Creating a standalone maven project in Non-Interactive Mode

```
-DgroupId=ineuron
-Dartifactlid=SwapDemoCmd
-DarchetypeArtifactid=maven-archetype-quickstart
-DinteractiveMode=false
```

```
D:\mavenpgms\>mvn archetype:generate -DgroupId=ineuron -DartifactId=SwapDemoApp -
DarchetypeArtifactId=maven-archetype-quickstart
             -Dpackage=in.ineuron -Dversion=1.0 -DinteractiveMode=false
```

```
SwapDemoApp
   src/main/java
            |=> in.ineuron.SwapDemoApp(main method)
   src/main/test
            |=> in.ineuron.App(test case code)
```

```
package in.ineuron;
class SwapApp
{
      int a,b;
      public void accept(int x,int y){
            a=x;
            b=y;
      }
      public void swapValues(){
            a=a+b;
            b=a-b;
            a=a-b;
      }
      void disp(){
            System.out.println("Value of a is :: "+a);
            System.out.println("Value of b is :: "+b);
      }
}
public class SwapDemoApp {
    public static void main( String[] args ){
            if(args.length!=2)
                  System.out.println("Plz enter 2 numbers...");
            else{
                  int x1= Integer.parseInt(args[0]);
                  int x2= Integer.parseInt(args[1]);

                  SwapApp s = new SwapApp();
                  s.accept(x1,x2);
                  System.out.println("Before swapping");
                  System.out.println("---------------");
                  s.disp();
                  s.swapValues();
                  System.out.println("After swapping");
                  System.out.println("---------------");
                  s.disp();
            }

    }
}
```

pom.xml[The above program should run from maven and it should accept command line arguments,so we use a plugin called "exec-maven-plugin"]

```
=======
<build>
    <plugins>
      <plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <version>3.1.0</version>
        <executions>
          <execution>
                  <id>ex2</id>
                  <phase>package</phase>
            <goals>
              <goal>java</goal>
            </goals>
          </execution>
        </executions>
        <configuration>
          <mainClass>in.ineuron.SwapDemoApp</mainClass>
              <arguments>
                  <argument>30</argument>
                  <argument>40</argument>
              </arguments>
        </configuration>
      </plugin>
    </plugins>
</build>
```

```
D:\Mavenpgms\SwapDemoApp>mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< ineuron:SwapDemoApp >-------------------------
[INFO] Building SwapDemoApp 1.1
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.3.0:resources (default-resources) @ SwapDemoApp ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,
i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory D:\Mavenpgms\SwapDemoApp\src\main\
resources
[INFO]
[INFO] --- compiler:3.10.1:compile (default-compile) @ SwapDemoApp ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e.
build is platform dependent!
[INFO] Compiling 1 source file to D:\Mavenpgms\SwapDemoApp\target\classes
[INFO]
[INFO] --- resources:3.3.0:testResources (default-testResources) @ SwapDemoApp ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,
i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory D:\Mavenpgms\SwapDemoApp\src\test\
resources
[INFO]
[INFO] --- compiler:3.10.1:testCompile (default-testCompile) @ SwapDemoApp ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e.
build is platform dependent!
[INFO] Compiling 1 source file to D:\Mavenpgms\SwapDemoApp\target\test-classes
[INFO]
```

```
[INFO] --- surefire:3.0.0:test (default-test) @ SwapDemoApp ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit.JUnit3Provider
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running in.ineuron.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.01 s - in
in.ineuron.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.3.0:jar (default-jar) @ SwapDemoApp ---
[INFO] Building jar: D:\Mavenpgms\SwapDemoApp\target\SwapDemoApp-1.1.jar
[INFO]
[INFO] --- exec:3.1.0:java (Swap2NumbersApp) @ SwapDemoApp ---
Before swapping
---------------
Value of a is :: 30
Value of b is :: 40

After swapping
---------------
Value of a is :: 40
Value of b is :: 30
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  4.270 s
[INFO] Finished at: 2023-04-05T20:07:36+05:30
[INFO] ------------------------------------------------------------------------
```

Working in eclipse to create a maven standalone project

```
Maven Project
=> Dont select checkbox create a simple project
=> Choose archetype as
            maven-archetype-quickstart (select version-1.4 from org.apache.maven)
=> provide
      groupid : pwskills
      artifactid: MavenProject1
      version : 1.0

MavenProject1
   |=> src/main/java
                  |=> in.pwskills.JdbcSelectApp(main code)
   |=> src/main/test
                  |=> in.pwskills.AppTest.java
   |=> pom.xml(mysql-connectector-java.jar)
```

JdbcSelectApp.java

```
package in.pwskills;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class JdbcSelectApp {

      public static void main(String[] args) throws Exception {

            String url = "jdbc:mysql://localhost:3306/octbatch";
            String username = "root";
            String password = "root123";

            Connection connection = DriverManager.getConnection(url, username,
password);
            System.out.println("Connection object is created:: " + connection);

            Statement statement = connection.createStatement();
            System.out.println("Statement object is created:: " + statement);

            String sqlSelectQuery = "SELECT SID,SNAME,SAGE,SADDRESS FROM STUDENT";
            ResultSet resultSet = statement.executeQuery(sqlSelectQuery);
            System.out.println("ResultSet object is created:: " + resultSet);

            System.out.println("SID\tSNAME\tSAGE\tSADDR");
            while (resultSet.next()) {
                  Integer id = resultSet.getInt(1);
                  String name = resultSet.getString(2);
                  Integer age = resultSet.getInt(3);
                  String team = resultSet.getString(4);
                  System.out.println(id + "\t" + name + "\t" + age + "\t" + team);
            }

            // Close the Connection
            connection.close();
            System.out.println("Closing the connection...");
      }
}
```

pom.xml
=======

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>pwskills</groupId>
  <artifactId>MavenProj1</artifactId>
  <version>1.2</version>

  <name>MavenProj1</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/com.mysql/mysql-connector-j -->
      <dependency>
            <groupId>com.mysql</groupId>
            <artifactId>mysql-connector-j</artifactId>
            <version>8.0.31</version>
      </dependency>

  </dependencies>

  <build>
    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults
(may be moved to parent pom) -->
      <plugins>
            <!-- Executing maven as java program-->
        <plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <version>3.1.0</version>
        <executions>
          <execution>
                  <id>JDBCSELECT APP</id>
                  <phase>package</phase>
            <goals>
              <goal>java</goal>
            </goals>
          </execution>
        </executions>
        <configuration>
          <mainClass>in.pwskills.JdbcSelectApp</mainClass>
        </configuration>
      </plugin>
      </plugins>
    </pluginManagement>
  </build>
</project>
```

Execution
=========

right click on project-> run=> maven build => type in goals as exec:java and click on run

output
======

```
Connection object is created:: com.mysql.cj.jdbc.ConnectionImpl@707f21d6
Statement object is created:: com.mysql.cj.jdbc.StatementImpl@518376cd
ResultSet object is created:: com.mysql.cj.jdbc.result.ResultSetImpl@20117d95
SID   SNAME SAGE  SADDR
7     dhoni 41    CSK
```

```
10    sachin      49    MI
18    kohli 36    RCB
33    pandya      31    GT
99    rahul 31    LSG
Closing the connection...
```

Maven features
==============

1. properties

In pom.xml file we use properties feature to link to the version of Spring module version

```
<properties>
      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
      <maven.compiler.source>1.8</maven.compiler.source>
      <maven.compiler.target>1.8</maven.compiler.target>
      <spring.version>5.3.26</spring.version>
</properties>

<dependencies>
<dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>${spring.version}</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
<dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-jdbc</artifactId>
      <version>${spring.version}</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-orm -->
<dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-orm</artifactId>
      <version>${spring.version}</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->
<dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>${spring.version}</version>
</dependency>
</dependencies>
```

2. exclusion

In pom.xml file we can use exclusion to exclude particular dependant jar without excluding a main jar

```
<dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>5.6.15.Final</version>
      <exclusions>
            <exclusion>
                  <groupId>org.jboss.logging</groupId>
                  <artifactId>jboss-logging</artifactId>
            </exclusion>
      </exclusions>
</dependency>

Working with repositories
=========================
1. local repository
2. central repository
3. remote repository

      refer: MavenProj1

pom.xml(mysqlconnector from central repository)
===============================================
<!-- https://mvnrepository.com/artifact/com.mysql/mysql-connector-j -->
<dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <version>8.0.32</version>
</dependency>


pom.xml(oracleconnector from central repository)
================================================
<!-- https://mvnrepository.com/artifact/com.oracle.database.jdbc/ojdbc8 -->
<dependency>
    <groupId>com.oracle.database.jdbc</groupId>
    <artifactId>ojdbc8</artifactId>
    <version>19.12.0.0</version>
</dependency>


pom.xml(oracleconnector from remote repository)
================================================
<!-- https://mvnrepository.com/artifact/com.oracle.jdbc/ojdbc8 -->
<dependency>
      <groupId>com.oracle.jdbc</groupId>
      <artifactId>ojdbc8</artifactId>
      <version>12.2.0.1</version>
</dependency>
<repositories>
      <repository>
            <id>sample</id>
            <url>https://broadinstitute.jfrog.io/artifactory/libs-release-local/</
url>
      </repository>
</repositories>


Creating a local repository in .m2 folder for oracle jar
=======================================================
D:\Mavenpgms\SwapDemoApp>mvn install:install-file -Dfile=C:\oraclexe\app\oracle\
product\11.2.0\server\jdbc\lib\ojdbc6.jar -DgroupId=nitinineuron -
DartifactId=nitinineuronoracleljar -Dversion=11.2 -Dpackaging=jar
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< ineuron:SwapDemoApp >------------------------
[INFO] Building SwapDemoApp 1.1
[INFO]   from pom.xml
```

```
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- install:3.1.0:install-file (default-cli) @ SwapDemoApp ---
[INFO] pom.xml not found in ojdbc6.jar
[INFO] Installing C:\oraclexe\app\oracle\product\11.2.0\server\jdbc\lib\ojdbc6.jar
to C:\Users\nitin\.m2\repository\nitinineuron\nitinineuronoracleljar\11.2\
nitinineuronoracleljar-11.2.jar
[INFO] Installing C:\Users\nitin\AppData\Local\Temp\
mvninstall5098441414756102901.pom to C:\Users\nitin\.m2\repository\nitinineuron\
nitinineuronoracleljar\11.2\nitinineuronoracleljar-11.2.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.320 s
[INFO] Finished at: 2023-04-05T22:10:27+05:30
[INFO] ------------------------------------------------------------------------
```

In our project pom.xml file we need to use
==

```
<dependency>
            <groupId>nitinineuron</groupId>
            <artifactId>nitinineuronoracleljar</artifactId>
            <version>11.2</version>
</dependency>
```

Maven Inheritance
==================

In one project pom.xml, we can add <parent> tag having other project info like <groupld>, <artifactld> and <version> to get other project dependencies and plugins.

=> Create one project called HBProj-01 it will have groupId,artifactId,version

pom.xml
=======

```
<groupId>pwskills</groupId>
<artifactId>HBproj-01</artifactld>
<version>1.1</version>
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
    <hibernate-version>5.6.15.Final</hibernate-version>
    <mysql-version>8.0.32</mysql-version>
</properties>

 <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
  </dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
<dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>${hibernate-version}</version>
</dependency>

<!-- https://mvnrepository.com/artifact/com.mysql/mysql-connector-j -->
<dependency>
      <groupId>com.mysql</groupId>
      <artifactId>mysql-connector-j</artifactId>
      <version>${mysql-version}</version>
</dependency>

</dependencies>
```

=> Create one more project called HBProj-02(child project)

pom.xml
=======

```
<parent>
      <groupId>iNeuron</groupId>
      <artifactId>HBproj-02</artifactld>
      <version>1.2</version>
</parent>
```

            refer: Hibernate-01, Hiberante-02

To see the nature of inheritance check for Hibernate-02 dependancies