Django Web Framework

DJANGO TEMPLATES AND TEMPLATE VARIABLES

Django Templates-

Till now we are directly typing HTML code inside the views.py python file. But it is not the correct or recommended way to write the HTML code. As it reduces the readability of the code, the backend python developer might confuse between the frontend HTML code and backend code and minor changes will take a lot of time.

To overcome this problem we have a powerful Django Feature: **Templates.**

Django templates are a key part of the Model-View-Template (MVT) architectural pattern used in Django web development. They define the structure and presentation of your web pages.

We can render templates from our views function. We have to write templates at project level only once and we can use these in multiple applications.

Django Template Language: While they primarily contain HTML, Django templates can also include special tags and filters. These provide logic and dynamic content within your templates. Django template language is primarily used for displaying dynamic content on webpage.

For example- When a user successfully add the employee details and clicks the submit button, if the backend logic is correct and the data saved to the database correctly, we can display content like “Data saved successfully”.

How to create a Template?

Step 1: Create a project named fourthproject_template1 and navigate to the project folder and then create a application

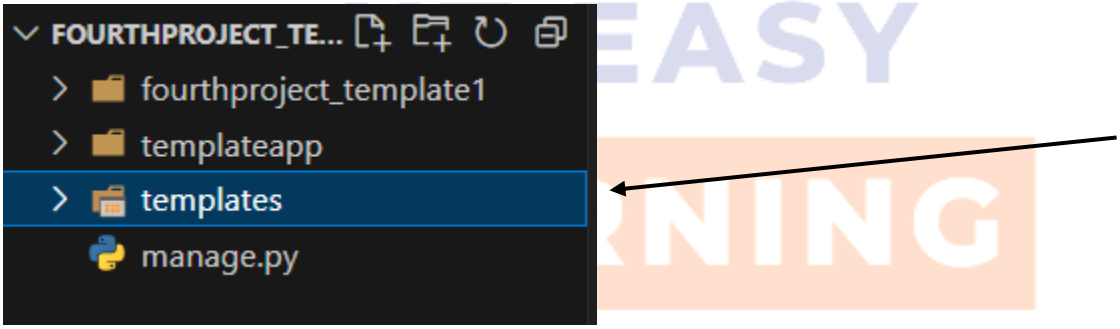
```
C:\Users\LENOVO\Documents\djangocourse>django-admin startproject fourthproject_template1

C:\Users\LENOVO\Documents\djangocourse>cd fourthproject_template1
```

```
C:\Users\LENOVO\Documents\djangocourse\fourthproject_template1>python manage.py startapp templateapp
```

Add the app in settings.py INSTALLED_APPS list.

NOTE: You can give any name of your choice for the project and application

Step 2: Create a folder named templates inside the root folder-



This folder will contain all our templates (HTML files) which are necessary and will be rendered as a response to the URL requests.

Step 3: Configure templates folder in Django settings.

Django templates are typically stored in a directory named **templates** within each application. The **TEMPLATES** setting in Django's settings file specifies the directories where Django looks for templates.

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

This list is where you'll add the template directory path

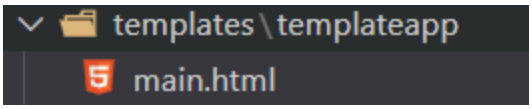
Add **'BASE_DIR , 'templates'** inside the **DIRS** list.

or

'DIRS': [BASE_DIR / 'templates']

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR,'templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

Step 4: Create HTML file inside templates folder. We are creating a folder inside templates folder with the same name as application name and inside that application folder we will create HTML file because a project may contain multiple applications and each application can contain multiple templates so keeping the templates of each application separate makes our project structure simple and easy to understand.



templates->templateapp->main.html

```
templates > templateapp > main.html > html
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>First Django Template Project</title>
7   </head>
8   <body>
9       <h1>Hello! This is my first template</h1>
10  </body>
11  </html>
```

TIP: To automatically get the HTML structure type '!' and then press tab key.

Step 6: Define function-based view for our template in views.py-

```
def display_view(request):
    return render(request,'templateapp/main.html')
```

The parameter (request) contains information about the request, such as the URL path, headers, cookies, and any data sent by the user (e.g., from a form submission).

Step 7: Define URL for our view in urls.py (application level as well as project level)

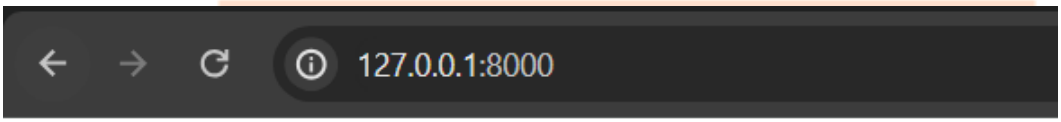
```
templateapp > urls.py > ...
1   from templateapp import views
2   from django.urls import path
3
4   urlpatterns=[
5       path('',views.display_view)
6   ]
```

```python
from django.contrib import admin
from django.urls import path,include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',include('templateapp.urls'))
]
```

Step 8: Start the development server:

python manage.py runserver

```
System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
 auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
May 18, 2024 - 16:28:22
Django version 5.0, using settings 'fourthproject_template1.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[18/May/2024 16:28:25] "GET / HTTP/1.1" 200 267
```



Hello! This is my first template

What is Template Variable, Template Tags and Template Filters?

Template Variables-

Variables are placeholders for data that will be replaced when the template is rendered. They are enclosed in double curly braces {{ }}.

By using template variable we can display dynamic content on our web page.

In the same project we are going to create a view to display data using template variable-

Practical 2:

Step 1: Create new HTML file and Update views.py and add a new view for template variable example-

templates > templateapp > displayname.html > html > body > h2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Template Variable</title>
</head>
<body>
    <h2>Hello! My name is {{name}}</h2>
</body>
</html>
```

templateapp > views.py > display_name_view

```
from django.shortcuts import render

# Create your views here.

def display_view(request):
    return render(request,'templateapp/main.html')


#Template variable example
def display_name_view(request):
    context={
        'name':'Harsh',
    }
    return render(request,'templateapp/displayname.html',context)
```

The context dictionary stores key-value pairs where the keys are variable names and the values are the data you want to pass to the template.

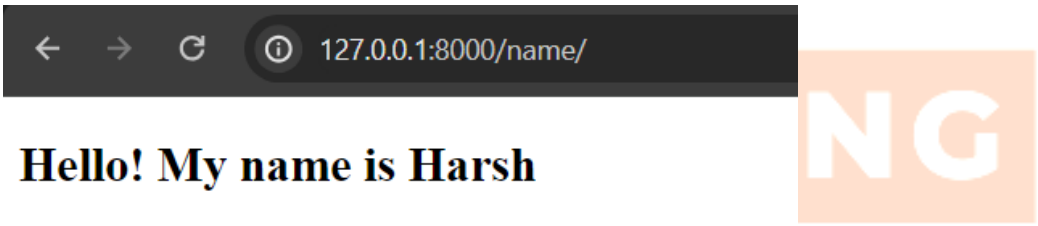
Step 2: Update application level urls.py and add URL pattern for our new view function-

templateapp > urls.py > ...

```python
from templateapp import views
from django.urls import path

urlpatterns=[
    path('',views.display_view),
    path('name/',views.display_name_view),
]
```

Step 3: Start the development server and visit the URL on browser-

```
[18/May/2024 17:39:59] "GET /name HTTP/1.1" 301 0
[18/May/2024 17:39:59] "GET /name/ HTTP/1.1" 200 246
```



Template variables are really important if we want to display dynamic content on our webpage. For example, the details of the user that is currently logged in to the system. Those details will be fetched from database and by using template variable we can display them on corresponding template. We will do that after learning models.

Practical 3: Let's display Name and Country-

Step 1: Create new HTML file-

templates > templateapp > show.html > html > body > h2 > br

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Details</title>
</head>
<body>
    <h2>
    Name: {{name}} <br/>
    Country: {{country}}
    </h2>
</body>
</html>
```

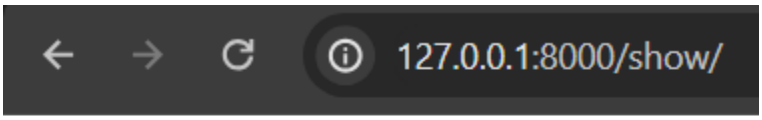
Step 2: Create a new view function-

```
def show_view(request):
    context={
        'name':'Harsh Trivedi',
        'country':'India',
    }
    return render(request,'templateapp/show.html',context)
```

Step 3: Update urls.py-

```
urlpatterns=[
    path('',views.display_view),
    path('name/',views.display_name_view),
    path('show/',views.show_view),
]
```

Step 4: Run the server and visit the URL-



Name: Harsh Trivedi
Country: India

Let's display current date and time also:

Step 1: Update the view and HTML file-

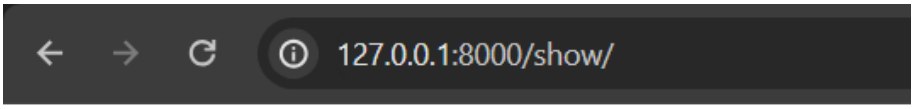
```python
def show_view(request):
    date=datetime.datetime.now
    context={
        'name':'Harsh Trivedi',
        'country':'India',
        'date':date,
    }
    return render(request,'templateapp/show.html',context)
```

templates > templateapp > show.html > html > body > h2

```
1   <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>User Details</title>
7   </head>
8  <body>
9      <h2>
10     Name: {{name}} <br/>
11     Country: {{country}} <br/>
12     Date and Time: {{date}}
13     </h2>
14  </body>
15  </html>
```

Step 2: Run the server and visit the URL-

```
System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
 auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
May 18, 2024 - 18:00:44
Django version 5.0, using settings 'fourthproject_template1.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[18/May/2024 18:01:07] "GET /show/ HTTP/1.1" 200 321
```



Name: Harsh Trivedi
Country: India
Date and Time: May 18, 2024, 6:01 p.m.

NOTE: The message you are seeing after starting the development server-

```
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
 auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
```

This message is related to database migrations. Till now we have not used database that is why applying migrations is not necessary. We will learn about migrations in detail when we start learning models.

HOMEWORK EXERCISE:

Display Student data like Name, College, City and Roll Number and date on a template using template variable.

Also see if we want to display subjects should we create multiple subject keys in context dictionary like subject 1, subject 2, etc.

We instead create a list of subjects like this- subjects = [‘DSA’,’Web development’,’Java’]. But how to display it on template?

We will see that in the next tutorial where we are going to learn about Template tags and filters.

If you are following the course series and learning the concepts easily, please subscribe to our YouTube channel. This will motivate us to create more educational and course videos and study material.

Join our growing community of tech enthusiasts! Subscribe to our YouTube channel, where we simplify programming languages in the easiest way possible. Gain a deeper understanding with our clear explanations and receive exclusive notes to enhance your learning journey. Don't miss out on valuable insights – hit that subscribe button now and embark on a programming adventure with us!

Subscribe to our channel:
https://www.youtube.com/@HTEASYLEARNING