

Les Widgets de sélection



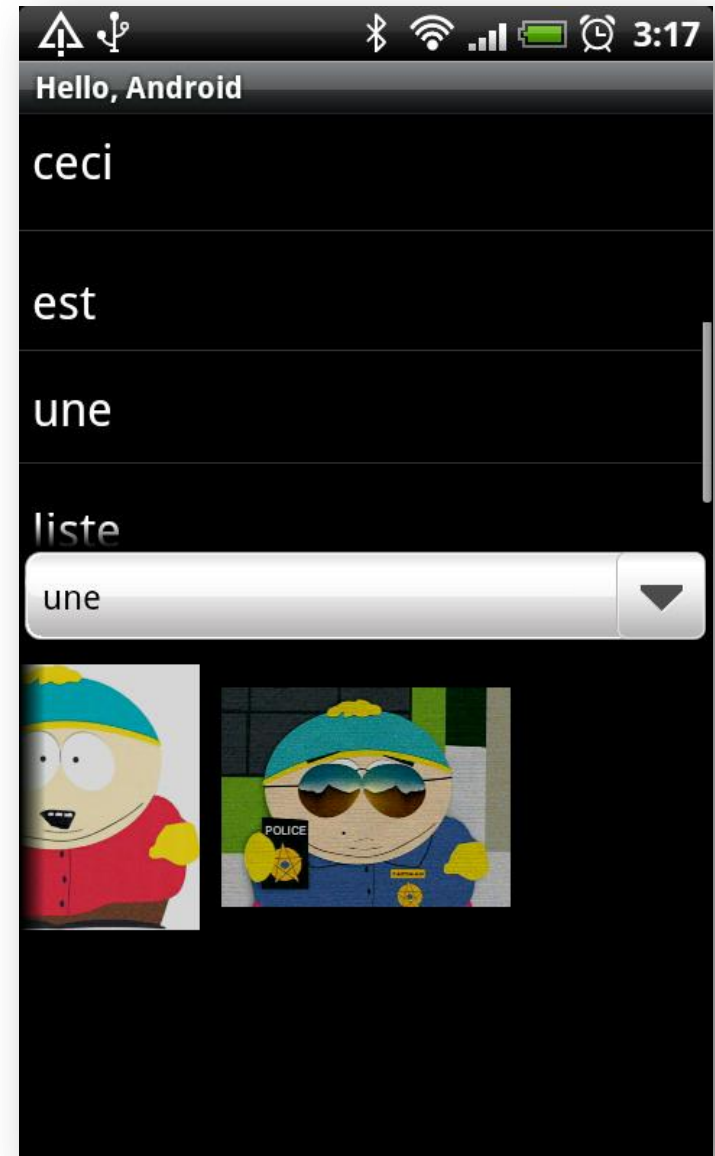
Application Android



Les widgets de sélection

Les widgets de sélection

- Widgets de sélection les plus utilisés
 - ListView
 - Spinner
 - Gallery
- Les widgets de sélection sont associés à des « adapteurs » qui offrent une interface commune permettant de fournir la listes des données d'un widget et de les convertir en vues spécifiques pour l'affichage



Les widgets de sélection

- Implémentation Xml

```
<ListView
    android:id = "@+id/list"
    android:layout_width="fill_parent"
    android:layout_height="200dip"/>

<Spinner
    android:id = "@+id/spinner"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>

<Gallery
    android:id = "@+id/gallery"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:spacing = "10dip"/>
```

Gallery

Les widgets de sélection

Création d'un Adapter pour l'objet Gallery

```
public class ImageAdapter extends BaseAdapter {
    int mGalleryItemBackground;
    private Context mContext;
    private Integer[] mImageIds = {
        R.drawable.images, R.drawable.image2, R.drawable.images_android3,
        R.drawable.cognicap, };

    public ImageAdapter(Context c) {
        mContext = c; }

    public int getCount() {
        return mImageIds.length; }

    public Object getItem(int position) {
        return position; }

    public long getItemId(int position) {
        return position; }

    public View getView(int position, View convertView, ViewGroup parent) {
        ImageView i = new ImageView(mContext);
        i.setImageResource(mImageIds[position]);
        i.setLayoutParams(new Gallery.LayoutParams(150, 100));
        i.setScaleType(ImageView.ScaleType.FIT_XY);
        return i; } }
```

Widgets de selection : Gallery

- Implémentation XML

```
<Gallery
    android:id="@+id/gallery"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>
```

- Implementation Java

```
public class WidgetsSelection extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Gallery g = (Gallery) findViewById(R.id.gallery);
        g.setAdapter(new ImageAdapter(this));
        g.setOnItemClickListener(new OnItemClickListener() {
            public void onItemClick(AdapterView parent, View v, int position, long id) {

                Toast.makeText(WidgetsSelection.this, "" + position, Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Widgets de selection : Gallery

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:paddingLeft="14dip" >

    <Gallery
        android:id="@+id/gallery"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:spacing="2dip"
        android:unselectedAlpha="0.5"
        android:gravity="fill_horizontal"/>
</LinearLayout>
```


Widgets de selection : Gallery

```
package com.example.gallery;

import android.app.Activity;
import android.os.Bundle;

public class Gallery extends Activity {
    /** Called when the activity is first created. */
    android.widget.Gallery galerie ;
    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        setContentView(R.layout.activity_main);
        ((android.widget.Gallery) findViewById(R.id.gallery)).
            setAdapter(new ImageAdapter(this));
    }
}
```

Widgets de selection : Gallery

```
public class ImageAdapter2 extends BaseAdapter {
    int mGalleryItemBackground;
    private Context mContext;
    private Integer[] mImageIds = {
        R.drawable.fadel1, R.drawable.fadel2, R.drawable.fadel3,
        R.drawable.fadel4};

    public ImageAdapter2(Context c) {
        mContext = c; }

    public int getCount() {
        return mImageIds.length; }

    public Object getItem(int position) {
        return mImageIds[position]; }

    public long getItemId(int position) {
        return position; }

    public View getView(int position, View convertView, ViewGroup parent) {
        ImageView i = new ImageView(mContext);
        i.setImageResource(mImageIds[position]);
        i.setLayoutParams(new Gallery.LayoutParams(300, 200));
        i.setScaleType(ImageView.ScaleType.FIT_XY);
        return i; } }
```

ListView

- Place les éléments en liste verticale et ajoute un ascenseur si nécessaire
 - **Séparateurs**
 - `android:divider` Pour définir la couleur des séparateurs ou pour utiliser une image comme séparateur.
 - `android:dividerHeight="unité"` Pour définir la hauteur des séparateurs (même s'ils contiennent une image)
 - **Type de choix**
 - `android:choiceMode="c"` (où c peut prendre les valeurs : `none`, `singlechoice`, `multipleChoice`) pour indiquer le mode de choix dans la liste (aucun, un seul, plusieurs).

Exemple de ListView

- En XML

- Dans le XML d'interface

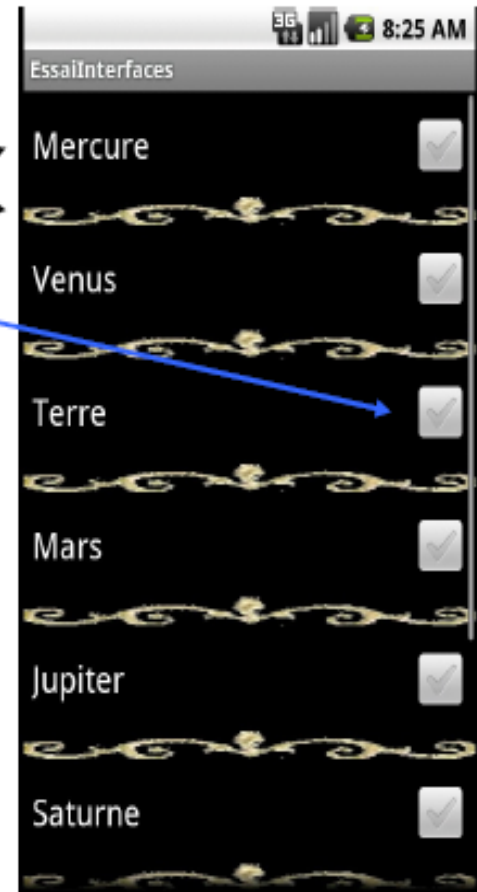
```
<ListView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/liste_de_planetes"
    android:entries="@array/planetes"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:divider="@drawable/separateur"
    android:dividerHeight="25px"
    android:choiceMode="multipleChoice"
/>
```

- Dans le XML de valeurs

```
<string-array name="planetes">
    <item>Mercure</item>
    ...
    <item>Neptune</item>
</string-array>
```

- Dans le code

```
ListView liste = (ListView) findViewById(R.id.liste_de_planetes);
String[] elements = getResources().getStringArray(R.array.planetes);
ArrayAdapter<String> adaptateur = new ArrayAdapter<String>(this,
    android.R.layout.simple_list_item_multiple_choice);
for (int i=0; i<elements.length; i++) adaptateur.add(elements[i]);
liste.setAdapter(adaptateur);
```



GridView

- Fonctionne comme ListView mais permet une présentation en plusieurs colonnes
- Exemple

- Dans le XML d'interface

```
<GridView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/liste_de_planetes"
    android:entries="@array/planetes"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:numColumns="2"
    android:stretchMode="columnWidth"
    android:columnWidth="60dp"
    android:gravity="fill_horizontal"
    android:choiceMode="multipleChoice"
/>
```

- Dans le code (même principe)

```
GridView table = (GridView) findViewById(R.id.liste_de_planetes);
String[] elements = getResources().getStringArray(R.array.planetes);
ArrayAdapter<String> adaptateur = new ArrayAdapter<String>(this,
    android.R.layout.simple_list_item_multiple_choice);
for (int i=0; i<elements.length; i++) adaptateur.add(elements[i]);
table.setAdapter(adaptateur);
```



Adapter l'affichage dans les items

Les widgets de sélection

Création d'un Adapter

Extension de la classe **BaseAdapter** et implémentation des méthodes

- **int getCount():** qui renvoie le nombre d'éléments à afficher
- **Object getItem(int position):** renvoie l'objet à la position passée en paramètre
- **long getItemId(int position):** renvoie l'identifiant de l'item à la position mentionnée
- **View getView(int position,..):** construit et renvoie la vue de l'item à la position fournie

Les widgets de sélection

Création d'un Adapter pour l'objet ListView

```
public class LivreAdapter extends BaseAdapter {

    List<Livre> biblio;
    LayoutInflater inflater;

    public LivreAdapter(Context context, List<Livre> biblio) {
        inflater = LayoutInflater.from(context);
        this.biblio = biblio;
    }

    @Override
    public int getCount() {
        return biblio.size();
    }

    @Override
    public Object getItem(int position) {
        return biblio.get(position);
    }

    @Override
    public long getItemId(int position) {
        return position;
    }
}
```


Les widgets de sélection

Création des interfaces XML

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

<ListView
    android:id="@+id/lvListe"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
</ListView>

</LinearLayout>
```

```
View lvListe;

<Livres> maBibliotheque = new ArrayList<Livre>();

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.mainlivre);
    lvListe = (ListView)findViewById(R.id.lvListe);
    RemplirLaBibliotheque();
    LivreAdapter adapter = new LivreAdapter(this, maBibliotheque);
    lvListe.setAdapter(adapter);
    adapter.notifyDataSetChanged(); }

private void RemplirLaBibliotheque() {
    maBibliotheque.clear();
    maBibliotheque.add(new Livre("Starcraft 2 : Les diables du ciel", "Will"));
    maBibliotheque.add(new Livre("L'art d android", "Mark Murphy"));
    maBibliotheque.add(new Livre("Le seuil des ténèbres", "Karen Chance"));
    maBibliotheque.add(new Livre("Starcraft 2 : Les diables du ciel", "Will"));
    maBibliotheque.add(new Livre("L'art d android", "Mark Murphy"));
    maBibliotheque.add(new Livre("Le seuil des ténèbres", "Karen Chance"));
    maBibliotheque.add(new Livre("Starcraft 2 : Les diables du ciel", "Will"));
    maBibliotheque.add(new Livre("L'art d android", "Mark Murphy"));
    maBibliotheque.add(new Livre("Le seuil des ténèbres", "Karen Chance"));
    maBibliotheque.add(new Livre("Starcraft 2 : Les diables du ciel", "Will"));
    maBibliotheque.add(new Livre("L'art d android", "Mark Murphy"));
    maBibliotheque.add(new Livre("Le seuil des ténèbres", "Karen Chance"));
```

```
public class MonActivite extends Activity {

    ListView lvListe;
    List<Livre> maBibliotheque = new ArrayList<Livre>();

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mainlivre);
        lvListe = (ListView)findViewById(R.id.lvListe);
        RemplirLaBibliotheque();
        LivreAdapter adapter = new LivreAdapter(this, maBibliotheque);
        lvListe.setAdapter(adapter);
        adapter.notifyDataSetChanged(); }
}
```

Les widgets de sélection

Création d'un Adapter pour l'objet ListView

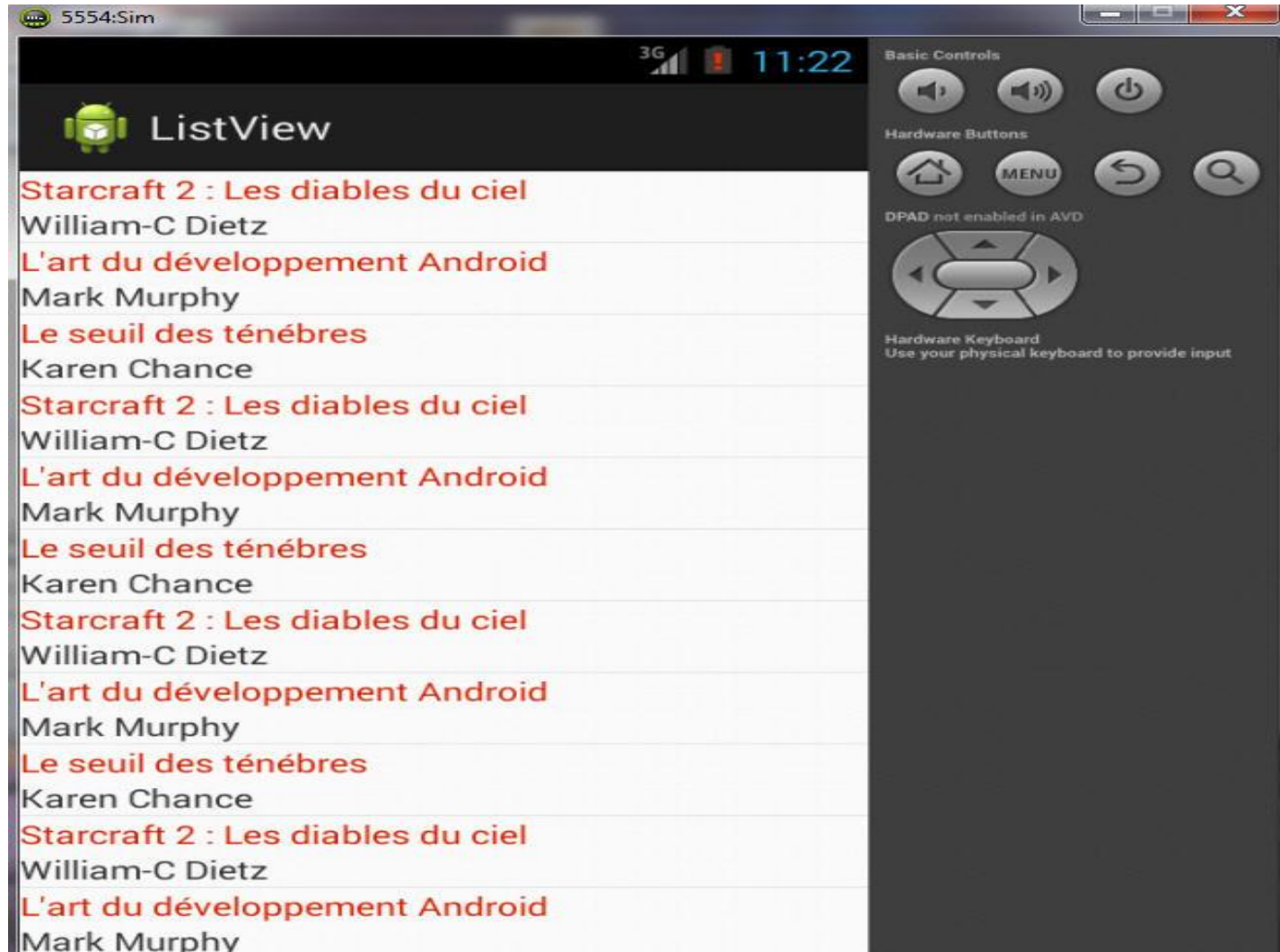
```
private class ViewHolder {
    TextView tvTitre;
    TextView tvAuteur;}

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    ViewHolder holder;
    if (convertView == null) {
        holder = new ViewHolder();
        convertView = inflater.inflate(R.layout.itemlivre, null);
        holder.tvTitre = (TextView) convertView.findViewById(R.id.tvTitre);
        holder.tvAuteur = (TextView) convertView.findViewById(R.id.tvAuteur);
        convertView.setTag(holder);
    } else {
        holder = (ViewHolder) convertView.getTag();
    }

    holder.tvTitre.setText(biblio.get(position).getTitre());
    holder.tvAuteur.setText(biblio.get(position).getAuteur());

    return convertView;
}
```

Les widgets de sélection



- Comment naviguer entre deux Activités ??

Intent

Lancement d'une intention:

startActivity()

On lance la deuxième activité avec cette méthode mais l'activité appelante ne sera pas prévenue de la fin de l'activité fille

startActivityForResult()

Cette méthode nous permet de naviguer entre 2 activités en lui passant un id comme paramètre. Lorsque l'activité fille se termine, l'activité mère sera prévenue via la méthode **onActivityResult();**

Intent

Création d'une intention:

On instancie la classe **Intent** pour passer d'une activité à une autre :

```
Intent intent = new Intent(this, WidgetsSelection.class);  
startActivityForResult(intent, 1);
```

On peut également créer une intention pour une action donnée.

```
final String requete = "http://www.google.fr/search?q=cognicap" ;  
Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(requete));  
startActivity(intent);
```

Alors comment on peut lancer
l'activité ?

Intent : Exemple

- Modification du listener du bouton au niveau de la première activité

```
final Intent intent = new Intent(this, WidgetsSelection.class);
Button bouton = (Button) this.findViewById(R.id.bouton);
bouton.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivityForResult(intent, 1);
    }});
```

- Envoie d'un paramètre vers l'activité fille :

```
intent.putExtra("site", "www.cognicap.com");
```

- Récupération du paramètre :

Activités/Intentions


Lancement d'activités et de sous activités

Pour démarrer une activité, il faut une intention et choisir comment la lancer

Création d'une intention:

-  On instancie la classe **Intent** en passant comme argument l'activité qu'on souhaite lancer

```
new Intent(this, BaseWidgetsActivity.class);
```

-  On peut également créer une intention pour une action donnée.

```
new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.novediagroup.com"));
```

Lancer une activité

- Lancer explicitement une activité

```
Intent demarre = new Intent(this, NomDeLaClasseDeLActiviteALancer.class);  
startActivity(demarre);
```

- Lancer implicitement une activité

– Exemple : lancer un navigateur sur une page :

```
Uri chemin = Uri.parse("http://www.google.fr");  
Intent naviguer = new Intent(Intent.ACTION_VIEW, chemin);  
startActivity(naviguer);
```

– Exemple : appeler un n° de téléphone :

```
Uri numero = Uri.parse("tel:0559574320");  
Intent appeler = new Intent(Intent.ACTION_CALL, numero);  
startActivity(appeler);
```

Lancer une activité et obtenir un retour

- Lancement (dans l'activité A)

```
static final int MON_CODE = 1; // code servant à identifier l'activité qui répond
Intent démarre = new Intent(this, NomDeLaClasseDeLActiviteB.class);
// ajouter les paramètres passés à B dans l'Intent démarre
startActivityForResult(intention, MON_CODE); // lancement de l'activité B
```
- Renvoi du code et des valeurs de retour (dans l'activité B)

```
Intent intent_retour = new Intent(); // Préparer un Intent pour les valeurs de retour
// Ajouter les valeurs de retour à l'Intent intent_retour
setResult(code, intent_retour); // renvoyer un code de retour (entier) et l'Intent de retour
finish(); // terminer l'activité B
```
- Traitement du code de retour (dans l'activité A)

```
protected void onActivityResult(int ident, int code_retour, Intent retour) { // surcharge
    switch(ident) {
        case MON_CODE : // c'est l'activité B qui a répondu
            // récupération des valeurs de retour contenues dans l'Intent retour
            // traitement selon le code de retour contenu dans l'entier code_retour
            return;
        ...
    }
}
```

Passer des paramètres à l'activité appelée

- La classe `Intent` permet de passer des paramètres à l'activité appelée et d'en récupérer les valeurs en retour
- Ajouter des paramètres (types simples ou tableaux)
Objet_intent.putExtra(String, val)
Le 1^{er} paramètre est un nom (clé)
Le second paramètre est la valeur :
 - De type simple (boolean, int, short, long, float, double, char)
 - Tableau de types simples
 - String et tableau de String
- L'activité appelée pourra récupérer ces paramètres par leur nom :

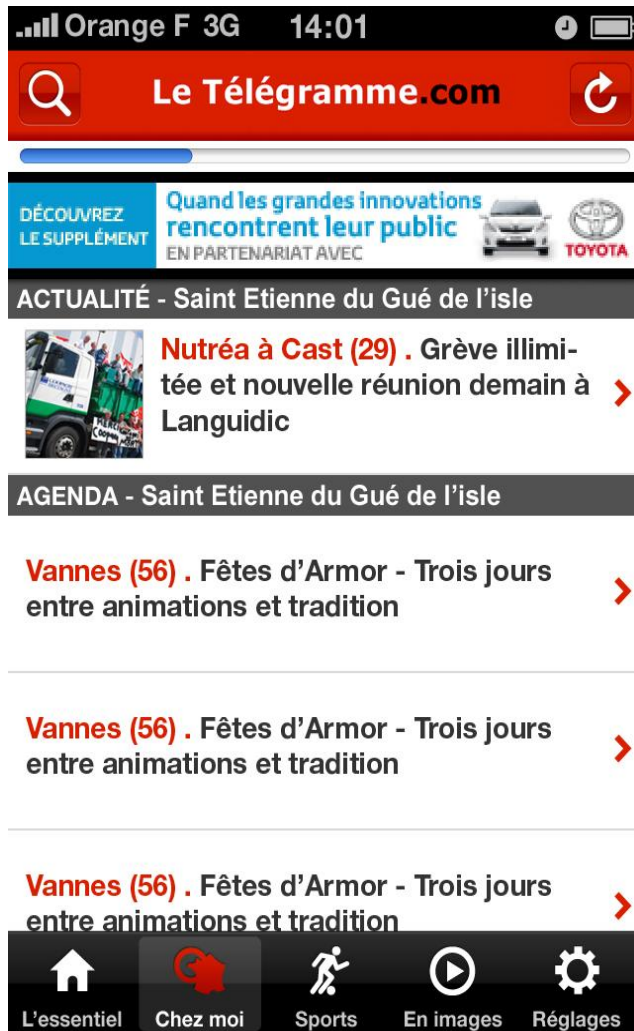
Récupérer les paramètres dans l'activité appelée

- L'activité lancée récupère l'objet de classe `Bundle` contenant les paramètres par :
`Bundle params = getIntent().getExtras()`
- Les paramètres sont récupérés dans ce `Bundle` par ses méthodes :
 - `getBoolean(String)`
 - `getInt(String)`
 - ...
 - `getBooleanArray(String)`
 - ...auxquelles on passe la clé en paramètre

Placer des valeurs de retour dans l'activité appelée

- Le méthode `setResult(int, Intent)` permet de renvoyer un code de retour et un Intent de retour
- L'activité appelée place les valeurs de retour dans cet Intent par `putExtra(String, val)` comme déjà vu pour les paramètres
- L'activité appelante récupère cet Intent comme dernier paramètre de la méthode :
`onActivityResult(int req, int code_retour, Intent retour)`
- Elle en extrait les paramètres par les méthodes de la classe Intent :
 - `getBooleanExtra(String)`
 - `getIntExtra(String)`
 - ...
 - `getBooleanArrayExtra(String)`
 - ...

2 – En local - Accueil



- La partie supérieure de la liste est réservée aux actualités. On affiche les actualités (il peut y en avoir plusieurs comme aucune).
- Actualité et Agenda sont des séparateurs, si l'une des rubriques est vide, son séparateur ne s'affiche pas
- L'ensemble de la liste scrolle ensemble

Erreur maquette : Le formatage des cartouches doit être identique à l'accueil de L'essentiel (titre sur 2 lignes puis chapeau sur 2 lignes pour les articles et début de la description de l'événement sur 2 lignes pour les événements)

2 – En local – Détail d'un événement



Vannes (56) . Fêtes d'Armor - Trois jours entre animations et traditions

Fêtes / Festival

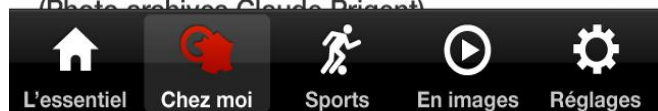
Le 23/08/2010, à 10h00

Lieu Douarnenez (29100) / Différents lieux

Tarifs Séance de 6 à 4,50€ / -12ans : 4€

Contact Nom du contact / mail@contact.fr / **01 61 45 56 56 / 06 65 65 65 23**

Philippe Le Gal, le directeur du Carré Magique, à Lannion, a dévoilé, hier, le programme de la saison. Outre une large place accordée à la danse, au cirque et au cinéma, le Carré Magique accueillera, entre autres, Marcel Azzola, accordéoniste qui a joué avec Brel, le 3 décembre, et Ibrahim Maalouf (ci-dessus), trompettiste de jazz qui compte au nombre de ses collaborations, Matthieu Chedid, Vincent Delerm et Sting. (Photo archives Claude Brizant)



- Le format d'affichage d'un événement sera le même que les articles d'actualité, seule la partie basse sera ajoutée
- La date et l'horaire, le lieu, le tarif et le contact seront récupérés dans le flux. Si pour un de ces éléments le flux est vide, on n'affiche rien
- Pour chaque titre lieu-tarif-contact on peut avoir plusieurs éléments, séparés par des « / »
- Si les éléments sont longs ils s'afficheront sur plusieurs lignes
- Les adresses mail ouvriront un nouveau mail en les touchant et les numéro de téléphone appelleront le numéro en les touchant
- Les partages seront à définir de la même manière que pour le partage d'un article (Facebook)
- Le bandeau de pub défile avec l'écran
- Le titre est constitué comme suit : **Nom_de_commune+point**[espace]titre_événement (idem dans la liste de l'écran précédent)

Erreur maquette : Pas de tab bar dans cet écran, mais bien le bouton partager, identique à celui présent dans un article classique