



JWT Class

Generates the JSON Claims Set in a JSON Web Token (JWT). The resulting Base64-encoded payload can be passed as an argument to create an instance of the `Auth.JWT` class.

Namespace

[Auth](#)

Usage

Use the methods in this class to generate the payload in a JWT bearer token for the OAuth 2.0 JWT bearer token flow. For more information and a full code sample, see [JWTBearerTokenExchange Class](#).

- [JWT Methods](#)

JWT Methods

The following are methods for `JWT`. All are instance methods.

- [clone\(\)](#)
Makes a duplicate copy of the JWT object.
- [getAdditionalClaims\(\)](#)
Returns a map of additional claims in the JWT, where the key string contains the name of the claim, and the value contains the value of the claim.
- [getAud\(\)](#)
Returns the audience (`aud`) claim that identifies the intended recipients of the JWT.
- [getIss\(\)](#)
Returns the issuer (`iss`) claim that identifies the issuer of the JWT.
- [getNbfClockSkew\(\)](#)
Returns the not before (`nbf`) claim that identifies the time before which the JWT must not be accepted for processing, while allowing some leeway for clock skew. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. To return the `nbf` claim for these JWTs, use the `getAdditionalClaims` method instead.
- [getSub\(\)](#)
Returns the subject (`sub`) claim that identifies the current user of the JWT.
- [getValidityLength\(\)](#)
Returns the length of time (in seconds) that the JWT is valid, which affects the expiration (`exp`) claim. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. To return the validity length for these JWTs, use the `getAdditionalClaims` method instead.
- [setAdditionalClaims\(additionalClaims\)](#)
Sets the additional claims in the JWT. Returned by the `getAdditionalClaims` method.
- [setAud\(aud\)](#)
Sets the audience (`aud`) claim in the JWT. Returned by the `getAud` method.
- [setIss\(iss\)](#)
Sets the issuer (`iss`) claim in the JWT. Returned by the `getIss` method.



- **setSub(sub)**
Sets the subject (sub) claim in the JWT. Returned by the `getSub` method.
- **setValidityLength(ValidityLength)**
Sets the length of time (in seconds) that the JWT is valid, which affects the expiration (exp) claim. Returned by the `getValidityLength` method. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. For these JWTs, the incoming JWT determines the validity length.
- **toJSONString()**
Generates the JSON object representation of the Claims Set as an encoded JWT payload.

clone()

Makes a duplicate copy of the JWT object.

Signature

```
public Object clone()
```

Return Value

Type: [JWT](#)

getAdditionalClaims()

Returns a map of additional claims in the JWT, where the key string contains the name of the claim, and the value contains the value of the claim.

Signature

```
public Map<String,Object> getAdditionalClaims()
```

Return Value

Type: [Map<String,Object>](#)

The claims returned depend on how the JWT was generated.

If the JWT was generated using other methods in the `Auth.JWT` class, this method returns the claims that were set using the `setAdditionalClaims` method.

For JWTs generated using methods in the [Auth.JWTUtil](#) class, the `getAdditionalClaims` method returns all claims except for these three.

- aud (audience)—Use the `getAud` method instead.
- iss (issuer)—Use the `getIss` method instead.
- sub (subject)—Use the `getSub` method instead.

For these JWTs, when the incoming JWT has a claim that stores an inner JSON list, the claim value is returned as a string.

getAud()

Returns the audience (aud) claim that identifies the intended recipients of the JWT.

Signature

```
public String getAud()
```

Return Value

Type: [String](#)



```
public String getIss()
```

Return Value

Type: [String](#)

getNbfClockSkew()

Returns the not before (*nbf*) claim that identifies the time before which the JWT must not be accepted for processing, while allowing some leeway for clock skew. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. To return the *nbf* claim for these JWTs, use the `getAdditionalClaims` method instead.

Signature

```
public Integer getNbfClockSkew()
```

Return Value

Type: [Integer](#)

getSub()

Returns the subject (*sub*) claim that identifies the current user of the JWT.

Signature

```
public String getSub()
```

Return Value

Type: [String](#)

getValidityLength()

Returns the length of time (in seconds) that the JWT is valid, which affects the expiration (*exp*) claim. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. To return the validity length for these JWTs, use the `getAdditionalClaims` method instead.

Signature

```
public Integer getValidityLength()
```

Return Value

Type: [Integer](#)

setAdditionalClaims(additionalClaims)

Sets the additional claims in the JWT. Returned by the `getAdditionalClaims` method.

Signature

```
public void setAdditionalClaims(Map<String,Object> additionalClaims)
```

Parameters

additionalClaims

Type: [Map<String,Object>](#)



Additional claims must not include any standard claims.

setAud(aud)

Sets the audience (*aud*) claim in the JWT. Returned by the `getAud` method.

Signature

```
public void setAud(String aud)
```

Parameters

aud

Type: [String](#)

Return Value

Type: void

setIss(iss)

Sets the issuer (*iss*) claim in the JWT. Returned by the `getIss` method.

Signature

```
public void setIss(String iss)
```

Parameters

iss

Type: [String](#)

Return Value

Type: void

setNbfclockSkew(nbfClockSkew)

Sets the not before (*nbf*) claim in the JWT. Returned by the `getNbfclockSkew` method. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. For these JWTs, the incoming JWT determines the *nbf* claim.

Signature

```
public void setNbfclockSkew(Integer nbfClockSkew)
```

Parameters

nbfClockSkew

Type: [Integer](#)

Return Value

Type: void

setSub(sub)

Sets the subject (*sub*) claim in the JWT. Returned by the `getSub` method.

Signature



Type: [String](#)

Return Value

Type: void

setValidityLength(ValidityLength)

Sets the length of time (in seconds) that the JWT is valid, which affects the expiration (`exp`) claim. Returned by the `getValidityLength` method. This method returns a `NoAccess` exception for JWTs generated using methods in the `Auth.JWTUtil` class. For these JWTs, the incoming JWT determines the validity length.

Signature

```
public void setValidityLength(Integer validityLength)
```

Parameters

validityLength

Type: [Integer](#)

Return Value

Type: void

toJSONString()

Generates the JSON object representation of the Claims Set as an encoded JWT payload.

Signature

```
public String toJSONString()
```

Return Value

Type: [String](#)

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](#)

POPULAR RESOURCES

[Documentation](#)
[Component Library](#)
[APIs](#)
[Trailhead](#)
[Sample Apps](#)
[Podcasts](#)

COMMUNITY

[Trailblazer Community](#)
[Events and Calendar](#)
[Partner Community](#)
[Blog](#)
[Salesforce Admins](#)
[Salesforce Architects](#)



Salesforce Tower, 415 Mission Street, 3rd Floor, San Francisco, CA 94105, United States

- [Privacy Information](#)
- [Terms of Service](#)
- [Legal](#)
- [Use of Cookies](#)
- [Trust](#)
- [Cookie Preferences](#)
- ☒ [Your Privacy Choices](#)
- [Responsible Disclosure](#)
- [Contact](#)