JWT AUTHENTICATION

https://jwt-auth.readthedocs.io/en/develop/

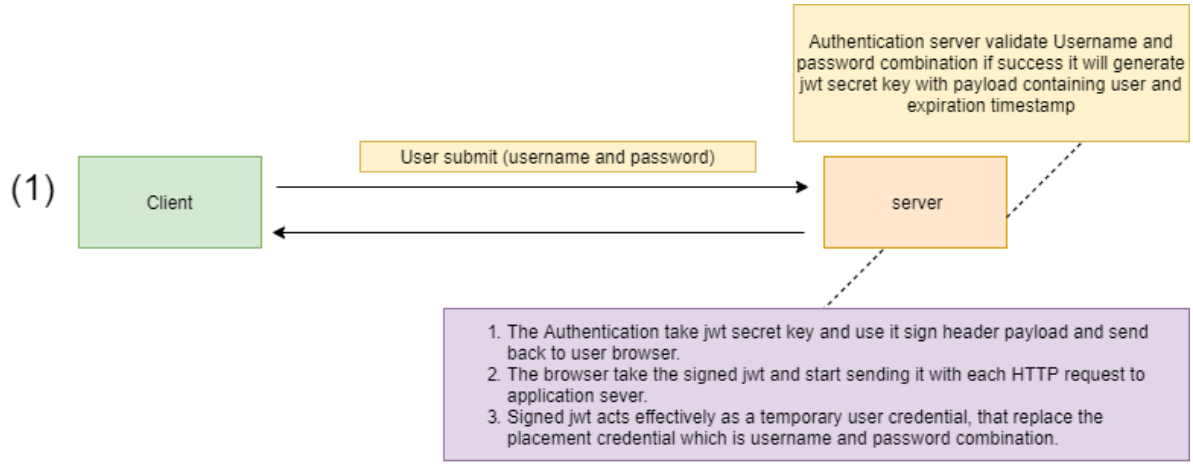
https://jwt.io/introduction

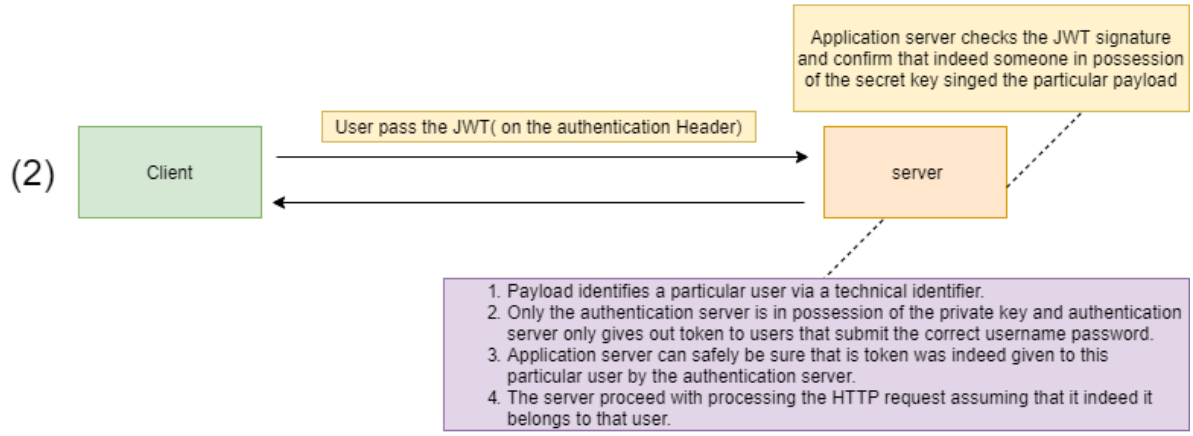
Authorization and authentication process of JWT

Authorization: The client use username, password or SMS login in exchange for a signed JWT token, the Server is valuation on username and password if success it will generate the JWT using secret key. after generate key the server return response secret key back to client. Whenever the user wants to access a protected route or resource, the user agent should send the JWT, typically in the **Authorization** header using the **Bearer** schema.

```
Authorization: Bearer <token>
```

Authentication: Unsign jwt token and verify user permission. have access to a certain interface, open interface to access.





What are JWTs?

A JSON Web Token (or JWT) is simply a JSON payload containing a particular claim. The **key property of JWTs** is that in order to confirm if they are valid we only need to look at the token itself.

A JWT is made of 3 parts: the Header, the Payload and the Signature

We don't have to contact a third-party service or keep JWTs in-memory between requests to confirm that the claim they carry is valid - this is because they carry a Message Authentication Code.

Why use JWTs?

- a centralized in-house custom developed authentication server
- more typically, a commercial product like a LDAP capable of issuing JWTs
- or even a completely external third-party authentication provider such as for example Auth0