Django create view

Django ModelForm[¶](https://learnbatta.com/course/django/create-view/" \l "django-modelform" \o "Permanent link)

* Django ModelForm is an advanced version of [django Form](https://docs.djangoproject.com/en/dev/topics/forms/" \l "the-form-class).
* ModelForm is associated with a django model.
* Model form can generate auto form fields using form meta class
* Can also define additional form fields.
* Let's look at the code for model and model form.

my\_app/models.[py](https://learnbatta.com/course/django/create-view/)

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"

my\_app/forms.[py](https://learnbatta.com/course/django/create-view/)

from django import forms

from my\_app.models import Contact

class ContactModelForm(forms.ModelForm):

class Meta:

model = Contact

fields = ['first\_name', 'last\_name', 'email']

* form Meta class attributes
  + *model*: defines model name
  + *fields*: form fields to include

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

* the template looks like below.

templates/form\_template.html

<form action="" method="POST" enctype="multipart/form-data">

{% csrf\_token %}

{{ form.as\_table }}

<input type="submit" value="submit" />

</form>

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

function based createview[¶](https://learnbatta.com/course/django/create-view/" \l "function-based-createview" \o "Permanent link)

* use function based create view for simple functionality.
* we will use contact model form for form validation.
* Let's look at the create view code below.

from django.http import HttpResponse

from django.shortcuts import render

from my\_app.forms import ContactModelForm

def fbv\_create\_view(request):

if request.method.lower() == "post":

form = ContactModelForm(request.POST)

if form.is\_valid():

instance = form.save()

return HttpResponse(f"contact saved: {instance}")

else:

form = ContactModelForm()

context = {'form': form}

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

class based createview[¶](https://learnbatta.com/course/django/create-view/" \l "class-based-createview" \o "Permanent link)

* Django comes with generic view CreateView
* It handles the requests GET and POST
* When a POST request is recieved then django generic view calls two methods form\_valid and form\_invalid after validating the requested data.
* CreateView also has other methods like get\_context\_data, get\_initial, etc.
* For current functionality we just need to override form\_valid method.
* Let's look at the code below.

from django.http import HttpResponse

from django.views.generic.edit import CreateView

from my\_app.forms import ContactModelForm

class CBVCreateView(CreateView):

form\_class = ContactModelForm

template\_name = 'form\_template.html'

def form\_valid(self, form):

instance = form.save()

return HttpResponse(f"contact saved: {instance}")

configure urls[¶](https://learnbatta.com/course/django/create-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\_create\_view', views.fbv\_create\_view, name='fbv\_create\_view'),

path('cbv\_create\_view', views.CBVCreateView.as\_view(), name='cbv\_create\_view'),

# ...

]

* Access the url <http://localhost:8000/my_app/fbv_create_view> to see the function based create view
* Access the url <http://localhost:8000/my_app/cbv_create_view> to see the class based create view
* Open above mentioned urls and fill in the contact info and submit the form to save the contact info to the database.

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

* <https://docs.djangoproject.com/en/dev/topics/forms/modelforms/>
* <https://docs.djangoproject.com/en/dev/ref/class-based-views/generic-editing/#django.views.generic.edit.CreateView>
* <https://ccbv.co.uk/projects/Django/4.1/django.views.generic.edit/CreateView/>