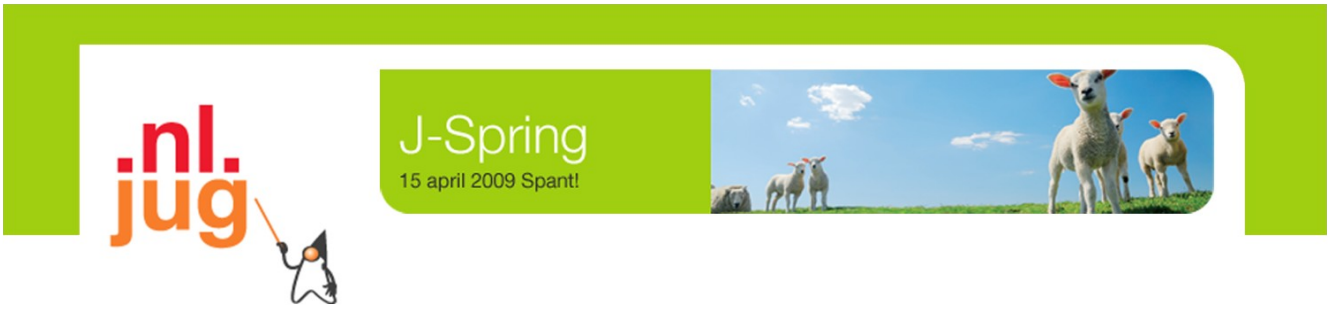


Practical Programming on Android

Introduction

- Koert Zeilstra is freelance software developer, used C++, OOP, Java, Swing, Struts, EJB, Spring, JSF, Android
- Palm Pilot – long battery life
- Palm V - Introduction JavaOne 1999, KVM

G1

- First Android phone: black, white, bronze
- Specifications: 192 MB RAM – enough in daily use
- More Android phones: Samsung, HTC, Motorola

What is Android

- Linux kernel and Dalvik virtual machine, application runs in process
- Develop in Java source code, compiled to Dalvik bytecode
    - Java 5: annotations, generics
- You can use regular debugger in Eclipse with Android plugin
- NextAction – started in october 2008, todo lists
- Google developer phone - synchronisation Gmail and contacts, Google calendar
- Android is attractive for developers:
    - Java source code, familiar development environment
    - Open platform

User interface

Layout in XML

```
<LinearLayout android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <TextView
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:text="@string/name_title"
    style="@style/label" />
  <EditText android:id="@+id/name"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="@string/initial_text"
    style="@style/editable_text">
    <requestFocus />
  </EditText>
</LinearLayout>
```

Activity

Controller of a screen

```
public class ContextEditorActivity extends Activity {
    private EditText nameWidget;
    protected void onCreate(Bundle savedValues) {
        super.onCreate(savedValues);
        setContentView(R.layout.context_editor);
        nameWidget = (EditText) findViewById(R.id.name);
    }
}
```

Connect Activity to XML layout

Binding via id of view component: `android:id="@+id/name"`, use findViewById

Eclipse plugin generates R class with layout and other resources, id's

advantage: auto-completion

Demo Eclipse

- View XML
- Preview layout, locale, portrait/landscape, screen size

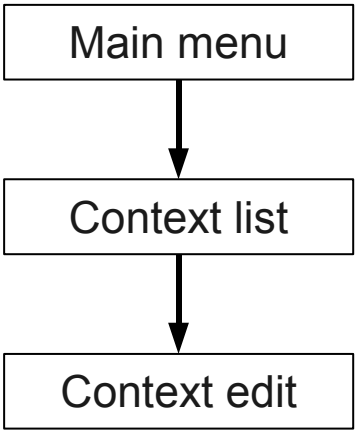
Buttons - context editor

Save button

```
saveButton = (Button) findViewById(R.id.save_button);
saveButton.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    finish();
  }
});
```

- Bind save button
- Listener onClick – user clicks on button
- Activity finish() - back to previous screen

Navigation

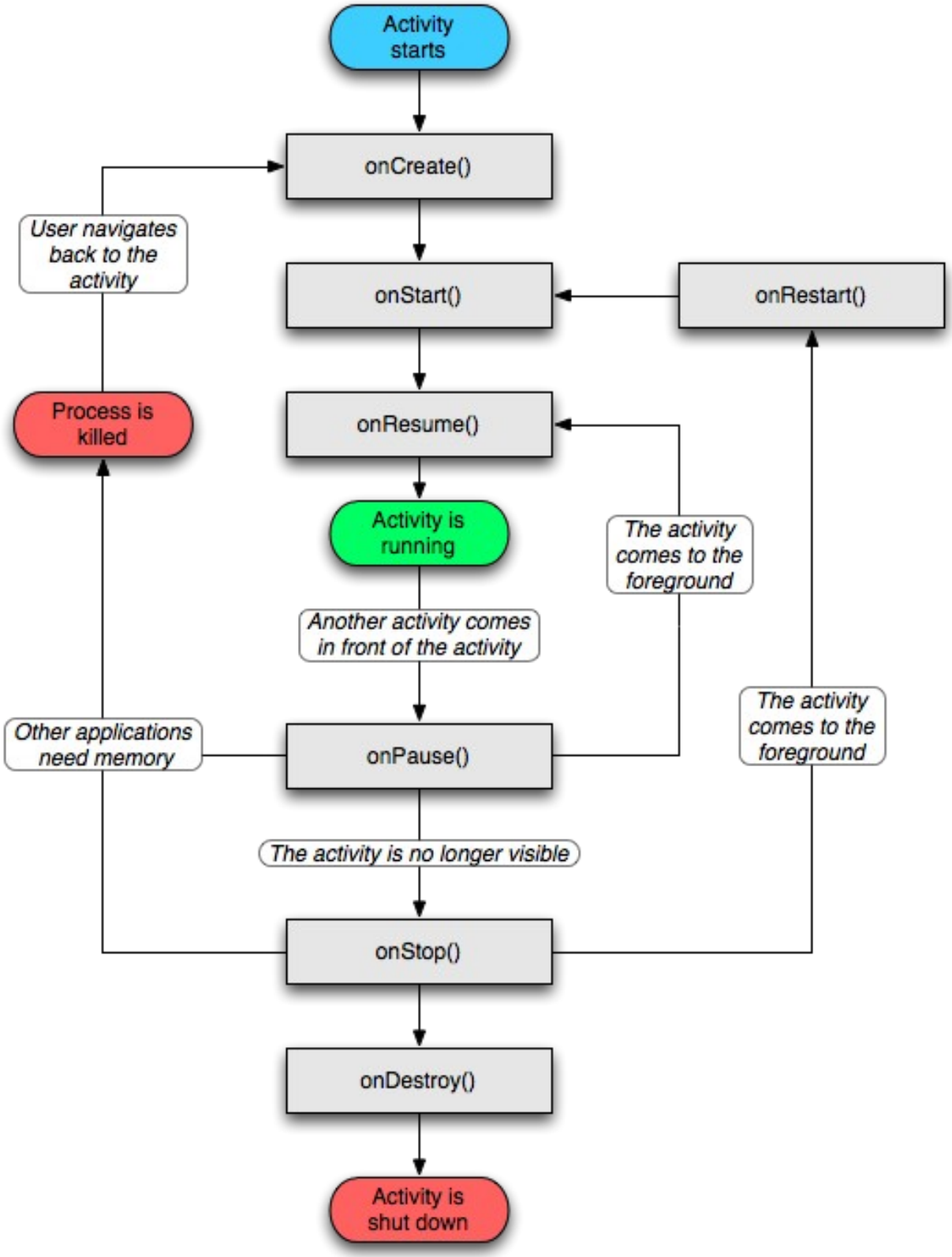


Navigation through application, finish() in Context edit navigates back to Context list.

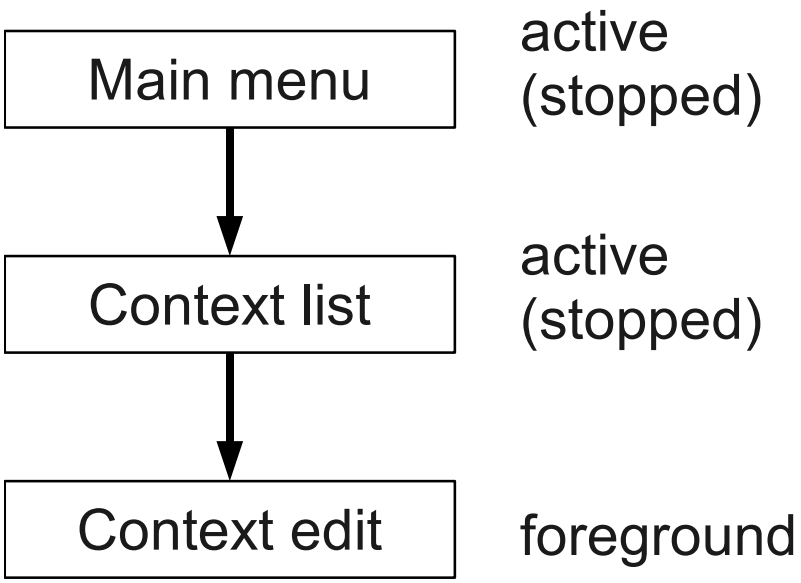
Go to next Activity:

```
Intent intent = new Intent(androidContext,
  ContextEditor.class);
startActivity(intent);
```

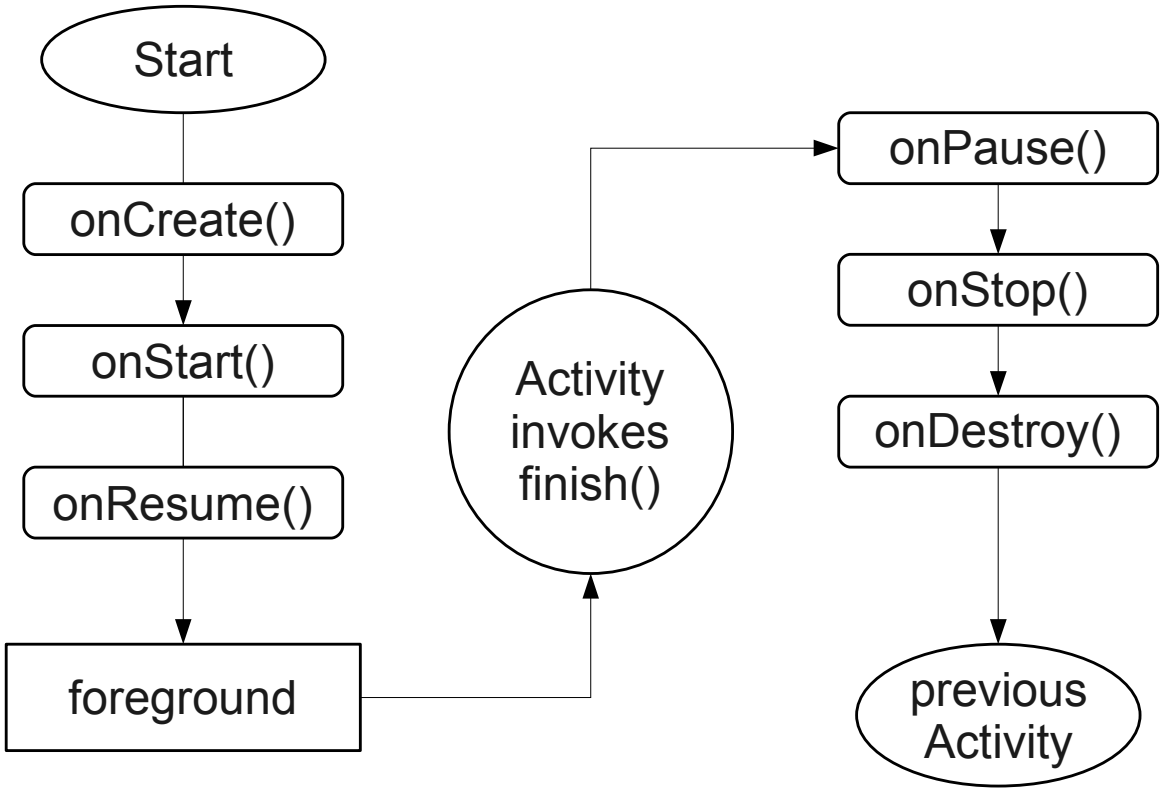
Lifecycle



Stack of activities


Main menu
active
(stopped)
Context list
active
(stopped)
Context edit
foreground

Normal flow



Receive phone call

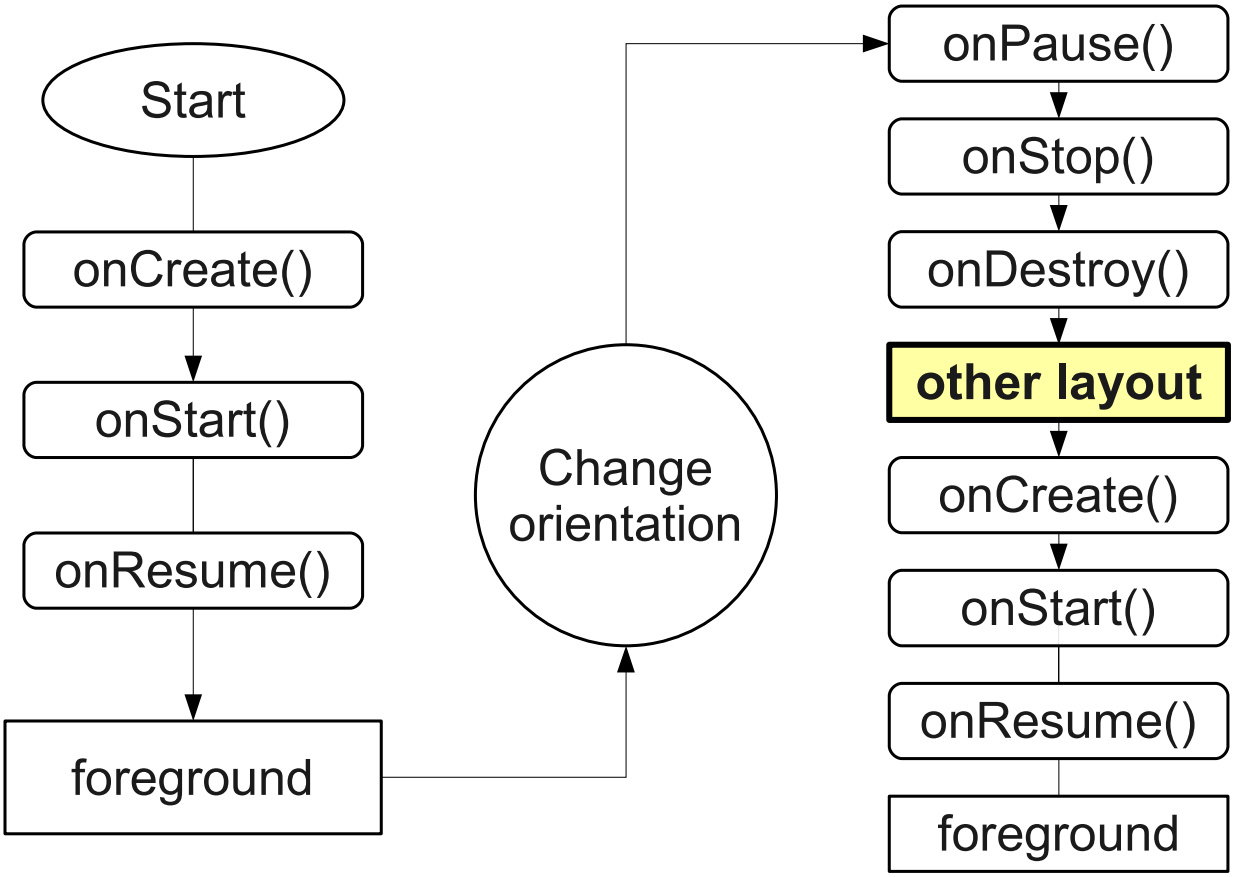
When you receive phone call: onPause() and after call: onResume()

Low on memory

Low on memory: onPause() and process is killed

Change orientation

Portrait/landscape



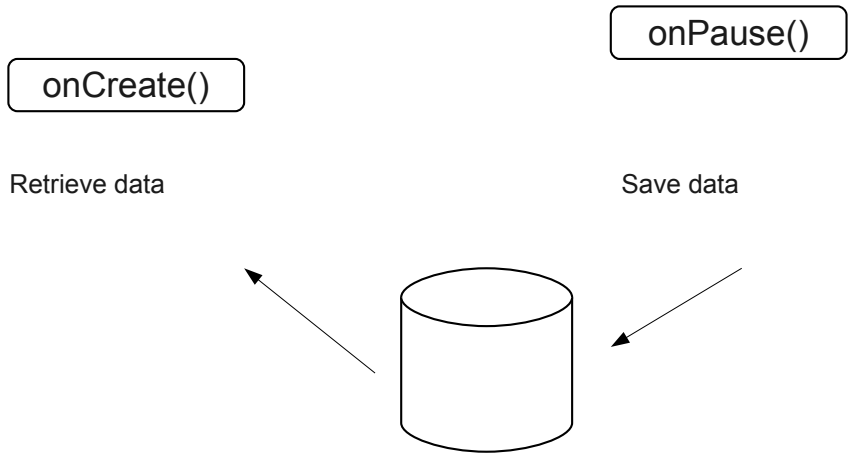
Activity on foreground – onPause(), onStop(), onDestroy()

Change layout – onCreate(), onStart(), onResume()

Practical data retrieve/save

When your application gets killed because of low memory and you want to save the data of user, you can do this in onPause(), in normal data save or maybe in temporary space somewhere and recover later. This is also useful in other situations, like back button.

Simple solution:



onCreate(): retrieve

onPause(): save

UI

Layout widgets

- AbsoluteLayout – x/y coordinates, inflexible
- LinearLayout – horizontal/vertical, simple, flexible
- RelativeLayout – position relative to other components, flexible, more complex
- TableLayout – rows/columns

View components:

List
Button
Text view/edit
Spinner
Google map, and more...

Navigation

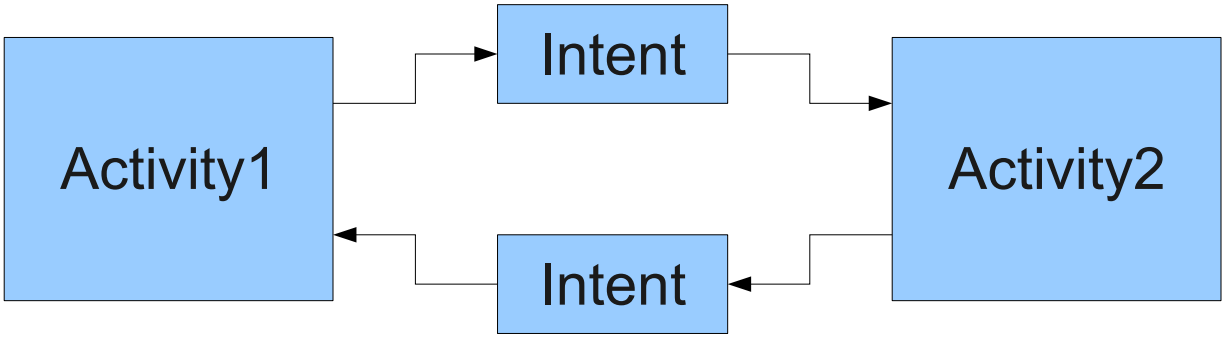
With parameters

```
Bundle bundle = new Bundle();
bundle.putString("myparameter", "Hello world");
intent.putExtras(bundle);
```

You can use Parcelable interface to add other classes to bundle.

Get results back

```
private static final int EDIT_ITEM = 2;
...
startActivityForResult(intent, EDIT_ITEM);
```



After Activity2 is finished, Activity1 gets back:

```
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
  switch (requestCode) {
    case NEW_ITEM:
      if (resultCode == Activity.RESULT_OK) {
        if (data != null) {
          Uri uri = data.getData();
          itemIdToSelect = ContentUris.parseId(uri);
        }
      }
      // ...
```

Using a URI:

```
Uri itemUri = Uri.parse("content://nextaction/task/1");
Intent intent = new Intent(Intent.ACTION_VIEW, itemUri);
startActivity(intent);
```

Target Activity receives:

```
public void onCreate(Bundle savedData) {
  Uri contextUri = getIntent().getData();
```

To other application

```
Intent intent = new Intent(Intent.ACTION_PICK,
  Uri.parse("content://contacts/people"));
startActivityForResult(intent, CHOOSE_CONTACT);

protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
  // data.getDate() == "content://contacts/people/2"
}
```

AndroidManifest.manifest

Match with URI

```
<activity android:name=".task.TaskViewActivity"
  android:label="@string/action_title">
 <intent-filter>
  <action android:name="android.intent.action.VIEW"/>
  <category android:name="android.intent.category.DEFAULT"/>
  <data android:scheme="content" android:pathPrefix="nextaction/task"/>
 </intent-filter>
</activity>
```

Start Activity

```
<activity android:name=".toplevel.activity.TopLevelActivity"
    android:label="@string/app_name">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Data Storage

Files

Internal:

```
File file = new File(fileName);
FileOutputStream output = null;
try {
 output = new FileOutputStream(file);
} finally {
 if (output != null) {
  try { output.close(); } catch (IOException e) {}
 }
}
```

SD Card:

```
File file = new File(
 Environment.getExternalStorageDirectory(),
 fileName);
```

Preferences

Reading

```
SharedPreferences settings = getSharedPreferences(
  "MyPrefsFile", MODE_PRIVATE);
String userName = settings.getString("userName", "defaultUser");
```

Writing

```
SharedPreferences settings = getSharedPreferences(
  "MyPrefsFile", MODE_PRIVATE);
SharedPreferences.Editor editor = settings.edit();
editor.putString("userName", userName);
editor.commit();
```

SQLite

Execute SQL directly

```
SQLiteDatabase db = ...
db.execSQL("CREATE TABLE context ("
 + "_id INTEGER PRIMARY KEY,"
 + "name TEXT"
 + ");");
```

Query

public Cursor query (SQLiteDatabase db,
String[] projectionIn, // selected fields
String selection, String[] selectionArgs, // where clause and values
String groupBy, String having, String sortOrder)

Example

```
SQLiteQueryBuilder contextQuery = new SQLiteQueryBuilder();
contextQuery.setTables("context");
contextQuery.setProjectionMap(CONTEXT_PROJECT_MAP);
Cursor cursor = contextQuery.query(db, new String[] {Task.ID},
 Task.NAME + " = ?",
 new String[] {contextName}, null, null, null);

CONTEXT_PROJECT_MAP = new HashMap<String, String>();
CONTEXT_PROJECT_MAP.put(GtdContext.ID,
 DatabaseHelper.TABLE_CONTEXT + "._id");
CONTEXT_PROJECT_MAP.put(GtdContext.NAME,
 DatabaseHelper.TABLE_CONTEXT + ".name");
```

SQL: context.name **as** name

```
if (cursor.moveToFirst()) {
 String name = cursor.getString(columnIndex);
}
```

Update

public int update (
String table,
ContentValues values, // map of update values
String whereClause,
String[] whereArgs)

Example

```
ContentValues values = new ContentValues();
values.put(GtdContext.Column.NAME, newName);
db.update(DatabaseHelper.TABLE_CONTEXT, values,
 GtdContext.ID + " = ?", new String[] {Integer.toString(contextId)});
```

ContentProvider

Standard interface for retrieving/storing data.

```
public class GtdProvider extends ContentProvider {
  ...
}
```

In AndroidManifest.xml

```
<provider android:name=".database.GtdProvider"
  android:authorities="net.kazed.android.gtd.database.GtdAndroid"/>
```

Query

```
public abstract Cursor query (
 Uri uri, // ID of item or items
 String[] projection, // selected fields
 String selection, String[] selectionArgs, // where clause and values
 String sortOrder)
```

Insert

```
public abstract Uri insert (Uri uri, ContentValues values)
```

Update

```
public abstract int update (Uri uri, ContentValues values, String
selection, String[] selectionArgs)
```

Delete

```
public abstract int delete (Uri uri, String selection, String[]
selectionArgs)
```

Example

```
public static final Uri CONTENT_URI =
 Uri.parse("content://net.kazed.android.gtd.database.GtdAndroid/context");

Uri itemUri = ContentUris.withAppendedId(GtdContext.CONTENT_URI, id);

Cursor cursor = getContentResolver.query(itemUri,
 new String[] {ID, NAME}, null, null, null);
```

Implement ContentProvider

Query

```
public Cursor query(Uri uri, String[] projection, String selection,
```

```
    String[] selectionArgs, String sort) {
 Cursor cursor = null;
 SQLiteQueryBuilder qb = new SQLiteQueryBuilder();
 SQLiteDatabase db = dbHelper.getReadableDatabase();

 switch (URI_MATCHER.match(uri)) {
 case CONTEXT_ID:
  qb.setTables(DatabaseHelper.TABLE_CONTEXT);
  qb.appendWhere("_id=" + uri.getPathSegments().get(1));
  cursor = qb.query(db, projection, selection, selectionArgs,
    null, null, "name DESC");
 break;
 // ...
 return cursor;
}
```

Setup URI matcher

```
public static final String PACKAGE =
 "net.kazed.android.gtd.database.GtdAndroid";


static {
 URI_MATCHER = new UriMatcher(UriMatcher.NO_MATCH);
 URI_MATCHER.addURI(PACKAGE, "context", CONTEXTS);
 URI_MATCHER.addURI(PACKAGE, "context/#", CONTEXT_ID);
 // ...
}
```

Implement insert, update, delete

```
public Uri insert (Uri uri, ContentValues values)
public int update (Uri uri, ContentValues values, String selection,
String[] selectionArgs)
public int delete (Uri uri, String selection, String[] selectionArgs)
```

Mime type/URI mapping

```
public static final String CONTENT_ITEM_TYPE =
 "vnd.android.cursor.item/vnd.nextaction.context";


public String getType(Uri uri) {
 String type = null;
 switch (URI_MATCHER.match(uri)) {
  case CONTEXTS:
  type = GtdContext.CONTENT_TYPE;
  break;
 // ...
 }
 return type;
}
```

Navigation with mime type

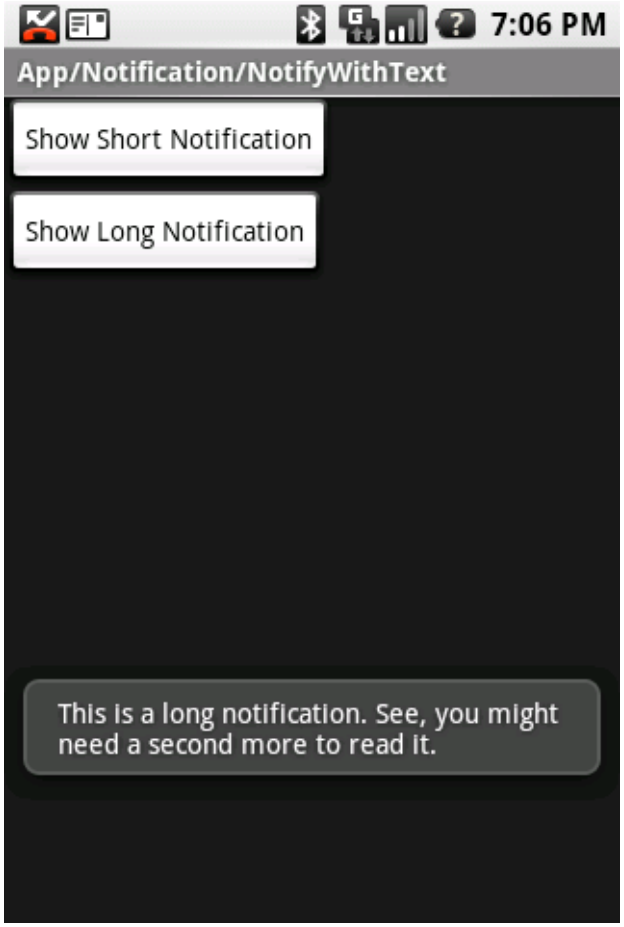
```
Uri itemUri = Uri.parse(
 "content://net.kazed.android.gtd.database.GtdAndroid/context/1");
Intent intent = new Intent(Intent.ACTION_VIEW, itemUri);
```

```
startActivity(intent);
```

Android manifest

```
<activity android:name=".context.activity.ContextEditorActivity"
  android:label="@string/title_context">
 <intent-filter>
  <action android:name="android.intent.action.EDIT" />
  <category android:name="android.intent.category.DEFAULT" />
  <data android:mimeType="vnd.android.cursor.item/vnd.nextaction.context"/>
 </intent-filter>
</activity>
```

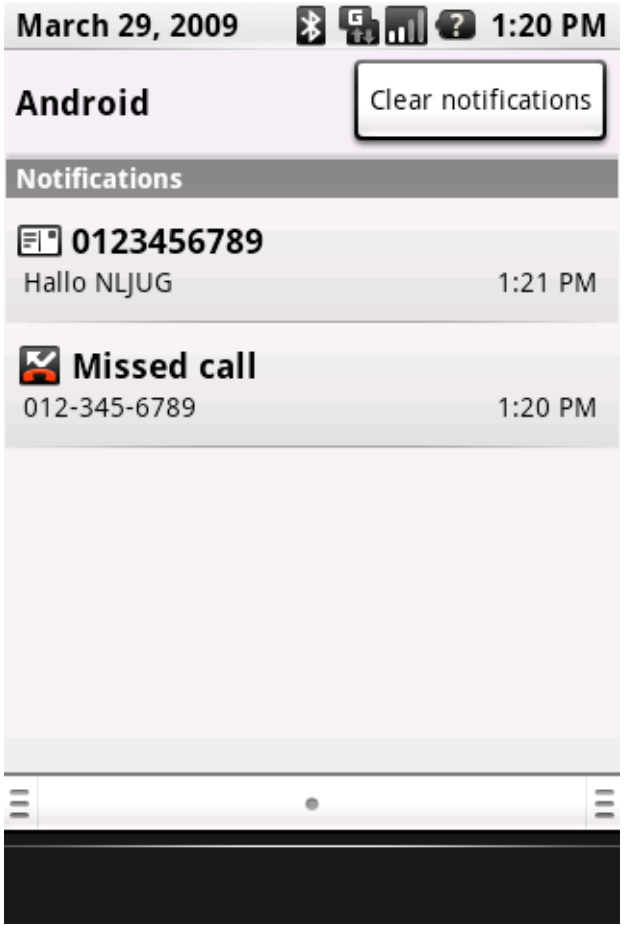
Notification



Toast popup

```
Toast toast = Toast.makeText(this,
  R.string.backup_import_failure, Toast.LENGTH_LONG);
toast.show();
```

Notification bar




```
NotificationManager notificationManager = (NotificationManager)
 context.getSystemService(Service.NOTIFICATION_SERVICE);
Resources resources = context.getResources();
Notification notification = new Notification(
 android.R.drawable.ic_menu_info_details,
 "Hello", System.currentTimeMillis());
Intent applicationIntent = new Intent(context, StartedTasksActivity.class);
PendingIntent pendingIntent = PendingIntent.getActivity(context, 0,
 applicationIntent, 0);
notification.setLatestEventInfo(context, "Title",
 "Notification text", pendingIntent);
notificationManager.notify(NOTIFICATION_ID, notification);
```

Other notification:

- Blinking LED's
- Ringtone
- Vibration

Background process

Alarm

```
AlarmManager alarmManager = (AlarmManager)
  context.getSystemService(Context.ALARM_SERVICE);
Intent processIntent = new Intent(context, StartDateProcessor.class);
PendingIntent alarmIntent = PendingIntent.getBroadcast(context, 0,
  processIntent, 0);
Date wakeupTime = // ...
alarmManager.set(AlarmManager.RTC_WAKEUP, wakeupTime, alarmIntent);
```

BroadcastReceiver

```
public class StartDateProcessor extends BroadcastReceiver {
 public void onReceive(Context context, Intent intent) {
 }
}
```

Send broadcast

```
Intent intent = new Intent("net.kazed.nextaction.NEW_TASK");
intent.putExtra("taskId", taskId);
sendBroadcast(intent);
```

Register in manifest:

```
<receiver android:name=".process.TaskProcessor">
 <intent-filter>
  <action android:name="net.kazed.nextaction.NEW_TASK"/>
 </intent-filter>
</receiver>
```

Service

```
public class MyService extends Service {
    public void onCreate() {
        super.onCreate();
    }

    public IBinder onBind(Intent intent) {
        return null;
    }

    public void onStart(Intent intent, int startId) {
        super.onStart(intent, startId);
    }
}
```

Register service

Manifest:

```
<service android:name=".app.LocalService"/>
```

Phone features

Camera

```
Camera camera = Camera.open();
ShutterCallback shutterCallback = new ShutterCallback() {
  public void onShutter() {
  }
};
PictureCallback rawCallback = new PictureCallback() {
  public void onPictureTaken(byte[] data, Camera camera) {
  }
};
PictureCallback jpegCallback = new PictureCallback() {
  public void onPictureTaken(byte[] data, Camera camera) {
  }
};
camera.takePicture(shutterCallback, rawCallback, jpegCallback);
```

GPS

```
LocationManager locationManager = (LocationManager)
  getSystemService(Context.LOCATION_SERVICE);

Criteria criteria = new Criteria();
criteria.setAccuracy(Criteria.ACCURACY_COARSE);
boolean enabledOnly = true;
String providerName = locationManager.getBestProvider(criteria, enabledOnly);
```

Location location = locationManager.getLastKnownLocation(providerName);

Tilt sensors

```
SensorManager sensorManager = (SensorManager)
  getSystemService(Context.SENSOR_SERVICE);
SensorListener listener = new SensorListener() {
  public void onAccuracyChanged(int sensor, int accuracy) {
  }
  public void onSensorChanged(int sensor, float[] values) {
  }
};
```

sensorManager.registerListener(listener, SensorManager.SENSOR_ACCELEROMETER);

Testing

Dependency injection

- Used in modern frameworks (Spring, Guice, EJB, JSF)
- Possible on Android? Discussion in developer google group
- DI is "heavy weight"
- costs too much performance and/or memory

- Android provides integration test: run on emulator

Java libraries

- not: bytecode generators (cglib)
- XML parsing

- xstream: easy write and read, somewhat large for small app
- SAX parser: small and lightweight

Performance

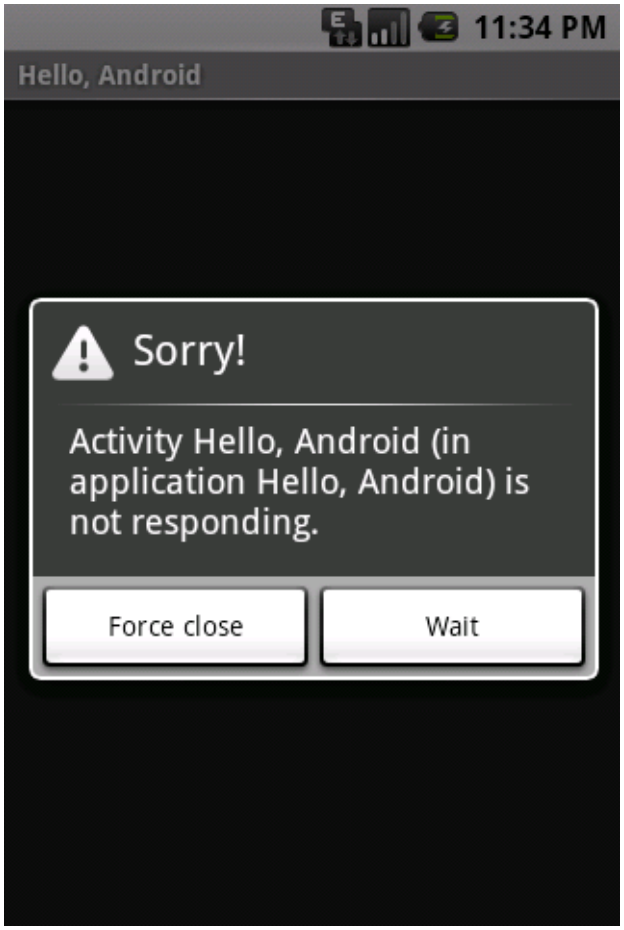
Avoid new

Prefer static helper methods

Avoid this of object lookup

See Android developer web site for more.

Responsiveness: put long process in thread with progress bar or service with listener.



Publishing your application

- Become developer ($25)
- Buy developer phone $399 + $173 (including shipping and customs to NL)

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.kazed.nextaction" android:versionCode="084"
android:versionName="0.8.4"
```

Provider

```
public class DatabaseHelper extends SQLiteOpenHelper {
  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    switch (oldVersion) {
    case 10:
      db.execSQL("ALTER TABLE " + DatabaseHelper.TABLE_TASK + " ADD COLUMN "
         + Task.TYPE + " INTEGER");
    }
  }
}
```

Publish on market, update

More information

Developer documentation: http://developer.android.com

SDK download, Tutorials with example code, reference documentation

Google groups: android-beginners, android-developers

Android market: http://www.android.com/market

Book: Professional Android Application Development - Reto Meier

Conclusion

After 12 jaar Palm Pilot, now internet everywhere and synchronization, more recharge
Familiar development environment: Eclipse with Android plugin

    - IDE, language, debugging
    - Components met listeners
    - Navigatie en stateless zoals web application
- New
    - Use Intent objects to link application together
    - Database provider for DAO, use URI's