**Maven**

Руководство по Maven <http://www.apache-maven.ru/>

Apache Maven — основы <https://habr.com/ru/post/77382/>

Maven - это инструмент для сборки Java проекта: компиляции, создания jar, создания дистрибутива программы, генерации документации. Простые проекты можно собрать в командной строке. Если собирать большие проекты с командной строки, то команда для сборки будет очень длинной, поэтому её иногда записывают в bat/sh скрипт. Но такие скрипты зависят от платформы. Для того чтобы избавиться от этой зависимости и упростить написание скрипта используют инструменты для сборки проекта.

Для платформы Java существуют два основных инструмента для сборки: Ant, Maven и Gradle.

**Терминология Maven**

Вся структура проекта описывается в файле pom.xml (POM – Project Object Model), который должен находиться в корневой папке проекта. Ключевым понятием Maven является артефакт — это, по сути, любая библиотека, хранящаяся в репозитории. Это может быть какая-то зависимость или плагин.

* Зависимости — это те библиотеки, которые непосредственно используются в вашем проекте для компиляции кода или его тестирования.
* Плагины же используются самим Maven'ом при сборке проекта или для каких-то других целей (деплоймент, создание файлов проекта для Eclipse и др.).
* В самом начале работы с Maven, пользователь непременно столкнется с таким понятием как архетип. Архетип — это некая стандартная компоновка файлов и каталогов в проектах различного рода (веб, swing-проекты и прочие). Другими словами, Maven знает, как обычно строятся проекты и в соответствии с архетипом создает структуру каталогов.

Как правило, название артефакта состоит из названия группы, собственного названия и версии. К примеру Spring будет иметь вот такое название в среде Maven: org.springframework.spring:2.5.5. Последний домен означает всегда artifactId, все, что перед ним – groupId.

**Основные преимущества Maven**

Независимость от OS. Сборка проекта происходит в любой операционной системе. Файл проекта один и тот же.

* Управление зависимостями. Редко какие проекты пишутся без использования сторонних библиотек(зависимостей). Эти сторонние библиотеки зачастую тоже в свою очередь используют библиотеки разных версий. Maven позволяет управлять такими сложными зависимостями. Что позволяет разрешать конфликты версий и в случае необходимости легко переходить на новые версии библиотек.
* Возможна сборка из командной строки. Такое часто необходимо для автоматической сборки проекта на сервере (Continuous Integration).
* Хорошая интеграция со средами разработки. Основные среды разработки на java легко открывают проекты которые собираются c помощью Maven. При этом зачастую проект настраивать не нужно - он сразу готов к дальнейшей разработке.
* Как следствие - если с проектом работают в разных средах разработки, то Maven удобный способ хранения настроек. Настроечный файл среды разработки и для сборки один и тот же - меньше дублирования данных и соответственно ошибок.
* Декларативное описание проекта.