

AppWidgets



AppWidget

- ▶ Good news: Bardzo proste!
- ▶ AppWidget = specjalny BroadcastReceiver
- ▶ a rozmiar etc, deklarujemy w `res/xml/my_widget.xml`

AppWidgetProvider

AppWidgetProvider, musi zostać zarejestrowany w **AndroidManifest.xml** (w `<application/>`):

```
<receiver android:name=".ui.appwidgets.MyWidgetProvider">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE">
        </action>
    </intent-filter>
    <meta-data android:name="android.appwidget.provider"
        android:resource="@xml/my_widget"/>
</receiver>
```

```
<appwidget-provider xmlns:android="http://schemas.android.  
    android:minWidth="294dp"  
    android:minHeight="72dp"  
    android:updatePeriodMillis="86400000"  
    android:previewImage="@drawable/preview_widget"  
    android:initialLayout="@layout/widget">  
</appwidget-provider>
```

Deklarowanie tego w XML jest wygodniejsze - mamy filtrowanie folderów (-v11).

AppWidget - implementacja

```
public class MyWidgetProvider extends AppWidgetProvider

    @Override
    public void onUpdate(Context context ,
                        AppWidgetManager appWidgetManager,
                        int[] appWidgetIds) {

        // Provider obsługuje WIELE (N) widgetów!
        final int N = appWidgetIds.length;

        // aktualizujemy każdego
        for (int i = 0; i < N; i++) {
            int appWidgetId = appWidgetIds[i];
            populateView(context , appWidgetManager ,
                        appWidgetId);
        }
    }
}
```

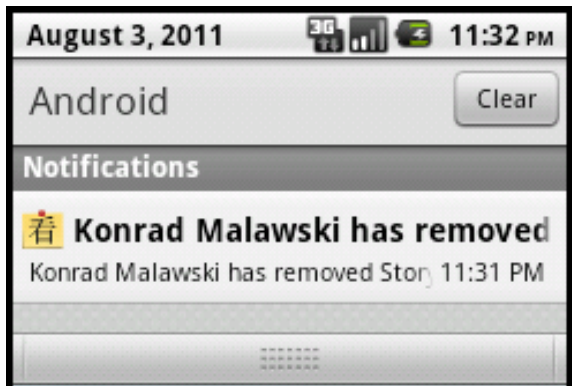
AppWidget - implementacja

```
private void populateView(Context context, AppWidgetManager manager) {  
    // Przygotowujemy intent do odpalenia "on click"  
    Intent intent = new Intent(context, ViewDetailsActivity.class);  
    PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, intent, 0);  
  
    // rejestrujemy onClickListener'a troszke inaczej:  
    RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.app_widget);  
    views.setOnClickPendingIntent(R.id.container, pendingIntent);  
  
    // aktualizujemy widok widżetu (prosimy menagera o to)  
    appWidgetManager.updateAppWidget(appWidgetId, views);  
}  
}
```

Notifications



Notificaion - przykład



NotificationManager

```
class MyActivity extends RoboActivity {  
    @Inject  
    NotificationManager notificationManager;  
}
```

NotificationManager

```
class MyActivity extends RoboActivity {  
    @Inject  
    NotificationManager notificationManager;  
}
```

Albo oczywiście **Service**.

NorificationManager - 1/3

```
int icon = R.drawable.ic_kanbanery;  
long when = System.currentTimeMillis();  
  
Notification notification = new Notification(icon, title,  
// ...
```

NorificationManager - 2/3

```
int icon = R.drawable.ic_kanbanery;  
long when = System.currentTimeMillis();  
  
Notification notification = new Notification(icon, title  
  
Intent notificationIntent = new Intent(this, ColumnsAct  
PendingIntent onClickIntent = PendingIntent.getActivity(  
  
// ...
```

NotificationManager - 3/3

```
int icon = R.drawable.ic_kanbanery;  
long when = System.currentTimeMillis();  
  
Notification notification =  
    new Notification(icon, title, when);  
  
Intent notificationIntent = new Intent(this,  
                                       ColumnsActivity.class);  
PendingIntent contentIntent = PendingIntent  
    .getActivity(this, 0, notificationIntent, 0);  
  
notification.setLatestEventInfo(context, title,  
                                msg, contentIntent);  
notification.flags = Notification.FLAG_AUTO_CANCEL;  
  
notificationManager.notify(ACTION_ID, // explain  
                           notification);
```