

Java Professional module #7

lecture #1. Apache Maven.Сборка проектов

Системы сборки

Сборка (англ. assembly) - двоичный файл, содержащий исполняемый код программы или другой, подготовленный для использования информационный продукт.

В Spring Boot выбор системы сборки является важной задачей. Исторически у разработчиков приложений Java имелось несколько вариантов утилит сборки проектов. Некоторые со временем стали непопулярными, и не без причин, и теперь сообщество разработчиков сконцентрировалось на двух: Maven и Gradle. Spring Boot поддерживает обе с одинаковым успехом.

Maven

Maven используется для автоматизации сборки проектов с использованием Java. Он помогает составить схему сборки конкретного программного обеспечения, а также его различных зависимостей. Он использует XML-файл для описания проекта, который вы собираете, зависимостей программного обеспечения от сторонних модулей и частей, порядка сборки, а также необходимых плагинов. Существуют заранее определенные цели для таких задач, как упаковка и компиляция. Для абсолютного большинства проектов определяемая соглашением структура Maven прекрасно подходит, так что менять ее обычно смысла не имеет.

Maven

1. Здесь есть четкая структура каталогов, и вы обязательно должны ей следовать. (При использовании плагинов IDE, она создается автоматически).
2. Каждый билд имеет атрибуты `groupId`, `artifactId` и `version`. Первые два уникальны и используются для определения в репозитории. Обычно `groupId` - доменное имя организации, а `artifactId` - название текущего проекта. Поэтому эта пара и является определяющей: нет двух компаний с одинаковым доменом, как и не существует двух одинаковых проектов в одной компании.

Maven

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
  </dependency>
</dependencies>
</project>
```

Управление зависимости проекта
Maven



Spring Boot с типичной структурой
проекта Maven

Gradle

Maven использует XML для конфигурирования сборки. Gradle - предметно-ориентированный язык на основе Groovy. И это его основное отличие от Maven. В остальном Gradle руководствуется теми же принципами. Здесь за выполнение всей работы также ответственны плагины и нет широкой свободы действий.

```
apply plugin: 'java'

repositories {
    mavenCentral()
}

jar {
    baseName = 'gradleExample'
    version = '0.0.1-SNAPSHOT'
}

dependencies {
    compile 'junit:junit:4.12'
}
```

Пример файла build.gradle проекта с HelloWorld

Выбор между Maven и Gradle

Более жесткий декларативный, можно даже сказать, догматичный подход Maven значительно повышает единообразие процесса сборки для различных проектов и сред. Если следовать пути Maven, проблемы практически исключены и можно сосредоточиться на коде, не возясь со сборкой.

Для простых проектов, а также проектов с очень сложными требованиями к сборке гибкость Gradle может оказаться лучшим вариантом. Но дополнительная гибкость Gradle, особенно в сложных проектах, означает и большее количество времени, потраченного на тонкую настройку, диагностику и устранение неполадок, когда все работает не так, как ожидалось.

В наших проектах мы будем делать акцент на Maven.