CSDLC

Table of Contents

Introduction

Design Decisions

- All adapters are within Linux based containers.
- Python would be the primary language.
- There is heavy leverage on Linux CLI tools/commands.
- Care would be taken to avoid load on the sites ex: parallel downloads would be avoided wherever possible.

Registry Site Patterns

- Each trial as a pop-up with a trial-specific URL ex: CTRI
- Paginated listing pages with trial-specific detail page ex: JPRN
- Single listing page with trial-specific detail page ex: NTR

Web Scraping Patterns

- Download all trial recordings in one go ex: clinicaltrials.gov in one single .zip file containing trial-specific XML page
- Mirror complete website including all trial-specific pages
- Download the main listing page, identify links for each page containing a list of a small set of trials, then download the trial-specific pages
- Download the trial-specific pages directly using the regex pattern for URL and trial ids ex: for CTRI.

Data Transformation Techniques

- Transform HTML into JSON using tools like pup
- Transform raw JSON to a JSON format that adheres to the WHO Dataset Version 1.3.1 using JSTL

Tools

- Docker
- pup
- JSLT
- yq, xq, jq
- XSLT
- Apache mirroring a site
- wget, curl, Linux commands, bash scripting (CSDLC.org v2)
- Python - beautiful soup, scrapy (opentrials.net)

Analysis of Registries

Chinese Clinical Trial Registry

- Paginated with trail-specific detail page of each version
- Also, download in XML is available for each clinical trial.
- Searched by id proj = ? (http://www.chictr.org.cn/showprojen.aspx?proj=121747)

Peruvian Clinical Trial Registry (REPEC)

- Downloaded in a single XML document

Sri Lanka (SLCTR)

- They are paginated with a trail-specific detail page.
- Search is done by Registration number ex. For SLCTR/2020/026 URL is Slctr.

Brazilian Clinical Trial Registry (ReBec)

- It is a Single listed page with trial-specific detail page.
- Each trial can be download in XML format http://ensaiosclinicos.gov.br/xml_ictrp/downloadxmlictrp/7975.

German Clinical Trials Register (DRKS) :

- Search results are paginated with the link https://www.drks.de/drks_web/setPage.do?page= number(1-1117)
- The total clinical trials download are available in XML format but limited to the first 1000 only.
- Clinical trials are listing pages with link https://www.drks.de/drks_web/navigate.do?navigationId=trial.HTML&TRIAL_ID=DRKS000xxxxx (00001 - 25000) trial-specific detail page.
- The download of the individual trial is available in pdf format.

The Netherlands National Trial Register (NTR)

- Search results are present on a single page https://www.trialregister.nl/trials.
- Clinical trials are listed with https://www.trialregister.nl/trial/xxxx (1-9247)

Clinical Research Information Service (CRiS), Republic of Korea

- The download of total trials available but the downloaded file doesn't contain the information accurately.
- Clinical trials are listed with the link https://cris.nih.go.kr/cris/en/search/search_result_st01.jsp?seq=xxxxx.

Pan African Clinical Trial Registry (PACTR)

- Search results are paginated.
- The download of all the trials under particular search was available.
- Clinical trials are paginated with https://pactr.samrc.ac.za/TrialDisplay.aspx?TrialID=xxxxx.
- The download of the individual trial is available in pdf format.

EU Clinical Trials Register (EU-CTR):

- PAGINATION :
  - https://www.clinicaltrialsregister.eu/ctr-search/search?query=&page=1
  - page = 1 to 1952
- EudraCT Number: e.g. 2004-003035-31 used to reference(This number is not continuous)
- TO VIEW CLINICAL RESULTS :
  - Referenced by EudraCT Number: e.g. 2004-003035-31
  - EudraCT Number 2004-003035-31 - Clinical trial results
  - This page has a summary report downloadable in pdf format.
- download each page trials (18 trials on each page) in txt format. Use URL in pagination to download each page. (There is no provision for downloading all trials at once)

Iranian Registry of Clinical Trials (IRCT) :

- https://www.irct.ir/search/result?query=covid&filters=%7B%22perPage%22%3A%2225%22%2C%22sortBy%22%3A%22relevance%22%2C%22displayFormat%22%3A%22brief%22%2C%22selected%22%3A%5B%5D%7D&page=2 : page=2: change the page number here
- https://www.irct.ir/search/result?query=@irct_id:IRCT20150902023864N2 : for each registry, there is ICRT number:**IRCT20150902023864N2** to access the trial
- **(no pattern found in the icrt number)**
  - (to search based on the ICRT number or go to advanced search page) : https://www.irct.ir/trial/53653
  - 53653: on changing this number, we get separate trials, but not all numbers have trials.
  - On this trial page, we have accurate data, and there is an option of download as pdf and XML.
- https://www.irct.ir/trial/53653/xml : xml page for trial with number 53653

ISRCTN :

- https://www.isrctn.com/search?q= : this gives all records
- https://www.isrctn.com/ISRCTN15984604 :for records with ISRCTN number(the numbers are not continuous)
- https://www.isrctn.com/search?q=&page=2034&searchType=basic-search: page=2034: to change the page number to view documents.
- The page has advanced search options but not with an id but with features.

Japan Primary Registries Network (JPRN):

- https://upload.umin.ac.jp/cgi-open-bin/ctr_e/index.cgi?sort=03&isicdr=1&page=1 : pages from 1-429
- for accessing each registry :
  for viewing details of registry:
  - https://upload.umin.ac.jp/cgi-open-bin/ctr_e/ctr_view.cgi?recptno=R000049421 : referred based on receipt number(not continuous) e.g. here the receipt number is R000049421.
  for viewing the history of the registry:

    - https://upload.umin.ac.jp/cgi-open-bin/ctr_e/ctr_his_list.cgi?recptno=R000049421referred based on receipt number (not continuous)e.g. here the receipt number is R000049421.
- For each registry there is a UMIN NO. :
    - C000000001 - C000000459
    - UMIN000000460 - UMIN000043287
    - (from 460th record the C convention is changed to UMIN)
    - you have to access the receipt number using UMIN number to get the data.

ANZCTR: Australian New Zealand Clinical Trial Registry:

- Main website link: ANZCTR
- Search link: http://www.anzctr.org.au/TrialSearch.aspx
- Download all trials/Selected trails/Selected trail summaries to XML
- As of 2 Feb 2021: 28661 trails
- Trail ID is a unique key for every trial.

Thai Clinical Trials Registry (TCTR)

- Main Website link: TCTR: Thai Clinical Trials Registry
- Search link: TCTR: Thai Clinical Trials Registry Search
- The keyword TCTR, when typed into the search bar, gives 4923 results.
- Each trial has an ID as a key and is downloaded as XML individually

Lebanon Clinical Trials Registry (LBCTR)

- Main website link: LBCTR: Home
- Search link: LBCTR: Search (20 trials on one page)
- The search link contains all the 96 trials of the registry, and they each have their unique IDs that contain the link to the information page.
- Every trial registry has its HTML page which has all the necessary information on the trial.
- The HTML page is of the format “https://lbctr.moph.gov.lb/Trials/Details/x”, where x is the last four/three/two numbers of the primary registry ID based on the preceding zeros.
- This x can be extracted from the HTML source code of the list view for every 20 trials and can be further used to access the specified URL pages

Cuban Public Registry of Clinical Trials (RPCEC)

- Main website link: Welcome to the Cuban Registry of Clinical Trials | Registro Público Cubano de Ensayos Clínicos
- There is no proper search page for this registry
- We can instead access the sorting all registries by date dashboard option to access all the 352 registries by URL in a pagination format: Date register of trial
- Each of these URLs follows a proper pattern allowing the conversion from the HTML source code of these pages into any other format by the below-given tools convenient

Report on Tools:

For extracting data from the registries, we require some web-scraping tools to be used which are listed below :

1. Requests
2. Lxml
3. Beautiful Soup
4. Scrapy
5. selenium

Sno	Tool	Description	Pros	Cons	Use cases
1	Requests	Requests allow the user to send requests to the HTTP server and GET response back in HTML or JSON response.	Simple, Basic/Digest Authentication, International Domains and URLs, Chunked Requests, HTTP(S) Proxy Support	1. retrieves only static content of a page 2. Can't handle websites made from javascript 3.Can't be used for parsing HTMLl	
2	Lxml	lxml, a high performance, blazingly fast, production-quality HTML, and XML parsing Python library. It works well when we're aiming to scrape large datasets. Combination of lxml and requests are used.	Fast parsing, pythonic API, lightweight, uses element trees.	1. Does not work well with poorly designed HTML 2. The official documentation is not very beginner-friendly	
3	Beautiful soup	It works very well with poorly designed HTML and has a lot of functions. It Pulls out data from XML and HTML files. urllib or Requests to get the HTML text from a webpage and then use Beautiful Soup to clean it up. Combination of beautiful soup and requests are used. Beautifulsoup also has the option to use Lxml as an HTML parser, but it is slower than pure lxml	good documentation, robust, easy to learn, automatic encoding detection	1.It is slower than lxml.	Used when the documents in the HTMLl are not structured.
4	Selenium	It was designed to automate test for web Applications	1. It can efficiently work with core Javascript concepts 2. It can easily handle AJAX and PZAX requests	1. Very slow 2. Difficult to set up 3. High CPU and memory usage 4. Not ideal for large projects	
5	Scrapy	Scrapy is an open-source collaborative framework for extracting the data from the websites.	1. Scrapy has built-in support extracting data from HTML sources using XPath expression and CSS expression. 2. It is a portable library 3. It can be Easily Extensible. 4. It is faster than other existing scraping libraries. It can extract the websites 20 times faster than other tools. 5. It consumes a lot less memory and CPU usage.	1.Its documentation is not that much significant for the beginners because it does not have a beginner-friendly documentation 2.It is not ideal for websites with fewer data in it	
6.	Pyquery	HTML/XML parser library which helps to fetch data by accessing DOM. You can access any DOM element with the help of CSS selectors.			
7.	Urllib	It is a python package which can be used for opening URLs. To open and parse information from HTTP or FTP protocols.	Offers better control over requests	More complicated than requests.	
8	xq,jq,yq	Used in converting HTML,XML,... to JSON format	`Yq` takes YAML input, converts it to JSON, and pipes it to jq,xq is similar to yq		
9	pup	pup is a command line tool for processing HTML.	Fast and Flexible	-	
10	JSLT	JSLT is a complete query and transformation language for	It easily converts from json to json(more simplified version)		

		JSON.			

Selected registries

1.German Clinical Trials Register (DRKS):

- Beautiful Soup: In this method, we are web scraping each trial at a time with its id. For each trial we get the output in HTML format as shown below:

```
<h1 class="organizational-data">
 Organizational Data
</h1>
<ul class="organizational-data">
 <li class="drksId">
  <label>
   DRKS-ID:
  </label>
  DRKS00000003
 </li>
 <li class="firstDrksPublishDate">
  <label>
   Date of Registration in DRKS:
  </label>
  2008/08/08
 </li>
 <li class="firstPartnerPublishDate">
  <label>
   Date of Registration in Partner Registry or other Primary Registry:
  </label>
  2008/02/01
 </li>
 <li class="investorInitiated">
  <label>
```

Since we are already getting data in HTML format we can directly convert it into JSON format with tools like pup…

- Lxml : In this method, we are web scraping each trial at a time with its id. For each trial we get the output in text format as shown below:

```
DRKS-ID:
        DRKS00000003


Trial Description
start of 1:1-Block title
Title
OSAKA: A multicenter, four arm, randomized, open label clinical studyinvestigating optimized dosin
end of 1:1-Block title
start of 1:1-Block acronym
Trial Acronym
[---]*
end of 1:1-Block acronym
start of 1:1-Block url
URL of the Trial
[---]*
end of 1:1-Block url
start of 1:1-Block public summary
Brief Summary in Lay Language
Why is this study being conducted?Different combinations of medications are known for preventing r
end of 1:1-Block public summary
start of 1:1-Block scientific synopsis
Brief Summary in Scientific Language
A multicenter, randomized, open, four armed, parallel group,comparative phase IIIb study. Subjects
```

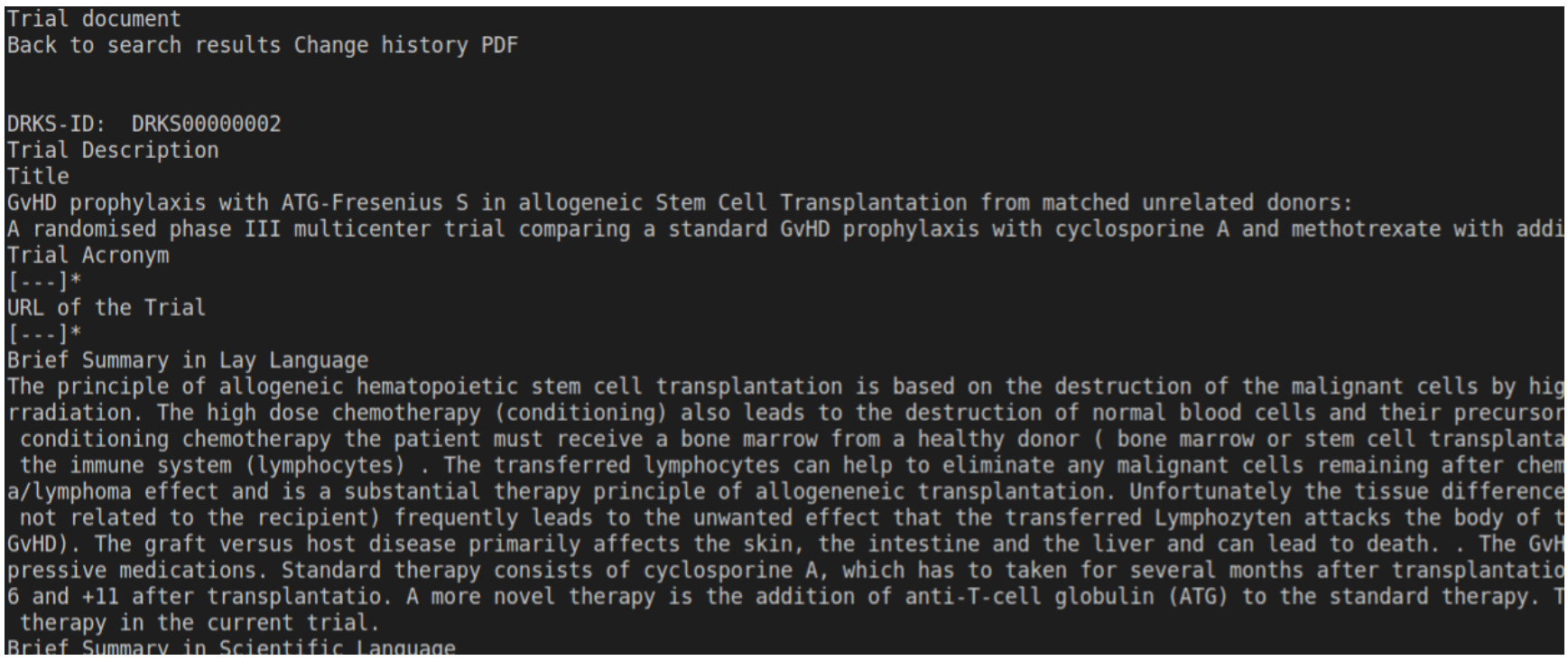
We have an option of getting each attribute like title, summary...separately.

- Scrapy(shell): In the method, we can extract the desired parts easily with selectors as a list. Also, the output is in simple HTML but not prettified as beautiful soup. Since it was done in the shell we couldn't able observe the runtime of the process.

```
In [27]: response.css('#content > div.trial > div.recruitmentCountries > ul').extract()
Out[27]: ['<ul class="recruitmentCountries">\n<li class="country">\n<label>DE:\n\t\t\t\t\t\t\t</label>\xa0\n\t\t\t\t\t\t\tGermany</li>\n
<li class="country">\n<label>DK:\n\t\t\t\t\t\t\t</label>\xa0\n\t\t\t\t\t\t\tDenmark</li>\n<li class="country">\n<label>SE:\n\t\t\t\t\t\t
\t</label>\xa0\n\t\t\t\t\t\t\tSweden</li>\n<li class="country">\n<label>NO:\n\t\t\t\t\t\t\t</label>\xa0\n\t\t\t\t\t\t\tNorway</li>\n<li
class="country">\n<label>FI:\n\t\t\t\t\t\t\t</label>\xa0\n\t\t\t\t\t\t\tFinland</li>\n<li class="country">\n<label>IS:\n\t\t\t\t\t\t\t</
label>\xa0\n\t\t\t\t\t\t\tIceland</li>\n<li class="country">\n<label>NL:\n\t\t\t\t\t\t\t</label>\xa0\n\t\t\t\t\t\t\tNetherlands</li>\n<l
i class="country">\n<label>CH:\n\t\t\t\t\t\t\t</label>\xa0\n\t\t\t\t\t\t\tSwitzerland</li>\n<li class="country">\n<label>AT:\n\t\t\t\t\t
\t\t</label>\xa0\n\t\t\t\t\t\t\tAustria</li>\n<li class="country">\n<label>IT:\n\t\t\t\t\t\t\t</label>\xa0\n\t\t\t\t\t\t\tItaly</li>\n<l
i class="country">\n<label>CZ:\n\t\t\t\t\t\t\t</label>\xa0\n\t\t\t\t\t\t\tCzech Republic</li>\n</ul>']
```

The normal scrapy project needs a little understanding of the library (methods).

- Selenium: In this method we are web scraping each trial at a time with its id.For each trial we get the output in text format. Using this tool we can fill the forms in the website like login,search,advanced search,filters ...etc



OBSERVATIONS :

- lxml runs faster than beautiful soup but it gives data in text format whereas beautiful soup gives in html format which is easily convertible to JSON format.But we can extract each field separately using lxml and store it in database/json format.
- Selenium has high running time.
- ***Scrapy needs a little understanding of the library.***
- ***Or Beautifulsoup***

2.ANZCTR: Australian New Zealand Clinical Trial Registry

- Direct Download of all the trials is available, which contains data in xml format

Search :

A close friend has been diagnosed with Acute Lymphoblastic Leukemia and I wanted to check if he is eligible to enroll in any immunotherapy treatments.

- How many registries have trials in ongoing/recruiting status on the above condition
- What are the eligibility criteria for each trial.
- Create an excel spreadsheet.