Django detail view

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

* Django detail view shows the single object information to the user.
* It fetches the object from the database using primary key or an equivalent.
* Let's write a view for contact detail view page.

contact model[¶](https://learnbatta.com/course/django/detail-view/#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"

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

* It's useful when we have a functionality with simple requirements.
* Look at the code below.

from django.http import Http404

from django.shortcuts import render

from my\_app.models import Contact

def fbv\_detail\_view(request, pk):

try:

obj = Contact.objects.get(id=pk)

except Contact.DoesNotExist:

raise Http404

context = {"object": obj}

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

* Detail view function takes two parameters request and pk
* request is an instance of HttpRequest class, which contains the request info
* pk is a url keyword argument used query the database to get the object.

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

* Django provides generic class DetailView
* We can inherit the DetailView and provide the required attibutes and/or methods to make it work.
* Only use if it fits your use case.
* Let's look at the code below.

from django.views.generic.detail import DetailView

from my\_app.models import Contact

class CBVListView(DetailView):

template\_name = 'detail\_template.html'

pk\_url\_kwarg = 'pk'

queryset = Contact.objects.all()

* In the above code pk\_url\_kwarg is used as a primary key in the query.
* Either we need to provider queryset attribute or get\_queryset() or get\_object() methods.
* There are other attributes and methods available to override based on the functionality.
* We can find more info at [ccbv](https://ccbv.co.uk/projects/Django/4.1/django.views.generic.detail/DetailView/)
* It sends the context {"object": <Model object>} to the template.

detail\_template.html[¶](https://learnbatta.com/course/django/detail-view/#detail_templatehtml)

* Let's look at the code for the template

<div>

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

<p>Email: {{ object.email }}</p>

</div>

* Template receives the context and reders the template.

configure urls[¶](https://learnbatta.com/course/django/detail-view/" \l "configure-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\_detail\_view/<int:pk>', views.fbv\_detail\_view, name='fbv\_detail\_view'),

path('cbv\_detail\_view/<int:pk>', views.CBVDetailView.as\_view(), name='cbv\_detail\_view'),

# ...

]

* Open url <http://127.0.0.1:8000/my_app/fbv_detail_view/1> to see the response for function based detail view
* Open url <http://127.0.0.1:8000/my_app/cbv_detail_view/1> to see the response for class based detail view
* Replace the 1 with any id in the contact table.
* It throws 404 status code if id does not exist in the table.

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

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