

Développement d'une application Desktop en Java FX

Table des matières

<i>Section 1 Quelques classes – la salle de réunion.....</i>	<i>3</i>
1.1 But.....	3
1.2 Enoncé.....	3

Section 1 Généralités

1.1 But

Les applications de bureaux graphiques – présentant une IHM – développées en Java peuvent utiliser plusieurs bibliothèques :

- SWING
- JavaFX, plus récent et plus souple

Ce document présente les généralités de JavaFX, un démarrage rapide.

1.2 Télécharger JavaFX

Utiliser le site <https://gluonhq.com/products/javafx/>

1.3 Outil de développement

L'outil IntelliJ est utilisé dans cette présentation.

Télécharger et installer la version Community
<https://www.jetbrains.com/fr-fr/idea/download/#section=windows>

Le placement d'objets graphiques dans l'application va se faire avec l'outil scene builder. Il est intégré dans IntelliJ, moyennant quelques installations supplémentaires lors du 1^{er} emploi.

L'outil peut aussi être utilisé en dehors de IntelliJ et offre dans ce cas plus d'espace pour dessiner. Scene builder est disponible à l'adresse <https://gluonhq.com/products/scene-builder/>

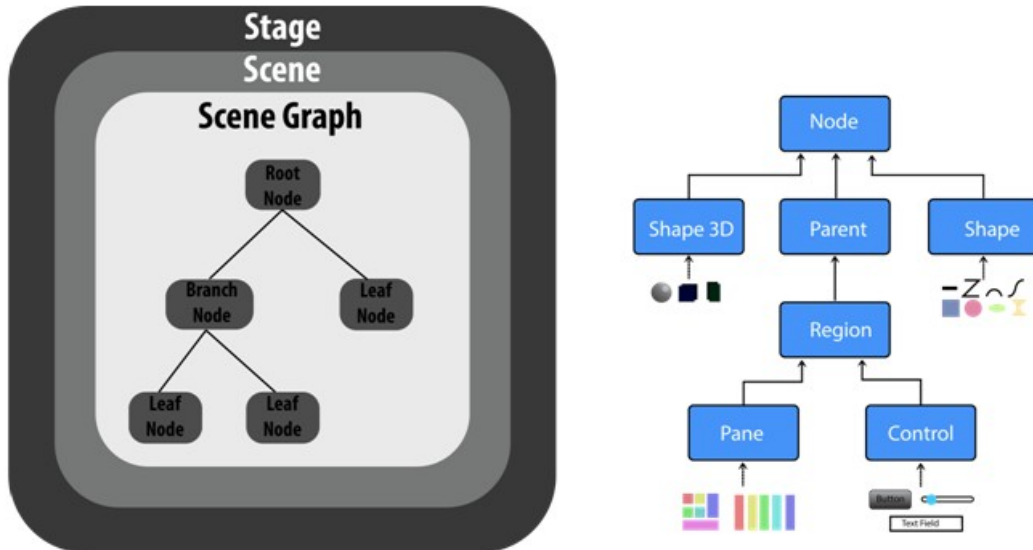
1.4 Liens web

https://www.youtube.com/watch?v=pZzU-e_doNE

<https://www.javatpoint.com/javafx-tutorial>

1.5 Architecture

Une vue dans Java FX est un contenu arborescent.



L'extrait de code suivant contient les instructions pour définir une fenêtre avec un label :

```
Label unLabel = new Label("I'm a Label on new Window");

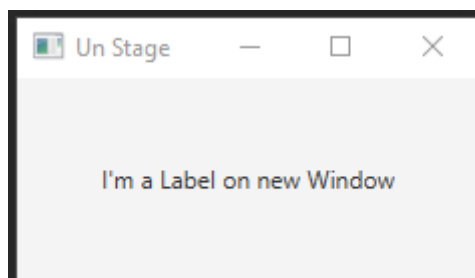
StackPane unLayout = new StackPane();
unLayout.getChildren().add(unLabel);

Scene uneScene = new Scene(unLayout, 230, 100);

// New window (Stage)
Stage newWindow = new Stage();
newWindow.setTitle("Un Stage");
newWindow.setScene(uneScene);

newWindow.show();
```

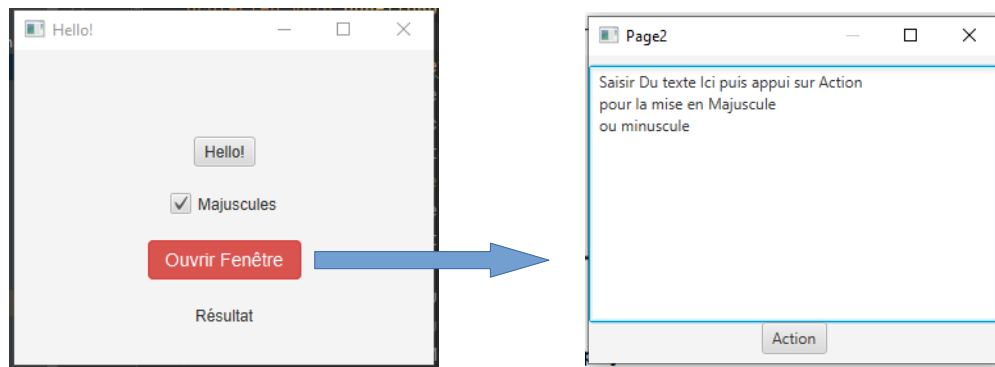
et affiche la fenêtre :



1.6 Créer une application simple

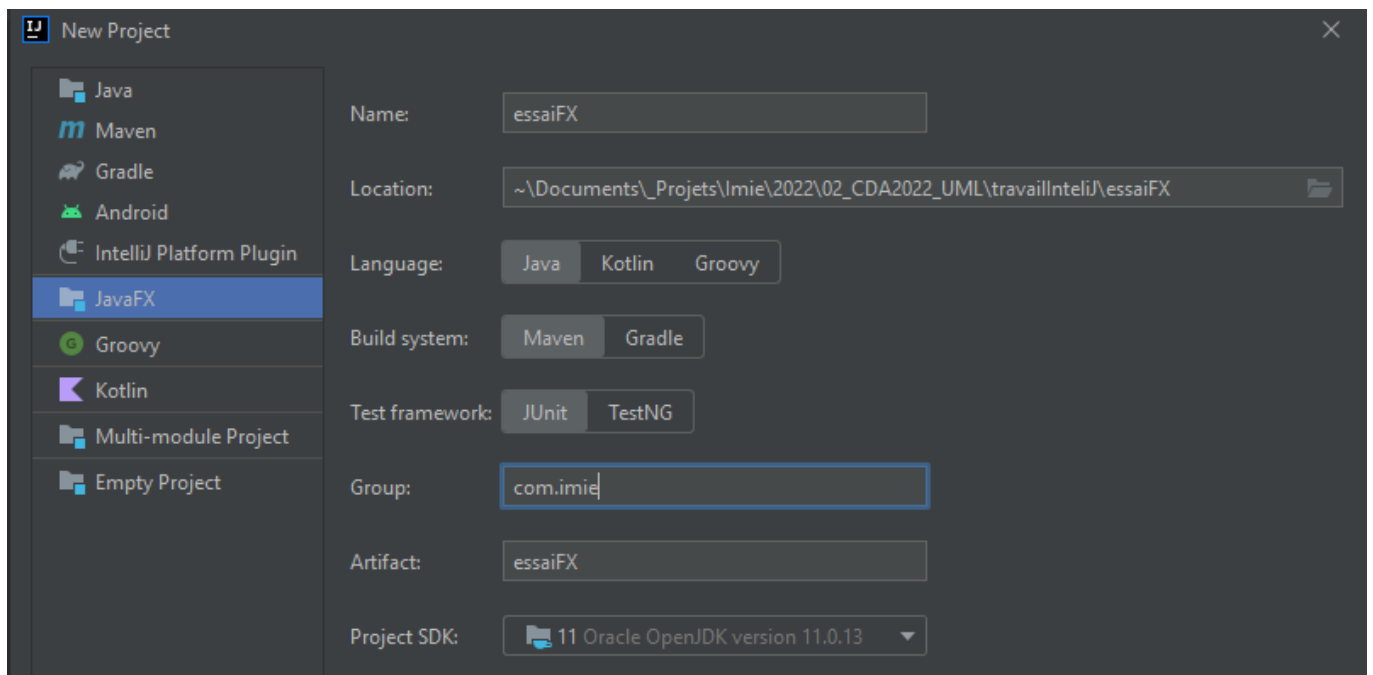
1.6.1 But

Nous allons créer une application de base avec 2 fenêtres. Il y a passage d'informations entre les fenêtres.

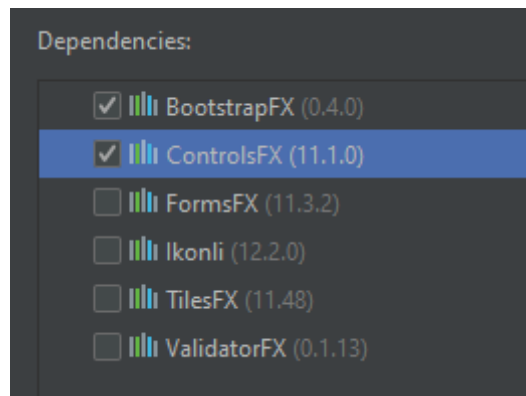


1.6.2 Créer le projet

Depuis IntelliJ, New Project puis saisir comme suit :



puis choisir les dépendances suivantes :



Ces options sont visibles sur les sites :

<https://github.com/kordamp/bootstrapfx>

<https://controlsfx.github.io/>

1.6.3 Lancer l'application

Run sur l'application. Utiliser le bouton de la fenêtre.

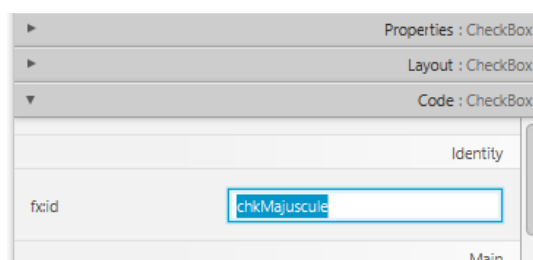
1.6.4 Modifier le contenu de la fenêtre

Editer le fichier hello-view.fxml.

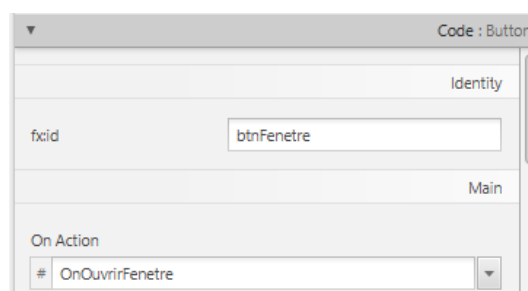
Clic sur l'onglet Scene Builder.

Ajouter sous le bouton existant :

- un checkbox. Le texte est « Majuscule », a pour identifiant chkMajuscule.

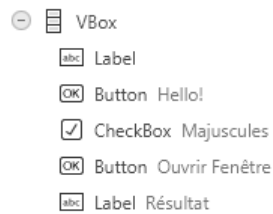


- Un bouton. Le texte est « Ouvrir Fenêtre ». L'identifiant est btnFenetre, l'action sur clic est OnOuvrirFenetre



- Un Label. Le texte est Résultat, L'id est : lblResult

L'arborescence de vue doit ressembler à ceci :



La vue ressemble à ceci :



Revenir sur le source fxml.

Placer le curseur sur `OnOuvrirFenetre` et accepter le Create Method que propose IntelliJ

```
ng="false" text="Majuscules" />
false" onAction="#OnOuvrirFenetre" text="Ouvr
```

Placer le curseur sur `chkMajuscule` et demander de créer un attribut (create Field in Page2Controller). Idem pour `lblResult`.

1.6.5 Ajouter du style

Nous avons fait le choix d'inclure BootstrapFX lors de la création du projet. Nous allons modifier la couleur du bouton `btnFenetre`, pour essai.

Dans `HelloApplication.java` ajouter le code suivant avant `stage.setScene(scene)` ;

:

```
HelloController controller = fxmlLoader.getController();
controller.btnFenetre.getStyleClass().setAll("btn", "btn-danger");
scene.getStylesheets().add(BootstrapFX.bootstrapFXStylesheet());
```

Essayer la vue.

1.6.6 Créer et appeler une nouvelle fenêtre

Le but est d'appeler une autre fenêtre sur appui du bouton btnFenetre de la fenêtre principale.

Nous allons de plus passer de l'information :

- depuis la fenêtre principale vers la nelle fenêtre : boolean qui indique si l'on souhaite traiter des majuscules ou non
- depuis la nelle fenêtre vers la fenêtre principale pour récupérer et afficher la chaîn,e transformée

Créer une nouvelle fenêtre :


Sur le nœud du projet resources / com.imie.essaifx faire New /FXML file. Donner pour nom Page2

Dans le fichier Page2.fxml indiquer le nom du contrôleur :

fx:controller=" Page2Controller "

Placer le curseur sur le mot Page2Controller et demander la création de la classe.

Ouvrir Scene Builder sur cette Page2.fxml

- Un  AnchorPane est placé comme élément par défaut dans la vue. Le remplacer par un VBox
- Pour cela, clic droit sur Anchor Pane et choisir Vrap in → VBox
- Supprimer le Anchor Pane
- Ajouter un TextArea avec un id : txtInput
- Ajouter un Bouton avec le texte Action, id : btnAction, et le On Action : OnAction

Dans le fichier Page2Controller placer ce code (garder les lignes actuelles du haut)

```
public class Page2Controller {  
    public TextArea txtInput;  
    private boolean isMajuscule=false;  
  
    public void setIsMajuscule(boolean isMajuscule) {  
        this.isMajuscule = isMajuscule;  
    }  
  
    public void OnAction(ActionEvent actionEvent) {  
        if (this.isMajuscule)  
            txtInput.setText(txtInput.getText().toUpperCase());  
        else  
            txtInput.setText(txtInput.getText().toLowerCase());  
    }  
}
```


Analyser ce code.

Appel de cette vue.

L'appel se fait depuis la vue principale, dans HelloController.java

Placer le code suivant au bon endroit :

```
public void OnOuvrirFenetre(ActionEvent actionEvent) throws IOException {
    FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource("Page2.fxml"));
    Scene scene = new Scene(fxmlLoader.load(), 320, 240);
    Stage stage = new Stage();
    stage.setTitle("Page2");
    stage.setScene(scene);
    Page2Controller controller = fxmlLoader.getController();

    controller.setIsMajuscule(chkMajuscule.isSelected());
    //controller.txtInput.setText("ABCD");
    stage.initModality(Modality.APPLICATION_MODAL);
    stage.showAndWait();
    lblResult.setText(controller.txtInput.getText());
}
```

Analyser ce code.

Lancer l'application et faites des essais de transformation de texte.

Ici dans cet exemple minimal, les vues et objets graphiques ont été définis par fxml.

Ils peuvent être définis aussi directement par du code java.