ITSMAP F13 Lesson 8 part 1

Androids **Persistence**/Data storage

Jesper Rosholm Tørresø

Data storage options are the following

(Click on links for developer.android.com)

- Shared Preferences Store private primitive data in key-value pairs.
- Internal Storage Store private data on the device memory.
- External Storage Store public data on the shared external storage.
- SQLite Databases Store **structured** data in a private database.
- Network Connection Store data on the web with your own network server.

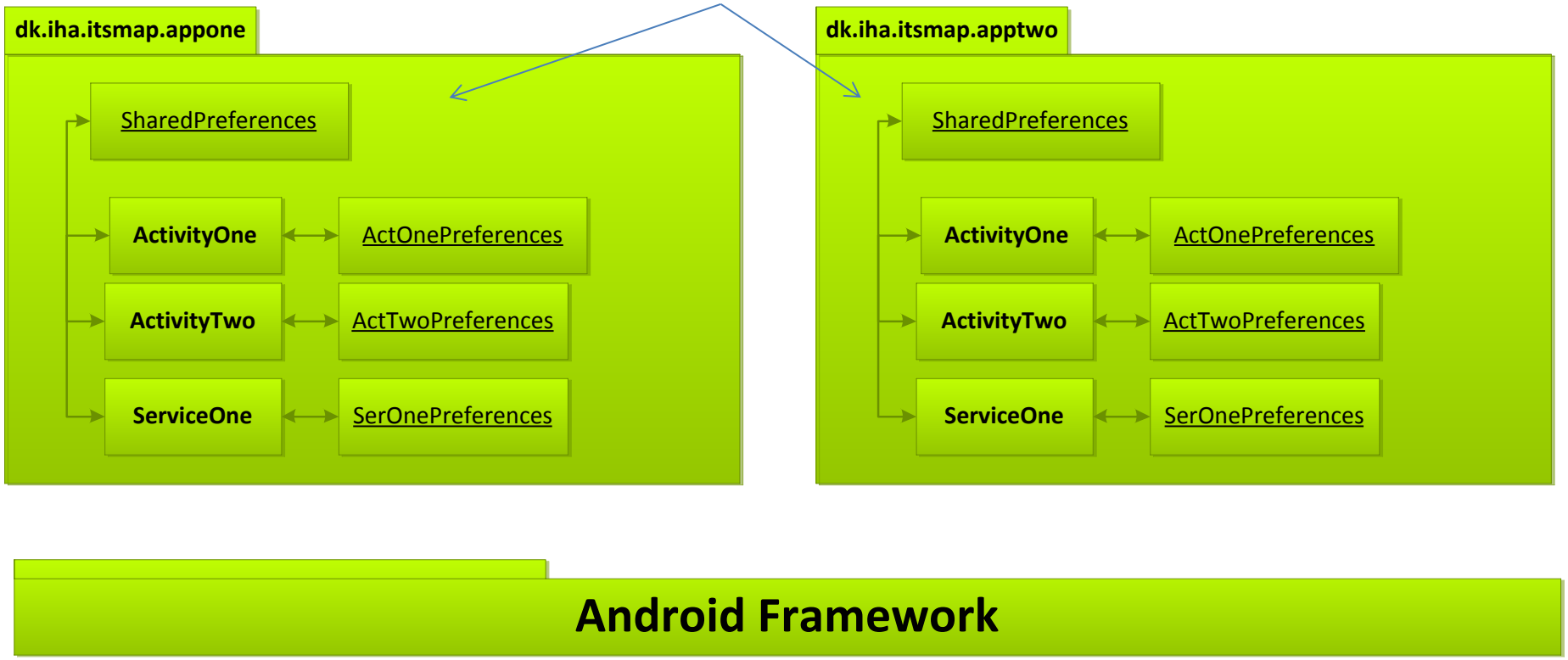
Android and Persistence
On device

- Shared preferences
  - Across Activities in a Application
- Preferences
  - For one specific Activity in one Application
- Files
- SQLite (Next part 2)

http://developer.android.com/guide/topics/data/data-storage.html

Scope and Preferences

Context for Apps



http://developer.android.com/reference/android/content/Context.html#getSharedPreferences(java.lang.String, int)

Simple Persistence
Shared Preferences

- Key / value mechanism
- Values Type mechanism
- Write - commit pattern
- Private for your application - can be shared see slide later on

```
SharedPreferences settings =
getSharedPreferences("MyPrefs",
MODE_PRIVATE);
SharedPreferences.Editor editor =
settings.edit();
editor.putBoolean("wifi_enabled", true);
editor.commit();
```

Type value mechanism

```
// Retrieve an editor to modify the shared
preferences.
   SharedPreferences settings =
   getSharedPreferences("MyPrefs",
   Activity.MODE_PRIVATE);
  SharedPreferences.Editor editor = settings.edit();

  // Store new primitive types in the shared
preferences object.
  editor.putBoolean("isTrue", true);
  editor.putFloat("lastFloat", 1f);
  editor.putInt("wholeNumber", 2);
  editor.putLong("aNumber", 3l);
  editor.putString("textEntryValue", "Not Empty");
  editor.commit();
```

Type value mechanism

```
// Get the stored preferences
  int mode = Activity.MODE_PRIVATE;
     SharedPreferences mySharedPreferences =
   getSharedPreferences(("MyPrefs",mode);


  // Retrieve the saved values.
  boolean isTrue = mySharedPreferences.getBoolean("isTrue", false);
  float lastFloat = mySharedPreferences.getFloat("lastFloat", 0f);
  int wholeNumber = mySharedPreferences.getInt("wholeNumber", 1);
  long aNumber = mySharedPreferences.getLong("aNumber", 0);
  String stringPreference =
        mySharedPreferences.getString("textEntryValue", "");
```

Accessing Shared Prefs

```
public class Calc extends Activity {
    public static final String PREFS_NAME = "MyPrefsFile";

    @Override
    protected void onCreate(Bundle state){
       super.onCreate(state);
       . . .

       // Restore preferences
       SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
       boolean silent = settings.getBoolean("silentMode", false);
       setSilent(silent);
    }

    @Override
    protected void onStop(){
       super.onStop();

      // We need an Editor object to make preference changes.
      // All objects are from android.context.Context
      SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
      SharedPreferences.Editor editor = settings.edit();
      editor.putBoolean("silentMode", mSilentMode);

      // Commit the edits!
      editor.commit();
    }
}
```

Activity/Service State
Get preferences *without a Key*

```
protected void saveActivityPreferences(){
  // Create or retrieve this activity' preference object.
  SharedPreferences activityPreferences =
getPreferences(Activity.MODE_PRIVATE);
  // Retrieve an editor to modify the shared preferences.
  SharedPreferences.Editor editor =
activityPreferences.edit();
  // Retrieve the View
  TextView myTextView =
(TextView)findViewById(R.id.myTextView);
  // Store new primitive types in the shared preferences
object.
  editor.putString("currentTextValue",
myTextView.getText().toString());
  // Commit changes.
  editor.commit();
}
```

Saving Activity instance state

```
// ** Listing 6-6:
private static final String TEXTVIEW_STATE_KEY =
"TEXTVIEW_STATE_KEY";

@Override
public void onSaveInstanceState(Bundle outState) {
        // Retrieve the View
        TextView myTextView = (TextView)
        findViewById(R.id.myTextView);

// Save its state
        outState.putString(TEXTVIEW_STATE_KEY,
        myTextView.getText().toString());
        super.onSaveInstanceState(outState);
}
```

Restoring Activity instance state

```
// ** Listing 6-7:
@Override
public void onCreate(Bundle icicle) {
  super.onCreate(icicle);
  setContentView(R.layout.main);

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

  String text = "";
  if (icicle != null && icicle.containsKey(TEXTVIEW_STATE_KEY))
    text = icicle.getString(TEXTVIEW_STATE_KEY);

  myTextView.setText(text);
}
```

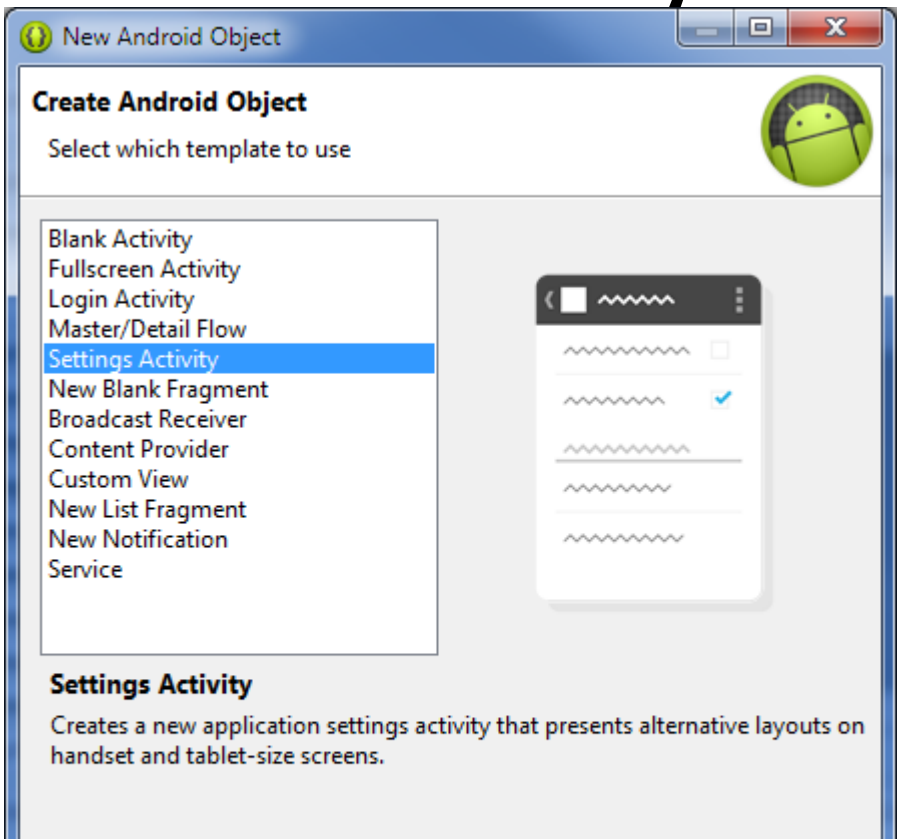
Preference Activity and Preference Framework

- Read [MEIER] ch 7 part “Creating a Settings activity for EQ viewer 1-16” for the principles
- Use the ADT Template **“Preference Activity”**

New->Other->Android Object->

Settings Activity

Template is version dependent



Listen for Changes in Preferences

On Shared Preference Change Listener skeleton implementation

```
public class MyActivity extends Activity implements
OnSharedPreferenceChangeListener
{
    @Override
    public void onCreate(Bundle SavedInstanceState) {
        // Register this OnSharedPreferenceChangeListener
        Context context = getApplicationContext();
        SharedPreferences prefs = PreferenceManager
        .getDefaultSharedPreferences(context);
        prefs.registerOnSharedPreferenceChangeListener(this);
    }

    public void onSharedPreferenceChanged(SharedPreferences prefs,
    String key) {
    // TODO Check the shared preference and key parameters and change UI or
    // behavior as appropriate.
    }
}
```

Share Across Applications

- Making any file MODE_WORLD_READABLE or (worse) MODE_WORLD_WRITEABLE is a bad idea. You lose any hope of security.
- If you wish to share data between two applications, there are a myriad of solutions, such as:
  - service with an API exposed by AIDL (Android Inteface Definition Language)
  - service with an API exposed via commands sent via startService() and responses sent via a Messenger or createPendingResult() PendingIntent or something
  - content provider
  - broadcast Intents
- All of those allow you to define permissions for integration and let you control the granularity of access.

It’s raw...

- Files stored in the res/raw folder
- Sounds, video, dictionaries....
- Will get an id in the **R** class

Ex

```
InputStream in=getResources().openRawResource(R.raw.words);
```

Plain old files...
files internal storage

- Open a Java stream and attach a reader/writer
- Read or write stuff
- close the stream

```
String FILENAME = "hello_file";
String string = "hello world!";

FileOutputStream fos = openFileOutput(FILENAME, Context.MODE_PRIVATE);
fos.write(string.getBytes());
fos.close();
```

Using external storage (SD card)

```
BroadcastReceiver mExternalStorageReceiver;
boolean mExternalStorageAvailable = false;
boolean mExternalStorageWriteable = false;

void updateExternalStorageState() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state)) {
        mExternalStorageAvailable = mExternalStorageWriteable = true;
    } else if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
        mExternalStorageAvailable = true;
        mExternalStorageWriteable = false;
    } else {
        mExternalStorageAvailable = mExternalStorageWriteable = false;
    }
    handleExternalStorageState(mExternalStorageAvailable,
            mExternalStorageWriteable);
}
```

Se more in complete example

http://developer.android.com/reference/android/os/Environment.html

Lesson 8 Exercise 1

- Make a new project with two Activities
  - One to read and show shared preferences values
    - Include an options menu to change to the PreferenceActivity.
  - A PreferenceActivity using the Android Preferences Framework (Use ADT Template or [MEIER] guide)
    - Setup some user options to save in SharedPreferences
- Start and stop and restart the App to see that options are persisted.