s14day21

上节内容回顾:

1、请求周期

url> 路由 > 函数或类 > 返回字符串或者模板语言?

Form 表单提交:

提交 -> url > 函数或类中的方法

- ....
HttpResponse('....')
render(request,'index.html')
redirect('/index/')

用户 < < 返回字符串

(当接受到 redirect 时)自动发起另外一个请求

--> url .....

Ajax:

```
$.ajax({
    url: '/index/',
    data: {'k': 'v', 'list': [1,2,3,4], 'k3': JSON.stringfy({'k1': 'v'})}, $(form 对象).serilize()
    type: 'POST',
    dataType: 'JSON':
    traditional: true,
    success: function(d){
        location.reload()                    # 刷新
        location.href = "某个地址"           # 跳转
    }
})
```

提交 -> url -> 函数或类中的方法

HttpResponse('{}')
render(request, 'index.html', {'name': 'v1'})
<h1>{{ name }}</h1> -->
<h1>v1</h1>

XXXXXXX redirect...

用户 <<<<< 字符串

2、路由系统 URL

a. /index/ -> 函数或类
b. /index/(\d+) -> 函数或类
c. /index/(?P<nid>\d+) -> 函数或类
d. /index/(?P<nid>\d+) name='root' -> 函数或类

reverse()
{% url 'root' 1%}

e. /crm/ include('app01.urls') -> 路由分发

f. 默认值

url(r'^index/', views.index, {'name': 'root'}),

```
        def index(request,name):
            print(name)
            return HttpResponse('OK')
```

g. 命名空间

```
        /admin/       include('app01.urls',namespace='m1')
        /crm/         include('app01.urls',namespace='m1')

        app01.urls
        /index/       name = 'n1'



        reverser('m1:n1')
```

3、

```
    def func(request):
        request.POST
        request.GET
        request.FILES
        request.getlist
        request.method
        request.path_info

        return render,HttpResponse,redirect
```

4、

```
    render(request, 'index.html')
    # for
    # if
    #  索引.     keys    values items       all
```

5、

```
    class User(models.Model):
        username = models.CharField(max_length=32)
        email = models.EmailField()

    有验证功能
        Django Admin
    无验证功能:
        User.objects.create(username='root',email='asdfasdfasdfasdf')
        User.objects.filter(id=1).update(email='666')



    class UserType(models.Model):
        name = models.CharField(max_length=32)
```

```
class User(models.Model):
    username = models.CharField(max_length=32)
    email = models.EmailField()
    user_type = models.ForeignKey("UserType")

user_list = User.objects.all()
for obj user_list:
    obj.username,obj.email,obj.user_type_id,obj.user_type.name,obj.user_type.id

user = User.objects.get(id=1)
user.

User.objects.all().values("username","user_type__name",)



class UserType(models.Model):
    name = models.CharField(max_length=32)


class User(models.Model):
    username = models.CharField(max_length=32)
    email = models.EmailField()
    user_type = models.ForeignKey("UserType")
    m = models.ManyToMany('UserGroup')


class UserGroup(models.Model):
    name = ....


obj = User.objects.get(id=1)
obj.m.add(2)
obj.m.add(2,3)
obj.m.add(*[1,2,3])

obj.m.remove(...)

obj.m.clear()


obj.m.set([1,2,3,4,5])

# 多个组，UserGroup 对象
obj.m.all()
obj.m.filter(name='CTO')
```

知识点：

URL

- 两个

Views

- 请求的其他信息

from django.core.handlers.wsgi import WSGIRequest

request.environ

request.environ['HTTP_USER_AGENT']

- 装饰器

FBV:

```
def auth(func):
    def inner(reqeust,*args,**kwargs):
        v = reqeust.COOKIES.get('username111')
        if not v:
            return redirect('/login/')
        return func(reqeust, *args,**kwargs)
    return inner
```

CBV:

```
from django import views
from django.utils.decorators import method_decorator

@method_decorator(auth,name='dispatch')
class Order(views.View):

    # @method_decorator(auth)
    # def dispatch(self, request, *args, **kwargs):
    #     return super(Order,self).dispatch(request, *args, **kwargs)

    # @method_decorator(auth)
    def get(self,reqeust):
        v = reqeust.COOKIES.get('username111')
        return render(reqeust,'index.html',{'current_user': v})

    def post(self,reqeust):
        v = reqeust.COOKIES.get('username111')
        return render(reqeust,'index.html',{'current_user': v})
```

Templates

- 母版...html

  extends

  include

- 自定义函数

  simple_tag

  a. app 下创建 templatetags 目录

  b. 任意 xxoo.py 文件

  c. 创建 template 对象 register

  d.

```
@register.simple_tag
```

def func(a1,a2,a3....)
return "asdfasd"

e. settings 中注册 APP

f. 顶部 {% load xxoo %}

g. {% 函数名 arg1 arg2 %}

缺点：

不能作为 if 条件

优点：

参数任意

filter

a. app 下创建 templatetags 目录

b. 任意 xxoo.py 文件

c. 创建 template 对象 register

d.

@register.filter

def func(a1,a2)

return "asdfasd"

e. settings 中注册 APP

f. 顶部 {% load xxoo %}

g. {{ 参数 1|函数名:"参数二，参数三" }} {{ 参数 1|函数名:数字 }}

缺点：

最多两个参数，不能加空格

优点：

能作为 if 条件

分页（自定义的分页）

XSS:

{{ page_str|safe }}

mark_safe(page_str)

cookie

客户端浏览器上的一个文件

{"user": 'dachengzi'}

session ： 装饰器

Models

- 一大波操作

Form 验证

-

缓存

中间件

信号

CSRF

Admin/ModelForm

作业：

主机管理：

1、单表操作

2、一对多

3、多对多

要求：

a. 删除对话框

b. 修改，添加新 URL

c. 基于 cookie 进行用户认证

d. 定制显示个数

e. 分页

预习：

Form： http://www.cnblogs.com/wupeiqi/articles/6144178.html

Model： http://www.cnblogs.com/wupeiqi/articles/6216618.html