ATOMIC OBJECT

Create a Local Copy of a Website with HTTrack

UX/DESIGN TOOLS **BY: BOBBY KILPATRICK, FEBRUARY 12, 2016**

I've recently been experimenting with **HTTrack** (http://www.httrack.com/), an open-source utility that makes it possible to download a full copy of any website. HTTrack is essentially a web crawler, allowing users to retrieve every page of a website merely by pointing the tool to the site's homepage.

From the HTTrack homepage:

> *"[HTTrack] allows you to download a World Wide Web site from the Internet to a local directory, building recursively all directories, getting HTML, images, and other files from the server to your computer. HTTrack arranges the original site's relative link-structure."*

I thought I'd share my experience with it.

Installing HTTrack

There are a couple of different ways to install HTTrack:

- **HTTrack Website** (http://www.httrack.com/): Download and install HTTrack manually. The download contains a README with detailed directions.
- **Homebrew** (http://brew.sh): Users of Homebrew can easily install HTTrack with the formula `brew install httrack`.

Basic Syntax

> *The syntax of HTTrack is quite simple. You specify the URLs you wish to start the process from, any options you might want to add ([-option]), any filters specifying places you should ([+]) and should not ([-]) go, and end the command line by pressing Enter. HTTrack then goes off and does your bidding.*

RELATED POSTS

UX/DESIGN TOOLS (HTTPS://SPIN.ATOMICOBJ /UX-DESIGN/UX-DESIGN-TOOLS/)

5 Tailwind CSS Anti-Patterns to Avoid (https://spin.atomicobje /tailwind-css-anti-patterns/)

UX/DESIGN TOOLS (HTTPS://SPIN.ATOMICOBJ /UX-DESIGN/UX-DESIGN-TOOLS/)

What to Consider When Selecting a Component Library (https://spin.atomicobje /component-libraries-2/)

UX/DESIGN TOOLS (HTTPS://SPIN.ATOMICOBJ /UX-DESIGN/UX-DESIGN-

At its most basic, HTTrack can be run by specifying just a single URL:

```
httrack http://example.com
```

This will unleash the program on the `http://example.com` domain with default settings. HTTrack retrieves this URL, then parses the page for more links. Any links found within the page are downloaded next and parsed for additional links. The process continues on until the crawler cannot find any links it hasn't already downloaded.

You can also add options to the basic command to customize HTTrack's behavior. For example, you can specify forbidden URLs and directories, alter download speeds, and limit downloads to a certain filetype. HTTrack has a huge number of options, accessible via `` `httrack --help` `` and at the **project website** (https://www.httrack.com/html/fcguide.html).

Custom Options

My goal for HTTrack was to create a static copy of the Atomic Object marketing website. To speed up my download and decrease the load on the server, I wanted to download only HTML, CSS, and JavaScript files. Images and other file types like videos and PDFs tend to be the largest files, so I intentionally omitted them.

Through trial and error, I came up with the following formula (broken out by line to make more readable):

```
1
2 httrack https://atomicobject.com \
3    -atomicobject.com/assets/* \
4    +atomicobject.com/*.css \
5    +atomicobject.com/*.js \
6    -atomicobject.com/documents/* \
7    -atomicobject.com/uploadedImages/* \
8    --path "~/httrack-copies/atomicobject/" \
9    --verbose \
```

Let's take a detailed look at what each option in the command does:

```
httrack https://atomicobject.com
```

As we saw in the basic syntax above, this points HTTrack at the site we want to copy.

```
-atomicobject.com/assets/* -atomicobject.com/documents/*
-atomicobject.com/uploadedImages/*
```

A rule that begins with a minus sign indicates something that we *don't* want HTTrack to

TOOLS/)

Prime Faces: A Worthy Bootstrap Alternative? (https://spin.atomicobje/prime-faces-vs-bootstrap/)

download. In this case, we've specified three URLs *not* to download, because this is where all of our image and other non-HTML assets are located.

Note that each URL includes a wildcard symbol ("*") at the end of the path. The use of the wildcard means that any file located within these three directories will match the rule, effectively disallowing the crawler from the entire directory.

```
+atomicobject.com/*.css +atomicobject.com/*.js
```

A rule preceded by a plus (+) sign indicates something we *do* want to download.

It's important to understand that HTTrack determines rule precedence from left to right. Because these rules come after (i.e., to the right) of the rule telling us to ignore the `/assets` directory, they will overrule it. That means that the assets directory will be ignored, unless the filename ends in .css or .js. This allows us to retrieve any CSS and JavaScript files, while still excluding other asset types, like images and videos.

```
--path "~/httrack-copies/atomicobject/"
```

The –path option specifies where we want HTTrack to save downloaded files. Without this option, files are downloaded to the current working directory.

```
--verbose
```

The verbose option tells HTTrack to output its log to the Terminal, allowing us to monitor the program as it runs.

Conclusion

With the above settings, I can create a full copy of all HTML, CSS, and JS files on the Atomic website in just under four minutes. If you're looking for an efficient tool to create a copy of a website, make sure to check out HTTrack.

Conversation