ASSOCIATION OF CANADIAN MAP LIBRARIES AND ARCHIVES
BULLETIN

Geospatial Data and Software Reviews

Meg Miller
University of Manitoba

GitHub and Jekyll for Publishing GIS Workshop Content

Authors
Evan Thornberry, GIS Librarian, University of British Columbia Libraries

Phil White, Earth, Environment & Geospatial Librarian, University of Colorado Boulder Libraries

Abstract

In this article, we describe GitHub in simple terms and demonstrate its practical value as a platform for delivering workshop instruction. This article stems from a virtual pre-conference workshop we delivered at the 2020 annual meeting of the Western Association of Map Libraries (WAML). We describe an easily replicated workflow for publishing workshop materials and documentation to the web using GitHub Pages and provide a GitHub repository that readers of this article can copy and customize to suit their own workshop needs.

Introduction

Background

During the past year, virtual instruction has moved from being a nice, value-added service to an absolute essential function of most academic libraries. This shift has presented both challenges and opportunities to libraries that routinely deliver workshops on technical topics—including geographic information systems (GIS). In an environment where we are unable to deliver instruction and assist students in person, it is imperative that we provide quality documentation to our learners so they can accomplish workshop objectives with as little friction as possible. We must also recognize our virtual learners' desire to participate in workshops asynchronously and at their own pace. For these reasons, virtual workshop instructors should strive to provide simple-to-follow workshop documentation that exists in an easily-accessed, modular, reproducible, and stable online environment, such as a properly managed website.

Creating dedicated websites for workshop instruction may sound like overkill and intimidating—librarians are not often trained in web-development and might also encounter regular IT barriers for publishing documents online. There may also be concerns about the cost of publishing workshop content to the web. But there are excellent free and flexible options for hosting workshop content on the web, including the version control and hosting platform GitHub. While many may think of GitHub as a website where developers publish and share computer code, it can in fact be used for a much wider range of things. For example, GitHub's file storing options make it advantageous for hosting static workshop webpages and data files. It also has several built-in project management features which can facilitate the development and management of workshop content across several collaborators.

Demystifying GitHub

With over 55 million users worldwide, GitHub is a popular website with users ranging from beginner coders to seasoned software developers. GitHub is essentially a community of users creating, maintaining and sharing code. GitHub is modelled on the free and open-source version control system Git, and from it borrows several terms and concepts used for creating, collaborating on, and documenting project-based work.

Git uses "repositories" to compartmentalize projects on a file system, and tracks changes to repository files as they are updated. When installed and used on a personal computer, for example, Git can work with file directories (or folders) containing data or to document files related to a project—once Git is initialized inside this directory it becomes a Git repository. When not using Git, files within the folder can be used and edited normally, however when files are changed or modified, Git can record the changes. Using Git, your repository preserves all earlier versions of the files stored in the folder. In the event that you make a mistake or accidentally delete a file, you can leverage the repository to revert back to an earlier iteration and retrieve past versions of your files. This is a basic concept of how Git is used as a version control system.

GitHub extends the functionality of Git by providing free cloud back-up and collaboration tools. Git users only need a free account on GitHub before they can "push" updates to the files from their local repository to a remote repository on GitHub. The owner of the remote repository can grant access to other GitHub users working on the same project, allowing for multiple "contributors" to a single project. Additionally, there are several other GitHub-based project management tools for working collaboratively which are covered below.

GitHub Features and Workflows

GitHub integrates several features and workflows which can be useful for creating and collaborating on content maintenance and development.

Forking and Cloning

Forking and Cloning are two main ways GitHub repositories can be duplicated. Forking someone else's repository will create a copy of it to be used as a foundation for your own work, on your own account (Figure 1). This method for copying a repository stays within the GitHub platform. Cloning a repository will copy an existing repository to a computer, which can be synced with your GitHub account. In either case, once a repository is forked or cloned, you own the copy and can work without disrupting the original.

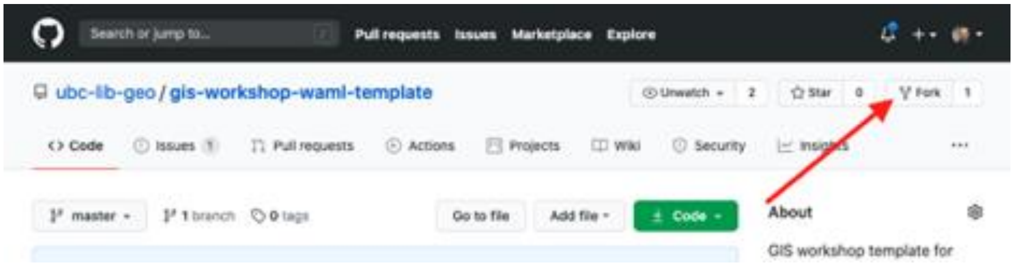


Figure 1. Forking allows one to copy a repository to their own GitHub account.

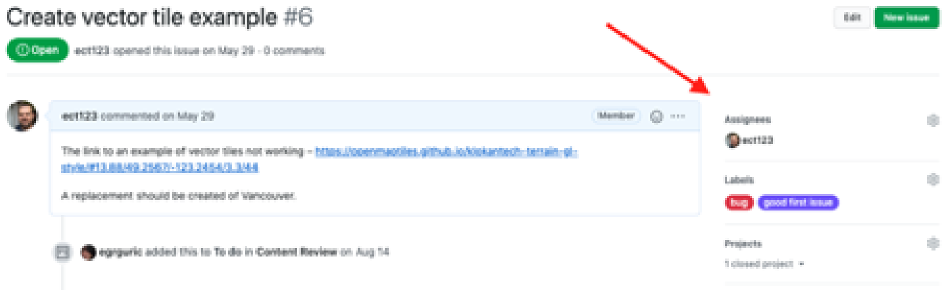
Pull Requests

A core feature of GitHub is its ability to create several "branches" of a single repository. Branches can be used by specific users to make changes to a workshop without disrupting the "main" production branch. Once work on a branch is completed, it is integrated to the "main" branch through a Pull Request. When submitting a Pull Request, the project owner is required to review work that has been done before accepting the changes.

Pull Requests can be submitted by users not belonging to the repository's collaborators. For example, workshop students can submit Pull Requests they think could be potential improvements to the workshop content. In this case the instructor would be notified of the request, and begin reviewing the improvement to determine if it should be integrated into the workshop.

GitHub Issues

GitHub's task tracking system is called Issues and is used for logging and managing work to be completed (Figure 2). Included in Issues are several "Labels" which can be applied to tasks so they can be organized and assigned, making it apparent what kind of work is in progress, good for beginners, part of a larger project, or considered future work.



Create vector tile example #6

ect123 opened this issue on May 29 · 0 comments

ect123 commented on May 29

The link to an example of vector tiles not working - https://openmaptiles.github.io/klokantech-terrain-gl-style/#13.88/49.2567/-123.2454/3.3/44

A replacement should be created of Vancouver.

egrguric added this to To do in Content Review on Aug 14

Figure 1: GitHub Issues enable tasks to be logged and assigned to a user.

Issues can be logged by any project collaborator—making them a low-barrier entrance for staff or students hoping to learn more. For example, students hoping to learn more about GIS can be added as collaborators to a GIS workshop repository and review the workshop on their own, adding Issues for things that need updating due to changing software versions, improvements in clarity, spelling and grammar, etc. Issues also facilitate a way for assigning work to specific users collaborating on a GitHub project, so work that is tagged as “good first issue” can be assigned to students or staff who might be less experienced with the subject.

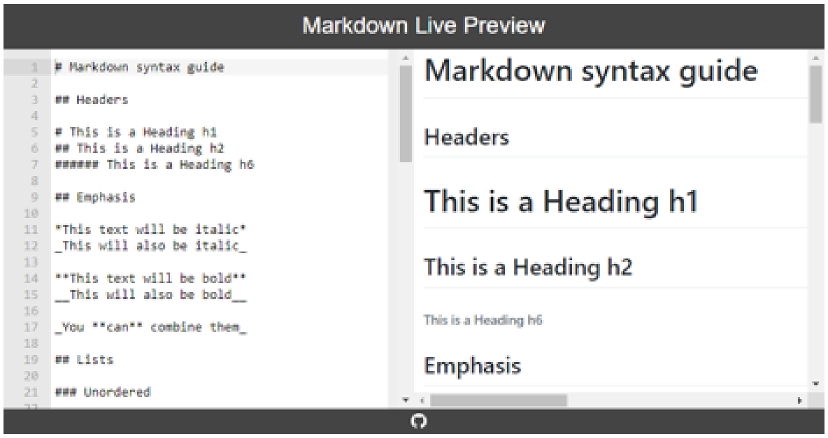
Organizational Accounts

A GitHub Organization is a group of shared accounts that have adequate file permissions to work on several simultaneous projects, making them an option for library departments or units working together on publishing open workshop content. Organizations can have an unlimited number of user accounts and public repositories.

GitHub Pages & Jekyll

One of GitHub’s most useful features for publishing open content is GitHub Pages. Pages publishes the content of a GitHub repository as a website by using HTML, CSS, and JavaScript files contained in the project. After updating a small number of GitHub Pages settings, your repository will be published as a public webpage with a standard URL (see Creating a GitHub Pages website). Only publicly viewable repositories can be published online. Public repository files can be viewed and downloaded, but not modified by outside collaborators unless it is approved through a Pull Request.

While one could use “hand coded” website files to create a GitHub Pages website, it can also be avoided by using Jekyll. Jekyll is a static website generator that transforms plain-text documents into web-ready files like HTML. Rather than using a complex markup language like HTML to structure and create web documents, Jekyll uses a simplified markup language called Markdown (Figure 3). Markdown syntax is very logical and easy to read, thus has a very low learning curve when used regularly. Additionally, there are numerous educational resources on the web for creating Markdown, or converting other documents like Excel tables and Word documents into Markdown. This means that GitHub Pages—which is powered by Jekyll—has the ability to create websites out of a collection of easy-to-create Markdown documents stored in a GitHub repository.



```
# Markdown syntax guide

## Headers

# This is a Heading h1
## This is a Heading h2
###### This is a Heading h6

## Emphasis

*This text will be italic*
_This will also be italic_

**This text will be bold**
__This will also be bold__

_You **can** combine them_

## Lists

### Unordered
```

Markdown syntax guide

Headers

This is a Heading h1

This is a Heading h2

This is a Heading h6

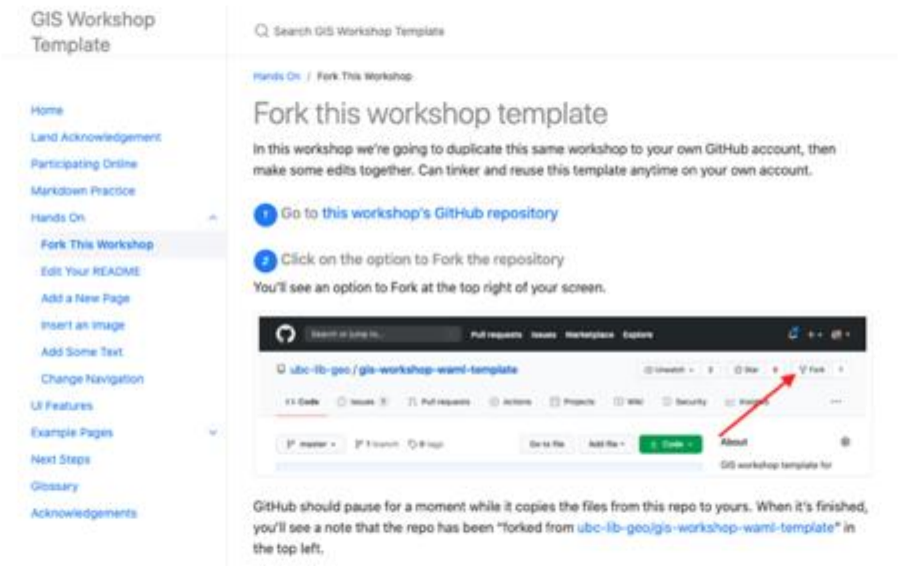
Emphasis

Figure 3. Raw Markdown (left) and its representation on a webpage (right) using markdownlivepreview.com.

Jekyll can also be stylistically and functionally customized by modifying or including several additional website files. However, extensive customization can be avoided through the use of a Jekyll website theme. Jekyll themes are packages of files that can be installed in a repository and used as a turn-key website for specific purposes and style—such as a blog, personal, or documentation website. The easiest method for selecting a theme for a GitHub Pages website is to use the GitHub Pages "Theme Chooser" which provides several basic styles for a website. However, you can also find several user-developed themes, or an openly licensed repository that is being used to generate a Jekyll website to copy and create your own.

Teaching GIS with GitHub and Jekyll Pages

In our example described below from the WAML pre-conference workshop, we provide a GitHub repository which can be used as a Jekyll theme template for delivering virtual GIS workshops. The template is built using the Just The Docs Jekyll theme, which presents a minimalist design and is ideal for focusing viewers on the instruction content. We encourage readers to visit the workshop repository and the GitHub Pages workshop website generated from the repository (Figure 4). We invite you to fork the repository to your own GitHub account and participate in the workshop asynchronously, or modify and adapt the template for your own instruction purposes.



Fork this workshop template

In this workshop we're going to duplicate this same workshop to your own GitHub account, then make some edits together. Can tinker and reuse this template anytime on your own account.

1 Go to this workshop's GitHub repository

2 Click on the option to Fork the repository

You'll see an option to Fork at the top right of your screen.

GitHub should pause for a moment while it copies the files from this repo to yours. When it's finished, you'll see a note that the repo has been "forked from ubc-lib-geo/gis-workshop-waml-template" in the top left.

Figure 4. A published GIS Workshop Template website using Jekyll.

WAML Workshop Template

A virtual workshop using GitHub and Jekyll was delivered as part of the program for the Western Association of Map Libraries (WAML) annual meeting in October, 2020. Titled *Teaching Workshops in a Virtual World Using GitHub Workflows*, the session included an introductory section which covered the basic GitHub and GitHub Pages workflows and terminology, and a hands-on section where students forked a GitHub repository, published a live workshop content website using a premade template, and created customized workshop pages.

The workshop was conducted from a published workshop template, which included a homepage with introductory information, an embedded slide presentation, and a hands-on section with step-by-step instructions. A link to the workshop website was provided to attendees at the beginning of the session for them to follow along with the live instruction. The session also used the workshop website as the template for students to fork and customize on their own GitHub account. This allowed students to tinker with working content, and gave them a usable foundation for a GIS workshop which they could customize further after the session ended.

A detailed summary of the session can be found at https://ubc-lib-geo.github.io/gis-workshop-waml-template/.

Conclusion

The GitHub and Jekyll ecosystem works as a simple and free way to publish open workshop and other educational content and can enhance workshop delivery through the use of web content like links and interactivity. The GitHub platform also includes several features that could ease content maintenance and collaboration, such as Issues, Collaborators, Organizations, and Pull Requests. It embraces openness and demonstrates to workshop participants the means by which they might also make their work open, shareable, and reproducible. GitHub also encourages the collection and use of documentation. Every repository requires a readme file, and Issues and Pull Requests document how a repository has been modified over time. And of course, an obvious benefit is that it is free—mitigating concerns about the cost of hosting instruction content online.

There are a few downsides to using GitHub, but we believe these are minimal. Like all software and technology, GitHub requires some practice to become comfortable with. Although most advanced GitHub users use a command-line shell for working with Git and GitHub, the GitHub web interface provides an easy entry-point for new users. All of the procedures in our workshop can be completed without use of the command-line, and walking through the workshop should help new users gain familiarity with GitHub workflows. GitHub also lends itself well to hosting workshop data. While it is designed to work with plain-text documents, it can also store non-text files like shapefiles and images. GitHub does have file and repository size limits, but we find these limits completely adequate for the purpose of distributing data in the context of a workshop.