CASE STUDY

Congruent Global
A Copernicus Consulting Group Company

Automated Scraping of Online Business Presence

Client

Our client, a leading marketing and digital analytics company, increases revenue for businesses through enhanced customer engagement and by shining a spotlight on vast amounts of consumer feedback to deliver insight and actionable intelligence. They serve local businesses, franchises & corporations by providing customized products and services that solve real problems. They partner with businesses to learn what matters most to their business and provide the expert support needed to increase revenue.

Objective

Our client analyses the performance of business by continually monitors the online presence of the business. This includes monitoring various listings of the business on major platforms such as Google, Facebook, Yelp, etc. for location information, online reviews, competitor monitoring etc. This is achieved through continuous scraping on these platforms. The data being scraped is publicly available and not proprietary.

Challenges

The objective of this engagement is to scrape, clean, transform online presence of the business such as business information & online reviews. This information can change any time and online reviews are posted continually by the customers. The main challenge of this engagement is to continuously scrape data to obtain new reviews. The other challenge is that the layout of the site can change anytime

For more case studies please visit https://congruentglobal.com/resources/

Solution

Block detection:

Crawlers will detect the blocks produced while accessing the third-party websites. Block detection is done by identifying the DOM elements or HTTP error codes produced by the third-party websites with respect to the block pattern.

Control blocked crawlers:

Once a crawler server is flagged as blocked by the third-party website then no further request will be made from that server to that website until that server gets unblocked. A separate process will run in the Crawler server for the configured time interval to check whether the block is released or not.

Avoiding blocks:

- Used different user agent string for each URL requests made from the Crawler server.
- Used stealth package that contains various evasion methods.
- Reduced the fetch rate when the block is frequent.
- Upgraded the technology periodically to support modern websites.
- Created the Crawler servers in different cloud providers with different GEO locations.

Central configuration:

Created a central configuration to perform the following actions,

- To turn OFF / turn ON crawling on all instances.
- To disable crawling for a specific website in a specific cloud provider.
- Update the max fetch rate for all / specific cloud providers.

Crawler management and monitoring:

- Automatically update the crawling rate dynamically based on the load in the DB.
- Identify DOM changes and then turn OFF the crawling process and send notifications.
- Real time crawl rate can be viewed in a dashboard.
- Additional hourly notifications will be sent to the team for diagnosis.

Deployment:

We are using 300+ servers for Crawling. Whenever code changes were made an SVN hook will pull the latest code in all the Crawler servers.
Reduced the infrastructure cost by assigning multiple IP addresses to a single instance.

Web scraping Ethics:

There are questions in general about the ethics of web scraping. Our solution takes care of the following aspects:

- Always scrape publicly available content only i.e., data which is available to users without logging into the website
- Respect the robots.txt file i.e., we won't request the URLs which are disallowed in the robots.txt file of a website.
- We use the publicly available cache sources like Google / Bing / Archive for websites which restricts web crawling completely.

Technology

The framework was written in Python which is a solid choice as a language to build fast, flexible systems. To Crawl websites various technologies like CasperJs and PuppeteerJs has been used.

To Crawl static websites Python and CasperJs has been used. To scrape dynamic and interactive websites PuppeteerJs is used.

To avoid block detection, we are using stealth plugin in PuppeteerJs. This plugin has various evasion methods to avoid blocks.

Benefits

By extending the system with robust configurations, Congruent was able to automate crawling and its maintenance tasks. We were also able to extend the solution with additional features to cover a wider range of data sources by ingesting data from partner sources. And by keeping the code in a centralized repository, crawlers are updated timely and only the latest code is executed. Centralized configuration enabled us to control the system on a matter of seconds.

For more case studies please visit https://congruentglobal.com/resources/