

[🏠](#) » Development

Development

Installation

This project manages dependencies using [poetry](#)

Ensure you have poetry installed (`pip install poetry`)

Then get setup with `poetry install`

```
git clone git@github.com:farridav/django-jazzmin.git
poetry install
```

Running the test project

Setup db tables etc.

```
python tests/test_app/manage.py migrate
```

Generate test data

```
python tests/test_app/manage.py reset
```

Run development server (with werkzeug debugger)

```
python tests/test_app/manage.py runserver_plus
```

Running the tests

Tests are run via github actions on any pull request into `master`, and are written for use with the [pytest](#) framework, we should have good enough tests for you to base your own off of, though where we are lacking, feel free to contribute, but keep it clean, concise and simple, leave the magic to the wizards.

 v: latest ▼

Run the test suite with your current python interpreter and Django version using `pytest` or

target an individual test with `pytest -k my_test_name`

Run against all supported Python and Django Versions using `tox`

Contribution guidelines

- Fork the project
- Make a pull request against this repositories `master` branch,
- Include tests unless its a trivial change
- Add a screenshot if your proposing UI changes
- Demonstrate the change within the `test_app` if possible
- No breaking changes please

Coding guidelines

- autoformat your code with `black`
- When fixing something display related, please bear the following in mind:
 - Try fixing the problem using HTML, else CSS, else JS
 - Try removing code, else changing code, else adding code

Serving documentation locally

You can serve the docs locally using `mkdocs serve -a localhost:8001` and visiting <http://localhost:8001>

Translations

Working with translations in jazzmin is, a bit unorthodox, as we are overriding django's templates, so it looks like we have a lot of strings that need translating, but in-fact they already have translation strings in Django, heres the process for dealing with translations, though we recommend not adding new strings that need translating if possible, and use suitable iconography instead (See [Font Awesome 5.13.0 free icons](#)), or use a string that is already translated upstream in Django.

Adding a new language

1. cd into the jazzmin folder
2. Add the desired language directory e.g `mkdir -p locale/de/LC_MESSAGES`
3. Run `django-admin makemessages`
4. `cd ../`

 [v: latest](#) ▼

5. Run `./cli.py locales --prune de` to remove the django provided strings
6. Go through the strings in the locale file, any that are not genuinely new strings introduced by jazzmin, find them in the codebase, and try making them match the ones provided in.djangos admin/admin docs translation files
 - [Django admin translations](#)
 - [Django admin docs translations](#)

Once you have finished, run `makemessages` again, until the file contains ONLY unique strings to jazzmin, there should only be a handful