



How to add Google reCAPTCHA to Django forms ?

Last Updated : 30 May, 2021

This tutorial explains to integrate Google's reCaptcha system to your Django site. To create a form in Django you can check out – [How to create a form using Django Forms ?](#)

Getting Started

Adding reCaptcha to any HTML form involves the following steps:

1. Register your site domain to reCaptcha Admin Console.
2. Add recaptcha keys to project settings.
3. Add reCaptcha scripts and input element to HTML template.
4. On form submission, make a post request to google reCaptcha API in the backend with the form's reCaptcha field value and recaptcha keys as the POST data.
5. Process the response from Google.

While this can be done manually, we will be using a third-party library as it makes the process much faster and simpler. Now let's make a sample contact form where we will integrate the reCaptcha

Initiate the Django project:

Make sure you are done with the django installation.

- Create a new django project :

```
django-admin startproject di_recaptcha
```

Django Views Model Template Forms Jinja Python SQLite Flask Json Postman Interview Ques

- Create a new app say "contact" :

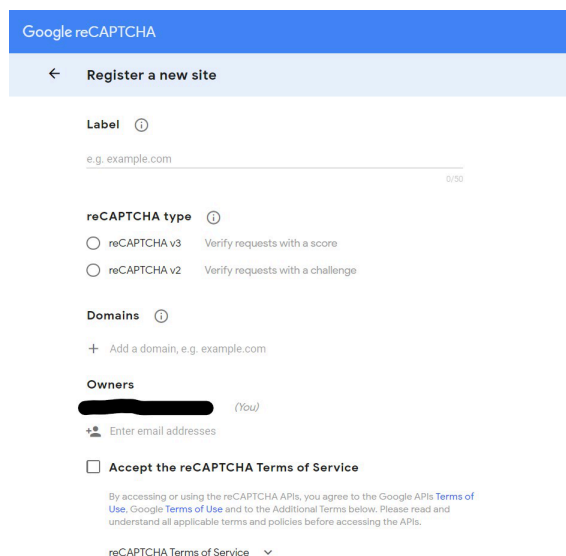
```
cd dj_recaptcha
python manage.py startapp contact
```

- Go to dj_recaptcha/settings.py add the contact app.

```
INSTALLED_APPS = [
    ...
    'contact',
    ...
]
```

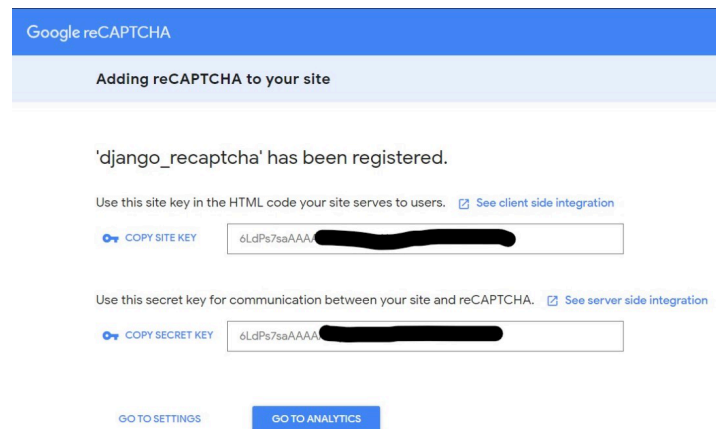
Registering to Google reCAPTCHA Admin Console:

First you need register your site on the [reCaptcha admin console](#). In the domains section add 127.0.0.1 since we are testing it out locally. Later on, you can add your production URL.

The screenshot shows the 'Register a new site' page in the Google reCAPTCHA Admin Console. The page has a blue header with 'Google reCAPTCHA' and a light blue sub-header with a back arrow and 'Register a new site'. Below this, there are several sections: 'Label' with a placeholder 'e.g. example.com' and a character count '0/50'; 'reCAPTCHA type' with two radio button options: 'reCAPTCHA v3' (selected) and 'reCAPTCHA v2'; 'Domains' with a plus icon and the text 'Add a domain, e.g. example.com'; 'Owners' with a redacted email address and the text '(You)'; and 'Accept the reCAPTCHA Terms of Service' with an unchecked checkbox and a link to the terms of service.

recaptcha admin console

You can specify whichever reCaptcha type you want, here we have selected v2 with " **I'm not a robot tickbox** ". You will get the API keys on form submission.



recaptcha api keys

Copy the site key and secret key into settings.py as follows:

```
RECAPTCHA_PUBLIC_KEY = Your_Site_Key
RECAPTCHA_PRIVATE_KEY = Your_Secret_key
```

As mentioned before we will be using a third-party library called django-recaptcha to make the process simpler for us. Let's just install it now using pip, enter the following command.

```
pip install django-recaptcha
```

Add the app to the INSTALLED_APPS list in settings.py

```
INSTALLED_APPS = [
    ...
    'contact',
    'captcha',
    ...
]
```

Now let's move on to creating a contact form in forms.py with email, feedback, and captcha as the fields.

Python3

```
# forms.py
```

```

from django import forms
from captcha.fields import ReCaptchaField
from captcha.widgets import ReCaptchaV2Checkbox

class ContactForm(forms.Form):
    email = forms.EmailField()
    feedback = forms.CharField(widget=forms.Textarea)
    captcha = ReCaptchaField(widget=ReCaptchaV2Checkbox)

```

The captcha field will be rendered as a checkbox field. If you specified a different type than the v2 checkbox while registering the site on recaptcha admin console, you need to change the widget attribute of ReCaptchaField above. If you don't specify one ReCaptchaV2Checkbox will be default There are three widgets that can be used with the ReCaptchaField class:

ReCaptchaV2Checkbox for Google reCAPTCHA V2 – Checkbox

ReCaptchaV2Invisible for Google reCAPTCHA V2 – Invisible

ReCaptchaV3 for Google reCAPTCHA V3

Make an HTML template say contact.html to render the form. We will be using the default styling with `{{ form.as_p }}`. If you want to manually style a Django form try [using widget tweaks](#).

HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Contact</title>
</head>
<body>
    <h2>Contact Form</h2>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Submit</button>
    </form>

```

```
</body>
</html>
```

Now create a view to handle the form submission in views.py.

Python3

```
# views.py
from django.shortcuts import render, HttpResponseRedirect
from .forms import ContactForm

def contact(request):
    if request.method == 'POST':
        form = ContactForm(request.POST)

        if form.is_valid():
            return HttpResponseRedirect("Yay! you are human.")
        else:
            return HttpResponseRedirect("OOPS! Bot suspected.")

    else:
        form = ContactForm()

    return render(request, 'contact.html', {'form':form})
```

Map the view to a URL. Here we are rendering the form on the homepage so let's add the URL mapping for the same.

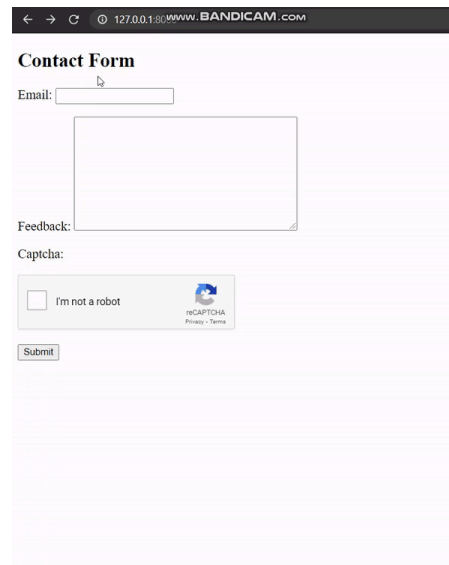
Python3

```
# urls.py
from django.contrib import admin
from django.urls import path
from contact import views

urlpatterns = [
    path('', views.contact, name='index'),
    path('admin/', admin.site.urls),
]
```

We are pretty much done, lets test it out by running the server and open 127.0.0.1:8000 in the browser.

```
python manage.py runserver
```

A screenshot of a web browser window. The address bar shows '127.0.0.1:8000/www.BANDICAM.COM'. The page title is 'Contact Form'. The form contains an 'Email:' label followed by a text input field. Below that is a 'Feedback:' label followed by a large text area. Underneath the text area is a 'Captcha:' label. The captcha widget includes a checkbox labeled 'I'm not a robot' and the reCAPTCHA logo with links for 'Privacy' and 'Terms'. At the bottom of the form is a 'Submit' button.

recaptcha demo

Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our [Mastering Django Framework - Beginner to Advanced Course](#) on GeeksforGeeks, designed for aspiring developers and experienced programmers. This comprehensive course covers everything you need to know about Django, from the basics to advanced features. Gain practical experience through **hands-on projects** and real-world applications, mastering essential Django principles and techniques. Whether you're just starting or looking to refine your skills, this course will empower you to build sophisticated web applications efficiently. Ready to enhance your web development journey? Enroll now and unlock your potential with Django!

P pans...



4

Previous Article

Determine The Face Tilt Using OpenCV - Python

Next Article

Youtube video downloader using Django

Similar Reads

Styling Django Forms with django-crispy-forms

Django by default doesn't provide any Django form styling method due to which it takes a lot of effort and precious time to beautifully style a form. django-crispy...

1 min read

How to customize Django forms using Django Widget Tweaks ?

Django forms are a great feature to create usable forms with just few lines of code. But Django doesn't easily let us edit the form for good designs. Here, we...

3 min read

Add Forms Dynamically In Django Using Formset And JavaScript

Formsets in Django allow us to manage a collection of forms as a single unit. They are particularly useful when we need to handle multiple forms on a single...

8 min read

How to Add HTML Attributes to Input Fields in Django Forms?

Django provides an easy-to-use system for creating and managing forms through its forms.Form and forms.ModelForm classes. These classes allow us to create...

4 min read

Render HTML Forms (GET & POST) in Django

Django is often called "Batteries Included Framework" because it has a default setting for everything and has features that can help anyone develop a website...

5 min read

How to create a form using Django Forms ?

Django forms are an advanced set of HTML forms that can be created using python and support all features of HTML forms in a pythonic way. This post...

3 min read

{{ form.as_p }} - Render Django Forms as paragraph

Django forms are an advanced set of HTML forms that can be created using python and support all features of HTML forms in a pythonic way. Rendering...

2 min read

{{ form.as_table }} - Render Django Forms as table

Django forms are an advanced set of HTML forms that can be created using python and support all features of HTML forms in a pythonic way. Rendering...

2 min read

{{ form.as_ul }} - Render Django Forms as list

Django forms are an advanced set of HTML forms that can be created using python and support all features of HTML forms in a pythonic way. Rendering...

2 min read

CharField - Django Forms

CharField in Django Forms is a string field, for small- to large-sized strings. It is used for taking text inputs from the user. The default widget for this input is...

5 min read

Article Tags :[Python](#)[Django-Projects](#)[Python Django](#)**Practice Tags :**[python](#)



Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305



Company

About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

Interview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation

OOAD

Puzzles

System Design Bootcamp

Interview Questions

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved