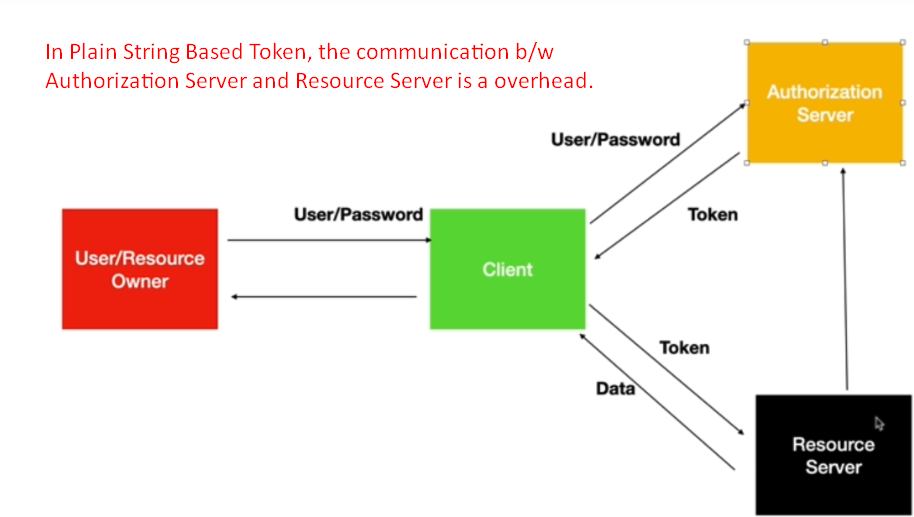
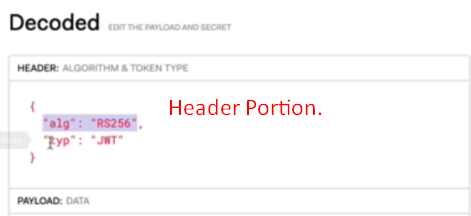
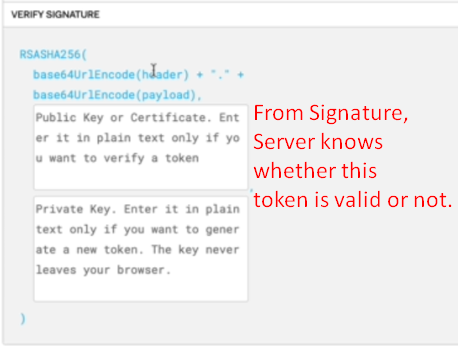
1. **JWT**: JSON Web Token.
2. It is a token format/standard to generate a token.
3. So far, we were using **Plain String Based Tokens**
   1. A client requests for a token.
   2. The authorization server returns a plain string based token.
4. There are disadvantages of this approach.
   1. The **client** sends token back to **Resource Server** when making a request for any URL.
   2. **Resource Server** has to validate this token again by sending the token to the **authorization server** or to a shared DB to fetch the UserDetails to check how much access a user has and is it validated or not.
   3. So, that is the overhead b/w the Resource Server or Communication Server.
5. **Solution**: JWT
6. Its JSON based Token Syntax or Standard.
7. It has 3 parts.
8. 
9. 
10. The algo in the Header is used to generate signature using symmetric or asymmetric key.
11. **Asymmetric Key**:
    1. The authorization server generates public and private key.
    2. The authorization server uses the private key and the public key is shared with all resource servers.
    3. The authorization server will use the private key to sign a token.
    4. Then the resource server uses the public key to verify the signature of the received token.  
       So that is why resource server needs not to make a call to authorization server to verify the received token from a request.
12. **Symmetric Key**:
    1. The public key is used by authentication server and resource servers.
13. 