DATA ACQUISITION (Webcrawling)

📅 15th JUNE 2022 (WEDNESDAY)
⏰ 1530 - 1730 (MYT)
📍 ONLINE
WEBCRAWLING & WEBSFCRAPPING
WHAT IS WEBCRAWLING & WEBSNAPING

What is web crawler?
process of repetitively finding and fetching hyperlinks starting from a list of starting URLs.

What is web scraping?
Web-scraping is automatically retrieving and processing information from websites.

Unstructured data from websites → WEBCRAWLING & WEBSNAPING → Structured data
BASIC LOGIC OF A CRAWLER

INPUT
- websites
- Bot/spider/crawler
- request

CRAWLER ENGINE
- parser

OUTPUT
- storage
- Extracted data
- CSV, XLSX, TXT, SQL, XML, HTML, JSON

In-house production/
Customized webcrawler

Webcrawling Tools
- botsol
- Web Scraper
- Octoparse
- parsehub

Pros
- Simple and easy to use (coding free)
- Point and click system

Cons
- Paid subscription for full version (free version – function limited)
- Dependencies on built-in scraper
1. Go to the webpage – get the URL of the page
2. Create pagination process to ensure each page will be visited
3. Build a loop for each product/item in a particular page
4. Extract the data for each item
5. Run the task and export/save the data
WEBCRAWLING
DOSM - STATSBDAA
PRICE INTELLIGENCE (PI)

- leveraging the capability of Big Data in collecting large data from various sources and transform them into better structure
- different prices of the same good can be obtained through various websites, providing a modernized price collection
- Transform from unstructured data into structured data to perform analysis

Managing massive data → data management → repository

- Extracting from relevant websites
- Automated web crawler
- Data consolidation on central local database
- Price Analysis
- Visualization
DATA ACQUISITION FLOW - WEBCRAWLING

<table>
<thead>
<tr>
<th>Crawler</th>
<th>Spark</th>
<th>Kafka</th>
<th>HBase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crawling service from a list of URL - store into HTML server</td>
<td>Parsing data from HTML server</td>
<td>Picking up and hold data that has been parse then insert bulk into HBase</td>
<td>Data stores into HBase</td>
</tr>
</tbody>
</table>
MariaDB
- To store all category URL listed to crawl

Pusher
- Act as an engine to send category URL to crawler workers

Beanstalkd
- A simple, fast work queue agent which act to distribute the URL equally to all available workers

Index and Content Crawler
- The main process for data crawling
- Contain workers

SSDB
- Storing big list of billions elements
- Use as a hash storage to prevent multiple crawl of the same page for content/detail page
Index Page

- Crawler will scrap basic information on the index page
  - Title
  - Price
  - URL (this URL will lead to the content/detail page)
  - Other information available

- The URL will be send back to the Beanstalk to go through content crawler

INDEX PAGE will be captured on a daily/weekly/monthly basis depends on the crawling frequency that has been determined.
Content /Detail Page

- Content crawler will scrap other information needed for this particular product (which was not available in index page)
- CONTENT/DETAIL PAGE will be captured when the page details are crawled.
HTML, CSS & JAVASCRIPT

CSS
PRESENTATION
“What does it look like?”

JavaScript
BEHAVIOR
“What does it do?”

HTML
STRUCTURE
“What does it mean?”

<table>
<thead>
<tr>
<th>HTML</th>
<th>CSS</th>
<th>Javascript</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure, Objects, Things</td>
<td>Looks, Style</td>
<td>Actions</td>
</tr>
<tr>
<td>&lt;p&gt; &lt;h1&gt; &lt;br&gt;</td>
<td>P {color: red;}</td>
<td>var x = 5;</td>
</tr>
<tr>
<td>nouns</td>
<td>adjectives</td>
<td>verbs</td>
</tr>
<tr>
<td>Walls, structure</td>
<td>Paint, curtains</td>
<td>Electrical, Plumbing, AC</td>
</tr>
</tbody>
</table>
## HTML ELEMENT, HTML TAGS

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;html&gt; ... &lt;/html&gt;</td>
<td>Declares the Web page to be written in HTML</td>
</tr>
<tr>
<td>&lt;head&gt; ... &lt;/head&gt;</td>
<td>Delimits the page’s head</td>
</tr>
<tr>
<td>&lt;title&gt; ... &lt;/title&gt;</td>
<td>Defines the title (not displayed on the page)</td>
</tr>
<tr>
<td>&lt;body&gt; ... &lt;/body&gt;</td>
<td>Delimits the page’s body</td>
</tr>
<tr>
<td>&lt;h n&gt; ... &lt;/hn&gt;</td>
<td>Delimits a level n heading</td>
</tr>
<tr>
<td>&lt;b&gt; ... &lt;/b&gt;</td>
<td>Set ... in boldface</td>
</tr>
<tr>
<td>&lt;i&gt; ... &lt;/i&gt;</td>
<td>Set ... in italics</td>
</tr>
<tr>
<td>&lt;center&gt; ... &lt;/center&gt;</td>
<td>Center ... on the page horizontally</td>
</tr>
<tr>
<td>&lt;ul&gt; ... &lt;/ul&gt;</td>
<td>Brackets an unordered (bulleted) list</td>
</tr>
<tr>
<td>&lt;ol&gt; ... &lt;/ol&gt;</td>
<td>Brackets a numbered list</td>
</tr>
<tr>
<td>&lt;li&gt; ... &lt;/li&gt;</td>
<td>Brackets an item in an ordered or numbered list</td>
</tr>
<tr>
<td>&lt;br&gt;</td>
<td>Forces a line break here</td>
</tr>
<tr>
<td>&lt;p&gt;</td>
<td>Starts a paragraph</td>
</tr>
<tr>
<td>&lt;hr&gt;</td>
<td>Inserts a horizontal rule</td>
</tr>
<tr>
<td>&lt;img src=&quot;...&quot; &gt;</td>
<td>Displays an image here</td>
</tr>
<tr>
<td>&lt;a href=&quot;...&quot; &gt; ... &lt;/a&gt;</td>
<td>Defines a hyperlink</td>
</tr>
</tbody>
</table>
DOCUMENT OBJECT MODEL (DOM)

DOM (Document Object Model)

- Root element: `<html>`
  - Element: `<head>`
    - Element: `<title>`
      - Text: "My title"
  - Element: `<body>`
    - Element: `<h1>`
      - Text: "A heading"
    - Element: `<a>`
      - Attribute: `href`
      - Text: "Link text"

Attribute | Value | Description
--- | --- | ---
class | class_rule or style_rule | The class of the element
id | id_name | A unique id for the element
style | style_definition | An inline style definition
LG UHD Monitor

**HOT DEAL**

**RM 1,497**

WAS RM 1,799

1/2 YEARLY CLEARANCE SALE: DETAILS

ADD TO CART

FIND IN STORE

**Specifications**

**General**

- **Brand**: LG
- **Model**: 32UN500-W
- **Product Type**: Monitors
- **Colour**: Black

**Description**

**Specifications**

- **Brand**: LG
- **Model**: 32UN500-W
- **Product Type**: Monitors
- **Colour**: Black

**HTML TAG**

<h1 class="fn product-title">LG 31.5-inch UHD 4K HDR Monitor (32UN500)</h1>

<span id="sec_discounted_price_75299" class="price-num">1,497</span>

<span id="sec_list_price_75299" class="list-price nowrap">1,799</span>
CRAWLER DESIGN DIAGRAMS

START

CRAWLER PROCESS 1
Access Home Page and extract categories URL

DATA 1
(Categories URL)

CRAWLER PROCESS 2
Access Product Categories page and extract Product details URL

DATA 2
(Product details URL)

END

CRAWLER PROCESS 3
Crawl product details page as HTML files

DATA 3
(HTML result - data)

<table>
<thead>
<tr>
<th>Process</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crawler Process 1</td>
<td>Access Home Page and extract the Categories URL, then store the categories URL for next process</td>
</tr>
<tr>
<td>Data 1</td>
<td>All Categories Page URL (link) stored to be used to access Product Index Page</td>
</tr>
<tr>
<td>Crawler Process 2</td>
<td>Access Product Categories Page to access product index and extract Product Details URL, then store the link for next process</td>
</tr>
<tr>
<td>Data 2</td>
<td>All Product Details Page Link stored to be used to access Product Detail Page</td>
</tr>
<tr>
<td>Crawler Process 3</td>
<td>Access Product Detail Page then crawl the product detail page as HTML files</td>
</tr>
<tr>
<td>Data 3</td>
<td>All product details HTML result from crawling process</td>
</tr>
</tbody>
</table>
1. Select main categories section

2. List all any needed category label for requirement in the next process

The main category label will be used as reference for the web crawler to take all sub-category that will be extracted in the next process
1. Crawl all sub category URL that related to all Main Categories from the first process

After we get the main category label, we should find the sub category link. This list of links will lead to each product by its sub category

List of subcategories
- en/category/meat-poultry/meat
- en/category/meat-poultry/offals
- en/category/meat-poultry/poultry
- en/category/meat-poultry/pre-packed
With subcategories URL crawled, we now can access to each product list under the sub categories.

In this process, the crawler will crawl all product URL link. The list of link will lead to detail page for each product.

Note: If the product list page contains multiple page, a looping for next (>) button will be click until there is no next page.
1. Crawler then will go to each of the list of product detail page URL and load it.

2. HTML page for each product will be saved in html format

3. For this example, html saved files will contain information needed such:
   - Title
   - Brand
   - Category
   - Price
   - Product information (if any)
Webscraping service with readily available template

- Amazon Product Data Crawler
- Google Maps Crawler
- Google Maps Review Crawler
Web scraping tool with a simple point-and-click interface, the ability to extract thousands of records from a website takes only a few minutes of scraper setup with many advanced features such as:

- Data scraping from multiple pages
- Multiple data extraction types

https://www.webscraper.io/
• Quickly scrape web data without coding
• Turn web pages into structured spreadsheets within clicks
• Free version consist of 10 crawlers with 2 concurrent local run (up to maximum of 10,000 records per export)
THANK YOU