Handling Multiple Applications in Django (urls.py Fix)

# 1. Problem Explanation

You created multiple apps (e.g., app1 and app2) and tried to import views into the project's urls.py file. Your current code:

from django.contrib import admin  
from django.urls import path  
# from app1 import views as ap1  
# from app2 import views as ap2  
  
from app1.views import home, myapp1  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('', ap1.home, name="home"), # ❌ ap1 not defined  
 path('myapp1/', ap1.myapp1, name="myapp1"),  
 path('myapp2/', ap2.myapp2, name="myapp2"),  
 path('myapp2\_me/', ap2.myapp2\_me, name="myapp2\_me"),  
]

Issue:  
- You imported only specific views from app1, but in your URL patterns you are calling ap1 and ap2 which are not imported (commented out).  
- This leads to a NameError.

# 2. Two Correct Ways to Fix the Import Issue

## Method 1: Import the whole views module with alias

from django.contrib import admin  
from django.urls import path  
from app1 import views as ap1 # Import app1 views with alias  
from app2 import views as ap2 # Import app2 views with alias  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('', ap1.home, name="home"), # Use alias for app1 view  
 path('myapp1/', ap1.myapp1, name="myapp1"),  
 path('myapp2/', ap2.myapp2, name="myapp2"),  
 path('myapp2\_me/', ap2.myapp2\_me, name="myapp2\_me"),  
]

Why use this?  
- Useful when you have many functions and want to avoid repetitive imports.  
- The alias (ap1, ap2) helps identify which app the view belongs to.

## Method 2: Import specific functions from each app

from django.contrib import admin  
from django.urls import path  
from app1.views import home, myapp1  
from app2.views import myapp2, myapp2\_me  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('', home, name="home"), # Direct function call  
 path('myapp1/', myapp1, name="myapp1"),  
 path('myapp2/', myapp2, name="myapp2"),  
 path('myapp2\_me/', myapp2\_me, name="myapp2\_me"),  
]

Why use this?  
- Clear which functions are being used.  
- Less typing because you call the function directly without alias.

# 3. Which Method Should You Use?

Method 1 (with alias) is better if you have many views in the app and want to group them logically.  
  
Method 2 (import specific functions) is better if you want clarity and don't have too many views.

# 4. Best Practice: Use App-level urls.py

Instead of importing all views into the project-level urls.py, the best practice is to create a separate urls.py for each app and include them in the project’s urls.py.  
  
Example:

app1/urls.py:

from django.urls import path  
from . import views  
  
urlpatterns = [  
 path('', views.home, name="home"),  
 path('myapp1/', views.myapp1, name="myapp1"),  
]

app2/urls.py:

from django.urls import path  
from . import views  
  
urlpatterns = [  
 path('myapp2/', views.myapp2, name="myapp2"),  
 path('myapp2\_me/', views.myapp2\_me, name="myapp2\_me"),  
]

project-level urls.py:

from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('app1/', include('app1.urls')), # include app1 urls  
 path('app2/', include('app2.urls')), # include app2 urls  
]

# 5. Summary

1. Method 1: Import the whole views module with alias.  
 Call views as ap1.home, ap2.myapp2.  
  
2. Method 2: Import specific view functions directly.  
 Call views as home, myapp2.  
  
3. Best practice: Use separate urls.py files for each app and include them in the project-level urls.py.