

UserStories Part 7

(Gemme userstories i json fil)

Denne opgave har til formål at vise en form for persistens i forbindelse med programmet UserStoryApp'en nemlig json gemt i en fil.

Trin 1: Opret et UserStoryService interface

Du skal lave et interface til din UserStoryService, som er et fælles interface for en række forskellige persistence-former (Mock-data, Json i én fil, database, EntityFramework).

Herved vil du komme til at implementere noget der minder om Adaptor Design Pattern

Lav et Interface ('IUserStoryPersistence') i mappen services, som indeholder følgende metoder:

GetAllUS – henter alle UserStories og returnerer dem i en collection (ICollection).

GetOneUS – henter en UserStory ud fra et id og returnerer UserStory'en. Hvis der ikke findes en UserStory med det id kastes en exception.

GetAllUSOrderByBusinessValues – henter alle UserStories og returnerer dem i en collection, hvor de UserStories med højest BusinessValues ligger først.

AddUS – Tilføjer en ny UserStory til collection og returnere sand hvis det lykkedes ellers falsk

DeleteUS – Sletter en UserStory ud fra et Id fra collection og returnerer den UserStory der slettes. Hvis der ikke findes en UserStory med det id kastes en exception.

UpdateUS – Opdaterer en UserStory ud fra et id med nye værdier. Der returneres sand hvis det lykkedes ellers falsk.

Trin 2: Implementer et MockUserStoryService

I den oprindelige UserStory opgave skulle du lave en MockUserStories. Denne implementering skal du adaptere over i det nye interface ('IUserStoryPersistence').

I mappen 'services' laver du en klasse 'MockUserStoryPersistence', der implementerer interfacet ' IUserStoryPersistence'.

Du skal i Startup.cs klassen tilføje din service klasse som en AddSingleton til services i ConfigureServices: a la

```
services.AddSingleton< IUserStoryPersistence, MockUserStoryPersistence>();
```

I dine 'controller' klasser bag dine razor pages skal du få injected et objekt af servicen a. la.

```
public CreateUserStoryModel(IUserStoryPersistence service){  
    _userStoryService = service;  
}
```

Tilpas din applikation til at benytte det nye interface.

Kør din applikation og se hvordan den virker.

Trin 3: Implementer et JsonUserStoryPersistence

I mappen 'services' laver du en klasse 'JsonUserStoryPersistence', der implementerer interfacet ' IUserStoryPersistence', men benytter at dataene skrives i Json format til en fil, hhv. læses fra filen.

Se i Razor-note 2 ss. 13-18, hvorledes du kan skriv hhv. læse i json format til en fil.

I Startup.cs skal du ændre din service til at benytte den nye service a.la:

```
services.AddSingleton< IUserStoryPersistence, JsonUserStoryPersistence >();
```

Mere skal du ikke ændre.

Kør din applikation og se at den virker.

Dog kan der blive et problem første gang, hvor der endnu ikke findes en json-fil.

Hvad er forskellen på de to implementeringer?

Hvorledes giver det sig udtryk?

Ekstra: Implementer et DBUserStoryPersistence

Lav en tabel til UserStories og implementer en ny service

DBUserStoryPersistence der laver sqlconnection til din database