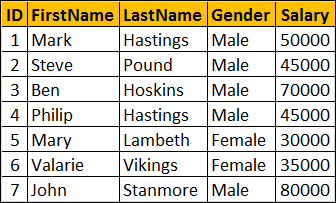
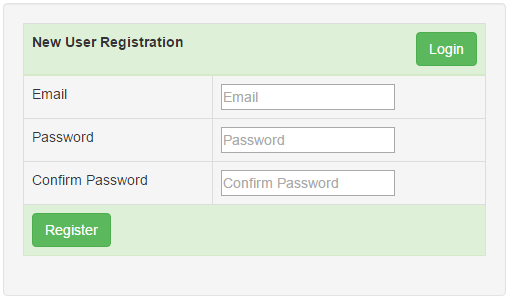
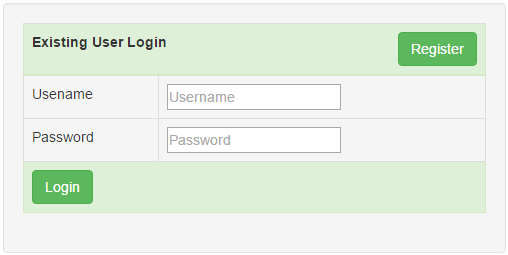
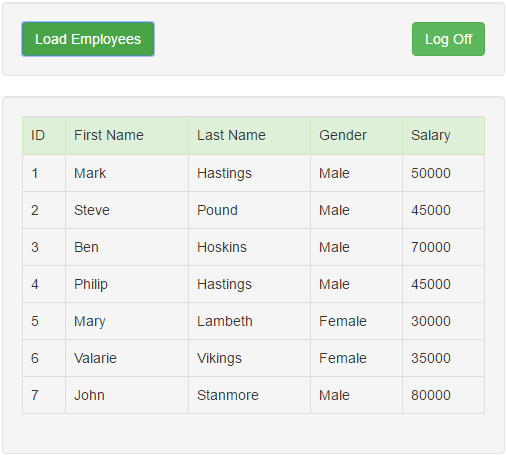
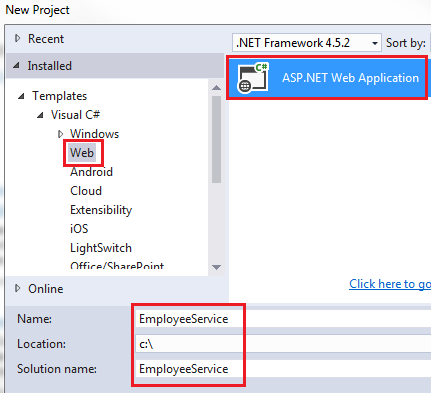
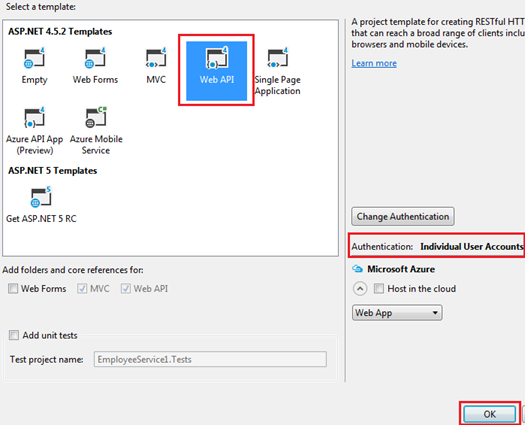
In this video and in a few upcoming videos, we will discuss step by step, how to implement **token based authentication in ASP.NET Web API** using OWIN middleware and Identity framework.  
  
  
  
We will be using the following Employees table for this demo.   
   
  
  
  
Only authenticated users should be able to access the above Employee data. In addition new users should be able to register. The new user registration form looks as shown below.   
   
  
Once a new user is registered with our application, he should be able to login into our system, using the login form shown below.   
   
  
If the user is successfully authenticated, then he will be redirected to the Data.html page, which displays the employee data from the Employees table as shown below. When the "Log Off" button is clicked, the user is logged out and redirected to the login page.   
   
  
In addition to OWIN middleware and ASP.NET Identity, we will use the following technologies. Click on the links if you are new to any of these technologies

* [ASP.NET Web API](https://www.youtube.com/playlist?list=PL6n9fhu94yhW7yoUOGNOfHurUE6bpOO2b)
* [SQl Server](https://www.youtube.com/playlist?list=PL08903FB7ACA1C2FB)
* [ADO.NET Entity Framework](https://www.youtube.com/playlist?list=PL6n9fhu94yhUPBSX-E2aJCnCR3-_6zBZx)
* [Bootstrap](https://www.youtube.com/playlist?list=PL6n9fhu94yhXd4xnk-j5FGhHjUv1LsF0V)
* [jQuery](https://www.youtube.com/playlist?list=PL6n9fhu94yhVDV697uvHpavA3K_eWGQap)

**Steps to build the ASP.NET Web API service to retrieve data from SQL Server**  
  
**Step 1 :** Run Visual Studio and Select File - New Project  
  
**Step 2 :**In the "New Project" dialog box, select "Web" under "Installed" templates. From the middle pane, select "ASP.NET Web Application". Name the project "EmployeeService" and click "OK"   
   
  
**Step 3 :** On the next screen select **"Web API"** and set Authentication to **"Individual User Accounts"** and click **"OK"**. The Individual User Accounts option uses a membership database in which the users that we register will be stored. We will discuss the Membership Database in a later video in this series.   
   
  
**Step 4 :** Execute the following script in SQL Server Management Studio to create the Employees table and populate it with test data

Create Database EmployeeDB

Go

Use EmployeeDB

Go

Create table Employees

(

     ID int primary key identity,

     FirstName nvarchar(50),

     LastName nvarchar(50),

     Gender nvarchar(50),

     Salary int

)

Go

Insert into Employees values ('Mark', 'Hastings', 'Male', 60000)

Insert into Employees values ('Steve', 'Pound', 'Male', 45000)

Insert into Employees values ('Ben', 'Hoskins', 'Male', 70000)

Insert into Employees values ('Philip', 'Hastings', 'Male', 45000)

Insert into Employees values ('Mary', 'Lambeth', 'Female', 30000)

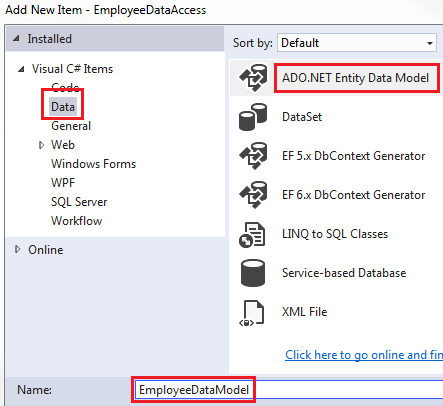
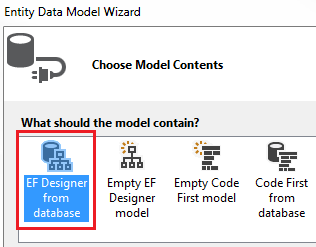
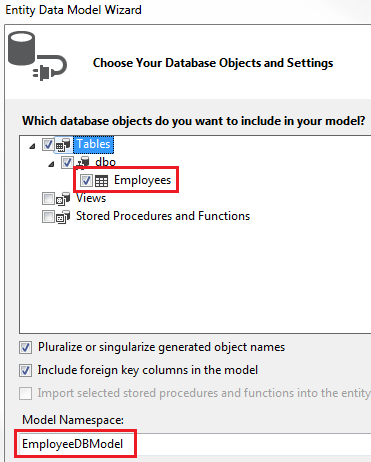
Insert into Employees values ('Valarie', 'Vikings', 'Female', 35000)

Insert into Employees values ('John', 'Stanmore', 'Male', 80000)

Go

**Step 5 :** Right click on "EmployeeService" project and select Add - New Item  
  
**Step 6 :** In the "Add New Item" window, 

* Select "Data" from the left pane
* Select ADO.NET Entity Data Model from the middle pane
* In the Name text box, type EmployeeDataModel and click Add

   
  
**Step 7 :** On the Entity Data Model Wizard, select **"EF Designer from database"** option and click next   
   
  
**Step 8 :** On the next screen, click "New Connection" button  
  
**Step 9 :** On "Connection Properties" window, set  
    Server Name = (local)  
    Authentication = Windows Authentication  
    Select or enter a database name = EmployeeDB  
    Click OK and then click Next  
  
**Step 10 :** On the nex screen, select "Employees" table and click Finish.   
   
  
**Step 11 :** Right click on the Controllers folder in EmployeeService project and select Add - Controller  
  
**Step 12 :**Select "Web API 2 Controller - Empty" and click "Add"  
  
**Step 13 :**On the next screen set the Controller Name = EmployeesController and click Add  
  
**Step 14 :** Copy and paste the following code in EmployeesController.cs

using System.Collections.Generic;

using System.Linq;

using System.Web.Http;

namespace EmployeeService.Controllers

{

    public class EmployeesController : ApiController

    {

        public IEnumerable<Employee> Get()

        {

            using(EmployeeDBEntities entities = new EmployeeDBEntities())

            {

                return entities.Employees.ToList();

            }

        }

    }

}

At this point when you navigate to /api/employees you should see all employees.  
  
**Step 15 :** Decorate the EmployeesController with [Authorize] attribute. 

[Authorize]

public class EmployeesController : ApiController

**Step 16 :** Build the solution and when you navigate to /api/employees we get "Authorization has been denied for this request"  
  
Since the EmployeesController is decorated with [Authorize] attribute, all the actions in the controller must be authenticated. Otherwise, we get 401 error - Authorization has been denied for this request.  
  
In our next video we will discuss how to register new users and in a video after that we will explore where these users are stored. In the subsequent video we will discuss implementing the "Login" page.