

 C# Corner

LEARN: How to Become a Software Architect


 Introduction to ASP.Net Identity


 become a member


Login


Post ▾


Ask Question


 Abhishek Arora

 Updated date, Dec 21, 2014



 45.5k

 8

 9



[Download Free .NET & JAVA Files API](#)

[Try Free File Format APIs for Word/Excel/PDF](#)

ASP.NET Identity is the membership system for authentication and authorization of the users by building an ASP.NET application. The ASP.NET Identity is a fresh look at what the membership system should be when you are building modern applications for the web, phone or tablet.

ASP.NET Identity allows you to add customized login/logout functionality and customized profile features that make it easy to customize the data about the logged-in user. Today there is a much broader array of data storage (which is increasing very quickly) options for web application and most of the developers want to enable their websites to use the social identity providers for Authentication and Authorization. Now what do the two terms Authentication and Authorization mean?

Authentication is used by the server to determine who is accessing their information or website. In authentication, the user or customer must prove their identity on a web server by log-in using email and word or using various social providers.

Authorization is a process by which a server determines if the client has permission to use a resource or access a file after the

way of logging into a website. Now the web has become more social where users are interacting using social channel providers such as Facebook, Twitter, Google Account, Microsoft Account and many more. Developers as well as users want to be able to login using these social channel identities to have a rich and dynamic experience on sites, that must enable log-in based redirection using such providers.


[become a member](#)
[Login](#)
[Post ▼](#)
[Ask Question](#)

The ASP.NET Identity can be used with all the frameworks such as ASP.NET MVC, Web Forms, Web Pages. By default the ASP.NET Identity will store the information of users in a database. Which is now also possible to store information on various storage providers based on the requirements of the application. The various data providers can be SharePoint, Azure table services and so on.

The Role based authorization also plays an important role in ASP.NET Identity. Now we can easily create roles such as "Admin", "Customer" and so on that allows us to add users to this role that also helps us to restrict the users to all the parts of the application.

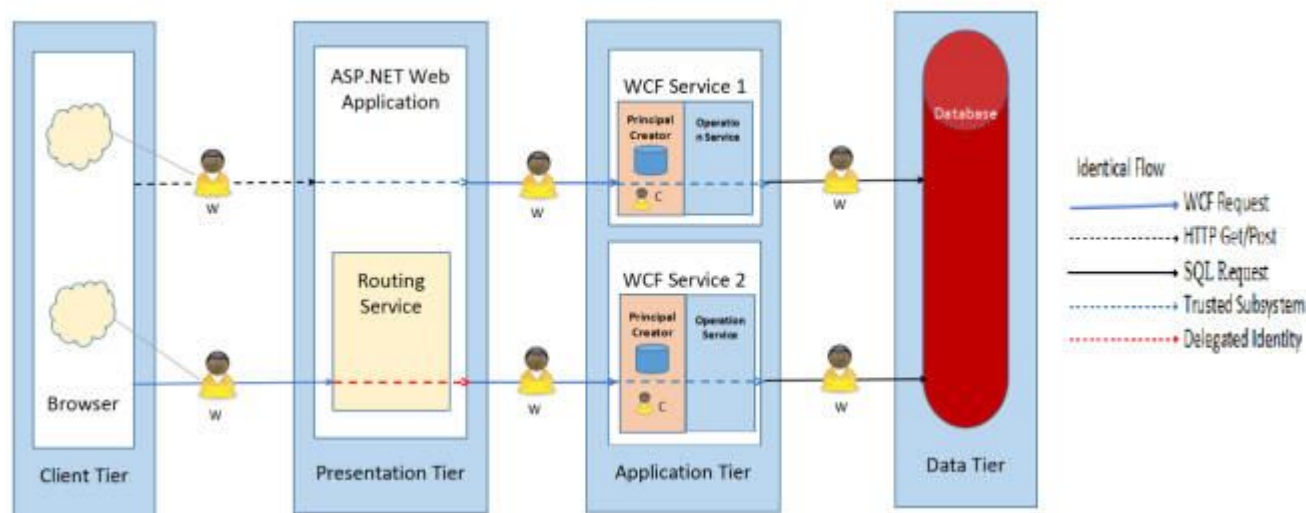


Figure 1: Flow of Identity Information in a Multi-Tier Application Using ASP.NET Identity 2.0

The ASP.NET Identifier is a game changer by bringing more security to your web application by introducing Two-Factor

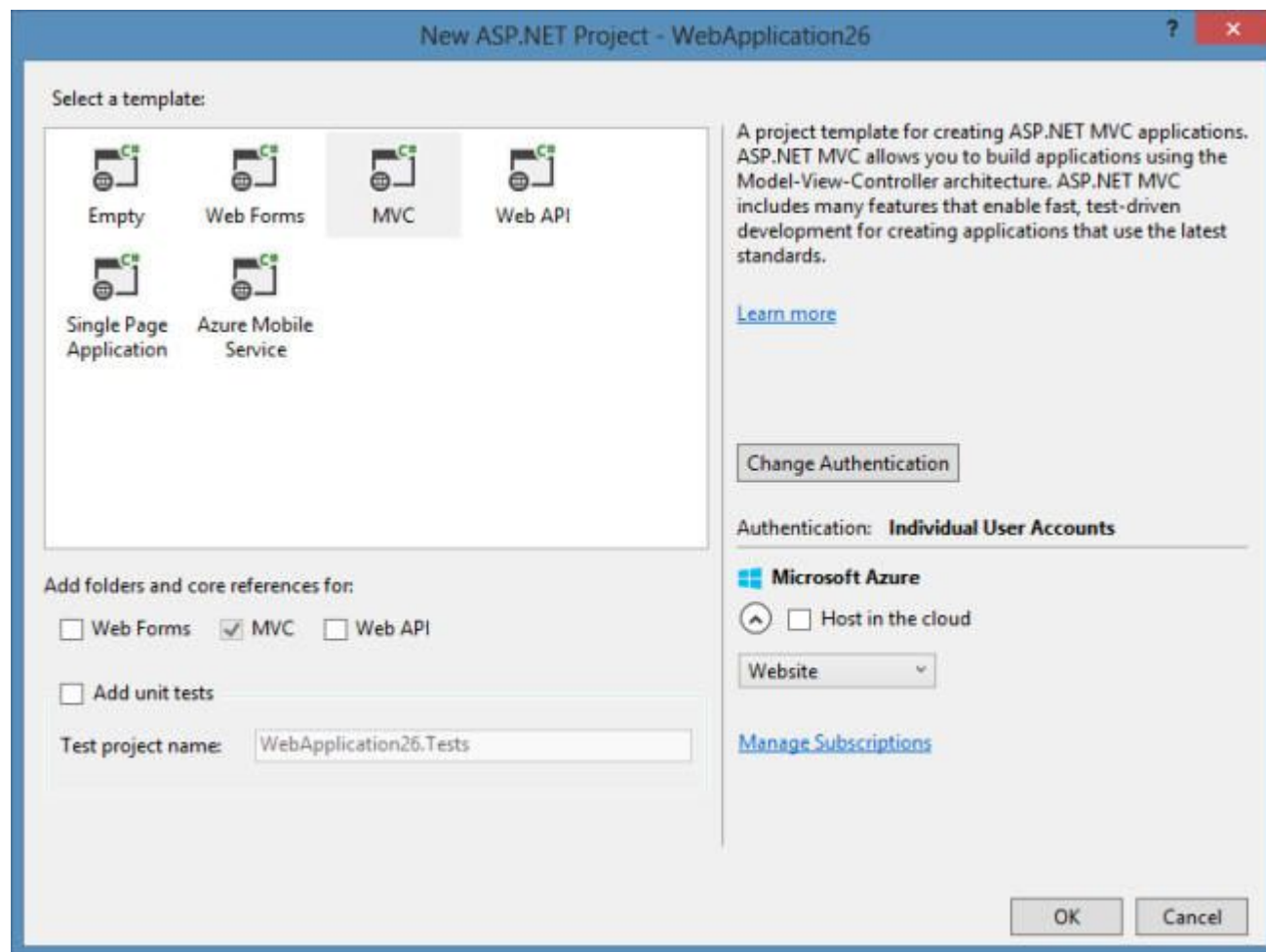


Features of ASP.NET Identity 2.0

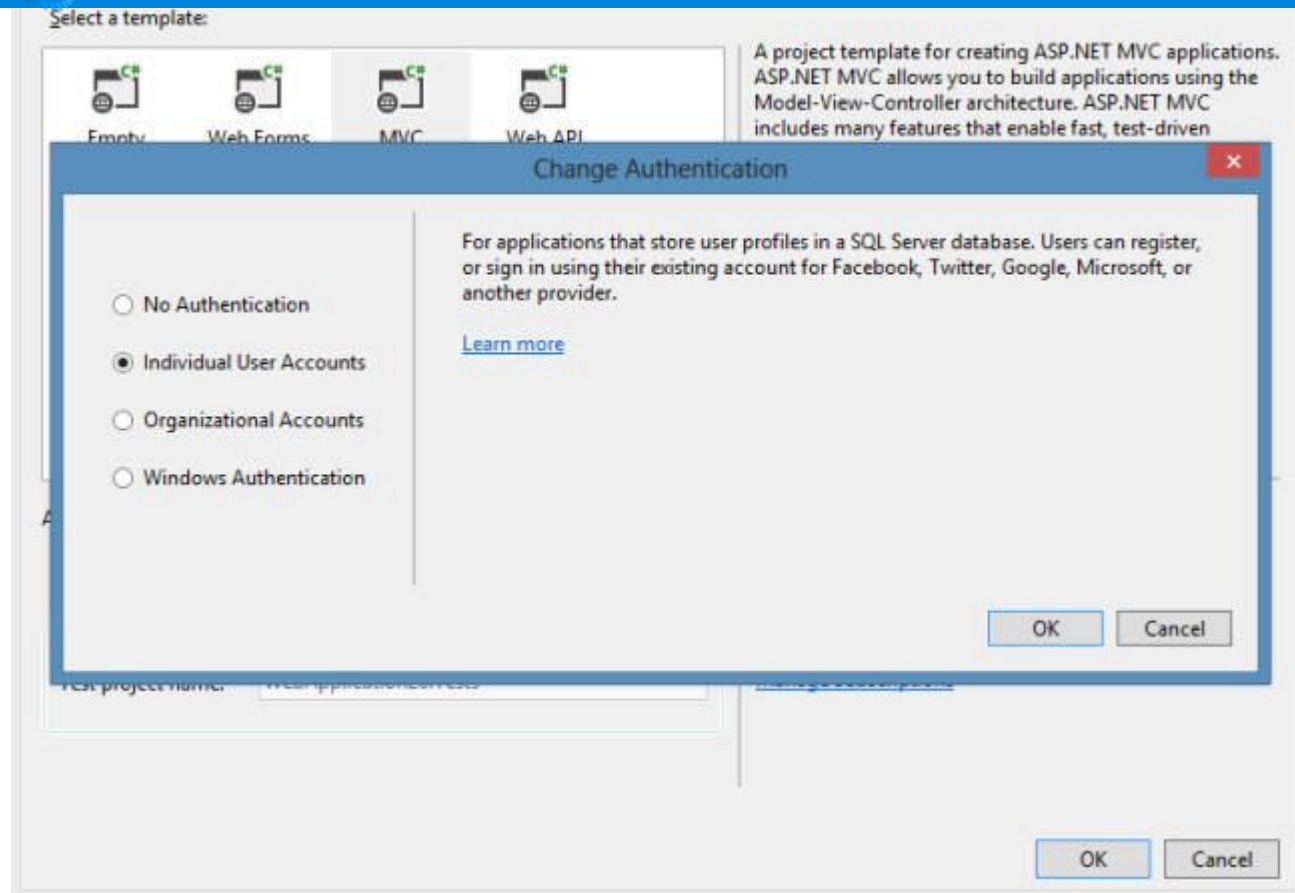
- **Two-Factor Authentication:** The Two-Factor Authentication adds one more level of security to your web application. Earlier we were using only single-factor authentication by simply entering a username and word to the web application that makes our account less secure because the hackers can hack your word and can steal your private data. The two-factor authentication needs your username and word along with a One-Time word (OTP). The One-Time word can be a hardware token or your mobile phone or can be a PIN. Whenever the user logs into his/her account they need to go through 2FA, first your username and word and another is a hardware token that generates one-time words for the second stage of the login process. Every time the word generated by the token will be unique and changes for each short duration of time. The Two-Factor Authentication makes our website much stronger compared with single-factor authentication. You can use ASP.NET Identity to secure Web Apps as well as Web APIs.
- **Account Lockout:** If the user enters the word and two factor codes incorrectly after the specific number of invalid attempts that can be configured by the developer his or her account will be locked for a specific period of time. The developers can also disable this feature.
- **Account Confirmation:** ASP.NET Identity allows us to confirm the account by confirming the email of the user. This feature is used by most of the websites to confirm their email before accessing their accounts and services. It also prevents fictitious accounts from being created.
- **word Reset:** This feature enables users to reset their word if they have forgotten their word.
- **Support IQueryable on Users and Roles:** IQueryable on UsersStore and RolesStore helps to easily get the list of Users and Roles.
- **Delete User account:** You could not delete a user using UserManager in earlier versions of ASP.NET Identity 1.0 but in ASP.NET 2.0 we can easily delete the user using UserManager.
- **Enhanced word Validator:** In earlier versions of ASP.NET Identifier the word validator was only validating the minimum



authentication system. ASP.NET is an open web interface for .NET (CLR) based library. Visual Studio 2013 will create the web application using ASP.NET Identity. The following figure will demonstrate the project template dialog of Visual Studio 2013.

[become a member](#)[Login](#)[Post ▼](#)[Ask Question](#)

Now let us create a MVC application and we will be able to see the Change Authentication button at the right hand side and by clicking on this we will be able to see the various modes of authentication as shown below.



As we can see, there are four modes of authentication named No Authentication, Individual User Accounts, Organizational Accounts and Windows Authentication. Let us see now what these terms really mean.

- **No Authentication:** This authentication mode doesn't require any user authentication for applications.
- **Individual User Accounts:** This type of application is configured to use ASP.NET Identity for user authentication that was already known as ASP.NET Membership in early days. ASP.NET Identity enables a user to register for an account either by creating an account on the website itself by creating a username and word or by signing in through social providers such



- **Windows Authentication:** This mode of authentication will support Windows Identity Foundation. IIS stands for Internet Information Services that acts as a web server for .NET application specially designed for windows Operating Systems. This authentication is intended for Intranet Applications.

The default selection of MVC application will be the "Individual User Account" that indicates that user account information will be stored in the application database which means that there will be no external based logins. By creating the MVC project with default selection we will find that the project will include AccountController and associated views for registering new users as well as for authenticating users and will also find a few references of OWIN assemblies along with an OWIN startup class. So to avoid difficulty we will be starting with an empty MVC project from startup.

Before proceeding further let's become familiarized with the parts of ASP.NET Identity. There are the following six important pieces of the ASP.NET Identity system:

- User
- Role
- User Manager
- Role Manager
- Authentication Manager
- Entity Framework DbContext

User

A user of the system is represented by the user object where a user object contains the basic authentication of a user by a user ID and word. The basic authentication is captured by IdentityUser class. The IdentityUser class can be inherited from custom class that can be used to capture profile information.

Role



User Manager

A User Manager is a class that allows you to manage users. Several tasks can be done using the User Manager class such as creating or removing a user account, changing words, or adding and removing users to a role and for this purposes ASP.NET Identity comes with the UserManager class.

Role Manager

A Role Manager is a class that allows us to manage roles. The role manager is responsible for creating or removing a role and checking whether a role exists in the system. This can be done using a role manager and for this purpose the RoleManager class is used.

Authentication Manager

All the classes stated above deal with users and roles respectively and they don't do any authentication by themselves. Signing in and signing out is the responsibility of the Authentication Manager. Similar to Forms Authentication, the local user account uses cookie-based authentication. The authentication manager is represented by the IAuthenticationManager interface.

Entity Framework DbContext

The database table schema is not rigidly fixed as in the case of the ASP.NET membership system in ASP.NET Identity. Based on the user and role objects the Entity Framework Code first approach is used to generate the table schema which means a separate column is created in the database for each piece of user profile. In the App_Data folder all the preceding tables are created by default in a separate database but we can also use our own database for storing this information by specifying the database. To accomplish this you can create a custom DbContext class that inherits from the IdentityDbContext base class.

Brought to you by: [JavaScript SDK for Bold BI dashboard and analytics embedding. Free trial](#)



C# Corner

LEARN: How to Become a Software Architect

Collapsible GridView in ASP.Net 2.0



Become a member

Login

Post ▼

Ask Question

asp.net

ASP.NET Identity 2.0

Entity Framework DbContext

Features of ASP.NET Identity 2.0

Abhishek Arora *TOP 500*

Dear readers, I am Computer Science graduate from Uttar Pradesh Technical University (UPTU), Ghaziabad

<https://www.c-sharpcorner.com/members/abhishek-arora>

217



825.3k



9



8



Type your comment here and press Enter Key (Minimum 10 characters)



good

Gowtham Rajamanickam

73 23.1k 4.7m



1



Apr 02, 2015



0

Reply



Thanks all..

Abhishek Arora

217 9.3k 825.3k



1



Dec 24, 2014



0

Reply



Nice one..

Manish Kumar Choudhary

186 10.9k 3.5m



2



Dec 22, 2014



0

Reply

[become a member](#)[Login](#)[Post ▼](#)[Ask Question](#)

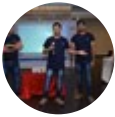
Nice..

[Humayun Kabir Mamun](#)

■ 361 ■ 5.5k ■ 71

Dec 22, 2014

2 0 Reply



good one.thanks for sharing

[Vithal Wadje](#)

■ 7 ■ 56.7k ■ 36.5m

Dec 22, 2014

2 0 Reply



Good start abhishek, Welcome to C# Corner.

[Dinesh Beniwal](#)

■ Editor ■ 67.2k ■ 7m

Dec 21, 2014

2 1 Reply



thank you sir..

[Abhishek Arora](#)

■ 217 ■ 9.3k ■ 825.3k

Dec 24, 2014

1



C# Corner

LEARN: How to Become a Software Architect

Discover if UX is Worth it for Your Career & Company - AMA

Create An Application To Get User Details In A Dialog Box In Angular With Angular Material Using Web API And SQL

[become a member](#)[Login](#)[Post ▼](#)[Ask Question](#)[View All ➔](#)

TRENDING UP

- 01 Getting Started With Microsoft Fluent UI React
- 02 What is Software-Defined Networking and Virtual Networks in Physical Networks
- 03 What are Virtualized Data Centers and VMware's SDDC Approach
- 04 What are Data Center Building Blocks and Network Virtualization Services?
- 05 What Is NSX Logical Switching And NSX Logical Routing?

06 What is Virtual Networking?


07 What are Virtual Switches and Standard Switches?












- 09 Build A Simple Todo App Using Microsoft Fluent UI React
- 10 Covid-19 Tracker Website With React, Material.UI And Chart.js

[become a member](#)[Login](#)[Post ▼](#)[Ask Question](#)[View All ➔](#)

LEARN: How to Become a Software Architect



become a member

Login

Post ▼

Ask Question

About Us Contact Us Privacy Policy Terms Media Kit Sitemap Report a Bug

C# Tutorials Common Interview Questions Stories Consultants Ideas Certifications

©2020 C# Corner. All contents are copyright of their authors.



C# Corner

LEARN: How to Become a Software Architect



Become a member

Login

Post ▼

Ask Question

FEATURED ARTICLES

Learn How to Build a Rapid POC - AMA