GABRIELE ROMANATO

Mascherare le password con JavaScript

In questo articolo, esploreremo uno snippet compatto di JavaScript che genera una password sicura, la maschera per la visualizzazione e fornisce un modo intuitivo per copiarla negli appunti. Questo è particolarmente utile per applicazioni web che gestiscono la creazione di account o funzionalità di gestione delle password.

Panoramica delle funzionalità principali

Il codice si concentra su tre compiti principali:

1. Generare una password casuale con lunghezza personalizzabile
2. Mascherare parte della password per una visualizzazione più sicura
3. Consentire agli utenti di copiare l'intera password negli appunti

Generazione della password

La funzione `generatePassword` crea una password robusta selezionando casualmente caratteri da un set definito che include lettere minuscole, maiuscole, numeri e simboli.

```
function generatePassword(length = 12) {
    const charset = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*
()_+';
    let password = '';
    for (let i = 0; i < length; i++) {
        let randomIndex = Math.floor(Math.random() * charset.length);
        password += charset[randomIndex];
    }
    return password;
}
```

Questa funzione genera di default una password di 12 caratteri, anche se la lunghezza può essere personalizzata. Garantisce un alto livello di entropia selezionando caratteri da un set variegato.

Offuscamento della password

Per aumentare la sicurezza dell'utente, la funzione `generateMask` oscura parzialmente la password, mostrando solo i primi quattro caratteri.

```
function generateMask(pass) {
    return pass.split('').map((char, index) => {
        return index < 4 ? char : '*';
    }).join('');
}
```

Questo approccio è particolarmente utile quando si visualizzano le password nei componenti dell'interfaccia utente, poiché evita l'esposizione completa pur permettendo all'utente di riconoscere quale password è in uso.

Visualizzazione e copia della password

Lo script seleziona due elementi del DOM: uno per mostrare la password mascherata e un altro per gestire l'azione di copia.

```
const passwordText = document.querySelector('.password-text');
const passwordCopy = document.querySelector('.password-copy');

const password = generatePassword();
passwordText.textContent = generateMask(password);
```

Il meccanismo di copia utilizza l'API `navigator.clipboard.writeText` per inserire in modo sicuro l'intera password negli appunti dell'utente. Il feedback viene fornito attraverso l'aggiornamento dinamico del testo del pulsante.

```
passwordCopy.addEventListener('click', () => {
    passwordCopy.innerText = 'Copia password';
    navigator.clipboard.writeText(password).then(() => {
        passwordCopy.innerText = 'Password copiata!';
        setTimeout(() => {
            passwordCopy.innerText = 'Copia password';
        }, 2000);
    }).catch(err => {
        passwordCopy.innerText = 'Copia fallita';
    });
});
```

Conclusione

Questo script, piccolo ma efficace, dimostra come costruire un'utilità per la gestione delle password user-friendly per applicazioni web moderne. Combina sicurezza (tramite l'offuscamento), usabilità (tramite integrazione con la clipboard) e semplicità, rendendolo un'ottima aggiunta a qualsiasi toolkit frontend.

Applicazioni Correlate

- ### JavaScript Password Mask

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

- **Python Placeholder Image**

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

-

  Go Placeholder Image

  Applicazione in Go per la creazione di immagini segnaposto.
  DockerDocker ComposeGoJavaScript