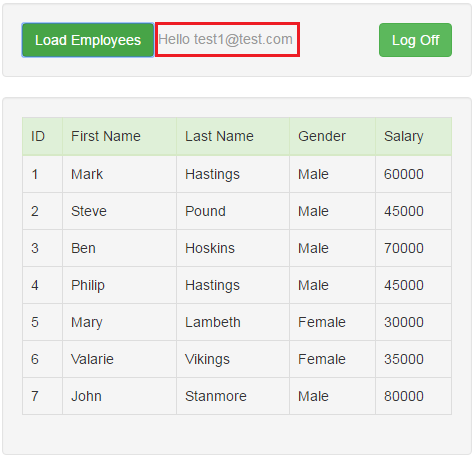
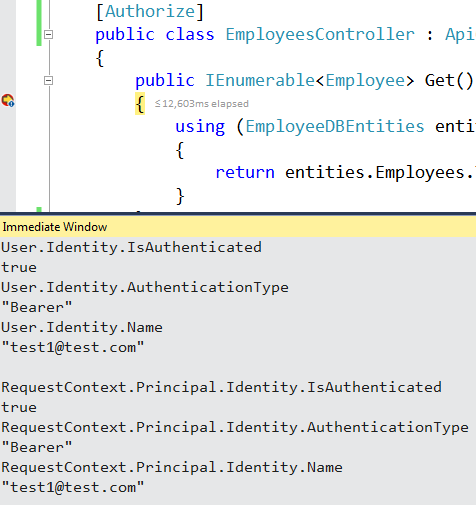
In this video we will discuss **how to get authenticated user identity name in asp.net web api and display it on the web page**. This is continuation to [Part 26](http://csharp-video-tutorials.blogspot.com/2016/12/aspnet-web-api-logout.html). Please watch [Part 26](http://csharp-video-tutorials.blogspot.com/2016/12/aspnet-web-api-logout.html) from [ASP.NET Web API tutorial](https://www.youtube.com/watch?v=0pcM6teVdKk&list=PL6n9fhu94yhW7yoUOGNOfHurUE6bpOO2b) before proceeding.   
  
We want to display the logged in username on the web page as shown below.   
   
  
  
  
**Step 1 :** On the Login.html page, store the username in the browser local storage. The **success()** function is called when the user is successfully logged in. The response that we get from the server includes userName property which has the userName that is used to login. We are storing the logged in username in the broswer localstorage using the key userName.  
  
success: function (response) {  
    localStorage.setItem("accessToken", response.access\_token);  
    localStorage.setItem("userName", response.userName);  
    window.location.href = "Data.html";  
}   
  
  
  
**Step 2 :** On the Data.html page include the following span element just below the **"Load Employees"** button. We will use this span element to display the logged in username.

<span id="spanUsername" class="text-muted"></span>

**Step 3 :** On the Data.html page, in $(document).ready() function include the following line of jQuery code to retrieve and display the logged in username.

$('#spanUsername').text('Hello ' + localStorage.getItem('userName'));

**How to get logged in user identity details in ASP.NET Web API controller**  
From the ASP.NET Web API controller use the User.Identity object to retrieve user details  
  
User.Identity.IsAuthenticated - Returns true or false depending on whether the user is authenticated  
User.Identity.AuthenticationType - Authentication Type used  
User.Identity.Name - Logged in username  
  
RequestContext.Principal.Identity.IsAuthenticated - Returns true or false depending on whether the user is authenticated  
RequestContext.Principal.Identity.AuthenticationType - Authentication Type used  
RequestContext.Principal.Identity.Name - Logged in username  
  
Instead of using User.Identity, we can also use RequestContext.Principal.Identity  
  
To see what the respective properties return, set a break point in the Get() method of Employees controller and run the application in debug mode. Navigate to Login page and login. On the Data.html page click "Load Employees" button. When the execution pauses at the breakpoint, open Immediate Window and type the above properties one by one and press enter. To open immediate window, in Visual Studio select Debug - Windows - Immediate.   
   
  
Now let us see what these properties return if we are not logged in.  
1. Add a new **Empty Web API 2 Controller** to the **EmployeeService**project. Name it TestController.  
2. Copy and paste the following code in TestController. 

using System.Web.Http;

namespace EmployeeService.Controllers

{

    public class TestController : ApiController

    {

        public string Get()

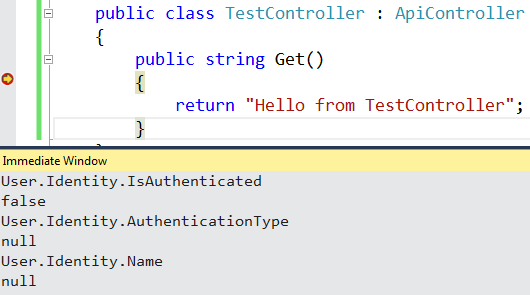
        {

            return "Hello from TestController";

        }

    }

}

3. Set a breakpoint on the Get() method in TestController. Run the application in Debug mode and navigate to /api/Test. When the execution pauses at the break point, open Immediate Window and type the above properties one by one and press enter.   
   
  
Since we are not logged in, notice IsAuthenticated property returns false. Since we are not authenticated AuthenticationType and Name properties return NULL.