

Sommaire

A- JSON Web Token	2
A.1. Structure du Token.....	2
A.2. La génération du Token.....	2
A.3. Pour la vérification du token.....	4
B- OAuth2	4
Qu'est-ce que OAuth?	4

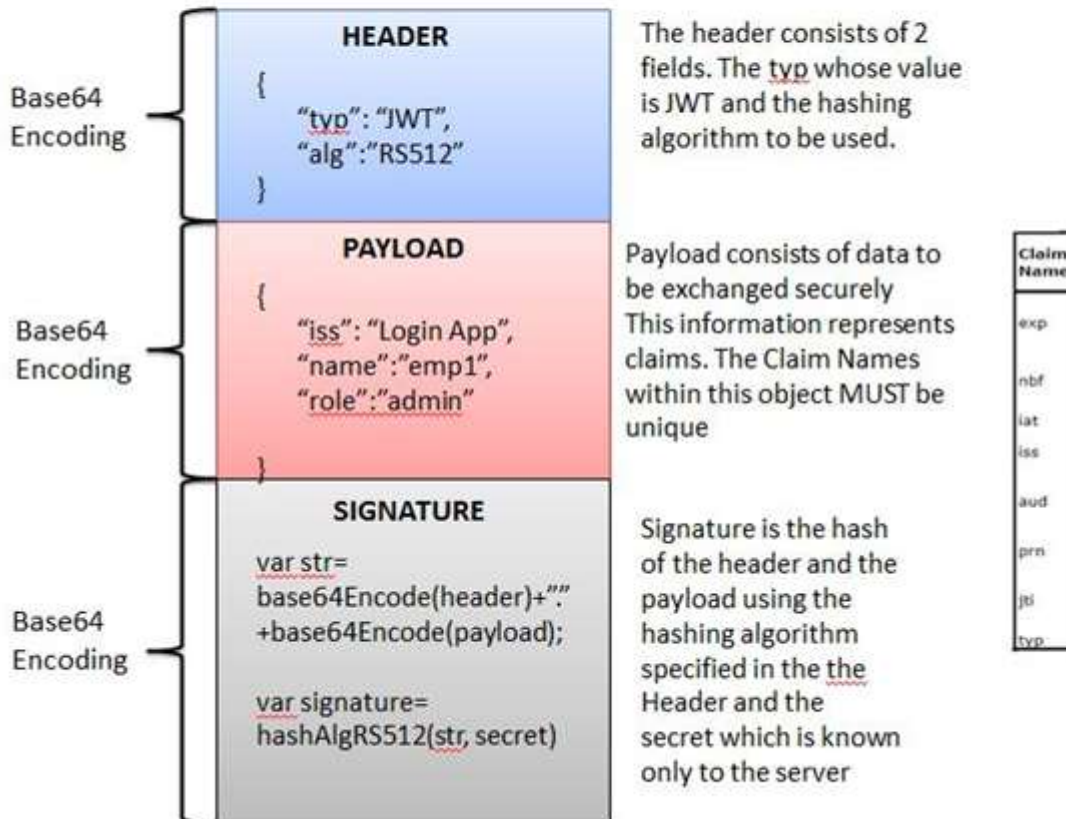
A- JSON Web Token

A.1. Structure du Token

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



Structure of JWT-



A.2. La génération du Token

Aller au lien : <http://jwtbuilder.jamiekurtz.com/> et suivre les étapes suivantes :

Standard JWT Claims

Issuer	<input type="text" value="Login"/>	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="2019-05-30T11:30:45.697Z"/>	Date/time when the token was issued. (defaults to now) now
Expiration	<input type="text"/>	Date/time at which point the token is no longer valid.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.

Additional Claims

Claim Type	Value	
<input type="text" value="idUser"/>	<input type="text" value="123"/>	✕
<input type="text" value="userName"/>	<input type="text" value="abbou"/>	✕

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)

Generated Claim Set (plain text)

```
{
  "iss": "Login",
  "iat": 1559215845,
  "exp": null,
  "aud": "",
  "sub": "",
  "idUser": "123",
  "userName": "abbou"
}
```

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

- Expiration is not a valid W3C date/time. Must be formatted as: YYYY-MM-DDThh:mm:ssZ

Signed JSON Web Token

Key 32 HS512 Create Signed JWT

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJMb2dpbiIsIm1hdCI6MTU1OTIxNTg0NSwiZXhwIjpudWxsLCJhdWQiOiIiLCJzdmIiOiIiLCJpZmVzZXIiOiIwMjM1LCJ1c2VybmFtZSI6ImFiYm91In0.sN5i09D5a0tj53Y6kRjFB_cn-a_UnmDZnA-RqEVNiijklq4kyvDQIngD-loW3P7-UIQFYqWa37Ckoh2q88lRQw
```

Copy JWT to Clipboard

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJMb2dpbiIsIm1hdCI6MTU1OTIxNTg0NSwiZXhwIjpudWxsLCJhdWQiOiIiLCJzdmIiOiIiLCJpZmVzZXIiOiIwMjM1LCJ1c2VybmFtZSI6ImFiYm91In0.sN5i09D5a0tj53Y6kRjFB_cn-a_UnmDZnA-RqEVNiijklq4kyvDQIngD-loW3P7-UIQFYqWa37Ckoh2q88lRQw

A.3. Pour la vérification du token

Aller au site <https://jwt.io/>

Encoded <small>PASTE A TOKEN HERE</small>	Decoded <small>EDIT THE PAYLOAD AND SECRET</small>					
<pre>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJMb2dpbiBBcHAIiLCJpYXQiOiE1NTYxMTc0MjksImV4cCI6bnVsbCwiYXVkIjoiaiwic3ViIjoiaiwizW1wSWQ0iJlYXAwMDEiLCJlbXB0YW11IjoiaZW1wbG95ZWUifQ.i8NikA2G-50nL7kBU3djsCPSSilqF6xLohAqqC6NjbFSsL_vKF95XxGdos16PhrA0RMghqgWDAP_h0S_v6z5Sg</pre>	<table border="1"><thead><tr><th>HEADER: ALGORITHM & TOKEN TYPE</th></tr></thead><tbody><tr><td><pre>{ "typ": "JWT", "alg": "HS512"}</pre></td></tr><tr><th>PAYLOAD: DATA</th></tr><tr><td><pre>{ "iss": "Login App", "iat": 1556117829, "exp": null, "aud": "", "sub": "", "empId": "emp001", "empName": "employee"}</pre></td></tr><tr><th>VERIFY SIGNATURE</th></tr></tbody></table>	HEADER: ALGORITHM & TOKEN TYPE	<pre>{ "typ": "JWT", "alg": "HS512"}</pre>	PAYLOAD: DATA	<pre>{ "iss": "Login App", "iat": 1556117829, "exp": null, "aud": "", "sub": "", "empId": "emp001", "empName": "employee"}</pre>	VERIFY SIGNATURE
HEADER: ALGORITHM & TOKEN TYPE						
<pre>{ "typ": "JWT", "alg": "HS512"}</pre>						
PAYLOAD: DATA						
<pre>{ "iss": "Login App", "iat": 1556117829, "exp": null, "aud": "", "sub": "", "empId": "emp001", "empName": "employee"}</pre>						
VERIFY SIGNATURE						

B- OAuth2

OAuth (Open Authorization) est un moyen simple de publier et d'interagir avec des données protégées. Il s'agit d'un standard ouvert pour l'authentification et l'autorisation basées sur des jetons sur Internet. Il permet aux services tiers, tels que Facebook, d'utiliser les informations de compte d'un utilisateur final sans révéler le mot de passe de cet utilisateur.

La spécification OAuth décrit cinq autorisations pour l'acquisition d'un jeton d'accès:

- Authorization code grant
- Implicit grant
- Resource owner credentials grant
- Client credentials grant
- Refresh token grant

Qu'est-ce que OAuth?

Prenons le cas d'utilisation de Quora. Allez sur **Quora.com**. Si vous êtes un nouvel utilisateur, vous devez vous inscrire. Vous pouvez vous inscrire en utilisant un compte google ou facebook. Ce faisant, vous autorisez Google ou Facebook à autoriser Quora à accéder à vos informations de profil avec Quora. Cette autorisation est faite en utilisant OAuth. Ici, vous n'avez en aucun cas partagé vos informations d'identification avec Quora.



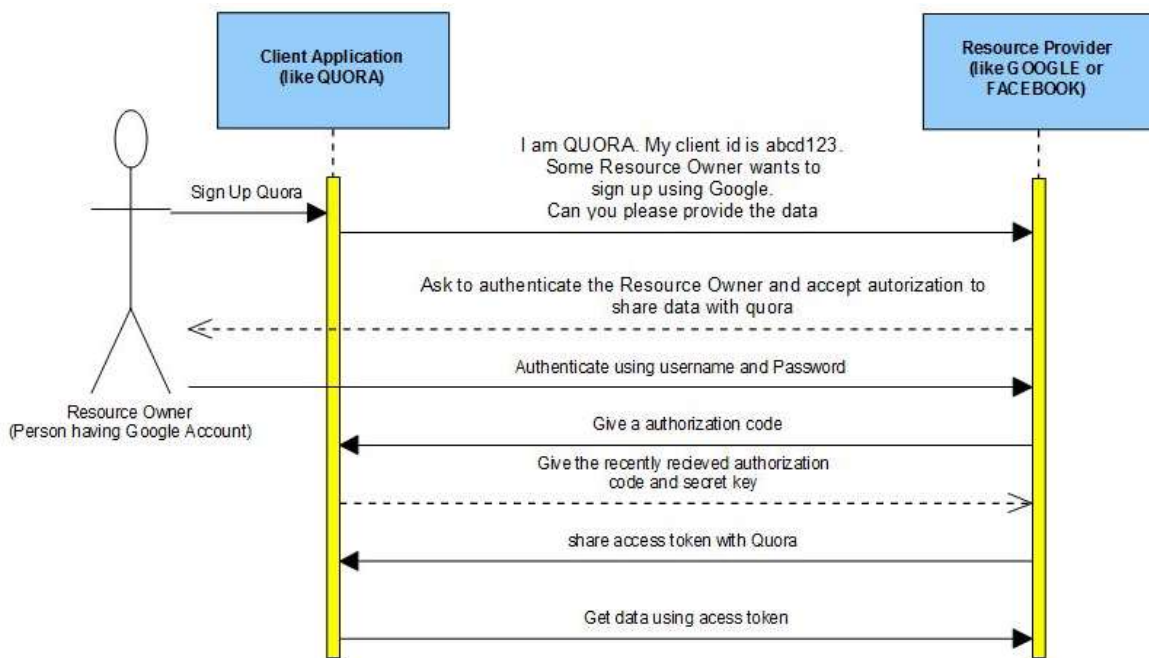
Dans l'exemple ci-dessus de Quora, nous avons 3 acteurs:

- **Resource Owner** - Ceci est l'utilisateur qui veut s'inscrire en utilisant Quora.
- **Client Application** - Ce sera Quora
- **Resource Server** - Ce sera Gmail ou Facebook.
- **Authorization Server** - Le serveur de ressources héberge les comptes d'utilisateur protégés et le serveur d'autorisation vérifie l'identité de l'utilisateur, puis émet des jetons d'accès à l'application.

Obtenir le code d'autorisation

Le flux est le suivant:

- « Resource Owner » demandera au « Client Application » d'obtenir des données du « Resource Server ».
- « Resource Server » demande au « Resource Owner » de s'authentifier et d'autoriser le partage de données.
- Une fois l'authentification réussie, « Resource Server » partage un code d'autorisation avec « client application ».



Récupération et utilisation du jeton d'accès

« **Client Application** » utilisant le code d'autorisation et la clé secrète demande le token à partir du « **Resource Server** ».

« **Resource Server** » partage le token avec « **Client Application** ».

À l'aide du token, « **Client Application** » peut maintenant obtenir les données JSON requises du « **Resource Server** ».

