

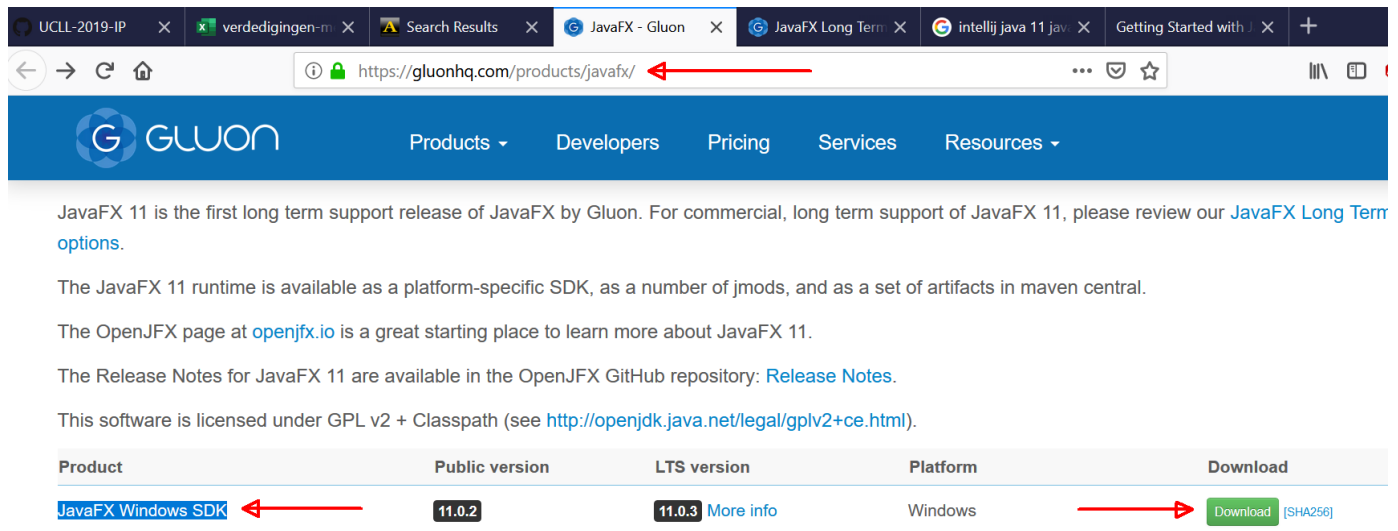
JavaFx en Java (JDK) 11 (of hoger)

Voer de eerste keer alle stappen uit!

Voor ieder nieuw project moet je altijd de stappen 2 en 4 uitvoeren.

1. Download de juist JavaFx SDK (Software Development Kit)

Surf naar <https://gluonhq.com/products/javafx/> en download de JavaFx SDK die je nodig hebt (Windows/Mac/Linux). Ik heb gekozen voor versie 11, omdat die LTS is (Long Time Support), maar je kunt net zo goed versie 12 kiezen (je moet wel minstens Java JDK 11 hebben voor JavaFx 12, en voor JavaFx 11 de JDK 10 (moet een OpenJDK build zijn, dus ik raad aan om gewoon JDK 11 te gebruiken).



JavaFX 11 is the first long term support release of JavaFX by Gluon. For commercial, long term support of JavaFX 11, please review our [JavaFX Long Term options](#).

The JavaFX 11 runtime is available as a platform-specific SDK, as a number of jmods, and as a set of artifacts in maven central.

The OpenJFX page at openjfx.io is a great starting place to learn more about JavaFX 11.

The Release Notes for JavaFX 11 are available in the OpenJFX GitHub repository: [Release Notes](#).

This software is licensed under GPL v2 + Classpath (see <http://openjdk.java.net/legal/gplv2+ce.html>).

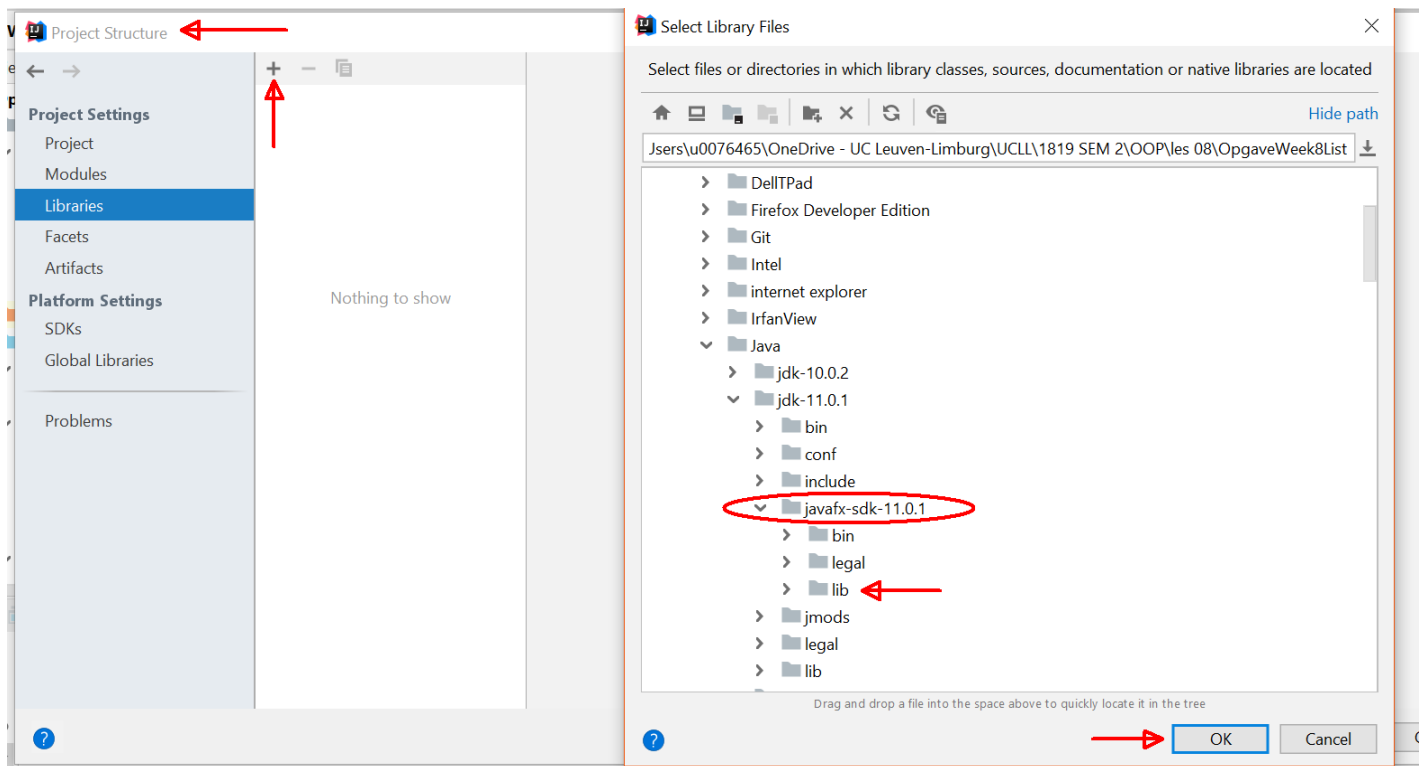
Product	Public version	LTS version	Platform	Download
JavaFX Windows SDK	11.0.2	11.0.3 More info	Windows	Download [SHA256]

Je krijgt een ZIP file, die je kunt unzippen waar je wilt. In mijn geval heb ik dat gedaan in “C:\Program Files\Java\jdk-11.0.1”, maar het maakt niet veel uit.

2. Voeg de library toe

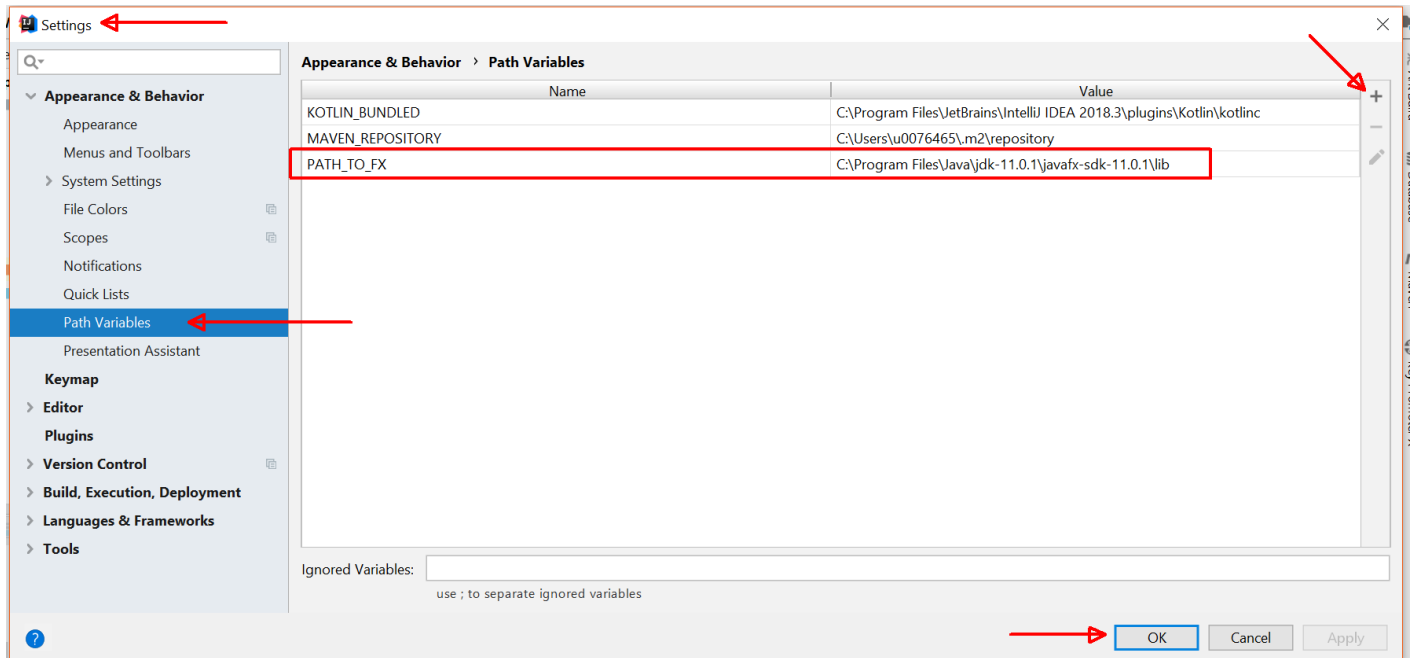
Vervolgens ga je naar je project en in het Project Structure venster (File – Project Structure of CTRL+SHIFT+ALT+S) voeg je een nieuwe library toe. Je selecteert gewoon de “lib” folder in je JavaFx SDK folder (in mijn geval dus “C:\Program Files\Java\jdk-11.0.1\javafx-sdk-11.0.1\lib”). Klik dan op OK.

Als je dat gedaan hebt zal de IDE de JavaFX classes herkennen. Maar let op, als je nu compileert dan krijg je nog steeds een fout: “Error: JavaFX runtime components are missing, and are required to run this application”.

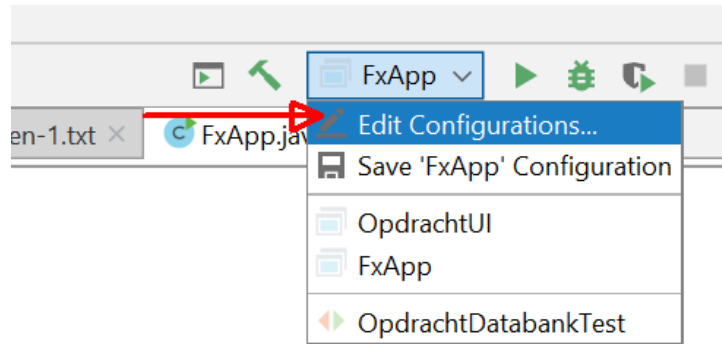


3. Zet een path naar JavaFx

Ga naar Settings (File – Settings of CTRL+ALT+S), zoek daar naar “Path” en bij “Path Variables” kun je dan een path variable definiëren. Klik op “+” en geef “PATH_TO_FX” in als variable (of iets van je eigen keuze) en als path geef je het pad naar de “lib” folder in je JavaFx SDK folder (in mijn geval dus “C:\Program Files\Java\jdk-11.0.1\javafx-sdk-11.0.1\lib”). Klik vervolgens op OK.



4. Pas de configuratie aan



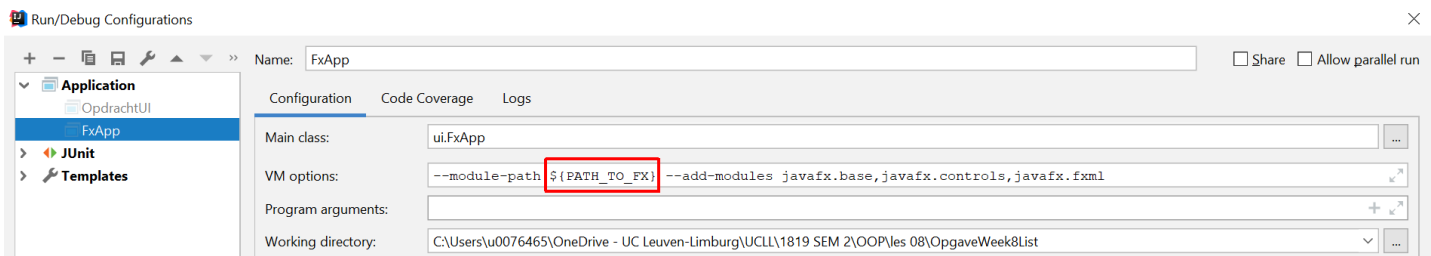
Kies VM-options, en vul daar `--module-path "C:\Program Files\Java\jdk-11.0.1\javafx-sdk-11.0.1\lib"` `--add-modules javafx.base,javafx.controls,javafx.fxml` in (je kunt het vak groter maken zoals op de figuur door op de pijltjes rechtsboven in het vak te klikken – en opnieuw kleiner door er nog eens op te klikken).

Merk op dat je hier het pad naar de “lib” folder op jouw machine moet inzetten.



Of, beter, aangezien je toch een path variable hebt aangemaakt, kun je in plaats van het echte pad in te vullen de path variable gebruiken: je vervangt `"C:\Program Files\Java\jdk-11.0.1\javafx-sdk-11.0.1\lib"` gewoon door `${PATH_TO_FX}`. Zie figuur hieronder.

Als je een andere naam gekozen hebt voor je path variable gebruik je die natuurlijk.



Beide methodes werken, maar de voorkeur gaat uit naar de laatste variant.

Bemerkt dat hier de modules `javafx.base`, `javafx.controls` en `javafx.fxml` geladen worden. Indien je er meer nodig hebt, kies je gewoon de juiste modules en plaats je die erbij, alles gescheiden door een komma.

5. Modules

De beschikbare modules zijn de JAR files in de “lib” folder van je JavaFx SDK. Merk op dat je de namen gebruikt zonder de .jar extensie in de VM-options!

