

## **Leerinhoud JavaFX Scene Builder.**

Gebruik van Scene Builder bij een JavaFX applicatie

**HO  
GENT**

# 1. Inleiding

- Met JavaFX Scene Builder kan je een GUI maken met "drag & drop".  
Met behulp van CSS (zie hoofdstuk 22) kan je ook de look-and-feel van je GUI aanpassen.
- JavaFX Scene Builder genereert FXML (FX Markup Language) - een XML om JavaFX GUI controls te definiëren en te schikken zonder java code te schrijven. Je hoeft echter geen (F)XML te kennen om dit hoofdstuk te bestuderen.

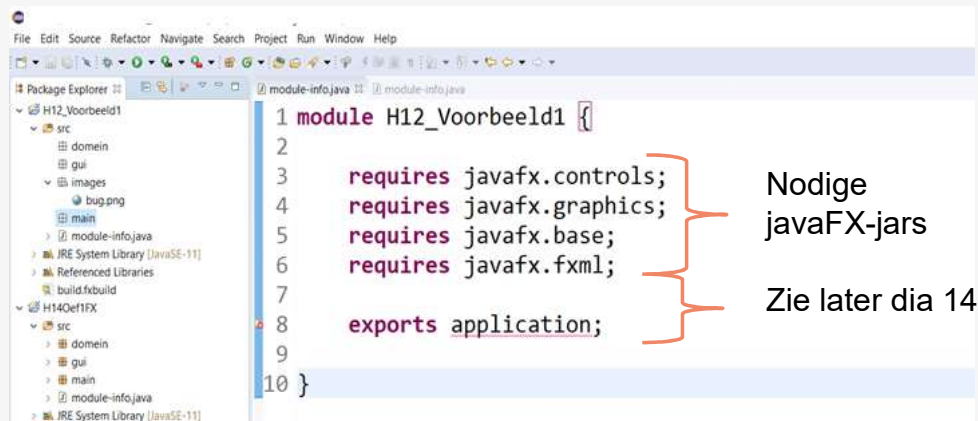
**HO  
GENT**

# 2. Een eerste voorbeeld: H12\_Voorbeeld1

- We maken FXVoorbeeld1 uit het vorige hoofdstuk opnieuw, nu m.b.v. Scene Builder
- Maak een nieuw JavaFX project in Eclipse:
  - File - New – Other... - **JavaFX Project** - H12\_Voorbeeld1
  - Voeg de 4 javafx-jars toe in het **Modulepath** van het project
  - Verwijder package application
  - Maak packages: **main**, **gui**, **domein** en **images** met daarin bug.png
  - Voeg module-info.java toe via **Configure / Create module-info.java** en pas aan (zie volgende slide)

**HO  
GENT**

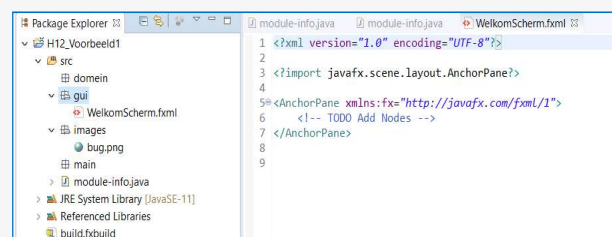
## 2. Een eerste voorbeeld: H12\_Voorbeeld1



**HO  
GENT**

## FXML bestand

- In package gui:
  - New / Other... / JavaFX
    - New FXML Document**
    - Name: WelkomScherf
- Klik op WelkomScherf.fxml met rechtermuisknop en selecteer **Open with SceneBuilder**.  
Ontwerp nu het scherm in Scene Builder

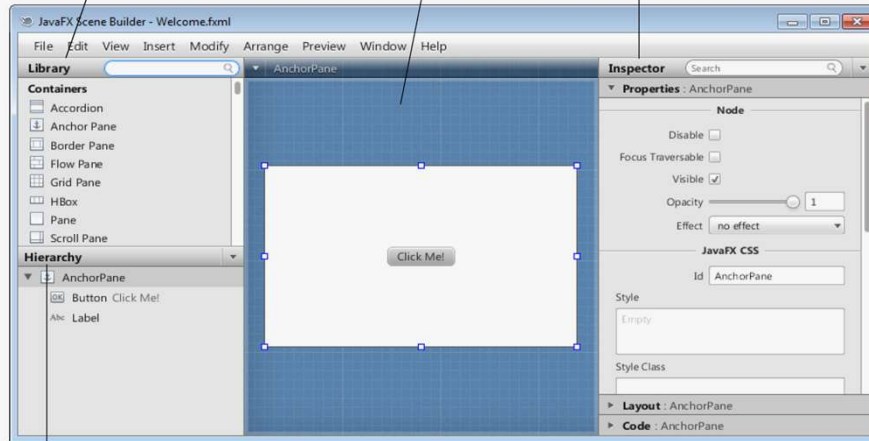


**HO  
GENT**

The **Library** contains JavaFX **Containers**, **Controls** and other items that can be dragged and dropped on the canvas

You use the content panel to design the GUI

You use the **Inspector** window to configure the currently selected item in the content panel



The **Hierarchy** window shows the structure of the GUI and allows you to select and reorganize controls

**Fig. 25.7** | JavaFX Scene Builder displaying the default GUI in `Welcome.fxml`.

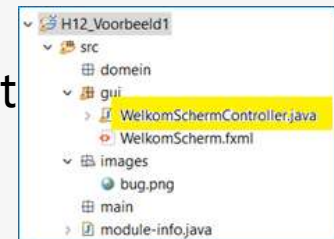
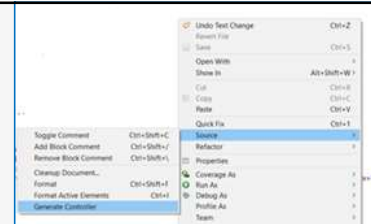
**HO  
GENT**

- Containers: VBox
  - Properties:
    - Alignment: CENTER
  - Layout:
    - Pref Width: 450
    - Pref Height: 300
- Controls: Label
  - Properties:
    - Tekst: Welkom bij JavaFX!
    - Font: System Bold, 30 px
- Controls: ImageView
  - Properties: Image
  - Layout:
    - Fit Width: 0
    - Fit Height: 0
- Controller: Use `fx:root` construct

**HO  
GENT**

## Controller

- Klik in de WelkomScherf-code met rechtermuisknop en kies **Source / Generate Controller**.  
=> WelkomScherfController wordt aangemaakt.



- De Controller-klasse wordt onze link tussen het domein en de gui.

**HO  
GENT**

```

1 package gui;
2
3 import java.io.IOException;
4
5
6
7
8 public class WelkomScherfController extends VBox } toegepaste layout
9 {
10     public WelkomScherfController()
11     {
12         FXMLLoader loader = new FXMLLoader
13             (getClass().getResource("WelkomScherf.fxml"));
14         loader.setController(this);
15         loader.setRoot(this);
16         try
17         {
18             loader.load();
19         } catch (IOException ex)
20         {
21             throw new RuntimeException(ex);
22         }
23     }
24 }
  
```

Bijbehorende  
fxml-bestand  
wordt  
ingeladen

**HO  
GENT**

## Vervelend

- Na elke wijziging in de Scene Builder kan je de Controller-klasse opnieuw laten genereren.

Eclipse laat **niet** toe de nieuwe wijzigingen in een bestaande Controller-klasse toe te voegen/aan te passen.

We kunnen dus best steeds een nieuwe Controller-klasse (met andere naam) aanmaken en handmatig de nieuwe toegevoegde componenten/opties aanpassen in onze eerste aangemaakte Controller-klasse.

**HO  
GENT**

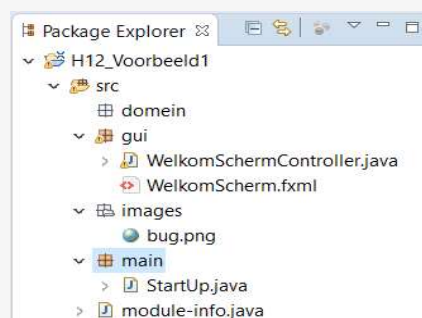
## 3 lagen model

In de package main:

New / Other... / JavaFX / Classes

**JavaFX Main Class**

Name: StartUp



**HO  
GENT**

```

1 package main;
2
3 import gui.WelkomScherController;
4
5
6
7
8 public class StartUp extends Application {
9
10     @Override
11     public void start(Stage primaryStage) {
12         WelkomScherController root = new WelkomScherController();
13         Scene scene = new Scene(root, 600, 400);
14
15         primaryStage.setTitle("Welkom!");
16         primaryStage.setScene(scene);
17         primaryStage.show();
18     }
19
20
21     public static void main(String[] args) {
22         Launch(args);
23     }
24 }

```

Object van  
de Controller-  
klasse!

**HO  
GENT**

## Aanpassen module-info.java

```

1 module H12_Voorbeeld1 {
2
3     requires javafx.controls;
4     requires javafx.graphics;
5     requires javafx.base;
6     requires javafx.fxml;
7
8
9     opens main to javafx.graphics, javafx.fxml;
10
11 }

```

Vanuit package  
**main**, scherm laten  
zien

**HO  
GENT**

# Run

**HO  
GENT**

## Bijkomend leermateriaal

- <http://docs.oracle.com/javase/8/javase-clienttechnologies.htm>
- <https://code.makery.ch/library/javafx-tutorial/>

**HO  
GENT**