GeoNode / **geonode-project**

A django template project for creating custom GeoNode projects. http://geonode.org

177 commits | **6** branches | **0** releases | **11** contributors

Branch: **2.8.0** | **New pull request** | **Create new file** | **Upload files** | **Find file** | **Clone or download**

afabiani Merge pull request #33 from code-geek/patch-1 ··· Latest commit `a45494f` on Apr 20

fixtures	GeoServer 2.12.x	6 months ago
project_name	- fix settings.py	3 months ago
scripts/misc	Settings and Local Settings updated with Master branch (2.9.x)	5 months ago
.gitignore	- Update .gitignore	5 months ago
Dockerfile	Use dockerfile with preinstalled libraries	2 years ago
Makefile	Added pull to hardreset	2 years ago
README.rst	Update README.rst	a month ago
dev_config.yml	Update dev_config.yml	2 months ago
docker-compose.override.yml	Fix docker syntax	2 years ago
docker-compose.yml	Do not build containers locally apart from the django app	2 years ago
jetty-runner.xml	sync with upstream on Jan. 2, 2017	a year ago
manage.py	- copyrights to OsGeo	9 months ago
pavement.py	- fix paver checks	3 months ago
playbook.yml	GeoServer 2.12.x	6 months ago
requirements.txt	Update requirements.txt	2 months ago
setup.py	Update setup.py	2 months ago

README.rst

{{ project_name|title }}

GeoNode template project. Generates a django project with GeoNode support.

Create a custom project

Note: You can call your geonode project whatever you like following the naming conventions for python packages (generally lower case with underscores (_). In the examples below, replace `my_geonode` with whatever you would like to name your project.

Using Docker

To setup your project using Docker, follow these instructions:

1. Install Docker (for Linux, Mac or Windows).
2. Run the following command in a terminal.:

```
docker run -v `pwd`:/usr/src/app geonode/django:geonode django-admin.py startproject --template=https://g
cd my_geonode
```

If you experience a permissions problem, make sure that the files belong to your user and not the root user.

Using a Python virtual environment

To setup your project using a local python virtual environment, follow these instructions:

1. Setup your virtualenvironment `mkvirtualenv my_geonode`
2. Install django into your virtualenviornment `pip install Django==1.8.19`
3. Create your project using the template project:

```
django-admin.py startproject --template=https://github.com/GeoNode/geonode-project/archive/2.8.0.zip -e p
```

4. Install Python deps

> Warning
>
> you might need to fix pygdal version accordingly to your system. If so, update it on `requirements.txt` and retry.

```
cd my_geonode
pip install celery==4.1.0
pip install -r requirements.txt --no-warn-conflicts
pip install -e . --no-warn-conflicts
```

6. Start the project:

```
DJANGO_SETTINGS_MODULE=my_geonode.settings paver reset
DJANGO_SETTINGS_MODULE=my_geonode.settings paver setup
DJANGO_SETTINGS_MODULE=my_geonode.settings paver sync
DJANGO_SETTINGS_MODULE=my_geonode.settings paver start
```

7. Access GeoNode from browser:

```
http://localhost:8000/
```

Note

default admin user is `admin` (with pw: `admin`)

Start your server

You need Docker 1.12 or higher, get the latest stable official release for your platform. Run docker-compose to start it up (get a cup of coffee or tea while you wait):

```
docker-compose up
```

Create the tables in your postgres database:

```
docker-compose run django python manage.py migrate
```

Set up a superuser so you can access the admin area:

```
docker-compose run django python manage.py createsuperuser
```

Access the site on http://localhost/

Recommended: Track your changes

Step 1. Install Git (for Linux, Mac or Windows).

Step 2. Init git locally and do the first commit:

> git init
>
> git add *

> git commit -m "Initial Commit"

Step 3. Set up a free account on github or bitbucket and make a copy of the repo there.

Configuring Requirements.txt

You may want to configure your requirements.txt, if you are using additional or custom versions of python packages. For example:

```
Django==1.8.19
six==1.10.0
django-cuser==2017.3.16
django-model-utils==3.1.1
pyshp==1.2.12
celery==4.1.0
Shapely>=1.5.13,<1.6.dev0
proj==0.1.0
pyproj==1.9.5.1
pygdal==2.2.1.3
inflection==0.3.1
git+git://github.com/<your organization>/geonode.git@<your branch>
```

Using Ansible

To run the Ansible playbook use something like this:

```
ANSIBLE_ROLES_PATH=~/workspaces/public ansible-playbook -e "gs_root_password=<new gs root password>" -e "gs_a
```

Configuration

Since this application uses geonode, base source of settings is `geonode.settings` ***module. It provides defaults for many items, which are used by geonode. This application has own settings module,*** `{{project_name}}.settings` ***, which includes*** `geonode.settings` ***. It customizes few elements:***

- static/media files locations - they will be collected and stored along with this application files by default. This is useful during development.
- Adds `{{project_name}}` to installed applications, updates templates, staticfiles dirs, sets urlconf to `{{project_name}}.urls` .

Whether you deploy development or production environment, you should create additional settings file. Convention is to make `{{project_name}}.local_settings` ***module. It is recommended to use*** `{{project_name}}/local_settings.py` ***.. That file contains small subset of settings for edition. It should:***

- not be versioned along with application (because changes you make for your private deployment may become public),
- have customized at least``DATABASES``, `SECRET_KEY` and `SITEURL` .

You can add more settings there, note however, some settings (notably `DEBUG_STATIC` , `EMAIL_ENABLE` , `*_ROOT` , and few others) can be used by other settings, or as condition values, which change other settings. For example, `EMAIL_ENABLE` defined in `geonode.settings` enables whole email handling block, so if you disable it in your `local_settings` , derived settings will be preserved. You should carefully check if additional settings you change don't trigger other settings.

To ilustrate whole concept of chanied settings:

```
+-----------------------+-------------+------------------------------+-------------+-----------------------
|  GeoNode configuration |             |   Your application default    |             |  (optionally) Your dep
|                        |             |        configuration          |             |
+========================|=============|==============================|=============|=======================
|                        | included by |                              | included by |
|   geonode.settings     |     ->      |  {{project_name}}.settings    |      ->     |  {{project_name}}.loca
+------------------------|-------------|------------------------------|-------------|-----------------------
```