

# Django - Generic Views

In some cases, writing views, as we have seen earlier is really heavy. Imagine you need a static page or a listing page. Django offers an easy way to set those simple views that is called generic views.

Unlike classic views, generic views are classes not functions. Django offers a set of classes for generic views in `django.views.generic`, and every generic view is one of those classes or a class that inherits from one of them.

There are 10+ generic classes –

```
>>> import django.views.generic
>>> dir(django.views.generic)

['ArchiveIndexView', 'CreateView', 'DateDetailView', 'DayArchiveView',
 'DeleteView', 'DetailView', 'FormView', 'GenericViewError', 'ListView',
 'MonthArchiveView', 'RedirectView', 'TemplateView', 'TodayArchiveView',
 'UpdateView', 'View', 'WeekArchiveView', 'YearArchiveView', '__builtins__',
 '__doc__', '__file__', '__name__', '__package__', '__path__', 'base', 'dates',
 'detail', 'edit', 'list']
```

This you can use for your generic view. Let's look at some example to see how it works.

## Static Pages

Let's publish a static page from the “static.html” template.

Our static.html –

```
<html>
  <body>
    This is a static page!!!
  </body>
</html>
```

If we did that the way we learned before, we would have to change the **myapp/views.py** to be –

```
from django.shortcuts import render
```

```
def static(request):
    return render(request, 'static.html', {})
```

and **myapp/urls.py** to be –

```
from django.conf.urls import patterns, url

urlpatterns = patterns("myapp.views", url(r'^static/', 'static', name = 'static'),)
```

The best way is to use generic views. For that, our myapp/views.py will become –

```
from django.views.generic import TemplateView

class StaticView(TemplateView):
    template_name = "static.html"
```

And our myapp/urls.py we will be –

```
from myapp.views import StaticView
from django.conf.urls import patterns

urlpatterns = patterns("myapp.views", (r'^static/$', StaticView.as_view()),)
```

When accessing /myapp/static you get –

**Static Page**  
page not found or type unknown

For the same result we can also, do the following –

No change in the views.py  
Change the url.py file to be –

```
from django.views.generic import TemplateView
from django.conf.urls import patterns, url

urlpatterns = patterns("myapp.views",
    url(r'^static/', TemplateView.as_view(template_name = 'static.html'),),)
```

As you can see, you just need to change the url.py file in the second method.

## List and Display Data from DB

We are going to list all entries in our Dreamreal model. Doing so is made easy by using the ListView generic view class. Edit the url.py file and update it as –

```
from django.views.generic import ListView
from django.conf.urls import patterns, url

urlpatterns = patterns(
    "myapp.views", url(r'^dreamreals/', ListView.as_view(model = Dreamreal,
        template_name = "dreamreal_list.html")),
)
```

Important to note at this point is that the variable pass by the generic view to the template is object\_list. If you want to name it yourself, you will need to add a context\_object\_name argument to the as\_view method. Then the url.py will become –

```
from django.views.generic import ListView
from django.conf.urls import patterns, url

urlpatterns = patterns("myapp.views",
    url(r'^dreamreals/', ListView.as_view(
        template_name = "dreamreal_list.html")),
    model = Dreamreal, context_object_name = "dreamreals_objects" ,)
```

The associated template will then be –

```
{% extends "main_template.html" %}
{% block content %}
Dreamreals: <p>
{% for dr in object_list %}
{{dr.name}}</p>
{% endfor %}
{% endblock %}
```

Accessing /myapp/dreamreals/ will produce the following page –

## List and Display Data from DB

---

Revision #1

Created 14 December 2022 11:09:52 by Admin

Updated 14 December 2022 11:10:28 by Admin