Maven

1. Установка

Самое тяжёлое было научить комп работать с командой mvn. Скачать архив и разархивировать его - это, конечно, ерунда. Но вот, что нужно прописать переменную среды и показать системе, где находится исполняемый файл, а главное, как именно это сделать. В общем с этим возникла одна из основных проблем.

Я сделал это двумя способами: командой в PowerSchell и с помощью мыши, так сказать. Команду я вообще не запомнил, помню только то, что в ней лучше не допускать ошибок, а то это может привести к плохим последствиям. В общем мышкой через параметры проще и безопасней). Не понял назначение переменной M2\_HOME (переменная для maven). Просто писали\говорили, что её лучше указать.

2. Создание пустого и не только проекта.

Тут всё более-менее понятно. Команда из семинара сработала с первого раза. Но я ещё посмотрел\почитал на эту тему и мне показалось более простым написать только: mvn archetype:generate -> там вылезает достаточно большой список уже "заполненных” проектов, но если нажать Enter без выбора номера, то потом всё будет как обычно: введите groupId, artefactId, version, package и что-то ещё. И пустой проект готов.

3. mvn-команды.

“Поигрался” с командами compile, clean, package, install. И в какой-то момент задался вопросом: “А можно ли запустить код с помощью mvn?” Нашёл команду

mvn exec:java –Dexec.mainClass=”ru.test.App”

Но у меня так и не получилось запустит код с помощью подобной команды. Через java тоже были сложности, однако я разобрался, где неправильно делал и смог запустить скомпилированный код через командную строку. Правда этот факт не помог мне с первоначальной проблемой...

4. Профили

Посмотрел одно видео (правда оно 6-ти летней давности) на эту тему. И немного с этим разобрался и потренировался:

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>1.1</version>

<executions>

<execution>

<phase>validate</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<tasks>

<echo>Hello, i'm running...</echo>

<echo>CURRENT ENV: ${app.env}</echo>

</tasks>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

<!-- </pluginManagement>-->

</build>

<profiles>

<profile>

<id>env-dev</id>

<properties>

<app.env>DEV</app.env>

</properties>

<activation>

<property>

<name>env</name>

<value>develop</value>

</property>

</activation>

</profile>

<profile>

<id>env-prod</id>

<properties>

<app.env>PROD</app.env>

</properties>

<activation>

<activeByDefault>true</activeByDefault>

</activation>

</profile>

</profiles>

</project>

5. Репозитории, конфиг-файлы, скоупы, модули.

Тоже немного посмотрел, почитал.

Касаемо репозиториев, как я понял, есть 3 вида:

* .m2 в домашней директории. Там хранятся все ранее скаченные библиотеки и туда же с помощью команды mvn install кидаются собственные “наработки”. Там же начинает свой поиск maven, когда мы подключаем dependency.
* MavenCentral. Если в первом пункте не нашлось нужной библиотеки идёт сюда. Как поменять это на что-нибудь другое. Я не нашёл.
* Local. Как я понял, если прописано в Pom-файле

<repositories>

<repository>

.............

</repository>

</repositories>

то в таком случае после второго пункта maven идёт сюда. Если такого нет, то пункт два конечный.

Файлы конфигурации есть тоже 3 видов:

* Непосредственно Pom.xml;
* Файл settings в директории ~\.m2. Но по умолчанию его там нет. Можно, соответственно, создать
* Аналогичный файл settings в директории с maven в папке conf\

Приоритет у них от 3 к 1. То есть 3 самый главный.

Scope нужен чтобы maven понимал, в какой момент нам нужна та или иная зависимость. Пишется в соответствующем тэге. Compile - используется по умолчанию, можно не указывать. Test, provided, runtime, import - ещё такие есть (думаю правильно написал).

Также посоздавал модули с помощью pom.xml

<modules>

<module>test-logic</module>

<module>test-gui</module>

</modules>

<parent>

<groupId>ru.test\_maven</groupId>

<artifactId>mavenEx</artifactId>

<version>1.0-SNAPSHOT</version>

</parent>

<artifactId>TestGUI</artifactId>

<version>1.0-SNAPSHOT</version>

Но тут я в подробности вообще не вдавался. Просто в основном ПОМЕ прописал модули, идеей их названия подсветились красным и сочетанием клавиш создал соответствующие модули и в них сгенерировались из ПОМКИ. Только таким способом не создаётся структура проекта в этих модулях. А только сами папки\модули с соответствующими названиями и в них только один файл pom.xml.

Gradle

На эту систему меня особо не хватило. Сам ничего не делал. Только посмотрел пару роликов (не считая, конечно, семинар и лекцию).

Для себя понял, что, когда поднимется данная тема на моём учебном пути или, когда нужно будет посмотреть что-нибудь сверх - нужно подробней изучить эту штуку и пощупать её на практике.

Потому что, как я понял, Gradle набирает обороты. Он быстрее собирает проект, он намного более гибкий (настраиваемый), правда у этого есть обратная сторона - стабильность из-за этого хуже (но это не вина gradle, а просто человеческий фактор - увеличивается вероятность ошибки), файл с конфигурацией более прост и читаем.