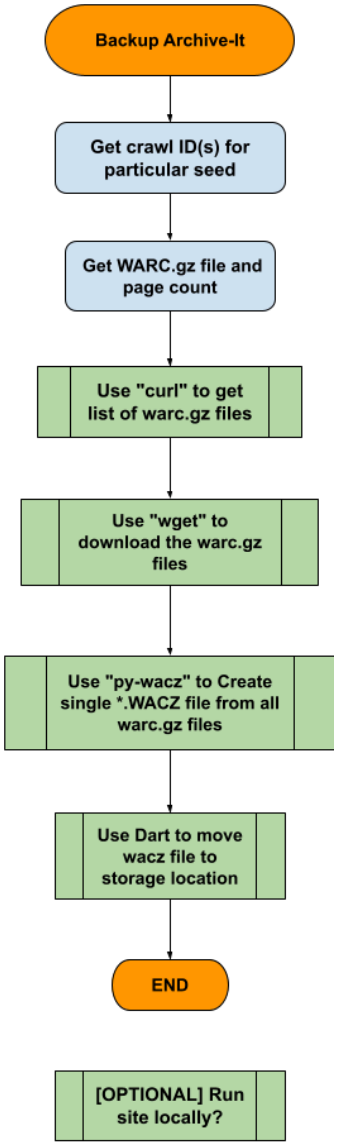
Process for backing up WARCs from Archive-it to MiDPN

Also available here:
https://github.com/mlibrary/digiPres/blob/main/webarchiving/warcs2mdpn/readme.md

Backup from Archive-It, partially based on this article:

How to find and download your WARC files with WASAPI – Archive-It Help Center



Basic Process:

- Get crawl ID(s) for particular seed
- Get WARC.gz file and page count
- Use "curl" to get list of warc.gz files
- Use "wget" to download the warc.gz files
- Use "py-wacz" to Create single *.WACZ file from all warc.gz files
- Create metadata and use DART to move wacz file to MiPDN

[OPTIONAL] Run site locally?
[Other Considerations]

Get list of Crawls:
Collection > Seed > List of Crawls:

1. From the Collection list, select Collection URL:
   https://partner.archive-it.org/1011/collections
2. Click on **Seeds**
3. From the Seed list, select **Seed**:
   https://partner.archive-it.org/1011/collections/13472/seeds/3316427
4. Get **Crawl IDs** from **Crawling History and use those in WASAPI:**
   https://warcs.archive-it.org/wasapi/v1/webdata?crawl=1993810
   https://warcs.archive-it.org/wasapi/v1/webdata?crawl=1991571

Webdata Query List

API endpoint that allows webdata files to be queried for and listed.

```
GET /wasapi/v1/webdata?crawl=1993810
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json ;utf-8
Vary: Accept

{
  "count": 3,
  "next": null,
  "previous": null,
  "files": [
    {
      "filename": "ARCHIVEIT-13472-TEST-JOB1993810-0-SEED3316427-20240625183026939-00000-w8s1rcf0.warc.gz",
      "filetype": "warc",
```

For crawl **1993810,** there are **3 warc.gz files**.

1. **Use "curl" to get a list of warc.gz files:**

   Note: An Archive-It WARC is no bigger than 1GB, so a single crawl can generate multiple WARCs.

**** Using sudo su is key here****
**** You also need https://jqlang.org/ installed to parse the locations ****

```
sudo su

curl --config curlConfig.txt

"https://warcs.archive-it.org/wasapi/v1/webdata?crawl=1993810" | jq -r
.files[].locations[0] > url.list
curl --config curlConfig.txt
"https://warcs.archive-it.org/wasapi/v1/webdata?crawl=1991571" | jq -r
.files[].locations[0] >> url.list
```

2. **Then, actually download the warc.gz files:**

```
wget --accept txt,gz -i url.list
```

**** If you have multiple crawls, you may want to use a batch process using a Bash script that pulls the crawls from the Seed Crawls list.

1. Click on **Crawling History**
2. To download CSV of Seed Crawls click on **Download Seed Crawls list**
3. Open the CSV in spreadsheet
4. Select the entries from the Crawl ID column and save these as plain text in a file (RWAcrawls.txt).

This file will be used as input in Bash script (the curl command is from https://support.archive-it.org/hc/en-us/articles/360015225051-How-to-find-and-download-your-WARC-files-with-WASAPI).

This script needs to run after `sudo su`.

```
  #!/bin/bash
  while IFS= read -r crawl; do
      curl -u username:password
"https://warcs.archive-it.org/wasapi/v1/webdata?crawl=${crawl}" | jq -r
.files[].locations[0] >> url.list
  done < "RWAcrawls.txt"
```

The '>>' appends each list of warcs to the same url.list file.

*** Need to check for crawls with count > 100, will need to use pagination option.***

3. **Use pywacz to create a single *.WACZ file from all warc.gz files** (using https://github.com/webrecorder/py-wacz):

```
wacz create -o africanElections.wacz -f *.warc.gz -t --detect-pages
```

*** Note this issue when running wacz: https://github.com/webrecorder/py-wacz/issues/50

Create a metadata.txt file containing fields, using information from Archive-it:

```
Collection: (same collection name from Archive-it)
CollectionID: (same collection ID from Archive-it)
Seed: (seed ID from Archive-it)
URL: (original site URL)
Title: (Title of the website)
Creator: (creator of the website)
Description: (same description from Archive-it)
Language:
Country:
Created:
Updated:
```

Use DART with MDPN configuration to transfer the wacz file and metadata.txt to MDPN for backup.

Save bag as:

UM_c[collectionID]_s[seedID]

Elections in Africa example: `UM_c13472_s3316427`

Run Site Locally

Move newly created africanElections.wacz to webserver location and update index.html file accordingly (see https://replayweb.page/docs/embedding/#self-hosting):

HTML:

```
<!doctype html>
<html class="no-overflow">
  <head>
    <title>WebArchive</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <script src="js/ui.js"></script>
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <section class="web-archive">
    <replay-web-page replayBase="js/" url="https://forgood.org.za/"
source="wacz/africanElections.wacz"></replay-web-page>
    </section>

  </body>
</html>
```

Start Apache webserver:

```
sudo apachectl restart
```

http://localhost/webarchive/AfricanElections/

**** Using Apache webserver here because of ReplayWeb's HTTPS CORS requirement, mentioned here https://replayweb.page/docs/user-guide/locations/, simple HTTPS will not work.

[Other Considerations] Extract just the files

If you do not need to capture the entire site but just the content, you can extract PDFs and/or other files such as images from WARC.gz files for easier access using `warc-extractor` from https://github.com/recrm/ArchiveTools.

Usage examples:

```
python3 warc-extractor.py http:content-type:pdf
python3 warc_extractor.py -dump content http:error:200
```