Django Example

Image Upload Path

> **Caution**
>
> Among the following codes provided as the guide, file upload part is a **sample code** and has insufficient security.
>
> As for the file upload, please use the code used within your project and refer the following code to handle the system link.

1. Preliminary Information

Example in the linking guide is constituted as follows.

- Project Name : **synapeditor_django**
- Application Name : **edit**

1.1. Adding the Application

```
# Project settings.py
...
INSTALLED_APPS = [
        'edit.apps.EditConfig', #Add application
        ...
]
...


# Project urls.py
...
urlpatterns = [
        path('edit/', include('edit.urls')), # Adding URL to use in the application
        ...
]
...
```

2. Uploading Images, Videos and Files

It is defined as follows for the upload. You may change the path or API format.

- Image Upload Path : **media**
- Image Upload API : **/edit/uploadFile**

2.1. Editor Configuration

```
// Synap Editor configuration object
var SynapEditorConfig = {
        ...
        'editor.upload.image.param': {
                'csrfmiddlewaretoken': '{{ csrf_token }}'
        },
        'editor.upload.image.api': '/edit/uploadFile',
        ...
}
```

2.2. Upload Path Configuration

```
# Project settings.py
...
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
...

# Project urls.py
...
urlpatterns = [
        ...
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
...
```

2.3. Linking URL and View

```
# Application urls.py
from django.urls import path
from . import views

urlpatterns = [
        ...
    path('uploadFile/', views.upload_file, name='upload_file'),
        ...
]
```

2.4. Creating view

After the upload, you shall respond with '**uploadPath**' included in JSON object.

```
# Application views.py
from django.http import JsonResponse
from .forms import UploadFileForm

# Upload the file.
def upload_file(request):
    if request.method == 'POST':
        form = UploadFileForm(request.POST, request.FILES)

        if form.is_valid():
            uploaded_file = form.save()
                        data = {
                                'uploadPath': uploaded_file.file.url
                        }

    return JsonResponse(data)
```

2.5. Creating form

```
# Application forms.py
from .models import UploadFile
from django import forms

class UploadFileForm(forms.ModelForm):
    class Meta:
        model = UploadFile
        fields = ('file', )
```

2.6. Creating model

```
# Application models.py
import uuid
from django.db import models

# Return the name of the file to save.
def get_file_name(instance, filename):
    ext = filename.split('.')[-1]
    return "%s.%s" % (uuid.uuid4(), ext)

class UploadFile(models.Model):
    file = models.FileField(upload_to=get_file_name)
```

3. Importing HWP, MS Word and Excel Documents

It is defined as follows for the upload. You may change the path or API format.

- Upload documents in **media** directory.
- Save the converted outcomes in **media/output** directory.
- The document import API is '**/edit/importDoc**'.

3.1. Editor Configuration

```
// Synap Editor configuration object
var SynapEditorConfig = {
        ...
        'editor.import.param': {
                'csrfmiddlewaretoken': '{{ csrf_token }}'
        },
        'editor.import.api': '/edit/importDoc/',
        ...
}
```

3.2. Upload Path Configuration

```
# Project settings.py
...
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
...

# Project urls.py
...
urlpatterns = [
        ...
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
...
```

3.3. Linking url and view

```
# Application urls.py
from django.urls import path
from . import views

urlpatterns = [
        ...
    path('importDoc/', views.import_doc, name='import_doc'),
        ...
]
```

3.4. Creating view

After the upload, you shall respond with '**serializedData**' and '**importPath**' included in JSON object.

```
# Application views.py
import os
import subprocess
import zipfile
import zlib
from django.conf import settings
from subprocess import call
from django.http import JsonResponse
from .forms import UploadFileForm

PROJECT_PATH = os.path.abspath(os.path.dirname(__name__)) # Absolute path of the project
MEDIA_ROOT = settings.MEDIA_ROOT # Absolute path for the file upload
# Import a document.
def import_doc(request):
    if request.method == 'POST':
        form = UploadFileForm(request.POST, request.FILES)

        if form.is_valid():
            uploaded_file = form.save()

                        # 1. Document conversion
            result = execute_converter(uploaded_file.file)

            if result['resultCode'] == 0:
                output_path = result['outputPath']
                                # 2. Decompressing the converted outcome
                unzip_path = unzip(output_path)
                pb_path = unzip_path + '/document.pb'

                                # 3. Serializing PB data
                serialized_data = serialize_pb(pb_path)
                common_prefix = os.path.commonprefix([output_path, PROJECT_PATH])
                import_path = '/' + os.path.relpath(unzip_path, common_prefix)

            data = {
                'serializedData': serialized_data, # Serialized pb data
                'importPath': import_path # Path in which the converted outcome is stored (it shall be the
path to which the browser can access)
            }

    return JsonResponse(data)

# Run the conversion module to convert the document.
def execute_converter(file):
    fname, ext = os.path.splitext(file.name)
    module_path = '{0}/sedocConverter/sedocConverter_exe'.format(PROJECT_PATH)
    font_path = '{0}/sedocConverter/fonts'.format(PROJECT_PATH)
    input_path = file.path
    output_path = '{0}/output/{1}.zip'.format(MEDIA_ROOT, os.path.basename(fname))
    temp_path = '{0}/temp'.format(PROJECT_PATH)
        args = [module_path, '-z', '-f', font_path, input_path, output_path, temp_path]
        result_code = None
```

```
        process = subprocess.Popen(args)
        try:
                result_code = process.wait(timeout=20) # The process shall be terminated if the conversion
is not completed within 20 seconds
    except subprocess.TimeoutExpired:
        process.kill()


    return {
                'resultCode': result_code,
                'outputPath': output_path
        }

#Decompress the converted file.
def unzip(zipFilePath):
    unzip_path, ext = os.path.splitext(zipFilePath)
    zip = zipfile.ZipFile(zipFilePath)
    zip.extractall(unzip_path)

    return unzip_path

# Read and serialize pb file and return it.
def serialize_pb(pbFilePath):
    serialized_data = []
    pb_file = open(pbFilePath, 'rb')
    pb_file.seek(16)
    pb_contents= pb_file.read()
    decompressed = zlib.decompress(pb_contents)

        for byte in decompressed:
        serialized_data.append(byte & 0xFF)

    pb_file.close()

        return serialized_data
```

3.5. Creating form

```
# Application forms.py
from .models import UploadFile
from django import forms

class UploadFileForm(forms.ModelForm):
    class Meta:
        model = UploadFile
        fields = ('file', )
```

3.6. Creating model

```
# Application models.py
import uuid
from django.db import models

# Return the name of the file to save.
def get_file_name(instance, filename):
    ext = filename.split('.')[-1]
    return "%s.%s" % (uuid.uuid4(), ext)

class UploadFile(models.Model):
    file = models.FileField(upload_to=get_file_name)
```

See also

- Java Spring Framework Example
- Java Servlet Example
- ASP.NET (C#) Example
- ASP(Classic) Example
- PHP Example
- PHP4 Example
- Django Example
- Ruby On Rails Example
- Wordpress plugin Example