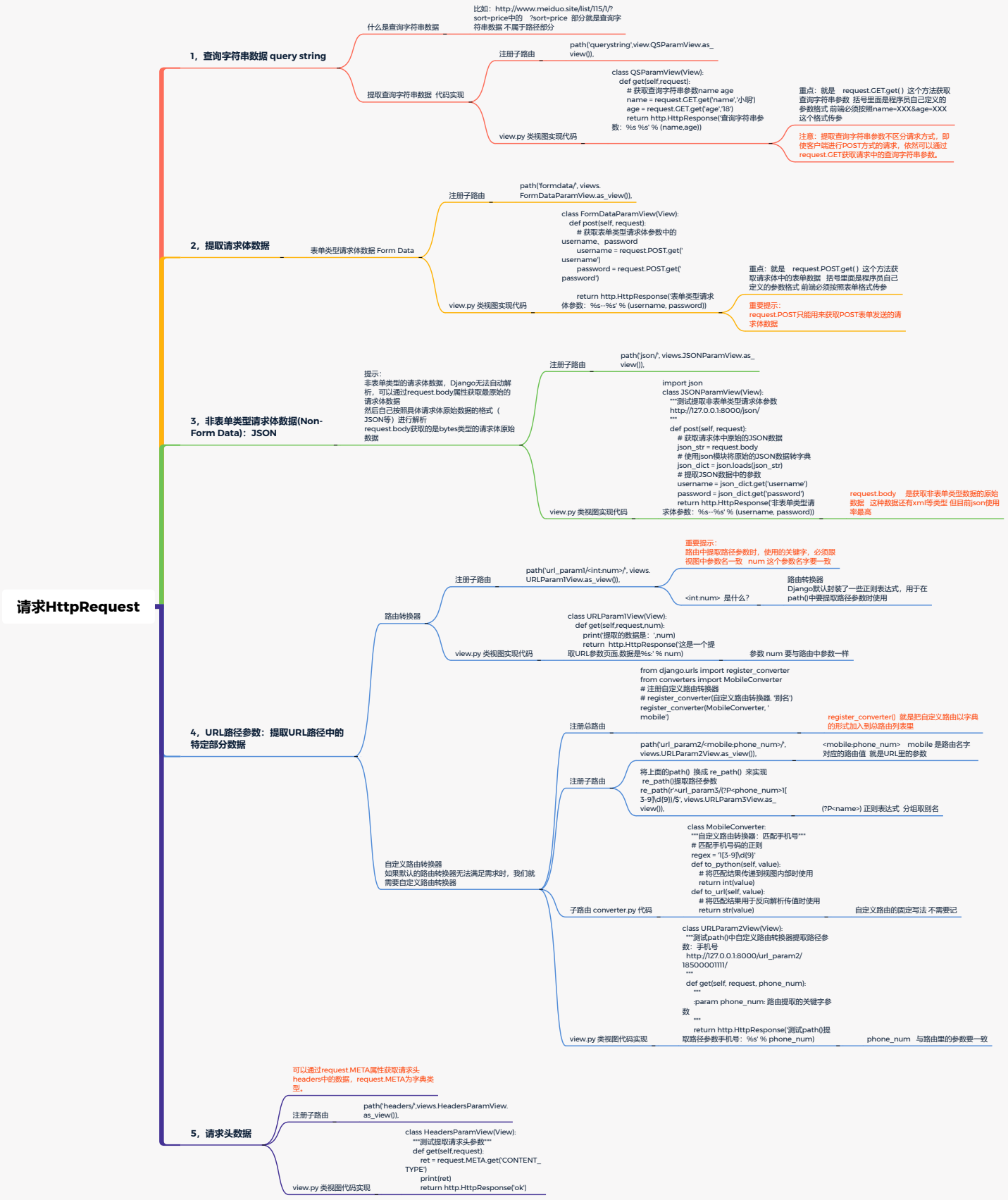
请求HttpRequest
1，查询字符串数据 query string
什么是查询字符串数据
比如：http://www.meiduo.site/list/115/1/?sort=price中的 ?sort=price 部分就是查询字符串数据 不属于路径部分
提取查询字符串数据 代码实现
注册子路由
path('querystring',view.QSParamView.as_view()),
view.py 类视图实现代码
class QSParamView(View):
def get(self,request):
# 获取查询字符串参数name age
name = request.GET.get('name','小明')
age = request.GET.get('age','18')
return http.HttpResponse('查询字符串参数：%s %s' % (name,age))
重点：就是 request.GET.get( ) 这个方法获取查询字符串参数 括号里面是程序员自己定义的参数格式 前端必须按照name=XXX&age=XXX这个格式传参
注意：提取查询字符串参数不区分请求方式，即使客户端进行POST方式的请求，依然可以通过request.GET获取请求中的查询字符串参数。
2，提取请求体数据
表单类型请求体数据 Form Data
注册子路由
path('formdata/', views.FormDataParamView.as_view()),
view.py 类视图实现代码
class FormDataParamView(View):
def post(self, request):
# 获取表单类型请求体参数中的username、password
username = request.POST.get('username')
password = request.POST.get('password')
return http.HttpResponse('表单类型请求体参数：%s--%s' % (username, password))
重点：就是 request.POST.get( ) 这个方法获取请求体中的表单数据 括号里面是程序员自己定义的参数格式 前端必须按照表单格式传参
重要提示：request.POST只能用来获取POST表单发送的请求体数据
3，非表单类型请求体数据(Non-Form Data)：JSON
提示：非表单类型的请求体数据，Django无法自动解析，可以通过request.body属性获取最原始的请求体数据然后自己按照具体请求体原始数据的格式（JSON等）进行解析request.body获取的是bytes类型的请求体原始数据
注册子路由
path('json/', views.JSONParamView.as_view()),
view.py 类视图实现代码
import json
class JSONParamView(View):
"""测试提取非表单类型请求体参数
http://127.0.0.1:8000/json/
"""
def post(self, request):
# 获取请求体中原始的JSON数据
json_str = request.body
# 使用json模块将原始的JSON数据转字典
json_dict = json.loads(json_str)
# 提取JSON数据中的参数
username = json_dict.get('username')
password = json_dict.get('password')
return http.HttpResponse('非表单类型请求体参数：%s--%s' % (username, password))
request.body 是获取非表单类型数据的原始数据 这种数据还有xml等类型 但目前json使用率最高
4，URL路径参数：提取URL路径中的特定部分数据
路由转换器
注册子路由
path('url_param1/<int:num>/', views.URLParam1View.as_view()),
重要提示：路由中提取路径参数时，使用的关键字，必须跟视图中参数名一致 num 这个参数名字要一致
<int:num> 是什么？
路由转换器 Django默认封装了一些正则表达式，用于在path()中要提取路径参数时使用
view.py 类视图实现代码
class URLParam1View(View):
def get(self,request,num):
print('提取的数据是：',num)
return http.HttpResponse('这是一个提取URL参数页面,数据是%s:' % num)
参数 num 要与路由中参数一样
自定义路由转换器
如果默认的路由转换器无法满足需求时，我们就需要自定义路由转换器
from django.urls import register_converter
from converters import MobileConverter
# 注册自定义路由转换器
# register_converter(自定义路由转换器, '别名')
register_converter(MobileConverter, 'mobile')
注册总路由
register_converter() 就是把自定义路由以字典的形式加入到总路由列表里
注册子路由
path('url_param2/<mobile:phone_num>/', views.URLParam2View.as_view()),
<mobile:phone_num> mobile 是路由名字 对应的路由值 就是URL里的参数
将上面的path() 换成 re_path() 来实现 re_path()提取路径参数
re_path(r'^url_param3/(?P<phone_num>1[3-9]\d{9})/$', views.URLParam3View.as_view()),
(?P<name>) 正则表达式 分组取别名
子路由 converter.py 代码
class MobileConverter:
"""自定义路由转换器：匹配手机号"""
# 匹配手机号码的正则
regex = '1[3-9]\d{9}'
def to_python(self, value):
# 将匹配结果传递到视图内部时使用
return int(value)
def to_url(self, value):
# 将匹配结果用于反向解析传值时使用
return str(value)
自定义路由的固定写法 不需要记
view.py 类视图代码实现
class URLParam2View(View):
"""测试path()中自定义路由转换器提取路径参数：手机号
http://127.0.0.1:8000/url_param2/18500001111/
"""
def get(self, request, phone_num):
"""
:param phone_num: 路由提取的关键字参数
"""
return http.HttpResponse('测试path()提取路径参数手机号：%s' % phone_num)
phone_num 与路由里的参数要一致
5，请求头数据
可以通过request.META属性获取请求头headers中的数据，request.META为字典类型。
注册子路由
path('headers/',views.HeadersParamView.as_view()),
view.py 类视图代码实现
class HeadersParamView(View):
"""测试提取请求头参数"""
def get(self,request):
ret = request.META.get('CONTENT_TYPE')
print(ret)
return http.HttpResponse('ok')