pip3 install virtualenvwrapper-win

mkvirtualenv my\_django\_environment

* deactivate — 退出當前的 Python 虛擬環境
* workon — 列出可用的虛擬環境
* workon name\_of\_environment — 激活指定的 Python 虛擬環境
* rmvirtualenv name\_of\_environment — 刪除指定的環境

1. 设置 Django 后端

首先，你需要安装 Django 并创建一个新的 Django 项目。假设你已经安装了 Python 和 pip。

bashCopy code

# 安装 Django pip install django

安裝django提示包

pip install django-stubs

# 创建一个新的 Django 项目 django-admin startproject backend

# 进入项目目录 cd backend

# 创建一个新的应用（例如叫做 api） python manage.py startapp api

接下来，配置 Django 以允许跨域资源共享（CORS），这对于前后端分离的应用是必要的。

# 安装 django-cors-headers pip install django-cors-headers

# 将 'corsheaders' 添加到你的 INSTALLED\_APPS 设置中

# 将 'corsheaders.middleware.CorsMiddleware' 添加到你的 MIDDLEWARE 设置中

# 添加一个 CORS\_ALLOW\_ALL\_ORIGINS 设置并设为 True（生产环境中可能需要更严格的设置）

编辑 backend/settings.py，进行相应配置。

2. 设置 React 前端

接下来，安装 Node.js 和 npm（如果你还没有安装的话），然后创建一个新的 React 应用。

# 安装 create-react-app npm install -g create-react-app

# 创建一个新的 React 应用 npx create-react-app frontend

# 进入项目目录 cd frontend

# 启动开发服务器 npm start

3. 整合 Django 和 React

开发环境：在开发环境中，你通常会分别运行 Django 和 React 的开发服务器。Django 服务 API 请求，而 React 从开发服务器提供 UI。

生产环境：在生产环境中，你会构建 React 应用，然后将构建产物（通常在 build 目录中）部署为 Django 的静态文件。

在 React 应用中运行 npm run build 生成静态文件。

将构建产物（React 应用的 build 目录）移动到 Django 项目中，可以通过配置 Django 的 STATICFILES\_DIRS 和 WHITENOISE 来提供服务。

4. 配置 Nginx 作为反向代理

在生产环境中，Nginx 会作为反向代理来服务静态文件，并将 API 请求转发给 Django。这部分配置会在将你的应用部署到服务器时进行。

这个流程提供了一个基础的起点。根据项目的具体需求，你可能需要进行额外的配置和开发，比如设置数据库、用户认证、环境变量管理等。

如果你对某个特定步骤需要更详细的指导或示例代码，请告诉我，我可以提供更具体的帮助。