Создаем файловый менеджер

О нашем проекте

Для начала определимся, что мы хотим от нашего файлового менеджера. В идеале — полную замену ASTRO и eStrongs. Ну, а пока нам требуется обеспечить базовую навигацию по каталогам, причем под «базовой» мы понимаем, что нам не нужно заходить в те папки, в которые имеет доступ только root. Помимо этого, в поле сверху у нас будет отображаться наше текущее расположение в структуре каталогов.  
  
Начинаем писать код

**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>  
  
\* This source code was highlighted with Source Code Highlighter.

Здесь у нас задается разметка для нашего основного Layout'а формы. TableLayout здесь значит, что элементы у нас выстроены в виде таблицы. Далее в верхней ячейке таблицы размещается элемент TextView (текстовое поле), а в нижней ячейке — ListView (список). Оба элемента имеют id, используя который, мы можем изменять содержимое элементов. Например, используя R.id.titleManager для нашего 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"  
/>  
  
\* This source code was highlighted with Source Code Highlighter.

Здесь мы задаем разметку для каждого элемента нашего ListView, то есть непосредственно для каждой отдельно взятой папки или каждого файла. В данном коде у нас задается ширина каждого элемента, высота, отступ (padding) и выравнивание center\_vertical — то есть центрование по вертикали.  
  
**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("/");

  //when application started

  @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);

    }

  }

}

\* This source code was highlighted with Source Code Highlighter.

В начале указывается название пакета (package name).  
Строки 2-18 отвечают за импорт нужных нам библиотек. Важно отметить, что импорт библиотек Eclipse может производить автоматически, как только встретит что-нибудь неизвестное.  
В данном коде у нас всего 5 достаточно очевидных функций, в которых несложно разобраться. И это — скелет приложения, обеспечивающий основную навигацию по файловой структуре. Однако у данного приложения существует одна проблема — при попытке зайти в директорию, доступ к которой разрешен только суперпользователю, мы получаем ошибку и приложение завершает работу.  
Соответственно, в следующий раз мы подумаем как избавиться от этой ошибки, а также о том, как сделать отображение иконок, соответствующих типу файла. Кроме того, мы подумаем о реализации функции Copy-Paste в нашем файловом менеджере.