

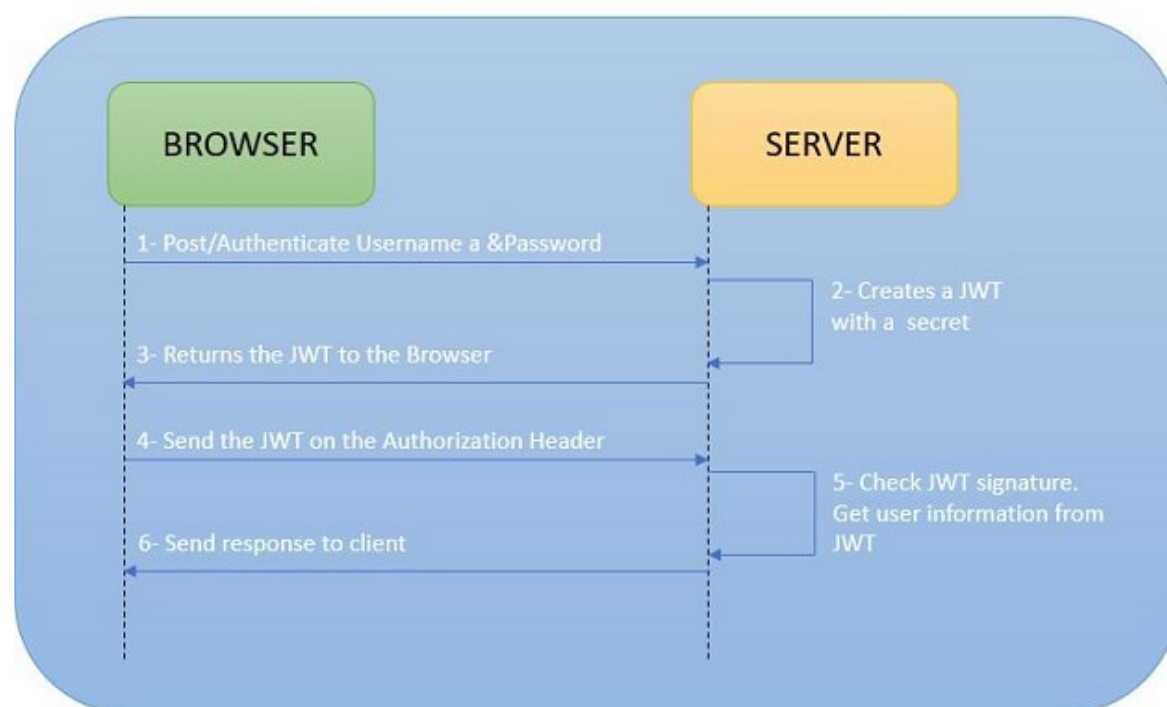
JWT (JSON Web Token)

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.

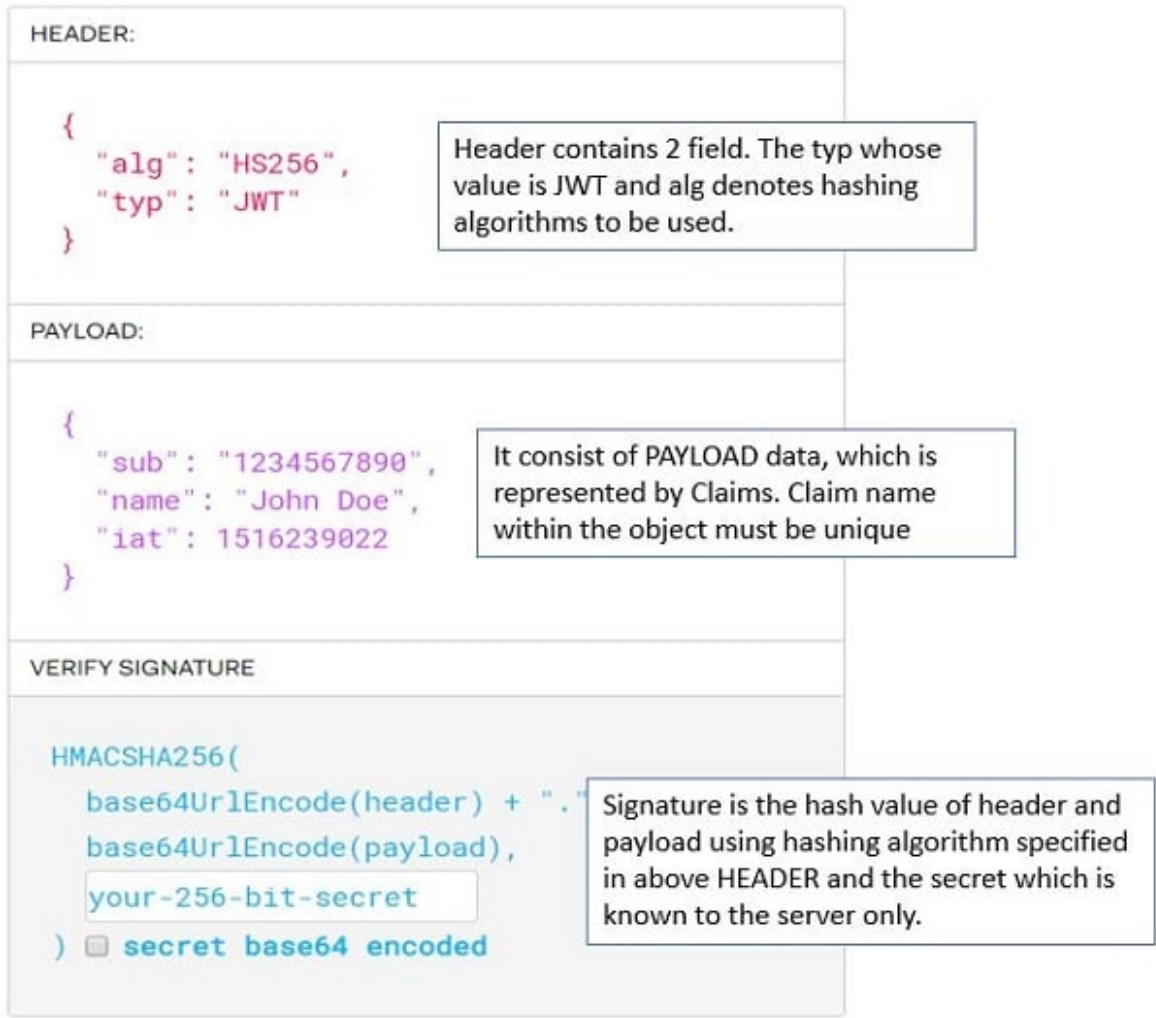
JWT Workflow



- 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:

Standard JWT Claims

Issuer	<input type="text" value="Online JWT Builder"/>	Identifier (or, name) of the server or system issuing the token. Typically a DNS name, but doesn't have to be.
Issued At	<input type="text" value="2019-11-11T16:50:28.422Z"/>	Date/time when the token was issued. (defaults to now) now
Expiration	<input type="text" value="2020-11-10T16:50:28.422Z"/>	Date/time at which point the token is no longer valid. (defaults to one year from now) now in 20 minutes in 1 year
Audience	<input type="text"/>	Intended recipient of this token; can be any string, as long as the other end uses the same string when validating the token. Typically a DNS name.
Subject	<input type="text"/>	Identifier (or, name) of the user this token represents.

Provide Claim data.

Additional Claims

Claim Type	Value	
<input type="text" value="GivenName"/>	<input type="text" value="John"/>	✕
<input type="text" value="Surname"/>	<input type="text" value="Doe"/>	✕

Use this section to define 0 or more custom claims for your token. The claim type can be anything, and so can the value.

If recipient of the token is a .NET Framework application, you might want to follow the Microsoft [ClaimType](#) names. You can also use the .NET-oriented claim buttons below.

clear all add one add email claim

add name claim (.NET) add role claim (.NET) add email claim (.NET)

We'll have the following claims in the payload.

Generated Claim Set (plain text)

```
{
  "iss": "Online JWT Builder",
  "iat": 1573491028,
  "exp": 1605027028,
  "aud": "",
  "sub": "",
  "GivenName": "John",
  "Surname": "Doe"
}
```

This section displays the claims that will be signed and base64-encoded into a complete JSON Web Token.

Sign the payload using the hashing algorithm.

Signed JSON Web Token

Key

password

8

HS512

Create Signed JWT

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJPbmtpbmUgSlIdUIEi1aWxkZXIiLCJpYXQiOiJlInzM0OTEwMjgsImV...

Copy JWT to Clipboard