

Node.js: verificare la disponibilità di un nome a dominio su ResellerClub

In questo articolo vedremo come verificare la disponibilità di un nome a dominio su ResellerClub con Node.js.

Prima di effettuare una richiesta alle HTTP API di ResellerClub dobbiamo compiere tre azioni preliminari:

1. Ottenere il nostro ID utente.
2. Ottenere la chiave API.
3. Abilitare l'indirizzo IP da dove effettuare la richiesta.

Quindi salviamo il nostro ID utente e la chiave API in un file di configurazione `.env`.

```
RESELLER_CLUB_API_KEY=valore  
RESELLER_CLUB_ID=valore
```

Ora possiamo creare una funzione helper per effettuare la richiesta GET.

```
const https = require('https');  
  
const API_ENDPOINTS = {  
  domains: {  
    base: 'domaincheck.httpapi.com',  
    path: '/api/domains/available.json',  
    params: 'auth-userid={id}&api-key=  
{key}&domain-name={domain}&tlds={tld}'  
  }  
}
```

```

};

const get = parameters => {
  const { id, key, domain, tld } = parameters;
  const queryString =
API_ENDPOINTS.domains.params.replace('{id}',
id).replace('{key}', key)
                                .replace('{domain}',
domain).replace('{tld}', tld);
  const options = {
    hostname: API_ENDPOINTS.domains.base,
    port: 443,
    path: API_ENDPOINTS.domains.path + '?' +
queryString,
    method: 'GET'
  };

  return new Promise((resolve, reject) => {

    const request = https.request(options,
response => {
      let chunks = [];

      response.on('data', chunk => {
        chunks.push(chunk);
      });

      response.on('end', () => {
        let body = Buffer.concat(chunks);
        resolve(JSON.parse(body.toString()));
      });

      response.on('error', err => {
        reject(err);
      });
    });
  });
};

```

```

        });
    });

    request.on('error', err => {
        reject(err);
    });

    request.end();

    });
};

```

L'endpoint richiede, oltre alla chiave API e all'ID utente, il nome a dominio e la sua estensione. Trattandosi di una query string, occorre prestare attenzione alla codifica dei parametri dell'URL qualora fossero presenti caratteri speciali.

Quindi creiamo una classe specifica.

```

class ResellerClub {
    constructor({ apiKey, id }) {
        this.apiKey = apiKey;
        this.id = id;
    }

    checkDomainAvailability({ name, extension }) {
        const params = {
            id: this.id,
            key: this.apiKey,
            domain: name,
            tld: extension
        };

        return get(params);
    }
}

```

```
    }  
  }  
  
  module.exports = ResellerClub;
```

Possiamo usare la nostra classe in questo modo:

```
'use strict';  
  
require('dotenv').config();  
  
const ResellerClub = require('./lib/ResellerClub');  
const RC = new ResellerClub({  
  apiKey: process.env.RESELLER_CLUB_API_KEY,  
  id: process.env.RESELLER_CLUB_ID  
});  
  
const init = async () => {  
  try {  
    const response = await  
RC.checkDomainAvailability({  
      name: 'gabrieleromanato',  
      extension: 'net'  
    });  
    console.log(response);  
  } catch(err) {  
    console.log(err);  
  }  
};  
  
init();
```

Esempio di risposta:

```
{ 'gabrieleromanato.net': { classkey: 'dotnet',  
status: 'available' } }
```