



The Kartoza Handbook



Your Open Source Geospatial Experts.
Kartoza Pty (Ltd.)
2022



Contents

Home	3
Template project title	3
Contributor covenant code of conduct	5
Examples of running instances	6
User	7
User documentation	7
Quickstart	8
Guide	9
Manual	10
Administrators	11
For administrators	11
Guide	12
Manual	13
Developers	14
For Developers	14
Guide	15
Python Reference Manual	42
Documentation Writing	43
API	62
DevOps	65
DevOps documentation	65
DevOps guide	66
DevOps documentation	67
About	68
About	68
Contributing to [PROJECT_NAME]	77
Disclaimer	80



1 Home

1.1 Template project title

Strapline

main_screenshot

1.1.1 Introduction

1.1.2 Key concepts

A **project** (dashboard) is the most important feature of PROJECT_NAME. Projects combine different elements (Reference datasets, indicators, and context layers) and enable data visualization/analysis for all end users.

Context layers are geospatial layers used to contextualise the information presented in a project.

Indicators layers derive from spreadsheets or other intelligence assets harvested by the platform.

Widgets are visual components such as charts generated by performing data analysis on country/regional levels and the indicator data.

1.1.2.1 Disclaimer

The software provided by this project is provided 'as is'. All information provided within the platform should be independently verified before using as the basis for action. The contributors and developers of this platform take no responsibility for any loss of revenue, life, physical harm or any other adverse outcome that may occur as a result of the use of this platform.

1.1.2.2 Purpose

1.1.2.3 Scope of project

1.1.2.4 Project roadmap

Project Roadmap

[Project Roadmap](#)

Contributing

We would love to collaborate with you! But first, please read our [contributor guidelines](#) which describe how to report issues, plan and contribute feature additions etc.

Code of conduct

Our community in this project is aligned with our [Code of Conduct](#) - please be sure to read and abide by that document in all interactions with our community.

Diversity statement

This project welcomes and encourages participation by everyone.

No matter how you identify yourself or how others perceive you: we welcome you. We welcome contributions from everyone as long as they interact constructively with our community.

While much of the work for our project is technical in nature, we value and encourage contributions from those with expertise in other areas, and welcome them into our community.

1.1.3 Project partners



Project Partner 1

[Project Partner 1](#)

Releases

[GitHub releases page](#) [Releases page](#)

Project Badges

license repo not found

release no releases or repo not found

github repo or version not found

issues repo not found

issues repo not found

issues repo not found

Project chatroom

[Chatroom](#)

Contributor License Agreement (CLA)

Contributions to this project will be subject to our [Contributor License Agreement](#)
License

This project is open source, published under the AGPL-3. You can read our license to find out what rights this license bestows to users and contributors.

[License](#)





1.1 Contributor covenant code of conduct

1.1.1 Our pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

1.1.2 Our standards

Examples of behaviour that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behaviour by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

1.1.3 Our responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behaviour and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behaviour.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

1.1.4 Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

1.1.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behaviour may be reported by contacting the project team. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

1.1.6 Attribution

This Code of Conduct is adapted from the [Contributor Covenant](https://www.contributor-covenant.org/version/1/4/code-of-conduct.html), version 1.4, available at <https://www.contributor-covenant.org/version/1/4/code-of-conduct.html>



1.1 Examples of running instances



1 User

1.1 User documentation

This is the homepage for all user related documentation.

The user content is divided into three sections:

- The [quickstart tutorial](#), which aims to get you familiar with the basics of platform in around 5 minutes.
- The [user guide](#), which describes common workflows in a tutorial format.
- The [user manual](#), which describes each page of the user interface and what the various options on that page do.



1.1 Quickstart

1.1.1 Installing the product

1.1.2 Getting started

1.1.2.1 Releases

[GitHub releases page](#) [Releases page](#)



1.1 Guide

1.1.1 User guide



1.1 Manual

1.1.1 User manual

This section of the documentation describes every page in the application and what the various components of that page do. The manual is intended to function as a reference for the application. For narrative / workflow based tutorials, you may prefer to work through our [user guide](#).



1 Administrators

1.1 For administrators

This is the homepage for all administrator related documentation.

The content is divided into two sections:

- The [user guide](#), which describes common workflows for system administrators in a tutorial format.
- The [user manual](#), which describes each page of the admin user interface and what the various options on that page do.



1.1 Guide

1.1.1 Administrator guide



1.1 Manual

1.1.1 Administrator manual



1 Developers

1.1 For Developers

This is the homepage for all developer related documentation.

- The [developer guide](#), which describes common workflows for developers in a tutorial format. This includes instructions on how to set up your developer environment, check out the code, run it locally etc.
- The [developer manual](#), which describes each python module and its classes, functions etc. This section is autogenerated.
- The [api guide](#), which describes common workflows for using the restful API.
- The [api manual](#), which describes each API endpoint and its parameters, return values etc. This section is autogenerated.



1.1 Guide

1.1.1 Developer guide

In this section of the documentation, we aim to onboard developers onto the process of setting up a developer workstation, understanding the system architecture, contributing back to the project and so forth. Here is a brief overview of the content provided here:

- **Architecture:** - Here we outline the system architecture using ER Diagrams, Software Component Diagrams etc.
- **Prerequisites:** - An overview of the software that needs to be installed before you can start.
- **Cloning:** - In this section we cover how to make a local working copy on your own machine.
- **IDE Setup:** - Here we walk through setting up your IDE.
- **Building:** - This covers tasks related to building the application so you can test it in your browser.
- **Configuration:** - This covers standard configuration tasks that need to be done once the site is brought online.
- **Design:** - This section describes the visual design approach for the user interface.
- **Workflows:** - Examples of typical workflows for adding features, fixing bugs etc.
- **Roadmap:** - Here we outline future plans for the project.



1.1.1 System architecture

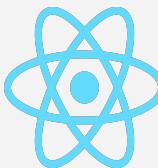
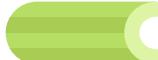
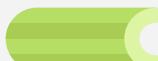
In this section, we outline the system architecture using ER Diagrams, Software Component Diagrams etc. and key libraries / frameworks used in this project.

1.1.1.1 Software components used



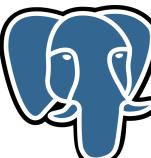
The following is a list, with brief descriptions, of the key components used in creating this platform. Please refer to their individual documentation for in-depth technical information.



Logo	Name	Notes
	Django	Django makes it easier to build better web apps more quickly and with less code.
	ReactJS	React lets you build user interfaces out of individual pieces called components. Create your own React components like Thumbnail , LikeButton , and Video . Then combine them into entire screens, pages, and apps.
	MUI	Move faster with intuitive React UI tools. MUI offers a comprehensive suite of free UI tools to help you ship new features faster. Start with Material UI, our fully-loaded component library, or bring your own design system to our production-ready components.
	Docker	Accelerate how you build, share, and run applications. Docker helps developers build, share, and run applications anywhere — without tedious environment configuration or management.
	Celery	Celery is a simple, flexible, and reliable distributed system to process vast amounts of messages, while providing operations with the tools required to maintain such a system. It's a task queue with focus on real-time processing, while also supporting task scheduling.
	Celery Beat	This extension enables you to store the periodic task schedule in your database. The periodic tasks can be managed from the Django Admin interface, where you can create, edit and delete periodic tasks and how often they should run.

Django REST framework is a powerful and flexible toolkit for building Web APIs.



Logo	Name	Notes
	Django Rest Framework	
	MapLibre	Open-source mapping libraries for web and mobile app developers.
		deck.gl is a WebGL-powered framework for visual exploratory data analysis of large datasets.
	PostGIS	PostGIS extends the capabilities of the PostgreSQL relational database by adding support storing, indexing and querying geographic data.
	PostgreSQL	PostgreSQL is a powerful, open source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.
	Tegola	An open source vector tile server written in Go, Tegola takes geospatial data and slices it into vector tiles that can be efficiently delivered to any client.



Logo	Name	Notes
	GDAL	GDAL is a translator library for raster and vector geospatial data formats that is released under an MIT style Open Source License by the Open Source Geospatial Foundation. As a library, it presents a single raster abstract data model and single vector abstract data model to the calling application for all supported formats. It also comes with a variety of useful command line utilities for data translation and processing.

1.1.1.2 Docker components

The following diagram represents the docker containers, ports and volumes that are used to compose this platform.

Architecture Docker Diagram

1.1.1.3 ER diagram

The following diagram represents all of the database entities that are created by the Django ORM (Object Relational Mapper). Right click the image and open it in its own tab to see it at full resolution.

Architecture ERD

 If you already have all of the above criteria met, you can move on to [Prerequisites](#) to start the process of getting your local development environment set up.



1.1.1 Project prerequisites



1.1.1 Checking out the code

This section outlines the process of checking out the code for local development.

🚩 Make sure you have gone through the [Prerequisites Section](#) before following these notes.

Git Code check out [PROJECT_URL]

Code:

```
git clone https://github.com/project-name/repository.git
```

📒 **Which branch to use?**: Note that we deploy our staging work from the `develop` branch and our production environment from the `main` branch. If you are planning on contributing improvements to the project, please submit them against the `develop` branch.

📝 Now that you have the code checked out, move on to the [IDE Setup](#) documentation.



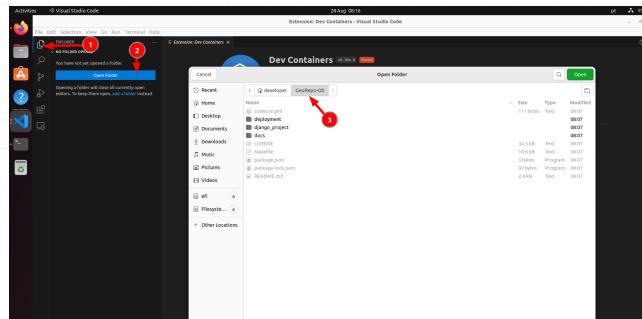
1.1.1 Setting up your IDE

This section outlines the process for configuring your IDE for development.

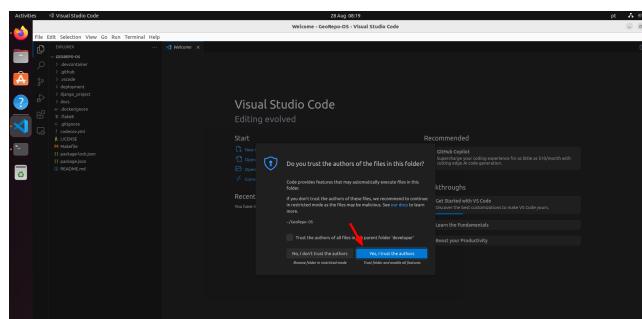
🚩 Make sure you have gone through the [Cloning Section](#) before following these notes.

1.1.1.1 VS Code setup

Open the project in VSCode (1, 2) by navigating to the place on your file system where you checked out the code in the pre-requisites step above (3).

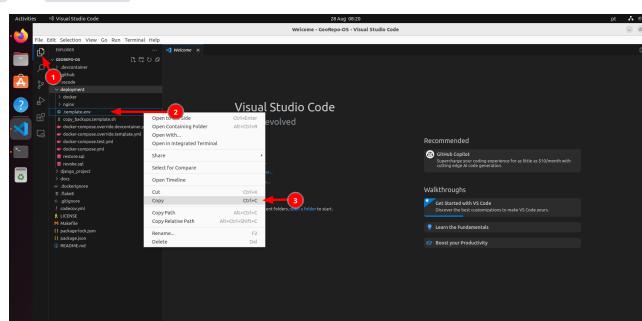


Accept the 'trust authors' prompt



Copying the .env

Copy the `template.env` to `.env`

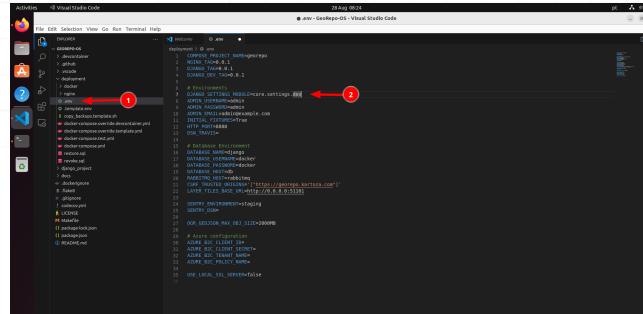
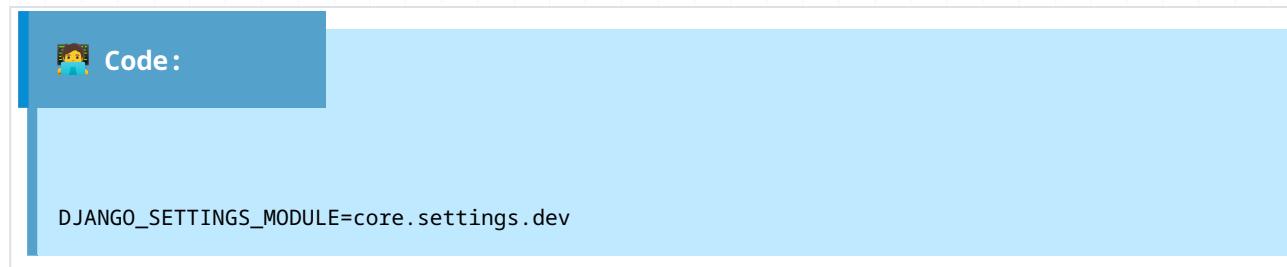


Edit the `.env` file and change the

Code:

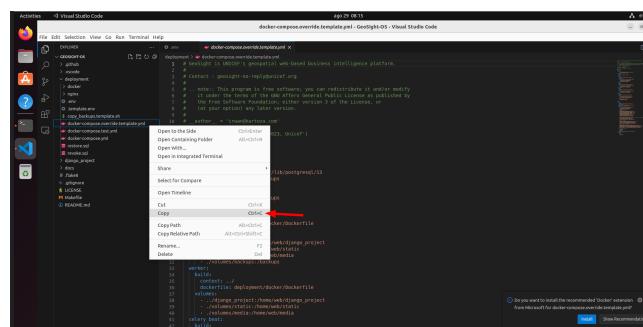
```
DJANGO_SETTINGS_MODULE=core.settings.prod
```

to

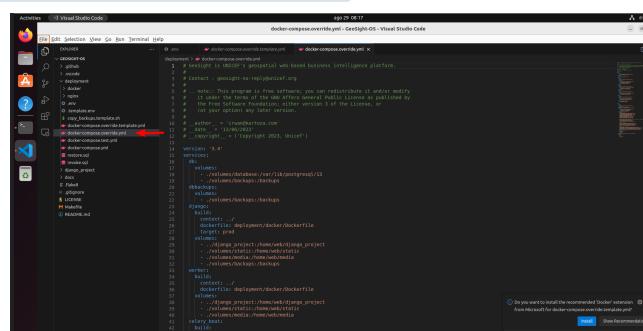


Override Docker configurations

We are going to copy the docker overrides template to a local file that will not be under version control.



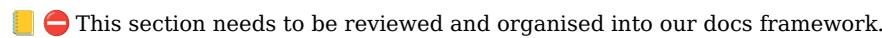
Rename the file to `docker-compose.override.yml`



Initially you will not need to change anything in this file, though you may want to take a look through the various configurations provided here if you want to tweak your local setup.

Now that you have your IDE set up, we can move on to [building the project](#).

1.1.1.2 Using PyCharm



This section is for using pycharm.

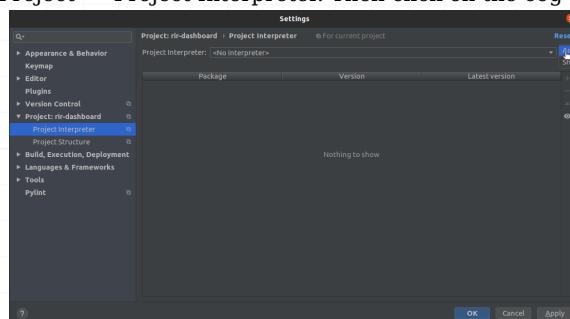
Requirements:

- Pycharm
 - Finished **Setting up the project**

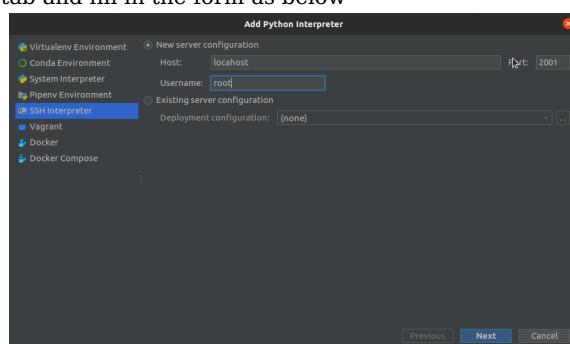


1.1.1.3 Setup interpreter

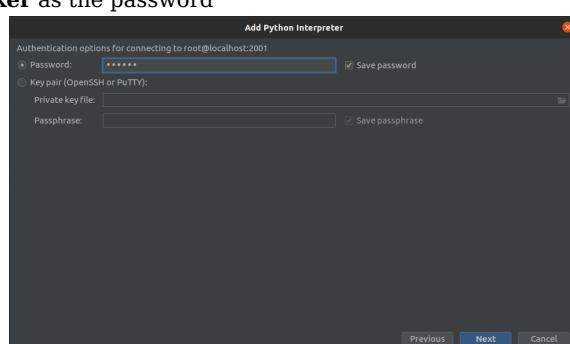
1. Go to File -> Settings -> Project -> Project Interpreter. Then click on the cog -> add



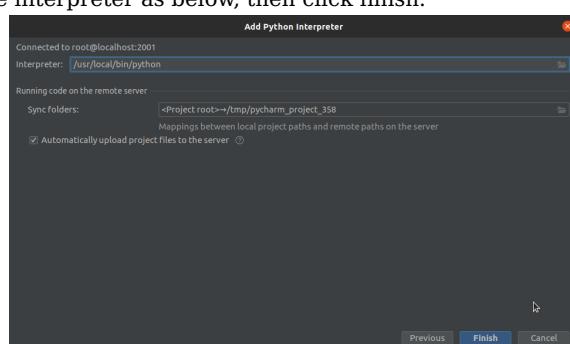
2. Go to the ssh interpreter tab and fill in the form as below



3. Click next and fill in **docker** as the password



4. Click next and change the interpreter as below, then click finish.



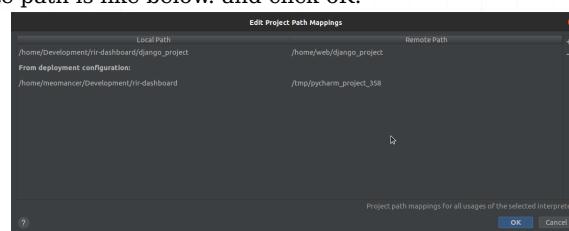


5. After you click finish, it will show all of the packages like below.

A screenshot of the PyCharm Project Interpreter settings window. The sidebar shows categories like Appearance & Behavior, Editor, Plugins, Version Control, Project, Build, Execution, Deployment, Languages & Frameworks, Tools, and Pylint. The 'Project' section is expanded, showing 'rir-dashboard' selected. The main area displays a table of packages with columns for Package, Version, and Latest version. The table includes packages such as Babel, Django, Faker, Jinja2, MarkupSafe, Pillow, Pygments, SecretStorage, Sphinx, alembic, amqp, asynctest, astroid, billiard, and cached-property.

Package	Version	Latest version
Babel	2.9.1	2.9.1
Django	3.2.8	4.0.3
Faker	13.3.2	13.3.4
Jinja2	3.0.3	3.1.1
MarkupSafe	2.0.1	2.1.1
Pillow	8.3.1	9.0.1
Pygments	2.11.2	2.11.2
SecretStorage	2.3.1	3.3.1
Sphinx	4.3.2	4.5.0
alembic	0.7.12	0.7.12
amqp	5.1.0	5.1.0
asynctest	3.4.1	3.5.0
asn1crypto	0.24.0	1.5.1
astroid	2.9.3	2.11.2
billiard	3.6.4.0	3.6.4.0
cached-property	1.5.2	1.5.2

6. In current page, click **path mappings**, click + button and put local path to where the project (django-project folder) and remote path is like below. and click OK.



Now the interpreter is done. When we restart the machine, we need to do **make up** to run the project.

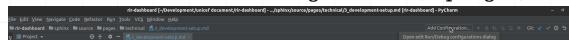
1.1.1.4 Setup run configuration



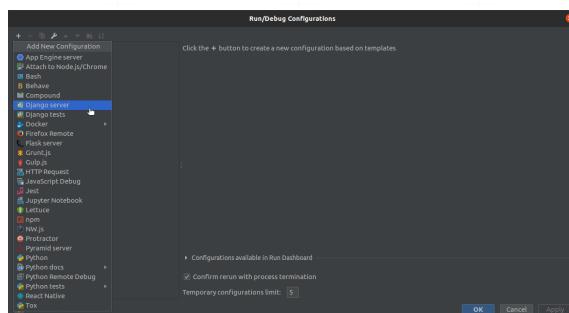
After the interpreter is done, we need configuration to run the project in development mode.



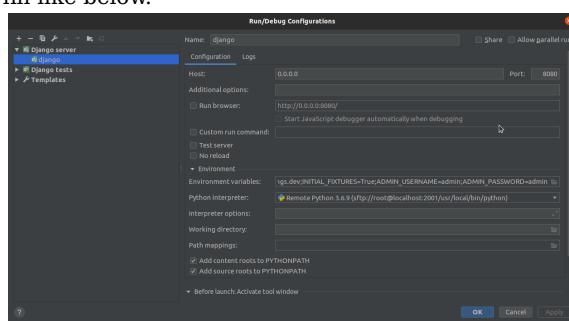
1. Click "Add configuration" like in the cursor in the image below. (top-right)



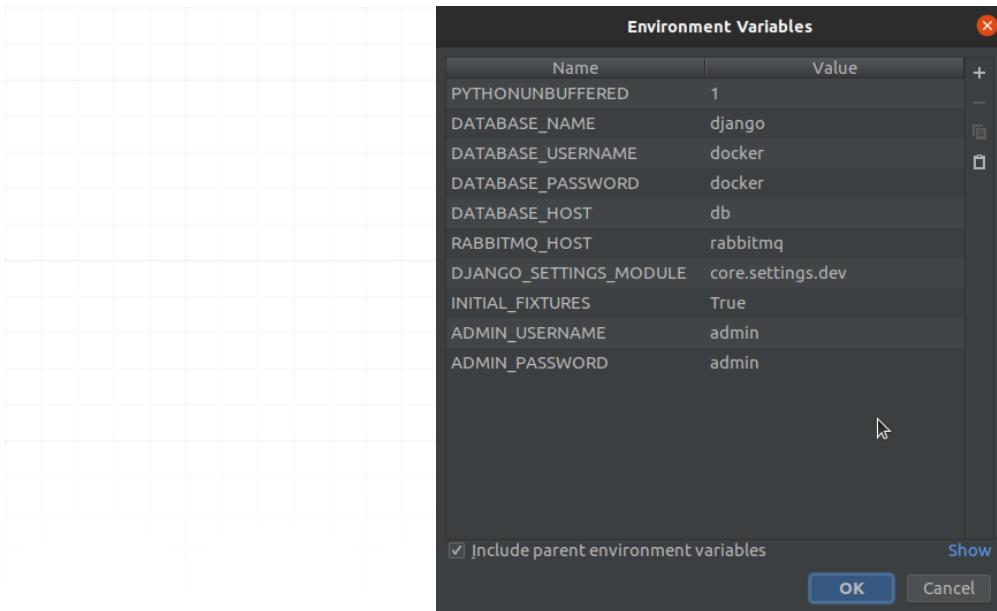
2. There will be a popup, and click +, then click **django server** like below



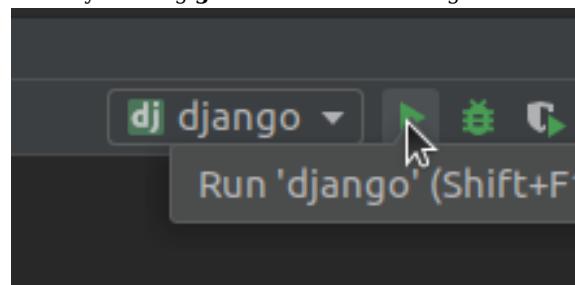
3. It will show the form and fill like below.



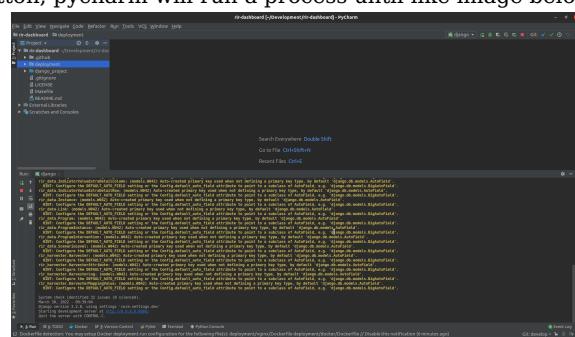
4. Don't click the OK yet, but click **Environment Variables** and add environments like below (by clicking + button).



5. After that, click OK.
6. Now we need to run the server by clicking **go** button in below image.



7. When we click the **go** button, pycharm will run a process until like image below.



8. Now it is done. We can access the development server in <http://localhost:2000/>

This development mode is DEBUG mode, and also whenever we change the code, the site will also change in runtime.

For more information how to set up on pycharm, please visit [Using a Docker Compose-Based Python Interpreter in PyCharm](#)

1.1.1.5 Quick setup guide

This content needs to be reviewed and moved to the readme.



Production

 **Code:**

```
git clone https://github.com/project-name/repository
cd project-name/deployment
docker-compose up -d
```

The web will be available at <http://127.0.0.1/>

To stop containers:

 **Code:**

```
docker-compose kill
```

To stop and delete containers:

 **Code:**

```
docker-compose down
```

Development

 **Code:**

```
git clone https://github.com/project-name/repository
cd project-name/deployment
cp .template.env .env
cp docker-compose.override.template.yml docker-compose.override.yml
```



After that,

- open new terminal
- on folder root of project, do

 **Code:**

```
make frontend-dev
```

Wait until it is done when there is sentence "webpack xxx compiled successfully in xxx ms".

After that, don't close the terminal. If it is accidentally closed, do `make frontend-dev` again

Next step: - Open new terminal - Do commands below

 **Code:**

```
make up  
make dev
```

Wait until it is on.

The web can be accessed using <http://localhost:2000/>

If the web is taking long time to load, restart project-name_dev container by `make dev-reload`.

The sequence should be `make frontend-dev`, after that run or restart project-name_dev.

To stop dev:

 **Code:**

```
make dev-kill
```

To reload container:

 **Code:**

```
make dev-reload
```



1.1.1 Project setup

1.1.1.1 Clone [PROJECT_NAME] repository

This will clone the [PROJECT_NAME] repository to your machine



Code:

```
git clone https://github.com/project/repository.git
```

1.1.1.2 Set up the project

This will set up the [PROJECT_NAME] project on your machine



Code:

```
cd [PROJECT_NAME]
cd deployment
cp docker-compose.override.template.yml docker-compose.override.yml
cp .template.env .env
cd ..
make up
```

Wait until everything is done.

After everything is done, open up a web browser and go to <http://127.0.0.1/> and the dashboard will open:

By Default, we can use the admin credential:



Code:

```
username : admin
password : admin
```

1.1.1.3 Set up different environment



To set up different environment, for example the Default credential, or the port of server, open **deployment/.env**. You can check the description below for each of variable.

 **Code:**

```
COMPOSE_PROJECT_NAME=[PROJECT_NAME]
NGINX_TAG=0.0.1 -> Change this for different nginx image
DJANGO_TAG=0.0.1 -> Change this for different django image
DJANGO_DEV_TAG=0.0.1 -> Change this for different django dev image

# Environments
DJANGO_SETTINGS_MODULE=core.settings.prod -> Change this to use different django config file
ADMIN_USERNAME=admin -> Default admin username
ADMIN_PASSWORD=admin -> Default admin password
ADMIN_EMAIL=admin@example.com -> Default admin email
INITIAL_FIXTURES=True
HTTP_PORT=80 -> Change the port of nginx

# Database Environment
DATABASE_NAME=django -> Default database name
DATABASE_USERNAME=docker -> Default database username
DATABASE_PASSWORD=docker -> Default database password
DATABASE_HOST=db -> Default database host. Change this if you use cloud database or any new docker
RABBITMQ_HOST=rabbitmq

# Onedrive
PUID=1000
PGID=1000
```

After you change the desired variable and do **make up**. It will rerun the project with new environment.

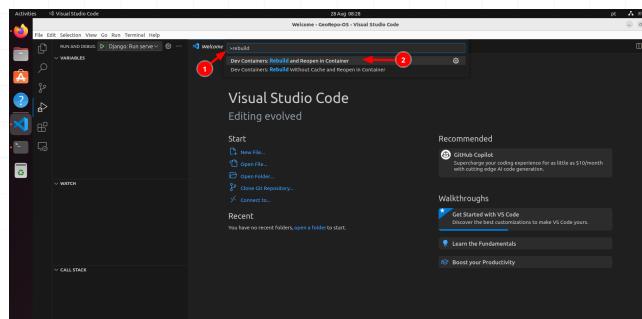


1.1.1 Building the dev environment

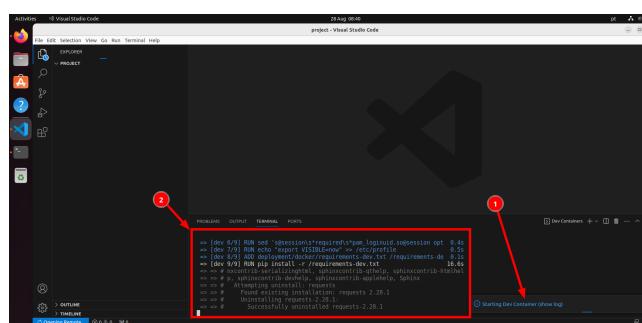
This section covers the process of building and running the application from your IDE.

👉 Make sure you have gone through the [IDE Setup Process](#) before following these notes.

Press **Ctrl -> P** **1** and then **>** and search for **Rebuild**. Select **Dev Containers: Rebuild and Reopen in Container** **2**. This will essentially mount your code tree inside a docker container and switch the development context of VSCode to be inside the container where all of the python etc. dependencies will be installed.

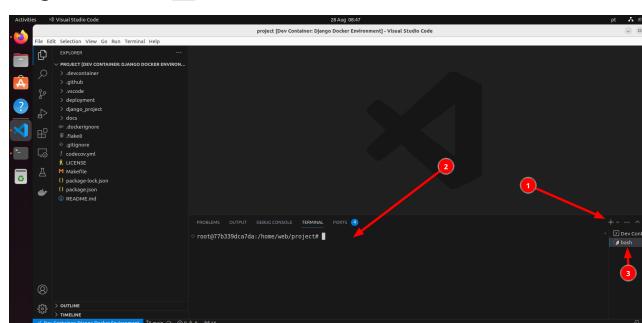


Once the task is running, a notification **1** will be shown in the bottom right of the VSCode window. Clicking in the notification will show you the setup progress **2**. Note that this make take quite a while depending on the internet bandwidth you have and the CPU power of your machine.



1.1.1.1 Open a dev container terminal

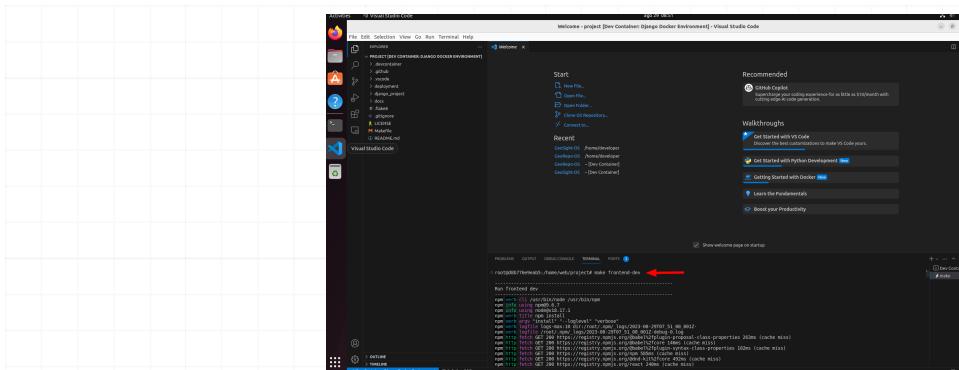
Open terminal within the dev container context by clicking the **+** icon in the terminal pane **1**. The new terminal **2** will show up in the list of running terminals **3**



1.1.1.2 Install FrontEnd libraries

Code:

make frontend-dev



1.1.1.3 Run django migration

Code:

```
cd /home/web/project/django_project
python manage.py migrate
```

1.1.1.4 Create super user

Code:

```
cd /home/web/project/django_project
python manage.py createsuperuser
```

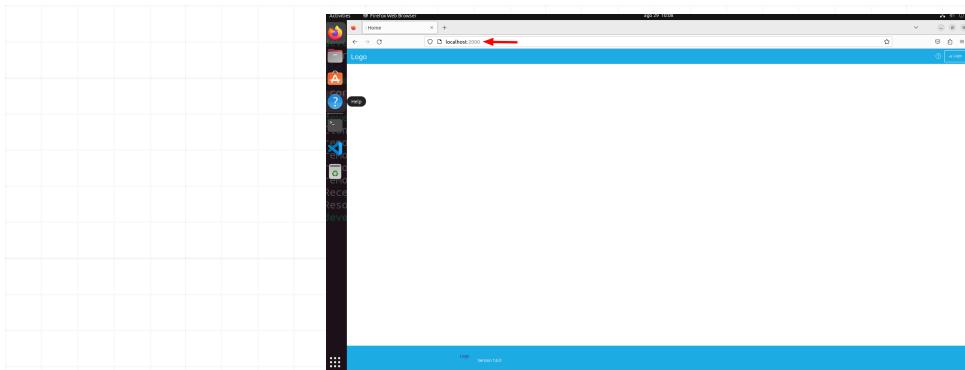
During this process you will be prompted for your user name (defaults to root), email address and a password (which you need to confirm). Complete these as needed.

1.1.1.5 Viewing your test instance

After completing the steps above, you should have the development server available on port 2000 of your local host:

Code:

```
http://localhost:2000
```



The site will be rather bare bones since it will need to be configured in the admin area to set up the theme etc.



1.1.1 Design

This section describes the visual design approach for the user interface.

Design Overview

1.1.1.1 Design sheet

The following documents provide guidance for the standard visual design we use when building the site:

Preview	Name	Notes
Colours	Colours	Colour Palettes
Buttons	Buttons	Designs for Buttons.
Form Fields	Form Fields	Form elements such as input boxes, select lists, radio buttons etc.
Icons	Icons	Standard iconography to be used in forms and views.
Tables	Tables	Design for tables and presentation of tabular data.
Others	Others	Other user interface elements.

Once you have reviewed the design sheets, you can move on to our [Roadmap](#).

1.1.1.2 Wire-frame model

Wire-frame Model



1.1.1 Project roadmap



1.1.1 Developer workflows

1.1.1.1 Adding a feature

- Create an Issue
- Wait for it to be added to a Sprint
- Functional Tests
- Playwright Tests
- Write end user documentation

1.1.1.2 Fixing a bug

- Claim an Issue
- Wait for it to be added to a Sprint
- Regression Test
- Implement Fix

1.1.1.3 Make PR for feature/bug fix

Committing to project

Follow our [Commit message conventions](#).

Pull request template

If it has related issues, add links to the issues(like `#123`) in the description. Fill in the [Pull Request template](#) by check your case.



1.1.1 Pull Request template



1.1.1 Commit message convention



title: Automatically Generated Python Documentation summary: DO NOT EDIT THIS FILE MANUALLY : It is created during the mkdocs build process date: 2023-08-03

1.1 Python Reference Manual

::: manage ::: initialize ::: core.celery ::: core.context_processors ::: core.wsgi ::: core.urls ::: core.asgi :::
core.settings.contrib ::: core.settings.utils ::: core.settings.prod ::: core.settings.project ::: core.settings.base :::
core.settings.dev ::: frontend.views ::: frontend.urls ::: frontend.apps ::: frontend.admin



1.1 Documentation Writing

1.1.1 Documentation

In this section of the documentation, we explain how to work with the documentation. The content is organised as follows:

- **Overview:** Here we outline the documentation system and the standard workflows needed.
- **Context help:** Here we outline the procedure for adding context help into the application and linking it to this documentation tree.



1.1.1 Documentation Overview

This document describes the easiest workflow for editing documentation.

Requirements:

1. You need to have a GitHub account and be logged in to your account
2. You need to have been given access to the repo by the repo manager

1.1.1.1 General Workflow

1. Go to the repo for the documentation.
2. Press the `.` key on your keyboard.
3. Select a file under the `src` folder to edit
4. Press `Ctrl-S` to save your work.
5. Go to the Source Control tab to commit your work.
6. Add a short, descriptive comment describing your changes.
7. Press the **Commit and push** button.
8. Wait a few minutes and your changes should be published live.

1.1.1.2 Adding a new page

Any new page you create needs to be also added to `mkdocs-base.yml` so that it gets 'built'. If you wish to build a page but not have it in the menu system, you can give it a blank menu description e.g.

Code:

```
# Pages to render but exclude from navigation
- "": developer/guide/templates/pull-request-template.md
```

Conversely, to ensure it is shown in the menu, find the right place in the navigation tree and then insert it with a short descriptor. e.g.

Code:

```
# Pages to render and include from navigation
- "My Menu Item": developer/guide/my-page.md
```

1.1.1.3 Adding images

You can easily upload images into the documentation sources and then add them to your document.

1. Take an image using your favourite screenshot tool.
2. Using your file manager, drag the file from your desktop into the `img` folder in the relevant part of the documentation you are working on.
3. **Shift+Drag** the image into your markdown document.
4. Edit the image description (the part in square brackets)

1.1.1.4 Adding links



You can add a link to any text by doing the following:

1. Copy the link from your web browser to your clipboard.
2. Either 2.1 Past the link directly into the document sources. 2.2 or, write some words in square brackets and paste the link in round brackets after, VSCode will create a markdown formatted link.

This is a normal link <https://staging-geosight.unitst.org/>, [this is a link](#).

The above in markdown:

 **Code:**

```
This is a normal link https://staging-geosight.unitst.org/, [this is a link](https://staging-geo
```

1.1.1.5 Page previews

You can easily preview the page you are working on by doing this:

1. Press **Ctl-Shift-V** to open a preview of the page you are currently working in.
2. Drag and drop the preview tab to the right side of the editing environment for a side-by-side view.

1.1.1.6 Leaving the editor viewer

How to leave the interactive editor.

1. Click the 'hamburger' menu and go to the repository.
2. Wait a few moments and the 'normal' GitHub page will load.



1.1.1 Documentation Overview

This section provides an overview of how the documentation system works.

1.1.1.1 Background

Welcome to the instructions for those wishing to contribute to our documentation!

Our documentation system is based on the popular [mkdocs](#) system. We encourage you to read the upstream documentation from mkdocs for the finer details of how mkdocs. In particular, read [this page](#) before you get to work. Mkdocs is not hard, but it may feel initially a little different if you are used to creating your documentation in word.

We use mkdocs because it has some great and useful features. Most especially:

- The code and the documentation are versioned together. This means that for whatever version of the code we have in our production release, we can be sure the documentation matches that release. We can also look back at changes over time through the version history when needed.
- It can render beautiful static (not needing a backend server) web sites with great features like search, styling, intuitive navigation etc.
- It can optionally render PDF documents from the same source markdown.
- An ecosystem of plugins and themes makes mkdocs really flexible and extensible.

Some terminology

In our documentation, we have organised the information using a specific nomenclature:

- **guide:** A guide is a multi-step workflow / tutorial that leads you through common activities.
- **manual:** A collection of documents that describe each part of the project in a matter of fact way without presupposing any specific workflow is being undertaken.

For each of the main topic areas (user, admin, developer, devops, api user) we provide guides and manuals as is appropriate.

Organisation

The documentation is broadly organised in the following structure. This structure may change over time as sections are added / moved / removed:



Code:

```
src
├── about
├── administrator
│   ├── guide
│   └── manual
├── developer
│   ├── documentation
│   ├── guide
│   └── manual
├── devops
│   ├── guide
│   └── manual
└── user
    ├── api
    ├── guide
    ├── manual
    └── quickstart
```



1.1.1.2 File naming conventions

Please follow these conventions when writing your documentation:

1. Write file names in all lower case, with hyphens separating words (no spaces). e.g. **important-file.md** not **Important File.md** or other deviations.
2. Place the image files in the img folder adjacent to your content.
3. Do not abbreviate any words in file names - but try to use terse, descriptive names.
4. Place your content into the appropriate place as outline in the structure above.
5. Each file should only one level one heading e.g. **# Page title**
6. Headings and subheadings should be written as a terse phrase starting with a capital letter and subsequent words in lower case unless they are proper nouns. Do not place a colon after the heading.
7. Use unicode number markers to reference numbers in images e.g. **1**
8. Every heading / sub-heading should have a single line underneath it summarising the content that will follow. This sentence will be used in the help center in the application to give the user a reference point for the text that will follow. It also makes the help text more consistent and readable.
9. For every section that images, the first image will be used as the thumbnail if the heading is referenced from the application.

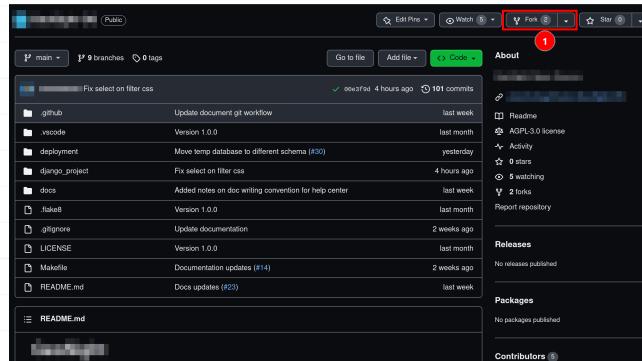
1.1.1.3 Contributing to the documentation



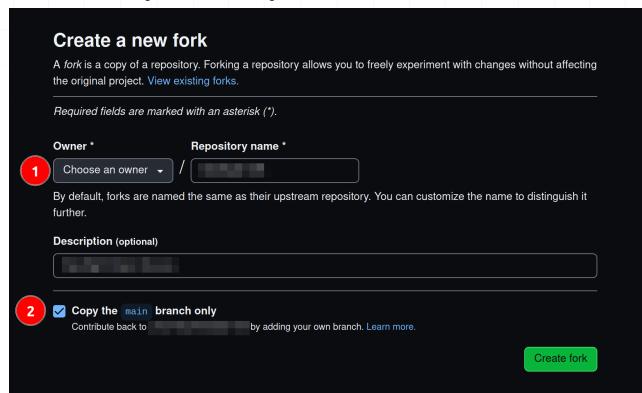
Note: The repo name in the images below is blurred out, you should work against the GeoRepo-OS repository when following these notes.

To start adding to the documentation navigate to the [home page](#) of the repository.

Once on the repository home page, **1** click on Fork



On the next page **1** Make sure your github account is selected as the owner and **2** make sure the "Copy the **main** branch only" option is ticked as you will only need the main branch.



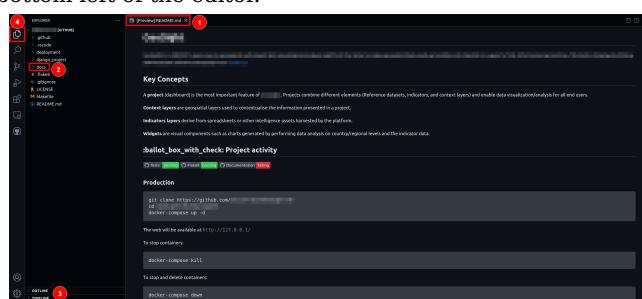
Once you have forked the repository, ensure that you are working in the main branch.



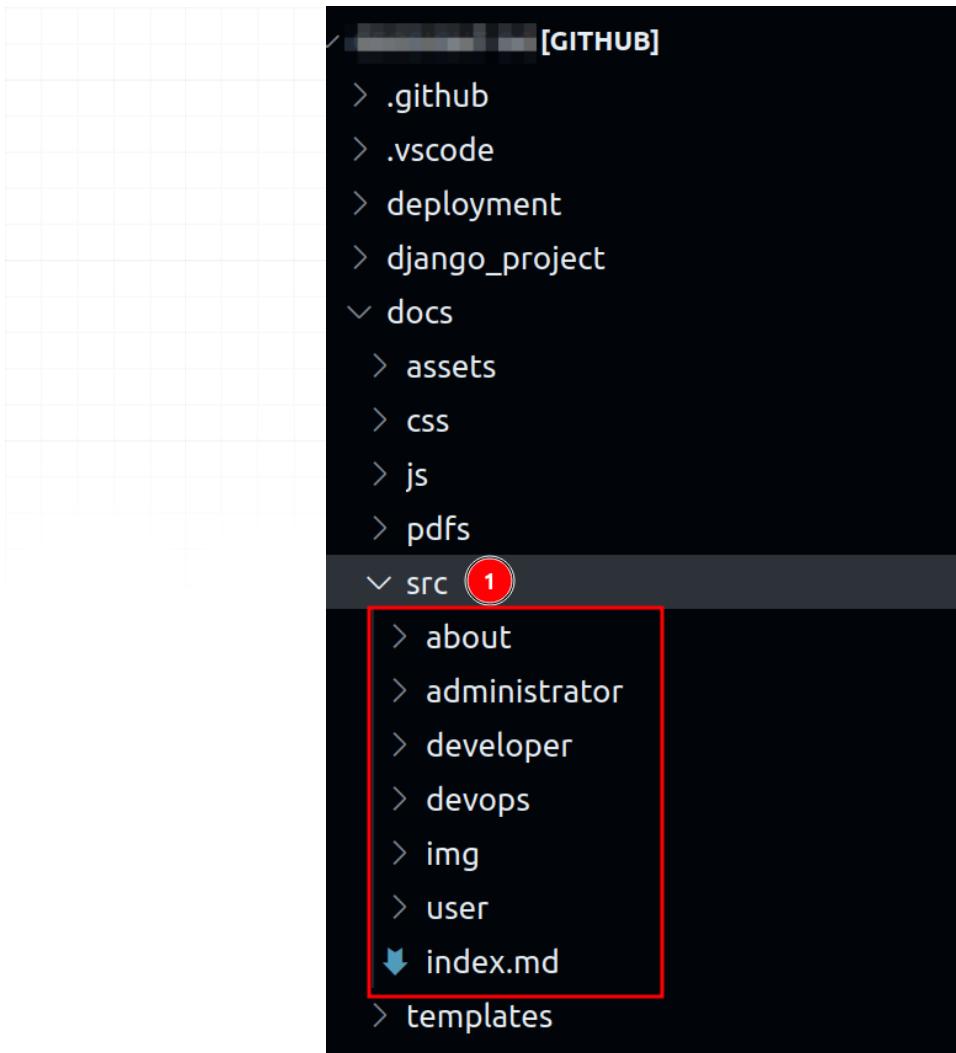
Once you are in your fork of the repository, click on **.** on your keyboard. This will open a web editor where you can begin adding to, or editing, the documentation.



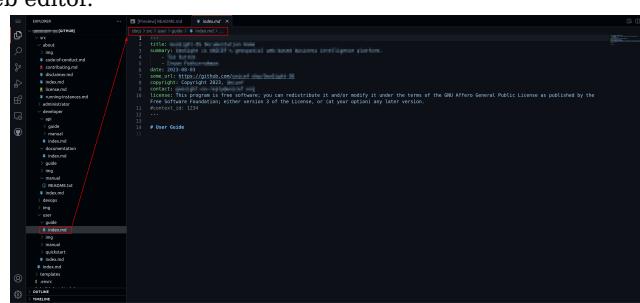
Once the editor loads, you will be greeted by **1** a preview of the project's README.txt. **2** Click on the **docs** directory in the menu on the left, this will expand the directory. To confirm that you are working in the right branch **3** it will say **main** in the bottom left of the editor.



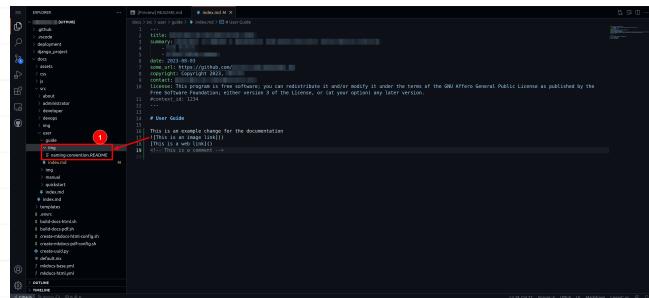
Once you have expanded the **docs** directory, **1** click on the **src** directory. All sub-directories within the **src** directory contain the files that become the documentation. Each sub-directory contains an **index.md** file that is required for the building of the documentation and static site. If you add a new sub-directory into any folder it must also contain a populated **index.md** file.



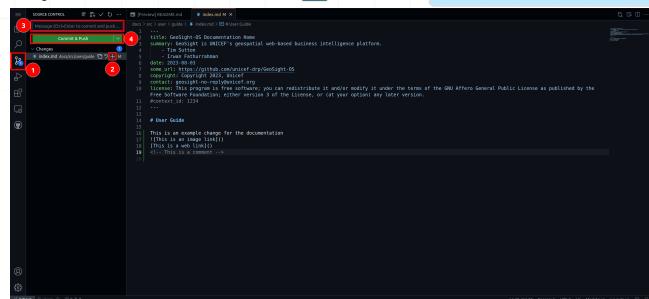
Once you have chosen which section of documentation you would like to edit (e.g. user/guide/index.md), click on the file and it will open in the web editor.



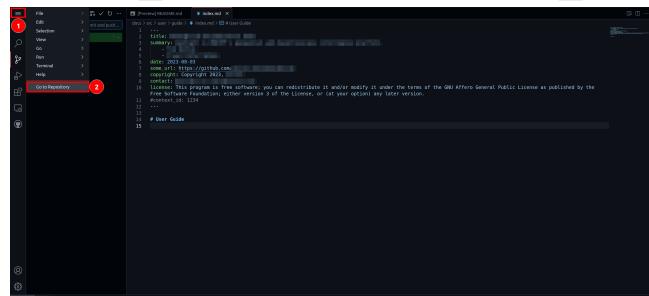
Once you have the file open, you can start adding to the documentation using [Markdown](#) syntax. If you need to add images to your documentation, add them to the relative `img` sub-directories following the naming conventions set out in the `naming-convention README` within the `img` folders.



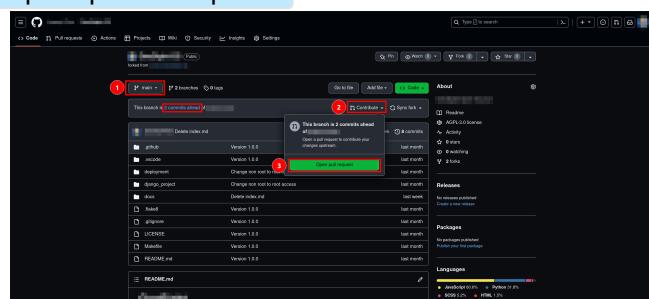
When you have completed making additions to (or editing) the documentation, **1** click on the source control tab then **2** click on the **plus** symbol next to the changes you are finished with to stage them, **3** add a commit message that is associated with the work you have done, and the **4** Click on **Commit & Push**



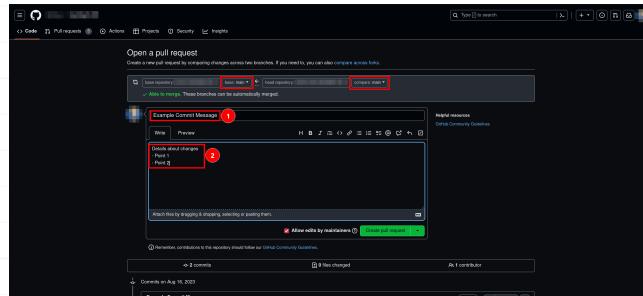
Once you have committed your changes, **1** Click on the burger menu and then **2** click on **Go To Repository**



Your fork of the repository will be opened in a new tab within your browser, navigate to that tab. Once there **1** ensure you are on the **main**, you should see how many commits ahead of the upstream branch you are, then **2** click on **Contribute -> Open pull request**.



On the next screen, **1** give your pull request a meaningful title, **2** give additional details regarding changes made in the larger text box, then **3** click on **Create pull request**. Also ensure you are creating a pull request to the upstream **main** branch from your **main** branch.



Once your pull request is opened you need to wait for it to be merged before you can open a new one.

1.1.1.4 Creating new files

Page metadata

Whenever you create a new file, you should add the standard header to the top of the file:

Code:

```
---
```

```
title: GeoRepo-OS Documentation
summary: GeoRepo is a UNICEF's geospatial web-based data storage and administrative boundary har
  - Tim Sutton
  - Dimas Tri Ciputra
  - Danang Tri Massandy
date: 2023-08-03
copyright: Copyright 2023, Unicef
contact: georepo-no-reply@unicef.org
license: This program is free software; you can redistribute it and/or modify it under the terms
#context_id: 1234
---
```

The summary can be updated to include your name for documents you contribute to.

The context_id is used to create a unique permalink to this document, and is optional. See further down in this document for more details.

Navigation

All pages need to be added to the 'Nav' section of the `mkdocs-base.yml` file. This will ensure that it is available in the menus and that mkdocs renders without errors. For example, if you wish to add a new page to the User Guide area, in Nav you would add a line like this:

- "Navigating the map": `users/guide/map-navigation.md`

Sometimes you may not wish to have the new page displayed in the menu system (this can be useful when, for example, it is accessed via a link in another page). In these cases, leave the menu entry part blank e.g.

- "": `users/guide/map-navigation.md`

Mkdocs will still generate the page but it will not be added to the menu.

1.1.1.5 Technical notes

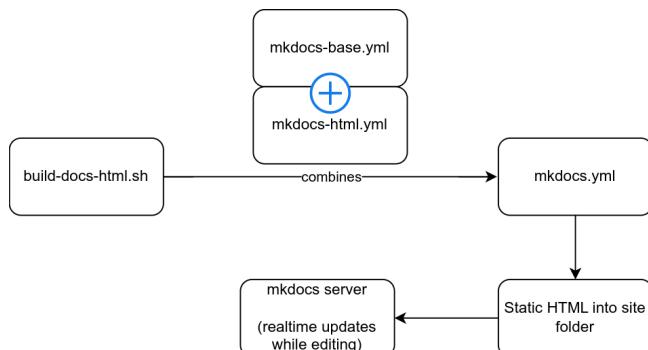


Working locally

If you want to work with the documentation locally (i.e. directly on your PC), you need to follow this general process:

1. Install python
2. Install pip
3. Install virtualenv (`pip install virtualenv`)
4. Make the docs folder your working directory (`cd docs`)
5. Create the virtual env in the docs folder (`python -m venv env`)
6. Activate the venv (`source env/bin/activate`)
7. Install the python modules listed in docs/requirements.txt (`pip install -r requirements.txt`)
8. If you are on Linux or macOS, open the docs directory in a shell and run `build-docs-html.sh`
9. In the docs directory, run `mkdocs serve`
10. Open your web browser at <https://localhost:8000> to view the rendered docs.

Note that `mkdocs serve` will dynamically re-render the docs any time you make a change. The process above is illustrated in the diagram below:



Hooks

We implement two hooks (plugins for mkdocs that are invoked during the docs rendering process).

- **uuid_redirects_hook.py** - this is used to handle page redirects from a uuid - see below for more details
- **python_manual_hook.py** - this is used to auto-generate the reference guide for the python modules, classes and functions that make up the code base for this project.

Permalinks for user and admin docs

If you are a developer, you should be aware of the workflow for creating user and administrator documentation for each page you create.

Every page should have a help link on it that leads to the appropriate manual page. The workflow for doing this is:

1. Generate a new page UUID using the provided python utility e.g. `./create-uuid.py`
2. Create a new page in the appropriate manual section e.g. `docs/src/user/manual/login.md`
3. In the metadata section at the top of the page, add the context id e.g. `context_id: V4cVEFd2TmwYJVb5HvWRwa`
4. In your django view, set up your help button to point to the site url and your context id. e.g. <https://siteurl/V4cVEFd2TmwYJVb5HvWRwa>

Whenever the user visits the page using the UUID URL, they will be redirected to the correct page e.g. <https://siteurl/login/>. This system protects us from file renaming and reorganising on the site, and ensures that the help link will always remain valid.



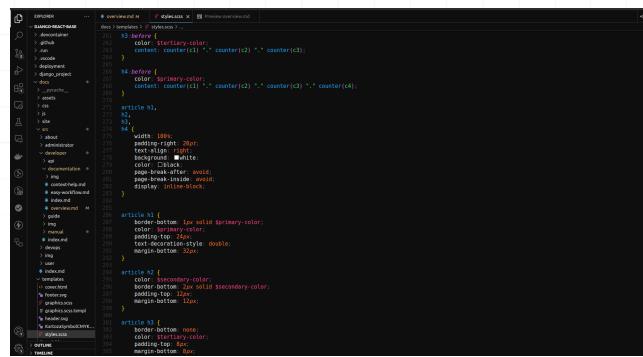
Generating PDFS

To generate PDFS, `cd` into `project_base_repo/docs` and then run the `build-docs-pdf` in a local terminal,

Modify PDFS Styling

To customise the styling of the pdfs , you can modify the `styles.scss` file in the `docs/templates` directory.

This is how it looks.



```
article {
    &::before {
        background-color: #fff;
        content: counter(c1) " " counter(c2);
    }
    &::after {
        color: $primary-color;
        content: counter(c1) " " counter(c2) " " counter(c3) " " counter(c4);
    }
    width: 100px;
    height: 20px;
    text-align: right;
    color: white;
    color: black;
    page-break-after: always;
    margin-bottom: 10px;
    display: inline-block;
}

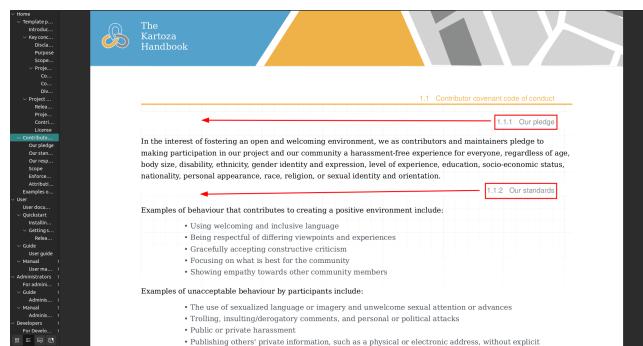
article h1 {
    border-bottom: 1px solid $primary-color;
    color: $primary-color;
    padding-bottom: 2px;
    text-decoration-style: double;
    margin-bottom: 12px;
}

article h2 {
    color: $secondary-color;
    border-bottom: 2px solid $secondary-color;
    padding-bottom: 12px;
}

article h3 {
    border-bottom: none;
    color: $secondary-color;
    padding-top: 10px;
    margin-bottom: 10px;
}
```

for example:

- We want to change the alignment of the headings from right to left.

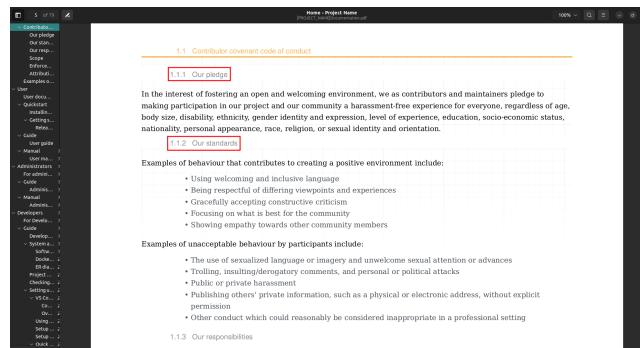


- To change the heading alignment we have to change the `text-align: left`.



```
267 color: $primary-color;
268 content: counter(c1) "." counter(c2) "." counter(c3) "." counter(c4);
269 }
270
271 article h1,
272 h2,
273 h3,
274 h4 {
275 width: 100%;
276 padding-right: 20px;
277 text-align: right;
278 background: white;
279 color: black;
280 page-break-after: avoid;
281 page-break-inside: avoid;
282 display: inline-block;
283 }
284
285
286 article h1 {
287 border-bottom: 1px solid $primary-color;
288 color: $primary-color;
289 padding-top: 24px;
290 text-decoration-style: double;
291 margin-bottom: 32px;
292 }
293
294 article h2 {
295 color: $secondary-color;
296 border-bottom: 2px solid $secondary-color;
297 padding-top: 12px;
298 margin-bottom: 12px;
299 }
300
301 article h3 {
302 border-bottom: none;
303 color: $tertiary-color;
304 padding-top: 8px;
305 margin-bottom: 8px;
```

- This is the PDF after changing the alignment



Modify Cover Page of PDFS

To customise the the cover page of the pdfs, you can modify the `cover.html` file in the `templates/cover.html`.

- As you can see in the below image the title of the pdf is **The Kartoza Handbook**.



- Now we want it to be **Kartoza Documentation**, to do this we have to change the title in the `cover.html` file.



```
docs > templates > cover.html > ...
1  <div>
2    <p style="font-size: xx-large; padding-top: 120px;">Kartoza Documentation</p>
3  </div>
4  <div class="cover-logo"></div>
5  <div>
6    <p style="font-size: large; padding-top: 120px;">Your Open Source Geospatial Experts.</p>
7    <p style="font-size: small; padding-top: 10px;">spatializing Pty (Ltd.)</p>
8    <p>•p-20-22/p>
9  </div>
```

- This is the cover page after changing the title.



Modify Document Title of PDFs

To change the document title of the pdfs, you can modify the `variable.scss` file in the `templates/variables.scss`.

- As you can see in the below image the document title is `The Kartoz Handbook`.



- Now we want it to be `Kartoza Documentation`, to do this we have to change the document title in the `variables.scss` file.

Modify the Navigation Bar Generating Static Site

To generate HTML, `cd` into `GEOREPO-OS/docs` and then run the `build-docs-html` in a local terminal. You can then run `mkdocs serve` to generate the static site on your local host, if there is a port conflict you can specify the port using the `-a` flag e.g `mkdocs serve -a 127.0.0.1:8001`.

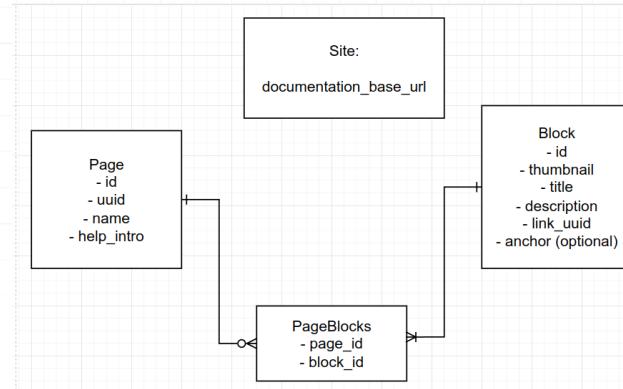


1.1.1 Context Help

This section describes how the context help system works.

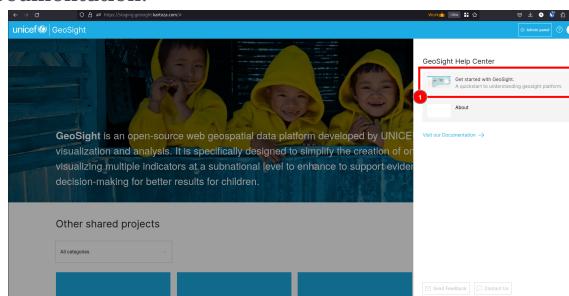
1.1.1.1 Overview

The context help system provides an integration between the web application and the documentation.

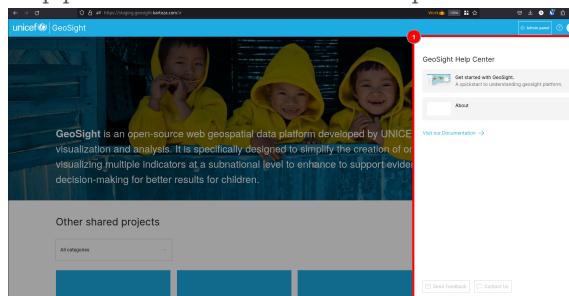


The following concepts are used:

- **Site:** This is the area for globally setting options relating to the context help system.
- **Blocks:** These are small components displayed as blocks 1 in the context help panel. Each block links to a section in the help documentation.



- **Pages:** These are context help panels that are each linked to a particular URL in the application.



- **Page Blocks:** These are links between pages and their constituent blocks. Each page can have many blocks and each block can be used in many pages.

1.1.1.2 Triggering context help



There are three ways to trigger context help:

- From the top nav bar 1

The screenshot shows the GeoSight homepage. At the top, there is a navigation bar with icons for user profile, search, and other links. Below the navigation bar is a banner featuring three children in yellow hooded jackets. The banner text reads: "GeoSight is an open-source web geospatial data platform developed by UNICEF for easy data visualization and analysis. It is specifically designed to simplify the creation of online maps for visualizing multiple indicators at a subnational level to enhance the support evidence-based decision-making for better results for children." Below the banner, there is a section titled "Other shared projects" with four project cards: "ACLED EMERGENCY", "Afghanistan", "Alg Project Demonstration", and "Alg_Test_FM". A red arrow points to the question mark icon in the top right corner of the navigation bar.

- From the side panel 1

The screenshot shows the GeoSight documentation center sidebar. On the left, there is a tree view of the documentation structure under "Documentation Center". One of the items, "Help", has a red arrow pointing to it. The main content area shows a table of recent activities with columns for "Project Name", "Description", "Category", and "Last Modified". A red arrow also points to the question mark icon in the bottom right corner of the sidebar.

- Next to widgets (not implemented yet)

1.1.1.3 Managing context help

We manage the context help system from django admin. You need to be a staff user to do this and have been granted read/write/update permissions on the documentation center related tables. First open your user menu 1 and then the Django Admin area 2.

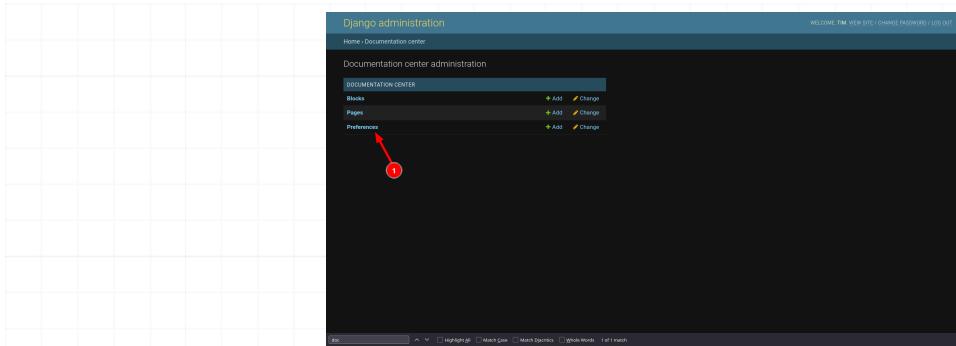
The screenshot shows the GeoSight documentation center sidebar. On the left, there is a tree view of the documentation structure under "Documentation Center". One of the items, "Help", has a red arrow pointing to it. The main content area shows a table of recent activities with columns for "Project Name", "Description", "Category", and "Last Modified". A red arrow also points to the question mark icon in the bottom right corner of the sidebar.

Next we can drill into the documentation center to see only the relevant admin activities 1.

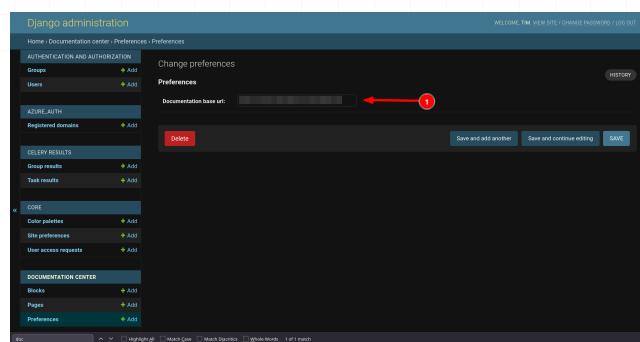
The screenshot shows the Django Admin interface. On the left, there is a tree view of the documentation structure under "DOCUMENTATION CENTER". A red box highlights this section, and a red arrow points to the "Blocks" item. The main content area shows a table of recent activities with columns for "Groups", "Users", "AZURE_AUTH", "CELERY RESULTS", "CORE", and "DOCUMENTATION CENTER". A red arrow also points to the question mark icon in the bottom right corner of the sidebar.

1.1.1.4 Site preferences

Our first activity is to edit the site preferences 1.

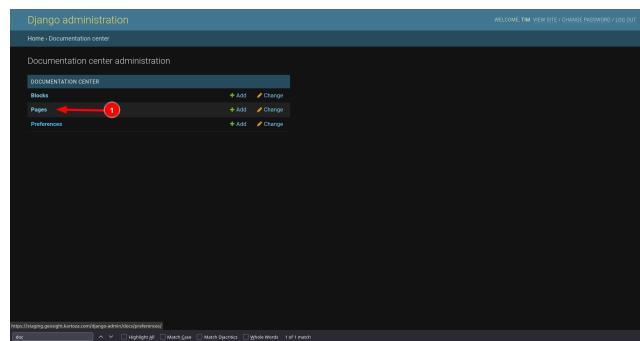


Here our goal is to set the documentation base url for the context help system 1. Use the URL for your documentation project i.e. <https://unicef-drp.github.io/GeoSight-OS>

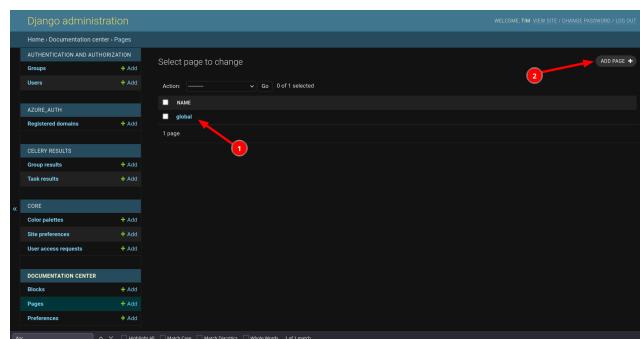


1.1.1.5 Managing Pages

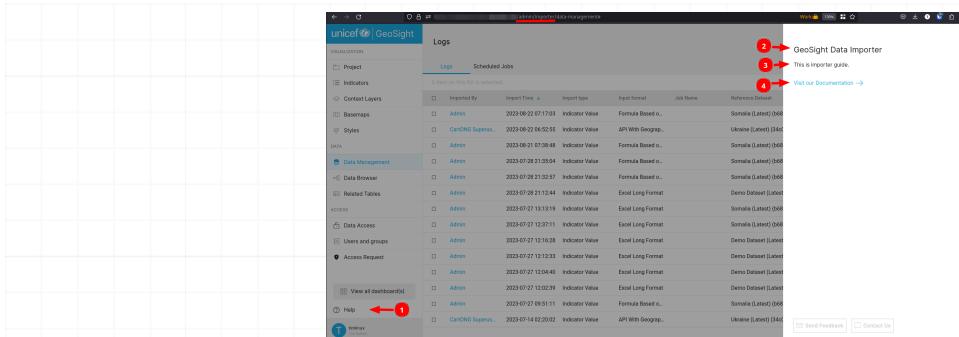
Next we need to create pages for each application end point where we want to include context sensitive help. First select pages 1 from the documentation center menu:



Now you can edit an existing page 1 or add a new one 2:

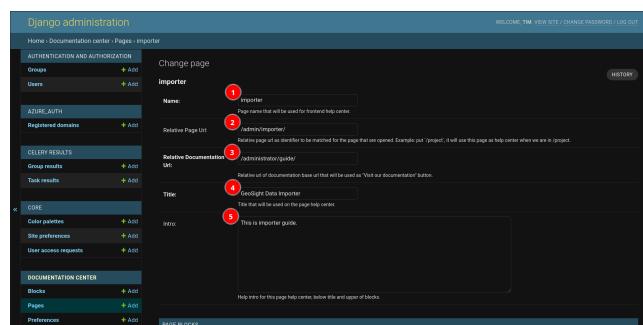


The page properties you can set are going to determine how the context help panel appears in the application when triggering the help panel 1:



Here is a detailed breakdown of the options:

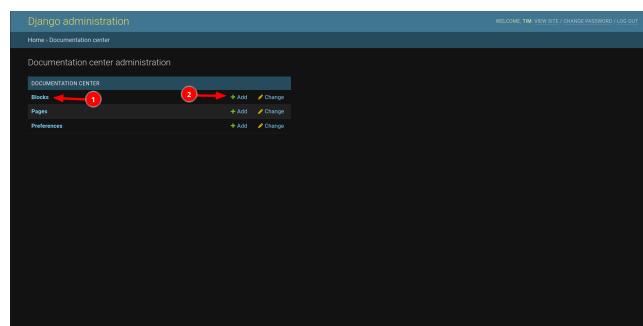
- 1 Name:** A symbolic name for you to easily recognise what the context help is about
- 2 Relative page url:** This is the relative path to our application page (see underline in image above) e.g. /admin/importer/
- 3 Url:** This is where the documentation is, relative to the documentation base URL. For example if your docs base URL is at: <https://unicef-drp.github.io/GeoSight-OS/> your help context document relative URL is /administrator/guide/. This will trigger the content panel link **3** above to point to this page.
- 4 Title:** This will be displayed as **1** above in the context help panel.
- 5 Intro:** This will be shown as lead-in text for the context help panel. See **2** above.



On the page form you can also manage the blocks that appear in the context help.

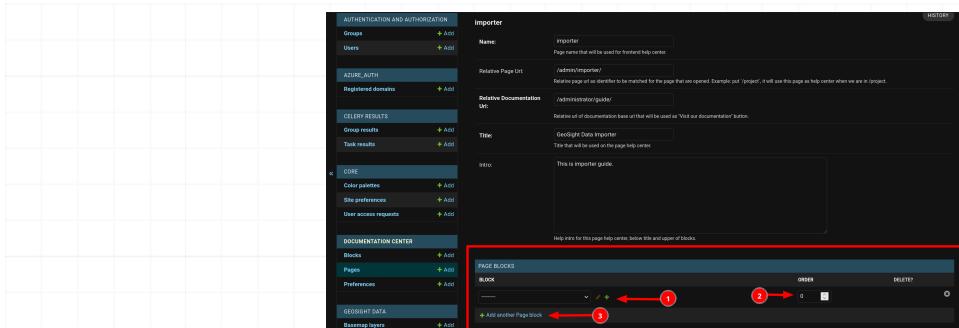
1.1.1.6 Managing blocks

You can view and manage existing blocks from the documentation center **1** or add a new block **2**. We first create a block then go back to the page to add it to the page.

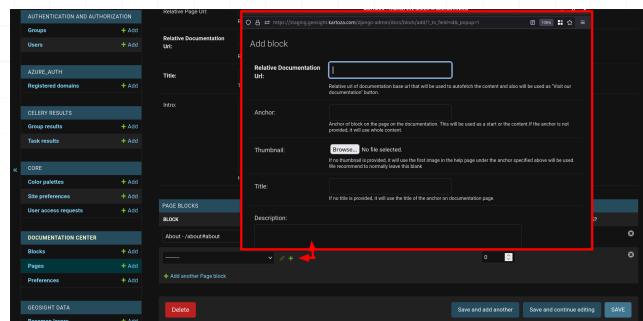


We can also create a new block directly from within a page.

- To set a block link within the page use **1**
- To set the order in which the block appears in the help panel use **2**. For ordering, lower number blocks will be listed before higher number blocks. It is up to you to ensure that each number used for ordering is unique.
- To create a new block component for a page, use **3** and then go and choose or create the actual block content.



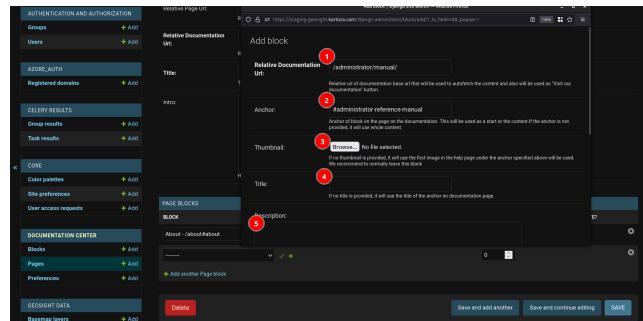
Let's create a new block within the context of an existing page (this is the more normal workflow).



Because we have created it within the page form, the block form appears as a popup window. We can deal with the form components as follows:

- **The relative url** 1 (based on the the documentation site wide base url preference) to the documentation page
- **The anchor** 2 - this is optional and should point to a heading / subheading within the page if populated.
- **The title** 3 - title for the block. This is automatically filled in from the page unless you specify an override title. Normally you would leave this blank.
- **The thumbnail** 4 for the block. This is automatically linked to the first image below the anchor in the help document. Usually you would leave this blank.
- **Description** 5 for the block. This will be automatically populated from the first paragraph in the linked help documentation.

⚠ As you can see, the block definition requires only minimal information - the URL and optionally the anchor on the page. Anything else you specify here will override information harvested from the page and likely make your documentation more difficult to maintain.



1.1.1.7 Page Blocks

Generally we will not directly use the PageBlocks management area - use the Page manager to add and remove blocks from your page.

1.1.1.8 Conclusion and notes

The system for context help has been designed in a generic way. That means that you can use any web site where you can reference content with page links and anchors.



➡ One word of caution. Do not use a web site as your documentation source if you do not control that site. The content from the help site is pulled directly into your application, which could put you in a compromising position if the content is not 'on message' for your organisation.



1.1 API

1.1.1 API documentation

This is the homepage for all API related documentation.

- The [api guide](#), which describes common workflows for using the restful API.
- The [api manual](#), which describes each API endpoint and its parameters, return values etc. This section is autogenerated.



1.1.1 API manual



1.1.1 API guide



1 DevOps

1.1 DevOps documentation

This section contains all documentation relevant to DevOps procedures.

The devops content is divided into two sections:

- The [devops guide](#), which describes common workflows for deployment and management of running instances in a tutorial format.
- The [devops manual](#), which provides links and references to deployment configuration files, frameworks and deployment systems used etc.



1.1 DevOps guide



1.1 DevOps documentation

- 1.1.1 Containers used
- 1.1.2 SDLC
- 1.1.3 Continuous integration
- 1.1.4 Testing deployments
- 1.1.5 Backend orchestration
 - 1.1.5.1 Deployments
 - 1.1.5.2 Kubernetes



1 About

1.1 About

1.1.1 License

Published under AGPL-3. See the [full license](#) text.

1.1.2 Contributors

Contributor	GitHub Handle
-------------	---------------

Hugh Mann

[NotARobot](#)

" "

Tim Sutton

[timlinux](#)



1.1.3 Project funders/backers

This project was conceptualized and initially funded by [FUNDER_1](#).

1.1.4 Logos

Another fine project contributed to by: [Kartoza.com](#)





1.0.0.1 GNU Affero General Public License

Version 3, 19 November 2007 Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU Affero General Public License is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, our General Public Licenses are intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

Developers that use our General Public Licenses protect your rights with two steps: **(1)** assert copyright on the software, and **(2)** offer you this License which gives you legal permission to copy, distribute and/or modify the software.

A secondary benefit of defending all users' freedom is that improvements made in alternate versions of the program, if they receive widespread use, become available for other developers to incorporate. Many developers of free software are heartened and encouraged by the resulting cooperation. However, in the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.

The GNU Affero General Public License is designed specifically to ensure that, in such cases, the modified source code becomes available to the community. It requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

An older license, called the Affero General Public License and published by Affero, was designed to accomplish similar goals. This is a different license, not a version of the Affero GPL, but Affero has released a new version of the Affero GPL which permits relicensing under this license.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions

"This License" refers to version 3 of the GNU Affero General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you".

"Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that **(1)** displays an appropriate copyright notice, and **(2)** tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under



this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that **(a)** is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and **(b)** serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.



You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- **a)** The work must carry prominent notices stating that you modified it, and giving a relevant date.
- **b)** The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- **c)** You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- **d)** If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- **a)** Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- **b)** Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either **(1)** a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or **(2)** access to copy the Corresponding Source from a network server at no charge.
- **c)** Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- **d)** Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- **e)** Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.



A “User Product” is either **(1)** a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or **(2)** anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- **a)** Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- **b)** Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- **c)** Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- **d)** Limiting the use for publicity purposes of names of licensors or authors of the material; or
- **e)** Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- **f)** Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.



All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated **(a)** provisionally, unless and until the copyright holder explicitly and finally terminates your license, and **(b)** permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a



consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either **(1)** cause the Corresponding Source to be so available, or **(2)** arrange to deprive yourself of the benefit of the patent license for this particular work, or **(3)** arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients.

"Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license **(a)** in connection with copies of the covered work conveyed by you (or copies made from those copies), or **(b)** primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Remote Network Interaction; Use with the GNU General Public License

Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source for any work covered by version 3 of the GNU General Public License that is incorporated pursuant to the following paragraph.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the work with which it is combined will remain governed by version 3 of the GNU General Public License.



14. Revised Versions of this License

The Free Software Foundation may publish revised and/or new versions of the GNU Affero General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU Affero General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU Affero General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU Affero General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.



To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.



Code:

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU Affero General Public License for more details.
```

```
You should have received a copy of the GNU Affero General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If your software can interact with users remotely through a computer network, you should also make sure that it provides a way for users to get its source. For example, if your program is a web application, its interface could display a “Source” link that leads users to an archive of the code. There are many ways you could offer source, and different solutions will be better for different programs; see section 13 for the specific requirements.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU AGPL, see <<http://www.gnu.org/licenses/>>.



1.1 Contributing to [PROJECT_NAME]

First off, thanks for taking the time to contribute! 🎉 😊 ✨

The following is a set of guidelines for contributing to this project. These are mostly guidelines, not rules. Use your best judgment, and feel free to propose changes to this document in a pull request.

1.1.1 Reporting bugs

Bugs are tracked as GitHub issues. Search the list and try reproduce on your local machine with a clean profile before you create an issue. When you create an issue, please provide the following information by filling in the template.

Explain the problem and include additional details to help maintainers reproduce the problem:

- **Use a clear and descriptive title** for the issue to identify the problem.
- **Describe the exact steps which reproduce the problem** in as many details as possible. Don't just say what you did, but explain how you did it. For example, if you moved the cursor to the end of a line, explain if you used a mouse or a keyboard.
- **Provide specific examples to demonstrate the steps.** Include links to files or GitHub projects, or copy/paste-able snippets, which you use in those examples. If you're providing snippets on the issue, use Markdown code blocks.
- **Describe the behaviour you observed after following the steps** and point out what exactly is the problem with that behaviour.
- **Explain which behaviour you expected to see instead and why.**
- **Include screenshots and animated GIFs** which show you following the described steps and clearly demonstrate the problem.

1.1.2 Suggesting enhancements

In case you want to suggest an enhancement, please follow this guideline to help maintainers and the community understand your suggestion. Before creating suggestions, please check [issue list](#) if there's already a request.

Create an issue and provide the following information:

- **Use a clear and descriptive title** for the issue to identify the suggestion.
- **Provide a step-by-step description of the suggested enhancement** in as many details as possible.
- **Provide specific examples to demonstrate the steps.** Include copy/paste-able snippets which you use in those examples, as Markdown code blocks.
- **Include screenshots and animated GIFs** which helps demonstrate the steps or point out the part of project which the suggestion is related to.
- **Explain why this enhancement would be useful** to most users.
- **List some other text editors or applications where this enhancement exists.**

1.1.3 First code contribution

Unsure where to begin contributing? You can start by looking through these [document](#), [good first issue](#) and [help wanted](#) issues:

- **document issues:** issues which should be reviewed or improved.
- **good first issues:** issues which should only require a few lines of code, and a test or two.
- **help wanted issues:** issues which should be a bit more involved than beginner issues.

1.1.4 Pull requests

1.1.4.1 Development workflow

- Set up your development environment
- Make change from a right branch
- Be sure the code passes tests



- Make a pull request

1.1.4.2 Development environment

- Prepare your machine and its packages installed.
- Checkout our repository
- Install dependencies by `pip install -r REQUIREMENTS-dev.txt`

1.1.4.3 Make changes

Checkout a branch

- **master**: PR Base branch.
- **production**: latest release branch with distribution files. Never make a PR on this.
- **gh-pages**: API docs, examples and demo

Check code style

Run the pylance extension and make sure all the tests pass.

Test

Run **TODO** and verify all the tests pass. If you are adding new commands or features, they must include tests. If you are changing functionality, update the tests if you need to.

Commit

Follow our [commit message conventions](#).

1.1.4.4 Yes! Pull request

Make your pull request, then describe your changes.

Title

Follow other PR title format on below.

The screenshot shows a light blue rectangular area representing a pull request template. On the left side, there is a dark blue vertical bar containing a small icon of a person with a gear and the word "Code:". To the right of this bar is a large light blue area. Inside this area, there is placeholder text for commit messages:

```
<Type>: Short Description (fix #111)
<Type>: Short Description (fix #123, #111, #122)
<Type>: Short Description (ref #111)
```

- capitalize first letter of Type
- use present tense: 'change' not 'changed' or 'changes'

Description

If it has related issues, add links to the issues (like `#123`) in the description. Fill in the [Pull Request Template](#) by check your case.

1.1.5 Code of conduct



This project and everyone participating in it is governed by the [Code of Conduct](#). By participating, you are expected to uphold this code. Please report unacceptable behaviour to tim@kartoza.com.

 **Note:**

This guide is based on [atom contributing guide](#), [CocoaPods](#) and [ESLint](#)



1.1 Disclaimer

Disclaimer about project.