СБОРКА МУЛЬТИ-МОДУЛЬНЫХ ПРОЕКТОВ С ПОМОЩЬЮ MAVEN.

Иванов Н.Ю[1]

Иванов Никита Юрьевич[1] – студент, *Поволжского государсттвенного университета телекоммуникаций и информатики, факультет информационных систем и технологий, кафедра «Программного обеспечения и управление в технических системах», г. Самара.*

Аннотация:
В статье анализируется использование сборки мульти-модульных проектов с помощью Maven. Приводятся плюсы использования инструмента сборки Maven в мульти-модульных проектах. Также демонстрируется пример использования мульти-модульности Maven.
Ключевые слова: *Maven, pom.xml, сборка, автоматизация сборки, мульти-модульность.*

Build multi-module projects with Maven.

Ivanov N.Y.[1]

Ivanov Nikita Yurievich – student, *Volga State University of Telecommunications and Informatics, Facility of Information Systems and Technologies, Department of Software and Management in Technical Systems, Samara.*

Annotation:
The article analyzes the use of the build of multi-modular projects using Maven. The advantages of using the Maven build tool in multi-module projects. Shows example of the use of Maven in multi-module project.
Keywords: *Maven, pom.xml, build, build automatization, multi-modularity*

УДК 00.1028

Мультимодульный проект Maven построен с помощью POM аггрегатора, который управляет группами подмодулей. В большинстве случаем агрегатор находится в корневом каталоге проекта. Подмодули являются обычными проектами Maven.

Существенным преимуществом такого подхода является уменьшение дублирования конфигурации. Например, есть приложение, которое состоит из

нескольких модулей, назовем их внешним модулем и внутренним. В процессе разработки функциональность обоих модулей меняется и в таком случае без специального инструмента сборки придется собирать оба модуля по отдельности или писать скрипт, который будет компилировать код, запускать тесты и показывать результаты. Через определенное время к проекту добавится еще больше модулей и с текущей моделью сборки им станет сложнее управлять и поддерживать.

Кроме того, в реальном мире проектам могут потребоваться определенные плагины Maven для выполнения различных операций в течении цикла сборки, совместного использования зависимостей и профилей.

Поэтому, используя мультимодульность Maven, модули проекта можно собирать в одну команду, и, емли порядок имеет значение, его можно задать. Кроме того, можно из родительского pom.xml можно поделиться огромным количеством конфигурации с дочерними файлами конфигурации Maven.

Maven поддерживает наследование таким образов, что каждый pom.xml имеет неявный родительский pom.xml, который называется Super POM и находится в бинарных файлах Maven. Следовательно, есть возможность создать собственный pom.xml, который будет родительским в рамках проекта. Затем туда включается вся конфигурация с зависимостями. Подмодули – это обычные проекты Maven, который наследуются от родительского pom.xml. Наследование позволяет делиться конфигурацией родительского pom.xml. Однако, если необходимо собрать весь проект за один раз, то необходимо явно объявить подмодули в родительском pom.xml.

Можно рассмотреть пример сборки на тестовом проекте. Ниже представлен родительский pom.xml проекта, в котором объявлены дочерние модули и тип упаковки.

```
<modelVersion>4.0.0</modelVersion>

<groupId>com.sobachken</groupId>
<artifactId>learningpro</artifactId>
<version>1.0.0-SNAPSHOT</version>
<modules>
    <module>learningpro-frontend</module>
    <module>learningpro-backend</module>
</modules>
<packaging>pom</packaging>
<name>learningpro</name>
```

Далее приведен пример определения родительского pom.xml в дочерних модулях.

```
<artifactId>learningpro-backend</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>learningpro-backend</name>
<packaging>jar</packaging>

<parent>
    <artifactId>learningpro</artifactId>
    <groupId>com.sobachken</groupId>
    <version>1.0.0-SNAPSHOT</version>
    <relativePath>../pom.xml</relativePath>
</parent>

<artifactId>learningpro-frontend</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>learningpro-frontend</name>

<parent>
  <artifactId>learningpro</artifactId>
  <groupId>com.sobachken</groupId>
  <version>1.0.0-SNAPSHOT</version>
  <relativePath>../pom.xml</relativePath>
</parent>
```

После выполнения команды clean install на родительском pom.xml можно увидеть следующее.

Список литературы / References

1. Документация Maven [Электронный ресурс] URL: https://maven.apache.org/guides/index.html