

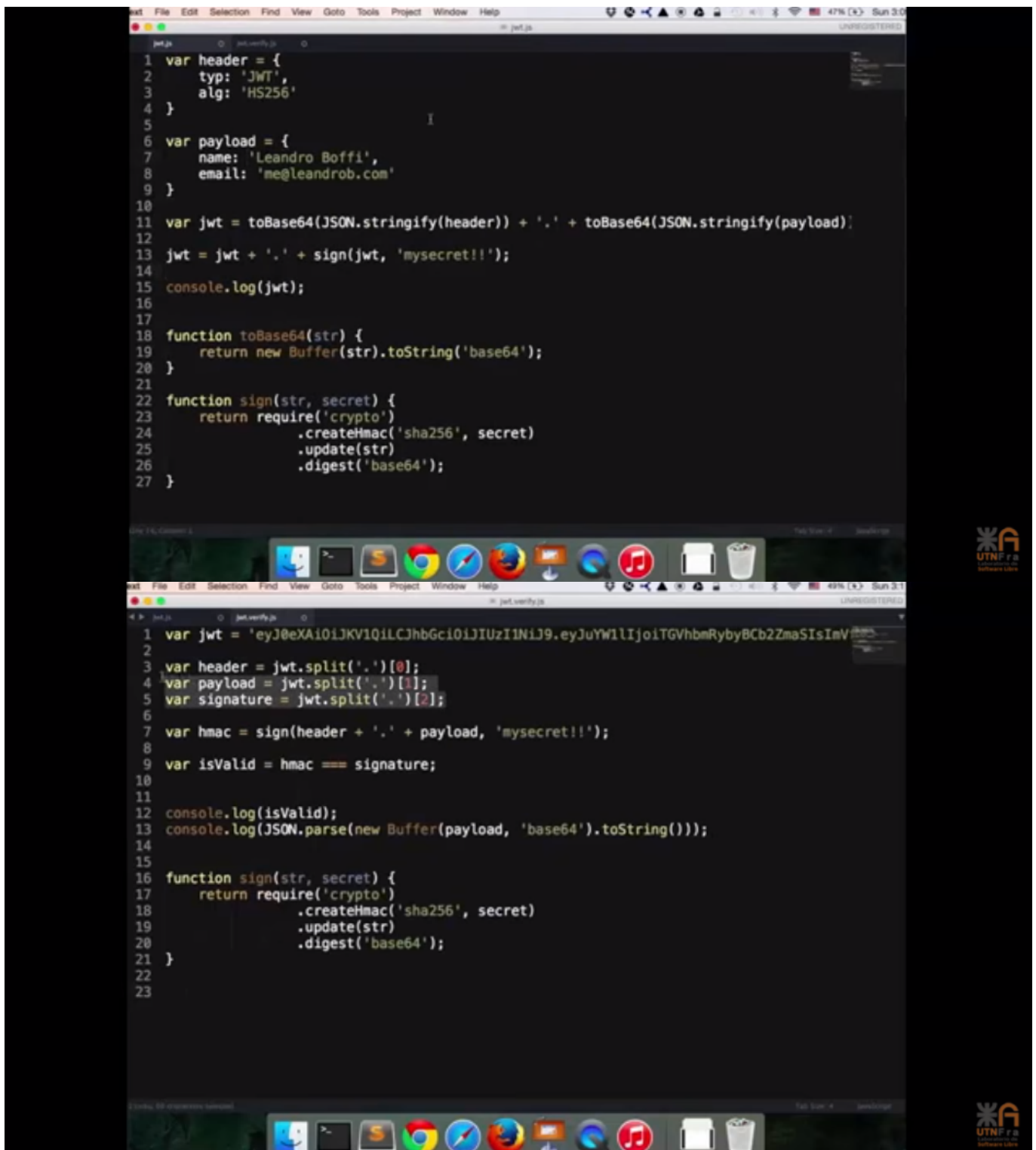
En este artículo vamos a revisar apuntes sobre [JWT JSON Web Token](#)

- Clave simetrica: es yo encripto y desencripto con la misma clave
- Clave asimetrica: yo tengo mi clave privada y las otras apps tienen una clave publica
- JWS Signature
- JWE Encription

Json Web Tokens (JWT)

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIjOiJlcmM  
zQlNjc4OTAzIm5hbWUiOiJKb2huc2h1IERvZSIsImFkbWludjpw  
OcnVlFQ.eoaDVGTCIRdfxUZXiPs3f8FmJDKDE_VCQF  
XqKxpLsts
```



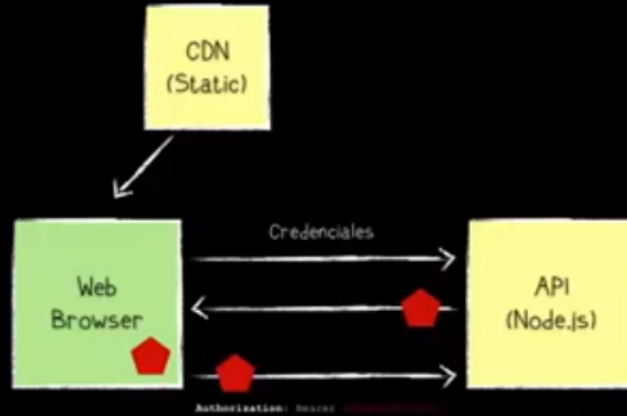


The image shows two screenshots of a code editor window. The top screenshot displays the code for generating a JWT token. The bottom screenshot displays the code for verifying a JWT token. The code is written in JavaScript and uses the 'crypto' module for signing and verification.

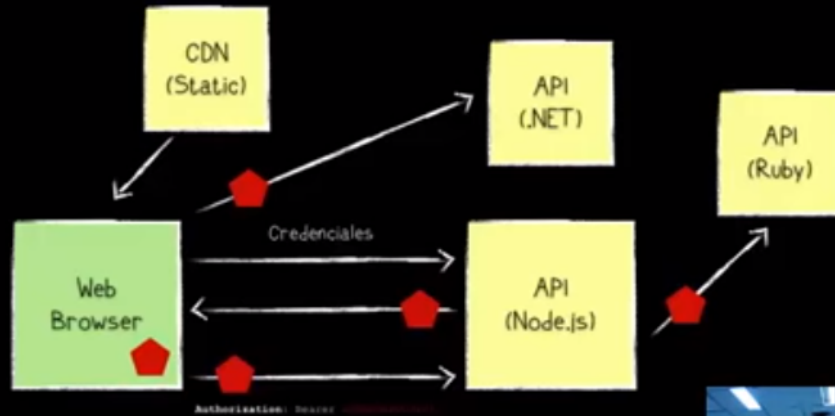
```
1 var header = {
2   typ: 'JWT',
3   alg: 'HS256'
4 }
5
6 var payload = {
7   name: 'Leandro Boffi',
8   email: 'me@leandrob.com'
9 }
10
11 var jwt = toBase64(JSON.stringify(header)) + '.' + toBase64(JSON.stringify(payload));
12
13 jwt = jwt + '.' + sign(jwt, 'mysecret!!');
14
15 console.log(jwt);
16
17
18 function toBase64(str) {
19   return new Buffer(str).toString('base64');
20 }
21
22 function sign(str, secret) {
23   return require('crypto')
24     .createHmac('sha256', secret)
25     .update(str)
26     .digest('base64');
27 }
```

```
1 var jwt = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1eW11Ijo1TGhbmRybyBCb2ZmaSIsImVudCI6ImVudCJ9';
2
3 var header = jwt.split('.')[0];
4 var payload = jwt.split('.')[1];
5 var signature = jwt.split('.')[2];
6
7 var hmac = sign(header + '.' + payload, 'mysecret!!');
8
9 var isValid = hmac === signature;
10
11
12 console.log(isValid);
13 console.log(JSON.parse(new Buffer(payload, 'base64').toString()));
14
15
16 function sign(str, secret) {
17   return require('crypto')
18     .createHmac('sha256', secret)
19     .update(str)
20     .digest('base64');
21 }
22
23
```

Token-Based Auth



Token-Based Auth



Json Web Tokens (JWT)

[Docs] [txt|pdf|xml] [Tracker] [WG] [Email] [Diff1] [Diff2] [Nits] [IPR]
Versions: (draft-jones-json-web-token) 00 01
02 03 20 21 22 23 24 25 26 27 28 29
30 31 32

OAuth Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 12, 2015

M. Jones
Microsoft
J. Bradley
Ping Identity
N. Sakimura
NRI
December 9, 2014

JSON Web Token (JWT)
draft-ietf-oauth-json-web-token-32

Abstract

JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. The claims in a JWT are encoded as a JavaScript Object Notation (JSON) object that is used as the payload of a JSON Web Signature (JWS) structure or as a



Para mi Conciencia

- JWS/JWE usan JWA que soporta
 - RS256 (RSASSA-PKCS-v1_5 usando SHA-256)
 - ES256 (ECDSA usando SHA-256)
- JWT incluye claims reservados en el payload
 - exp: Expiration
 - nbf: Not Before
 - iat: Issued At
 - aud: Audience



Video

Adjunto PDF Oficial de JWT Cookbook: [jwt-handbook.pdf](#)

Julio Pari (IT Architect IBM)



Si te ha interesado este artículo y deseas un apoyo o asesoría en algún requerimiento, envíame un mensaje a: (info@juliopari.com) o sino a través de LinkedIn:
<https://www.linkedin.com/in/juliopari/>
