

# Django

- 1.
  - 1.1.
- 2. , ,
  - 2.1.
  - 2.2.
  - 2.3. url view
  - 2.4. view
  - 2.5. form
  - 2.6. model
- 3. HWP, Word, Excel
  - 3.1.
  - 3.2.
  - 3.3. url view
  - 3.4. view
  - 3.5. form
  - 3.6. model



## 1.

- : **synapeditor\_django**
- : **edit**

### 1.1.

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

# urls.py
...
urlpatterns = [
    path('edit/', include('edit.urls')), # url
    ...
]
...
```

## 2. , ,

. API .

- : **media**
- API: **/edit/uploadFile**

### 2.1.

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

## 2.2.

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

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

## 2.3. url view

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

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

## 2.4. view

JSON **'uploadPath'** .

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

# .
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. form

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

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

## 2.6. model

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

# .
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. HWP, Word, Excel

. API .

- **media** .
- **media/output** .
- API '**edit/importDoc**'.

### 3.1.

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

### 3.2.

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

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

### 3.3. url view

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

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

### 3.4. view

JSON '**serializedData**' '**importPath**' .

```
# 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__)) #
MEDIA_ROOT = settings.MEDIA_ROOT #
# .
def import_doc(request):
    if request.method == 'POST':
        form = UploadFileForm(request.POST, request.FILES)

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

            # 1.
            result = execute_converter(uploaded_file.file)

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

                # 3. PB
                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
                    'importPath': import_path # ( )
                }

            return JsonResponse(data)

# .
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) # 20
        except subprocess.TimeoutExpired:
            process.kill()

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

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

    return unzip_path

# pb serialize .
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. form

```

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

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

```

### 3.6. model

```

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

# .
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)

```

- [Java Spring Framework](#)
- [Java Servlet](#)

- [ASP.NET \(C#\)](#)
- [ASP\(Classic\)](#)
- [PHP](#)
- [PHP4](#)
- [Django](#)
- [Ruby On Rails](#)
- [Wordpress plugin](#)