Django Templates Tutorial – Creating A Simple Template

July 3, 2016 by Belal Khan — 1 Comment

Welcome to **Django Templates Tutorial.** In this tutorial, we will create a very simple Django Template. I already posted a couple of tutorials for getting started with Python Django Framework. In this Django Templates Tutorial, we will see how to create a simple template.

If you haven't read the previous post, then It would be good if you go through the last post first. You can visit the earlier tutorials from the given links.

- **Python Django Tutorial for Beginners – Getting Started**
- **Django Views Tutorial – Creating Your First View**

And after reading the posts mentioned above, come back to this Django Template Tutorial to know how the template works in Django.

Contents [hide]

What is a Template?

If I tell you what are Templates in Django, then the answer is "A Django Template is a sequence of text that helps in separation of the presentation layer of a document from its data."

Why Templates?

You may think that why use templates? In the last tutorial where we learned about creating a view. The code we used there in our view is

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("<h1>Welcome to Mailing System Python Django App</h1>")
```

As you can see we have hardcoded the HTML inside HttpResponse and, it is not a convenient way to design the page. It will take more effort. And also when we need to change the design of the page, we need to shift python code. That is why we separate the presentation layer from the python code. And we will learn HOW TO do this, in this **Django Templates Tutorial.**

Django Templates Tutorial

Now let's begin our Django Templates Tutorial, as we know now why we need it. I am using PyCharm (But as the Community Edition don't support Django you need to use PowerShell to command prompt to create the project, then you can open the project in PyCharm or any other source code editor like Sublime, etc.c.

Creating a new Django Project

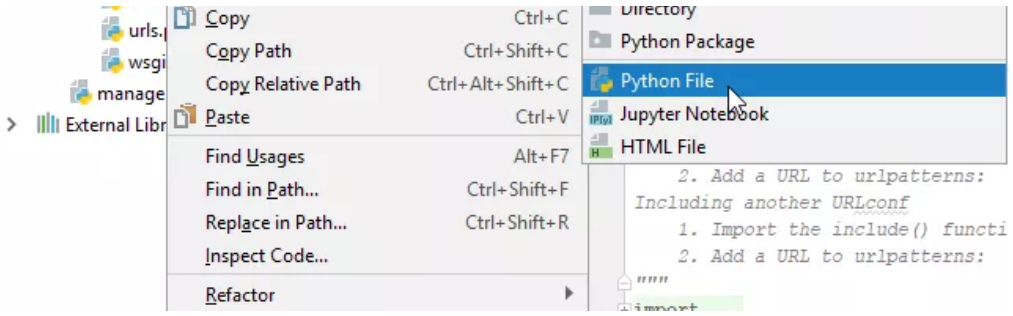
- Run this command to create a new project.

```
django-admin startproject MyDjangoApp
```

- Now open the created project with PyCharm. (You can use Sublime Notepad++ as well).

Creating views.py

To define our views inside the project we will create a new file named **views.py**.



Defining a URL

Now to see our template we will define a URL pattern.

- Come inside the **urls.py** file of your project and modify it as follows.

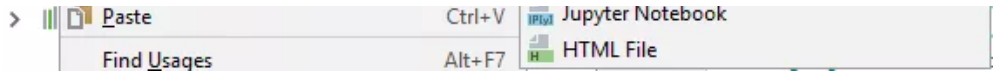
```
from django.conf.urls import url
from django.contrib import admin

#importing the views we created
from . import views

urlpatterns = [
    url(r'^admin/', admin.site.urls),

    #defining a new URL pattern
    #when we will go here our template will render
    url(r'^firsttemplate/', views.index),
]
```

What we did?

The above code is very simple and we already did it in the last tutorial about **creating a simple view in django.**

Creating a Template

First we need a directory to store all the templates, this will help us in separating the design in a particular location. So create a new directory in you project and name it templates.



So as I already said template is nothing but an HTML file containing the presentation of our page. So lets create a template.

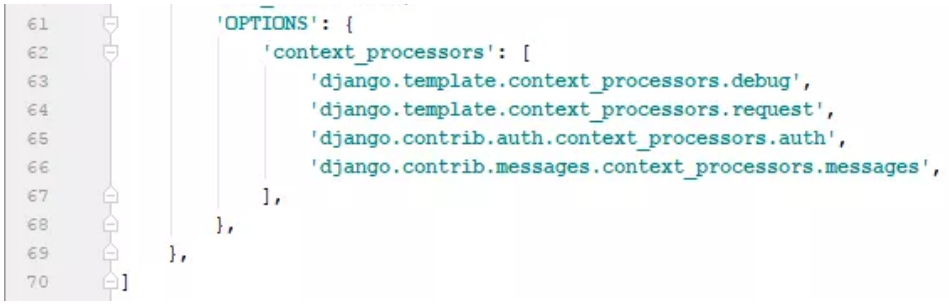
- Right click on templates folder that we created and go to **new->html file**, put a file name and click on create. I have created **index.html.**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Django Templates Tutorial</title>
</head>
<body>
    <h1>Our First Template</h1>
</body>
</html>
```


- Above you can see a very simple html file. And it is actually our template.

Defining Template Directory in Settings

- Only creating the folder is not enough, we need to define the folder in **settings.py**. See the image below.



```
        '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 step is very important, or else you will get **TemplateNotFoundException**.

Rendering Template

Template will be rendered via view. But this time we do not need to hardcode the html inside our python code. As we already separated the html and now it is an individual file.

To render the template we created follow these steps.

- Come inside **views.py** file that we created. And write the following code.

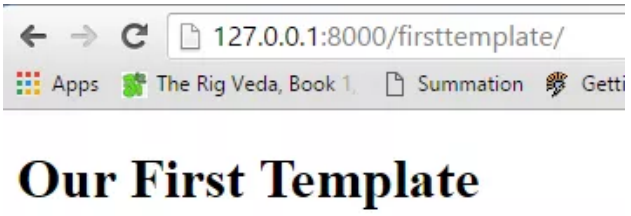
```
from django.http import HttpResponse

#importing loading from django template
from django.template import loader

#our view which is a function named index
def index(request):

    #getting our template
    template = loader.get_template('index.html')

    #rendering the template in HttpResponse
    return HttpResponse(template.render())
```

What we did?

In the above code first we imported loader. We need to it load the template.

Running Your Project

- Now lets try running our project. So again run the command **python manage.py runserver** in the terminal inside PyCharm.
- **http://127.0.0.1:8000/firsttemplate/**
- Go to the above URL and you will see the template we created.



- **BINGO!** our template is rendered successfully.

Passing Values to Template

- We can also pass values to our template. This is needed when we are making a dynamic web app. So lets see how we can do this. Its pretty easy. Just follow these steps.
- Go to your template file index.html and change it as follow

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Django Templates Tutorial</title>
</head>
<body>
<div style="width:50%;margin:0 auto; font-size:250%">
    <h1>User Profile</h1>
    <table>
        <tr>
            <th>Name</th>
            <td>{{ name }}</td>
```

```
            <th>Course</th>
            <td>{{ course }}</td>
        </tr>
        <tr>
            <th>Address</th>
            <td>{{ address }}</td>
        </tr>
    </table>
</div>
</body>
</html>
```

What we did?

In the above html file we just created a simple table. And where we want to pass the variable value we have written a variable instead of writing a static html. Just remember wherever we need to put a variable in html page we need to write it between **{{ }}** .

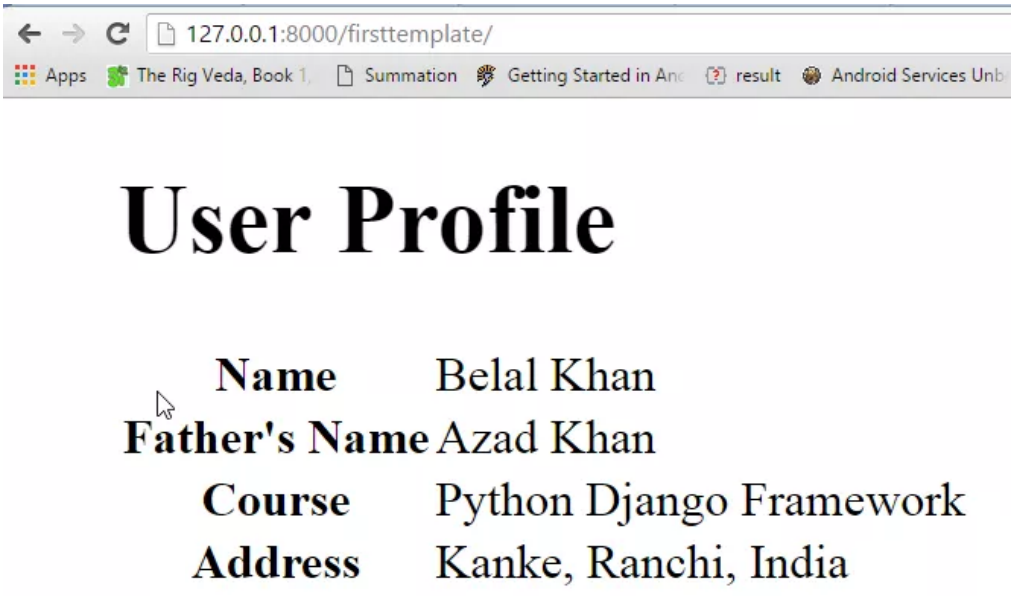
- Now while rendering the template we will pass the values needed. So come inside **views.py** file and modify it as follows.

```
from django.http import HttpResponse

#importing loading from django template
from django.template import loader

#our view which is a function named index
def index(request):

    #getting our template
    template = loader.get_template('index.html')

    #creating the values to pass
    context = {
        'name':'Belal Khan',
        'fname':'Azad Khan',
        'course':'Python Django Framework',
```

What we did?

We have created a variable named context containing the values to pass. It is very much similar to JSON object. And then we are passing the context with request, while rendering our template.

Now refresh your page and you will see the following.



As you can see the values passed are showing up on the page. Bingo! Its working fine.

Next Post: **Django Forms Tutorial – Working with Forms in Django**

So thats all for this **Django Templates Tutorial**. Feel free to leave your comments if having any ideas or feedback regarding this **Django Templates Tutorial**. We will do some more cool stuffs in upcoming tutorials. Thank You 🙂

Share this:

Related

Django Forms Tutorial - Working with Forms in Django
July 4, 2016
In "Python Django"

Django Views Tutorial - Creating Your First View
July 1, 2016
In "Python Django"

Python Django Tutorial for Beginners - Getting Started
June 30, 2016
In "Python Django"

Filed Under: Python Django
Tagged With: django templates, django templates tutorial

About Belal Khan

Hey this is Belal Khan, a computer science student from St. Xavier's College, Ranchi. Here I post about Python and Django.

Comments

ashish jullia says

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

POST COMMENT

Notify me of follow-up comments by email.

Notify me of new posts by email.

POPULAR TUTORIALS