1. How to access https domain

* Step 1: export file domain.crt
* Step 2: open cmd and go to C:\Program Files\Java\jdk1.8.0\_144\jre\lib\security
* Step 3: add https domain by using keytool cmd like:

*keytool -import -v -trustcacerts -alias oreo88.com -file path\domain.crt -keystore cacerts -keypass changeit -storepass changeit*

**Noted**: yourdomain = *oreo88.com*

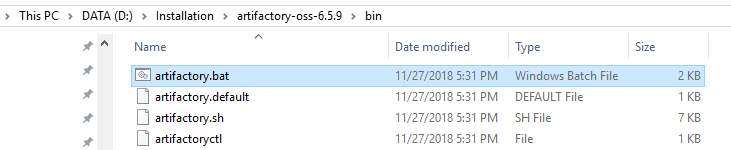
yourcertfile = *domain.crt*

* Step 4: add some lines java code

String certificatesTrustStorePath = **"C:\\Program Files\\Java\\jdk1.8.0\_144\\jre\\lib\\security\\cacerts"**;  
System.*setProperty*(**"javax.net.ssl.trustStore"**, certificatesTrustStorePath);  
System.*setProperty*(**"javax.net.ssl.trustStorePassword"**, **"changeit"**);

Reference link: <https://stackoverflow.com/questions/9619030/resolving-javax-net-ssl-sslhandshakeexception-sun-security-validator-validatore>

1. Install Maven local repository
2. Download Artifactory package, unzip this package, set up environment to bin of the unzipped file
3. Run artifactory.bat file under Administrator



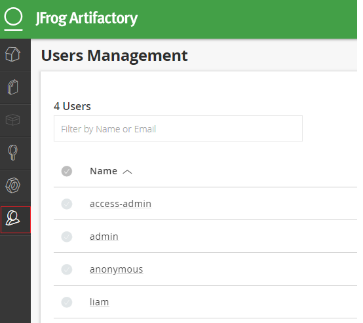
Watch two videos

<https://www.youtube.com/watch?v=dMYcca37WBs>

<https://www.youtube.com/watch?v=Lg4a6Sc_Xco>

1. Configure user permission. There are a default user “***admin/password***” to sign in this page <http://localip:8081/artifactory/webapp/#/home>.

Step 1: Navigate to Admin icon to

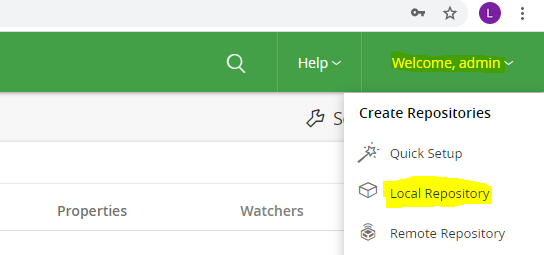
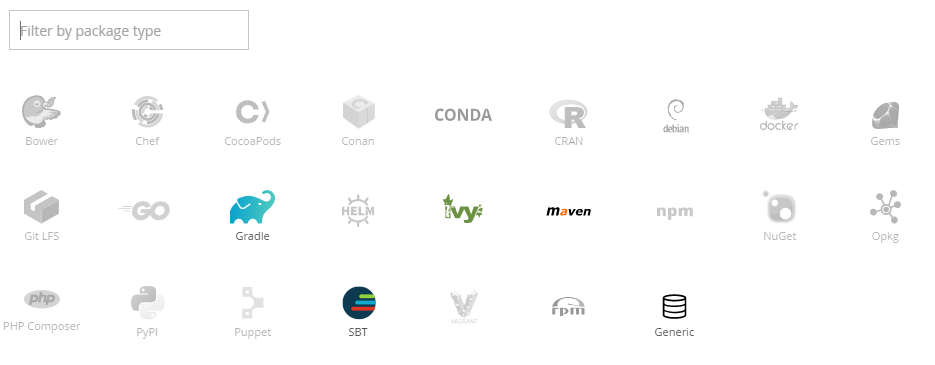


Step 2: Configure steps within <https://inthecheesefactory.com/blog/how-to-setup-private-maven-repository/en>. The purpose of this configuration allows users to have permissions on these repositories such as delete, override, deploy, read etc.

<https://dzone.com/articles/how-to-setup-a-maven-repository-in-minutes>

Step 3: How to create a **lib-local-repository**

Click Welcome Admin > Local Repository > Select Maven type

Step 4: Get Repository info

Click Home > **lib-local-repository**



**Step 5 & 6 is for machines which want to connect Maven repository**

Step 5: copy ***settings.xml*** file to .m2 under *C:\Users\{username}\.m2*

<?xml version="1.0" encoding="UTF-8"?>

<settings xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.1.0 http://maven.apache.org/xsd/settings-1.1.0.xsd" xmlns="http://maven.apache.org/SETTINGS/1.1.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<servers>

<server>

<username>**admin**</username>

<password>**password**</password>

<id>**central**</id>

</server>

<!--server>

<username>**admin**</username>

<password>**password**</password>

<id>**snapshots**</id>

</server-->

</servers>

<profiles>

<profile>

<repositories>

<repository>

<snapshots />

<id>snapshots</id>

<name>CODEWS08-releases</name>

<url>**http://172.21.1.152:8081/artifactory/lib-local-repository**</url>

</repository>

</repositories>

<pluginRepositories>

<pluginRepository>

<snapshots />

<id>central</id>

<name>CODEWS08-releases</name>

<url>**http://172.21.1.152:8081/artifactory/lib-local-repository**</url>

</pluginRepository>

</pluginRepositories>

<id>artifactory</id>

</profile>

</profiles>

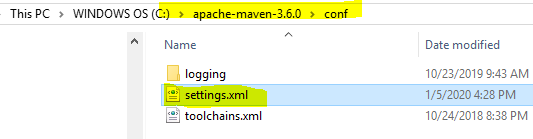
<activeProfiles>

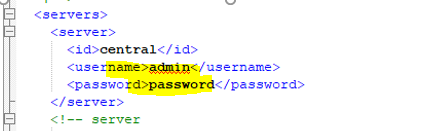
<activeProfile>artifactory</activeProfile>

</activeProfiles>

</settings>

Step 6: On You can set up username and password settings.xml in Maven Home where you configure MAVEN\_HOME. Now, you can run maven deploy to get jar file to this maven repository





Step 7: Configure local maven repository on Jenkins because there is many AD user login on this machine so we have to let it know which .m2 repository Jenkins uses to get settings.xml containing username and password of Artifactory server.