**The Last Stage of iBookmarks**

The iBookmark program now lets you edit new URL’s as well as preview them. It also includes a CAB (Context Action Bar) to delete and edit bookmark items. We could consider the app finished, even though we could still apply a number of solution. In particular, we should enable the “Edit” feature, perhaps re-using the BookmarkItemEdit activity if possible. Anyway, here is the complete source code, as usual. Note in particular the activity\_bookmark\_item.xml layout for a sophisticated layout creating borderless buttons.

**AndroidManifest.xml**

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

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

package=*"org.mathcs.ibookmark"*

android:versionCode=*"1"*

android:versionName=*"1.0"* >

<uses-sdk

android:minSdkVersion=*"11"*

android:targetSdkVersion=*"18"* />

<uses-permission android:name=*"android.permission.INTERNET"*/>

<application

android:allowBackup=*"true"*

android:icon=*"@drawable/ic\_launcher"*

android:label=*"@string/app\_name"*

android:theme=*"@style/AppTheme"* >

<activity

android:name=*"org.mathcs.ibookmark.BookmarkActivity"*

android:label=*"@string/app\_name"* >

<intent-filter>

<action android:name=*"android.intent.action.MAIN"* />

<category android:name=*"android.intent.category.LAUNCHER"* />

</intent-filter>

</activity>

<activity

android:name=*"UrlViewer"*

android:parentActivityName=*"BookmarkActivity"* >

<!-- Parent activity meta-data to support 4.0 and lower -->

<meta-data

android:name=*"android.support.PARENT\_ACTIVITY"*

android:value=*"com.example.myfirstapp.MainActivity"* />

</activity>

<activity

android:name=*"BookmarkItemEdit"*

android:parentActivityName=*"BookmarkActivity"*>

<!-- Parent activity meta-data to support 4.0 and lower -->

<meta-data

android:name=*"android.support.PARENT\_ACTIVITY"*

android:value=*"com.example.myfirstapp.MainActivity"* />

</activity>

</application>

</manifest>

**strings.xml**

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

<resources>

<string name=*"app\_name"*>iBookmark</string>

<string name=*"action\_settings"*>Settings</string>

<string name=*"hint\_title"*>Title</string>

<string name=*"hint\_http"*>http://</string>

<string name=*"text\_preview"*>Preview</string>

<string name=*"menu\_add"*>Add URL</string>

<string name=*"menu\_del"*>Remove URL</string>

<string name=*"menu\_edit"*>Edit URL</string>

</resources>

**bookmark.xml (Menu)**

<menu xmlns:android=*"http://schemas.android.com/apk/res/android"* >

<item

android:id=*"@+id/action\_settings"*

android:orderInCategory=*"100"*

android:showAsAction=*"never"*

android:title=*"@string/action\_settings"*/>

<item

android:id=*"@+id/id\_menu\_add"*

android:showAsAction=*"ifRoom|withText"*

android:title=*"@string/menu\_add"*

android:icon=*"@android:drawable/ic\_menu\_add"*>

</item>

</menu>

**context.xml (Menu)**

<menu xmlns:android=*"http://schemas.android.com/apk/res/android"* >

<item

android:id=*"@+id/id\_menu\_del"*

android:showAsAction=*"ifRoom|withText"*

android:title=*"@string/menu\_del"*

android:icon=*"@android:drawable/ic\_menu\_delete"*>

</item>

<item

android:id=*"@+id/id\_menu\_edit"*

android:showAsAction=*"ifRoom|withText"*

android:title=*"@string/menu\_edit"*

android:icon=*"@android:drawable/ic\_menu\_edit"*>

</item>

</menu>

**activity\_bookmark.xml (Layout)**

<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=*".BookmarkActivity"* >

<ListView

android:id=*"@+id/id\_list"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_weight=*"1"* >

</ListView>

</LinearLayout>

**activity\_bookmark\_edit.xml (Layout)**

<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=*".BookmarkActivity"* >

<EditText

android:id=*"@+id/id\_input\_title"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:hint=*"@string/hint\_title"* />

<EditText

android:id=*"@+id/id\_input\_url"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"wrap\_content"*

android:hint=*"@string/hint\_http"*

android:inputType=*"textUri"* />

<RelativeLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"48dp"*>

<View

android:layout\_width=*"match\_parent"*

android:layout\_height=*"1dip"*

android:layout\_marginLeft=*"4dip"*

android:layout\_marginRight=*"4dip"*

android:background=*"?android:attr/dividerVertical"*

android:layout\_alignParentTop=*"true"*/>

<View

android:id=*"@+id/LayoutHelper"*

android:layout\_width=*"1dip"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentTop=*"true"*

android:layout\_alignParentBottom=*"true"*

android:layout\_marginBottom=*"4dip"*

android:layout\_marginTop=*"4dip"*

android:background=*"?android:attr/dividerVertical"*

android:layout\_centerHorizontal=*"true"*/>

<Button

android:id=*"@+id/id\_button\_preview"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentBottom=*"true"*

android:layout\_alignParentLeft=*"true"*

android:layout\_alignParentTop=*"true"*

android:layout\_toLeftOf=*"@id/LayoutHelper"*

android:background=*"?android:attr/selectableItemBackground"*

android:text=*"@string/text\_preview"* />

<Button

android:id=*"@+id/id\_button\_okay\_itemedit"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"match\_parent"*

android:layout\_alignParentRight=*"true"*

android:layout\_alignParentTop=*"true"*

android:background=*"?android:attr/selectableItemBackground"*

android:text=*"@android:string/ok"*

android:layout\_alignParentBottom=*"true"*

android:layout\_toRightOf=*"@id/LayoutHelper"*/>

</RelativeLayout>

<WebView

android:id=*"@+id/id\_webview\_itemedit"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"* />

</LinearLayout>

**Bookmark.java**

package org.mathcs.ibookmark;

/\*\*

\* Class to define a "bookmark". By definition it is comprised of a "title", a "url", and

\* a state of being selected or not.

\*

\* @author Bert Wachsmuth

\* @version 1.0 (10/2013)

\*/

import java.io.BufferedReader;

import java.io.IOException;

import java.io.PrintWriter;

public class Bookmark

{

private String url;

private String title;

private boolean isSelected;

public Bookmark(String title, String url)

{

super();

this.url = url;

this.title = title;

isSelected = false;

}

public Bookmark(BufferedReader in) throws IOException

{

this(in.readLine(), in.readLine());

}

public String getUrl()

{

return url;

}

public void setUrl(String url)

{

this.url = url;

}

public String getTitle()

{

return title;

}

public void setTitle(String title)

{

this.title = title;

}

public boolean isSelected()

{

return isSelected;

}

public void setSelected(boolean isSelected)

{

this.isSelected = isSelected;

}

public void save(PrintWriter out)

{

out.println(title);

out.println(url);

}

public String toString()

{

return title + "[" + url + "]";

}

}

**BookmarkView.java**

package org.mathcs.ibookmark;

/\*\*

\* Defines how a bookmark is rendered in a list, i.e. how it is represented visually.

\* Serves as template to all items in a list of bookmarks.

\*

\* @author Bert Wachsmuth

\* @version 1.0 (10/2013)

\*/

import android.content.Context;

import android.graphics.Color;

import android.widget.LinearLayout;

import android.widget.TextView;

public class BookmarkView extends LinearLayout

{

private TextView url;

private TextView title;

public BookmarkView(Context context)

{

super(context);

this.url = new TextView(context);

this.title = new TextView(context);

setOrientation(LinearLayout.VERTICAL);

addView(title);

addView(url);

}

public void setData(Bookmark item)

{

title.setText(item.getTitle());

url.setText(item.getUrl());

if (item.isSelected())

{

title.setBackgroundColor(Color.GRAY);

url.setBackgroundColor(Color.GRAY);

}

else

{

title.setBackgroundColor(Color.WHITE);

url.setBackgroundColor(Color.WHITE);

}

}

}

**BookmarkAdapter.java**

package org.mathcs.ibookmark;

/\*\*

\* Class to contain a list of bookmarks that mediates between the list and the

\* view onto the list (a ListView). Provides methods to add, delete, and edit

\* members of the list but has no UI.

\*

\* @author Bert Wachsmuth

\* @version 1.0 (10/2013)

\*/

import java.io.BufferedReader;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

import android.content.Context;

import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

public class BookmarkAdapter extends BaseAdapter

{

private ArrayList<Bookmark> data;

private Context context;

private boolean isDirty;

public BookmarkAdapter(Context context)

{

super();

this.context = context;

this.data = new ArrayList<Bookmark>();

this.isDirty = false;

}

public void add(Bookmark item)

{

data.add(item);

isDirty = true;

}

public Bookmark get(int position)

{

return (Bookmark)data.get(position);

}

public void remove(int position)

{

data.remove(position);

isDirty = true;

}

@Override

public int getCount()

{

return data.size();

}

@Override

public Object getItem(int position)

{

return data.get(position);

}

@Override

public long getItemId(int position)

{

return 0;

}

@Override

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

{

BookmarkView view = null;

if (convertView != null)

view = (BookmarkView)convertView;

else

view = new BookmarkView(context);

view.setData(data.get(position));

return view;

}

public boolean isDirty()

{

return isDirty;

}

public void saveBookmarks(PrintWriter out) throws IOException

{

out.println(data.size());

for (int i = 0; i < data.size(); i++)

data.get(i).save(out);

isDirty = false;

}

public void loadBookmarks(BufferedReader in) throws IOException

{

data.clear();

int size = Integer.parseInt(in.readLine());

for (int i = 0; i < size; i++)

data.add(new Bookmark(in));

isDirty = false;

}

public void deselectAll()

{

for (int i = 0; i < data.size(); i++)

data.get(i).setSelected(false);

}

}

**BookmarkActivity.java**

package org.mathcs.ibookmark;

/\*\*

\* Main activity of this app. Shows a list of bookmarks and lets you add, delete, or

\* edit them. You can also tap on a bookmark in the list to view the corresponding web

\* page. The list is saved to the local device inside the app directory; the data file

\* is a text file as follows:

\*

\* line 1: version info (a string)

\* line 2: number of bookmarks saved (int)

\* line 3: title of 1st bookmark (string)

\* line 4: url of 1st bookmark (string)

\* line 5: title of 2nd bookmark (string)

\* line 6: url of 2nd bookmark (string)

\* ...

\*

\* @author Bert Wachsmuth

\* @version 1.0 (10/2013)

\*/

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.io.PrintWriter;

import android.os.Bundle;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.view.ActionMode;

import android.view.Menu;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.view.View;

import android.widget.AbsListView;

import android.widget.AdapterView;

import android.widget.ListView;

import android.widget.Toast;

public class BookmarkActivity extends Activity

{

public final static String KEY\_URL = "url";

public final static String KEY\_TITLE = "title";

public final static String DEFAULT\_URL = "www.shu.edu";

public final static String BOOKMARK\_FILE\_NAME = "bookmarks.txt";

public final static int ITEM\_EDIT\_REQUEST\_CODE = 0;

private BookmarkAdapter data;

private ListView list;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_bookmark);

data = new BookmarkAdapter(this);

list = (ListView) this.findViewById(R.id.id\_list);

list.setAdapter(data);

list.setChoiceMode(ListView.CHOICE\_MODE\_MULTIPLE\_MODAL);

list.setDrawSelectorOnTop(true);

list.setOnItemClickListener(new AdapterView.OnItemClickListener()

{

@Override

public void onItemClick(AdapterView<?> adapter, View view, int pos,

long arg3)

{

startBookmark(pos);

}

});

list.setMultiChoiceModeListener(new AbsListView.MultiChoiceModeListener()

{

@Override

public boolean onCreateActionMode(ActionMode mode, Menu menu)

{

// Inflate the menu for the CAB

MenuInflater inflater = mode.getMenuInflater();

inflater.inflate(R.menu.context, menu);

return true;

}

@Override

public void onItemCheckedStateChanged(ActionMode mode, int position,

long id, boolean checked)

{

// Here you can do something when items are selected/de-selected,

// such as update the title in the CAB

data.get(position).setSelected(checked);

data.notifyDataSetChanged();

mode.setTitle(list.getCheckedItemCount() + " selected");

}

@Override

public boolean onActionItemClicked(ActionMode mode, MenuItem item)

{

// Respond to clicks on the actions in the CAB

if (item.getItemId() == R.id.id\_menu\_del)

deleteSelectedItems();

else if (item.getItemId() == R.id.id\_menu\_edit)

editSelectedItem();

mode.finish();

return false;

}

@Override

public void onDestroyActionMode(ActionMode mode)

{

// can make necessary updates to the activity when CAB is removed

data.deselectAll();

data.notifyDataSetChanged();

}

@Override

public boolean onPrepareActionMode(ActionMode mode, Menu menu)

{

// You can perform updates to the CAB due to invalidate() request

return false;

}

});

}

@Override

public boolean onCreateOptionsMenu(Menu menu)

{

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.bookmark, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item)

{

if (item.getItemId() == R.id.id\_menu\_add)

addItem();

return super.onOptionsItemSelected(item);

}

@Override

public void onPause()

{

super.onPause();

if ((data != null) && (data.isDirty()))

{

saveData();

}

}

@Override

public void onResume()

{

super.onResume();

loadData();

}

@Override

public void onActivityResult(int requestCode, int resultCode, Intent intent)

{

super.onActivityResult(requestCode, resultCode, intent);

// requestCode is ignored since we only use one. But resultCode is used!

if (resultCode == RESULT\_CANCELED)

;

else if (resultCode == RESULT\_OK)

{

String title = intent.getStringExtra(KEY\_TITLE);

String url = intent.getStringExtra(KEY\_URL);

data.add(new Bookmark(title, url));

saveData();

}

}

private void startBookmark(int position)

{

Intent intent = new Intent(this, UrlViewer.class);

intent.putExtra(KEY\_URL, data.get(position).getUrl());

startActivity(intent);

}

private void addItem()

{

Intent intent = new Intent(this, BookmarkItemEdit.class);

startActivityForResult(intent, ITEM\_EDIT\_REQUEST\_CODE);

}

private void editSelectedItem()

{

// To Do

}

private void deleteSelectedItems()

{

for (int i = data.getCount()-1; i >= 0; i--)

{

if (data.get(i).isSelected())

data.remove(i);

}

data.notifyDataSetChanged();

}

private void saveData()

{

try

{

PrintWriter out = new PrintWriter(new OutputStreamWriter(

openFileOutput(BOOKMARK\_FILE\_NAME,

Context.MODE\_PRIVATE)));

data.saveBookmarks(out);

out.close();

}

catch (Exception ex)

{

Toast.makeText(this, "Error saving data: " + ex.getMessage(),

Toast.LENGTH\_SHORT).show();

}

}

private void loadData()

{

try

{

BufferedReader in = new BufferedReader(new InputStreamReader(

openFileInput(BOOKMARK\_FILE\_NAME)));

data.loadBookmarks(in);

data.notifyDataSetChanged();

in.close();

}

catch (Exception ex)

{

Toast.makeText(this, "Error loading data: " + ex.getMessage(),

Toast.LENGTH\_SHORT).show();

}

}

}

**BookmarkItemEdit.java**

package org.mathcs.ibookmark;

/\*\*

\* Activity to edit a bookmark by editing its title and its address. Also lets you

\* preview the address.

\*

\* @author Bert Wachsmuth

\* @version 1.0 (10/2013)

\*/

import android.app.Activity;

import android.app.AlertDialog;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.webkit.WebChromeClient;

import android.webkit.WebView;

import android.webkit.WebViewClient;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

public class BookmarkItemEdit extends Activity

{

private TextView urlField = null;

private TextView titleField = null;

private Button preview = null;

private Button okay = null;

private WebView web = null;

private AlertDialog alert = null;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_bookmark\_item);

getActionBar().setDisplayHomeAsUpEnabled(true);

setTitle("Add/Edit Bookmark");

preview = (Button)this.findViewById(R.id.id\_button\_preview);

okay = (Button)this.findViewById(R.id.id\_button\_okay\_itemedit);

urlField = (TextView) this.findViewById(R.id.id\_input\_url);

titleField = (TextView)this.findViewById(R.id.id\_input\_title);

urlField.setText("http://");

web = (WebView)this.findViewById(R.id.id\_webview\_itemedit);

web.getSettings().setJavaScriptEnabled(true);

web.setWebChromeClient(new WebChromeClient()

{

public void onProgressChanged(WebView view, int progress)

{

// Activity, WebView measure progress with different scales

// Progress meter will automatically disappear at 100%

setProgress(progress \* 1000);

}

});

web.setWebViewClient(new WebViewClient()

{

public void onReceivedError(WebView view, int errorCode,

String description, String failingUrl)

{

Toast.makeText(BookmarkItemEdit.this, "Oh no! " + description,

Toast.LENGTH\_SHORT).show();

}

});

preview.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

String url = urlField.getText().toString().trim();

if ((url == null) ||

(url.equals("")) ||

(url.equals("http://")))

showAlert("The web address is empty - can't preview.");

else

web.loadUrl(url);

}

});

okay.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent data = new Intent();

String title = titleField.getText().toString().trim();

String url = urlField.getText().toString().trim();

if ((title == null) || (title.equals("")))

showAlert("The Title can not be empty.");

else if ((url == null) ||

(url.equals("")) ||

(url.equals("http://")))

showAlert("The Address can not be empty.");

else

{

data.putExtra(BookmarkActivity.KEY\_URL, url);

data.putExtra(BookmarkActivity.KEY\_TITLE, title);

setResult(RESULT\_OK, data);

finish();

}

}

});

}

private void showAlert(String msg)

{

if (alert == null)

{

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle("Warning");

builder.setPositiveButton("Dismiss", null);

alert = builder.create();

}

alert.setMessage(msg);

alert.show();

}

}

**UrlViewer.java**

package org.mathcs.ibookmark;

/\*\*

\* Minimal "web browser like" activity that shows a web page. Javascript is enabled

\* and links should be active within the activity.

\*

\* @author Bert Wachsmuth

\* @version 1.0 (10/2013)

\*/

import android.app.Activity;

import android.os.Bundle;

import android.view.Window;

import android.webkit.WebChromeClient;

import android.webkit.WebView;

import android.webkit.WebViewClient;

import android.widget.Toast;

public class UrlViewer extends Activity

{

private WebView web;

private String url;

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

// Must call this before setting the content view!!!

getWindow().requestFeature(Window.FEATURE\_PROGRESS);

web = new WebView(this);

setContentView(web);

getActionBar().setDisplayHomeAsUpEnabled(true);

web.getSettings().setJavaScriptEnabled(true);

web.setWebChromeClient(new WebChromeClient()

{

public void onProgressChanged(WebView view, int progress)

{

// Progress meter will automatically disappear at 100%

setProgress(progress \* 1000);

}

});

web.setWebViewClient(new WebViewClient()

{

public void onReceivedError(WebView view, int errorCode,

String description, String failingUrl)

{

Toast.makeText(UrlViewer.this, "Oh no! " + description,

Toast.LENGTH\_SHORT).show();

}

});

url = getIntent().getStringExtra(BookmarkActivity.KEY\_URL);

setTitle(url);

web.loadUrl(url);

}

}