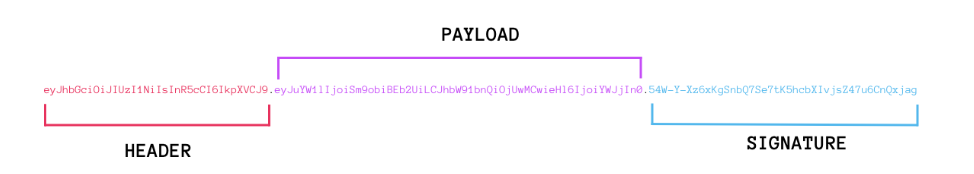
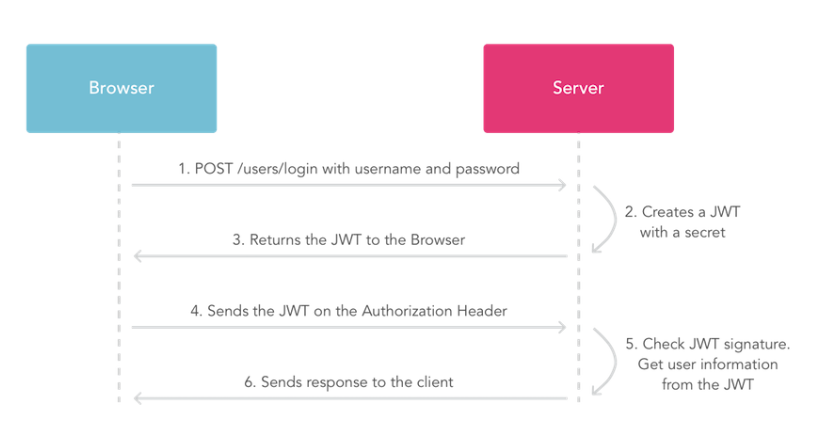
# **Auth JWT**

## What is JWT (Json Web Token)?

JSON Web Token (JWT) is a token based authentication between two servers. In this a JSON object that is digitally signed using JSON Web Signature (JWS) and encrypted using JSON Web Encryption (JWE).



## Authentication sequence for JWT



## What is Auth JWT?

Auth JWT is a helper to achieve JWT authentication in ASP.Net Core applications in an easy and flexible way.

## Hashing Algorithms Used

* SHA1
* SHA256
* SHA384
* SHA512

## Components of Auth JWT

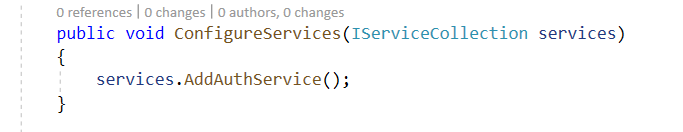
* **TokenRequestModel** is a class which contain properties for Auth JWT payload.
  + [Note: All properties are not mandatory.]
* ***Issuer***
* ***ExpiryInSeconds***
* ***UserId***
* ***User***
* ***Role***
* ***Audience***
* ***JwtId***
* ***Subject***
* ***CustomProperty***
* **AlgorithmType** is constant class to get the algorithm details
* ***SHA1***
* ***SHA256***
* ***SHA384***
* ***SHA512***
* **JWTModule** is a class which contains all the interfaces for Auth JWT.
* ***CreateToken(<TokenRequestModel>, <secrect>, <AlgoritmType>)*** -> Function creates the token
* ***VerifyToken(<token>, <secrect>, <ValidateModel Optional>)*** -> Function verifies the token.
* **TokenResponseModel** is a class which contains the Status and Content of the Auth JWT response.
* ***Status***
* ***Content***
* **ValidateModel** is a class which contain properties to be verified in the VerifyToken() function.
  + [Note: All properties should match the **TokenRequestModel**.]
* ***Issuer***
* ***UserId***
* ***User***
* ***Role***
* ***Audience***
* ***JwtId***
* ***Subject***
* ***CustomProperty***

## Configuration of Auth.JWT

### Step 1:

Add the service to the **ConfigureServices(IServiceCollection services)** function.

***services.AddAuthService();***



### Step 3

Implement Constructor injection in the required classes, to get the object of Auth.JWT.



## Sample Code Snippet

In the below code, I have used Auth.JWT to create and validate the token in a simple class for better understanding.

|  |
| --- |
| class TestAuthJwt  {  private readonly JWTModule \_module;  private readonly TokenRequestModel \_reqModel;  private readonly ValidateModel \_validateModel;  public TestAuthJwt(JWTModule module, TokenRequestModel reqModel, ValidateModel validateModel)  {  \_module = module;  \_reqModel = reqModel;  \_validateModel = validateModel;  }  public void Execute()  {  string secrect = "F4760D";  \_reqModel.Issuer = "authjwt\_team";  \_reqModel.ExpiryInSeconds = "1000";  \_reqModel.UserId = "U1324322";  \_reqModel.User = "sambeet";  \_reqModel.Role = "admin";  \_reqModel.Audience = "authjwt\_app";  \_reqModel.JwtId = "J4433421";  \_reqModel.Subject = "authjwt\_subject";  \_reqModel.CustomProperty.Add("CustomField1","auth\_custom1");  \_reqModel.CustomProperty.Add("CustomField2", "auth\_custom2");  var result = \_module.CreateToken(\_reqModel, secrect, AlgorithmType.SHA256);  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\* Create Token Result\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.WriteLine(Newtonsoft.Json.JsonConvert.SerializeObject(result));  \_validateModel.Issuer = "authjwt\_team";  \_validateModel.UserId = "U1324322";  \_validateModel.User = "sambeet";  \_validateModel.Role = "admin";  \_validateModel.Audience = "authjwt\_app";  \_validateModel.JwtId = "J4433421";  \_validateModel.Subject = "authjwt\_subject";  \_validateModel.CustomProperty.Add("CustomField1", "auth\_custom1");  \_validateModel.CustomProperty.Add("CustomField2", "auth\_custom2");  var verifyResult = \_module.VerifyToken(result.Content, secrect, \_validateModel);  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\* Verify Token Result\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.WriteLine(Newtonsoft.Json.JsonConvert.SerializeObject(verifyResult));  Console.ReadKey();  }  } |