How to Create HTTPS Communication With Express:

Generating Private Key and Certificate

- Go to the *bin* folder and then create the private key and certificate by typing the following at the prompt:

```
openssl genrsa 1024 > private.key
openssl req -new -key private.key -out cert.csr
openssl x509 -req -in cert.csr -signkey private.key -out certificate.pem
```

Note for Windows Users

- If you are using a Windows machine, you may need to install openssl. You can find some openssl binary distributions here. Also, this article gives the steps for generating the certificates in Windows. Another article provides similar instructions. Here's an online service to generate self-signed certificates.

Configuring the HTTPS Server

- Open the *www* file in the *bin* directory and update its contents as follows:

```
==================================================
. . .

var https = require('https');
var fs = require('fs');

. . .

app.set('secPort',port+443);

. . .

/**
 * Create HTTPS server.
 */

var options = {
  key: fs.readFileSync(__dirname+'/private.key'),
  cert: fs.readFileSync(__dirname+'/certificate.pem')
};

var secureServer = https.createServer(options,app);
```

```
/**
 * Listen on provided port, on all network interfaces.
 */

secureServer.listen(app.get('secPort'), () => {
   console.log('Server listening on port ',app.get('secPort'));
});
secureServer.on('error', onError);
secureServer.on('listening', onListening);

. . .
```

```
===================================================
```

Open *app.js* and add the following code to the file:

```
. . .

// Secure traffic only
app.all('*', (req, res, next) => {
  if (req.secure) {
    return next();
  }
  else {
    res.redirect(307, 'https://' + req.hostname + ':' + app.get('secPort') +
req.url);
  }
});

. . .
```