how to deliver the content. For example, if an Accelerated Mobile Page (AMP) version exists, Google can redirect mobile users to the faster AMP page (the encoded feed link often contains both the original and AMP URLs) ³ ⁴. This ensures users get a version optimized for their device.
- **Scraping and Policy Control:** Because the redirect uses a client-side mechanism (not a standard HTTP 3xx redirect), basic scrapers can't easily bypass it ⁵. This may be intentional to discourage third-party services from mass-harvesting Google News content. It forces clients to either use Google's interfaces or perform extra steps, aligning with Google's terms of service which forbid uncontrolled automated access. In practice, Google News will even attempt to **block "robots" or scripts** that try to fetch the final article through these links ².

In summary, Google's RSS design gives them **more control** – they can log referral traffic and enforce their policies by not handing out raw URLs in the feed.

Can the Original Article URL Be Decoded or Extracted?

Yes, but it's non-trivial. The `CBMi...` portion of the Google News link is essentially **encoded data** (often base64) that represents the target URL (and sometimes an AMP URL) along with some metadata ³. Originally, this encoding could be deciphered offline:

- **Older Format (pre mid-2024):** The path after `/articles/` was base64-encoded binary data containing the article URL. Developers discovered the format: a short prefix, then the length and bytes of the main URL, then the length and bytes of an AMP URL (optional), and sometimes a suffix ³. By decoding the base64 and skipping the prefix/suffix bytes, one could extract the original link. For example, a script could Base64-decode the `CBMi...` string, strip magic bytes `0x08 0x13 0x22` at the start and `0xD2 0x01 0x00` at the end, then read the URL bytes in between ⁶ ⁷. This yielded the publisher's URL in plaintext.
- **New Format (introduced ~July 2024):** Google changed the encoding such that the raw URL is no longer present in the decoded data (likely to further obfuscate it). If the decoded string begins with a

marker like "AU_yqL" (indicating the new scheme), the **original URL cannot be obtained by offline decoding alone** ⁸. In this new scheme, Google now requires an extra lookup: the RSS link's HTML triggers a **client-side script/API call** to retrieve the final URL. In other words, the feed link is now a pointer that needs Google's servers (or a simulation of them) to resolve the true article link ⁹.

Despite these hurdles, it is possible to get the real URL programmatically by emulating what a browser or Google's script does. Several methods have emerged, as described next.

Approaches to Resolve Google News Redirects

Developers and tinkerers have crafted **workarounds to follow or decode** Google News RSS links:

- **HTML Parsing & Internal API Call:** One reliable approach is to fetch the Google News link's HTML and inspect it for embedded data. The page often contains a `<c-wiz>` element or script tag with parameters (`data-p` attribute or similar) that include encoded info about the target ¹⁰. By parsing those and then calling Google's hidden batch API, you can obtain the real URL. For example, an open-source script uses Google's **undocumented batchexecute endpoint** (`/_/DotsSplashUi/data/batchexecute`) to resolve the link ¹¹ ¹². The script essentially sends a specially crafted POST request (with parameters extracted from the page) to Google, which returns a JSON containing the article's URL. This technique **mimics Google's own redirect logic** in a headless manner. Both JavaScript and Python implementations of this exist – e.g. a Stack Overflow answer demonstrates using Axios + Cheerio to grab `data-p` from the page and then POST to the batchexecute API to get the final URL ¹³ ¹².
- **Base64 Decoding (Legacy Method):** For older-style links (or when Google News still encoded the URL in the link), some RSS tools directly decode the `CBMi...` string. A community-developed TypeScript function does `atob()` on the path and then extracts the URL bytes after removing known prefixes/suffixes ⁷. This yields the clean article link for feeds that use the older encoding scheme. However, as noted, this fails on newer feeds that hide the URL (in those cases the function detects the "AU_yqL" pattern and falls back to the batchexecute call) ⁸.
- **Headless Browser or Automation:** An alternative is using a headless browser (Puppeteer, Selenium, etc.) to **let Google's own JavaScript redirect run**. Essentially, you programmatically open the `news.google.com/rss/articles/...` URL in a browser context and wait for it to navigate to the final page, then grab the final URL. This approach works consistently (since it's exactly what a real user's browser does), but it's heavier and slower. A contributor on Stack Overflow notes that emulating the chain of requests via script is complex, and using Puppeteer/Selenium might be the more straightforward solution in many cases ⁹. The downside is performance and resource usage if you need to do this for many links.
- **Direct HTML Scraping of Meta Refresh:** In some cases (especially older behavior), the Google News redirect page includes the target URL in a simple HTML meta refresh or a snippet of JavaScript. For example, users found that the HTML contained the destination link in plaintext multiple times ¹⁴. A lighter approach is to perform a GET request and then use a regex or DOM parser to extract any `http://` or `https://` substring that looks like the external news link, without executing JS.

Some low-code tools (like Make.com and n8n) leveraged this: they fetch the Google link (with proper headers to avoid consent blocks) and then parse out the final URL by taking the last occurrence of an `http` link in the response ¹⁴ ¹⁵. This can be effective if Google's page still embeds the URL in the HTML (for example, in a `<script>` variable or meta tag).

- **Third-Party RSS Services:** Recognizing this issue, a few services automate the cleanup. FiveFilters' Feed Control, for instance, can take a Google News RSS feed and **rewrite it with direct source links** by internally resolving each Google URL ¹. Similarly, projects like **morss.it** and community workflows (e.g. an n8n workflow template) exist to convert Google's feed into a more standard feed with original URLs. These typically use one of the methods above under the hood (e.g. decoding or automated fetch) to produce the cleaned feed.
- **Libraries and APIs:** The scraping community has also produced reusable tools. For example, a PyPI package *googlenewsdecoder* provides a simple function to decode Google News article URLs to their originals ¹⁶. Underneath, it implements the logic of trying the direct decode first and falling back to Google's API, including handling **rate limiting and proxies** ¹⁷. Such libraries encapsulate the complexity, so RSS aggregators or scrapers can integrate them to seamlessly get final links.

In practice, many developers combine approaches: try quick decoding, if that fails, do a headless or API call. The existence of these solutions shows it is feasible to extract the real URLs – Google's obfuscation can be overcome with careful engineering.

Developer Experiences and Solutions in the Wild

Scrapers initially struggled with these redirect links. Simple attempts like calling `fetch()` or `axios` with redirects enabled would not work – the response would just return the same Google URL, because no server-side 301/302 was present ¹⁸ ⁵. Questions on forums from 2023–2024 confirm the confusion: “*Why can't I get the destination URL? Is it even a redirect?*” ¹⁹. The answer turned out to be that Google's redirect is **JavaScript-driven**, not an HTTP redirect, thus invisible to standard HTTP clients.

Over time, the community reverse-engineered solutions. One **GitHub gist by huksley** documented the binary format and provided a decoder script (later inspiring other tools) ³ ⁷. On Q&A sites, users shared code to programmatically simulate Google's internal request chain ¹³ ¹². These code snippets have been adapted into production scrapers and open-source libraries. For example, the *google-news-url-decoder* project credits huksley's original script and has iterated through multiple releases to keep up with Google's changes ²⁰. By late 2024, developers noted that Google's new encoding required sending a post request to Google's backend, leading to updates that added that capability (and things like proxy support to avoid blocks) ¹⁷.

Some news aggregators have chosen to avoid the headache altogether by using official publisher feeds or other news APIs. But for those who rely on Google News's aggregated feed (since it can gather stories from many sources with one query), these custom decoders are now a common part of the workflow.

Limitations and Risks of Following Google's Redirects

While it's possible to get the final article links, **there are caveats and risks:**

- **TOS and Legal Considerations:** Scraping Google News in an automated way might violate Google's Terms of Service if done excessively or for certain uses. Google provides the RSS feeds openly, but using them to, say, republish content or for heavy commercial use could be against their policies. There's also the aspect of bypassing Google's click tracking intentionally – which Google likely discourages. Caution and moderation are advised (as well as reading Google's usage terms).
- **Rate Limiting and Blocking:** Google monitors traffic to its services. Rapidly resolving many Google News URLs can trigger rate limits – for instance, receiving HTTP 429 “Too Many Requests”. Developers have observed this and responded by throttling requests and using proxy IPs to distribute the load ¹⁷. The need for a `CONSENT=YES` cookie/header (to bypass EU consent pages) is another hurdle – without it, Google may serve a consent interstitial instead of the redirect page ²¹. In short, programmatic access might require human-like headers, delay between requests, and fallback strategies to avoid Google identifying and blocking the client as a bot.
- **Fragility of Undocumented APIs:** The batchexecute approach relies on an internal Google API that is **undocumented and unsupported**. Google could change the request format or endpoint at any time, potentially breaking decoders until they reverse-engineer the new method. In July 2024 they did exactly that – introducing a new scheme that temporarily stumped existing decoders ⁸. Anyone building a pipeline around this should be prepared for maintenance when Google makes changes.
- **Complex Redirect Logic:** Since the redirect isn't a simple 301, if your extraction logic is flawed you might end up not with a clean URL but a Google News page or an AMP link when you didn't want one. For example, earlier encodings carried both the AMP and original URLs; picking the wrong bytes could yield the AMP URL (which is a valid page but maybe not the one you prefer). It's important to correctly parse the data format. Also, if a publisher's link itself has multiple redirects (HTTP redirects on their side), your pipeline might need to handle those too (e.g. some news sites redirect HTTP to HTTPS, etc.).
- **Redirect Loops or Failures:** If implemented incorrectly, a decoder might repeatedly query Google and never reach a result (e.g. if you keep requesting the RSS link expecting a 3xx redirect). Fortunately, known solutions avoid loops by inspecting the content rather than continuously following. But a poor implementation could get stuck or time out. Additionally, heavy use of headless browsers can be resource-intensive and slow, which is a practical limitation if you need to process lots of articles quickly.
- **IP-Based Blocks and Captchas:** In aggressive scraping scenarios, Google may start showing CAPTCHA or other bot challenges. For instance, if one tries to fetch too many links without proper spacing, Google could present a consent or verification page instead of the redirect. Using residential proxies or rotating IPs has been a strategy to mitigate this, as noted by developers when the bandwidth/usage spiked after Google's 2024 change ²². It's a cat-and-mouse game: push too hard, and Google's defenses kick in.

In conclusion, **extracting the real article URLs from Google News RSS is achievable** and widely practiced using the methods above. Google's decision to use redirect links is driven by tracking and control, which in turn makes our lives a bit harder – but the community has answered with clever reverse engineering. If you implement a decoder, do so carefully: respect Google's servers (to avoid bans), and keep an eye out for changes. With proper handling, you can reliably get clean publisher links from the Google News feed ¹ ⁸ and integrate them into your news processing pipeline. Just be aware that you're leveraging an unofficial solution to sidestep Google's intentional indirection.

Sources: Google News RSS documentation and user forums; community guides and code from GitHub and Stack Overflow illustrating Google News URL decoding ¹ ² ³ ⁸ ⁵ ¹⁷, and practical insights from developers who've tackled Google's redirects in scraping workflows ⁹ ¹⁴.

¹ Google News RSS feeds (2024) - FiveFilters.org

<https://www.fivefilters.org/2021/google-news-rss-feeds/>

² Google news RSS working but not showing news preview : r/rss

https://www.reddit.com/r/rss/comments/1gwe5ds/google_news_rss_working_but_not_showing_news/

³ ⁶ ⁷ ⁸ ¹¹ ²² This script decodes Google News generated encoded, internal URLs for RSS items · GitHub

<https://gist.github.com/huksley/bc3cb046157a99cd9d1517b32f91a99e>

⁴ ¹⁸ ¹⁹ Not able to fetch redirect on any Google News links · node-fetch node-fetch · Discussion #1754 · GitHub

<https://github.com/node-fetch/node-fetch/discussions/1754>

⁵ ¹⁴ ¹⁵ ²¹ Scrape Article Content from Redirected Google News Links - How To - Make Community

<https://community.make.com/t/scrape-article-content-from-redirected-google-news-links/34778>

⁹ ¹⁰ ¹² ¹³ typescript - How to Resolve Google News Redirects to Get the Final Article URL Using Axios? - Stack Overflow

<https://stackoverflow.com/questions/79444019/how-to-resolve-google-news-redirects-to-get-the-final-article-url-using-axios>

¹⁶ ¹⁷ googlenewsdecoder · PyPI

<https://pypi.org/project/googlenewsdecoder/>

²⁰ GitHub - SSujitX/google-news-url-decoder: A Python script to decode Google News article URLs.

<https://github.com/SSujitX/google-news-url-decoder>