• Setting up URL patterns in urls.py for routing requests to views.

**1. Basic URL Configuration**

First, let’s create the urls.py file for your app if it doesn’t already exist. Here’s how you can define URL patterns in your urls.py:

* python

# myapp/urls.py

from django.urls import path

from . import views

urlpatterns = [

path('', views.home, name='home'), # URL pattern for the home view

path('about/', views.about, name='about'), # URL pattern for the about view

]

**2. Include App URLs in Project URLs**

Next, you need to include your app’s URLs in the project’s main urls.py file. This file is usually found in the project’s root directory.

* python

# myproject/urls.py

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls), # URL pattern for the admin site

path('', include('myapp.urls')), # Include the app’s URLs

]

**3. Creating Views**

Make sure you have corresponding views defined in your views.py file for the URLs you set up.

* python

# myapp/views.py

from django.shortcuts import render

def home(request):

return render(request, 'home.html') # Render the home template

def about(request):

return render(request, 'about.html') # Render the about template

**4. Creating Templates**

You should also have templates corresponding to the views you defined. Place these templates in your app’s templates directory.

**home.html**:

html

<!DOCTYPE html>

<html>

<head>

<title>Home</title>

</head>

<body>

<h1>Welcome to the Home Page!</h1>

</body>

</html>

**about.html**:

html

<!DOCTYPE html>

<html>

<head>

<title>About</title>

</head>

<body>

<h1>About Us</h1>

</body>

</html>

• Integrating templates with views to render dynamic HTML content.

• Integrating templates with views to render dynamic HTML content.

Integrating templates with views in Django allows you to render dynamic HTML content based on the data from your models. Here’s how you can achieve this step-by-step:

**1. Define Your Models**

First, define your models in models.py to represent the data you want to render dynamically.

* python

# myapp/models.py

from django.db import models

class Car(models.Model):

make = models.CharField(max\_length=50)

model = models.CharField(max\_length=50)

year = models.IntegerField()

def \_\_str\_\_(self):

return f"{self.make} {self.model} ({self.year})"

**2. Create Views to Handle Requests**

Next, create views in views.py that will fetch the necessary data from the models and pass it to the templates.

* python

# myapp/views.py

from django.shortcuts import render

from .models import Car

def car\_list(request):

cars = Car.objects.all()

return render(request, 'car\_list.html', {'cars': cars})

**3. Define URL Patterns**

Set up URL patterns in urls.py to route requests to the appropriate views.

* python

# myapp/urls.py

from django.urls import path

from . import views

urlpatterns = [

path('cars/', views.car\_list, name='car\_list'),

]

**4. Create Templates**

Create HTML templates in the templates directory to display the data. Use Django's templating language to insert dynamic content.

* html

<!-- myapp/templates/car\_list.html -->

<!DOCTYPE html>

<html>

<head>

<title>Car List</title>

</head>

<body>

<h1>Available Cars</h1>

<ul>

{% for car in cars %}

<li>{{ car.make }} {{ car.model }} ({{ car.year }})</li>

{% endfor %}

</ul>

</body>

</html>

**5. Configure Settings**

Ensure your project's settings are configured to locate your templates. In settings.py, make sure the TEMPLATES setting is properly configured.

* python

# myproject/settings.py

TEMPLATES = [

{

...

'DIRS': [os.path.join(BASE\_DIR, 'templates')],

...

},

]

**6. Run the Development Server**

Start the Django development server to see your dynamic content in action.

python manage.py runserver