JWTUtil Class

Contains methods for validating a JSON Web Token (JWT) from an external identity provider as part of the OAuth 2.0 token exchange flow. Use these methods as part of the `validateIncomingToken` method in the `Auth.Oauth2TokenExchangeHandler` class.

Namespace

Auth

Usage

See Token Exchange Handler Validation and Subject Mapping.

If the methods in this class fail, Salesforce returns an `Auth.JWTValidationException` exception.

- JWTUtil Methods

JWTUtil Methods

The following are methods for `JWTUtil`.

- parseJWTFromStringWithoutValidation(incomingJWT)
  Parses a JWT from an encoded string into a header, payload, and signature. Use this method to decode the JWT without validating it.
- validateJWTWithCert(incomingJWT, certDeveloperName)
  Parses and validates the JWT using a certificate saved in Salesforce. The certificate can be self-signed or signed by a certificate authority.
- validateJWTWithKey(incomingJWT, publicKey)
  Parses and validates the JWT using a public key from the external identity provider.
- validateJWTWithKeysEndpoint(incomingJWT, keysEndpoint, shouldUseCache)
  Parses and validates the JWT using a remote JSON Web Key Set (JWKS) endpoint on your external identity provider.

parseJWTFromStringWithoutValidation(incomingJWT)

Parses a JWT from an encoded string into a header, payload, and signature. Use this method to decode the JWT without validating it.

Signature

```
public static Auth.JWT parseJWTFromStringWithoutValidation(String incomingJWT)
```

Parameters

incomingJWT

Type: String

The JWT from your identity provider.

Return Value

Type:Auth.JWT

Signature

```
public static Auth.JWT validateJWTWithCert(String incomingJWT, String certDeveloperName)
```

Parameters

incomingJWT

Type: String

The JWT from your identity provider.

certDeveloperName

Type: String

A certificate saved in the Certificate and Key Management page in Setup.

Return Value

Type: Auth.JWT

validateJWTWithKey(incomingJWT, publicKey)

Parses and validates the JWT using a public key from the external identity provider.

Signature

```
public static Auth.JWT validateJWTWithKey(String incomingJWT, String publicKey)
```

Parameters

incomingJWT

Type: String

The JWT from your identity provider.

publicKey

Type: String

The public key from your identity provider.

Return Value

Type: Auth.JWT

validateJWTWithKeysEndpoint(incomingJWT, keysEndpoint, shouldUseCache)

Parses and validates the JWT using a remote JSON Web Key Set (JWKS) endpoint on your external identity provider.

Signature

```
public static Auth.JWT validateJWTWithKeysEndpoint(String incomingJWT, String keysEndpoint,
Boolean shouldUseCache)
```

Parameters

incomingJWT

Type: String

The JWT from your identity provider.

endpoint must conform to the specification defined in RFC 7517: JSON Web Key (JWK).

To test HTTP callouts to the JWKS endpoint, use the `Auth.HttpCalloutMockUtil` class.

shouldUseCache

Type: Boolean

Indicates whether the certificate from the keys endpoint is cached.

Return Value

Type: Auth.JWT

DID THIS ARTICLE SOLVE YOUR ISSUE?
Let us know so we can improve!

Share your feedback

DEVELOPER CENTERS
Heroku
MuleSoft
Tableau
Commerce Cloud
Lightning Design System
Einstein
Quip

POPULAR RESOURCES
Documentation
Component Library
APIs
Trailhead
Sample Apps
Podcasts
AppExchange

COMMUNITY
Trailblazer Community
Events and Calendar
Partner Community
Blog
Salesforce Admins
Salesforce Architects

Privacy Information Terms of Service Legal Use of Cookies Trust Cookie Preferences

Your Privacy Choices Responsible Disclosure Contact