1. SSL jetty Configuration (<https://www.eclipse.org/jetty/documentation/current/jetty-ssl-distribution.html#_default_jsse_ssl_configuration>)

|  |  |
| --- | --- |
| Version: 9.4.20.v20190813 |  |

|  |  |  |
| --- | --- | --- |
| **SSL in the Jetty Distribution** | | |
| [**Previous**](https://www.eclipse.org/jetty/documentation/current/configuring-ssl.html) | **Chapter 6. Configuring Jetty Connectors** [**Home**](https://www.eclipse.org/jetty/documentation/current/index.html) | [**Next**](https://www.eclipse.org/jetty/documentation/current/configuring-security.html) |

[**Contact the core Jetty developers at www.webtide.com**](http://www.webtide.com/)

*private support for your internal/customer projects ... custom extensions and distributions ... versioned snapshots for indefinite support ... scalability guidance for your apps and Ajax/Comet projects ... development services for sponsored feature development*

**SSL in the Jetty Distribution**

[Default JSSE SSL Configuration](https://www.eclipse.org/jetty/documentation/current/jetty-ssl-distribution.html#_default_jsse_ssl_configuration)

[Conscrypt SSL Configuration](https://www.eclipse.org/jetty/documentation/current/jetty-ssl-distribution.html#jetty-conscrypt-distribution)

[Client Certificate Authentication](https://www.eclipse.org/jetty/documentation/current/jetty-ssl-distribution.html#client-certificate-authentication)

When making use of the Jetty Distribution, enabling SSL support is as easy as activating the appropriate module. Jetty supports both the default [JSSE](https://docs.oracle.com/javase/8/docs/technotes/guides/security/jsse/JSSERefGuide.html) provider and the [Conscrypt](https://github.com/google/conscrypt/" \t "_top) provider as SSL implementations.

**Default JSSE SSL Configuration**

For the default SSL support, simply activate the ssl [module](https://www.eclipse.org/jetty/documentation/current/startup-modules.html):

$ cd /path/to/mybase

$ java -jar ${JETTY\_HOME}/start.jar --add-to-startd=ssl

INFO : server initialised (transitively) in ${jetty.base}/start.d/server.ini

INFO : ssl initialised in ${jetty.base}/start.d/ssl.ini

INFO : Base directory was modified

$ tree

.

├── etc

│   └── keystore

└── start.d

├── server.ini

└── ssl.ini

When you open start.d/ssl.ini, you will see several commented properties ready for use when configuring SslContextFactory basics.

To highlight some of the more commonly used properties:

**jetty.ssl.host**

Configures which interfaces the SSL/TLS Connector should listen on.

**jetty.ssl.port**

Configures which port the SSL/TLS Connector should listen on.

**jetty.httpConfig.securePort**

If a webapp needs to redirect to a secure version of the same resource, then this is the port reported back on the response locationline (having this be separate is useful if you have something sitting in front of Jetty, such as a Load Balancer or proxy).

**jetty.sslContext.keyStorePath**

Sets the location of the keystore that you configured with your certificates.

**jetty.sslContext.keyStorePassword**

Sets the Password for the keystore.

**Conscrypt SSL Configuration**

Enabling Conscrypt SSL is just as easy as default SSL - enable both the conscrypt and ssl [modules](https://www.eclipse.org/jetty/documentation/current/startup-modules.html):

$ cd ${JETTY\_HOME}

$ java -jar ../start.jar --add-to-start=ssl,conscrypt

ALERT: There are enabled module(s) with licenses.

The following 1 module(s):

+ contains software not provided by the Eclipse Foundation!

+ contains software not covered by the Eclipse Public License!

+ has not been audited for compliance with its license

Module: conscrypt

+ Conscrypt is distributed under the Apache Licence 2.0

+ https://github.com/google/conscrypt/blob/master/LICENSE

Proceed (y/N)? y

INFO : server transitively enabled, ini template available with --add-to-start=server

INFO : conscrypt initialized in ${jetty.base}/start.d/conscrypt.ini

INFO : ssl initialized in ${jetty.base}/start.d/ssl.ini

MKDIR : ${jetty.base}/lib/conscrypt

DOWNLD: https://repo1.maven.org/maven2/org/conscrypt/conscrypt-openjdk-uber/1.0.0.RC11/conscrypt-openjdk-uber-1.0.0.RC11.jar to ${jetty.base}/lib/conscrypt/conscrypt-uber-1.0.0.RC11.jar

MKDIR : ${jetty.base}/etc

COPY : ${jetty.home}/modules/conscrypt/conscrypt.xml to ${jetty.base}/etc/conscrypt.xml

COPY : ${jetty.home}/modules/ssl/keystore to ${jetty.base}/etc/keystore

INFO : Base directory was modified

No additional Conscrypt configuration is needed. SSL-specific parameters, like keyStorePath and keyStorePassword can still configured as in the example above, making use of the ${JETTY\_BASE}/start.d/ssl.ini file.

**Client Certificate Authentication**

To enable client certificate authentication in the Jetty Distribution, you need to enable the both the ssl and https modules.

$ cd /path/to/mybase

$ java -jar /path/to/jetty-dist/start.jar *--add-to-startd=ssl,https*

**$JETTY\_BASE/start.d/ssl.ini.**

# Module: ssl

--module=ssl

jetty.ssl.host=0.0.0.0

jetty.ssl.port=8583

jetty.sslContext.keyStorePath=etc/keystore

jetty.sslContext.trustStorePath=etc/truststore

jetty.sslContext.keyStorePassword=OBF:

jetty.sslContext.keyManagerPassword=OBF:

jetty.sslContext.trustStorePassword=OBF:

# Enable client certificate authentication.

jetty.sslContext.needClientAuth=true

**$JETTY\_BASE/start.d/https.ini.**

*# Module: https*

--module=https

|  |  |  |
| --- | --- | --- |
| [**Previous**](https://www.eclipse.org/jetty/documentation/current/configuring-ssl.html) | [**Top**](https://www.eclipse.org/jetty/documentation/current/configuring-connectors.html) | [**Next**](https://www.eclipse.org/jetty/documentation/current/configuring-security.html) |
| Configuring SSL/TLS | [**Home**](https://www.eclipse.org/jetty/documentation/current/index.html) | Chapter 7. Configuring Security |

See an error or something missing? [**Contribute to this documentation at  Github!**](http://github.com/eclipse/jetty.project)*(Generated: 2019-08-13)*