**Issue** : Error when accessing sms service via https using app.sms.baseUrl = https://www.myvaluefirst.com/smpp/sendsms?username={username}&password={password}&from={from}&to={to}&text={text}

**Exception:** javax.net.ssl.SSLException: Unsupported record version Unknown-0.0

**Observations**: java 8 by default uses TLSv1.2 protocol. And for backward compatibility, It specify that a server using TLSv1.0 MUST negotiate with the client using higher version of TLS protocol.

<http://www.ietf.org/rfc/rfc5246.txt>

[page 86]

If a TLS server receives a ClientHello containing a version number

greater than the highest version supported by the server, it MUST

reply according to the highest version supported by the server.

But some buggy TLSv1.0 servers don't follow this specification and simply close the connection when the client offers a version newer than TLSv1.0.

[page 87]

Note: some server implementations are known to implement version

negotiation incorrectly. For example, there are buggy TLS 1.0

servers that simply close the connection when the client offers a

version newer than TLS 1.0. Also, it is known that some servers will

refuse the connection if any TLS extensions are included in

ClientHello. Interoperability with such buggy servers is a complex

topic beyond the scope of this document, and may require multiple

connection attempts by the client.

The JBossWeb/2.0.1.GA server of [myvaluefirst.com](http://myvaluefirst.com)/smpp/ is one of them.

[https->connection:] The connection uses TLS 1.0.

The server does not support the TLS renegotiation extension.

<http://www.oracle.com/technetwork/java/javase/documentation/cve-2014-3566-2342133.html>

**S**o if we use java.lang.System.setProperty("https.protocols", “TLSv1"); The error is gone.

But as the property is set globally, a connection to any other service will also be negotiated to TLSv1.0 when the server of the service is using higher version of TLS.