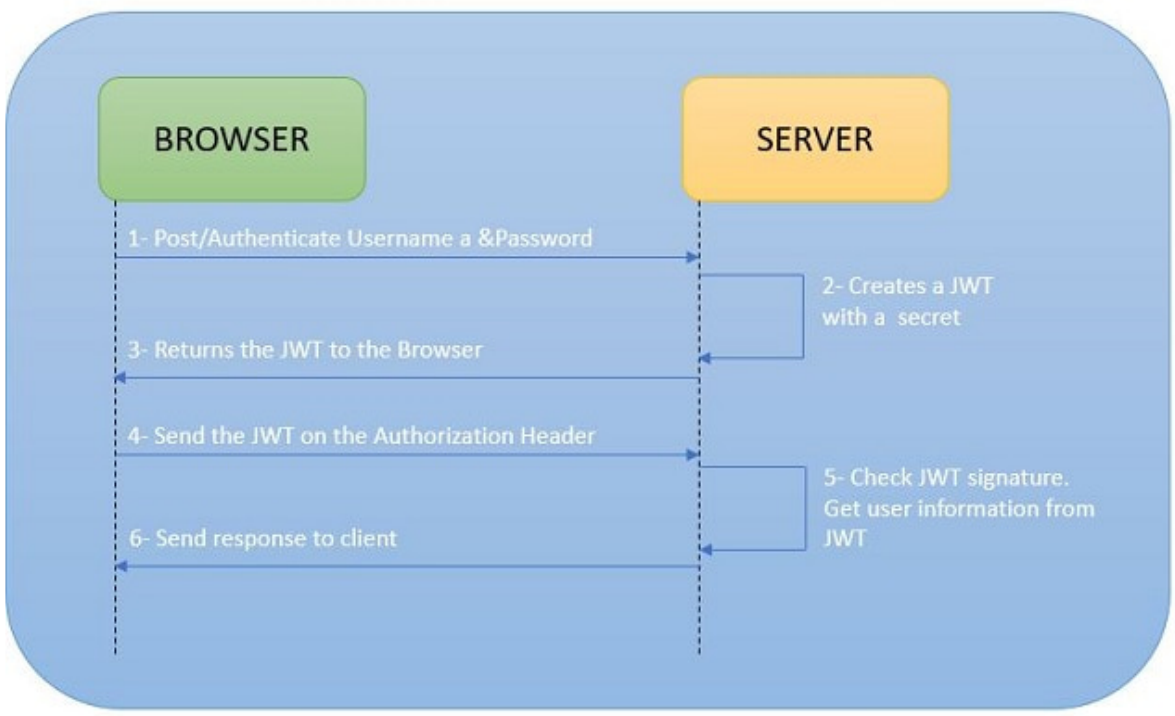
JWT (JSON Web Token) Authentication Introduction

JSON Web Token (JWT) is an open standard (RFC 7519) that specifies a compact and self-contained way of transmitting information securely as a JSON object between parties. This information can be verified and trusted as it has been digitally signed. It can also hold all the user's claim, like authorization information, so that the service provider does not need to access the database to validate user roles and permissions for each request; data is extracted from the token.

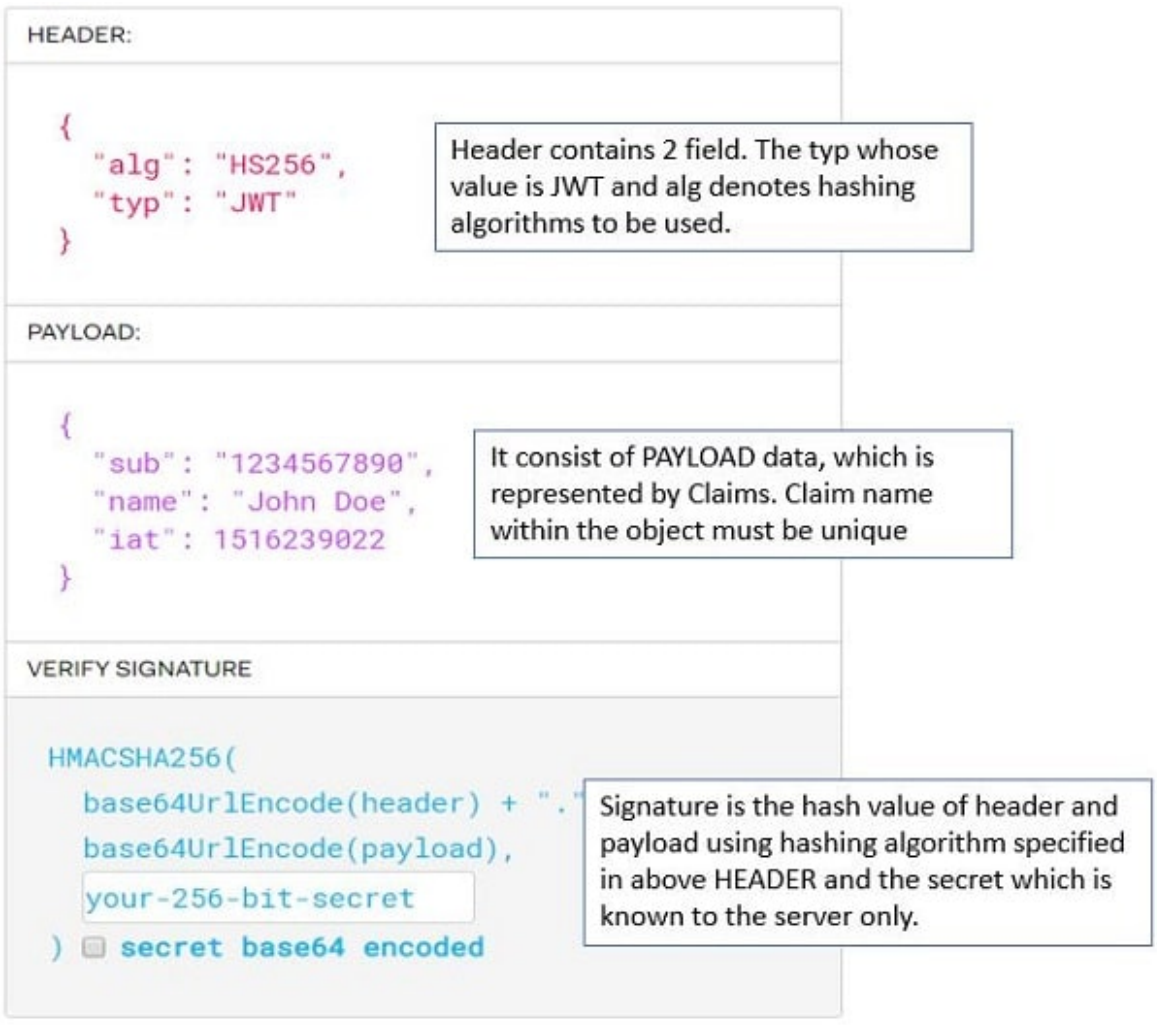
JWT Workflow



- Customers sign in by submitting their credentials to the provider.
- Upon successful authentication, it generates JWT containing user details and privileges for accessing the services and sets the JWT expiry date in payload.
- The server signs and encrypts the JWT if necessary and sends it to the client as a response with credentials to the initial request.
- Based on the expiration set by the server, the customer/client stores the JWT for a restricted or infinite amount of time.
- The client sends this JWT token in the header for all subsequent requests.
- The client authenticates the user with this token. So we don't need the client to send the user name and password to the

server during each authentication process, but only once the server sends the client a JWT.

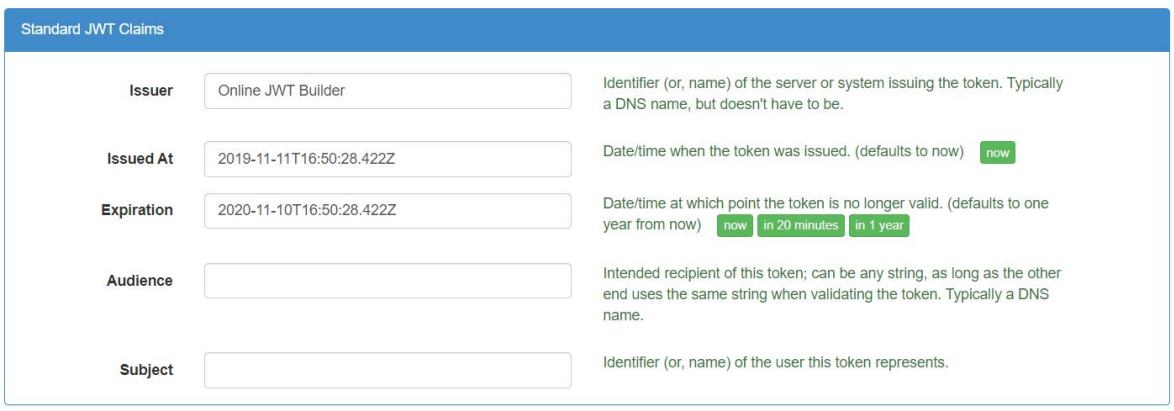
JWT Structure



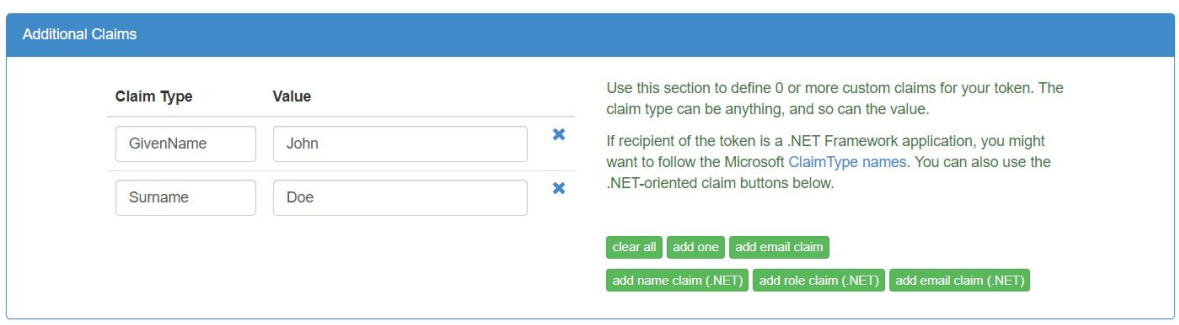
Create JWT Token Online

Will generate JWT Token by using [JWT Online Token Generator](#).

Provide the payload as given below:



Provide Claim data.



We'll have the following claims in the payload.

Sign the payload using the hashing algorithm.