# 验证码

目的：防止暴力请求，减轻业务服务器、数据库压力

实现：

1. 安装Pillow包

pip install Pillow==3.4.1

1. 创建视图

from PIL import Image, ImageDraw, ImageFont

from django.utils.six import BytesIO

...

def verify\_code(request):

#引入随机函数模块

import random

#定义变量，用于画面的背景色、宽、高

bgcolor = (random.randrange(20, 100), random.randrange(

20, 100), 255)

width = 100

height = 25

#创建画面对象

im = Image.new('RGB', (width, height), bgcolor)

#创建画笔对象

draw = ImageDraw.Draw(im)

#调用画笔的point()函数绘制噪点

for i in range(0, 100):

xy = (random.randrange(0, width), random.randrange(0, height))

fill = (random.randrange(0, 255), 255, random.randrange(0, 255))

draw.point(xy, fill=fill)

#定义验证码的备选值

str1 = 'ABCD123EFGHIJK456LMNOPQRS789TUVWXYZ0'

#随机选取4个值作为验证码

rand\_str = ''

for i in range(0, 4):

rand\_str += str1[random.randrange(0, len(str1))]

#构造字体对象，ubuntu的字体路径为“/usr/share/fonts/truetype/freefont”

font = ImageFont.truetype('FreeMono.ttf', 23)

#构造字体颜色

fontcolor = (255, random.randrange(0, 255), random.randrange(0, 255))

#绘制4个字

draw.text((5, 2), rand\_str[0], font=font, fill=fontcolor)

draw.text((25, 2), rand\_str[1], font=font, fill=fontcolor)

draw.text((50, 2), rand\_str[2], font=font, fill=fontcolor)

draw.text((75, 2), rand\_str[3], font=font, fill=fontcolor)

#释放画笔

del draw

**#存入session，用于做进一步验证**

request.session['verifycode'] = rand\_str

#内存文件操作

**验证时用**

buf = BytesIO()

#将图片保存在内存中，文件类型为png

im.save(buf, 'png')

#将内存中的图片数据返回给客户端，MIME类型为图片png

return HttpResponse(buf.getvalue(), 'image/png')

1. 配置url

url(r'^verify\_code/$', views.verify\_code),

1. 输入http://127.0.0.1:8000/verify\_code

