



# Sven Ruppert

speaking java since 1996

Principal IT Consultant



Standort - München

# Twittert bitte ;-)

Zeigt der Community das ihr aktiv seid !!

Principal IT Consultant

codecentric 

Standort - München

@SvenRuppert

# Sven Ruppert

speaking java since 1996

Projekte aus den Bereichen:

- Automobilindustrie
- Energieversorgung
- Finanzen / Leasingbranche
- Luft - Raumfahrt / Satelliten
- Behörden / UN / Weltbank

Wo?

- Europa
- Asien - Indien bis Malaysia

# JavaFX 8 - JumpStart

folgendes erwartet uns:

- Intro
- Hello - World
- SceneBuilder - Demo
- FXML - was ist das denn?
- Swing und JavaFX
- afterburner.fx - Intro
- CDI bootstrap - Intro
- TestFX
- Beispiele
- DrillDownPieChart
- CrossLanguage Injection
- JavaFX on IOT
- JavaFX on iOS
- JavaFX on Android

# JavaFX 8 - JumpStart

## Intro

Swing war gestern.... heute ist JavaFX ;-)

# JavaFX 8 - JumpStart

## Intro

Swing war gestern.... heute ist JavaFX ;-)

Seit JavaFX8...

.. ist JavaFX im JDK enthalten

# JavaFX 8 - JumpStart

## Intro

Swing war gestern.... heute ist JavaFX ;-)

Seit JavaFX8...

.. ist JavaFX im JDK enthalten

.. auch für JDK7 verfügbar, aber muss explizit eingebunden werden.



# JavaFX 8 - JumpStart

## Intro

Swing war gestern.... heute ist JavaFX ;-)

Seit JavaFX8...

.. ist JavaFX im JDK enthalten

.. auch für JDK7 verfügbar, aber muss explizit eingebunden werden.

Beachte:

Der Wechsel von JavaFX 2.2 auf JavaFX8 bedeutet recht viel Arbeit, basierend auf dem was verwendet worden ist.

# JavaFX 8 - JumpStart

## Intro

Swing war gestern.... heute ist JavaFX ;-)

Seit JavaFX8...

.. ist JavaFX im JDK enthalten

.. auch für JDK7 verfügbar, aber muss explizit eingebunden werden.

Beachte:

Der Wechsel von JavaFX 2.2 auf JavaFX8 bedeutet recht viel Arbeit, basierend auf dem was verwendet worden ist.

z.B. Builder sind nun alle deprecated und Signaturen und Konstruktoren haben sich geändert.

# JavaFX 8 - JumpStart

Hello World

Wie gehabt.. auch wir starten damit...

# JavaFX 8 - JumpStart

Hello World

Wie gehabt.. auch wir starten damit...

Aufgabe:

erzeuge einen Button,  
bei Verwendung das Event auf die KommandLine schreiben..

# JavaFX 8 - JumpStart

## Hello World

JAVA

```
public class JavaFXPlain extends Application {
    public static void main(String[] args) { launch(args); }
    @Override
    public void start(Stage primaryStage) {
        final AnchorPane rootNode = new AnchorPane();
        final Button button = new Button();
        button.setText("Ohhh drück mich..");
        button.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) { System.out.println("event = " + event); }
        });
        rootNode.getChildren().add(button);
        final Scene scene = new Scene(rootNode, 320, 240);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
}
```

# JavaFX 8 - JumpStart

Hello World

Application:

Die Hauptklasse einer JavaFX Anwendung erbt von `javafx.application.Application`.

# JavaFX 8 - JumpStart

Hello World

Application:

Die Hauptklasse einer JavaFX Anwendung erbt von `javafx.application.Application`.

Methode: `start`

Die `start()` Methode ist der Haupt-Einstiegspunkt für eine JavaFX Anwendung.

# JavaFX 8 - JumpStart

Hello World

Application:

Die Hauptklasse einer JavaFX Anwendung erbt von `javafx.application.Application`.

Methode: `start`

Die `start()` Methode ist der Haupt-Einstiegspunkt für eine JavaFX Anwendung.

Stage:

Die Instanz der Klasse `Stage` ist der TOP-Level Kontainer für alle JavaFX Komponenten



# JavaFX 8 - JumpStart

## Hello World

Application:

Die Hauptklasse einer JavaFX Anwendung erbt von `javafx.application.Application`.

Methode: `start`

Die `start()` Methode ist der Haupt-Einstiegspunkt für eine JavaFX Anwendung.

Stage:

Die Instanz der Klasse `Stage` ist der TOP-Level Kontainer für die JavaFX Anwendung.

Scene:

Die Instanz der Klasse `Scene` ist der Kontainer für den gesamten Kontent.



# JavaFX 8 - JumpStart

## Hello World

JAVA

```
public class JavaFXPlain extends Application {
    public static void main(String[] args) { launch(args); }
    @Override
    public void start(Stage primaryStage) {
        final AnchorPane rootNode = new AnchorPane();
        final Button button = new Button();
        button.setText("Ohhh drück mich..");
        button.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) { System.out.println("event = " + event); }
        });
        rootNode.getChildren().add(button);
        final Scene scene = new Scene(rootNode, 320, 240);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
}
```

# JavaFX 8 - JumpStart

## Scenebuilder

Der SceneBuilder ist ein Werkzeug von Oracle selbst.

# JavaFX 8 - JumpStart

## Scenebuilder

Der SceneBuilder ist ein Werkzeug von Oracle selbst.

.. ermöglicht die visuelle Gestaltung von GUI Elementen.

# JavaFX 8 - JumpStart

## Scenebuilder

Der SceneBuilder ist ein Werkzeug von Oracle selbst.

.. ermöglicht die visuelle Gestaltung von GUI Elementen.

.. kann jederzeit eine Preview starten (klickbar)

# JavaFX 8 - JumpStart

## Scenebuilder

Der SceneBuilder ist ein Werkzeug von Oracle selbst.

.. ermöglicht die visuelle Gestaltung von GUI Elementen.

.. kann jederzeit eine Preview starten (klickbar)

.. speichert das Ergebnis in einer Datei mit der Endung fxml

# JavaFX 8 - JumpStart

## Scenebuilder

Der SceneBuilder ist ein Werkzeug von Oracle selbst.

.. ermöglicht die visuelle Gestaltung von GUI Elementen.

.. kann jederzeit eine Preview starten (klickbar)

.. speichert das Ergebnis in einer Datei mit der Endung fxml

.. FXML ist die Beschreibung der Anordnungen und

# JavaFX 8 - JumpStart

## Scenebuilder

Der SceneBuilder ist ein Werkzeug von Oracle selbst.

.. ermöglicht die visuelle Gestaltung von GUI Elementen.

.. kann jederzeit eine Preview starten (klickbar)

.. speichert das Ergebnis in einer Datei mit der Endung fxml

.. FXML ist die Beschreibung der Anordnungen und

.. .. CSS Anweisungen (das sollte besser nicht gemacht werden )



# JavaFX 8 - JumpStart

## Scenebuilder

Beginnen wir mit der Applikation....

# JavaFX 8 - JumpStart

## Scenebuilder

Beginnen wir mit der Applikation....

.. diese muss die FXML-Datei laden

# JavaFX 8 - JumpStart

## Scenebuilder

Beginnen wir mit der Applikation....

.. diese muss die FXML-Datei laden

.. das passiert mit dem FXMLLoader

# JavaFX 8 - JumpStart

## Scenebuilder

Beginnen wir mit der Applikation....

.. diese muss die FXML-Datei laden

.. das passiert mit dem FXMLLoader

.. und das Ergebnis ist die Instanz der in der FXML-Datei definierten Root-Komponente

# JavaFX 8 - JumpStart

## Scenbuilder

JAVA

```
public class Main extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        URL resource = getClass().getResource("SceneBuilderDemo.fxml");
        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(resource);
        AnchorPane root = loader.load();
        primaryStage.setScene(new Scene(root, 300, 240));
        primaryStage.show();
    }
    public static void main(String[] args) { launch(args); }
}
```

# JavaFX 8 - JumpStart

## Scenebuilder

Nun sehen wir uns die FXML-Datei an...

# JavaFX 8 - JumpStart

## Scenebuilder

Nun sehen wir uns die FXML-Datei an...

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<AnchorPane maxHeight="240.0" maxWidth="320.0" minHeight="240.0" minWidth="320.0"
    prefHeight="240.0" prefWidth="320.0"
    xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
    fx:controller="org.rapidpm.course.javafx.jumpstart.step002.SceneBuilderDemoController">
    <children>
        <Button fx:id="button" mnemonicParsing="false" onAction="#onButtonClicked" text="Bitte drücke mich..."/>
    </children>
</AnchorPane>
```

# JavaFX 8 - JumpStart

## Scenebuilder

### fx:controller

.. Definition welcher Controller verwendet werden soll



# JavaFX 8 - JumpStart

## Scenebuilder

### fx:controller

- .. Definition welcher Controller verwendet werden soll
- .. Der Controller muss einen Default Constructor haben

# JavaFX 8 - JumpStart

## Scenebuilder

### fx:controller

- .. Definition welcher Controller verwendet werden soll
- .. Der Controller muss einen Default Constructor haben
- .. jede Komponente kann seinen eigenen Controller bekommen (später mehr)

# JavaFX 8 - JumpStart

## Scenebuilder

### fx:controller

- .. Definition welcher Controller verwendet werden soll
- .. Der Controller muss einen Default Constructor haben
- .. jede Komponente kann seinen eigenen Controller bekommen (später mehr)

**onAction="#onButtonClicked"**

# JavaFX 8 - JumpStart

## Scenebuilder

### fx:controller

- .. Definition welcher Controller verwendet werden soll
- .. Der Controller muss einen Default Constructor haben
- .. jede Komponente kann seinen eigenen Controller bekommen (später mehr)

### **onAction="#onButtonClicked"**

- .. Name der Methode die aufgerufen werden soll

# JavaFX 8 - JumpStart

## Scenebuilder

### fx:controller

- .. Definition welcher Controller verwendet werden soll
- .. Der Controller muss einen Default Constructor haben
- .. jede Komponente kann seinen eigenen Controller bekommen (später mehr)

### **onAction="#onButtonClicked"**

- .. Name der Methode die aufgerufen werden soll
- .. beachte # vor dem Namen, sonst kein Matching!

# JavaFX 8 - JumpStart

## Scenebuilder

```
public class SceneBuilderDemoController {  
    public void onButtonClicked(ActionEvent actionEvent){  
        System.out.println("actionEvent = " + actionEvent);  
    }  
}
```

JAVA

# JavaFX 8 - JumpStart

## Scenebuilder

```
public class SceneBuilderDemoController {  
    public void onButtonClicked(ActionEvent actionEvent){  
        System.out.println("actionEvent = " + actionEvent);  
    }  
}
```

JAVA

Der Controller ist selber zu schreiben!

# JavaFX 8 - JumpStart

Scenebuilder

JSR 223-compatible scripting engine



# JavaFX 8 - JumpStart

## Scenebuilder

JSR 223-compatible scripting engine

.. Hiermit können wir die Controller in z.B. folgenden Sprachen schreiben

# JavaFX 8 - JumpStart

## Scenebuilder

JSR 223-compatible scripting engine

.. Hiermit können wir die Controller in z.B. folgenden Sprachen schreiben

.. .. JavaScript, Groovy, Jython, Clojure

# JavaFX 8 - JumpStart

## Scenebuilder

JSR 223-compatible scripting engine

.. Hiermit können wir die Controller in z.B. folgenden Sprachen schreiben

.. .. JavaScript, Groovy, Jython, Clojure

Beispiel anhand von JavaScript

# JavaFX 8 - JumpStart

## Scenebuilder

Beispiel anhand von JavaScript

1) die **fx:controller** Anweisung aus der fxml-Datei entfernen

# JavaFX 8 - JumpStart

## Scenebuilder

Beispiel anhand von JavaScript

1) die **fx:controller** Anweisung aus der fxml-Datei entfernen

2) die verwendete Sprache in der fxml-Datei definieren,  
hier `<?language javascript?>`

# JavaFX 8 - JumpStart

## Scenebuilder

Beispiel anhand von JavaScript

1) die **fx:controller** Anweisung aus der fxml-Datei entfernen

2) die verwendete Sprache in der fxml-Datei definieren,  
hier `<?language javascript?>`

3) Action-Methodennamen ändern (sprachabhängig) ->  
`onAction="onButtonClicked(event);"`

# JavaFX 8 - JumpStart

## Scenebuilder

Beispiel anhand von JavaScript

- 1) die **fx:controller** Anweisung aus der fxml-Datei entfernen
- 2) die verwendete Sprache in der fxml-Datei definieren, hier **<?language javascript?>**
- 3) Action-Methodennamen ändern (sprachabhängig) -> **onAction="onButtonClicked(event);"**
- 4a) Definition der Methode in der Sprache innerhalb der FXML-Datei

# JavaFX 8 - JumpStart

## Scenebuilder

Beispiel anhand von JavaScript

- 1) die **fx:controller** Anweisung aus der fxml-Datei entfernen
- 2) die verwendete Sprache in der fxml-Datei definieren, hier `<?language javascript?>`
- 3) Action-Methodennamen ändern (sprachabhängig) -> **`onAction="onButtonClicked(event);"`**
- 4a) Definition der Methode in der Sprache innerhalb der FXML-Datei
- 4b) Definition der Methode in der Sprache in einer externen datei , incl Referenz



# JavaFX 8 - JumpStart

## Scenbuilder

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<?language javascript?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<AnchorPane maxHeight="240.0" maxWidth="320.0"
    minHeight="240.0" minWidth="320.0" prefHeight="240.0" prefWidth="320.0"
    xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1">
  <fx:script>
    function onClicked() { buttonID.setText("Calling the JavaScript"); }
  </fx:script>
  <!--<fx:script source="fxml_example.js"/>-->
  <children>
    <Button fx:id="buttonID" mnemonicParsing="false" onAction="onButtonClicked(event);"
      text="Bitte drücke mich..."/>
  </children>
</AnchorPane>
```

# JavaFX 8 - JumpStart

## FXML

Mit FXML kann man die Anordnung/Komposition der GUI Elemente und die Logik voneinander trennen

# JavaFX 8 - JumpStart

## FXML

Mit FXML kann man die Anordnung/Komposition der GUI Elemente und die Logik voneinander trennen

Wie sollte ich da vorgehen?

# JavaFX 8 - JumpStart

## FXML

Mit FXML kann man die Anordnung/Komposition der GUI Elemente und die Logik voneinander trennen

Wie sollte ich da vorgehen?

Wie feingranular wird diese Trennung in Komponenten gemacht?

# JavaFX 8 - JumpStart

## FXML

Mit FXML kann man die Anordnung/Komposition der GUI Elemente und die Logik voneinander trennen

Wie sollte ich da vorgehen?

Wie feingranular wird diese Trennung in Komponenten gemacht?

Wann verwende ich FXML, wann besser nicht?

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Bei der Verwendung von FXML gibt es prinzipiell immer 3 Komponenten.

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Bei der Verwendung von FXML gibt es prinzipiell immer 3 Komponenten.

.. fxml Datei: Hier werden die Komponenten positioniert.

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Bei der Verwendung von FXML gibt es prinzipiell immer 3 Komponenten.

.. fxml Datei: Hier werden die Komponenten positioniert.

.. Controller Klasse: Abbildung der Logik bzw Interaktion der GUI Elemente.



# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Bei der Verwendung von FXML gibt es prinzipiell immer 3 Komponenten.

.. fxml Datei: Hier werden die Komponenten positioniert.

.. Controller Klasse: Abbildung der Logik bzw Interaktion der GUI Elemente.

.. CSS: Anpassungen der GUI Elemente ( Farben / Formen / Größen ..) - **noch nicht gezeigt**

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Bei der Verwendung von FXML gibt es prinzipiell immer 3 Komponenten.

.. fxml Datei: Hier werden die Komponenten positioniert.

.. Controller Klasse: Abbildung der Logik bzw Interaktion der GUI Elemente.

.. CSS: Anpassungen der GUI Elemente ( Farben / Formen / Größen ..) - **noch nicht gezeigt**

nette Theorie... Wir wollen ein Beispiel...

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Beginnen wir mit der Hpt-Ansicht: MainPane.fxml

# JavaFX 8 - JumpStart

## FXML - Wie sollte ich da vorgehen?

Beginnen wir mit der Hpt-Ansicht: MainPane.fxml

XML

```
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.Button?>
<?import org.rapidm.course.javafx.jumpstart.step003.sample.main.edit.EditPane?>

<fx:root type="org.rapidm.course.javafx.jumpstart.step003.sample.main.MainPane"
  fx:controller="org.rapidm.course.javafx.jumpstart.step003.sample.main.MainPaneController"
  xmlns:fx="http://javafx.com/fxml" >
  <VBox>
    <Label fx:id="lb" text="Hello World"/>
    <Button fx:id="btn" onAction="#onDoSomething" text="und los geht es"/>
    <EditPane fx:id="editPane"/>
  </VBox>
</fx:root>
```

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Beginnen wir mit der Hpt-Ansicht: MainPane.fxml

.. Hier werden Std.Komponenten und eine eigene Komponente positioniert.

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Beginnen wir mit der Hpt-Ansicht: MainPane.fxml

.. Hier werden Std.Komponenten und eine eigene Komponente positioniert.

.. Der Controller für die Hpt.-Ansicht wird gesetzt.

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Der MainPaneController

```
public class MainPaneController {
    @FXML Button btn;
    @FXML EditPane editPane;
    public MainPaneController() {
        System.out.println("MainPaneController = OK" );
    }
    public void onDoSomething(ActionEvent actionEvent){
        System.out.println("actionEvent = " + actionEvent);
        btn.setText(Instant.now()+"");
        editPane.setLabelText(); //Label v Edit veraendern
    }
}
```

JAVA

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Der MainPaneController

..was bedeuten die Attribute annotiert mit @FXML ?

```
public class MainPaneController {
    @FXML Button btn;
    @FXML EditPane editPane;
    public MainPaneController() {
        System.out.println("MainPaneController = OK" );
    }
    public void onDoSomething(ActionEvent actionEvent){
        System.out.println("actionEvent = " + actionEvent);
        btn.setText(Instant.now()+"");
        editPane.setLabelText(); //Label v Edit veraendern
    }
}
```

JAVA



# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Der MainPaneController

..was bedeuten die Attribute annotiert mit @FXML ?

**BEACHTTE!**  
Hier existiert ein Lebenszyklus.

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Der MainPaneController

..was bedeuten die Attribute annotiert mit @FXML ?

**BEACHTTE!**

Hier existiert ein Lebenszyklus.

Bei der Kombination mit anderen Technologien wird das wichtig !!

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Der MainPaneController

..was bedeuten die Attribute annotiert mit @FXML ?

**BEACHTTE!**

Hier existiert ein Lebenszyklus.

Bei der Kombination mit anderen Technologien wird das wichtig !!

Was fehlt noch ?

# JavaFX 8 - JumpStart

## FXML - Wie sollte ich da vorgehen?

Zur Erinnerung: MainPane.fxml

XML

```
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.Button?>
<?import org.rapidm.course.javafx.jumpstart.step003.sample.main.edit.EditPane?>

<fx:root type="org.rapidm.course.javafx.jumpstart.step003.sample.main.MainPane"
  fx:controller="org.rapidm.course.javafx.jumpstart.step003.sample.main.MainPaneController"
  xmlns:fx="http://javafx.com/fxml" >
  <VBox>
    <Label fx:id="lb" text="Hello World"/>
    <Button fx:id="btn" onAction="#onDoSomething" text="und los geht es"/>
    <EditPane fx:id="editPane"/>
  </VBox>
</fx:root>
```

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Der MainPaneController

..was bedeuten die Attribute annotiert mit @FXML ?

BEACHTEN!

Hier existiert ein Lebenszyklus.

Bei der Kombination mit anderen Technologien wird das wichtig !!

Was fehlt noch ?

Die Definition der eigenen Komponente: **MainPane**

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Die Klasse: MainPane

```
public class MainPane extends AnchorPane {  
    public MainPane(){  
        URL resource = getClass().getResource("MainPane.fxml");  
        FXMLLoader loader = new FXMLLoader(resource);  
        loader.setRoot(this);  
        try {  
            loader.load();  
        } catch (IOException e) { e.printStackTrace(); }  
        AnchorPane.setBottomAnchor(this, 0.0);  
        AnchorPane.setTopAnchor(this, 0.0);  
        AnchorPane.setLeftAnchor(this, 0.0);  
        AnchorPane.setRightAnchor(this, 0.0);  
    }  
}
```

JAVA

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Nun haben wir unsere erste Komponente.

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Nun haben wir unsere erste Komponente.

Wir arbeiten mit Namenskonventionen.



# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Nun haben wir unsere erste Komponente.

Wir arbeiten mit Namenskonventionen.

GUI-Komponente: MainPane

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Nun haben wir unsere erste Komponente.

Wir arbeiten mit Namenskonventionen.

GUI-Komponente: MainPane

FXML-Datei: MainPane.fxml

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Nun haben wir unsere erste Komponente.

Wir arbeiten mit Namenskonventionen.

GUI-Komponente: MainPane

FXML-Datei: MainPane.fxml

Controller: MainPaneController

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Nun haben wir unsere erste Komponente.

Wir arbeiten mit Namenskonventionen.

GUI-Komponente: MainPane

FXML-Datei: MainPane.fxml

Controller: MainPaneController

So werden alle Komponenten aufgebaut!

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Nun haben wir unsere erste Komponente.

Wir arbeiten mit Namenskonventionen.

GUI-Komponente: MainPane

FXML-Datei: MainPane.fxml

Controller: MainPaneController

So werden alle Komponenten aufgebaut!

.. EditPane ..

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

EditPane, besteht aus...

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

EditPane, besteht aus...

GUI-Komponente: EditPane



# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

EditPane, besteht aus...

GUI-Komponente: EditPane

FXML-Datei: EditPane.fxml



# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

EditPane, besteht aus...

GUI-Komponente: EditPane

FXML-Datei: EditPane.fxml

Controller: EditPaneController

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

EditPane, besteht aus...

GUI-Komponente: EditPane

FXML-Datei: EditPane.fxml

Controller: EditPaneController

Aber wo kommt die Geschäftslogik hin?

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

EditPane, besteht aus...

GUI-Komponente: EditPane

FXML-Datei: EditPane.fxml

Controller: EditPaneController

Aber wo kommt die Geschäftslogik hin?

NEIN ! nicht in den Controller ;-)

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Die zu verwendende Geschäftslogik - Klasse Service

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Die zu verwendende Geschäftslogik - Klasse Service

```
public class Service {  
    public String calculate(){  
        return Instant.now().toString();  
    }  
}
```

JAVA

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Die zu verwendende Geschäftslogik - Klasse Service

```
public class Service {  
    public String calculate(){  
        return Instant.now().toString();  
    }  
}
```

JAVA

Hierbei handelt es sich um ein POJO

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Die zu verwendende Geschäftslogik - Klasse Service

```
public class Service {  
    public String calculate(){  
        return Instant.now().toString();  
    }  
}
```

JAVA

Hierbei handelt es sich um ein POJO

Diese soll im Controller verwendet werden...

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

EditController:

```
public class EditPaneController {
    private Service service = new Service();
    public @FXML Button btn;
    public @FXML Label lb;
    public void onDoSomething(ActionEvent actionEvent){
        System.out.println("actionEvent = " + actionEvent);
        btn.setText(service.calculate());
    }
    public void useService(){ //Holder Komponente triggert Kind
        lb.setText(service.calculate());
    }
}
```

JAVA



# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

EditController:

```
public class EditPaneController {  
    private Service service = new Service();  
    //.. snipp  
    public void onDoSomething(ActionEvent actionEvent){  
        btn.setText(service.calculate());  
    }  
    //.. snipp  
}
```

JAVA

Hier wird der Service noch statisch verwendet.. -> CDI

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Refactoring :

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Refactoring :

.. Main bleibt wie gehabt...

# JavaFX 8 - JumpStart

FXML - Wie sollte ich da vorgehen?

Refactoring :

.. Main bleibt wie gehabt...

```
public class Main extends Application {  
    @Override  
    public void start(Stage primaryStage) throws Exception {  
        MainPane root = new MainPane();  
        primaryStage.setTitle("Hello World");  
        primaryStage.setScene(new Scene(root, 300, 275));  
        primaryStage.show();  
    }  
    public static void main(String[] args) { launch(args); }  
}
```

JAVA

# JavaFX 8 - JumpStart

## FXML - Wie sollte ich da vorgehen?

Refactoring :

.. gemeinsame GUI-Teile werden in ein BasePane verschoben..

JAVA

```
public abstract class BasePane<T> extends AnchorPane {
    public T controller;
    private void init() {
        AnchorPane.setBottomAnchor(this, 0.0); AnchorPane.setTopAnchor(this,0.0);
        AnchorPane.setLeftAnchor(this,0.0); AnchorPane.setRightAnchor(this,0.0);
        URL resource = getClass().getResource(getFXMLName()+".fxml");
        FXMLLoader loader = new FXMLLoader(resource);
        loader.setRoot(this);
        try { loader.load(); controller = loader.getController(); }
        catch (IOException e) {e.printStackTrace(); }
    }
    public abstract String getFXMLName();
    public BasePane(Node... children) { super(children); init(); }
    public BasePane() { init(); }
}
```

# JavaFX 8 - JumpStart

## FXML - Wie sollte ich da vorgehen?

Refactoring :

.. gemeinsame GUI-Teile werden in ein BasePane verschoben..

.. dadurch reduzieren sich die GUI-Komponenten..

```
public class MainPane extends BasePane<MainPaneController> {  
    @Override  
    public String getFXMLName() {  
        return MainPane.class.getSimpleName();  
    }  
}
```

JAVA

```
public class EditPane extends BasePane<EditPaneController> {  
    @Override  
    public String getFXMLName() {  
        return EditPane.class.getSimpleName();  
    }  
    public void setLabelText(){ controller.useService(); }  
}
```

JAVA

# JavaFX 8 - JumpStart

## FXML

DONE - Wie sollte ich da vorgehen?

Wie feingranular wird diese Trennung in Komponenten gemacht?

Wann verwende ich FXML, wann besser nicht?

# JavaFX 8 - JumpStart

## FXML

DONE - Wie sollte ich da vorgehen?

Wie feingranular wird diese Trennung in Komponenten gemacht?

.. Alles was zu einem Segment auf dem Screen zusammen gefasst werden kann.



# JavaFX 8 - JumpStart

## FXML

DONE - Wie sollte ich da vorgehen?

Wie feingranular wird diese Trennung in Komponenten gemacht?

.. Alles was zu einem Segment auf dem Screen zusammen gefasst werden kann.

.. Immer wenn es wiederkehrende Teile sind

# JavaFX 8 - JumpStart

## FXML

DONE - Wie sollte ich da vorgehen?

Wie feingranular wird diese Trennung in Komponenten gemacht?

.. Alles was zu einem Segment auf dem Screen zusammen gefasst werden kann.

.. Immer wenn es wiederkehrende Teile sind

Aber !!

# JavaFX 8 - JumpStart

## FXML

DONE - Wie sollte ich da vorgehen?

Wie feingranular wird diese Trennung in Komponenten gemacht?

.. Alles was zu einem Segment auf dem Screen zusammen gefasst werden kann.

.. Immer wenn es wiederkehrende Teile sind

Aber !!

.. Die Verwendung von fxml kostet Zeit und Ressourcen

# JavaFX 8 - JumpStart

## FXML

DONE - Wie sollte ich da vorgehen?

Wie feingranular wird diese Trennung in Komponenten gemacht?

.. Alles was zu einem Segment auf dem Screen zusammen gefasst werden kann.

.. Immer wenn es wiederkehrende Teile sind

Aber !!

.. Die Verwendung von fxml kostet Zeit und Ressourcen

.. FXML - File cachen / nicht immer wieder neu vom Filesystem lesen

# JavaFX 8 - JumpStart

## FXML

DONE - Wie sollte ich da vorgehen?

Wie feingranular wird diese Trennung in Komponenten gemacht?

.. Alles was zu einem Segment auf dem Screen zusammen gefasst werden kann.

.. Immer wenn es wiederkehrende Teile sind

Aber !!

.. Die Verwendung von fxml kostet Zeit und Ressourcen

.. FXML - File cachen / nicht immer wieder neu vom Filesystem lesen

.. FXML - File BITTE ->NICHT<- in einen Persistence-Layer legen!!!!

# JavaFX 8 - JumpStart

## FXML

DONE - Wie sollte ich da vorgehen?

DONE - Wie feingranular wird diese Trennung in Komponenten gemacht?

Wann verwende ich FXML, wann besser nicht?

# JavaFX 8 - JumpStart

## FXML

DONE - Wie sollte ich da vorgehen?

DONE - Wie feingranular wird diese Trennung in Komponenten gemacht?

Wann verwende ich FXML, wann besser nicht?

.. NICHT bei der Erstellung von Komponenten wie Anzeigen/Buttons

# JavaFX 8 - JumpStart

## FXML

DONE - Wie sollte ich da vorgehen?

DONE - Wie feingranular wird diese Trennung in Komponenten gemacht?

Wann verwende ich FXML, wann besser nicht?

.. NICHT bei der Erstellung von Komponenten wie Anzeigen/Buttons

.. sehr gute Komponenten .. siehe ENZO von Gerrit Grunwald

.. Sourcen sind OpenSource -> bitbucket



# JavaFX 8 - JumpStart

## FXML

DONE - Wie sollte ich da vorgehen?

DONE - Wie feingranular wird diese Trennung in Komponenten gemacht?

Wann verwende ich FXML, wann besser nicht?

.. NICHT bei der Erstellung von Komponenten wie Anzeigen/Buttons

.. sehr gute Komponenten .. siehe ENZO von Gerrit Grunwald

.. Sourcen sind OpenSource -> bitbucket

.. JA bei ScreenElementen die aus n Komponenten bestehen und eine Iteraktion untereinander haben

# JavaFX 8 - JumpStart

## FXML

DONE - Wie sollte ich da vorgehen?

DONE - Wie feingranular wird diese Trennung in Komponenten gemacht?

DONE - Wann verwende ich FXML, wann besser nicht?

codecentric AG

# Pause...

Kaffee ich brauche !

Dauer: 15min



# JavaFX 8 - JumpStart

## Swing und JavaFX

Bei Alt-Projekten stellt sich die Frage....

... wie wird migriert?

# JavaFX 8 - JumpStart

## Swing und JavaFX

Bei Alt-Projekten stellt sich die Frage....

... wie wird migriert?

Die Lösung heist : JFXPanel

# JavaFX 8 - JumpStart

## Swing und JavaFX

Bei Alt-Projekten stellt sich die Frage....

... wie wird migriert?

Die Lösung heist : JFXPanel

.. in der Swing Anwendung wird ein JFXPanel eingefügt und dort die jeweilige JavaFX Komponente

# JavaFX 8 - JumpStart

## Swing und JavaFX

Bei Alt-Projekten stellt sich die Frage....

... wie wird migriert?

Die Lösung heist : JFXPanel

.. in der Swing Anwendung wird ein JFXPanel eingefügt und dort die jeweilige JavaFX Komponente

Problem: Die Kommunikation zwischen den GUI Threads

# JavaFX 8 - JumpStart

## Swing und JavaFX

Bei Alt-Projekten stellt sich die Frage....

... wie wird migriert?

Die Lösung heist : JFXPanel

.. in der Swing Anwendung wird ein JFXPanel eingefügt und dort die jeweilige JavaFX Komponente

Problem: Die Kommunikation zwischen den GUI Threads

.. es wird hier ein Kommunikationsmedium benötigt.



# JavaFX 8 - JumpStart

## Swing und JavaFX

Bei Alt-Projekten stellt sich die Frage....

... wie wird migriert?

Die Lösung heist : JFXPanel

.. in der Swing Anwendung wird ein JFXPanel eingefügt und dort die jeweilige JavaFX Komponente

Problem: Die Kommunikation zwischen den GUI Threads

.. es wird hier ein Kommunikationsmedium benötigt.

.. CDI hat sich hier sehr gut bewährt.

# JavaFX 8 - JumpStart

## Swing und JavaFX

Bei Alt-Projekten stellt sich die Frage....

... wie wird migriert?

Die Lösung heist : JFXPanel

.. in der Swing Anwendung wird ein JFXPanel eingefügt und dort die jeweilige JavaFX Komponente

Problem: Die Kommunikation zwischen den GUI Threads

.. es wird hier ein Kommunikationsmedium benötigt.

.. CDI hat sich hier sehr gut bewährt. -> CDI bootstrap

# JavaFX 8 - JumpStart

## Swing und JavaFX

Problem: Die Kommunikation zwischen den GUI Threads

.. es wird hier ein Kommunikationsmedium benötigt.

.. CDI hat sich hier sehr gut bewährt. -> CDI bootstrap

.. Spring wird gerade getestet -> Spring bootstrap

# JavaFX 8 - JumpStart

## Swing und JavaFX

Problem: Die Kommunikation zwischen den GUI Threads

.. es wird hier ein Kommunikationsmedium benötigt.

.. CDI hat sich hier sehr gut bewährt. -> CDI bootstrap

.. Spring wird gerade getestet -> Spring bootstrap

.. afterburner.fx hilft dabei leider nicht

# JavaFX 8 - JumpStart

## TestFX

Bei TestFX handelt es sich um ein TestFramework für JavaFX Komponenten

.. Es ist OpenSource

.. basiert auf JUnit

# JavaFX 8 - JumpStart

## TestFX

Bei TestFX handelt es sich um ein TestFramework für JavaFX Komponenten

.. Es ist OpenSource

.. basiert auf JUnit

**JAVA**

```
public class TextInputControlsTest extends GuiTest {
    public static final String TEXT_FIELD = ".text-field";
    @Override
    protected Parent getRootNode() { return new VBox(new TextField()); }
    @Test
    public void shouldClearText() {
        clickOn(TEXT_FIELD).type("Some text");
        verifyThat(TEXT_FIELD, hasText("Some text"));
        push(TAB); // To change focus from the TextField.
        clearTextIn(TEXT_FIELD);
        verifyThat(TEXT_FIELD, hasText(""));
    }
}
```

# JavaFX 8 - JumpStart

## TestFX

Bei TestFX handelt es sich um ein TestFramework für JavaFX Komponenten

.. Es ist OpenSource

.. basiert auf JUnit

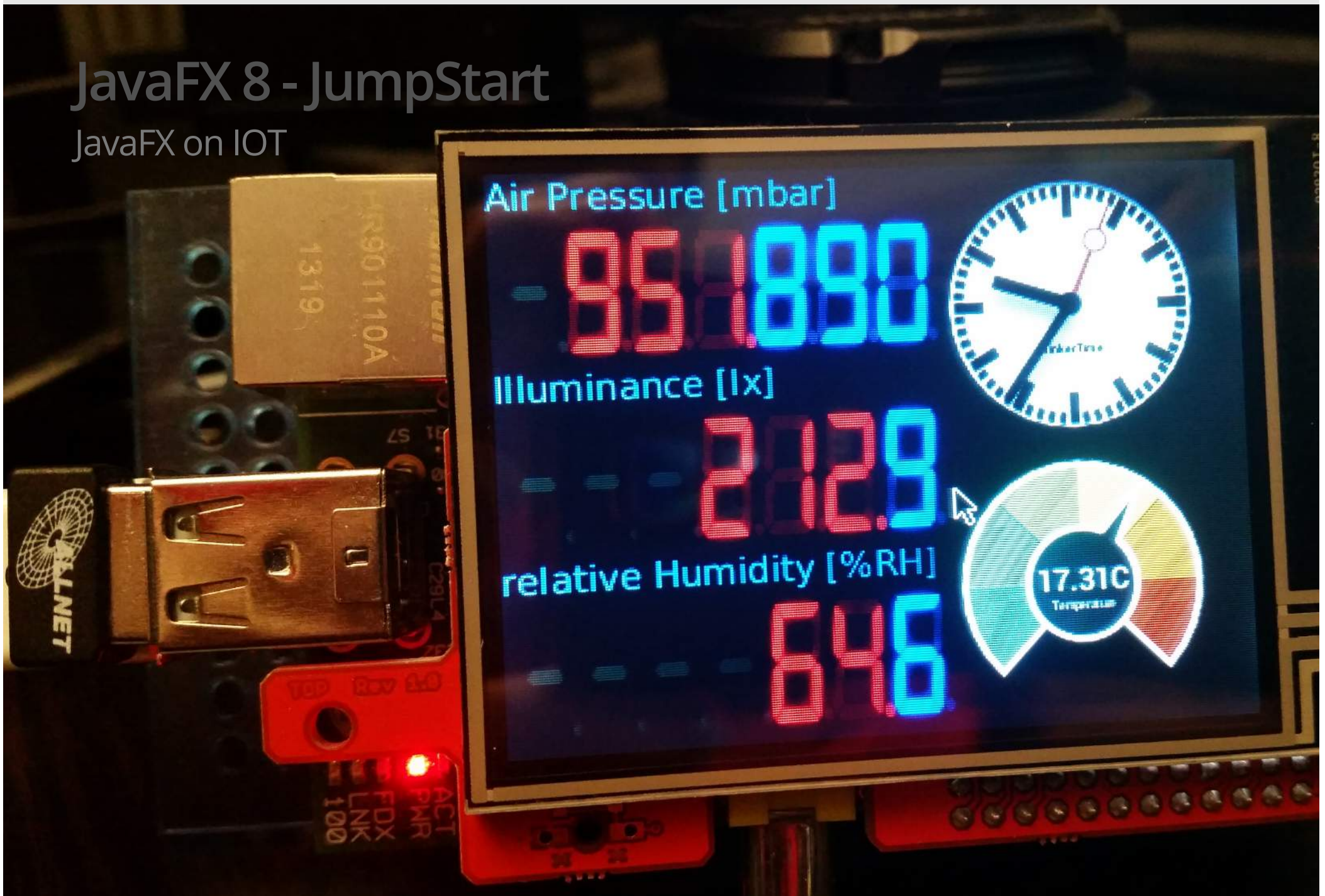
JAVA

```
public class TextInputControlsTest extends GuiTest {
    public static final String TEXT_FIELD = ".text-field";
    @Override
    protected Parent getRootNode() { return new VBox(new TextField()); }
    @Test
    public void shouldClearText() {
        clickOn(TEXT_FIELD).type("Some text");
        verifyThat(TEXT_FIELD, hasText("Some text"));
        push(TAB); // To change focus from the TextField.
        clearTextIn(TEXT_FIELD);
        verifyThat(TEXT_FIELD, hasText(""));
    }
}
```

Aber sehen wir uns das doch einfach mal an ;-)

# JavaFX 8 - JumpStart

JavaFX on IOT





# JavaFX 8 - JumpStart

JavaFX on ...

iOS - RoboVM - JavaFX nativ unter iOS

# JavaFX 8 - JumpStart

JavaFX on ...

iOS - RoboVM - JavaFX nativ unter iOS

Android - <https://bitbucket.org/javafxports/android/>

# Book about Reflection

my book with Dr, Heinz Kabutz

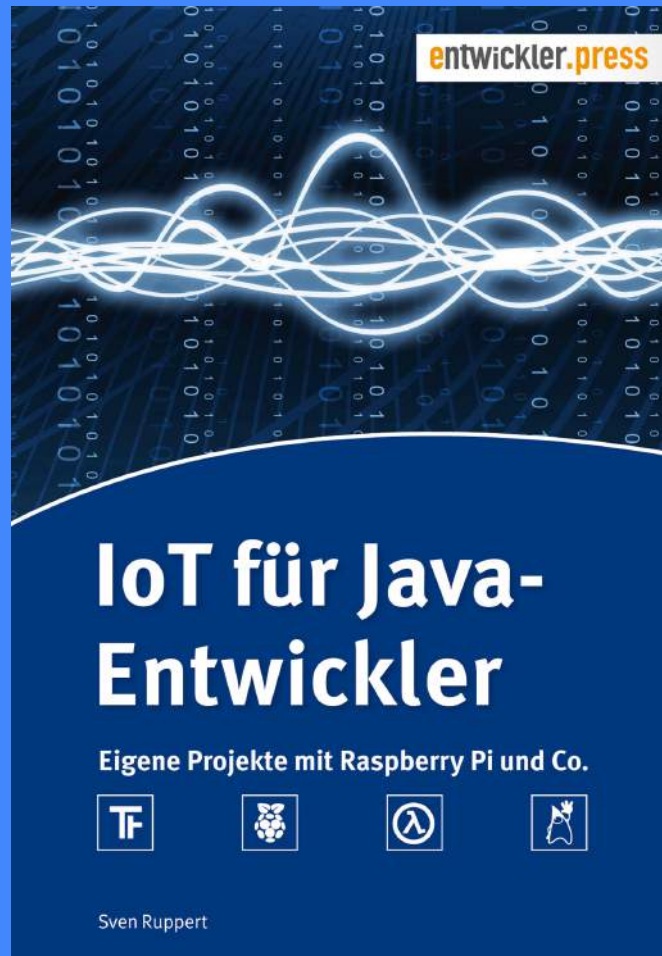
coming soon

JavaFX , CDI und Reflection... ein teuflisches Paar

z.B. ...dynamic static proxies... ;-)

# IoT für Java-Entwickler

coming soon





2015 - erstmalig in der Schweiz ???

Wer hat Lust mit zu helfen?? Bei Interesse bitte bei mir melden.

- Twitter: @SvenRuppert
- mail: sven.ruppert@gmail.com

# Deutschland und Schweiz

JavaFX und Java8 Kickstarter (auch inhouse) -> 2 Tage

<http://www.codecentric.de/java8>

- JavaFX
- Lambdas
- Default Methoden
- Functional - Interfaces
- Date/Time API
- Streams API
- Concurrency mit Streams und StampedLock
- Auswirkungen auf API Design/Evolution
- pre JDK8 Designpatterns migrieren auf JDK8 Pattern

# <Thank You!>



g+ [www.google.com/+SvenRuppert](http://www.google.com/+SvenRuppert)

twitter [@SvenRuppert](https://twitter.com/SvenRuppert)

www [www.codecentric.de](http://www.codecentric.de)

github [github.com/svenruppert](https://github.com/svenruppert)

