Secrets NET8 EF8 Blazor VS

# Place a Secret in an Azure Key Vault

## Create Azure Key Vault

If you haven’t already, create an Azure Key Vault:

Launch Microsoft Azure:

Type “key vault” into search:

Select Key vaults: 

Select <+Create>:  to display the Create a key vault form:

A screenshot of a computer

Description automatically generated

Select your Subscription: (In this example I selected my “Visual Studio Professional Subscription”

Select you Resurce Group (or create a new resource group)

Enter your key vault name (I use a prefix followed by a dash for all of my names wherever I can). In this instance I am using kv-xxxxxxx. (This must be globally unique for all users in the Region)

Select the Azure Region (I use Centra US or East US 2 since both of them provide a full slate of services)

Select the Standard pricing tier since it is free.

I use the defaults for the Recovery options

Select <Next>:

A screenshot of a computer

Description automatically generated

I don’t make any changes to the Access Configuration page.

Select <Next>:

A screenshot of a computer

Description automatically generated

I use the defaults for the Networking page since I want to be able to access the key vault from my personal workstation.

Select <Review + create>

A screenshot of a computer

Description automatically generated

Review your entries and select <Create> to deploy the Key Vault.

A white background with black text

Description automatically generated

Select <Go to resource>:

A screenshot of a computer

Description automatically generated

You will probably not have enough access rights to add a secret.

Select <Access control (IAM)> on the left side nav bar:

A screenshot of a computer

Description automatically generated

If you are the owner of the Azure Subscription you can add yourself.

Select <Role assignments> from the top menu:

A screenshot of a computer

Description automatically generated

Select <Add> and <Add role assignment> from the top menu:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Enter “Key Vault” into the search by role name box:

A screenshot of a computer

Description automatically generated

Select “Key Vault Administrator”:

Select <Members> from the top menu:

A screenshot of a computer

Description automatically generated

Select <+ Select members>

The Select members dialog displays and you can select yourself and others:

A screenshot of a computer

Description automatically generated

Select <Select> to add the members to the Key Vault Administrators group.

A screenshot of a computer

Description automatically generated

You aren’t done yet:

Select <Next> or <Review + assign> to get the Review + assign form:

A screenshot of a computer

Description automatically generated

Select <Review and assign> to assign the selected users to the role.

Now, go back and do the same thing for Key Vault Reader so you can have access to the secrets from your workstation.

When completed, your Role Assignments form should look something like this:

A screenshot of a computer

Description automatically generated

Now you should be able to add the SyncFusion secret:

Select <Secrets> from the menu on the left:

A screenshot of a computer

Description automatically generated

Select <+ Generate/Import> :

A screenshot of a computer

Description automatically generated

Select Upload Options of “Manual”

Enter a Name for the key. For this example I have used “SyncFusionKey”

Paste your Sync Fusion Activation Key into the Secret Value

Enter a description of the key in Content Type. This shows up on other screens.

Clear bot the Set Activation Date and Set Expiration Date checkboxes.

Enabled is the default.

Select <Create> to create the secret and you should see something like:

A red arrow pointing to a white box

Description automatically generated

# Register your application with Azure

You need to register your application with Azure as an “Enterprise application”:

In Azure, enter “Enterprise Applications” in search and select :

You should see a list of your currently registered applications:

A screenshot of a computer

Description automatically generated

Select <+ New application> :

You should see a confusing page something like the following that has little to do with registering our new application:

A screenshot of a computer

Description automatically generated

Select <+ Create your own application>  and the Create your own application dialog displays:

A screenshot of a computer application

Description automatically generated

Enter the name of your application in the name text.

Select the “Register an application” radio button.

Select <Create>

The Register an application form displays:

A screenshot of a computer application

Description automatically generated

For this exercise, accept the defaults unless you know you want make additional selections.

Select <Register>

The Browse Microsoft Entra Gallery form displays where you can register another application, or:

Select “Enterprise applications” from the breadcrumb bar:

A blue and white box with text

Description automatically generated

The enterprise applications form displays:

A screenshot of a computer

Description automatically generated

Select your application from the list to display the application values:

A screenshot of a computer

Description automatically generated

There are many options and additional data associated with an application registration, but we are only interested in the Name, Application ID, and Object ID:

## Name

The application name is “SyncFusion\_NET8\_EF8\_Blazor\_VS\_V1”

## Application ID

The Application ID is “c8263da7-0d9b-4430-80ff-f1d52a031e45”

## Object ID

The Object ID is “1bdc698d-c4a5-4603-9140-1a69d7387977”

# Get Values from App registrations

Load the application on the App Registrations screen to get the client ID and tenant ID

A screenshot of a computer

Description automatically generated

# Prepare your project

Using PowerShell run the following on the directory containing your project:

dotnet add package Azure.Identity

dotnet add package Azure.Security.KeyVault.Secrets

These statements loaded the necessary NuGet packages.

# Add Azure Secrets Class to VS Project

In the root of the project, add a new class named Secrets.cs with the following code:

using Azure.Identity;

using Azure.Security.KeyVault.Secrets;

namespace KeyVault01.Secrets

{

public class Secrets()

{

public string GetSecret(string SecretName)

{

string userAssignedClientId = "7e717e34-790b-4670-8a71-bb862d4883c1";

var credential = new DefaultAzureCredential(new DefaultAzureCredentialOptions

{

// set the user assigned identity in Azure

ManagedIdentityClientId = userAssignedClientId

// exclude Managed Identity

,ExcludeManagedIdentityCredential = true

});

var client = new SecretClient(

new Uri("https://kv-ltslm-dev.vault.azure.net/"), credential);

var secret = client.GetSecret(SecretName);

return secret.Value.Value.ToString();

}

}

}

## clientId

The clientId maps to the Azure Application “Application ID”

## clientSecret

The clientSecret maps to the Azure Secret “6c37d01d5c0a4ed9a449cb63b70945b5”

## vaultUrl

The vaultUrl maps to the path portion of the Secret Identifier:

“https://kv-xxxxxxx.vault.azure.net/secrets/SyncFusionKey”

## tenantId

# Add Azure Secrets to Programs.cs

Add the following code to the Programs.cs file to retrieve a secret:

// retrieve SQL connection string from Key Vault

Secrets secrets = new Secrets();

string SqlConnectionString = secrets.GetSecret("ltSLM-SQLServer-DEV");

Console.WriteLine(SqlConnectionString);

# Add a SyncFusion component to a Blazor/Razor page