

JavaScript: generare uno pseudo-UUID

In JavaScript possiamo generare uno pseudo-UUID.

La soluzione è la seguente:

```
'use strict';

function generatePseudoUUID() {
    const d = new Date().getTime();
    return 'xxxxxxxx-xxxx-4xxx-yxxx-
xxxxxxxxxxxx'.replace(/[xy]/g, function(c) {
        return (d + Math.random() * 16) % 16 | 0;
    });
}
```

Questa funzione JavaScript genera un pseudo UUID (Universally Unique Identifier) utilizzando il timestamp corrente e un pattern specifico. All'interno della funzione, viene creata una variabile `d` che memorizza il timestamp corrente ottenuto utilizzando il metodo `getTime()` dell'oggetto `Date`. Viene restituita una stringa che rappresenta l'UUID generato. La stringa segue un pattern specifico: `'xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'`. Ogni "x" e "y" nel pattern rappresentano un carattere esadecimale.

La stringa di pattern viene elaborata utilizzando il metodo `replace()` con un'espressione regolare `[xy]`. Questo metodo sostituisce ogni occorrenza di "x" o "y" nella stringa con un valore generato dalla funzione di callback

fornita come secondo argomento. La funzione di callback viene eseguita per ogni corrispondenza trovata nella stringa di pattern. Riceve un parametro `c` che rappresenta il carattere corrispondente ("x" o "y") e restituisce il valore generato per quel carattere.

All'interno della funzione di callback, viene generato un valore esadecimale per il carattere corrente utilizzando l'espressione $(d + \text{Math.random}() * 16) \% 16 | 0$. Ossia:

- `Math.random()` restituisce un numero casuale compreso tra 0 (incluso) e 1 (escluso).
- Moltiplicando `Math.random()` per 16, otteniamo un numero compreso tra 0 e 15 (estremi inclusi).
- Aggiungendo il timestamp corrente `d` a questo numero, otteniamo un valore compreso tra `d` e `d + 15`.
- Calcolando il resto della divisione per 16 $(d + \text{Math.random}() * 16) \% 16$, otteniamo un numero compreso tra 0 e 15.
- Utilizzando l'operatore bitwise OR `| 0`, convertiamo il risultato in un intero, troncando eventuali decimali. Questo garantisce che il valore finale sia un numero intero compreso tra 0 e 15.

Il valore generato viene quindi utilizzato per sostituire il carattere corrispondente nella stringa di pattern. Alla fine, la funzione restituisce la stringa risultante che rappresenta l'UUID generato.