

TP Java ME

Installation du "Wireless Toolkit 2.5.2" de Sun

L'environnement de Sun pour commencer à faire du développement pour Java Micro Edition s'appelle le Wireless Toolkit. Nous utilisons dans ce TP la version 2.5.2. Dans tout cet énoncé on le désignera par WTK (son nom habituel).

0°) Préparez vous un répertoire de travail. Il devra contenir :

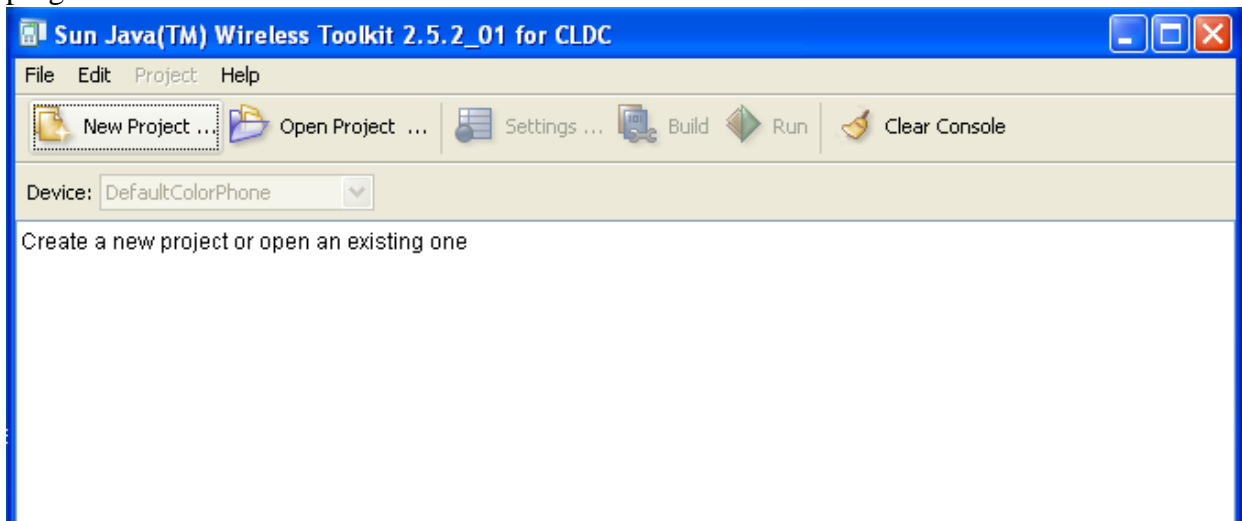
- un sous répertoire de nom `Telecharger` qui contiendra les données téléchargées.
- un sous répertoire de nom `WTK` qui contiendra le WTK installé.

1°) Télécharger cet environnement à partir de l'URL <http://java.sun.com/products/sjwtoolkit/index.jsp>

On utilisera le lien "[Sun Java Wireless Toolkit 2.5.2_01 for CLDC](#)". Il y a plusieurs pages web à parcourir, lisez rapidement ces pages ! Vous devez finir par télécharger le programme `sun_java_wireless_toolkit-2.5.2_01-win.exe`. Mettez-le dans votre répertoire `Telecharger`.

2°) Lancer l'installation de cet environnement. Il y a plusieurs écrans qui apparaissent lors de cette installation. On placera cet environnement dans le répertoire `WTK`.

3°) Lancer l'exécution de cet environnement. Pour lancer cet environnement aller dans le répertoire où vous avez installé le WTK, puis dans son sous répertoire `bin` et lancer le programme `ktoolbar.exe`. Vous devez obtenir la fenêtre ci-dessous :



Remarque : Dans une ancienne version un raccourci était installé dans le menu des programmes. et on pouvait lancer donc cet environnement par :

démarrer | Programmes | Sun Java (TM) Wireless Toolkit 2.5.2 for CLDC | Wireless Toolkit 2.5.2 (sous Windows XP)

L'environnement "Wireless Toolkit"

4°) Quand cet environnement a été lancé, exécuter les démonstrations proposées par exemple : Demos, CityGuide, UIDemo par **OpenProject** puis **Run**. Votre souris ne doit servir qu'à appuyer sur les touches de l'émulateur.

5°) Changer de **Device** (de **DefaultColorPhone** à **QuertyDevice**) et tester certaines midlets.

Le client Java ME ClientCompteBancaireMIDlet

6°) Vous allez écrire une MIDlet qui sera un client permettant d'atteindre et de faire des opérations sur un compte bancaire.

6.1) Pour cela cliquer **New Project**, Entrer un nom de projet (EDJavaMECNAM) et un nom de classe (ClientCompteBancaireMIDlet) puis cliquer **Create Project**. Accepter les configurations par défaut dans la fenêtre suivante : cliquer **OK**.

6.2) Vérifier qu'un nouveau projet a été créé : le répertoire EDJavaMECNAM a été créé sous C:\Documents and Settings\nomDUtilisateurDeVotreMachine\j2mewtk\2.5.2\apps.

6.3) Dans ce répertoire EDJ2MECNAM ont été créés plusieurs fichiers et sous-répertoires. Editer votre MIDlet dans le fichier ClientCompteBancaireMIDlet.java :

fichier ClientCompteBancaireMIDlet.java

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import java.io.*;

public class ClientCompteBancaireMIDlet extends MIDlet implements CommandListener,
Runnable {
    private Form mMainForm, connectForm;
    private ChoiceGroup cg;
    private Command cmdExit, cmdTransaction;
    private TextField tf;
    private String reponseDeLaServlet = "";

    // les 3 méthodes abstraites de MIDlet
    public void destroyApp(boolean condition){}
    public void pauseApp(){}
    public void startApp(){
        Display.getDisplay(this).setCurrent(mMainForm);
    }
    // La methode de l'interface CommandListener
    public void commandAction(Command c, Displayable d) {
        if (c == cmdExit) {
            destroyApp(true);
            notifyDestroyed();
        } else if (c == cmdTransaction) {
            Thread t = new Thread(this);
            t.start();
            Display.getDisplay(this).setCurrent(connectForm);
        }
    }

    public void run() {
        envoiRequete();
    }

    public void envoiRequete() {
        HttpURLConnection hc = null;
        InputStream in = null;
        String url = "http://localhost:8080/JavaMeSMB111/GereCompteBancaireServlet";

        try {
            int choix = cg.getSelectedIndex();
            String urlComplete = url;
```

```

String valeur = null;
switch (choix) {
case 0 :
    // A FAIRE 1
    // Compléter urlComplete pour pouvoir obtenir le solde du compte bancaire
    // On pourra s'inspirer du code lignes suivantes
    break;
case 1 :
    valeur = tf.getString();
    urlComplete += "?operation=debiter&quantite="+valeur;
    break;
case 2 :
    valeur = tf.getString();
    // A FAIRE 2
    // Compléter urlComplete pour pouvoir crediter valeur euro sur le compte
bancaire
    // On pourra s'inspirer du code lignes précédentes !
    break;
}
System.out.println(urlComplete);

hc = (HttpConnection)Connector.open(urlComplete);
in = hc.openInputStream();

int contentLength = (int)hc.getLength();
byte[] raw = new byte[contentLength];
int length = in.read(raw);

in.close();
hc.close();

// traite la réponse
reponseDeLaServlet = new String(raw, 0, length);
mMainForm = null;
mMainForm = new Form("MIDlet JMF pour manipuler le compte bancaire");
remplirMainForm();
Display.getDisplay(this).setCurrent(mMainForm);
} catch (IOException ioe) {
    System.out.println("Pb dans l'envoi de la requête : " + ioe);
}
}

private void remplirMainForm() {
    cg = new ChoiceGroup("Sur votre compte bancaire voulez vous : ",
ChoiceGroup.EXCLUSIVE);
    mMainForm.append(cg);
    cg.append("Obtenir le solde\nou bien", null);
    cg.append("Retirer\nou", null);
    cg.append("Crediter", null);
    tf = new TextField("la somme de : ", "", 7, TextField.DECIMAL );
    mMainForm.append(tf);
    mMainForm.append(new StringItem(null, "euro"));
    mMainForm.append(new StringItem(null, reponseDeLaServlet));

    cmdExit = new Command("Exit", Command.EXIT, 0);
    mMainForm.addCommand(cmdExit);
    cmdTransaction = new Command("Lancer transaction", Command.EXIT, 0);
    mMainForm.addCommand(cmdTransaction);
    mMainForm.setCommandListener(this);
}

private void remplirConnectForm() {
    StringItem messageLabel = new StringItem(null, "En cours de connexion (eh
oui !)");
    connectForm.append(messageLabel);
    connectForm.setCommandListener(this);
}

```

```

public ClientCompteBancaireMIDlet() {
    mMainForm = new Form("MIDlet pour manipuler le compte bancaire");
    remplirMainForm();
    connectForm = new Form("Connexion à la servlet");
    remplirConnectForm();
}
}

```

6.4) Sauvegarder cette MIDlet ClientCompteBancaireMIDlet.java dans ce répertoire
 (C:\Documents and Settings\nomDUtilisateurDeVotreMachine\j2mewtk\2.5.2\apps\src)

6.5) Construire la MIDlet par l'environnement WTK en cliquant **Build**. Lancer l'exécution en cliquant **Run**. Vous devez obtenir l'interface graphique ci dessous :



Evidemment toute la partie réseau ne fonctionne pas encore : c'est l'objet de la suite de ce TP.

Le serveur GereCompteBancaireServlet

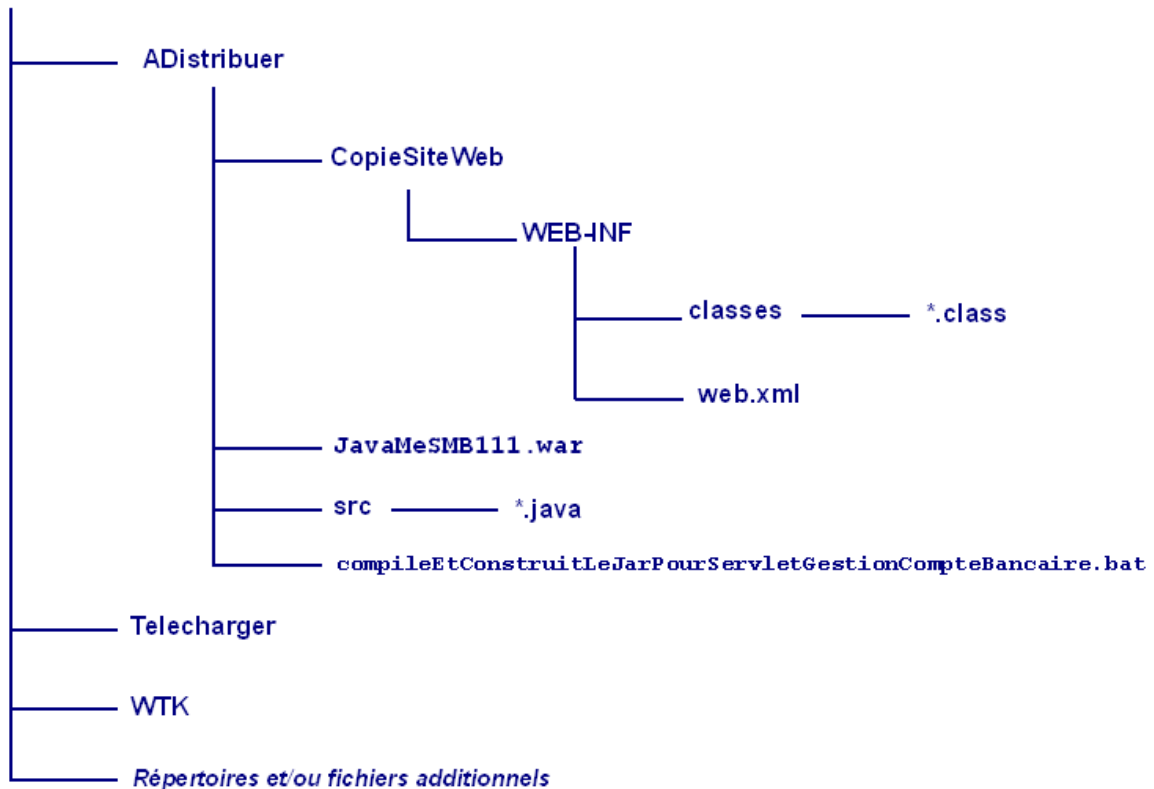
Le programme qui va recevoir les requêtes de la MIDlet et y répondre est une servlet.

7°) Récupérer à l'URL

<http://cedric.cnam.fr/~farinone/SMB111/JavaME/>

le fichier `ADistribuer.zip`. Ouvrez ce fichier dans votre répertoire de travail. Vous devez désormais avoir une arborescence comme :

racine du répertoire de travail



Vous avez donc dans le répertoire `ADistribuer` :

- le `.war` de cette application web
- une copie du site web qui sera finalement mise sous tomcat
- les sources `*.java` pour votre servlet
- un script qui permet de construire cette application web

8°) Lancer tomcat.

Déployer le fichier `JavaMeSMB111.war` dans tomcat (utiliser Tomcat Manager cf. TP sur les servlets/JSP).

9°) Lisez le code de la servlet `GereCompteBancaireServlet` (fichier `GereCompteBancaireServlet.java`) ainsi que le fichier `web.xml`.

9.1) En déduire l'URL qu'il faut utiliser pour obtenir le solde du compte bancaire.

9.2) De même en lisant le code de la servlet `GereCompteBancaireServlet`, déduire l'URL pour débiter 52 euro.

9.3) De même en lisant le code de la servlet `GereCompteBancaireServlet`, déduire l'URL pour créditer 52 euro.

9.4) Vérifier vos réponses aux questions 9.1, 9.2, 9.3 en accédant à ces URLs par un navigateur.

L'architecture client-serveur (MIDlet client, servlet serveur)

10°) Editer la MIDlet `ClientCompteBancaireMIDlet.java` (qui se trouve dans le répertoire `(C:\Documents and Settings\nomDUtilisateurDeVotreMachine\j2mewtk\2.5.2\apps\src)`).

Aux vues des réponses aux questions 9.x, compléter dans cette MIDlet les parties A FAIRE 1 et A FAIRE 2.

11°) Reconstituez la MIDlet (par Build) et lancer l'exécution de cette MIDlet (par Run). Tester l'obtention du solde et des opérations créditer et débiter.

Vous devez désormais pouvoir manipuler votre compte bancaire par un client web ou un téléphone portable (eh oui!)