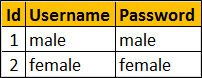
In this video we will discuss **how to implement basic authentication in ASP.NET WEB API**. This is continuation to [Part 17](https://www.youtube.com/watch?v=xIzlD-frEw4&index=17&list=PL6n9fhu94yhW7yoUOGNOfHurUE6bpOO2b). Please watch [Part 17](https://www.youtube.com/watch?v=xIzlD-frEw4&index=17&list=PL6n9fhu94yhW7yoUOGNOfHurUE6bpOO2b) from [ASP.NET Web API tutorial](https://www.youtube.com/playlist?list=PL6n9fhu94yhW7yoUOGNOfHurUE6bpOO2b).  
  
Create Users table in SQL Server   
   
  
  
  
**SQL Scrip to create the table**

Create Table Users

(

     Id int identity primary key,

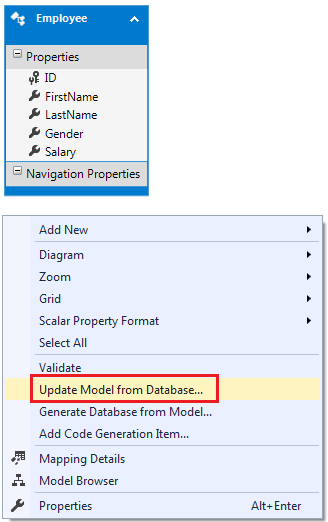
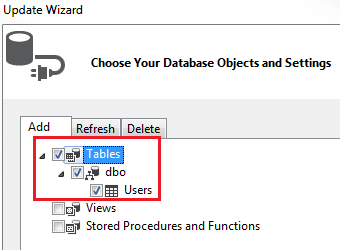
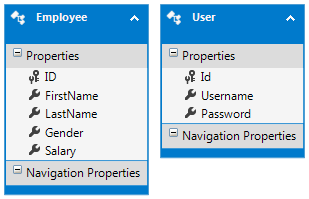
     Username nvarchar(100),

     Password nvarchar(100)

)

Insert into Users values ('male','male')

Insert into Users values ('female','female')

If we login with "male" username, we want to display only "male" employees and if we login with "female" username, we want to display only female employees.  
  
  
  
**Update ADO.NET Entity Data Model**  
1. Open **"EmployeeDataModel.edmx"** in EmployeeDataAccess project  
  
2. Right click on the designer surface, and select **"Update Model from Database"**option   
   
  
3. On the next screen, expand **"Tables"** and select **"Users"**table and click **"Finish"**   
   
  
4. At this point you will have the **"Users"** table added to the Entity Data Model   
   
  
**Create a class that checks if the username and password are valid**  
1. Add a new class file to **EmployeeService** Web API project. Name it **EmployeeSecurity.cs**  
  
2. Copy and paste the following code in it

using EmployeeDataAccess;

using System;

using System.Linq;

namespace EmployeeService

{

    public class EmployeeSecurity

    {

        public static bool Login(string username, string password)

        {

            using (EmployeeDBEntities entities = new EmployeeDBEntities())

            {

                return entities.Users.Any(user =>

                       user.Username.Equals(username, StringComparison.OrdinalIgnoreCase)

                                          && user.Password == password);

            }

        }

    }

}

**Create basic authentication filter**  
1. Add a new class file to **EmployeeService**Web API project. Name it **BasicAuthenticationAttribute.cs**  
  
2. Copy and paste the following code in it

using System;

using System.Net;

using System.Net.Http;

using System.Security.Principal;

using System.Text;

using System.Threading;

using System.Web.Http.Controllers;

using System.Web.Http.Filters;

namespace EmployeeService

{

    public class BasicAuthenticationAttribute : AuthorizationFilterAttribute

    {

        public override void OnAuthorization(HttpActionContext actionContext)

        {

            if (actionContext.Request.Headers.Authorization == null)

            {

                actionContext.Response = actionContext.Request

                    .CreateResponse(HttpStatusCode.Unauthorized);

            }

            else

            {

                string authenticationToken = actionContext.Request.Headers

                                            .Authorization.Parameter;

                string decodedAuthenticationToken = Encoding.UTF8.GetString(

                    Convert.FromBase64String(authenticationToken));

                string[] usernamePasswordArray = decodedAuthenticationToken.Split(':');

                string username = usernamePasswordArray[0];

                string password = usernamePasswordArray[1];

                if (EmployeeSecurity.Login(username, password))

                {

                    Thread.CurrentPrincipal = new GenericPrincipal(

                        new GenericIdentity(username), null);

                }

                else

                {

                    actionContext.Response = actionContext.Request

                        .CreateResponse(HttpStatusCode.Unauthorized);

                }

            }

        }

    }

}

**Enable basic authentication**  
1. The BasicAuthenticationAttribute can be applied on a specific controller, specific action, or globally on all Web API controllers.  
  
2. To enable basic authentication across the entire Web API application, register BasicAuthenticationAttribute as a filter using the Register() method in WebApiConfig class

config.Filters.Add(new RequireHttpsAttribute());

3. You can also apply the attribute on a specific controller, to enable basic authentication for all the methods in that controller  
  
4. In our case let's just enable basic authentication for Get() method in EmployeesController. Also modify the implementation of the Get() method as shown below.

[BasicAuthentication]

public HttpResponseMessage Get(string gender = "All")

{

    string username = Thread.CurrentPrincipal.Identity.Name;

    using (EmployeeDBEntities entities = new EmployeeDBEntities())

    {

        switch (username.ToLower())

        {

            case "male":

                return Request.CreateResponse(HttpStatusCode.OK,

                    entities.Employees.Where(e => e.Gender.ToLower() == "male").ToList());

            case "female":

                return Request.CreateResponse(HttpStatusCode.OK,

                    entities.Employees.Where(e => e.Gender.ToLower() == "female").ToList());

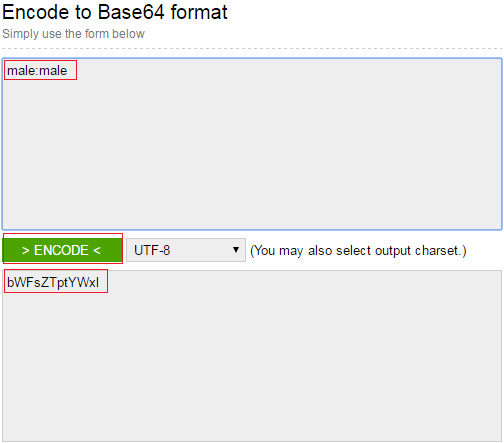
            default:

                return Request.CreateResponse(HttpStatusCode.BadRequest);

        }

    }

}

**Testing basic authentication using fiddler**  
1. The username and password need to be colon (:) separated and base64 encoded.   
  
2. Just google with the string - base64 encode. The first web site that you get is <https://www.base64encode.org/>  
  
3. Enter the username and password separated by colon (:) in **"Encode to Base64 format"**  textbox, and then click **"Encode"**button   
   
  
4. Opend Fiddler and issue a Get request to http://localhost:35171/api/employees  
Testing basic authentication using fiddler.png  
  
5. The request succeeds and we get the data as expected.  
  
6. If we supply a username and password that does not exist we get 401 Unauthorized  
  
In our next video we will discuss **how to pass basic authentication credentials to the Web API service using jQuery AJAX**.