**[CookieAuthenticationOptions类 （MVC身份验证）](http://blog.csdn.net/fanbin168/article/details/49923505)**

2015-11-19 09:55 3806人阅读 [评论](http://blog.csdn.net/fanbin168/article/details/49923505#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/fanbin168/article/details/49923505#report)

http://static.blog.csdn.net/images/category_icon.jpg分类：

C#*（106）* http://static.blog.csdn.net/images/arrow_triangle%20_down.jpgASP.NET自带工具类

版权声明：本文为博主原创文章，未经博主允许不得转载。

1

[参考资料：MVC新特性](http://www.cnblogs.com/webabcd/p/3510032.html)

[重要的参考资料：使用OWIN验证的中间件](http://codego.net/608804/)

[最重要的参考资料：ASP.NET Identity身份验证和基于角色授权](http://www.cnblogs.com/OceanEyes/archive/2015/09/06/4787576.html)

定义与介绍

**[csharp]** [view plain](http://blog.csdn.net/fanbin168/article/details/49923505) [copy](http://blog.csdn.net/fanbin168/article/details/49923505)

1. using Microsoft.Owin;
2. using Microsoft.Owin.Infrastructure;
3. using Microsoft.Owin.Security;
4. using System;
6. namespace Microsoft.Owin.Security.Cookies
7. {
8. // 摘要:
9. //     包含由cookiesauthenticationmiddleware使用的选项
10. public class CookieAuthenticationOptions : AuthenticationOptions
11. {
12. // 摘要:
13. //    创建与默认值初始化的选项的一个实例
14. public CookieAuthenticationOptions();
16. // 摘要:
17. //     确定用于创建的cookie的域。不提供默认。
18. public string CookieDomain { get; set; }
19. //
20. // 摘要:
21. //确定是否浏览器应该允许由客户端JavaScript来访问该cookie。默认值是true，
22. //这意味着该cookie将只通过HTTP请求，而不是提供给脚本的页面上。
23. public bool CookieHttpOnly { get; set; }
24. //
25. // 摘要:
26. // 确定用于保存标识cookie名称。默认值是“.AspNet.Cookies”。
27. // 如果你改变了AuthenticationType的名称，这个值应该被改变，特别是如果你的系统使用cookie认证中间件多次。
28. public string CookieName { get; set; }
29. //
30. // 摘要:
31. //确定用于创建cookie中的路径。默认值是“/”为最高的浏览器兼容性。
32. public string CookiePath { get; set; }
33. //
34. // 摘要:
35. //确定是否饼干应只对HTTPS请求转交。默认是限制cookie来HTTPS请求，
36. //如果这是在做签到的页面也HTTPS。如果你有一个HTTPS登录页面和网站的部分是HTTP，您可能需要修改这个值。
37. public CookieSecureOption CookieSecure { get; set; }
38. //
39. // 摘要:
40. //控制cookie将多少时间留在其创建点有效。到期信息是受保护的cookie票。
41. //正因为如此，即使它被传递给服务器过期cookie将被忽略后浏览器应该清除它
42. public TimeSpan ExpireTimeSpan { get; set; }
43. //
44. // 摘要:
45. //该LoginPath属性通知，它应该改变传出401 Unauthorized状态码为302重定向到指定的登录路径中间件。
46. //将产生的401的当前的URL被添加到LOGINPATH作为命名由ReturnUrlParameter查询字符串参数。
47. //一旦请求到LOGINPATH赋予一个新的登入标识，ReturnUrlParameter值用于浏览器重定向回造成原始未经授权的状态码的网址。
48. //如果LoginPath为null或空，中间件也不会去找401 Unauthorized状态码，并在登录时它不会自动重定向。
49. public PathString LoginPath { get; set; }
50. //
51. // 摘要:
52. //如果LogoutPath设置在中间件然后一个请求发送到该路径将重定向基于所述ReturnUrlParameter。
53. public PathString LogoutPath { get; set; }
54. //
55. // 摘要:
56. //提供者可以被分配给由在启动时所述应用程序创建的对象的实例。
57. //中间件呼吁提供这给在某些地方应用控制在哪里处理正在发生的方法。
58. //如果它不提供的默认实例被提供其中不执行任何时调用方法。
59. public ICookieAuthenticationProvider Provider { get; set; }
60. //
61. // 摘要:
62. //所述ReturnUrlParameter确定其由中间件所附当一个401 Unauthorized状态码被改变成302重定向到登录路径查询字符串参数的名称。
63. //这也是查询字符串参数看起来当一个请求到达该登录路径或注销路径上，为了执行该操作后，返回到原来的URL的。
64. public string ReturnUrlParameter { get; set; }
65. //
66. // 摘要:
67. //该SlidingExpiration设置为true，以指示中间件到一个新的到期时间重新发一个新的Cookie它处理一个请求是超过一半的过期窗口中的任何时间。
68. public bool SlidingExpiration { get; set; }
69. //
70. // 摘要:
71. //该SystemClock提供访问系统的当前时间坐标。如果它不提供的默认实例被用于其中要求DateTimeOffset.UtcNow。
72. //这通常是不会被替换，除非进行单元测试。
73. public ISystemClock SystemClock { get; set; }
74. //
75. // 摘要:
76. //所述TicketDataFormat用于保护和不保护的身份和被存储在cookie值等性能。
77. //如果它不提供一个缺省数据处理程序是使用包含在数据保护服务创建在IAppBuilder.Properties。
78. //在ASP.NET运行时，默认的数据保护服务是基于计算机密钥，并DPAPI在不同的进程中运行时。
79. public ISecureDataFormat<AuthenticationTicket> TicketDataFormat { get; set; }
80. }
81. }

使用（位置：MVC50/App\_Start/Startup.Auth.cs）

**[csharp]** [view plain](http://blog.csdn.net/fanbin168/article/details/49923505) [copy](http://blog.csdn.net/fanbin168/article/details/49923505)

1. using Microsoft.AspNet.Identity;
2. using Microsoft.Owin;
3. using Microsoft.Owin.Security.Cookies;
4. using Owin;
6. namespace MVC50
7. {
8. public partial class Startup
9. {
10. // 有关配置身份验证的详细信息，请访问 http://go.microsoft.com/fwlink/?LinkId=301864
11. public void ConfigureAuth(IAppBuilder app)
12. {
13. // 使应用程序可以使用 Cookie 来存储已登录用户的信息
14. app.UseCookieAuthentication(new CookieAuthenticationOptions
15. {
16. AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
17. LoginPath = new PathString("/Account/Login")
18. });
19. // Use a cookie to temporarily store information about a user logging in with a third party login provider
20. app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);
22. // 取消注释以下行可允许使用第三方登录提供程序登录
23. //app.UseMicrosoftAccountAuthentication(
24. //    clientId: "",
25. //    clientSecret: "");
27. //app.UseTwitterAuthentication(
28. //   consumerKey: "",
29. //   consumerSecret: "");
31. //app.UseFacebookAuthentication(
32. //   appId: "",
33. //   appSecret: "");
35. // 支持使用 google 账户登录
36. app.UseGoogleAuthentication();
37. }
38. }
39. }

