

# Django. DRF.

То, что нужно знать о Django и DRF, чтобы начать их правильно применять.

# Django

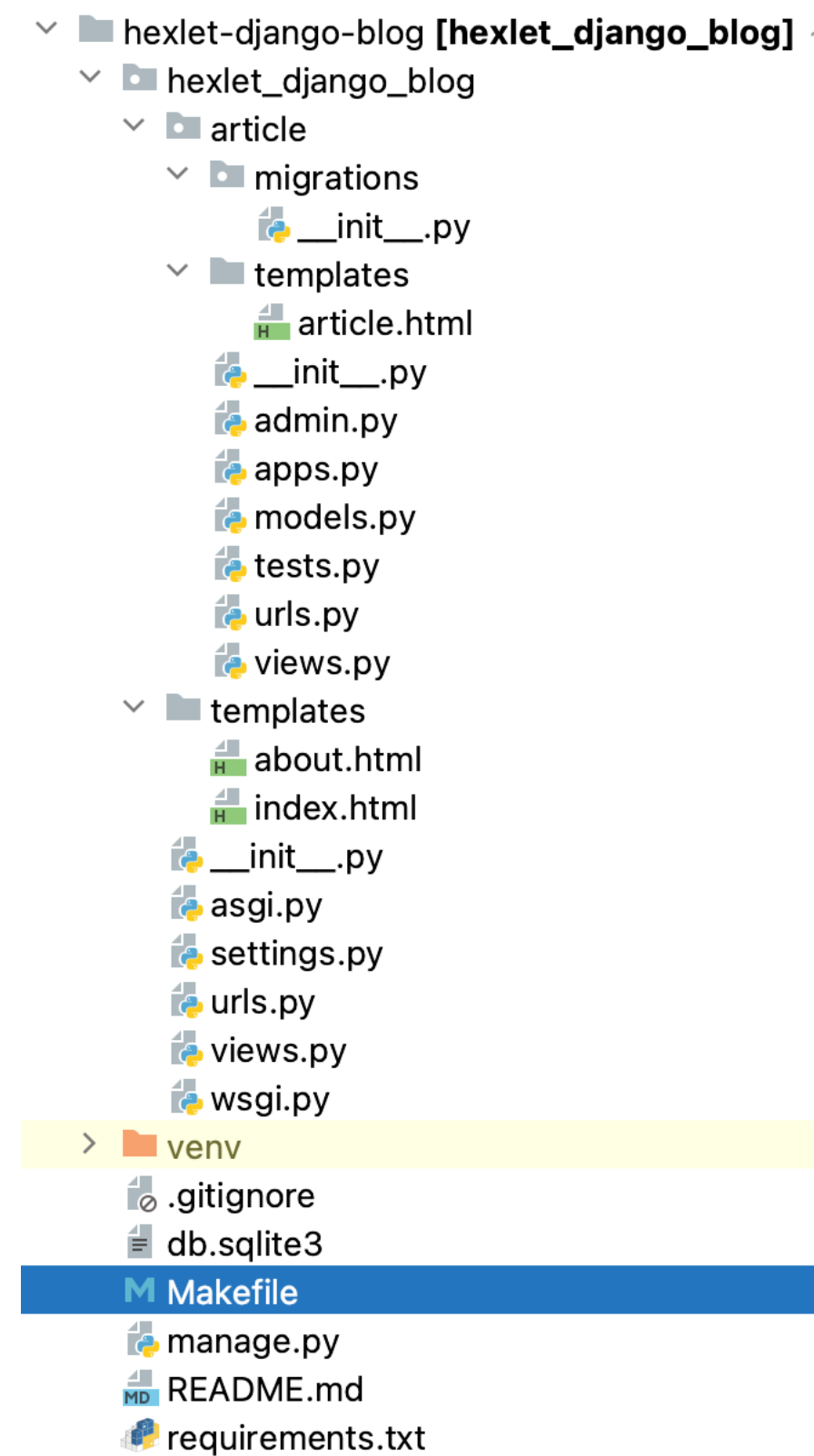
# Настройка и управление

## Работа с проектом на удобном уровне абстракции

- Настройка и управление проектом в django осуществляется посредством работы с `manage.py`

# Структура приложения

Что генерируется в приложении и как мы это используем



# Запросы

## Как мы принимаем запросы к системе

- Views - нужны для представлений и навешивания вызова функций на урлы
- Urls - нужны для менеджмента углов

# Приложения

## Декомпозиция проекта - зачем и как

- Приложения нужны для выделения зон ответственности

# Шаблонизация

## Шаблоны в Django и примеры применения

- Шаблоны нужны для работы с отображением страниц
- В частности наиболее частая задача в системах с подходом REST - работа с админ панелями

# 12 факторов

**Что это за идея**

- <https://12factor.net/ru/>



# Конфигурация

## Как управлять окружением

```
1  from dotenv import load_dotenv
2  import os
3
4  load_dotenv()
5
6  print(os.getenv('KEY'))
```

# Django Rest Framework

# APIView

## Представления

```
from rest_framework.views import APIView
from rest_framework.response import Response
from rest_framework import authentication, permissions
from django.contrib.auth.models import User

class ListUsers(APIView):
    """
    View to list all users in the system.

    * Requires token authentication.
    * Only admin users are able to access this view.
    """
    authentication_classes = [authentication.TokenAuthentication]
    permission_classes = [permissions.IsAdminUser]

    def get(self, request, format=None):
        """
        Return a list of all users.
        """
        usernames = [user.username for user in User.objects.all()]
        return Response(usernames)
```

# APIViewSet

## Аккумуляции представлений

```
from django.contrib.auth.models import User
from django.shortcuts import get_object_or_404
from myapps.serializers import UserSerializer
from rest_framework import viewsets
from rest_framework.response import Response

class UserViewSet(viewsets.ViewSet):
    """
    A simple ViewSet for listing or retrieving users.
    """
    def list(self, request):
        queryset = User.objects.all()
        serializer = UserSerializer(queryset, many=True)
        return Response(serializer.data)

    def retrieve(self, request, pk=None):
        queryset = User.objects.all()
        user = get_object_or_404(queryset, pk=pk)
        serializer = UserSerializer(user)
        return Response(serializer.data)
```