Django list view

Django list view[¶](https://learnbatta.com/course/django/list-view/#django-list-view)

* List view is used to show the list data to the user.
* For example: Show list of contacts to the user.

django contact Model[¶](https://learnbatta.com/course/django/list-view/#django-contact-model)

my\_app/models.py

from django.db import models

class Contact(models.Model):

first\_name = models.CharField(max\_length=30)

last\_name = models.CharField(max\_length=30)

email = models.EmailField(max\_length=255)

phone = models.CharField(max\_length=10, null=True)

class Meta:

db\_table = "contact"

django list view template[¶](https://learnbatta.com/course/django/list-view/#django-list-view-template)

tempaltes/list\_template.html

* Below template recieves the contacts from the context and renders the info.

<h1>Contacts</h1>

{% for contact in contacts %}

<div>

<p><strong>Name:</strong> {{ contact.first\_name }} </p>

<p><strong>Email:</strong> {{ contact.email }} </p>

</div>

{% endfor %}

function based list view[¶](https://learnbatta.com/course/django/list-view/#function-based-list-view)

* It's easy to write and use.
* If we have complex functionality then go for generic ListView
* Let's look at the code

my\_app/views.py

from django.shortcuts import render

from my\_app.models import Contact

def fbv\_list\_view(request):

contacts = Contact.objects.all()

context = {'contacts': contacts}

return render(request, 'list\_template.html', context)

class based list view[¶](https://learnbatta.com/course/django/list-view/#class-based-list-view)

* Django provides generic ListView for list functionality.
* We can modify it as needed.
* Let's look at the code below.

from django.views.generic.list import ListView

from my\_app.models import Contact

class CBVListView(ListView):

template\_name = 'list\_template.html'

context\_object\_name = 'contacts'

queryset = Contact.objects.all()

* It either requires attribute queryset or method get\_queryset to return the query results.
* We can have pagination by using attribute paginate\_by
* In template we can access the query results either with context\_object\_name value or with name object\_list

configure the urls[¶](https://learnbatta.com/course/django/list-view/" \l "configure-the-urls" \o "Permanent link)

* open my\_app/urls.py and add below code to it.

from django.urls import path

from my\_app import views

urlpatterns = [

# ...

path('fbv\_list\_view', views.fbv\_list\_view, name='fbv\_list\_view'),

path('cbv\_list\_view', views.CBVListView.as\_view(), name='cbv\_list\_view'),

# ...

]

* Access the url <http://localhost:8000/my_app/fbv_list_view> to see the function based list view
* Access the url <http://localhost:8000/my_app/cbv_list_view> to see the class based list view

References[¶](https://learnbatta.com/course/django/list-view/#references)

* <https://docs.djangoproject.com/en/dev/ref/class-based-views/generic-display/#listview>
* <https://ccbv.co.uk/projects/Django/4.1/django.views.generic.list/ListView/>