**BatteryAppWidget** (12/18/15)

// I finished reading about Annotation - @nonNull, @UiThread, @MainThread, @WorkerThread, @BinderThread

--------------------

BatterAppWidget extends AppWidgetProvider

// AppWidgetProvide extends BroadcastRecevier

**1. onReceive**(@NonNull Context, @NonNull intent)

if ACTION\_EXPLICIT\_UPDATE || ACTION\_BOOT\_COMPLETED

doExplicitUpdate(..)

if ACTION\_BOOT\_COMPLETED

need to start the alarm, get alarm interval from SP

2. **doExplicitUpdate**(..)

get all the appWidgetIds then call onUpdate(..)

3. **onUpdate(**..)

for each appWidgetId

call updateAppWidget(..)

4. **updateAppWidget**(...)

/\* these steps will be called the first time and every time appWidget updated, even

though I didn't add any new info to the widget layout, but it still do the work

unless, I can delete it is not the first time update, then I can eliminate some steps

or I have the configure activity, then I know that all subsequence update it is not

the first time call.

Maybe I do that, I will have configuration activity tell the user the default settings - level=40, 12 hr, and android notify only, and let them now they can change the default settings later, for now they can just click ok,

\*/

// then I can skip the RemoteViews update

// I may need to move this code to configuration activity

get RemoteViews from context.getPackageName() and R.layout.fragment\_show\_image

update the textview in the remoteViews

create intent with Action ="com.xfsi.batterycheckter.action.GOTO\_MAIN\_ACTIVITY");

create PendingIntent

bind this PendingIntent with onClick of widget

call widget manger to update the widget

appWidgetManager.updateAppWidget(appWidgetId, views);

get the current battery level

get level from SP

compare level => send out notifications - text, email, notify

**5. sendSms**

get phones from PS

then execute SendTextBG class in the background

**6. sendMail**

execute SendMailBG class in the background

**7. createMessage**

// this message is used for both text and email

get date&time

get current level and interval from SP

// get widget name why it comes from BatteryAppWidgetConfiureActivity

CharSequence widgetName = BatteryAppWidgetConfigureActivity.loadTitlePref(context, appWidgetId)

return composed message.

**8. generateNotification**

create icon for the notification

get current time

get NotificationManager

// touch battery icon => main activity

create Intent with MainActivity.class

create PendingIntent with CLEAR\_TOP | SINGLE\_TOP

create another PendingIntent for Level button => CheckLevelActivity.class

create another PendingIntent for interval button => TimeIntervalActivity.class

build NotificationCompat

set icon, when, text, ticker message, default sound, vibrate

add action for levelActivity and intervalActivity

setConentIntent to MainActivity

set Priority= high

notificationMgr.notify(..)

**9. errorNotify**

// this is used for when I can't send email out.

similar to 8. but without two buttons Level and time intervaal,

it still has battery icon, when touch go to MainActivity

**10. getBatteryLevel** => float current battery level

get one time battery value from the system

context.**getApplicationContext().registerReceiver(null**,

new IntentFilter(Intent.ACTION\_BATTERY\_CHANGED));

// I need to check out this info what work what not work

// register receiver to get one time Battery change event

// null means no receiver => now???

// context.registerReceiver(..) work on real device, but not on emaulator

// Intent batteryIntent = context.registerReceiver(null, new IntentFilter(Intent.ACTION\_BATTERY\_CHANGED));

/\* K This works on emulator

context.getApplicatonContext().registerReceiver(...)

The Context that is passed to onReceive() is blocked from calling registerReceiver()

even with a null BroadcastReceiver.

\*/