

# JSON Web Token (JWT) overview

OIDC primer - a course on OpenID Connect

Credits: Roland Hedberg, Ioannis Kakavas

# JWT in the context of OIDC

The **OpenID Connect** protocol, as we have seen, is a simple REST/JSON- based identity federation protocol layered on OAuth 2.0.

It uses the **JSON Web Token (JWT)** and **JSON Object Signing and Encryption (JOSE)** formats both to represent security tokens and to provide security for other protocol messages.

In particular, to guarantee security:

- JWT is used to **represent information** (i.g. to describe a user identity)

# JSON Web Token (JWT)

JSON Web Token (JWT) is a **transport format** to represent a set of claims as JSON object.

It has the characteristic of being:

- **Versatile**, you can represent all information you need
- **Compact**, the representation does not add significant overhead
- **URL-safe**, the string obtained can be passed in URLs or HTTP headers

# JWT Representation

JWT token is a **sequence of URL-safe parts** separated by period '.' characters.

The main parts commonly are:

- Header
- Payload
- Digital signature

```
eyJhbGciOiJIUzI1NiIsImtpZCI6Imkwd25uIn0.eyJzdWUiOiJqb2UiLCJhdWQiOiJpbV9vaWNfY2xpZW50IiwianRpljoidWY5MFNlNHdzY0ZoY3RVVDZEdHZiMlIsImlzcyI6Imh0dHBzOlwvXC9sb2NhbGhvc3Q6OTAzMlhdCi6MTM5NDA2MDg1MywiZXhwIjoxMzk0MDYxMTUzLCJub25jZSI6ImU5NTdmZmJhLTlhNzgtNGVhOS04ZWNhLWFIOGM0ZWY5Yzg1NiIsImFOX2hhc2giOiJ3Zmd2bUU5VnhqQXVkc2w5bGM2VHFBIn0.lr4L-oT7Dji7Re0eSZDStAdOKHwSvjZfR-OpdWSOmsrw0QVeI7oalcehyKUFpPFDXDR0-RsEzqno0yek-_U-Ui5EM-yv0PiaUOmJK1U-ws_C-fCplUFSE7SK-TrCwaOow4_7FN5L4i4NAa_WqgOjZPlOT8o3kKyTkBL7GdITL8rEe4BDK8L6mLqHJrFX4SsEduPk0CyHJSykrqzYS2MEJIncocBBI4up5Y5g2BNEb0aV4VZwYjmrV9oOUC_yC1Fb4Js5Ry1t6P4Q8q_2ka5OcArlo188XH7IMgPA2GnwSFGHBhccjpxhN7S46ubGPXRBnsnrPx6Ruor2cl46d9ARQ
```

# JWT Representation - Header

Value	Value Decoded
<code>eyJhbGciOiJSUzI1NiIsImtpZCI6Imkwd25uIn0</code>	<pre>{   "alg": "RS256",   "kid": "i0wnn" }</pre>

# JWT Representation - Payload

Value	Value Decoded
eyJzdWliOiJqb2UiLCJhdWQiOiJpbV9vaWNfY2xpZW50liwianRpljoidWY5MFNlNHdzY0ZoY3RVVDZEdHZiMiIsImIzcyI6Imh0dHBzOlwvXC9sb2NhbGhvc3Q6OTAzMSIsImhhdCI6MTM5NDA2MDg1MywiZXhwljoxMzk0MDYxMTUzLCJub25jZSI6ImU5NTdmZmJhLTlhNzgtNGVhOS04ZWNhLWFIQGM0ZWY5Yzg1NiIsImF0X2hhc2giOiJ3Zmd2bUU5VnhqQXVkc2w5bGM2VHFBIn0	<pre>{   "sub": "joe",   "aud": "im_oic_client",   "jti": "uf90SK4wscFhctUT6Dtvb2",   "iss": "https://localhost:9031",   "iat": 1394060853,   "exp": 1394061153,   "nonce": "e957ffba-9a78-4ea9-ae8c4ef9c856",   "at_hash": "wfgvmE9VxjAudsl9lc6TqA" }</pre>

# JWT Representation - Digital signature

Value	Value Decoded
lr4L-oT7Dji7Re0eSZDstAdOKHwSvjZfR-OpdWSOmsr w0QVeI7oalcehyKUFpPFDXDR0-RsEzqno0yek-_U-Ui 5EM-yv0PiaUOmJK1U-ws_C-fCplUFSE7SK-TrCwaOo w4_7FN5L4i-4NAa_WqgOjZPlOT8o3kKyTkBL7GdITL8 rEe4BDK8L6mLqHJrFX4SsEduPk0CyHJSyKRqzYS2MEJ IncocBBI4up5Y5g2BNEb0aV4VZwYjmrV9oOUC_yC1F b4Js5Ry1t6P4Q8q_2ka5Ocarlo188XH7IMgPA2GnwS FGHBhccjphN7S46ubGPXRBNsnrPx6RuoR2cl46d9AR Q	N/A <i>(signature represented as specified by JOSE)</i>

# JSON Signing and Encryption (JOSE)

**JOSE** is a framework intended to provide a method to **securely transfer claims** (such as authorization information) between parties. The JOSE framework provides a collection of specifications to serve this purpose.

As by its name, JOSE deals with:

- Digital **signature** of claims
- **Encryption** of claims



# JOSE Signature

JOSE permit to **describe the algorithms** used for signing as defined in the JSON Web Algorithm (JWA) specification.

In the “alg” field JOSE specifies the signature method used:

- **None**: no digital signature
- **HS256**: HMAC w/ SHA-256 hash
- **RS256**: RSA PKCS v1.5 w/ SHA-256 hash
- **ES256**: ECDSA w/ P-256 curve and SHA-256 hash
- ...

# JOSE Encryption

JOSE permit to **describe** also **the algorithms** used for encryption as defined in the JSON Web Algorithm (JWA) specification.

In the “alg” field JOSE specifies the encryption method used:

- **None**: no digital signature
- **RSA1\_5**: RSA 1.5
- **RSA-OAEP-256**: RSA Optimal Asymmetric Encryption Padding 256 bit
- **A256KW**: AES Keywrap w/ 256 key
- **dir**: direct encryption
- **ECDH-ES+A256KW**: EC Diffie Hellman Ephemeral+Static key agreement w/ AES256 key
- ...

# JSON Web Key (JWK)

A JSON Web Key (JWK) is a JSON data structure that represents a **cryptographic key**.

Using a JWK rather than one or more parameters allows for a generalized key as input that can be applied to a number of different algorithms that may expect a different number of inputs.

# Summary

The JW\* standards permit to represent relevant information in a versatile, concise and URL safe manner.

1. **JWT** (*JSON Web Token*) is used to represent user information and ID tokens
2. **JWS** (*JSON Web Signature*) is used to sign the message
3. **JWE** (*JSON Web Encryption*) is used to describe encryption used for the message
4. **JWA** (*JSON Web Algorithms*) is used to describe the security algorithm used
5. **JWK** (*JSON Web Key*) is used to describe the key used by security algorithm

Q&A

Thanks for your attention!