# Building an External Documentation Center

<br>

In this tutorial, we provide detailed steps and guidance for users who want to build their own technical documentation center and host the documents on GitHub. This tutorial is suitable for users with a basic computer operation knowledge, but with limited technical expertise and familiarity with Git, GitHub, and Markdown. By following this tutorial, you will learn how to set up a Git environment, register for a GitHub account and authorize, configure the VS Code editor, and use Markdown to write and push documents, ultimately establishing a complete external documentation center.

Through this tutorial, you will be able to:

\* Install and configure Git to establish version control for your documentation center.

\* Register and authorize a GitHub account to host your documents.

\* Install and configure the VS Code editor to provide an efficient documentation editing environment.

\* Master the basic grammar of Markdown to write formatted documents.

\* Use the knowledge gained to create, edit, and push documents to a GitHub repository, establishing your own documentation center.

## Operating Environment

<br>

The software and versions used in this tutorial are as follows:

\* Operating system: Windows 11

\* Git: 2.41.0.windows.3

\* GitHub: Free, pro, and team

\* Read the Docs: Version 3.1

\* VS Code: Version 1.81

## Prerequisites

<br>

Before starting, please ensure that you have knowledge of the following:

1. The principles of Git branch management and the meanings of common Git commands such as Clone, Pull, Commit, Stage, Push, Merge, etc. For more information, refer to the [Git Reference](https://git-scm.com/docs).

2. Basic GitHub operations and its relationship with Git. For more information, refer to the [GitHub getting started documentation](https://docs.github.com/zh/get-started).

3. Understanding of Read the Docs. For more information, refer to [Read the Docs: documentation simplified](https://docs.readthedocs.io/en/stable/).

4. Interface operations of VS Code and its relationship with Git commands. For more information, refer to the [Introduction to Git in VS Code](https://code.visualstudio.com/docs/sourcecontrol/intro-to-git).

5. Basic Markdown syntax. For specific syntax, refer to the [Markdown Guideline](https://www.markdownguide.org/basic-syntax/).

## Operating Steps

### Step 1: Download Git

1. Go to the official Git website download page: https://git-scm.com/downloads.

2. On the download page, your operating system will be automatically detected as Windows, and the installation program for Windows will be provided automatically. Click the "Download" button.

3. After the download is complete, double-click the downloaded installation program file and follow the installation wizard to proceed with the installation process. During the installation process, you can choose installation options and installation paths, and usually you can keep the default settings.

4. Wait for the installation to complete, then in the terminal enter ``git --version`` to verify if Git has been installed correctly.

### Step 2: Register GitHub

1. Go to the [GitHub official website](https://github.com/) and click on the \*\*Sign up\*\* button on the top right corner.

2. Follow the registration prompts to create a GitHub account.

### Step 3: Register Read the Docs

1. Go to [readthedocs.org](https://readthedocs.org/) and click on the \*\*Sign up\*\* button on the top right corner.

2. On the account registration page, select \*\*Sign up with Github\*\*.

3. Follow the prompts on the page to authorize your account.

4. Fill in the necessary information such as username and email, then click \*\*Create account\*\* to register your account.

### Step 4: Download the Editor (VS Code)

1. Go to the [VS Code official website](https://code.visualstudio.com/Download) to download the VS Code client.

2. Open the VS Code installation program locally and follow the prompts to install it.

### Step 5: Authorize GitHub Account in the Editor

1. Open the VS Code editor, click on the left \*\*Source code (Source Control)\*\* icon, then click \*\*Clone Repository\*\*, and then click \*\*Clone from GitHub\*\*.

2. A prompt will appear asking you to log in to GitHub, click \*\*Allow\*\*, which will open a browser page for you to log in to your GitHub account.

3. After logging in to GitHub in the browser, return to VS Code.

4. Allow VS Code to open the GitHub URI in a new window.

5. After a few seconds, you can click on the \*\*Accounts\*\* on the left of VS Code to confirm that your VS Code account has been successfully logged in to GitHub.

### Step 6: Configure the Editor (Optional)

It is recommended to install the following two extensions to improve your Markdown writing experience. If you do not need these additional features, you can still continue to use the regular Markdown editor.

\* Markdown All in One:

\* Function: This extension provides a series of shortcuts for the Markdown editor to insert common Markdown syntax. It simplifies the process of inserting links, images, headings, lists, and other commonly used elements.

\* Advantages: You can trigger automatic completion by typing short commands to insert specific Markdown formats, saving typing time and reducing the possibility of errors.

\* Markdown Preview Enhanced:

\* Function: This extension provides enhanced Markdown preview functionality that goes beyond the basic preview. It supports MathJax, flowcharts, sequence diagrams, and other extended features, as well as multiple themes and styles.

\* Advantages: You can preview the Markdown rendering effects in real-time inside the editor, and use more advanced functions to enhance document readability and presentation. For example, you can draw complex flowcharts and charts to make your documents more colorful and diverse.

To install these extensions, follow these steps:

a. Open the VS Code editor, click on the left \*\*Extensions\*\* icon, then search for and install the following extensions: - Markdown all in one - Markdown preview enhanced b. After installation, click on the left \*\*Extensions\*\* icon again, then open the \*\*Installed\*\* tab to see the downloaded extensions.

### Step 7: Create a Project

There are two ways to create a project in a GitHub repository: by copying an existing repository or creating a new one. To quickly configure the Read the Docs repository, it is recommended to directly copy an existing Read the Docs configuration repository to your personal space:

1. Visit an existing GitHub repository such as ``https://github.com/AI-Assisted-Technical-Communication/AI-Translation`` in GitHub.

2. Refer to [Fork a repo](https://docs.github.com/en/get-started/quickstart/fork-a-repo), copy the repository to your personal space.

### Step 10: Sync to Read the Docs

1. Log in to Read the Docs and select \*\*Import a Project\*\* on your account homepage.

2. On the Import a Repository page, you will see a list of public projects from your GitHub account.

\* If your documentation repository appears in the list, click on the repository name to automatically connect to your GitHub repository.

\* If your documentation repository is not in the list, select \*\*Import Manually\*\* for manual configuration.

3. Follow the prompts to create your project and configure initial settings.

Notes:

- Ensure that you select the correct GitHub repository on Read the Docs and enter the repository name or URL accurately.

- During project configuration, make sure to choose the appropriate document type and default branch to ensure correct document building and version control.

- Ensure that your repository includes the ``readthedocs.yml`` and ``conf.py`` files to configure the building process. These files can be used to define build settings, specify dependencies, and more. Configure the build process according to your specific needs. For example, you can specify the document building tool, install dependencies, set up the build environment, etc.

- After creating the project, Read the Docs will automatically build the documentation once. Check the build status in a timely manner and verify the generated documentation. If any issues arise, you can return to the configuration page to make adjustments.

### Step 11: Check Online Documentation

After Read the Docs automatically builds your documentation, you can click on the \*\*Documents\*\* link in the top right corner to view and read the built documentation.

## Related Information

\* Git Official Documentation: <https://git-scm.com/book/zh/v2>

\* GitHub Official Documentation: <https://docs.github.com/cn>

\* VS Code Official Documentation: <https://code.visualstudio.com/docs>

\* Read the Docs Documentation: <https://docs.readthedocs.io/en/stable/>