logical ADVANTAGE

CHARLOTTE TECH TALKS

Authentication Using Tokens for AngularJS, OWIN, ASP.NET Web API & Identity

Mark A. Wilson
Senior Developer
MarkW@LogicalAdvantage.com

www.DeveloperInfra.com
@DeveloperInfra

Mark A. Wilson

www.developerinfra.com

@DeveloperInfra

Software craftsman, consultant and agile .NET/JavaScript web developer. User group leader and event planner. Loving husband, dog foster, and Disney aficionado.

Logical Advantage

www.logicaladvantage.com

Our mission is to partner with our clients to provide strategies and solutions that maximize the ROI of Enterprise Asset Intelligence and Human Capital Development.

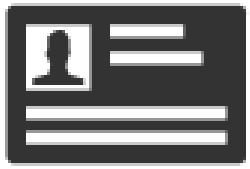
Charlotte, NC

Enterprise Developers Guild
www.developersguild.org

Spark Conference
www.sparkconf.org

Authentication Using Tokens for AngularJS, OWIN, ASP.NET Web API & Identity

Authentication



Who You Are

Authentication

Cookie-Based

Pros

- Decades-Old & Widely Used
- Default for New Projects

Cons

- Man-In-The-Middle (MITHM)
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)

Token-Based

Pros

- Stateless & Scalable Servers
- Mobile Friendly
- Pass Authentication To Other Applications
- Extra Security

Cons

- Cross-Site Scripting (XSS)

- https://scotch.io/tutorials/the-ins-and-outs-of-token-based-authentication
- http://sitr.us/2011/08/26/cookies-are-bad-for-you.html

Authentication

Cookie-Based

Pros

- Decades-Old & Widely Used
- Existing Server App Support

Cons

- Man-In-The-Middle (MITHM)
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)

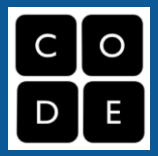
Token-Based

Pros

- Stateless & Scalable Servers
- Mobile Friendly
- Cross-Domain / Cross-Origin Resource Sharing (CORS) AJAX
- Decoupled Token Generation
- CSRF Protected
- CDN Asset Hosting
- Improved Performance
- Standard-based JWT

- https://auth0.com/blog/2014/01/07/angularjs-authentication-with-cookies-vs-token/

Token-Based Authentication



DEMO

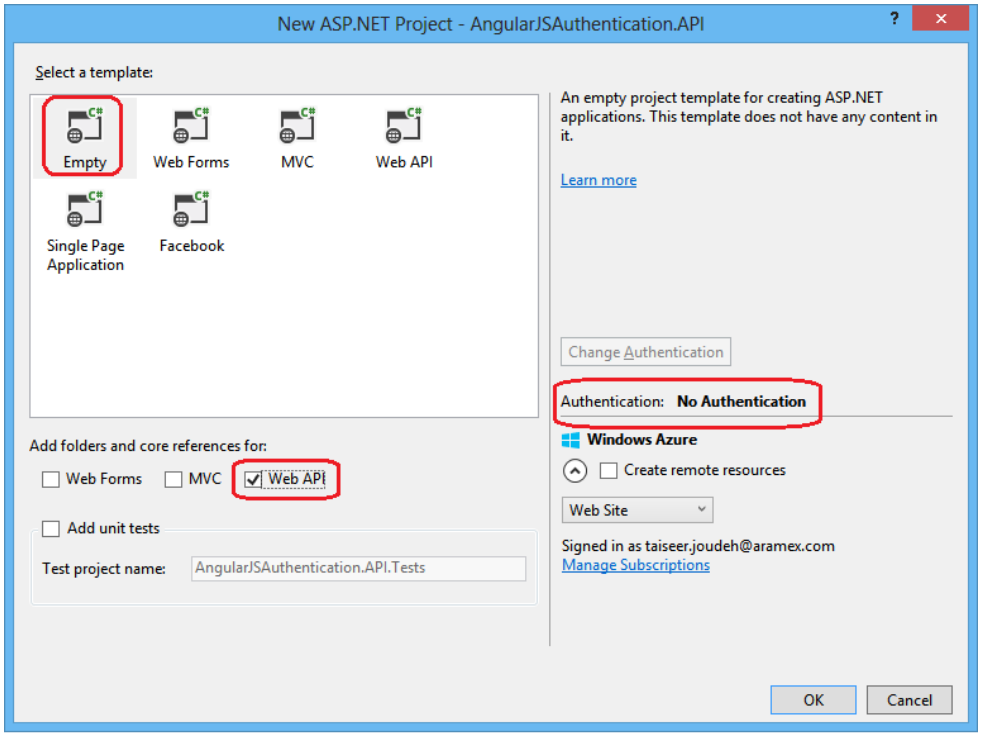
Demo

Code based on the online tutorial, "Token Based Authentication using ASP.NET Web API 2, Owin, and Identity" by Taiseer Joudeh, MVP.

- http://bitoftech.net/2014/06/01/token-based-authentication-asp-net-web-api-2-owin-asp-net-identity/

Lightweight Solutions & Projects

Less is more.



NuGet Packages

The magical number 7.

Microsoft.AspNet.WebApi

Microsoft.AspNet.WebApi.Owin

Microsoft.Owin.Host.SystemWeb

Microsoft.Owin.Security.OAuth

Microsoft.Owin.Cors

*Microsoft.AspNet.Identity.Owin

*Microsoft.AspNet.Identity.EntityFramework

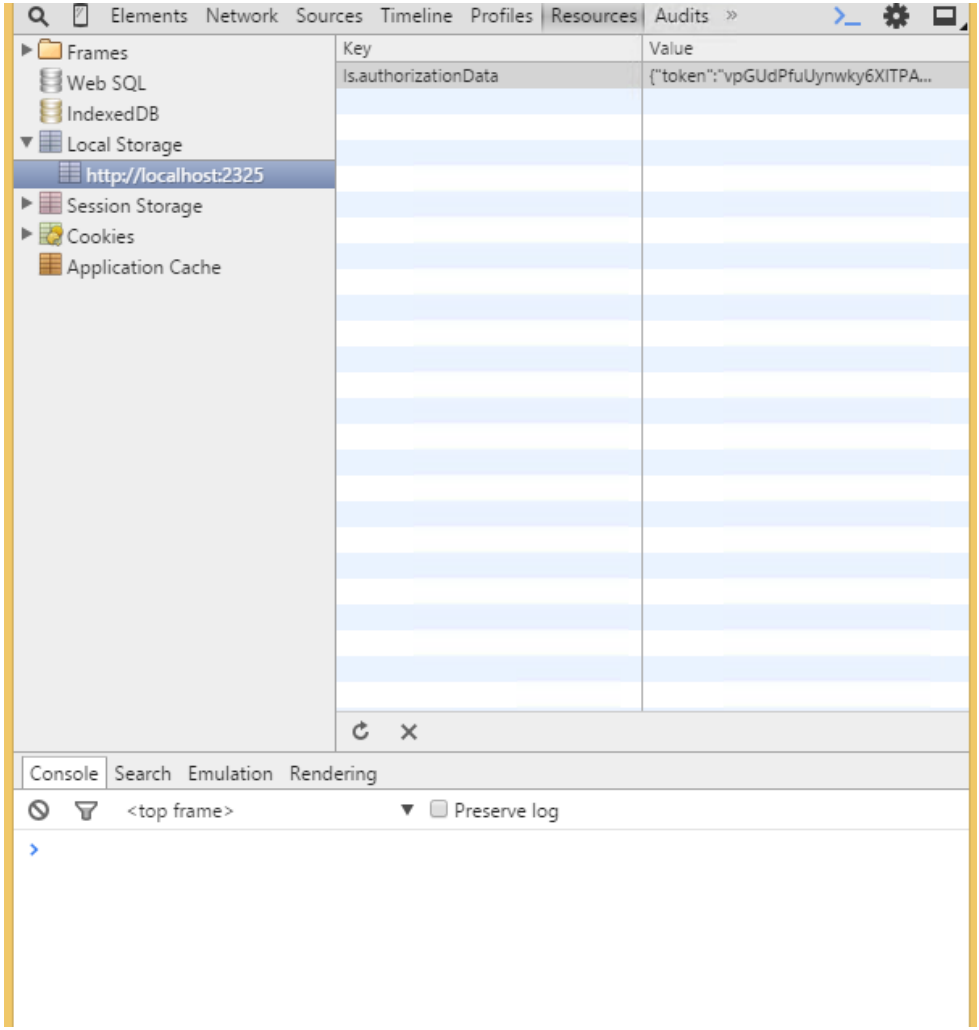
*Authentication API Project Only

Example Token

It is JSON.

Local Storage

Other options include JavaScript or a Cookie.



AngularJS Service

Move along, nothing to see here.

sService.js

```
'use strict';
app.factory('ordersService', ['$http', 'ngAuthSettings', function ($http, ngA

    var serviceBase = ngAuthSettings.apiServiceBaseUri;

    var ordersServiceFactory = {};

    var _getOrders = function () {

        return $http.get(serviceBase + 'api/orders').then(function (results)
            return results;
        });
    };

    ordersServiceFactory.getOrders = _getOrders;

    return ordersServiceFactory;

}]);
```


AngularJS Interceptor

Pre- and post-processing.

nterceptorService.j

```
'use strict';
app.factory('authInterceptorService', ['$q', '$injector','$location', 'localS

    var authInterceptorServiceFactory = {};
    var $http;

    var _request = function (config) {

        config.headers = config.headers || {};

        var authData = localStorageService.get('authorizationData');
        if (authData) {
            config.headers.Authorization = 'Bearer ' + authData.token;
        }

        return config;
    }

    var _responseError = function (rejection) {
        var deferred = $q.defer();
        if (rejection.status === 401) {
            var authService = $injector.get('authService');
            authService.refreshToken().then(function (response) {
                _retryHttpRequest(rejection.config, deferred);
            }, function () {
                authService.logOut();
                $location.path('/login');
                deferred.reject(rejection);
            });
        } else {
            deferred.reject(rejection);
        }
        return deferred.promise;
    }

    var _retryHttpRequest = function (config, deferred) {
        $http = $http || $injector.get('$http');
        $http(config).then(function (response) {
```


ASP.NET Web API Controller

You shall not pass!

ctedController.cs

```
larJSAuthentication.ResourceServer | AngularJSAuthentication.ResourceServer. | Get()
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Net;
5  using System.Net.Http;
6  using System.Security.Claims;
7  using System.Web.Http;
8
9  namespace AngularJSAuthentication.ResourceServer.Controllers
0  {
1      [Authorize]
2      [RoutePrefix("api/protected")]
           0 references | Mark A. Wilson, 134 days ago | 1 change
3      public class ProtectedController : ApiController
4      {
5          [Route("")]
               0 references | Mark A. Wilson, 134 days ago | 1 change
6          public IEnumerable<object> Get()
7          {
8              var identity = User.Identity as ClaimsIdentity;
9
0              return identity.Claims.Select(c => new
1              {
2                  Type = c.Type,
3                  Value = c.Value
4              });
5          }
6      }
7  }
```


Shared MachineKey

Do not use the same keys for dev & prod.

config

```
<?xml version="1.0" encoding="utf-8"?>

<!--
  For more information on how to configure your ASP.NET application, please v
  http://go.microsoft.com/fwlink/?LinkId=301879
  -->
<configuration>
  <configSections>
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigF

  <!-- For more information on Entity Framework configuration, visit http://g
  <appSettings />
  <connectionStrings>
    <add name="AuthContext" connectionString="Data Source=.;Initial Catalog=A
  </connectionStrings>
  <system.web>
    <compilation debug="true" targetFramework="4.5" />
    <httpRuntime targetFramework="4.5" />
    <machineKey
      decryption="AES"
      decryptionKey="88274072DD5AC1FB6CED8281B34CDC6E79DD7223243A527D46C09CF6
      validation="SHA1"
      validationKey="A970D0E3C36AA17C43C5DB225C778B3392BAED4D7089C6AAF76E3D42
  </system.web>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="Microsoft.Owin" publicKeyToken="31bf3856ad364
        <bindingRedirect oldVersion="0.0.0.0-3.0.1.0" newVersion="3.0.1.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="Microsoft.Owin.Security.OAuth" publicKeyToken
        <bindingRedirect oldVersion="0.0.0.0-3.0.1.0" newVersion="3.0.1.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="Microsoft.Owin.Security.Cookies" publicKeyTok
```


Demo

Code based on the online tutorial, "Token Based Authentication using ASP.NET Web API 2, Owin, and Identity" by Taiseer Joudeh, MVP.

- http://bitoftech.net/2014/06/01/token-based-authentication-asp-net-web-api-2-owin-asp-net-identity/

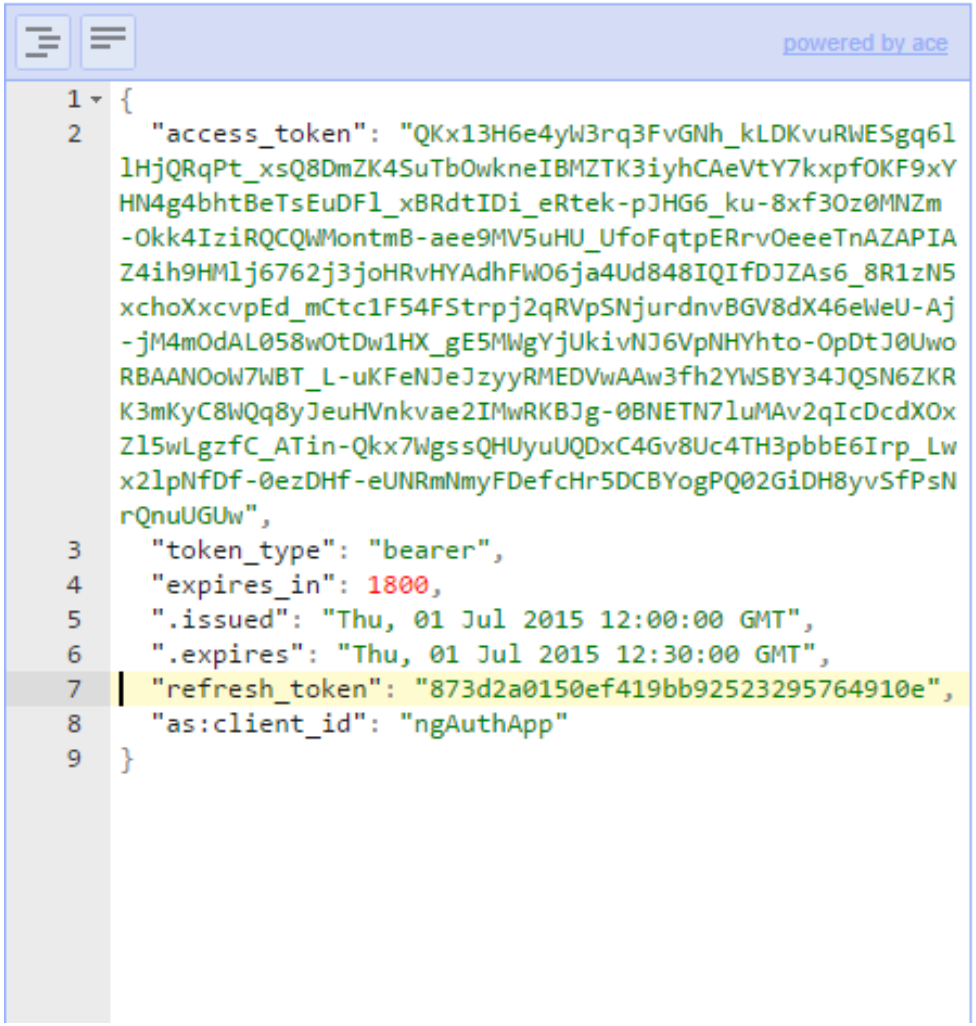
Lessons Learned

- Use HTTPS over TLS/SSL!
- Forms Authentication
- Third-Party Logins (Facebook, Twitter, etc.)
- Custom Grant Type
- Entity Framework
- [Public-facing] Production Ready?
  - Auth0.com
  - OAuth.io
  - Thinktecture IdentityServer
  - DotNetOpenAuth
  - Spring Social for .NET
  - etc.

OAuth Refresh Tokens

Example Token

It is a GUID.



Lessons Learned

Pros

- Updating access token content.
- Maintaining authenticated users list.
- Revoking access from authenticated users.
- Prompting to login multiple times not necessary.

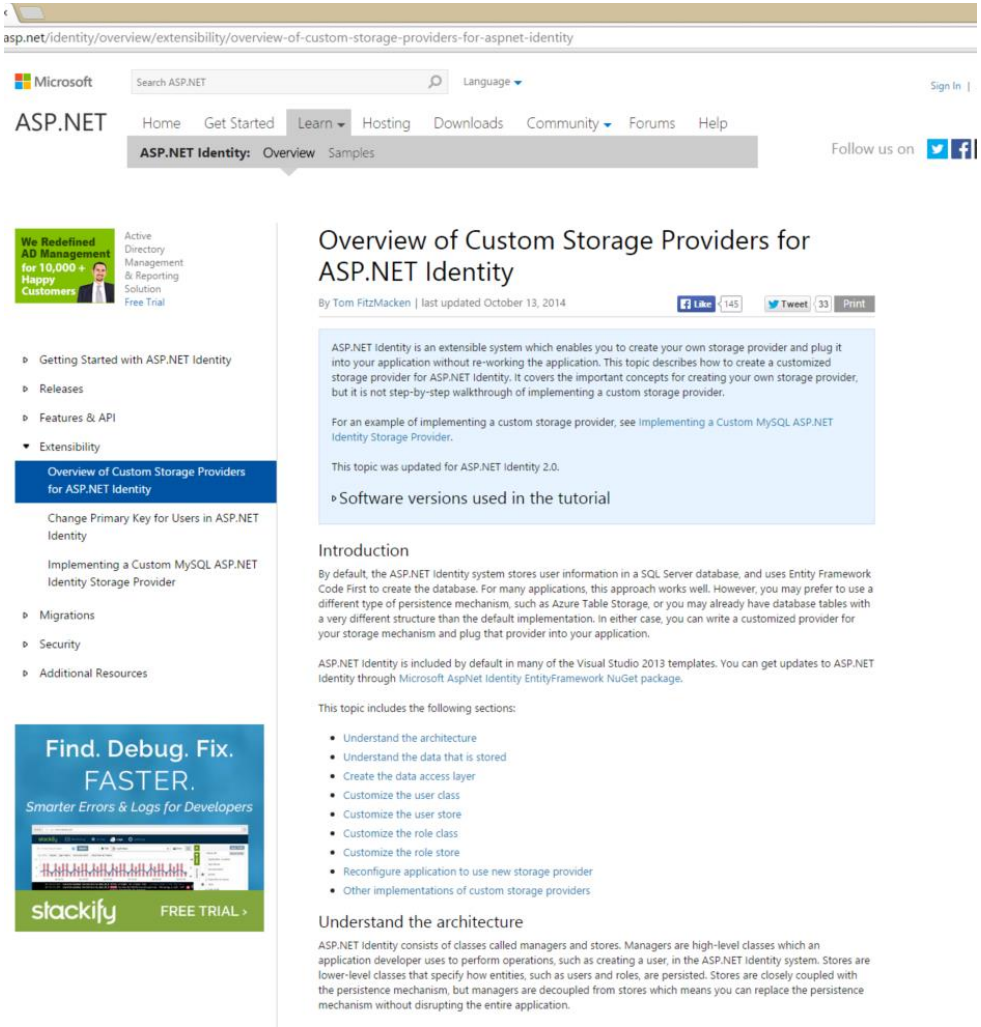
Cons

- Adding a lot of complexity.
- Using third-party logins?

Custom ASP.NET Identity Provider

Custom Provider

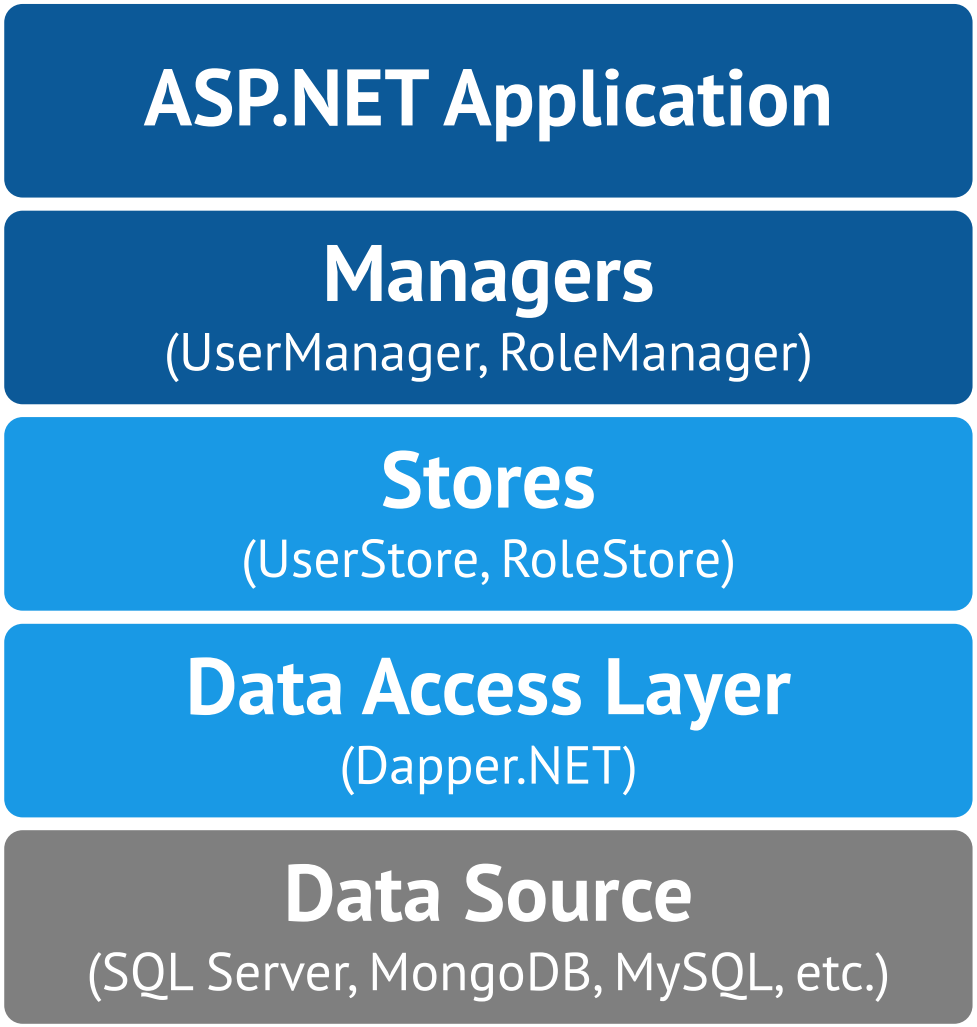
For when you want something other than Entity Framework or a GUID primary key.



- http://www.asp.net/identity/overview/extensibility/overview-of-custom-storage-providers-for-aspnet-identity

Architecture

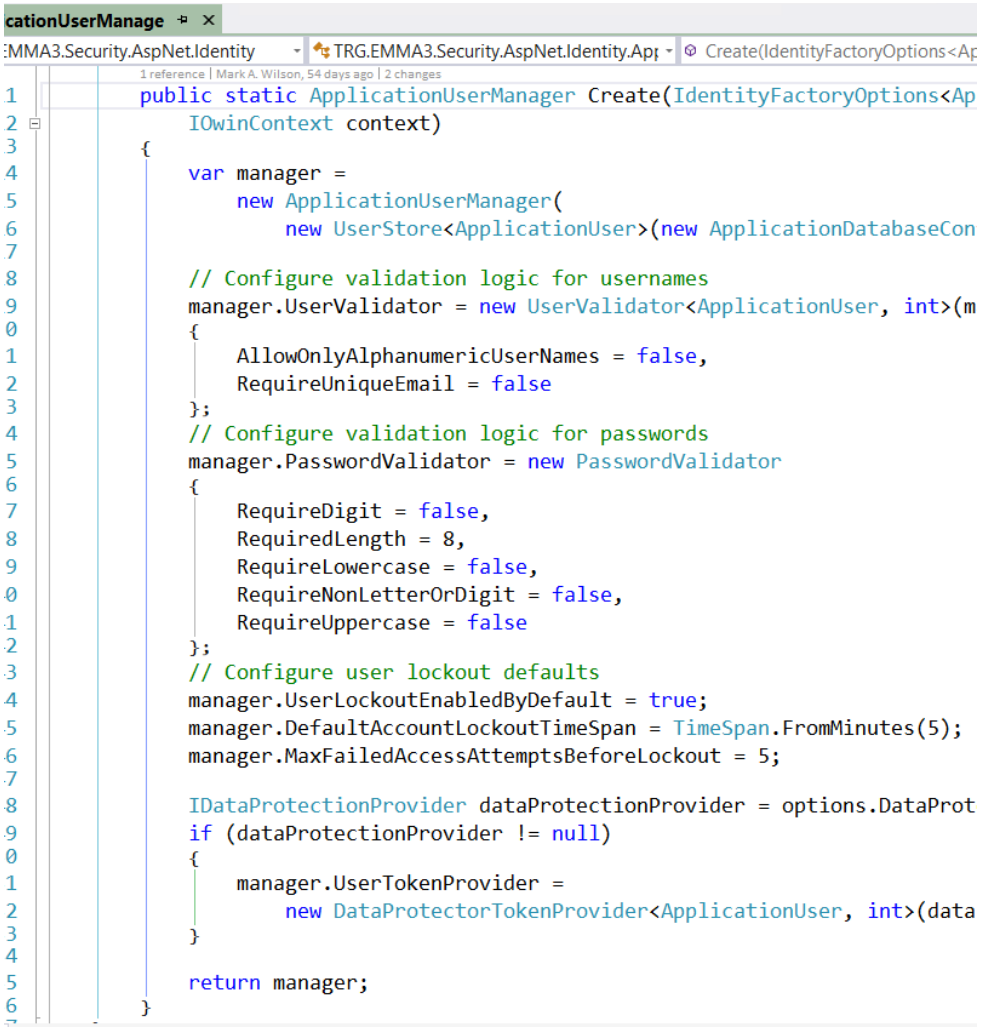
Your application interacts with the managers, and stores interact with the data access layer.



- http://www.asp.net/identity/overview/extensibility/overview-of-custom-storage-providers-for-aspnet-identity

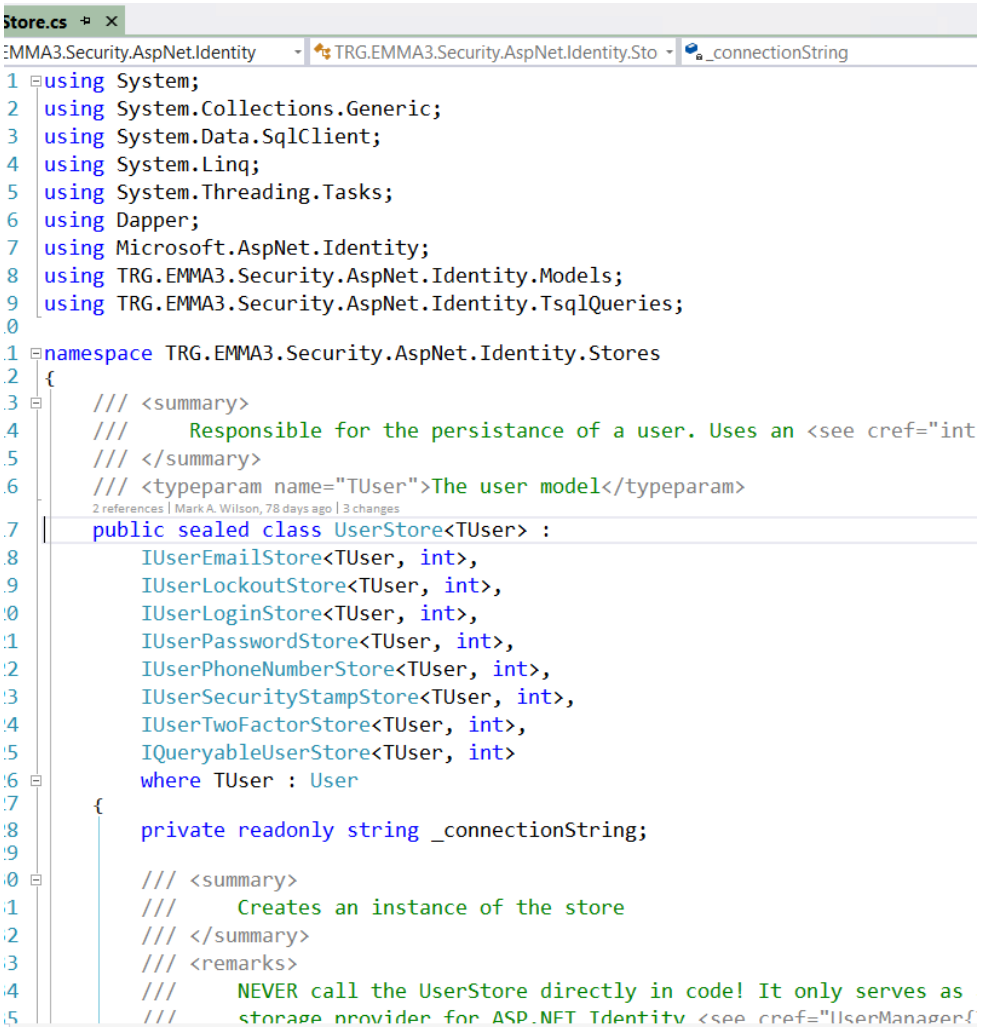
UserManager

All configuration.



UserStore

Gets/Sets don't alter data in the Data Source (DB).



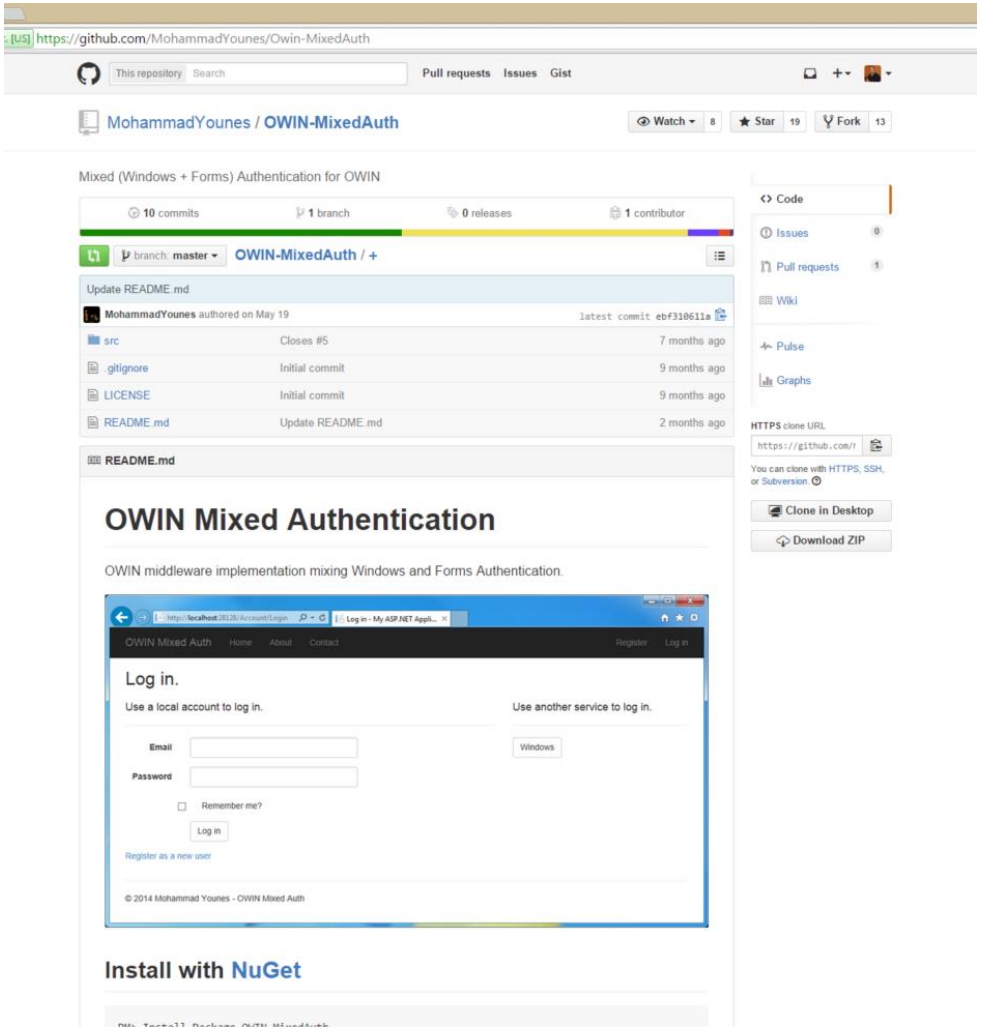
Lessons Learned

- Pretty easy to create.
- Permits integer primary key.
- Interact with managers, not stores.
- Gets/Sets don't alter data in the Data Source (DB).
- In OAuthConfig, configure the database context, user manager, and role manager to use a single instance per request.

Mixed Authentication

OWIN MixedAuth

Third-party Windows login provider.



- https://github.com/MohammadYounes/Owin-MixedAuth

Lessons Learned

- Proof of concept?
- Enable Windows Authentication.
- Required small code changes.
- Uses a pop-up window.
- Uses a cookie.

Resources

- **Fiddler** (http://www.telerik.com/fiddler)
- **Token Based Authentication using ASP.NET Web API 2, Owin, and Identity** (http://bitoftech.net/2014/06/01/token-based-authentication-asp-net-web-api-2-owin-asp-net-identity/)
- **Overview of Custom Storage Providers for ASP.NET Identity** (http://www.asp.net/identity/overview/extensibility/overview-of-custom-storage-providers-for-aspnet-identity)
- **OWIN Mixed Authentication** (https://github.com/MohammadYounes/Owin-MixedAuth)

Authentication Using Tokens for AngularJS, OWIN, ASP.NET Web API & Identity

Mark A. Wilson
Senior Developer
MarkW@LogicalAdvantage.com

www.DeveloperInfra.com
@DeveloperInfra