

IDENTIFIKATION: Sommerhus#6 / PELE

Overordnede mål

Ideen er at du skal kunne håndtere flere samtidige brugere i RazorPages.

Se i øvrigt opgave [Sommerhus#1](#) og [Sommerhus#2](#) og [Sommerhus#3](#) og [Sommerhus#4](#) og [Sommerhus#5](#)

Opgave 1: Lav en bruger

I projektet SommerhusLib tilføj en klasse 'User' med properties Name og IsAdmin.
(hvis du vil have flere bruger-roller end 'alm' og 'admin'; skal du lave en enum).

Husk at 'build'e projektet.

Opgave 2: Understøt Session i SommerhusApp'en

Du skal i program.cs understøtte at din applikation skal benytte Session.
Under builder tilføj:

```
builder.Services.AddSession();
```

og under app tilføj

```
app.UseSession();
```

I mappen Services lav en klasse SessionHelper, der her to metoder

```
public static User GetUser(HttpContext context)
public static void SetUser(User user, HttpContext context)
```

I GetUser-metoden skal du ud fra HttpContext – session hente en string og json deserialisere den til et User-objekt

```
String? s = context.Session.GetString("User");
Return JsonSerializer.Deserialize<UserService>(s);
```

I Set-user metoden skal du serialisere User-objektet og gemme det i session objektet

```
String s = JsonSerializer.Serialize(us);  
context.Session.SetString("User", s);
```

Opgave 3: Lav en login-side

I mappen pages/sommerhuse lav en ny RazorPage Login.

I login-controllerdelen lav to properties [BindProperty] til navn og password

Lav en OnPost-metode der tjekker navn og password, hvis det er OK lav et User-objekt og gem det i session – med din SessionHelper.

I Login-viewdelen lav en form med to felter til at indtaste navn og password, samt en knap til at logge ind.

Opgave 4: Lav et tjek på index-side

Du skal beskytte din index-side ved at tjekke om brugeren er logget ind ellers sendes vedkommende til login-siden.

Dvs. i OnGet læs Session objektet og hvis User-objektet er tomt redirect til login

Ekstra A: lav et tjek for om brugeren er administrator for at kunne benytte Create, Update eller Delete siderne (måske har du kun create)

Ekstra B: lav en klasse ValidateUser, der indeholder en liste af brugere samt deres password og roller samt en metode til at tjekke (Validere) om et navn og password er korrekt.

Benyt den i login-controller-delen til at godkende login.

Ekstra C: lad din ValidateUser være en service som tilføjes AddSingleton....

Ekstra D: Nu er det jo ikke så smart at have password liggende og flyde, så

De password, der er i ValidateUser klassen, skal være hash'et. Tjek et password ved at hash'e passwordet og undersøg om det er en samme hashkode som i ValaidateUser

Benyt: `Crypto.HashPassword(password)`