Tutorials

Tutorial (deutsch): JSON Web Token bzw. JWT

Was ist JSON Web Token bzw. JWT?

JSON Web Token bzw. JWT ist ein sog. Access-Token, das auf JSON basiert. Es ermöglicht, dass Daten im JSON-Format zwischen zwei Parteien ausgetauscht werden. Es beinhaltet z.B. Informationen über den Absender und darüber, ob dieser die benötigten Zugriffsrechte hat, um beispielsweise bestimmte Daten abzurufen. Ein JWT kann auch verschlüsselt werden. JWT-Implementierungen stehen für viele Plattformen zur Verfügung (siehe Webseite https://jwt.io/ - für die Inhalte des Links übernehme ich keine Haftung!).

Wo wird JSON Web Token bzw. JWT eingesetzt?

JSON Web Token bzw. JWT wird meist zur Benutzer-Authentifizierung bzw. zum Benutzer-Login verwendet und ist besonders für REST-Anwendungen geeignet. Zudem ist es flexibler als die Authentifizierung über Cookies. Dadurch, dass sämtliche Informationen bzgl. der Authentifizierung im Token übertragen werden (stateless session), ist keine Speicherung der Sitzung/Session auf dem Server und somit auch keine Datenbank-Abfrage notwendig.

Wie funktioniert die Kommunikation mit JSON Web Token bzw. JWT?

Gibt der Benutzer erfolgreich seinen Benutzernamen und sein Passwort ein, wird das JSON Web Token bzw. JWT vom Server zurückgegeben und lokal auf dem Client gespeichert. Ab sofort wird das JWT bei Anfragen vom Client an den Server immer als Parameter mitgesendet, so dass der Server das JWT entschlüsseln und bei erfolgreicher Überprüfung die Anfrage ausführen kann. Für den Benutzer ist die Kommunikation über JWT nicht sichtbar. Es wird empfohlen, ein Ablauf-Datum für den JWT einstellen (z.B. 24 Stunden). Zudem wird ausdrücklich empfohlen, JWT immer in Verbindung mit SSL bzw. TLS verwenden (Übertragung über HTTPS).

Woraus besteht ein JSON Web Token bzw. JWT?

Ein JSON Web Token bzw. JWT besteht aus drei Teilen, die Base64-kodiert und durch einen Punkt voneinander getrennt sind:

```
Header.Payload.Signature
```

Header:

- Besteht zumeist aus den zwei folgenden Teilen.
- Typ (immer JWT, beschreibt den IANA-Medientyp application/jwt).
- Signatur-Methode bzw. Verschlüsselungs-Algorithmus (z.B. HS256, RS256 oder ES256, none möglich, aber nicht empfohlen).

- Beispiel:

```
{
 "alg": "HS256",
 "typ": "JWT"
}?
```

Payload:

- Enthält die eigentlichen Informationen, die an die Anwendung übermittelt werden sollen im JSON-Format als Key-/Value-Paare, wobei die Keys die sog. Claims sind.
- Es wird zwischen verschiedenen Claims unterschieden.
- Registrierte Claims (Standard-Claims aus dem IANA JSON Web Token Claim Register, z.B. iss für issuer/Aussteller, aud für audience/Zieldomäne, exp für expiration time/Verfallsdatum/-uhrzeit).
- Öffentliche Claims (keine Einschränkung, dürfen nicht mit den registrierten Claims kollidieren).
- Private Claims (spezifischere Informationen wie z.B. User-ID, dürfen nicht mit registrierten und öffentlichen Claims kollidieren).
- Alle Claims sind optional, es können beliebig viele Claims enthalten sein, man sollte sich allerdings auf die notwendigsten beschränken.
- Beispiel:

```
{
 "sub": "9876543210",
 "name": "Hein Blöd",
 "admin": true,
 "exp": 30
}?
```

Signatur:

- Der Aufbau der Signatur wird durch JSON Web Signature bzw. JWS definiert.
- Damit die Signatur funktioniert, muss ein geheimer Schlüssel (secret) verwendet werden, den die Anwendung kennt (dies stellt sicher, dass die Nachricht nicht verändert wurde und verifiziert zusätzlich den Absender des JWT).
- Anschließend muss eines der drei folgenden Verfahren zur Anwendung kommen.
- Keine Signatur/Sicherung bzw. "alg": "none" im Header: Es wird keine Signatur generiert (nicht empfohlen, JWT besteht nur aus Header und Payload).
- Signatur (JWS).
- Signatur (JWS) und Verschlüsselung (JWE): JSON Web Encryption bzw. JWE zusätzlich zur JWS (verschlüsselt die Inhalte des Payloads).
- Aus diesen Daten wird die Base64-kodierte Signatur erstellt.

Wie wird die Signatur eines JSON Web Token bzw. JWT erstellt?

```
var encodedString = base64UrlEncode(header) + "." + base64UrlEncode(pa
yload);
var hash = HMACSHA256(encodedString, secret);
```

Wie wird der eigentliche JSON Web Token bzw. JWT erstellt?

```
var jwt = base64UrlEncode(header) + "." + base64UrlEncode(payload) + "
." + hash;
```

Wie wird der JSON Web Token bzw. JWT übertragen?

- Für die Übertragung des JSON Web Token bzw. JWT gibt es zwei Möglichkeiten.
- Das JWT kann über die URL übertragen werden. Beispiel:

  ```
  Beispiel: http://www.testseite.de/path?jwt_token=[JWT]?
  ```

- Das JWT kann über den Header übertragen werden (im Authorization-Feld als Bearer Token oder alternativ im Cookie-Feld).

Eindeutige ID: #2690
Verfasser:
Letzte Änderung: 2022-10-27 08:00