

## 文件传输

django rest framework 中提供了 `FileField` 的模型字段用于文件传输。

原型为 `class FileField(upload_to='', storage=None, max_length=100, **options)`

和图片的上传一样，要在 `settings.py` 里设置上传文件的地址，即文件在服务器中存在谁没位置，如：

```
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, "media")
```

同时需要在视图中使用解析文件表单的解析器 `MultiPartParser` 和 `FormParser`，示例代码如下：

```
from rest_framework.views import APIView
from rest_framework.parsers import MultiPartParser, FormParser
from rest_framework.response import Response
from rest_framework import status
from .serializers import FileSerializer
class FileView(APIView):
    parser_classes = (MultiPartParser, FormParser)
    def post(self, request, *args, **kwargs):
        file_serializer = FileSerializer(data=request.data)
        if file_serializer.is_valid():
            file_serializer.save()
            return Response(file_serializer.data, status=status.HTTP_201_CREATED)
        else:
            return Response(file_serializer.errors,
                            status=status.HTTP_400_BAD_REQUEST)
```

最终前端传输的文件会存到 `seetings.py` 中指定的服务器内的路径上。

### 名词解释

- 解析器：用来把前端传入的数据解析成后端能看懂的样子，可以用 `QueryDict` 填充前端请求携带的数据 `request.data`，上面两个解析器可以解析内容类型为 `multipart/form` 的 Http 请求
- `Multipart/form-data`：一种请求类型（`Content Type`），在 HTTP 请求携带文件时会用到。涉及到上传文件、图片等的操作，基本上全部都会使用请求类型为 `Multipart/form-data` 的请求方式来完成上传，该方式本质是 `POST` 方式。

想进一步了解可以查看参考文献。

## SMILES 调研

### SMILES

SIMILES 是一种表示有机物结构式和化学反应的分子语言，任何有机物都有唯一的 SIMILES 的表示方法。下面是一些例子。

SMILES	名字	SMILES	名字
CC	乙烷	[OH3+]	水合氢离子
O=C=O	二氧化碳	[235U]	铀-235
C#N	氰化氢	F/C=C/F	E-二氟乙烯
CCN(CC)CC	三乙胺	F/C=C\F	Z-二氟乙烯
CC(=O)O	醋酸	N[C@H](C)C(=O)O	L-丙氨酸

SMILES 可以以字符串的形式读取和表示，在指定基团时，可以参考潘逸轩同学提出的方法，给出此基团在字符串中的起始和结束位置，利用字符串切片读取基团对应的 SMILES 子串，再根据子串判断出指定基团的类别。对于目标基团也可采用类似的方法传输，并与判断基团类别，再根据反应前后的基团类别确定反应类型（实质上就是一个分类讨论的过程）。

有关前端以何形式将结构式传给后端，前端的回应是可以根据后端的需求来做，因此或许可以传一个包含原反应物结构式的 SMILES 字符串和指定基团的 SMILES 字符串的文本文件就可以完成前后端结构式的传递。

## rdkit

如果查询数据库时，数据库对有机物格式有其他要求，如要求 .mol 文件，可通过 Python 的包 rdkit 将 SMILES 字符串转化成 .mol 等文件。

示例如下：

```
>>> m1 = Chem.MolFromSmiles('C1CCCC1') # 将SMILES串转成mol格式
>>> print(Chem.MolToMolBlock(m1))

      RDKit          2D

  4  4  0  0  0  0  0  0  0  0  0999 v2000
    1.0607    0.0000    0.0000 c   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
...
  4  1  1  0
M  END
```

rdkit 的功能：

- 读取分子：SIMILES/SMARTS、.sdf、.mol、.mol2、其他格式
- 输出分子：SIMILES/SMARTS、.sdf、.mol、其他格式
- 分子可视化：单个展示、批量展示、3D展示

## pysmile

pysmile 可以通过给定的 SMILES 串来确定元素之间的键连关系（以列表形式给出），也可以画出大致结构式并保存，在后端的开发中或许也用得上。示例代码如下：

```
from pysmiles import read_smiles
import networkx as nx
import matplotlib.pyplot as plt

smiles = 'N#CC#N' # 给定的SMILES表达式
mol = read_smiles(smiles) # 读取到一个networkx的网络结构中
print(mol.nodes) # 打印节点信息
print(mol.edges) # 打印边信息
print(nx.to_numpy_matrix(mol)) # 打印邻接矩阵信息，该邻接矩阵表示键连关系

elements = nx.get_node_attributes(mol, name = "element")
nx.draw(mol, with_labels=True, labels=elements)
plt.savefig('pysmiles.png') # 保存图层
```

## 参考文献

- [django rest framework 使用接口上传文件 - 北凉柿子 \(beiliangshizi.com\)](#)
- [Parsers - Django REST framework中文站点 \(q1mi.github.io\)](#)
- [HTTP content-type | 菜鸟教程 \(runoob.com\)](#)
- [Multipart/form-data - 简书 \(jianshu.com\)](#)
- [SMILES:一种简化的分子语言 - 简书 \(jianshu.com\)](#)
- [RDKit|一站式搞定分子读取、输出、可视化 - 简书 \(jianshu.com\)](#)
- [pysmiles: 一个用于读写SMILES表达式的python库 - DECHIN - 博客园 \(cnblogs.com\)](#)