**طریقه راه‌اندازی Captcha Image در پروژه**

**مرحله اول**

وارد نشانی ذیل، شده و کد موجود در قسمت Package Reference را کپی کرده و در داخل فایل Server.csproj قرار می‌دهیم:

<https://www.nuget.org/packages/DNTCaptcha.Core>

Server.csproj:

<ItemGroup>

<PackageReference Include="DNTCaptcha.Core" Version="4.4.2" />

</ItemGroup>

یک‌بار پروژه را Rebuild می‌کنیم، تا عملیات دانلود Nuget صورت گیرد.

**مرحله دوم**

در داخل همه فایل‌های appsettings (Production و Development) یک کلید به نام CaptchaImageEncryptionKey و به شکل ذیل ایجاد می‌کنیم:

"ApplicationSettings": {

"CaptchaImageEncryptionKey": "My Key is Hello, World!",

**نکته مهم:** طول این کلید باید حداقل شانزده کاراکتر باشد!

**مرحله سوم**

در پوشه ذیل:

/Pages/Shared/PartialViews

یک Partial View به نام \_CaptchaImage ایجاد می‌کنیم و کد ذیل را در داخل آن می‌نویسیم:

<div class="mb-3">

<dnt-captcha asp-captcha-generator-max="9999"

asp-captcha-generator-min="1000"

asp-captcha-generator-language="English"

asp-captcha-generator-display-mode="ShowDigits"

asp-use-relative-urls="true"

asp-placeholder="Security Code"

asp-validation-error-message="@(Resources.Messages.Errors.CaptchaImage)"

asp-font-name="Tahoma"

asp-font-size="20"

asp-fore-color="#333333"

asp-back-color="#FCF6F5FF"

asp-text-box-class="form-control"

asp-text-box-template= "<div class='input-group'><span class='input-group-text'><span class='bi bi-lock'></span></span>{0}</div>"

asp-validation-message-class="text-danger"

asp-refresh-button-class="bi-arrow-counterclockwise btn-lg"

asp-show-refresh-button="true"

asp-dir="ltr" />

</div>

**مرحله چهارم**

کد ذیل را در داخل فایل Program.cs و قبل از دستور builder.Build می‌نویسیم:

var captchaImageEncryptionKey =

builder.Configuration.GetSection

(key: "ApplicationSettings:CaptchaImageEncryptionKey").Value;

if(string.IsNullOrWhiteSpace(value: captchaImageEncryptionKey) ||

captchaImageEncryptionKey.Length < 16)

{

captchaImageEncryptionKey = "My Key is Hello, World!";

}

builder.Services

.AddControllers();

// using DNTCaptcha.Core;

builder.Services

.AddDNTCaptcha(options =>

{

// .UseDistributedSerializationProvider()

// options.UseSessionStorageProvider()

// -> It doesn't rely on the server or client's times. Also, it's the safest one.

// options.UseMemoryCacheStorageProvider()

// -> It relies on the server's times. It's safer than the CookieStorageProvider.

// .UseDistributedCacheStorageProvider()

// -> It's ideal for scalability using

// `services.AddStackExchangeRedisCache()` for instance.

options.UseCookieStorageProvider

(sameSite: Microsoft.AspNetCore.Http.SameSiteMode.Strict)

// -> It relies on the server and client's times. It's ideal.

// for scalability because it doesn't save anything in the server's memory.

// This is optional!

// Don't set this line (remove it) to use the installed system's fonts

// (FontName = "Tahoma") or if you want to use a custom font, make sure

// that font is present in the wwwroot/fonts folder and use a good

// and complete font!

//.UseCustomFont(fullPath: System.IO.Path.Combine

// (env.WebRootPath, "fonts", "IRANSans(FaNum)\_Bold.ttf"))

.AbsoluteExpiration(minutes: 10)

.ShowThousandsSeparators(show: false)

.WithNoise(pixelsDensity: 25, linesCount: 3)

.WithEncryptionKey(key: captchaImageEncryptionKey)

// This is optional!

// Change it if you don't like the default names.

.InputNames(component: new DNTCaptcha.Core.DNTCaptchaComponent

{

CaptchaInputName = "DNTCaptchaInputText",

CaptchaHiddenInputName = "DNTCaptchaText",

CaptchaHiddenTokenName = "DNTCaptchaToken",

})

// This is optional!

// Change it if you don't like its default name.

.Identifier(className: "dntCaptcha")

;

});

و دستور ذیل را بعد از دستور builder.Build می‌نویسیم:

app.MapRazorPages();

app.MapControllers();

**مرحله پنجم**

در پوشه Pages وارد فایل \_ViewImports.cshtml می‌شویم و دستور ذیل را به آن اضافه می‌کنیم:

@addTagHelper \*, DNTCaptcha.Core

@using Infrastructure

@using Infrastructure.TagHelpers

@addTagHelper \*, Server

@addTagHelper \*, DNTCaptcha.Core

@addTagHelper \*, Infrastructure.TagHelpers

@addTagHelper \*, Server.Infrastructure.TagHelpers

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

**مرحله ششم:**

فرض کنید که می‌خواهیم از این Captcha Image در صفحه Register استفاده نماییم.

برای این منظور در پوشه Pages و در پوشه Account وارد فایل Register.cshtml می‌شویم و در محلی که مایل هستیم، دستور ذیل را می‌نویسیم:

<**partial** **name**="~/Pages/Shared/PartialViews/\_CaptchaImage.cshtml" />

**مرحله هفتم:**

وارد فایل Register.cshtml.cs می‌شویم و سرویس‌های ذیل را به Constructor اصطلاحا Inject می‌کنیم:

DNTCaptcha.Core.DNTCaptchaOptions

DNTCaptcha.Core.IDNTCaptchaValidatorService

public RegisterModel

(DNTCaptcha.Core.IDNTCaptchaValidatorService captchaValidatorService,

Microsoft.Extensions.Options.IOptions<DNTCaptcha.Core.DNTCaptchaOptions> captchaOptions) :

{

CaptchaOptions = captchaOptions.Value;

CaptchaValidatorService = captchaValidatorService;

}

private DNTCaptcha.Core.DNTCaptchaOptions CaptchaOptions { get; init; }

private DNTCaptcha.Core.IDNTCaptchaValidatorService CaptchaValidatorService { get; init; }

**مرحله هشتم**

در تابع Post دستور ذیل را قبل از چک کردن ModelState.IsValid می‌نویسیم:

if (CaptchaValidatorService.HasRequestValidCaptchaEntry

(DNTCaptcha.Core.Language.English, DNTCaptcha.Core.DisplayMode.ShowDigits) == false)

{

ModelState.AddModelError(CaptchaOptions.CaptchaComponent

.CaptchaInputName, Resources.Messages.Errors.CaptchaImageIsNotCorrect);

}

**نکته مهم:**

دستور ذیل با Captcha Image مشکل دارد!

builder.Services

.AddRouting(options =>

{

options.LowercaseQueryStrings = true;

});

**نکته مهم:**

تمام دستورات ذیل باید در صفحه و دقیقا به همین ترتیب وجود داشته باشند:

<script src="/lib/jquery/jquery.js"></script>

<script src="/lib/jquery-validate/jquery.validate.js"></script>

<script src="/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.js"></script>

<script src="/lib/jquery-ajax-unobtrusive/jquery.unobtrusive-ajax.js "></script>