

# Django Cheat Sheets

Quick references for commands, files, common snippets, DRF, testing, and deployment.

## CLI & Management

```
django-admin startproject proj
python manage.py startapp app
python manage.py runserver
python manage.py makemigrations && python manage.py migrate
python manage.py createsuperuser
python manage.py shell
```

## File Map

Project: settings.py, urls.py, asgi.py/wsgi.py  
App: models.py, views.py, urls.py, templates/, static/, admin.py, forms.py

## Views

```
from django.shortcuts import render, redirect, get_object_or_404
from django.http import HttpResponseRedirect, JsonResponse
from django.contrib.auth.decorators import login_required

@login_required
def hello(request):
    return HttpResponseRedirect("Hi!")
```

## URLs

```
from django.urls import path, include
from . import views
urlpatterns = [
    path('', views.index, name='index'),
]
```

## Templates

```
{% extends 'base.html' %}
{% block content %}
    <h1>{{ title }}</h1>
    <a href="{% url 'index' %}">Home</a>
    
{% endblock %}
```

## ORM

```
# Create
Book.objects.create(title='TDD', author='Beck', owner=request.user)
# Read
Book.objects.filter(owner=request.user, status='read').order_by('-created_at')
# Update
book = Book.objects.get(pk=1, owner=request.user)
book.status = 'reading'
book.save()
# Delete
Book.objects.filter(owner=request.user, status='to_read').delete()
```

## Forms

```
from django import forms
class MyForm(forms.Form):
    name = forms.CharField(max_length=100)
```

## Admin & Auth

```
from django.contrib import admin
from django.contrib.auth.mixins import LoginRequiredMixin
from django.contrib.auth.decorators import login_required
```

## Settings

```
INSTALLED_APPS += ['myapp']
TEMPLATES[0]['APP_DIRS'] = True
STATIC_URL = 'static/'
LOGIN_URL = '/accounts/login/'
LOGIN_REDIRECT_URL = '/'
```

## DRF Kickstart

```
# pip install djangorestframework
INSTALLED_APPS += ['rest_framework']

from rest_framework import serializers, viewsets, routers
from django.urls import path, include

class TaskSerializer(serializers.ModelSerializer):
    class Meta:
        model = Task
        fields = '__all__'

class TaskViewSet(viewsets.ModelViewSet):
    queryset = Task.objects.all()
    serializer_class = TaskSerializer

router = routers.DefaultRouter()
router.register(r'tasks', TaskViewSet)

urlpatterns = [path('api/', include(router.urls))]
```

## Testing (pytest)

```
# pip install pytest pytest-django
def test_homepage(client, django_user_model):
    user = django_user_model.objects.create_user('a', password='x')
    client.force_login(user)
    resp = client.get('/')
    assert resp.status_code == 200
```

## Deployment (Basics)

- Use environment variables for secrets (DJANGO\_SECRET\_KEY, DATABASE\_URL).
- Collect static: python manage.py collectstatic.
- Run Gunicorn/Uvicorn behind Nginx.
- Prefer Postgres in production.