**BASIC AUTHENTICATION**

1.Create a .NET core WEB API project in Visual Studio.

A screenshot of a computer

Description automatically generated

2. Create a BasicAuthenticationHandler.cs class file.

A screenshot of a computer

Description automatically generated

3. Inherit the AuthenticationHandler class.



4.Create a constructor of BasicAuthenticationHandler and implement the method HandleAuthenticateAsync().

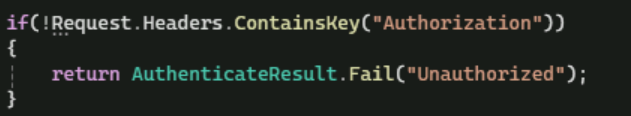


5.In the HandleAuthenticateAsync() method, we have to add few conditions before going ahead of authorizing the endpoints.

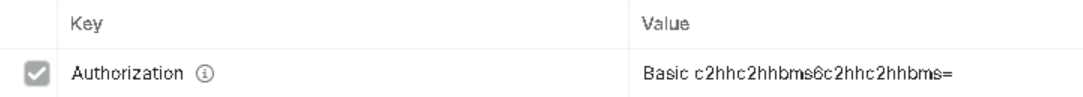
6.In the header check if “Authorization” key is present or not, if not return “Unauthorized”.

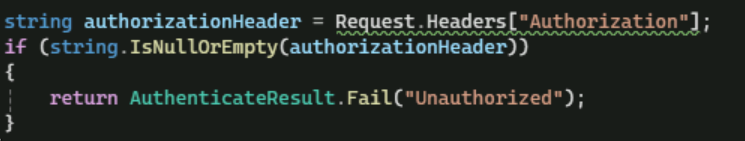
A white background with black lines

Description automatically generated



7.Check if the value of “Authorization” key is null or empty, if it is return “Unauthorized”.





8.Check if the value starts with “basic “ . OrdinalIgnoreCase is used as it compares as if they are converted to Uppercase.

A black text on a white background

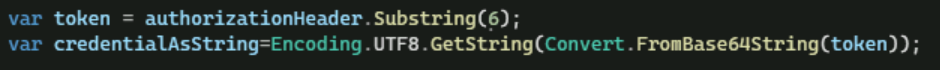
Description automatically generated

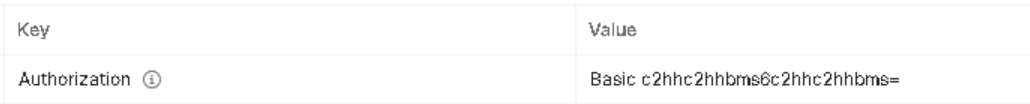
A black background with white text

Description automatically generated

9.Get the value of Authorization key starting from the 7th position.

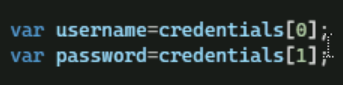
The username and password passed on by the user will be in the form of key:value pair. It will be converted to base64 string. Now in order to check that against the username and password specified or mentioned in the program it should be converted back from base64 string to key value pair.





Where 0th index will have username and 1st index will have password.





Check if there are more than two words, if yes that is not the username and password. Return “Unauthorized.

A black background with white text

Description automatically generated

10.Check if the username and password obtained after conversion from base64 string matches with the username and password mentioned in the program. If yes,

* Using the Claim class, generate on object of it by passing the NameIdentifier of ClaimTypes and username.
* Generate the identity using the claims object and with word “Basic”.
* Using this identity generate the claimsPrincipal.
* Using this claimsPrincipal return the Success message passing the AuthenticationTicket by passing the claimsPrincipal and Scheme.Name.

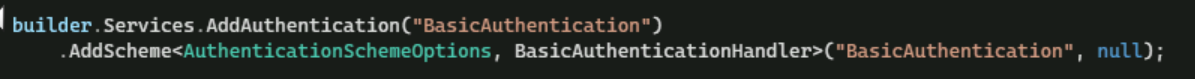
Scheme.Name is the name of the Authentication as specified in the program.cs.

If username and password does not match, then return “Unauthorized”.

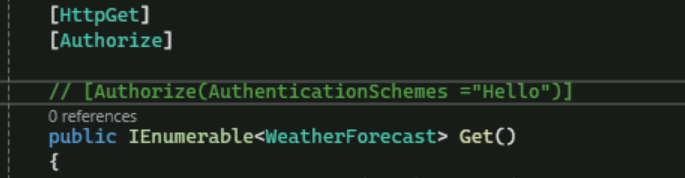
A screen shot of a computer code

Description automatically generated

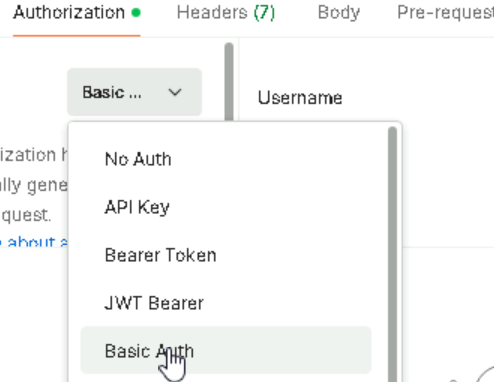
11.Once the implementation is done, add the service in program.cs



12.In the controller, decorate the necessary endpoints to be authorized by adding [Authorize] tag.



13.Test the endpoints through postman by selecting the Basic Auth option inside the Authorization.



14.By specifying the username and password mentioned or hardcoded in the program exactly, we can obtain the positive result.

15.If the username and password is matched we will be able to access the endpoint with a message of 200 Ok success in postman.

16.If not, we will be getting a 401 unauthorized failure message.