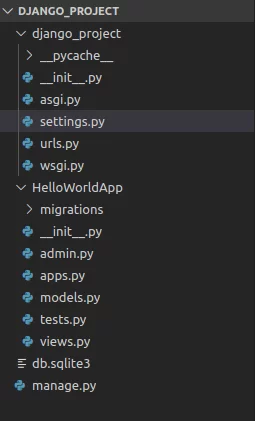
<https://www.askpython.com/django/django-app-structure-project-structure>

* Manage.py. ...
* \_init\_.py. ...
* setting.py. ...
* urls.py. ...
* wsgi.py. ...
* asgi.py. ...
* APPs.



A settings file is just a Python module with module-level variables.

When you use Django, you have to tell it which settings you’re using. Do this by using an environment variable, [**DJANGO\_SETTINGS\_MODULE**](https://docs.djangoproject.com/en/3.2/topics/settings/#envvar-DJANGO_SETTINGS_MODULE).

or

django-admin runserver --settings=mysite.settings

In your Django apps, use settings by importing the object **django.conf.settings**

Note that **django.conf.settings** isn’t a module – it’s an object. So importing individual settings is not possible:

**from** **django.conf.settings** **import** DEBUG *# This won't work.*

You shouldn’t alter settings in your applications at runtime

settings.DEBUG = **True** *# Don't do this!*

only:

**from** **django.conf** **import** settings

settings.configure(DEBUG=**True**)

**Either configure() or**[**DJANGO\_SETTINGS\_MODULE**](https://docs.djangoproject.com/en/3.2/topics/settings/#envvar-DJANGO_SETTINGS_MODULE)**is required**

**import** **django**

**from** **django.conf** **import** settings

**from** **myapp** **import** myapp\_defaults

settings.configure(default\_settings=myapp\_defaults, DEBUG=**True**)

django.setup()

*# Now this script or any imported module can use any part of Django it needs.*

**from** **myapp** **import** models

<https://docs.djangoproject.com/en/3.2/ref/django-admin/>

In addition, **manage.py** is automatically created in each Django project. It does the same thing as **django-admin** but also sets the [**DJANGO\_SETTINGS\_MODULE**](https://docs.djangoproject.com/en/3.2/topics/settings/#envvar-DJANGO_SETTINGS_MODULE) environment variable so that it points to your project’s **settings.py** file.

**check**

**dbshell**

**diffsettings**

**dumpdata**[**¶**](https://docs.djangoproject.com/en/3.2/ref/django-admin/#dumpdata)

**loaddata**

**makemigrations**

**migrate**

**runserver**

**sendtestemail**

**shell**

**startapp**

**startproject**

**test**

**testserver**[**¶**](https://docs.djangoproject.com/en/3.2/ref/django-admin/#testserver)

django-admin testserver mydata.json

**changepassword**

**createsuperuser**

**collectstatic**

**findstatic**

django-admin migrate --traceback

django-admin migrate --verbosity 2

<https://djbook.ru/rel1.9/topics/forms/index.html>

class form

csrf

widget

is\_valid

form.as\_...

**Рендеринг полей вручную**

<https://djbook.ru/rel1.9/ref/contrib/admin/index.html>

admin**.**site**.**register(Author)

**Декоратор для регистрации**

register**(***\*models***,***site=django.admin.sites.site***)**[**¶**](https://djbook.ru/rel1.9/ref/contrib/admin/index.html#django.contrib.admin.register)

ModelAdmin.empty\_value\_display

ModelAdmin.exclude

**from** django.contrib **import** admin

**class** **AuthorAdmin**(admin**.**ModelAdmin):

fields **=** ('name', 'title')

**class** **AuthorAdmin**(admin**.**ModelAdmin):

exclude **=** ('birth\_date',)

**from** django.contrib **import** admin

**class** **FlatPageAdmin**(admin**.**ModelAdmin):

fieldsets **=** (

(None, {

'fields': ('url', 'title', 'content', 'sites')

}),

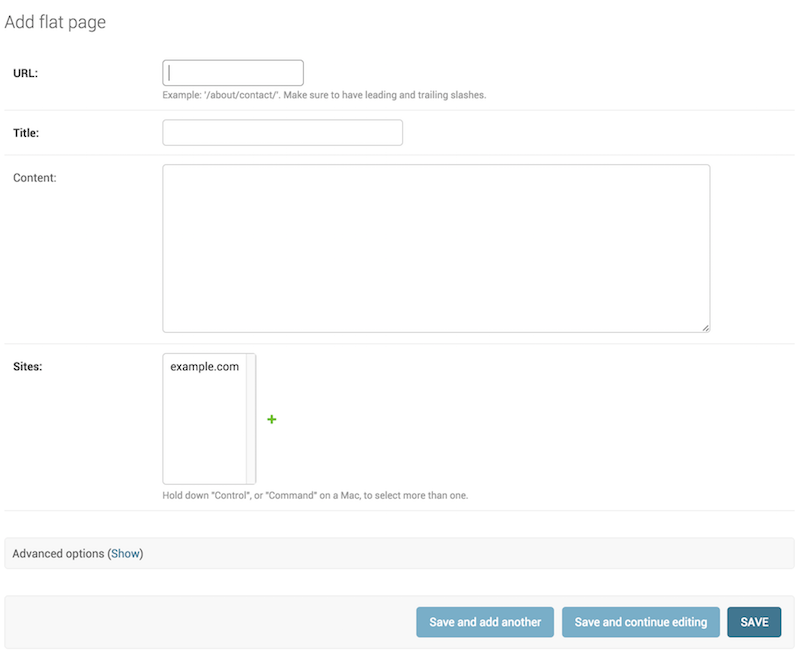
('Advanced options', {

'classes': ('collapse',),

'fields': ('registration\_required', 'template\_name'),

}),

)



ModelAdmin.list\_display

list\_display указывает какие поля отображать на странице списка объектов.

ModelAdmin.list\_filter

Укажите list\_filter, чтобы определить фильтры данных в правой панели страницы списка объектов