assignment no 8: To use pre-built and user defined modules in Node.js Application

---

1. With the help of suitable examples use following Node.js pre-built modules in Node.js Application.

A. File System Module (fs)

```
12
13    const fs = require('fs');
14
15    fs.readFile('./sample.txt','utf-8',(err,data)=>{
16        if(!err)
17            console.log(data);
18    })
```

PROBLEMS   OUTPUT   **TERMINAL**   DEBUG CONSOLE

```
const fs = require('fs');

fs.writeFile('./sample.txt','Welcome',(err)=>{
    if(err)
        console.log(err);
})
```

Output ->

```
📄 sample.txt
1    Welcome
```

B. Secure HTTP Module (https)

```js
JS app.js > ...
   1    const express = require('express');
   2    const https = require('https');
   3    const path = require('path');
   4    const fs = require('fs');
   5
   6    const app = express();
   7
   8    app.get('/',(req,res)=>{
   9        res.send("Hello from SSL");
  10    })
  11
  12    const sslServer = https.createServer({
  13        key : '',
  14        cert:''
  15    } , app)
  16
  17
  18
  19    sslServer.listen(3443,()=>{
  20        console.log("Secure server on port 3443");
  21    })
  22    app.listen(8080,()=>{
  23        console.log("Server is running on port 8080");
  24    })
  25
```

```
Generating RSA private key, 2048 bit long modulus (2 primes)
......+++++
.................................................................+++++
e is 65537 (0x010001)
```

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:pavanshinde74947@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

```
Signature ok
subject=C = IN, ST = Some-State, O = Internet Widgits Pty Ltd, emailAddress = pavanshinde74947@gmail.com
Getting Private key
PS C:\Users\ASUS\OneDrive\Desktop\PL3-Assignment\7\cert>
```
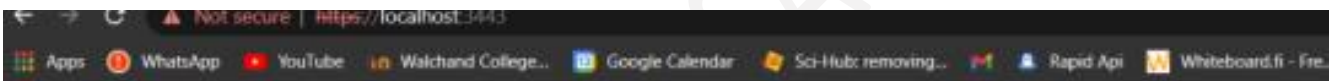
```js
app.js > [ ] sslServer > ⚙ cert
 1    const express = require('express');
 2    const https = require('https');
 3    const path = require('path');
 4    const fs = require('fs');
 5
 6    const app = express();
 7
 8    app.get('/',(req,res)=>{
 9        res.send("Hello from SSL");
10    })
11
12    const sslServer = https.createServer({
13        key : fs.readFileSync(path.join(__dirname,'cert','key.pem')),
14        cert: fs.readFileSync(path.join(__dirname,'cert','cert.pem'))
15    } , app)
16
17
18
19    sslServer.listen(3443,()=>{
20        console.log("Secure server on port 3443");
21    })
22    app.listen(8080,()=>{
23        console.log("Server is running on port 8080");
24    })
25
26
```
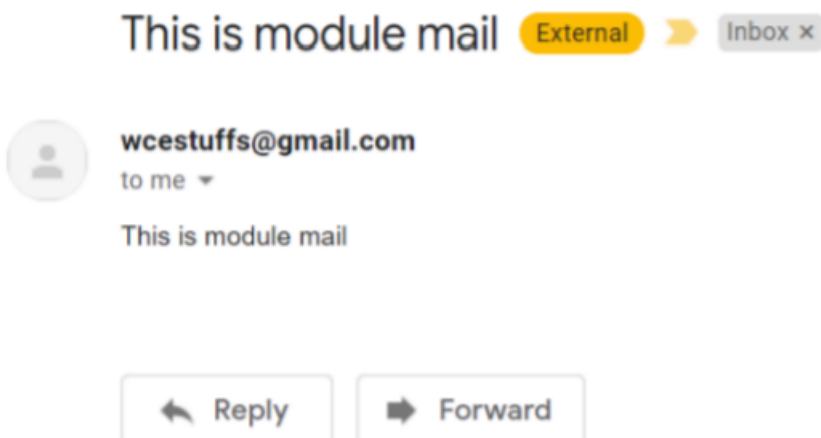
← → C ⚠ Not secure | https://localhost:3443

Apps ⊙ WhatsApp ▶ YouTube in Walchand College.. 📅 Google Calendar 🔴 Sci-Hub: removing.. M 📧 Rapid Api 📋 Whiteboard.fi - Fre..

Hello from SSL

Certificate   ×

General Details Certification Path

    Certificate Information

    This CA Root certificate is not trusted. To enable trust,
    install this certificate in the Trusted Root Certification
    Authorities store.

        Issued to: Internet Widgts Pty Ltd

        Issued by: Internet Widgts Pty Ltd

        Valid from 23-11-2021 to 23-11-2022

                                        Issuer Statement

                                              OK

C. Nodemailer Module (nod emailer)

```js
const nodemailer = require("nodemailer");

const sendMail = async(mail) => {
    var transporter = nodemailer.createTransport({
        service: 'Gmail',
        auth: {
            user: 'wcestuffs@gmail.com',
            pass:
        }
    });

    console.log('created');
    transporter.sendMail({
    from: 'wcestuffs@gmail.com',
      to: mail,
      subject: 'This is module mail',
      text: 'This is module mail'
    });
}

module.exports = sendMail
```

```js
const express = require('express')
const sendMail = require('./mailer.js')


const app = express()


app.get('/send',(req,res) =>{
    const { mail } = req.query

    sendMail(mail)

    res.send('Done')

})

app.listen(3000)
```

Email from http server -

This is module mail External ▶ Inbox ×

wcestuffs@gmail.com
to me ▼

This is module mail

↩ Reply   ➡ Forward

2. Create a user defined module and use it in Node.js Application that uses HTTP server

```
const nodemailer = require("nodemailer");

const sendMail = async(mail) ⇒ {
    var transporter = nodemailer.createTransport({
        service: 'Gmail',
        auth: {
            user: 'wcestuffs@gmail.com',
            pass:
        }
    });

    console.log('created');
    transporter.sendMail({
    from: 'wcestuffs@gmail.com',
        to: mail,
        subject: 'This is module mail',
        text: 'This is module mail'
    });
}

module.exports = sendMail
```

```
1   const express = require('express')
2   const sendMail = require('./mailer.js')
3
4
5   const app = express()
6
7
8   app.get('/send',(req,res) ⇒{
9       const { mail } = req.query
10
11      sendMail(mail)
12
13      res.send('Done')
14
15  })
16
17  app.listen(3000)
```

This is module mail External ▸ Inbox ×

wcestuffs@gmail.com
to me ▾

This is module mail

↩ Reply   ➡ Forward