Create JWT Functions Plugin - V2.1.0

# Overview

This plugin enables you to create signed and unsigned JSON Web Tokens(JWTs). It supports signing and verification of JWTs using the following algorithms - RS256, RS384, RS512, PS256, PS384, PS512, ES256, ES384, ES512, HS256, HS384, HS512, decoding JWTs and generating HMAC SHA 256, 384 or 512 key generation.

Functions

1. createjwttokenps256withscs - Deprecated use createsignedjwt instead
2. createjwttokenrs256withscs - Deprecated use createsignedjwt instead
3. createjwttokenwithrs256 - Deprecated since version 2.0.0
4. createsignedjwt
5. createunsignedjwt
6. decodejwt
7. generatehmacshakey
8. verifyjwt

# Create Signed JWT - createsignedjwt()

This function returns a JWT signed with the specified signature algorithm.

It supports the following signing algorithms - RS256, RS384, RS512, PS256, PS384, PS512, ES256, ES384, ES512, HS256, HS384, HS512.

The provided Secure Credential Store (SCS) identified by the ‘scsKey’ must contain a ‘privatekey’ field holding the RSA [RS or PS signing algorithms] or ECDSA [ES signing algorithms] private key content without the BEGIN and END markers in PKCS8 format. For HS signing algorithms, the provided SCS must contain a ‘secretkey’ field holding the generated HMAC SHA key content.

Provide a text value for the ‘audience’ parameter to generate JWT with a single aud claim.

**Note: If expiry is provided with a null value or 0 it will be set to 3600 second as default.**

## Parameters

| **Sl No** | **Name** | **Type** | **Description** |
| --- | --- | --- | --- |
| 1 | kid (Optional) | Text | The value to identify which key was used in signing the JWT. |
| 2 | issuer (Optional) | Text | The value to identify the principal that issued the JWT. |
| 3 | subject (Optional) | Text | The value to identify the principal that is the subject of the JWT. |
| 4 | audience (Optional) | List of Text | The value to identify the recipients the JWT is intended for. |
| 5 | jti (Optional) | Text | The value to uniquely identify the generated JWT. Provide null to disable jti. |
| 6 | expiry (Optional) | Number(Integer) | Provide a non-negative value in seconds to modify the expiry time of the generated JWT. Default 3600 seconds (1 hour) |
| 7 | scope (Optional) | Text | Provide the scope for the token. |
| 8 | scsKey | Text | Provide the Third-party Credential Store key. The Credential Store must contain the field 'privatekey' or ‘secretkey’ (For HS256, 384 and 512 algorithms) which must hold the private key content without the BEGIN and END markers. And not in DER format. |
| 9 | usePerUserCredentials (Optional) | Boolean | Provide whether to use user specific credentials. |
| 10 | algorithm | Text | The signing algorithm to be used. Valid values are - RS256, RS384, RS512, PS256, PS384, PS512, ES256, ES384, ES512, HS256, HS384, HS512 |
| 11 | customClaims (Optional) | Dictionary | The custom claims to be added in the JWT. The provided value must be a dictionary with text or list of text as values. Any value of type other than text will produce the JWT with the value in the object string notation. Only a dictionary with text or list of text values is accepted. |

**Output** - The signed JSON Web Token (JWT)

# Create Unsigned JWT - createunsignedjwt()

This function returns an unsigned JWT. Provide a text value for the ‘audience’ parameter to generate JWT with a single aud claim.

**Note: If expiry is provided with a null value or 0 it will be set to 3600 second as default.**

## Parameters

| **Sl No** | **Name** | **Type** | **Description** |
| --- | --- | --- | --- |
| 1 | kid (Optional) | Text | The value to identify which key was used in signing the JWT. |
| 2 | issuer (Optional) | Text | The value to identify the principal that issued the JWT. |
| 3 | subject (Optional) | Text | The value to identify the principal that is the subject of the JWT. |
| 4 | audience (Optional) | List of Text | The value to identify the recipients the JWT is intended for. |
| 5 | jti (Optional) | Text | The value to uniquely identify the generated JWT. Provide null to disable jti. |
| 6 | expiry (Optional) | Number(Integer) | Provide a non-negative value in seconds to modify the expiry time of the generated JWT. Default 3600 seconds (1 hour) |
| 7 | scope (Optional) | Text | Provide the scope for the token. |
| 8 | customClaims (Optional) | Dictionary | The custom claims to be added in the JWT. The provided value must be a dictionary with text or list of text as values. Any value of type other than text will produce the JWT with the value in the object string notation. Only a dictionary with text or list of text values is accepted. |

**Output** - The RS384 signed JSON Web Token (JWT)

# Verify JWT -verifyjwt()

This function verifies the signature and expiry of the signed JWT. It returns true if the signed JWT has not expired and is valid and false otherwise. If an unsigned JWT is provided as input to the ‘token’ parameter, this function will throw an error.

If the algorithm specified is HS256 or HS384 or HS512, then the function expects the ‘secretkey’ field in the specified SCS to hold the HmacSHA256 or 384 or 512 key respectively. For all other supported algorithms, ‘publickey’ field must be present holding the public key generated using RSA [RS and PS signing algorithms] or ECDSA [ES signing algorithms] respectively.

The provided public key must contain only the key contents in plain format without the BEGIN and END markers.

## Parameters

| **Sl No** | **Name** | **Type** | **Description** |
| --- | --- | --- | --- |
| 1 | token | Text | The value of the generated java web token. |
| 2 | algorithm | Text | The value of the algorithm used. Valid values:  HS256, HS384, HS512 , RS256, R384, RS512 ,  ES256, ES384, ES512 , PS256, PS384, PS512 |
| 7 | scsKey | Text | Provide the Third-party Credential Store key. The Credential Store must contain the field 'publickey' which must hold the private key content without the BEGIN and END markers in plain PKCS8 format. |
| 8 | usePerUserCredentials (Optional) | Boolean | Provide whether to use user specific credentials. |

**Output** - Boolean indicating the status.

# Decode JWT -decodejwt()

This function decodes and returns the header and payload contents of the JWT without verification.

Note: All header and payload contents are returned only as text.

## Parameters

| **Sl No** | **Name** | **Type** | **Description** |
| --- | --- | --- | --- |
| 1 | token | Text | The value of the generated java web token. |

**Output** - The decoded token dictionary without verifying the signature.

# Generate Hmac Sha Key - generatehmacshakey()

This function to generate a Hmac-SHA key that can be used to sign a JWT using HS\* algorithm.

## Parameters

| **Sl No** | **Name** | **Type** | **Description** |
| --- | --- | --- | --- |
| 1 | **algorithm** | Text | Specify the desired HMAC algorithm. Valid values are HS256, HS384, HS512 |

**Output** - The generated HMAC SHA Key.