

# Django

xbZhong

2025-01-29

[本页PDF](#)

## 项目结构

1. `manage.py`: 与项目交互的文件，一般在终端输入 `python manage.py [子命令]`
2. `settings.py`: 项目的设置项
3. `urls.py`: 配置url路径
4. `wsgi.py`: 项目与WSGI协议兼容的web服务器入口

## 入门

- Django中定义函数时 `request` 是一个必加的参数，它包含了向服务器发送请求的请求体的所有信息
- 给请求体返回数据时要导入 `HttpResponse` 对象
- `app` 是项目的一个模块，各种各样的 `app` 组成了项目
- `urlpatterns` 位于项目的 `urls.py` 当中，可以将 `url` 与视图函数（也就是访问这个url要执行的函数）进行绑定

```
from django.contrib import admin
from django.http import HttpResponse
from django.urls import path
## 视图函数
def welcome(request):
    return HttpResponse("Welcome to DjangoProject")
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', welcome)
]
```

- 启动: `python manage.py runserver`
- 创建app: `python manage.py startapp [app名字]`

## URL携带参数

- 从请求信息中解析参数并获取
  - 使用 `request.GET.get` 解析参数，如解析 `http://127.0.0.1:8000/app1?id=3`，会得到3这个数字
  - 直接在网址中解析，如 `http://127.0.0.1:8000/app1/水浒传`，使用 `/app1/<book_name>` 的形式作为path，调用视图函数时会进行参数传递，只需要在视图函数中进行输出即可
    - `/app1/<book_name>` 中可以对 `book_name` 进行类型定义，如 `/app1/<int:book_id>`

```

## app1/views.py
from django.http import HttpResponseRedirect
from django.shortcuts import render

def book_id(request):
    id = request.GET.get('id')
    return HttpResponseRedirect(f'您查找的图书id是{id}')

def book_name(request,book_name):
    return HttpResponseRedirect(f'您查找的图书名称为{book_name}')


## urls.py
from django.contrib import admin
from django.http import HttpResponseRedirect
from django.urls import path
## 导入app1的views文件，将网址和视图函数进行映射
from app1 import views
def welcome(request):
    return HttpResponseRedirect("Welcome to DjangoProject")
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', welcome),
    # http://127.0.0.1:8000/app1?id=3
    path('app1/',views.book_id),
    # http://127.0.0.1:8000/app1/1
    path('app1/<book_name>',views.book_name)
]

```

## path函数

- 定义为 `path(route,view,name=None,kwarg=None)`
  - `route`参数：这个参数可以指定url中需要传递的参数，从而在数据库中获取需要的信息。
  - `view`参数：可以是一个视图函数或者时类视图，或者是函数返回值、
  - `name`参数：给这个url取个别名

## 路由模块化

- 使用 `include`方法在项目的主 `urls` 导入app的 `urls.py` 文件，便于统一管理

```

# app2/urls.py 路由文件
from django.urls import path
from . import views
urlpatterns = [
    path('list/', views.movie_list, name='movie_list'),
    path('detail/<int:movie_id>', views.movie_detail, name='movie_detail')
]

# app2/views.py 视图函数文件
from django.http import HttpResponse
from django.shortcuts import render

def movie_list(request):
    return HttpResponse('电影列表')

def movie_detail(request, movie_id):
    return HttpResponse(f'您查找的电影id为: {movie_id}')

# urls.py
from django.contrib import admin
from django.http import HttpResponse
from django.urls import path, include
from app1 import views
import app2

urlpatterns = [
    # app2
    path('app2/', include("app2.urls"))
]

```

## 路由反转

- 使用 `reverse` 通过视图函数得到 `url`
  - 里面传递的参数是视图函数的别名和 `url` 要传递的参数

```
def welcome(request):
    print(reverse('book_id_search'))
    print(reverse('book_name_search',kwargs={'book_name':'sanguoyanyi'}))
    return HttpResponse("Welcome to DjangoProject")

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', welcome),
    # http://127.0.0.1:8000/app1?id=3
    path('app1/',views.book_id,name='book_id_search'),
    # http://127.0.0.1:8000/app1/1
    path('app1/<book_name>',views.book_name,name='book_name_search'),
    # app2
    path('app2/',include("app2.urls"))
]
```

## DTL (模板渲染)

---