

Django+验证码 功能

参考文献:

<https://django-simple-captcha.readthedocs.io/en/latest/usage.html>
--django-simple-captcha官网 文档
https://blog.csdn.net/qq_37648632/article/details/83149803
----博客文档
极验验证 顶象 都在做滑动验证 很多

流程

1. 安装 django-simple-captcha

```
pip install django-simple-captcha
```

2. 注册captcha 在 APP中

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'captcha',  
]
```

2. 也可以个性化定制, 在 settings.py中添加如下代码, 详情参考官方文档:

```
# 设置 captcha 图片大小  
CAPTCHA_IMAGE_SIZE = (80, 45)  
# 字符个数  
CAPTCHA_LENGTH = 4  
# 超时(minutes)  
CAPTCHA_TIMEOUT = 1  
  
# django_simple_captcha 验证码配置其他配置项查看文档  
# 默认格式  
CAPTCHA_OUTPUT_FORMAT = '%(image)s %(text_field)s %(hidden_field)s',  
  
CAPTCHA_NOISE_FUNCTIONS = ('captcha.helpers.noise_null', # 没有样式  
    # 'captcha.helpers.noise_arcs', # 线  
    # 'captcha.helpers.noise_dots', # 点  
)  
# 图片中的文字为随机英文字母, 如 mdsh  
CAPTCHA_CHALLENGE_FUNCT = 'captcha.helpers.random_char_challenge'  
# 图片中的文字为数字表达式, 如2+2=  
#CAPTCHA_CHALLENGE_FUNCT = 'captcha.helpers.math_challenge'
```

```
# 超时(minutes)
CAPTCHA_TIMEOUT = 1
```

3. 执行数据迁移，生成数据表 captcha_captchastore

```
python manage.py migrate
```

4. 添加路由

```
urlpatterns = [
    path('admin/', admin.site.urls),
    # 图片验证码 路由
    path('captcha/', include('captcha.urls'))
]
```

再添加ajax刷新请求的二级路由，ajxa请求地址： captcha/refresh_captcha/

```
path('refresh_captcha/', views.refresh_captcha), # 刷新验证码, ajax
path("img_check/", views.img_check, name='img_check') #验证验证码
```

5. 在views.py中添加代码，

```
# 验证码需要导入的模块
from captcha.models import CaptchaStore
from captcha.helpers import captcha_image_url
# 验证码刷新
def img_refresh(request):
    if not request.is_ajax():
        return HttpResponse('不是Ajax请求')
    new_key = CaptchaStore.generate_key()
    to_json_response = {
        'hashkey': new_key,
        'image_url': captcha_image_url(new_key),
    }
    return HttpResponse(json.dumps(to_json_response))

# 验证
def img_check(request):
    print('验证用户输入的图片验证码...')
    if request.is_ajax():
        cs =
CaptchaStore.objects.filter(response=request.GET.get('code'),

hashkey=request.GET.get('hashkey'))
    print(cs)
    if cs:
        json_data = {'status': 1}
    else:
        json_data = {'status': 0}
    return JsonResponse(json_data)
else:
    # raise Http404
    json_data = {'status': 0}
    return JsonResponse(json_data)
```

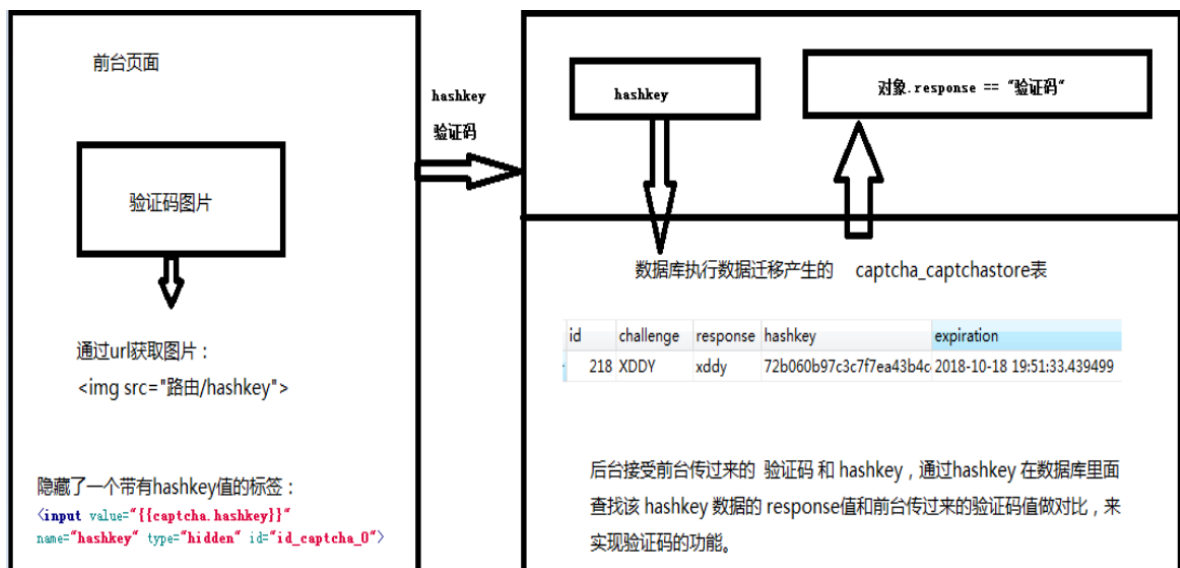
6. 前端

```
<li class="col-xs-12 col-sm-12 col-md-12 col-lg-12" >
    验证码:{{ register_form.captcha }}
    <b id="msg"></b>
</li>
```

2. ajax

```
<script type="text/javascript">
    //刷新图片验证码
    $(function(){
        //刷新
        $('#captcha').click(function(){
            $.getJSON("/img_refresh/",function(result){
                $('#captcha').attr('src',
result['image_url']);
                $('#id_captcha_0').val(result['hashkey'])
            });
            return false;
        });
        //验证
        $('#id_captcha_1').blur(function(){
            console.log('用户输入
的:'+$('#id_captcha_1').val());
            // 获取输入框和隐藏字段id_captcha_0的数值
            code= $('#id_captcha_1').val();
            hashkey=$('#id_captcha_0').val();
            $.getJSON('/img_check/',
            {'code':code, 'hashkey':hashkey},function(res){
                //1 验证码正确,0验证码错误
                if(res.status){
                    $(".reg_sub :input").attr("type","submit");
                    $("#msg").html('验证码正确')
                }else{
                    $(".reg_sub :input").attr("type","button");
                    $("#msg").html('验证码错误')
                }
            })
        })
    })
</script>
```

原理



极验验证

- 1.官网直接注册账号申请key和value
www.geetest.com 极验滑动验证码官网
- 2.从[Github: gt3-python-sdk](#)下载 .zip 文件
- 3.找到里面Django的文件夹对照官网直接copy
- 4.上代码--基本是copy，少数需自己修改

代码示例

view

```
# 请在官网申请ID使用，示例ID不可使用
pc_geetest_id = "b46d1900d0a894591916ea94ea91bd2c"
pc_geetest_key = "36fc3fe98530eea08dfc6ce76e3d24c4"

# 处理极验 获取验证码的视图
def get_geetest(request):
    user_id = 'test'
    gt = GeetestLib(pc_geetest_id, pc_geetest_key)
    status = gt.pre_process(user_id)
    request.session[gt.GT_STATUS_SESSION_KEY] = status
    request.session["user_id"] = user_id
    response_str = gt.get_response_str()
    return HttpResponse(response_str)

-----
urls
# 极验滑动验证码 获取验证码的url
url(r'^pc-geetest/register', views.get_geetest),
```

HTML

```

// 极验滑动验证码
var handlerPopup = function (captchaObj) {
    // appendTo方法用于将验证按钮插到宿主页面，使其显示在页面上。
    captchaObj.appendTo('#popup-captcha');
    // onReady监听验证按钮的 DOM 生成完毕事件
    captchaObj.onReady(function () {
        $("#wait").attr('class', 'hidden');
    });

    // onSuccess:监听验证成功事件，进行二次验证
    captchaObj.onSuccess(function () {
        $("#login-button").removeAttr('disabled');
        $("#login-button").removeAttr('title');
    });

    $("#login-button").click(function () {
        // getValidate:获取用户进行成功验证(onSuccess)所得到的结果，该结果用于进行服务端
        SDK 进行二次验证。
        var validate = captchaObj.getValidate();
        if (!validate) {
            return 0;
        } else {
            $.ajax({
                url: "/login/", // 进行二次验证
                type: "post",
                dataType: "json",
                data: {
                    username: $('input[name="user"]').val(),
                    password: $('input[name="pwd"]').val(),
                    csrfmiddlewaretoken: $("
                    [name='csrfmiddlewaretoken']").val(),

                    geetest_challenge: validate.geetest_challenge,
                    geetest_validate: validate.geetest_validate,
                    geetest_seccode: validate.geetest_seccode,
                },
                success: function (data) {
                    if (data.status) {
                        $('#error').text(data.msg);
                        captchaObj.reset(); // 让验证回到初始状态
                    } else {
                        location.href = data.msg;
                    }
                },
            });
        }
    });
};

// 当input框获取焦点时将之前的错误清空
$('input[name="user"],input[name="pwd"]').focus(function () {
    // 将之前的错误清空
    $('#error').text('');
});

// 验证开始需要向网站主后台获取id, challenge, success (是否启用failback)
$.ajax({
    url: "/pc-geetest/register?t=" + (new Date()).getTime(), // 加随机数防止缓存

```

```
type: "get",
dataType: "json",
success: function (data) {
    // 使用initGeetest接口
    // 参数1: 配置参数
    // 参数2: 回调, 回调的第一个参数验证码对象, 之后可以使用它做appendTo之类的事件
    initGeetest({
        gt: data.gt, // 验证 id, 极验后台申请得到
        challenge: data.challenge, // 验证流水号, 后服务端 SDK 向极验服务器申请得
        product: "popup", // 产品形式, 包括: float, embed, popup。注意只对PC版验证
        offline: !data.success, // 表示用户后台检测极验服务器是否宕机, 一般不需要关
        width: '100%', // 设置按钮的长度
        // 更多配置参数请参见: http://www.geetest.com/install/sections/idx-
        client-sdk.html#config
    }, handlerPopup);
},
});
```