

Read the Docs tutorial

In this tutorial you will learn how to host a public documentation project on Read the Docs Community.

Note

Find out the [differences between Read the Docs Community and Read the Docs for Business](#).

In the tutorial we will:

1. Import a Sphinx project from a GitHub repository (no prior experience with Sphinx is required).
2. Tailor the project's configuration.
3. Explore other useful Read the Docs features.

If you don't have a GitHub account, you'll need to [register for a free account](#) before you start.

Preparing your repository on GitHub

1. [Sign in to GitHub](#) and navigate to the [tutorial GitHub template](#).
2. Click the green `Use this template` button, then click `Create a new Repository`. On the new page:

Owner

Leave the default, or change it to something suitable for a tutorial project.

Repository name

Something memorable and appropriate, for example `rtd-tutorial`.


Visibility

Make sure the project is "Public", rather than "Private".

3. Click the green `Create repository` button to create a public repository that you will use in this Read the Docs tutorial, containing following files:

`.readthedocs.yaml`

Read the Docs configuration file. Required.

  [stable](#) ▼

`README.rst`

Description of the repository.

`pyproject.toml`

Python project metadata that makes it installable. Useful for automatic documentation generation from sources.

`lumache.py`

Source code of the fictional Python library.

`docs/`

Directory holding all the fictional Python library documentation in reStructuredText, the Sphinx configuration `docs/source/conf.py` and the root document `docs/source/index.rst`.

ericholscher Merge pull request #52 fr... 00dc2a7 19 hours ago 3 commits

File	Description	Last Commit
docs	Use a YAML file and Python requirements	yesterday
.gitignore	Basic structure	2 years ago
.readthedocs.yaml	Use a YAML file and Python requirements	yesterday
README.rst	Basic structure	2 years ago
lumache.py	Basic structure	2 years ago
pyproject.toml	Basic structure	2 years ago

README.rst

Template for the Read the Docs tutorial

Contributors 3

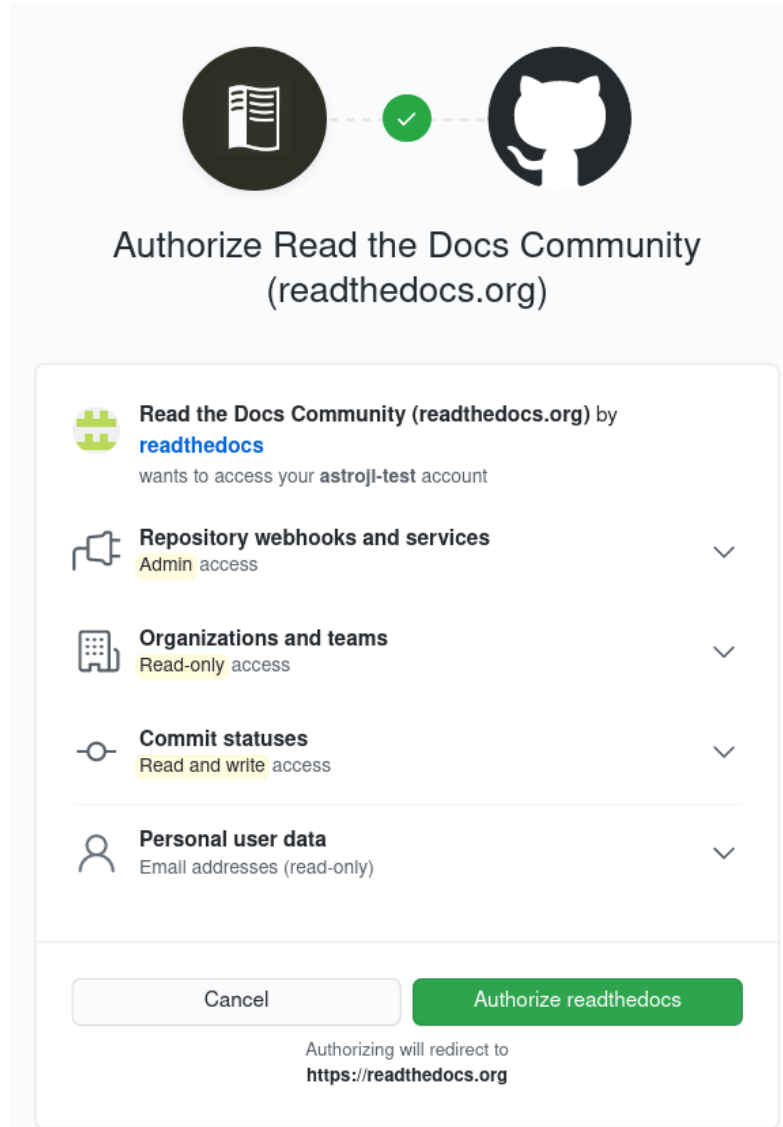
- ericholscher Eric Holscher
- humitos Manuel Kaufmann
- astrojuanlu Juan Luis Cano Rodríguez

GitHub template for the tutorial

Creating a Read the Docs account

To create a Read the Docs account: navigate to the [Sign Up page](#) and choose the option

`Sign up with GitHub`. On the authorization page, click the green `Authorize readthedocs` button.




GitHub authorization page

Note

Read the Docs needs elevated permissions to perform certain operations that ensure that the workflow is as smooth as possible, like installing [webhooks](#). If you want to learn more, check out [Permissions for connected accounts](#).

After that, you will be redirected to Read the Docs to confirm your e-mail and username. Click the [Sign Up »](#) button to create your account and open your [dashboard](#).

When you have clicked the link in your email from Read the Docs to “verify your email address” and finalize the process, your Read the Docs account will be ready to create your first project.

 **Read the Docs**

astrojl-test

Your primary email address is not verified. Please [verify it here](#).

Confirmation e-mail sent to juanluis@readthedocs.org.

Successfully signed in as [astrojl-test](#).

Ready to share your documentation?

You don't have any projects yet, but you can start building documentation by importing one. Not sure how to start documenting your project? Check out the [Getting Started Guide](#) to learn how.

Want to try a demo? [Import our own demo project](#).

Import a Project

Search all docs


Thanks!


Your [support](#) of Read the Docs helps make the site better each and every month.

Welcome to your Read the Docs dashboard!

Importing the project to Read the Docs

To import your GitHub project to Read the Docs:

1. Click the **Import a Project** button on your [dashboard](#).
2. Click the **+** button to the right of your `rtd-tutorial` project. If the list of repositories is empty, click the  button.

 **Read the Docs**


astrojl-test

Import a Repository

We're only showing public repositories. For private projects and other features, please use [Read the Docs for Business](#).


No remote repositories found, try [refreshing your accounts](#).

« previous next »

 You can import your project manually if it isn't listed here or connected to one of your accounts.

Import Manually

Filter repositories

 astrojl-test

Import projects workflow

3. Enter some details about your Read the Docs project:

Name

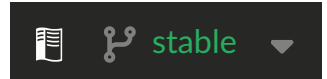
The name of the project, used to create a unique subdomain for each project. so it is better if you prepend your username, for example `{username}-rtd-tutorial`.

Repository URL

The URL that contains the documentation source. Leave the automatically filled value.

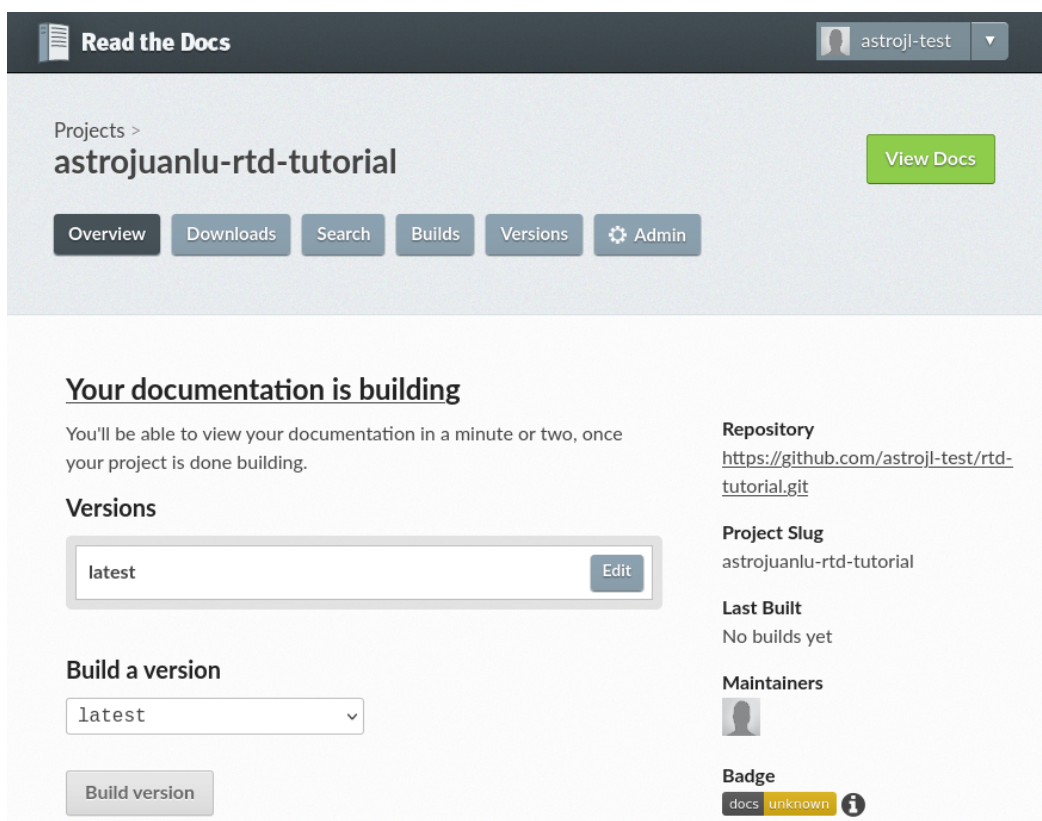
Default branch

Name of the default branch of the project, leave it as `main`.



Then click the [Next](#) button to create the project and open the [project home](#).

You just created your first project on Read the Docs! 🎉



Project home

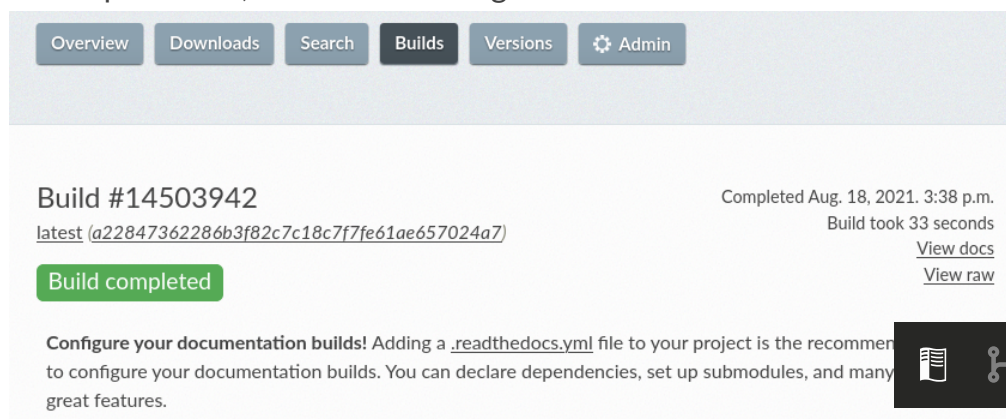
Checking the first build

Read the Docs will build your project documentation right after you create it.

To see the build logs:

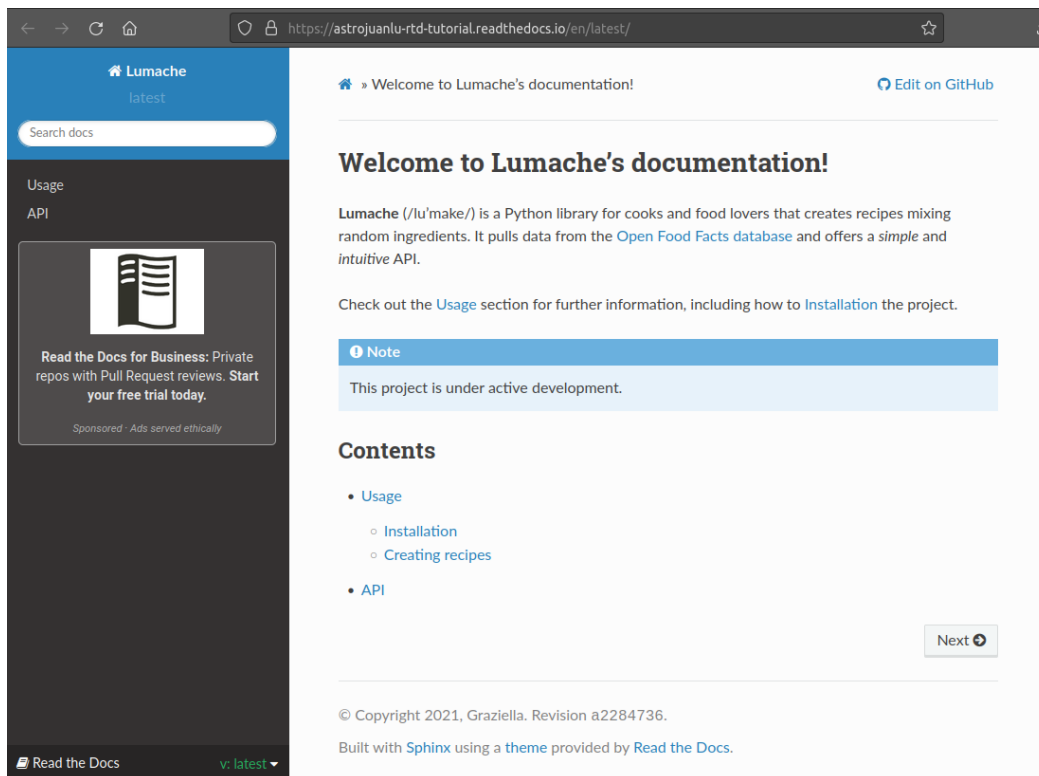
1. Click the [Your documentation is building](#) link on the [project home](#).

- If the build has not finished by the time you open it, you will see a spinner next to a “Installing” or “Building” indicator, meaning that it is still in progress.
- If the build has finished, you’ll see a green “Build completed” indicator, the completion date, the elapsed time, and a link to the generated documentation.



First successful documentation build

2. Click on [View docs](#) to see your documentation live!



HTML documentation live on Read the Docs

Note

Advertisement is one of our main sources of revenue. If you want to learn more about how do we fund our operations and explore options to go ad-free, check out our [Sustainability page](#).

If you don't see the ad, you might be using an ad blocker. Our EthicalAds network respects your privacy, doesn't target you, and tries to be as unobtrusive as possible, so we would like to kindly ask you to [not block us](#) ❤️

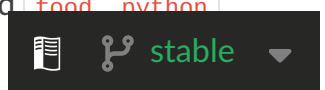
Configuring the project

To update the project description and configure the notification settings:

1. Navigate back to the [project page](#) and click the [Admin](#) button, to open the Settings page.
2. Update the project description by adding the following text:

Lumache (/lu'make/) is a Python library for cooks and food lovers that creates recipes mixing random ingredients.


3. Set the project homepage to <https://world.openfoodfacts.org/>, and add `food python` to the list of public project tags.
4. To get a notification if the build fails, click the [Email Notifications](#) link on the left, add your email address, and click the [Add](#) button.

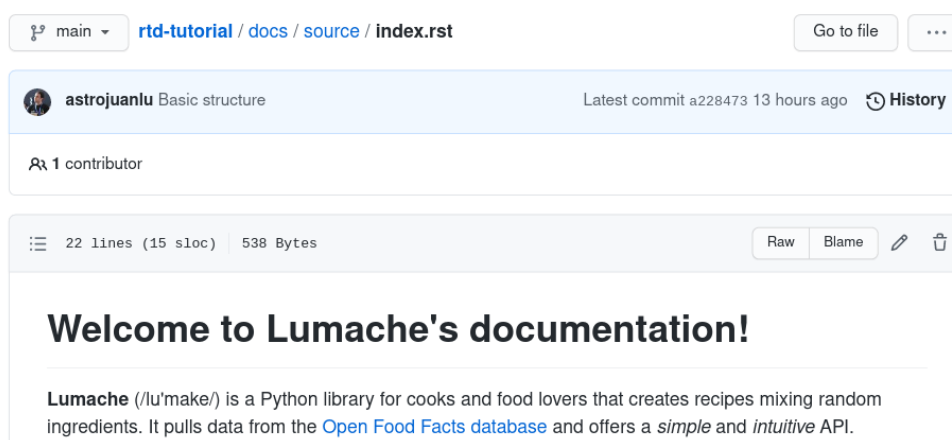


Triggering builds from pull requests

Read the Docs can [trigger builds from GitHub pull requests](#) and show you a preview of the documentation with those changes.

To trigger builds from pull requests:

1. Click the **Settings** link on the left under the **Admin** menu, check the “Build pull requests for this project” checkbox, and click the **Save** button at the bottom of the page.
2. Make some changes to your documentation:
 1. Navigate to your GitHub repository, locating the file `docs/source/index.rst`, and clicking on the  icon on the top-right with the tooltip “Edit this file” to open a web editor (more information [on their documentation](#)).



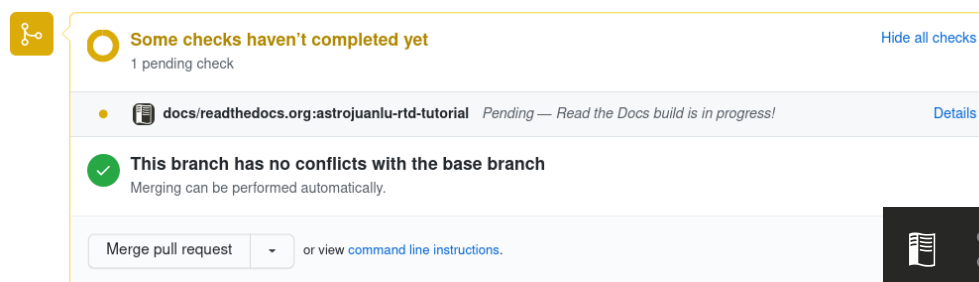
File view on GitHub before launching the editor

2. In the editor, add the following sentence to the file:

`docs/source/index.rst`

```
Lumache hosts its documentation on Read the Docs.
```

3. Write an appropriate commit message, choose the “Create a **new branch** for this commit and start a pull request” option.
4. Click the green **Propose changes** button to open the new pull request page, then click the **Create pull request** button below the description.



Read the Docs building the pull request from GitHub

After opening the pull request, a Read the Docs check will appear indicating that it is building the documentation for that pull request. If you click the [Details](#) link while your project is building the build log will be opened. After building this link opens the documentation directly.

Adding a configuration file

The Admin tab of the [project home](#) has some *global* configuration settings for your project.

Build process configuration settings are in `.readthedocs.yaml` [configuration file](#), in your Git repository, which means it can be different for every version or branch of your project (more on [versioning](#)).

Using different Python versions

To build your project with Python 3.8 instead of the latest Python version, edit the `.readthedocs.yaml` file and change the Python version to 3.8 like this:

`.readthedocs.yaml`

```
version: 2

build:
  os: "ubuntu-22.04"
  tools:
    python: "3.8"

python:
  install:
    - requirements: docs/requirements.txt

sphinx:
  configuration: docs/source/conf.py
```

The [purpose of each key](#) in the `.readthedocs.yaml` configuration file is:

`version`

Required, specifies [version 2 of the configuration file](#).

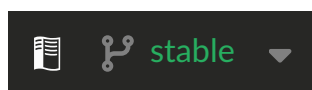
`build.os`

Required, specifies the Docker image used to build the documentation. [states the name of the base image](#).

`build.tools.python`

Specifies the [Python version](#).

`python.install.requirements`



Specifies what [Python dependencies](#) to install.

After you commit these changes, go back to your project home, navigate to the “Builds” page, and open the new build that just started. You will notice that one of the lines contains `python -mvirtualenv`: if you click on it, you will see the full output of the corresponding command, stating that it used Python 3.8.6, the latest version of Python 3.8, to create the virtual environment.

```
python3.8 -mvirtualenv /home/docs/checkouts/readthedocs.org/user_builds/astrojuanlu-rtd-tu
Using base prefix '/home/docs/.pyenv/versions/3.8.6'
New python executable in /home/docs/checkouts/readthedocs.org/user_builds/astrojuanlu-rtd-tu
Also creating executable in /home/docs/checkouts/readthedocs.org/user_builds/astrojuanlu-
Installing setuptools, pip, wheel...
done.
Command time: 3s Return: 0
```

Read the Docs build using Python 3.8

Making build warnings more visible

If you navigate to your HTML documentation, you will notice that the index page looks correct but the API section is empty. This is a very common issue with Sphinx, and the reason is stated in the build logs. On the build page you opened before, click on the [View raw](#) link on the top right, which opens the build logs in plain text, and you will see several warnings:

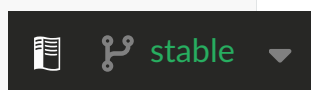
```
WARNING: [autosummary] failed to import 'lumache': no module named lumache
...
WARNING: autodoc: failed to import function 'get_random_ingredients' from module
'lumache'; the following exception was raised:
No module named 'lumache'
WARNING: autodoc: failed to import exception 'InvalidKindError' from module 'lumache';
the following exception was raised:
No module named 'lumache'
```

To spot these warnings more easily and help you to address them, add the

`sphinx.fail_on_warning` option to your Read the Docs configuration file.

To fail on warnings to your Read the Docs project, edit the `.readthedocs.yaml` file in your project, add the three lines of `sphinx` configuration below, and commit the file:

`.readthedocs.yaml`



```
version: 2

build:
  os: "ubuntu-22.04"
  tools:
    python: "3.8"

python:
  install:
    - requirements: docs/requirements.txt

sphinx:
  configuration: docs/source/conf.py
  fail_on_warning: true
```

If you navigate to your “Builds” page, you will see a `Failed` build, which is expected because we’ve configured Sphinx to fail on warnings and several warnings were encountered during the build.

To learn how to fix these warnings, see the next section.

Installing Python dependencies

The reason `sphinx.ext.autosummary` and `sphinx.ext.autodoc` fail to import the [Making build warnings more visible](#), is because the `lumache` module is not installed.

You will need to specify those installation requirements in `.readthedocs.yaml`.

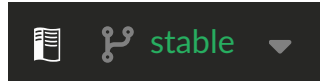
To install your project dependencies and make your code available to Sphinx, edit `.readthedocs.yaml`, add the `python.install` section and commit it:

`.readthedocs.yaml`

```
python:
  install:
    - requirements: docs/requirements.txt
    # Install our python package before building the docs
    - method: pip
      path: .
```

Now, Read the Docs installs the Python code before starting the Sphinx build, which will finish seamlessly. If you go now to the API page of your HTML documentation, you will see the `lumache` summary! :tada:

Enabling PDF and EPUB builds



Sphinx can build several other formats in addition to HTML, such as PDF and EPUB. You

might want to enable these formats for your project so your users can read the documentation offline.

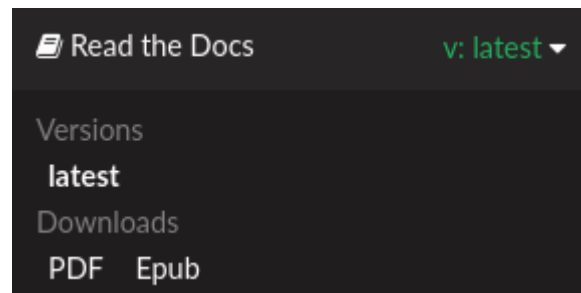
To do so, add the following `formats` to your `.readthedocs.yaml`:

`.readthedocs.yaml`

```
sphinx:
  configuration: docs/source/conf.py
  fail_on_warning: true

formats:
  - pdf
  - epub
```

After this change, PDF and EPUB downloads will be available both from the “Downloads” section of the [project home](#), as well as the [flyout menu](#).



Downloads available from the flyout menu

Versioning documentation

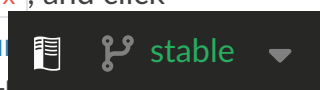
Read the Docs supports having [several versions of your documentation](#), in the same way that you have several versions of your code. By default, it creates a `latest` version that points to the default branch of your version control system (`main` in the case of this tutorial), and that’s why the URLs of your HTML documentation contain the string `/latest/`.

Creating a new version of your documentation

Read the Docs automatically creates documentation versions from GitHub branches and tags that [follows some rules](#) about looking like version numbers, such as `1.0`, `2.0.3` or `4.x`.

To create version `1.0` of your code, and consequently of your documentation:

1. Navigate to your GitHub repository, click the branch selector, type `1.0.x`, and click “Create branch: 1.0.x from ‘main’” (more information [in the GitHub docs](#)).
2. Check that you now have version `1.0.x` in your [project home](#), click on the [versions](#) button, and under “Active Versions” you will see two entries:



- The `latest` version, pointing to the `main` branch.
- A new `stable` version, pointing to the `origin/1.0.x` branch.

Active Versions

latest	main	Public	Edit
stable	origin/1.0.x	Public	Edit

List of active versions of the project

When you created your branch, Read the Docs created a new special version called `stable` pointing to it. When it's built it will be listed in the [flyout menu](#).

Setting stable as the default

To set `stable` as the *default version*, rather than `latest`, so that users see the `stable` documentation when they visit the [root URL](#) of your documentation:

1. In the `Admin` menu of your project home, go to the `Settings` link, choose `stable` in the “Default version*” dropdown, and hit `Save` at the bottom.

Modifying versions

Both `latest` and `stable` are now *active*, which means that they are visible for users, and new builds can be triggered for them. In addition to these, Read the Docs also created an *inactive* `1.0.x` version, which will always point to the `1.0.x` branch of your repository.

Activate a version

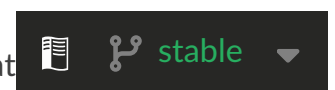
Active versions are built whenever new code is pushed to that branch or tag.

1.0.x	Public	Activate
main	Public	Activate

List of inactive versions of the project

To activate the `1.0.x` version:

1. On your [project home](#), go to the “Versions”, locate `1.0.x` under “Activate” and click the `Activate` button.
2. On the “Activate” page with “Active” and “Hidden” checkboxes, check only “Active” and



Note

Read more about [hidden versions](#) in our documentation.

Getting project insights

Once your project is up and running, you will probably want to understand how readers are using your documentation, addressing some common questions like:

- what are the most visited pages?
- what are the most frequently used search terms?
- are readers finding what they are looking for?

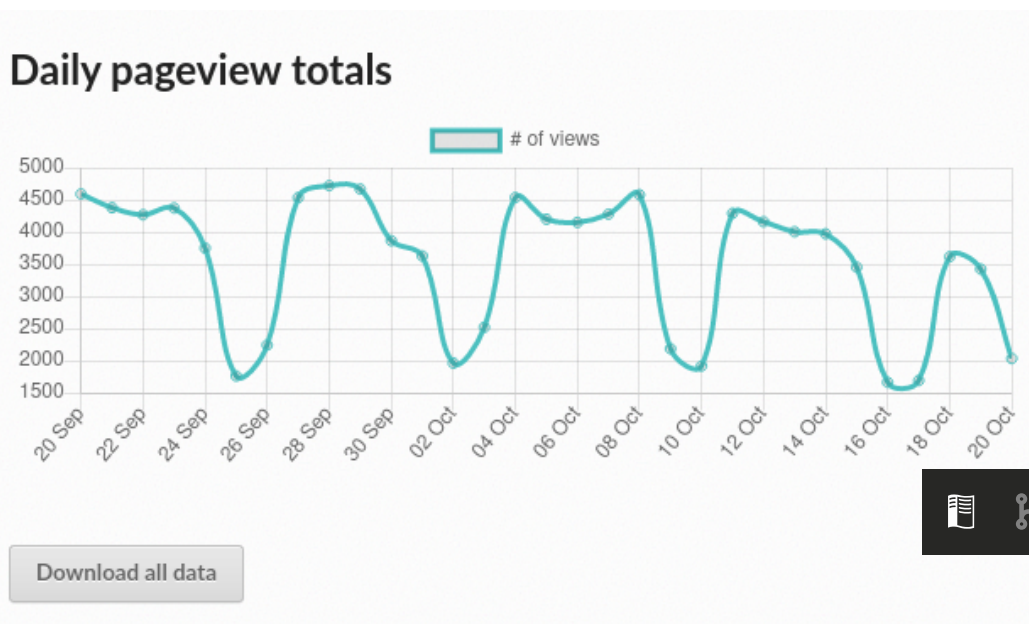
Read the Docs has traffic and search analytics tools to help you find answers to these questions.

Understanding traffic analytics

The Traffic Analytics view gives you a simple overview of how your readers browse your documentation. It respects visitor privacy by not storing identifying information about your them.

This page shows the most viewed documentation pages of the past 30 days, plus a visualization of the daily views during that period.

To see the Traffic Analytics view, go back the [project page](#) again, click the [Admin](#) button, and then click the [Traffic Analytics](#) section. You will see the list of pages in descending order of visits, and a similar visualization to this one:



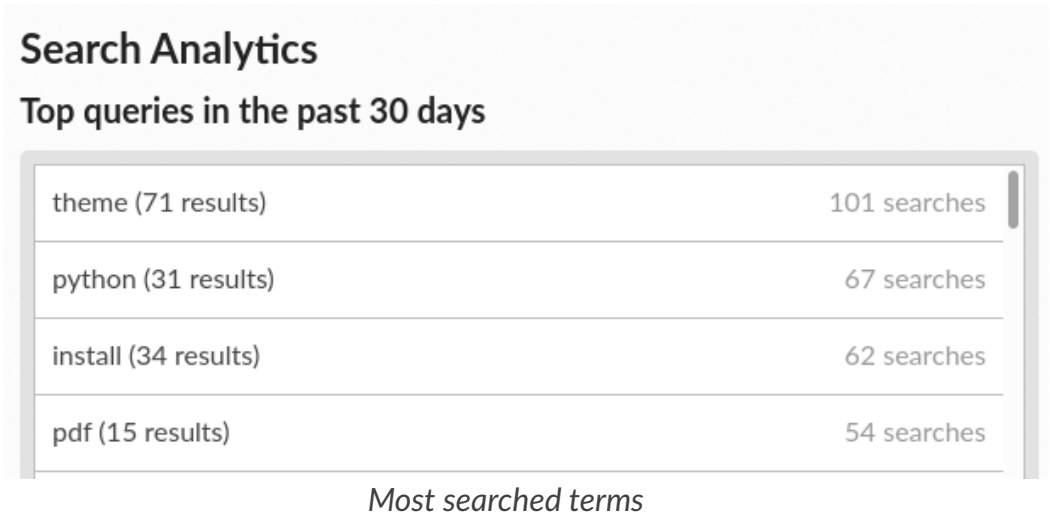
You can also download this data in [CSV](#) format for closer inspection. To do that, scroll to the bottom of the page and click the [Download all data](#) button.

Understanding search analytics

As well as traffic analytics, Read the Docs shows [what terms your readers are searching for](#). This can inform decisions on what areas to focus on, or what parts of your project are less understood or more difficult to find.

To generate some artificial search statistics on the project, go to the HTML documentation, locate the Sphinx search box on the left, type `ingredients`, and press the `Enter` key. You will be redirected to the search results page, which will show two entries.

Next, go back to the [Admin](#) section of your project page, and then click the [Search Analytics](#) section. You will see a table with the most searched queries (including the `ingredients` one you just typed), how many results did each query return, and how many times it was searched. Below the queries table, you will also see a visualization of the daily number of search queries during the past 30 days.

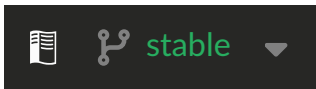


Like the Traffic Analytics, you can also download the whole dataset in CSV format by clicking on the [Download all data](#) button.

Where to go from here

This is the end of the tutorial. You have accomplished a lot:

1. Forked a GitHub repository.
2. Connected it to Read the Docs.
3. Built its HTML documentation.
4. Customized the build process.



5. Added new documentation versions.
6. Browsed the project analytics.

Nice work!

Here are some resources to help you continue learning about documentation and Read the Docs:

- Learn more about the platform [features](#).
- Learn about other supported documentation generators in the [Sphinx tutorial](#) or the [MkDocs User Guide](#).
- See a list of Read the Docs [Example projects](#).
- Learn how to do specific tasks in the [How-to guides A-Z](#).
- Learn about private project support and other enterprise features in [our commercial service guide](#).
- Join a global community of fellow `documentarians` in [Write the Docs](#) and [its Slack workspace](#).
- Contribute to Read the Docs in [Contributing to Read the Docs](#), we appreciate it!

Happy documenting!