Indiegogo Scraper Manual

The Indiegogo Scraper is a tool that enables you to extract data from Indiegogo projects using the Selenium library. This manual will guide you through the necessary steps to set up and use the scraper effectively.

Prerequisites

Before using the scraper, ensure that you have the following prerequisites:

1. Python: Make sure you have Python installed on your system. You can download it from the official Python website: python.org.
2. Required Libraries: Install the necessary libraries by running the following commands in your terminal or command prompt:
   * selenium: Enables automated browser interaction.
   * pandas: Provides data manipulation and analysis capabilities.
   * psycopg2 (optional): Enables integration with PostgreSQL databases.
3. ChromeDriver: Download and install ChromeDriver, which Selenium will use to interact with the Chrome browser. Ensure that you download the appropriate version of ChromeDriver compatible with your Chrome browser version.

Usage

Follow these steps to use the Indiegogo Scraper:

1. Set up the development environment:
   1. Import the necessary libraries in your Python script.
   2. Install ChromeDriver and ensure it is accessible from your script.
2. Initializing the scraper:
   1. Configure Selenium to use ChromeDriver.
   2. Open the Indiegogo trending page as the starting point for scraping. (It will show 12 projects)
3. Loading more projects:
   1. Locate the "Show More" button on the page.
   2. Click the button to load additional projects. (Clicking the button will show 12 more projects)
   3. Repeat the previous step as desired to load more projects.
4. Collecting project links:
   1. Identify the HTML elements containing project links.
   2. Extract the links and store them in a list.
5. Extracting project data:
   1. Create a dictionary or data structure to store the extracted data.
   2. Iterate through the list of project links.
   3. Visit each project page and extract the desired data.
   4. Append the extracted data to the dictionary or data structure.
6. Saving the data:
   1. Choose an appropriate output format, such as CSV or a PostgreSQL database.
   2. Implement code to save the extracted data in the desired format.
7. Close the scraper:
   1. Call the appropriate method to close the browser.

Refer to the documentation and examples provided by the Selenium library for more detailed information on using Selenium and performing specific actions within the scraper.

Extra notes

* When you open the first page of Indiegogo's trending section, you will see 12 projects. Each time you click the "Show more" button, an additional 12 projects will be displayed.
* When extracting the links, our code collects all the links present on the page and stores them in a list.
* Upon visiting a page or following a link, our code extracts relevant data, including the project's status, title, tagline, creator, campaign count, location, funding amount, number of backers, funding percentage, goal type, days left, story, tags, and discussion.
* Although the variables "faq" and "updates" contain textual values that can also be extracted, we have indented them as they are not used in our research.
* We have included code to save the output into a PostgreSQL database. However, this code is indented as it was not utilized in our research. Nevertheless, it is available if needed for future use.
* Some projects use a "comments" tab instead of a "discussion" tab. In these cases, the scraper fills the "discussion" column with "Error." The difference is that commenting on the "discussion" tab requires being a backer, while the "comments" tab only requires having an account. Since our research focuses on backer sentiment, we did not analyze the "comments" tab.
* During the scraping process, the code efficiently extracts data from approximately 220 projects per hour.

Replicability and Scalability:

* The code provided is specifically designed for scraping data from the Indiegogo website. Due to variations in data organization on other websites, the code may not be directly applicable to them. While certain sections of the code can still be utilized, modifications would be required to update the variables and their respective locations for scraping on different websites.
* Additionally, the current Indiegogo code relies on the "show more" button for loading additional content, as opposed to a pagination system with page numbers. If the target website employs a page numbering system, the code would need to be adjusted accordingly to accommodate this functionality.
* To ensure replicability and scalability on different websites, adaptability is crucial. Variables, scraping locations, and pagination mechanisms should be updated accordingly to suit the specific structure and features of the target website.