[This is a description of the step-by-step technical process by which we 'scan' each Initial URL. These steps come after the initial process of building the website index.]

First off, before any scans take place, the process of ingesting the Initial URL list into the database populates the following fields: * `Initial URL` * `Initial Domain` * `Initial Base Domain` * `Initial Top Level Domain` * `Agency` * `Bureau` * `Branch` * `Data Source` * `Public` * `Filtered`

When scanning commences, this core file dictates which scans are run. Due to the nature of the code base, the scans run asynchronously (i.e. they don't necessarily run in the order they are written in the code). Each scan operates separately and don't talk to each other.

The current scans are:

- **primary** - Loads the Initial URL and analyzes the resulting Final URL, generating most of the Site Scanning data.
- **dns** - Analyzes the DNS of the Final URL using a Node.js library (instead of Puppeteer).
- **notFound** - Tests for proper 404 behavior using an https service (instead of Puppeteer).
- **robotsTxt** - Appends `/robots.txt` to the Initial URL, loads it, and analyzes the resulting `robots.txt` Final URL.
- **sitemapXml** - Appends `/sitemap.xml` to the Initial URL, loads it, and

Primary Scan

The primary scan uses Puppeteer to load a Initial URL in a headless Chrome/Chromium browser. It then (again asynchronously) runs a number of what might be thought of as scan components, the list of which can be found here and the code for which can be found here.

At the moment, these 'scan components' are: * urlScan - which loads the Initial URL and then notes whether it redirects; what the Final URL is; whether it is live; what its server status code, filetype, and base domain are; and whether the Final URL is on the same domain and same website as the Initial URL. This populates the `URL`, `Domain`, `Base Domain`, `Top Level Domain`, `Media Type`, `Live`, `Redirects`, and `Status Code` fields. * For `Live` - it is marked `TRUE` if the final server status code is one of these - 200, 201, 202, 203, 204, 205, 206. * For `Redirect` - it is marked `TRUE` if there are one or more components in the redirect chain of the request method.
* cmsScan - which looks for certain code snippets in the page html and headers that indicate the use of a certain CMS. These code snippets are borrowed from the great work of Wappalyzer, specifically the files in this folder. In order to detect the use of Cloud.gov pages, the x-server response header is also checked to see if it contains `cloud.gov pages`. This populates the `Infrastructure - CMS Provider` field.
* cookieScan - which uses Puppeteer's built in functionality to note the domains of

DNS Scan

- The IPv6 test looks within the DNS for the presence of an AAAA record. This populates the `DNS - IPv6` field.
- For the `hostname` field, only certain results that contain one of these strings are included, so as to better highlight the most common cloud services. This populates the `DNS - Hostname` field.

Not Found Scan

- A random string is added as a path after the Target URL to test how the site handles 404 errors. This populates the `Target URL - 404 Test` field.

Robots.txt Scan

`/robots.txt` is appended to the Target URL and loaded. The scan then notes whether it redirects; what the Final URL is; whether it is live; what its server status code, filetype, and file size are; whether the Final URL is live and if the entire path is `/robots.txt`; what the robots.txt crawl delay is (if it is there); and what the urls of sitemaps listed in the robots.txt are. This populates the `Robots.txt - Detected`, `Robots.txt - Target URL - Redirects`, `Robots.txt - Final URL`, `Robots.txt - Final URL - Live`,`Robots.txt - Final URL - Status Code`, `Robots.txt - Final URL - Media Type`, `Robots.txt - Final URL - Filesize`, `Robots.txt - Crawl Delay`, and `Robots.txt - Sitemap Locations` fields. * For `Robots.txt - Final URL - Live` - it is marked `TRUE` if the final server status code is one of these - 200, 201, 202, 203, 204, 205, 206. * For `Robots.txt - Target URL - Redirects` - it is marked `TRUE` if the are one or more components in the redirect chain of the request method.
* For `Robots.txt - Detected`, the analysis looks at whether `Robots.txt - Final URL - Live` is `TRUE` for its decision (as opposed to the relevant server status code).

Sitemap.xml Scan

`/sitemap.xml` is appended to the Target URL and loaded. The scan then notes whether it redirects; what the Final URL is; whether it is live; what its server status code, filetype, and file size are; whether the Final URL is live and if the entire path is `/sitemap.xml`; what the sitemap.xml item count is; and what count of urls ending in `.pdf` are in it. This populates the `Sitemap.xml - Detected`, `Sitemap.xml - Target URL - Redirects`, `Sitemap.xml - Final URL`, `Sitemap.xml - Final URL - Live`, `Sitemap.xml - Final URL - Status Code`, `Sitemap.xml - Final URL - Media Type`, `Sitemap.xml - Final URL - Filesize`, `Sitemap.xml - Items Count`, and `Sitemap.xml - PDF Countfields`. * For `Sitemap.xml - Final URL - Live` - it is marked `TRUE` if the final server status code is one of these - 200, 201, 202, 203, 204, 205, 206. * For `Sitemap.xml - Target URL - Redirects` - it is marked `TRUE` if the are one or more components in the redirect chain of the request method.
* For `Sitemap.xml - Detected`, the analysis looks at whether `Robots.txt - Final URL - Live` is `TRUE` for its decision (as opposed to the relevant server status code).

Other Notes

- In the above folders, the x.ts files are the scans/scan components themselves and the x.spec.ts files are the test files.