

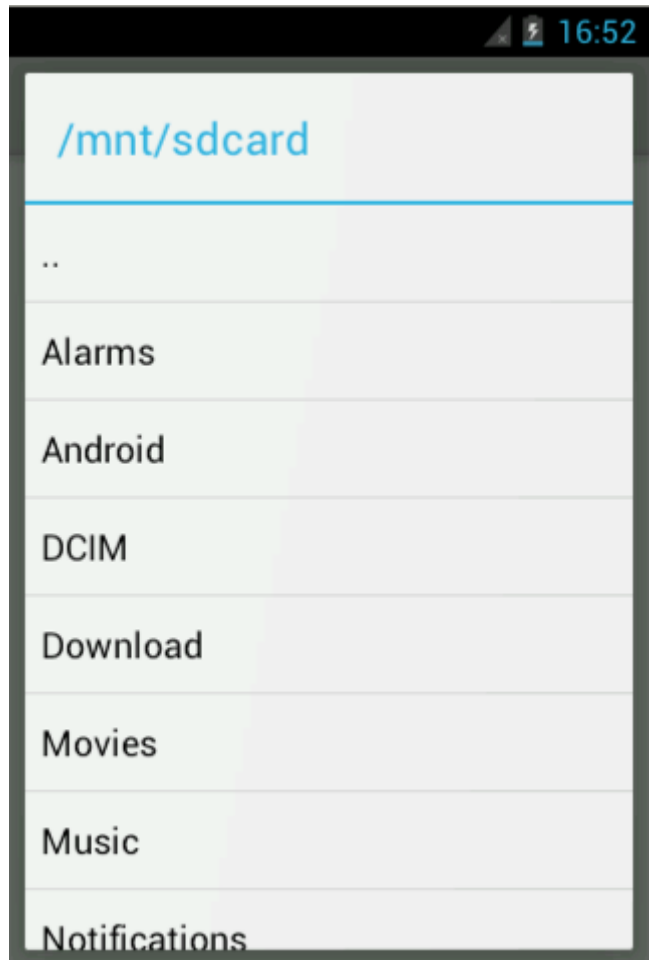
Simple Android File Chooser

By [Roger Keays](#), 3 June 2015

Surprisingly, the Android API doesn't include a file chooser. I'm not sure why that is, I guess the developers don't want to make the assumption that the device has a user filesystem available. Anyway, it's not my job to speculate on what the Android developers talk about over lunch. I just need a file chooser.

It seems like there are a few options out there but I wanted something uber simple and ultra light. It's just for users to select an import file.

My solution is a single Java class utilising regular android Dialog and ListViews. It just wires up the event handlers to read from the disk and refresh the list. You provide a FileSelectedListener implementation to handle the file selection event. Menial stuff that you don't want to bother implementing yourself. It can be used like this:



```
new FileChooser(activity).setFileListener(new FileSelectedListener() {  
    @Override public void fileSelected(final File file) {  
        // do something with the file  
    }).showDialog();
```

Here is the FileChooser class which you can cut and paste into your project. Public domain code so do what you want with it.

Peace.

```
package au.com.ninthavenue.patterns.android.dialogs;
```

```
import android.app.Activity;  
import android.app.Dialog;  
import android.os.Environment;  
import android.view.View;
```

```

import android.view.ViewGroup;
import android.view.WindowManager.LayoutParams;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import java.io.File;
import java.io.FileFilter;
import java.util.Arrays;

public class FileChooser {
    private static final String PARENT_DIR = "..";

    private final Activity activity;
    private ListView list;
    private Dialog dialog;
    private File currentPath;

    // filter on file extension
    private String extension = null;
    public void setExtension(String extension) {
        this.extension = (extension == null) ? null :
            extension.toLowerCase();
    }

    // file selection event handling
    public interface FileSelectedListener {
        void fileSelected(File file);
    }
    public FileChooser setFileListener(FileSelectedListener fileListener) {
        this.fileListener = fileListener;
        return this;
    }
    private FileSelectedListener fileListener;

    public FileChooser(Activity activity) {
        this.activity = activity;
        dialog = new Dialog(activity);
        list = new ListView(activity);
        list.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override public void onItemClick(AdapterView<?> parent, View view,

```

```

        String fileChosen = (String) list.getItemAtPosition(which);
        File chosenFile = getChosenFile(fileChosen);
        if (chosenFile.isDirectory()) {
            refresh(chosenFile);
        } else {
            if (fileListener != null) {
                fileListener.fileSelected(chosenFile);
            }
            dialog.dismiss();
        }
    }
});
dialog.setContentView(list);
dialog.getWindow().setLayout(LayoutParams.FILL_PARENT, LayoutParams.FILL_PARENT);
refresh(Environment.getExternalStorageDirectory());
}

public void showDialog() {
    dialog.show();
}

/**
 * Sort, filter and display the files for the given path.
 */
private void refresh(File path) {
    this.currentPath = path;
    if (path.exists()) {
        File[] dirs = path.listFiles(new FileFilter() {
            @Override public boolean accept(File file) {
                return (file.isDirectory() && file.canRead());
            }
        });
        File[] files = path.listFiles(new FileFilter() {
            @Override public boolean accept(File file) {
                if (!file.isDirectory()) {
                    if (!file.canRead()) {
                        return false;
                    }
                } else if (extension == null) {
                    return true;
                } else {

```

```

        return file.getName().toLowerCase().endsWith(extens
    }
    } else {
        return false;
    }
}
});

// convert to an array
int i = 0;
String[] fileList;
if (path.getParentFile() == null) {
    fileList = new String[dirs.length + files.length];
} else {
    fileList = new String[dirs.length + files.length + 1];
    fileList[i++] = PARENT_DIR;
}
Arrays.sort(dirs);
Arrays.sort(files);
for (File dir : dirs) { fileList[i++] = dir.getName(); }
for (File file : files ) { fileList[i++] = file.getName(); }

// refresh the user interface
dialog.setTitle(currentPath.getPath());
list.setAdapter(new ArrayAdapter(activity,
    android.R.layout.simple_list_item_1, fileList) {
    @Override public View getView(int pos, View view, ViewG
        view = super.getView(pos, view, parent);
        ((TextView) view).setSingleLine(true);
        return view;
    }
});
}

}

/**
 * Convert a relative filename into an actual File object.
 */
private File getChosenFile(String fileChosen) {
    if (fileChosen.equals(PARENT_DIR)) {

```

```
        return currentPath.getParentFile();
    } else {
        return new File(currentPath, fileChosen);
    }
}
```

About Roger Keays



Roger Keays is an artist, an engineer, and a student of life. He has no fixed address and has left footprints on 40-something different countries around the world. Roger is addicted to surfing. His other interests are music, psychology, languages, the proper use of semicolons, and finding good food.