

译 ASP.NET Core 2.0身份和角色管理入门

2018年12月08日 15:25:50 寒冰屋 阅读数 921

目录

介绍

身份验证和授权

身份验证

授权

背景

先决条件

使用代码

第1步：创建数据库

第2步：创建ASP.NET Core

更新appsettings.json

步骤3：在Startup.cs文件中添加Identity Service

第4步：注册并创建您的第一个用户

迁移

刷新数据库

第5步：创建角色并为角色分配用户

第6步：创建管理页面和设置授权

创建Admin控制器

设置授权

第7步：按用户角色显示隐藏菜单

Download ASPNETCoreUserIdentity - 1.5 MB (建议按照文章自己先做)

介绍

在本文中，我们将详细介绍如何在MVC应用程序中使用ASP.NET Core Identity来创建用户角色并根据用户角色显示菜单。

在这里，我们将看到如何做：

- 创建默认管理员用户
- 创建默认管理员角色
- 将未经身份验证的用户重定向到登录页面
- 仅为授权管理员用户显示管理页面菜单

ASP.NET Identity允许我们为系统添加登录功能。在这个演示中，我们将使用SQL Server来存储用户详细信息和配置文件数据。我们将Identity进行新用户注册，登录以及维护用户配置文件数据。如果我们谈论登录，重要的部分是登录用户是否经过身份验证并且是否有权查看页

身份验证和授权

身份验证



检查有效用户。这里的问题是如何检查用户是否有效。当用户第一次访问网站时，他/她将注册该网站。他们所有的信息，如用户名，密码，电子邮件地址，都存储在网站数据库中。当用户输入他/她的用户ID和密码时，将使用数据库检查信息。如果用户输入了与数据库中相同的用户ID和密码，则他/她将被重定向到网站的主页。如果用户输入的UserID或密码与数据库不匹配，则登录页面将显示一条消息，例如“请输入有效的用户名或密码”。检查网站的整个过程称为身份验证。

授权

用户通过身份验证后，需要通过他/她的角色将其重定向到相应的页面。例如，当管理员登录时，则需要重定向到管理页面。如果会计师登录，则她重定向到其“帐目”页面。

背景

先决条件

确保已安装计算机中的所有先决条件。如果没有，那么一个接一个地下载并安装它们。

1. 首先，从此[链接](#)下载并安装Visual Studio 2017
2. SQL Server 2014或更高版本

使用代码

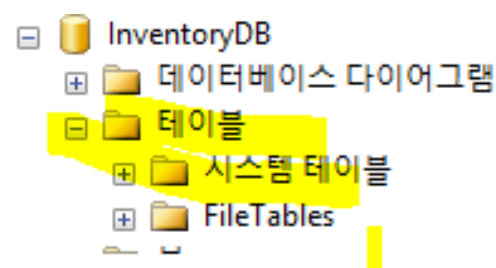
第1步：创建数据库

首先，我们将创建一个数据库，并使用我们的新数据库连接在 *appsettings.json* 文件中设置连接字符串 **DefaultConnection**。我们将使用此类ASP.NET Core Identity表。

创建数据库：运行以下脚本以创建我们的数据库。

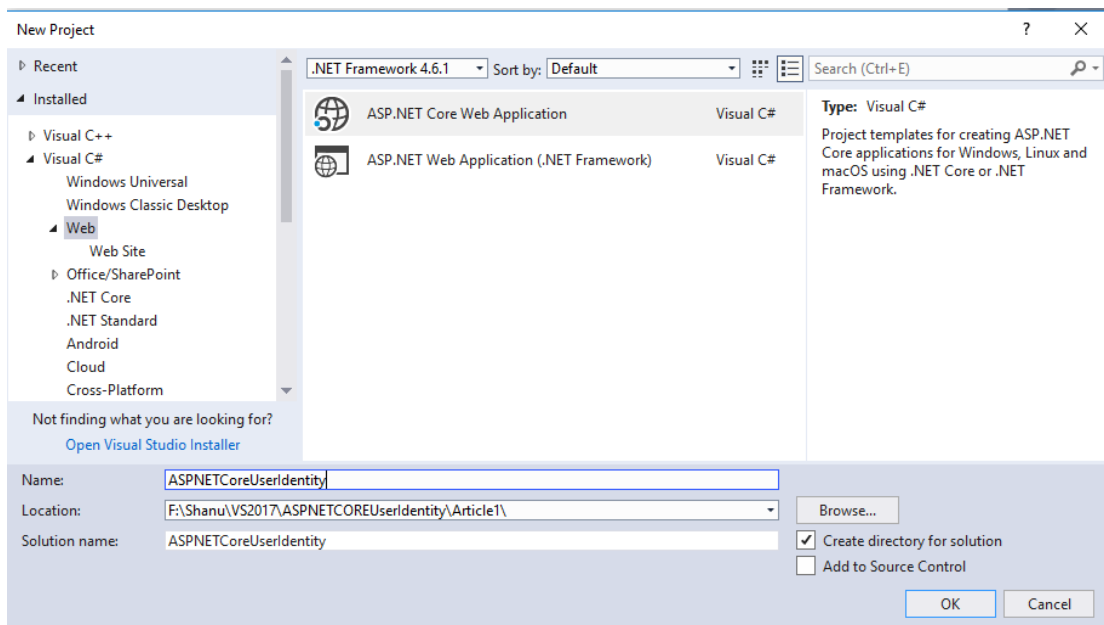
```
1 USE MASTER
2 GO
3
4 -- 1) Check for the Database Exists .If the database is exist then drop and create new DB
5 IF EXISTS (SELECT [name] FROM sys.databases WHERE [name] = 'InventoryDB' )
6     DROP DATABASE InventoryDB;
7 GO
8
9 CREATE DATABASE InventoryDB;
10 GO
11
12 USE InventoryDB
13 GO
```

运行数据库脚本后，我们可以看到已创建数据库并且尚未创建表。

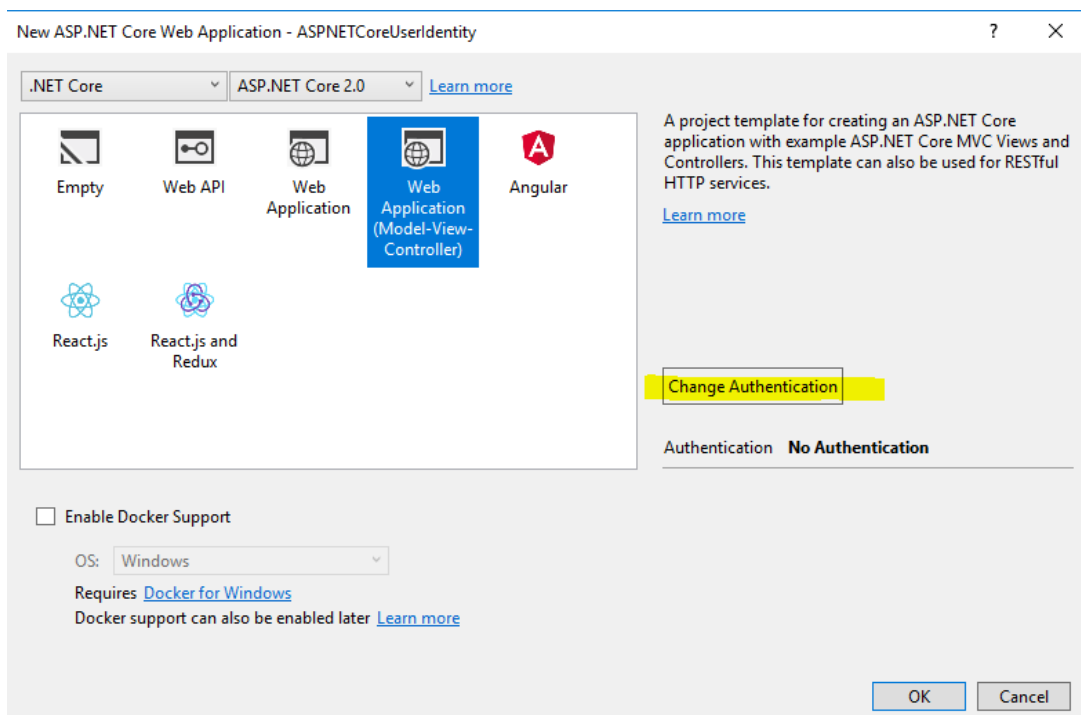


第2步：创建ASP.NET Core

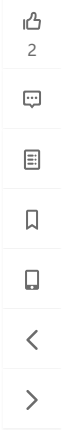
安装Visual Studio 2017后，单击“开始”，然后单击“程序”，选择“Visual Studio 2017”——单击“Visual Studio 2017”。单击Project，选择Web，然后选择 **ASP.NET Core Web Application**。输入您的项目名称并单击。

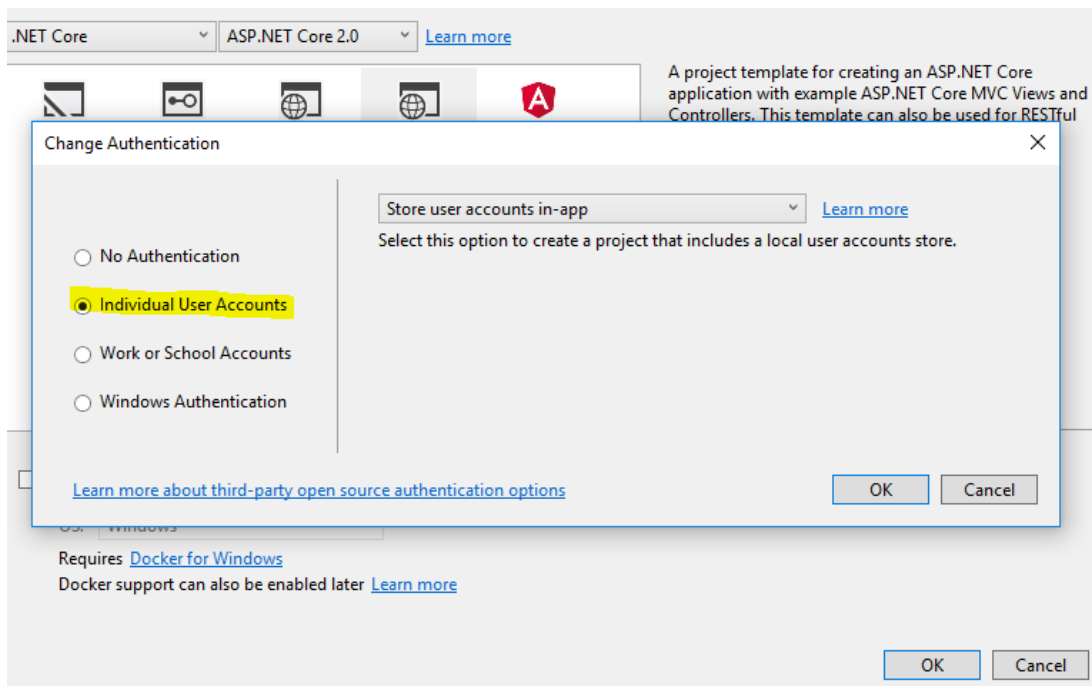


选择Web应用程序（模型-视图-控制器），然后单击“更改身份验证”。



选择个人用户帐户，然后单击确定以创建项目。





更新appsettings.json

在`appsettings.json`文件中，我们可以找到 `DefaultConnection` 连接字符串。在连接字符串中，更改你的SQL Server名称，UID和PWD以在一创建和存储所有用户详细信息。

```
1 "ConnectionStrings": {
2   "DefaultConnection": "Server= YOURSERVERNAME;Database=InventoryDB;
3     user id= YOURSQLUSERID;password=YOURSQLPASSWORD;Trusted_Connection=True;
4     MultipleActiveResultSets=true"
5 },
```

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Server= [redacted];Database=InventoryDB;user id= [redacted];password=[redacted];Trusted_Connection=True;MultipleActiveResultSets=true"
  },
  "Logging": {
    "IncludeScopes": false,
    "LogLevel": {
      "Default": "Warning"
    }
  }
}
```

步骤3：在Startup.cs文件中添加Identity Service

默认情况下，在ASP.NET Core应用程序中，Identity Service将添加到`Startup.cs`文件中的 `ConfigureServices`方法中。您还可以在用户注册时强度，还可以使用以下代码设置默认登录页/注销页以及`AccessDeniedPath`。

```
1 services.AddIdentity<ApplicationUser, IdentityRole>()
2     .AddEntityFrameworkStores<ApplicationDbContext>()
3     .AddDefaultTokenProviders();
4
5 //Password Strength Setting
6 services.Configure<IdentityOptions>(options =>
7 {
8     // Password settings
9     options.Password.RequireDigit = true;
10    options.Password.RequiredLength = 8;
11    options.Password.RequireNonAlphanumeric = false;
12    options.Password.RequireUppercase = true;
13    options.Password.RequireLowercase = false;
14    options.Password.RequiredUniqueChars = 6;
15
16    // Lockout settings
17    options.Lockout.DefaultLockoutTimeSpan = TimeSpan.FromMinutes(30);
18    options.Lockout.MaxFailedAccessAttempts = 10;
19    options.Lockout.AllowedForNewUsers = true;
20
21    // User settings
```

```

22 |         options.User.RequireUniqueEmail = true;
23 |     });
24 |
25 |     //Setting the Account Login page
26 |     services.ConfigureApplicationCookie(options =>
27 |     {
28 |         // Cookie settings
29 |         options.Cookie.HttpOnly = true;
30 |         options.ExpireTimeSpan = TimeSpan.FromMinutes(30);
31 |         options.LoginPath = "/Account/Login"; // If the LoginPath is not set here,
32 |                                           // ASP.NET Core will default to /Account/Login
33 |         options.LogoutPath = "/Account/Logout"; // If the LogoutPath is not set here,
34 |                                           // ASP.NET Core will default to /Account/Logout
35 |         options.AccessDeniedPath = "/Account/AccessDenied"; // If the AccessDeniedPath is
36 |                                           // not set here, ASP.NET Core
37 |                                           // will default to
38 |                                           // /Account/AccessDenied
39 |         options.SlidingExpiration = true;
40 |     });

```



下面是我们如何在ConfigureService方法中添加ASP.NET Core身份服务如下:

```

public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<ApplicationDbContext>(options =>
    {
        options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection"));
    });

    services.AddIdentity<ApplicationUser, IdentityRole>()
        .AddEntityFrameworkStores<ApplicationDbContext>()
        .AddDefaultTokenProviders();

    //Password Strength Setting
    services.Configure<IdentityOptions>(options =>
    {
        // Password settings
        options.Password.RequireDigit = true;
        options.Password.RequiredLength = 8;
        options.Password.RequireNonAlphanumeric = false;
        options.Password.RequireUppercase = true;
        options.Password.RequireLowercase = false;
        options.Password.RequiredUniqueChars = 6;

        // Lockout settings
        options.Lockout.DefaultLockoutTimeSpan = TimeSpan.FromMinutes(30);
        options.Lockout.MaxFailedAccessAttempts = 10;
        options.Lockout.AllowedForNewUsers = true;

        // User settings
        options.User.RequireUniqueEmail = true;
    });

    //Setting the Account Login page
    services.ConfigureApplicationCookie(options =>
    {
        // Cookie settings
        options.Cookie.HttpOnly = true;
        options.ExpireTimeSpan = TimeSpan.FromMinutes(30);
        options.LoginPath = "/Account/Login"; // If the LoginPath is not set here, ASP.NET Core will default to /Account/Login
        options.LogoutPath = "/Account/Logout"; // If the LogoutPath is not set here, ASP.NET Core will default to /Account/Logout
        options.AccessDeniedPath = "/Account/AccessDenied"; // If the AccessDeniedPath is not set here, ASP.NET Core will default to /Account/AccessDenied
        options.SlidingExpiration = true;
    });

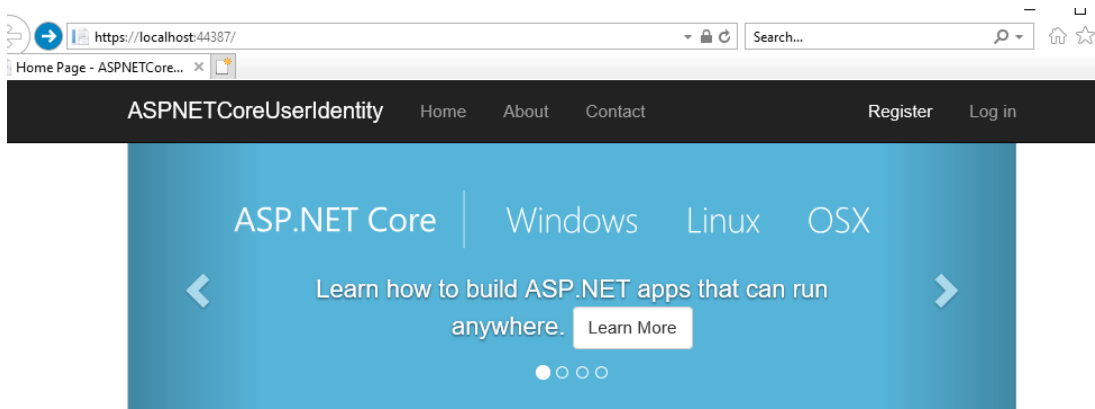
    // Add application services.
    services.AddTransient<EmailSender, EmailSender>();

    services.AddMvc();
}

```

第4步：注册并创建您的第一个用户

现在，我们的ASP.NET Core Web应用程序已准备好供用户在我们的网站上注册，且用户也可以在注册后登录我们的系统。我们将通过在后续添加角色来执行授权。构建并运行应用程序以注册您的第一个默认管理员用户。



单击注册链接注册我们的第一个用户。(PS: 建议注册用户和下图一样, 否则下面创建Admin角色时, 其方法中的对应用户名改为您自行注册的问题, 把Configure方法中最后一句注释掉, 创建角色在下面的步骤中会介绍, 但是在创建数据库表之前不需要——如果您的此文提供的案例不虑。)

[ASPNETCoreUserIdentity](#)[Home](#)[About](#)[Contact](#)[Register](#)[Log in](#)


Register

Create a new account.

Email

Password

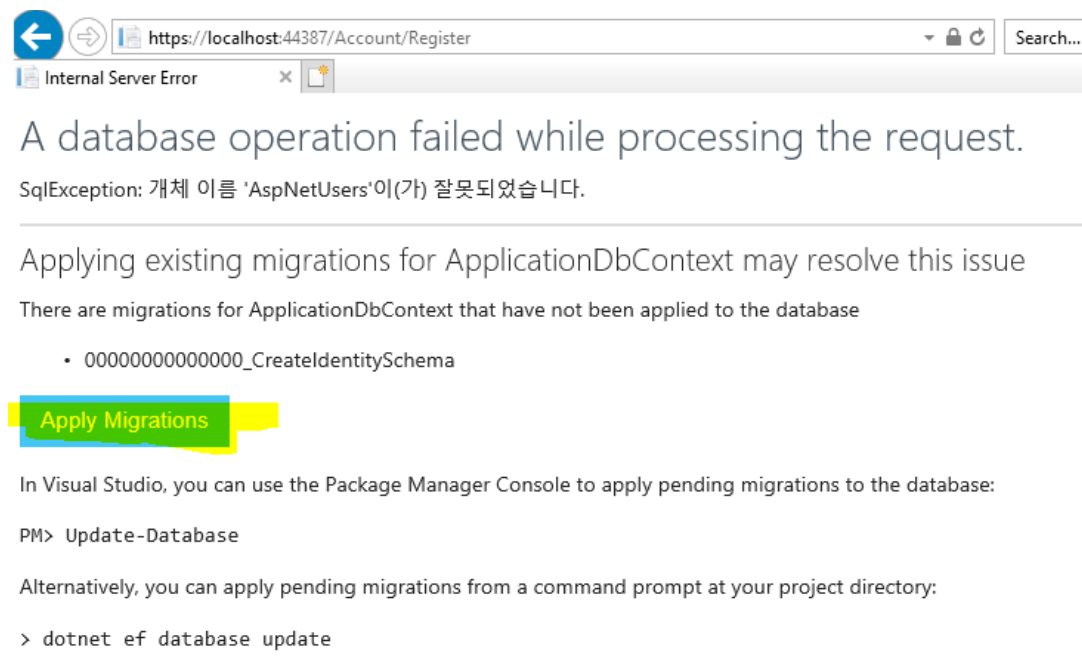
Confirm password



Register

迁移

当我们点击“注册”按钮时，我们可以看到下面的页面。不要对这个页面感到恐慌，因为第一次运行我们需要进行迁移，只需单击Apply Migrat



我们可以看到确认为“已应用迁移”，然后单击“尝试刷新页面消息”。

- 🗨️
- 👍
2
- 💬
- 📄
- 🔖
- 📱
- <
- >

				
---	---	---	---	---



SQLException: 개체 이름 'AspNetUsers'이(가) 잘못되었습니다.

There are migrations for ApplicationDbContext that have not been applied to the database

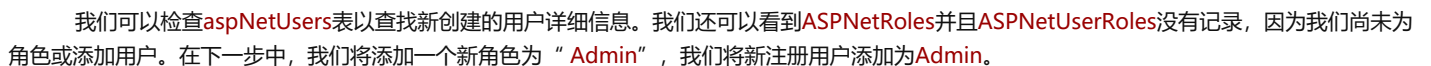
- Migrations Applied Try refreshing the page

Alternatively, you can apply pending migrations from a command prompt at your project directory:

刷新页面，我们可以看到新注册的用户已登录我们的网站。



刷新数据库时，我们可以看到所有的Identity表都已创建。



2

 < >

startup

目程序目




```

0 references | 0 exceptions
public void Configure(IApplicationBuilder app, IHostingEnvironment env, IServiceProvider services)
{
    if (env.IsDevelopment())
    {
        app.UseBrowserLink();
        app.UseDeveloperExceptionPage();
        app.UseDatabaseErrorPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
    }

    app.UseStaticFiles();

    app.UseAuthentication();

    app.UseMvc(routes =>
    {
        routes.MapRoute(
            name: "default",
            template: "{controller=Home}/{action=Index}/{id?}");
    });

    CreateUserRoles(services).Wait();
}

```

当我们构建并运行应用程序时，我们可以看到ASPNetRoles表中添加了新角色，并且我们可以看到我们的默认用户已被分配了 Admin角色。

InventoryDBQuery.s...ventoryDB (sa (32))

```

select * from aspNetUsers
select * from ASPNetRoles
Select * From ASPNetUserRoles

```

100 %

결과 메시지

	Id	AccessFailedCount	ConcurrencyStamp	Email
1	2f7354da-02a0-4bfb-97f4-f6d55e78db32	0	30b8db25-19eb-4168-9c7d-646ff3021c7c	sys

	Id	ConcurrencyStamp	Name	NormalizedName
1	2fa97882-0bca-457d-b584-37c6af82e718	0ba2af3b-6803-4661-9f92-ff2531d410f4	Admin	ADMIN

	UserId	RoleId
1	2f7354da-02a0-4bfb-97f4-f6d55e78db32	2fa97882-0bca-457d-b584-37c6af82e718

第6步：创建管理页面和设置授权

现在有一个用于ASP.NET Core Web应用程序的Admin用户。下一步，让我们创建一个新页面并将此页面的授权设置为仅登录，仅管理员。为此，我们创建了一个新命名名为Admin的Controller。

创建Admin控制器

右键单击Controller文件夹并单击Add New Controller，选择MVC Controller - Empty，然后单击Add。

Add Empty MVC Controller

Controller name:
AdminController

Add
Cancel

输入您的Controller名称为Admin，然后单击Add。

Add MVC View

View name:

Template:

Model class:

Data context class:

Options:

☐ Create as a partial view

☒ Reference script libraries

☒ Use a layout page:

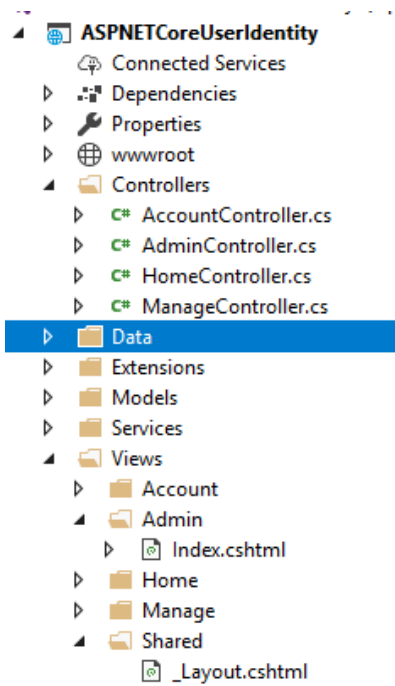
...

(Leave empty if it is set in a Razor _viewstart file)

在控制器中，右键单击Index，然后单击添加视图。单击“添加”按钮以创建视图页面。

```
1
2  @{
3      ViewData["Title"] = "Admin Page";
4  }
5
6  <h2>Admin Page</h2>
7  <h4>Only Admin Can View this page !!</h4>
8
```

我们可以看到我们的管理控制器和管理视图已创建。



打开 *Admin / Index.cshtml* 页面以根据需要进行设计。在这里，我添加了如下的简单文字：

```

1
2   @{
3       ViewData["Title"] = "Admin Page";
4   }
5
6   <h2>Admin Page</h2>
7   <h4>Only Admin Can View this page !!</h4>
8

```

接下来，我们创建一个新菜单以显示Admin Page。要创建新菜单，请从 Views / Shared 文件夹中打开 _Layout.cshtml。添加如下图所示

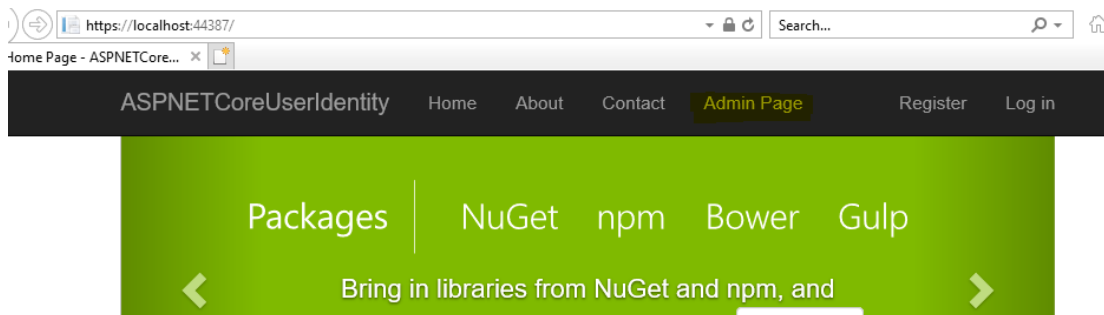
```

<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <li><a asp-area="" asp-controller="Home" asp-action="Index">Home</a></li>
    <li><a asp-area="" asp-controller="Home" asp-action="About">About</a></li>
    <li><a asp-area="" asp-controller="Home" asp-action="Contact">Contact</a></li>
    <li><a asp-area="" asp-controller="Admin" asp-action="Index">Admin Page</a></li>
  </ul>
  @await Html.PartialAsync("_LoginPartial")
</div>

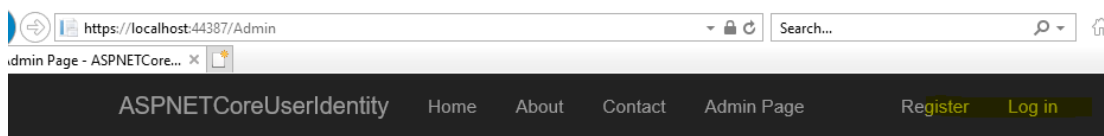
```

现在我们已经创建了管理页面，并为管理员添加了菜单。我们仅为Admin用户创建了此页面，其他用户或未登录的用户不应该看到此页面。如我们的应用程序会发生什么。

我们可以看到新的菜单“Admin Page”已经创建，现在对所有人开放。这意味着任何人都可以单击该链接并查看该页面的内容。



在这里，我们可以看到，我们可以通过登录查看管理页面。



Admin Page

Only Admin Can View this page !!

设置授权

为避免这种情况，我们在管理页面控制器中使用授权。打开我们的 Admin 控制器并添加以下代码行：

```

1 [Authorize(Roles = "Admin")]
2 public IActionResult Index()
3 {
4     return View();
5 }

```

```

namespace ASPNETCoreUserIdentity.Controllers
{
    0 references
    public class AdminController : Controller
    {
        [Authorize(Roles = "Admin")]
        0 references | 1 request | 0 exceptions
        public IActionResult Index()
        {
            return View();
        }
    }
}

```



如果我们运行我们的应用程序并单击Admin Page，它将自动重定向到Log in页面。

请注意，只有管理员角色成员才能查看管理页面，因为我们仅为管理员角色设置了授权。如果您想添加更多角色，我们可以使用逗号，如下面的

```
[Authorize(Roles = "Admin,SuperAdmin,Manager")]
```

第7步：按用户角色显示隐藏菜单

现在让我们更进一步的只显示登录管理员用户的管理员菜单。为此，我们从 *Views/Shared* 文件夹下打开 *Layout.cshtml* 并编辑新添加的菜单，码。在这个代码中，首先我们检查用户是否已经过身份验证，意味着登录，然后我们检查用户是否具有查看菜单的授权。

```

1 <li>
2     @if (User.Identity.IsAuthenticated)
3     {
4         @if (User.IsInRole("Admin"))
5         {
6             <a asp-area="" asp-controller="Admin" asp-action="Index">Admin Page</a>
7         }
8     }
9 </li>

```

以下是我们的代码：

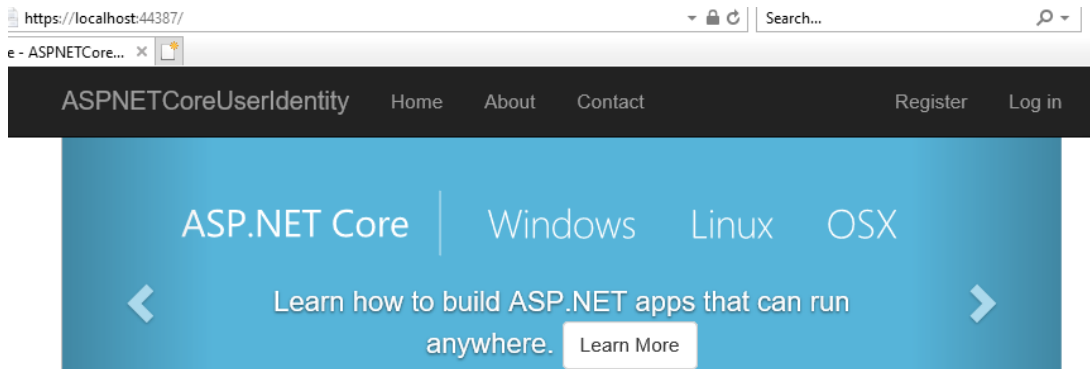


```

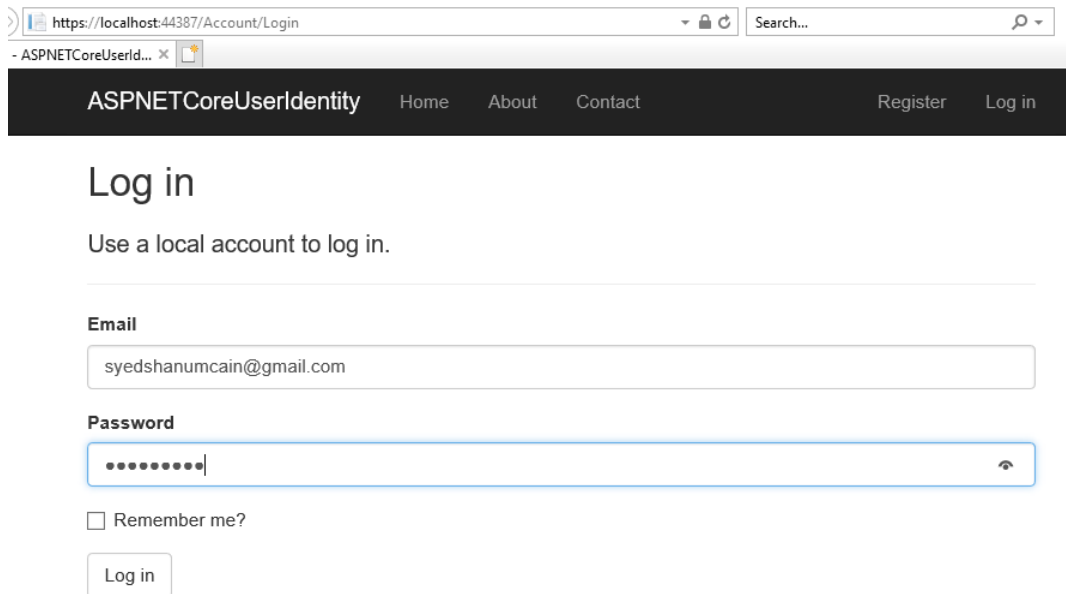
<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <li><a asp-area="" asp-controller="Home" asp-action="Index">Home</a></li>
    <li><a asp-area="" asp-controller="Home" asp-action="About">About</a></li>
    <li><a asp-area="" asp-controller="Home" asp-action="Contact">Contact</a></li>
    <li>
      @if (User.Identity.IsAuthenticated)
      {
        @if (User.IsInRole("Admin"))
        {
          <a asp-area="" asp-controller="Admin" asp-action="Index">Admin Page</a>
        }
      }
    </li>
  </ul>
  @await Html.PartialAsync("_LoginPartial")

```

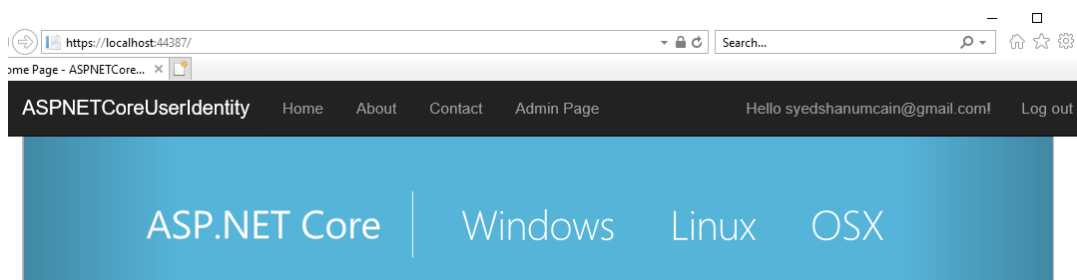
运行应用程序，默认情况下我们可以看到“Admin Page”不会显示在我们的顶层菜单中。仅登录管理员角色用户可以查看菜单。



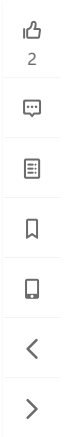
让我们通过我们最初创建的管理员用户登录来试试这个。



登录后，我们可以看到管理员用户现在可以查看Admin Page菜单。



让我们尝试创建普通用户，因为我们现在在注册新用户。



Register

Create a new account.

Email

test@test.com

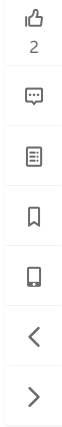
Password

.....

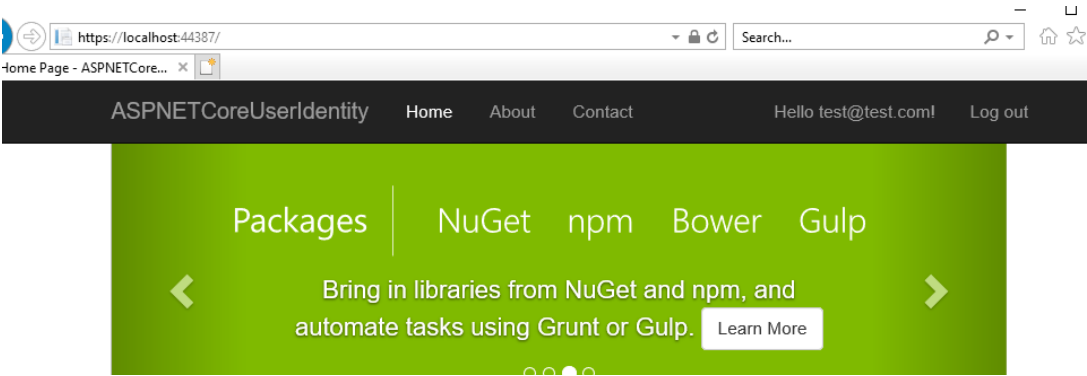
Confirm password

.....

Register



注册后，我们可以看到，对于这个用户，我们没有添加“Admin&rdquorole”，他无权查看Admin Page。



参考链接：<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?tabs=visual-studio%2Caspnetcore2x>

原文地址：<https://www.codeproject.com/Articles/1235077/Getting-Started-With-ASP-NET-Core-Identity-And-Rol>

[推动全社会公益氛围形成，使公益与空气和阳光一样触手可及。](#)
公益缺你不可，众多公益项目等你PICK——百度公益 让公益像「空气和阳光」一样触手可及！
gongyi.baidu.com



想对作者说点什么

ASP.NET Core 2.1中基于角色的授权

阅读数 201

授权是用来描述用户能够做什么的过程。例如，只允许管理员用户可以在电脑上进行软... 博文 来自：[dotNET跨...](#)

asp.net core 2.0 Web简单使用：二、添加默认用户，修改登录方式

阅读数 1745

后台登录一般懒得用邮箱了，太长，所以改成用户名登录。首先添加默认用户先删除... 博文 来自：[wuyt200...](#)

使用依赖注入的ASP.NET Core 2.0用户角色基础动态菜单管理

阅读数 734

目录介绍背景先决条件使用代码第1步-创建数据库创建数据库第2步-创建ASP.NETCor... 博文 来自：[寒冰屋的...](#)

Asp.Net Core-Identity 配置

阅读数 4688

在本章中，我们将安装和配置Identity框架，这只需要一点点工作。如果您使用Visual... 博文 来自：[我为女狂](#)





跟着微软学ASP.NET Core(六)-Identity获取当前登陆用户

阅读数 2333

在使用了asp.netcore自带的Identity建立了项目,使得项目只带了一个带有登陆功能的... 博文 来自: 黑猫拉面

AspNet Core 2.0 之旅---NLog日志的使用教程

阅读数 3397

1、安装NLog, 搜索 NLog.Web.AspNetCore, 并将它安装在web项目中。2、配置... 博文 来自: huanghu...

asp.net core 2.0 高清中文版+完整书签

11-07

ASP.NET Core 是一个跨平台的高性能开源框架, 用于生成基于云且连接 Internet 的新式应用程序。使用 ASP.NET Core, 您可以: 建置 We...

下载

asp.net core 2.0 Web简单使用: 四、用户列表、添加、删除

阅读数 1032

继续, 之前做的是用户注册登录。但是没有管理用户的地方, 简单做一个列表, 添加... 博文 来自: wuyt200...

asp.net core 2.0 Web简单使用: 一、用户注册登录

阅读数 3815

不在软件公司要写软件好难啊, 没时间啊。所以大概会很慢慢很慢才能把我想要的做出... 博文 来自: wuyt200...

使用依赖注入的ASP.NET Core 2.0用户角色基础动态菜单..._CSDN博客

ASP.NET Core——身份验证UI安装 - 寒冰屋的专栏 - CSDN博客

ASP.NET Core——身份验证UI安装2018年12月13日 22:04:43 寒冰屋 阅读数:2...AspNet Core 2.0 多角色权限...

ASP.NetCore2.0入门.pdf

02-11

ASP.NetCore2.0入门.pdf ASP.NetCore2.0入门.pdf ASP.NetCore2.0入门.pdf ASP.NetCore2.0入门.pdf ASP.NetCore2.0入门

下载

Angular 7和ASP.NET Core 2.1入门 - 寒冰屋的专栏 - CSDN博客

要求在ASP.NET Core 2.2中确认电子邮件——第1部分 - ..._CSDN博客

来自: 寒冰屋的专栏 对ASP.NET Core MVC 2开发web应用程序的一些看法 07-...AspNet Core 2.0身份和角色管...

AspNet Core 2.0 项目实战

阅读数 94

https://www.cnblogs.com/oorz/p/9052498.html

博文 来自: qq_4027...

ASP.NET Core 2.0身份验证和授权系统揭秘 - 刘联其的个..._CSDN博客

要求在ASP.NET Core 2.2中确认电子邮件——第2部分 - ..._CSDN博客

来自: 寒冰屋的专栏 下载 Pro Entity Framework Core 2 for ASP.NET Core...AspNet Core 2.0身份和角色管理...

asp.net core 2.0 Web简单使用: 五、添加示例

阅读数 576

添加Model添加Anime对象usingSystem;usingSystem.Collections.Generic;usingS... 博文 来自: wuyt200...

AspNet Core 2.0 之旅---在window 上 部署WEB应用程序

阅读数 1417

这两天夜晚没事儿的时候, 就把自己做的Web应用程序发布到自己的电脑上, 来记录... 博文 来自: huanghu...

逐步创建ASP.NET Core 2.2应用程序 - 寒冰屋的专栏 - CSDN博客

博文 来自: 寒冰屋的专栏 asp.net MVC core 2.2 +core EF 中文PDF教程 ...AspNet Core 2.0身份和角色管理入...

AspNet Core WebApi 身份验证、注册、用户管理 - Vold..._CSDN博客

Learning ASP.NET Core 2.0

02-10

包含了ASP.NET Core 2.0从开发到Devops所有相关要点的书

下载



香港研究生费用一览表



Asp.net Core 2.1 用户、角色创建 - georgeyp01的博客 - CSDN博客

Asp.net Core是个好东西,以前做个用户、角色创建和管理还得自己写数据库,现在...博文 来自: 寒冰屋的专栏 ASP....

ASP.NetCore2.0入门 英文版

ASP.NET Core 2.0 英文版 新手入门 官方文档 2017-10-31 09:11:25

将asp.net core2.0项目部署在IIS上运行

前言: 与ASP.NET时代不同, ASP.NETCore不再是由IIS工作进程 (w3wp.exe) 托管... 博文 来自: hailang2l...

新消息, asp.net core 2.0不再支持.net framework

昨天asp.net core 2.0 preview刚出来, 前一段时间github asp.net区就议论纷纷了, 以后... 论坛

asp.net core 2.0 Cookie 使用

直接上代码1、配置startup.cs中publicvoidConfigureServices(IServiceCollectionse... 博文 来自: 秦秋雨的...

asp.net core 2.0 入门攻略 第一章 环境部署

一、首先大家先部署环境安装因为大家当前是研究学习用途, 所以建议大家使用VS社... 博文 来自: 赫的BLOG

人工智能3.0, 多模块自由选择, 领取学习图谱, 进军AI领域!

程序员转型/入门人工智能领域需具备什么条件?

ASP.NET Core Identity 框架 - ASP.NET Core 基础教程 - 简单...

原文:ASP.NETCoreIdentity框架-ASP.NETCore基础教程-简单教程, 简单编程ASP.N... 博文 来自: weixin_3...

ASP.net core 2.0

Introduction to ASP .NET Core 8/5/2017 • 2 min to read • Edit Online By Daniel Roth, Rick Anderson,

怎么在asp.net core2.0中使用多组cookies

怎么在asp.net core2.0中使用多组cookies var user = new ClaimsPrincipal(new Claims... 论坛

微信小程序上传下载文件服务器端代码.ASPNETCORE2.0

微信小游戏 小程序 上传/下载 文件的服务器代码示例 asp.net core 2.0 对应客户端接口 wx.uploadFile wx.downloadFile

win10下ASP.NET Core 2.0部署环境搭建

此文用于记录在win10环境下, 新建的Asp.netCore2.0Web应用项目如何运行在IIS上... 博文 来自: rwcat的...

Python应用的六大发展方向! 学完python你最想做什么技术岗?

零基础python学习方法, 快来挑战!

ASP.NET Core 实战: 将 .NET Core 2.0 项目升级到 .NET Core ...

一、前言 最近一两个星期, 加班, 然后回去后弄自己的博客, 把自己的电脑从Wi... 博文 来自: dotNET跨...

AspnetCore 2.0 自动API文档生成组件, 支持protobuffer

关于API文档自动生成, 用于对APP端的开发帮助文档生成, 默认ProtoBuffer传输格... 博文 来自: dotNET跨...

asp.net core2.1 部署centos7/linux系统 -- 安装部署 (一)

1、安装dotnetsdk (添加产品密钥与yum源) 添加yum源: sudorpm-Uvhhttps://p... 博文 来自: hailang2l...

ASP.NET Core 2.0系列学习笔记-NLog日志配置文件

一、新建ASP.NETCore2.0MVC项目, 使用NuGet在浏览中搜索: NLog.Web.AspNe... 博文 来自: ChaiTSim...

【ASP.NET CORE 2.0】在CentOS上部署ASP.NET CORE 2.0程...

突然来的需求, 让以前ASP.NET写的系统要部署到Linux上, 所以研究了一下。1.安装... 博文 来自: 花海的专栏

人工智能怎么学? 对于转型的程序员有什么要求?

从国内的招聘网站看不得不说AI的岗位及薪资较优势, 但是程序员转型有什么要求?

2

10-31
下载

02-27
下载

10-15
下载

ASP.NET Core 2.0 依赖注入

阅读数 2886

问题如何使用 ASP.NETCore 服务容器进行依赖注入？答案创建一个服务publicinterf... 博文 来自: dotNET跨...

netCore2.0 Api 跨域(Cors)

阅读数 7135

1、在使用netCore2.0使用WebApi的过程中涉及到了跨域处理。在Microsoft.AspNe... 博文 来自: 越一步之距

将传统 ASP.NET 应用迁移到 .NET Core

阅读数 679

点击蓝字关注我现在越来越多的人在谈论.NETCore。诚然，.NETCore是未来,但是.NE... 博文 来自: dotNET跨...

Jasig cas 单点登录系统Server&Java Client配置

阅读数 6012

原文地址: http://blog.csdn.net/puma_dong/article/details/11564309版权声明: ... 博文 来自: 竹林幽深

跟着老桂学ASP.NET Core 2.0

阅读数 1611

.netcore作为微软开发技术中跨平台的利器，2.0的发布已经有一段时间了，asp.netc... 博文 来自: dotNET跨...

重磅！6月份PYPL编程语言排行榜Python再次成为第一名，凭什么？

看完Python的就业前景分析，这么火是有原因的！

《ASP.NET Core跨平台开发从入门到实战》及《Learning ASP.NET Core 2.0》

04-28

《ASP.NET Core跨平台开发从入门到实战》及《Learning ASP.NET Core 2.0》

下载

asp.net identity（微软首推的身份验证）2.0分析-基于vs2015默...

阅读数 2893

因为总是感觉，asp.netidentity用起来不太舒服，比如代码的扩展性，以及维护以后... 博文 来自: 赫的BLOG

Asp.Net Core WebApi 身份验证、注册、用户管理

阅读数 2570

Asp.NetCoreWebApi身份验证、注册、用户管理用户服务用户实体、用户模型及数... 博文 来自: Voldemo...

.net core 权限管理实现方案之一

11-22

该示例代码是针对asp.net core mvc权限管理的一种实现方式，不一定是最好的，但是能满足权限的有效控制

下载

怎么实现使用手机号、邮箱、用户名登录

阅读数 5493

实现使用手机号或是邮箱或是用户名登录简单逻辑

博文 来自: 亦念戈涯

Asp.net Core 2.1 用户、角色创建

阅读数 116

刚刚学习，有些地方表达可能不对，自己理解就好。这就是个【学习笔记】。回头可... 博文 来自: georgey...

ASP.NET CORE2.0高级编程.pdf

12-01

介绍 入门 创建 Web 应用 创建 Web API 教程 创建 Razor 页面 Web 应用 Razor 页面入门 添加模型 已搭建基架的 Razor 页面 SQL Server Lo...

下载

一步一步带你做WebApi迁移ASP.NET Core2.0

阅读数 5948

随着ASP.NETCore2.0发布之后，原先运行在WindowsIIS中的ASP.NETWebApi站点... 博文 来自: 心想事成

将 ASP.NET Core 2.0 项目升级至 ASP.NET Core 2.1.3X

阅读数 672

阅读本文大概需要3.3分钟。在上一篇文章《ASP.NetCore运行错误HttpError502.5解... 博文 来自: dotNET跨...

什么是ASP.NET Core 2.0?

阅读数 1118

什么是ASP.NETCore2.0? ==LearningASP.NETCore2.0ASP.NET的第一个预览版本... 博文 来自: leiyaio_...

Programming ASP.NET Core pdf版 2018最新发布

05-17

asp.net 开发牛人 2018最新版 已转成pdf 全网首发 Published 5/9/2018 1st Edition 416 pages Book 978-1-5093-0441-7 eBo

下载

Asp.Net Core实战(干货)

阅读数 4449


序言使用.NETCore，团队可以更容易专注的在.netcore上工作。比如核心类库（如Sy... 博文 来自: qq_4256...


asp.net core 2.2 pdf


01-16


A S P .N E T C o r e 是一个跨平台的高性能开源框架，用于生成基于云且连接 I n t e r n e t 的新式应用程序。使用 A S P .N E T C o r e ，您可以： ...


下载


2
























ASPNETCORE 缓存篇

阅读数 2995

一、缓存的作用怎样理解缓存： 其实所有的程序，架构，优化，线程...等技术手段... 博文 来自： shunqixi...

ASP.NET Core 2.1 : 十四.静态文件与访问授权、防盗链

阅读数 510

ASP.NETCore2.1:十四.静态文件与访问授权、防盗链我的网站的图片不想被公开浏览... 博文 来自： Orbita_w...

ASP.NET Core 和 EF Core 系列教程——入门

阅读数 2万+

ASP.NETCore和EntityFrameworkCore系列教程——入门(1/10)作者：TomDykstra... 博文 来自： 若即的专栏

python图片处理类之~PIL.Image模块(ios android icon图标自...

阅读数 13万+

1.从pyCharm提示下载PIL包nn http://www.pythonware.com/products/pil/nn nn... 博文 来自： 专注于co...

jquery/js实现一个网页同时调用多个倒计时(最新的)

阅读数 58万+

jquery/js实现一个网页同时调用多个倒计时(最新的)nn最近需要网页添加多个倒计时. ... 博文 来自： Websites

在linux上安装redis

阅读数 2904

环境ncentos6.5 x86 32位nredis3.2.8nredis简介n Redis是一个Key-Value 存... 博文 来自： 御前两把...

Android Multimedia框架总结 (二十三) MediaCodec补充及M...

阅读数 1万+

前言：前面几章都是分析MediaCodec相关源码，有收到提问，说MediaCodec到底... 博文 来自： 何俊林

linux上安装Docker(非常简单的安装方法)

阅读数 31万+

最近比较有空，大四出来实习几个月了，作为实习狗的我，被叫去研究Docker了，汗... 博文 来自： 我走小路...

农村女孩高中才接触电脑，却成为程序媛的心路历程

阅读数 2586

突发奇想：n 今天坐在工位上，玩着电脑，突然回想起自己刚开始接触计算机的画面... 博文 来自： 赵亚兰的...

运营管理视频教程 机器学习教程 Objective-C培训 交互设计视频教程 颜色模型

mysql关联查询两次本表 native底部 react extjs glyph 图标 云计算2.0培训 2.0学习 javaafx



寒冰屋

关注

原创 49 粉丝 68 喜欢 93 评论 69

等级： 博客 5 访问： 25万+

积分： 4800 排名： 1万+

勋章： 恒 5

腾讯云

学生服务器套餐
10元/月

• 1核2G • 1M宽带
• 50GB存储

立即购买

最新文章

实时 Git，在版本控制之前控制源码

Redis 6 将采用全新协议 RESP3，以提供客户端缓存功能



2



WPS Office 2019 For Linux 8722 发布，引入pdf组件

腾讯开源软件镜像站上线

Visual Studio Code 1.36 发布，结束对Linux 32 位系统的支持

个人分类

CSharp.NET	154篇
架构及框架	86篇
web前端	57篇
数据库	38篇
UML	2篇

展开

归档

2019年7月	16篇
2019年6月	58篇
2019年5月	39篇
2019年4月	45篇
2019年3月	42篇

展开

热门文章

Windows 全新终端 Windows Terminal
阅读数 39493

联想拯救者Y7000系统安装之路（Win10系统）
阅读数 18968

未能加载文件或程序集
"Newtonsoft.Json, Version=4.5.0.0"[已
阅读数 15839

Ubuntu 16 安装pycharm专业破解版方法
阅读数 15657

PyCharm专业版的安装与破解
阅读数 15096

最新评论


Windows 全新终端 Wind...
DILIGENT203: 我就想知道，复制粘贴快捷键是什么

Windows 全新终端 Wind...
LLJJYY001: 标题党，没干活


Windows 全新终端 Wind...
Asgard_Hu: 标题党，没干活


.NET Core中Quartz....
mzl87: [reply]ynat123456[reply] https://www.quartz-scheduler.net/documentation/qu ...


.NET Core中Quartz....
ynat123456: 请问一下，如果有多个job的话，好像只能执行后面那个，这个问题改怎么处理呢？



2
























买香港云服务器就选亿速云

低延迟免备案云服务器，高
直连稳定独享大带宽香港服
速度快稳定有保障



CSDN学院



CSDN企业招聘

👤 QQ客服

✉ kefu@csdn.net

🗣 客服论坛

☎ 400-660-0108

工作时间 8:30-22:00

关于我们 招聘 广告服务 网站地图

🔍 百度提供站内搜索 京ICP备19004658号

京公网安备11010502030143

©1999-2019 北京创新乐知网络技术有限公司

网络110报警服务 经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉



2

