GABRIELE ROMANATO

Utilizzo dell'oggetto crypto in JavaScript

L'oggetto `crypto` in JavaScript fornisce funzionalità crittografiche che possono essere utilizzate per generare numeri casuali sicuri, creare hash, cifrare e decifrare dati. È disponibile nelle moderne API Web e nei contesti Node.js.

Generazione di numeri casuali sicuri

Per generare numeri casuali crittograficamente sicuri, possiamo usare il metodo `crypto.getRandomValues()`:

```
const array = new Uint8Array(10);
crypto.getRandomValues(array);
console.log(array);
```

Questo metodo riempie un array tipizzato con numeri casuali sicuri.

Creazione di un hash

Possiamo utilizzare l'API `crypto.subtle.digest` per creare un hash, ad esempio SHA-256:

```
async function generateHash(message) {
    const encoder = new TextEncoder();
    const data = encoder.encode(message);
    const hashBuffer = await crypto.subtle.digest('SHA-256', data);
    const hashArray = Array.from(new Uint8Array(hashBuffer));
    const hashHex = hashArray.map(byte => byte.toString(16).padStart(2, '0')).join('');
    return hashHex;
}

generateHash("Hello, world!").then(console.log);
```

Questo codice converte una stringa in un hash SHA-256.

Crittografia e decrittografia

Con l'API `crypto.subtle`, possiamo cifrare e decifrare dati usando AES-GCM:

```
async function encryptData(key, data) {
    const encoder = new TextEncoder();
    const encodedData = encoder.encode(data);
    const iv = crypto.getRandomValues(new Uint8Array(12));
    const encrypted = await crypto.subtle.encrypt(
        { name: "AES-GCM", iv: iv },
        key,
        encodedData
    );
    return { encrypted, iv };
}

async function decryptData(key, encrypted, iv) {
    const decrypted = await crypto.subtle.decrypt(
        { name: "AES-GCM", iv: iv },
        key,
```

```
        encrypted
    );
    return new TextDecoder().decode(decrypted);
}
```

Questa implementazione utilizza AES-GCM, un algoritmo sicuro e moderno per la crittografia simmetrica.

Conclusione

L'oggetto `crypto` in JavaScript fornisce potenti strumenti per la sicurezza delle applicazioni, consentendo la generazione di numeri casuali sicuri, la creazione di hash e l'implementazione di tecniche di crittografia moderne.

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