**Role-Based Authentication in ASP.NET MVC**

##### **What are the Roles?**

Roles are nothing but the permissions given to a particular user to access some resources. So in some other words, we can say that, once a user is authenticated then what are the resources the user can access are determined by his roles. A single user can have multiple roles and Roles plays an important part in providing security to the system. For example, Admin, Customer, Accountant, etc.

##### **SQL Script:**

In order to understand the Roles, let add some data into the tables. Please use the below SQL Script to insert some test data to Employee, Users, RoleMaster, and UserRolesMapping table.

-- Inserting data into Employee table

**INSERT** **INTO** Employee **VALUES**('Anurag', 'Software Engineer', 10000)

**INSERT** **INTO** Employee **VALUES**('Preety', 'Tester', 20000)

**INSERT** **INTO** Employee **VALUES**('Priyanka', 'Software Engineer', 20000)

**INSERT** **INTO** Employee **VALUES**('Ramesh', 'Team Lead', 10000)

**INSERT** **INTO** Employee **VALUES**('Santosh', 'Tester', 15000)

-- Inserting data into Users table

**INSERT** **INTO** Users **VALUES**('Admin','admin')

**INSERT** **INTO** Users **VALUES**('User','user')

**INSERT** **INTO** Users **VALUES**('Customer','customer')

-- Inserting data into Role Master table

**INSERT** **INTO** RoleMaster **VALUES**('Admin')

**INSERT** **INTO** RoleMaster **VALUES**('User')

**INSERT** **INTO** RoleMaster **VALUES**('Customer')

-- Inserting data into User Roll Mapping table

**INSERT** **INTO** UserRolesMapping **VALUES**(1, 1, 1)

**INSERT** **INTO** UserRolesMapping **VALUES**(2, 1, 2)

**INSERT** **INTO** UserRolesMapping **VALUES**(3, 1, 3)

**INSERT** **INTO** UserRolesMapping **VALUES**(4, 2, 2)

**INSERT** **INTO** UserRolesMapping **VALUES**(5, 3, 3)

As you can see, the user with id 1 having three roles whiles the user with id 2 and 3 having only one role.

##### **Creating the Role Provider:**

Create a class file with the name **UsersRoleProvider** within the Models folder and then copy and paste the following code. This class implements the **RoleProvider** class. If you go to the definition of **RoleProvider** class then you can see it is an **abstract class**. As it is an abstract class we need to implement all the methods of that class. The **RoleProvider** class belongs to **System.Web.Security** namespace.

**using** *System;*

**using** *System.Collections.Generic;*

**using** *System.Linq;*

**using** *System.Web;*

**using** *System.Web.Security;*

**namespace** *SecurityDemoMVC.Models*

**{**

**public** **class** UsersRoleProvider : RoleProvider

**{**

**public** **override** string ApplicationName

**{**

**get**

**{**

**throw** new NotImplementedException**()**;

**}**

**set**

**{**

**throw** new NotImplementedException**()**;

**}**

**}**

**public** **override** **void** AddUsersToRoles**(**string**[]** usernames, string**[]** roleNames**)**

**{**

**throw** new NotImplementedException**()**;

**}**

**public** **override** **void** CreateRole**(**string roleName**)**

**{**

**throw** new NotImplementedException**()**;

**}**

**public** **override** **bool** DeleteRole**(**string roleName, **bool** throwOnPopulatedRole**)**

**{**

**throw** new NotImplementedException**()**;

**}**

**public** **override** string**[]** FindUsersInRole**(**string roleName, string usernameToMatch**)**

**{**

**throw** new NotImplementedException**()**;

**}**

**public** **override** string**[]** GetAllRoles**()**

**{**

**throw** new NotImplementedException**()**;

**}**

**public** **override** string**[]** GetRolesForUser**(**string username**)**

**{**

**using** **(**EmployeeDBContext context = new EmployeeDBContext**())**

**{**

var userRoles = **(from** user in context.Users

**join** roleMapping in context.UserRolesMappings

**on** user.ID **equals** roleMapping.UserID

**join** role in context.RoleMasters

**on** roleMapping.RoleID **equals** role.ID

**where** user.UserName == username

**select** role.RollName**)**.ToArray**()**;

**return** userRoles;

**}**

**}**

**public** **override** string**[]** GetUsersInRole**(**string roleName**)**

**{**

**throw** new NotImplementedException**()**;

**}**

**public** **override** **bool** IsUserInRole**(**string username, string roleName**)**

**{**

**throw** new NotImplementedException**()**;

**}**

**public** **override** **void** RemoveUsersFromRoles**(**string**[]** usernames, string**[]** roleNames**)**

**{**

**throw** new NotImplementedException**()**;

**}**

**public** **override** **bool** RoleExists**(**string roleName**)**

**{**

**throw** new NotImplementedException**()**;

**}**

**}**

**}**

In the above class, we only modify the implementation of the **GetRolesForUser** method. This method takes the Username as an input parameter and based on the username we need to fetch the User Roles as an array and return that array.

##### **Configuring Role Provider in the web.config file:**

Add the following code within the system.web section of your web.config file.

**<roleManager** defaultProvider="usersRoleProvider" enabled="true" **>**

**<providers>**

**<clear/>**

**<add** name="usersRoleProvider" type="SecurityDemoMVC.Models.UsersRoleProvider"**/>**

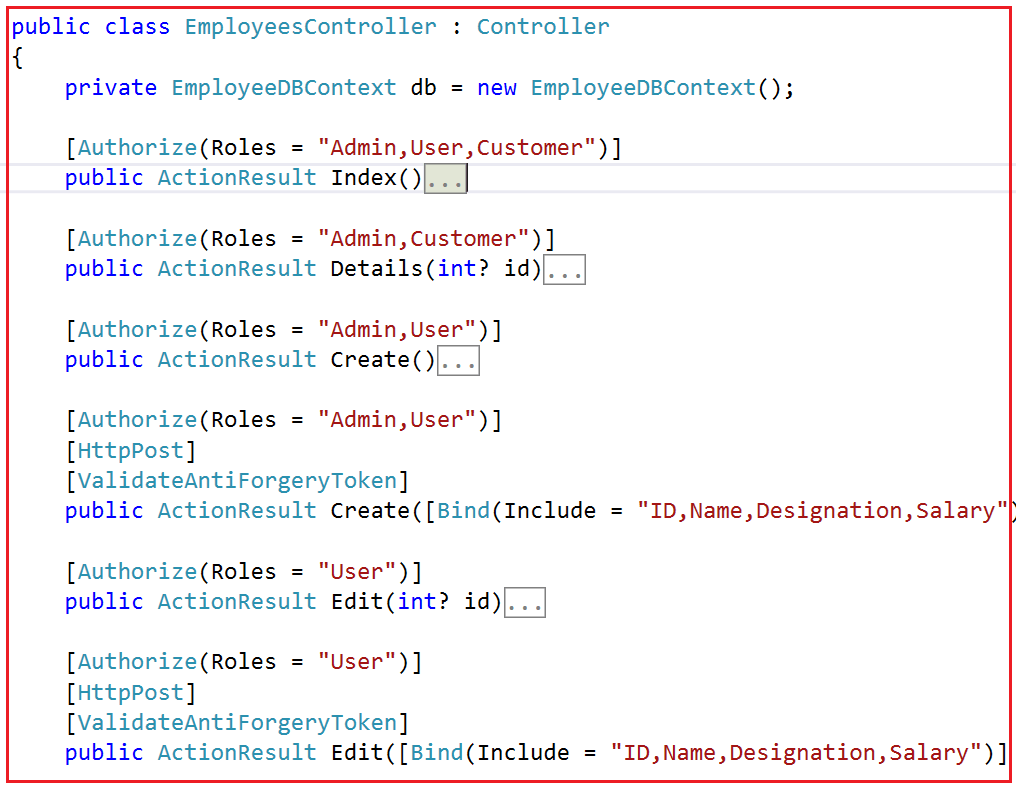
**</providers>**

**</roleManager>**

Basically here we are adding our Role Providers. Before adding the Role Providers first we clear all roles. The name you can give anything but the type value is going to be the full name of your Role Provider i.e. including the namespace. Here you can add any number of Role Providers. You have to provide the default provider which is going to be used as default in the default provider parameter of role manager and you need to enable it by setting the value to true of enabled property.

##### **Modifying the Employees Controller:**

Please modify the Authorize attribute to include Roles as shown below.



First, we remove the Authorize attribute from the Controller Level and applied it at the action method level. Here you can pass multiple roles separated by a comma. As per your business requirement set the Roles and test by yourself.