

In this video tutorial we will learn how to make an invisible reCaptcha 3 using the Google API.

For those who do not know: CAPTCHA is an acronym for the expression "Completely Automated Public Turing test to tell Computers and Humans Apart" (test of fully automated public for differentiation between computers and humans). It serves to protect your system against automatic loops of spam systems, for example malicious systems to test passwords in a login system, or to log successively on a form automatically.

The first step is to access the reCaptcha website by [clicking here](#).

Registering on Google's reCaptcha

We need to first register our domain in the recaptcha, for this just follow the tutorial [clicking here](#). For V 3.0 : <https://www.google.com/recaptcha/admin#v3signup>

After registering, you will have two important codes: the SITE_KEY that runs on the client side and the SECRET_KEY that is to be used on the server side. With this data we will create the Config.php file including this data in two constants:

code source

```
1. <? php
2. # client-side
3. define ( 'SITE_KEY' , 'YOUR SITE KEY' );
4.
5. # server-side
6. define ( 'SECRET_KEY' , 'YOUR SECRET KEY' );
```

Configuring reCaptcha client side

Later we need to create a test form and install the recaptcha javascript, as per the code below:

code source

```
1. <body>
2.   <form id = "Form1" name = "Form1" action = "controller
3.     <input type = "text" id = "Name" name = "Name"> <b
4.     <input type = "hidden" id = "g-recaptcha-response"
5.     <input type = "submit">
6.   </ form>
7.
8.   <script src = "https: //www.google.com/recaptcha/api.j
9.   <script src = "libraries / Javascript.min.js"> </ scri
10. </ body>
```

We need to call the grecaptcha.execute function by passing our SITE_KEY:

code source

```
1. grecaptcha. ready ( function () {
2.   grecaptcha. execute ( 'SUA SITE KEY' , { action: 'home
3.   . then ( function (token) {
```

```
4. document.getElementById ( 'g-recaptcha-respon  
5. } );  
6. } );
```

Configuring reCaptcha server side

We will create two files: ControllerForm.php that will receive the form data and instantiate the second file: ClassCaptcha.php.

ControlllerForm file:

code source

```
1. include ("../ class /ClassCaptcha.php");  
2.  
3. $ ObjCaptcha = new Captcha ();  
4. $ Return = $ ObjCaptcha-> getCaptcha ($ _POST [ 'g-recaptc  
5.  
6. var_dump ($ Return);  
7. if ($ Return-> success == true && $ Return-> score> 0.5) {  
8.     echo "worked out";  
9. } else {  
10.     echo "you are a robot";  
11. }
```

ClassCaptcha File:

code source

```
1. include ("../config/Config.php");  
2. class Captcha {  
3.  
4.     public function getCaptcha ($ SecretKey)  
5.     {  
6.         $ Answer = file_get_contents ("https: //www.google  
7.         $ Return = json_decode ($ Response);  
8.         return $ Return;  
9.     }  
10. }
```

That's the way to make an invisible captcha with Google's reCaptcha.

Invisible Recaptcha v3 do Google

<https://www.youtube.com/watch?v=w6WF9rw6-3c&feature=youtu.be>



ruslan-website.com/recaptcha/3/ (ruslan-website.com)

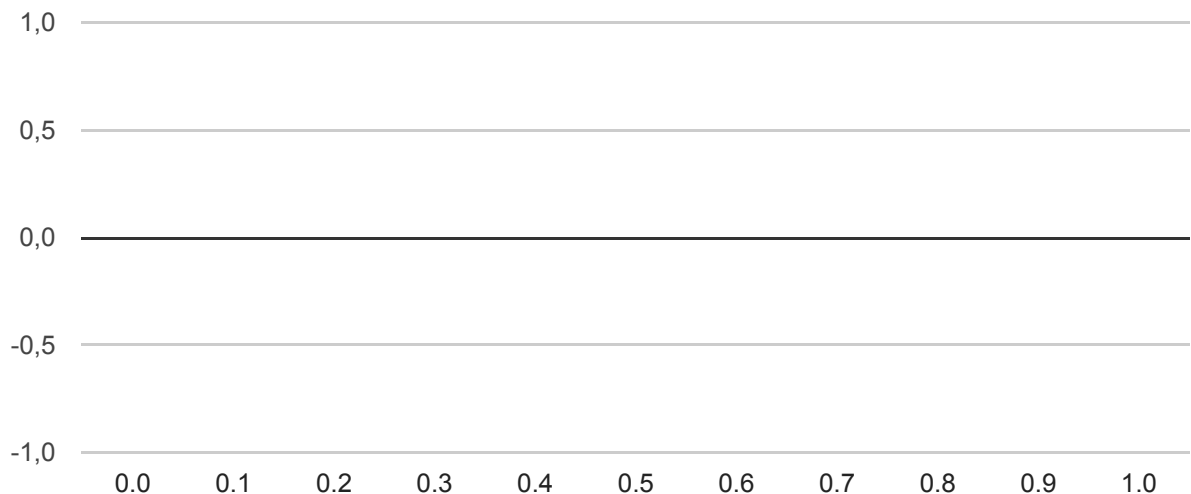
**Analytics**

7 days from: ▾

10/13/2018

Top 10 Actions ?

To view a more detailed breakdown of your traffic, add [action tags](#) to your site.

Number of requests ?Score distribution ?

i Adding reCAPTCHA to your site

[Read the setup instructions](#) or view this site on a larger screen.

▼ Keys

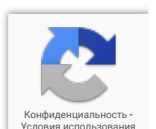
Site key

Use this in the HTML code your site serves to users.

```
6Lc50HUUAAAAAKg-bgWMsW2UB7wWcyYRiLsKVSRf
```

Secret key

Use this for communication between your site and Google. Be sure to keep it a secret.





ruslan-website.com



Paste this snippet before the closing `</head>` tag on your HTML template:

```
<script src='https://www.google.com/recaptcha/api.js?render=6Lc50HUUAAAAAKg-bgWMsW2UB7wWcyYRiLsKVSRf'></script>
```

Paste this snippet in the JavaScript callback for an action on your website.

```
<script>
  grecaptcha.ready(function() {
    grecaptcha.execute('6Lc50HUUAAAAAKg-bgWMsW2UB7wWcyYRiLsKVSRf', {action:
    'action_name'})
      .then(function(token) {
        ...
      });
  });
</script>
```

The [reCAPTCHA documentation site](#) provides details on client side integration for v3.

▼ Step 2: Server side integration

URL: <https://www.google.com/recaptcha/api/siteverify>

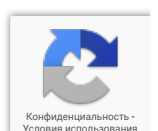
secret (required)	6Lc50HUUAAAAAD9ajK2cUozV4Fm5NcGxa-iOjsKf
response (required)	The value of 'g-recaptcha-response'.
remoteip	The end user's ip address.




The [reCAPTCHA documentation site](#) describes more details and advanced configurations.







Key Settings

Delete Key





ruslan-website.com


Domain
(one per line)

ruslan-website.com

Owners
(one per line)

steppe.ego@gmail.com

[Advanced Settings](#)

☒ Send alerts to owners  [Discard changes](#) [Save changes](#)

reCAPTCHA v3 returns a score for each request without user friction. The score is based on interactions with your site and enables you to take fine-grained action. Register reCAPTCHA v3 keys [here](#).

This page explains how to enable and customize reCAPTCHA v3 on your webpage.

Frontend integration

```
<script src="https://www.google.com/recaptcha/api.js?render=reCAPTCHA_site_key">
</script>
greaptcha.ready(function() {
  greaptcha.execute('reCAPTCHA_site_key', {action: 'homepage'}).then(function() {
    ...
  });
});
</script>
```

1. Load the JavaScript api with your sitekey
2. Call greaptcha.execute on an action or when the page loads
3. Send the token to your backend with the request to [verify](#)

Tips

greaptcha.ready() will run your function as soon as the reCAPTCHA library has loaded. To avoid race conditions with the api.js, either include the api.js before your scripts that call greaptcha, or you can continue to use the [onload callback](#) defined with the v2 API.

Placement on your website

reCAPTCHA v3 will never interrupt your users, so you can run it whenever you like without affecting conversion. reCAPTCHA works best when it has the most context about interactions with your site, which comes from seeing both legitimate and abusive behavior. For this reason, we recommend including reCAPTCHA verification on forms or actions as well as in the background of pages for analytics.

Note: You can execute reCAPTCHA as many times as you'd like with different actions on the same page.

Interpreting the score

reCAPTCHA v3 returns a score (1.0 is very likely a good interaction, 0.0 is very likely a bot). Based on the score, you can take variable action in the context of your site. Every site is different, but below are some examples of how sites use the score. As in the examples below, take action behind the scenes instead of blocking traffic to better protect your site.

Use case	Recommendation
homepage	See a cohesive view of your traffic on the admin console while filtering scrapers.
login	With low scores, require 2-factor-authentication or email verification to prevent brute force attacks.
social	Limit unanswered friend requests from abusive users and send risky comments to moderation.
e-commerce	Put your real sales ahead of bots and identify risky transactions.

reCAPTCHA adapts to your use cases by seeing real traffic on your site. For this reason, scores in a staging environment may differ from production. As reCAPTCHA v3 doesn't ever interrupt the user flow, you can first run reCAPTCHA without taking action and then decide on thresholds by looking at your traffic in the [admin console](#). By default, you can use a threshold of 0.5.

Actions

reCAPTCHA v3 introduces a new concept: actions. When you specify an action name in each place you execute reCAPTCHA you enable two new features:

- a detailed break-down of data for your top ten actions in the [admin console](#)
- adaptive risk analysis based on the context of the action (abusive behavior can vary)

Importantly, when you verify the reCAPTCHA response you should also verify that the action name matches the one you expect.

```
<script>
  grecaptcha.ready(function() {
    grecaptcha.execute('reCAPTCHA_site_key', {action: 'homepage'});
  });
</script>
```

Note: actions may only contain alphanumeric characters and slashes, and must not be user-specific.

Site Verify Response

Make the request to [verify the response token](#) as with reCAPTCHA v2 or Invisible reCAPTCHA.

The response is a JSON object:

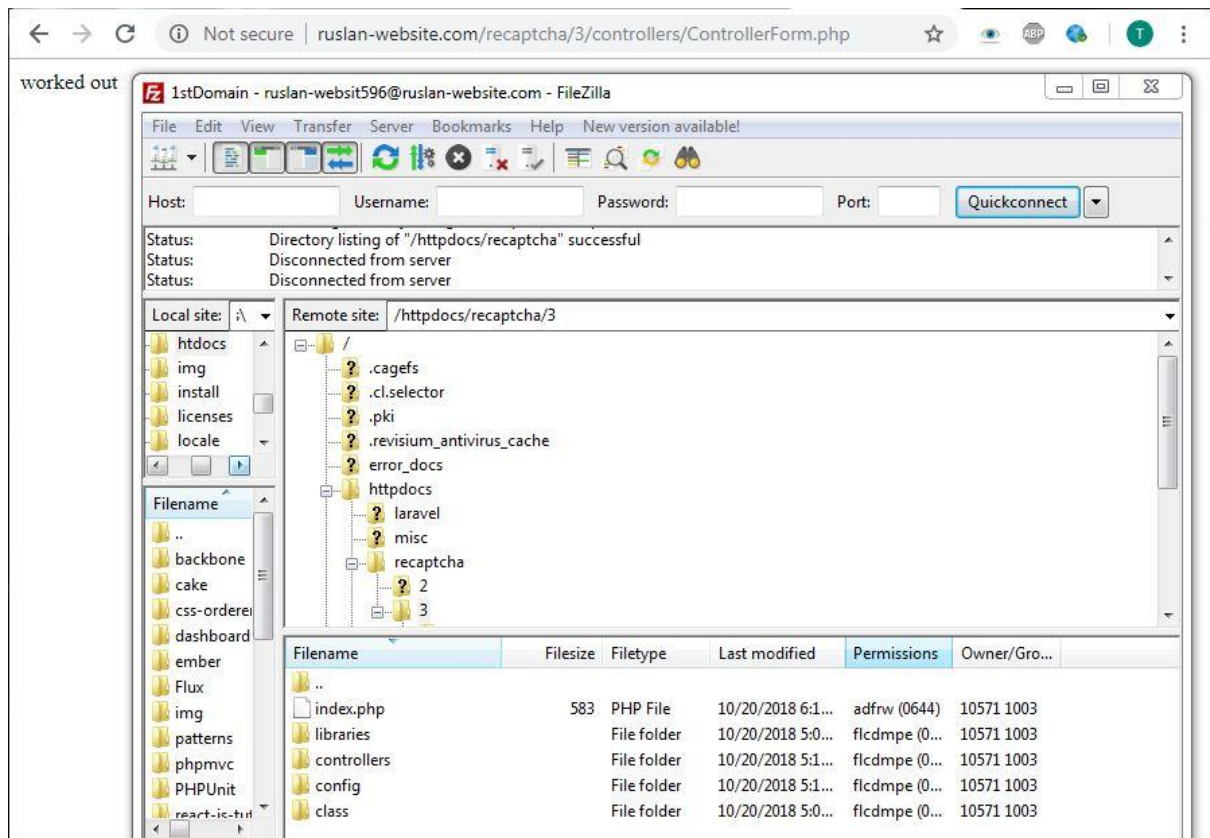
```
{
  "success": true|false,      // whether this request was a valid reCAPTCHA token
  "score": number             // the score for this request (0.0 - 1.0)
  "action": string            // the action name for this request (important to v
  "challenge_ts": timestamp,  // timestamp of the challenge load (ISO format yyyy
  "hostname": string,         // the hostname of the site where the reCAPTCHA was
  "error-codes": [...]        // optional
}
```

Another helpful site:

[php - How to Verify Google Recaptcha V3 Response - Stack Overflow](#)

<https://stackoverflow.com/questions/50405977/how-to-verify-google-recaptcha-v3-response#>

Results:



Another useful tutorial:

Google reCAPTCHA v3 - Invisible google reCAPTCHA without user friction – YouTube