



DIPLOMADO

Desarrollo de sistemas con tecnología Java

Módulo 11

Tokens JWT

Mtro. Alfonso Gregorio Rivero Duarte



1. JWT



1.1 Introducción

Introducción

¿Alguna vez has estado en un evento en el que necesitabas presentar un documento de identificación para demostrar que realmente fuiste la persona que compró el boleto?

El acto de solicitar un documento de identificación es una forma de autenticación para que usted reciba la autorización para ingresar.

En la web, este proceso funciona de manera similar.

- Para realizar solicitudes de algunos servicios o acceder a determinadas páginas, deberá identificarse de alguna forma y esta identificación deberá ser segura y única.

¿Qué es un Token?

Actualmente, escuchamos mucho la palabra token relacionada con NFT (acrónimo de “non-fusible tokens”), metaverso, criptomonedas, etc. Sin embargo, fuera de este medio, un token es una firma digital, una clave.

Cuando abres una cuenta bancaria, debes definir una contraseña y tus datos personales.

Estos datos se convierten en una firma digital que lo identificarás de manera única para ese banco y cada vez que accedas a tu banco e ingreses tu contraseña y datos personales, el banco comprenderá y confirmará que tú es el usuario que inició la sesión.

Similar a ingresar el evento cuando presentamos nuestro documento de identidad.

¿Qué es un Token?

Existen varios algoritmos y estándares que transforman tu información en un token, es decir, una clave de autenticación única, que tiene sentido para el servicio o la aplicación a la que intentas acceder en ese momento.

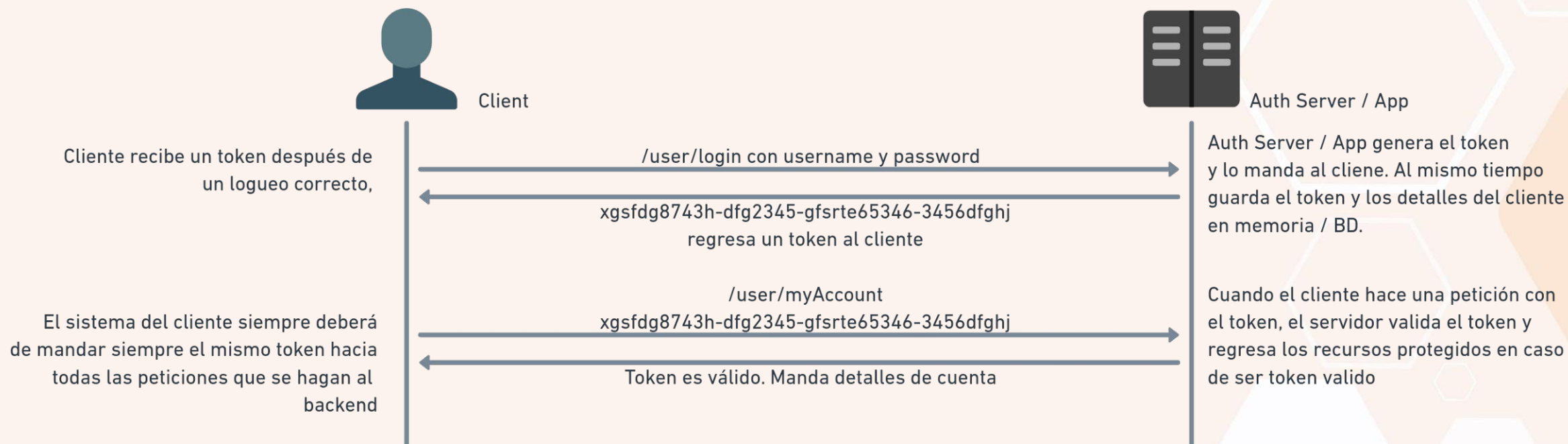
Uno de estos estándares es JWT, que es seguro porque permite la autenticación entre las dos partes a través de un token firmado.



Rol de Tokens

Un token puede ser un string plan, o hasta un UUID, o puede ser un JSON de tipo JSON Web Token (JWT) que se genera cuando se autentica un usuario por primera vez en un sistema

En cada request a algún recurso restringido, el cliente manda el token en el query string o en el header de autenticación.
El servidor valida el token y si es válido, regresa el recurso solicitado al cliente



Ventajas de los tokens

Token nos ayuda a no compartir las credenciales para cada solicitud. Es un riesgo para la seguridad enviar credenciales a través de la red con frecuencia.

Los tokens se pueden invalidar durante cualquier actividad sospechosa sin invalidar las credenciales del usuario.

Se pueden crear tokens con una vida útil corta.

Los tokens se pueden utilizar para almacenar información relacionada con el usuario, como roles/autoridades, etc.

Ventajas de los tokens

Reutilizabilidad

- Podemos tener muchos servidores separados, ejecutándose en múltiples plataformas y dominios, reutilizando el mismo token para autenticar al usuario.

Sin estado, más fácil de escalar. El token contiene toda la información para identificar al usuario, eliminando la necesidad del estado de la sesión. Si usamos un balanceador de carga, podemos pasar el usuario a cualquier servidor, en lugar de estar vinculados al mismo servidor en el que iniciamos sesión.

Ya usamos tokens en CSRF y JSESSIONID

- El token CSRF protegió nuestra aplicación de los ataques CSRF
- JSESSIONID es el token predeterminado generado por Spring Security que nos ayudó a no compartir las credenciales con el backend cada vez.

¿Qué es JWT?

Un JWT es un estándar para la autenticación y el intercambio de información definido por RFC7519.

Es posible almacenar objetos JSON de forma segura y compacta. Este token es un código **Base64** y se puede firmar con un par de claves secretas o privadas/públicas.

Los tokens firmados pueden verificar la integridad de la información que contienen, a diferencia de los tokens cifrados que ocultan esta información.

Si un JWT está firmado por un par de claves pública/privada, la empresa certifica que la parte que tiene la clave privada está realmente firmada.

¿Cuándo y dónde puedo usar JWT?

Se puede utilizar, por ejemplo, en un escenario de autorización.

Una vez que el usuario haya iniciado sesión, puede ver cada solicitud y verificar que incluye el JWT, lo que le permite acceder a rutas, servicios y otros recursos.

Otro escenario para el uso de JWTs es el intercambio de información porque, una vez firmado, es posible estar seguro de que los remitentes son quienes dicen ser.

Además, podemos identificar si el contenido de la empresa ha cambiado o no debido a la composición de un JWT.

¿Cómo surgió JWT?

Forma parte de una familia de especificaciones: la familia JOSE.

JOSE significa JSON Object Signing and Encryption, en inglés JSON Object Signing and Encryption.

JWT es parte de esta familia de especificaciones y representa el token. A continuación, puedes ver otras especificaciones de esta familia:

- JWT (JSON Web Tokens): representa el propio token;
- JWS (JSON Web Signature): representa la firma del token;
- JWE (JSON Web Encryption): representa la firma para el cifrado de tokens;
- JWK (JSON Web Keys): representa las claves para la firma;
- JWA (JSON Web Algorithms): representa los algoritmos para firmar el token.

¿Cómo surgió JWT?

- JWT significa JSON Web Token. Es una implementación de token que estará en formato JSON y diseñado para usarse con peticiones web.
- JWT es el token más común y favorito para uso de muchos sistemas que se usa básicamente por sus características especiales y ventajas ante otros tipos.
- Tokens JWT pueden ser usados en cualquiera de los escenarios de Autenticación / Autorización en conjunto con algún tipo de intercambio de información. Esto significa que el token puede contener cierta información del usuario (Claims)

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MzE1MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

1. Header
2. Payload
3. Signature (Optional)

Componentes básicos de un JSON Web Token

Un JWT tiene una estructura básica compuesta por encabezado, carga útil y firma. Estas tres partes están separadas por puntos (.).

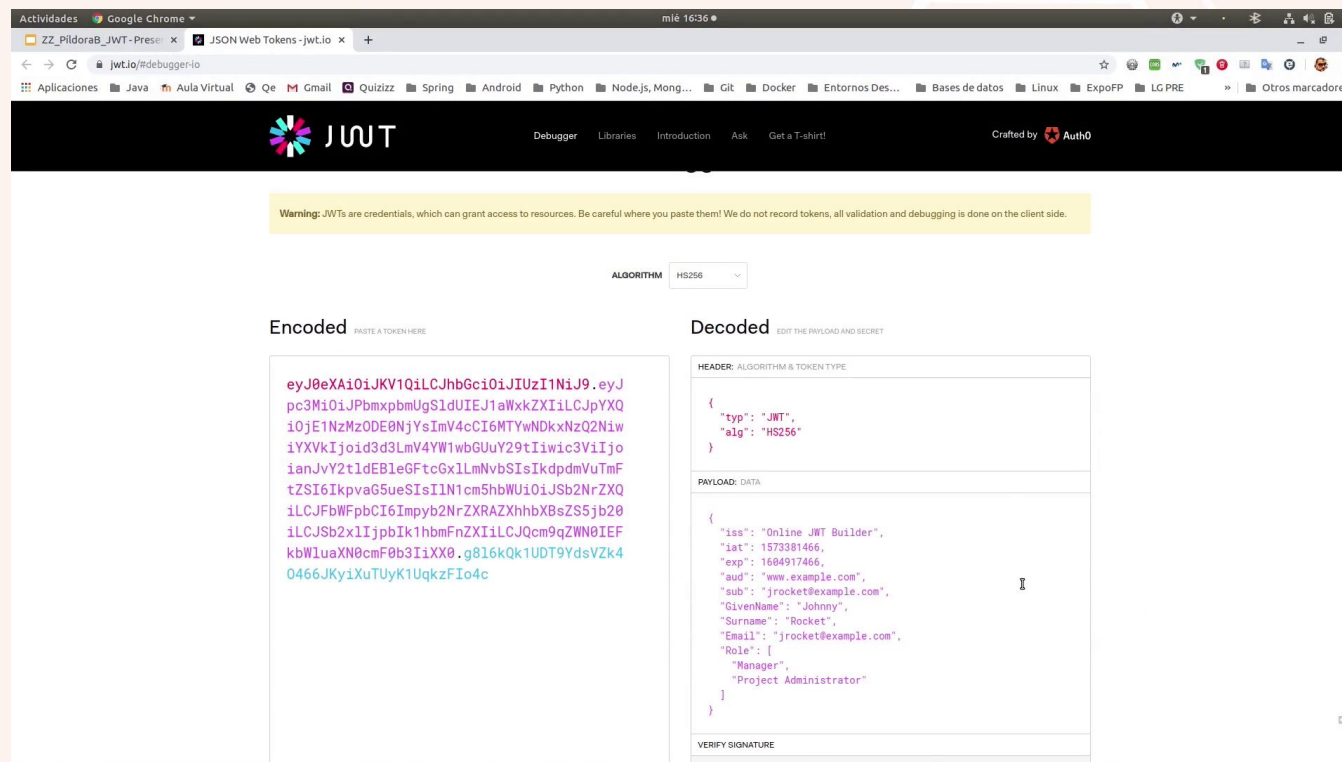
De esta forma quedaría algo así como: header.payload.signature.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.ikFGEvw-Du0f30vBaA742D_wqPA5BBHXgUY6wwqab1w
```

Componentes básicos de un JSON Web Token

Podemos utilizar un debugger online para decodificar ese token y visualizar su contenido. Si accedemos al mismo y pegamos dentro el texto completo, se nos mostrará lo que contiene:

<https://jwt.io/>



The screenshot shows the JWT.io online debugger interface. The browser address bar displays "jwt.io/#debugger-io". The page has a dark header with the JWT logo and navigation links: Debugger, Libraries, Introduction, Ask, and Get a T-shirt. A warning message states: "Warning: JWTs are credentials, which can grant access to resources. Be careful where you paste them! We do not record tokens, all validation and debugging is done on the client side." Below the warning, there is a dropdown menu for "ALGORITHM" set to "HS256". The interface is split into two main sections: "Encoded" and "Decoded". The "Encoded" section contains a long string of base64-encoded characters. The "Decoded" section shows the decoded token structure, including the header and payload. The header is {"typ": "JWT", "alg": "HS256"}. The payload is {"iss": "Online JWT Builder", "iat": 1573381466, "exp": 1684917466, "aud": "www.example.com", "sub": "jrocket@example.com", "GivenName": "Johnny", "Surname": "Rocket", "Email": "jrocket@example.com", "Role": ["Manager", "Project Administrator"]}. The "VERIFY SIGNATURE" section is empty.

Componentes básicos de un JSON Web Token

Podemos ver el contenido del token sin necesidad de saber la clave con la cual se ha generado, aunque no podremos validarlo sin la misma.

Como hemos dicho, un token tres partes:

- **Header:** son los encabezados del token donde pasamos básicamente dos piezas de información: el alg que informa qué algoritmo se usa para crear la firma y el type que indica el tipo de token.

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

Componentes básicos de un JSON Web Token

- **Payload:** donde aparecen los datos de usuario y privilegios, así como toda la información que queramos añadir, todos los datos que creamos convenientes.

```
{  
  "email" : "nombre@alura.com.br"  
  "password" : "HuEKW489!j445*"  
}
```

Componentes básicos de un JSON Web Token

La información se proporciona como pares key/value (clave-valor); las claves se denominan claims en JWT. Hay tres tipos diferentes de claims:

- ***Los claims registrados*** son los que figuran en el IANA JSON Web Token Claim Register y cuyo propósito se establece en un estándar. Algunos ejemplos son el emisor del token (iss, de issuer), el dominio de destino (aud, de audience) y el tiempo de vencimiento (exp, de expiration time). Se utilizan nombres de claim cortos para abreviar el token lo máximo posible.

Componentes básicos de un JSON Web Token

- ***Los claims públicos*** pueden definirse a voluntad, ya que no están sujetos a restricciones. Para que no se produzcan conflictos en la semántica de las claves, es necesario registrar los claims públicamente en el JSON Web Token Claim Register de la IANA o asignarles nombres que no puedan coincidir.
- ***Los claims privados*** están destinados a los datos que intercambiamos especialmente con nuestras propias aplicaciones. Si bien los claims públicos contienen información como nombre o correo electrónico, los claims privados son más concretos. Por ejemplo, suelen incluir datos como identificación de usuario o nombre de departamento. Al nombrarlos, es importante asegurarse de que no vayan a entrar en conflicto con ningún claim registrado o público.

Componentes básicos de un JSON Web Token

Todos los claims son opcionales, por lo que no es obligatorio utilizar todos los claims registrados. En general, el payload puede contener un número ilimitado de claims, aunque es aconsejable limitar la información del JWT al mínimo. Cuanto más extenso sea el JWT, más recursos necesitará para la codificación y la decodificación.

Here are the base claims that the ColdBox Security JWT token creates for you automatically:

- **Issuer (iss)** - The issuer of the token (defaults to the application's base URL)
- **Issued At (iat)** - When the token was issued (Unix timestamp)
- **Subject (sub)** - This holds the identifier for the token (defaults to user id)
- **Expiration time (exp)** - The token expiry date (Unix timestamp)
- **Unique ID (jti)** - A unique identifier for the token (md5 of the sub and iat claims)
- **Scopes (scope)** - A space-delimited string of scopes attached to the token
- **Refresh Token (cbsecurity_refresh)** - If you use refresh tokens, this custom claim will be added to the payload.

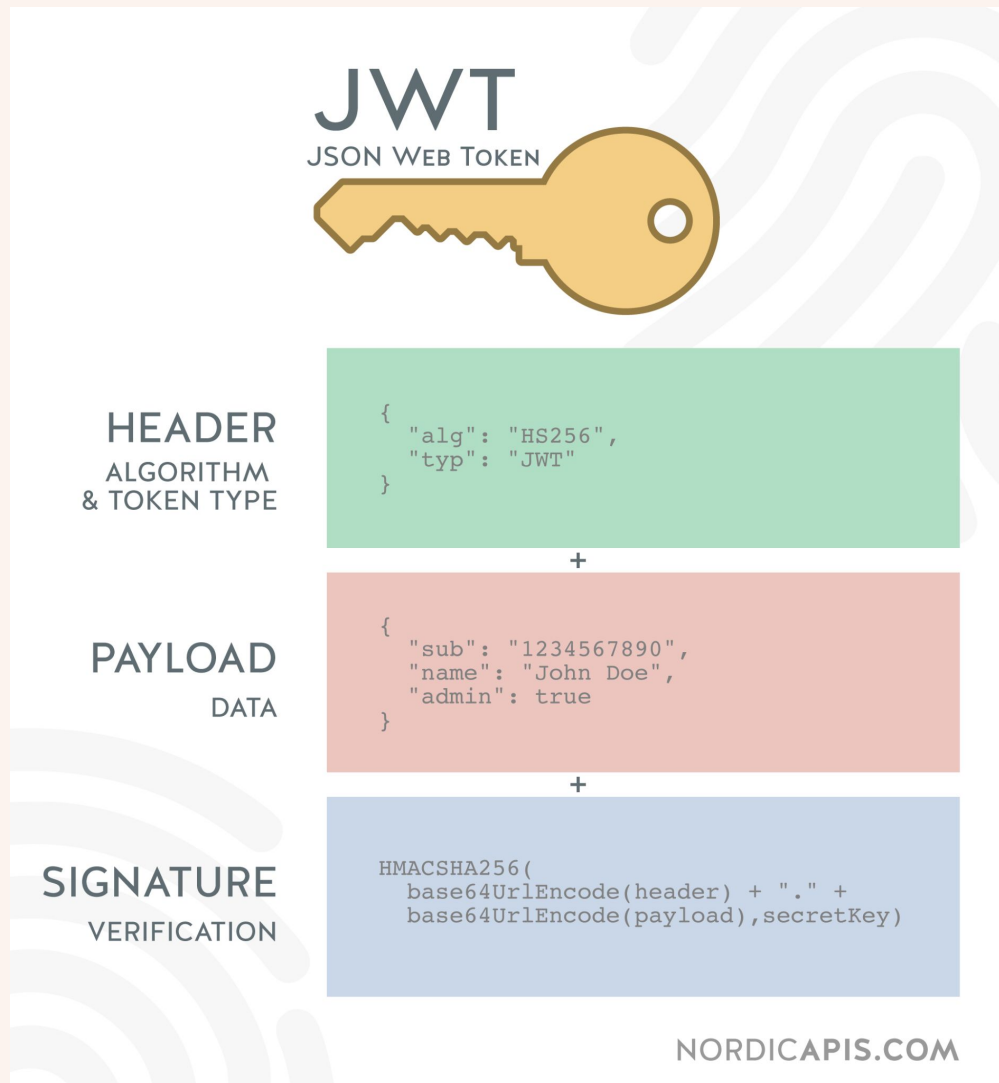
Componentes básicos de un JSON Web Token

- **Signature:** La firma del token (firma) está compuesta por la codificación del encabezado y el payload más una clave secreta y es generada por el algoritmo especificado en el encabezado.

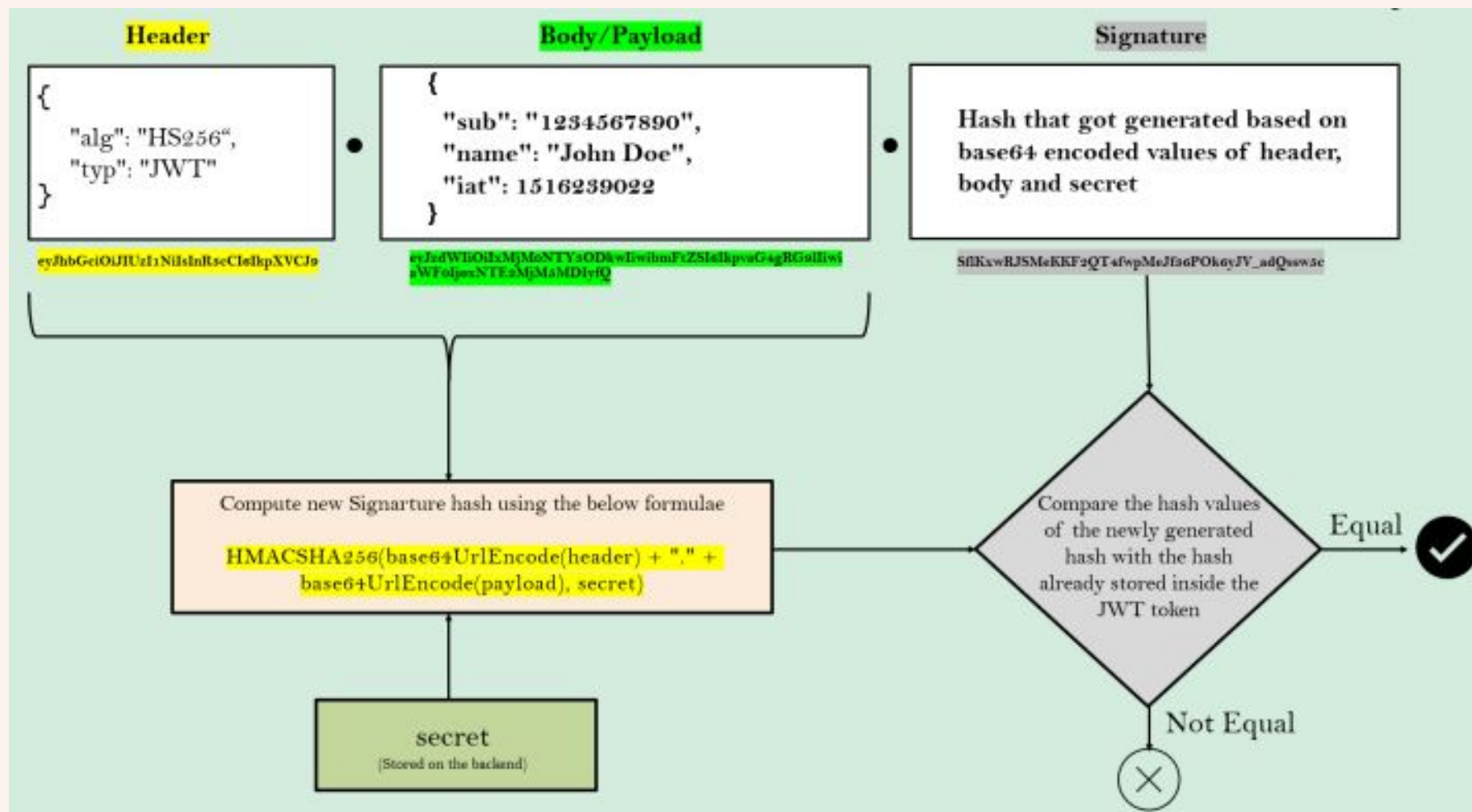
```
HS256SHA256(  
    base64UrlEncode(header) + "." + base64UrlEncode(payload), secre
```

El resultado son tres cadenas separadas por puntos que se pueden usar fácilmente en entornos HTML y protocolos HTTP.

Componentes básicos de un JSON Web Token



Validación JWT



Ciclo de vida de un Token

Vamos a ver ahora el ciclo de vida de un token JWT, si lo queremos utilizar en el marco de un proceso de autenticación.

Como hemos visto, JWT no es un estándar de autenticación, sino que simplemente un estándar que nos permite hacer una comunicación entre dos partes de identidad de usuario. Con este proceso, además, podríamos implementar la autenticación de usuario de una manera que fuera relativamente segura.

Ciclo de vida de un token JWT



¿Qué aplicaciones le puedo dar?

Se utilizan para mantener sesión del lado del cliente: permiten que el cliente pueda loguearse, y que el contenido de la sesión esté protegido mediante la autenticación por encriptación.

Securización de APIS: Cada request que el usuario realice a un servicio debe llevar el token para acceder a un recurso restringido, en el caso de que el token no sea válido o haya expirado, el usuario no podrá acceder al recurso.

¿Qué tokens puedo utilizar en mi API?

- **Token de Autorización:** Es generado por un servicio, generalmente cuando un usuario se autentica. Contiene información relacionada con el usuario.
- **Token de 'Refresco':** Es un token que se utiliza para renovar el token de autenticación, y evitar pedirle nuevamente al usuario sus credenciales. Tiene un tiempo de vida mayor al del token de autenticación.

mytoken.json

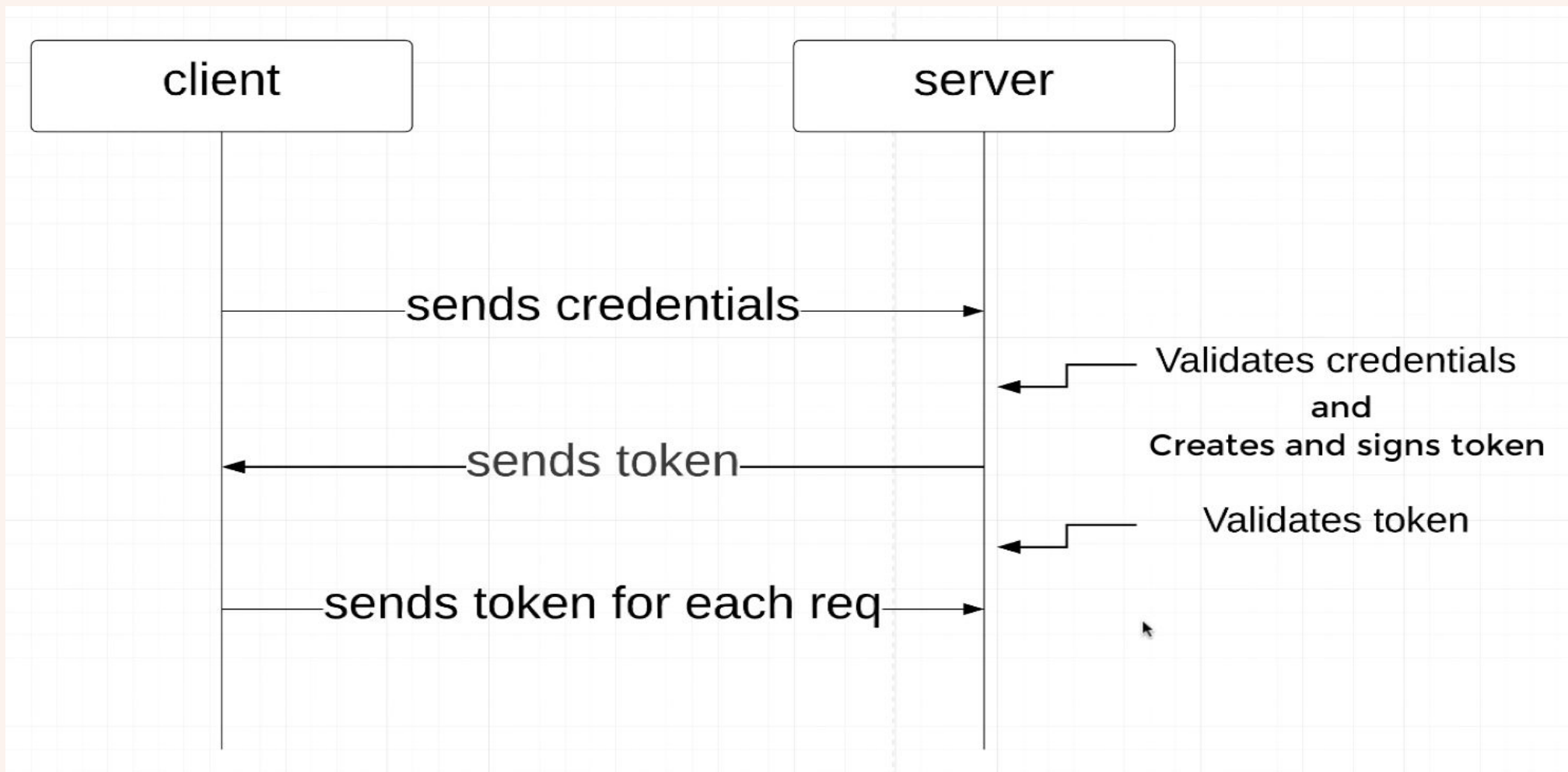
```
{
  "iat": 1569340662,
  "scope": "",
  "iss": "http://127.0.0.1:56596/",
  "sub": 123,
  "exp": 1569344262,
  "jti": "12954F907C0535ABE97F761829C6BD11"
}
```

myRefreshToken.json

```
{
  "iat": 1569340662,
  "scope": "",
  "iss": "http://127.0.0.1:56596/",
  "sub": 2222,
  "exp": 1569344262,
  "jti": "234234CDDEEDD",
  "cbsecurity_refresh" : true
}
```


JSON Web Token

- + Fast
- + Stateless
- + Used across many services
- Compromised Secret Key
- No visibility to logged in users
- Token can be stolen



JWT {JSON WEB TOKEN}

With Love By

@ Sec_v8

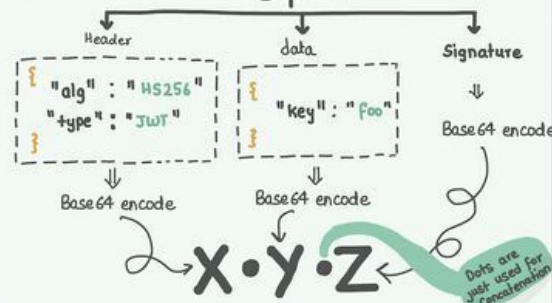
1 {"what": "JSON"}

* A file format to store data in key: value format



2 JWT Structure

3 parts



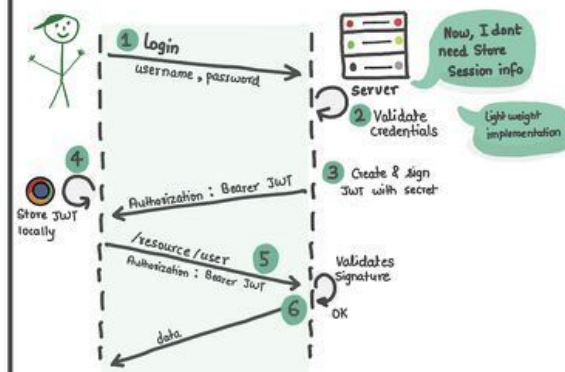
JWT = abc123.F32401.C3h72110A.1234abcd123 X Y Z

X: I am just Encoded Header

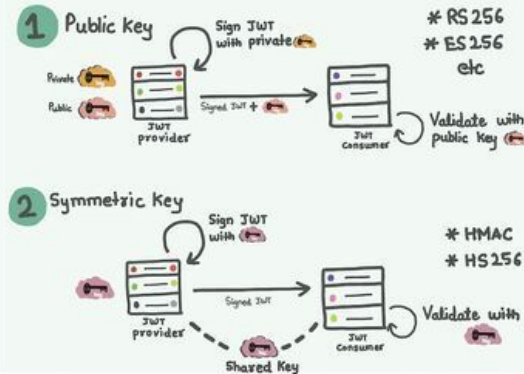
Y: I am just Encoded Data

Z: I am Encoded Signature (Read above below)

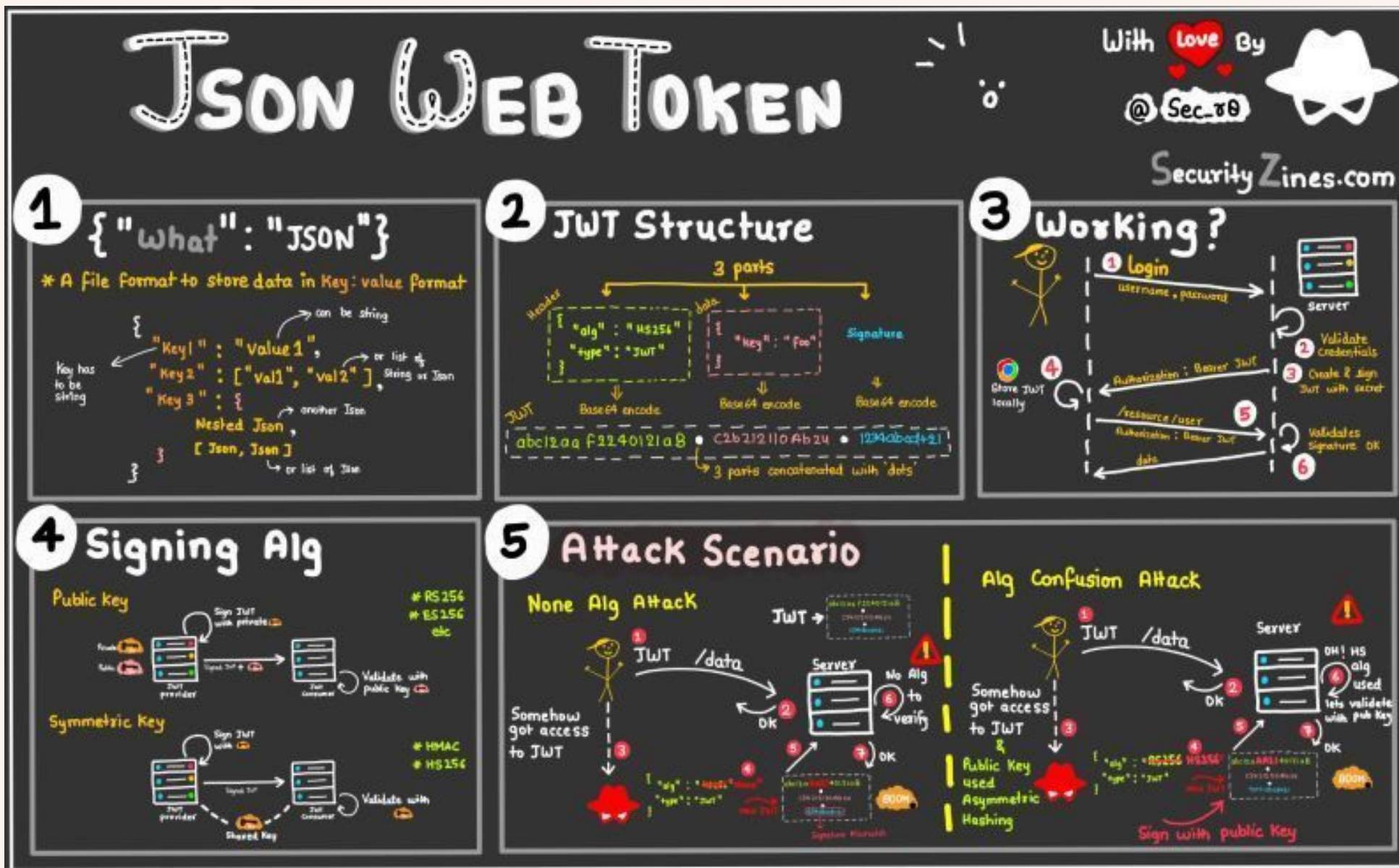
3 Working?



4 Signing Alg



SecurityZines.com In Collaboration with ByteByteGo



blog.amigoscode.com

JSON WEB TOKEN

STRUCTURE OF A JSON WEB TOKEN (JWT)



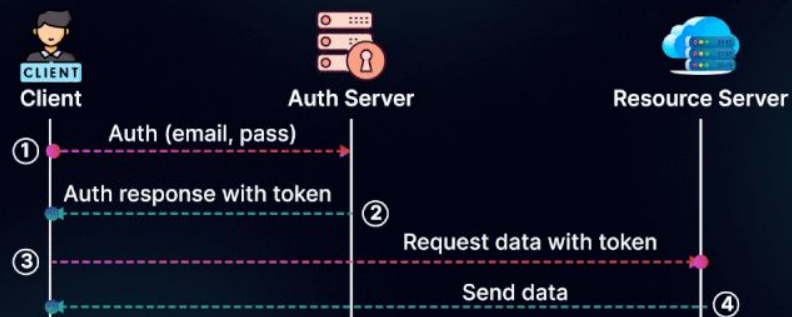
Header includes metadata info with two parts:
typ → Token Type
alg → hashing algorithm

Payload includes statement about an entity & additional data called **claims** which is of 3 types:

- PUBLIC
- REGISTERED
- PRIVATE

Signature is made up of base64 header, base64 payload, secret, & cryptographic algorithm.

BASIC JWT AUTH FLOW



amigoscode.com

DDTIC_DSJ_PLI_2024

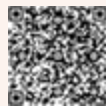
Contacto

Mtro. Alfonso Gregorio Rivero Duarte
Senior Data Manager - CBRE

devil861109@gmail.com

Tels: (+52) 55 289970 69

Redes sociales:



<https://www.linkedin.com/in/alfonso-gregorio-rivero-duarte-139a9225/>