

## VC-JWT Functions

[Suggest Edits](#)

Verite conforms to the [VC-JWT Test Suite](#), which is the interoperability test suite for the Verifiable Credential 1.1 data model.

Verite relies on [did-jwt-vc](#) for the core VC-JWT signing and verifying, using the same preprocessing as is used in the [did-jwt-vc implementation of JWS-Test-Suite](#) to ensure conformance with the VC data model version 1.1 (including cross-compatibility with other verifiable encodings).

Examples of the direct use of these sign/verify functions are below.

## Create VC-JWT

```
JavaScript import { buildIssuer, CredentialPayload, encodeVerifiableCredential } from "verite"
```

```
const issuer = "did:key:z6MksGKh23mH2z2FpeND6WxJttd8TWhkTga7mtbM1x1z2M65m" const signer = buildIssuer( issuer,
    "f10465e2546027554ca1584ca53971dfc3bf44f9b287cb15b5732ad84adb4e63be5aa9b3df96e696f4eaa500ec0b58bf5dfdce59200571b44288cc9981279a238" ) const issuanceDate = new Date() const
Payload:CredentialPayload = { "@context": [ "https://www.w3.org/2018/credentials/v1", type: "VerifiableCredential", issuer: issuer, issuanceDate: issuanceDate, credentialSubject: { degree: { type:
    "BachelorDegree", name: "Baccalaureat en musiques numériques" } } } const result = await encodeVerifiableCredential(vcPayload, signer)
```

## Verify VC-JWT

```
JavaScript import { verifyVerifiableCredential } from "verite"
```

[illegible]