# Lab 1: Crawling, indexation and webpage research

During this web mining laboratory, we explored the website crawling with Crawler4j and its indexation with Solr via jsoup, which is HTML parser. We finally used the indexation to perform searches.

We used docker-solr as docker image for our Solr.

## 1 Crawler

### 1.1 Core

We started by creating a core for Solr, named *core_one* using the following commands:

```
$ docker exec -u 0 -it docker-solr bash to attach to the docker bash
```

then

```
$ bin/solr create -c core_one to create a default core.properties
```

As we set *update.autoCreateFields* to **false** in our core. We had to create two custom fields: *doc_title_en* and *doc_body_en*, which will be used to store the title and the body of each retained pages by the crawler.

### 1.2 Crawler4j

We configured our crawler, **MyCrawler.java** to work with our Solr core *core_one*. Concerning our first crawler configuration:

- starting page: wikipedia.org at "Veganism" page
- domain limitation: yes
- maximum pages to fetch: 70
- maximum deepness: 3
- politeness delay: 500
- https: yes
- FILTERS: custom binary files

#### 1.2.1 shouldVisit function

We are applying the **FILTERS** pattern matching on the URL and verify that the domain.

#### 1.2.2 visit function

The crawler retrieves the HTML page and parses them with jSoup. It creates a Solr document with the id field (hashcode of the page), the *doc_title_en* (title from the page), and *doc_body_en* (body content from the page) and finally adds the document into the current Solr instance.
To avoid Solr overloads, we set a loader for the commits. Indeed, the program will stock 50 documents before committing them to the Solr as a batch.
Each visited page has its content indexed by Solr.

### 1.3 Tries and fails

- We first indexed **Vegan.com**, but the pages were not meaningful from a feature point of view, indeed it had no categories.
- We then indexed **arxiv.org**, but their *"robots.txt"* policy was not allowing us to go anywhere.

## 2 Specific Indexation

## 3 Research

## 4 Theorical questions

**Please explain what strategy should be adopted for indexing pages in several languages (each page is composed of only one language, but the corpus includes pages in several languages). What should you watch out for? Please explain the process you propose.**

We imagine different solutions to handle multiple languages with Solr:

- The first solution is to defined different schema fields for every language like "`title_fr`" or "`title_en`" and applying filters to each language. The downside of this solution is the high memory consumption and complexity when many languages are present.
- The second solution is to use a collection of same fields for all languages, add a field to store the langue and then apply a filter on it (e.g. `fq=language:english`). The downside of this solution is that we cannot use language specific features like lemmatisation and stemming.
- The third solution is to create a Solr core for each language and route the queries to the right core.

If we have few languages to querying we proposed to use the first solution but if we have a lot of languages we propose to use the third solution.

**Solr allows by default to do a fuzzy search. Please explain what it is and how Solr implements it. Some first names may have a lot of spelling variations (eg Caitlin : Caitilin, Caitlen, Caitlinn, Caitlyn, Caitlyne, Caitlynn, Cateline, Catelinn, Catelyn, Catelynn, Catlain, Catlin, Catline, Catlyn, Catlynn, Kaitlin, Kaitlinn, Kaitlyn, Kaitlynn, Katelin, Katelyn, Katelynn, etc). Is it possible to use, while keeping a good performance, the fuzzy research made available by Solr to do research taking into account such variations? If not what alternative(s) do you see, please justify your answer.**