ListFragment. Основы

Самое распространённое использование фрагментов - списки и связанное с ним содержание. При использовании списков на планшете в альбомной ориентации справа оставалось слишком много пустого пространства. Фрагменты позволяют использовать данное пространство с пользой.

Представим себе ситуацию - у нас есть список **ListView**, содержащий ссылки. При щелчке на одной из ссылок мы открываем вторую активность, состоящую из какого-нибудь компонента: **TextView** или **WebView**. По сути, один экран у нас сменяется другим. Можно реализовать эту задачу по другому. На планшетах много места. Почему бы не расположить **ListView** и **TextView** на одном экране рядышком? И когда пользователь будет щёлкать слева на элементе списка, то в правой части будет обновляться содержимое **TextView**. Такой подход нам знаком, например, при чтении электронных писем - слева список писем, а справа - содержание выбранного письма.

Для связывания данных используются адаптеры **ListAdapter**, **ArrayAdapter**, **SimpleAdapter**, **SimpleCursorAdapter** и т.д. Подключение следует производить в методе **onActivityCreated()**.

Стандартная разметка подгружается автоматически, для собственной разметки используйте метод **onCreateView()**.

Начнём с простых примеров. Мы знаем, что для создания списка используется компонент **ListView**. Если наш экран должен состоять только из списка, то можно использовать готовую активность [ListActivity](http://developer.alexanderklimov.ru/android/listactivity.php), в которой уже встроен список и реализованы необходимые методы.

**ListFragment** работает по такому же принципу. По сути это обычный фрагмент, в который встроили **ListView**, избавив нас от написания лишнего кода.

Если изучить исходники фрагмента, то можно встретить следующие строки кода, которые могут пригодиться.

View mListContainer; // родительский контейнер (android.R.id.listContainer)

ListAdapter mAdapter; // адаптер списка

ListView mList; // список (android.R.id.list)

TextView mEmptyView; // текстовое поле для пустого списка (android.R.id.empty)

View mProgressContainer; // компонент для показа анимации загрузки

Как правило, **ListFragment** используют в паре с другим обычным фрагментом. А пока мы попробуем обойтись одним фрагментом. Создайте новый проект или используйте уже готовый проект и добавьте новый класс, который наследуется от **ListFragment**. Назовём новый класс **SingleListFragment**:

import android.support.v4.app.ListFragment;

public class SingleListFragment extends ListFragment {

}

Как и с обычными фрагментами, списочный фрагмент может быть из библиотеки поддержки или из обычных классов Android.

Посмотрим, как выглядит данный фрагмент по умолчанию. Мы не будем создавать свою разметку и свой адаптер, так как списочный фрагмент уже включает в себя необходимые компоненты.

Разместим фрагмент в разметке основной активности **MainActivity**. У нас это файл **activity\_main.xml**:

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context=".MainActivity">

<fragment

android:id="@+id/listFragment"

android:name="ru.alexanderklimov.as21.SingleListFragment"

android:layout\_width="match\_parent"

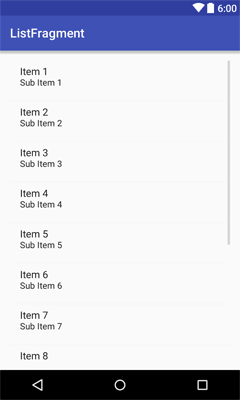
android:layout\_height="match\_parent"

tools:layout="@android:layout/list\_content">

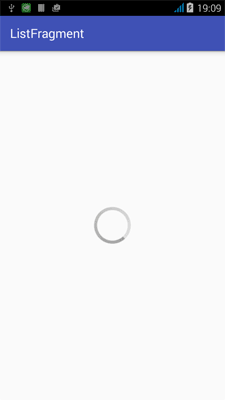
</fragment>

</LinearLayout>

В атрибуте **android:name** вы указываете полное имя класса вашего фрагмента. Вы можете переключиться в графический режим и из контекстного меню выбрать пункт **list\_content**, чтобы увидеть экран со списком. Ничего необычного и интересного.



Запустим проект и посмотрим, что получилось. На экране мы увидим белую страницу с индикатором прогресса. Списком здесь и не пахнет. Впрочем, это не удивительно, так как мы не подготовили данных для списка.



Не будем ничего выдумывать, а просто скопируем массив строк из урока про [ListView](http://developer.alexanderklimov.ru/android/views/listview.php) и вставим его в класс фрагмента. А в методе **onActivityCreated()** свяжем массив с адаптером и передадим его списочному фрагменту.

package ru.alexanderklimov.as21;

import android.os.Bundle;

import android.support.v4.app.ListFragment;

import android.widget.ArrayAdapter;

public class SingleListFragment extends ListFragment {

// определяем массив типа String

final String[] catNames = new String[]{"Рыжик", "Барсик", "Мурзик",

"Мурка", "Васька", "Томасина", "Кристина", "Пушок", "Дымка",

"Кузя", "Китти", "Масяня", "Симба"};

@Override

public void onActivityCreated(Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

ListAdapter adapter = new ArrayAdapter<>(getActivity(),

android.R.layout.simple\_list\_item\_1, catNames);

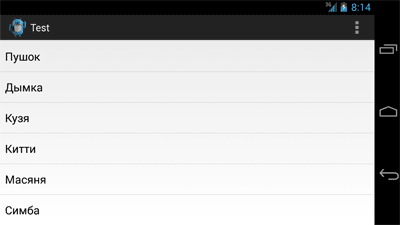
setListAdapter(adapter);

}

}

Мы используем метод **onActivityCreated()**, так как именно здесь можно быть уверенным, что все необходимые компоненты фрагмента загрузились и фрагмент готов к использованию в составе активности. Метод **onCreateView()** в данном случае использовать не обязательно. В остальном код идентичен с кодом для **ListActivity** - массив, адаптер, связывание массива с адаптером.

Запускаем проект и видим список с именами котов.



Если вы хотите видеть выбранный элемент постоянно активным, то используйте другую системную разметку **simple\_list\_item\_activated\_1**.

Если нужна своя разметка для списка, то поступаем точно также, как в уроке с [ListActivity](http://developer.alexanderklimov.ru/android/listactivity.php" \l "custom). Создаём в папке **res/layout** новый файл, скажем **listfragment.xml** и размещаем нужные элементы:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical" >

<ListView

android:id="@id/android:list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#00FF00" >

</ListView>

<TextView

android:id="@id/android:empty"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center"

android:text="@string/empty" >

</TextView>

</LinearLayout>

В шаблоне нужно разместить **ListView** с обязательным идентификатором **@id/android:list**. Компонент **TextView** будет показан в том случае, если нет данных для списка. Он также должен иметь обязательный идентификатор **@id/android:empty**. Помните, в начале статьи я приводил исходник системного фрагмента?

Разметка подключается в методе **onCreateView()**:

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

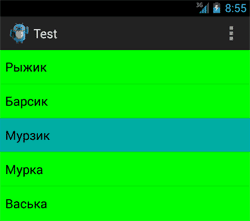
Bundle savedInstanceState) {

return inflater.inflate(R.layout.listfragment, null);

}

Запустив проект, мы увидим список с зелёным фоном. Если отключить адаптер, то можно увидеть уже свой компонент **TextView** со своим текстом, который вы определили в ресурсе **@string/empty**.

Если в **ListView** вы добавите атрибут **android:choiceMode="singleChoice"** для одиночного выбора, то в адаптере рекомендую использовать другой ресурс **android.R.layout.simple\_list\_item\_activated\_1**. В этом случае выбранный элемента списка будет иметь другой цвет, что позволит быстро определять выделенный элемент.



Нажатия на элементах списка

Определять нажатия на отдельных элементах списка можно через метод фрагмента **onListItemClick()**:

@Override

public void onListItemClick(ListView l, View v, int position, long id) {

super.onListItemClick(l, v, position, id);

Toast.makeText(getActivity(), "Вы выбрали позицию: " + position, Toast.LENGTH\_SHORT).show();

}

Если вас интересует текст выбранного элемента:

@Override

public void onListItemClick(ListView l, View v, int position, long id) {

super.onListItemClick(l, v, position, id);

TextView textView = (TextView) v;

String itemText = textView.getText().toString(); // получаем текст нажатого элемента

Toast.makeText(getActivity(), "Вы выбрали " + itemText, Toast.LENGTH\_SHORT).show();

}

Если вы хотите настраивать внешний вид каждого элемента, например, разместить значок, то опять ничего нового здесь нет. Нужно создать свою разметку для элемента списка и написать свой адаптер. Простой пример. Создадим новую разметку **res/layout/listfragment\_row.xml**:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal" >

<ImageView

android:id="@+id/imageViewIcon"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

<TextView

android:id="@+id/textViewName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

</LinearLayout>

В классе фрагмента прописываем свой адаптер. В методе **onActivityCreated()** присоединяем свой адаптер, а в методе **onListItemClick()** меняем код.

package ru.alexanderklimov.as21;

import android.content.Context;

import android.os.Bundle;

import android.support.v4.app.ListFragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ArrayAdapter;

import android.widget.ImageView;

import android.widget.ListAdapter;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

public class SingleListFragment extends ListFragment {

// определяем массив типа String

final String[] catNames = new String[]{"Рыжик", "Барсик", "Мурзик",

"Мурка", "Васька", "Томасина", "Кристина", "Пушок", "Дымка",

"Кузя", "Китти", "Масяня", "Симба"};

@Override

public void onActivityCreated(Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

MyListAdapter myListAdapter = new MyListAdapter(getActivity(),

R.layout.listfragment\_row, catNames);

setListAdapter(myListAdapter);

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

return inflater.inflate(R.layout.listfragment, null);

}

@Override

public void onListItemClick(ListView l, View v, int position, long id) {

super.onListItemClick(l, v, position, id);

Toast.makeText(getActivity(),

getListView().getItemAtPosition(position).toString(),

Toast.LENGTH\_LONG).show();

}

public class MyListAdapter extends ArrayAdapter<String> {

private Context mContext;

public MyListAdapter(Context context, int textViewResourceId,

String[] objects) {

super(context, textViewResourceId, objects);

mContext = context;

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

// return super.getView(position, convertView, parent);

LayoutInflater inflater = (LayoutInflater) mContext

.getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);

View row = inflater.inflate(R.layout.listfragment\_row, parent,

false);

TextView catNameTextView = (TextView) row.findViewById(R.id.textViewName);

catNameTextView.setText(catNames[position]);

ImageView iconImageView = (ImageView) row.findViewById(R.id.imageViewIcon);

// Присваиваем значок

iconImageView.setImageResource(R.drawable.ic\_launcher\_cat);

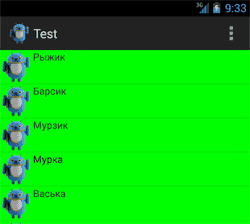
return row;

}

}

}

Получаем результат:



Множественный выбор

Вы можете использовать список с множественным выбором. Модификация минимальна, просто установите нужный режим у **ListView**.

listfragment.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:paddingLeft="8dp"

android:paddingRight="8dp" >

<ListView

android:id="@id/android:list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:choiceMode="multipleChoice"

android:drawSelectorOnTop="false" />

<TextView

android:id="@id/android:empty"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:text="Нет данных" />

</LinearLayout>

В классе фрагмента напишем код, заменив системную разметку на **simple\_list\_item\_multiple\_choice**.

package ru.alexanderklimov.as21;

import android.content.Context;

import android.os.Bundle;

import android.support.v4.app.ListFragment;

import android.util.SparseBooleanArray;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ArrayAdapter;

import android.widget.ImageView;

import android.widget.ListAdapter;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

public class SingleListFragment extends ListFragment {

// определяем массив типа String

final String[] catNames = new String[]{"Рыжик", "Барсик", "Мурзик",

"Мурка", "Васька", "Томасина", "Кристина", "Пушок", "Дымка",

"Кузя", "Китти", "Масяня", "Симба"};

@Override

public void onActivityCreated(Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

ListAdapter adapter = new ArrayAdapter(getActivity(),

android.R.layout.simple\_list\_item\_multiple\_choice, catNames);

setListAdapter(adapter);

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

return inflater.inflate(R.layout.listfragment, null);

}

@Override

public void onListItemClick(ListView l, View v, int position, long id) {

super.onListItemClick(l, v, position, id);

String prompt = "Вы выбрали: "

+ getListView().getItemAtPosition(position).toString() + "\n";

prompt += "Выбранные элементы: \n";

int count = getListView().getCount();

SparseBooleanArray sparseBooleanArray = getListView()

.getCheckedItemPositions();

for (int i = 0; i < count; i++) {

if (sparseBooleanArray.get(i)) {

prompt += getListView().getItemAtPosition(i).toString() + "\n";

}

}

Toast.makeText(getActivity(), prompt, Toast.LENGTH\_LONG).show();

}

}

