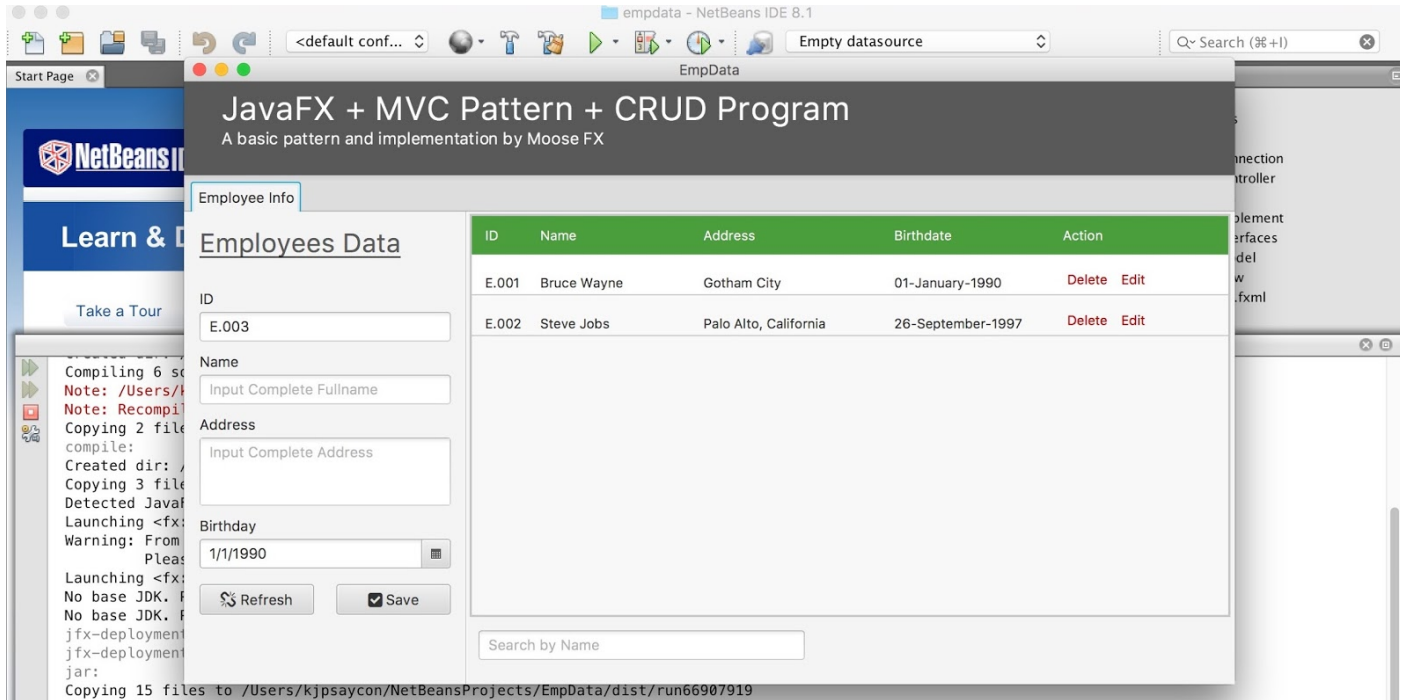


JAVAFX MVC PATTERN + CRUD



This time I will giving a tutorial how to CRUD (create, read, update, delete) in JavaFX. using also a pattern to this will give you insight right in the code. The demo might have been a little old or rusty, but according to many Java programmers ***including me*** this would be a fresh start with your advancement. Please be ready and for the preparation:

1. Netbeans 8.1
2. Java JDK 1.8 and JRE 8
3. XAMPP (MAC and WIN) *Lets use MYSQL for database for now
4. Scene Builder 2.0
5. Great understanding about MVC pattern
6. Great understanding about Interfaces, Implementation and CSS.
7. A cup of hot Green Tea, Jollibee Pies-to-Go, and a 24K Magic by Bruno Mars 🎵

Create a Database and Table

For Database:

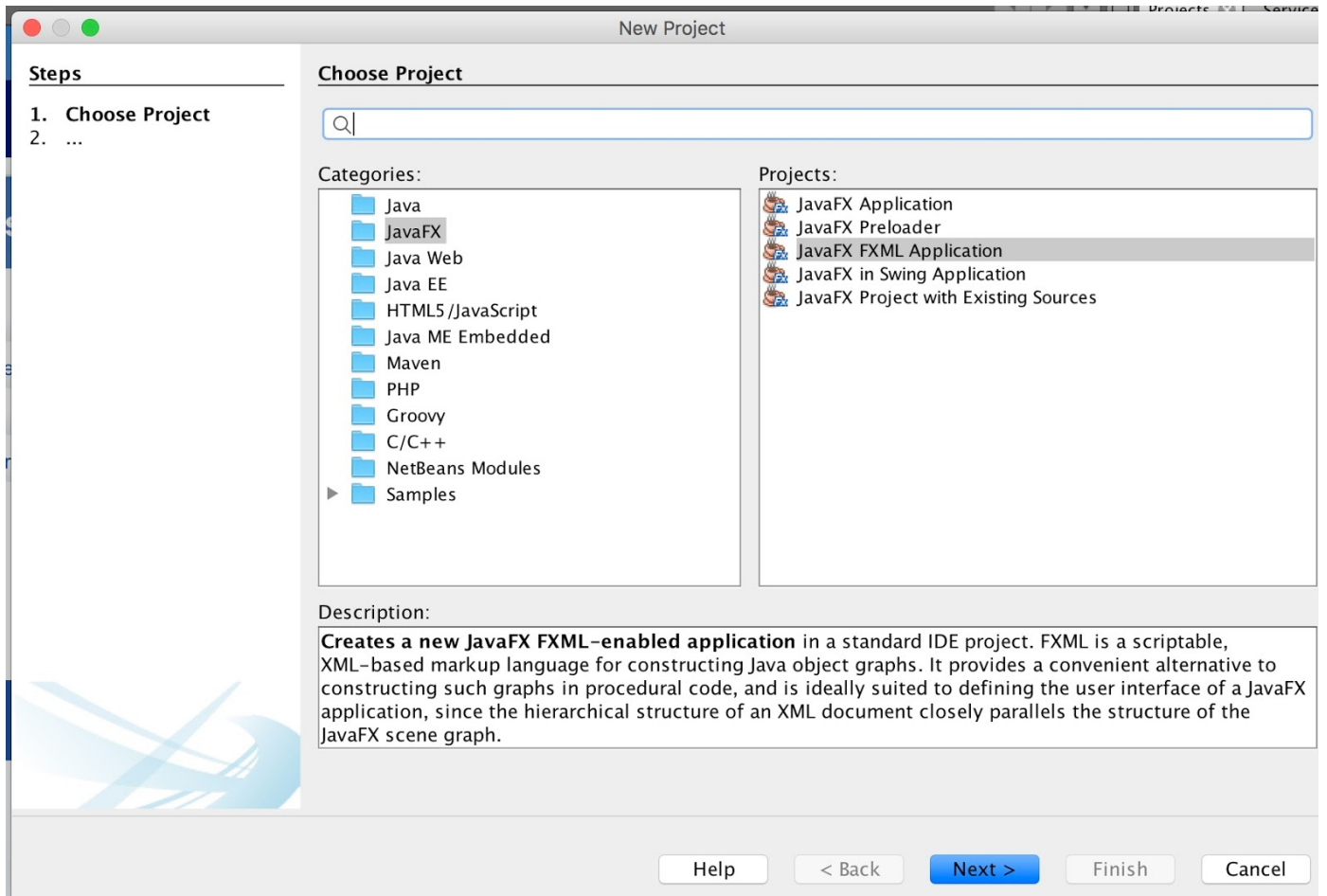
```
CREATE DATABASE `empdb` /*!40100 COLLATE 'latin1_swedish_ci' */;
```

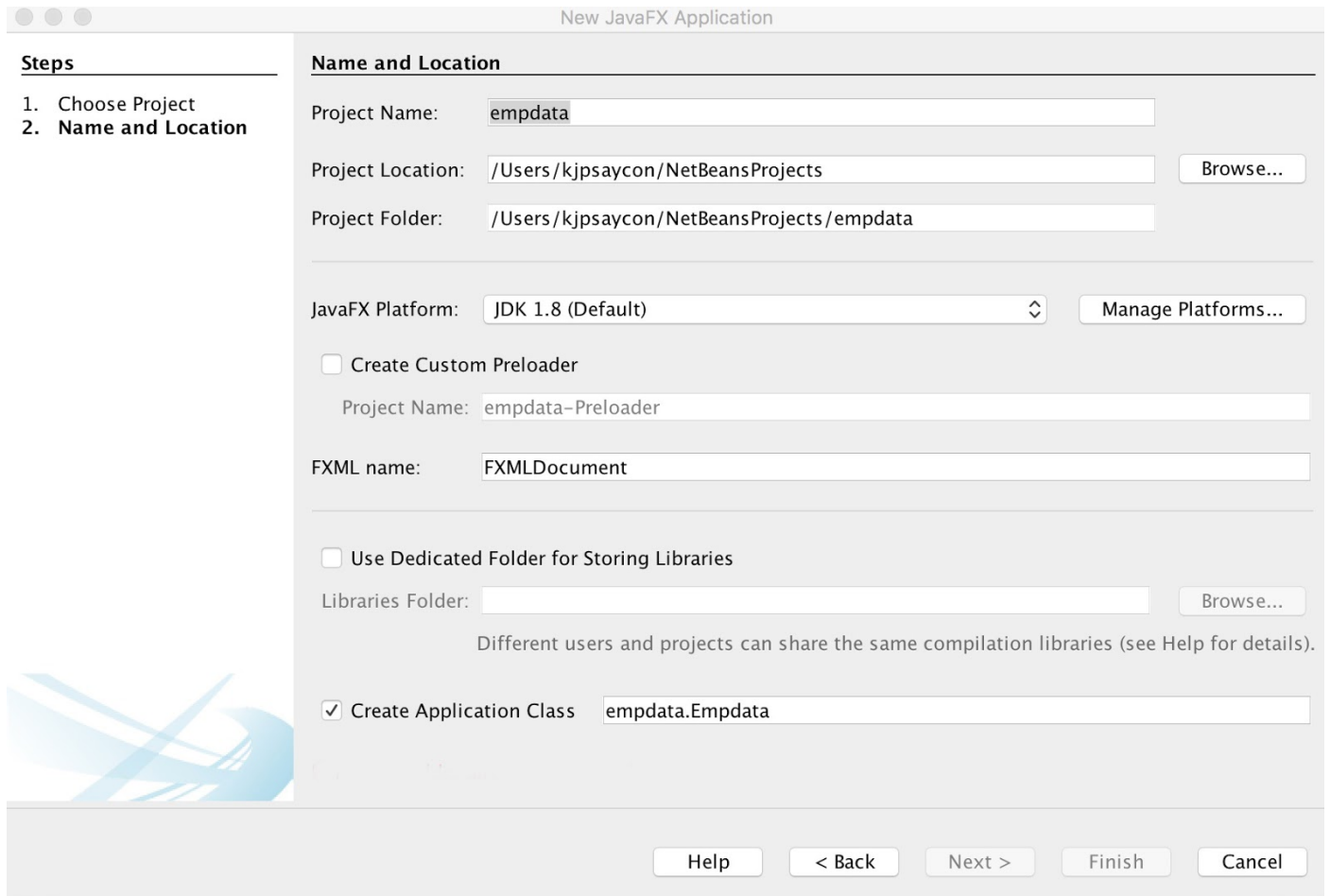
For Table:

```
CREATE TABLE `emptable` (  
  `id` varchar(5) NOT NULL,  
  `name` varchar(50) DEFAULT NULL,  
  `address` varchar(50) DEFAULT NULL,  
  `birthdate` date DEFAULT NULL,
```

```
PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Go to Netbeans and Create a Project:





The image shows a 'New JavaFX Application' dialog box. On the left, a 'Steps' sidebar lists '1. Choose Project' and '2. Name and Location', with the second step being active. The main area is titled 'Name and Location' and contains several input fields and checkboxes. The 'Project Name' is 'empdata', 'Project Location' is '/Users/kjpsaycon/NetBeansProjects', and 'Project Folder' is '/Users/kjpsaycon/NetBeansProjects/empdata'. The 'JavaFX Platform' is set to 'JDK 1.8 (Default)'. There is an unchecked checkbox for 'Create Custom Preloader' with a sub-field for 'Project Name' set to 'empdata-Preloader'. The 'FXML name' is 'FXMLDocument'. Another unchecked checkbox is for 'Use Dedicated Folder for Storing Libraries' with a 'Libraries Folder' field. A checked checkbox is for 'Create Application Class' with a field containing 'empdata.Empdata'. At the bottom are buttons for 'Help', '< Back', 'Next >', 'Finish', and 'Cancel'.

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

JavaFX Platform:

☐ Create Custom Preloader

Project Name:

FXML name:

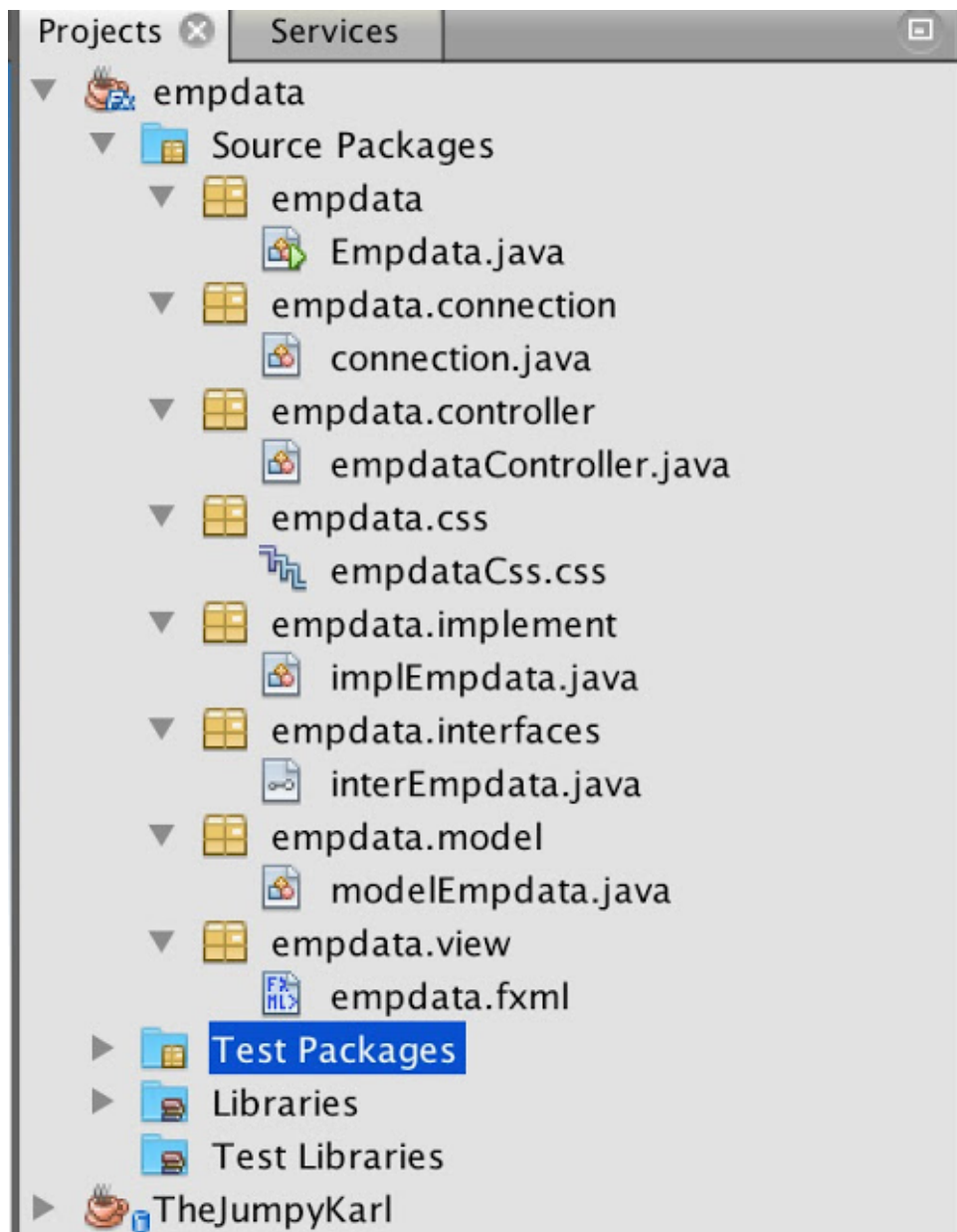
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Application Class

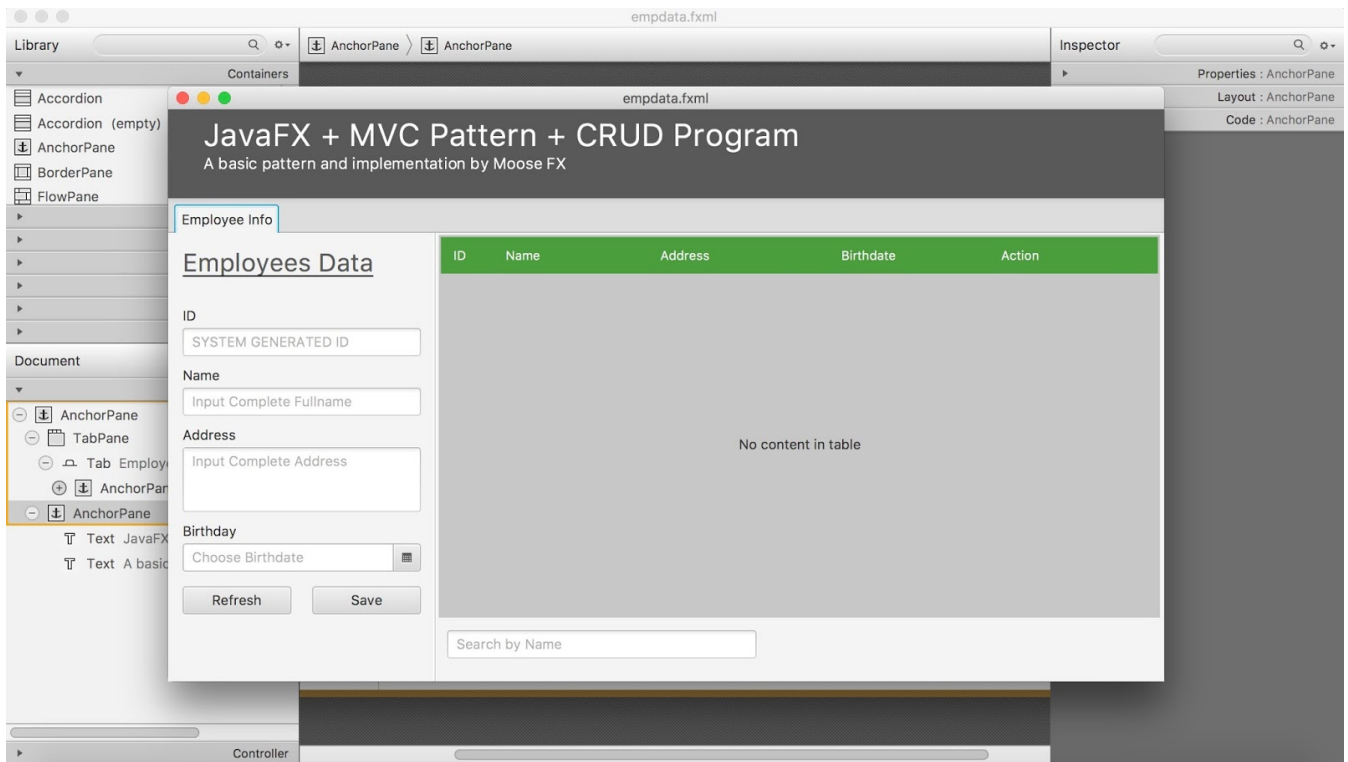
Create a Package just like this one or create your own way to organize the Package (*NOTE Java Ruling 101: You cannot put a package inside a package.)



Design View as shown below using Scene Builder 2.0

to edit double click the empdata.fxml on the empdata.view package

Do not forget after design set the properties fx: id and also to the controller class.



connection.java

```
package empdata.connection;
```

```
import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;
```

```
import java.sql.Connection;
```

```
import java.sql.SQLException;
```

```
import javax.swing.*.*;
```

```
/**
```

```
*
```

```
* @author kjpsaycon
```

```
*/
```

```
public class connection {
    private Connection con;
```

```
    public connection(){
    }
}
```

```
    public Connection connect(){
        if(con == null){
```

```

        MySQLDataSource db = new MySQLDataSource();
        db.setDatabaseName("empdb");
        db.setUser("root");
        db.setPassword("");
        try {
            con = db.getConnection();
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null,"Error","SQL Error:
"+e,JOptionPane.ERROR_MESSAGE);
        }
    }
    return con;
}
}

```

modelEmpdata.java

```

package empdata.model;

import java.text.SimpleDateFormat;
import java.util.Date;
import javafx.beans.property.ObjectProperty;
import javafx.beans.property.SimpleObjectProperty;
import javafx.beans.property.SimpleStringProperty;
import javafx.beans.property.StringProperty;

/**
 *
 * @author kjpsaycon
 */
public class modelEmpdata {
    private final StringProperty id = new SimpleStringProperty();
    private final StringProperty name = new SimpleStringProperty();
    private final StringProperty address = new SimpleStringProperty();
    private final ObjectProperty<Date> birthDate = new SimpleObjectProperty<>();
}

```

```
private String dateFormat;
```

```
public modelEmpdata() {  
}
```

```
public String getId() {  
    return id.get();  
}
```

```
public void setId(String value) {  
    id.set(value);  
}
```

```
public StringProperty idProperty() {  
    return id;  
}
```

```
public String getName() {  
    return name.get();  
}
```

```
public void setName(String value) {  
    name.set(value);  
}
```

```
public StringProperty nameProperty() {  
    return name;  
}
```

```
public String getAddress() {  
    return address.get();  
}
```

```
public void setAddress(String value) {  
    address.set(value);  
}
```

```
}

public StringProperty addressProperty() {
    return address;
}

public Date getBirthDate() {
    return birthDate.get();
}

public void setBirthDate(Date value) {
    birthDate.set(value);
}

public ObjectProperty BirthDateProperty() {
    return birthDate;
}

public String getDateFormat() {
    Date date = getBirthDate();
    SimpleDateFormat df = new SimpleDateFormat("dd-MMMM-yyyy");
    String format = df.format(date);
    return format;
}

public void setDateFormat(String dateFormat) {
    this.dateFormat = dateFormat;
}

}
```

interEmpdata.java

```
package empdata.interfaces;
```

```
import empdata.model.modelEmpdata;  
import javafx.collections.ObservableList;
```

```
/**
```

```
 *
```

```
 * @author kjpsaycon
```

```
 */
```

```
public interface interEmpdata {  
    ObservableList<modelEmpdata> getAll();  
    ObservableList<modelEmpdata> likeByName(String a);  
    void autold(modelEmpdata m);  
    void insert(modelEmpdata m);  
    void delete(modelEmpdata m);  
    void update(modelEmpdata m);  
}
```

```
implEmpdata.java
```

```
-----  
package empdata.implement;
```

```
import empdata.connection.connection;  
import empdata.model.modelEmpdata;  
import java.sql.Date;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import empdata.interfaces.interEmpdata;
```

```

public class implEmpdata implements interEmpdata {
    connection k;

    @Override
    public void insert(modelEmpdata m) {
        k = new connection();
        PreparedStatement ps;
        try {
            ps = k.connect().prepareStatement("insert into tablebiodata values(?,?,?,?)");
            ps.setString(1, m.getId());
            ps.setString(2, m.getName());
            ps.setString(3, m.getAddress());
            ps.setDate(4, (Date) m.getBirthDate());
            ps.execute();
        } catch (Exception e) {
            Logger.getLogger(implEmpdata.class.getName()).log(Level.SEVERE, null, e);
        }
    }
}

```

```

    @Override
    public void delete(modelEmpdata m) {
        k = new connection();
        PreparedStatement ps;
        try {
            ps = k.connect().prepareStatement("delete from tablebiodata where id = ?");

            ps.setString(1, m.getId());
            ps.execute();
        } catch (Exception e) {
            Logger.getLogger(implEmpdata.class.getName()).log(Level.SEVERE, null, e);
        }
    }
}

```

```

    @Override
    public void update(modelEmpdata m) {
        k = new connection();
    }
}

```

```

PreparedStatement ps;
try {
    ps = k.connect().prepareStatement("update tablebiodata set name=?, address=?,
birthdate=? where id = ?");
    ps.setString(4, m.getId());
    ps.setString(1, m.getName());
    ps.setString(2, m.getAddress());
    ps.setDate(3, (Date) m.getBirthDate());
    ps.execute();
} catch (Exception e) {
    Logger.getLogger(implEmpdata.class.getName()).log(Level.SEVERE, null, e);
}
}

```

@Override

```

public ObservableList<modelEmpdata> getAll() {
    k = new connection();
    ObservableList<modelEmpdata> listData = FXCollections.observableArrayList();
    try {
        String sql = "select * from tablebiodata";
        ResultSet rs = k.connect().createStatement().executeQuery(sql);
        while (rs.next()) {
            modelEmpdata m = new modelEmpdata();
            m.setId(rs.getString(1));
            m.setName(rs.getString(2));
            m.setAddress(rs.getString(3));
            m.setBirthDate(rs.getDate(4));
            listData.add(m);
        }
    } catch (Exception ex) {
        Logger.getLogger(implEmpdata.class.getName()).log(Level.SEVERE, null, ex);
    }
    return listData;
}

```

@Override

```
public ObservableList<modelEmpdata> likeByName(String a) {
    k = new connection();
    ObservableList<modelEmpdata> listData = FXCollections.observableArrayList();
    try {
        String sql = "select * from tablebiodata where name like '%" + a + "%'";
        ResultSet rs = k.connect().createStatement().executeQuery(sql);
        while (rs.next()) {
            modelEmpdata m = new modelEmpdata();
            m.setld(rs.getString(1));
            m.setName(rs.getString(2));
            m.setAddress(rs.getString(3));
            m.setBirthDate(rs.getDate(4));
            listData.add(m);
        }
    } catch (Exception ex) {
        Logger.getLogger(implEmpdata.class.getName()).log(Level.SEVERE, null, ex);
    }
    return listData;
}
```

@Override

```
public void autold(modelEmpdata m) {
    k = new connection();
    try {
        ResultSet rs = k.connect().createStatement().executeQuery("select * from
tablebiodata");
        while(rs.next()){
            String code = rs.getString(1).substring(2);
            String auto = ""+(Integer.parseInt(code)+1);
            String nulled = "";
            if (auto.length()==1) {
                nulled = "00";
            }else if (auto.length()==2) {
                nulled = "0";
            }else if (auto.length()==3) {
```

```

        nulled = "";
    }
    m.setId("E."+nulled+auto);
}
if (m.getId()==null) {
    m.setId("E.001");
}
} catch (SQLException ex) {
    Logger.getLogger(implEmpdata.class.getName()).log(Level.SEVERE, null, ex);
}
}
}

```

empdataController.java

```

package empdata.controller;

import empdata.implement.implEmpdata;
import empdata.model.modelEmpdata;
import de.jensd.fx.fontawesome.AwesomeDude;
import de.jensd.fx.fontawesome.AwesomeIcon;
import java.io.IOException;
import java.net.URL;
import java.sql.Date;
import java.time.LocalDate;
import java.util.ResourceBundle;
import javafx.beans.property.SimpleBooleanProperty;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.control.Alert;

```

```
import javafx.scene.control.Button;
import javafx.scene.control.DatePicker;
import javafx.scene.control.Hyperlink;
import javafx.scene.control.Tab;
import javafx.scene.control.TableCell;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.HBox;
import javafx.stage.StageStyle;
import javafx.util.Callback;
import empdata.interfaces.interEmpdata;
```

```
/**
 * FXML Controller class
 *
 * @author kjpsaycon
 */
```

```
//controller empdata
public class empdataController implements Initializable {
    @FXML
    private Tab tabBirthChart,tabBiodata;
    @FXML
    private TextField txtId;
    @FXML
    private TextField txtName;
    @FXML
    private TextArea txtAddress;
    @FXML
    private DatePicker cmbDate;
```

```

@FXML
private Button btnSave;

@FXML
private TableView<modelEmpdata> tableData;

@FXML
private TableColumn<modelEmpdata, String> colId;

@FXML
private TableColumn<modelEmpdata, String> colName;

@FXML
private TableColumn<modelEmpdata, String> colAddress;

@FXML
private TableColumn<modelEmpdata, String> colDate;

@FXML
private TableColumn colAction;

@FXML
private TextField txtSearch;

@FXML
private Button btnRefresh;

@FXML
private AnchorPane paneLoadGraph;
interEmpdata crudData = new implEmpdata();
ObservableList<modelEmpdata> listData;
private String statusCode,statusClick;
ObservableList<modelEmpdata> listDelete;
/**
 * Initializes the controller class.
 * @param url
 * @param rb
 */
@Override
public void initialize(URL url, ResourceBundle rb) {
    colId.setCellValueFactory((TableColumn.CellDataFeatures<modelEmpdata, String>
cellData) ->
        cellData.getValue().idProperty());
    colName.setCellValueFactory((TableColumn.CellDataFeatures<modelEmpdata,
String> cellData) ->

```

```

        cellData.getValue().nameProperty());
        colAddress.setCellValueFactory((TableColumn.CellDataFeatures<modelEmpdata,
String> cellData) ->
            cellData.getValue().addressProperty());
        colDate.setCellValueFactory(new PropertyValueFactory("dateFormat"));
        colAction.setCellValueFactory(new Callback<TableColumn.CellDataFeatures<Object,
Boolean>,
            ObservableValue<Boolean>>() {
                @Override
                public ObservableValue<Boolean> call(TableColumn.CellDataFeatures<Object,
Boolean> p) {
                    return new SimpleBooleanProperty(p.getValue() != null);
                }
            });

        colAction.setCellFactory(new Callback<TableColumn<Object, Boolean>,
TableCell<Object, Boolean>>() {
            @Override
            public TableCell<Object, Boolean> call(TableColumn<Object, Boolean> p) {
                return new ButtonCell(tableData);
            }
        });
        listData = FXCollections.observableArrayList();
        AwesomeDude.setIcon(btnSave, AwesomeIcon.CHECK_SQUARE, "15px");
        AwesomeDude.setIcon(btnRefresh, AwesomeIcon.CHAIN_BROKEN, "15px");
        cmbDate.setValue(LocalDate.of(1990, 01, 01));
        statusCode = "0";
        statusClick = "0";
        showData();
        autold();
        tableData.getSelectionModel().clearSelection();

    }

```



```
private void dialog(Alert.AlertType alertType,String s){
    Alert alert = new Alert(alertType,s);
    alert.initStyle(StageStyle.UTILITY);
    alert.setTitle("Info");
    alert.showAndWait();
}
```

```
private void clear(){
    txtId.clear();
    txtName.clear();
    txtAddress.clear();
    txtSearch.clear();
    cmbDate.setValue(LocalDate.of(1990, 01, 01));
    statusCode = "0";
}
```

```
private void showData(){
    listData = crudData.getAll();
    tableData.setItems(listData);
}
```

```
private void autold(){
    modelEmpdata m = new modelEmpdata();
    crudData.autold(m);
    txtId.setText(m.getId());
}
```

@FXML

```
private void actionSave(ActionEvent event) {
    modelEmpdata m = new modelEmpdata();
    m.setId(txtId.getText());
    m.setName(txtName.getText());
    m.setAddress(txtAddress.getText());
    m.setBirthDate(Date.valueOf(cmbDate.getValue()));
    if (statusCode.equals("0")) {
        crudData.insert(m);
    }
}
```

```

    }else{
        crudData.update(m);
    }
    dialog(Alert.AlertType.INFORMATION, "Data has saved");
    showData();
    clear();
    autold();

}

```

@FXML

```

private void tableDataClick(MouseEvent event) {
    if (statusClick.equals("1")) {
        statusCode = "1";
        try {
            modelEmpdata click = tableData.getSelectionModel().getSelectedItem().get(0);
            txtId.setText(click.getId());
            txtName.setText(click.getName());
            txtAddress.setText(click.getAddress());
            cmbDate.setValue(LocalDate.parse(click.getBirthDate().toString()));
        } catch (Exception e) {
        }
    }
}

```

@FXML

```

private void actionSearch(KeyEvent event) {
    listData = crudData.likeByName(txtSearch.getText());
    tableData.setItems(listData);
}

```

@FXML

```

private void actionRefresh(ActionEvent event) {
    clear();
    showData();
}

```

```

        autold();
    }

private class ButtonCell extends TableCell<Object, Boolean> {
    final Hyperlink cellButtonDelete = new Hyperlink("Delete");
    final Hyperlink cellButtonEdit = new Hyperlink("Edit");
    final HBox hb = new HBox(cellButtonDelete, cellButtonEdit);
    ButtonCell(final TableView tblView){
        hb.setSpacing(4);

        //cell delete
        cellButtonDelete.setOnAction((ActionEvent t) -> {
            statusClick = "1";
            int row = getTableRow().getIndex();
            tableData.getSelectionModel().select(row);
            tableDataClick(null);
            modelEmpdata m = new modelEmpdata();
            m.setld(txtld.getText());
            crudData.delete(m);
            showData();
            clear();
            autold();
            dialog(Alert.AlertType.INFORMATION, "Data has successfully removed");
            statusClick = "0";
            statusCode = "0";
        });

        //cell edit
        cellButtonEdit.setOnAction((ActionEvent event) -> {
            statusClick = "1";
            int row = getTableRow().getIndex();
            tableData.getSelectionModel().select(row);
            tableDataClick(null);
            statusClick = "0";
        });
    }
}

```

```

@Override
protected void updateItem(Boolean t, boolean empty) {
    super.updateItem(t, empty);
    if(!empty){
        setGraphic(hb);
    }else{
        setGraphic(null);
    }
}
}
}
}
}

```

empdataCSS.css

```

-----
/*
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
*/
/*
    Author    : kjpsaycon
*/

 hyperlink{
    -fx-text-fill: #cb0006;
}

 headerPane{
    -fx-background-color: derive(grey, -30%);
}

 table-view {
    -fx-base: white;
    -fx-control-inner-background: white;

```

```
-fx-background-color: #c7c7c7;
-fx-table-cell-border-color: transparent;
-fx-table-header-border-color: transparent;
-fx-padding: 2;
}

.table-view .column-header-background {
    -fx-background-color: #4f9d42;
}

.table-view .column-header, .table-view .filler {
    -fx-size: 35;
    -fx-border-width: 0 0 0 0;
    -fx-background-color: transparent;
    -fx-border-color:
        transparent
        transparent
        derive(-fx-base, 80%)
        transparent;
    -fx-border-insets: 0 10 1 0;
}

.table-view .column-header .label {
    -fx-font-size: 12px;
    -fx-font-family: "Segoe UI bold";
    -fx-text-fill: white;
    -fx-alignment: center-left;
}

.table-view:focused .table-row-cell:filled:focused:selected {
    -fx-background-color: -fx-focus-color;
}

.table-cell {
    -fx-cell-size: 4.0em;
    -fx-padding: 1em 0em 0.1em 0.1em;
```

```
-fx-font: 12px "Segoe UI";  
-fx-alignment: bottom-left;  
}
```

```
.table-row-cell:empty {  
    -fx-background-color: #f8f8f8;  
    -fx-base: transparent;  
    -fx-control-inner-background: transparent;  
    -fx-table-cell-border-color: transparent;  
    -fx-table-header-border-color: transparent;  
    -fx-padding: 0;  
}
```

```
.table-row-cell:empty .table-cell {  
    -fx-border-width: 0px;
```

```
}Empdata.java
```

```
package empdata;
```

```
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.stage.Stage;
```

```
/**  
 *  
 * @author kjpsaycon  
 */
```

```
//main class javafx
```

```
public class Empdata extends Application {
```

```
    @Override
```

```
    public void start(Stage stage) throws Exception {
```

```
        Parent root =
```

```
FXMLLoader.load(getClass().getResource("/empdata/view/empdata.fxml"));
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("EmpData");
    stage.show();
}

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    launch(args);
}

}
```

I think it is a clear tutorial to all the source code has been inserted above. Regarding scene on the design builder please analyze and understand well, or if necessary, ask to google and can also ask in the comments. Other tutorial for introduction to JavaFX editing and coding for Style Builder, connecting FXML to the Controller Class and many more stay tune to this blog! Happy Magic 24KCoding. - [Karl Saycon](#)