# How to achieve a basic authorization in ASP.NET Core

## Introduction

ASP.NET Core is the new cross platform framework for web, this sample will show how to achieve a basic authorization in ASP.NET Core.

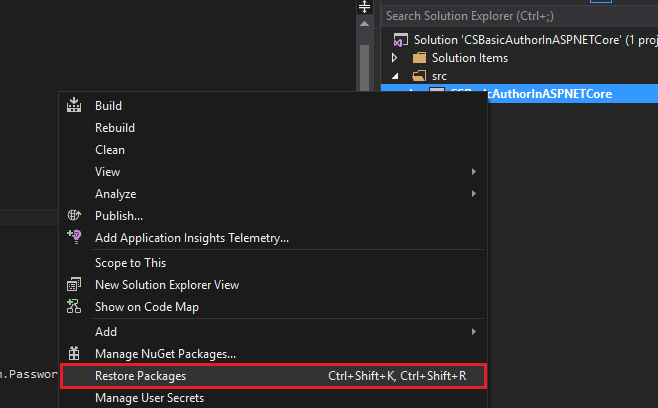
In this sample, you can login to the web server, and visit the authorized page.

## Sample prerequisites

* .NET Core 1.0 or later version(s). [[Visual Studio 2015 Home page installer](https://www.visualstudio.com/en-us/visual-studio-homepage-vs.aspx)]
* Microsoft Visual Studio 2015 update3 or above. [[.NET Core + Visual Studio tooling](http://go.microsoft.com/fwlink/?LinkID=798306)]

## Building the sample

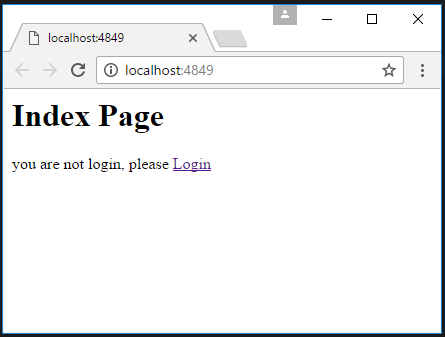
* Open the sample solution “**CSBasicAuthorInASPNETCore.sln**” in folder “/**CSBasicAuthorInASPNETCore**” using Visual Studio.
* Right click on project “**CSBasicAuthorInASPNETCore**” and select Restore packages.



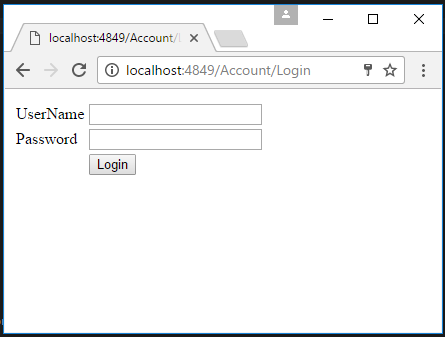
* Press **F6 Key** or select **Build -> Build Solution** from the menu to build the sample.

## Running the sample

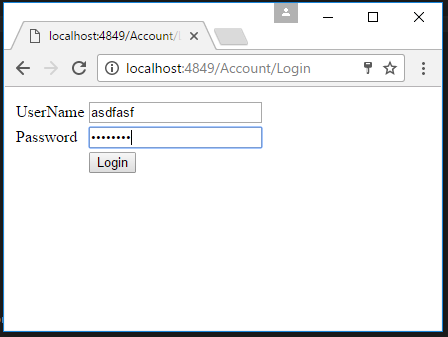
* Open the Sample solution using Visual Studio, then press **F5 Key** or select **Debug -> Start Debugging** from the menu.
* When the web application is running, you can see the page in browser.



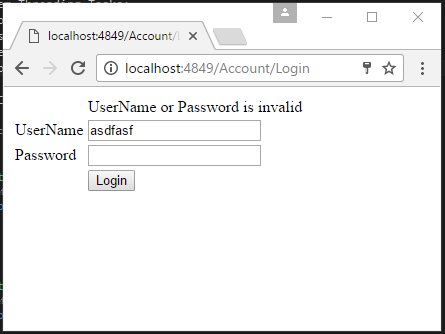
* Click Login



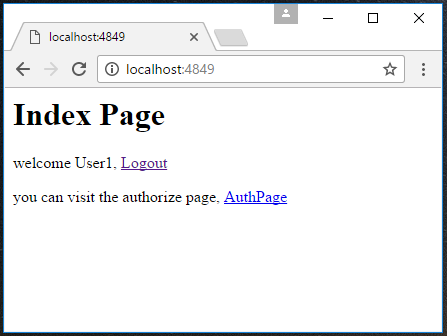
* Type an invalid user name or password, and click **Login**



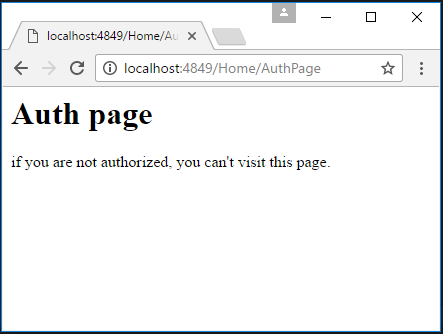
* Then server will retrun an error message on page.



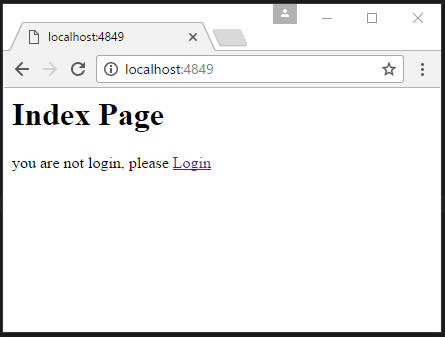
* Now, type user name as “**User1**” and password as “**112233**” and click **login.** You can see login successed, and redirect to home page.



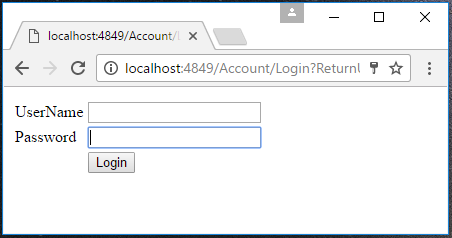
* Click Link “**AuthPage**”, you can visit the authorize page.



* Go back to previous page, and click **Logout**, you can see you have been logout.



* Now go to the authorize page, <http://localhost:4849/home/AuthPage> , your will had been redirect to login page.



## Using the code

**Startup.cs**

public void ConfigureServices(IServiceCollection services)

{

services.AddMvc();

services.AddAuthorization();

}

public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory loggerFactory)

{

app.UseCookieAuthentication(new CookieAuthenticationOptions

{

AuthenticationScheme = "Cookie",

LoginPath = new PathString("/Account/Login"),

AccessDeniedPath = new PathString("/Account/Forbidden"),

AutomaticAuthenticate = true,

AutomaticChallenge = true

});

app.UseMvc(routes =>

{

routes.MapRoute(

name: "default",

template: "{controller=Home}/{action=Index}/{id?}");

});

}

**HomeController.cs**

[HttpGet]

[AllowAnonymous]

public IActionResult Index()

{

return View();

}

[Authorize]

[HttpGet]

public IActionResult AuthPage()

{

return View();

}

**AccountController.cs**

[HttpGet]

public IActionResult Login()

{

return View();

}

[HttpPost]

public async Task<IActionResult> Login(User userFromFore)

{

var userFromStorage = TestUserStorage.UserList

.FirstOrDefault(m => m.UserName == userFromFore.UserName && m.Password == userFromFore.Password);

if (userFromStorage != null)

{

//you can add all of ClaimTypes in this collection

var claims = new List<Claim>()

{

new Claim(ClaimTypes.Name,userFromStorage.UserName)

//,new Claim(ClaimTypes.Email,"emailaccount@microsoft.com")

};

//init the identity instances

var userPrincipal = new ClaimsPrincipal(new ClaimsIdentity(claims, "SuperSecureLogin"));

//signin

await HttpContext.Authentication.SignInAsync("Cookie", userPrincipal, new AuthenticationProperties

{

ExpiresUtc = DateTime.UtcNow.AddMinutes(20),

IsPersistent = false,

AllowRefresh = false

});

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

}

else

{

ViewBag.ErrMsg = "UserName or Password is invalid";

return View();

}

}

public async Task<IActionResult> Logout()

{

await HttpContext.Authentication.SignOutAsync("Cookie");

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

}

## More information

ASP.NET Core Authorization

<https://docs.asp.net/en/latest/security/authorization/>

A workshop for moving through the various new pieces in ASP.NET v5 Authorization

<https://github.com/blowdart/aspnetauthorizationworkshop>