

Guide d'installation OpenJFX sous macOS

Prérequis : Installation de Java

Avant d'installer OpenJFX, vous devez avoir Java installé sur votre système.

⚠ IMPORTANT - Compatibilité des versions : La version d'OpenJFX doit correspondre à la version de Java. Par exemple : - Java 11 → OpenJFX 11 - Java 17 → OpenJFX 17 - Java 21 → OpenJFX 21

Installer Java via Homebrew

```
brew install openjdk
```

Configurez le PATH en ajoutant à votre ~/.zshrc (ou ~/.bash_profile si vous utilisez bash) :

```
# Pour Apple Silicon (M1/M2/M3)
export PATH="/opt/homebrew/opt/openjdk/bin:$PATH"
```

```
# Pour Intel Mac
# export PATH="/usr/local/opt/openjdk/bin:$PATH"
```

Rechargez votre configuration :

```
source ~/.zshrc
```

Vérifiez l'installation :

```
java -version
javac -version
```

Installer une version spécifique de Java

Si vous avez besoin d'une version particulière :

```
# Java 11
brew install openjdk@11
```

```
# Java 17 (LTS)
brew install openjdk@17
```

```
# Java 21 (LTS)
brew install openjdk@21
```

Puis configurez le PATH avec la version souhaitée :

```
# Pour Java 17 par exemple
export PATH="/opt/homebrew/opt/openjdk@17/bin:$PATH"
```

Installation d'OpenJFX

Méthode 1 : Via Homebrew (recommandé)

```
brew install openjdk openjfx
```

Cette commande installe Java (JDK) et OpenJFX.

Si vous avez déjà installé Java, vous pouvez installer uniquement OpenJFX :

```
brew install openjfx
```

Méthode 2 : Installation manuelle depuis le site officiel

1. Téléchargez OpenJFX depuis openjfx.io
2. Choisissez la version macOS (SDK)
3. Extrayez l'archive dans un répertoire, par exemple ~/javafx ou /Library/Java/JavaFX
4. Notez le chemin du répertoire lib

Vérification de l'installation

Vérifier Java

```
java -version  
javac -version
```

Vous devriez voir la version de Java installée (OpenJDK).

Vérifier OpenJFX

```
brew list openjfx
```

Pour trouver le chemin d'installation :

```
brew info openjfx
```

Le chemin typique est /opt/homebrew/opt/openjfx/lib (Apple Silicon) ou /usr/local/opt/openjfx/lib (Intel).

Vérifier la compatibilité des versions

```
# Vérifier La version de Java  
java -version
```

```
# Vérifier La version d'OpenJFX  
brew info openjfx | grep openjfx:
```

Les numéros de version majeure doivent correspondre (par exemple, Java 17.x.x avec OpenJFX 17.x.x).

Script de lancement d'application JavaFX

Créez un fichier lancer-app.sh :

```
#!/bin/bash

# Se placer dans Le répertoire du script
cd "$(dirname "$0")"

# Détection automatique de L'architecture
if [ -d "/opt/homebrew/opt/openjfx/lib" ]; then
    # Apple Silicon
    JAVAFX_PATH="/opt/homebrew/opt/openjfx/lib"
elif [ -d "/usr/local/opt/openjfx/lib" ]; then
    # Intel
    JAVAFX_PATH="/usr/local/opt/openjfx/lib"
else
    echo "OpenJFX non trouvé. Installez-le avec : brew install openjfx"
    exit 1
fi

CLASSPATH="lib/*:EspFx.jar:Thermoptim.jar" # Ajout du dossier lib et du jar principal
MAIN_CLASS="rg.espfx.EspFx"

# Modules JavaFX nécessaires
MODULES="javafx.controls,javafx.fxml,javafx.graphics,javafx.base"

# Lancement de L'application avec options macOS
java --module-path "$JAVAFX_PATH" \
    --add-modules "$MODULES" \
    -cp "$CLASSPATH" \
    -Xdock:name="EspFx" \
    -Dapple.laf.useScreenMenuBar=true \
    "$MAIN_CLASS"
```

Rendre le script exécutable

```
chmod +x lancer-app.sh
```

Lancement de l'application

```
./lancer-app.sh
```

Script universel (Intel + Apple Silicon)

Pour un script qui fonctionne sur les deux architectures :

```
#!/bin/bash
```

```

# Détection automatique de l'architecture
if [ -d "/opt/homebrew/opt/openjfx/lib" ]; then
    # Apple Silicon
    JAVAFX_PATH="/opt/homebrew/opt/openjfx/lib"
elif [ -d "/usr/local/opt/openjfx/lib" ]; then
    # Intel
    JAVAFX_PATH="/usr/local/opt/openjfx/lib"
else
    echo "OpenJFX non trouvé. Installez-le avec : brew install openjfx"
    exit 1
fi

APP_JAR="MonApplication.jar"
MODULES="javafx.controls,javafx.fxml,javafx.graphics,javafx.base"

java --module-path "$JAVAFX_PATH" \
    --add-modules "$MODULES" \
    -jar "$APP_JAR"

```

Configuration pour une installation manuelle

Si vous avez installé OpenJFX manuellement, modifiez le chemin dans le script :

```
JAVAFX_PATH="$HOME/javafx/lib" # Adaptez selon votre installation
```

Modules JavaFX disponibles

Ajustez la variable MODULES selon vos besoins :

- javafx.base - Classes de base
- javafx.controls - Composants UI (boutons, tables, etc.)
- javafx.fxml - Support FXML
- javafx.graphics - Graphiques et rendu
- javafx.media - Support multimédia
- javafx.swing - Intégration Swing
- javafx.web - Navigateur web intégré

Exemple complet avec options JVM

```

#!/bin/bash

# Détection du chemin OpenJFX
if [ -d "/opt/homebrew/opt/openjfx/lib" ]; then
    JAVAFX_PATH="/opt/homebrew/opt/openjfx/lib"
elif [ -d "/usr/local/opt/openjfx/lib" ]; then
    JAVAFX_PATH="/usr/local/opt/openjfx/lib"
else
    echo "Erreur: OpenJFX non trouvé"

```

```

        exit 1
    fi

    APP_JAR="MonApplication.jar"
    MODULES="javafx.controls,javafx.fxml"

    # Options pour macOS
    java --module-path "$JAVAFX_PATH" \
        --add-modules "$MODULES" \
        -Xmx512m \
        -Xdock:name="Mon Application" \
        -Xdock:icon="icon.icns" \
        -Dapple.laf.useScreenMenuBar=true \
        -Dfile.encoding=UTF-8 \
        -jar "$APP_JAR" "$@"

```

Options spécifiques macOS : - -Xdock:name : Nom affiché dans le Dock - -Xdock:icon : Icône personnalisée dans le Dock - -Dapple.laf.useScreenMenuBar=true : Menu dans la barre macOS

Création d'une application .app

Pour créer une véritable application macOS, créez la structure suivante :

```

MonApp.app/
├── Contents/
│   ├── Info.plist
│   ├── MacOS/
│   │   └── lanceur.sh
│   └── Resources/
│       ├── MonApplication.jar
│       └── icon.icns

```

Fichier Info.plist :

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>CFBundleName</key>
    <string>Mon Application</string>
    <key>CFBundleExecutable</key>
    <string>lanceur.sh</string>
    <key>CFBundleIconFile</key>
    <string>icon.icns</string>
    <key>CFBundleIdentifier</key>
    <string>com.exemple.monapp</string>
    <key>CFBundleVersion</key>
    <string>1.0</string>

```

```
</dict>
</plist>
```

Fichier lanceur.sh dans MacOS/ :

```
#!/bin/bash
cd "$(dirname "$0")/../Resources"

if [ -d "/opt/homebrew/opt/openjfx/lib" ]; then
    JAVAFX_PATH="/opt/homebrew/opt/openjfx/lib"
else
    JAVAFX_PATH="/usr/local/opt/openjfx/lib"
fi

java --module-path "$JAVAFX_PATH" \
    --add-modules javafx.controls,javafx.fxml \
    -Xdock:name="Mon Application" \
    -jar MonApplication.jar
```

Rendez le script exécutable :

```
chmod +x MonApp.app/Contents/MacOS/lanceur.sh
```

Dépannage

Erreur “java: command not found”

Java n’est pas installé ou n’est pas dans le PATH.

Installez Java :

```
brew install openjdk
```

Puis configurez le PATH dans votre ~/.zshrc :

```
export PATH="/opt/homebrew/opt/openjdk/bin:$PATH" # Apple Silicon
# ou pour Intel : export PATH="/usr/local/opt/openjdk/bin:$PATH"
```

Rechargez : source ~/.zshrc

Erreur d’incompatibilité de version

Si vous obtenez des erreurs étranges au lancement, vérifiez que les versions correspondent :

```
java -version # Notez la version majeure (11, 17, 21, etc.)
brew info openjfx # Doit avoir la même version majeure
```

Si les versions ne correspondent pas, installez une version spécifique :

```
# Exemple pour Java 17
brew install openjdk@17
brew install openjfx@17
```

Erreur “JavaFX runtime components are missing”

Vérifiez que : - OpenJFX est bien installé : `brew list openjfx` - Le chemin `JAVAFX_PATH` est correct - Les modules nécessaires sont bien spécifiés

Trouver le chemin d’installation

```
brew --prefix openjfx
```

Puis ajoutez `/lib` au chemin retourné.

Problème de sécurité macOS

Si macOS bloque l’application, allez dans : **Préférences Système > Confidentialité et sécurité** et autorisez l’exécution.

Versions

Pour vérifier la version installée :

```
brew info openjfx
```

Pour mettre à jour :

```
brew upgrade openjfx
```

Alternative : jpackage

Pour créer un vrai bundle macOS professionnel, utilisez `jpackage` (inclus avec JDK 14+) :

```
jpackage --type app-image \
  --name "MonApplication" \
  --input target \
  --main-jar MonApplication.jar \
  --main-class com.exemple.MainApp \
  --module-path /opt/homebrew/opt/openjfx/lib \
  --add-modules javafx.controls,javafx.fxml \
  --icon icon.icns
```