A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

public class MvcApplication : System.Web.HttpApplication

{

protected void Application\_Start()

{

AreaRegistration.RegisterAllAreas();

FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

RouteConfig.RegisterRoutes(RouteTable.Routes);

BundleConfig.RegisterBundles(BundleTable.Bundles);

Task.Run(() =>

{

var vaultService = new VaultService();

var secretValue = vaultService.GetSecretValueAsync("/my-secret-password", "password", "secret").Result;

var a = secretValue;

}).Wait();

}

}

public class VaultService

{

private readonly IVaultClient \_vaultClient;

public VaultService()

{

var vaultAddress = @"http://localhost:8200";

var vaultToken = "testtoken";

\_vaultClient = new VaultClient(new VaultClientSettings(vaultAddress, new TokenAuthMethodInfo(vaultToken)));

}

public async Task<string> GetSecretValueAsync(string path, string key,string mountPoint)

{

try

{

Secret<SecretData> secret = \_vaultClient.V1.Secrets.KeyValue.V2.ReadSecretAsync(

path: path,

mountPoint: mountPoint

).Result;

return secret.Data.Data[key].ToString();

}

catch (Exception ex)

{

Console.WriteLine($"Error retrieving secret from Vault: {ex.Message}");

return null;

}

}

}