

# 07 Django 视图Views

## View视图

### HttpRequest 和HttpResponse

Django中的视图主要用来接受Web请求，并做出响应。

视图的本质就是一个Python中的函数

视图的响应分为两大类

- 1) 以Json数据形式返回 (JsonResponse)
- 2) 以网页的形式返回
  - 2.1) 重定向到另一个网页 (HttpResponseRedirect)
  - 2.2) 错误视图(4XX,5XX) (HttpResponseNotFound, HttpResponseForbidden, HttpResponseNotAllowed等)

视图响应过程：

浏览器输入-> urls 路由匹配 -> 视图响应 -> 回馈到浏览器

视图参数：

- 1) 一个HttpRequest的实例，一般命名为request
- 2) 通过url正则表达式传递过来的参数

位置：

通常在应用下的views.py中定义

错误视图：

- 1) 404视图 (页面没找到)
- 2) 400视图 (客户操作错误)
- 3) 500视图 (服务器内部错误)

### HttpRequest

服务器在接收到Http请求后，会根据报文创建HttpRequest对象

视图中的第一个参数就是HttpRequest对象

Django框架接收到http请求之后会将http请求包装为HttpRequest对象，之后传递给视图。

request常用属性和方法：

属性： path 请求的完整路径

method 请求的方法，常用GET,POST

GET 类似字典的参数，包含了get的所有参数

POST 类似字典的参数，包含了post所有参数

FILES 类似字典的参数，包含了上传的文件

COOKIES 字典，包含了所有COOKIE

session 类似字典，表示会话

META['REMOTE\_ADDR']

方法： is\_ajax() 判断是否是ajax()，通常用在移动端和JS中

get\_full\_path() 返回包含参数字符串的请求路径。

QueryDict： 类字典的对象

类似字典的数据结构。与字典的区别：可以存在相同的键。

QueryDict中数据获取方式

`dict['uname']` 或 `dict.get('uname')`

获取指定key对应的所有值

`dict.getlist('uname')`

## HttpResponse

服务器返回给客户端的数据

HttpResponse由程序员自己创建：

- 1) 不使用模板，直接调用`HttpResponse()`，返回`HttpResponse`对象。
- 2) 调用模板，进行渲染。

使用render

`render(request, template_name[, context])`

`request` 请求体对象

`template_name` 模板路径

`context` 字典参数，用来填坑

属性：`content` 返回的内容

`charset` 编码格式

`status_code` 响应状态码(2xx, 3xx, 4xx, 5xx)

方法：

`write(xxx)` 直接写出文本

`flush()` 冲刷缓冲区

`set_cookie(key, value='xxx', max_age=None)` 设置cookie

`delete_cookie(key)` 删除cookie

HttpResponse子类

`HttpResponseRedirect`

响应重定向：可以实现服务器内部跳转

`return HttpResponseRedirect('/grade/2030')`

使用的时候推荐使用反向解析

`JsonResponse`

返回Json数据的请求，通常用在异步请求上

`JsonResponse (dict)`

返回json数据时，`Content-type`是`application/json`