**ASP.Net Core MVC: Drag and Drop multiple File upload using jQuery**

In this article I will explain with an example, how to upload multiple files with Drag and Drop in HTML5 supported browsers using jQuery and AJAX in ASP.Net Core MVC.

Multiple files such as Image, PDF, Word, Excel, etc. and also large Music and Video files such as MP3 and MP4 will be easily uploaded by simple Drag and Drop of files in HTML5 supported browsers using the [jQuery FileDrop](https://github.com/weixiyen/jquery-filedrop) plugin in ASP.Net Core MVC.

**Note**: For beginners in ASP.Net MVC Core, please refer my article [ASP.Net MVC Core Hello World Tutorial with Sample Program example](https://www.aspsnippets.com/Articles/ASPNet-MVC-Core-Hello-World-Tutorial-with-Sample-Program-example.aspx).

**View**

The following View consists of two HTML DIV elements and a Button. One DIV will be the area for dropping files and the other will display the list of uploaded files.

Configuring the Plugin

The [jQuery FileDrop](https://github.com/weixiyen/jquery-filedrop) plugin has been applied to the dropSection DIV and the Button ID is set into the fallback\_id property of the [jQuery FileDrop](https://github.com/weixiyen/jquery-filedrop) plugin.

**Note**: For storing and accessing Static Files such as JavaScript in ASP.Net Core, please refer my article: [Static Files (Images, CSS and JS files) in ASP.Net Core](https://www.aspsnippets.com/Articles/Static-Files-Images-CSS-and-JS-files-in-ASPNet-Core.aspx).

Plugin properties

**url** – URL of the Action method.

**paramname** – Name of the Action method parameter which will contain the File data.

**maxfiles** – Maximum number of simultaneous upload of Files.

**maxfilesize** – Maximum allowed File size in MB.

Plugin events

**dragOver** – Raised when File is dragged into the Drop section.

**dragLeave** – Raised when the File of dragged out of the Drop section.

**drop** – Raised when the File is dragged and dropped.

**uploadFinished** – Raised when the upload of a file is completed.

**afterAll** – Raised when upload of all files dragged together are completed.

Plugin validations

**allowedfileextensions** – Only the Files with the specified extension will be allowed to be uploaded.

**allowedfiletypes** – Only the Files with the specified MIME type (Content Type) will be allowed to be uploaded.

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Index</title>

<style type="text/css">

body {

font-family: Arial;

font-size: 10pt;

}

#dropSection {

height: 300px;

width: 600px;

background-color: skyblue;

}

#btnUpload {

display: none;

}

.active {

background-color: yellow !important;

}

</style>

</head>

<body>

<div id="dropSection">

</div>

<br />

Uploaded Files:

<hr />

<div id="uploadedFiles">

</div>

<input type="button" id="btnUpload" value="Upload" />

<script type="text/javascript" src='https://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.8.3.min.js'></script>

<script src="~/Scripts/filedrop.js"></script>

<script type="text/javascript">

$(function () {

$("#dropSection").filedrop({

fallback\_id: 'btnUpload',

fallback\_dropzoneClick: true,

url: '@Url.Action("Upload")',

//allowedfiletypes: ['image/jpeg', 'image/png', 'image/gif', 'application/pdf', 'application/doc'],

allowedfileextensions: ['.doc', '.docx', '.pdf', '.jpg', '.jpeg', '.png', '.gif'],

paramname: 'fileData',

maxfiles: 5, //Maximum Number of Files allowed at a time.

maxfilesize: 2, //Maximum File Size in MB.

dragOver: function () {

$('#dropSection').addClass('active');

},

dragLeave: function () {

$('#dropSection').removeClass('active');

},

drop: function () {

$('#dropSection').removeClass('active');

},

uploadFinished: function (i, file, response, time) {

$('#uploadedFiles').append(file.name + '<br />')

},

afterAll: function (e) {

//To do some task after all uploads done.

}

})

})

</script>

</body>

</html>

**Namespaces**

You will need to import the following namespace.

using System.IO;

**Controller**

The Controller consists of two Action methods.

Action method for handling GET operation

Inside this Action method, simply the View is returned.

Action method for handling POST operation

This Action method handles the POST operation i.e. uploading of Files.

It accepts the fileData parameter which is a collection of type IFormFile.

**Note**: The name of the IFormFile parameter must be equal to the paramname property value of the [jQuery FileDrop](https://github.com/weixiyen/jquery-filedrop) plugin.

First a check is performed whether Directory (Folder) exists if not then the Directory (Folder) is created inside the www Folder using IHostingEnvironment interface and then a loop is executed and one by one each file is saved to the Directory (Folder).

**Note**: For more details about IHostingEnvironment interface, please refer [Using IHostingEnvironment in ASP.Net Core](https://www.aspsnippets.com/Articles/Using-IHostingEnvironment-in-ASPNet-Core.aspx).

public class HomeController : Controller

{

    private IHostingEnvironment Environment;

    public HomeController(IHostingEnvironment \_environment)

    {

        Environment = \_environment;

    }

public IActionResult Index()

{

return View();

}

[HttpPost]

public IActionResult Upload(List<IFormFile> fileData)

{

        string path = Path.Combine(this.Environment.WebRootPath, "Uploads");

        if (!Directory.Exists(path))

        {

            Directory.CreateDirectory(path);

        }

foreach (IFormFile postedFile in fileData)

{

if (postedFile != null)

{

string fileName = Path.GetFileName(postedFile.FileName);

             string filePath = Path.Combine(path, fileName);

             using (FileStream stream = new FileStream(filePath, FileMode.Create))

            {

                postedFile.CopyTo(stream);

            }

}

}

return Content("Success");

}

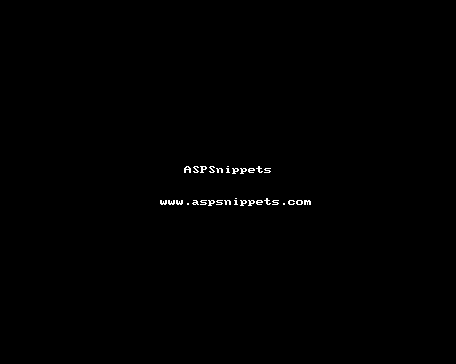
}

**Uploading Large files**

The maximum upload File Size Limit in ASP.Net Core is 28 MB, which means user will not be allowed to upload Files of size greater than 28 MB.

Thus if you want to upload larger files, please refer my article: [Uploading Large Files in ASP.Net Core](https://www.aspsnippets.com/Articles/Uploading-Large-Files-in-ASPNet-Core.aspx).

**Screenshot**



**Downloads**

[Drag\_Drop\_File\_Upload\_Core\_MVC.zip](https://www.aspsnippets.com/Handlers/DownloadFile.ashx?File=8265930f-f971-4d59-b334-6ea542be7325.zip)