



Remarques sur le TP

Jean-Marc Farinone
Maître de conférences
Conservatoire National des Arts et Métiers

Exercice 1 : une authentification



The image shows a mobile application interface for authentication. At the top, a dark header bar contains the text "AuthenticationProjet" in white. Below the header, the instruction "Enter votre nom et votre mot de passe" is displayed. There are two input fields: the first contains the text "toto", and the second is a password field with six dots and a vertical cursor, highlighted by an orange border. Below the password field is a large, light gray button labeled "Connecter". At the bottom of the screen, there is a smaller, rounded button labeled "Connection OK".

L'IHM



- On peut construire toute l'IHM en glisser déposer et faire afficher l'écran sans avoir écrit une seule ligne de code
- Pour les événements (clic sur le bouton Connecter), il faut écrire du code (voir le cours)
remarque 1 : si vous êtes dans une méthode d'une classe interne à l'activité (qui est un Context) et que vous voulez récupérer ce Context, il faut écrire *ActiviteEnglobante.this*
- remarque 2 : récupérer le texte d'un composant graphique EditText utilise la méthode

```
public CharSequence getText ()
```


Convertir un CharSequence en String est fait à l'aide de la méthode `toString()`

Une solution (1/2)

- Pour les questions 1, 2, 3, le code de la classe peut être :

```
package cnam.rsx116;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class Exo1ProjetActivity extends Activity {
    private EditText etNom;
    private EditText etMotDePasse;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Button button = (Button)findViewById(R.id.button1);
        // Enregistrer l'auditeur auprès du bouton
        button.setOnClickListener(lAuditeurDuBouton);
        etNom = (EditText)findViewById(R.id.editText1);
        etMotDePasse = (EditText)findViewById(R.id.editText2);
    }
}
```

Une solution (2/2)

```
private OnClickListener lAuditeurDuBouton = new OnClickListener() {
    public void onClick(View v) {
        String chaineAAfficher = null;
        String leNomUtilisateur = etNom.getText().toString();
        String leMotDepasse = etMotDePasse.getText().toString();
        if (leMotDepasse.equals(leNomUtilisateur + "pw")) {
            chaineAAfficher = "bonne authentication";
        } else {
            chaineAAfficher = "MAUVAISE authentication, recommencer";
        }
        // chaineAAfficher = leMotDepasse;
        Toast leToast = Toast.makeText(Exo1ProjetActivity.this, chaineAAfficher,
Toast.LENGTH_LONG);
        leToast.show();
    }
};
}
```

Solution question 4 (1/2)

- On peut, en utilisant la correction précédente, écrire dans le code de la classe :

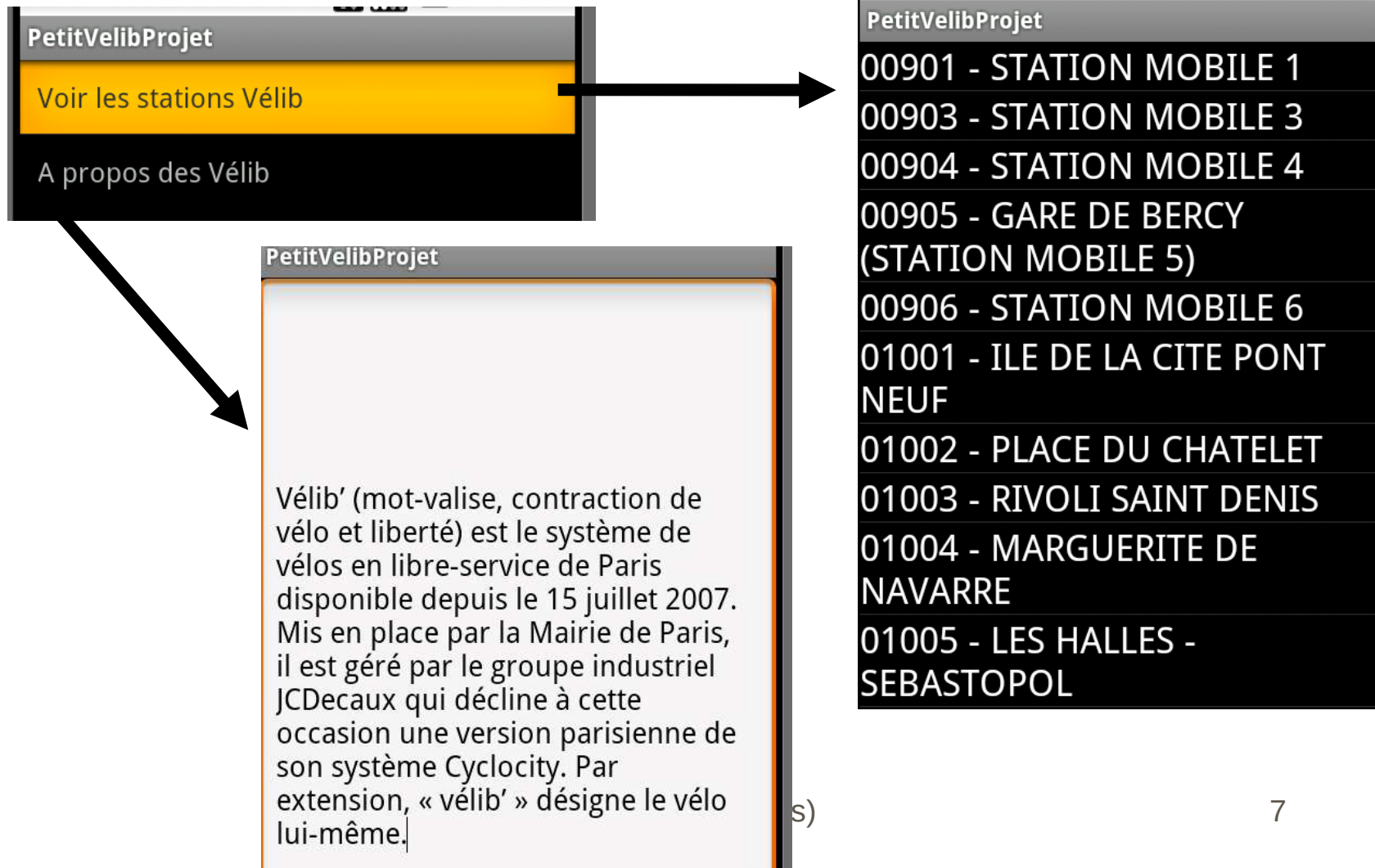
```
if (leMotDepasse.equals(leNomUtilisateur + "pw")) {  
    Intent i0 =  
        new Intent(Exo1ProjetActivity.this, BonneAuthentificationActivity.class);  
    startActivity(i0);  
}
```

- Il ne faut pas oublier de déclarer la nouvelle classe dans le fichier `AndroidManifest.xml`

```
<activity android:name=".BonneAuthentificationActivity" >
```

Exercice 2 :

ListView, Velibs and Cie !



Première étape : une solution

- Pas de difficulté en suivant les diapos du cours
- Le code peut être :

```
package cnam.rsx116;

import android.app.ListActivity;
import android.content.res.Resources;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class Exo2ProjetActivity extends ListActivity
{
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ListView lv = getListView();
        lv.setTextFilterEnabled(true);
        Resources res = getResources();
        String [] lesItems = new String[] {
            res.getString(R.string.seeAllStations),
            res.getString(R.string.aboutUs)
        };

        setListAdapter(new ArrayAdapter<String>(this, R.layout.list_item,
lesItems));
    }
}
```

Seconde étape : une solution

- C'est, en gros (!), le premier exercice. Il faut simplement coder l'action sur un item de ListView plutôt que sur un bouton. Le code peut être :

```
ListView lv = ...
lv.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        // le code lancé lors de la sélection d'un item
        if (position == 1) { // le second item de la ListView
            String chaineAAfficher = "Bravo les vélibs";
            Toast leToast = Toast.makeText(Exo2ProjetActivity.this,
chaineAAfficher, Toast.LENGTH_LONG);
            leToast.show();
        }
    }
});
```

Troisième étape : une solution

- Le but est de faire le traitement XML du fichier XML se trouvant à l'URL donné. On construit un `ArrayAdapter<StationVelib>` qui sera l'adapter pour la `ListView` à afficher
- C'est un peu long mais il n'y a pas de réelle difficulté si on suit l'énoncé et les notes de cours



Fin