# ASP. NET MVC Roles Demo

Amit Ahlawat

CSE 686 – Internet Programming

Spring 2014

### Goals of document

- 1) Show how to make a group of pages available only after registration with built-in authentication and registration mechanisms
- 2) Show to make specific pages available only to select users
- 3) Show to make specific pages available only to select roles

# Using Built-in Registration and Authentication Mechanisms

Create an ASP .NET MVC application.

sDemo - Microsoft Visual Studio				Quick Launch (Ctrl+Q)
IT VIEW PROJECT BUILD DEBUG	TEAM TOOLS TEST ARCHITECTUR	େ ANALYZE WINDOW HELP ୦୦×ପୁଂA <sub>ਦ</sub>		S
ASP.NET application 🗢 ×	E 686\ Projectr\ Polec\ Demo\ PolecDemo\ PolecD	amo\Project Readme.html		Solution Explorer
ertoses (in ordang (searco) (in ordanic (co		enourojeconcouncinani		Search Selution Suplayer (Chilue)
Your ASP.N	NET applicat	ion		Image: Solution 'RolesDemo' (L project)         Image: Solution 'RolesDemo' (L project)         Image: Solution 'RolesDemo' (L project)         Image: RolesDemo 'RolesDemo' (L project)         Image: RolesDemo 'RolesDemo' (R Project)         Image: RolesDemo 'RolesDemo' (R RolesDemo)         Image: RolesDemo 'RolesDemo' (R RolesDemo)         Image: RolesDemo 'RolesDemo' (R RolesDemo)         Image: RolesDemo 'RolesDemo)         Image: RolesDemo)         Image: RolesDemo)
Congratulations! You've c	reated a project			<ul> <li>▶ iii fonts</li> <li>▶ iii Models</li> </ul>
This application	CUSTOMIZE APP	DEPLOY	GET HELP	Gripts     Gripts     Gripts     Gripts     Gripts     Gripts     Gripts
consists of:	Get started with ASP.NET MVC Change the site's theme	Ensure your app is ready for production Windows Azure Hosting providers	Get help Get more templates	<ul> <li>A Home</li> <li>B Shared</li> <li>Test</li> </ul>
Sample pages showing basic nav between Home, About, and Contact	Add more libraries using NuGet Configure authentication Customize information about the website users			00     ViewStart.cshml       vi2     Veisconifig       Di fraviconico     b       Olobalasas     ViewStart.cshml       ViewStart.cshml     viewscanfig
Theming using Bootstrap	Get information from social			
how to register and sign in	Add HTTP services using ASP.NET			
ASP.NET features managed using NuGet	Secure your web API			
	Add real-time web with ASP.NET SignalR			
	Add components using Scaffolding			✓ Solution Explorer Team Explorer Class View

Add a new controller to the application.

Your ASP.NET application 🗢 🗙		Solution Explorer	• 4 ×
URL: C:\Users\TA Grading\Desktop\TAs\Amit\CSE 6 Add Scaffold	Cong Paget, Number 1971		d 🖻 🕨 - 22
▲ Installed			(+;) (;+)
Common	MVC 5 Controller - Empty	MVC 5 Controller - Empty by Microsoft	l project)
Your ASP.N	MVC 5 Controller with read/write actions	v5.0.0.0 An empty MVC controller.	
	MVC 5 Controller with views, using Entity Framework	Id: MvcControllerEmptyScaffolder	
	Web API 2 Controller – Empty		troller.cs
Congratulations! You've cre	Web API 2 Controller with actions, using Entity Framework		mencs
This application	Web API 2 Controller with read/write actions		
consists of:	Web API 2 OData Controller with actions, using Entity Framework		
			html
Sample pages showing basic nav between Home, About, and			
Contact			
Theming using Bootstrap			.html
Authentication, if selected, shows how to register and sign in			
ASP.NET features managed using			
NuGet		Add Cancel	
			,
Add components using Scaffolding		- Solution Explorer Team Ex	plorer Class View

Name the controller 'TestController'.

ASP.NET application				Solution Explorer	- <b>-</b>
C:\Users\TA Grading\Desktop\TAs\Amit\C	E 686\Projects\Roles\Demo\RolesDemo\RolesDemo\Project	_Readme.html	•	C C A To - ≓ R I T B ≠ - S	
Your ASP.N	NET application		Ĺ	Search Solution Explorer (Ctrl+)	ع
Congratulations! You've of This application consists of:	CUSTOMIZE APP Get started with ASP.NET N Change the site's theme		Add Cancel	b         If conts           b         Models           b         Scripts           a         Views           b         Account           b         Home           b         Shared           i         Text	
Sample pages showing basic nav between Home, About, and Contact Theming using Bootstrap	Add more libraries using N Configure authentication Hosting providers Customize information about the website users Get information from social	g providers		00 Uvevdart.citrmi ↓ Web.config b favicon.ico ↓ Gifobahaax ↓ packages.config ↓ Project.Readme.htmi ↓ C* Startup.c.t	
Authentication, if selected, shows how to register and sign in	providers Add HTTP services using ASP.NET			Þ ý⊡ Web.config	
ASP.NET features managed using NuGet	Web API Secure your web API Add real-time web with ASP NET				
	SignalR Add components using Scaffolding		-	Solution Explorer Team Explorer Class View	

Add a view for the Index method of the TestController.

TestController.cs 🗢 × Your ASP.NET application		-	Solution Explorer
t RolesDemo.Controllers.TestController	~ 🖗 Index()	*	○ ○ ☆ 'o - ≠ Q 司 ŋ ○ ≠ -
⊡using System;		+	Search Solution Explorer (Ctrl+:)
using System.Collections.Generic;		<u>^</u>	
using System Leh:			I Solution 'KolesDemo' (1 project)
using System.Web.Mvc:			A B KolesDemo
			Properties
namespace RolesDemo.Controllers	Add View		App Data
{			App_Data
public class TestController : Controller	View manage		b Content
{			A G Controllers
÷ //	index		C* AccountController.cs
// GET: /Test/	Template:		C* HomeController.cs
nublic ActionResult Index()	Empty (without model)		C# TestController.cs
{	Model class		fonts
return View();			Models
}			Scripts
	Data context class:		Views
1	· · · · · · · · · · · · · · · · · · ·		Account
	View options:		▶ Home
	Create as a partial view		P Shared
	Reference script libraries		I est
			Web config
			Favicen ice
			Clobal asax
	(Leave empty if it is set in a Razor viewstart file)		packages.config
			Project_Readme.html
			C= Startup.cs
	Add Cancel		▶ ♥ Web.config
		-	
100.97 -			Solution Evolorer Team Evolorer Clarr View



Here the view is shown. An additional heading has been added (This is a test page).

Run the application and navigate to /Test/Index as shown:



To add the requirement of user authentication to view a page, we will use the [Authorize] attribute as shown :



The above should not allow any user to view page /Test/Index unless logged in.

Run the application and navigate to /Test/Index.



The page is redirected to the login page.

Log in.	account to log in.	Use another service to log in.
User name Password		There are no external authentication service: configured. See this article for details on setting up this ASP.NET application to suppo logging in via external services.
Register if you	Log in don't have a local account.	
© 2014 - My AS	SP.NET Application	

Create a new account called `testUser'.

User name	testUser	
Password		
Confirm password		
	Register	
© 2014 - My ASP.NET A	Application	

#### After registration, navigate to /Test/Index



Now the user is logged in, /Test/Index can be viewed.



Add another action to the TestController called 'Foo'.



#### Add a view to the action 'Foo'.

Index.cshtml TestController.cs + × Your ASP.NET applie	ation	Solution Explorer
to RolesDemo.Controllers.TestController	- 🌚 Foo()	· · · · · · · · · · · · · · · · · · ·
using System.collections.Generic; using System.Linq; using System.Heb; using System.Heb;		Search Solution Explorer (Ctrl+;)     Golution 'RolesDemo' (1 project)     ●
<pre>□namespace RolesDemo.Controllers {</pre>	Add View	▶ ■■ References App_Data
Oreferences public class TestController : Controller {	View name: Foo	App_Start     Gontent     Controllers
<pre>// // GET: /Test/ [Authorize]</pre>	Template: Empty (without model)	C* AccountController.cs C* HomeController.cs
<pre>Oreferences Display Content of the second seco</pre>	Model class:	C* TestController.cs      Finite Controller.cs      Models
<pre>return View(); }</pre>	Data context class:	<ul> <li>Scripts</li> <li>Views</li> <li>Account</li> </ul>
Oreferences public ActionResult Foo() {	View options: Create as a partial view	Home     Shared
return View(); } }	Reference script libraries     Vie a layout page:	(@) Index.cshtml (@) _ViewStart.cshtml
L	(Leave empty if it is set in a Razor_viewstart file) Add Cancel	<ul> <li>Y⊒ Web.config</li> <li>Braviconico</li> <li>Al Global.sax</li> <li>Packages.config</li> <li>Project.Readme.html</li> <li>cr Startup.cs</li> <li>Y⊇ Web.config</li> </ul>
		·

An additional heading has been added to the new page (Foo : test page).



The [Authorize] attribute can be applied to the Controller as well. That will affect every action in that Controller.

Remove the [Authorize] attribute from the Index action and add it to the TestController as follows:



Run the application, and navigate to /Test/Index :



Navigation to /Test/Index redirects the page to login page.

Log in - My ASP.NET App	
← → C 🗋 localhost:1305/Account/Login?ReturnUrl=%2FTest%2FIndex	☆ =
Application name Home About Contact	Register Log in
Log in.	
Use a local account to log in.	Use another service to log in.
User name Password	There are no external authentication services configured. See this article for details on setting up this ASP.NET application to support logging in via external services.
Remember me?     Log in	
Register if you don't have a local account.	
© 2014 - My ASP.NET Application	

Now try navigating to /Test/Foo. Since the attribute was applied to the controller, this navigation should also redirect to login page if user is not logged in.



Navigation to /Test/Foo redirects to login page.

Application	name Home About Contact	Register Log in
Log in.		
Use a local	account to log in.	Use another service to log in.
User name		There are no external authentication services configured. See this article for details on setting up this ASP.NET application to support
Password		logging in via external services.
	Remember me?	
	Log in	
Register if you	don't have a local account.	

There are situations when you want to exclude some of actions, but still apply the [Authorize] attribute to the controller. For this, the [AllowAnonymous] attribute is used.

Apply the [AllowAnonymous] attribute to the Foo action as follows:



From the above, navigation to action methods other than 'Foo' should require authentication.

Run the application and navigate to /Test/Foo:



The page is accessible even though user is logged out, because of the [AllowAnonymous] attribute.



### How to make specific pages available only to select users

For this demo, we need to create two test users. Run the application and click Register.



Register a user named 'admin'.

Register - My ASP.NET Ap × → C □ localhost:1305//	Account/Register	<u>.</u>
Application name	Home About Contact	Register Log in
Register. Create a new accou	int.	
User name	admin	
Password		
Confirm password		
	Register	
© 2014 - My ASP.NET A	Register	

After registering 'admin' user, register another user named 'employee'.

Register - My ASP.NET Ap ×		-	n Autor	Authors	AaB	-	-		
← → C □ localhost:1305/A	ccount/Register								☆ =
Application name	Home About	Contact					Register	Log in	
Register. Create a new accou	int.								
User name	employee								
Password									
Confirm password									
	Register								
© 2014 - My ASP.NET Ap	plication								

To specify which users are allowed to view a page after authentication, we use the [Authorize] attribute, and specify a comma separated list of users.

Remove the [Authorize] attribute from the TestController.

Add the [Authorize (Users="admin")] attribute to Foo action method to specify only `admin' user can view that page after logging in.



Run the application and navigate to /Test/Foo.



Because of the [Authorize] attribute, the page is redirected to login page.

Login as the 'admin' user.

Log in - My ASP.NET App		
← → C 🗋 localhos	st:1305/Account/Login?ReturnUrl=%2FTest%2FFoo	☆ =
Application r	name Home About Contact	Register Log in
Log in.		
Use a local	account to log in.	Use another service to log in.
User name Password	admin	There are no external authentication services configured. See this article for details on setting up this ASP.NET application to support logging in via external services.
	Remember me?	
	Log in	
Register if you	don't have a local account.	
© 2014 - My AS	SP.NET Application	

After logging in, the 'admin' user can view the page.

Foo - My ASP.NET Applica ×	The second	Reality of State	 	march. Nong		
← → C 🕒 localhost:1305/Te	est/Foo					☆ =
Application name	Home About	Contact		Hello admin!	Log off	
Foo Foo test page © 2014 - My ASP.NET App	plication					

Log out the 'admin' user. Navigate to /Test/Foo as follows:



The page will be redirected. Now try logging in as 'employee'. Since 'employee' is not in the list of allowed users, even after logging in, 'employee' must not be able to navigate to /Test/Foo.

Application	name Home About Contact	Register Log in
Log in.		
Use a local	account to log in.	Use another service to log in.
User name Password	employee	There are no external authentication services configured. See this article for details on setting up this ASP.NET application to suppor logging in via external services.
	Remember me?	
	Log in	
Register if you	don't have a local account.	

Now try navigating to /Test/Foo. You will observe that the page is navigated to the login page again, even though the 'employee' user is logged in. It won't allow the navigation to /Test/Foo unless a user in the [Authorize] attribute list is logged in.

Log in - My ASP.NET Appi x	
← → C Discalhost:1305/Account/Login?ReturnUrl=%2FTest%2FFoo	☆ =
Application name Home About Contact	Hello employee! Log off
Log in.	
Use a local account to log in.	Use another service to log in.
User name	There are no external authentication services configured. See this article for details on setting up this ASP.NET application to support
Password  Remember me?	logging in via external services.
Log in	
Register if you don't have a local account.	
© 2014 - My ASP.NET Application	

Log in as the 'admin' user again.

Application name Home About Cont	=%2FTest%2FFoo 😪
Log in.	
Use a local account to log in.	Use another service to log in.
User name admin Password	There are no external authentication services configured. See this article for details on setting up this ASP.NET application to support logging in via external services.
Cog in	
Register if you don't have a local account.	

It can be observed that the navigation was allowed, once a user in the list logged in.

Foo	My ASP.NET Applice ×	- 10. (80.)	-	-	-	-	-	Traperty of	mart. No.		
← ⇒	C localhost:1305/Test/Foo										☆ =
	Application name Home	About	Contact						Hello admir	n! Log off	
	Foo										
	Foo : test page										_
	© 2014 - My ASP.NET Application										

To allows both 'admin' and 'employee' users to view that page, edit the 'Users' list in the [Authorize] attribute of the Foo action method and add 'employee' to that list as follows:



Run the application and navigate to /Test/Foo.



It will be redirected to login page.

Login as the 'employee' user.

Application name Home About Contact	Register Log in
Log in. Use a local account to log in.	Use another service to log in.
User name employee Password ·······	There are no external authentication services configured. See this article for details on setting up this ASP.NET application to suppor logging in via external services.
Remember me?     Log in	
Register if you don't have a local account.	

Navigate to /Test/Foo, and the page will be displayed for 'employee' user.

1 Foo -	My ASP.NET Applic: ×		-	-	(and (a)	-	 -	Implement	Annes A.	-		
$\leftarrow \rightarrow$	C localhost:1305/	/Test/Foo										☆ =
	Application name	Home	About	Contact					Hello er	nployee!	Log off	
	Foo											
	Foo : test page											
	© 2014 - My ASP.NET A	Application										

How to make specific pages available only to select roles

In Visual Studio 2013, the ASP .NET Web configuration tool was removed which assisted in managing roles.

To complete this task, we have to manage roles by using an alternative method, by editing some tables using SQL Server Explorer.

The tables we are interested in are:

- 1) AspNetRoles stores role id and name
- 2) AspNetUserRoles stores mapping of user id to role id
- 3) AspNetUsers stores user id and user name

The first step is to create some users in the website. For the remainder of this task, it will be assumed that the following users exist:

- 1) admin
- 2) employee
- 3) employee2

The first step is to open Server Explorer and get to the window where we can run some queries in the tables.

Server Explorer appears as an option in the left column as shown:



Click and navigate to the Server Explorer and open the tables menu as shown:

Serv	Serv	/er E	plore	r -= X	
er Ex	Ę۵	×	+## #	ta 11 St 📴	
plo		ി	Data	Connections	
rer			🔁 D	efaultConnection (RolesDemo)	
5			4 📋	Tables	
olb			⊳	_MigrationHistory	
2 2			⊳	AspNetRoles	
Pa			⊳	AspNetUserClaims	
ge			⊳	AspNetUserLogins	
Insp			⊳	AspNetUserRoles	
Dect			⊳		
ğ			Þ 🧉	Views	
			Þ 🧉	Stored Procedures	
			Þ 📫	Functions	
			Þ 📫	Synonyms	
			Þ 🧉	Types	
			Þ 🧉	Assemblies	
	⊳	•	Serve	rs	
	⊳	S	Share	Point Connections	
	⊳		Wind	ows Azure	

Right click and select new query.



First let's examine the 3 tables. Use select statements as shown below.

Too	SQL	.Query3.sql	* + X	Foo.	cshtml	Index.cs
lbox	Þ		周		aspnet-Role	sDemo-201
Page Inspector		□ select   select select	<pre>* from * from * from</pre>	Asp Asp	NetRoles; NetUserRol NetUsers;	es;

In Server Explorer you can highlight and execute individual lines. The execute button is the green play button in the top left corner.



Select * from AspNetRo select * from AspNetUs select * from AspNetUs select * from AspNetUs 100 % ↓ ST-SQL ↑↓ ⊞ Results Id Name	les; erRoles; ers; Message		
<pre> select * from AspNetRoles select * from AspNetUserR select * from AspNetUsers 0% → □ T-SQL ↑↓ ■Results </pre>	; oles; ; Message		
Userid Roleid			
Id UserName 1 009e87dd-7c36-47e9-8eab-776f39e289d9 admin	PasswordHash AOoZ5Rd7qgoCOk7B617x/QPBU8ZkbXzkzSN1y7WQhazPm+E	SecurityStamp de333473-13f5-47d7-b9a3-2c80c74421bc	Discriminator ApplicationUser
2         4d33b74f-91e2-4778-9904-33ecfa754e3b         testUser           3         64682311-5437-445a-b696-2234b48fbc1c         employee           4         b20de7a-ef90-4b70-959d-87ab3c83622a         employee2	AEIzLuH3N8Yx0yjTEuch50C9T9/ZhE8P+4XVOAdr/GGSrLtP5h1T AKUvBBoklfTXb7PX4Gj8spPsT92F+05PwTkZfyrGMq9rO2GBPv7 AIJ06MJeIW10hjrHfOHuSfBjno5rlRdXBo+aleTdQ1nZ05FNVZTJH	16df5549-8b66-4d22-8464-1c0f90f978a7 3849eee9-157e-4a58-940a-4b23559c3619 0bc2aa58-a181-4f4d-a87e-d21af205c15a	ApplicationUser ApplicationUser ApplicationUser

As can be seen, the AspNetRoles, and AspNetUserRoles tables are empty, whereas the AspNetUsers contains details about the current users.

In this task, we will create two roles, namely admin and employee and assign them to users as follows:

- 1) admin user -> admin role
- 2) employee & employee2 user -> employee role

The next step is to create the two roles, admin and employee.

Use insert statements as shown below:



Execute the insert statements (you can highlight them both and click the execute button).



Check the contents of the AspNetRoles table.

<pre>select * from AspNetRoles;</pre>
<pre>select * from AspNetUserRoles;</pre>
<pre>select * from AspNetUsers;</pre>
<pre>insert into AspNetRoles(Id,Name) values(1,'admin');</pre>
<pre>insert into AspNetRoles(Id,Name) values(2,'employee');</pre>
100 % -
⊈ T-SQL ↑↓      ⊞ Results      B Message
Id Name
1 1 admin
2 2 employee

The next step is to identify the user ids we want to map.

Record the user ids of the desired users somewhere (in this demo, those are just pasted in the same window to keep track of them).



	<pre>select * from AspNetUsers;</pre>										
	<pre>insert into AspNetRoles(Id,Name) values(1,'admin');</pre>										
	<pre>insert into AspNetRoles(Id,Name) values(2,'employee');</pre>										
B	□admin - 009e87dd-7c36-47e9-8eab-776f39e289d9										
	employee - 64682311-5437-445a-b696-2234b48fbc1c										
	employee2 - b20de7fa-ef90-4b70	-959d-87ab	3c83622a								
100 %	6 -										
2	T-SQL ↑↓ ⊞Results 🗈 Mess	age									
	ld	UserName	PasswordHash	SecurityStamp	Discriminator						
1	009e87dd-7c36-47e9-8eab-776f39e289d9	admin	AOoZ5Rd7qgoCOk7B617x/QPBU8ZkbXztxzSN1y7WQhazPm+E	de333473-13f5-47d7-b9a3-2c80c74421bc	Application User						
2	4d33b74f-91e2-47f8-9904-33ecfa754e3b	testUser	AEIzLuH3N8Yx0yjTEuch50C9T9/ZhE8P+4XVOAdr/GGSrLtP5h1T	16df5549-8b66-4d22-8464-1c0f90f978a7	Application User						
3	64682311-5437-445a-b696-2234b48fbc1c	employee	AKUvBBoklfTXb7PX4Gj8spPsT92F+O5PwTkZfyrGMq9rO2GBPv7	3849eee9-157e-4a58-940a-4b23559c3619	Application User						
4	b20de7fa-ef90-4b70-959d-87ab3c83622a	employee2	AIJ06MJeIW10h jrHfOHuSfB jn o 5 rlRdXB o + a leTdQ1nZ05FNVZTJH	0bc2aa58-a181-4f4d-a87e-d21af205c15a	Application User						

The next step is to insert the mappings between user id and role id. From before, we inserted the role ids:

1) admin role -> id 1
2) employee role -> id 2
employee - 64682311-5437-445a-b696-2234b48fbc1c
employee2 - b20de7fa-ef90-4b70-959d-87ab3c83622a
insert into AspNetUserRoles(UserId,RoleId) values('009e87dd-7c36-47e9-8eab-776f39e289d9',1);
insert into AspNetUserRoles(UserId,RoleId) values('64682311-5437-445a-b696-2234b48fbc1c',2);
insert into AspNetUserRoles(UserId,RoleId) values('b20de7fa-ef90-4b70-959d-87ab3c83622a',2);

Execute the insert statements.



Check the contents of the AspNetUserRoles table.

	⊡select * from AspNetRoles;						
	<pre>select * from AspNetUserRoles;</pre>						
	<pre>select * from AspNetUsers;</pre>						
	<pre>insert into AspNetRoles(Id,Name) values(1,'admin');</pre>						
	<pre>insert into AspNetRoles(Id,Name) values(2,'employee');</pre>						
100 % -							
I T-SQL ↑↓      I Results      Message							
	Userld	RoleId					
1	009e87dd-7c36-47e9-8eab-776f39e289d9	1					
2	64682311-5437-445a-b696-2234b48fbc1c	4682311-5437-445a-b696-2234b48fbc1c 2					
3	b20de7fa-ef90-4b70-959d-87ab3c83622a 2						

Now our user to role mapping is complete.

The last step is to use these roles in Authorize attribute. Edit the Test Controller as follows (replacing a user list with role list):

```
TestController.cs + ×
SQLQuery3.sql *
                    Foo.cshtml
                                    Index.cshtml
🔩 RolesDemo.Controllers.TestController
   ⊡using System;
     using System.Collections.Generic;
     using System.Linq;
     using System.Web;
    using System.Web.Mvc;
   namespace RolesDemo.Controllers
     {
         // [Authorize]
         0 references
         public class TestController : Controller
   Ė
         {
   Ė
             11
             // GET: /Test/
             0 references
             public ActionResult Index()
   Ė
             {
                  return View();
              }
              [Authorize(Roles="employee")]
              // [AllowAnonymous]
             0 references
             public ActionResult Foo()
   Ė
             {
                  return View();
             }
         }
    }
```

This should only allow users in employee role (employee & employee2) to view the result of Foo Action method.

Run the application and login as employee user.

Log in - My ASP.NET App	pi ×					
← → C 🗋 localhost:1305/Account/Login						
Application	name Home About Contact					
Log in.						
Use a local account to log in.						
User name	employee					
Password	••••••					
	Remember me?					
	Log in					
Register if you don't have a local account.						
© 2014 - My ASP.NET Application						

## Navigate to Test/Foo as follows:

Foo - My ASP.NET Applic: ×	August of	Autoritation in the local division of the lo	Added	Authors (	Aat	-	 August 1	Audio an	
← → C 🖞 localhost:1305/Test/Foo	About	Contact					Hello em	ployee!	Log off
E e e									
Foo : test page									
© 2014 - My ASP.NET Application									

As expected this user can view this page. Repeat the test for employee2 user.



As expected, users in employee role can access the page.

Now let's test the admin user, which belongs to admin role.

Login as the admin user:

Log in - My ASP.NET Appl ← → C D localhos	× st:1305/Account/Login					
Application n	name Home About Contact					
Log in.						
Use a local account to log in.						
User name	User name admin					
Password						
	Remember me?					
	Log in					
Register if you don't have a local account.						
© 2014 - My ASF	P.NET Application					

Navigate to Test/Foo.

← → C Diocalhost:1305/Account/Login?ReturnUrl=%2FTest%2FFoo						
Application name Home About Contact	Hello admin! Log off					
Log in.						
Use a local account to log in.	Use another service to log in.					
User name	There are no external authentication services configured. See this article for details on					
Password	setting up this ASP.NET application to support logging in via external services.					
Remember me?      Log in						
Register if you don't have a local account.						

As can be seen, the admin user is not allowed to view the page. This is as we expected, given we only authorized the employee role to access the page.