

# Node.js: ottenere gli header HTTP da un URL con curl

In questo tutorial vedremo come ottenere gli header HTTP da un URL usando curl e Node.js.

Il comando curl va eseguito dalla shell ed il suo output va trasformato in un oggetto letterale avente come proprietà e valori rispettivamente il nome ed il valore di ciascun header HTTP.

Possiamo definire la seguente classe:

```
'use strict';

const util = require('util');
const exec =
util.promisify(require('child_process').exec);
const validator = require('validator');

class cURL {
  constructor(url) {
    this.url = url;
  }

  validate() {
    return typeof this.url === 'string' &&
validator.isURL(this.url);
  }

  parseHeaders(output) {
    if(!output || output.length === 0) {
      return null;
    }
  }
}
```

```

    }
    const lines = output.split(/\r\n/);
    let headers = {};
    lines.forEach(line => {
        let values = line.split(/:\s/);
        if(values.length === 2 && typeof values[0]
=== 'string' && typeof values[1] === 'string') {
            let [key, value] = values;
            headers[key] = value.trim();
        }
    });
    return headers;
}

async getHeaders() {
    if(!this.validate()) {
        throw new Error('Invalid parameter.');
```

Il comando da eseguire dalla shell è `curl -I <URL>`. L'URL deve essere validato e se supera tale validazione può essere utilizzato con il metodo

core `exec()` che in questo esempio restituisce una Promise grazie al metodo di utility `promisify()`.

L'output del comando è formattato su più righe. In ciascuna riga il nome dell'header è separato dal suo valore tramite il pattern `:[spazio]`, quindi possiamo sfruttare tale schema per riuscire a separare righe ed header e a creare l'oggetto letterale restituito dal metodo principale della classe.

Esempio d'uso:

```
'use strict';

const curl = require('./lib/cURL');
const curlRequest = new
curl('https://gabrielromanato.com');

(async() => {
  try {
    console.log(await curlRequest.getHeaders());
  } catch(err) {
    console.log(err);
  }
})();
```