django路由系统

Ringo Yungpo Kao

December 29, 2017

# django路由系统

## django简介

在Django的urls中我们可以根据一个URL对应一个函数名来定义路由规则如下：

每一个urls对应一个views里的函数 ## 基本的urls对应 {#基本的urls对应 }

urlpatterns = [  
url(r'^login/$', views.login),  
url(r'^index/$', views.index),  
url(r'^$', views.login),  
]

## 基于app的路由

根据app对路由规则进行一次分类

当app的urls很多的时候，那么就不能再工程的urls下面去设置 应该这样设置：

1. 首先在helloword下面的urls这样设置

#导入include  
from django.conf.urls import url,include  
from django.contrib import admin  
#导入app下的view函数  
from helloapp import views  
urlpatterns = [  
 url(r'^admin/', admin.site.urls),  
 #使用helloapp的urls规则  
 url(r'^helloapp/',include('helloapp/urls'))  
   
]

1. 在helloapp下面创建一个urls.py

内容：

from django.conf.urls import url,include  
from django.contrib import admin  
#导入app下的view函数  
from . import views  
urlpatterns = [  
 url(r'^admin/', admin.site.urls),  
 url(r'^login/',views.login),  
 url(r'^index/',views.index),  
 url(r'^reg/',views.reg),  
 url(r'^layout/',views.layout),  
]

1. 在helloapp下面的views里创建上面urls对应的函数

from django.shortcuts import render,HttpResponse  
# Create your views here.  
def index(request):  
 # return HttpResponse('<h1>hello world!!!</h1>')  
 return render(request,'index.html')  
def login(request):  
 return HttpResponse('login')  
def reg(request):  
 return HttpResponse('reg')  
def layout(request):  
 return HttpResponse('layout')

1. 访问：

http://127.0.0.1:8000/helloapp/admin/ admin后台管理  
http://127.0.0.1:8000/helloapp/layout/  
http://127.0.0.1:8000/helloapp/login/  
http://127.0.0.1:8000/helloapp/reg/  
http://127.0.0.1:8000/helloapp/index/

## 动态路由（传一个参数）

比如分页：当urls大量过多的时候比如几百个的时候，那么肯定不会去写几百个路由规则 所有这个时候就需要动态urls，使用正则表达式来完成

1. 在helloapp下面的urls写入以下内容：

from django.conf.urls import url,include  
from django.contrib import admin  
#导入app下的view函数  
from . import views  
urlpatterns = [  
 url(r'^admin/', admin.site.urls),  
 url(r'^book/(\d+)$', views.book), //正则匹配  
]

1. 在helloapp下面的views写入以下内容：

from django.shortcuts import render,HttpResponse  
# Create your views here.  
def book(request,num):  
 print(num)  
 return HttpResponse(num)

当用户访问http://127.0.0.1:8000/helloapp/book/数字的时候 django会在自动把参数传给views里的book函数

## 动态路由（传多个参数）

多个参数它是已/来分割的

来一个url的加法 1. 在helloapp下面的urls写入以下内容：

from django.conf.urls import url,include  
from django.contrib import admin  
#导入app下的view函数  
from . import views  
urlpatterns = [  
 url(r'^admin/', admin.site.urls),  
 url(r'^book/(\d+)/(\d+)$', views.book),  
]

1. 在helloapp下面的views写入以下内容：

def book(request,num1,num2):  
 print(num1,num2)  
 num = int(num1) + int(num2)  
 return HttpResponse(num)

他的顺序是：正序的，你先给他传那个值，第一个参数就是那个

## 动态的路由（Key:value的形式）

1. 在helloapp下面的urls写入以下内容：

from django.conf.urls import url,include  
from django.contrib import admin  
#导入app下的view函数  
from . import views  
urlpatterns = [  
 url(r'^admin/', admin.site.urls),  
 url(r'^book/(?P<k1>\d+)/(?P<k2>\d+)$', views.book),  
 这里?p<v1>这里的v1就是key，vlaue就是传进去的值,  
]

1. 在helloapp下面的views写入以下内容：

from django.shortcuts import render,HttpResponse  
# Create your views here.  
def book(request,k1,k2):  
 print(k1,k2)  
 return HttpResponse(k1+k2)

这样我们就不必按照顺序去取了，可以通过key，value的方式来取传进来的值

注：可以根据传来的值来进行判断返回给用户指定的url，过程略。

## 基于反射的动态路由

仅仅是通过反射来实现的，通过文件找到里面的函数然后执行！ 但是在Django中不建议使用此方法。因为不同的WEB框架建议你使用不同的方式， Django就不建议使用反射

## 总结

简而言之，django的路由系统作用就是使views里面处理数据的函数与请求的url建立映射关系。使请求到来之后，根据urls.py里的关系条目，去查找到与请求对应的处理方法，从而返回给客户端http页面数据

django 项目中的url规则定义放在project 的urls.py目录下， 默认如下：

from django.conf.urls import url  
from django.contrib import admin  
   
   
urlpatterns = [  
 url(r'^admin/', admin.site.urls),  
]

url()函数可以传递4个参数，其中2个是必须的：regex和view，以及2个可选的参数：kwargs和name。下面是具体的解释： - regex： regex是正则表达式的通用缩写，它是一种匹配字符串或url地址的语法。Django拿着用户请求的url地址，在urls.py文件中对urlpatterns列表中的每一项条目从头开始进行逐一对比，一旦遇到匹配项，立即执行该条目映射的视图函数或二级路由，其后的条目将不再继续匹配。因此，url路由的编写顺序至关重要！

需要注意的是，regex不会去匹配GET或POST参数或域名，例如对于https://www.example.com/myapp/，regex只尝试匹配myapp/。对于https://www.example.com/myapp/?page=3,regex也只尝试匹配myapp/。

如果你想深入研究正则表达式，可以读一些相关的书籍或专论，但是在Django的实践中，你不需要多高深的正则表达式知识。

性能注释：正则表达式会进行预先编译当URLconf模块加载的时候，因此它的匹配搜索速度非常快，你通常感觉不到。 - view： 当正则表达式匹配到某个条目时，自动将封装的HttpRequest对象作为第一个参数，正则表达式“捕获”到的值作为第二个参数，传递给该条目指定的视图。如果是简单捕获，那么捕获值将作为一个位置参数进行传递，如果是命名捕获，那么将作为关键字参数进行传递。

* kwargs： 任意数量的关键字参数可以作为一个字典传递给目标视图。
* name 对你的URL进行命名，可以让你能够在Django的任意处，尤其是模板内显式地引用它。相当于给URL取了个全局变量名，你只需要修改这个全局变量的值，在整个Django中引用它的地方也将同样获得改变。这是极为古老、朴素和有用的设计思想，而且这种思想无处不在。