

Node.js: how to check SSL certificates with the SSL Labs API and WebSockets

In this tutorial we will see how to use the SSL Labs API to check the quality of an SSL certificate with Node.js and WebSockets.

To use the SSL Labs API you need to make calls at regular intervals to the same endpoint to get the current status of the certificate evaluation process.

To do this, we need to create a class that allows us to repeat the same action without running the risk of memory leaks related to the incorrect handling of the `setInterval()` method.

```
'use strict';

const { EventEmitter } = require('events');

class Scheduler extends EventEmitter {
  constructor(action = function () {}, ms = 2000) {
    super();
    this.action = action;
    this.handle = undefined;
    this.interval = ms;
    this.on('timeout', () => {
      this.action(this);
    });
  }

  start() {
    if (!this.handle) {
```

```

        this.handle = setInterval(() =>
this.emit('timeout'), this.interval);
    }
}

stop() {
    if (this.handle) {
        clearInterval(this.handle);
        this.handle = undefined;
    }
}
}

module.exports = Scheduler;

```

Using events we can control with much more precision the repetition of the same HTTP request to the remote API. We then create a class to make that request. We will use got modules (version 11.8.3) for HTTP requests and validator for data validation.

```

'use strict';

const API_URL =
'https://api.ssllabs.com/api/v3/analyze';
const got = require('got');
const validator = require('validator');

class ApiClient {
    static fetch(host) {
        if(!validator.isFQDN(host)) {
            return Promise.reject(new Error('Invalid
parameter.'));
        }
        return got(`${API_URL}?host=${host}`).json();
    }
}

```

```

    }
}

module.exports = ApiClient;

```

In ExpressJS we can create a WebSocket using the HTTP server instance created by the application. Then we receive the host name from the client and return the data that comes to us from the API remote at regular intervals. We use the ws module to create and manage WebSockets.

```

const wsServer = new ws.Server({ noServer: true });
wsServer.on('connection', socket => {

    socket.on('message', message => {
        const host = message.toString();
        const task = new Scheduler(async timer => {
            try {
                const data = await
Client.fetch(host);
                const { status, endpoints } = data;
                socket.send(JSON.stringify({
                    status,
                    endpoints
                }));
                if(status.toLowerCase() === 'ready')
{
                    timer.stop();
                }
            } catch(err) {
                timer.stop();
            }
        }, 2000);
        task.start();
    });
});

```

```
});  
});
```

When SSL Labs informs us that the tests have finished (the final status is "READY") we also have access to the final grade (from A to F) contained in the endpoint arrays which corresponds to the list of tests performed.