



# Upload And Display Image In ASP.NET Core 3.1



Farhan Ahmed



Updated date, Feb 03, 2020



45.4k



5



14

[Download Free .NET & JAVA Files API](#)[Try Free File Format APIs for Word/Excel/PDF](#)

In this article, I will explain how to upload and display images. This article will help you to understand image tag helper and IFormFile in asp.net core. Image Tag Helper enhances the `img` tag to provide cache busting behavior for static image files. A unique cache-busting string is appended as a query parameter to the image source. The file on the host web server changes, a unique request URL is generated that includes the updated request parameter. IFormFile which is a C# representation of the file is used to process or save the file.

Here is the complete [code](#).

This is how we display images in asp.net core.

```
01. 
```

## Step 1

Start up Visual Studio 2019. Now click on create new project and Choose ASP.NET Core Web Application and click on "Next".

After clicking next, another wizard will open. Under the project name, give a meaningful name to your project and click on create



That will open up another new wizard. Select ASP.Net Core 3.0 from the dropdown. If not, select default. Choose Web Application (Model-View-Controller) template and click on create which will create ASP.Net Core Application.



## Step 2

Now right click on Models folder and “Add” class and name it Employee.

```
01. using System;
02. using System.Collections.Generic;
03. using System.ComponentModel.DataAnnotations;
04. using System.Linq;
05. using System.Threading.Tasks;
06.
07. namespace MvcCoreUploadAndDisplayImage_Demo.Models
08. {
09.     public class Employee
10.     {
11.         [Key]
12.         public int Id { get; set; }
13.
14.         [Required(ErrorMessage = "Please enter first name")]
15.         [Display(Name = "First Name")]
16.         [StringLength(100)]
17.         public string FirstName { get; set; }
18.
19.         [Required(ErrorMessage = "Please enter last name")]
```

```

22.         public string LastName { get; set; }
23.
24.         [Display(Name = "Full Name")]
25.         public string FullName { get; set; }
26.
27.         [Required(ErrorMessage = "Please enter age")]
28.         public int Age { get; set; }
29.
30.         [Required(ErrorMessage = "Please choose gender")]
31.         public string Gender { get; set; }
32.
33.         [Required(ErrorMessage = "Please enter position")]
34.         public string Position { get; set; }
35.
36.         [Required(ErrorMessage = "Please enter office")]
37.         public string Office { get; set; }
38.
39.         [Required(ErrorMessage = "Please enter salary")]
40.         public int Salary { get; set; }
41.
42.         [Required(ErrorMessage = "Please choose profile image")]
43.         public string ProfilePicture { get; set; }
44.
45.         public Employee()
46.         {
47.             FullName = FirstName + " " + LastName;
48.         }
49.     }
50. }

```

### Step 3

Right click on project "Add" folder ViewModels and "Add" class EmployeeViewModel.

```
01 using Microsoft.AspNetCore.Http;
```

```
04. namespace MvcCoreUploadAndDisplayImage_Demo.ViewModels
05. {
06.     public class EmployeeViewModel
07.     {
08.         [Required(ErrorMessage = "Please enter first name")]
09.         [Display(Name = "First Name")]
10.         public string FirstName { get; set; }
11.
12.         [Required(ErrorMessage = "Please enter last name")]
13.         [Display(Name = "First Name")]
14.         public string LastName { get; set; }
15.
16.         [Required(ErrorMessage = "Please enter age")]
17.         public int Age { get; set; }
18.
19.         [Required(ErrorMessage = "Please choose gender")]
20.         public string Gender { get; set; }
21.
22.         [Required(ErrorMessage = "Please enter position")]
23.         public string Position { get; set; }
24.
25.         [Required(ErrorMessage = "Please enter office")]
26.         public string Office { get; set; }
27.
28.         [Required(ErrorMessage = "Please enter salary")]
29.         public int Salary { get; set; }
30.
31.         [Required(ErrorMessage = "Please choose profile image")]
32.         [Display(Name = "Profile Picture")]
33.         public IFormFile ProfileImage { get; set; }
34.     }
35. }
```

#### Step 4

```
01. using System;
02. using System.Collections.Generic;
03. using System.Text;
04. using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
05. using Microsoft.EntityFrameworkCore;
06. using MvcCoreUploadAndDisplayImage_Demo.Models;
07.
08. namespace MvcCoreUploadAndDisplayImage_Demo.Data
09. {
10.     public class ApplicationDbContext : IdentityDbContext
11.     {
12.         public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
13.             : base(options)
14.         {
15.         }
16.
17.         public DbSet<Employee> Employees { get; set; }
18.     }
19. }
```

## Step 5

Now create images folder under wwwroot folder of the project.

## Step 6

Now go to tools, select nuget package manager, then click on package manager console. It will bring the console window below in Visual Studio 2019. Now write the following commands to enable migration.

- add-migration InitialModel (add migration can be anything you wish to)
- update-database

Now open HomeController which is added when we created new project. Write the following code for Index and New IActionResult methods.

## Controller Complete Code

```
01. using Microsoft.AspNetCore.Hosting;
02. using Microsoft.AspNetCore.Mvc;
03. using Microsoft.EntityFrameworkCore;
04. using MvcCoreUploadAndDisplayImage_Demo.Data;
05. using MvcCoreUploadAndDisplayImage_Demo.Models;
06. using MvcCoreUploadAndDisplayImage_Demo.ViewModels;
07. using System;
08. using System.IO;
09. using System.Threading.Tasks;
10.
11. namespace MvcCoreUploadAndDisplayImage_Demo.Controllers
12. {
13.     public class HomeController : Controller
14.     {
15.
16.         private readonly ApplicationDbContext dbContext;
17.         private readonly IWebHostEnvironment webHostEnvironment;
18.         public HomeController(ApplicationDbContext context, IWebHostEnvironment hostEnvironment)
19.         {
20.             dbContext = context;
21.             webHostEnvironment = hostEnvironment;
22.         }
23.
24.         public async Task<IActionResult> Index()
25.         {
26.             var employee = await dbContext.Employees.ToListAsync();
27.             return View(employee);
28.         }
29.
```

```

32.         return View();
33.     }
34.
35.     [HttpPost]
36.     [ValidateAntiForgeryToken]
37.     public async Task<IActionResult> New(EmployeeViewModel model)
38.     {
39.         if (ModelState.IsValid)
40.         {
41.             string uniqueFileName = UploadedFile(model);
42.
43.             Employee employee = new Employee
44.             {
45.                 FirstName = model.FirstName,
46.                 LastName = model.LastName,
47.                 FullName = model.FirstName + " " + model.LastName,
48.                 Gender = model.Gender,
49.                 Age = model.Age,
50.                 Office = model.Office,
51.                 Position = model.Position,
52.                 Salary = model.Salary,
53.                 ProfilePicture = uniqueFileName,
54.             };
55.             dbContext.Add(employee);
56.             await dbContext.SaveChangesAsync();
57.             return RedirectToAction(nameof(Index));
58.         }
59.         return View();
60.     }
61.
62.     private string UploadedFile(EmployeeViewModel model)
63.     {
64.         string uniqueFileName = null;
65.
66.         if (model.ProfileImage != null)

```

```

69.         uniqueFileName = Guid.NewGuid().ToString() + "_" + model.ProfileImage.FileName;
70.         string filePath = Path.Combine(uploadsFolder, uniqueFileName);
71.         using (var fileStream = new FileStream(filePath, FileMode.Create))
72.         {
73.             model.ProfileImage.CopyTo(fileStream);
74.         }
75.     }
76.     return uniqueFileName;
77. }
78. }
79. }

```

## Step 8

Right click on New IActionResult method. "Add" view with default name "New". Write the following code.

```

01. @model MvcCoreUploadAndDisplayImage_Demo.ViewModels.EmployeeViewModel
02.
03. @{
04.     ViewData["Title"] = "New";
05. }
06.
07. <div class="col-md-10 mx-auto py-4">
08.     <div class="card">
09.         <div class="card-header bg-primary text-uppercase text-white">
10.             <h4>Add New </h4>
11.         </div>
12.         <div class="card-body">
13.             <form enctype="multipart/form-data" asp-action="New">
14.                 <div class="row">
15.                     <div class="col-md-6">
16.                         <div class="form-group">
17.                             <label asp-for="FirstName" class="control-label"></label>
18.                             <input asp-for="FirstName" class="form-control" />

```



```

20.         </div>
21.     </div>
22.     <div class="col-md-6">
23.         <div class="form-group">
24.             <label asp-for="LastName" class="control-label"></label>
25.             <input asp-for="LastName" class="form-control" />
26.             <span asp-validation-for="LastName" class="text-danger"></span>
27.         </div>
28.     </div>
29. </div>
30. <div class="row">
31.     <div class="col-md-6">
32.         <div class="form-group">
33.             <label asp-for="Gender" class="control-label"></label>
34.             <select asp-for="Gender" class="form-control">
35.                 <option value="">Choose Gender</option>
36.                 <option value="Male">Male</option>
37.                 <option value="Female">Female</option>
38.             </select>
39.             <span asp-validation-for="Gender" class="text-danger"></span>
40.         </div>
41.     </div>
42.     <div class="col-md-6">
43.         <div class="form-group">
44.             <label asp-for="Age" class="control-label"></label>
45.             <input asp-for="Age" class="form-control" />
46.             <span asp-validation-for="Age" class="text-danger"></span>
47.         </div>
48.     </div>
49. </div>
50. <div class="row">
51.     <div class="col-md-6">
52.         <div class="form-group">
53.             <label asp-for="Position" class="control-label"></label>
54.             <input asp-for="Position" class="form-control" />
55.             <span asp-validation-for="Position" class="text-danger"></span>

```

```

57.         </div>
58.         <div class="col-md-6">
59.             <div class="form-group">
60.                 <label asp-for="Office" class="control-label"></label>
61.                 <input asp-for="Office" class="form-control" />
62.                 <span asp-validation-for="Office" class="text-danger"></span>
63.             </div>
64.         </div>
65.     </div>
66.     <div class="row">
67.         <div class="col-md-6">
68.             <div class="form-group">
69.                 <label asp-for="Salary" class="control-label"></label>
70.                 <input asp-for="Salary" class="form-control" />
71.                 <span asp-validation-for="Salary" class="text-danger"></span>
72.             </div>
73.         </div>
74.         <div class="col-md-6">
75.             <div class="form-group">
76.                 <label asp-for="ProfileImage" class="control-label"></label>
77.                 <div class="custom-file">
78.                     <input asp-for="ProfileImage" class="custom-file-
79. input" id="customFile">
80.                         <label class="custom-file-
81. label" for="customFile">Choose file</label>
82.                     </div>
83.                     <span asp-validation-for="ProfileImage" class="text-danger"></span>
84.                 </div>
85.             </div>
86.             <div class="form-group">
87.                 <input type="submit" value="submit" class="btn btn-sm btn-primary rounded-
0 text-uppercase" />
88.                 <a asp-action="Index" class="btn btn-sm btn-primary rounded-0 text-uppercase">
89. <i class="fas fa-fast-backward"></i> Back to List</a>
90.             </div>

```

```

90.     </div>
91. </div>
92. </div>
93.
94. @section scripts{
95.     <script type="text/javascript">
96.         // Add the following code if you want the name of the file appear on select
97.         $(".custom-file-input").on("change", function () {
98.             var fileName = $(this).val().split("\\").pop();
99.             $(this).siblings(".custom-file-label").addClass("selected").html(fileName);
100.         });
101.     </script>
102. }

```

## Step 9

Right click on Index IActionResult method "Add" view with default name "Index." If you don't have it then write the following code.

```

01. @model IEnumerable<MvcCoreUploadAndDisplayImage_Demo.Models.Employee>
02.
03. @{
04.     ViewData["Title"] = "Home Page";
05. }
06.
07. <h4 class="text-center text-uppercase">List of employee</h4>
08.
09. <p>
10.     <a asp-action="New" class="btn btn-sm btn-primary rounded-0 text-uppercase"><i class="fas fa-
plus-circle"></i> Add New</a>
11. </p>
12. <table class="table table-bordered">
13.     <thead>
14.         <tr>
15.             <th>@Html.DisplayNameFor(model => model.FullName)</th>
16.             <th>@Html.DisplayNameFor(model => model.Gender)</th>

```

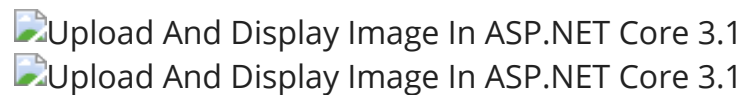
```

19.         <th>@Html.DisplayNameFor(model => model.Office)</th>
20.         <th>@Html.DisplayNameFor(model => model.Salary)</th>
21.     </tr>
22. </thead>
23. <tbody>
24.     @foreach (var employee in Model)
25.     {
26.         <tr>
27.             <td>
28.                 
32.                 @employee.FullName
33.             </td>
34.             <td>@employee.Gender</td>
35.             <td>@employee.Age</td>
36.             <td>@employee.Office</td>
37.             <td>@employee.Position</td>
38.             <td>@employee.Salary</td>
39.         </tr>
40.     }
41. </tbody>
42. </table>

```

## Step 10

Build and Run your application by pressing ctrl+F5



Brought to you by: [Embed Analytics and Dashboards into your product with a JavaScript SDK. Free trial.](#)

## Getting Started With ASP.NET Core 3.1 - Part One

ASP.NET Core

ASP.NET Core 3.1

Display Image In ASP.NET Core 3.1

Image In ASP.NET Core 3.1

Upload Image In ASP.NET Core 3.1



Farhan Ahmed *TOP 100*

ASP.NET Full-Stack Developer | Blogger



87



4.5m



2



14



5



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



Tank you! solved a big problem of mine!

[Otavio Rockenbach](#)

1906 7 0



May 14, 2020



1



0

Reply



Your tutorial is quite self explanatory.

[Aluh Johnson](#)

1910 3 0



May 08, 2020



1



1

Reply



Thank you Aluh johnson. Great to hear this.

[Farhan Ahmed](#)

87 21.4k 4.5m



May 08, 2020



0

Very nice article about image upload technique



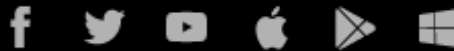
Hello could u please help me . I want to Create Edit Action in order Edit Multiple Images Please Help me out

**Muhammad Sami**

1901 12 0

Apr 04, 2020

0 0 Reply



[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.