JDK 17 and Java 17

To build Java and Grails based modules using JDK 17 and to run them under Java 17 a few changes are needed.
Please add any additional changes you notice to this page!

Goal

JDK 17 and Java 17 are the current LTS JDK and LTS Java versions since September 2021 and will receive free public updates at least until September 2027.

Currently most modules and the build environments use the previous LTS version, JDK 11 and Java 11. To reduce the maintenance burden JDK 11 and Java 11 should not be used for the next flower releases. Bug fixes and hot fixes for modules of a released flower release continue to use their initial LTS version.

Dockerfile

Copy the sample Dockerfile from https://github.com/folio-org/folio-tools/tree/master/folio-java-docker/openjdk17#sample-module-dockerfile

Jenkinsfile

Change the buildNode from jenkins-agent-jav11 to jenkins-agent-java17. Additionally, if

healthChkCmd = 'curl --silent --fail -o /dev/null http://localhost:8081/admin/health || exit 1'

exists in the Jenkinsfile, change it to

healthChkCmd = 'wget --no-verbose --tries=1 --spider http://localhost:8081/admin/health || exit 1'

GitHub Actions

For projects using GitHub Actions, be sure to upgrade the version of Java used. Under the actions/setup-java@v2 job should be a key java-version; changing this to "17" is sufficient.

POM Files

If declared, be sure to change the Java version and source/target compilation versions. This can be done under the main properties block:

<java.version>17</java.version>
<maven.compiler.source>17</maven.compiler.source>
<maven.compiler.target>17</maven.compiler.target>

Note: for this to work, be sure that the source/target versions are not overridden in the maven-compiler-plugin configuration

Dependencies

Upgrade lombok for Java 17 if the build fails with IllegalAccessError: class lombok.javac.apt.LombokProcessor, for example

MODUBSUB-242 - Upgrade lombok to 1.18.24 for Java 17

Upgrade mockito for Java 17 if the build fails with InaccessibleObjectException: Unable to make protected final, for example

EDGPATRON-95 - Upgrade mockito from 1.95 to 4.6.1 for Java 17

Upgrade pl.project13.maven.git-commit-id-plugin for Java 17 if the build fails with Unable to make ... accessible: module java.base does not "opens java.util" to unnamed module, for example

RMB-930 - Upgrade git-commit-id-plugin fixing java.base does not "opens java.util" for JDK 17

Reflection

Some access using reflection is no longer allowed in Java 17 and causes InaccessibleObjectException. Rewrite, for example: https://github.com/folio-org/mod-login-saml/pull/129/files

Testing on Vagrant

To use Java 17 on the provided Vagrant boxes, the following commands must be used:
# Install JDK/JRE 17
sudo apt update && sudo apt install -y openjdk-17-jdk

# Install new version of Maven, as the one provided with Ubuntu 20.04 has a bug with JDK 17
# See https://github.com/m-thirumal/installation_guide/blob/39187a6e9acff22b6800c7a407370478f1df5a77/maven /upgrade_maven.md for details
wget https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/apache-maven-3.8.6-bin.tar.gz
sudo mkdir -p /usr/local/apache-maven
sudo mv apache-maven-3.8.6-bin.tar.gz /usr/local/apache-maven
cd /usr/local/apache-maven
sudo tar -xzvf apache-maven-3.8.6-bin.tar.gz

# Add environment variables
 tee ~/.profile <<-"EOF"
export M2_HOME=/usr/local/apache-maven/apache-maven-3.8.6
export M2=$M2_HOME/bin
export MAVEN_OPTS="-Xms256m -Xmx512m"
export PATH=$M2:$PATH
EOF

# Then, to synchronize the environment variables you can restart your SSH session, or
source ~/.profile

# Check it works
mvn --version

Help

Ask on #development Slack channel if some module doesn't build or run under JDK 17 or Java 17 and there is no obvious fix.