SAPIENZA
UNIVERSITÀ DI ROMA

Giovanni Stilo, Ph.D.
stilo@di.uniroma1.it

Taste the Soup

Web Scraping and jSoup Introduction

Web Scraping

- **Web scraping** (web harvesting or web data extraction) is a computer software technique of extracting information from websites. Usually, such software programs **simulate human exploration** of the World Wide Web by either implementing low-level Hypertext Transfer Protocol (HTTP), or embedding a fully-fledged web browser, such as Internet Explorer or Mozilla Firefox.

WS Techiniques (0)

- **Human copy-and-paste:**
  - Sometimes even the best web-scraping technology cannot replace a human's manual examination and copy-and-paste, and sometimes this may be the only workable solution when the websites for scraping explicitly set up barriers to prevent machine automation.
- **Text grepping** and regular expression matching:
  - A simple yet powerful approach to extract information from web pages can be based on the UNIX grep command or regular expression.

WS Techiniques (1)

- **Data mining algorithms:**
  - Many websites have large collections of pages ***generated dynamically*** from an underlying structured source like a database. Data of the same category are typically encoded into **similar pages** by a common script or template.
    In data mining, a program that **detects** such **templates** in a particular information source, extracts its content and translates it into a **relational form** is called a wrapper.

- **HTML parsers:**
  - Some semi-structured data query languages, such as XQuery and the HTQL, can be used to parse HTML pages and to retrieve and transform page content.

WS Techiniques (2)

- **DOM parsing:**
  - Parse and control web pages into a DOM tree, that permit to control parts of the pages retieve.
- **Semantic annotation recognizing:**
  - The pages being scraped may embrace metadata or semantic markups and annotations, which can be used to locate specific data snippets.
- **Computer vision web page analyzers:**
  - There are efforts using machine learning and computer vision that attempt to identify and extract information from web pages by interpreting pages visually as a human being might.[3]

Wrapper

- **Wrapper:**
  - In data mining is a program that **extracts** content of a particular information source and translates it into a relational form.
  - Many web pages present structured data:
    - telephone directories,
    - product catalogs, etc.
  - All data ar formatted for human browsing using HTML language. Structured data are typically descriptions of objects retrieved from underlying databases and displayed in Web pages following some fixed templates.
  - The scope is to **translate HTML content into a relational form**. Wrappers are commonly used as such translators. Formally, a wrapper is a function from a page to the set of tuples it contains.

JSoup

- **jsoup** is a Java library for working with real-world HTML. It provides a very *convenient API* for extracting and manipulating data, using the best of DOM, CSS, and jquery-like methods.
  - scrape and parse HTML from a URL, file, or string
  - find and extract data, using DOM traversal or CSS selectors
  - manipulate the HTML elements, attributes, and text
  - clean user-submitted content against a safe white-list, to prevent XSS attacks
  - output tidy HTML
- jsoup is designed to deal with all varieties of HTML found in the wild; from pristine and validating, to invalid tag-soup; **jsoup** will create a sensible parse tree.

DOM Overview

- **Finding elements**
  - getElementById(String id)
  - getElementsByTag(String tag)
  - getElementsByClass(String className)
  - getElementsByAttribute(String key) (and related methods)
  - Element siblings: siblingElements(), firstElementSibling(), lastElementSibling(); nextElementSibling(), previousElementSibling()
  - Graph: parent(), children(), child(int index)

- **Element data**
  - attr(String key) to get and attr(String key, String value) to set attributes
  - attributes() to get all attributes
  - id(), className() and classNames()
  - text() to get and text(String value) to set the text content
  - html() to get and html(String value) to set the inner HTML content
  - outerHtml() to get the outer HTML value
  - data() to get data content (e.g. of script and style tags)
  - tag() and tagName()

Selector Overview

- tagname: find elements by tag, e.g. **a**
- ns|tag: find elements by tag in a namespace, e.g. **fb|name** finds <fb:name> elements
- #id: find elements by ID, e.g. **#logo**
- .class: find elements by class name, e.g. **.masthead**
- [attribute]: elements with attribute, e.g. **[href]**
- [^attr]: elements with an attribute name prefix, e.g. **[^data-]** finds elements with HTML5 dataset attributes
- [attr=value]: elements with attribute value, e.g. **[width=500]**
- [attr^=value], [attr$=value], [attr*=value]: elements with attributes that start with, end with, or contain the value, e.g. **[href*=/path/]**
- [attr~=regex]: elements with attribute values that match the regular expression; e.g**. img[src~=(?i)\.(png|jpe?g)]**
- *: all elements, e.g. *****

jSoup How

```
String html = "<html><head><title>First parse</title></head>"
  + "<body><p>Parsed HTML into a doc.</p></body></html>”;

Document doc = Jsoup.parse(html);
```

Or

```
String url=“http://www.romatoday.it/eventi/”;

Document doc = Jsoup.connect(url).get();

Elements newsHeadlines = doc.select(”a");
```

Try jsoup online: (http://try.jsoup.org/)

CLASSWORK:

For each event in:

http://www.romatoday.it/eventi/

Select the related information and provide them

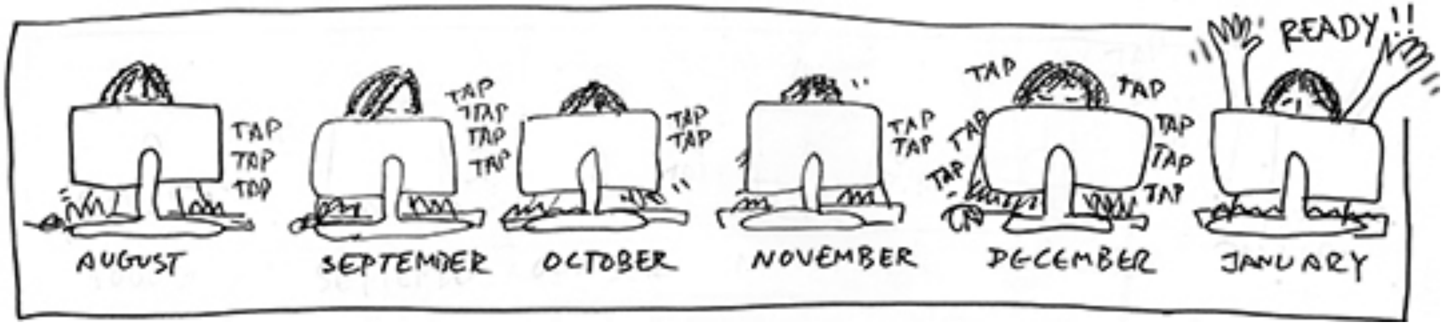
In a structured way.

Maven Snippet

```
<dependency>
        <groupId>org.jsoup</groupId>
        <artifactId>jsoup</artifactId>
        <version>1.8.2</version>
</dependency>
```

http://jsoup.org/

Let’s Try?!?!


TAP TAP TAP
AUGUST
TAP TAP TAP TAP
SEPTEMBER
TAP TAP
OCTOBER
TAP TAP
NOVEMBER
TAP TAP TAP TAP TAP TAP TAP TAP
DECEMBER
READY!!
JANUARY