



reCAPTCHA

Usabilidad de la verificación de Google CAPTCHA en su version 3.

CAPTCHA: Completely Automated Public Turing test to tell Computers and Humans Apart, traducido al español es algo así como **"Test de Turing completamente automático para diferenciar a los humanos de los ordenadores"**

Anexación del CAPTCHA en un formulario

El primer paso para la usabilidad de google es agregar la API que se proporciona en su documentación de google <https://developers.google.com/recaptcha/docs/v3> ,Esta API se agrega en el header o bien puede ser incluida desde otro archivo de tu Javascript.

API: `<script src="https://www.google.com/recaptcha/api.js"></script>`

El segundo paso es el desarrollo de un formulario en estructura HTML con su action para enviar el formulario, method para definir si esta información viajará a travez de GET o POST, y por último un id que sera el indentificador del formulario.

Para este ejemplo se creara tres input:

- Nombre con tipo texto
- Correo con tipo texto
- TOKEN con tipo oculto

FORMULARIO:

```
<form action="enviar.php" method="post" id="form_prueba">
  <div class="mb-3">
    <label for="name" class="form-label">Nombre</label>
    <input type="text" class="form-control" name="name">
  </div>
  <div class="mb-3">
    <label for="email" class="form-label">Correo</label>
    <input type="email" class="form-control" name="email">
  </div>
  <input type="hidden" name="token" id="googlecp3" value="">

  <button type="submit">Submit</button>
</form>
```

El tercer paso es eliminar el botón submit que viene por defecto en el formulario y remplazarlo con el que proporciona google en su documentación.

BOTÓN SUBMIT DE GOOGLE:

```
<form action="enviar.php" method="post" id="form_prueba">
  <div class="mb-3">
    <label for="name" class="form-label">Nombre</label>
    <input type="text" class="form-control" name="name">
  </div>
  <div class="mb-3">
    <label for="email" class="form-label">Correo</label>
    <input type="email" class="form-control" name="email">
  </div>
  <input type="hidden" name="token" id="googlecp3" value="">
  <!--Site-key de localhost-->
  <button class="g-recaptcha"
    data-sitekey="reCAPTCHA_site_key_public"
    data-callback='onSubmit'
    data-action='submit'>Submit
  </button>
</form>
```

El cuarto paso es hacer uso del callback que es una funcion de devolución de llamada para manejar el token. este script se puede agregar en el footer o bien puede ser incluida desde otro archivo Javascript.

Tambien agregamos una configuración más para obtener el token y enviarlo en el input que se encuentra oculto en el formulario.

FUNCION DE LLAMADA (ANTES):

```
<script>
  function onSubmit(token) {
    document.getElementById("form_prueba").submit();
  }
</script>
```

FUNCION DE LLAMADA (DESPUES):

```
<script>
  function onSubmit(token) {
    var token = token;
    document.getElementById("googlecp3").value = token;
    document.getElementById("form_prueba").submit();
  }
</script>
```

Recepción de la información

El primer paso es crear un archivo para la recepción de datos con el nombre que se haya puesto en el action del formulario, en este ejemplo tenemos enviar.php por lo tanto crearemos un archivo con ese nombre.

```
action="enviar.php" method="post" id="form_prueba"
```

El segundo paso es crear las variables en php para recibir los datos por POST.

```
$nombre = $_POST['name'];  
$email  = $_POST['email'];  
$token  = $_POST['token'];
```

El tercer paso es crear una variable con la clave privada del reCaptcha:

```
$private_key="private_key_capcha";
```

Verificación de token con el servidor

Crear la configuración de validacion con el servidor con cURL:

```
// call curl to POST request  
$ch = curl_init();  
curl_setopt($ch, CURLOPT_URL, "https://www.google.com/recaptcha/api/siteverify");  
curl_setopt($ch, CURLOPT_POST, 1);  
curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query(array('secret' =>  
$private_key, 'response' => $token)));  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
$response = curl_exec($ch);  
curl_close($ch);  
$arrResponse = json_decode($response, true);  
$arraycapcha=var_dump($arrResponse);  
  
// verify the response  
if($arrResponse["success"] == '1' ) {  
    echo '<div class="alert alert-warning" role="alert">'. 'Nombre registrado:  
' . $nombre. '</div><br>';  
    echo '<div class="alert alert-warning" role="alert">'. 'Correo recibido:  
' . $email. '</div><br>';  
    echo '<div class="alert alert-success" role="alert">'. 'TOKEN:  
' . $token. '</div>';  
} else {  
    // spam submission  
    // show error message  
    echo "Error";  
}
```

Imagen de muestra

Nombre

alevi

Correo

email@prueba.com

enviar

Recepción

```
array(5) { ["success"]=> bool(true) ["challenge_ts"]=> string(20) "2021-03-13T20:11:24Z" ["hostname"]=> string(9) "localhost" ["score"]=> float(0.90000000000000002220446049250313080847263336181640625) ["action"]=> string(6) "submit" }
```

Nombre registrado: alevi

Correo recibido: email@prueba.com

TOKEN: 03AGdBq24wjvRqe9JMVSO5GDPbnsZWCzzFNI6kCZHGHIC-wISN1NhbqXBS7pS-is-E5EfetO2XcTm1CiqE0j2mnDBPHwJaq_es04pzOpJ9tu1Nf99JiGSrEISs8rNc2fLD_ZOal00NS8FpSsSiiNRY49tNbiKs2gRwH1s0uir-U-XdgMHMs9xlCit5FIhAiaJwak1ybBc47cV0NVhcAbqF-cKxeLx_Lxr7i3-A8uix9BG4G7WvmnopHomca6L_aoiUbdcf-xokoxsLRuZ94I9IFjdjP29Byr3i9zB0Crp7PUHkMntH3LLDLV0CnbUDqachp4OHUMGpP2k7O3IKXd6OrOC4cr981TS5FPGjle-e3OizYuXuW78dEBobBpD8xbAvVyrY8fps0ld8HRrX5tnQfAb4IYSt4tBr4Fo5jNf26reZ8hKTf8kDutBibcsf6ZUSPColo8myg3mj