

GitHub Pages

The current website for UMN Kernel Object is set up as a [GitHub pages](#) repository. This is a relatively simple system that publishes files pushed to a GitHub repository to a publicly accessible website.

Our web content is hosted in the repository [UMN-Kernel-Object/UMN-Kernel-Object.github.io](https://UMN-Kernel-Object.github.io). This repository naming pattern is required to use GitHub pages. Any files that are added to this repository (most importantly, any HTML, CSS, and JavaScript files) are published to the repository's corresponding website. GitHub automatically updates this website's content whenever a push is made to the main branch of this repository.

If you visit <https://UMN-Kernel-Object.github.io>, you will notice that the web page is a simple placeholder with the contents "Hello, World!" – the contents of the `index.html` file in our GitHub pages repository.

Automatic Documentation Generation

Our group's documentation website is located at <https://UMN-Kernel-Object.github.io/docs>. The content for this documentation comes from the `docs` folder in our GitHub pages repository. Currently, this web content is not written by hand. Instead, it is automatically generated by the `sphinx` tool from files in the RST markup language. This is the same infrastructure behind the documentation for the [Linux kernel](#) and the [Python](#) programming language.

The documentation's source RST files are maintained in a separate GitHub repository: [UMN-Kernel-Object/documentation](https://UMN-Kernel-Object.github.io/documentation). This repository uses GitHub's "Actions" feature to automatically generate HTML files from the original RST and push these to the main branch of UKO's GitHub pages repository.

GitHub actions are specified using a YAML file. You can find the definition of an action to perform automatic RST compilation and a push to GitHub pages under the directory `.github/actions` in our documentation repository.

Here is a brief overview of what the current GitHub action does:

1. Clones a copy of the documentation repository
2. Sets up a `sphinx` Docker container and uses it to compile RST into HTML and CSS files suitable for publication on the web.
3. Sets up a temporary git installation with a private key granting this action access to our GitHub pages repository.
4. Clones a copy of the GitHub pages repo, copies in our newly compiled HTML and CSS files
5. Makes a new commit to the pages repository with a message indicating the current date and time

6. Pushes this commit to the main branch of the UKO GitHub pages repo

Note that some care is needed because this GitHub action requires access to a private SSH key to give it the ability to automatically push to GitHub pages. You may have noticed that the action's temporary private key file is generated from `secrets.DEPLOY_KEY_PRIVATE` in the action's YAML file.

This leverages GitHub's "Secrets" feature in which sensitive content can be stored alongside the usual code in a repository. You can find the definition of this secret value if you go to the documentation repo's settings page, click on "Secrets and variables" and then click on "Actions" from this drop-down menu.

The UMN-Kernel-Object.github.io repository has the corresponding public key listed as a "Deploy key" (found under its settings menu), meaning any person (or code) in possession of the private key is allowed to access and push to this repository. The current deploy key is titled "Kernel DocBot" as it is only used by our GitHub action to automatically update documentation content.

Modifying Our Website

If you would like to make changes to our group's website, the easiest way is to simply push changes to our GitHub pages repository. You can use whatever HTML, CSS, and JavaScript you'd like.

Our documentation is handled slightly differently. You should instead take the following steps:

1. Clone our documentation repository and make whatever changes you'd like as local commits. You'll need to learn RST to do this, but many tutorials are available on the web.
2. Push your changes to our documentation repo (or submit a patch and have them merged into the main branch if we have a documentation maintainer)
3. Verify that the documentation repo's GitHub action completed successfully, applying your changes to our GitHub pages website.
4. You should then see your changes on our documentation website!

Future Work

Here are a few ideas for valuable contributions one could make to our documentation:

- Create a "landing page" for our website. This should say something about who we are and what we do so that interested students (or other collaborators) can visit this page to learn about us.
- Clean up our existing documentation website. The current site has improper formatting, missing biographies, and has not yet merged in the Caesar tutorial branch.
- Learn how to make our documentation look halfway decent. Sphinx probably has some concept of "themes" we can use for this.
- Add new material to our documentation reflecting the work of our subgroups