

Node.js: query DNS con il comando dig

In questo tutorial vedremo come effettuare delle query DNS usando il comando dig in Node.js.

Creiamo una classe di utility che accetta come parametri il nome del dominio e il tipo di record da ricercare.

```
'use strict';

const { exec } = require('child_process');

class Dig {
  constructor(domain, record) {
    this.domain = domain;
    this.record = record;
    this.cmd = `dig +nocmd ${this.domain}
${this.record} +noall +answer`;
  }

  lookup() {
    const self = this;
    return new Promise(( resolve, reject ) => {
      exec(self.cmd, (error, stdout, stderr) =>
      {
        if(error || stderr) {
          reject('DNS lookup failed');
        }
        resolve(stdout);
      });
    });
  }
}
```

```
    }  
  }  
  
  module.exports = Dig;
```

Utilizziamo il metodo `exec()` della classe `core child_process` per eseguire il comando `dig` dalla shell. Con le Promise possiamo restituire l'output del comando o l'eventuale errore generato.

Quindi usiamo questa classe nella nostra applicazione nel modo seguente.

```
'use strict';  
  
const express = require('express');  
const bodyParser = require('body-parser');  
const path = require('path');  
const port = process.env.PORT || 3000;  
const app = express();  
const Dig = require('./lib/Dig');  
  
app.disable('x-powered-by');  
  
app.use('/public',  
  express.static(path.join(__dirname, '/public'), {  
    maxAge: 0,  
    dotfiles: 'ignore',  
    etag: false  
  }));  
  
app.use(bodyParser.json());  
app.use(bodyParser.urlencoded({ extended: false }));  
  
app.get('/', (req, res) => {
```

```
    res.sendFile(path.join(__dirname) +
'/views/index.html');
});

app.post('/api/lookup', async (req, res) => {
    const { domain, type } = req.body;
    const dig = new Dig(domain, type);
    try {
        const lookup = await dig.lookup();
        res.send(lookup);
    } catch(err) {
        res.send(err);
    }
});

if (app.get('env') === 'development') {
    app.use((err, req, res, next) => {
        res.status(err.status || 500);
    });
}

app.use((err, req, res, next) => {
    res.status(err.status || 500);
});

app.listen(port);
```

Codice di esempio

GitHub