6/11/15

Ramblings on maintaining state in list when switch to new activity from manual mode && should the timer widget continue to work /run during this time?

Thinking of making better use of the Circuit class and the associated array maCircuits which is not being used as it should be...

* Does it make sense to only update the Circuit Array info upon onPause() or even onStop()?
* Then in onResume() or whatever I could restore the ListItem array / View info with these values
* Then make the Circuit Array the Singleton. I could create a Singleton Bundle if there is more data that should be saved.
* This approach would also mean minimal changes to the ManualMode Activity and CustomListAdapter which is very appealing (but only if it makes sense). I'd just need to add a few additional items to the array list.

In doing this, we have a Model, in the sense that it can be used for other Activities that do other things but need some of the same circuit info.

OK. For **Singleton** vs Extend Application, I think that I'm going to go with the Singleton. This decision is based on this StackOverflow entry (http://stackoverflow.com/questions/3826905/singletons-vs-application-context-in-android) and Dianne Hackborn (she's an Android framework engineer) comments to it. She continually insisted that the Singleton was the right approach.

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

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

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

5/21/15

How to configure OspiMessage class and what does what as far as Event, message strings, ResultListeners, EventBus posts, etc.

In ManualModeAct - who calls SendMessage or SendCommand??

1 - onClick for GET button calls SendCommand with the string from text field.

2 - onEvent (EventBus) calls SendMessage (event.message) when CircuitOnOffEvent is posted

1 - onStart() calls SendCommand ("/sn0") just to get things started.

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

CustomListAdapter -

* posts Events when CircuitOnOffButtons are toggled which then call / invoke onEvent() in ManualModeAct.
* EventBus.getDefault().post( new CircuitOnOffEvent( CircuitOnOffEvent.CIRCUIT\_OFF, mH.position+1));
  + This causes onEvent to be called once the CircuitOnOffEvent constructs a new "turn circuit x off" message.

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

onEvent() right now each Activity has its own onEvent() with different argument typed event - then calls SendMessage() Can we move this stuff to OspiMessage new class?

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

**Can these just be OspiMessages??** 6/15 \_ I made them extend OspiMessage

CircuitOnOffEvent - Basically only has a constructor and all that does is create a String called message which contains something like this : URL+STATION\_NUMBER+mCircuitNum+"="+mStatus+"&t=0"

ChangeProgramEvent - Basically only has a constructor and all that does is create a String called message which contains the full string needed to change a program schedule.

* might need to work the format of the string into different subclasses or possibly just by parsing the arguments into the constructor a little (or a lot).
* or maybe keep these since their formats are pretty specialized and still move the SendMessage() to OspiMessage?
* or can probably create a static method called onEvent or SendMessage(). I'd like to make onEvent() the singleton but since I'm using the default event bus, isn't that implied? If so, how to use it the way I want?

Currently:

SendMessage( String command)

{

new OspiGetResultsAsyncTask( activityContext,

OspiGetResultsAsyncTask.OSPI\_GET ).execute(command);

}

Is Context a problem if I put into OspiMessage class?? I think I might be OK, but not really sure. There are different ways to get the context:

* getApplicationContext(),
* getContext(),
* getBaseContext()
* or this (when in the activity class)

Also, I should be able to pass in the Activity context when I create the Message instance as I did for OspiGetResultAsyncTask.