



# Add Google reCAPTCHA V3 in PHP contact form

Google has released Google reCAPTCHA v3, a new and improved version of recaptcha. It improves the security of your web forms against spam bots and other sorts of abuse. The spam score is used by Google reCAPTCHA v3, which means that the reCAPTCHA v3 API delivers the spam score of each request based on the user activity.

## How to add Google reCAPTCHA V3 in PHP Contact form

In comparison to Google reCAPTCHA v2, this reCAPTCHA v3 is more easier to use because the user does not have to click on the checkbox, which is required in Google reCAPTCHA v2. It simply calculates a spam score based on the supplied data and user activity and determines whether the action is spam or not.

In this [PHP TUTORIAL](#), we'll look at how to use a contact form to implement Google reCAPTCHA v3 in PHP. If you have a contact form or any other type of form on your website and are concerned about spam, you've come to the correct spot. We recommend that you follow this guide all the way through so that you have a thorough understanding of how to defend your forms from spambot attacks.

You can see a demo of Google recaptcha v3 here, and I've prepared a functioning code so you can quickly download Google recaptcha v3 code and incorporate it into your project or application.

## Google reCAPTCHA v3 in PHP: Step to step explanation

Developers are paying greater attention to the newest version of Google reCAPTCHA v3 because it eliminates one extra click in your HTML forms. In your HTML forms, this reCAPTCHA v3 will not display any kind of captcha. This implies it is hidden and will not appear in your frontend forms on the web page; it can only be seen in the right bottom corner.

The spam score is returned by Google reCAPTCHA v3 on each request without the end-user having to do anything. Because Google reCAPTCHA v3 is transparent and does not interfere

with user engagement, we may use it anywhere on the website and in user interactions, but it works best when we are capturing user input.

To use Google reCAPTCHA V3 in a PHP Contact form, follow the steps below.

- From the Google reCAPTCHA Admin panel, generate a Site key and a Secret key.
- In PHP, create a simple contact form.
- Create a PHP file that uses the Google reCAPTCHA V3 API to validate the form.

### **In PHP, create a simple contact form.**

To demonstrate Google reCAPTCHA, we constructed a simple contact form in HTML and Bootstrap. We supply a js file that may be used to contact the Google reCAPTCHA API from an HTML form.

```
<script async src="https://www.google.com/recaptcha/api.js?render=  
<YOUR_SITE_KEY>"></script>
```

We'll replace YOUR SITE KEY> with your produced site key below. We also use recaptcha response as a hidden field, and we'll use a javascript call to replace the value of this input field during runtime. As a result, the Google reCAPTCHA API generates a one-of-a-kind key and saves it in this input field. This value will be used in the PHP file to validate the input and calculate the spam score.

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <title>Google reCAPTCH V3 Example</title>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
    <link rel="nofollow"  
href="https://justpaste.it/redirect/4ul9z/https%3A%2F%2Fmaxcdn.bootstrapcdn.com%2  
Fbootstrap%2F4.5.2%2Fcss%2Fbootstrap.min.css">  
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">  
</script>  
    <script  
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js">  
</script>
```

```

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js">
</script>
<script async src="https://www.google.com/recaptcha/api.js?render=
<YOUR_SITE_KEY>"></script>
<style>
body{background-color:#f4f8fa;}
</style>
</head>
<body>
<div class="container col-sm-5" style="background-color:#ffffff; padding:25px;
border: 1px solid #d9d8d8;">
<h1 style="font-size: 21px; font-weight: bold;">Demo of Integrate Google
reCAPTCHA V3 in PHP</h1>

<form action="" method="post">
<div id="alert_message"></div>
<div class="form-group">
<label for="pwd">Name:</label>
<input type="text" class="form-control" id="name" placeholder="Enter your
name" name="name" required>
</div>
<div class="form-group">
<label for="email">Email:</label>
<input type="text" class="form-control" id="email" placeholder="Enter your
email" name="email" required>
</div>
<div class="form-group">
<label for="email">Comment:</label>
<textarea name="comment" class="form-control" id="comment"
placeholder="Enter your comment" required></textarea>
</div>
<input type="hidden" name="recaptcha_response" value=""
id="recaptchaResponse">
<input type="submit" name="submit" value="Submit" class="btn btn-success
btn-lg" style="padding: 6px 46px; 16px 0 0 0;">
<div style="font-size: 12px; padding-top: 12px; color: #808080;">Note: This form
is protected by Google reCAPTCHA.</div>
</form>
</div>

```



```

    }
  });
});
});
</script>

```

Create a PHP file(submit\_enquiry.php)

This PHP file takes all of the information from the contact form and sends it to Google's reCAPTCHA API. You must replace YOUR SECRET KEY> with the secret key you generated. To generate the spam score, we now use the reCAPTCHA API and send these variables to it.

When you call the API, you'll get something like this:

```
{ "success": true, "challenge_ts": "2020-11-06T15:31:26Z", "hostname": "localhost", "score": 0.9, "action": "submit" }
```

```

<?php
if(isset($_POST['name']) && $_POST['name']!= "" && isset($_POST['email']) &&
$_POST['email']!= "")
{
    // This is Google API url where we pass the API secret key to validate the user request.
    $google_recaptcha_url = 'https://www.google.com/recaptcha/api/siteverify';
    $recaptcha_secret_key = '<YOUR_SECRET_KEY>'; // Add your generated Secret key
    $set_recaptcha_response = $_POST['recaptcha_response'];
    // Make the request and capture the response by making below request.
    $get_recaptcha_response = file_get_contents($google_recaptcha_url . '?secret=' .
    $recaptcha_secret_key . '&response=' . $set_recaptcha_response);

    print( $get_recaptcha_response);
    $get_recaptcha_response = json_decode($get_recaptcha_response);
    $success_msg="";
    $err_msg="";
    // Set the Google recaptcha spam score here and based the score, take your action

```

```
if ($get_recaptcha_response->success == true && $get_recaptcha_response->score
>= 0.5 && $get_recaptcha_response->action == 'submit') {
    $success_msg = "Your message has been sent successfully.";
} else {
    $err_msg = "Something went wrong. Please try again after sometime.";
}
} else {
    $err_msg = "Please enter the required fields in this form";
}
// Get the response and pass it into your ajax as a response.
$return_msg = array(
    'error' => $err_msg,
    'success' => $success_msg
);
echo json_encode($return_msg);
?>
```

Now we'll go over how to implement Google reCAPTCHA V3 to a PHP Contact form step by step. To prevent spam, I recommend using this reCAPTCHA in your web forms or in your application.

<https://www.phptpoint.com/>