Аспектно - ориентированное программирование

Инструменты для сборки и управления

Лекция №6 Антонов В.В.

Что такое maven?

"Maven is a project development management and comprehension tool"

с сайта maven.apache.org

- ● инструмент для сборки и управления
- проектами [на Java] (build tool)
- ● инструмент для управления ЖЦ проекта
- ● инструмент для автоматизации

Другие утилиты для сборки проектов

- shell/bat скрипты
- ● make
- ● cmake
- ● scons
- ● ant

Почему Maven?

- на текущий момент одна из самых широко распространенных утилит для сборки в мире Java (загляните в исходники почти любого проекта от apache.org - найдете там pom.xml)
- огромный актуальный репозиторий артефактов в репозиториях maven
- поддерживается большинством современных IDE (Eclipse, IntelliJ IDEA, NetBeans и т.д.)

Ключевые преимущества

- **декларативный язык описания проекта** (POM)
- ● автоматическое управление зависимостями
- ● огромный, поддерживаемый в актуальном состоянии репозиторий артефактов
- ● модульная расширяемая за счет плагинов архитектура, огромное количество плагинов

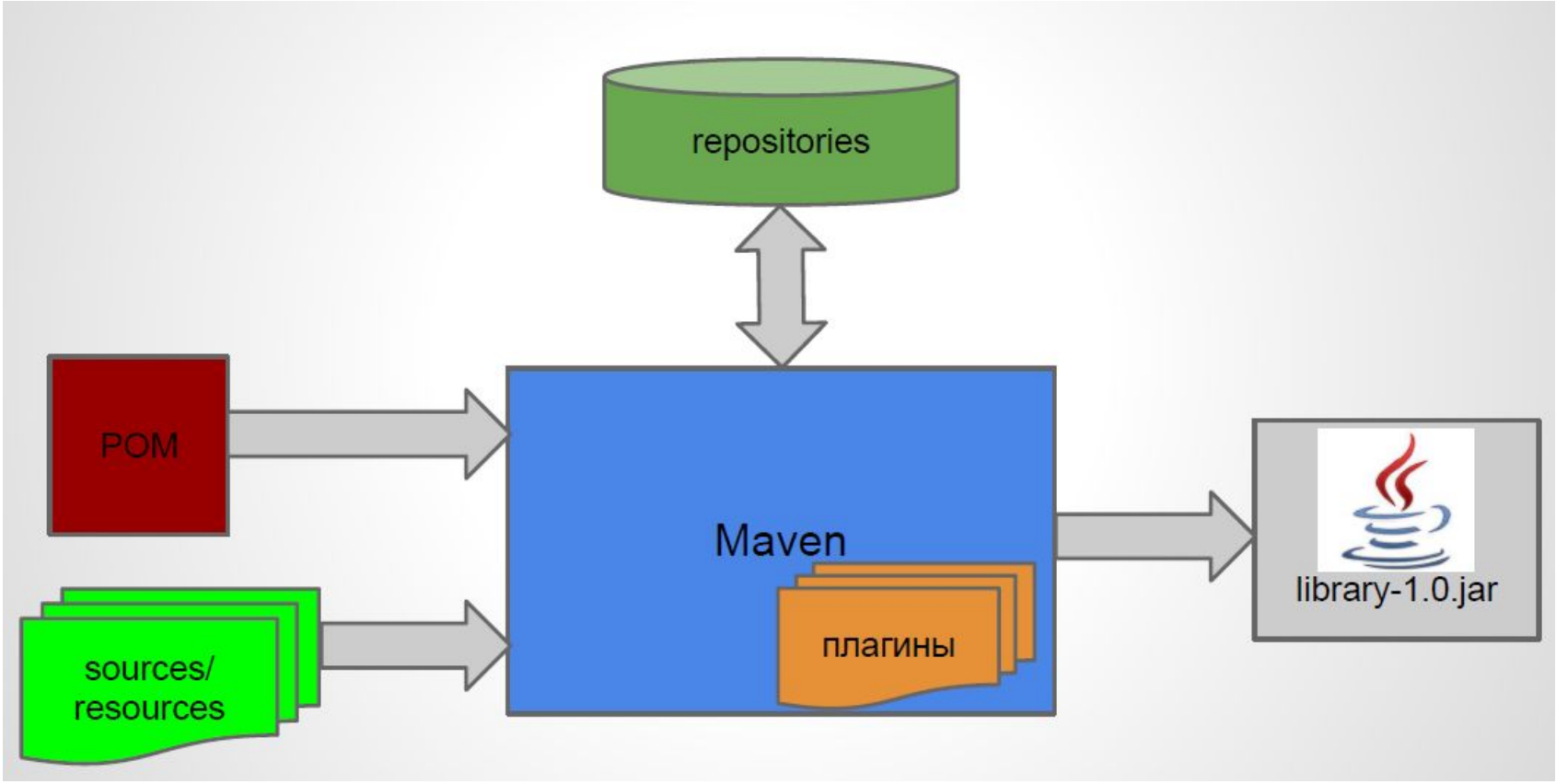
Главные недостатки

- сложность освоения
- неочевидность (контринтуитивность) в некоторых моментах
- не очень хорошая документация
- огромное количество плагинов (трудно сориентироваться)
- трудно разобраться если что то пошло не так (возникла ошибка)
- необходим доступ в Интернет или собственный репозиторий артефактов

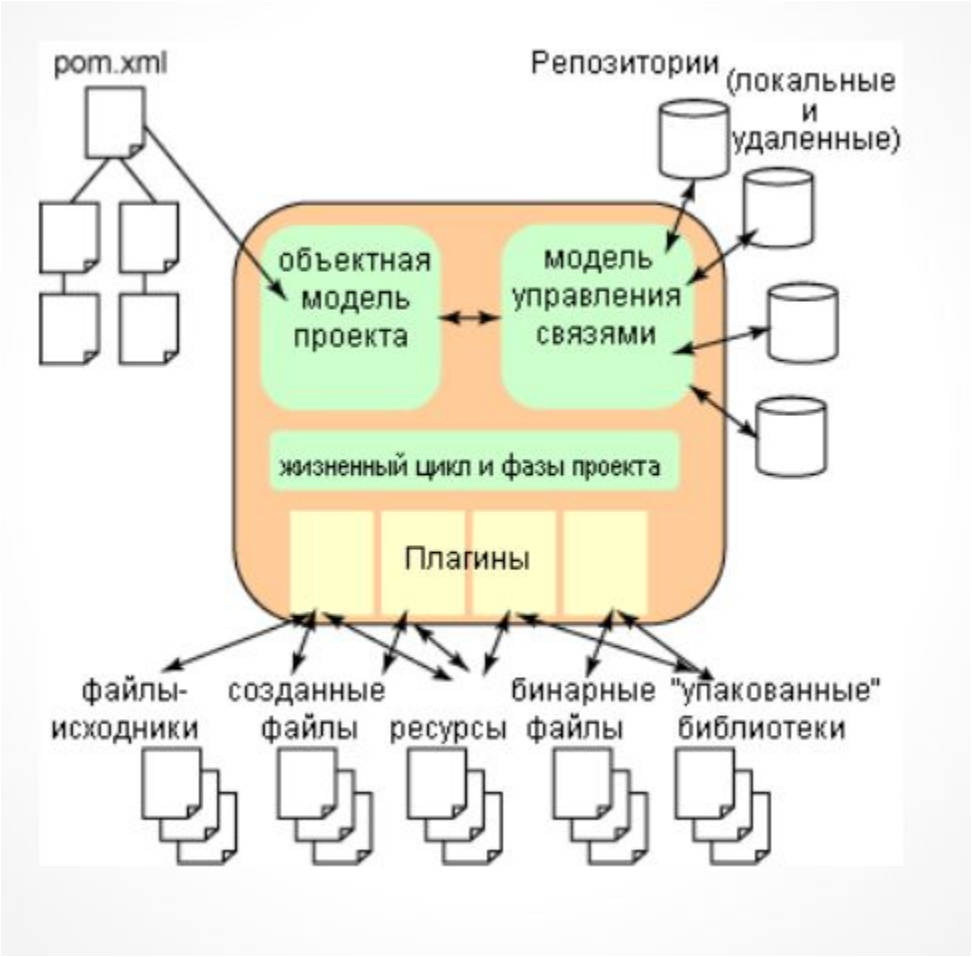
Установка Maven

- ● требует наличия на машине JDK версии >=1.5
- ● скачиваем с сайта проекта http://apache.maven.org
- ● разворачиваем архив
- ● прописываем переменную окружения M2_HOME
- ● прописываем путь $M2_HOME/bin в PATH
- ● запуск командой mvn

Как все это работает?



Еще раз как все ЭТО работает?



Артефакт

- Что есть Артефакт? Да все что угодно что производит наш проект (jar,war, ear и т.п.) или использует maven (плагин)
- Результатом работы Maven является создание (построение) артефакта, а так же ряд дополнительных действий над ним (тестирование, инсталляция в локальный репозиторий, deployment)
- ● Сам артефакт зависит от других артефактов (наших и внешних, плагинов maven)

Координаты артефакта

- groupId
- artifactId
- [packaging]

default jar

- ● version

в формате mmm.nnn.bbb-ssssss-dd , необязательными являются поля

ssssss (спецификатор SNAPSHOT,RELEASE и т.п.) и dd (номер сборки)

- ● [classifier]

Примеры maven координат

log4j

```
<groupId>log4j</groupId>
<artifactId>log4j</artifactId>
<version>1.2.16</version>
```

spring

```
<groupId>org.springframework</groupId>
<artifactId>spring-core</artifactId>
<version>3.1.0.RELEASE</version>
```

POM файл

- POM - Project Object Model, xml файл, обычно называется pom.xml
- POM файл содержит описание нашего проекта (декларативный стиль!) и все специфические его настройки.
- Пример минимального POM файла (данный пример работает!!!):

```
<project xmlns=...>
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.codehaus.mojo</groupId>
    <artifactId>my-project</artifactId>
    <version>1.0</version>
</project>
```

Минимальные требования к POM

- Минимально POM файл проекта должен содержать лишь версию модели и координаты артефакта проекта.

Репозитории

- Репозиторий maven это файловое хранилище с метаинформацией и быстрым поиском и доступом
- ● Бывают двух типов
- ○ local (находятся в ~/.m2/repository)
- ○ remote (например, стандартный http://repo1.maven.org/maven2 или внутренний репозиторий компании, например, Nexus)
- ● используются для хранения и получения зависимостей (dependencies) проекта и плагинов maven

Lifecycle (жизненный цикл) проекта

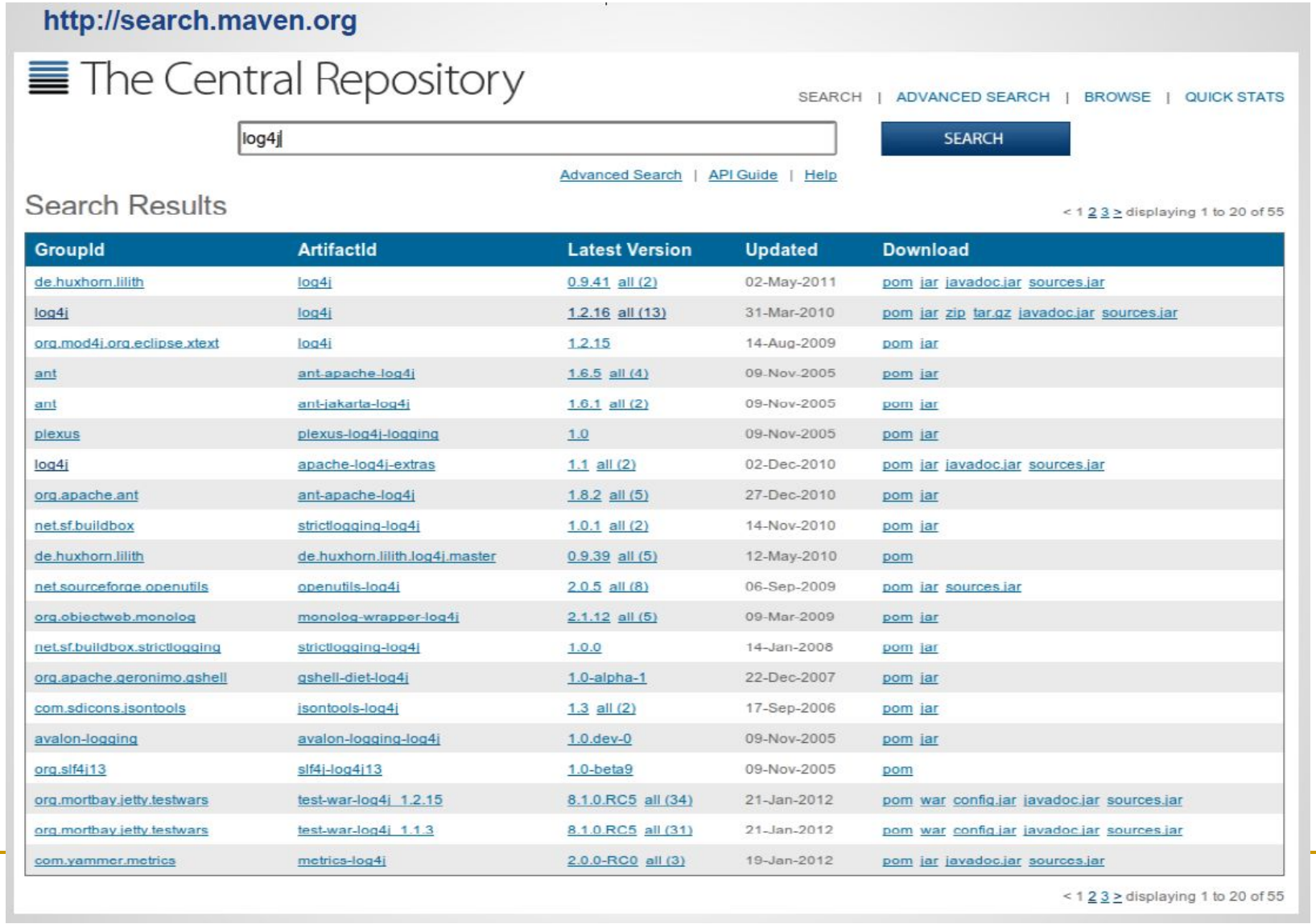
Существуют три стандартных lifecycles:

- ● clean - очистка проекта
- ● default - построение проекта из исходных кодов
- ● site - построение вторичных артефактов

(документация, wiki, сайт и т.п.)

- Жизненный цикл состоит из фаз. К каждой фазе может быть привязан ноль или более goal-ов различных плагинов. По умолчанию, набор фаз с привязанными плагинами стандартен и зависит от типа артефакта проекта (конкретно - от типа packaging).

Как найти нужную библиотеку



http://search.maven.org

The Central Repository

SEARCH | ADVANCED SEARCH | BROWSE | QUICK STATS

log4j

SEARCH

Advanced Search | API Guide | Help

Search Results

< 1 2 3 > displaying 1 to 20 of 55

GroupId	ArtifactId	Latest Version	Updated	Download
de.huxhorn.lilith	log4j	0.9.41 all (2)	02-May-2011	pom jar javadoc.jar sources.jar
log4j	log4j	1.2.16 all (13)	31-Mar-2010	pom jar zip tar.gz javadoc.jar sources.jar
org.mod4j.org.eclipse.xtext	log4j	1.2.15	14-Aug-2009	pom jar
ant	ant-apache-log4j	1.6.5 all (4)	09-Nov-2005	pom jar
ant	ant-jakarta-log4j	1.6.1 all (2)	09-Nov-2005	pom jar
plexus	plexus-log4j-logging	1.0	09-Nov-2005	pom jar
log4j	apache-log4j-extras	1.1 all (2)	02-Dec-2010	pom jar javadoc.jar sources.jar
org.apache.ant	ant-apache-log4j	1.8.2 all (5)	27-Dec-2010	pom jar
net.sf.buildbox	strictlogging-log4j	1.0.1 all (2)	14-Nov-2010	pom jar
de.huxhorn.lilith	de.huxhorn.lilith.log4j.master	0.9.39 all (5)	12-May-2010	pom
net.sourceforge.openutils	openutils-log4j	2.0.5 all (8)	06-Sep-2009	pom jar sources.jar
org.objectweb.monolog	monolog-wrapper-log4j	2.1.12 all (5)	09-Mar-2009	pom jar
net.sf.buildbox.strictlogging	strictlogging-log4j	1.0.0	14-Jan-2008	pom jar
org.apache.geronimo.gshell	gshell-diet-log4j	1.0-alpha-1	22-Dec-2007	pom jar
com.sdicons.jsontools	jsontools-log4j	1.3 all (2)	17-Sep-2006	pom jar
avalon-logging	avalon-logging-log4j	1.0.dev-0	09-Nov-2005	pom jar
org.slf4j13	slf4j-log4j13	1.0-beta9	09-Nov-2005	pom
org.mortbay.jetty.testwars	test-war-log4j_1.2.15	8.1.0.RC5 all (34)	21-Jan-2012	pom war config.jar javadoc.jar sources.jar
org.mortbay.jetty.testwars	test-war-log4j_1.1.3	8.1.0.RC5 all (31)	21-Jan-2012	pom war config.jar javadoc.jar sources.jar
com.yammer.metrics	metrics-log4j	2.0.0-RC0 all (3)	19-Jan-2012	pom jar javadoc.jar sources.jar

< 1 2 3 > displaying 1 to 20 of 55