Node.js: usare bcrypt

bcrypt è una funzione di hashing che in Node.js può essere utilizzata con un modulo NPM specifico.

Il modulo bcrypt.js può essere usato sia in Node.js che nel browser. In Node.js fa uso del modulo core `crypto` per la creazione sicura di numeri casuali.

Questo modulo fornisce sia metodi sincroni che asincroni. Poiché l'operazione di cifratura di una stringa è computazionalmente dispendiosa, si consiglia di utilizzare i metodi asincroni `hash()` (cifratura della stringa) e `compare()` (verifica dell'uguaglianza tra una stringa non cifrata e la sua versione cifrata) che possono essere usati sia con Promise che con funzioni di callback. Così facendo si evita di bloccare l'Event Loop.

`hash()` può essere usato specificando la stringa da cifrare ed il numero di round necessario all'operazione. Un valore compreso tra 8 e 12 è considerato un ragionevole compromesso tra sicurezza e performance.

```
'use strict';

const bcrypt = require('bcryptjs');

const hashPassword = async str => {
    try {
        const hashedPwd = await bcrypt.hash(str, 12);
        return hashedPwd;
    } catch(err) {
        return '';
    }
};
```

`compare()` viene usato per verificare se la stringa passata come primo argomento corrisponde all'hash del secondo argomento. Restituisce un valore booleano.

```
const verifyPassword = async (str, hash) => {
    try {
       const isValid = await bcrypt.compare(str,
hash);
       return isValid;
    } catch(err) {
        return false;
    }
};
```

Riferimenti

bcrypt