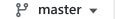


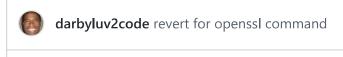
Code Pull requests Actions Projects Security Insights



/ opensel

 ${\mathfrak V}$ 

**fullstack-angular-and-springboot** / bonus-content / secure-https-communication / **openssl-setup.md** 



৪২ 1 contributor

# OpenSSL Setup - Generate key and selfsigned certificate

This document includes instructions for OpenSSL setup. It also includes steps for generating key and self-signed certificate.

There are different steps for each operating system. Choose the steps for your environment.

- MS Windows
- Mac or Linux

# Free Book: OpenSSL Cookbook

A free book on OpenSSL is available online: OpenSSL Cookbook, by Ivan Ristić

# **MS Windows**

### **Install OpenSSL**

For MS Windows, you need to install OpenSSL

- 1. In your web browser, visit the link: https://slproweb.com/products/Win32OpenSSL.html
- 2. In the table, move to the entry: Win64 OpenSSL v1.1.x Light.
- 3. Select the MSI download link
- 4. Once downloadeded to your computer, run the MSI file
- 5. During the installation process, select all of the defaults.
- 6. Update your system path environment variable to point to the openssl installation
  - a. Open the MS Windows Control Panel
  - b. Select **System > Advanced System Settings**
  - c. Click Environment Variables
  - d. In the System variables section, select the Path variable and click the Edit button.
  - e. At the beginning of the path, add: c:\Program Files\OpenSSL-Win64\bin;
    - **NOTE:** Be sure to update with the installation directory on your computer accordingly.
  - f. Click **Ok** and proceed to close all of the dialogs.

#### **Verify OpenSSL installation**

Let's verify the OpenSSL installation

- 1. Open a new command-prompt window.
- 2. Type the following command:

```
openssl help
```

3. You will see the version of openssl installed. If so then openssl is installed successfully. :-)

## **Generate Key and Self-Signed Certificate**

- 1. Open a command-prompt window.
- 2. Move into the directory of your Angular ecommerce project.

cd angular-ecommerce

3. Create a new directory for your output files

```
mkdir ssl-localhost
```

- 4. Create a configuration file for the OpenSSL utility.
  - a. In the directory: angular-ecommerce
  - b. Create a new file named: localhost.conf
- 5. Open the localhost.conf file and enter the following:

```
[req]
# Don't prompt the user when running openssl certificate generation
prompt = no
# Reference to the section containing the Distinguished Name (information
about your company/entity)
distinguished_name = dn
[dn]
# Country, State and Locality (city)
C = US
ST = Pennsylvania
L = Philadelphia
# Organization and Organizational Unit (department name, group name)
0 = luv2code
OU = Training
# Common Name (fully qualified domain name of your website server)
CN = localhost
```

- 6. Save the file.
- 7. In the terminal window, run this command to generate the key and certificate. Be sure to enter this command as a single line.

```
openssl req -x509 -out ssl-localhost\localhost.crt -keyout ssl-
localhost\localhost.key -newkey rsa:2048 -nodes -sha256 -days 365 -config
localhost.conf
```

Argument	Description
req -x509	generate X.509 certificate
-out ssl- localhost/localhost.crt	name of output certificate file
-keyout ssl- localhost/localhost.key	name of output key file
-newkey rsa:2048	create new certificate request and a new private key using algorithm RSA and key size of 2048 bits
-nodes	No DES encryption. The generated private key will not be encrypted
-sha256	Use the SHA256 message digest to sign the request
-days 365	Certificate is valid for 365 days
-config localhost.conf	Name of config file

Detailed docs available here.

8. This command generates the following output:

```
Generating a 2048 bit RSA private key
. . . . . . . +++
writing new private key to 'ssl-localhost/localhost.key'
```

- 9. The command generates two files: localhost.crt and localhost.key.
- 10. View the newly generated files in the ssl-locahost directory.

```
dir ssl-localhost
```

Sample output

localhost.crt localhost.key

11. View the contents of your certificate.

openssl x509 -noout -text -in ssl-localhost/localhost.crt

#### Sample Output

```
Certificate:
    Data:
        Version: ...
        Serial Number: 13535095018565170476 (0xbbd6513516bc752c)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=Pennsylvania, L=Philadelphia, O=luv2code,
OU=Training, CN=localhost
        Validity
            Not Before: May 29 21:25:12 2021 GMT
            Not After: May 29 21:25:12 2022 GMT
        Subject: C=US, ST=Pennsylvania, L=Philadelphia, O=luv2code,
OU=Training, CN=localhost
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                DNS:localhost
    Signature Algorithm: sha256WithRSAEncryption
        a2:9c:22:7c:73:ed:03:3f:ec:00:ce:c0:f6:0b:20:b4:09:6d:
```

Congrats! You have successfully generated a key and self-signed certificate. You can now return to the videos and continue with the course.

## Mac or Linux

On Mac/Linux, openssl is already included with the operating system. There is nothing additional to install.

#### **Generate Key and Self-Signed Certificate**

- 1. Open a terminal window.
- 2. Move into the directory of your Angular ecommerce project.

```
cd angular-ecommerce
```

3. Create a new directory for your output files.

```
mkdir ssl-localhost
```

- 4. Create a configuration file for the OpenSSL utility.
  - a. In the directory: angular-ecommerce
  - b. Create a new file named: localhost.conf
- 5. Open the localhost.conf file and enter the following:

```
[req]
# Don't prompt the user when running openssl certificate generation
prompt = no
# Reference to the section containing the Distinguished Name (information
about your company/entity)
distinguished_name = dn
[dn]
# Country, State and Locality (city)
C = US
ST = Pennsylvania
L = Philadelphia
# Organization and Organizational Unit (department name, group name)
0 = luv2code
OU = Training
# Common Name (fully qualified domain name of your website server)
CN = localhost
```

- 6. Save the file.
- 7. In the terminal window, run this command to generate the key and certificate.

```
openssl req -x509 \
  -out ssl-localhost/localhost.crt \
  -keyout ssl-localhost/localhost.key \
  -newkey rsa:2048 -nodes -sha256 -days 365 \
  -config localhost.conf
```

Argument	Description
req -x509	generate X.509 certificate

Argument	Description
-out ssl- localhost/localhost.crt	name of output certificate file
-keyout ssl- localhost/localhost.key	name of output key file
-newkey rsa:2048	create new certificate request and a new private key using algorithm RSA and key size of 2048 bits
-nodes	No DES encryption. The generated private key will not be encrypted
-sha256	Use the SHA256 message digest to sign the request
-days 365	Certificate is valid for 365 days
-config localhost.conf	Name of config file

Detailed docs available here.

8. This command will generate the following output:

```
Generating a 2048 bit RSA private key
. . . . . . . +++
.......+++
writing new private key to 'ssl-localhost/localhost.key'
```

310 lines (224 sloc) | 9.31 KB

10. View the newly generated files in the ssl-localhost directory.

ls ssl-localhost

Sample output

localhost.crt localhost.key

11. View the contents of your certificate.

openssl x509 -noout -text -in ssl-localhost/localhost.crt

#### Sample Output

```
Certificate:
    Data:
        Version: ...
        Serial Number: 13535095018565170476 (0xbbd6513516bc752c)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=Pennsylvania, L=Philadelphia, O=luv2code,
OU=Training, CN=localhost
        Validity
            Not Before: May 29 21:25:12 2021 GMT
            Not After : May 29 21:25:12 2022 GMT
        Subject: C=US, ST=Pennsylvania, L=Philadelphia, O=luv2code,
OU=Training, CN=localhost
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                DNS:localhost
    Signature Algorithm: sha256WithRSAEncryption
        a2:9c:22:7c:73:ed:03:3f:ec:00:ce:c0:f6:0b:20:b4:09:6d:
        . . .
```

Congrats! You have successfully generated a key and self-signed certificate. You can now return to the videos and continue with the course.