**Single file upload*:***

@inject FileUploadService service

@inject IHostEnvironment host

<div class="row c-controls align-items-center">

<div class="col-md-6 ps-5">

<label>Maximum size of file</label>

</div>

<div class="col-md-6">

<**DropdownList** @bind-Value="fileSizeSingle"

**Source**="@(SelectListItem.CreateList(filesSize.Select(t => t + "M").ToArray()))" />

</div>

<div class="col-md-6 ps-5">

<label>Fileupload</label>

</div>

<div class="col-md-6">

<**Fileupload** **Title**="Single file upload" **Style**="width:130px;" **Service**="service" **OnChange**="ShowProrgres"

**OnFileUploadCompleted**="FileUploadCompleted" **RelatedPath**="@relatedPath"

**MaxAllowedSize**="Convert.ToInt32(filesSize[fileSizeSingle - 1] \* 1000\_000)" />

</div>

</div>

</div>

<**PopupWindow** @bind-Status="status" **AutoHide**="enableAutoHide">

<div class="p-2">

<**FileUploadProgress Service**="service" />

</div>

</**PopupWindow**>

@code

{

WindowStatus status;

int fileSizeMultiple = 1;

int fileCount = 1;

bool enableAutoHide;

string relatedPath;

double[] filesSize = new double[] { 0.5, 1, 2, 5, 10, 20, 50 };

int[] fileCounts = new int[] { 2, 3, 5, 8, 10 };

protected override void OnInitialized()

{

if (host.IsProduction())

relatedPath = host.ContentRootPath + "\\Data";

else

relatedPath = "F://Data";

base.OnInitialized();

}

void ShowProrgres(InputFileChangeEventArgs e)

{

enableAutoHide = false;

status = WindowStatus.Open;

}

void FileUploadCompleted()

{

enableAutoHide = true;

}

}

***Multiple file upload:***

@inject FileUploadService service

@inject IHostEnvironment host

<div class="row c-controls">

<div class="col-md-6">

<label>Maximum size of files</label>

</div>

<div class="col-md-6">

<**DropdownList** @bind-Value="fileSizeMultiple"

**Source**="@(SelectListItem.CreateList(filesSize.Select(t => t + "M").ToArray()))" />

</div>

<div class="col-md-6">

<label>Maximum numer of files file</label>

</div>

<div class="col-md-6">

<**DropdownList** @bind-Value="fileCount"

**Source**="@(SelectListItem.CreateList("Two", "Three", "Five", "Eight", "Ten"))" />

</div>

<div class="col-md-6">

<label>Fileupload</label>

</div>

<div class="text-center col-md-6">

<**Fileupload Multiple** **Title**="Multiple file upload" **Style**="width:145px;height:45px;"

**Service**="service" **MaxFilesCount**="Convert.ToInt32(fileCounts[fileCount - 1])"

**MaxAllowedSize**="Convert.ToInt32(filesSize[fileSizeMultiple - 1] \* 1000\_000)"

**OnChange**="ShowProrgres" **RelatedPath**="@relatedPath **OnFilesUploadCompleted**="FileUploadCompleted" />

</div>

</div>

<**PopupWindow** @bind-Status="status" **AutoHide**="enableAutoHide">

<div class="p-2">

<**FileUploadProgress Service**="service" />

</div>

</**PopupWindow**>

@code

{

WindowStatus status;

int fileSizeMultiple = 1;

bool enableAutoHide;

string relatedPath;

double[] filesSize = new double[] { 0.5, 1, 2, 5, 10, 20, 50 };

int[] fileCounts = new int[] { 2, 3, 5, 8, 10 };

protected override void OnInitialized()

{

if (host.IsProduction())

relatedPath = host.ContentRootPath + "\\Data";

else

relatedPath = "F://Data";

base.OnInitialized();

}

async Task SaveFileDataWithContent(InputFileChangeEventArgs file)

{

imageDataWithContent = await file.File.GetByteArrayAsync();

}

async Task SaveFileDataWithoutContent(InputFileChangeEventArgs file)

{

imageDataWithoutContent = await file.File.GetByteArrayAsync();

}

void ShowProrgres(InputFileChangeEventArgs e)

{

enableAutoHide = false;

status = WindowStatus.Open;

}

void FileUploadCompleted()

{

enableAutoHide = true;

}

}