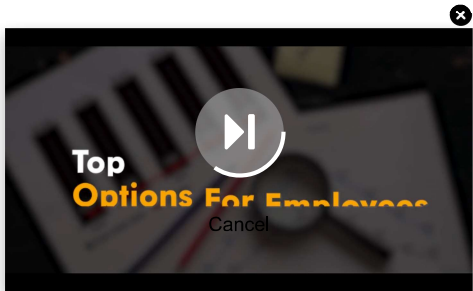




Django Howto's (/howto/django/)

- Django Change Password (/howto/django/django-change-password/)
- Use Dynamic URL Inside the Template in Django (/howto/django/django-template-url/)
- Use the Include Tag in Django (/howto/django/django-template-include/)
- Django OR Filter (/howto/django/django-filter-or/)
- Use jQuery in a Django App (/howto/django/django-jquery/)
- Django Bootstrap (/howto/django/django-bootstrap/)
- Django Download File
- Reset Admin Password



X

Top

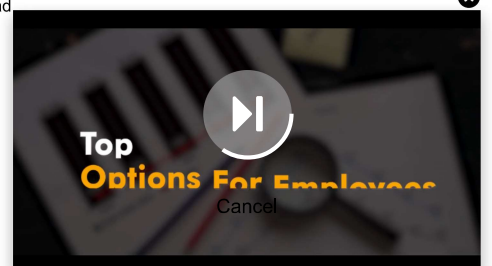
Options For Employees

Cancel

▶



 **ezoic** (<https://www.ezoic.com/what-is-ezoic/>) report this ad



X

Top

Options For Employees

Cancel

▶

Django Download File

Django (/tags/django/) Django Download (/tags/django-download/)

Created: May-06, 2022



(<http://www.facebook.com/sharer.php?>

src=bm&u=https%3a%2f%2fwww.delftstack.com%2fhowto%2fdjango%2fdjang



(<https://twitter.com/intent/tweet/>

Django%20Download%20File&url=https%3a%2f%2fwww.delftstack.com%2fho



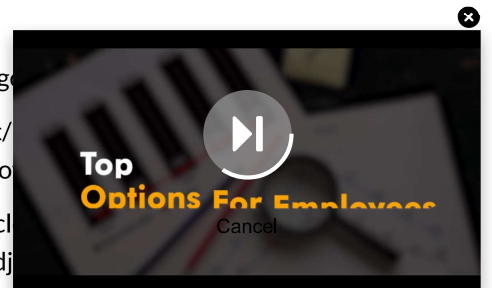
(<https://www.linkedin.com/shareArticle>


```
mini=true&url=https%3a%2f%2fwww.delftstack.com%2fhowto%2fdjango%2fdj
```

file%2f&title=Diango%20Howto%27s-

Django%20I

file%2f&sur

ango-download-^(x)

This article shows how to create a website where the admin can upload a file and make the user download a file from a web page in Django. 

Upload and Download a File in Django

Let's go to the terminal and create a demo project to demonstrate to you. Follow the commands to create a Django project and an app in that project.

```
django-admin startproject django_project
```

```
python manage.py startapp blog
```

Let's open this newly created Django project in the available text editor.

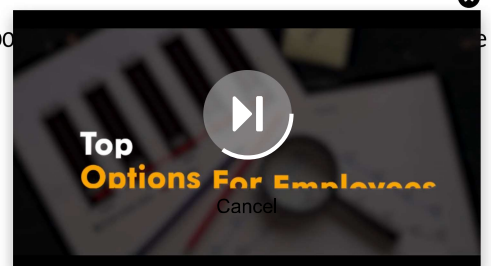
We will first create a folder for our uploaded files, so let's open `settings.py`. Find the `INSTALLED_APPS` list and pass the app name as an item.

```
INSTALLED_APPS=[ 'blog.apps.BlogConfig', ]
```

Now we need to create a folder where the files will get uploaded. Find the `STATIC_URL` variable and write the following code below the variable.

```
STATICFILES_DIRS=[os.path.join(BASE_DIR, 'static')]
```

Now we need to set our `MEDIA_URL` equals to `'/media/'` and then set `MEDIA_ROOT` to the path where the uploaded files will be saved.



```
MEDIA_ROOT=os.path.join(BASE_DIR, 'media_cdn')
```

Now let's add our media and static rules to the `urls.py`. Let's define the path for `home` and `download` functions that will pull from the `views.py` file.

We are also going to set the URL for the download file.

We cannot use the `url` object because this is deprecated. So instead of this, we can easily use the `re_path` object.

```
re_path(r'^download/(?P<path>.*)$', serve, {'document_root': settings.MEDIA_ROOT})
```

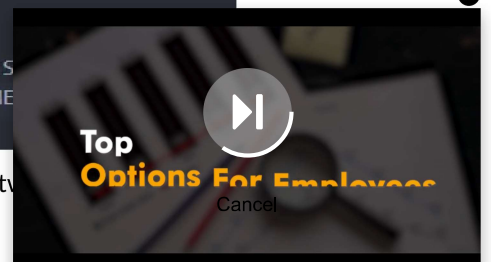
In our case, we are using this code.

```
import imp
from django.conf import settings
from django.contrib import admin
from django.urls import path
from blog import views
from django.conf import settings
from django.conf.urls.static import static
from django.urls import re_path
from django.views.static import serve

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.home),
    re_path(r'^download/(?P<path>.*)$', serve, {'document_root': settings.MEDIA_ROOT}),
]

if settings.DEBUG:
    urlpatterns+=static(settings.STATIC_URL,document_root=settings.S
    urlpatterns+=static(settings.MEDIA_URL,document_root=settings.ME
```

Let's create our model inside the `models.py` file, and in this model, we will have two `FileField`, and the second would be `CharField`.



We also need to create a string dunder method to return the page's title.

```
from django.db import models

class FilesAdmin(models.Model):
    adminupload=models.FileField(upload_to='media')
    title=models.CharField(max_length=50)
    def __str__(self):
        return self.title
```

We need to register our FilesAdmin model in the admin.py file.

```
from .models import FilesAdmin

admin.site.register(FilesAdmin)
```

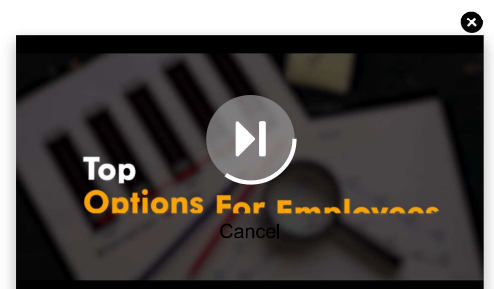
We will create a home.html file within the blog templates inside the blog folder. Let's add the following code in the home.html file among the body tag.

```
{% for post in file %}
<h2>{{post.title}}</h2>
<a href="{{post.adminupload.url}}" download="{{post.adminupload.url}}">Download</a>
{% endfor %}
```

We will get all objects from the FilesAdmin model in the home function. Then pass the context in the home.html file and render this page.

We need to define our home and download functions in the views.py file. We will get the path in the download function, where the file will download by default.

In the next line, we will check whether the directory exists or not. Then we will open a file in reading binary mode and return it in the HTTP response.




```

from django.shortcuts import render
from requests import Response
from .models import FilesAdmin

def home(request):
    # get all objects
    OUR_CONTEXT={'file':FilesAdmin.objects.all()}
    return render(request,'blog/home.html',OUR_CONTEXT)

import os
from django.conf import settings
from django.http import HttpResponse, Http404

def download(request, path):
    # get the download path
    download_path = os.path.join(settings.MEDIA_ROOT, path)
    if os.path.exists(download_path):
        with open(download_path, 'rb') as fh:
            response = HttpResponse(fh.read(), content_type="application/adminupload")
            response['Content-Disposition'] = 'inline; filename=' + os.path.basename(download_path)
            return response
    raise Http404

```

Now we need to make migration by running these commands.

```
python manage.py makemigrations
```

```
python manage.py migrate
```

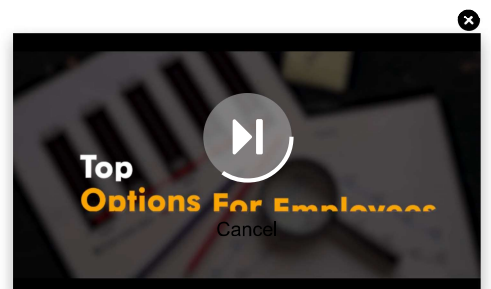
Create a superuser.

```
python manage.py createsuperuser
```

Let's run the Django server and try to upload an mp3 file from the admin panel.



Let's open the home page and see if we can download it or not.



[Download](#)



Write for us

DelftStack articles are written by software geeks like you. If you also would like to contribute to DelftStack by writing paid articles, you can check the write for us (/write-for-us/) page.

< [Django Bootstrap \(/Howto/Django/Django-Bootstrap/\)](#)

> [Reset Admin Password In Django](#)

