

Node.js: il metodo `util.promisify()`

Il metodo `util.promisify()` è una funzione fornita dal modulo "util" di Node.js che permette di convertire le funzioni asincrone `callback-style` in funzioni che restituiscono una `Promise`. Questo metodo è particolarmente utile quando si utilizzano librerie asincrone che seguono lo stile `callback`, in quanto semplifica la gestione delle `Promise`.

Il metodo accetta come argomento una funzione che segue lo stile `callback`, ovvero una funzione che prende come ultimo argomento una `callback` che viene chiamata con eventuali errori o dati come parametri. La funzione restituita da questo metodo invece restituisce una `Promise` che risolve con i dati della `callback` o viene rigettata con eventuali errori.

Ad esempio, supponiamo di avere una funzione asincrona che effettua una chiamata API e restituisce i dati tramite una `callback`:

```
function fetchData(callback) {  
  // effettua la chiamata API  
  // chiama la callback con i dati o l'errore  
}
```

Utilizzando il metodo `util.promisify()` è possibile convertire questa funzione in una funzione che restituisce una `Promise`:

```
'use strict';  
  
const util = require('util');  
const fetchDataPromise = util.promisify(fetchData);  
  
fetchDataPromise().then(data => {  
  // gestisci i dati
```

```
}).catch(error => {  
  // gestisci l'errore  
});
```

In questo modo, la chiamata `fetchDataPromise()` restituisce una `Promise` che può essere gestita con il metodo `then()` se la `Promise` viene risolta con successo, o con il metodo `catch()` se viene rigettata con un errore.

In conclusione, il metodo `util.promisify()` semplifica la gestione delle promesse quando si utilizzano librerie asincrone che seguono lo stile `callback`, riducendo la necessità di gestire manualmente le `callback`.