حضرت خواجہ سیدنا معین الدین حسن چشتی سنجاری اجمیری رحمۃ اللہ علیہ

HOME

BLOGS   ⌄

1

FORUMS   ⌄

VIDEOS

SEARCH

INTERVIEW   ⌄

COMMUNITY   ⌄

CONTACT

🔍 | Ex Send Email ASP.Net

ASP.Net Core Razor Pages: Display PDF files from Database in View

Posted: 01 Jul 2022   Author: Mudassar Khan   Views: 11500   �

| ASP.Net | JavaScript | SQL Server | HTML5 | PDF | Core | Razor Pages |

⬇ **Download Code**

Disclaimer

This site makes use of Cookies. Please refer Privacy Policy for more details.

Allow   Reject

In this article I will explain with an example, how to upload and display PDF files from Database

PDF.js plugin

The PDF file will be displayed (rendered) in Browser using PDF.js JavaScript plugin.

The following files of PDF.js JavaScript plugin will be used.

1. pdf_viewer.min.css

2. pdf.min.js

3. pdf.worker.min.js

2

Database

This article makes use of a table named tblFiles whose schema is defined as follows.

Column Name	Data Type	Allow Nulls
id	int	☐
Name	varchar(50)	☐
ContentType	nvarchar(200)	☐
Data	varbinary(MAX)	☐
		☐

Note: You can download the database table SQL by clicking the download link below.

Download SQL file

Namespaces

You will need to import the follo

```
using System.IO;
using System.Data.SqlC
using System.Collectio
using Microsoft.AspNet
using Microsoft.Extensi
```

Disclaimer

This site makes use of Cookies. Please refer Privacy Policy for more details.

Allow   Reject

the records of the Files from database.

```
public class FileModel
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string ContentType { get; set; }
    public byte[] Data { get; set; }
}
```

3

Razor PageModel (Code-Behind)

The PageModel consists of three Handler methods.

Handler method for handling GET operation

Inside this Handler method, the GetFiles method is called and the public property Files is set.

Inside the GetFiles method, the records from the tblFiles table are fetched using DataReader and generic list of FileModel class objects is populated.

 Note: For details about reading Connection String from AppSettings.json, please refer my article, . Net Core: Read Connection String from AppSettings.json file.

Finally, the generic list of FileModel public property is returned to the Razor Page.

Handler method for handling POST operation for uploading Files

This Handler method gets called when a File is selected and the Upload Button is clicked, and it gets the uploaded file in the IF

 Note: In case the IFormFile pa
 ore: IFormFile always returns N

The uploaded file is converted t
inserted into the database table

Disclaimer

This site makes use of Cookies. Please refer Privacy Policy for more details.

Allow   Reject

After successful insert of the File, <u>RedirectToPage</u> is called which redirects to the IndexModel <u>Get</u>

When the <u>View</u> Link inside the HTML Table (Grid) is clicked, then the ID of the particular PDF file is sent to this Handler method and using the File ID, the File's binary data is fetched from the database.

Note: The following Handler method performs File display and hence the return type is set to <u>Json Result</u>. For more details, please refer my article <u>Using jQuery AJAX in ASP.Net Core Razor Pages</u>.

Finally, JSON is returned back to the <u>jQuery</u> <u>AJAX</u> function.

4

```csharp
public class IndexModel : PageModel
{
    public List<FileModel> Files { get; set; }
    private IConfiguration Configuration;

    public IndexModel(IConfiguration _configuration)
    {
        Configuration = _configuration;
    }

    public void OnGet()
    {
        this.Files = this.GetFiles();
    }

    public IActionResult OnPostUploadFile(IFormFile postedFile)
    {
        string fileName = Path.GetFileName(postedFile.FileName);
        string contentType = postedFile.ContentType;
        using (MemoryS
        {
            postedFile
            string con
nn");
            using (SqlC
            {
```

Disclaimer

This site makes use of Cookies. Please refer <u>Privacy Policy</u> for more details.

Allow   Reject

```
                           string query = "INSERT INTO tblFiles VALUES (@Name, @Con



                           cmd.connection = con;
                                cmd.Parameters.AddWithValue("@Name", fileName);
                                cmd.Parameters.AddWithValue("@ContentType", contentT
ype);

                                cmd.Parameters.AddWithValue("@Data", ms.ToArray());
                                con.Open();
                                cmd.ExecuteNonQuery();
                                con.Close();
                           }
                      }
                 }

            return RedirectToPage("Index");
        }


        public JsonResult OnPostGetPDF(int fileId)
        {
             byte[] bytes;
             string fileName, contentType;
             string constr = this.Configuration.GetConnectionString("MyCon
n");
             using (SqlConnection con = new SqlConnection(constr))
             {
                  using (SqlCommand cmd = new SqlCommand())
                  {
                       cmd.CommandText = "SELECT Name, Data, ContentType FROM t
blFiles WHERE Id=@Id";
                            cmd.Parameters.AddWithValue("@Id", fileId);
                            cmd.Connection = con;
                            con.Ope
                            using
                            {
                                 sd
                                 by
                                 co
                                 fi
```

5