## How to Implement Google Recaptcha V3 in Django class view

**Dependencies:**

Python 3.7, Django 2.2

**Customize Django bulit-in password_reset view and password_reset form**

Let’s suppose the app name is “landing” which takes over all password-reset request.

*In urls.py*

```python
from landing import views

urlpatterns = [
    # other urls
    path('password_reset/',
views.PasswordResetGoogleRecaptchaView.as_view(), name='password_reset')
]
```

*In views.py*

```python
from django.contrib.auth.views import PasswordResetView
from landing.forms import PasswordResetGoogleRecaptchaForm

class PasswordResetGoogleRecaptchaView(PasswordResetView):
    form_class = PasswordResetGoogleRecaptchaForm
```

Notice that we inherit Django builit-in view **PasswordResetView** and let it use our custom **PasswrodResetGoogleRecaptchaForm** declared in *forms.py*.

*In forms.py*

```python
from django.contrib.auth.forms import PasswordResetForm

class PasswordResetGoogleRecaptchaForm(PasswordResetForm):
    recaptcha = forms.CharField(
        widget=forms.HiddenInput(),
        max_length=1024,
        required=False
    )
    def clean(self):
        cleaned_data = super(PasswordResetGoogleRecaptchaForm,
self).clean()
        recaptcha_token = cleaned_data.get('recaptcha')
        recaptcha_data = {
            'secret': settings.GOOGLE_RECAPTCHA_SECRET_KEY,
            'response': recaptcha_token
        }
        r =
requests.post('https://www.google.com/recaptcha/api/siteverify',
data=recaptcha_data)
        result = r.json()
        if result.get('success') and result.get('score') > 0.5:
            # client is human
            print(result)
            pass
        else:
            raise forms.ValidationError('You are robot!')
```

Our form is built by inheriting Django builit-in PasswordResetForm. In the form, we add a new field called "recaptcha" which is responsible to pass Recaptcha token over to back-end. Notice that using HiddenInput in "recaptcha" field makes itself hidden visually from client browser.

We also add a clean() function in which all the magic happens. It returns result from Google recaptcha api by using which we can build our logic to determine if it is a robot or not.

You may check what the expected result here.
https://developers.google.com/recaptcha/docs/v3#site_verify_response

**Customize password_reset_form.html**

By default, our "PasswordResetGoogleRecaptchaView" will render content in templates/registration/ password_reset_from.html.

```
<form id="reset-password-form" method="POST">
    {% csrf_token %}
    {{ form.non_field_errors }}
    <div class="input-group mb-3">
        <input type="email" class="form-control" placeholder="Email"
id="id_email" name="email" maxlength="254"
               required="" autofocus>
        <div class="input-group-append">
            <div class="input-group-text">
                <span class="fas fa-envelope"></span>
            </div>
        </div>
    </div>
    {{ form.recaptcha }}
    <div class="row">
        <input type="submit" class="btn btn-primary btn-block" value="Send
me instructions!">
    </div>
</form>

<script src="{% static 'js/jquery.min.js' %}"></script>
<script
src="https://www.google.com/recaptcha/api.js?render=YOUR_GOOGLE_RECAPTCHA_
SITE_KEY"></script>
<script>
  $('document').ready(function submitResetForm() {
    grecaptcha.ready(function () {
      grecaptcha.execute("YOUR_GOOGLE_RECAPTCHA_SITE_KEY", {action:
'homepage'}).then(function (token) {
          $('input#id_recaptcha').val(token);
      });
    });
  });
</script>
```

In the html file, all we need to do is to display form, integrate recaptcha javascript and fill token in recaptcha field. Behind the scenes, $('input#id_recaptcha').val(token) is the key piece of codes that fills token in recaptcha field. If you would like to enable google Recaptcha site key in multiple pages, passing it as global variable will be a better solution.