Node.js: verificare se un sito è stato aggiunto a Cloudflare

In questo tutorial vedremo come verificare se un sito è stato correttamente aggiunto a Cloudflare usando Node.js.

Un sito correttamente aggiunto a Cloudflare presenta sempre i record NS impostati sui server DNS di Cloudflare e opzionalmente un header HTTP con il valore di questo provider che indica l'abilitazione della funzionalità proxy.

Per prima cosa useremo il modulo NPM **got** (versione 11.8.3 per evitare di usare gli import dinamici) per reperire gli header HTTP del sito. Il modulo NPM **validator** verrà usato per la validazione dei parametri delle classi.

```
'use strict';

const validator = require('validator');
const got = require('got'); // 11.8.3

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

    async send() {
        if(!this.isValidURL(this.url) ) {
            return { error: 'Invalid parameter.' };
        }

        try {
            const response = await  got(this.url);
```

```
            return response.headers;
        } catch(err) {
            return { error: 'Request error.' };
        }
    }

    isValidURL(value) {
        if(typeof value !== 'string') {
            return false;
        }

        return validator.isURL(value);
    }
}
```

Ora dobbiamo reperire l'array dei record NS del dominio utilizzando l'oggetto core `Resolver` del modulo `dns` che ci permette di effettuare query DNS utilizzando specifici resolver.

```
'use strict';

const { Resolver } = require('dns').promises;
const resolver = new Resolver();
const nameservers = ['8.8.8.8', '8.8.4.4'];
const validator = require('validator');

resolver.setServers(nameservers);

class DNS {
    constructor(host) {
        this.host = host;
        this.recordType = 'NS';
    }
```

```
    async resolve() {
        if(!this.isValidHost(this.host) ) {
            return { error: 'Invalid parameter.' };
        }

        try {
            return await  resolver.resolve(this.host,
this.recordType);
        } catch(err) {
            return { error: 'Record not found.' };
        }
    }

    isValidHost(value) {
        if(typeof value !== 'string') {
            return false;
        }

        return validator.isFQDN(value);
    }
}
```

Non ci resta che creare la classe principale che sfrutterà le due classi helper ed aggiungerà due metodi che verificheranno la presenza dei record NS di Cloudflare e dell'header HTTP con il valore specifico.

```
class Cloudflare {
    constructor(host) {
        this.host = host;
        this.dns = new DNS(this.host);
        this.request = new
Request(`https://${this.host}`);
    }
```

```
    hasHostNSRecords(records) {
        if(!Array.isArray(records) || records.length
=== 0) {
            return false;
        }
        return records.every(record => { return
record.includes('ns.cloudflare.com'); });
    }

    hasHostHTTPHeader(headers) {
        if(typeof headers !== 'object' || headers ===
null) {
            return false;
        }
        return typeof headers.server !== 'undefined'
&& headers.server === 'cloudflare';
    }

    async check() {
        try {
            const records = await this.dns.resolve();
            const headers = await
this.request.send();
            return this.hasHostNSRecords(records) ||
this.hasHostHTTPHeader(headers);
        } catch(err) {
            return false;
        }
    }
}

module.exports = Cloudflare;
```

Esempio d'uso:

```
'use strict';

const Cloudflare = require('./lib/Cloudflare');
const cloudFlare = new
Cloudflare('gabrieleromanato.dev');

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