Django File Upload

File upload to the server using Django is a very easy task. Django provides built-in library and methods that help to upload a file to the server.

The **forms.FileField()** method is used to create a file input and submit the file to the server. While working with files, make sure the HTML form tag contains **enctype="multipart/form-data"**property.

Let's see an example of uploading a file to the server. This example contains the following files.

**Template (index.html)**

It will create an HTML form which contains a file input component.

1. <body>
2. <form method="POST" **class**="post-form" enctype="multipart/form-data">
3. {% csrf\_token %}
4. {{ form.as\_p }}
5. <button type="submit" **class**="save btn btn-default">Save</button>
6. </form>
7. </body>

**Form (forms.py)**

1. from django **import** forms
2. **class** StudentForm(forms.Form):
3. firstname = forms.CharField(label="Enter first name",max\_length=50)
4. lastname  = forms.CharField(label="Enter last name", max\_length = 10)
5. email     = forms.EmailField(label="Enter Email")
6. file      = forms.FileField() # **for** creating file input

**View (views.py)**

Here, one extra parameter **request.FILES** is required in the constructor. This argument contains the uploaded file instance.

1. from django.shortcuts **import** render
2. from django.http **import** HttpResponse
3. from myapp.functions.functions **import** handle\_uploaded\_file
4. from myapp.form **import** StudentForm
5. def index(request):
6. **if** request.method == 'POST':
7. student = StudentForm(request.POST, request.FILES)
8. **if** student.is\_valid():
9. handle\_uploaded\_file(request.FILES['file'])
10. **return** HttpResponse("File uploaded successfuly")
11. **else**:
12. student = StudentForm()
13. **return** render(request,"index.html",{'form':student})

**Specify URL (urls.py)**

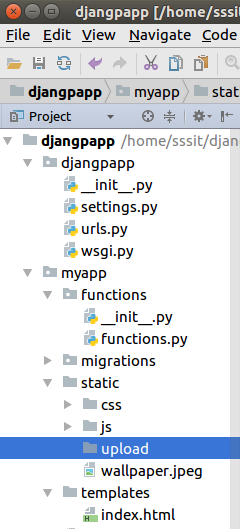
1. from django.contrib **import** admin
2. from django.urls **import** path
3. from myapp **import** views
4. urlpatterns = [
5. path('admin/', admin.site.urls),
6. path('index/', views.index),
7. ]

**Upload Script (functions.py)**

This function is used to read the uploaded file and store at provided location. Put this code into the **functions.py** file. But first create this file into the project.

1. def handle\_uploaded\_file(f):
2. with open('myapp/static/upload/'+f.name, 'wb+') as destination:
3. **for** chunk in f.chunks():
4. destination.write(chunk)

Now, create a directory **upload** to store the uploaded file. Our project structure looks like below.

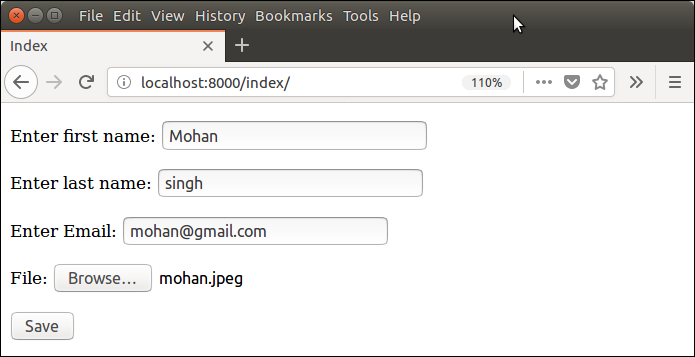


Initially, this directory is empty. so, let's upload a file to it and later on it will contain the uploaded file.

**Start Server**

1. python manage.py runserver

**Output**



Submit this form and see the **upload** folder. Now, it contains the uploaded file.

