Crawl Data

1. ***Thế nào là Crawl Data***

Crawl Data còn được gọi là quét/cào dữ liệu. Đây là thuật ngữ được sử dụng để chỉ quá trình các công cụ tìm kiếm như Google, Bing,… thu thập dữ liệu từ nội dung của website, ứng dụng,… Những dữ liệu được thu thập có thể bao gồm như text, hình ảnh, html, video,…

2. ***Tìm hiểu các kỹ thuật: Beauty Soup, Scrapy và Selenium***

* **Beauty Soup**

Beauty Soup được phát triển chỉ để quét dữ liệu. Đây là một thư viện quét web Python với các tính năng rất mạnh mẽ để quét dữ liệu không chỉ nhanh mà còn hiệu quả. Công cụ quét web này hoạt động với trình phân tích cú pháp và nó đi kèm với trình phân tích cú pháp HTML trong thư viện. Tuy nhiên, bạn cũng có thể sử dụng nó với một số trình phân tích cú pháp Python khác.

Ngoài ra, Beautiful Soup còn rất tốt để trích xuất dữ liệu từ các trang web HTML và XML cũng như một số ngôn ngữ đánh dấu khác. Sử dụng phần mềm để thu thập dữ liệu đòi hỏi kiến thức về lập trình Python nhưng các quy trình liên quan không phức tạp lắm. May mắn thay, có tài liệu chi tiết về cách sử dụng phần mềm và chúng thậm chí còn bao gồm các tập lệnh và mã Python.

Mặc dù phần mềm được phát triển dưới dạng thư viện Python 2 nhưng nó sẽ tự động được chuyển đổi thành Python 3 sau khi quá trình cài đặt hoàn tất. Bạn có thể cài đặt Beautiful Soup trên máy tính **Linux và Windows**. Tuy nhiên, vì đây không phải là thư viện Python tiêu chuẩn nên việc cài đặt rất kỹ thuật và có thể không dễ dàng đối với người mới bắt đầu.

* **Selenium**

Công cụ này được phát triển để thử nghiệm các ứng dụng web. Tuy nhiên, công dụng của nó đã vượt xa mức đó vì nó có thể xử lý một số tác vụ tự động hóa.

Selen có ba loại – Selenium WebDriver, Selenium IDE và Selenium Grid. Để quét web, bạn có thể sử dụng Selenium WebDriver với các API tự động hóa trình duyệt nổi bật.

Là một công cụ quét web, Selenium có một số thách thức. Điều này bao gồm những khó khăn trong việc vượt qua CAPTCHA, 2FA, tải xuống tệp, kiểm tra hiệu suất, tìm kiếm liên kết, phụ thuộc kiểm tra và mã phản hồi HTTP. Hơn nữa, người dùng Selenium có thể gặp rắc rối với các nền tảng tự động hóa như Gmail, Facebook và các nền tảng email khác.

Selenium là một công cụ tự động hóa web dựa trên trình duyệt. Nó không có sẵn dưới dạng ứng dụng để bạn cài đặt trên PC. Tuy nhiên, nó sẽ hoạt động với các trình duyệt **Google Chrome, Mozilla Firefox, Microsoft Edge, Opera và Safari**. Selenium cũng có một trình duyệt gốc – Selenium Internet Explorer – và nó sẽ hoạt động với các trình duyệt chuyên dụng như **HtmlUnitDriver**.

* **Scrapy**

Trước đây gọi là ScrapingHub, Scrapy là một phần mềm nguồn mở để trích xuất dữ liệu trên bất kỳ trang web nào. Là một phần mềm mã nguồn mở nên rất nhiều người đã đóng góp vào sự phát triển của nó nhưng hiện tại nó đang được quản lý bởi Zyte. Scrapy là công cụ trích xuất dữ liệu được nhiều người lựa chọn vì nó nhanh, mạnh và dễ mở rộng.

Với công cụ quét web này, bạn có thể xây dựng và khởi chạy các trình thu thập dữ liệu web để thu thập dữ liệu các trang web. Bạn có thể lưu trữ những con nhện này trên máy chủ của riêng bạn hoặc lưu trữ chúng trên Đám mây Zyte Scrapy. Dễ dàng mở rộng có nghĩa là bạn có thể thêm nhiều chức năng hơn mà không ảnh hưởng đến bất kỳ tính năng nào hiện có. Phần mềm được viết bằng Python.

Các yêu cầu web bạn gửi bằng Scrapy được lên lịch và xử lý không đồng bộ; trong khi một yêu cầu được gửi đi, một yêu cầu khác có thể được gửi ngay cả khi yêu cầu trước đó vẫn chưa được xử lý hoàn toàn. Đây là một trong những ưu điểm chính của phần mềm. Scrapy có thể được sử dụng trên nhiều hệ điều hành bao gồm **Windows, Mac, Linux và BSD.**

***3. Mục đích sử dụng của các công cụ***

* **Beautiful Soup:**
* Mục đích chính: Beautiful Soup là một thư viện Python mạnh mẽ được sử dụng để phân tích và trích xuất dữ liệu từ mã HTML hoặc XML.
* Sử dụng:
* Trích xuất thông tin từ trang web: Bạn có thể sử dụng Beautiful Soup để tìm kiếm và trích xuất dữ liệu cụ thể từ các trang web. Ví dụ, bạn có thể lấy tiêu đề, mô tả, và các phần tử HTML khác từ một trang web.
* Xử lý dữ liệu: Beautiful Soup cũng có thể được sử dụng để xử lý và chuyển đổi dữ liệu sau khi nó đã được trích xuất từ trang web.
* **Scrapy:**
* Mục đích chính: Scrapy là một framework Python dành cho việc khai thác dữ liệu trên web. Nó cung cấp một cách tổ chức và quản lý quy trình khai thác dữ liệu trên nhiều trang web.
* Sử dụng:
* Khai thác dữ liệu trên nhiều trang web: Scrapy cho phép bạn xây dựng các spider (các chương trình tự động) để tự động khai thác dữ liệu từ nhiều trang web khác nhau và tự động lặp lại quy trình này.
* Lập kế hoạch và theo dõi quá trình khai thác: Scrapy giúp bạn quản lý việc khai thác dữ liệu và theo dõi tiến trình.
* **Selenium:**
* Mục đích chính: Selenium là một công cụ tự động hóa trình duyệt web. Nó không chỉ giúp bạn trích xuất dữ liệu từ trang web mà còn cho phép bạn tương tác với trang web như một người dùng thực sự.
* Sử dụng:
* Tương tác với trang web động: Selenium thường được sử dụng khi trang web chứa nhiều JavaScript và các tương tác người dùng khác nhau. Bạn có thể tự động hoá các thao tác như điều hướng, điền vào biểu mẫu, và lấy dữ liệu động từ trang web.
* Kiểm thử ứng dụng web: Selenium cũng được sử dụng để kiểm tra và kiểm thử ứng dụng web bằng cách tự động tương tác với giao diện người dùng.

Tóm lại, Beautiful Soup được sử dụng chủ yếu cho việc phân tích và trích xuất dữ liệu từ HTML, Scrapy được sử dụng cho việc tự động khai thác dữ liệu trên nhiều trang web, và Selenium được sử dụng để tương tác với trang web động và thực hiện các thao tác như một người dùng thực sự.

**4. Nêu sự khác biệt giữa các công cụ này trong Web Crawl Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **Đặc điểm** | **Beautiful Soup** | **Scrapy** | **Selenium** |
| **Loại công cụ** | Thư viện Python | Framework Python | Công cụ tự động hóa trình duyệt |
| **Mục đích chính** | Phân tích và trích xuất dữ liệu từ mã HTML/XML | Tổ chức và quản lý quá trình khai thác dữ liệu từ nhiều trang web | Tương tác với trang web động và thực hiện các thao tác người dùng |
| **Sử dụng cơ bản** | Trích xuất thông tin từ trang web, xử lý dữ liệu | Xây dựng các spider để tự động khai thác dữ liệu, quản lý việc khai thác | Tự động hoá tương tác với trang web như điều hướng, điền biểu mẫu, kiểm thử ứng dụng web |
| **Công cụ bổ trợ phổ biến** | Requests, lxml | None | None |
| **Độ phức tạp** | Thấp | Trung bình đến cao | Trung bình đến cao |
| **Sử dụng phù hợp** | Trích xuất dữ liệu cố định từ trang web | Khai thác dữ liệu từ nhiều trang web, lặp đi lặp lại quá trình khai thác | Tương tác với trang web động hoặc kiểm thử ứng dụng web |
| **Điểm mạnh** | Dễ sử dụng, tốc độ nhanh | Tổ chức quá trình khai thác, xử lý ngoại lệ | Tương tác đầy đủ với trang web động |
| **Hạn chế** | Không thể tự động điều hướng, không thực hiện các tác vụ động | Có học hỏi và thiết lập phức tạp hơn | Cần một trình duyệt thực sự và chậm hơn |

5. Nêu ra giải pháp chọn lựa các công cụ này trong từng tính huống cụ thể

* **Trích xuất dữ liệu tĩnh từ một số trang web đơn giản:**
* Giải pháp: Beautiful Soup.
* Lý do: Beautiful Soup dễ sử dụng và thích hợp để trích xuất dữ liệu tĩnh từ các trang web đơn giản mà không cần tương tác hoặc định kỳ lặp lại.
* **Khai thác dữ liệu từ nhiều trang web khác nhau:**
* Giải pháp: Scrapy.
* Lý do: Scrapy cho phép bạn tổ chức và quản lý quá trình khai thác trên nhiều trang web, xử lý ngoại lệ và thực hiện các tác vụ khai thác phức tạp.
* **Tương tác với trang web động và thực hiện thao tác người dùng:**
* Giải pháp: Selenium.
* Lý do: Selenium cho phép bạn tương tác với trang web như một người dùng thực sự, điều này hữu ích khi cần thao tác trên các trang web sử dụng JavaScript hoặc khi thực hiện kiểm thử ứng dụng web.
* **Kết hợp cả trích xuất dữ liệu và tương tác với trang web động:**
* Giải pháp: Kết hợp Beautiful Soup hoặc Scrapy với Selenium.
* Lý do: Trong trường hợp cần trích xuất dữ liệu từ trang web và sau đó thực hiện các tương tác động như điều hướng và điền biểu mẫu, bạn có thể kết hợp Beautiful Soup hoặc Scrapy để trích xuất dữ liệu với Selenium để thực hiện các tương tác người dùng.
* **Hiệu suất là yếu tố quan trọng:**
* Giải pháp: Scrapy hoặc Beautiful Soup với nhiều luồng (multithreading/multiprocessing).
* Lý do: Scrapy có thể được cấu hình để chạy đồng thời trên nhiều luồng, và bạn có thể tối ưu hóa hiệu suất bằng cách sử dụng nhiều luồng trong Beautiful Soup để trích xuất dữ liệu từ nhiều trang cùng một lúc.