



Send Email Using Templates In ASP.NET Core



Nishan Aryal



Updated date Aug 15 2019



142k



12



12



[Download Free .NET & JAVA Files API](#)

[Try Free File Format APIs for Word/Excel/PDF](#)



[SendEmailusingTemplatesinASPNETCoreApplications.zip](#)

In this article, we will discuss how to send emails using a Template file in ASP.NET Core Application. Sending email using Template file is similar to sending emails using plain text. The only difference is that now, the source of the content is a text file rather than direct content for email.

If you don't know how to send email in ASP.NET Core applications, please refer to my previous article [Configure Email Service In ASP.NET Core Using MailKit](#). Let's assume, we have already configured email service in ASP.NET Core application.

Milestones

1. Create a Simple Email Template.
2. Configure Email Setting in ASP.NET Core Application.
3. Read content from Email Template using StreamReader()
4. Personalize Email Content
5. Send Email to User

We will keep the template file separate so that we can change the file design and content anytime. Let's keep these template files under wwwroot => Templates => EmailTemplate.

Step 1

For this, create a new folder named Templates under wwwroot folder.

Step 2

Create a new folder named Email Template under Templates folder.

Step 3

Add a new Item in Email Template (right click on Email Template folder >> add New Item).

Step 4

Select ASP.NET Core and then HTML Page.

Step 5

Set a name for the HTML file. In my case, I have named it as Welcome_EmailTemplate.html (But I will be using Confirm_Account_Registration.html for demo purposes whose design is the same with a different file name).

 Send Email using Templates in ASP.NET Core Applications

Once you have created Email Templates, the Solution Explorer looks like below. (In my case, there are 5 Email Templates)

 Send Email using Templates in ASP.NET Core Applications

Send Email using Templates in ASP.NET Core Applications

The code for the above sample Email Template is given below.

Code Snippet

```
01. <html xmlns="http://www.w3.org/1999/xhtml">
02. <head>
03.     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
04.     <title>Welcome Email from TCP</title>
05. </head>
06.
07. <body>
08.     <table width="100%" border="0" cellspacing="0" cellpadding="0">
09.         <tr>
10.             <td align="center" valign="top" bgcolor="#ffe77b" style="background-color:#ffe77b;">
11.                 <br>
12.                 <br>
13.                 <table width="600" border="0" cellspacing="0" cellpadding="0">
14.                     <tr>
15.                         <td height="70" align="left" valign="middle"></td>
16.                     </tr>
17.                     <tr>
18.                         <td align="left" valign="top">
19. 
21.                         </tr>
22.                         <tr>
23.                             <td align="left" valign="top" bgcolor="#564319" style="background-
color:#564319; font-family:Arial, Helvetica, sans-serif; padding:10px;">
<div style="font-size:36px; color:#ffffff;">
<b>{0}</b></div>
</td>
</tr>
</table>
</tr>
</table>
</body>
</html>
```

```

26.         <b>{1} : ASP.NET Core Demp App</b>
27.     </div>
28. </td>
29. </tr>
30. <tr>
31.     <td align="left" valign="top" bgcolor="#ffffff" style="background-
color:#ffffff;">
32.         <table width="100%" border="0" cellspacing="0" cellpadding="0">
33.             <tr>
34.                 <td align="center" valign="middle" style="padding:10px; color:#
size:28px; font-family:Georgia, 'Times New Roman', Times, serif;">
35.                     Congratulations! <small>You are registered.</small>
36.                 </td>
37.             </tr>
38.         </table>
39.         <table width="95%" border="0" align="center" cellpadding="0" cellspacing="0">
40.             <tr>
41.                 <td width="40%" align="center" valign="middle" style="padding:10px;">
42.                     
43.                 </td>
44.                 <td align="left" valign="middle" style="color:#525252; font-
family:Arial, Helvetica, sans-serif; padding:10px;">
45.                     <div style="font-size:16px;">
46.                         Dear {2},
47.                     </div>
48.                     <div style="font-size:12px;">
49.                         Thank you for showing your interest in our website.
50.                         All you need to do is click the button below (it only takes 30 seconds)
51.                         You won't be asked to log in to your account -
we're simply verifying ownership of this email address.
52.                     <hr>
53.                     <center>
54.
55.                         <button type="button" title="Confirm Account Registration" />

```

```

57.                                     Confirm Email Now
58.                                     </a>
59.                                 </button>
60.
61.                             </center>
62.                         </div>
63.                     </td>
64.                 </tr>
65.             </table>
66.             <table width="100%" border="0" cellspacing="0" cellpadding="0">
67.                 <tr>
68.                     <td align="center" valign="middle" style="padding:5px;">
69.                         
74.                 <tr>
75.                     <td align="left" valign="middle" style="padding:15px; font-
family:Arial, Helvetica, sans-serif;">
76.                         <div style="font-size:20px; color:#564319;">
77.                             <b>Please keep your credentials confidential for future
78.                         </div>
79.                         <div style="font-size:16px; color:#525252;">
80.                             <b>Email                :</b> {2}
81.                             <br />
82.                             <b>Username :</b> {3}
83.                             <br />
84.                             <b>Password :</b> {4}
85.                         </div>
86.                     </td>
87.                 </tr>
88.             </table>

```

```

90.         <tr>
91.             <td align="left" valign="middle" style="padding:15px; backgroun
color:#564319; font-family:Arial, Helvetica, sans-serif;">
92.                 <div style="font-size:20px; color:#fff;">
93.                     <b>Update your password now.</b>
94.                 </div>
95.                 <div style="font-size:13px; color:#ffe77b;">
96.                     Weak passwords get stolen and lead to hacked accounts.
97.                     <br>
98.                     <br>
99.                     <a href="#" style="color:#ffe77b; text-
decoration:underline;">CLICK HERE</a> TO CHANGE PASSOWORD
100.                 </div>
101.             </td>
102.         </tr>
103.     </table>
104.     <table width="95%" border="0" align="center" cellpadding="0" cellspacing="0">
105.         <tr>
106.             <td width="50%" align="left" valign="middle" style="padding:10px">
107.                 <table width="75%" border="0" cellspacing="0" cellpadding="0">
108.                     <tr>
109.                         <td align="left" valign="top" style="font-
family:Verdana, Geneva, sans-serif; font-size:14px; color:#000000;">
110.                             <b>Follow Us On</b>
111.                         </td>
112.                     </tr>
113.                     <tr>
114.                         <td align="left" valign="top" style="font-
family:Verdana, Geneva, sans-serif; font-size:12px; color:#000000;">
115.                             <table width="100%" border="0" cellspacing="0">
116.                                 <tr>
117.                                     <td width="33%" align="left" valign="mi
118.                                         <a href="https://twitter.com" title
119.                                             
126.                                     </td>
127.                                     <td width="33%" align="left" valign="mi
128.                                     <a href="https://facebook.com" titl
129.                                     
139.        <b>Hours:</b> Mon-Fri 9:30-5:30, Sat. 9:30-
3:00, Sun. Closed <br>
140.        <b>Customer Support:
</b> <a href="mailto:name@yourcompanyname.com" style="color:#564319; text-
decoration:none;">name@yourcompanyname.com</a><br>
141.        <br>
142.        <b>Company Address</b><br>
143.        Company URL: <a href="http://www.yourcompanyname.com" targe
decoration:none;">http://www.yourcompanyname.com</a>
144.    </td>
145.    </tr>
146.    </table>
147.    </td>
148.    </tr>
149.    <tr>
150.        <td align="left" valign="top">


```

```
153.         <br>
154.         <br>
155.     </td>
156. </tr>
157. </table>
158. </body>
159. </html>
```

Now, our Email Template is ready.

Scenario - I : Verifying User's Email Ownership on Registration

We need to send a confirmation email to verify the ownership of the email to the user, just after first registration with his/her email on our application.

The process is,

- Collect user details from Registration Form
- Register Method of Account Controller of type [HttpPost] creates a new user using the following code.
-

```
01. //Take user details from Registration Form ie. model carries data from Registration form to
    here
02. var user = new ApplicationUser { UserName = model.Email, Email = model.Email };
03.
04. //CreateAsync inside UserManager creates new user when userdetails and password supplied.
05. var result = await _userManager.CreateAsync(user, model.Password);
```

- Generate unique code for new user details.
-

- A unique random callbackUrl is generated using the generated code and UserId.

```
01. var callbackUrl = Url.Action(nameof(ConfirmEmail), "Account", new { userId = user.Id, code = code }, protocol: HttpContext.Request.Scheme);
```

- Send callbackUrl to the user via email.

```
01. string Message = "Please confirm your account by clicking <a href=\"\" + callbackUrl +  
    "\">here</a>";  
02.  
03. //Send Email to User  
04. await _emailSender.SendEmailAsync(model.Email, subject, messageBody);
```

- When user clicks the send callback URL, he/she is sent back to the ConfirmEmail Method of Account Controller.

```
01. //Find User Details by userId  
02. var user = await _userManager.FindByIdAsync(userId);  
03. if (user == null)  
04. {  
05.     return View("Error");  
06. }  
07. var result = await _userManager.ConfirmEmailAsync(user, code);  
08. ViewData["Message"] = "Your email has been confirmed. Please Login now. ";  
09. ViewData["MessageValue"] = "1";
```

- This line will match the received user details and code via callbackUrl and sets ConfirmedEmail to true in database.

Now, during this process, if we want to send email using our custom Email Templates, we need to

1. Read content of Email Template
2. Read User Details received from User Registration Form.

5. Send email to user. (Using Mail Kit for now)

Reading Contents from Template File

Step 1

Get Information about the web Hosting Environment where our application is running.

```
01.     private readonly UserManager<ApplicationUser> _userManager;
02.     private readonly SignInManager<ApplicationUser> _signInManager;
03.     private readonly IEmailSender _emailSender;
04.     private readonly ISmsSender _smsSender;
05.     private readonly ILogger _logger;
06.     private readonly string _externalCookieScheme;
07.     private IHostingEnvironment _env;
08.
09.     public AccountController(
10.         UserManager<ApplicationUser> userManager,
11.         SignInManager<ApplicationUser> signInManager,
12.         IOptions<IdentityCookieOptions> identityCookieOptions,
13.         IEmailSender emailSender,
14.         ISmsSender smsSender,
15.         ILoggerFactory loggerFactory,
16.
17.         IHostingEnvironment env)
18.     {
19.         _userManager = userManager;
20.         _signInManager = signInManager;
21.         _externalCookieScheme = identityCookieOptions.Value.ExternalCookieAuthenticationScheme;
22.         _emailSender = emailSender;
23.         _smsSender = smsSender;
24.         _logger = loggerFactory.CreateLogger<AccountController>();
25.
26.         _env = env;
```

Step 2

get wwwroot Folder

```
01. var webRoot = _env.WebRootPath; //get wwwroot Folder
```

Step 3

Get TemplateFile located at wwwroot/Templates/EmailTemplate/Register_EmailTemplate.html

```
01. //Get TemplateFile located at wwwroot/Templates/EmailTemplate/Register_EmailTemplate.html
02.     var pathToFile = _env.WebRootPath
03.         + Path.DirectorySeparatorChar.ToString()
04.         + "Templates"
05.         + Path.DirectorySeparatorChar.ToString()
06.         + "EmailTemplate"
07.         + Path.DirectorySeparatorChar.ToString()
08.         + "Confirm_Account_Registration.html";
```

Step 4

Initialie BoduBuilder()

```
01. var builder = new BodyBuilder();
```

Step 5

Read Content of Template file using StreamReader and append to BodyBuilder()

```
01. using (StreamReader SourceReader = System.IO.File.OpenText(pathToFile))
02. {
```

```
05.
06. }
```

Passing value and Formatting the Content of Template with Dynamic Values

Step 1

use `string.Format(format item, dynamic values as parameters)` In our case, `{x}` values in Templates are to replace by dynamic values.

```
01. //{0} : Subject
02. //{1} : Current DateTime
03. //{2} : Email
04. //{3} : Username
05. //{4} : Password
06. //{5} : Message
07. //{6} : callbackURL
```

Code Snippet

```
01. string messageBody = string.Format(builder.HtmlBody,
02.                                     subject,
03.                                     String.Format("{0:dddd, d MMMM yyyy}", DateTime.Now),
04.                                     model.Email,
05.                                     model.Email,
06.                                     model.Password,
07.                                     Message,
08.                                     callbackUrl
09.                                     );
```

Send Email using Mailkit

If you don't know how to send email using Mail kit in ASP.NET Core Applications please refer to my previous blog. For now I am skipping setting up email service.

```
primary'>Confirmation Link</a>";
03. ViewData["MessageValue"] = "1";
```

await _emailSender.SendEmailAsync(model.Email, subject, messageBody); line will send email to user email.

Code used in Register Method of Account Controller

```
01. // POST: /Account/Register
02. [HttpPost]
03. [AllowAnonymous]
04. [ValidateAntiForgeryToken]
05. public async Task<IActionResult> Register(RegisterViewModel model, string returnUrl = null)
06. {
07.     ViewData["ReturnUrl"] = returnUrl;
08.     if (ModelState.IsValid)
09.     {
10.         var user = new ApplicationUser { UserName = model.Email, Email = model.Email };
11.         var result = await _userManager.CreateAsync(user, model.Password);
12.
13.         // var user = await _userManager.FindByEmailAsync(model.Email);
14.         if (result.Succeeded)
15.         {
16.             // Send an email with this link
17.             var code = await _userManager.GenerateEmailConfirmationTokenAsync(user);
18.             var callbackUrl = Url.Action(nameof(ConfirmEmail), "Account", new { userId = user.Id, code = code });
19.             //Email from Email Template
20.             string Message = "Please confirm your account by clicking <a href=\"" + callbackUrl + "\">Click here</a>";
21.             // string body;
22.
23.             var webRoot = _env.WebRootPath; //get wwwroot Folder
24.
25.             //Get TemplateFile located at wwwroot/Templates/EmailTemplate/Register_EmailTemplate.cshtml
26.             var pathToFile = _env.WebRootPath
27.                 + Path.DirectorySeparatorChar.ToString()
```

```
30.         + "EmailTemplate"
31.         + Path.DirectorySeparatorChar.ToString()
32.         + "Confirm_Account_Registration.html";
33.
34.     var subject = "Confirm Account Registration";
35.
36.     var builder = new BodyBuilder();
37.     using (StreamReader SourceReader = System.IO.File.OpenText(pathToFile))
38.     {
39.         builder.HtmlBody = SourceReader.ReadToEnd();
40.     }
41.     //{0} : Subject
42.     //{1} : DateTime
43.     //{2} : Email
44.     //{3} : Username
45.     //{4} : Password
46.     //{5} : Message
47.     //{6} : callbackURL
48.
49.     string messageBody = string.Format(builder.HtmlBody,
50.         subject,
51.         String.Format("{0:dddd, d MMMM yyyy}", DateTime.Now),
52.         model.Email,
53.         model.Email,
54.         model.Password,
55.         Message,
56.         callbackUrl
57.     );
58.
59.
60.     await _emailSender.SendEmailAsync(model.Email, subject, messageBody);
61.
62.     ViewData["Message"] = $"Please confirm your account by clicking this link: <a href='{Url.Action("ConfirmAccountRegistration", "Account", new {id = model.Id}, "ConfirmAccountRegistration.html")}'>Confirmation Link</a>";
63.     ViewData["MessageValue"] = "1";
```

```
66.         return RedirectToLocal(returnUrl);
67.     }
68.     ViewData["Message"] = $"Error creating user. Please try again later";
69.     ViewData["MessageValue"] = "0";
70.     AddErrors(result);
71. }
72.
73. // If we got this far, something failed, redisplay form
74. return View(model);
75. }
```

Application Execution and Output

Step 1

Now let's rebuild the solution and run the application.

Step 2

Let's register as a New user /Account/Register



Send Email using Templates in ASP.NET Core Applications

Once we register as a new user, a confirmation email is sent to the registered email. Upon checking the email inbox, we receive an email like this -



Send Email using Templates in ASP.NET Core Applications

We receive the personalized email sent from the Demo ASP.NET Core Application.

Scenario where we need Email Templates

- Notices and Call for events
- Welcome Email to New Users.
- Automatic Monthly report to Admins and Staffs.
- and many more....

Summary

So far, we have learned how to send personalized emails to users creating Email Templates .

You may also like

- [How to configure Email Services in ASP.NET Core Applications.](#)
- [How to export data to Excel from Web API in ASP.NET Core using Kendo UI](#)
- [Sorting and Grouping in Kendo Grid using Web API in ASP.NET Core Application](#)
- [Call Web API Using JQuery AJAX In ASP.NET Core Web Applications](#)
- [How to create Image Thumbnail in ASP.NET Web Application](#)

Next Recommended Article

[ASP.NET Core 2.0 - Applying Bootstrap Template](#)

ASP.NET Core

ASP.NET Core Applications

STemplates In ASP.NET Core Applications



Nishan Aryal *TOP 1000*

Working as Technical Consultant and Project Manager, adequate knowledge on Kendo-UI, JavaScript, SQL Server, Power BI, IOT, Azure, and ASP.NET MVC and ASP.NET Core. Lets learn together and share together is what I belie... [Read more](#)



657



1.4m



1

12 12



Type your comment here and press Enter Key (Minimum 10 characters)



it is very nice template. thanks

[Pritesh Maturkar](#)

1476 416 1.7k

1 Oct 03, 2019
0 Reply



This is nice ... but why dont you simply use a view with a model then use RenderPartialView ?

[Benoit Robin](#)

1864 28 0

1 Apr 12, 2019
0 Reply



How would the html template handle more complex objects than replacing strings? For example , a List of items in an email?

[John](#)

1872 20 0

1 Nov 27, 2018
0 Reply



Thank you for the article. But i am getting Input string was not in a correct format,error .Please help me in this

[Komal Sharma](#)

1762 130 0

1 Jul 26, 2018
1 Reply



Hi [Komal Sharma](#), can you please send me a screenshot of the error, so that I can help you.

[Nishan Aryal](#)

657 2.4k 1.4m

1 Jul 31, 2018



Sir in the mail template images are not showing please help me

[Saheb Jana](#)

1691 201 0

1 May 15, 2018
1 Reply

Please check the image src



Thank you for your article,nice

[Nasrin Bagheri](#)

1889 3 0

Dec 20, 2017

1 0 Reply



Thank you for your article nice post

[Adha Dimas](#)

1865 27 0

Nov 22, 2017

1 0 Reply



Nice Post about Email Template

[Rijwan Ansari](#)

1521 371 11.2k

Jul 21, 2017

1 0 Reply



Nice Article.....

[Thiruppathi R](#)

283 7.2k 1.1m

May 31, 2017

2 1 Reply



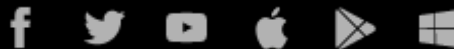
Thanks [Thiruppathi R](#) for reading my articles.

[Nishan Aryal](#)

657 2.4k 1.4m

Jun 01, 2017

1



[About Us](#) [Contact Us](#) [Privacy Policy](#) [Terms](#) [Media Kit](#) [Sitemap](#) [Report a Bug](#) [FAQ](#) [Partners](#)

[C# Tutorials](#) [Common Interview Questions](#) [Stories](#) [Consultants](#) [Ideas](#) [Certifications](#)

©2020 C# Corner. All contents are copyright of their authors.