



Kontrollera tillgång

Översikt

- Implementera autentisering och auktorisering.
- Tilldela roller och medlemskap.

.eerec

Lektion 1: Implementera autentisering och auktorisering

- Introduktion.
- Lokal autentisering.
- Claim-Based Authentication.
- Federated Authentication.
- Förhindra tillgång.

.eerec

Introduktion

- Autentisering är ett viktigt krav, för de flesta webbapplikationer.
- Utvecklare väljer att bara visa en del av applikationen, för alla användare.
- Webbapplikationen kräver att användare autentiserar sig för att komma åt andra delar av applikationen.
- ASP.NET inkluderar ett antal olika autentiseringssystem för denna funktion.

.eetec

Lokal autentisering

- *Membership providers*, hjälper dig med att skapa en säker autentisering.
- Du kan använda claim-based eller federated system, för att låta extern användare autentisera sig själv, för att sedan få tillgång till informationen.
- *Authentication providers*, inkluderar programkod som körs när ASP.NET behöver autentisera användare, mot information som finns i AD DS eller på SQL-server.

.eetec

Lokal autentisering (forts.)

Provider	Beskrivning
ActiveDirectoryMembershipProvider	Ger möjlighet att endast arbeta med AD DS som förråd för användare, för din webbapplikation.
SqlMembershipProvider	Arbetar tillsammans med aspnet_regdb.exe, för att arbeta med specifikt schema för tabell.
SimpleMembershipProvider	Nya generationen, fungerar med alla versioner av SQL Server, även i Azure. Kräver tre parametrar: namn på tabell, användare ID och användarenamn.
UniversalProvider	Funger med databaser som fungerar med Entity Framework, bara med schema från Microsoft.

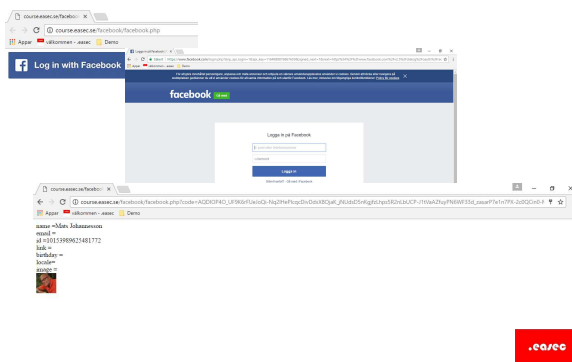
.eetec

Claim-Based Authentication

- Claim-Based Authentication är en modell som tillhandahåller funktioner för:
 - Single sign-on.
 - Hjälper till att autentisera användare.
 - Hjälper till att lagra information om användare.
- Integrerar applikation med system för identiteter, från andra plattformar eller organisationer.

.eefcc

Claim-Based Authentication (forts.)



.eefcc

Federated Authentication

- Federation litar på claim-based authentication, för att tillåta externa partners, såsom företag man litar på, att få tillgång till applikationen.
- Windows Identity Foundation (WIF) tillhandahåller stöd genom WSFederationAuthenticationModule.
- Autentisering låter Security Token Service att bearbeta claims från partners, kommer att extrahera ut användareinformation från dessa.
- FederatedPassivSignin används för att omdirigera användare till inloggnings sida.

.eefcc

Förhindra tillgång

- Du kan förhindra tillgång, genom att implementera attributet [Authorize()].
- Web.config filen kan inte användas, för MVC-baserade applikationer skickar begäran till händelser, inte till sidor.
- Om du placerar attributet på klassnivå, måste användare ha loggat på, innan de kan använda controllerklass.
- [AllowAnonymous()] kommer att tillåta anonym användare att få tillgång till specificerad del i din programkod.

.eeec

Övning Kontroller tillgång till händelser i controller



.eeec

Lektion 2: Tilldela roller och medlemskap

- Introduktion.
- Role providers.
- Lägg till användare i roll.
- Egen skräddarsydd Role Provider.
- Referera till tjänst.
- Lägg till logik för Membership Provider.

.eeec

Introduktion

- Roller och medlemskap är ett komplement till autentisering.
- Du måste först känna till vilka roller som behövs för din applikation, innan du kan implementera funktionen.
- Även skräddarsydda providers kan skapas, stöd för FoxPro eller Oracle databaser.

.eeec

Role providers

- Role Providers fungerar på liknande sätt, som för användare.
- Två typer:
 - ActiveDirectoryRoleProvider, använder AD DS.
 - SQLRoleProvider, fungerar bara med schema genererade från Microsoft.
- Dessa två typer har restriktioner, exempelvis att de inte kan arbeta med Non-SQL eller med schema som inte är genererade av Microsoft.
- SimpleRoleProvider och UniversalProvider är exempel på nya generationen av providers, dessa har inte dessa restriktioner.

.eeec

Lägga till användare i roll

- När du använder SimpleRoleProvider eller UniversalRoleProvider, kan du ladda in grundinformation, genom att direkt editera tabell.
- Du kan även editera tabell genom att använda funktionen AddUserToRoles.
- Exempelvis:

```
AddUserToRoles("Mats", new string[] {"Admin", "Staff"})
```

.eeec

Egen skräddarsydd Role Provider

- ASP.NET tillåter skräddarsydda role providers.
- Måste skapa klass, som ärver klassen RoleProvider.
- I denna klass skriver du din programkod.
- När detta är klar, måste du lägga till referens till din egna provider i web.config.

.eeec

Referera till tjänst

- Referens till SimpleMembershipProvider måste läggas till i web.config.
- Nästa steg är att skapa tabell för användare, du kan modifiera AccountModel.cs för att generera tabell.
- Internet Application Template.

.eeec

Lägga till logik för Membership Provider

- Om du behöver lägga till funktionalitet till MembershipProvider, exempelvis egen logik för autentisering, måste du utveckla Custom Membership provider.
- Därefter måste göra override på ValidateUser.
- När du har lagt till logik i skräddarsydd membership provider, måste web.config-filen modifieras.

.eeec

Repetitionsfrågor

.course
