# How to save Image to database and render it in ASP.NET Core

## Introduction

This sample demonstrates how to save image and render it with database in ASP.NET Core.

## Sample prerequisites

* Visual Studio 2017 or above. [[Visual Studio Home Page](https://www.visualstudio.com/)]
* SQL Server 2008 R2 or later version(s).

## Building the sample

* Create database with below structure.

CREATE TABLE [dbo].[Images]

(

[Id] UNIQUEIDENTIFIER NOT NULL PRIMARY KEY,

[Name] VARBINARY(200) NOT NULL,

[Data] VARBINARY(MAX) NOT NULL,

[Length] INT NOT NULL,

[Width] INT NOT NULL,

[Height] INT NOT NULL,

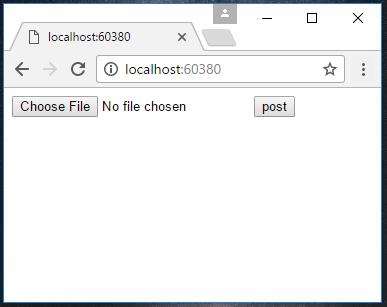
[ContentType] VARCHAR(50) NOT NULL

)

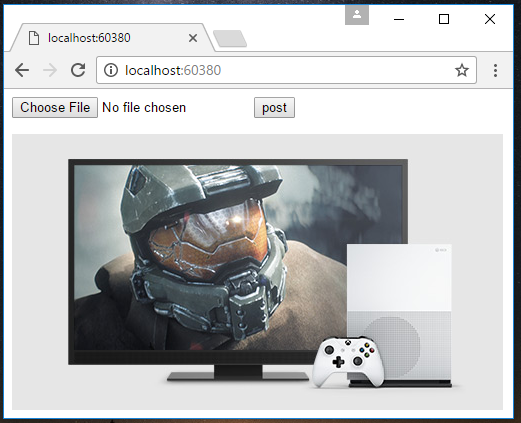
* Open the sample solution “**CSFileIOWithDBASPNETCore.sln**” using Visual Studio.
* Right click the project “**CSFileIOWithDBASPNETCore**” and select **Restore Packages**.
* Open the file “**appsettings.json**” in project “**CSFileIOWithDBASPNETCore**”, and in the section “**ConnectionStrings**”, set the “**SampleConnection**” value as your SQL Server database connection string.
* 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.



* Select an image to upload, and then click the **post** button.
* Then the image will be saved in database, and rendering all image in default page.



## Using the code

The HomeController.cs

public class HomeController : Controller

{

[HttpGet]

public IActionResult Index()

{

using (ImageDBContext dbContext = new ImageDBContext())

{

List<Guid> iamgeIds = dbContext.Images.Select(m => m.Id).ToList();

return View(iamgeIds);

}

}

[HttpPost]

public IActionResult UploadImage(IList<IFormFile> files)

{

IFormFile uploadedImage = files.FirstOrDefault();

if (uploadedImage == null || uploadedImage.ContentType.ToLower().StartsWith("image/"))

{

using (ImageDBContext dbContext = new ImageDBContext())

{

MemoryStream ms = new MemoryStream();

uploadedImage.OpenReadStream().CopyTo(ms);

System.Drawing.Image image = System.Drawing.Image.FromStream(ms);

Models.Image imageEntity = new Models.Image()

{

Id = Guid.NewGuid(),

Name = uploadedImage.Name,

Data = ms.ToArray(),

Width = image.Width,

Height = image.Height,

ContentType = uploadedImage.ContentType

};

dbContext.Images.Add(imageEntity);

dbContext.SaveChanges();

}

}

return RedirectToAction("Index");

}

[HttpGet]

public FileStreamResult ViewImage(Guid id)

{

using (ImageDBContext dbContext = new ImageDBContext())

{

Models.Image image = dbContext.Images.FirstOrDefault(m => m.Id == id);

MemoryStream ms = new MemoryStream(image.Data);

return new FileStreamResult(ms, image.ContentType);

}

}

}

Views/Home/Index.cshtml

@model IList<Guid>

<form action="/Home/UploadImage" enctype="multipart/form-data" method="post">

<input type="file" name="files" />

<button>post</button>

</form>

@foreach (var item in Model)

{

<img src="/Home/ViewImage/@item" />

}

## More information

Entity Framework core document

<https://docs.microsoft.com/en-us/ef/core/>

System.Drawing for .NET Core

<https://www.nuget.org/packages/CoreCompat.System.Drawing/1.0.0-beta001> .