Autenticação e autorização com ASP.Net MVC

Publicado em **26 de julho de 2008** por **Giovanni Bassi**

Like 0 Send 0 Tweetar 0

Dei uma boa olhada na nova funcionalidade de autenticação e autorização para o ASP.Net MVC, e posso dizer que ficou bem legal. Gostei do que vi, ficou simples e fácil de aplicar. Assim como todas essas novas funcionalidades, esta está vindo também através de um Interception Filter. Basicamente o que o filtro faz é verificar se você está autenticado, e opcionalmente, autorizado a acessar uma determinada ação ou controller.

O atributo é o AuthorizeAttribute, e você pode ver o que ele está fazendo no repositório do fonte no Codeplex. Note que, basicamente, ele está verificando se o usuário está autenticado, olhando para a user.Identity.IsAuthenticated:

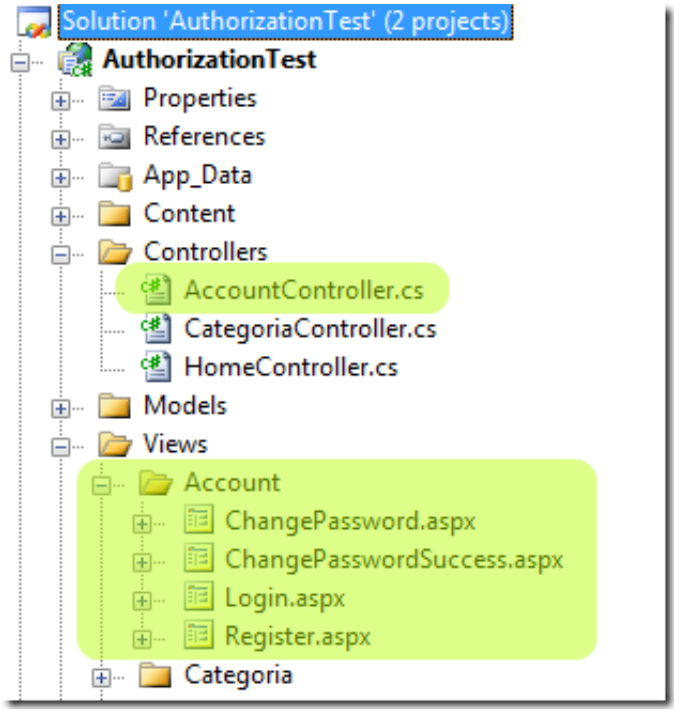
```
IPrincipal user = filterContext.HttpContext.User;
if (!user.Identity.IsAuthenticated) {
filterContext.Cancel = true;
filterContext.Result = new HttpUnauthorizedResult();
return;
```

Nada de mais, certo? Para autorizar é mais ou menos a mesma coisa, ele pega a lista de usuários ou papéis (roles) e verifica:

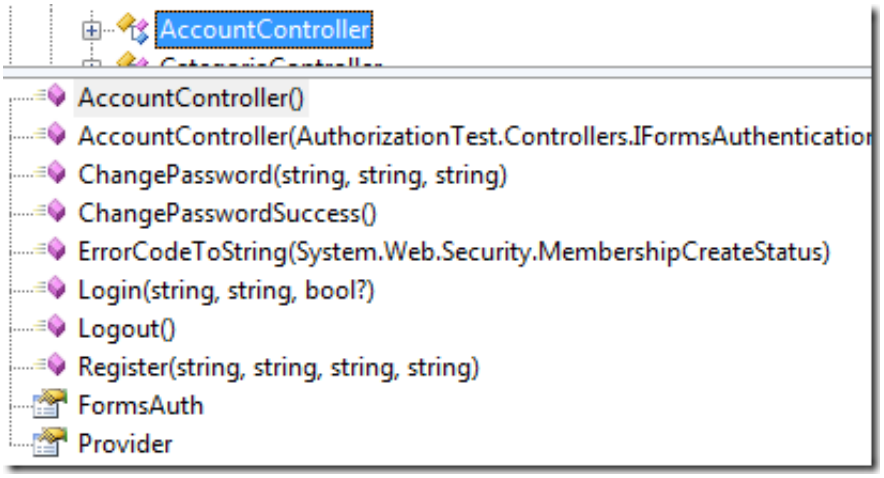
```
if (!String.IsNullOrEmpty(Users)) {
IEnumerable<string> validNames = SplitString(Users);
bool wasMatch = validNames.Any(name => String.Equals(name, user
if (!wasMatch) {
filterContext.Cancel = true;
filterContext.Result = new HttpUnauthorizedResult();
```

```
return;
}
}
if (!String.IsNullOrEmpty(Roles)) {
IEnumerable<string> validRoles = SplitString(Roles);
bool wasMatch = validRoles.Any(role => user.IsInRole(role));
if (!wasMatch) {
filterContext.Cancel = true;
filterContext.Result = new HttpUnauthorizedResult();
}
}
```

Além deste atributo, que faz o trabalho pesado de autenticação e autorização, há um controller para auxiliar no contato com o aplicativo em si. Ele se chama AccountController, e já vem no projeto. Há também algumas views que esse controle utiliza. Dêem uma olhada no Solution Explorer, com os itens novos destacados:



Já o controller em si traz os seguintes métodos:



Como podemos ver, ele tem ações para trocar senha, logar, deslogar, e criar um novo usuário (register). Ele não lista usuários, cria grupos e outras tarefas administrativas. Pelo que eu li nos blogs da Microsoft, eles esperam que tarefas administrativas sejam feitas de outra forma, e sugerem que no IIS 7 isso já é mais fácil. Isso pode querer dizer que não vai rolar um sistema de gestão de usuários no MVC. Para compensar isso há a boa notícia de que eles vão evoluir um pouco mais esse controlador e as views, permitindo trocar os mecanismos de autenticação e autorização, se não quisermos usar o membership do ASP.Net, que é onde o atributo [Authorize] e o controlador estão acoplados atualmente. Será possível estender este modelo de 2 formas então:

1. Trocando o membership do ASP.Net por algum outro framework;
2. Trocando o provider de membership do ASP.Net. Para permitir isso, o construtor do controlador já inicia com uma injeção de dependência bonita, dêem uma olhada:

```
16 public AccountController()
17     : this(null, null)
18 {
19 }
20
21 public AccountController(IFormsAuthentication formsAuth,
MembershipProvider provider)
22 {
23     FormsAuth = formsAuth ?? new
FormsAuthenticationWrapper();
24     Provider = provider ?? Membership.Provider;
25 }
```

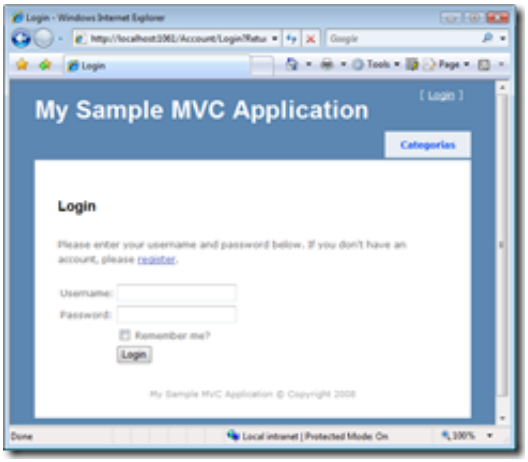
Essa interface, IFormsAuthentication, é nova, e é parte do MVC. O MembershipProvider é o velho conhecido MembershipProvider do ASP.Net. Dessa forma, fica fácil injetar um outro provider, para testar, e até mesmo trocá-lo. Para arrancar de vez o Membership do ASP.Net (primeira opção acima), a recomendação até agora é refazer o controlador. Mas isso vai melhorar.

Vimos que há view para trocar senha, para fazer login e se cadastrar. Como fica? Vou mostrar um screenshots abaixo, mas antes fiz um controlador simples, onde um usuário

normal só pode ver uma lista de categorias, e o usuário admin pode alterar. Vejam o controlador:

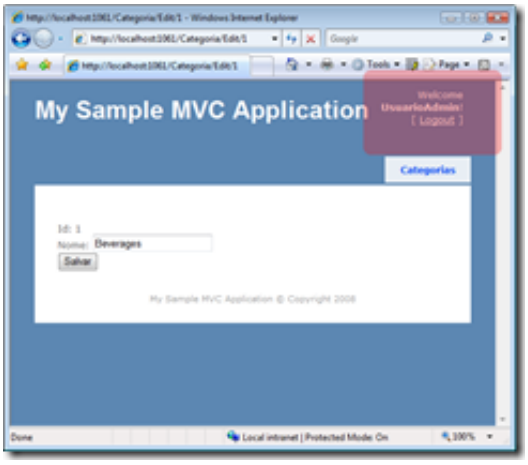
```
 1 public class CategoriaController : Controller
 2 {
 3     [Authorize(Roles="All,Admin")]
 4     public ActionResult Index()
 5     {
 6         return View("Index", (new
Northwind2005Model.NorthwindEntities()).
 7             Categories.ToList());
 8     }
 9
10     [Authorize(Roles = "Admin")]
11     public ActionResult Edit(int id)
12     {
13         var northwind = new
Northwind2005Model.NorthwindEntities();
14         var categoria = (from cat in northwind.Categories
15                          where cat.CategoryID == id
16                          select cat).First();
17         return View(categoria);
18     }
19
20     [Authorize(Roles = "Admin")]
21     public ActionResult Update(int id)
22     {
23         var northwind = new
Northwind2005Model.NorthwindEntities();
24         var categoria = (from cat in northwind.Categories
25                          where cat.CategoryID == id
26                          select cat).First();
27         BindingHelperExtensions.UpdateFrom(categoria,
Request.Form);
28         northwind.SaveChanges();
29
30         return Index();
31     }
32 }
```

Naturalmente a ação index, que é a padrão, lista os usuários, mas antes eu preciso estar autenticado. Então, quando eu clico na aba de categorias, sou direcionado à view de login pelo controlador:



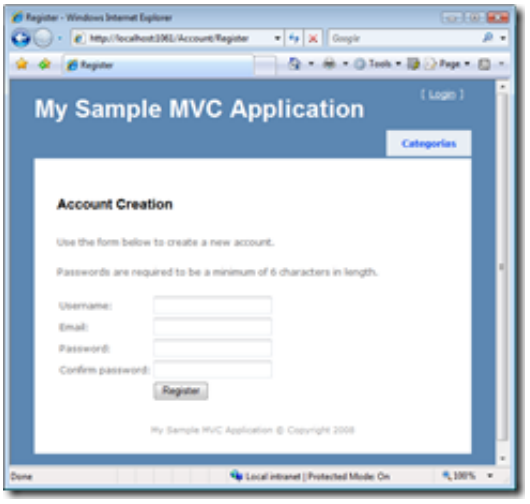
Após autenticar, por algum motivo, ele me redireciona de volta à Home (porque será?). Tenho que clicar novamente na aba de categorias que criei para finalmente ver as categorias, exibida pela ação index (notem o login aparecendo em destaque):

Mas eu loguei com um usuário comum. Ao tentar editar, sou direcionado ao login de novo, como se fosse para logar como admin. Logo como admin, ele volta à Home de novo (!?), navego até as categorias e tento editar. Agora já consigo editar e salvar. Essa é ação Edit, note mais uma vez o login no canto superior direito. Agora é o usuário admin:

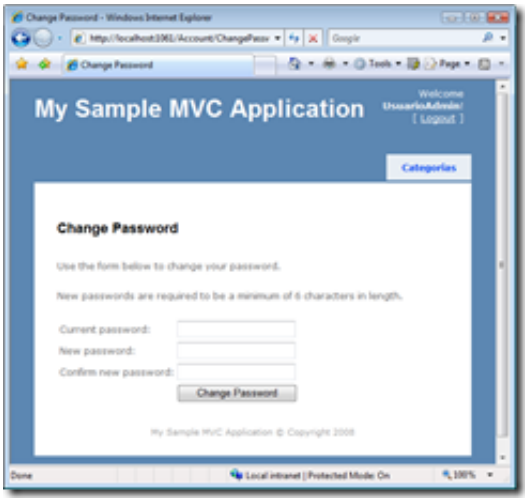


Não gostei dessa história. Porque não posso ser direcionado de volta à mesma página, para exibir uma mensagem? Busquei uma forma simples. Ainda não há.

É possível ainda incluir um novo usuário via link de Register, ação também chamada Register (pode ser visto na primeira imagem do IE acima):



E trocar de senha:



Uma das coisas que achei mais interessantes está na ação de login: ela posta para si mesma, e dessa forma controla exibição de erros, como senha muito curta. Muito inteligente.

Conclusão dessa verificação do código de autorização e autenticação: com esse exemplo já dá para confirmar que o negócio funciona direitinho, mas precisa ainda evoluir um pouco em usabilidade e extensibilidade. Vamos aguardar agora o Preview 5, que deve sair daqui uns 2 meses.

Like 0 Send 0 Tweetar 0

Esse post foi publicado em **Microsoft** e marcado **ASP.Net MVC** por **Giovanni Bassi**. Guardar **link permanente [http://blog.lambda3.com.br/2008/07/autenticao-e-autorizao-com-asp-net-mvc/]** .

Sobre Giovanni Bassi

Arquiteto e desenvolvedor, agilista, escalador, provocador.
Programa porque gosta, e começou a trabalhar com isso porque acha que trabalhar como administrador é meio chato. Por esse motivo sempre diz que nunca mais vai virar gerente de ninguém. E também porque acredita que pessoas autogerenciadas funcionam melhor e por acreditar que heterarquia é melhor que hierarquia. Mas isso é outro assunto.
Foi reconhecido **Microsoft MVP** depois que alguém notou que ele não dormia a noite pra ficar escrevendo artigos, cuidando e participando do **.Net Architects**, gravando o podcast **Tecnoretórica**, escrevendo posts no **blog** e falando o que bem entende no twitter **@giovannibassi**. E por falar nisso é no twitter que conta pra todos que gerencia de projetos deve ser feita pelo time e não por um gerentes, que greves em TI são coisas sem sentido e que stored procedure com regras de negócio são malígnas.
Você já deve ter percebido (até porque está lá na primeira frase) que **Giovanni** é agilista. De tanto gostar disso ele **trouxe** os programas de certificação e treinamento **PSD** e **PSM** da **Scrum.org** pro Brasil, e por causa deles, do MVP e de algum trabalho que aparece tem que ficar indo pros EUA de vez enquando, coisa que prefere não fazer. (É bem comum você ouvir ele perguntando porque a Scrum.org e a Microsoft não estão na Itália, por exemplo.)
Junto com alguns Jedis criou a **Lambda3**, que, apesar de ser pequena e de não ser muito comum no Brasil, insiste em fazer projetos e consultoria direito. Por causa da Lambda3 ele tem trabalhado mais do que quando era consultor independente, mas menos do que a maioria das pessoas. Quer dizer, isso se você considerar que os trabalhos junto à comunidade não são trabalho, caso contrário ele trabalha mais que a maioria das pessoas.
Recentemente ele resolveu que merecia viver melhor e ganhar uns anos de vida e desistiu de ser sedentário, fazendo algum barulho de vez em quando com os amigos no twitter com a hashtag #DotNetEmForma. Por causa do convite recente de amigos do lado Open Source (que ele respeita e admira), começou a escalar, e agora está sempre com as mãos machucadas. Mas ainda dá pra programar. Você encontra ele sempre em algum evento, como o TechEd, e o DNAD, mas também outros menos comuns para o pessoal do .NET, como a RubyConf. Nesses eventos, ou ele está vendo palestras, ou batendo papo com alguém, ou codando alguma aplicação que alguém achou que dava pra fazer durante o evento.
Ver todas as mensagens por Giovanni Bassi →

8 comentários

0

Deixar uma mensagem...

Mais novos **Comunidade** **Compartilhar**

Avatar

Rodrigo • 2 meses atrás

E se eu quiser criar roles personalizadas? sem ter que usar annotation(acho que é isso), Tem como? como seria?

Responder Compartilhar ›

Avatar

Giovanni Bassi • 3 anos atrás

É seguro.

1 Responder Compartilhar ›

Avatar

Mateus • 3 anos atrás

Giovanni, minha dúvida é se o Membership é seguro contra ataques de SQL Injection, pretendo fazer um projeto e publica-lo na web, dai a pergunta se alguem saber q estou usando p Banco de dados do Aspnet (ASPNETDB.MDF) alguém pode tentar deletar alguma tabela ou remover registros pelo fato deste cara saber os nomes das tabelas e seus campos.

1 Responder Compartilhar ›

Avatar

Giovanni Bassi • 5 anos atrás

Mario, dê uma olhada na categoria do MVC aqui no blog, tem um monte de links:
http://unplugged.giggio.net/ca...

Responder Compartilhar ›

Avatar

Marcio Cleber • 5 anos atrás

Giovanni,

Me indique alguns links desse assunto. achei alguns e estou estudando.

Valeu!!

Responder Compartilhar ›

Avatar

Marcio Cleber • 5 anos atrás

Olá Giovanni,

Sou novo na área de .NET, mas valeu pela informação.
Achei que roles era apenas pra MVC.

Valeu!!

Responder Compartilhar ›

Avatar

Giovanni Bassi • 5 anos atrás

Marcio,
Para isso existem roles, que já existe desde o começo do ASP.Net.
Você utiliza a própria infra-estrutura do ASP.Net para isso.

Responder Compartilhar ›

Avatar

Marcio Cleber • 5 anos atrás

Olá,

Achei muito interessante seu post, porém gostaria de saber como eu faço para criar dois tipos de usuários, o usuário normal e o usuário admin.