To add JavaFX to WSL2, you have two primary options: use the **system's package manager** to install `openjfx` or **download the standalone JavaFX SDK** and configure your project build path (e.g., with Maven or Gradle). Since JavaFX 25 is likely the next major release (as of current date), these general methods will apply.

Prerequisites

- Ensure WSL2 is installed and running with a Linux distribution (e.g., Ubuntu).
- Ensure you have a JDK installed within your WSL2 environment. If not:

  bash

  ```bash
  sudo apt update
  sudo apt install openjdk-21-jdk # Or the version compatible with JavaFX 25
  ```

Option 1: Install OpenJFX via Package Manager (Recommended)

This is the simplest way to get the relevant JavaFX libraries (OpenJFX) on your Linux distribution within WSL2.

1. **Update your package lists**:

   bash

   ```bash
   sudo apt update
   ```

2. **Install the OpenJFX package**:

   bash

   ```bash
   sudo apt install openjfx
   ```

   This command installs the OpenJFX libraries and makes them available to the system's OpenJDK installation. The system package manager will handle compatibility and dependencies.

Option 2: Use a Standalone JavaFX SDK with your Project

If you need a specific version or are managing multiple projects with different JavaFX versions, you can download the SDK from the Gluon website and configure your project's build tool (like Maven or

Gradle).

1. **Download the JavaFX SDK for Linux (x64)** from the Gluon website to a location in your WSL2 filesystem (e.g., `~/downloads`).
2. **Unzip the archive**:

   ```bash
   tar -xvf openjfx-XX-linux-x64.tar.gz # Replace XX with the actual version number
   ```

3. **Configure your project** (e.g., using Maven):
   1. You'll need to specify the path to the JavaFX SDK's `lib` folder in your build configuration (like `pom.xml`).
   2. For running from the command line, you need to add VM options to specify the module path and required modules:

      ```bash
      java --module-path /path/to/javafx-sdk-25/lib --add-modules javafx.controls,javafx.fxml
      ```

   3. Most developers use an IDE (like VS Code, IntelliJ) which offers integrated tools and wizards for setting up JavaFX projects with Maven archetypes, which simplifies the configuration of module paths and dependencies.

Running GUI Applications in WSL2

For JavaFX applications (which are GUI apps) to display correctly in WSL2, you must have a compatible display server set up (like X server or Wayland). Modern Windows 11 versions include **WSLg**, which provides automatic GUI support. Ensure WSLg is working correctly on your system.

Once set up, your JavaFX application should run just like any native Linux GUI application when executed within the WSL2 environment.