

### P3 - ASP.NET Core

*Subsemnatul Craciun Ștefan Ciprian declar pe propria răspundere că acest cod nu a fost copiat din Internet sau din alte surse. Pentru documentare am folosit următoarele surse:*

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/razor-pages/?view=aspnetcore-3.1>

[ASP Net Core WCF EF Razor Pages - Tutorial](#)

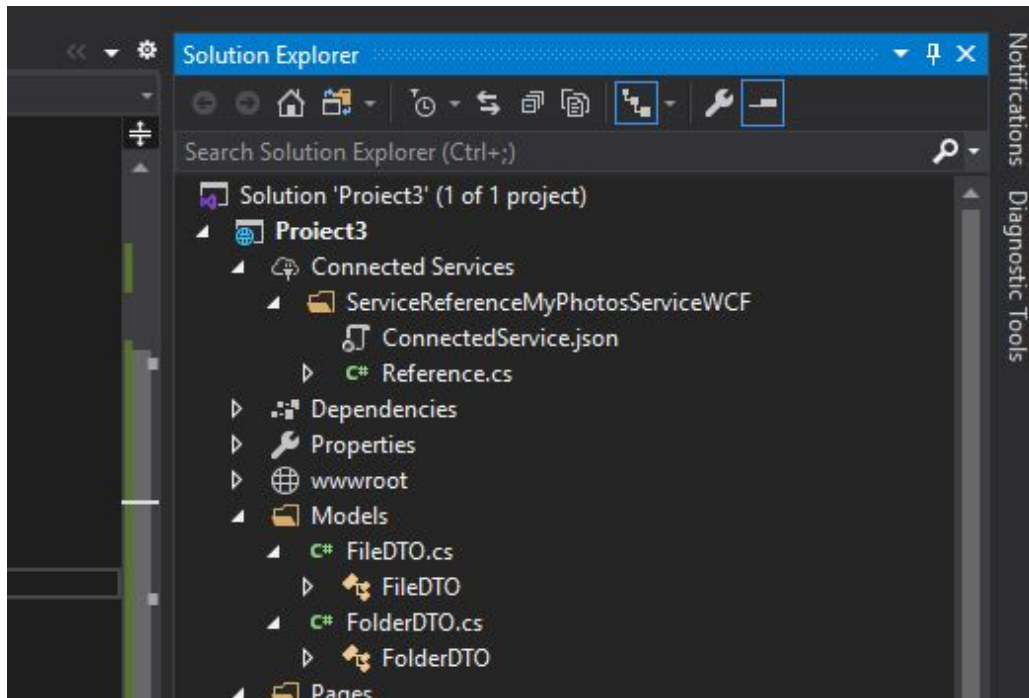
Pentru acest proiect am folosit informațiile din serviciul WCF din proiectul 2. Înainte sa fac conexiunea cu serviciul WCF din proiectul 2 am adaugat o nouă metoda în "ApiStatic.cs" pentru a afișa toate imaginile în funcție de id-ul unui Folder și anume:

```
public static List<File> GetFilesByFolderId(int id)
{
    using (Model1Container context = new Model1Container())
    {
        context.Configuration.ProxyCreationEnabled = false;
        var query = from ctx in context.Files
                     where ctx.FolderId == id
                     select ctx;
        return query.Distinct().ToList();
    }
}
```

Am facut update si in serviciul "MyPhotosServiceWCF" din proiectul 2:

```
public List<File> GetFilesByFolderId(int id)
{
    return ApiStatic.GetFilesByFolderId(id);
}
```

Am urmat pasii pentru a crea un proiect de tip ASP.Net Core, am pornit proiectul 2 și apoi am conectat Proiect 3 cu serviciul WCF:

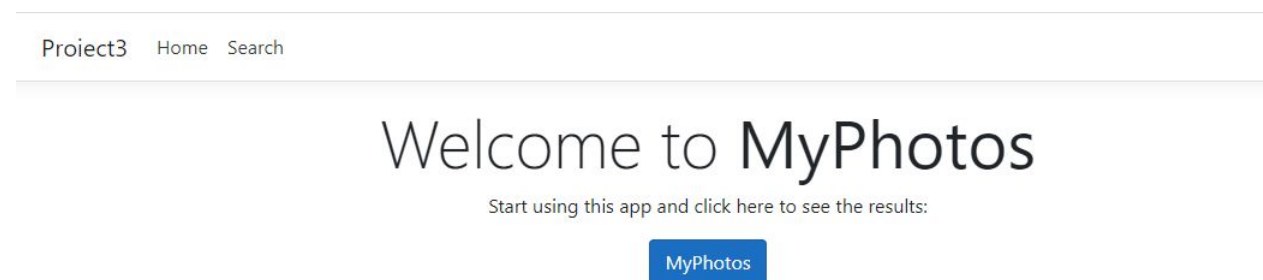


Următorul pas a fost sa creez folderul “Models” unde am făcut cate un DTO pentru File and Folders, fiind asemănătoare cu cele din Proiect 2.

In “Pages” am creat 3 directoare: Files, Folders, Search

- In Files am creat Index Razor page
- In Folder -> Index Razor page
- In Search -> Index Razor page

Cand vom rula prima data aplicația vom vedea un mesaj cu “Welcome to MyPhotos” si un buton care va lista câte directoare avem în aplicația de fotografii.



După cum am precizat cand vom face click pe butonul “MyPhotos” vom avea o lista cu directoarele din baza de date, directoare în care avem fotografii sau clipuri salvate:

Proiect3 Home Search

## MyPhotos > Folders

Click on the Folder name to see all the photos:

Id	Name	Location	Date
1	<a href="#">Demo</a>	C:\Users\Ciprian Craciun\source\repos\Proiect2-ObjectWCF\Demo	4/19/2020 1:27:30 AM
2	<a href="#">Demo2</a>	C:\Users\Ciprian Craciun\source\repos\Proiect2-ObjectWCF\Demo2	4/19/2020 3:28:33 PM
3	<a href="#">Test3</a>	C:\Users\Ciprian Craciun\source\repos\Proiect2-ObjectWCF\Test3	5/23/2020 2:10:32 AM

În fișierul “Index.cshtml.cs” in metoda “OnGetAsync” am returnat către client lista cu directoare din baza de date.

După cum se observa si in imagine, fiecare nume din tabel are un link construit cu parametrii **id** si **sortOrder**

```
</td>  
▼<td>  
  <a href="/Files?id=1&sortOrder=Name">Demo</a> == $0  
</td>
```

In acest fel cand vom face click pe numele unui director vom lista toate imaginile din acel director ordonate alfabetic:

## MyPhotos &gt; Files

FileName	FileDesc	FilePath	FileTags	FileDate
backend-encounterEntryNoteList	File Added just for test	C:\Users\Ciprian Craciun\source\repos\Proiect2-ObjectWCF\Demo2\backend-encounterEntryNoteList	test, tag, con	5/23/2020 2:38:47 AM
con1	Connexion photo	C:\Users\Ciprian Craciun\source\repos\Proiect2-ObjectWCF\Demo2\con1	Same, con	5/22/2020 10:10:35 PM
con2	Second connexion!	C:\Users\Ciprian Craciun\source\repos\Proiect2-ObjectWCF\Demo2\con2	Con, Con2	5/22/2020 10:11:21 PM
image-saved	This is other image for connexion with I	C:\Users\Ciprian Craciun\source\repos\Proiect2-ObjectWCF\Demo2\image-saved	con, test	5/23/2020 2:40:31 AM

[< Back to Folders](#)

Pentru partea de căutare am adăugat un buton în navigatie “Search”, va afișa o pagina goala cu un Form și un input de tip text.

Folosim aceasta pagina pentru utilizatorul care va dori sa caute o poza după un anumit cuvânt cheie.

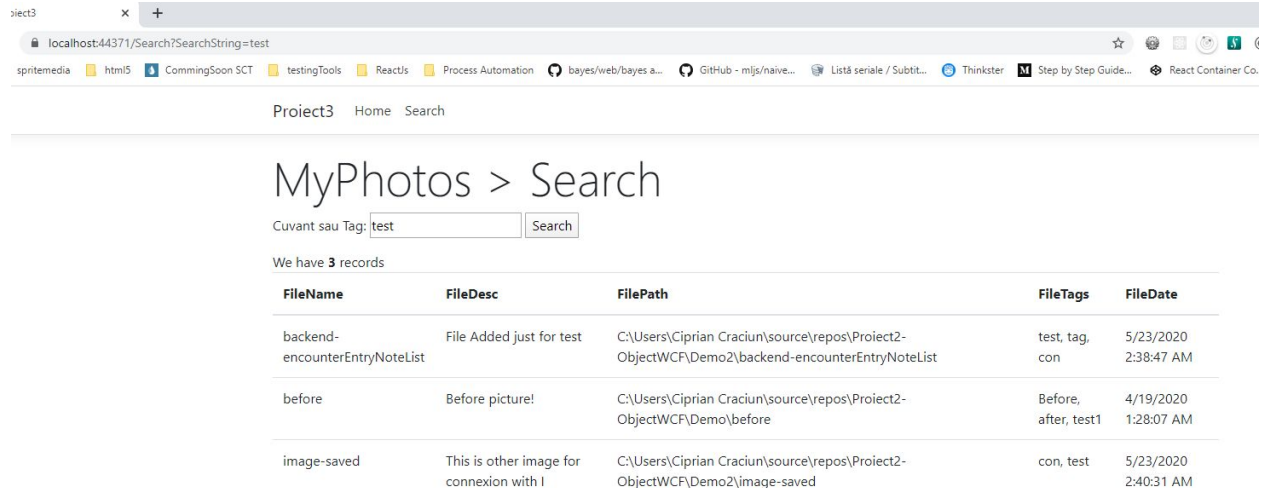
## MyPhotos &gt; Search

Cuvant sau Tag:  

We have 0 records

FileName	FileDesc	FilePath	FileTags	FileDate
----------	----------	----------	----------	----------

De asemenea afisam și câte imagini au fost găsite: “We have 3 records”

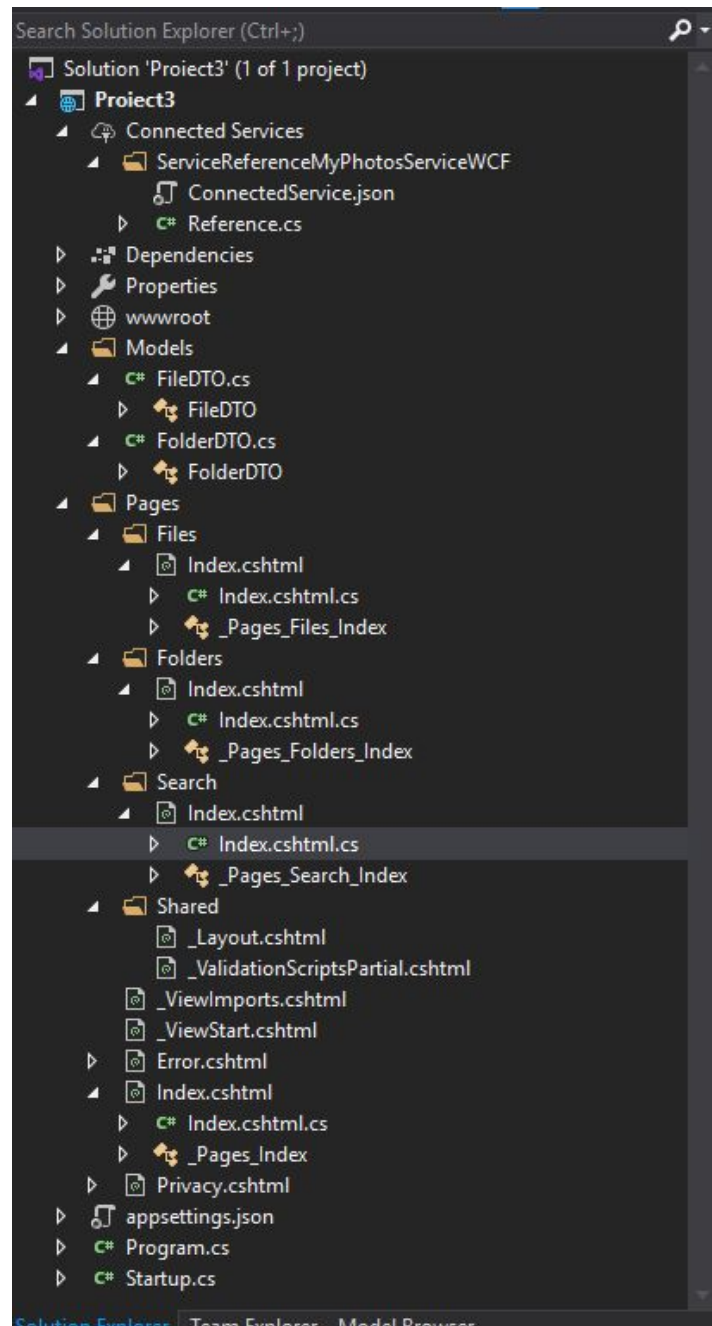


Pentru aceasta functionalitate folosesc un parametru “SearchString” care ia valoarea din From si apoi in Search -> Index.cshtml.cs voi avea:

```
[BindProperty(SupportsGet = true)]
public string SearchString { get; set; }
```

```
public async Task OnGetAsync()
{
    if (!string.IsNullOrEmpty(SearchString))
    {
        var items = await
client.SearchInDescAndTagsAsync(SearchString);
        foreach (var cc in items)
        {
            FileDTO files = new FileDTO();
            files.FolderId = cc.FolderId;
            files.FileName = cc.FileName;
            files.FileDesc = cc.FileDesc;
            files.FilePath = cc.FilePath;
            files.FileTags = cc.FileTags;
            files.FileDate = cc.FileDate;
            Files.Add(files);
        }
    }
}
```

Structura finală a proiectului arăta ca în imaginea de mai jos:



Am creat pe git si un folder numit **assets** unde am adaugat toate imaginile din acest document plus gif unde demonstrez cum funcționează proiectul.