JavaScript: salvare un JSON Web Token (JWT) in un cookie e usarlo con le Fetch API

I JSON Web Token (JWT) sono diventati una delle tecniche di autenticazione più popolari per le applicazioni web moderne. Tuttavia, la loro gestione può risultare complessa. Uno dei problemi più comuni è come salvare e gestire un JWT una volta che è stato generato. In questo articolo, vediamo come salvare un JWT in un cookie e poi utilizzarlo con le Fetch API di JavaScript.

Il salvataggio di un JWT in un cookie è relativamente semplice. Prima di tutto, è necessario generare il token e inserirlo all'interno di un cookie. Ci sono diverse librerie JavaScript disponibili per aiutare a farlo. Ad esempio, la libreria js-cookie è una delle più popolari e consente di salvare facilmente un valore in un cookie:

```
import Cookies from 'js-cookie';

const jwt = 'my-json-web-token';
Cookies.set('jwt', jwt);
```

Una volta salvato il JWT in un cookie, possiamo utilizzarlo per autenticarci alle nostre API utilizzando le Fetch API di JavaScript. In genere, dovremmo inviare il JWT come header di autorizzazione con ogni richiesta alle nostre API:

```
import Cookies from 'js-cookie';

const jwt = Cookies.get('jwt');

fetch('https://my-api.com/data', {
  headers: {
    Authorization: `Bearer ${jwt}`,
  },
})
  .then((response) => response.json())
  .then((data) => console.log(data))
  .catch((error) => console.error(error));
```

In questo esempio, recuperiamo il JWT dal cookie utilizzando la libreria js-cookie e lo utilizziamo come header di autorizzazione con la Fetch API di JavaScript. Questo ci consente di autenticarci con le nostre API e ricevere i dati richiesti.

In conclusione, salvare un JWT in un cookie e utilizzarlo con le Fetch API di JavaScript è relativamente semplice. Ci sono diverse librerie disponibili per aiutare a gestire i cookie e gli header di autorizzazione, e utilizzando questi strumenti possiamo creare un'esperienza di autenticazione sicura e affidabile per i nostri utenti.