

How to Add Pages to Wagtail

You should read this file if you are tasked with adding pages to the StreetCard site, which uses *Wagtail* and *Django*. The process consists of four steps, each having a bit of work to do.

Step 1 - Create HTML pages

The first step is to, of course, make the HTML for the page you want.

However, the HTML you will be making is the one that Wagtail will use, which contains stuff that regular HTML does not, which is hinted by the uses of `{% ... %}` and `{{ ... }}` throughout the page. These are related to Wagtail. I recommend looking up Wagtail tutorials online regarding this.

Make sure you use `{% extends "base.html" %}` at the top when creating the page as `base.html` already contains everything that all the site pages have and thus you don't need to work more than you need to. Also, if you need to add a page to the navigation bar (which is in `base.html`), then wait until *after* Step 4 to do that.

In doubt, check the pages in `streetcard/home/templates/home` directory as a reference point.

Step 2 - Create Page Classes

Then, you have to make a Python class for that page, usually in the structure of `[PAGE_NAME]Page`, where `[PAGE_NAME]` is in *UpperCamelCase*. It should be the same as the name of the html file for said page (i.e. `[page_name]_page.html`, where `[page_name]` is in all lowercase with underscores ('_') between words).

The Python code for the page in the `models.py` file should look more-or-less like this, with the page name being `example`:

```
class ExamplePage(Page)

    class Meta:
        verbose_name = 'example_page'

    body = RichTextField(blank=True)

    template = "home/example_page.html"

    content_panels = Page.content_panels + [
        FieldPanel('body'),
    ]
```

Of course, depending on what kind of page it is, it would look different from this, but it is something you can fall back on if you are stuck.

Step 3 - Make Migrations

Now, you have to update Django by making migrations and migrating the new classes you made.

First, you need to set up the virtual environment, which is necessary to run the web server, opening the terminal under the `env/Scripts` directory and running the following command

```
activate
```

Or `activate.bat` if you on Windows.

Go to the `streetcard` directory from the main directory (by typing `cd ../../streetcard` into the terminal) and run the two commands:

```
py manage.py make-migrations  
py manage.py migrate
```

There should be no errors thrown in your direction if you did this right. If there are any, fix them and repeat the two commands again (given if you haven't closed the terminal. In that case, repeat the process from the beginning of Step 3 again).

Step 4 - Create Pages through Wagtail Admin

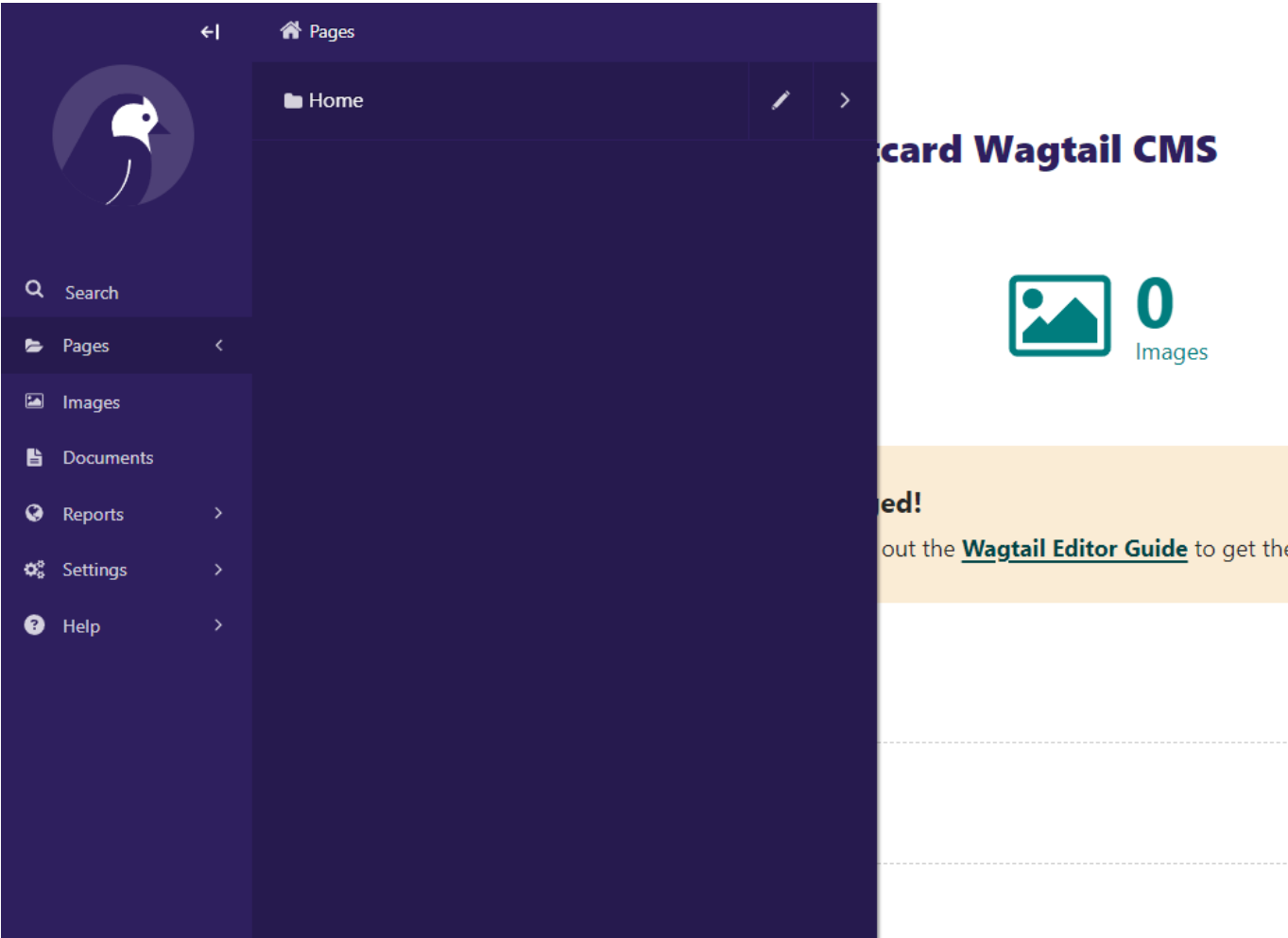
This step is important, as the pages you worked on won't be recognized or accessed to until you actually go through adding the page through Wagtail Admin. It is also where you add the contents of the page. I'll show you how that is done.

First, if you are still in your virtual environment (if not, repeat Step 3 again but without the migration commands), and start up the server in the same directory as before:

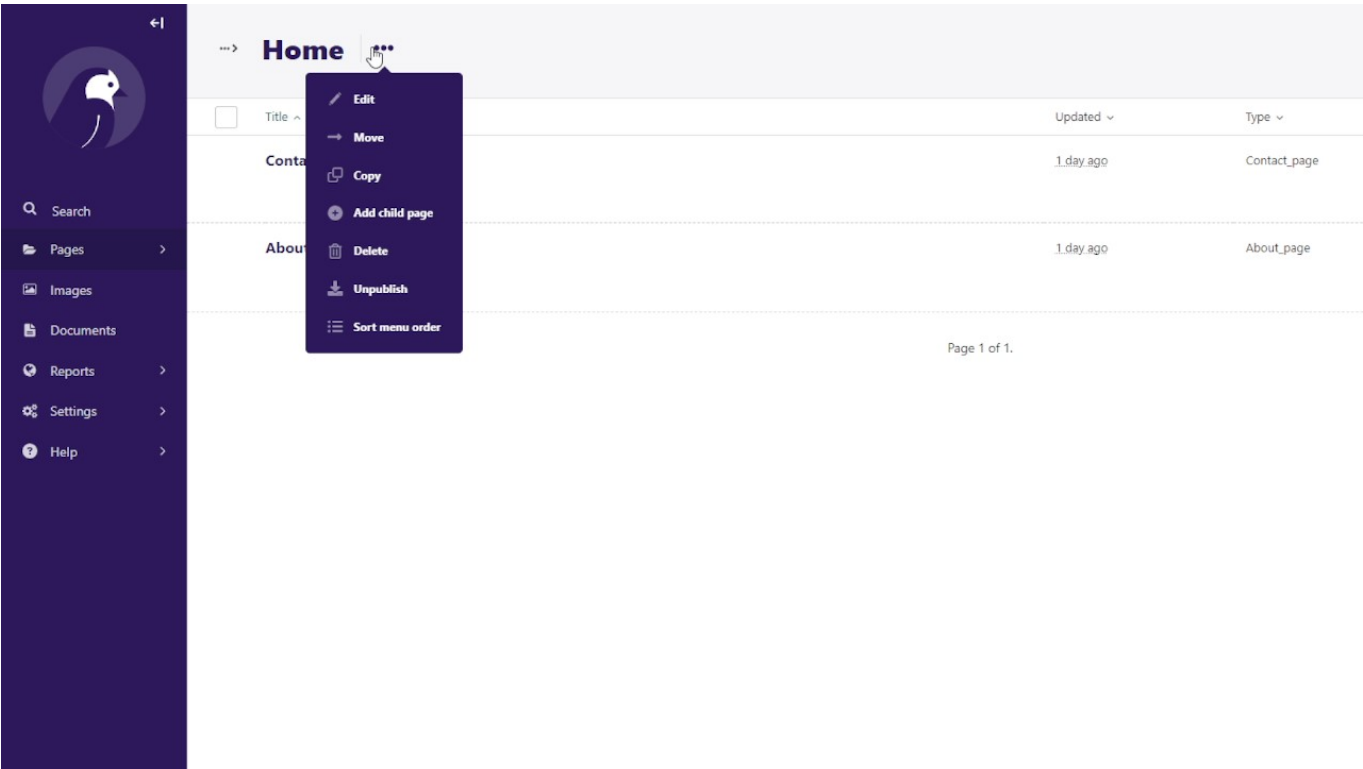
```
py manage.py runserver
```

Then, go to `127.0.0.1:8000/admin` and log in. there should either be a unified superuser account that everyone shares or you have to create one if you don't have one already. If you need to create a superuser for any reason, close the server (by pressing `Ctrl + C`) and run the command `python manage.py createsuperuser` and follow the steps it gives you (make sure you remember that information). Once you are done, start the server back up again and log in.

Next, go to **Pages > Home**, which can be found the image below:



Then, you go press the "..." button and select "Add child page".



Then follow the steps, selecting the HTML file that you want to add to the site (in this case, `example_page.html`), and fill in the necessary fields (you can edit them later).

After that, you should be all done! Congratulations, you added your page to the website!