

LEKCE 11

Refactoring, Tidbits, Delegates

Závěrečný úkol



Jak to jde?

Závěrečný projekt - Flow Odevzdávání

ToDoList Categories
Backend

2.12.

ToDoList Categories
Frontend

9.12.

ToDoList Categories
Final

16.12.

Úprava backendu,
aby podporoval kategorie

- Models
- Entity + Context
- Database Schema
 - Migrations
- DTOs
- Endpoint Controller
- Testy

Úprava frontendu,
aby podporoval kategorie

- Dashboard component
- EditItem component
- Opravy BE na základě zpětné vazby

Prostor pro vychytání chyb a vylepšení programu a kódu.

- Vylepšení vzhledu pro uživatele frontendu
- Opravy FE na základě zpětné vazby
- Refactoring
 - jednoduchost
 - přehlednost
 - pojmenování

Konfigurace

Proč je konfigurace důležitá?

- Odděluje kód od nastavení
- Usnadňuje údržbu a nastavování parametrů
- Flexibilita nasazení

Prakticky v ToDoList aplikaci

- Nastavení logování
- Nastavení databázového připojení (vývojová x produkční)
- Nastavení připojení HTTP klienta na backend

Validace



Prakticky v ToDoList aplikaci

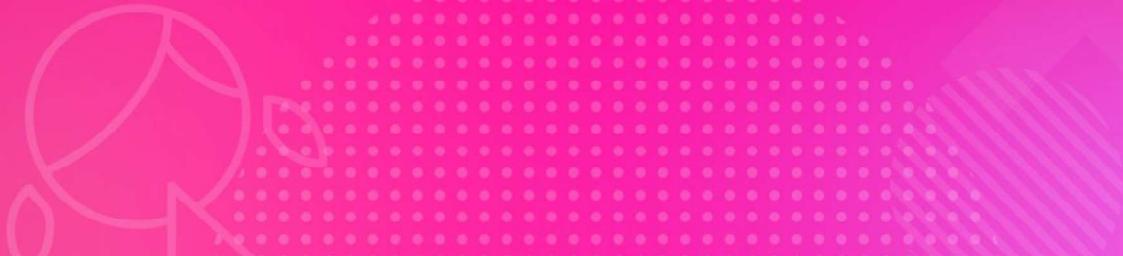
- Klientská validace v EditToDoItem (ToDoItemView)
- Serverová validace - REST API (DTOs)
- Serverová validace - databáze (EF Core ToDoItem Entity)

Dokumentace OpenAPI



ProducesResponseType<Type>(StatusCode)

Delegáti



```
public delegate void MessageTakingDelegate(string message);  
public delegate int NumberOperationDelegate(int n1, int n2);
```

Jak se používají?

Deklarace delegáta

```
public delegate void MyDelegate(string input);
```

Deklarace a Inicializace delegátového typu

```
public MyDelegate mujDelegat = MojeMetodaSplnujiciDelegat;
```

Invokace (volání)

```
mujDelegat("zprava");  
mujDelegat.Invoke("zprava");
```

Action, Func

(předpřipravení delegáti)

Action<ParameterType1, ...>

Func<ParameterType1, ..., ReturnType>

Lambda výrazy

(anonymní funkce)

Context



(input-parameters) => expression

(input-parameters) => { <sequence-of-statements> }

Kdy použít co?

Použití	Delegáti	Anonymní funkce (lambdy)
Jednorázové volání	✗ Příliš složité	✓ Jednoduché a přehledné
Více metod (multicast)	✓ Delegát zvládá připojení více metod	✗ Není vhodné
Znovupoužitelnost	✓ Lepší pro logiku používanou na více místech	✗ Není ideální, pokud je třeba opakovaně používat
Jednoduchost	✗ Vyžaduje více kódu	✓ Krátké a snadné na psaní
Podpora closure	✗ Delegáti nezachycují okolní kontext	✓ Lambdy umožňují použití closure