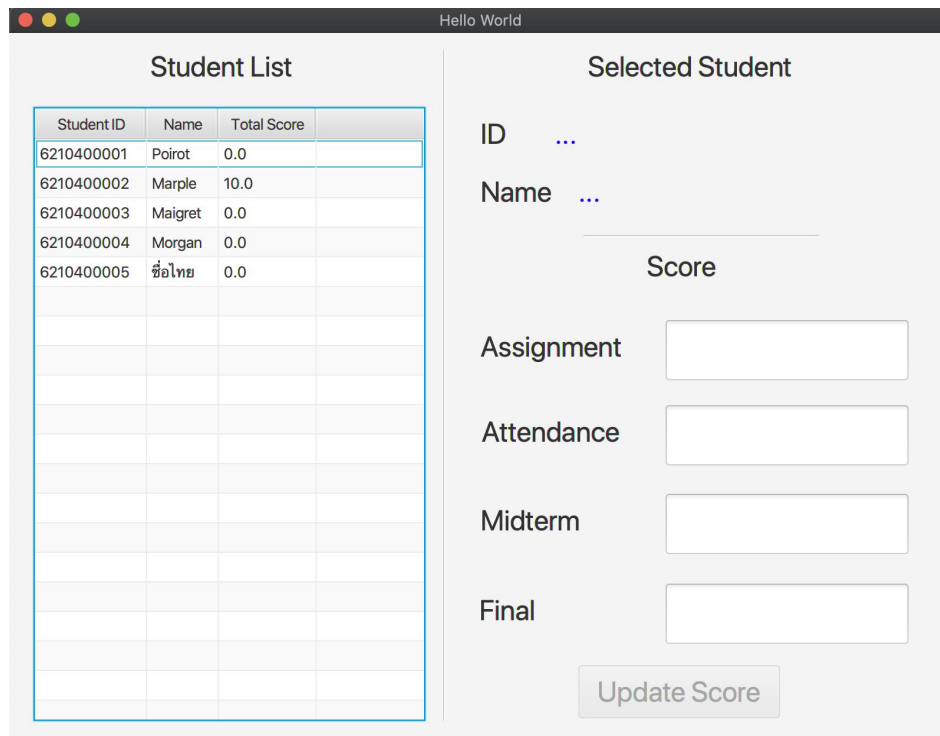


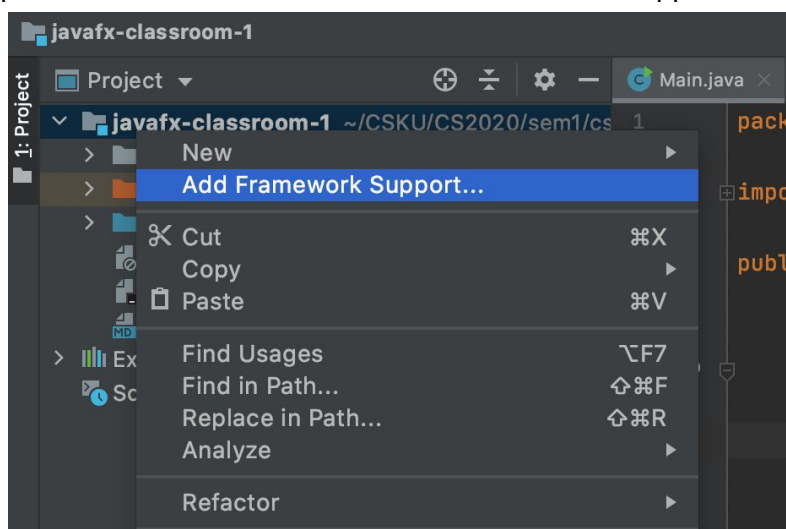
การเพิ่ม maven ไปใน JavaFX project บน IntelliJ IDEA

1. เริ่มต้นจากดาวน์โหลดตัวอย่าง JavaFX Project จาก <https://github.com/saacsos/javafx-classroom/releases/tag/v1>
2. เลือก Open or Import Project จากนั้นลองรัน Main Class ควรจะได้หน้าต่างโปรแกรมดังนี้

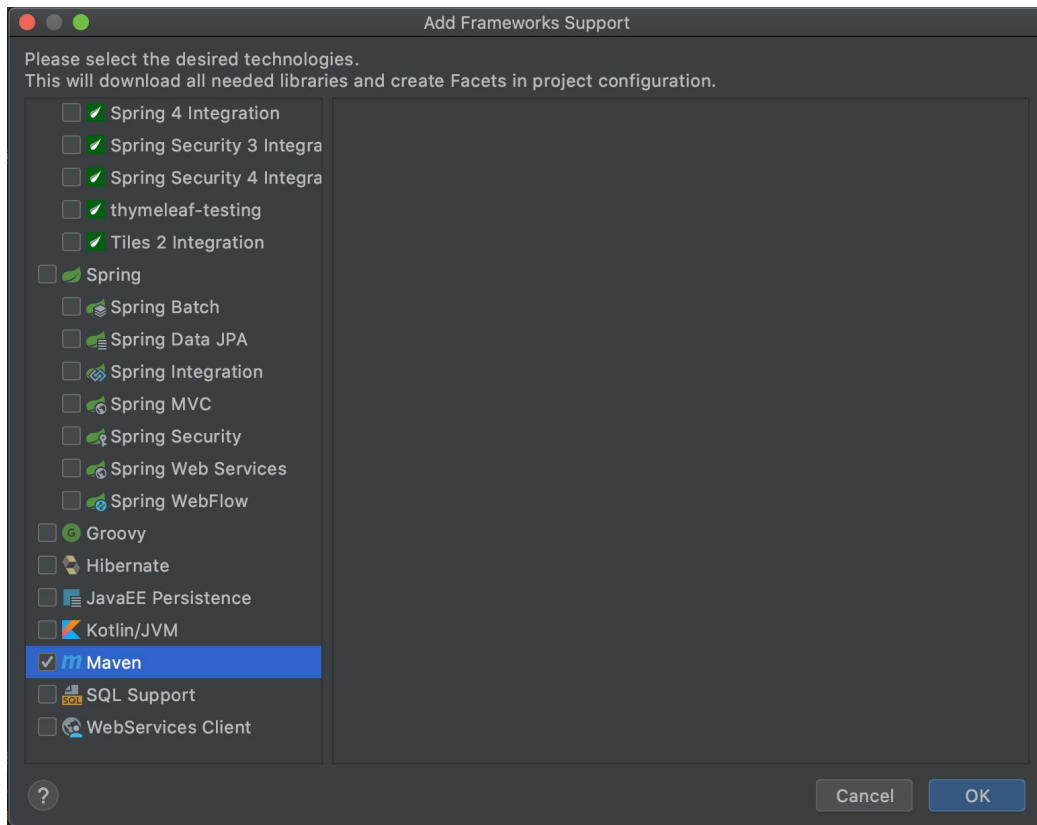


สังเกตการจัดการ package ใน folder src ว่ามีการจัดการอย่างไร ก่อนดำเนินการต่อ

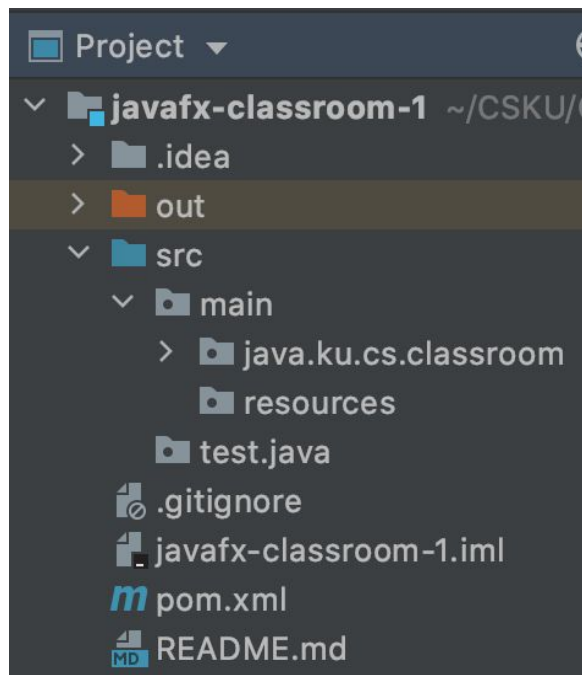
3. ที่ Project panel คลิกขวาที่ชื่อโปรเจกต์ แล้วเลือก Add Framework Support...



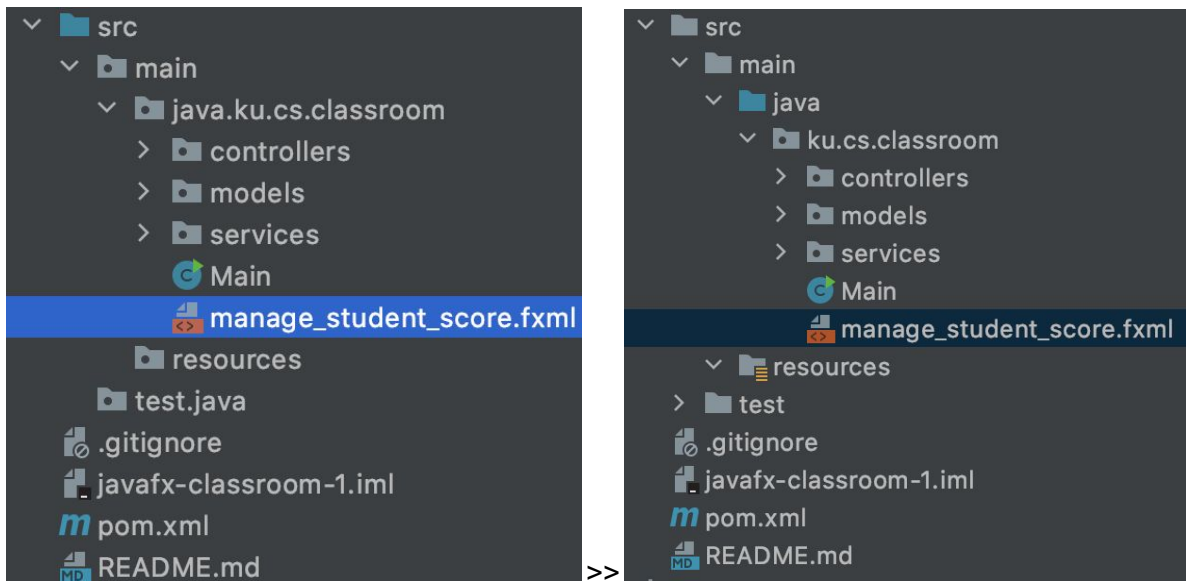
4. เลื่อนหา Maven เลือก แล้วกดปุ่ม OK



5. สังเกตว่า มีไฟล์ pom.xml และการจัดการ package ใน folder src เปลี่ยนแปลงไปจากเดิม



6. แก้ไขไฟล์ pom.xml โดยเปลี่ยน groupId เป็น ชื่อ package เดิม ก่อนเพิ่ม maven จากนั้นคลิกขวาที่ไฟล์ pom.xml เลือก Maven > Reload project



7. ค้นหา Main Class แล้วรันอีกครั้ง อาจพบปัญหา
java: diamond operator is not supported in -source 1.5
(use -source 7 or higher to enable diamond operator)
ให้แก้ไขไฟล์ pom.xml โดยเพิ่ม plugin maven-compiler-plugin
จากนั้น Reload Maven Project

```
<?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>ku.cs.classroom</groupId>
  <artifactId>javafx-classroom-1</artifactId>
  <version>1.0-SNAPSHOT</version>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <source>1.8</source>
```

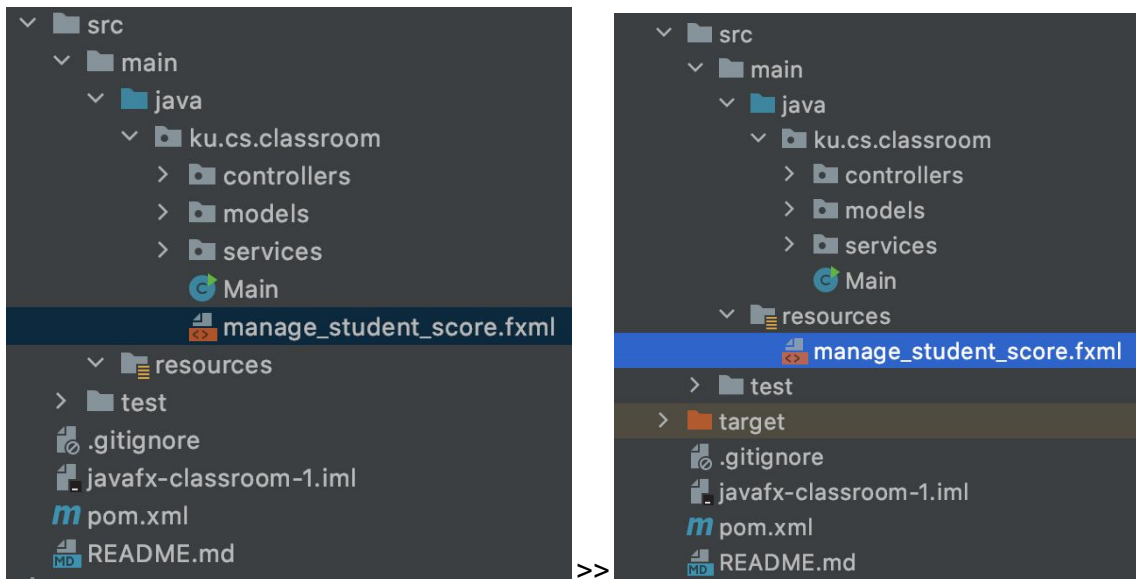
```

        <target>1.8</target>
    </configuration>
</plugin>
</plugins>
</build>

</project>

```

8. รันไฟล์ Main Class อีกครั้ง จะพบว่ามี error `java.lang.reflect.InvocationTargetException` เนื่องจากไฟล์ fxml ต้องย้ายไปอยู่ใน folder resources



9. ตรวจสอบ Main Class การอ้างอิงไฟล์ fxml จาก resources จะต้องอ้างจาก path /

```

package ku.cs.classroom;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        Parent root =
FXMLLoader.load(getClass().getResource("/manage_student_score.fxml"));
        primaryStage.setTitle("Hello World");
        primaryStage.setScene(new Scene(root, 800, 600));
    }
}

```

```
        primaryStage.show();  
    }  
  
    public static void main(String[] args) {  
        launch(args);  
    }  
}
```

10. ลองรัน Main class อีกครั้ง จะต้องได้หน้าต่างของ Application

ตัวอย่างการใช้งาน Maven Dependencies

11. เพิ่ม dependency bootstrapfx-core ในไฟล์ pom.xml (คำค้น: "maven bootstrapfx-core")
จากนั้น Reload Maven Project

```
<?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>ku.cs.classroom</groupId>
  <artifactId>javafx-classroom-1</artifactId>
  <version>1.0-SNAPSHOT</version>

  <dependencies>
    <!-- https://mvnrepository.com/artifact/org.kordamp.bootstrapfx/bootstrapfx-core -->
    <dependency>
      <groupId>org.kordamp.bootstrapfx</groupId>
      <artifactId>bootstrapfx-core</artifactId>
      <version>0.2.4</version>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

12. ที่ Main Class เพิ่ม css file ให้กับ Scene

```
package ku.cs.classroom;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        Parent root =
FXMLLoader.load(getClass().getResource("/manage_student_score.fxml"));
        primaryStage.setTitle("Hello World");
        primaryStage.setScene(new Scene(root, 800, 600));
        primaryStage.getScene()
                    .getStylesheets()
                    .add("org/kordamp/bootstrapfx/bootstrapfx.css");
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

13. Apply Style ให้กับปุ่ม ในคลาส ScoreController

```
package ku.cs.classroom.controllers;

// ...

public class ScoreController {

    // ...

    @FXML
    public void initialize() {
        updateScoreButton.setDisable(true);
        updateScoreButton.setStyleClass().setAll("btn", "btn-success");
        dataSource = new StudentHardcodeDataSource();
        students = dataSource.getStudentsData();
        students.findById("6210400002").addScore(ScoreType.ASSIGNMENT, 10);
    }
}
```

```

        showStudentData();

studentsTable.getSelectionModel().selectedItemProperty().addListener((observable, oldValue, newValue) -> {
    if (newValue != null) {
        showSelectedStudent(newValue);
    }
});
}

// ...
}

```

14. ลองรัน Main Class สังเกตปุ่ม Update Score จะเป็นสีเขียว

Student List

| Student ID | Name | Total Score | | |
|------------|---------|-------------|--|--|
| 6210400001 | Poirot | 0.0 | | |
| 6210400002 | Marple | 10.0 | | |
| 6210400003 | Maigret | 0.0 | | |
| 6210400004 | Morgan | 0.0 | | |
| 6210400005 | ชื่อไทย | 0.0 | | |

Selected Student

ID ...

Name ...

Score

Assignment

Attendance

Midterm

Final

การ build JavaFX Project ที่ใช้ Maven Framework

15. หากใช้วิธีการ build artifact ที่เคยสอนไป จะมีปัญหาการรัน jar file ที่ไม่มี dependency ติดไปกับ jar file ด้วย ให้แก้ไขไฟล์ pom.xml ดังนี้

- a. เพิ่ม properties เพื่อกำหนดค่าต่าง ๆ ดังนี้
out.folder คือชื่อ folder ที่บรรจุ jar file ที่ build เสร็จแล้ว
out.filename คือ ชื่อของ jar file ที่ build เสร็จแล้ว
project.mainClass คือ Main Class ของโปรเจกต์ ต้องระบุ package ด้วย

```
<?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>ku.cs.classroom</groupId>
  <artifactId>javafx-classroom-1</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <out.folder>62104xxxxx</out.folder>
    <out.filename>62104xxxxx-jar</out.filename>
    <project.mainClass>ku.cs.classroom.Main</project.mainClass>
  </properties>

  <dependencies>
    <!--
https://mvnrepository.com/artifact/org.kordamp.bootstrapfx/bootstrapfx-core
-->
    <dependency>
      <groupId>org.kordamp.bootstrapfx</groupId>
      <artifactId>bootstrapfx-core</artifactId>
      <version>0.2.4</version>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

```
        </plugin>
    </plugins>
</build>

</project>
```

b. เพิ่ม plugin ที่ช่วยเหลือในการ build jar file ดังนี้

```
<?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>ku.cs.classroom</groupId>
    <artifactId>javafx-classroom-1</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <out.folder>62104xxxxx</out.folder>
        <out.filename>62104xxxxx-jar</out.filename>
        <project.mainClass>ku.cs.classroom.Main</project.mainClass>
    </properties>

    <dependencies>
        <!--
https://mvnrepository.com/artifact/org.kordamp.bootstrapfx/bootstrapfx-core
-->
        <dependency>
            <groupId>org.kordamp.bootstrapfx</groupId>
            <artifactId>bootstrapfx-core</artifactId>
            <version>0.2.4</version>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.1</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
```

```
    </configuration>
  </plugin>
```

```
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-jar-plugin</artifactId>
    <version>2.4</version>
    <configuration>
      <archive>
        <manifest>
          <addClasspath>true</addClasspath>
          <classpathPrefix>lib/</classpathPrefix>
          <mainClass>${project.mainClass}</mainClass>
        </manifest>
        <manifestEntries>
          <Class-Path>.</Class-Path>
        </manifestEntries>
      </archive>
      <finalName>${out.folder}/${out.filename}</finalName>
    </configuration>
  </plugin>
```

```
  <plugin>
    <artifactId>maven-resources-plugin</artifactId>
    <version>2.6</version>
    <executions>
      <execution>
        <id>copy-resources</id>
        <phase>validate</phase>
        <goals>
          <goal>copy-resources</goal>
        </goals>
      </execution>
    </executions>
    <configuration>
```

```
      <outputDirectory>${basedir}/target/${out.folder}</outputDirectory>
      <resources>
        <resource>
          <directory>resources</directory>
          <filtering>true</filtering>
        </resource>
      </resources>
    </configuration>
  </execution>
</executions>
</plugin>
```

```
<plugin>
```

```

        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-dependency-plugin</artifactId>
        <executions>
            <execution>
                <id>copy-dependencies</id>
                <phase>prepare-package</phase>
                <goals>
                    <goal>copy-dependencies</goal>
                </goals>
                <configuration>

<outputDirectory>${project.build.directory}/${out.folder}/lib</outputDirectory>

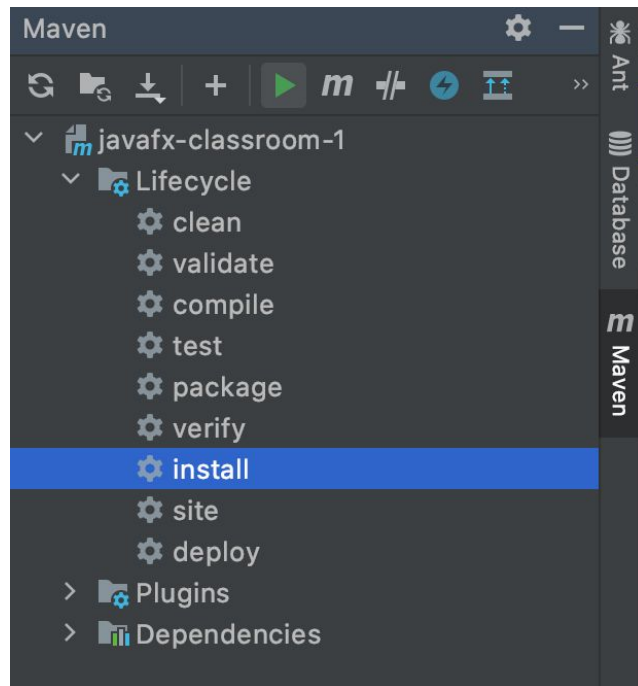
                    <overwriteReleases>false</overwriteReleases>
                    <overwriteSnapshots>false</overwriteSnapshots>
                    <overwriteIfNewer>true</overwriteIfNewer>
                </configuration>
            </execution>
        </executions>
    </plugin>
</plugins>
</build>

</project>

```

c. จากนั้น Reload Maven Project

16. เปิด panel Maven (ทางขวาของ IntelliJ IDEA) แล้วเลือก Lifecycle > install จากนั้นกดปุ่ม Run Maven Build (สามเหลี่ยมสีเขียว)



จะได้ jar file ที่ build เสร็จแล้ว ใน folder target ซึ่งสามารถรันได้พร้อม dependency โดยจะต้องมี folder lib อยู่กับ jar file ด้วย จึงจะรันได้
สังเกตว่าใน folder lib จะมี dependency bootstrapfx-core ด้วย

