音乐商店技术文档

# 1身份验证

1. 创建用户模型
2. 种子函数创建实例
3. 上下文类添加用户实体集
4. 访问该行为时，需要进行身份验证

[AdminFilter] //绑定过滤器

[Authorize]//验证特性

public ActionResult Download()

{

return View();

}

1. 验证特性也可放在**控制器**前，这时访问该控制器的任何行为都需要身份验证。
2. 验证特性也可以设置**全局限制App\_start /FilterConfig.cs**

public static void RegisterGlobalFilters(GlobalFilterCollection filters)

{

filters.Add(new HandleErrorAttribute());

filters.Add(new AuthorizeAttribute());

}

1. 配置文件设置表单授权处理路由

<system.web>

<compilation debug="true" targetFramework="4.5.2" />

<httpRuntime targetFramework="4.5.2" />

<!--配置文件设置表单授权处理路由-->

<authentication mode="Forms">

<forms loginUrl="~/Authentication/Login"></forms>

</authentication>

<httpModules>

<add name="ApplicationInsightsWebTracking" type="Microsoft.ApplicationInsights.Web.ApplicationInsightsHttpModule, Microsoft.AI.Web" />

</httpModules>

</system.web>

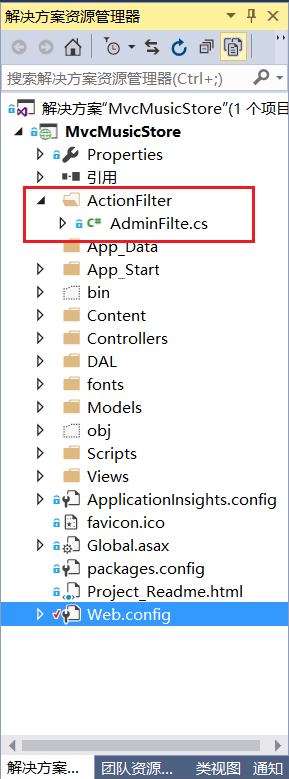
1. 设置匿名访问控制器/行为

[AllowAnonymous] //匿名特性

匿名特性可使需要身份验证的控制器或行为免验证

# 2过滤器

1. 在项目中创建一个过滤器文件夹，并创建一个过滤器



public class AdminFilter : ActionFilterAttribute

{

public override void OnActionExecuting(ActionExecutingContext filterContext)

{

if (!Convert.ToBoolean(filterContext.HttpContext.Session["IsAdmin"]))

{

filterContext.Result = new ContentResult() { Content = "没有访问权限" };

}

}

}

1. 绑定数据库

[AdminFilter] //绑定过滤器

[Authorize]//验证特性

public ActionResult Download()

{

return View();

}

1. 3关闭数据库连接

/// <summary>

/// 关闭数据库连接

/// </summary>

/// <param name="disposing"></param>

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

# 3方法

1. RedirectToAction

使用操作名称和控制器名称重定向到指定的操作。

# 4html辅助器

1. @Html.ValidationMessage验证错误是提示

<div>

@Html.ValidationMessage("CredentialError", new { style = "color:red;" })

@using (Html.BeginForm("Login", "Authentication", FormMethod.Post))

{

@Html.LabelFor(c => c.UserName)

@Html.TextBoxFor(x => x.UserName)

<br />

@Html.LabelFor(c => c.Password)

@Html.PasswordFor(x => x.Password)

<br />

<input type="submit" name="BtnSubmit" value="Login" />

}

</div>

public ActionResult Login(UserDetails u)

{

if (ModelState.IsValid)

{

UserStatus status = GetUserValidity(u);

bool IsAdmin = false;

if (status == UserStatus.AuthenticatedAdmin)

{

IsAdmin = true;

}

else if (status == UserStatus.AuthentucatedUser)

{

IsAdmin = false;

}

else

{

ModelState.AddModelError("CredentialError", "用户名密码无效");

return View();

}

//设置cookie 保存在浏览器

FormsAuthentication.SetAuthCookie(u.UserName, false);

// 保存在服务器端,关闭浏览器后session就清空

Session["IsAdmin"] = IsAdmin;

return RedirectToAction("Index", "Home");

}

else

{

return View();

}

}

1. @Html.ActionLink（）

返回指定的链接文本和操作的定位点元素（a 元素）。

可指定访问到某个控制器下的某个行为

1. @Html.DisplayFor（）

可返回指定的model中的属性的值