

Guide to Mozu GitHub

Using the GitHub app for your local desktop is tremendously helpful for working in GitHub. You can use the website portal or the command shell if you prefer.

You will need to download the repo for MozuDocumentation and branch gh-pages to your desktop only once, or again if you move machines. You can work within that repo locally and push your changes through the app.



All content in this repo is PUBLIC! You may not want to post files that should be paid for, hidden behind a login, or internal only. Partner docs may be ok, but may need permission.

The repo is accessible here: <https://github.com/Mozu/MozuDocumentation> This is the “code” view. You use this to download your repo or work with the repo.

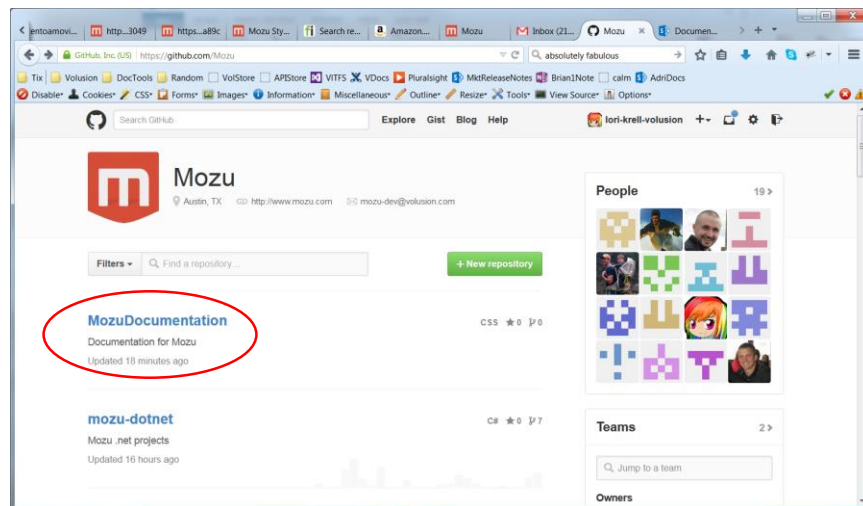
The website link is here: <http://mozu.github.io/MozuDocumentation/> You can use the links to the PDFs and other files directly from this location. For example, <http://mozu.github.io/MozuDocumentation/GettingStarted/GettingStartedwithMozuSiteBuilder.pdf> could be used in your SiteBuilder pages in Drupal or in email to someone, even your release notes!

This file is also saved in the repo if you ever need to review it:

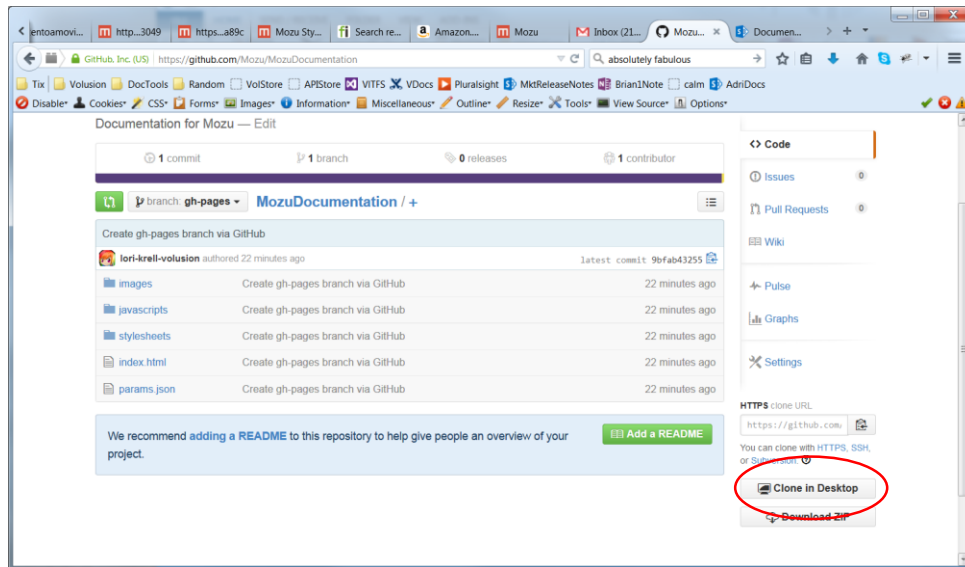
<http://mozu.github.io/MozuDocumentation/GuidetoMozuGitHub.pdf>

Download the Repo

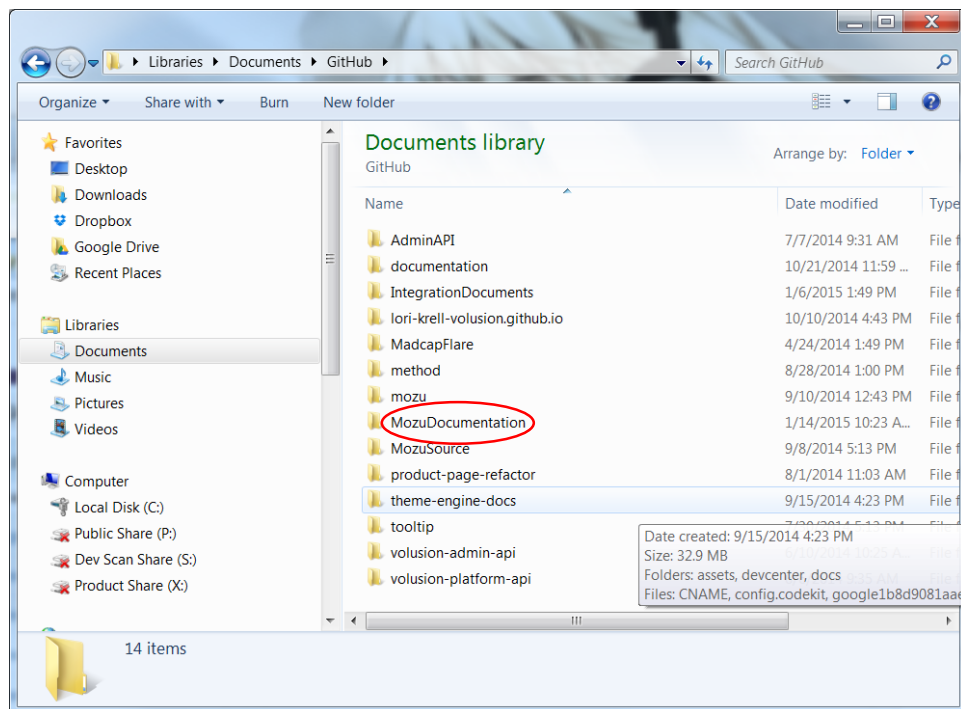
1. Visit the repo online through GitHub: <https://github.com/Mozu>
2. Click MozuDocumentation.



3. You should see a page with files and a branch of gh-pages. This is the “website” branch for the repo that should always be active and public.
4. Click Clone in Desktop.



5. Select a location for the repo. A folder will be created called MozuDocumentation. You can save this into a current location with other repos, or separate in another location. GitHub will keep track for you. For example, I keep all of my GitHub repos in a GitHub folder. Any content added to these folders will automatically be detected by GitHub for publishing back into the repo!



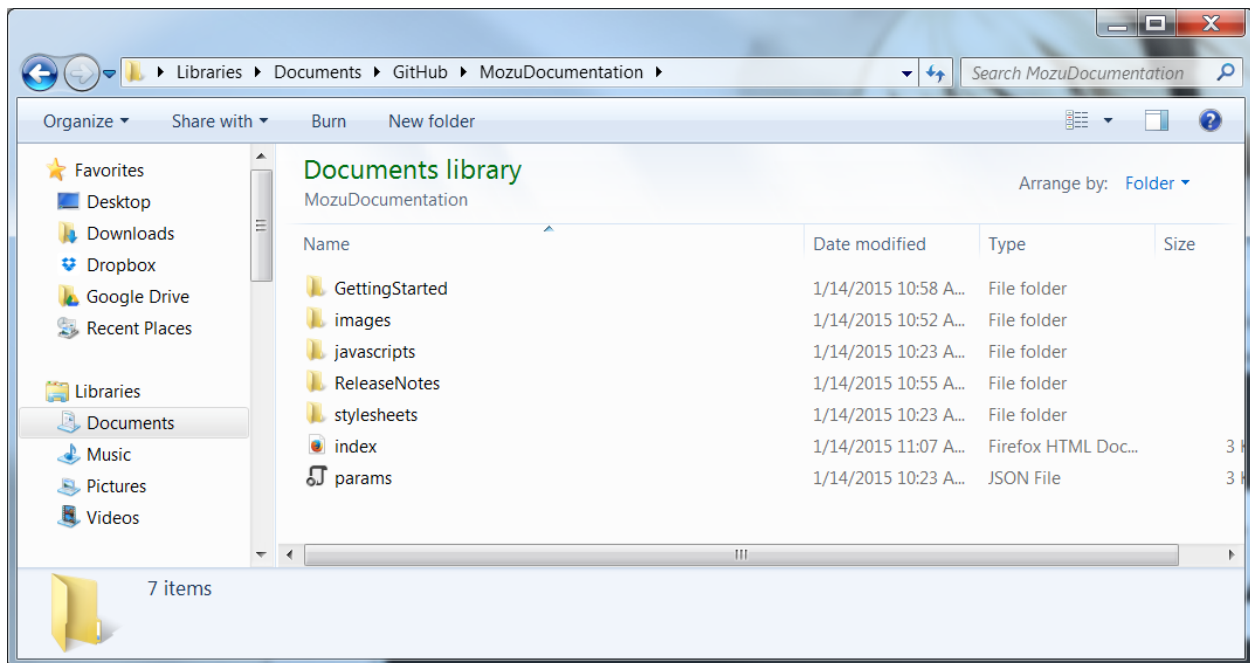
Adding a PDF or Output to GitHub

You will work directly within the copied repo branch in your local. You can continue to work in these files editing content and adding files prior to publishing. As you make changes and add files, GitHub will automatically recognize the changes, display the delta, and walk you through the publishing process through the app.

The repo will have the following folders, files, and structures. You can add PDFs to the ReleaseNotes and GettingStarted folders, or make new folders for your content. If you want to post HTML5 output from Flare, make a new folder, paste your content, and link to the landing page or index file in that folder.



File names should not have spaces in them. Use camel case, dashes, and underscores as needed. Spaces can make linking problematic.



How to Add Content:

1. With multiple writers working in the repo, open your GitHub app and check for any updates. Click the Sync button at top-right to make sure you have the latest files locally.



2. Open your folder for MozuDocumentation.
3. Add your PDF or HTML output files to the appropriate folders. Or make a new folder for the content.
4. Edit the index HTML file in the main folder of the repo. You can use Notepad, Notepad++, or other simple text program.
5. Each section of documentation is in an `<h4></h4>`: Release Notes, Getting Started, and App Documentation. All app docs should be in the MozuIntegrations repo and added to that Index file.
6. Copy and paste or follow the conventions of the code to add a new line item. For release notes, they are in descending order, latest at the top. Remember no spaces in the names and the folder name. Add a short description to help anyone landing on this page.

```

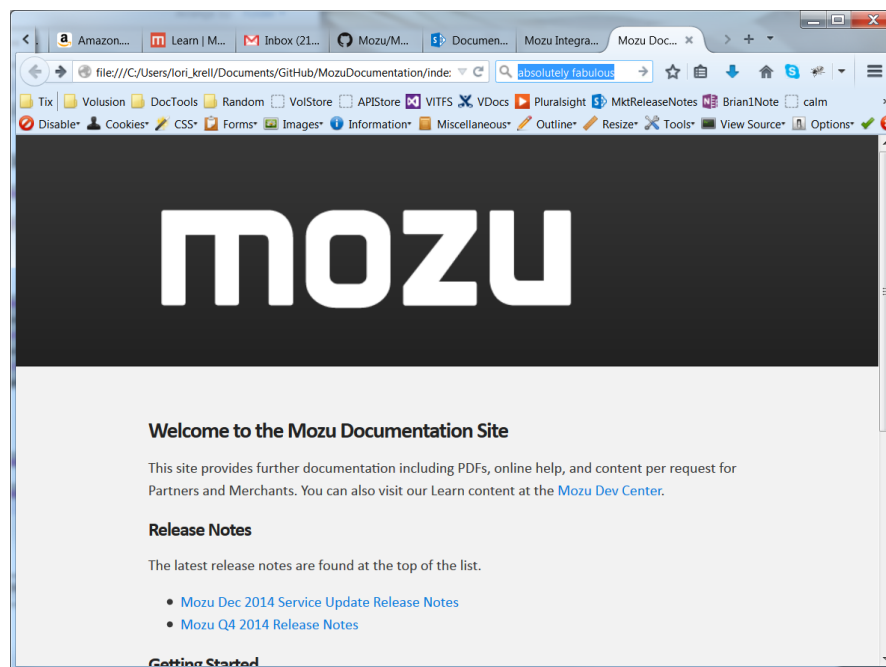
<h4>Release Notes</h4>
<p>The latest release notes are found at the top of the list.</p>
<ul>
<li><a href="ReleaseNotes/MozuDecember2014ServiceUpdateReleaseNotes.pdf">Mozu Dec 2014
Service Update Release Notes</a></li>
<li><a href="ReleaseNotes/MozuQ42014ReleaseNotes.pdf">Mozu Q4 2014 Release Notes</a></li>
</ul>

<h4>Getting Started</h4>
<ul>
<li><a href="GettingStarted/GettingStartedwithMozuAdminGeneralSettings.pdf">Getting
Started with Mozu Admin General Settings</a> : Configure channels, locations, shipping, payment,
checkout, tax, user, role, and app settings.</li>
<li><a href="GettingStarted/GettingStartedwithMozuCatalog.pdf">Getting Started with Mozu
Catalog</a> : Create catalogs, product attributes, product types, categories, products, and
more.</li>
<li><a href="GettingStarted/GettingStartedwithMozuSiteBuilder.pdf">Getting Started with
Mozu Site Builder</a> : Create, configure, and manage your Mozu themes, widgets, pages, templates
and more in Site Builder.</li>
</ul>

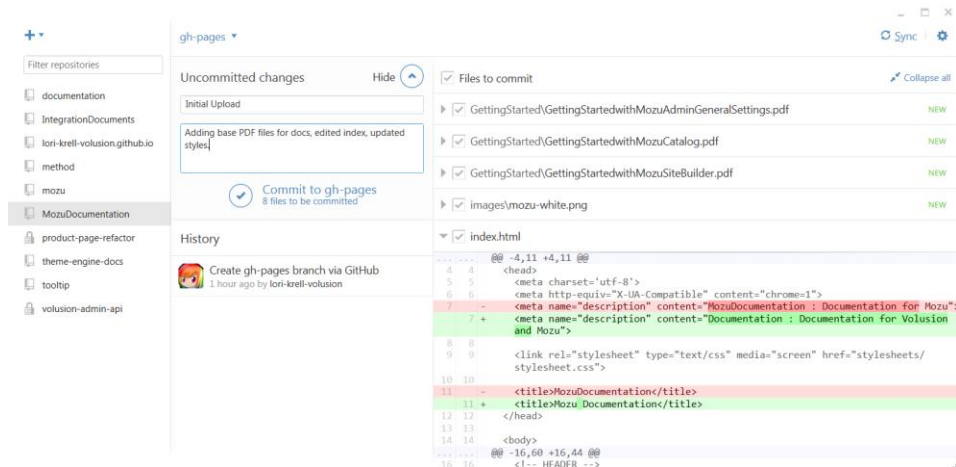
<h4>App Documention</h4>
<ul>
<li><a href="http://mozu.github.io/IntegrationDocuments/">Mozu App Documentation</a> :
Repository of online help and PDF content for Mozu apps.</li>
<li><a href="http://mozu.github.io/IntegrationDocuments/Mozu-App-Certification-
Guide.pdf">Mozu App Certification Guide</a>
</ul>

```

7. Save the Index file changes. Open that file to get a preview in your browser.



8. When all work is complete, open your GitHub app. Any updates display to the right with an Uncommitted changes section.
9. Enter a Summary and Description for the changes.
10. Click Commit to gh-pages.



11. When the commit is packaged and ready to publish, click Sync.



12. All changes upload. If merge issues occur, they will display in the app. When complete, a new history entry displays. You can click this summary to see all changes committed.

You are done! You can view the changes immediately after sync here:

<http://mozu.github.io/MozuDocumentation/>