# How to achieve upload progress bar in ASP.NET Core

## Introduction

We can use the FormData Object as the data payload, use ajax to send it, and subscribe “progress” event of XMLHttpRequest.upload to monitor upload progress.

And sample demonstrates how to achieve it in ASP.NET Core.

## Sample prerequisites

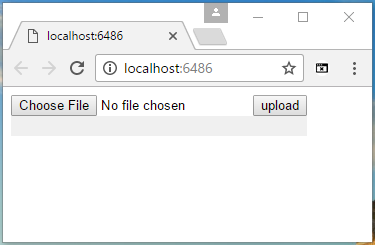
* IE 10 or above versions
* .NET Core 1.0 or later version(s). [[.NET Core + Visual Studio tooling](https://go.microsoft.com/fwlink/?LinkID=827546)]
* Microsoft Visual Studio 2015 update3 or above. [[Visual Studio 2015 installer](https://www.microsoft.com/en-sg/download/details.aspx?id=48146)]

## Building the sample

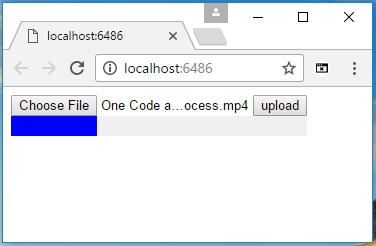
* Open the sample solution “**CSUploadProgressBar.sln**” using Visual Studio.
* Right click on the project “**CSUploadProgressBar**” and select Restore packages.
* Press **F6 Key** or select **Build -> Build Solution** from the menu to build the sample.

## Running the sample

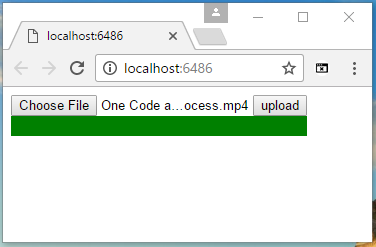
* Open the Sample solution using Visual Studio, then press F5 Key or select Debug -> Start Debugging from the menu.
* When the web application is running, you can see the page in browser.



* Click **Choose File** to select a large file, then click the **upload** buttom.



* When upload is finished, the progress color will be green.



## Using the code

At web.config, you need to set the maxRequestLength

<configuration>

<system.web>

<httpRuntime executionTimeout="100000" maxRequestLength="524288000" />

</system.web>

<system.webServer>

<security>

<requestFiltering>

<requestLimits maxAllowedContentLength="524288000" />

</requestFiltering>

</security>

……

</system.webServer>

</configuration>

And seem things you need config at Startup.cs again.

public void ConfigureServices(IServiceCollection services)

{

services.AddMvc();

services.Configure<FormOptions>(x =>

{

x.ValueLengthLimit = int.MaxValue;

x.MultipartBodyLengthLimit = int.MaxValue;

});

}

In Controller, this just an upload target address.

public class HomeController : Controller

{

public IActionResult Index()

{

return View();

}

public JsonResult UploadFile(IList<IFormFile> files)

{

return Json(new { state = 0, message = string.Empty });

}

}

At finally, this is the core code.

<input type="file" id="file" />

<button>upload</button>

<div style="position:relative;width:296px;background:#f0f0f0">

<div id="progress" style="background: blue; height: 20px;width:0"></div>

</div>

<script>

$("button").click(function () {

var progressEle = $("#progress");

progressEle.css("background", "blue");

var data = document.getElementById("file").files[0];

var formData = new FormData();

formData.append("files", data);

$.ajax({

url: "/Home/UploadFile",

data: formData,

processData: false,

contentType: false,

type: "POST",

xhr: function () {

var xhr = new window.XMLHttpRequest();

xhr.upload.addEventListener("progress", function (evt) {

if (evt.lengthComputable) {

var progress = Math.round((evt.loaded / evt.total) \* 100);

progressEle.width(progress + "%");

}

}, false);

return xhr;

},

success: function (data) {

if (data.state == 0) {

progressEle.css("background", "green");

}

}

});

});

</script>

## More information

Upload large file to ASP.NET Core

<https://docs.microsoft.com/en-us/aspnet/core/mvc/models/file-uploads>

<https://www.codeproject.com/Questions/1115853/Increase-upload-file-size-in-ASP-NET-core>

MDN document

<https://developer.mozilla.org/en-US/docs/Web/API/FormData>

<https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/Using_XMLHttpRequest#Submitting_forms_and_uploading_files>