

Dezvoltarea Aplicațiilor Web utilizând ASP.NET Core MVC

Laborator 3

EXERCITII:

Exercițiul 1:

Să se creeze un nou proiect (**VEZI Curs 3** - Crearea unei aplicații în ASP.NET Core 9.0). În cadrul proiectului să se adauge un nou Controller numit **ExamplesController** și să se implementeze următoarele cerințe, creând metodele aferente în acest Controller (Actions):

1. Să se adauge o rută numită **Concatenare** care să răspundă prin **/concatenare** urmată de 2 parametri de tip string. Să se afișeze valorile lor concatenate.
2. Să se adauge o rută numită **Produs** care primește 2 parametri de tip întreg. Al doilea parametru este opțional. Să se afișeze produsul celor 2 parametri. Dacă al doilea parametru lipsește se va afișa mesajul “Introduceți ambele valori”. Pagina se va accesa prin ruta: **/produs/param1/param2**
3. Să se adauge o rută numită **Operație** care primește 3 parametri astfel: doi de tip întreg și unul de tip string. Parametrul de tip string reprezintă o operație (plus, minus, ori, div) efectuată între parametrii de tip întreg. Să se afișeze pe ecran toți parametrii împreună cu operația, cât și rezultatul acestora. În cazul în care un parametru lipsește se va afișa pe ecran mesajul “Introduceți parametrul 1/2/3” – în funcție de parametrul care lipsește. De asemenea, toți parametrii sunt opționali. Ruta se va accesa prin URL-ul: **/operatie/param1/param2/op**
Parametrul op poate fi: “plus”, “minus”, “ori”, “div”

Exercițiul 2:

Presupunem că avem o aplicație în care există resursa **Student**. Să se adauge un Controller – **StudentsController** și toate rutele asociate operațiilor de tip C.R.U.D. – CREATE, READ, UPDATE, DELETE.

De exemplu: în **StudentsController** o să avem următoarele metode:

- **Index** – pentru afișarea tuturor studenților
- **Show** – afișarea unui singur student, în funcție de id-ul acestuia
- **Create** – crearea unui nou student
- **Edit** – editarea unui singur student
- **Delete** – ștergerea unui singur student

Rutele trebuie să conțină toți parametrii necesari operațiilor.
De asemenea, mare atenție să nu existe conflict între rute.

Explicații:

Index – afișarea tuturor studenților. Se vor realiza două rute, în două moduri:

Varianta 1: accesarea paginii în care se vor afișa toți studenții va avea loc folosind URL-ul: **/Students/Index**. Ruta o să se numească **StudentsIndex**

Varianta 2: accesarea paginii va avea loc folosind URL-ul: **students/all**
Ruta o să se numească **StudentsAll**

Show – afișarea unui singur student, în funcție de id-ul acestuia. Se va realiza o rută numită **StudentShow**, având URL-ul: **students/{id}**. Id-ul este un parametru opțional. În Controller se va afișa doar un mesaj: “Afișare student cu ID: ”

Create – crearea unui nou student. Se va realiza o rută numită **StudentCreate**, având URL-ul: **students/new**. În Controller se va afișa mesaj: “Creare student în baza de date”.

Edit – editarea unui singur student. Se va realiza o rută numită **StudentEdit**, având URL-ul: **students/{id}**. Id-ul este un parametru opțional. În Controller se va afișa doar un mesaj: “Editare student cu ID: ”.

Ce observați în acest caz?

Delete – stergerea unui singur student. Se va realiza o rută numită **StudentDelete**, având URL-ul: **students/{id}**. Id-ul este un parametru opțional. În Controller se va afișa doar un mesaj: “Ștergere student cu ID: ”

Exercițiul 3:

Să se creeze Controllerul **SearchController**. În cadrul acestuia să se adauge metoda: **NumarTelefon**.

Metoda reprezintă o pagină de căutare, care primește un parametru după care se face căutarea. Adăugați metoda, împreună cu toți parametrii ei, cât și rutele asociate, împreună cu următoarele constrângeri. În final, afișați valorile parametrilor în pagină.

- Ruta pentru căutarea numărului de telefon este de forma: **search/telefon/{telefon}**. Ruta primește un singur parametru reprezentat de un șir de caractere de lungime = 10 caractere. Toate caracterele sunt cifre, șirul începe cu caracterul 0, după caracterul 0 trebuie să fie obligatoriu încă 9 cifre (<https://regexr.com/>).

Ce observați în cazul în care numărul de telefon lipsește din request sau în cazul în care acesta are lungimi diferite? Aceste verificări se vor realiza la nivel de Controller.

De exemplu:

```
public string NumarTelefon(string? telef)
{
    if (telef == null)
    {
        return "Introduceti numarul de telefon";
    }

    if (telef.Length < 10)
        return "Numarul de telefon nu are suficiente cifre";

    else if (telef.Length > 10)
        return "Numarul de telefon are prea multe cifre";
    else
        return "Cautare pentru nr de telefon: " + telef;
}
```

Exemplu integrare expresie regulată:

```
app.MapControllerRoute(
    name: "NrTelefon",
    pattern: "search/telefon/{telef:regex()}?",
    defaults: new { controller = "Search", action =
        "NumarTelefon" });
```