New issue

Jump to bottom

Add support for recursive archiving of entire domains, or across domains to a given depth (using a crawler) #191

Open **diego898** opened this issue on Mar 22, 2019 · 22 comments

Labels help wanted size: hard status: idea-phase touches: data/schema/architecture why: functionality

diego898 commented on Mar 22, 2019

Is it in scope to have it be possible to archive:

- an entire blog (all posts) by only passing the root url?
- have this archive process only be additive? (even if posts are later deleted, I can browse the local history myself)

48 ❤ 10 1

pirate commented on Mar 23, 2019 • edited Member

I'd say this is currently our 2nd most-requested feature :) It's definitely in the roadmap: #120 but not planned anytime soon because it's not ArchiveBox's primary use-case and it's extremely difficult to do well. For now I recommend using another software to do the crawling to produce a list of URLs of all the pages, and then pipe that list into archivebox to do the actual archiving.

Eventually, the idea plan is roughly to expose similar flags on ArchiveBox as are available on `wget`:

- `--mirror`
- `--level=5`
- `--span-hosts`
- `--recursive`
- `--no-parent`

https://www.gnu.org/software/wget/manual/wget.html#Recursive-Retrieval-Options-1

These flags together should cover all the use cases:

1. archiving an entire domain with all pages
2. archiving an entire domain but only below the current directory level
3. archiving recursively from a single page across all domains to a given depth

I anticipate it will take a while to get to this point though (12+ months likely), as we first have to build or integrate a crawler of some sort, and web crawling is an extremely complex process with lots of subtle nuance around configuration and environment.

The process will also naturally be additive the moment snapshot support is added: #179.

Unfortunately, doing mirroring / full-site crawling properly is extremely non-trivial, as it involves building or integrating with an existing crawler/spider. Even just the logic to parse URLs out of a page is deceivingly complex, and there are tons of intricacies around mirroring that don't need to be considered when doing the kind of single-page archiving that ArchiveBox was designed for.

Currently this is blocked by setting up our proxy archiver which has support for deduping response data in the WARC files, then we'll also need to pick a crawler, or integrate with an existing one from here.

For people landing on this issue and looking for an immediate solution, I recommend using this command (which is exactly what's used by ArchiveBox right now, but with a few recursive options added):

```
wget --server-response \
     --no-verbose \
     --adjust-extension \
     --convert-links \
     --force-directories \
     --backup-converted \
     --compression=auto \
     -e robots=off \
     --restrict-file-names=unix \
     --timeout=60 \
     --warc-file=warc \
     --page-requisites \
     --no-check-certificate \
     --no-hsts \
     --span-hosts \
     --no-parent \
     --recursive \
     --level=2 \
     --warc-file=$(date +%s) \
     --user-agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.75 Safari/537.
     https://example.com
```

Set `--level=[n]` to the depth of links you want to follow during archiving, or add `--mirror` and remove `--span-hosts` and `--no-parent` if you want to archive an entire domain.

16 ❤ 1

pirate added size: hard status: idea-phase why: functionality labels on Mar 23, 2019

pirate changed the title ~~Feature Request: Recursive Archiving~~ **Add support for recursively archiving an entire site / domain like wget --mirror** on Mar 25, 2019

bcarothe commented on Apr 2, 2019

I am interested in this feature, particularly in limited depth/level recursion. I would like ArchiveBox to be able to .archive a page from a specified URL and then archive any links/content that may not be on the same domain. My particular interest is to be able to click around on the archived page to local archived version of it rather than the actual URL (similar to how the Wayback Machine handles it).

Keep up the good work and good luck!

1

knowncolor mentioned this issue on Apr 30, 2019

Follow and archive all links to a given depth. Similar to #191 but not restricted to a domain. #226

Closed

pirate changed the title ~~Add support for recursively archiving an entire site / domain like wget --mirror~~ **Add support for recursive crawler archiving for entire domains, or across domains to a given depth** on Apr 30, 2019

pirate changed the title ~~Add support for recursive crawler archiving for entire domains, or across domains to a given depth~~ **Add support for recursive archiving of entire domains, or across domains to a given depth (using a crawler)** on Apr 30, 2019

pirate mentioned this issue on Sep 20, 2019

Feature Request: Permit users to specify link depth for archiving #270

Closed

theAkito commented on Sep 22, 2019

I downloaded this app and took this ability for granted. Just by accident I found it not to work, yet. When can this be expected to be implemented?

pirate commented on Sep 23, 2019 • edited Member

Not for a while, it's a very tricky feature to implement natively, I'd rather integrate an existing crawler and use ArchiveBox to just process the generated stream of URLs. Don't expect this feature anytime soon unless you feel like implementing it yourself, for now you can check out some of the alternative software on the wiki:

https://github.com/pirate/ArchiveBox/wiki/Web-Archiving-Community

2

theAkito commented on Sep 23, 2019

@pirate Thank you very much for the reference. Will look into it, as long as the feature is not implemented into ArchiveBox.

1

pirate mentioned this issue on Jan 5, 2020

Question: Cloning whole site approaches #307

Closed

pirate mentioned this issue on Jul 7, 2020

Question: Get all links from the same domain #354

Closed

pirate mentioned this issue on Jul 30, 2020

Feature Request: Mirror from admin webui #401

Closed

pirate added help wanted touches: data/schema/architecture labels on Jan 23, 2021

pirate pinned this issue on Apr 6, 2021

GlassedSilver commented on Apr 11, 2021

This got pinned I see.

Does this mean this feature can expect to see some level of work sometime soon?

I swear to God, once this gets added I'll be damn happy. It's something I've been dreaming of having, domain-level snapshotting to go back in time for various endeavors.

Just imagine... browsing playstation.com or such 20-30 years later, relieving the PS4 and PS5 era in the context of them being retro, but being able to get all those articles etc back.

When I find old screenshots of websites and my OS that are pushing 15 years or more now I get nostalgia tickles, a feature like this would go beyond seeing what I saw back in the day and enable me to discover stuff that escaped my eyes at all to begin with at solely MY discretion.

The level of archivist comfort here is immeasurable.

This combined with automated re-snapshotting over time... Incredible!

Bonus question: would this also lay the foundation to enable use cases where I may want to archive not the entire domain, but all items of a certain user's feed?
e.g.: for future reference I may want to "subscribe" to a Twitter user locally and not miss any of their tweets. Surely the crawling method here could be utilized for this as well, right?
Correct me if I'm wrong and dreaming. :)

2

pirate commented on Apr 11, 2021 • edited Member

You can already archive everything on a user's feed with the `--depth=1` flag, it just doesn't support depth >1.

However, you can achieve full recursive archiving if you do multiple passes of `--depth=1` archiving (breadth-first), e.g.:

```
archivebox add --depth=1 https://example.com
archivebox list https://example.com | archivebox add --depth=1
archivebox list https://example.com | archivebox add --depth=1
archivebox list https://example.com | archivebox add --depth=1
...
# etc as many levels deep as you want, it wont duplicate stuff so it's safe to re-run until theres nothing new that it discovers
```

This is nice for a number of reasons, you can keep an eye on progress and make sure it's not accidentally downloading all of youtube by accident, and the URLs are added in order of depth and can be tagged separately during the adding process if you want. It also allows you to manually rate-limit, to avoid being blocked by / taking down the site you're archiving.

I really don't want to implement my own full recursive crawler, it's a lot of work and really difficult to maintain. Also a big support burden as crawlers constantly break and need fixing and extra config options to handle different desired behavior on different kinds of sites. I would much rather people use one of the many great crawlers/spiders that are already available and pipe the URLs into archivebox (e.g. Scrapy). https://scrapy-do.readthedocs.io/en/latest/quick-start.html

As it stands, I'm unlikely to add a crawler directly into ArchiveBox anytime soon because I barely have enough time to maintain ArchiveBox-as-is, but I'm not opposed to improving the ergonomics around using it with a crawler with smaller PRs, or reviewing a proposed design if someone wants to contribute a way to build scrapy or another existing crawler into AB.

This issue is pinned because we get a lot of requests for it and I'd rather make this thread easy to find so people know what the status is.

4

larshaendler commented on Jun 10, 2021

@pirate I used a mix of the two suggestion you offered and build myself a workaround. That does the trick for a local version that I can navigate offline.

1. Run wget first to get a list of all urls of my page.
   `wget --spider --recursive --no-verbose --output-file=wgetlog.txt https://mydomain.com/`
2. Run sed to remove all wget clutter:
   sed -n "s@.+ URL:([^]+) .+@\1@p" wgetlog.txt | sed "s@&@&@" > myurls.txt
3. Manually open the myurls.txt to remove any pages that look fishy or dont make sense to keep.
4. Drop all urls in archivebox in a single command
   `xargs archivebox add < ~/Downloads/myurls.txt`

4a. Alternative way, drop each line with a single command
`xargs -0 -n 1 archivebox add < <(tr \\n \\0 < ~/Downloads/myurls.txt)`

The result pages can be navigated locally because archivebox is intelligent enough to find all linked offline versions. But it is of course not a single page dump.

6

pirate mentioned this issue on Jun 27, 2021

Question: FileNotFoundError - 'single-file' and others #774

Closed

francwalter commented on Jul 2, 2021

> You can already archive everything on a user's feed with the `--depth=1` flag, it just doesn't support depth >1.
>
> However, you can achieve full recursive archiving if you do multiple passes of `--depth=1` archiving (breadth-first), e.g.:
>
> ```
> archivebox add --depth=1 https://example.com
> archivebox list https://example.com | archivebox add --depth=1
> archivebox list https://example.com | archivebox add --depth=1
> archivebox list https://example.com | archivebox add --depth=1
> ...
> # etc as many levels deep as you want, it wont duplicate stuff so it's safe to re-run until theres nothing new that it discover
> ```
>
> This is nice for a number of reasons, you can keep an eye on progress and make sure it's not accidentally downloading all of youtube by accident, and the URLs are added in order of depth and can be tagged separately during the adding process if you want. It also allows you to manually rate-limit, to avoid being blocked by / taking down the site you're archiving.

Is there a way to exclude urls not within example.org in this setting? Is there an option maybe?
Thanks, frank

pirate commented on Jul 7, 2021 • edited Member

@francwalter I've added a new option ~~`URL_WHITELIST`~~ `URL_ALLOWLIST` in `5a2c78e` for this usecase. Here's an example of how to exclude everything except for URLs matching `*.example.com`:

```
export URL_ALLOWLIST='^http(s)?:\/\/(.+)?example\.com\/?.*$'

# then run your archivebox commands
archivebox add --depth=1 'https://example.com'
archivebox list https://example.com | archivebox add --depth=1
archivebox list https://example.com | archivebox add --depth=1
...
# all URLs that don't match *.example.com will be excluded, e.g. a link to youtube.com would not be followed
# (note that all assets required to render each page are still archived, URL_DENYLIST/URL_ALLOWLIST does not apply to inline images
```

I've also documented the new allowlist support here: https://github.com/ArchiveBox/ArchiveBox/wiki/Configuration#URL_ALLOWLIST

It will be out in the next release v0.6.3, but if you want to use it early you can run from the `dev` branch:
https://github.com/ArchiveBox/ArchiveBox#install-and-run-a-specific-github-branch

4 ❤ 1

pirate mentioned this issue on Aug 2, 2021

Can archivebox download an entire website for offline view, not just a web page? #820

Closed

pirate mentioned this issue on Nov 23, 2021

Feature Request: Whole-site archiving with link-rewriting to point to archived versions #893

Closed

9 tasks

kiwimato commented on Dec 12, 2022

I may be doing something wrong, but it doesn't work for me

```
$ export URL_WHITELIST='^http(s)?:\/\/(.+)?redacted\.com\/?.*$'
$ archivebox add --depth=1 'https://redacted.com'
[i] [2022-12-12 00:28:42] ArchiveBox v0.6.3: archivebox add --depth=1 https://redacted.com
    > /data

[+] [2022-12-12 00:28:42] Adding 1 links to index (crawl depth=1)...
[Errno 2] No such file or directory: 'https:/redacted.com'
    > Saved verbatim input to sources/1670804922-import.txt
    > Parsed 1 URLs from input (Generic TXT)

[*] Starting crawl of 1 sites 1 hop out from starting point
    > Downloading https://redacted.com contents
    > Saved verbatim input to sources/1670804922.925886-crawl-redacted.com.txt
    > Parsed 30 URLs from input (Generic TXT)
    > Found 0 new URLs not already in index

[*] [2022-12-12 00:28:43] Writing 0 links to main index...
    √ ./index.sqlite3

$ archivebox list https://redacted.com | archivebox add --depth=1
[i] [2022-12-12 00:29:55] ArchiveBox v0.6.3: archivebox list https://redacted.com
    > /data

[i] [2022-12-12 00:29:55] ArchiveBox v0.6.3: archivebox add --depth=1
    > /data

[+] [2022-12-12 00:29:56] Adding 2 links to index (crawl depth=1)...
[Errno 21] Is a directory: '/data/archive/1670804464.144965'
[Errno 2] No such file or directory: 'https:/redacted.com'
[Errno 2] No such file or directory: '"redacted.com"'
    > Saved verbatim input to sources/1670804996-import.txt
    > Parsed 2 URLs from input (Generic TXT)

[*] Starting crawl of 1 sites 1 hop out from starting point
    > Downloading https://redacted.com contents
    > Saved verbatim input to sources/1670804996.266747-crawl-redacted.com.txt
    > Parsed 30 URLs from input (Generic TXT)
    > Found 0 new URLs not already in index

[*] [2022-12-12 00:29:56] Writing 0 links to main index...
    √ ./index.sqlite3

$ archivebox list https://redacted.com
[i] [2022-12-12 00:30:20] ArchiveBox v0.6.3: archivebox list https://redacted.com
    > /data

/data/archive/1670804464.144965 https://redacted.com "redacted.com"
$
```

JustGitting commented on Dec 17, 2022

@kiwimato I've got the same problem. Seems the `archivebox list` command does not output what the piped `archivebox add` command is expecting.

I've modified the list command to the following and it seems to be working. Feedback and corrections welcome.

I used the substring option as felt it was more flexible... I'm new to archivebox.

`2>/dev/null` is used to redirect extra info to null, it cleans up the output of the list command.

```
export URL_WHITELIST='^http(s)?:\/\/(.+)?example\.com\/?.*$'

# then run your archivebox commands
```

```
archivebox add --depth=1 'https://example.com'
archivebox list --csv url -t substring https://example.com 2>/dev/null | archivebox add --depth=1
```

❤ 1

kiwimato commented on Dec 17, 2022 • edited

@JustGitting I got around this by actually using browsterix-cralwer which is awesome and does this out of the box. It also has the ability to create WACZ files directly which can then be used with web-replay-gen

Thank you for posting a solution btw :)

3

JustGitting commented on Dec 17, 2022

@kiwimato great to hear you found a workaround. I had a look at browsterix-crawler, but hoped to avoid needing to run docker. I like simple commands :-).

GRMrGecko commented on Jan 18, 2023

I'm stuck with using SiteSucker and storing into my own local directory until this is added. There are some sites I want to archive entirely for my personal browsing at a later date because I do not know if the site owner will continue to keep their site up. For an example, I have an archive of http://hampa.ch/ which will be useful for a lot of the things I do personally.

1

This comment was marked as duplicate. Sign in to view

KenwoodFox commented on Jul 6, 2023

I'm working out a way to hopefully archive webcomics, the trouble is most comics store their images as separate but very similar web pages ie domain.org/d/20210617 then domain.org/d/20210618 and domain.org/d/20210619. Maybe rather than trying to cram and everything-crawler into ArchiveBox, we could just improve the connections that could be made to more generalized crawlers? Even a python slot to type in some bs4 or something :3

melyux commented on Jul 11, 2023

I saw **@larshaendler** said above that:

> The result pages can be navigated locally because archivebox is intelligent enough to find all linked offline versions. But it is of course not a single page dump.

Is this true? My internal hyperlinks on snapshots seem to go to the originals, not to their archived offline versions already on ArchiveBox. Even without adding a full-blown crawler, I'm sure a much simpler enhancement would be to, upon adding a new snapshot, rewrite all hyperlinks pointing to that snapshot's original URL to now point to the offline version. And check if any hyperlinks on the current snapshot exist offline in the archive and rewrite those to point to the offline versions.

KenwoodFox commented on Jul 13, 2023

> I saw **@larshaendler** said above that:
>
> > The result pages can be navigated locally because archivebox is intelligent enough to find all linked offline versions. But it is of course not a single page dump.

> Is this true? My internal hyperlinks on snapshots seem to go to the originals, not to their archived offline versions already on ArchiveBox. Even without adding a full-blown crawler, I'm sure a much simpler enhancement would be to, upon adding a new snapshot, rewrite all hyperlinks pointing to that snapshot's original URL to now point to the offline version. And check if any hyperlinks on the current snapshot exist offline in the archive and rewrite those to point to the offline versions.

I've actually also experienced this too, i tried it out just the other day infact! Even if i archive one path, then archive another with that archived path in it. The links still point back to the original. Dosn't seem to matter what order i archive the sites in.

KenwoodFox commented on Jul 13, 2023

Can archivebox display wacz files? because that would be great to pair that crawler with a docker archivebox

pirate commented on Aug 16, 2023 Member

No, we don't have wacz support yet but I'm friends with the folks that designed that spec, and I'd love to integrate with browsertrix / ArchiveWeb.page + ReplayWeb.page at some point in the future to improve our wacz support.

For now there are many higher priority things on my plate, mainly the event-sourcing refactor of the ArchiveBox internals (using django-Huey-monitor), before I'm ready to add new extractors / UI.

❤ 2

TomLucidor commented last month

@pirate this is exactly the feature I would look for (mixing with AI for saerching pages) when HTTrack gets antiquated, and thank **@larshaendler** for the idea of extracting URLs using WGet first before archiving. What is the current best way to do this assuming we are not using WGet (on windows)?

Assignees

No one assigned

Labels

help wanted | size: hard | status: idea-phase | touches: data/schema/architecture | why: functionality

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

14 participants