

# UserStories Part 1

Dette er en repetitions opgave til Razor Pages (basic). Der er fokus på arkitektur og brug af MockData og Service (som Singleton).

## 1. Nyt projekt: UserStories

Opret et nyt projekt: *UserStories* af typen 'ASP.Net Core Web Application' - vælg 'Web Application'

## 2. Afprøvning

Kør projektet og verificer at templateen virker.

## 3. Mappestruktur

Under projektet oprettes følgende mapper:

- *Models*
- *Services*
- *MockData*

## 4. Class UserStory

I mappen *Models* oprettes en klasse *UserStory* med følgende properties:

- *Id* (int)
- *Title* (string)
- *Description* (string)

Der skal være et *private static nextId* felt til automatisk at optælle *Id* (sikre at *Id* er unik). Konstruktøren skal tage to string som parameter og den skal initialisere *Id*, *Title* og *Description*, ved brug af *nextId* og parametrene. Der skal også være en default konstruktør (dvs konstruktør uden parametre).

## 5. MockUserStories

I mappen *MockData* oprettes en ny klasse *MockUserStories*. Klassen skal indeholde en statisk liste af *UserStory* objekter og initialiseres ala:

```
private static List<UserStory> userStories = new List<UserStory>()
{
    new UserStory("Create Story", "As P.O I want to create a new Story So ..."),
    new UserStory("Edit Story", "As P.O I want to edit a Story So ..."),
    new UserStory("Move Story", "As team member I want to move a Story So ..."),
    new UserStory("Delete Story", "As team member I want to delete a Story So ...")
};
```

Der skal desuden være en statisk metode *GetMockUserStories()* der returnere listen.

## 6. **UserStoryService**

I mappen Service oprettes en ny klasse *UserStoryService*.

Klassen skal indeholde en liste af UserStory objekter:

- *private List<UserStory> userStories;*

Og en konstruktør der initialiserer *userStories* ved et kald af *GetMockUserStories()* fra *MockUserStories*.

*Samt en metode GetUserStories() der returnerer userStories.*

## 7. **Startup.cs - registrering af UserStoryService**

UserStoryService skal registreres i Startup klassen, det gøres ved at tilføje

- *services.AddSingleton<UserStoryService, UserStoryService>();*

under linien: *services.AddRazorPages();* i *ConfigureServices(IServiceCollection services)* metoden

## 8. **Razor Pages – UserStories**

I mappen Pages oprettes en ny Razor Pages (empty) UserStories

- Indsæt '*<h1>User Stories</h1>*' i *UserStories.cshtml*
- Tilføj et nyt link i *\_layout.cshtml* filen til den nye side
- Kør og afprøv at link og page virker

## 9. **UserStoriesModel (UserStories.cshtml.cs)**

Vi skal nu have injected vores UserStoryService, så servicen kan benyttes til at hente listen af UserStories i MockData:

- Tilføj et instance field til servicen: *private UserStoryService userStoryService;*
- Tilføj en property til Listen af UserStories:

```
public List<UserStory> UserStories { get; private set; }
```

- Inject servicen i konstruktøren:

```
public UserStoriesModel(UserStoryService userStoryService)  
{  
    this.userStoryService = userStoryService;
```

```
}
```

- Opdater *OnGet()* metoden så den initialiserer listen *UserStories* med *MockData* ved at kalde *GetUserStories()* på servicen:

```
UserStories = userStoryService.GetUserStories();
```

## 10. Opdater Razor Page – UserStories

Vi skal have opdateret siden, så den viser alle vores *UserStory*. Tilføj en *foreach* løkke der indsætter alle *UserStory* i en list:

```
<ul>
  @foreach (var userstory in @Model.UserStories)
  {
    <li>
      <div>(@userstory.Id) @userstory.Title </div>
      <div>@userstory.Description</div>
    </li>
  }
</ul>
```

## 11. Afprøvning

Kør og afprøv at der er "hul igennem" og du kan få vist alle *UserStories* fra *MockData*:

