

## WEB SYSTEMS 202

Marinda Taljaard

Office 09 02 29

marinda.taljaard@mandela.ac.za

Using Roles in ASP.NET MVC Authentication

### ADMIN

- Test marks remains
  - Scripts now available
- Module evaluation – 3 questions – please complete in Moodle

## TEST FEEDBACK

- Extra Content Placeholders / Incorrect placement
- Root level web.config
  - Duplicate connection string entries
  - Relative path to database not used
- Validation
  - Not specifying which control to validate
  - Display Dynamic
  - Regular expression validator
    - Remember to specify expression
    - Still needed Required field validator too
- Authorization:
  - AddCustomer and Customers saved in separate folder
  - Second web.config to specify allow and deny properties



## AUTHORIZATION AND AUTHENTICATION

- Complete access to anonymous users
  - Default situation
- Restricted access to some pages
  - Authentication required on some pages
- Restricted access to all pages
  - All users must be authenticated
- Restricted access to all pages, but different levels of access available
  - Any authenticated users can access “general” pages
  - Only admin users can access “admin” pages



## ONLY TWO USER LEVELS?

- Can add a Boolean field to the user table
  - Incorporate into query to determine access



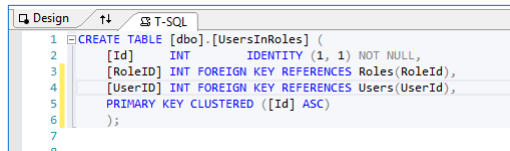
## MAKING PROVISION FOR USER ROLES

- Update model and database
  - Table for role information (... , RoleID, RoleName)
  - Table for role allocation (... , RoleID, UserID)
- Even though this module does not focus on databases:
  - Foreign keys required for tables – creating the relationship/association between tables



## RELATIONSHIPS/ASSOCIATIONS

- Update in Database



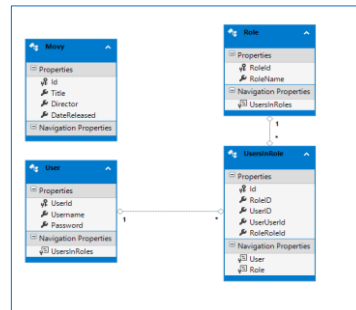
```

1 CREATE TABLE [dbo].[UsersInRoles] (
2     [Id] INT IDENTITY (1, 1) NOT NULL,
3     [RoleId] INT FOREIGN KEY REFERENCES Roles(RoleId),
4     [UserId] INT FOREIGN KEY REFERENCES Users(UserId),
5     PRIMARY KEY CLUSTERED ([Id] ASC)
6 );
7
8

```

- Using Entity Framework

- Right-click in open space, add association

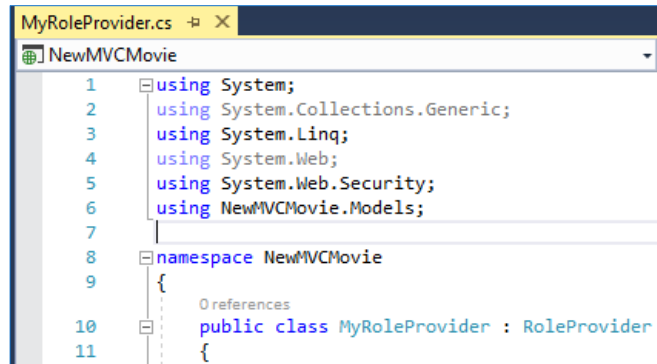


## CUSTOM ROLE PROVIDER

- Role Provider uses a separate database for storing user roles
- Create and configure a custom role provider to use application existing database
- Right click in Solution Explorer
  - Add New Item, Visual C#, Class
  - Enter name: MyRoleProvider.cs
- Or create a new controller, and add the required methods and views*

## CUSTOM ROLE PROVIDER

- Import namespace:
  - Using System.Web.Security;
  - Using <your project namespace>.Models;



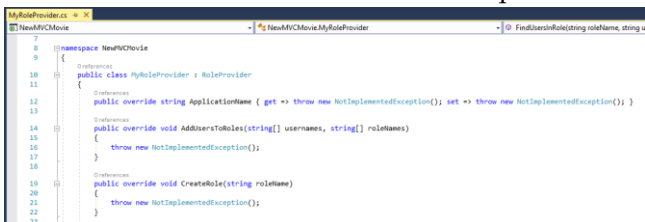
```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Security;
6  using NewMVCMovie.Models;
7
8  namespace NewMVCMovie
9  {
10     public class MyRoleProvider : RoleProvider
11     {

```

## CUSTOM ROLE PROVIDER

- Click on RoleProvider – look for quick actions next to the line numbers
  - Select “Implement abstract class” – serves as base class to be used for inheritance
- For minimum functionality implement:
  - GetRolesForUser – get a list of all user roles (might be a list)
    - Get the Role(s) for a specific user (multi-table query)
  - IsUserInRole – check if a user has a specific role



```


7  namespace NewMVCMovie
8  {
9      public class MyRoleProvider : RoleProvider
10     {
11         public override string ApplicationName { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }
12
13         public override void AddUsersToRoles(string[] usernames, string[] roleNames)
14         {
15             throw new NotImplementedException();
16         }
17
18         public override void CreateRole(string roleName)
19         {
20             throw new NotImplementedException();
21         }
22
23     }

```

## CUSTOM ROLE PROVIDER

- Other available methods:
    - AddUsersToRoles
    - CreateRole
    - DeleteRole
    - FindUsersInRole
    - GetAllRoles
    - GetUsersInRole
    - RemoveUsersFromRoles
    - RoleExist
- 

## USING A ROLES CONTROLLER

- Add methods and views – at least this functionality:
  - Index view – listing all roles
  - Create view [HttpPost] – adding a new role
  - Edit view [HttpPost] – updating role information
  - Delete view [HttpPost] – removing a role
  - ManageUsers view – allocating a role to a user
- 

## APPLYING AUTHORIZATION

- Only Admin users allowed:
  - [Authorize(Roles = "Admin")]
  - *Use the RoleName as you have in the table*
- Any logged in users allowed:
  - [Authorize]
- Anonymous users allowed:
  - [AllowAnonymous]



## WEB.CONFIG


- Specify roleManager Provider

```
<authentication mode="Forms">
  <forms loginUrl="/Security/Login" defaultUrl="/Home/Index"></forms>
</authentication>

<roleManager enabled="true" defaultProvider="MyRoleProvider">
  <providers>
    <clear/>
    <add name="MyRoleProvider" type="NewMVCMovie.MyRoleProvider"></add>
  </providers>
</roleManager>
```



## PORTFOLIO

- Portfolio
    - Due Tuesday 16 October, 12:00 midday
    - Submit on server (not Moodle)
      - MVC projects are big – do not start the copy process 11:55....
    - Submit details to follow later
  - Choose one of the three given topics
  - Specific requirements in Portfolio document
  - Marking Rubric also in Portfolio document
  - “No Plagiarism” declaration document will be made available on Moodle – to be signed and submitted in printed form
- 

## RESOURCES

- Entity Framework
    - <https://docs.microsoft.com/en-us/ef/ef6/fundamentals/relationships>
  - Adding Admin roles
    - [https://www.youtube.com/watch?v=ufbnsO\\_DBIY](https://www.youtube.com/watch?v=ufbnsO_DBIY)
    - [https://www.youtube.com/watch?v=WAYI6ar\\_U6U](https://www.youtube.com/watch?v=WAYI6ar_U6U)
    - <https://www.youtube.com/watch?v=798wMMCbaTY&t=2s>
  - Custom Role Provider
    - <http://kitsula.com/Article/Custom-Role-Provider-for-MVC>
    - <https://techbrij.com/role-based-menu-asp-net-mvc>
- 