

Dezvoltarea Aplicatiilor Web utilizand ASP.NET Core MVC

Laborator 3

EXERCITII:

Exercitiul 1:

Sa se creeze un nou proiect (**VEZI Curs 3** - Crearea unei aplicatii in ASP.NET Core 6.0). In cadrul proiectului sa se adauge un nou Controller numit **ExamplesController** si sa se implementeze urmatoarele cerinte, creand metodele aferente in acest Controller (Actions):

1. Sa se adauge o ruta numita **Concatenare** care sa raspunda prin **/concatenare** urmata de 2 parametri de tip string. Sa se afiseze valorile lor concatenate.
2. Sa se adauge o ruta numita **Produs** care primeste 2 parametri de tip intreg. Al doilea parametru este optional. Sa se afiseze produsul celor 2 parametri. Daca al doilea parametru lipseste se va afisa mesajul "Introduceti ambele valori". Ruta se va accesa prin ruta: **/produs/param1/param2**
3. Sa se adauge o ruta numita **Operatie** care primeste 3 parametri astfel: 2 de tip intreg si unul de tip string. Parametrul de tip string reprezinta o operatie (plus, minus, ori, div) efectuata intre parametrii de tip intreg. Sa se afiseze pe ecran toti parametrii impreuna cu operatia, cat si rezultatul acestora. In cazul in care un parametru lipseste se va afisa pe ecran mesajul "Introduceti parametrul 1/2/3" – in functie de parametrul care lipseste. De asemenea, toti parametrii sunt optionali. Ruta se va accesa prin URL-ul: **/operatie/param1/param2/op**
Parametrul op poate fi: "plus", "minus", "ori", "div"

Exercitiul 2:

Presupunem ca avem o aplicatie in care exista resursa Student. Sa se adauge un Controller – **StudentsController** si toate rutele asociate operatiilor de tip C.R.U.D. – CREATE, READ, UPDATE, DELETE.

De exemplu: in **StudentsController** o sa avem urmatoarele metode:

- **Index** – pentru afisarea tuturor studentilor
- **Show** – afisarea unui singur student, in functie de id-ul acestuia
- **Create** – crearea unui nou student
- **Edit** – editarea unui singur student
- **Delete** – stergerea unui singur student

Rutele trebuie sa contina toti parametrii necesari operatiilor.
De asemenea, mare atentie sa nu existe conflict intre rute.

Explicatii:

Index – afisarea tuturor studentilor. Se vor realiza doua rute, in doua moduri.

Varianta 1: accesarea paginii in care se vor afisa toti studentii va avea loc folosind URL-ul: **/Students/Index**. Ruta o sa se numeasca **StudentsIndex**

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

Show – afisarea unui singur student, in functie de id-ul acestuia. Se va realiza o ruta numita **StudentShow**, avand URL-ul: **students/{id}**. Id-ul este un parametru optional. In Controller se va afisa doar un mesaj: “Afisare student cu ID: ”

Create – crearea unui nou student. Se va realiza o ruta numita **StudentCreate**, avand URL-ul: **students/new**. In Controller se va afisa mesaj: “Creare student in baza de date”.

Edit – editarea unui singur student. Se va realiza o ruta numita **StudentEdit**, avand URL-ul: **students/{id}**. Id-ul este un parametru optional. In Controller se va afisa doar un mesaj: “Editare student cu ID: ”.

Ce observati in acest caz?

Delete – stergerea unui singur student. Se va realiza o ruta numita **StudentDelete**, avand URL-ul: **students/{id}**. Id-ul este un parametru optional. In Controller se va afisa doar un mesaj: “Stergere student cu ID: ”

Exercitiul 3:

Sa se creeze Controllerul **SearchController**. In cadrul acestuia sa se adauge urmatoarele metode: **NumarTelefon, CNP**.

Fiecare metoda reprezinta o pagina de cautare care primeste un parametru dupa care se face cautarea. Adaugati metodele impreuna cu toti parametrii lor, cat si rutele asociate impreuna cu urmatoarele constrangeri si afisati valorile parametrilor in pagina:

- Ruta pentru cautarea numarului de telefon este de forma: **search/telefon/{telefon}**. Ruta primeste un singur parametru reprezentat de un sir de caractere de lungime = 10 caractere. Toate caracterele sunt cifre, sirul incepe cu caracterul 0, dupa caracterul 0 trebuie sa fie obligatoriu inca 9 cifre (<https://regexr.com/>).

Ce observati in cazul in care numarul de telefon lipseste din request sau in cazul in care acesta are lungimi diferite? Aceste verificari se vor realiza la nivel de controller.

De exemplu:

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

Exemplu integrare expresie regulata:

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

- Ruta CNP (/search/cnp/{cnp}) va primi un singur parametru reprezentat de un sir de caractere de lungime = 13. Toate caracterele sunt cifre. Sirul incepe cu 1,2,5 sau 6; (<https://regexr.com/>).

Ce observati?

Indicatii:

Pentru preluarea primei cifre:

```
String first_letter = cnp.Substring(0, 1);
```