

ASP.NET Identity

What is ASP.NET Identity?

ASP.NET Identity is a framework for security and user management.

- Developed by Microsoft.
- ASP.NET Identity replaces the previous membership and security solutions.
- ASP.NET Identity 2.0 was released in the spring 2014
- Works with ASP.NET MVC, Web API, Web Pages, SignalR and Web Forms.
- Compatible from .NET 4.5

What is ASP.NET Identity?

ASP.NET was developed to create a robust security solution which can cope with modern demands.

It also relieves the developers from creating a new user management system for every new application.

Why ASP.NET Identity?

ASP.NET Identity is the current security library from Microsoft.

ASP.NET Identity is currently the preferred way to handle user authentication in ASP.NET applications

It comes with the latest safety features such as:

- Account confirmation
 - Both email and/or SMS confirmation
- Improved password handling
- Two-Factor Authentication

Easy use of social login providers thanks to OAuth

Allowing you to delegate the user authentication process.

Example of such providers : Google, Facebook or Twitter

Claims-based authentication

Allows the developer to create more detailed identities and control the authorization in greater detail.

Identity is integrated into the OWIN pipeline

OWIN decouples the communication between your application and the server/database

Visual Studio 2013 come with a range of templates which implement ASP.NET Identity to the following project types:

- MVC
- Web Forms
- Web API
- Single Page Application

Why ASP.NET Identity

If you are using an earlier version of Visual Studio, the templates listed in the previous slide does not contain ASP.NET Identity.

However you can easily add ASP.NET Identity via the Package Manager Console, and implement your own functionality

```
PM> Install-Package Microsoft.AspNet.Identity.Core
```

```
PM> Install-Package Microsoft.AspNet.Identity.EntityFramework
```

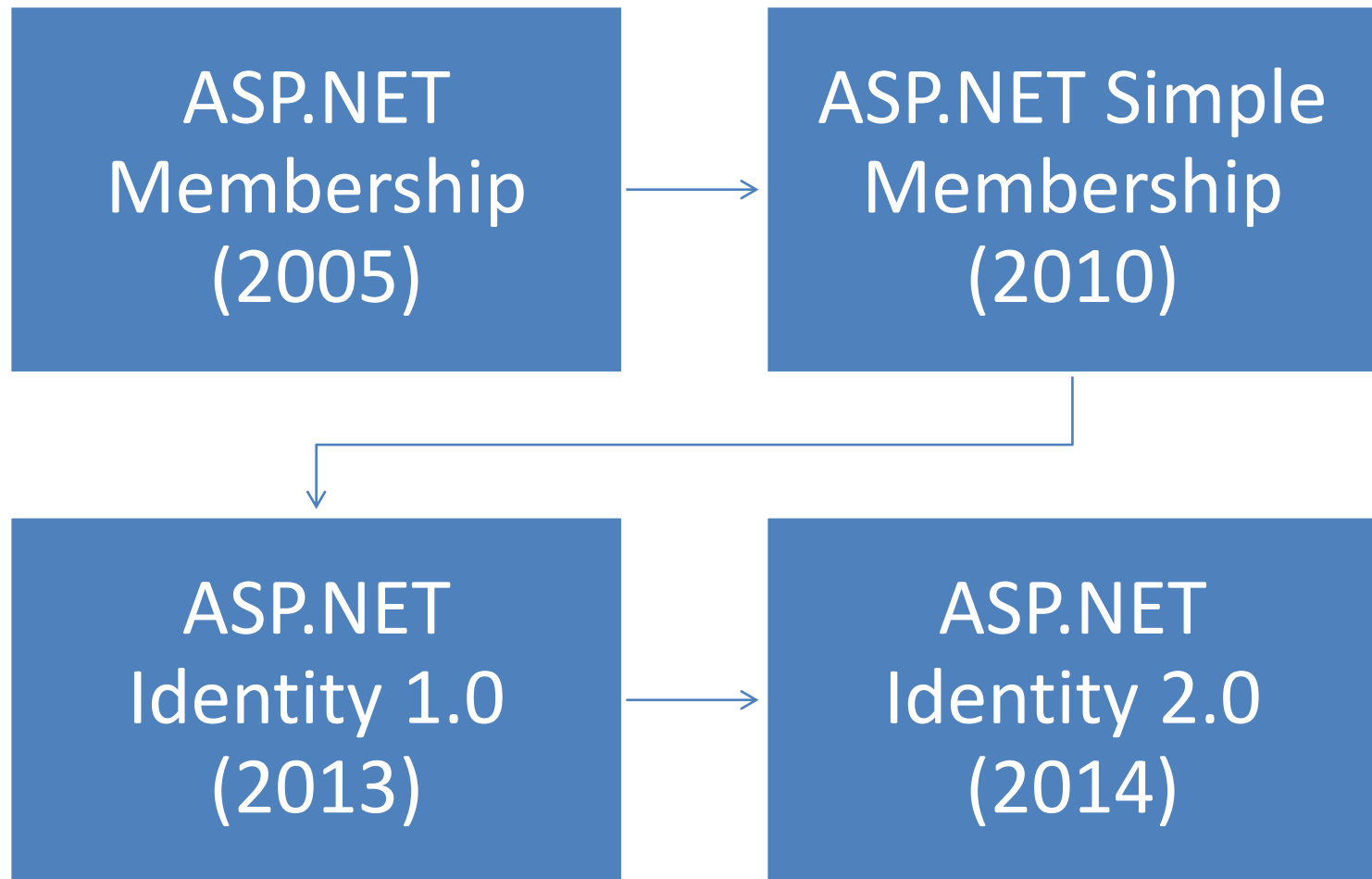
```
PM> Install-Package Microsoft.AspNet.Identity.Owin
```

When should I use ASP.NET Identity?

- When your application will handle a large number of external users
- When you don't want to handle user authentication yourself
- When you feel the need for flexibility which claims-based authorization gives you
- When you want your application to have a secure and strong safety structure

Before ASP.NET Identity?

What was before ASP.NET Identity?



What was before ASP.NET Identity?

General flaws with the previous security libraries.

- Difficulties adding custom data to the database
- No longer up to date with todays security standards
 - Not using modern password storage techniques
 - Does not support two-factor authentication
 - Does not support claims-based authentication

Think twice about using membership and simplemembership

<http://brockallen.com/2012/09/02/think-twice-about-using-membershipprovider-and-simplemembership/>

ASP.NET Membership

- Forms based authentication.
- Designed with only SQL servers in mind.
- No Owin integration.
- No social login providers.
- Difficult to customize user information
- Does not follow SRP (Single Responsibility Principle)

ASP.NET Simple Membership

- Built upon ASP.NET Membership
- Designed for ASP.NET MVC
- Supports OAuth/OpenId
- Tied to Microsoft databases (SQL)
- Does not follow SRP

ASP.NET Identity 1.0

- Released in 2013
- Simple & flexible way to store and handle user data
- Supports Roles & Claims
- Social Login Providers using OAuth
- Requires .NET 4.5

ASP.NET Identity 2.0

- Released in 2014
- Security Token Provider
- Two-Factor authentication
- Account confirmation (Email/SMS)
- Account lockout
- Password reset (Email/SMS)

Resources

- ASP.NET official site
<http://www.asp.net/identity>
- ASP.NET security forum
<http://forums.asp.net/25.aspx/1?Security>
- Brock Allen (Blog)
<http://brockallen.com/category/asp-net-security/>
- Dominick Baier (Blog) Mostly focused on Web-API
<http://leastprivilege.com/category/net-security/>
- Pro ASP.NET MVC 5 Platform

