Scrape Task List

Question: What is a Scrape Task?

Answer: A Scrape Task is an automated job that collects data from a target platform, such as Instagram.1 It gathers items like followers, posts, hashtags, or content based on a specific location.2 The purpose of a scrape task is to produce structured data that you can use for various purposes, including outreach, analytics, audience building, or storage. Each task runs with assigned bots on a chosen server and method, and it reports its success or failure and the number of items it has collected.

Question: What are scrape tasks in this system?

Answer: In this system, scrape tasks are automated jobs that are used to collect specific data from a target platform. This data can include information such as followers, posts, hashtags, or content related to a particular location.3 These tasks are essential for generating structured data that can be used for things like outreach, analytics, and audience building. Each task operates with assigned bots, a chosen server, and a specified method, and it provides a report on its performance.

Question: Why do I need scrape tasks?

Answer: You need scrape tasks to automatically collect structured data from platforms. This data is crucial for a variety of purposes, such as finding new audiences, conducting market research, or gathering content for your campaigns. Without scrape tasks, you would have to manually collect all the information, which is inefficient and time-consuming. Each task is run with specific bots on a chosen server and provides a detailed report on its outcome.

Question: What is the purpose of scraping jobs?

Answer: The purpose of scraping jobs is to automate the collection of data from a target platform. These jobs are designed to gather specific information like followers, posts, or content related to a hashtag or location. The data they collect is structured and can be used for valuable tasks such as outreach, building an audience, or performing analytics.4 Each job is configured to run with specific bots and a designated server, and it provides a report on its success and the data it has collected.

Question: How do I view the Scrape Task List?

Answer: To view the Scrape Task list, click on the Scraping Task option in the Dashboard sidebar. The Scrape Task Management Table will then be displayed, showing all your existing scrape tasks.

Question: Where can I see my scraping tasks?

Answer: You can see your scraping tasks by clicking Scraping Task on the Dashboard sidebar. This will open the Scrape Task Management Table, which lists all of your current scraping jobs.

Question: How do I open the scrape task table?

Answer: To open the scrape task table, click on the Scraping Task button on the Dashboard sidebar. The Scrape Task Management Table will then load, showing all of your scraping jobs.

Question: Show all scrape jobs?

Answer: To see all your scrape jobs, go to the Dashboard sidebar and click Scraping Task. This will bring up the Scrape Task Management Table, where all of your scrape jobs are listed.

Question: How do I add or make a scrape task?

Answer: To add a scrape task, follow these steps:

1. On the Dashboard sidebar, click **Scrape Tasks**.
2. On the Scrape Task List page, click **Add Scrape Task** (located at the top-right).
3. You will be redirected to a five-step form.
    - **Step 1: Scrape General Settings**
        - **Scrape Name:** Enter a descriptive name (e.g., "Scrape Followers of Competitor").
        - **Service:** Select the target platform (e.g., Instagram).
        - Click **Go to Scrape Type Selection**.
    - **Step 2: Scrape Type Selection**
        - Choose the data type: **By Username** (followers/following), **By Hashtag** (posts/media), **By Location** (geographic data), or **By Keywords** (search terms).
        - Click **Go to Scrape Method & Storage**.
    - **Step 3: Scrape Method & Storage**
        - **Method:** Choose **Browser (UI-based)** for a slower, more realistic approach, or **Android (Mobile API)** for a faster, lightweight option.
        - **Storage:** Select **Local** to save data on the server or **Cloud** for cloud storage.
        - **Select Server:** Assign a server for the task.
        - Click **Go to Account Selection**.

- **Step 4: Account Selection**
  - Select one or more accounts from your existing bot profiles or from logged-in profile servers to perform the scraping.
  - Click **Go to Advanced Settings**.
- **Step 5: Advanced Settings**
  - **Request Limit per Account per Day:** Set a limit (e.g., 100, recommended: 50–150).
  - **Max Threads:** Define the number of parallel scraping threads (e.g., 10).
  - **Start Scraping Upon Save:** Enable this to run the task immediately.
  - Click **Save** to create and start the task.

Question: Where can I find Add Scrape Task?

Answer: To find the option to add a scrape task, first click on Scrape Tasks in the Dashboard sidebar. On the subsequent list page, the Add Scrape Task button is located at the top-right. Clicking it will take you to a multi-step form where you can configure your new scrape job.

Question: How do I create a scrape job?

Answer: To create a scrape job, click Scrape Tasks on the Dashboard sidebar. From the Scrape Task List, click Add Scrape Task. This will open a five-step form to guide you through the setup process. You'll need to configure general settings (name, service), select the data type, choose a scraping method and storage location, assign accounts, and finally, set advanced limits before you save and run the task.

Question: How do I start a scrape task?

Answer: To start a scrape task, you first need to create one. Navigate to the Dashboard sidebar, click Scrape Tasks, and then Add Scrape Task. This will take you to a five-step form where you define the task. In the final step, Advanced Settings, you can enable the Start Scraping Upon Save option. After clicking Save, the task will begin immediately.

Question: How do I configure Scrape General Settings?

Answer: To configure the general settings for a new scrape task, first click Scrape Tasks on the Dashboard sidebar. Then, on the Scrape Task List page, click Add Scrape Task. This will redirect you to a multi-tab form.

In the first tab, **Scrape General Settings**, you'll do the following:

- **Scrape Name:** Enter a descriptive name for your task (e.g., "Scrape Followers of Competitor").
- **Service:** Select the platform you want to target (e.g., "Instagram").

Once you have completed these steps, click **Go to Scrape Type Selection** to proceed.

Question: What is Scrape Name?

Answer: The Scrape Name is a descriptive name you assign to a scrape task to help you identify it later. When you create a new scrape task, you enter this name in the Scrape General Settings tab, which is the first step in the process.

Question: How do I select a service?

Answer: To select a service, you need to create a new scrape task first. Click Scrape Tasks on the Dashboard sidebar, then click Add Scrape Task. In the first tab, Scrape General Settings, you will find the Service field, where you can choose the target platform for your scraping job (e.g., Instagram).

Question: Where do I start when creating a scrape task?

Answer: You start by going to the Scrape Tasks section on the Dashboard sidebar. From there, click the Add Scrape Task button. This will take you to the first tab of the form, Scrape General Settings, where you will name your task and select the service you want to scrape.

Question: How do I choose a Scrape Type?

Answer: To choose a Scrape Type, follow these steps:

1. On the Dashboard sidebar, click **Scrape Tasks**.
2. On the Scrape Task List page, click **Add Scrape Task** (top-right).
3. You will be redirected to a step-by-step form with 5 tabs.
4. In the **Scrape Type Selection** tab, you can choose from these options:
    - **By Username:** To scrape followers or following of a specific user.
    - **By Hashtag:** To scrape posts or media associated with a hashtag.
    - **By Location:** To scrape data based on a geographic location.
    - **By Keywords:** To scrape based on specific search terms or keywords.

Once you have selected a type, you can proceed to the next tab.

Question: What scrape options are available?

Answer: The available scrape options are:

- **By Username:** To scrape followers or following of a specific user.
- **By Hashtag:** To scrape posts or media associated with a hashtag.
- **By Location:** To scrape data based on a geographic location.
- By Keywords: To scrape based on specific search terms or keywords.
  You can find these options in the Scrape Type Selection tab when adding a new scrape task.

Question: How do I scrape followers or hashtags?

Answer: To scrape followers or hashtags, you need to create a new scrape task and select the appropriate type.

1. Go to the Dashboard sidebar and click **Scrape Tasks**, then **Add Scrape Task**.
2. In the **Scrape Type Selection** tab, choose **By Username** to scrape followers or following, or choose **By Hashtag** to scrape posts or media associated with a hashtag.
3. After making your selection, you can move on to the next step to configure your task.

Question: Can I scrape by keywords?

Answer: Yes, you can scrape by keywords. To do so, you must create a new scrape task. On the Scrape Type Selection tab of the task creation form, select the By Keywords option. This will allow you to scrape data based on specific search terms.

Question: How do I set Scrape Method and Storage?

Answer: To set the scrape method and storage, you must be in the Scrape Method & Storage tab of the scrape task creation form. You can reach this form by clicking Scrape Tasks on the Dashboard sidebar and then Add Scrape Task.

In this tab, you will find these options:

- **Method:**
  - **Browser (UI-based):** This method uses a simulated browser session. It's slower but more visually accurate.

    - **Android (Use Mobile Phone):** This method uses API requests, which is faster and lightweight but may be subject to platform restrictions.
- **Storage:**
    - **Local:** This option saves the scraped data directly on the current server or machine.
    - **Cloud:** This option saves the data to cloud storage, but requires that you have a cloud storage service set up beforehand.
- **Select Server:** Here, you assign a server that will handle the scraping task.

After making your selections, click **Next** to continue.

Question: What is the difference between Browser and Android method?

Answer: The Browser method uses a simulated browser session to scrape data, making it slower but more accurate in mimicking real user behavior. The Android method, on the other hand, uses API requests. This method is faster and more lightweight but may encounter more restrictions from the target platform. You can choose between these two options in the Scrape Method & Storage tab when creating a new scrape task.

Question: Where is the scraped data stored?

Answer: The scraped data can be stored in one of two places, depending on your selection in the Scrape Method & Storage tab. You can choose Local to save the data on your current server or machine, or you can choose Cloud to save it to a cloud storage service, which must be set up in advance.

Question: How do I select a server for scraping?

Answer: You select a server for scraping in the Scrape Method & Storage tab when you are creating a new scrape task. This tab is part of the five-step form you access by clicking Scrape Tasks and then Add Scrape Task on the Dashboard sidebar. In the Select Server field, you can assign a server to handle the scraping job.

Question: How do I select Accounts for a Scrape Task?

Answer: To select accounts for a scrape task, you must be in the Account Selection tab of the scrape task creation form. You can get there by clicking Scrape Tasks on the Dashboard sidebar and then Add Scrape Task.

In this tab, you can choose the accounts that will perform the scraping from one of two places:

- From the list of available bot profiles that are already created in your system.
- From logged-in profile servers, which are accounts that are already linked through servers.
  You can assign one or multiple accounts depending on the scale of your scraping needs. Once you have assigned the accounts, continue to the Advanced Settings tab.

Question: Which bots will run my scrape task?

Answer: The bots that will run your scrape task are the ones you select in the Account Selection tab when creating the task. You can choose from either your available bot profiles or from accounts that are already logged in through your profile servers. You have the flexibility to assign a single account or multiple accounts, depending on how large the task is.

Question: Can I assign multiple accounts?

Answer: Yes, you can assign multiple accounts to a single scrape task. This is done in the Account Selection tab of the scrape task creation form. You can select one or more accounts from your list of available bot profiles or from accounts logged in through your profile servers, which allows you to scale your scraping efforts.

Question: How do I pick profiles for scraping?

Answer: To pick profiles for scraping, you'll need to create a new scrape task and navigate to the Account Selection tab. Here, you'll see a list of your available bot profiles and accounts linked through profile servers. You can choose one or more of these accounts to execute the scraping task, depending on your needs.

Question: How do I configure Advanced Scraping Settings?

Answer: To configure advanced scraping settings, you'll need to be in the Advanced Settings tab of the scrape task creation form. You can get there by clicking Scrape Tasks on the Dashboard sidebar and then Add Scrape Task. In this tab, you can fine-tune the task's performance and limits with these options:

- **Request Limit per Account per Day:** This sets a daily limit on the number of requests each account can make (e.g., 100). A recommended range is 50–150 to avoid account bans.

- **Max Threads:** This option determines the number of parallel scraping threads that will run simultaneously (e.g., 10). A higher number can increase speed but may also increase the risk of detection.
- **Start Scraping Upon Save:** Enable this option if you want the scraping task to begin running immediately after you save it.

Once you have configured these settings, click **Submit** to create and start the task.

Question: What are request limits?

Answer: Request limits are a feature in the Advanced Settings tab of a scrape task that allow you to set a daily maximum on the number of requests each account can make. For example, a setting of 100 means each account will not make more than 100 requests per day. This is a crucial setting to prevent accounts from being flagged or banned by platforms, with a recommended range of 50–150 requests per day.

Question: How do I control scraping speed?

Answer: You can control the scraping speed by adjusting the Max Threads setting in the Advanced Settings tab when creating a scrape task. This setting determines the number of parallel scraping threads that will run at the same time. A higher number of threads can increase the speed of the scraping job, but it may also make the activity more noticeable to the target platform.

Question: Can I start scraping immediately after saving?

Answer: Yes, you can start scraping immediately after saving by enabling the Start Scraping Upon Save option in the Advanced Settings tab of the scrape task creation form. When you activate this setting and click Submit, the scraping task will begin running right away.

Question: What columns appear in the Scrape Task table and what do they mean?

Answer: The Scrape Task table includes the following columns and their meanings:

- **SR No.:** The serial number of the task.
- **Service:** The target platform for the scraping (e.g., Instagram).
- **Name:** The user-defined name for the scrape task.
- **OS:** The operating system used by the task (e.g., Android).

- **Metrics:** The configuration metrics for the task, such as the number of Threads and Requests per Day.
- **Childbots:** The list or number of bots assigned to the task.
- **Type:** The type of scrape being performed (e.g., username, hashtag, location, or keywords).
- **Value:** The specific value being scraped (e.g., #fitness, @someuser, or a location ID).
- **Scraped So Far:** The total number of records that have been successfully scraped to date.
- **Bot Status:** The current status of the associated bots (e.g., logged in, idle, active, or challenged).
- **Requests Sent:** The total number of scraping requests that have been issued.
- **Media Downloaded:** The number of media items that have been downloaded by the task.
- **Media Stored:** The number of media items that have been successfully saved to storage.
- **Failed Requests:** A count of the scraping attempts that have failed, which is useful for debugging.
- **Successful Requests:** A count of the successful responses that have been received.
- **Action:** The row-level actions you can perform, such as Edit, Delete, or View details.

Question: What do the table headings mean?

Answer: The table headings in the Scrape Task table mean the following:

- **SR No.:** Serial number.
- **Service:** The platform being scraped.
- **Name:** The name you gave the task.
- **OS:** The operating system running the task.
- **Metrics:** Task configurations like Threads and Requests per Day.
- **Childbots:** The number of bots assigned.
- **Type:** The scrape method (e.g., username, hashtag).
- **Value:** The specific value for the scrape (e.g., @someuser).
- **Scraped So Far:** Total successful records.
- **Bot Status:** The current state of the bots (e.g., active, idle).
- **Requests Sent:** Total requests made.
- **Media Downloaded:** Media items downloaded.
- **Media Stored:** Media items saved.

- **Failed Requests:** Failed attempts count.
- **Successful Requests:** Successful responses count.
- **Action:** Options to edit, delete, or view the task.

Question: Explain fields in scrape tasks table.

Answer: The fields in the Scrape Task table are:

- **SR No.:** Task serial number.
- **Service:** The platform the task is scraping from.
- **Name:** The custom name you gave the task.
- **OS:** The operating system of the task.
- **Metrics:** The task's configuration, like Max Threads.
- **Childbots:** The number of bots running the task.
- **Type:** The scrape type, such as username or hashtag.
- **Value:** The specific value, like a username or hashtag name.
- **Scraped So Far:** The total number of records that have been successfully scraped.
- **Bot Status:** The current status of the bots.
- **Requests Sent:** The total number of requests sent.
- **Media Downloaded:** The number of media items downloaded.
- **Media Stored:** The number of media items successfully stored.
- **Failed Requests:** A count of requests that failed.
- **Successful Requests:** A count of successful requests.
- **Action:** Actions such as editing or deleting the task.

Question: What are Actions for Multiple Scraping Tasks?

Answer: The Actions for Multiple Tasks feature allows you to manage several scraping tasks simultaneously instead of individually. You can apply a single action to multiple selected tasks using the Select an Action dropdown menu.

Steps to Use Actions for Multiple Tasks

1. On the Dashboard sidebar, click the **Resource Creation** tab.
2. From the toggle list, click **Scrape Tasks**.
3. In the Scrape Task List table, check the boxes next to the tasks you want to manage.
4. At the top of the table, open the **Select an Action** dropdown.
5. Choose one of the following actions:
    - **Start selected tasks:** To begin running the chosen scraping jobs.

- **Stop selected tasks:** To pause active scraping jobs.
- **Delete selected tasks:** To permanently remove selected scraping tasks.
- **Generate Report:** To create a report summarizing the selected scraping jobs.
- **Export CSV File:** To download scraped data in CSV format.

6. Click the **Apply Action Button** to execute the selected action.

Question: How do I start or stop multiple scrape tasks at once?

Answer: To start or stop multiple scrape tasks at once, navigate to the Scrape Tasks list via the Dashboard sidebar. Check the boxes next to the tasks you wish to manage. Then, from the Select an Action dropdown at the top of the table, choose either Start selected tasks or Stop selected tasks and click Apply Action Button.

Question: Can I delete several scrape tasks together?

Answer: Yes, you can delete several scrape tasks together. Go to the Scrape Tasks list, select the tasks you want to remove by checking their boxes, and then choose Delete selected tasks from the Select an Action dropdown menu. Finally, click the Apply Action Button to permanently remove them.

Question: Where is the bulk action option for scrape tasks?

Answer: The bulk action option for scrape tasks is located at the top of the Scrape Task List table. After selecting the tasks you wish to manage, you can find the Select an Action dropdown menu and the Apply Action Button there.

Question: How can I export scraped data from multiple tasks?

Answer: To export scraped data from multiple tasks, go to the Scrape Tasks list, select the tasks you want to export, and choose Export CSV File from the Select an Action dropdown. Click Apply Action Button to download the data in CSV format.

Question: Is there a way to generate reports for several scrape jobs at once?

Answer: Yes, you can generate reports for several scrape jobs at once. On the Scrape Tasks list, select the tasks you want to include in the report. Then, from the Select an Action dropdown menu, choose Generate Report and click the Apply Action Button to create a single report summarizing the selected jobs.

Question: Best practices for Scrape Tasks

Answer: The best practices for Scrape Tasks are:

- **Naming:** Use meaningful task names that include the type and target.
- **Limits:** Begin with conservative request limits (50–80 requests per account per day) for new accounts. A safe recommended range is 50-150.
- **Distribution:** Distribute the scraping workload across multiple bots and servers to minimize the risk of being detected.
- **Threads:** Gradually increase the **Max Threads** count. High thread counts can improve speed but also increase risk.
- **Monitoring:** Monitor **Failed Requests** and **Bot Status** daily and investigate any sudden spikes in failures.
- **Storage:** Store raw data and backups, and export CSV files periodically for auditing.
- **Testing:** Always test each new task on a small sample (e.g., 1–2 bots, for a short duration) before you scale it up.

Question: How do I avoid blocks?

Answer: To avoid blocks when running scrape tasks, you should follow these best practices:

- Set **conservative request limits** for your accounts, especially new ones (50–80 requests per account per day is a good starting point, with a safe range of 50-150).
- **Distribute your workload** across multiple bots and servers.
- Increase the **Max Threads** gradually to avoid aggressive behavior that could trigger a block.
- **Monitor your tasks daily** for any signs of trouble, like a spike in failed requests.

Question: What limits should I set?

Answer: When setting limits for scrape tasks, you should start with a conservative request limit per account per day of around 50–80 requests, especially for new accounts. The recommended safe range is 50-150. You should also start with a low number for Max Threads and increase it gradually as you monitor the performance to find a balance between speed and safety.

Question: How to scale scraping?

Answer: To scale scraping effectively, you should:

- **Distribute your workload** across multiple bots and servers. This helps to spread out the activity and reduce the risk of being detected.
- **Gradually increase your Max Threads** count to optimize for speed without triggering bans.
- **Monitor your performance** by regularly checking for failed requests and bot status.
- Use a **test-and-learn approach**, starting with a small sample before you scale to a large number of resources.
- **Store raw data and backups** so that you have a record of your scraping activities.

Question: Troubleshooting common scrape-list issues

Answer: To troubleshoot common scrape-list issues, consider the following:

- **No results:** Verify that the **Type** and **Value** settings are correct. For example, check for correct hashtag spelling or ensure the username is valid.
- **High failed requests:** This often indicates an issue with your setup. Check your proxies, bot login statuses, server connectivity, and whether you've hit platform rate limits.
- **Task stuck:** Check the status of your bots and the server's load. You can also view the server logs for any timeouts or errors that may be causing the task to get stuck.
- **Export/report failures:** Ensure that you have the proper permissions to export data and that the scrape task has actually produced data to be exported.

Question: Why is my task stuck?

Answer: If your scrape task is stuck, you should troubleshoot by checking the status of your bots and the server's load. It's also a good idea to view the server logs to look for any timeouts or other errors that might be preventing the task from proceeding.

Question: Why are no results returned?

Answer: If your scrape task returns no results, first check to ensure that the Type and Value you've entered are correct. A common mistake is a typo in a hashtag or an incorrect username. Double-check these values to ensure they are valid for the target platform.

Question: What does a high failed requests count mean?

Answer: A high number of failed requests can indicate several problems. You should check the status of your proxies to ensure they are working, verify that your bots are logged in correctly, confirm that you have good server connectivity, and consider whether you might have hit the platform's rate limits, which can cause requests to fail.