

Node.js: creare e servire un JSON Web Token (JWT) in ExpressJS

Un JSON Web Token (JWT) è uno standard aperto (RFC 7519) che definisce un modo compatto e sicuro per trasmettere informazioni tra le parti come un oggetto JSON firmato digitalmente. In Node.js, si può creare e servire un JWT utilizzando il modulo `jsonwebtoken` insieme al framework ExpressJS.

Per iniziare, è necessario installare il modulo `jsonwebtoken` eseguendo il comando `npm install jsonwebtoken` nella console del terminale. Successivamente, è necessario importare il modulo in un file JavaScript e definire una chiave segreta per la firma del JWT.

```
const jwt = require('jsonwebtoken');
const secretKey = 'my_secret_key';
```

In seguito, si può creare un JWT con i dati che si desidera includere nel token, utilizzando il metodo `sign()` del modulo `jsonwebtoken`.

```
'use strict';

const userData = {
  name: 'John Doe',
  email: 'johndoe@email.com',
  role: 'user'
};
```

```
const token = jwt.sign(userData, secretKey, {
  expiresIn: '1h' });
```

Il JWT creato conterra i dati specificati nella variabile `userData`, firmati con la chiave segreta `secretKey` e con una durata di validità di un'ora.

Per servire il JWT tramite ExpressJS, è necessario creare un'API che restituisca il token come una risposta HTTP. Per esempio:

```
'use strict';

const express = require('express');
const app = express();

app.get('/api/token', (req, res) => {
  const token = jwt.sign(userData, secretKey, {
    expiresIn: '1h' });
  res.json({ token });
});

app.listen(3000, () => {
  console.log('Server running on port 3000');
});
```

In questo esempio, viene creato un'API GET che restituisce il JWT come un oggetto JSON con la proprietà `token`. Quando si richiama questa API tramite il browser o un'applicazione, si riceverà un JWT valido per un'ora.

In sintesi, creare e servire un JWT in Node.js tramite ExpressJS è un processo relativamente semplice che richiede solo poche righe di codice utilizzando il pacchetto `'jsonwebtoken'`. Una volta creato il token, si può

includere in una richiesta HTTP come header Authorization o in un cookie, per autenticare l'utente in una applicazione web o mobile.