# CS 410 – Text Information Systems – Tech Review

# Web Crawling Software (Tools/Frameworks/Libraries)

**By Cheerla3 (Mohana Venkata Kalyan Cheerla)**

## Introduction:

This article goes over various Web Crawling Software (Tools/Frameworks/Library) available in the Technical market by describing their various key things including their background, features, pros and cons. The main objective is article creation is to prepare a One Stop Solution for the recently emerged powerful web crawling software packages. This article will be definitely helpful when someone has confusion in choosing the most appropriate crawler to accomplish their Web Crawling/Scraping activities. So, I am going to discuss them below without further delay.

## Body:

### 1. Scrapy:

Scrapy is a fast high-level web crawling and web scraping framework, used to crawl websites and extract structured data from their pages. It is an open source web scraping framework in Python used to build web scrapers. It provides all the tools that are needed to efficiently extract data from websites, process them as needed, and store them in the preferred structure and format. Scrapy has a couple of handy built-in export formats such as JSON, XML, and CSV. It runs on Linux, Mac OS, and Windows systems.

**Features :**

- Built-in support for extracting data from HTML/XML sources using extended CSS selectors and XPath expressions.
- Generating feed exports in multiple formats (JSON, CSV, XML).
- Built on Twisted
- Robust encoding support and auto-detection.
- Fast and powerful.

**Programming Language:** Python

**Pros:**

- Suitable for broad crawling and easy to get started.
- Large developer community.
- Easy setup and detailed documentation.
- Active Community.

**Cons:**

- Does not handle JavaScript natively.
- Does not run in a fully distributed environment natively.
- Cannot be scaled dynamically.

## 2. Apache Nutch:

Apache Nutch is a highly extensible and scalable open source web crawler software project. When it comes to best open source web crawlers, Apache Nutch definitely has a top place in the list. Apache Nutch is popular as a highly extensible and scalable open source code web data extraction software project great for data mining. Nutch can run on a single machine but a lot of its strength is coming from running in a Hadoop cluster. Many data analysts and scientists, application developers, and web text mining engineers all over the world use Apache Nutch. Apache Nutch is a cross-platform solution written in Java.

Apache Nutch also provides extensible interfaces such as Parse and Apache Tika. Nutch has integration with systems like Apache Solr and Elastic Search. It extends its custom functionality with its flexible plugin system which is necessary for most use cases, but you may spend time writing your own plugins.

**Features:**

- Fetching and parsing are done separately by default
- Supports a wide variety of document formats: Plain Text, HTML/XHTML+XML, XML, PDF, ZIP and many others
- Uses XPath and namespaces to do the mapping
- Distributed file system (via Hadoop)
- Link-graph database
- NTLM authentication

**Programming Language:** Java

**Pros:**

- Highly extensible and Flexible system for web crawling.

- Implements search when combined with open source search platforms like Apache Lucene or Apache Solr.
- Dynamically scalable with Hadoop.

**Cons:**

- Difficult to setup
- Poor documentation
- Some operations take longer, as the size of crawler grows

## 3. Heritrix:

Heritrix is a web crawler designed for web archiving, written by the Internet Archive. It is available under a free software license and written in Java. The main interface is accessible using a web browser, and there is a command-line tool that can optionally be used to initiate crawls. Heritrix runs in a distributed environment. It is scalable, but not dynamically scalable. This means you must decide on the number of machines before you start web crawling.

### Features :

- HTTP authentication
- NTLM Authentication
- XSL Transformation for link extraction
- Search engine independence
- Mature and stable platform
- Highly configurable
- Runs from any machine

**Programming Language:** Java

**Pros:**

- Excellent user documentation and easy setup
- Extensible, good performance and decent support for distributed crawls

**Cons:**

- Not dynamically scalable

## 4. Storm Crawler:

StormCrawler is a library and collection of resources that developers can leverage to build their own crawlers. The framework is based on the stream processing framework Apache Storm and all operations occur at the same time such as – URLs being fetched, parsed, and indexed continuously – which makes the whole data crawling process more efficient and good for large scale scraping.

It comes with modules for commonly used projects such as Apache Solr, Elasticsearch, MySQL, or Apache Tika and has a range of extensible functionalities to do data extraction with XPath, sitemaps, URL filtering or language identification.

**Features:**

- Scalable
- Resilient
- Low latency
- Easy to extend
- Polite yet efficient

**Programming Language:** Java

**Pros:**

- Appropriate for large scale recursive crawls
- Suitable for Low latency web crawling

**Cons:**

- Does not support document deduplication.

## 5. Node Crawler:

Nodecrawler is a popular web crawler for NodeJS, making it a very fast data crawling solution. If you prefer coding in JavaScript, or you are dealing with mostly a Javascript project, Nodecrawler will be the most suitable web crawler to use. Its installation is pretty simple too. It features server-side DOM & automatic jQuery insertion with Cheerio (default) or JSDOM.

**Features:**

- Server-side DOM & automatic jQuery insertion with Cheerio (default) or JSDOM
- Configurable pool size and retries
- Control rate limit
- Priority queue of requests
- forceUTF8 mode to let crawler deal for you with charset detection and conversion
- Compatible with 4.x or newer version

**Programming Language:** Javascript

**Pros:**

- Easy Installation.

**Cons:**

- No Promise support.

## 6. HTTrack:

HTTrack is a free and open-source web crawler that lets you download sites. All you need to do is start a project and enter the URLs to copy. The crawler will start downloading the content of the website and you can browse at your own convenience. HTTrack is fully configurable and has an integrated help system. It has versions for Linux and Windows users.

**Features:**

- Multilingual Windows and Linux/Unix interface.
- Mirror one site, or more than one site together.
- Filter by file type, link location, structure depth, file size, site size, accepted or refused sites or filename.
- Proxy support to maximize speed, with optional authentication.

**Programming Language:** C

**Pro**

- Highly configurable for multiple systems

**Cons**

- Not as easy to use as other products.
- May take time to download an entire website.

## 7. Beautifulsoup:

Beautiful Soup is a Python library designed for quick turnaround projects like web scraping. It sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

**Features:**

- Beautiful Soup automatically converts incoming documents to Unicode and outgoing documents to UTF-8.
- Beautiful Soup sits on top of popular Python parsers like lxml and html5lib, allowing you to try out different parsing strategies or trade speed for flexibility.

**Programming Language:** Python

**Pros:**

- Easy to use.
- Light weight

**Cons:**

- Not so many options.

## 8. MechanicalSoup:

MechanicalSoup is a python library that is designed to simulate the behavior of a human using a web browser and built around the parsing library BeautifulSoup. If you need to scrape data from simple sites or if heavy scraping is not required, using MechanicalSoup is a simple and efficient method. MechanicalSoup automatically stores and sends cookies, follows redirects and can follow links and submit forms.

**Features:**

- Lightweight
- Cookie Handlers.

**Programming Language:** Python

**Pros:**

- Easy to use.
- Light weight

**Cons:**

- Not so many options.

## 9. PySpider:

PySpider is a Powerful Spider(Web Crawler) System in Python. It supports Javascript pages and has a distributed architecture. PySpider can store the data on a backend of your choosing database such as MySQL, MongoDB, Redis, SQLite, Elasticsearch, Etc. You can use RabbitMQ, Beanstalk, and Redis as message queues.

**Features:**

- Powerful WebUI with script editor, task monitor, project manager and result viewer
- Supports heavy AJAX websites.
- Facilitates more comfortable and faster scraping.

**Programming Language:** Python

**Pro**

- Very good support to other scripting technologies like Javascript and AJAX
- Very fast.

**Cons**

- No.

## 10. SuperCrawler:

Supercrawler is a Node.js web crawler. It is designed to be highly configurable and easy to use. When Supercrawler successfully crawls a page (which could be an image, a text document or any other file), it will fire your custom content-type handlers. Define your own custom handlers to parse pages, save data and do anything else you need.

**Features:**

- Link Detection : Supercrawler will parse crawled HTML documents, identify links and add them to the queue.
- Robots Parsing : Supercrawler will request robots.txt and check the rules before crawling. It will also identify any sitemaps.
- Sitemaps Parsing : Supercrawler will read links from XML sitemap files, and add links to the queue.
- Concurrency Limiting : Supercrawler limits the number of requests sent out at any one time.
- Rate limiting : Supercrawler will add a delay between requests to avoid bombarding servers.
- Exponential Backoff Retry : Supercrawler will retry failed requests after 1 hour, then 2 hours, then 4 hours, etc. To use this feature, you must use the database-backed or Redis-backed crawl queue.
- Hostname Balancing : Supercrawler will fairly split requests between different hostnames. To use this feature, you must use the Redis-backed crawl queue.

**Programming Language:** Javascript

**Pro**

- Link Detection
- Robots Parsing
- Sitemaps Parsing
- Concurrency Limiting
- Rate limiting
- Exponential Backoff
- Hostname Balancing

**Cons**

- Complexity in usage.

## Conclusion:

Each Crawler has its own advantages and disadvantages. So it is very essential to choose the most appropriate one for the project requirement.

## References:


- https://github.com/brendonboshell/supercrawler
- http://nodecrawler.org/
- http://www.httrack.com/
- https://mechanicalsoup.readthedocs.io/
- https://en.wikipedia.org/wiki/Beautiful_Soup_(HTML_parser)
- https://en.wikipedia.org/wiki/Scrapy
- https://scrapy.org/
- https://docs.scrapy.org/en/latest/
- https://github.com/rivermont/spidy