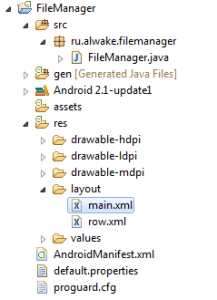
Файловый менеджер для Android своими руками

Пошагово разберём, как создать свой собственный несложный файловый менеджер. Конечно же, примеры использованного кода будут полезны и в чисто обучающих целях для новичков в Android-программировании.

Итак, перед нами стоит цель – обеспечить минимальный функционал файлового менеджера, как-то – базовая навигация по каталогам, то есть даже без доступа к папкам, на которые имеет права root. Также нужно отображать в верхней части экрана текущее расположение в файловой системе.

Разберем структуру проекта:

[](http://fans-android.com/wp-content/uploads/2012/11/proj_structure.png)

*/res/drawable-\*dpi*– каталоги с ресурсами, которые предназначены для устройств с разной разрешающей способностью экрана. На данном этапе здесь содержится одинокий файл *icon.png*– иконка для приложения.

*/res/layout*– директория содержит xml-файлы, которые описывают структуру и внешний вид форм и их элементов. Сразу после создания проекта, в данной папке находится файл *main.xml*, также создаем файл *row.xml*, который будет описывать внешний вид рядов – элементов списка в файловом дереве нашего файлового менеджера.

*/res/values* – в данной папке размещены любые константы, которые могут быть применены для работы проекта.

Редактировать .xml-файлы можно в визуальном и текстовом режимах, причем второй предпочтительнее. А поэтому для редактирования кода нажимаем правой кнопкой мышки на необходимом файле .xml в Package Explorer и выбираем пункт «Open with» -> «Text Editor».

Файл *FileManager.java* содержит основной класс для основной формы приложения. Здесь же будет находится весь написанный нами код.

*AndroidManifest.xml*– содержит все основные свойства проекта, в том числе заданные при его создании (например, название). Так что при необходимости изменить название проекта в процессе разработки приложения редактировать нужно именно этот файл.

Приступим к написанию кода, по ходу дела разбираясь, за что он отвечает. Файл *main.xml*:

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

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

android:orientation="vertical"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent">

<TableRow>

<TextView android:id="@+id/titleManager"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:padding="5dip"

/>

</TableRow>

<TableRow>

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

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_weight="2"

android:drawSelectorOnTop="false"/>

</TableRow>

</TableLayout>

В данном коде задана разметка для основного Layout`а. TableLayout означает, что элементы будут выстроены в виде таблицы. В верхней ячейке этой таблицы находится элемент TextView, то есть текстовое поле, а в нижней – ListView, то есть список. Оба эти элемента имеют id, используя которые, можно изменять их содержимое. К примеру, с использованием R.id.titleManager для текстового поля TextView.

Файл *row.xml*:

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

<TextView

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

android:layout\_width="fill\_parent"

android:layout\_height="40sp"

android:padding="5dip"

android:gravity="center\_vertical"

/>

В данных строках задана разметка для каждого элемента ListView, то есть для каждой папки и файла. В коде указана ширина элементов, высота, отступы (padding) и выравнивание – центрирование по вертикали (center\_vertical).

Файл *FileManager.java*:

package ru.alwake.filemanager;

import java.io.File;

import java.util.ArrayList;

import java.util.List;

import android.app.AlertDialog;

import android.app.ListActivity;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.DialogInterface.OnClickListener;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import android.widget.TextView;

public class FileManager extends ListActivity {

private List<String> directoryEntries = new ArrayList<String>();

private File currentDirectory = new File("/");

//Старт приложения

@Override

public void onCreate(Bundle icicle) {

super.onCreate(icicle);

//set main layout

setContentView(R.layout.main);

//browse to root directory

browseTo(new File("/"));

}

//browse to parent directory

private void upOneLevel(){

if(this.currentDirectory.getParent() != null) {

this.browseTo(this.currentDirectory.getParentFile());

}

}

//browse to file or directory

private void browseTo(final File aDirectory){

//if we want to browse directory

if (aDirectory.isDirectory()){

//fill list with files from this directory

this.currentDirectory = aDirectory;

fill(aDirectory.listFiles());

//set titleManager text

TextView titleManager = (TextView) findViewById(R.id.titleManager);

titleManager.setText(aDirectory.getAbsolutePath());

} else {

//if we want to open file, show this dialog:

//listener when YES button clicked

OnClickListener okButtonListener = new OnClickListener(){

public void onClick(DialogInterface arg0, int arg1) {

//intent to navigate file

Intent i = new Intent(android.content.Intent.ACTION\_VIEW, Uri.parse("file://" + aDirectory.getAbsolutePath()));

//start this activity

startActivity(i);

}

};

//listener when NO button clicked

OnClickListener cancelButtonListener = new OnClickListener() {

public void onClick(DialogInterface arg0, int arg1) {

//do nothing

//or add something you want

}

};

//create dialog

new AlertDialog.Builder(this)

.setTitle("Подтверждение") //title

.setMessage("Хотите открыть файл "+ aDirectory.getName() + "?") //message

.setPositiveButton("Да", okButtonListener) //positive button

.setNegativeButton("Нет", cancelButtonListener) //negative button

.show(); //show dialog

}

}

//fill list

private void fill(File[] files) {

//clear list

this.directoryEntries.clear();

if (this.currentDirectory.getParent() != null)

this.directoryEntries.add("..");

//add every file into list

for (File file : files) {

this.directoryEntries.add(file.getAbsolutePath());

}

//create array adapter to show everything

ArrayAdapter<String> directoryList = new ArrayAdapter<String>(

this, R.layout.row, this.directoryEntries);

this.setListAdapter(directoryList);

}

//when you clicked onto item

@Override

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

//get selected file name

int selectionRowID = position;

String selectedFileString =

this.directoryEntries.get(selectionRowID);

//if we select ".." then go upper

if(selectedFileString.equals("..")){

this.upOneLevel();

} else {

//browse to clicked file or directory using browseTo()

File clickedFile = null;

clickedFile = new File(selectedFileString);

if (clickedFile != null)

this.browseTo(clickedFile);

}

}

}

В начале указано имя пакета (Package Name). Строки со 2й по 18ю отвечают за импорт необходимых для работы приложения библиотек. Eclipse умеет производить подгрузку и импорт нужных библиотек автоматически, если встречает что-либо неизвестное. Вообще, в выше изложенном коде всего пять наглядных функций, это и есть скелет приложения, который обеспечивает базовую навигацию по файловой системе Android-девайса. Единственная оговорка – при попытке открыть директорию, для доступа к которой нужны root-права, приложение выдаст ошибку и будет закрыто.