

GABRIELE ROMANATO

Gestire i JSON Web Token (JWT) in Vue.js

I JSON Web Token (JWT) sono ampiamente utilizzati per l'autenticazione stateless nelle applicazioni web. In questo articolo vedremo come gestirli correttamente in un'applicazione Vue.js, coprendo le fasi di login, salvataggio del token, invio nelle richieste e logout.

1. Login e ricezione del JWT

Dopo che l'utente invia le proprie credenziali, il backend restituisce un JWT. Questo token deve essere salvato per l'utilizzo nelle successive richieste.

```
// Esempio di chiamata login con Axios
axios.post('/api/login', {
  username: 'utente',
  password: 'password'
})
.then(response => {
  const token = response.data.token;
  localStorage.setItem('jwt', token);
})
.catch(error => {
  console.error('Errore di login:', error);
});
```

2. Impostare il token nelle richieste

Ogni richiesta autenticata al server deve includere il token JWT nell'header Authorization.

```
// Configurazione globale di Axios
axios.interceptors.request.use(config => {
  const token = localStorage.getItem('jwt');
  if (token) {
    config.headers.Authorization = `Bearer ${token}`;
  }
  return config;
});
```

3. Decodificare il token (opzionale)

Puoi decodificare il token per leggere i dati in esso contenuti, ad esempio l'ID utente o il ruolo.

```
import jwtDecode from 'jwt-decode';

const token = localStorage.getItem('jwt');
if (token) {
  const decoded = jwtDecode(token);
  console.log('Utente loggato:', decoded);
}
```

4. Logout

Per disconnettere l'utente, basta rimuovere il token dal localStorage.

```
localStorage.removeItem('jwt');
```

5. Proteggere le route

Puoi usare i navigation guard di Vue Router per bloccare l'accesso alle route protette se l'utente non ha un token valido.

```
router.beforeEach((to, from, next) => {  
  const token = localStorage.getItem('jwt');  
  if (to.meta.requiresAuth && !token) {  
    next('/login');  
  } else {  
    next();  
  }  
});
```

Conclusione

La gestione dei JWT in Vue.js richiede attenzione ma segue uno schema semplice: login, salvataggio del token, uso nelle richieste, eventuale decodifica e logout. È importante anche gestire la scadenza del token e i casi di token non valido, per garantire la sicurezza dell'applicazione.

Applicazioni Correlate



-

JavaScript Password Mask

Un esempio in JavaScript di mascheramento di una password con l'aggiunta della funzionalità di copia negli appunti.

[Docker](#)[Docker Compose](#)[JavaScript](#)



Python Placeholder Image

Applicazione sviluppata in Python con Flask per la creazione di immagini segnaposto.

Docker Docker Compose Python Flask JavaScript



-

Go Placeholder Image

Applicazione in Go per la creazione di immagini segnaposto.

[Docker](#)[Docker Compose](#)[Go](#)[JavaScript](#)