

# Comment ajouter le ReCaptcha (sans plugin)

CSS, Elementor, JS, PHP

Web  Compétences : 4

**Personne n'aime les commentaires laissés sur un site web par des robots ou des gens mal intentionnées. La plupart des commentaires sont des messages dans une langue qui n'est pas celle du site, souvent en russe ou en chinois. Mais pas de panique, Google nous offre gratuitement la possibilité de contrer ces spams avec la technologie du ReCaptcha.**

Publié mardi 30 novembre 2021, 18h58
Modifié lundi 26 août 2024, 10h04



By Olivier Paudex

## Introduction

Le ReCaptcha de Google utilise un moteur d'analyse de risques avancés et de défis adaptatifs pour empêcher les logiciels malveillants de se livrer à des activités abusives sur un site web. Il permet aux utilisateurs légitimes de se connecter et d'interagir avec tous les formulaires du site ainsi que de bloquer tous les autres.

# Version 2 vs Version 3

La version 2 présentait une sorte de challenge que l'utilisateur humain devait résoudre afin de poursuivre.

La version 3 analyse le comportement humain à l'aide d'un algorithme retournant un score. Si celui-ci est inférieur à **0.5 sur 1**, la probabilité qu'il s'agisse d'un robot est très élevé et l'utilisateur sera alors bloqué.

**Les deux versions sont gratuites jusqu'à concurrence d'un million de requêtes par mois.**

> Ce site utilise la version 3 et cette publication explique en détails la marche à suivre pour activer le ReCaptcha en version 3.

# Obtenir les clés du ReCaptcha

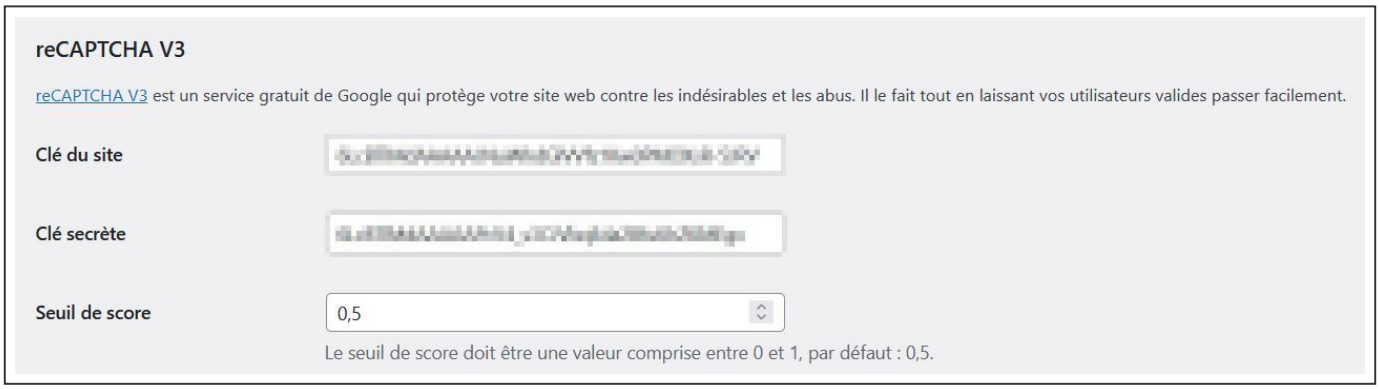
La première tâche à accomplir est de générer les clés du ReCaptcha qui seront nécessaire pour que celui-ci puisse envoyer des requêtes de validation.

- Ouvrez le site de Google – **https://www.google.com/recaptcha/about/**
- Cliquez sur **V3 Admin Console**.
- Créez un compte sur Google si ce n'est pas encore fait.
- Sélectionnez la version 3.
- Entrez votre ou vos domaines.
- Sauvegardez.
- Récupérez les deux clés d'authentification (celle du site et la secrète).

# Implémentation avec Elementor

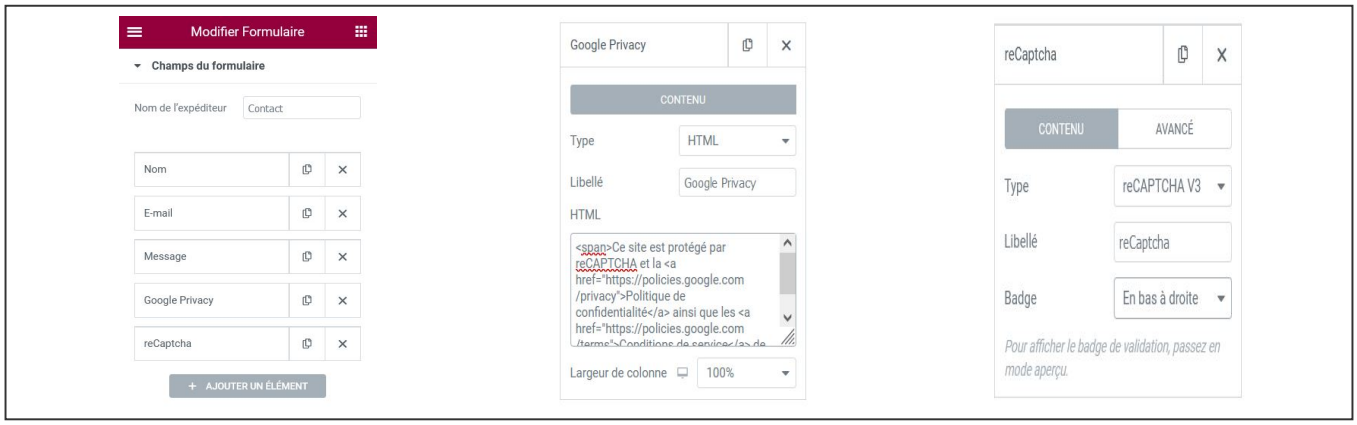
La deuxième partie est tout aussi simple. Il suffit d'implémenter le ReCaptcha dans les réglages d'Elementor pour que ceux-ci soient disponibles dans tous les formulaires.

- Dans un premier temps, il faut configurer Elementor avec les nouvelles clés ReCaptcha obtenues.



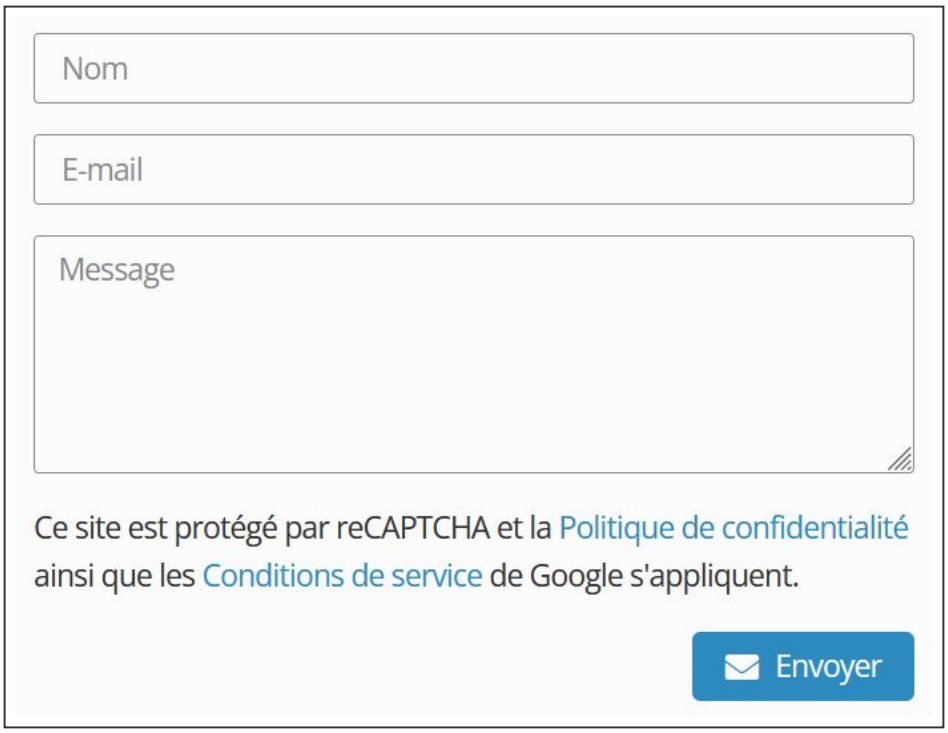

*Les réglages du ReCaptcha pour Elementor*

- Ensuite, il faut créer un formulaire. Dans cet exemple, c'est le formulaire de contact du site.
- Il faut choisir la version 3 du ReCaptcha. La position du badge n'a pas d'importance
- Dans cet exemple, un avertissement a été ajouté. Il a été créé avec un champ de type HTML.




*Les champs du formulaire de contact*

Voici le résultat. Le champ ReCaptcha affiche un badge, mais il est possible de le cacher avec un peu de CSS.




*Le formulaire de contact*

# Cacher le badge

Voici le code CSS pour cacher le badge.

```
/* Recaptcha badge */
.grecaptcha-badge {
  display: none;}
```

# Un peu de code ...

La troisième tâche à effectuer est la plus difficile. Il va falloir coder l'implémentation du ReCaptcha pour tous les formulaires qui ne sont pas réalisés avec Elementor, comme par exemple, celui des commentaires. En effet, même si Elementor possède un widget pour afficher la zone de commentaires dans un modèle de type unique, le formulaire est celui fourni de base avec WordPress, donc pas de moyen d'ajouter un champ de Recaptcha sur ce dernier.

Commencez par créer trois fichiers dans le thème enfant.

- **Enqueue_JS_Scripts.php**
- **Comments_Form_Recaptcha.js**
- **Comments_Form_ReCaptcha.php**

Modifiez le fichier **"functions.php"** pour y inclure les deux nouveaux fichiers PHP. Le fichier Javascript sera quant à lui, appeler par l'action PHP de mise en queue ci-dessous.

```
include_once ($dir . '/php/Comments_Form_ReCaptcha.php');
include_once ($dir . '/php/Enqueue_JS_Scripts.php');
```

## Le fichier d'appel des fonctions Javascript

- Créez un fichier **PHP** et nommez-le comme bon vous semble. Dans cet exemple, il porte le nom de **"Enqueue_JS_Scripts.php"**. Il va servir à mettre en queue deux fichiers Javascript.

Collez le code PHP ci-dessous :

```
// ReCaptcha library
if (!function_exists('Comments_Form_Recaptcha')) {
  function Comments_Form_Recaptcha() {
    // Only on single pages
    if (!is_admin() && (is_single())) {
      $jquery = array ('jquery');
      $version = '1.0';
      $in_footer = true;
    wp_enqueue_script('google-recaptcha', 'https://www.google.com/recaptcha/api.js?render=[Col
    wp_enqueue_script('Comments_Form_Recaptcha', get_stylesheet_directory_uri() . '/js/Comments
    }
  }
}add_action("wp_enqueue_scripts", "Comments_Form_Recaptcha");
```

- Le premier fichier **"api.js"** est une API de Google. C'est elle qui va permettre d'envoyer des requêtes d'authentification. Il faut lui indiquer **la clé du site** précédemment créée.
- Le deuxième fichier **"Comments_Form_Recaptcha.js"** va nous permettre de récupérer le jeton d'authentification (token, en anglais).

## Le fichier de récupération du jeton

- Créez un fichier **"Comments_Form_Recaptcha.js"** ou comme bon vous semble et collez le code ci-dessous :
- N'oubliez pas de remplacer la clé du site, la même que dans le fichier PHP ci-dessus.
- Attention de bien nommez l'élément jeton par son nom **"recaptcha"**. C'est le champ contenant le jeton.

```
jQuery(document).ready(function($) {
  $('#commentform').submit(function(event) {

    // Get the token
    grecaptcha.execute('[Collez ici la clé du site]', {action: 'submit'}).then(function(token)
      document.getElementById('recaptcha').value = token;
    });
  });});
```

## Le fichier de vérification et validation du jeton

- Créez un fichier **"Comments_Form_ReCaptcha.php"** ou comme bon vous semble et collez le code ci-dessous.

- Le filtre **"add_google_recaptcha"**, va créer un champ supplémentaire de type caché dont le nom est **"recaptcha"**. Il va se créer au chargement du formulaire de commentaires. Le reste du code affiche un message d'avertissement concernant la police et les termes d'utilisation de Google.

> La fonction **"pl__()"** permet de traduire du texte à l'aide du plugin **"Polylang"**. Il faut le supprimer si votre site ne l'utilise pas.

- L'action **"verify_google_recaptcha"**, va, quant à lui, vérifier que le champ **"recaptcha"** ci-dessus n'est pas vide et appeler une fonction de validation qui sera exécuter avant que le commentaire soit enregistré sur le site.
- La fonction **"validate_google_recaptcha"**, va, pour finir, construire la requête HTTP à l'aide de la clé secrète et du token. Google va nous retourner une réponse positive ou négative en fonction du contenu du commentaire.

```
function add_google_recaptcha($args) {

  // Add a recaptcha field
  $field = '<p class="recaptcha"><input id="recaptcha" name="recaptcha" type="hidden"></input>
  $args['submit_field'] = $field . $args['submit_field'];
  // Add a Google warning
  $message = '<p class="recaptcha-warning">' . pl__('This site is protected by reCAPTCHA and')
  $message .='<a href="https://policies.google.com/privacy">' . pl__('Privacy Policies') . ' </
  $message .='<a href="https://policies.google.com/terms">' . pl__('Terms of Service') . ' </a
  $args['comment_notes_after'] = $message;
  return $args;
}
add_filter('comment_form_defaults','add_google_recaptcha');
/**********************************************************************************************
function verify_google_recaptcha() {

  // Verify the token before publishing the comment
  if (isset($_POST['recaptcha']) && $_POST['recaptcha']) {
    if (!validate_google_recaptcha($_POST['recaptcha'])) {
      wp_die("ReCaptcha Token is not valid or suspicious");
    }
  }
}
add_action('pre_comment_on_post', 'verify_google_recaptcha');
/**********************************************************************************************

function validate_google_recaptcha($recaptcha) {
  // Build HTTP query
  $recaptcha_postdata = http_build_query(array(
    'secret' => '[Collez ici la clé secrète]',
    'response' => $recaptcha,
    'remoteip' => $_SERVER['REMOTE_ADDR']));
  // Send query to Google with POST method
  $recaptcha_opts = array('http' => array(
    'method'  => 'POST',
    'header'  => 'Content-type: application/x-www-form-urlencoded',
    'content' => $recaptcha_postdata));
  // Verify the token
  $recaptcha_context  = stream_context_create($recaptcha_opts);
  $recaptcha_response = json_decode(file_get_contents("https://www.google.com/recaptcha/api/si
  // Return a response
  if ($recaptcha_response['success']) {
    return true;
  }
  return false;}
```

# Mot de la fin

Il existe bien entendu d'autres manières pour filtrer les commentaires et messages indésirables sur un site web, à commencer par une kyrielle de plugins. Cette solution a été adapté de la version 2 qui était plus simple et n'utilisait pas de Javascript, mais Google a décidé d'utiliser la fonction **"grecaptcha.execute"** d'où l'appel en Javascript dans la version 3.