

How to

Generate Docs

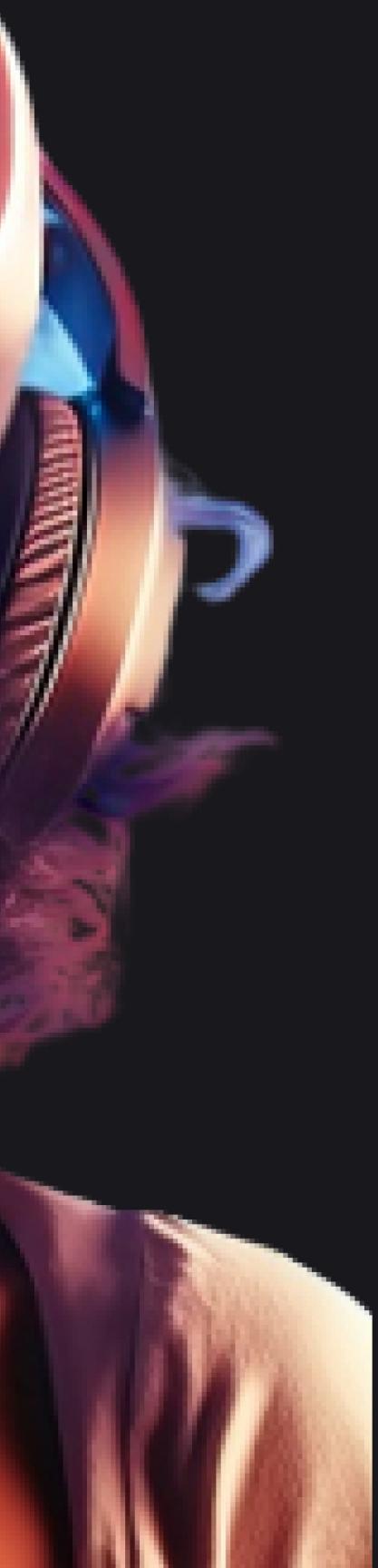
Of Django Application



#pythonworld



Introduction...



Django's admindocs app pulls documentation from the docstrings of **models**, **views**, **template tags**, and **template filters** for any app in INSTALLED_APPS and makes that documentation available from the Django admin.

Add `admindocs` to `INSTALLED_APPS`

`admindocs` is a build-in app that comes with django, just need to add in `INSTALLED_APPS`



settings.py

```
INSTALLED_APPS = [  
  
    "django.contrib.admindocs",  
  
    "django.contrib.admin",  
    "django.contrib.auth",  
    "django.contrib.contenttypes",  
    "django.contrib.sessions",  
    "django.contrib.messages",  
    "django.contrib.staticfiles",  
]
```

Add this path

Open urls.py file and add this path to the urlpatterns



urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path("admin/doc/", include("django.contrib.admindocs.urls")),
    path("admin/", admin.site.urls),
]
```



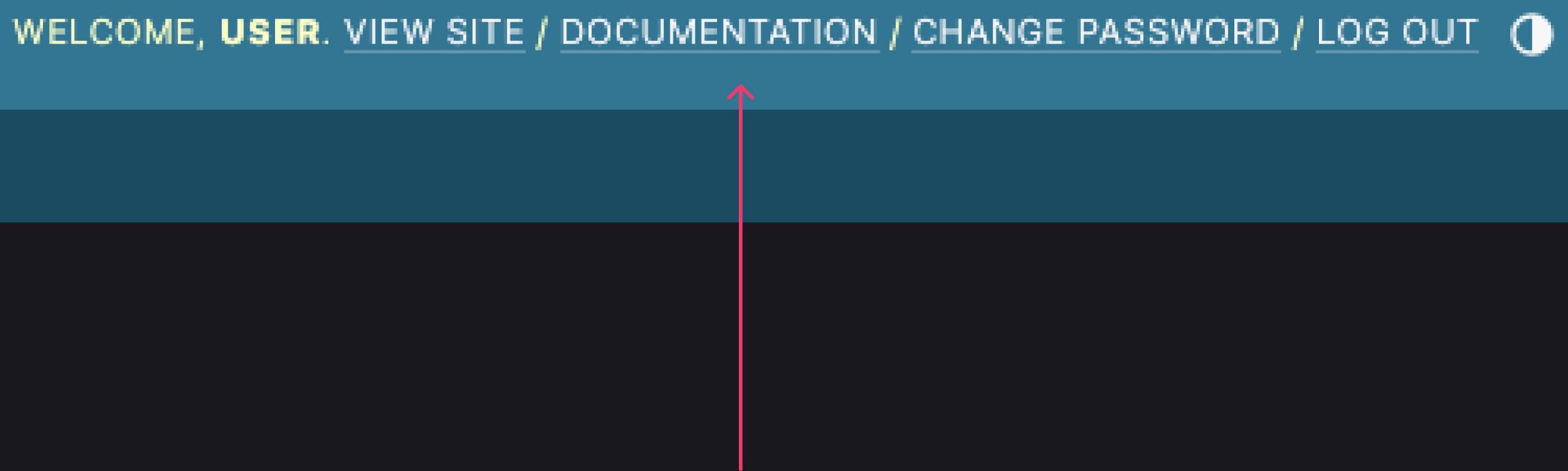
add this admin doc path
before admin path.

Install docutils

Docutils is a modular system for processing documentation into useful formats, such as HTML, XML, and LaTeX.

```
pip install docutils
```

Open django admin



Click on documentation, it will pulls documentation from the docstrings of models, views, template tags, and template filters

Write model docs.

You can simply use **doc string** to write the docs for your defined models.



models.py

```
class BlogEntry(models.Model):
    """
    Stores a single blog entry, related to
    :model:`blog.Blog` and :model:`auth.User`.
    """

    models.SlugField()
    author = models.ForeignKey(
        User,
        models.SET_NULL,
        blank=True,
        null=True,
    )
    blog = models.ForeignKey(Blog, models.CASCADE)

    def publish(self):
        """Makes the blog entry live on the site."""
        ...
```

Write View docs.

You can simply use **doc string** to write the docs for your defined views functions.



views.py

```
from django.shortcuts import render
from myapp.models import MyModel

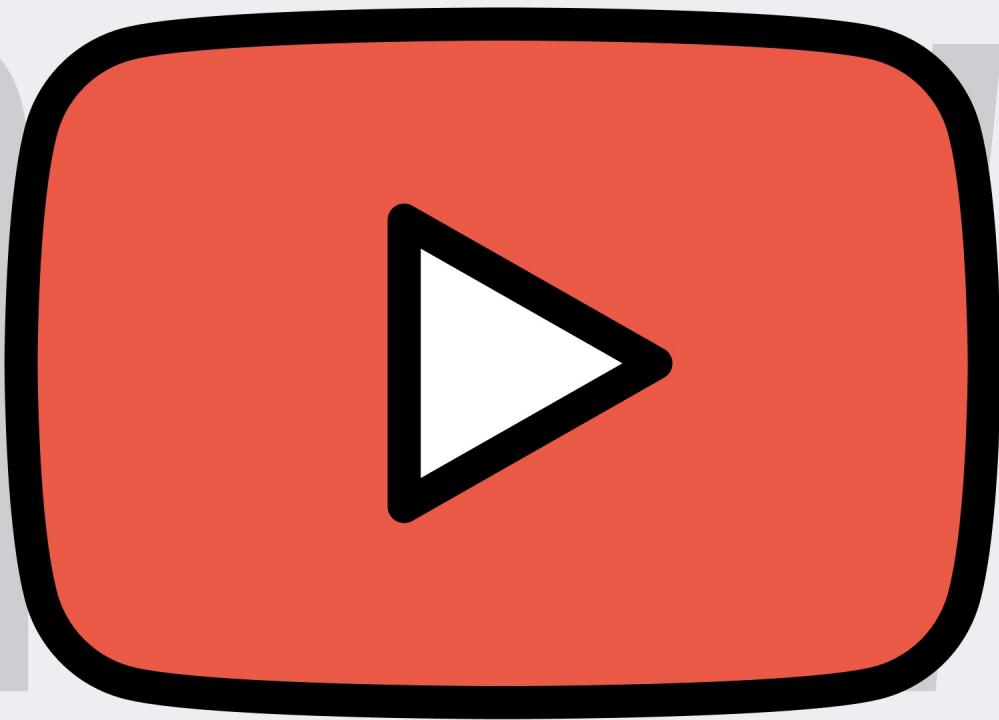
def my_view(request, slug):
    """
    Display an individual :model:`myapp.MyModel`.

    **Context**
    ``mymodel``
        An instance of :model:`myapp.MyModel`.

    **Template:**
    :template:`myapp/my_template.html`
    """
    context = {"mymodel": MyModel.objects.get(slug=slug)}
    return render(request, "myapp/my_template.html", context)
```

For more django related content

Subscribe To My Youtube Channel



Link in bio

DID YOU FIND THIS HELPFUL



Follow for more

#pythonworld