✅ Step 1: Check Your Setup

- **Java Version**: You need **JDK 14 or later** (check with `java -version`).
- **JavaFX SDK**: Download from https://openjfx.io and extract it.
    - Example path:
        - Windows: `C:\javafx-sdk-21`
        - macOS/Linux: `/opt/javafx-sdk-21`

✅ Step 2: Compile Your Project

From your project folder:

Shell
mkdir out
javac --module-path "C:\javafx-sdk-21\lib" --add-modules javafx.controls,javafx.fxml -d out $(find . -name "*.java")
Show more lines

(Adjust path for macOS/Linux.)

✅ Step 3: Create JAR

Create a manifest file `manifest.mf`:

```
None
Manifest-Version: 1.0
Main-Class: your.package.MainClass
```

Then package:

Shell
jar cfm MyApp.jar manifest.mf -C out .

Show more lines

✅ Step 4: Use jpackage to Create a Native App

Run:

```
jpackage --name MyApp \
--input . \
--main-jar MyApp.jar \
--main-class your.package.MainClass \
--module-path "C:\javafx-sdk-21\lib" \
--add-modules javafx.controls,javafx.fxml \
--type exe
```

Show more lines

For macOS:

Shell

```
jpackage --name MyApp \
--input . \
--main-jar MyApp.jar \
--main-class your.package.MainClass \
--module-path /opt/javafx-sdk-21/lib \
--add-modules javafx.controls,javafx.fxml \
--type dmg
```

Show more lines

This will create:

- **Windows**: `MyApp.exe` installer
- **macOS**: `MyApp.dmg` installer
- **Linux**: `MyApp.deb` or `MyApp.rpm`

✅ Step 5: Double-Click and Run

After installation, you'll have a **native app icon** that launches without needing `java --module-path ...`.

✅ Optional Enhancements:

- Add an **icon**: `--icon myicon.ico` (Windows) or `.icns` (macOS).
- Add **license**, **version**, etc. with `--app-version`, `--license-file`.

🔥 **Result:** Your JavaFX app will behave like a normal desktop app—double-clickable, no command line needed.