

Practical – 10

Link to my GitHub repository: <https://github.com/RaviThakur322/Node-JS-Practical-GF2023485999/tree/main>

1. Goal: Set up an **mTLS** server that:

Requires client certificates

Supports certificate reload without restart

2. Tasks: Create **mtls-server.js** that watches cert files and updates TLS context.

Project Structure:

mtls-server.js:

```
JS mtls-server.js U X
Practical-10 > JS mtls-server.js > ...
1  import https from 'https';
2  import fs from 'fs';
3  import path from 'path';
4  import crypto from 'crypto';
5
6  const certDir = path.resolve('./certs');
7  const certPath = path.join(certDir, 'server.crt');
8  const keyPath = path.join(certDir, 'server.key');
9  const caPath = path.join(certDir, 'ca.crt');
10
11 function loadOptions() {
12   console.log('Loading certificates...');
13   console.log('Server cert size:', fs.statSync(certPath).size);
14   console.log('Key size:', fs.statSync(keyPath).size);
15   console.log('CA cert size:', fs.statSync(caPath).size);
16
17   return {
18     key: fs.readFileSync(keyPath, 'utf8'),
19     cert: fs.readFileSync(certPath, 'utf8'),
20     ca: fs.readFileSync(caPath, 'utf8'),
21     requestCert: true,
22     rejectUnauthorized: true,
23     secureOptions:
24       crypto.constants.SSL_OP_NO_TLSv1 |
25       crypto.constants.SSL_OP_NO_TLSv1_1,
26   };
27 }
28
29 let options = loadOptions();
30
31 const server = https.createServer(options, (req, res) => {
32   if (!req.socket.authorized) {
33     res.writeHead(401, { 'Content-Type': 'text/plain' });
34     res.end('Unauthorized client certificate');
35     console.log('Unauthorized client tried to connect');
36     return;
37   }
38 })
```

```
JS mTLS-server.js U X
Practical-10 > JS mTLS-server.js > ...
31 const server = https.createServer(options, (req, res) => {
38   res.writeHead(200, { 'Content-Type': 'text/plain' });
39   res.end('mTLS connection established');
40   console.log('Authorized client connected');
41 });
42
43 server.listen(8443, () => {
44   console.log('mTLS server listening on port 8443');
45 });
46
47 server.on('error', (err) => {
48   console.error('Server error:', err);
49 });
50
51 [certPath, keyPath, caPath].forEach((filePath) => {
52   fs.watchFile(filePath, () => reloadCert());
53 });
54
55 function reloadCert() {
56   try {
57     console.log('Reloading TLS certificates...');
58     options = loadOptions();
59     const secureContext = https.createSecureContext(options);
60     server.setSecureContext(secureContext);
61     console.log('TLS certificates reloaded successfully');
62   } catch (err) {
63     console.error('Failed to reload certificates:', err);
64   }
65 }
```

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
● PS C:\Users\sharm\nodejs> cd Practical-10
○ PS C:\Users\sharm\nodejs\Practical-10> node mTLS-server
Loading certificates...
Server cert size: 1108
Key size: 1732
CA cert size: 1130
mTLS server listening on port 8443
Authorized client connected
Authorized client connected
Authorized client connected
```

