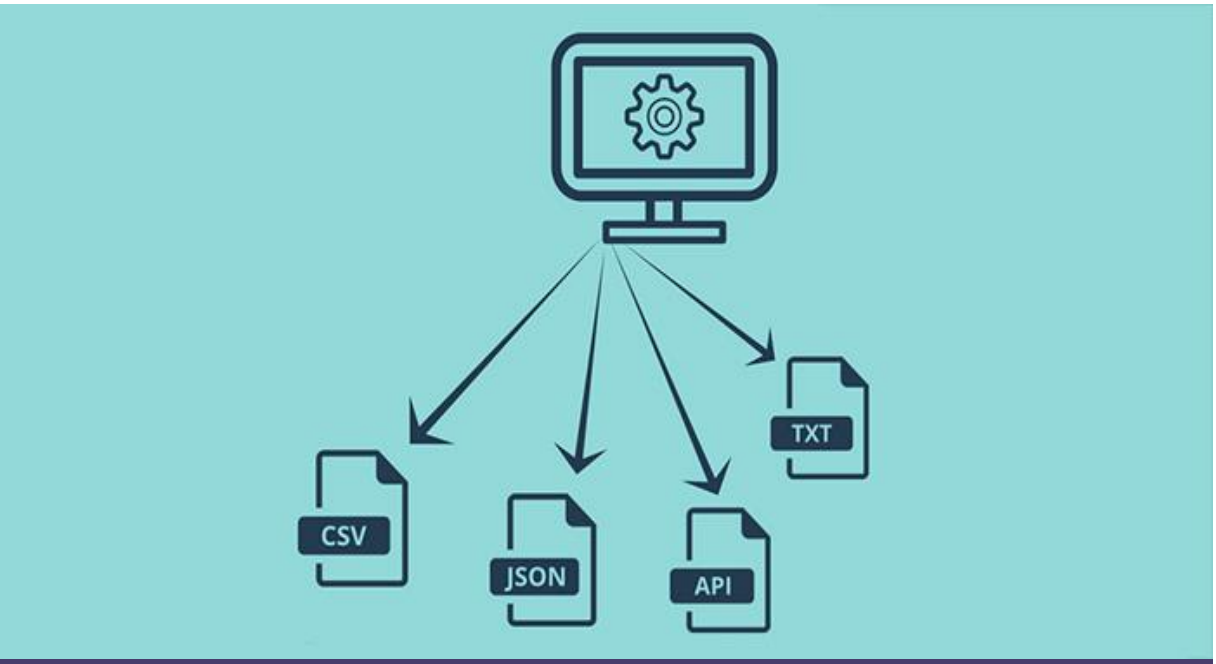


WEB SCRAPING

Node.js

Profesor:
Andrey Alfaro

Estudiante:
Yeth Penado

CETAV
Centro de Tecnología y Artes Visuales

TABLA DE CONTENIDOS

WEB SCRAPING

¿Qué es Web Scraping?

El Web Scraping hace referencia a una técnica para obtener datos de un sitio web. Es un conjunto de acciones ejecutadas por programas de software dirigidas a la extracción de grandes cantidades de datos de sitios web. De esta forma, se extrae el código HTML y con él, los valores almacenados en la base de datos. Gracias a estos programas, se puede automatizar la obtención de información, y hacerlo de un modo rápido, seguro y sin fallos. También se conoce como Web Harvesting o Web Data Extraction.

Entre los tipos de datos que pueden extraerse mediante esta técnica, se encuentran:

- Imágenes.
- URL.
- Direcciones de correo electrónico.
- Números de teléfono.
- Datos de texto (como párrafos).

Similar a su antecesor Web Crawling[1], el Web Scraping va un poco más allá y utiliza sus spiders para extraer la información, y almacenarla en el formato que necesitemos, para su posterior análisis. Estos datos pueden ser almacenados en una computadora y ser guardados como cualquier otro archivo, o bien, ser accedidos de forma continua mediante un API. Entre los formatos en que pueden guardarse estos archivos, están el CSV y JSON.

Algunos conceptos importantes relacionados al Web Scraping son:

- **Screen Scraping:** Es un método para extraer datos de una página web cuando la página está abierta en un navegador.
- **Cloud Scraping:** Utiliza la URL de una página web para extraer datos, sin tener que abrir un navegador web.
- **Captcha:** Diferencia entre un humano y una máquina. Muchos sitios web lo usan para evitar la extracción automática de datos.

[1] Con el objetivo de organizar toda la información disponible en la red, los primeros motores de búsqueda desarrollaron lo que se conocen como web crawlers: bots para rastrear todos los sitios webs existentes.

¿Para qué se utiliza el Web Scraping?

Dado que el propósito del Web Scraping es la extracción de datos, es una práctica muy importante y extendida, principalmente para el mundo empresarial. Algunos usos que pueden dársele al WS son:

- Vigilar de cerca los precios de diferentes productos.
- Analizar datos dados y / o reutilizar los datos para diversos fines estadísticos.
- Detección de cambios en sitios web.
- Obtención de precios para comparadores.
- Aplicaciones para business intelligence.
- Generación de leads[2]/contactos de venta.
- Compra y venta de productos y servicios.
- Dar seguimiento a las tendencias del momento.
- Recopilación de datos para análisis Big Data, Machine Learning (ML) y Artificial Intellingence (AI).

¿Qué resuelve esta tecnología?

En tiempos pasados, la extracción de datos estaba limitada a estadísticas, encuestas y llamadas telefónicas. Con el Web Scraping, esta tarea se automatizó y se unificó en un solo proceso. Ahora, extraer datos de un sitio web es tan sencillo como correr un software que se encargue de realizar el análisis del sitio en cuestión y procese los datos que el cliente necesita. Este último proceso se realiza de forma rápida y sencilla, a diferencia de la forma antigua, donde era necesario invertir en un grupo de personas que realizara la extracción y proceso de datos por sí mismos, lo que resultaba en un proceso mucho más extenso.

¿Quiénes están utilizando esta metodología?

El uso de Web Scraping está muy extendido en la actualidad. Esto es más común en empresas dedicadas al turismo, marketing, compra y venta de productos, bienes raíces, redes sociales, agencias de seguridad nacional y motores de búsqueda, como Yahoo. Es complicado señalar exactamente quiénes están usando esta tecnología, dado su uso extendido y sobre todo, la controversia que existe alrededor de esta práctica. Sin

[2] Un lead es un usuario que ha entregado sus datos a una empresa y que pasa a ser un registro de su base de datos, con el que la organización puede interactuar.

embargo, por mencionar algunos ejemplos específicos, se puede mencionar a grandes compañías que utilizan Web Scraping, como Amazon, Mercado Libre, Wallmart y LinkedIn.

El mejor ejemplo de uso es Google: Lo primero que hacen los bots es leer la petición del usuario y empezar a rastrear las webs que tiene indexadas (registradas) en busca de la que mejor se adapte a su consulta. Cuando encuentra coincidencias entre lo que el usuario quiere saber y el contenido de las páginas web, lo muestra como un listado.

¿Controversia alrededor de Web Scraping: ¿Es legal?

Su uso es legal, sí. El Scraping no supone una ilegalidad en sí mismo, sin embargo, no es ningún secreto que esta técnica pueda utilizarse con fines maliciosos, de ahí surge su mala fama. Sin embargo, hablando en términos legales, esta práctica está avalada y aprobada, salvo en casos específicos (o lugares específicos), dado que las leyes de cada país son distintas, así que hay lugares donde el Web Scraping puede ser más restrictivo que en otros.

Un ejemplo de mal uso de Scraping es el siguiente: Suponer que una empresa desea obtener la lista de precios de una tienda online. Para ello, envía bots a visitar la tienda y extraer los datos. Si envía demasiadas visitas y el servidor de la web en cuestión no puede responder a la demanda, podría incluso hacer que la web no funcione correctamente o no funcione del todo, lo que resultaría en una pérdida de ingresos para el sitio rastreado. A nivel legal, esto incurriría en una demanda por denegación de servicio (DDoS).

En los últimos años, se han llevado casos a las cortes de justicia por casos de Web Scraping. Algunos de ellos fueron LinkedIn vs HiQ, Ryanair vs Atrápalo, y nada como el más célebre de los últimos tiempos, Facebook vs Cambridge Analytica.

Entonces, ¿cómo realizar Scraping en el marco de la legalidad? Algunos marcos éticos a seguir son:

- No utilizar propiedad intelectual o marcas registradas.
- No violar derechos de autor.
- No realizar competencia desleal.
- No sobrecargar a los servidores de los sitios scrapeados.

La legalidad del Scraping no reside en la extracción de datos como práctica. En cada caso hay que saber si es legal el uso que se le dan a esos datos y los métodos que vamos a se van a emplear.

Medidas de protección contra el Web Scraping

El administrador de un sitio web puede usar varias medidas para detener o ralentizar un bot. Algunas técnicas incluyen:

- Bloquear una dirección IP de forma manual o según criterios como la geolocalización. Esto también bloqueará toda navegación desde esa dirección.
- Usar tokens de falsificación de solicitud (CSRF): Al usar tokens CSRF en la aplicación o sitio web, se evita que las herramientas automatizadas realicen solicitudes arbitrarias a las URL de los invitados. Un token CSRF puede estar presente como un campo de formulario oculto.
- Deshabilitar cualquier API de servicio web que el sistema del sitio pueda exponer.
- Los bots a veces declaran quiénes son (utilizando strings de agente de usuario) y pueden bloquearse sobre esa base usando el archivo robots.txt[3] ; ' googlebot ' por ejemplo.
- Los bots pueden bloquearse al monitorear el exceso de tráfico. También se pueden bloquear con herramientas para verificar que es una persona real la que accede al sitio, como un CAPTCHA.
- Servicios comerciales anti-bot: las empresas ofrecen servicios anti-bot y anti-scraping para sitios web.
- Localización de bots con un honeypot u otro método para identificar las direcciones IP de los rastreadores automáticos.
- Ofuscación[4] usando sprites CSS para mostrar datos tales como números de teléfono o direcciones de correo electrónico. Sin embargo, esta técnica sacrifica la accesibilidad para usuarios que utilizan lectores de pantalla.
- Debido a que los bots dependen de la consistencia en el código front-end del sitio web, agregar pequeñas variaciones al HTML - CSS que rodea los datos importantes y los elementos de navegación requeriría una mayor participación humana en la configuración inicial de un bot y, si se hace de manera efectiva, puede generar que el sitio web sea demasiado difícil de scrapear debido a la capacidad disminuida para automatizar el proceso de scraping.

[3] Un archivo robots.txt en un sitio web funcionará como una petición que especifica que determinados robots no hagan caso a archivos o directorios específicos en su búsqueda.
[4] La ofuscación es el acto deliberado de crear código fuente o código de máquina que es difícil de entender para los humanos.

- Los sitios web pueden declarar si el scraping está permitido o no en el archivo robots.txt y permitir el acceso parcial, limitar la frecuencia de rastreo, especificar el momento óptimo para rastrear y más.

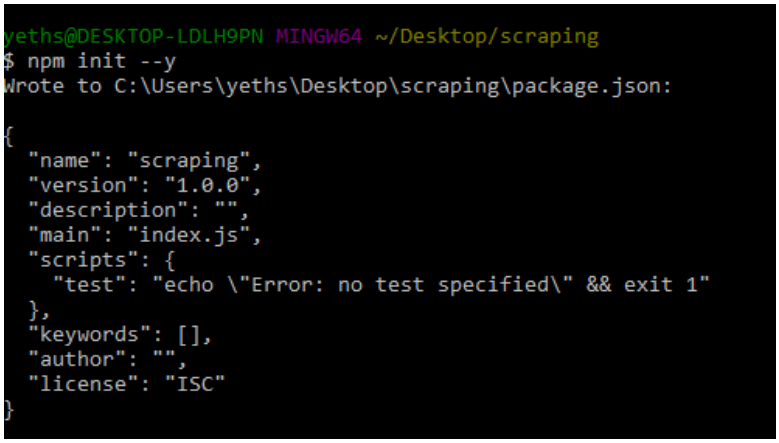
¿Cómo se implementa el Web Scraping en Node.js?

La regla general en Programación, es que no existe una única forma de hacer las cosas, siempre hay caminos distintos para realizar una misma acción. El caso del Web Scraping en Node.js no es la excepción, sin embargo, actualmente la forma más utilizada para scrapear, es el uso de la librería Cheerio. En unas pocas líneas de código, Cheerio extrae los datos que se solicitan de una página web. Esta librería forma parte del core de jQuery, diseñada específicamente para el lado del servidor.

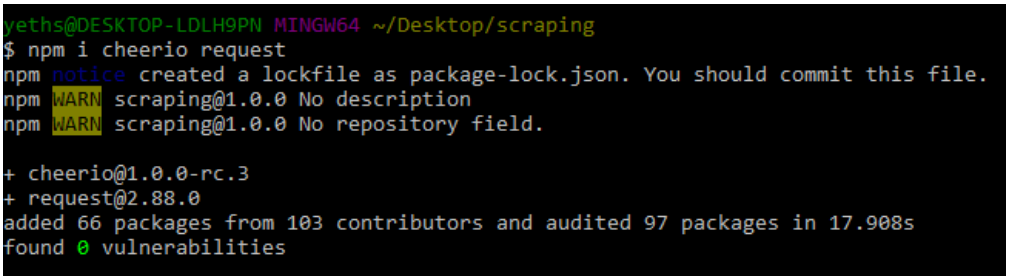
Ejemplo de código

A continuación, se presenta una implementación sencilla de Web Scraping utilizando la librería Cheerio. El sitio web de donde se extraerán los datos es: https://www.parquelalibertad.org/cetav/noticias

1. Inicializar un proyecto con Node. (Comando: `npm init --y`)



2. Instalar las dependencias de Cheerio, con el comando `npm i cheerio request`.



Extracción de datos

3. En el archivo .js se requieren tres variables: la que llama a la Librería (cheerio), el módulo fileSystem (fs) que permite almacenar los datos extraídos en el sistema de ficheros del proyecto y el módulo request, que se encarga de realizar las peticiones GET y HTTP.

```
const cheerio = require('cheerio'); // Llamado a la Librería
const fs = require('fs'); // Permite almacenar en la PC los datos que se deseen analizar
const request = require('request'); // Realiza peticiones GET y HTTP
```

4. Establecer el sitio al que se desea hacer el request, en este caso, será la sección de Noticias del sitio web del Cetav. Comando: `node miscript.js`

```
request("https://www.parquelalibertad.org/cetav/noticias", (err, res, body) => {
  if (!err && res.statusCode == 200) {
    let $ = cheerio.load(body); // Carga y analiza todo el HTML de la página
    $('img', '.view-noticias').each(function() {
      const urlImg = $(this).attr('src');
      images.push(urlImg);
    }); // Crea un array con los datos scrapeados
  }
```

Nota: *Al ser una librería basada en el core de jQuery, utiliza la misma sintaxis que este.*

5. `Let $ = cheerio.load(body);` hace referencia a un método propio de esta librería, en este caso, lo que ejecuta esta línea de código, es la extracción del <body> HTML.

6. `$('img', '.view-noticias')` 'img' hace referencia al tipo de dato que se desea extraer, que en este caso, son imágenes. '.view-noticias' hace referencia al contenedor donde se contienen todas las noticias, por ende, las imágenes también.

7. Con esta data, se genera un Array, que puede ser manipulado según desee el programador.

Almacenamiento de datos a nivel local

8. Para extraer estos datos y almacenarlos en el sistema de ficheros del proyecto:

```
// Recorre los archivos scrapeados y los almacena en una carpeta local
for(let i = 0; i < images.length; i++) {
  request(images[i]).pipe(fs.createWriteStream(`img/${i}.jpg`));
};
});
```

9. Recorre el array donde se guardó la data y guarda los archivos extraídos en una carpeta destinada.

Otras herramientas para realizar Scraping

Es necesario tener en cuenta que el Web Scraping no es algo exclusivo de Node.js; esta práctica lleva realizándose casi desde los inicios de Internet, por lo cual, no es de sorprenderse ver otros lenguajes de programación involucrados en el tema de Scraping.

A continuación, se presentan algunas de las herramientas más populares en lo que respecta a Web Scraping.

Softwares

- Import.io
- Dexi.io
- Octoparse
- Mozenda
- OutwitHub

Lenguajes de Programación

- Python
- Java
- Javascript
- PHP
- R

Plugins

- Web Scraper - Chrome
- Data Scrapper – Chrome
- Scraper – Wordpress
- Crawlomatic – Wordpress

Otras herramientas

- Webhose.io
- ParseHub
- VisualScraper
- FMiner

Conclusión

Utilizado por miles, recelado por otros, el Web Scraping es una técnica que llegó para quedarse. Tomando en cuenta sus ventajas y desventajas, es innegable el poder y el valor diferencial que ofrece esta herramienta a las empresas que la utilizan.

Cada vez más, los datos van siendo de acceso público, subidos y autorizados por los propios usuarios, por lo que un uso legal, ético e inteligente del Web Scraping ofrece grandes ventajas en cuanto a mercadeo se refiere.

Sin embargo, sigue abierto el debate sobre la privacidad y los derechos de los usuarios sobre sus propios datos y hasta qué punto las empresas hacen uso de esta información.

Bibliografía

- https://www.antevenio.com/blog/2019/03/que-es-el-web-scraping-y-para-que-sirve/
- https://www.prowebscraper.com/blog/the-ultimate-guide-to-web-scraping-for-non-programmers/
- https://blog.datary.io/web-scraping-que-es-legalidad-usos-y-el-porque-de-su-valor-diferencial/
- https://www.youtube.com/watch?v=LoziivfAAjE&t=2s
- https://www.techdirt.com/articles/20090605/2228205147.shtml
- https://www.imperva.com/docs/gated/WP_Detecting_and_Blocking_Site_Scraping_Attacks.pdf
- https://www.redeszone.net/2019/03/11/web-scraping-funcion-extraer-datos/
- https://blog.semseoymas.com/que-utilidad-tiene-el-web-scraping-175.htm
- https://papelesdeinteligencia.com/herramientas-de-web-scraping/
- https://diariodeuneletrado.wordpress.com/2017/03/22/es-legal-el-web-scrapping-de-webscrapping-y-legalidad/
- https://www.imperva.com/learn/application-security/web-scraping-attack/
- https://www.kdnuggets.com/2018/09/octoparse-web-scraping.html
- https://www.kdnuggets.com/2018/07/ultimate-list-web-scraping-tools-software.html/2
- https://cheerio.js.org/
- https://stackabuse.com/web-scraping-with-node-js/