

Django template view

template view

- It's simple view to display simple html page with/without using the context.
- we will be using the code from [Create django app](#)
- create a template templates/my_template.html
- we will be using the below template in the view

templates/my_template.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>{{ title }}</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.c
ss">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></
script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"
></script>
</head>
<body>
  <div class="jumbotron text-center">
    <h1>{{title}}</h1>
    <p>Resize this responsive page to see the effect!</p>
  </div>
</body>
</html>
```

function based template view

- open my_app/views.py and add below code to it.
- my_app/views.py

```
from django.shortcuts import render

def my_template_view(request):
    context = {"title": "Simple template view"}
    return render(request, "my_template.html", context)
```

class based template view¶

- Django comes with generic reusable views.
- Lets use TemplateView

```
from django.views.generic import TemplateView

# ...
class MyTemplateView(TemplateView):
    template_name = "my_template.html"

    def get_context_data(self):
        context = {"title": "Simple template view"}
        return context
```

configure urls¶

- open my_app/urls.py and add below code to it.

```
from django.urls import path
from my_app import views

urlpatterns = [
    # ...
    path('fbv_template_view', views.my_template_view,
    name='my_template_view'),
    path('cbv_template_view', views.MyTemplateView.as_view(),
    name='my_template_view'),
    # ...
]
```

- Access the url http://localhost:8000/my_app/fbv_template_view to see the function based template view
- Access the url http://localhost:8000/my_app/cbv_template_view to see the class based template view