
terra-backend-crud

Release 0.3.11

Oct 10, 2019

Contents

1	Installation	3
1.1	Requirements	3
1.2	With pip	3
1.3	With git	3
2	Configuration	5
3	Example of use	7

Terralego backend app

1.1 Requirements

Must be installed, for the package to work:

- terracommon.terra
- template_model
- template_engines

1.2 With pip

From Pypi:

```
pip install xxxxxxxxxx-xxxxxxxxxxxx
```

From Github:

```
pip install -e https://github.com/Terralego/terra.backend.crud.git@master#egg=django-  
↪template-engines
```

1.3 With git

```
git clone https://github.com/Terralego/terra.backend.crud.git  
cd terra.backend.crud  
python setup.py install
```


CHAPTER 2

Configuration

In your project :

- settings

```
INSTALLED_APPS = [  
    ...  
    # apps required by CRUD  
    'geostore',  
    'template_model',  
    'template_engines',  
    # CRUD app  
    'terra_geocrud',  
    ...  
]  
...  
TEMPLATES = [  
    ...  
    # if you want to render odt templates  
    {'BACKEND': 'template_engines.backends.odt.OdtEngine'},  
    # if you want to render docx templates  
    {'BACKEND': 'template_engines.backends.docx.DocxEngine'},  
]
```

- urls

```
urlpatterns = [  
    ...  
    path('', include('terra_geocrud.urls', namespace='terra_geocrud')),  
    ...  
]
```

You can customize default url and namespace by including `terra_geocrud.views` directly

Run migrations

```
./manage.py migrate
```

- ADMIN :

you can disable and / or customize admin

- SETTINGS :

Waiting for settings definition directly in models.

Settings should be overridden with TERRA_GEOCRUD settings in your project settings file:

```
...
TERRA_GEOCRUD = {
    'EXTENT': [[-90.0, -180.0], [90.0, 180.0]], # default value for map extent. API_
    ↳serialize this for layer extent if there is no features in it (as default)
}
...
```

CHAPTER 3

Example of use

- By default, api endpoints are available under

```
/api/crud/
```

- There are 4 endpoint:

```
/api/crud/groups/          -> manage groups of CRUD views
/api/crud/views/          -> manage CRUD views (a view creation create_
↳ its associated layer)
/api/crud/settings/       -> get ordered menu with views classified by_
↳ group or not, and basic map settings
/api/crud/template/<template_pk>/render/<pk>/ -> fill a template with a feature
```

- A command is available to create default views for each existing layer

```
./manage.py create_default_crud_views
```

- **START GUIDE**

- First, you need to create crud views for your geostore layers with the command or the admin.
- These views can be grouped, and will be listed by the frontend api
- Then, you can customize default layer-schema by providing your own property groups, which will groups properties as json schema nested objects.

ADMIN

- access to /admin/terra_geocrud/