

GABRIELE ROMANATO

Menu

Parsing del DOM con Node.js

Il parsing del DOM è una tecnica essenziale per estrarre, manipolare o analizzare i contenuti di documenti HTML o XML. In Node.js, esistono strumenti efficaci per interagire con il DOM e automatizzare attività come il web scraping o l'elaborazione di file HTML. In questo articolo vedremo come utilizzare librerie come `jsdom` e `cheerio`.

Parsing del DOM con jsdom

`jsdom` è una libreria che implementa un ambiente DOM completo per Node.js. È particolarmente utile per simulare un browser e manipolare HTML come faresti con JavaScript lato client.

Installazione

```
npm install jsdom
```

Esempio di utilizzo

Ecco un esempio di come caricare e analizzare un documento HTML con `jsdom`:

```
const { JSDOM } = require('jsdom');

const htmlContent = `
<!DOCTYPE html>
<html>
  <body>
    <h1>Titolo</h1>
    <p>Paragrafo di esempio.</p>
  </body>
</html>`;

const dom = new JSDOM(htmlContent);
const document = dom.window.document;

const title = document.querySelector('h1').textContent;
console.log('Titolo:', title);

const paragraph = document.querySelector('p').textContent;
console.log('Paragrafo:', paragraph);
```

Questo codice crea un oggetto `document` che ti permette di accedere agli elementi del DOM come faresti in un browser.

Parsing del DOM con Cheerio

Cheerio è un'altra libreria potente e leggera per analizzare e manipolare HTML. Ha una sintassi simile a jQuery, rendendola molto intuitiva per chi ha familiarità con questo strumento.

Installazione

```
npm install cheerio
```

Esempio di utilizzo

Ecco come utilizzare Cheerio per analizzare un documento HTML:

```
const cheerio = require('cheerio');

const htmlContent = `
<!DOCTYPE html>
<html>
  <body>
    <h1>Titolo</h1>
    <p>Paragrafo di esempio.</p>
  </body>
</html>`;

const $ = cheerio.load(htmlContent);

const title = $('h1').text();
console.log('Titolo:', title);

const paragraph = $('p').text();
console.log('Paragrafo:', paragraph);
```

Con Cheerio, puoi selezionare e manipolare elementi HTML utilizzando selettori CSS, rendendo il codice compatto e leggibile.

Confronto tra jsdom e Cheerio

Ecco una rapida comparazione tra queste due librerie:

Caratteristica	jsdom	Cheerio
Simulazione del DOM	Completa, simula un browser	Parziale, focalizzata sull'analisi
Performance	Più lenta per grandi file	Molto veloce
Sintassi	Vanilla JavaScript	Simile a jQuery

Conclusioni

Entrambe le librerie sono ottime per analizzare il DOM in Node.js, ma la scelta dipende dalle tue esigenze. Usa jsdom per simulare un ambiente browser completo e cheerio per compiti più leggeri e

mirati. Con un po' di pratica, queste librerie possono semplificare notevolmente il lavoro con documenti HTML.

Applicazioni Correlate



-

Node.js Placeholder Image

Applicazione per la generazione con Node.js di immagini segnaposto.

DockerDocker ComposeNode.jsJavaScriptExpressJS



-

Node.js URL Shortener

Implementazione in Node.js di un sistema per l'abbreviazione degli URL.

Docker Docker Compose Node.js JavaScript Express JS MongoDB



JS

-

JavaScript App Hash Change

Applicazione che sfrutta gli hash degli URL per gestire contenuto dinamico in JavaScript.
DockerDocker ComposeNode.jsJavaScriptExpressJSMYSQL