

Maven

Maven

Maven — инструмент для управления и сборки проектов — настоящий помощник Java-программиста.

Стандартная структура для Maven-проекта:

- в папке *src/main/java* содержатся java-классы;
- в *src/main/resources* — ресурсы, которые использует наше приложение (HTML-страницы, картинки, таблицы стилей и тд);
- *src/test* — для тестов.

dependencies

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>example.com</groupId>
  <artifactId>example</artifactId>
  <version>1.0-SNAPSHOT</version>

  <dependencies>
    <dependency>
      <groupId>commons-io</groupId>
      <artifactId>commons-io</artifactId>
      <version>2.6</version>
    </dependency>
  </dependencies>
</project>
```

Виды Maven-репозиториев

1. внешний (глобальный)
2. локальный, у тебя на компьютере

Все библиотеки, которые ты добавляешь в свои проекты, сохраняются в локальном репозитории. Когда Maven добавляет необходимую зависимость в проект, он сначала проверяет локальный репозиторий на наличие такой библиотеки, и только если не находит ее там — обращается к внешнему.

Фазы maven-проекта

1. `clean` — удаляются все скомпилированные файлы из каталога `target` (место, в котором сохраняются готовые артефакты);
2. `validate` — идет проверка, вся ли информация доступна для сборки проекта;
3. `compile` — компилируются файлы с исходным кодом;
4. `test` — запускаются тесты;
5. `package` — упаковываются скомпилированные файлы (в `jar`, `war` и т.д. архив);
6. `verify` — выполняются проверки для подтверждения готовности упакованного файла;
7. `install` — пакет помещается в локальный репозиторий. Теперь он может использоваться другими проектами как внешняя библиотека;
8. `site` — создается документация проекта;
9. `deploy` — собранный архив копируется в удаленный репозиторий.

Запуск фазы

- через командную строку:

mvn package

- Через IDE

До запуска *package* выполняются фазы *validate*, *compile*, *test*.

Исключение — фаза *clean*. Ее желательно вызывать перед каждой новой сборкой проекта. Фазы можно перечислить через пробел:

mvn clean package.

Также **у каждой фазы** есть пре- и пост-фазы: например, *pre-deploy*, *post-deploy*, *pre-clean*, *post-clean*, но используются они довольно редко.

Плагины

У каждой фазы есть цели (goal). Стандартные цели заложены по умолчанию, дополнительные добавляются Maven-плагинами.

Чтобы добавить Мавен-плагин в проект, его описание, аналогично зависимостям, нужно поместить в pom.xml в тег `<build>` и `<plugins>`.

Пример добавление плагина для проверки того, что все наши внешние библиотеки используют самые последние версии

```
<build>
  <plugins>
    <plugin>
      <groupId>com.soebes.maven.plugins</groupId>
      <artifactId>uptodate-maven-plugin</artifactId>
      <version>0.2.0</version>
      <executions>
        <execution>
          <goals>
            <goal>dependency</goal>
          </goals>
          <phase>validate</phase>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```