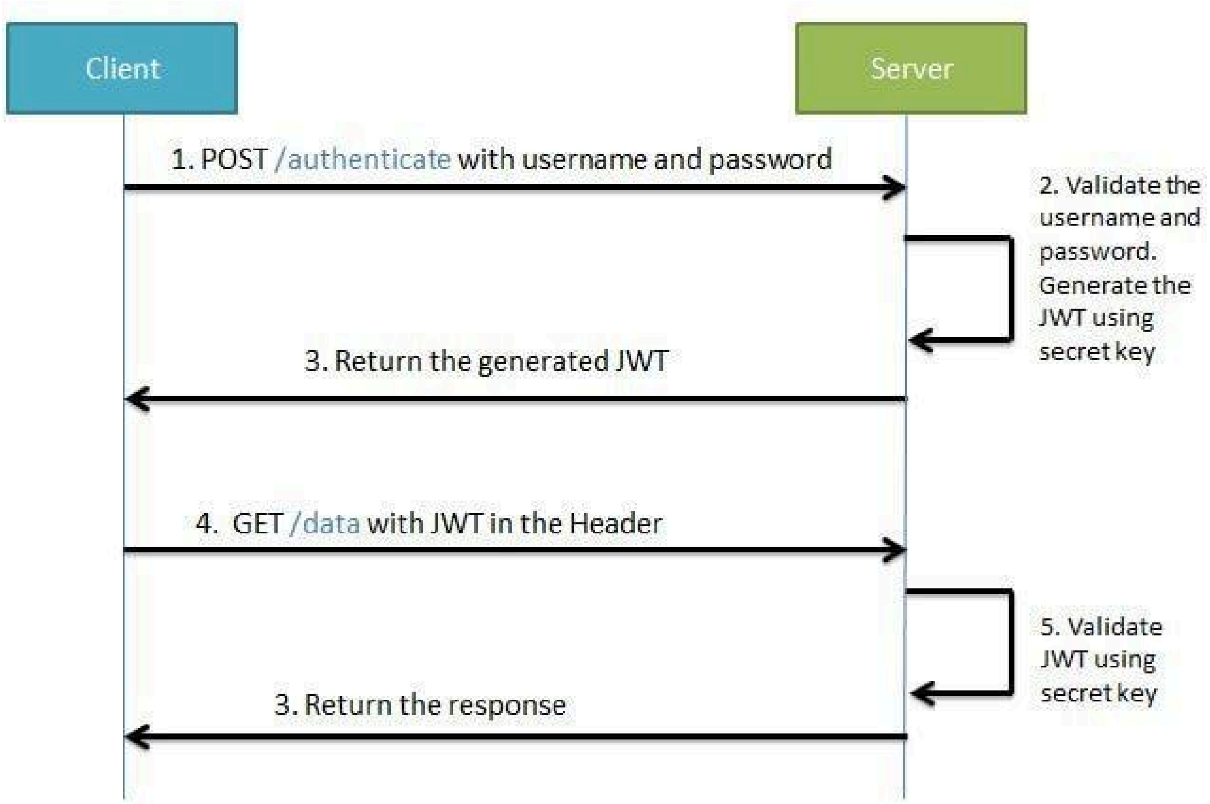
JWT

JWT stands for JSON Web Token. JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

This information can be verified and trusted because it is digitally signed. The client will need to authenticate with the server using the credentials only once.

During this time the server validates the credentials and returns the client a JSON Web Token(JWT). For all future requests the client can authenticate itself to the server using this JSON Web Token(JWT) and so does not need to send the credentials like username and password.

Workflow of how JWT is used


Client
Server
1. POST /authenticate with username and password
2. Validate the username and password. Generate the JWT using secret key
3. Return the generated JWT
4. GET /data with JWT in the Header
5. Validate JWT using secret key
3. Return the response

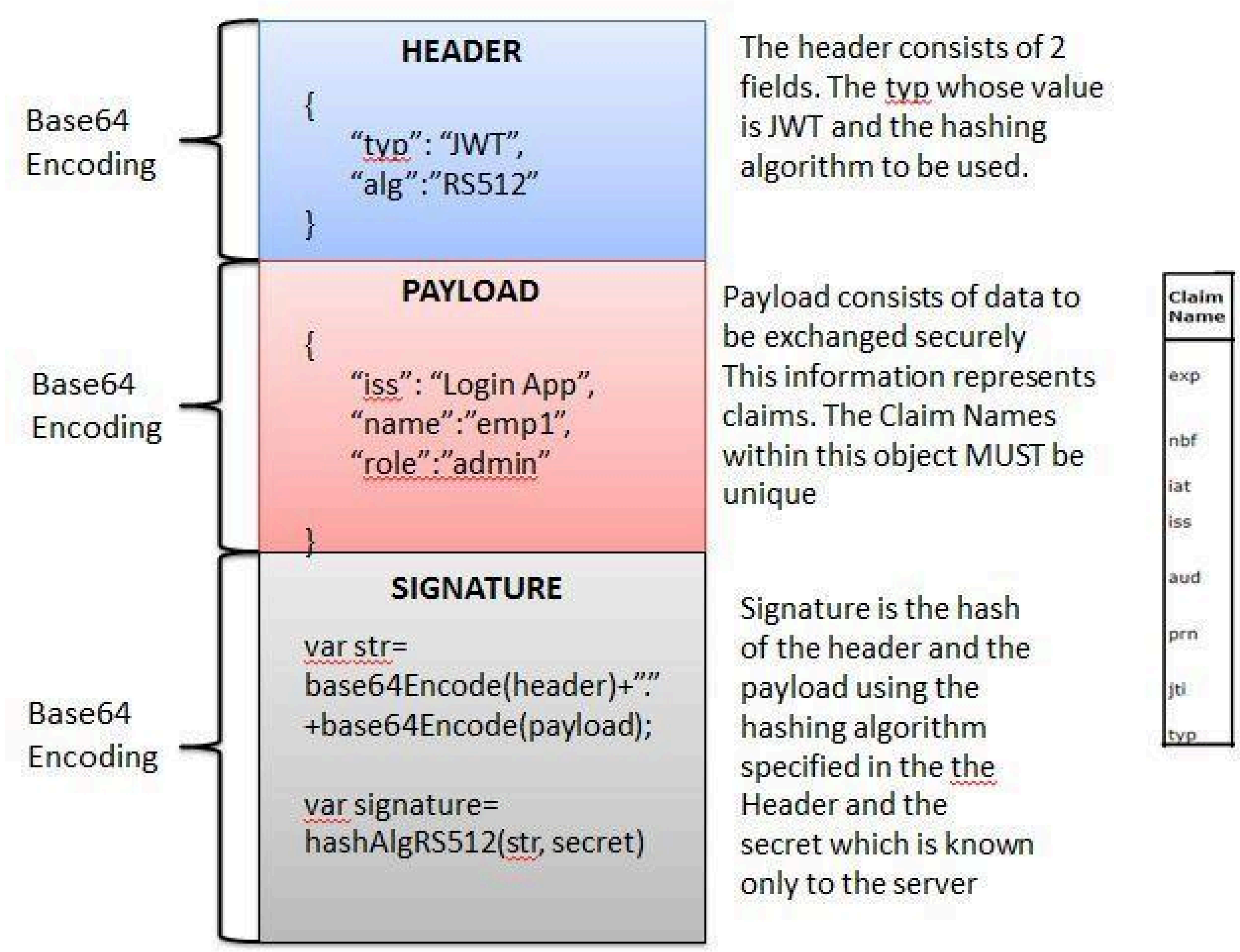
During the first request the client sends a POST request with username and password. Upon successful authentication the server generates the JWT sends this JWT to the client. This JWT can contain a payload of data. On all subsequent requests the client sends this JWT token in the header. Using this token the server authenticates the user. So we don't need the client to send the user name and password to the server during each request for authentication, but only once after which the server issues a JWT to the client. A JWT payload can contain things like user ID so that when the client again sends the JWT, you can be sure that it is issued by you, and you can see to whom it was issued.

Structure of JWT

JWT has the following format –**header.payload.signature**

HEADER
.
PAYLOAD
.
SIGNATURE

Structure of JWT-



```
{
    "typ": "JWT",
    "alg":"RS512"
}
```

The header consists of 2 fields. The typ whose value is JWT and the hashing algorithm to be used.

```
{
    "iss": "Login App",
    "name":"emp1",
    "role":"admin"

}
```

Payload consists of data to be exchanged securely This information represents claims. The Claim Names within this object MUST be unique

```
var str=
base64Encode(header)+"."
+base64Encode(payload);

var signature=
hashAlgRS512(str, secret)
```

Signature is the hash of the header and the payload using the hashing algorithm specified in the the Header and the secret which is known only to the server

Claim Name
exp
nbf
iat
iss
aud
prn
jti
typ

An important point to remember about JWT is that the information in the payload of the JWT is visible to everyone. So we should not pass any sensitive information like passwords in the payload. We can encrypt the payload data if we want to make it more secure. However we can be sure that no one can tamper and change the payload information. If this is done the server will recognize it.