

Установка maven

JSP & Servlets
1 уровень, 0 лекция

ОТКРЫТА

1.1 Большие программы

Мы уже научились писать маленькие программы, поэтому теперь будем учиться писать большие. Как известно, чем больше и сложнее программа, тем больше за ее разработку платят денег:) И начнем с небольшой предыстории...

С ростом размера программ разработчики столкнулись с двумя новыми для себя обстоятельствами:

- Над одной программой работает большое количество людей.
- Нет такого человека, который бы знал весь код программы.

Очень часто начали возникать ситуации, когда программист фиксил баг в одном месте программы и при этом ломал что-то в другом. В release documentation даже появилась такая шутка:

Список изменений:

- Исправили старые баги :)
- Добавили новые :(

Тогда же придумали два подхода к решению этой проблемы: технический и менеджерский.

Технический подход заключался в том, что программы разбивали на части: **библиотеки и модули**. Каждый такой модуль был небольшим кирпичиком из которых потом выстраивались большие проекты. Библиотеки же – это такие универсальные компоненты, которые могут использоваться в разных программах.

Менеджерский подход был еще интереснее – они ограничивал количество людей, которые могут работать над одним проектом/библиотекой. Эмпирически даже вывели правило: **команда должна быть настолько большой, чтобы "ее можно было накормить двумя пиццами"**. Обычно это означает, что если над проектом работает более **8 человек**, то его нужно разделить на два проекта.

В сообществе Java-разработчиков популярным стало написание библиотек на все случаи жизни и выкладывание их в общий доступ. Таким образом, Java-программисты могли не писать снова один и тот же код (который зачастую был сыроват и содержал баги), а пользоваться готовыми и **проверенными решениями**.

Дополнительным стимулом стало то, что язык Java получил большую популярность при написании серверных решений (работал на бэкенде). Во-первых, у серверного ПО более высокие требования к надежности, **и использование проверенных временем библиотек всегда предпочтительнее, чем написание своего кода**.

Во-вторых, у серверов практически нет ограничений на размер кода. Разработчик мобильного приложения старается впихнуть его в 10 мегабайт, десктопного приложения – в 100 мегабайт. А бэкенд-разработчик на Java может напихать в проект несколько десятков гигабайт библиотек и ему слова никто не скажет :)

Это, кстати, не шутка. Легко можно встретить бэкенд-проект из нескольких десятков модулей и с парой сотен библиотек. Вот только описывать (и изменять!) сценарии сборки таких проектов стало чрезвычайно трудно.

И тут появился Maven.

1.2 Знакомство с Maven

Maven – это специальный “фреймворк” для управления сборкой проектов. Он стандартизирует 3 вещи:

- Описание проекта;

- Сценарии сборки проектов,
- Зависимости между библиотеками.

Предшественником Maven’а был **Ant**, а наследником является **Gradle**. Но именно Maven развил и довел до совершенства три перечисленных стандарта, а также регламентировал их взаимодействие. Именно он вывел работу Java-сообществ на новый уровень. Давай же посмотрим на него подробнее.



Технически Maven – это специальная программа/сервис, основная цель которой **управлять процессом сборки** проектов. Ее можно просто скачать в виде архива и распаковать в любую директорию. Специальный установщик для этого не нужен.

Графического интерфейса у нее нет – **все команды отдаются ей с помощью консоли**. Чтобы еще более комфортно с ней работать, рекомендуется прописать в своей ОС специальные переменные окружения (environment variables).

Так же у Maven есть специальный **репозиторий** (директория/папка), где он хранит библиотеки, которые используются им при сборке проектов. Тебе нужно будет выбрать какую-нибудь папку на диске и назначить ее в качестве репозитория.

Еще из интересного можно отметить наличие глобального Maven-репозитория для всех библиотек, но об этом расскажем немного позже.

1.3 Загрузка и установка Maven

У Maven есть официальный сайт maven.apache.org. Там очень много документации по проекту, так что, если возникнут сложности или дополнительные вопросы – заходи, не стесняйся.

Также на странице downloads (<https://maven.apache.org/download.cgi>) можно скачать архив с maven (apache-maven-3.8.5-bin.zip). Распакованный архив займет где-то 10 Мб, хотя для локального maven репозитория со временем потребуется несколько сотен мегабайт памяти.

Maven написан на Java и требует JRE не ниже 7 версии, а также прописанные переменные окружения типа JAVA_HOME.

Просто создай на компьютере папку для Maven, например, **d:\devtools**, и распакуй в нее архив с Maven. В итоге у тебя должна получиться папка типа **d:\devtools\maven\bin**, где будут находиться основные бинарные файлы проекта.

1.4 Переменные окружения

После этого нужно добавить путь к папке bin из распакованного архива в переменную среды PATH.

Чтобы установить переменную среды (environment variable) в Windows 10, нужно перейти в Панель управления — Система — Дополнительные параметры системы. Затем нажать “Переменные среды”, найти PATH и выбрать “Изменить”, после чего добавить путь **d:\devtools\maven\bin** в конец строки. Обрати внимание, путь должен вести именно к папке bin.

В ОС на основе Unix переменную среды можно добавить консольной командой:

```
1 export PATH=/opt/apache-maven-3.8.5/bin:$PATH
```

Если ты все сделал правильно, то в консоли нужно набрать команду: «mvn -v». В ответ ты увидишь что-то типа:

```
1 C:\Users\Zapp>mvn -v
2 Apache Maven 3.0.5 (r01de14724cdef164cd33c7c8c2fe155faf9602da; 2013-02-19 15:51:28+0200)
3 Maven home: T:\apache-maven-3.0.5\bin\..
4 Java version: 1.8.0_65, vendor: Oracle Corporation
5 Java home: C:\Program Files\Java\jdk1.8.0_65\jre
6 Default locale: en_US, platform encoding: Cp1251
7 OS name: "windows 7", version: "6.1", arch: "amd64", family: "dos"
```

1.5 Локальный репозиторий Maven

Также ты можешь задать специальную папку, где Maven будет хранить jar-библиотеки, которые будет использовать при сборке проектов. Такую папку называют – **локальный maven репозиторий**.

Если такая папка не задана, то Maven создаст ее в домашней директории текущего пользователя. У меня это директория:
C:\Users\Zapp\.m2

Папка имеет довольно специфическое имя “.m2”. Хотя пользователей Linux оно не пугает – там это довольно распространенный подход к именованию различных ”репозиториев” и/или любого другого хранилища служебной информации.

Важно! Не располагай Maven в системных папках, так как при работе ему понадобятся права на запись в эти папки, что может вызывать нездоровый интерес антивируса или операционной системы.

Maven до версии 3.5 требовал указать переменную окружения с именем M2_HOME, но теперь это не нужно.

Более подробно о конфигурировании Maven можно почитать по ссылке: <https://maven.apache.org/configure.html>

< Список квестов

Следующая лекция >

– +48 +

Комментарии (31) популярны новые старые

JavaCoder

Введите текст комментария

G QA Automation Engineer

25 сентября, 20:56

...

Ubuntu (*Unix тоже сойдёт):
sudo apt install maven
mvn -v

Ответить

–

0

+

Anonymous #3163965 Уровень 2, Russian Federation 13 сентября, 14:03 ...

объяснять мне пожалуйста. Почему только командный строка? проще работать через иде. програмист глупый? не любить простота?

Ответить

0

Andrei Lead Software Developer (

8 октября, 10:49

...

прогрмисты деляться на инженеров и кодеров - первые понимают что делают и им без разницы где это делать, вторые умеют только говнокодить без вникания что-почему-как и без любимой IDE даже не знают как запустить проект без нее

Ответить

0

Кирилл Руковицын Уровень 46, Москва

14 октября, 14:32

...

ух сколько злости)

Ответить

0

Andrei Lead Software Developer (

15 октября, 12:20

...

Это не злость - это опыт ;).
Очень печально когда в проект приходит "лид" или "синиор" разарботчик, а логи читать не умеет, не понимает что такое стек-трейс, не знает что такое брейкпоинт и как им пользоваться даже в IDE.

Ответить

+1

Artem Уровень 16, Новосибирск, Russian Federation

11 сентября, 21:49

...

Подскажите, я сейчас на 10 уровне, стоит лезть сюда и пытаться разобраться во всем?

Ответить

0

Oleg Software Developer

10 октября, 23:57

...

Не стоит. Если хватит сил проходи до 30+ уровня, а лучше до 40. А потом сюда заходи в продолжение обучения. Данный ветка в карте квестов интуитивно понятно расположен после прохождения основного древа синтаксис-кор-коллекции-многопоточка-JPA-стажировка.... Вот идеальный путь по курсу

Ответить

0

Artem Уровень 16, Новосибирск, Russian Federation

12 октября, 13:12

...

Окей, понял... Спасибо

Ответить

0

Каролина Уровень 68, Киев, Беларусь

1 сентября, 19:45

...

Подробная инструкция для **windows** со скринами:
<https://phoenixnap.com/kb/install-maven-windows>

А вот как можно установить maven не заводя переменных окружения для **mac OS**:
1) Для начала устанавливаем менеджер пакетов - вводим в консоль команду:

```
1 /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD
```

2) устанавливаем maven - вводим в консоль команду:

```
1 brew install maven
```

3) проверяем что установилось вводим в консоль:

```
1 mvn -version
```

или короткая форма:

```
1 mvn -v
```

Ответить

+2

Andrei Lead Software Developer (

8 октября, 10:50

...

зачем такие сложности?
качаем мавен в виде zip арихва с оф сайта -> распаковываем куда хотим -> добавляем в системные переменные путь к мавену.

Ответить

0

GSTV11 Уровень 35, Москва

28 августа, 23:12

...

у кого не получилось - терминал надо перезапускать

Ответить

0

GSTV11 Уровень 35, Москва

28 августа, 23:00

...

добавьте скрины, пожалуйста

Ответить

0

Антон

Уровень 68

23 августа, 23:19

...

При вводе "mvn -v" система выдавала ошибку ""The JAVA_HOME Environment variable is not defines correctly". Оказалось, ищет и не находит JDK (<https://www.oracle.com/java/technologies/downloads/>). После установки JDK в переменной окружения JAVA_HOME нужно указать путь на папку с JDK. Затем все заработало, MAVEN_HOME и M2_HOME не затребовало. Еще может выдавать эту ошибку, если JDK установлен, но путь к нему в JAVA_HOME указа неверно.

Ответить

0

+

LindX

Java Developer в Home

20 августа, 22:25

...

M2_HOME нужен. даже на 3.8.6

Ответить

0

+

Владимир

Уровень 68, Тель-Авив, Израиль

18 августа, 16:27

...

двумя пиццами можно только гномов накормить

Ответить

+4

+

Влад

Уровень 68, Минск

24 июля, 18:34

...

Maven версии 3.8.6 не заработал пока не указал переменные JAVA_HOME и MAVEN_HOME(Так только у меня?

Ответить

0

+

Danicimo

Java Junior Developer

28 июля, 07:36

...

Так у всех и с любой версией maven) Переходи на линкус))

Ответить

0

+

Антон

Уровень 68, Иваново, Россия

14 августа, 16:27

...

У меня и с ними не заработал((

Ответить

0

+

Andrei

Lead Software Developer (

15 октября, 12:17

...

JAVA_HOME и MAVEN_HOME переменные подсказывают ОС где искать javac и mvn в системе. Без них тоже можно запускать - но тогда надо полностью пути прописывать в командной строке.

Ответить

0

+

observer

Java Developer в monobank

14 июня, 16:57

...

Скажем так, это больше похож на "ознакомительный блок". Тут более глубже рассматривается этапы разработки, сборки и т.п.

Ответить

0

+

Facepalm

Уровень 32, Москвачкала, Россия

9 июня, 23:59

...

Не понял половины слов с 2-х уровней про maven, лучше глянуть видео в ютубе и проскочить на 3 уровень, а то почувствуете себя дурачком

Ответить

+1

+

Alex

QA Manual Engineer

11 июля, 18:21

...

Поддерживаю, лучше один раз увидеть, чем что раз услышать. [Видео с ютуб](#)

Ответить

+4

+

Facepalm

Уровень 32, Москвачкала, Россия

8 июня, 13:19

...

Если у вас при проверке командой "mvn -v" выходит - "The JAVA_HOME Environment variable is not defines correctly" значит у вас не установлен JDK (качаете отсюда последнюю версию <https://freessoft.ru/windows/java-se-development-kit-jdk/versions>) И всё при следующей проверке "mvn -v" выведет то что нужно.

Ответить

0

+

Николай Зернов

Уровень 59, Санкт-Петербург, Россия

8 июня, 13:04

...

Устанавливал версию 3.8.5 на macOS, и без указания переменной окружения M2_HOME установка не получалась.

Ответить

0

+

Sasha

Java Developer в UKEESS

3 июня, 10:53

...

Спасибо, наконец-то дождался) Я уже правда работаю, но все равно интересно посидеть здесь поностальгировать)

Ответить

+10

+

Айбелив Айкенфлаев

Уровень 36, Москва, Россия

19 июля, 21:51

...

В свете информации о том, что вы уже работаете, наличие ностальгии воодушевляет, т.к. говорит о том, что данный ресурс оставил положительные впечатления, а, следовательно, и я тут не зря время трачу)

Ответить

👍 +8 👎

Байсал Уровень 2, Алматы, Kazakhstan

31 мая, 11:56 ⋮

Ответить

👍 0 👎

Модератор Безработный

29 мая, 02:02 ⋮

и 10 лет не прошло)

Ответить

👍 +4 👎

Роман Backend Developer в ЭЛАР

28 мая, 04:08 ⋮

Third)))

Ответить

👍 +1 👎

Тёма Панфилов Уровень 18, Подольск

28 мая, 02:03 ⋮

Second

Ответить

👍 +1 👎

Mikhail Khanov Уровень 19, Russian Federation

27 мая, 02:08 ⋮

First

Ответить

👍 0 👎

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

 Русский 

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ



