JWT JWT Structure and Claims Reference

toolpilot.dev/cheatsheets/jwt-cheat-sheet/

JWT structure (header.payload.signature), standard claims, signing algorithms, and security best practices.

JWT Structure

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
.
eyJzdWIiOiIxMjM0IiwibmFtZSI6IkFsaWNlIiwiaWF0IjoxNjE2MjM5MDIyfQ
.
SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c

[Header].[Payload].[Signature]

Each part is Base64URL-encoded JSON (no padding =)
```

Header Fields

Field	Meaning	Example
alg	Signing algorithm	HS256, RS256, ES256
typ	Token type	"JWT"
kid	Key ID (for key rotation)	"key-2026-01"
cty	Content type (nested JWT)	"JWT"

Standard Payload Claims (RFC 7519)

Claim	Full name	Type	Description
iss	Issuer	string	Who issued the token
sub	Subject	string	User ID or entity
aud	Audience	string/array	Intended recipient
exp	Expiration	NumericDate	Unix timestamp when expires
nbf	Not Before	NumericDate	Valid from this time
iat	Issued At	NumericDate	When token was issued
jti	JWT ID	string	Unique token identifier

Signing Algorithms

Algorithm	Type	Key	Best for
HS256	HMAC-SHA256	Shared secret	Microservices, same issuer/verifier
RS256	RSA-SHA256	Private/public key	Public APIs, third-party verification
ES256	ECDSA-SHA256	Private/public key	Mobile, IoT (smaller key)
PS256	RSA-PSS-SHA256	Private/public key	High-security APIs
none	Unsigned	None	NEVER use in production

Security Best Practices

- Always verify signature — never trust unverified tokens
- Check exp claim before trusting payload
- Use short expiry (15min–1hr) with refresh tokens
- Never store sensitive data in payload (it's readable!)
- Reject alg: "none" tokens explicitly

- Use RS256/ES256 for public APIs (asymmetric is safer)
- Rotate secrets regularly and use kid for key versioning
- Store tokens in httpOnly cookies, not localStorage

Free developer cheat sheets at https://toolpilot.dev/cheatsheets/ — © 2026 toolpilot.dev