Holloway's Angular Static Website Builder User Guide

An user guide for operating the builder.

Version: v0.0.1 – 1st Edition

1. Table of Content

2. Prologue

Thank you for using Holloway's Angular Static Website Builder. Depending on the end user, should this document is not the same as the original website context, then it serves as a general user guide for all downstream derivatives. Should you have any inquiries, please feel free to contact my team at:

1. GitHub upstream: https://github.com/ChewKeanHo/APP_Website_Angular
2. GitHub discussion: https://github.com/orgs/ChewKeanHo/discussions

3. Legal Compliance & Licenses

By default, the builder's most upstream distribution is licensed under MIT license. For any derivatives downstream created from the upstream template repository, they're, by default, configured to be Proprietary License.

Please refer to the website repository maintainer for the actual legal requirements.

4. Operating the Builder

The builder is fairly easy to operate thanks to the Holloway's AutomataCI governance. One only needs to execute the following in sequences:

```
$ ./automataCI/ci.sh.ps1 env
$ ./automataCI/ci.sh.ps1 setup
$ ./automataCI/ci.sh.ps1 prepare
$ ./automataCI/ci.sh.ps1 test
$ ./automataCI/ci.sh.ps1 build          # build the website artifact
$ ./automataCI/ci.sh.ps1 package        # prepare for publication
$ ./automataCI/ci.sh.ps1 release        # publish
```

5. Initializing the Template

Generally, the template had configured the Angular (at least version 17) to perform server-side rendering (SSR) with pre-render enabled for static site generation (SSG). Moreover, it flattens the file structure for simplicity sake matching the Hugo Static Site Filesystem.

Once you created a repository for this template, you should:

1. update all the files inside "**src/**" directory including this document.
2. Update the key configurations inside "**srcANGULAR/angular.json**" directory such as but not limited to:
    1. **projects.architect.configurations.production.baseHref** – set your base URL here.
    2. **projects.architect.configurations.production.deployUrl** – set your base URL here.

Give the engine a run and it should be fine.

6. Modified Template Filesystem

The Angular filesystem is modified from Angular 17 boilerplate generated structures. Unless otherwise specified herein, everything should operate normally matching Angular 17 and above specifications. Otherwise, you should be aware of the following:

6.1. assets/ Directory

This is configured to be the "static" directory known in static site generation communities with the output locked at root level ("/"). Hence, anything that is placed in shall be copied directly during build.

6.2. foundations/ Directory

This is configured to house all the baseline files like the Angular required app.html base file, website-specific CSS and JS files etc that will be consumed by Angular to generate the output. They are pulled in by Angular via the *angular.json* configuration file.

6.3. contents/ Directory

This is configured to house the pages standalone "Components" in a URL based pathing directory organization. At root level ("**contents/**"), you are strongly advised not to touch the "app.html" and "app.ts" as they're used by Angular to bootstrap your application.

Any further pages (e.g. "**contents/en/**") are registered via the "**angular.routes.ts**" file which defines the URL pathing and load the required pages.

6.4. services/ Directory

This is configured to house the actual Angular UI Component (not standalone) consumed by the Pages (standalone components). It can also house other services like database and etc.

6.5. prerender-routes.txt

This is configured to tell Angular what are the pages to be per-rendered as a static page. It complies to the Angular prerender specifications (https://angular.io/guide/prerendering).

The reason "**discoverRoutes**" configuration is unused mainly due to the fact that Angular cannot guess or find out parameterized routes – only the website developer knows. That's why this text file is created from the get go.

7. PWA

The PWA is enabled by default referencing (https://angular.io/guide/service-worker-intro). The "**manifest.webmanifest**" file and all the associated icons files are located inside the "**assets/**" directory. Please update them accordingly.

8. Basics SEO

The basic files for SEO (e.g. "**robots.txt**", "**browserconfig.xml**" are created by default and located inside "**assets/**" directory. Unless required, the basic files are suffice to facilitate search engine crawlers.

8.1.1. Open Graph & X (Formally "Twitter")

This feature is not implemented since it can hamstrung a developer from one another especially when using Angular 17 SSR feature. Hence, it's best to do it inside the pages with Title and Meta classes offered by Angular libraries.

9. GitHub Pages

For those who wants to deploy to GitHub Pages seamlessly, the required ".nojekyll" and "CNAME" files are located inside "**assets/**" directory.

Please update the CNAME file matching the verifiable domain.

10. Epilogue

That’s all from us. If you wish to keep in touch, please feel free to join us at:
https://github.com/orgs/ChewKeanHo/discussions