

Django REST Framework API Structure

DRF-focused structure for pure JSON APIs. No HTML templates.

#django #python #api #rest #drf #backend

 PNG

 PDF

 Copy

 Prompt

Project Directory

myproject/

manage.py

> config/ Project configur...

__init__.py

> settings/

__init__.py

base.py

local.py

production.py

urls.py API root + vers...

wsgi.py

asgi.py

> apps/

> users/ Auth & user man...

__init__.py

admin.py

apps.py

models.py

serializers.py DRF serializers

views.py ViewSets here

urls.py

permissions.py Custom permissi...

> migrations/

__init__.py

> tests/

__init__.py

test_api.py API endpoint te...

> products/ Example resource

__init__.py

admin.py

apps.py

models.py

serializers.py

views.py

urls.py

filters.py django-filter i...

pagination.py Custom paginati...

> migrations/

__init__.py

> tests/

__init__.py

test_api.py

factories.py Test data facto...

__init__.py

> api/ API versioning ...

__init__.py

> v1/ Version 1 endpo...

__init__.py

urls.py Aggregates all ...

schema.py OpenAPI customi...

> core/ Shared utilities

__init__.py

exceptions.py Custom API exce...

pagination.py Default paginat...

permissions.py Shared permissi...

throttling.py Rate limiting c...

renderers.py Custom JSON ren...

> requirements/

base.txt

local.txt

production.txt

.env.example

.gitignore

pytest.ini

Why This Structure?

This structure is optimized for pure JSON APIs. No templates folder, no static files. The `api/` folder provides versioning, while each app contains its own serializers, views, and permissions. The `core/` folder holds shared DRF utilities.

Key Directories

api/v1/ - Version 1 URL aggregation and schema config

apps/{app}/serializers.py - DRF serializers per app

apps/{app}/views.py - ViewSets and APIViews

core/ - Shared pagination, permissions, throttling

API Versioning Strategy

URLs are versioned via path prefix (`/api/v1/`). When v2 is needed, create `api/v2/` with its own `urls.py` . Apps can export multiple serializer versions (`UserSerializerV1` , `UserSerializerV2`) consumed by different API versions.

Getting Started

- `pip install -r requirements/local.txt`
- `cp .env.example .env`
- `python manage.py migrate`
- `python manage.py createsuperuser`
- `python manage.py runserver`
- Visit `/api/v1/docs/` for Swagger UI

App File Pattern

models.py - Django models only

serializers.py - All DRF serializers for this resource

views.py - ViewSets or APIViews

filters.py - django-filter FilterSets

permissions.py - Resource-specific permissions

When To Use This

- Backend for SPA (React, Vue, etc.)
- Mobile app backends
- Microservice APIs
- Third-party integrations
- Headless CMS backends

Recommended DRF Settings

- `DEFAULT_AUTHENTICATION_CLASSES` : TokenAuthentication, SessionAuthentication
- `DEFAULT_PERMISSION_CLASSES` : IsAuthenticated
- `DEFAULT_PAGINATION_CLASS` : `core.pagination.StandardPagination`
- `DEFAULT_FILTER_BACKENDS` : DjangoFilterBackend, SearchFilter, OrderingFilter
- `DEFAULT_THROTTLE_RATES` : `{"anon": "100/hour", "user": "1000/hour"}`

Trade-offs

No admin styling - Admin works but no custom templates

Serializer duplication - Different serializers for list/detail/create

Testing overhead - API tests require more setup than view tests