

```
/home/manu/CloudStation/ProjetsJAVA/MonnaieFXML/src/service/ServiceMonétaire.java
```

```
package service;

import java.io.*;
import java.net.*;
import java.util.*;

public class ServiceMonétaire {
    private int port;
    private final double TAUX = 6.55957;

    public static void main(String[] args) { new ServiceMonétaire(); }

    private ServiceMonétaire() {
        try {
            lireConfiguration();
            run();
        }
        catch (IOException ex) {
            System.err.println("Impossible de trouver ou de lire le fichier de configuration");
        }
    }

    private void lireConfiguration() throws IOException {
        Properties configuration = new Properties();
        configuration.load(getClass().getResourceAsStream("../config/connexion.properties"));
        port = Integer.parseInt(configuration.getProperty("port"));
    }

    private void run() {
        try {
            ServerSocket service = new ServerSocket(port);
            System.out.println("Démarrage du service");
            while (true) {
                Socket client = service.accept();
                System.out.println("Requête de "+client.getInetAddress().getHostAddress());
                traitement(client);
                client.close();
            }
        }
        catch (IOException ex) {
            System.err.println("Ce numéro de service est déjà utilisé");
        }
    }

    private void traitement(Socket client) throws IOException {
        Scanner requête = new Scanner(client.getInputStream());
        requête.useLocale(Locale.US);
        PrintWriter réponse = new PrintWriter(client.getOutputStream(), true);
        double valeur = requête.nextDouble();
        String monnaie = requête.next();
        switch (monnaie) {
            case "€" : réponse.println((valeur*TAUX)+" F"); break;
            case "F" : réponse.println((valeur/TAUX)+" €"); break;
        }
    }
}
```

/home/manu/CloudStation/ProjetsJAVA/MonnaieFXML/src/fx/ConversionFXML.java

```

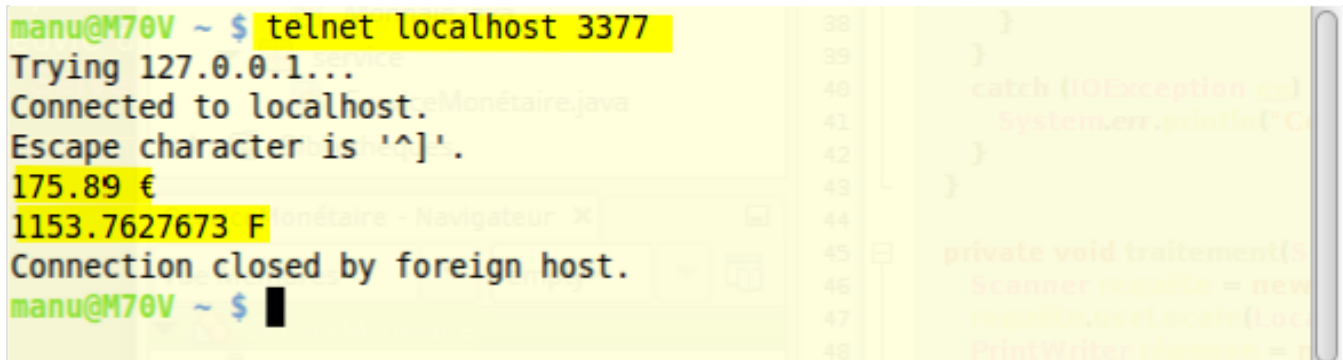
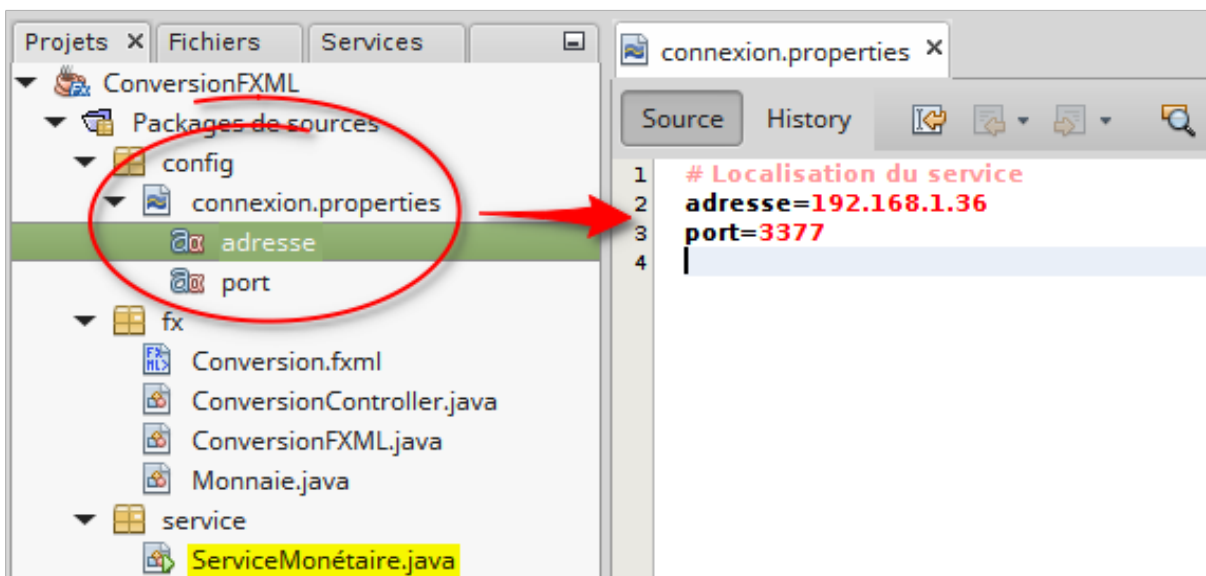
package fx;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.*;
import javafx.stage.Stage;

public class ConversionFXML extends Application {

    @Override
    public void start(Stage fenêtre) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("Conversion.fxml"));
        Scene scene = new Scene(root);
        fenêtre.setTitle("Conversion entre les €uros et les francs");
        fenêtre.setResizable(false);
        fenêtre.setScene(scene);
        fenêtre.show();
    }
}

```



```
/home/manu/CloudStation/ProjetsJAVA/MonnaieFXML/src/fx/Monnaie.java
```

```
package fx;

import javafx.beans.property.*;
import javafx.geometry.Pos;
import javafx.scene.control.TextField;
import javafx.util.converter.CurrencyStringConverter;

public class Monnaie extends TextField {
    private final StringProperty symbole = new SimpleStringProperty("€");
    private final DoubleProperty valeur = new SimpleDoubleProperty(0.0);
    private CurrencyStringConverter valeurMonnaie = new CurrencyStringConverter();

    public Monnaie() { setAlignment(Pos.CENTER_RIGHT); }

    public double getValeur() {
        setValeur(valeurMonnaie.fromString(getText()).doubleValue());
        return valeur.get();
    }

    public void setValeur(double nombre) {
        valeur.set(nombre);
        setText(valeurMonnaie.toString(nombre));
    }

    public String getSymbole() { return symbole.get(); }

    public void setSymbole(String monnaie) {
        symbole.set(monnaie);
        valeurMonnaie = new CurrencyStringConverter("#,##0.00 "+monnaie);
        setText(valeurMonnaie.toString(valeur.get()));
    }

    public DoubleProperty valeurProperty() { return valeur; }
    public StringProperty symboleProperty() { return symbole; }
}
```

/home/manu/CloudStation/ProjetsJAVA/MonnaieFXML/src/fx/Conversion.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import fx.*?>
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<AnchorPane prefHeight="81.0" prefWidth="430.0" xmlns="http://javafx.com/javafx/8"
  xmlns:fx="http://javafx.com/fxml/1" fx:controller="fx.ConversionController">
  <children>
    <Button layoutX="370.0" layoutY="14.0" onAction="#calculerFrancs" text="-&gt; F" />
    <fx.Monnaie fx:id="euros" layoutX="14.0" layoutY="14.0" onAction="#calculerFrancs"
      prefHeight="25.0" prefWidth="349.0" text="0,00 €" />
    <fx.Monnaie fx:id="francs" layoutX="14.0" layoutY="45.0" onAction="#calculerEuros"
      prefHeight="25.0" prefWidth="349.0" symbole="F" />
    <Button layoutX="370.0" layoutY="45.0" onAction="#calculerEuros" text="-&gt; €" />
  </children>
</AnchorPane>
```

The screenshot shows a JavaFX application window with a light gray border. It contains two rows of input fields. The top row has a text field with the value "6 556 920,69 €" and a button labeled "-> F" to its right. The bottom row has a text field with the value "43 010 580,25 F" and a button labeled "-> €" to its right. The buttons are light blue with a darker blue border.

```
/home/manu/CloudStation/ProjetsJAVA/MonnaieFXML/src/fx/ConversionController.java
```

```
package fx;

import java.io.*;
import java.net.*;
import java.util.*;
import javafx.fxml.*;
import javafx.scene.control.*;

public class ConversionController implements Initializable {
    @FXML private Monnaie euros, francs;
    private String adresse;
    private int port;
    private Alert problème;

    @FXML
    private void calculerFrancs() {
        francs.setValeur(communication(euros.getValeur()+" €"));
    }

    @FXML
    private void calculerEuros() {
        euros.setValeur(communication(francs.getValeur()+" F"));
    }

    private double communication(String monnaie) {
        try {
            Socket service = new Socket(adresse, port);
            PrintWriter requête = new PrintWriter(service.getOutputStream(), true);
            Scanner réponse = new Scanner(service.getInputStream());
            requête.println(monnaie);
            réponse.useLocale(Locale.US);
            return réponse.nextDouble();
        }
        catch (IOException ex) {
            problème.showAndWait();
            return 0.0;
        }
    }

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // Configuration boîte d'alerte
        problème = new Alert(Alert.AlertType.ERROR);
        problème.setTitle("Alerte");
        problème.setHeaderText("Problème de communication avec le service");
        problème.setContentText("Service hors d'atteinte");
        // Récupération des informations sur la localisation du service
        ResourceBundle config = ResourceBundle.getBundle("config/connexion");
        adresse = config.getString("adresse");
        port = Integer.parseInt(config.getString("port"));
    }
}
```

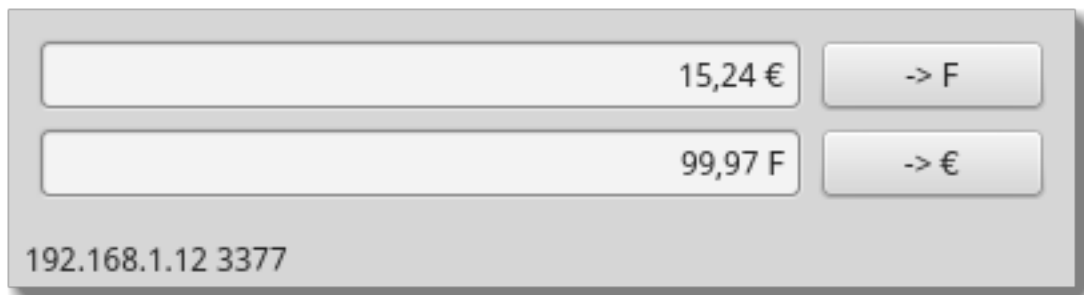
```
#ifndef PRINCIPAL_H
#define PRINCIPAL_H

#include "ui_principal.h"
#include <QtNetwork>

class Principal : public QMainWindow, private Ui::Principal
{
    Q_OBJECT

public:
    explicit Principal(QWidget *parent = 0);
private slots:
    void calculerEuros();
    void calculerFrancs();
    void reception();
private:
    void configuration();
    void soumettre(QString monnaie);
private:
    enum {AUCUN, EURO, FRANC} commande = AUCUN;
    QTcpSocket service;
    QString adresse;
    int port;
};

#endif // PRINCIPAL_H
```



```
#include "principal.h"

Principal::Principal(QWidget *parent) : QMainWindow(parent)
{
    setupUi(this);
    configuration();
    connect(&service, SIGNAL(readyRead()), this, SLOT(reception()));
}

void Principal::configuration()
{
    QSettings config("connexion.ini", QSettings::IniFormat);
    adresse = config.value("Localisation/adresse").toString();
    port = config.value("Localisation/port").toInt();
    barreEtat->showMessage(QString("%1 %2").arg(adresse).arg(port));
}

void Principal::calculerFrancs()
{
    commande = FRANC;
    soumettre(QString("%1 €").arg(euro->value()));
}

void Principal::calculerEuros()
{
    commande = EURO;
    soumettre(QString("%1 F").arg(franc->value()));
}

void Principal::soumettre(QString monnaie)
{
    service.connectToHost(adresse, port);
    QTextStream requete(&service);
    requete << monnaie << endl;
}

void Principal::reception()
{
    if (service.canReadLine())
    {
        double monnaie = service.readLine().split(' ')[0].toDouble();
        switch (commande) {
            case FRANC : franc->setValue(monnaie); break;
            case EURO : euro->setValue(monnaie); break;
            case AUCUN : barreEtat->showMessage("Choisissez votre monnaie");
        }
    }
}
```

