

Formation Android



Épisode III

Xavier MARIN



@XavMarin

<https://github.com/Giwi>

CTO chez @qaobee

Et si on faisait un Twitter like?



Objectifs de l'atelier

Maitriser et être autonome sur les éléments suivants:

- Mécanisme de Pulling
- Composants graphiques avancés

Ce qu'il nous manque

- Un service de polling
- Deux solutions possibles
 - Service
 - TimerTask
- Une liste de messages

Utilisons le TimerTask, plus simple

TimerTask

```
Timer timer = new Timer();
final Handler handler = new Handler();
TimerTask timerTask = new TimerTask() {
    public void run() {
        handler.post(new Runnable() {
            public void run() {
                // le code du pull
            }
        });
    }
};
timer.schedule(timerTask, 0, 2000);
```

Liste

Pour afficher une ListView il faut:

- Layout : comment est afficher un item
- Model : item à afficher
- Adapter : en charge d'afficher les items
- Assembler le tout

Liste : Item layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="5dip">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:id="@+id/msg_user" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/msg_message" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:id="@+id/msg_date" />
</LinearLayout>
```


Liste : Model

```
public class Message {  
    public Message(String username, String message, long date) {  
        this.username = username;  
        this.msg = message;  
        this.date = date;  
    }  
  
    public String getUsername() { return username; }  
  
    public String getMsg() { return msg; }  
  
    String username;  
    String msg;  
  
    public long getDate() {  
        return date;  
    }  
  
    long date;  
}
```

Liste : Adapter

- Etendre BaseAdapter
- Remplir les méthodes
- Utiliser le pattern ViewHolder

Liste : Adapter

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    ViewHolder vh;
    if(convertView==null){
        // inflate the layout
        LayoutInflater inflater = ((Activity)context).getLayoutInflater();
        convertView = inflater.inflate(R.layout.item_message, parent, false);
        vh = new ViewHolder();
        vh.username = (TextView) convertView.findViewById(R.id.msg_user);
        vh.message = (TextView) convertView.findViewById(R.id.msg_message);
        vh.date = (TextView) convertView.findViewById(R.id.msg_date);
        convertView.setTag(vh);
    } else {
        vh = (ViewHolder) convertView.getTag();
    }
    vh.username.setText(messages.get(position).getUsername());
    vh.message.setText(messages.get(position).getMsg());
    try {
        vh.date.setText(DateHelper.getFormattedDate(messages.get(position).getDate()));
    } catch (ParseException e) {
        e.printStackTrace();
    }

    return convertView;
}
```

On assemble le tout

```
listView = (ListView) findViewById(R.id.tchat_list);  
adapter = new MessagesAdapter(this);  
listView.setAdapter(adapter);
```

Bon, on code?



A faire

- Une page de login
- Une page de création de compte
- Une page pour le tchat (Affichage et post)



A votre tour !

<https://cesi-giwisoft.rhcloud.com/apidoc/>

