Building a Docs as Code system

Lisha Li

About Me

- I live in Vancouver Canada
- 14+ years of experiences writing end-user and developer docs
- Experiences with doc infrastructure design:
  - Built a DITA XML documentation system
  - Built a Confluence documentation system
  - Built a Docs-as-Code documentation system

Table of Contents

What is Docs as Code and why?

An approach or system that you create and maintain your documentation using the same tools as you do with your programming code.

Benefits:

- Keep your documentation in sync with your code
- Use the same tools to review code and documentation changes
- Writers and developers can work together
- Automated testing, building, and publishing

Manage

Manage your documentation using a source control system:

- GitHub
- Bitbucket
- GitLab

Write

Write the documentation in plain text markup:

- Markdown
- reStructuredText (for Python project)
- Asciidoc

Use any text editor:

- Visual Studio Code (https://code.visualstudio.com)
- Atom (https://atom.io)
- Notepad++ (https://notepad-plus-plus.org)
- Online IDE tool such as Gitpod (https://www.gitpod.io)

Build

Build the documentation using a static site generator (SSG). SSG creates HTML pages from a template and a given content source.

The following website lists all the popular SSGs, their popularity, and the templates they use:

https://jamstack.org/generators

In general:

- Markdown - Any SSGs
- reStructuredText - Sphinx (https://www.sphinx-doc.org)
- AsciiDoc - Antora (https://antora.org)

Look

Customize the look of the HTML pages using standard web technologies:

- CSS
- JavaScript
- Customize the default template
- Create your own template

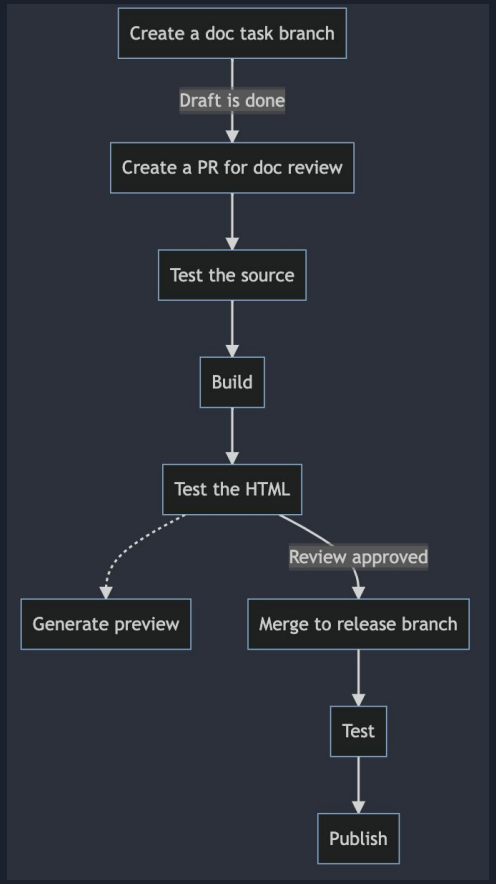
Test

Test the documentation source and output using content and code linters and HTML checkers:

- Style and prose linter
  - Vale (https://vale.sh/docs/)
- HTML checker
  - Htmltest (https://github.com/wjdp/htmltest)
  - Muffet (https://github.com/raviqqe/muffet)
  - HTMLProofer (https://github.com/gjtorikian/html-proofer)
  - LinkChecker (https://github.com/linkchecker/linkchecker/)
- Code linter
  - Pylint (https://pylint.pycqa.org)
  - JSLint (https://www.jslint.com/)

Docs as Code CI/CD Workflow

- CI refers to continuous integration
  - An automation process that changes are regularly built, tested, and merged to repo
- CD refers to continuous deployment
  - An automation process to release changes from repo to production



Demo

I will do a demo using a docs as code demo system I built in GitHub
https://github.com/lilisha100/docs-as-code-demo

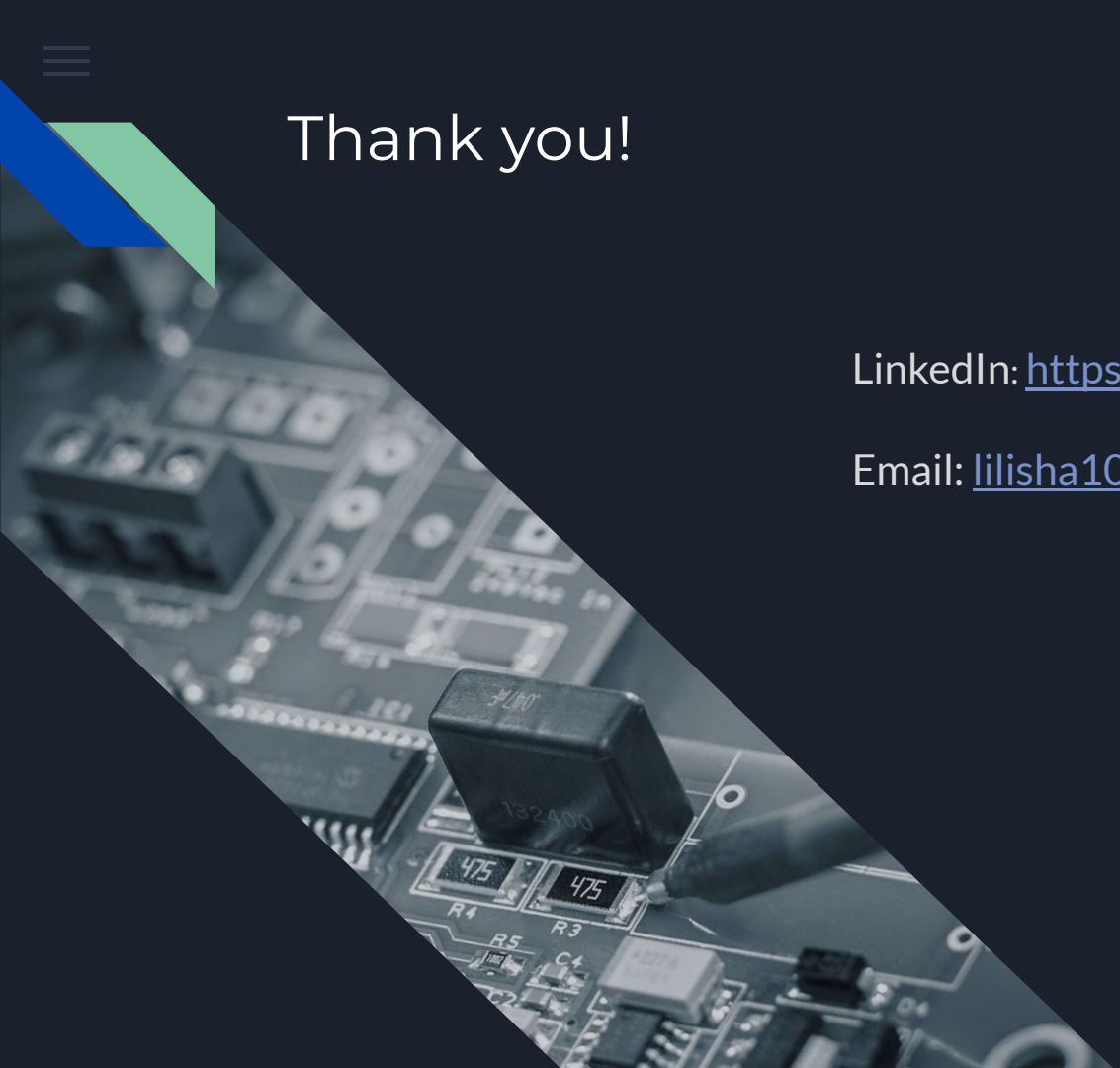
- DocFX to build developer guide/client and REST API references
- Content source in markdown and from docfx-seed project
- Vale
- Htmltest
- Gitpod for doc editing and preview
- GitHub Pages to host the doc online
- GitHub Action to automate the workflow

Summary

Things you need to set up a docs as code system:

- Git repo
- Markup language
- Authorizing tools
- Static Site Generator
- Doc testing tools
- Website design
- Website hosting
- CI/CD workflow

Thank you!



LinkedIn: https://www.linkedin.com/in/lishali/

Email: lilisha100@hotmail.com