**ИНТЕГРАЦИЯ Tomcat 9.0.0.M9 и JTLS**

Сначала следует ознакомиться с инструкцией *How\_to\_set\_gost\_tls\_tomcat.docx* в папке *Doc\WebServerIntegration\Tomcat*.

**Дополнительные сведения.**

В связи с серьезными изменениями, которым подвергся код *tomcat* (в частности, tomcat-coyote.jar, начиная примерно с версии 8.5), сделан адаптер *JCPTomcatAdapter*, расширяющий стандартный класс *JCPJSSEImplementation*, для обеспечения взаимодействия *tomcat* *9.0.0.M9* и *cpSSL*.

В tomcat'е появились два класса, предоставляющих несколько разные алгоритмы работы с *Connector*'ом tomcat'а (*lib\tomcat-coyote.jar\org\apache\tomcat\util\net\):*

*jsse/JSSEImplementation*

*openssl/OpenSSLImplementation*

По умолчанию, когда в *Connector*’е не задан параметр *sslImplementationName*, используется *OpenSSLImplementation*. Но *JSSEImplementation* ближе всего к старому *tomcat*.

[*https://tomcat.apache.org/tomcat-9.0-doc/config/http.html#SSL\_Support\_-\_SSLHostConfig*](https://tomcat.apache.org/tomcat-9.0-doc/config/http.html#SSL_Support_-_SSLHostConfig)

[*https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html*](https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html)

Класс *JCPJSSEImplementation* модуля *JCPTomcatAdapter* расширяет класс *JSSEImplementation*, обеспечивая корректное чтение параметров коннектора в новом формате (*Connector* -> *SSLHostConfig* -> *Certificate*) для *cpSSL* (некоторые параметры типа алгоритмов не используются).

Исходные коды адаптера приложены в виде *maven*-проекта *JCPTomcatAdapter*. Проект имеет несколько зависимостей - *tomcat-coyote, tomcat-juli, JCP, cpSSL, JCPRevCheck, JCryptoP*. Собирать проект следует с помощью *JDK* 10. Собранная библиотека *JCPTomcatAdapter* лежит в папке *Doc\WebServerIntegration\Tomcat9\JCPTomcatAdapter\target* проекта.

Для использования *JCPTomcatAdapter* готовую библиотеку следует положить в папку *{tomcat\_dir}\lib\ tomcat’*а, где *{tomcat\_dir} –* папка установленного веб-сервера*.* Также в эту папку необходимо положить следующие библиотеки из дистрибутива JCP:

*ASN1P.jar*

*asn1rt.jar*

*cpSSL.jar*

*forms\_rt.jar*

*JCP.jar*

*JCPRevCheck.jar*

*JCPRevTools.jar*

*JCryptoP.jar*

\*\*\*

Пример коннектора (*tomcat* 9.0.0.M9) с включенной аутентификацией клиента:

*<Connector*

*port="8443"*

*protocol="org.apache.coyote.http11.Http11NioProtocol"*

***sslImplementationName****="****ru.CryptoPro.ssl.tomcat.jsse.JCPJSSEImplementation****"*

*maxThreads="150"*

*secure="true"*

*scheme="https"*

*SSLEnabled="true" >*

*<SSLHostConfig*

*protocols="TLSv1"*

*sslProtocol="GostTLS"*

*truststoreFile="c:\tomcat9\test\_ca.store"*

*truststorePassword="1"*

*truststoreProvider="JCP"*

*truststoreType="CertStore"*

*certificateVerification="true" >*

*<Certificate*

*certificateKeystoreFile="c:\tomcat9\test\_ca.store"*

*certificateKeystorePassword="1"*

*certificateKeystoreProvider="JCP"*

*certificateKeystoreType="HDImageStore" />*

*</SSLHostConfig>*

*</Connector>*

Используется реализация *JCPJSSEImplementation* модуля *JCPTomcatAdapter*.

При инициализации *JCPJSSEImplementation* происходит добавление необходимых для работы провайдеров: *JCP, cpSSL, JCPRevCheck, JCryptoP*.

Выбор ключевого контейнера сервера происходит по типу контейнера *certificateKeystoreType* и паролю для доступа к нему *certificateKeystorePassword*. Т.к. передается только информация о доверенном хранилище корневых сертификатов, то его пароль должен совпадать с паролем контейнера, который предполагается использовать в качестве серверного, т.е. *truststoreFile* совпадает с *certificateKeystoreFile*, *truststorePassword* - с *certificateKeystorePassword*.

*truststoreFile* - хранилище доверенных корневых сертификатов формата *CertStore* (JCP).

*truststorePassword* - пароль к нему.

*truststoreProvider* - провайдер для работы с хранилищем.

*truststoreType* - тип хранилища.

*certificateVerification* - параметр определяет, включена ли на сервере аутентификация клиента.

*certificateKeystoreFile* - хранилище доверенных корневых сертификатов формата *CertStore* (JCP).

*certificateKeystorePassword* - пароль к нему.

*certificateKeystoreProvider* - провайдер для работы с хранилищем.

*certificateKeystoreType* - тип хранилища.

В примере *c:\tomcat9\test\_ca.store* - это хранилище корневых сертификатов.

При использовании JCSP вместо JCP в качестве провайдера в секции *SSLHostConfig/Certificate* следует задать в *certificateKeystoreProvider=”JCSP”* и *certificateKeystoreType=”<тип\_контейнера>”*, где *тип\_контейнера* – это тип контейнера, поддерживаемый JCSP и определяющий место хранения серверного контейнера, например, реестр (тип контейнера «REGISTRY»), папка (тип контейнера «HDIMAGE») и другие.

Для использования провайдера JCSP вместо JCP необходимо добавить этот провайдер методом Security.addProvider(new JCSP()) в конструкторе класса *JCPJSSEImplementation* (по аналогии с другими провайдерами), а также добавить JCSP в зависимости в файле pom.xml и пересобрать *JCPTomcatAdapter.*

Перед сборкой проекта *JCPTomcatAdapter* необходимосоздать локальный репозиторий в папке проекта – repository, и установить в него модули *JCP.jar* ,*cpSSL.jar, JCryptoP.jar, JCPRenCheck.jar* (при использовании JCSP – *JCSP.jar*) Например, так (в командной строке):

*cd* *JCPTomcatAdapter*

*mkdir repository*

*mvn org.apache.maven.plugins:maven-install-plugin:2.3.1:install-file -Dfile="JCP.jar" -DgroupId=ru.cprocsp -DartifactId=JCP -Dversion=2.5 -Dpackaging=jar -DlocalRepositoryPath="repository" -DcreateChecksum=true*

*mvn org.apache.maven.plugins:maven-install-plugin:2.3.1:install-file -Dfile="cpSSL.jar" -DgroupId=ru.cprocsp -DartifactId=cpSSL -Dversion=2.5 -Dpackaging=jar -DlocalRepositoryPath="repository" -DcreateChecksum=true*

*mvn org.apache.maven.plugins:maven-install-plugin:2.3.1:install-file -Dfile="JCryptoP.jar" -DgroupId=ru.cprocsp -DartifactId=JCryptoP -Dversion=2.5 -Dpackaging=jar -DlocalRepositoryPath="repository" -DcreateChecksum=true*

*mvn org.apache.maven.plugins:maven-install-plugin:2.3.1:install-file -Dfile="JCPRevCheck.jar" -DgroupId=ru.cprocsp -DartifactId=JCPRevCheck -Dversion=2.5 -Dpackaging=jar -DlocalRepositoryPath="repository" -DcreateChecksum=true*

*file* – устанавливаемый файл (положение зависит от места хранения дистрибутива), *groupId* – идентификатор группы (см. *pom.xml*), *artifactId* – идентификатор модуля (см. *pom.xml*), *version* – версия модуля (см. *pom.xml*, зависит от версии дистрибутива), *localRepositoryPath* – папка локального дистрибутива. Для проверки соответствия путей, имен и идентификаторов следует смотреть *pom.xml* проекта. Версия плагина *maven-install-*plugin может отличаться*.*

После открытия проекта *JCPTomcatAdapter* зависимости должны быть отображены корректно. Сборка производится командой *build: clean compile package*