GMB Scraper: User Guide

Version 1.0

Welcome to your new GMB Scraper! This guide will walk you through everything you need to know, from the initial technical setup to strategic use, all in simple, easy-to-understand language.

This powerful tool automates the process of collecting business information from Google Maps, saving you hours of manual work.

Note: This guide assumes you have a basic understanding of how to run commands in a terminal or command prompt.

Sheet Link: GMB Scraper

Table of Contents

- Understanding the Log and Progress Files
- Finding Your Scraped Data

Chapter 4: Strategies, Use Cases, and Customization

- Strategic Use Cases for Your Data
- How to Customize the Scraper
- Maintaining Your Scraper's Health

Chapter 5: How the Magic Happens (A Simple Look at the Script)

- What is Selenium?
- The Journey of a Keyword: The Script's Logic

Appendix: Troubleshooting & Frequently Asked Questions (FAQ)

- Common Problems and Solutions

Chapter 1: Introduction to the Tool

What is this Automation Tool?

Imagine you need a list of all the gyms in New York City. Normally, you would have to search on Google Maps, click on each business, and manually copy-paste their name, address, phone number, and rating into a spreadsheet. This is incredibly time-consuming and prone to errors.

This tool solves that problem.

It is a smart Python script that acts like a highly efficient assistant. You give it a list of search terms (like "gyms in New York City"), and it automatically:

1. Opens a real Chrome browser.
2. Searches Google for your term.
3. Clicks the "More businesses" button to see the full list.
4. Methodically scrapes the details of each business—Name, Rating, Number of Reviews, Phone Number, Address, and more.
5. Saves all this valuable data into a clean, organized Excel file on your computer.

It's designed to behave like a human to avoid being blocked and even has a system to pause and alert you if it encounters a CAPTCHA, so you can solve it and let the script continue its work.

Who is this Tool For?

This tool is designed for anyone who needs to gather local business data at scale. You do not need to be a professional coder to use it, but you should be comfortable following setup instructions.

It's perfect for:

- **Sales Teams:** To build targeted lead lists for cold calling or outreach campaigns.
- **Marketing Agencies:** To conduct competitor analysis and identify market saturation for local clients.
- **Local SEO Specialists:** To gather data on competitors' ratings, review counts, and business categories.
- **Market Researchers:** To analyze the business landscape in different geographical areas.
- **Small Business Owners:** To find potential partners or analyze their local competition.

Key Features at a Glance

- **Fully Automated Scraping:** Just provide your keywords and the tool does the rest.
- **Google Sheets Integration:** Easily manage your keyword lists from a central Google Sheet.
- **Human-like Behavior:** Uses random delays and a real browser profile to minimize the chance of being blocked.
- **Progress Tracking:** The tool remembers which keywords it has completed, so you can stop and resume without scraping the same data twice.
- **Smart CAPTCHA Handling:** If a CAPTCHA appears, the script pauses, notifies you via email, and waits for you to solve it before continuing.
- **Email Crash Alerts:** If a critical error occurs, the script will email you with the details so you can fix it.
- **Rich Data Extraction:** Gathers key business details including Name, Rating, Review Count, Category, Address, and Phone Number.
- **Clean Excel Output:** All data is saved in a ready-to-use `.xlsx` file.

How It Works: A Simple Overview

The process is straightforward. Think of it as giving a set of instructions to a very dedicated robot.

1. **You Provide the Keywords:** You add your search terms (e.g., "plumbers in London", "cafes near me") to a simple Google Sheet.
2. **You Configure the Settings:** In a single file, you tell the script where your project is, how to access your Google Sheet, and where to send email alerts.
3. **You Create a Browser Profile:** You run a one-time setup script to log in to a Google account in a dedicated Chrome profile. This makes the scraper look like a regular user.
4. **The Automation Kicks In:** You run the main script from your terminal.
5. **It Reads and Searches:** The script reads the first keyword from your sheet, opens Chrome, and performs the Google search.
6. **It Clicks and Scrapes:** It finds the "More businesses" link, clicks it, and begins scraping the data from each listing on the page. It then clicks "Next" to continue to the following pages.
7. **It Records Its Work:** After finishing a keyword, it saves it to a progress file so it won't be repeated. All collected data is stored.
8. **It Creates the Final Report:** Once all keywords are processed, the script compiles all the data into a single Excel file named `GMB_Scraped_Data.xlsx` in your project folder.

This cycle repeats for every keyword, ensuring a thorough and automated data collection process.

Chapter 2: Getting Started: The Full Setup Process

This chapter will guide you through the one-time setup of your GMB Scraper. Follow these steps carefully to ensure everything works perfectly.

Prerequisites: What You'll Need

1. **Python 3.8+:** If you don't have it, download it from python.org. During installation, make sure to check the box that says **"Add Python to PATH"**.
2. **A Code Editor:** A program like Visual Studio Code is highly recommended.
3. **A Google Account:** A standard Gmail account is needed for two things: creating a browser profile and sending error notifications.
4. **Project Files:** You should have all the project files (`gmb_scraper.py`, `config.py`, etc.) downloaded in a single folder on your computer.

Step 1: Set Up Your Google Cloud Platform (GCP) Project & API

This is required to allow the script to read keywords from your Google Sheet.

1. **Create a GCP Project:**
    - Go to the Google Cloud Console.
    - Click the project dropdown at the top and click **"New Project"**.
    - Give it a name (e.g., "GMB Scraper Project") and click **"Create"**.
2. **Enable APIs:**
    - Make sure your new project is selected.
    - In the search bar, search for and enable the **"Google Drive API"**.
    - Search for and enable the **"Google Sheets API"**.
3. **Create a Service Account:**
    - In the search bar, go to **"Service Accounts"**.
    - Click **"+ Create Service Account"**.
    - Give it a name (e.g., "sheets-reader") and click **"Create and Continue"**.
    - For the role, select **"Basic" > "Editor"** and click **"Continue"**.
    - Click **"Done"**.
4. **Generate a JSON Key:**
    - Find your newly created service account in the list. Click the three dots under "Actions" and select **"Manage keys"**.
    - Click **"Add Key" > "Create new key"**.
    - Choose **JSON** as the key type and click **"Create"**.

    - A `.json` file will be downloaded to your computer.
5. **Finalize:**
    - Rename the downloaded file to `gcp_credentials.json`.
    - Move this file into your main project folder.

Step 2: Set Up Your Google Sheet

This is where you will list the keywords you want to scrape.

1. **Make a Copy of the Template Sheet:**
    - Open the master template sheet here: GMB Scraper Sheet
    - In the menu, click **File > Make a copy**.
    - Give your copy a new name (e.g., "My GMB Keywords") and save it to your own Google Drive.
2. **Share the Sheet with Your Service Account:**
    - Open your `gcp_credentials.json` file. Find the value associated with `"client_email"`. It will look something like `sheets-reader@...iam.gserviceaccount.com`. Copy this entire email address.
    - Go to your new Google Sheet, click the **"Share"** button in the top right.
    - Paste the `client_email` address into the sharing dialog, give it **Editor** access, and click **"Send"**.
3. **Add Your Keywords:**
    - In the `GMB lists` tab, start adding your keywords to the `Keywords` column.
    - We have included a sample CSV file (`GMB Scraper - GMB lists.csv`) in the project folder so you can see the format.

Step 3: Set Up Your Python Environment

This ensures the script has all the tools it needs to run without interfering with other Python projects.

1. **Open a Terminal:**
    - Navigate to your project folder in your file explorer.
    - In the address bar, type `cmd` and press Enter (on Windows) or right-click and select "Open in Terminal" (on Mac/Linux).
2. **Create a Virtual Environment:**

    - In the terminal, run this command: `python -m venv venv`
    - This creates a `venv` folder inside your project directory.
3. **Activate the Environment:**
    - **On Windows:** `venv\Scripts\activate`
    - **On Mac/Linux:** `source venv/bin/activate`
    - You should see `(venv)` appear at the beginning of your terminal prompt.
4. **Install Required Packages:**
    - With the environment active, run this command: `pip install -r requirements.txt`
    - This will automatically install Selenium, gspread, pandas, and all other necessary libraries.

Step 4: Configure the

This is the central control file for the scraper. Open `config.py` in your code editor and update the following settings.

- `PROJECT_ROOT`: Replace the example path with the **full, absolute path** to your project folder.
    - *How to get it:* In your project folder, right-click on the `config.py` file and select "Copy as path". Paste it and remove the `\config.py` part at the end.
    - *Example (Windows):* `r"C:\Users\YourName\Documents\GMB_Scraper"`
-
- `SHEET_NAME`: The exact name of your Google Sheet (e.g., "My GMB Keywords").
- `ENABLE_EMAIL_NOTIFICATIONS`: Set to `True` to receive email alerts for CAPTCHAs and crashes, or `False` to disable them.
- `SENDER_EMAIL`: The Gmail address you want to send notifications from.
- `SENDER_PASSWORD`: **Do not use your regular Gmail password.** You need to generate an "App Password".
    - Go to your Google Account settings > Security.
    - Enable 2-Step Verification if it's not already on.
    - Click on "App passwords", generate a new password for "Mail" on "Windows Computer", and copy the 16-character password provided. Paste it here.
-
- `RECIPIENT_EMAIL`: A list of email addresses that should receive the alerts. You can add more than one.

Step 5: Create Your Master Chrome Profile (A Crucial First Step!)

This one-time setup creates a dedicated browser profile and logs you into Google, which helps the scraper appear as a real user and avoid blocks.

1. Make sure your virtual environment is active (`(venv)` is visible in your terminal).
2. Run the following command: `python create_master_profile.py`
3. A new Chrome browser window will open. Follow the instructions printed in your terminal:
    - Go to `google.com`.
    - **Sign in** to any Google account.
    - Accept any cookie pop-ups.
    - **Manually close the browser window** when you are done.
4. The script will automatically close after 90 seconds. A new folder named `Chrome-Master-Profile` will now be in your project directory. **Do not delete this folder.**

Step 6: Activating the Scraper

Congratulations! The setup is complete. It's time to run the scraper.

1. Make sure your virtual environment is still active.
2. In your terminal, run the main script: `python gmb_scraper.py`
3. A Chrome window will open and the script will begin its work. You will see progress updates printed in the terminal. Sit back and let the automation do its job!

Chapter 3: Using the Scraper Day-to-Day

Now that the setup is complete, using the tool is simple. Here's how to manage your scraping campaigns.

Your Daily Workflow

1. **Add New Keywords:** Open your Google Sheet and add any new search terms you want to scrape to the bottom of the `Keywords` column in the `GMB lists` sheet.
2. **Activate the Scraper:** Open your terminal, navigate to the project folder, activate the virtual environment (`venv\Scripts\activate`), and run the script: `python gmb_scraper.py`.
3. **Monitor Progress:** Watch the terminal for real-time updates on which keyword is being processed and how many listings have been found.
4. **Check for Alerts:** Keep an eye on your email for any CAPTCHA or error notifications if you have them enabled.
5. **Find Your Data:** Once the script finishes, a file named `GMB_Scraped_Data.xlsx` will appear in your project folder containing all the collected business information.

How to Handle a CAPTCHA Alert

Google sometimes presents a "I'm not a robot" challenge (CAPTCHA) to verify you are human. The script is designed to handle this gracefully.

1. **The Script Pauses:** The script will detect the CAPTCHA and pause its execution.
2. **You Get an Alert:** If email notifications are on, you will receive an email telling you that manual action is required. The terminal will also display a message.
3. **Solve the CAPTCHA:** Go to the open Chrome window that the script is controlling. You will see the CAPTCHA page. Simply solve it as you normally would.
4. **The Script Resumes:** Once you solve it, the script will automatically detect that the CAPTCHA is gone and will resume scraping from where it left off.

Understanding the Log and Progress Files

The script creates two important files to help you understand what it's doing:

- `gmb_scraper.log`: This is a detailed, technical log of every action the script takes. If something goes wrong, the error details will be here.

- `gmb_completed_keywords.txt`: This is a simple list of all the keywords the scraper has already finished. This file ensures that if you stop and restart the script, it won't waste time scraping the same keywords again. To re-scrape a keyword, simply delete it from this file.

Finding Your Scraped Data

All the data collected by the scraper is saved in a single Excel file:

- **File Name:** `GMB_Scraped_Data.xlsx`
- **Location:** In your main project folder.

Each time you run the script, new data is added to the collection, and the Excel file is overwritten with the complete, updated dataset at the end of the process.

Chapter 4: Strategies, Use Cases, and Customization

This tool is powerful, but using it strategically will give you the best results.

Strategic Use Cases for Your Data

- **Targeted Lead Generation:** Scrape keywords like "roofing contractors in Dallas" or "dental clinics in Chicago" to build highly relevant lists for your sales team. The phone numbers are perfect for cold calling campaigns.
- **Market Research:** Analyze a new market by scraping for your business type across multiple cities. You can quickly see the number of competitors, their average ratings, and how established they are (based on review counts). This can help you identify underserved areas.
- **Competitor Analysis:** Create a list of your direct competitors and scrape their GMB listings. This gives you a baseline of their online presence, including their exact business category, address, and customer ratings.
- **Data Enrichment:** If you have a list of company names but lack contact details, you can use their names as keywords to find their phone numbers and addresses.

How to Customize the Scraper

You can tweak some settings directly in the `gmb_scraper.py` file for more advanced control.

- **Change the Scrape Depth:**
    - Open `gmb_scraper.py` and find the line: `MAX_GMB_PAGES_TO_SCRAPE = 10`.
    - You can increase this number to scrape deeper into the results for a keyword, or decrease it for faster, more superficial scrapes.
- **Adjust Timings (Advanced):**
    - Find the `DELAY_CONFIG` dictionary in `gmb_scraper.py`.
    - These values control the random pauses between actions. Increasing them makes the scraper slower but safer.
    - **Warning:** Do not set these values too low, as it can increase the risk of being temporarily blocked by Google.
- **Update Website Selectors (For Experts):**
    - The `serp_selectors.py` file contains the CSS selectors the script uses to find elements on Google's webpage. If Google updates its website design and the scraper stops working, these selectors may need to be updated. This is for advanced users only.

Maintaining Your Scraper's Health

- **Refresh Your Profile:** Over time, your Google login session in the master profile can expire, leading to more frequent CAPTCHAs. If you notice this happening, simply run the `refresh_profile.py` script. It will delete the old profile and guide you through creating a fresh one.
- **Reset Progress:** If you want the scraper to forget all its progress and start from scratch, simply delete the `gmb_completed_keywords.txt` file.

Chapter 5: How the Magic Happens (A Simple Look at the Script)

You don't need to understand code to use this tool, but it can be helpful to know what's happening behind the scenes.

What is Selenium?

Selenium is the core technology that powers this scraper. It's a tool that allows code to take control of a web browser. Instead of just downloading the HTML of a page, Selenium can open a browser, move the mouse, type in a search box, and click buttons—just like a real person. This is why it's so effective at navigating modern websites.

The Journey of a Keyword: The Script's Logic

When you run the script, it follows a very logical, step-by-step process for each keyword in your list:

1. **Wake Up & Read:** The script starts and reads all your settings from `config.py`. It then connects to your Google Sheet and reads the list of keywords.
2. **Check History:** It opens `gmb_completed_keywords.txt` to see which keywords it has already processed.
3. **Pick a Keyword:** It takes the first keyword from your sheet that is *not* in its history file.
4. **Launch Browser:** It opens a new Chrome window using the session data stored in your `Chrome-Master-Profile` folder. This makes it look like you are the one browsing.
5. **Go to Google:** It navigates to `google.com`, finds the search box, and types in your keyword.
6. **Find the "Golden Link":** It scans the search results page for the "More businesses" link and clicks it to access the full GMB list.
7. **Scrape and Paginate:**
    - It waits for the business listings to load.
    - It carefully extracts the data for each business on the page.
    - It then looks for the "Next" page button and clicks it.
    - This loop continues until it either finds no "Next" button or reaches the `MAX_GMB_PAGES_TO_SCRAPE` limit.
8.
9. **Record & Rest:** After finishing all pages for a keyword, it adds the keyword to the `gmb_completed_keywords.txt` file and takes a short, random break to mimic human behavior.
10. **Repeat:** It moves to the next keyword in your list and starts the process all over again.

11. **Final Report:** Once all keywords are done, it takes all the data it has collected and saves it to the `GMB_Scraped_Data.xlsx` file.

Appendix: Troubleshooting & Frequently Asked Questions (FAQ)

Q: I ran the script, but it crashed immediately with a
A: This is almost always an issue with a file path in `config.py`.

- **Check** Make sure this is the full, correct path to your project folder. Use "Copy as path" to be sure.
- **Check** Ensure this file is named exactly that and is located directly inside your project folder.

Q: I'm getting a
A: This means the script can't find your Google Sheet. Check three things:

1. The `SHEET_NAME` in `config.py` exactly matches the name of your Google Sheet file.
2. The `WORKSHEET_NAME` in `config.py` (`GMB lists`) exactly matches the name of the tab inside your sheet.
3. You have correctly shared your sheet with the `client_email` from your `gcp_credentials.json` file, giving it **Editor** permissions.

Q: The script runs, but no data is being scraped and the Excel file is empty.
A: This can happen if Google changes its website layout, breaking the script's ability to find elements.

- Check the `gmb_scraper.log` file. Look for messages like "Could not find 'More businesses' button".
- This is an advanced issue. The selectors in `serp_selectors.py` may need to be updated to match Google's new HTML structure.

Q: I keep getting CAPTCHAs every time I run the script.
A: Your Google session has likely expired or become untrusted. The best solution is to refresh your browser profile.

1. Stop the script if it's running.
2. Run this command in your terminal: `python refresh_profile.py`
3. Follow the on-screen instructions to log in to Google again in the new browser window. This usually solves the problem.

Q: How do I reset everything and start over from scratch?
A: To do a complete reset:

1. Delete the `gmb_completed_keywords.txt` file.
2. Delete the `GMB_Scraped_Data.xlsx` file (if it exists).
3. (Optional) For a full browser reset, delete the entire `Chrome-Master-Profile` folder and run `python create_master_profile.py` again.