References

Examples

See Android ApiDemos
com.example.android.apis.app classes
Keystroke Example

Show how to react
to keystrokes not handled by views
Keystroke Activity Methods

public boolean onKeyDown(int keyCode, KeyEvent event)

    Called when a key was pressed down and not handled by any of the views inside of the activity

public boolean onKeyMultiple(int keyCode, int repeatCount, KeyEvent event)

    Called when multiple down/up pairs of the same key have occurred in a row.

public boolean onKeyUp(int keyCode, KeyEvent event)

    Called when a key was released and not handled by any of the views inside of the activity
KeyDownExample

public class GoodStuff extends Activity {
    private Button keyText;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        keyText = (Button) this.findViewById(R.id.update);
    }

    public boolean onKeyDown(int keyCode, KeyEvent event) {
        keyText.setText(String.valueOf(keyCode));
        switch (keyCode) {
            case (KeyEvent.KEYCODE_T):
                showToast();
                break;
            default:
                return super.onKeyDown(keyCode, event);
        }
        return true;
    }
}

Warning KeyEvent.KEYCODE_T is not the ascii of t or T
private void showToast() {
    keyText.setText("N");
    Toast.makeText(this, "This is a test", Toast.LENGTH_LONG).show();
}

Toast displays a text dialog that will vanish in a short time.
public class GoodStuff extends Activity {
    private Button keyText;

    Will use keystroke to start examples

    public boolean onKeyDown(int keyCode, KeyEvent event) {
        keyText.setText(String.valueOf(keyCode));
        switch (keyCode) {
            case (KeyEvent.KEYCODE_T):
                showToast();
                break;
            case (KeyEvent.KEYCODE_A):
                etc.
                etc.
                default:
                return super.onKeyDown(keyCode, event);
        }
        return true;
    }
}
Service

A component
  Runs in the background for an indefinite period of time
  Not interacting with the user
## Existing Services

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVITY_SERVICE</td>
<td>Interacting with the global system state.</td>
</tr>
<tr>
<td>ALARM_SERVICE</td>
<td>Receiving intents at a time of your choosing.</td>
</tr>
<tr>
<td>AUDIO_SERVICE</td>
<td>Management of volume, ringer modes and audio routing.</td>
</tr>
<tr>
<td>CLIPBOARD_SERVICE</td>
<td>Accessing and modifying global clipboard.</td>
</tr>
<tr>
<td>CONNECTIVITY_SERVICE</td>
<td>Management of network connections.</td>
</tr>
<tr>
<td>KEYGUARD_SERVICE</td>
<td>Controlling keyguard.</td>
</tr>
<tr>
<td>LAYOUT_INFLATER_SERVICE</td>
<td>Inflating layout resources in this context.</td>
</tr>
<tr>
<td>LOCATION_SERVICE</td>
<td>Controlling location updates.</td>
</tr>
<tr>
<td>NOTIFICATION_SERVICE</td>
<td>Informing the user of background events.</td>
</tr>
<tr>
<td>POWER_SERVICE</td>
<td>Controlling power management</td>
</tr>
<tr>
<td>SEARCH_SERVICE</td>
<td>Handling searches.</td>
</tr>
<tr>
<td>SENSOR_SERVICE</td>
<td>Accessing sensors.</td>
</tr>
<tr>
<td>TELEPHONY_SERVICE</td>
<td>Management the telephony features of the device.</td>
</tr>
<tr>
<td>VIBRATOR_SERVICE</td>
<td></td>
</tr>
<tr>
<td>WALLPAPER_SERVICE</td>
<td>Accessing wallpapers.</td>
</tr>
<tr>
<td>WIFI_SERVICE</td>
<td>Management of Wi-Fi access.</td>
</tr>
<tr>
<td>WINDOW_SERVICE</td>
<td>Accessing the system's window manager.</td>
</tr>
</tbody>
</table>
private void vibrate() {
    keyText.setText("V");
    Vibrator vibrate = (Vibrator) getSystemService(VIBRATOR_SERVICE);
    vibrate.vibrate(2000);
}
Vibrate Permission

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.sdsu.cs683"
    android:versionCode="1"
    android:versionName="1.0.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".GoodStuff"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.VIBRATE"/>
</manifest>
Notifications

![Notification Screenshots]

- GoodStuff
- Android
- December 04, 2008
- Title Text
- Message Text
- 10:25 PM
- Clear notifications
- 5:30 PM
private void notifyUser() {
    NotificationManager noteManager = (NotificationManager)
        getSystemService(NOTIFICATION_SERVICE);
    Notification note = new Notification(R.drawable.happy,
        "This is a test", System.currentTimeMillis());

    note.defaults = Notification.DEFAULT_SOUND;

    PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
        new Intent(this, GoodStuff.class), 0);

    note.setLatestEventInfo(this, "Title Text", "Message Text", contentIntent);

    // We use a layout id because it is a unique number.
    noteManager.notify(R.layout.main, note);
}
# PendingIntent.getActivity

PendingIntent getActivity(Context context, int requestCode, Intent intent, int flags)

<table>
<thead>
<tr>
<th>context</th>
<th>Context in which this PendingIntent should start the activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>requestCode</td>
<td>not used</td>
</tr>
<tr>
<td>intent</td>
<td>Intent of the activity to be launched</td>
</tr>
<tr>
<td>flags</td>
<td>FLAG_ONE_SHOT, FLAG_NO_CREATE, FLAG_CANCEL_CURRENT</td>
</tr>
</tbody>
</table>
private void notificationOff() {
    NotificationManager noteManager = (NotificationManager)
        getSystemService(NOTIFICATION_SERVICE);
    noteManager.cancel(R.layout.main);
}

Turning It Off
Alarms

PendingIntent selfIntent;
private void alarmOn() {
    selfIntent = PendingIntent.getActivity(this, 0, new Intent(this, GoodStuff.class), 0);
    long alarmTime = SystemClock.elapsedRealtime() + 60* 1000;

    AlarmManager am = (AlarmManager)getSystemService(ALARM_SERVICE);
    am.set(AlarmManager.ELAPSED_REALTIME_WAKEUP, alarmTime, selfIntent);

    Toast.makeText(this, "Set alarm", Toast.LENGTH_LONG).show();
}

private void alarmOff() {
    AlarmManager am = (AlarmManager)getSystemService(ALARM_SERVICE);
}