add a project with only empty api selected

create model class

create databaseContext class

define connection string in Web.config

install entity framework via nugat package manager

Commands

1.enable-migrations

2.add-migration initial

3.update-database –verbose

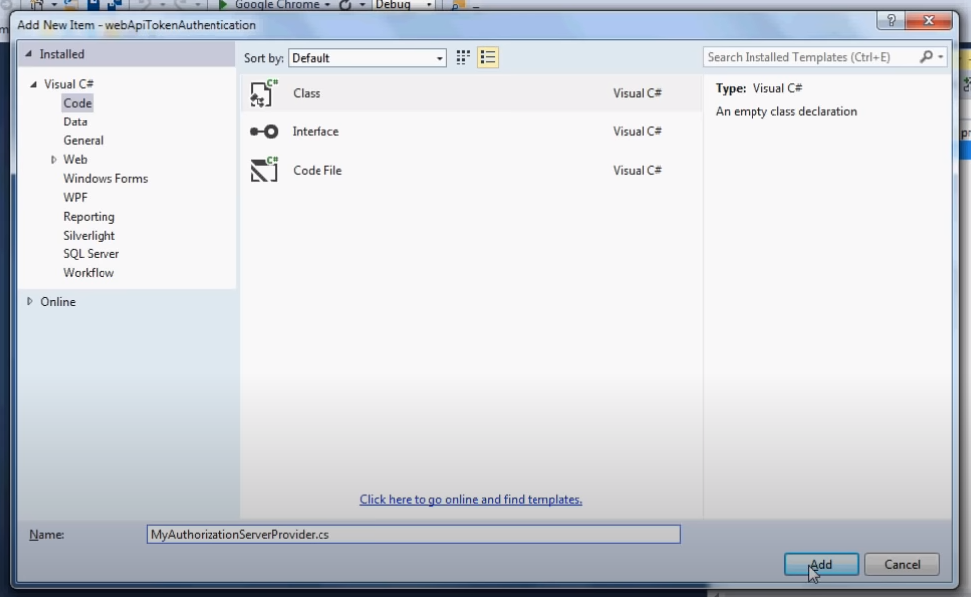
============================= Code for token based authentication ================

<https://www.youtube.com/watch?v=rMA69bVv0U8>

Install through nugat package

1. OWIN
2. Microsoft.Owin.Host.System.Web
3. Microsoft.Owin
4. Microsoft.Owin.Security
5. Microsoft.Owin.Security.OAuth
6. Microsoft.Owin.Cors

Create a class MyAuthorizationProvider



public class MyAuthorizationProvider:OAuthAuthorizationServerProvider

{

public override async Task ValidateClientAuthentication(OAuthValidateClientAuthenticationContext context)

{

context.Validated();

}

public override async Task GrantResourceOwnerCredentials(OAuthGrantResourceOwnerCredentialsContext context)

{

var identity = new ClaimsIdentity(context.Options.AuthenticationType);

if (context.UserName=="admin" && context.Password=="admin")

{

identity.AddClaim(new Claim(ClaimTypes.Role, "admin"));

identity.AddClaim(new Claim("username", "admin"));

identity.AddClaim(new Claim(ClaimTypes.Name, "Anurag"));

context.Validated(identity);

}

else if (context.UserName=="user" && context.Password=="user")

{

identity.AddClaim(new Claim(ClaimTypes.Role, "user"));

identity.AddClaim(new Claim("username", "user"));

identity.AddClaim(new Claim(ClaimTypes.Name, "Raj"));

context.Validated(identity);

}

else

{

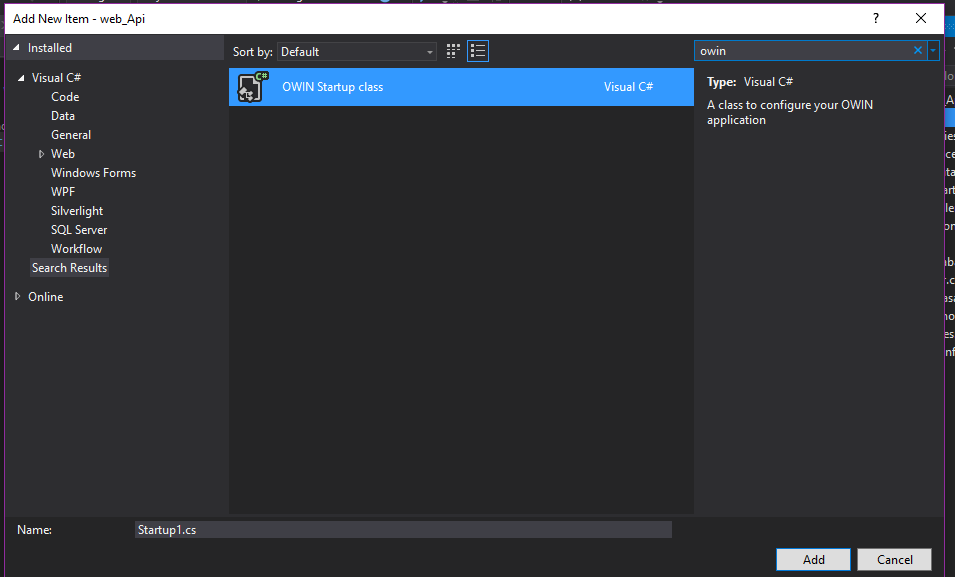
context.SetError("Invalid ", "Username Or Password incorrenct");

}

}

}

Now add OWIN Startup class



Rename as Startup

Then code this

public class Startup

{

public void Configuration(IAppBuilder app)

{

// enable cors

app.UseCors(Microsoft.Owin.Cors.CorsOptions.AllowAll);

var myProvider = new MyAuthorizationProvider();

OAuthAuthorizationServerOptions options = new OAuthAuthorizationServerOptions

{

AllowInsecureHttp = true,

TokenEndpointPath = new PathString("/token"),

AccessTokenExpireTimeSpan = TimeSpan.FromDays(1),

Provider = myProvider

};

app.UseOAuthAuthorizationServer(options);

app.UseOAuthBearerAuthentication(new OAuthBearerAuthenticationOptions());

HttpConfiguration config = new HttpConfiguration();

WebApiConfig.Register(config);

}

}

Add a class for unauthorized user handle AuthorizeAttribute

Now add this code

public class AuthorizeAttribute : System.Web.Http.AuthorizeAttribute

{

protected override void HandleUnauthorizedRequest(HttpActionContext actionContext)

{

if (!HttpContext.Current.User.Identity.IsAuthenticated)

{

base.HandleUnauthorizedRequest(actionContext);

}

else

{

actionContext.Response = new System.Net.Http.HttpResponseMessage(System.Net.HttpStatusCode.Forbidden);

}

}

}

Add a Controller for test Authentication/Authorization

TestController