## JAVAFX-3

## ##理解布局

布局容器 (Layoutcontainer) 或面板 (Pane) 允许对JavaFX应用程序场景图中的UI控件进行灵活、动态的排布。JavaFX Layout API包括下列容器类:

- BorderPane类将其内容节点放到上、下、左、右、中各个区域中。
- HBox类将其内容节点横向排成一行。
- VBox类将其内容节点纵向排成一列。
- StackPane类将其内容节点摞在一起。
- GridPane类允许开发者创建一个灵活的网格,按行列来布局其内容节点。
- FlowPane类将其内容按行或列进行"流式"布局,当遇到横向或纵向的边界时自动进行换行或换列。
- TilePane类将其内容放到统一大小的单元格中。
- AnchorPane类允许开发者创建锚节点,将控件停靠于布局的上下左右各边,也可以居中停靠。

## ##JAVA FXml

- 附上官方文档 https://docs.oracle.com/javase/8/javafx/api/toc.htm
- 使用 SceneBuilder 构建场景 http://www.yiibai.com/javafx/javafx-tutorial-for-beginners.html
- 这里说明两点 注意控制器 我们还是以 简单的布局为例子

标签



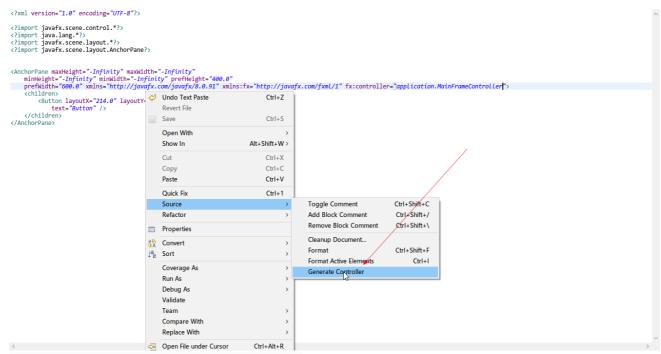
t="400.0"

9.91" xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.MainFrameController">

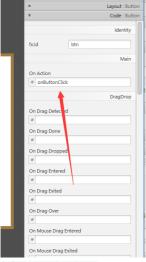
Parsing="false"

控制器

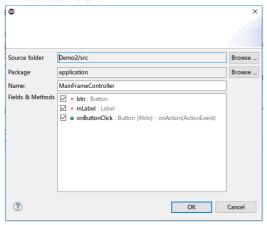
- 逆向生产控制器



- 注意要写组件的fx: id 值 不要搞错了 写成 id 就不行



- 生成控制器代码



- 编写核心代码(这部分代码一定要熟悉)

```
public void start(Stage primaryStage) {
    try {
       Parent root = FXMLLoader.load(this.getClass().getResource("MainFrame.fxml"));
       Scene scene = new Scene(root);
       scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
       primaryStage.setScene(scene);
       primaryStage.show();
    } catch(Exception e) {
       e.printStackTrace();
    }
- 控制层代码(fxml)
)
L public class MainFrameController {
    private Button btn;
10
   @FXML
    private Label mLabel;
    // Event Listener on Button[#btn].onAction
   @FXML
3⊝
    public void onButtonClick(ActionEvent event) {
)
)
      System.out.println("click");
L
        mLabel.setText("click");
    }
3 }
- 安卓mvvp写法
try {
    Parent root = FXMLLoader.load(this.getClass().getResource("MainFrame.fxml"));
    Scene scene = new Scene(root);
    scene.getStylesheets().add(getClass().getResource("application.css").toExternalFor
    primaryStage.setScene(scene);
     * 这里注意加上#字
   * 这种方式的优先级>在fxml 或者说其他地方写控制器的方式的
    Label lab =(Label) root.lookup("#lab");
    Button btn =(Button) root.lookup("#btn");
    btn.setOnAction((e)->{
        lab.setText("click");
    });
    primarvStage.show():
- 运行效果
                     click
```