

# Google reCaptcha (with ASP.NET MVC)

Sanjeet Singh

\* R&D Dept., Syscom Corporation Ltd.

**Abstract-** This paper talks about  
•Why Google reCaptcha?  
•Description about old and new Google reCaptcha?  
•How to use?

**Index Terms-** Why Google ReCaptcha, old Google reCaptcha, new Google reCaptcha, how to use;

## INTRODUCTION

Dear Readers,

First of all, I would say this paper is just for ASP.net developers.

Google reCaptcha protects us from spam and abuse. You may have heard of the word Electronic Spamming, it is the use of electronic messaging systems to send unsolicited messages, especially advertising, as well as sending messages repeatedly on the same site. So Google reCaptcha save us from such spam by forcing user to verify that he/she is not a robot.

reCaptcha is a free google service and uses an advanced risk analysis and adaptive Captchas to keep away automated software from engaging in abusive activities on our site and allow valid user pass through with ease.

There are software that can increase the number of hit to the server which can impact the server functionality and if number of request rises above the server capacity, it can cause server down. Google reCaptcha can prevent such conditions.

Now in this paper I have consolidated both old and new Google reCaptcha and how to use them. I have also included code for each reCaptcha as well.

### Description about old and new Google reCaptcha:

**Old Google reCaptcha:** the old reCaptcha's are either text based verification or image based verification where user has to type the text or select the images from the given images to prove that they are human not robot. The major problem with old reCaptcha is that they are not user friendly and kind of frustrating.

**New Google reCaptcha:** they are more user friendly, the user just has to check the checkbox and they are good to go. If without checking the checkbox you submit the request validation failed. If verification failed multiple times than verification will changed into image based verification, where user has to choose that matches the criteria to validate the request. Hence this also increase the verification strength.

### How to Use:

Step 1: First you need to sign up for reCaptcha. Go to <https://www.google.com/recaptcha/intro/index.html>

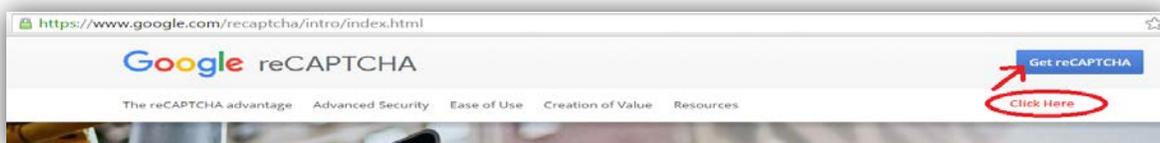


Figure 1: Sign Up for ReCaptcha



### Old reCaptcha:

- In your mvc application open view (.cshtml file) where you want to use reCaptcha and add the following code. Replace Site/Public key with your Site key.

```
using (Html.BeginForm("FormSubmit", "Recaptcha", FormMethod.Post))
{
    <div class="editor-label">
        <script type="text/javascript" src='https://www.google.com/recaptcha/api/challenge?k=Site/Public Key'>
        </script>
        <noscript>
            <iframe src='https://www.google.com/recaptcha/api/challenge?k=Site/Public Key' height="300" width="500" frameborder="0"></iframe><br />
            <textarea name="recaptcha_challenge_field" rows="3" cols="40"></textarea>
            <input type="hidden" name="recaptcha_response_field" value="manual_challenge" />
        </noscript>
    </div>
    <p>
        <input type="submit" value="Create" />
    </p>
}
</fieldset>
<span style="display:inline-block;margin:20px;font-size:20px;padding:20px;border:1px solid #030303;">
    @ViewBag.Message;
</span>
```

Figure 4: reCaptcha View

- Open the controller of the above view and add FormSubmit method which is going to validate your reCaptcha. Replace Secret key with your secret key

```
[HttpPost]
0 referencas
public ActionResult FormSubmit()
{
    bool flag = false;
    string Message = "";
    string[] result;
    HttpWebRequest request = (HttpWebRequest)WebRequest.Create("https://www.google.com/recaptcha/api/verify");
    request.ProtocolVersion = HttpVersion.Version10;
    request.Timeout = 0x7530;
    request.Method = "POST";
    request.UserAgent = "reCAPTCHA/ASP.NET";
    request.ContentType = "application/x-www-form-urlencoded";
    string formData = string.Format("privatekey={0}&remoteip={1}&challenge={2}&response={3}", new object[]{
        HttpUtility.UrlEncode("Secret key"),
        HttpUtility.UrlEncode(Dns.GetHostEntry(Dns.GetHostName()).AddressList[1].ToString()),
        HttpUtility.UrlEncode(Request.Form["recaptcha_challenge_field"]),
        HttpUtility.UrlEncode(Request.Form["recaptcha_response_field"])
    });

    byte[] formByte = Encoding.ASCII.GetBytes(formData);
    using (System.IO.Stream requestStream = request.GetRequestStream())
    {
        requestStream.Write(formByte, 0, formByte.Length);
    }
    try
```

```
{
    using (WebResponse httpResponse = request.GetResponse())
    {
        using (System.IO.TextReader readStream = new System.IO.StreamReader(httpResponse.GetResponseStream(), Encoding.UTF8))
        {
            result = readStream.ReadToEnd().Split(new string[] { "\n", "@\n" }, StringSplitOptions.RemoveEmptyEntries);
            Message = result[1];
            flag = Convert.ToBoolean(result[0]);
        }
    }
}
catch (Exception ex)
{
    flag = false;
}
if (flag)
{
    ViewBag.Message = "Google reCaptcha validation success";
}
else
{
    ViewBag.Message = "Google reCaptcha validation failed";
}

return View("Index");
```

Figure 5: reCaptcha Controller code

- Now browse your site to check reCaptcha. On click of create if captcha is valid Google reCaptcha validation success message will be shown otherwise reCaptcha validation failed message will be shown.

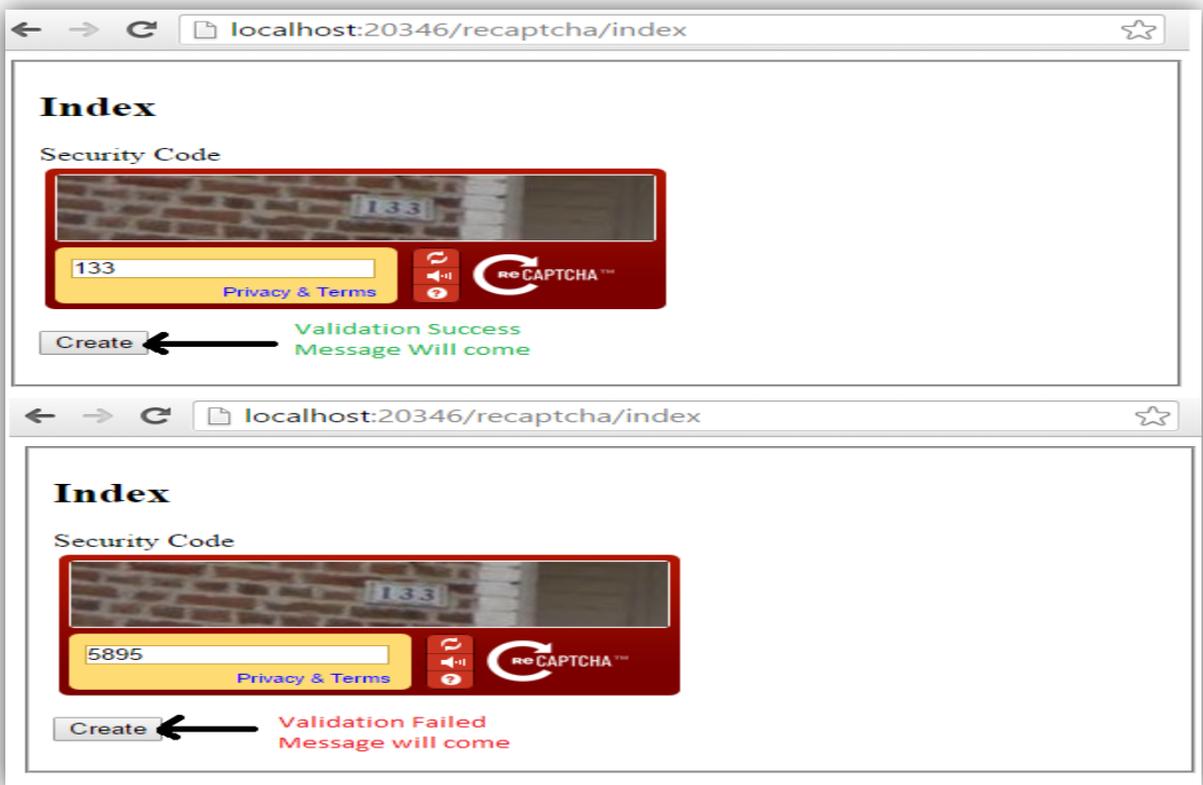


Figure 6: reCaptcha Validation

### New reCaptcha:

- In your mvc application open view (.cshtml file) where you want to use reCaptcha and add the following code. Replace Site/Public key with your Site key.

```
<div>
    @using (Html.BeginForm("FormSubmit", "Recaptcha", FormMethod.Post))
    {
        <div class="g-recaptcha" data-sitekey="Public/Site key"></div>
        <input type="submit" value="Submit" />
    }
</div>
<span style="display:inline-block;margin:20px;font-size:20px;padding:20px;border:1px solid #D3D3D3;">
    @ViewBag.Message;
</span>
<script src='https://www.google.com/recaptcha/api.js'></script>
```

Figure 7: reCaptcha View Code

- Open the controller of the above view and add FormSubmit method which is going to validate your reCaptcha. Replace Secret key with your secret key

```
[HttpPost]
0 references
public ActionResult FormSubmit()
{
    var response=Request["g-recaptcha-response"];
    string secretkey = "secret key";
    var client = new System.Net.WebClient();
    var result = client.DownloadString(string.Format("https://www.google.com/recaptcha/api/siteverify?secret={0}&response={1}",secretkey,response));
    var obj = JObject.Parse(result);
    var status = (bool)obj.SelectToken("success");
    ViewBag.Message = status ? "Google reCaptcha validation success" : "Google reCaptcha validation failed";
    return View("Index");
}
```

Figure 8: reCaptcha Controller code

- Now browse your site to check reCaptcha. On click of Submit if checkbox is checked Google reCaptcha validation success message will be shown otherwise reCaptcha validation failed message will be shown.

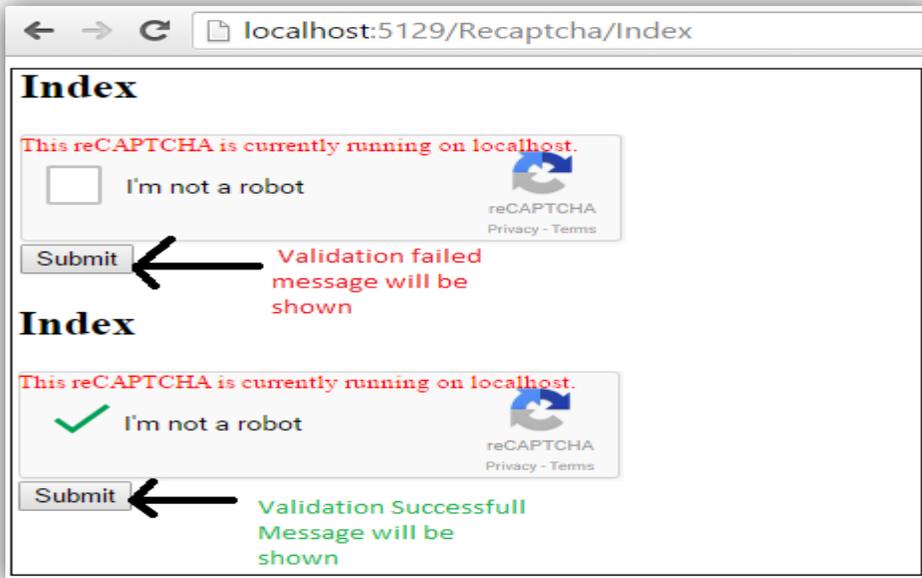


Figure 9: reCaptcha Validation

- If validation failed multiple times now on checking the checkbox security measures will increase by increasing the strength of verification through selection of Images.

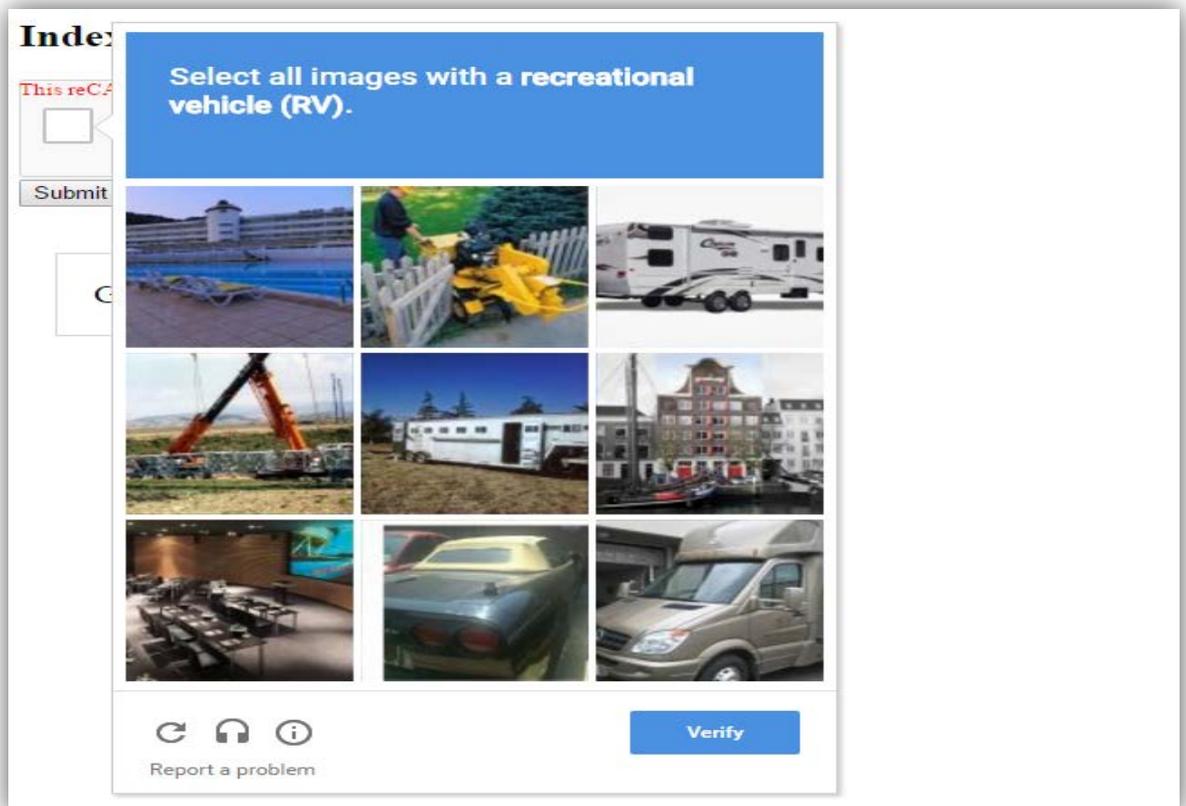


Figure 10: reCaptcha Validation

#### IV. CONCLUSION

Google reCaptcha is very useful to developer as well as server administrator, it prevents invalid request to the server which can hamper the server functionality. It keep your site safe and functional.  
Thus a server become more functional and can handle multiple number of users.  
Google reCaptcha prevents spam and are easy to implement.

#### ACKNOWLEDGMENT

This work was supported by Syscom Corporation Ltd.

#### REFERENCES

- [1] <https://www.google.com/recaptcha/intro/index.html>
- [2] <https://en.wikipedia.org/wiki/reCaptcha>

#### AUTHORS

**First Author** – Sanjeet Singh, B.Tech, Software Engineer, sanjeetsingh1991@gmail.com