**Generate SSL certificates via OPENSSL.**

openssl genrsa -out mohan-key.pem 2048

openssl req -new -sha256 -key mohan-key.pem -out mohan-csr.pem

openssl x509 -req -in mohan-csr.pem -signkey mohan-key.pem -out mohan-cert.pem

**Server TLS application**

'use strict';

var tls = require('tls');

var fs = require('fs');

const PORT = 1337;

const HOST = '127.0.0.1'

var options = {

    key: fs.readFileSync('mohan-key.pem'),

    cert: fs.readFileSync('mohan-cert.pem')

};

var server = tls.createServer(options, function(socket) {

    // Send a friendly message

    socket.write("I am the server sending you a message.");

    // Print the data that we received

    socket.on('data', function(data) {

        console.log('Received: %s [it is %d bytes long]',

            data.toString().replace(/(\n)/gm,""),

            data.length);

    });

    // Let us know when the transmission is over

    socket.on('end', function() {

        console.log('EOT (End Of Transmission)');

    });

});

// Start listening on a specific port and address

server.listen(PORT, HOST, function() {

    console.log("I'm listening at %s, on port %s", HOST, PORT);

});

// When an error occurs, show it.

server.on('error', function(error) {

    console.error(error);

    // Close the connection after the error occurred.

    server.destroy();

});

**Client TLS application**

'use strict';

var tls = require('tls');

var fs = require('fs');

const PORT = 1337;

const HOST = '127.0.0.1'

// Pass the certs to the server and let it know to process even unauthorized certs.

var options = {

    key: fs.readFileSync('mohan-key.pem'),

    cert: fs.readFileSync('mohan-cert.pem'),

    rejectUnauthorized: false

};

var client = tls.connect(PORT, HOST, options, function() {

    // Check if the authorization worked

    if (client.authorized) {

        console.log("Connection authorized by a Certificate Authority.");

    } else {

        console.log("Connection not authorized: " + client.authorizationError)

    }

    // Send a friendly message

    client.write("I am the client sending you a message.");

});

client.on("data", function(data) {

    console.log('Received: %s [it is %d bytes long]',

        data.toString().replace(/(\n)/gm,""),

        data.length);

    // Close the connection after receiving the message

    client.end();

});

client.on('close', function() {

    console.log("Connection closed");

});

// When an error ocoures, show it.

client.on('error', function(error) {

    console.error(error);

    // Close the connection after the error occurred.

    client.destroy();

});

**Download and Install openssl using the following link**

https://code.google.com/archive/p/openssl-for-windows/downloads

**Material**

URL: <https://github.com/auth0/node-jsonwebtoken/issues/642>

<https://riptutorial.com/node-js/example/19326/tls-socket--server-and-client>