

HonoJsonWebKey

August 17, 2026

HonoJsonWebKey

Kind: Interface**Source:** `src/utils/jwt/jws.ts`**Part of:** [Utils](#)

`HonoJsonWebKey` represents a JSON Web Key with a required `kid` property. The key identifier lets JWT handling code select the matching key from a key set when processing a token.

Properties

Property	Type
<code>kid</code>	<code>string</code>

Diagram

```
graph LR
  JWT[JWT header] --> KID[kid]
  KID --> JWK[HonoJsonWebKey]
  JWK --> Verify[JWT verification]
```

Usage

```
import type { HonoJsonWebKey } from 'hono/utils/jwt/jws'

const signingKey = {
  kid: 'active-signing-key',
} as HonoJsonWebKey

const keys: HonoJsonWebKey[] = [signingKey]
```

```
const key = keys.find((candidate) => candidate.kid === 'active-signing-key')
```

AI Coding Instructions

- Set `kid` to the identifier expected by the JWT header.
- Keep `kid` values unique within a key set.
- Preserve the remaining JSON Web Key fields when passing keys to JWT signing or verification code.
- Match keys by `kid` before importing them for cryptographic operations.

How it works

`HonoJsonWebKey` is an exported TypeScript interface that extends the platform `JsonWebKey` type with an optional string `kid` property. [src/utils/jwt/jws.ts:21-25](#)

It has no runtime implementation, validation, error path, or direct side effect of its own; it is a type declaration. [src/utils/jwt/jws.ts:23-25](#)

It is one accepted member of `SignatureKey`, alongside a string or `CryptoKey`, so it can be passed to JWT signing and verification routines. [src/utils/jwt/jws.ts:27-37](#)

[src/utils/jwt/jws.ts:39-48](#) When such an object is imported as a signing or verification key, the code passes it to Web Crypto as a JWK with `crypto.subtle.importKey('jwk', ...)`; import failures are propagated by the awaited Web Crypto call.

[src/utils/jwt/jws.ts:66-75](#) [src/utils/jwt/jws.ts:95-104](#)

For `Jwt.sign`, if the supplied key is an object containing `alg`, its `alg` value selects the signing algorithm and its `kid` value is written into the JWT header.

[src/utils/jwt/jwt.ts:56-75](#)

For JWKS verification, `verifyWithJwks` accepts an array of `HonoJsonWebKey` values, requires the decoded token header to contain `kid`, and selects the first key whose `kid` strictly equals the header's `kid`. [src/utils/jwt/jwt.ts:197-216](#)

[src/utils/jwt/jwt.ts:228-251](#) If no matching key exists, it throws `JwtTokenInvalid`; if the selected key declares `alg` and it differs from the JWT header algorithm, it throws `JwtAlgorithmMismatch`. [src/utils/jwt/jwt.ts:248-256](#)

Relationships

- IMPORTS → `getRuntimeKey`
- IMPORTS → `decodeBase64`

- IMPORTS → `CryptoKeyUsage`
- IMPORTS → `JwtAlgorithmNotImplemented`
- IMPORTS → `utf8Encoder`