

Шпаргалка по MAVEN

Программирование

Maven: Шпаргалка по сборке, запуску и библиотекам

Сборка JAR-файла и запуск проекта

1. Перейти в папку с проектом

```
cd /путь/к/проекту
```

2. Очистить старые файлы и собрать JAR

```
mvn clean package
```

✓ После сборки JAR-файл появится в папке `target/`, например:

```
target/myproject-1.0-SNAPSHOT.jar
```

3. Запустить JAR-файл

```
java -jar target/myproject-1.0-SNAPSHOT.jar
```

✓ Запуск в фоне (чтобы процесс не остановился после выхода из терминала)

```
nohup java -jar target/myproject-1.0-SNAPSHOT.jar > output.log 2>&1 &
```

4. Остановить запущенный процесс

Найти **PID** процесса:

```
ps aux | grep myproject
```


Остановить процесс:

```
kill -9 <PID>
```

2 Как подключать библиотеки в Maven?

Все зависимости указываются в `pom.xml` в разделе `<dependencies>`.

1. Найти библиотеку

 Перейти на [Maven Repository](#), найти нужную библиотеку и скопировать `<dependency>`.

2. Добавить в `pom.xml`

Пример для **Gson**:

```
XML

<dependencies>
  <dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.10.1</version>
  </dependency>
</dependencies>
```

3. Скачать библиотеки и перекомпилировать проект

```
mvn clean compile
```

3 Основные команды Maven

Команда	Что делает
<code>mvn clean</code>	Удаляет старые файлы сборки (<code>target/</code>)
<code>mvn compile</code>	Компилирует код
<code>mvn package</code>	Собирает JAR-файл в <code>target/</code>
<code>mvn install</code>	Устанавливает JAR в локальный репозиторий (<code>~/.m2/</code>)
<code>mvn dependency:tree</code>	Показывает дерево зависимостей
<code>mvn exec:java -Dexec.mainClass="com.example.Main"</code>	Запускает проект без сборки JAR
<code>mvn versions:display-dependency-updates</code>	Показывает доступные обновления библиотек