

DIY Scraping Techniques (Reddit & Community)

Community-Sourced Techniques

Practical techniques from Reddit, HN, and forums — the stuff not in vendor blog posts.

Grid-Scanning / Map Splitting (Essential)

Overcome the **120-result limit** per search by dividing areas into smaller grid cells. Auto-adjust zoom to ~16 per cell. Can use QGIS + Python to generate coordinate grids. Tune tightness to urban density.

```
https://www.google.com/maps/search/{query}/@{lat},{long},16z
```

Used by every serious scraper (gosom, Apify actors, Octoparse). Hexagonal grid sampling is the academic version.

"Search in This Area" Automation

Moving the map and clicking "search in this area" reveals different/more results than initial search. Automate by systematically panning across coordinates.

HTTP-Only / No Browser Approach (50x Cheaper)

Replicate Google Maps' internal protobuf HTTP calls directly — no headless browser. One HN user reduced costs from **\$50K/mo to \$1K/mo**. "Built entirely with cURL, avoiding headless browsers since they're really slow." Google rarely updates their map API structure.

Dual-Architecture Engine

Use high-performance HTTP extraction for bulk data, only spin up headless browsers for dynamic elements. Extracts Booking URL, FID, CID from metadata. Faster than pure Playwright/Puppeteer.

Place ID Pre-Extraction

Get Place IDs first (cheap/fast), then pipe them for direct detail extraction without browser search overhead — data comes back "almost instantly."

CID-Based Direct Access

CID (Customer ID) is a 64-bit decimal that **never changes**, even through rebrand/relocation. More permanent than Place ID (which expires after 12 months).

```
https://www.google.com/maps/place/?cid=<CID_NUMBER>
```

Find CID: URL `?cid=` parameter, or inspect Knowledge Panel for "ludocid".

Reference: [Scrap.io CID Guide](#), [CID Converter](#)

LLM-Powered Query Expansion

Use LLMs to generate search term synonyms to overcome per-query result limits. "dentist" becomes "dental clinic", "oral surgeon", "dental practice", "orthodontist" etc. Multiplies coverage without geographic splitting.

Other Community Tricks

- **Ctrl+S page saving:** Bot performs search, saves entire page, parses offline later
- **Google Search local pack mining:** Harvest Maps data from regular Google SERPs (lower detection risk)
- `udm=1` **parameter:** Returns Google Places tab results — but needs browser context for location
- **Geocoding API grid trick:** Grid area, geocode lat/lng to addresses, dedupe. 80%+ coverage
- **"No website" filter:** Gold mine for web dev agencies — scrape businesses without websites

Sources

- [r/webscraping — Scraping GMaps at Scale](#)
 - [r/SaaS — Dual-Architecture GMaps Scraper](#)
 - [r/webscraping — Google Maps Data Extraction](#)
-

Revision #1

Created 2026-06-05 23:48:42 UTC by Ben Adrian Sarmiento

Updated 2026-06-05 23:48:42 UTC by Ben Adrian Sarmiento