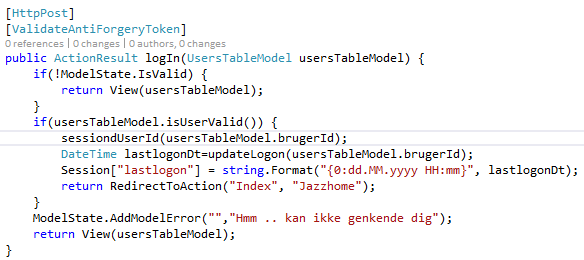
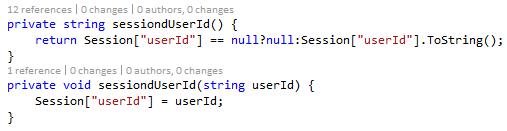
### Adgangskontrol

I første omgang blev det medfødte framework, til sikring mod uautoriseret adgang, flyttet over på MySQL. Det var i sig selv en omfattende øvelse. Herefter skulle nogle af de anvendte komponenter customiseres for at kunne blive anvendt på den måde, der var lagt op til i teoridelen. Det viste sig at være en større opgave at tilegne sig viden om, hvordan det skulle gøres. Derfor blev så lavet en mindre elegang løsning, som var hurtigere at få sat i værk.

I logIn-metoden testes det indtastede brugerId mod databasen. Hvis brugeren er kendt i forvejen, registreres brugerId i en sessionsvariabel med navnet userId, og web-siden /Jazzhome/Index returneres, ellers forbliver brugeren på logIn-siden; denne gang med en fejlmeddelslse.





Efterfølgende anvendes følgende kode ind i samtlige metoder, der skal beskyttes mod uautoriseret adgang.



Hvis der er en tekst i sessionsvariablen userId, er brugeren autoriseret.

I Web.config-filen kan man sætte en timeout på sessionen. Her er den sat til ti minutter. Hvis man ikke vælger en værdi selv, anvendes tyve minutter.



MVC-frameworket anvender filtre for at tilføre funktionalitet, der skal anvendes på tværs af flere metoder og/eller klasser. Som et konkret eksempel herpå, er [Authorize] og [AllowAnonymous]

Disse to annoteringer/filtre/Attributter bruges i forbindelse med den medfødte adgangskontrol til at fortælle, at en klasse eller metode skal beskyttes hhv. kan tilgås uden yderligere autorisation.

I en hjemmegjort en af slagsen, public class AdgangskontrolAttribute : ActionFilterAttribute {, kan man se, hvad der er af tilgængelig information i bl.a. den medfødte authentication. Her kan ses, at der er logget ind med ”a@b.c” og at brugeren er authenticated.

Det bliver så slutmålet at anvende de tilgængelige variable, men først flyttes den fungerende hjemmelavede løsning til AdgangskontrolAttribute. Og relevante klasser annoteres med [AdgangskontrolAttribute]. Det er en mere elegant løsning end at anvende metoden sessiondUserId() hver gang.

