

## 参考

<https://qiita.com/yasushi-jp/items/97fbd3487690e2a79933>

## maven を使った HelloWorld

### プロジェクト作成

```
mvn archetype:generate -DgroupId=com.example -DartifactId=sample
```

### groupId

ルートパッケージに使用される

### artifactId

プロジェクト名。ディレクトリ名などに使用される

### ビルド

- mvn compile
  - プログラムをコンパイル
- mvn package
  - jar ファイル作成
- mvn package -DskipTests=true
  - テストをスキップして jar ファイル作成
- mvn dependency:build-classpath
  - クラスパスを表示。コマンドラインで手動で実行したいときとか便利。
- mvn install dependency:copy-dependencies
  - 必要な jar ファイルを target の dependency にダウンロードする

### 実行

```
java -cp target xxxxx.jar com.example.App
```

### 実行 (exec-maven-plugin を使う)

pom.xml の plugins に以下を追記。

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>exec-maven-plugin</artifactId>
  <version>1.2.1</version>
  <configuration>
    <mainClass>com.example.App</mainClass>
  </configuration>
</plugin>
```

mainClass はメインクラス。

```
mvn exec:java
```

## 使い方いろいろ

### コンパイルの対象のファイルを減らす

標準のままだと全ての Java ファイルがコンパイルの対象となる。

変更のあったファイル (class ファイルより新しい Java ファイルのみ) をコンパイルするほうが効率が良い

pom.xml の plugins に以下を追加すると新しいファイルのみコンパイルの対象となる。

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.1</version>
  <configuration>
    <useIncrementalCompilation>false</useIncrementalCompilation>
  </configuration>
</plugin>
```

## コンパイルの対象のファイルを減らす ( コマンドライン )

pom.xml に記載せずに useIncrementalCompilation を指定することもできる

```
mvn compile -Dmaven.compiler.useIncrementalCompilation=false
```

## ソースパス

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>build-helper-maven-plugin</artifactId>
  <executions>
    <execution>
      <id>add-source</id>
      <phase>generate-sources</phase>
      <goals>
        <goal>add-source</goal>
      </goals>
      <configuration>
        <sources>
          <source>../demo2/src/main/java</source>
        </sources>
      </configuration>
    </execution>
  </executions>
</plugin>
```

## 一部ソース除外

ソースパスを追加した場合などで、不要なファイルがある場合は除外できる。  
ただし、指定方法は絶対パス等ではなく、

```
<exclude>**/some/full/directory/*</exclude>
<exclude>**/some/single/File.java</exclude>
```

などのような指定方法をする。

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <configuration>
    <excludes>
      <exclude>**/demo2/Demo2Application.java</exclude>
    </excludes>
  </configuration>
</plugin>
```

## ローカルリポジトリ (.m2) のパスを変更

環境変数で変更する方法

<https://stackoverflow.com/questions/16591080/maven-alternative-m2-directory>

```
export MAVEN_OPTS="-Dmaven.repo.local=/path/to/repository"
```