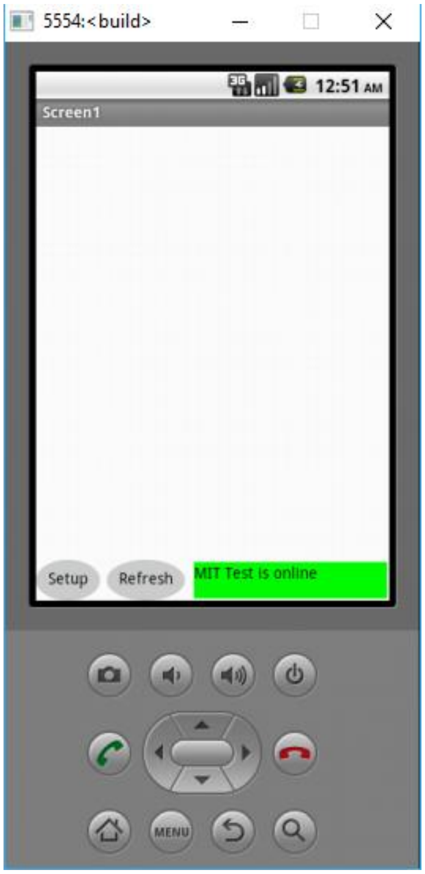
Particle App Template User Manual

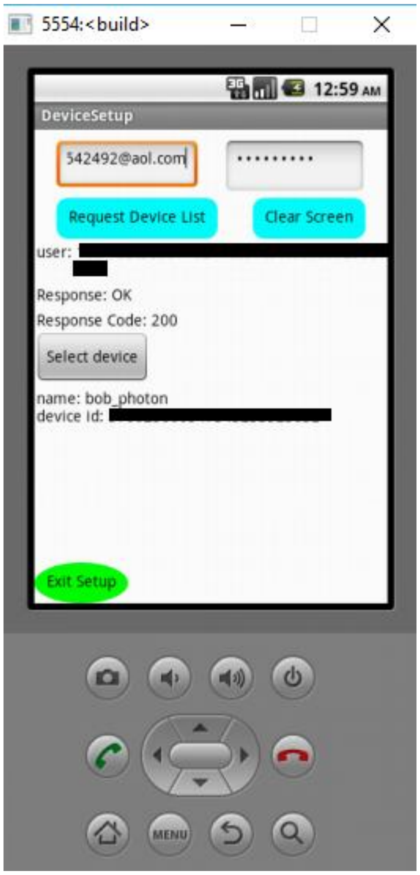
Version 1.0; date: 3/25/18

NOTICE: Use of this document is subject to the terms of use described in the document "Terms_of_Use_License_and_Disclaimer" that is included in this release package. This document can also be found at:

https://github.com/TeamPracticalProjects/Particle_App_Template/blob/master/Terms_of_Use_License_and_Disclaimer.pdf



MIT Test is online



542492@aol.com
•••••••••

user: [redacted]
Response: OK
Response Code: 200

name: bob_photon
device id: [redacted]

Table of Contents

Table of Contents

1. Introduction.

This project provides a Template for creating apps that can communicate with Particle (https://www.particle.io/) devices (e.g. Photon, Electron) over the Internet, via the Particle Cloud. The template, as well as apps that will be created from it, is written in MIT App Inventor 2 (http://ai2.appinventor.mit.edu). MIT App Inventor 2 (AI2) is an easy-to-use, graphical programming system that is completely web based and requires no complex IDE installation. At present, AI2 allows the user to create Android apps; however, the MIT team has announced plans to support iOS in 2018. Furthermore, the AI2 apps can be run on Windows and Mac OSX via readily available Android emulators for these operating systems.

This document describes how to use the Template itself. As such, it can easily be referenced from a User Manual for any app that is subsequently created from the Template.

This document does not describe how a developer can use the Template to create an app that communicates with Particle devices. The latter is the subject of a companion document "Publishing AI2 Apps" that is also published in this repository:

https://github.com/TeamPracticalProjects/Particle_App_Template/blob/master/Documentation/Publishing%20AI2%20Apps.pdf

2. Overview.

Figure 1 shows the contents of the two screens that comprise this Template. The first screen is called "Screen1" and is the opening screen for any app made using AI2 (unless, of course, the developer renames this screen). Most of "Screen1" is left unpopulated; it is intended that an app made from this Template will use this real estate for its visual components. However, the bottom of "Screen1" contains visual components that are part of the Template and these components should not be tampered with by an app developer who uses this Template.

The second screen is the "Device Setup Screen" and this screen is for the exclusive use of the Template. If an app developer needs more screen real estate than is available on "Screen1", the developer can always add additional screens to the app.

The following is a brief description of each visual component on the Template:

2.1. "Screen1" Visual Components.

"User App Area": This is the real estate on "Screen1" that is available for the visual components of an app that is created using this Template. In the AI2 Designer window, this area is within the layout component called "VerticalScrollArrangement1".

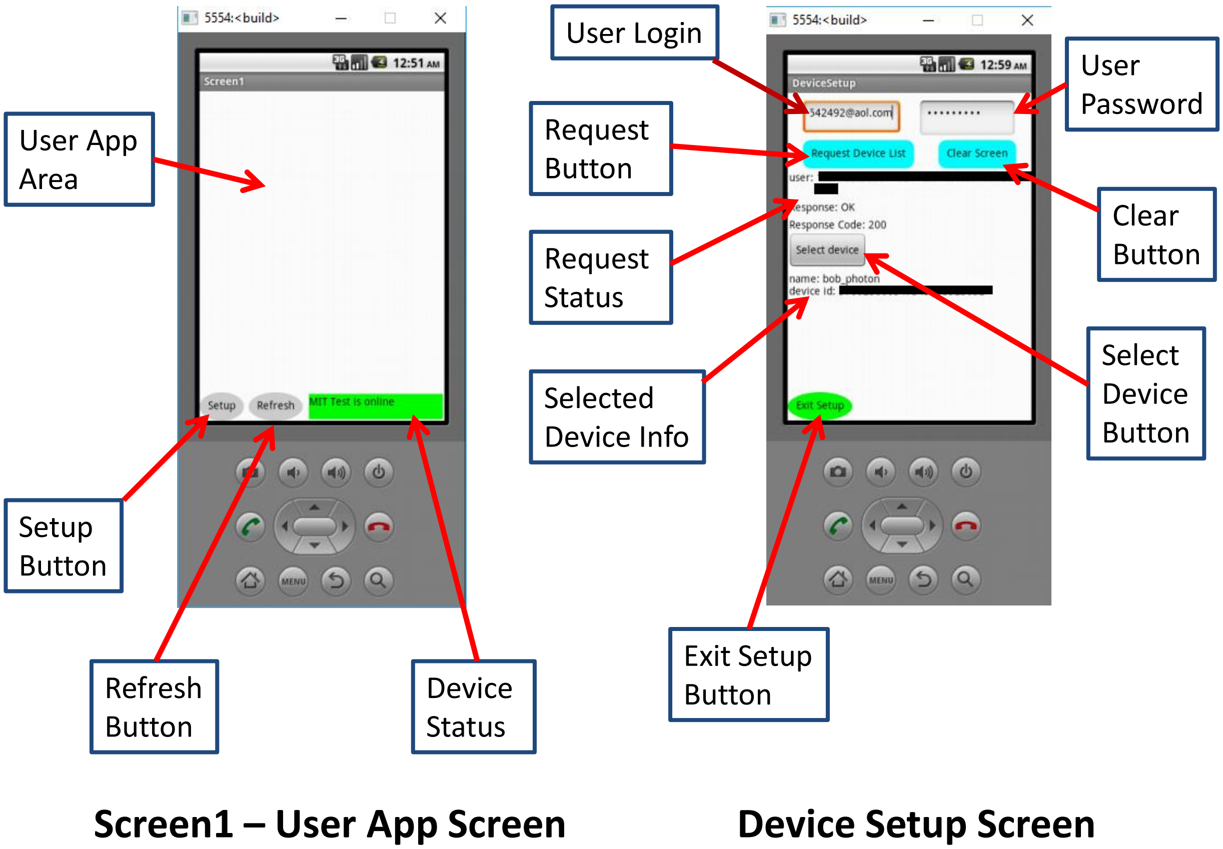


Figure 1. Overview.

<u>“Setup” Button</u>: When tapped, this button takes the app user to the “Device Setup Screen”.

<u>“Refresh” Button</u>: When tapped, the currently selected Particle device will be “pinged” over the Internet and its status (online, offline) will be displayed in the “Device Status” component of “Screen1”. A user may tap “Refresh” at any time that they wish to find out the current status of the selected device[1].

[1] Note that opening the app <u>may or may not</u> automatically ping the device. The Android OS caches screens from recent apps and if it decides to use a cached screen, the Template will not receive an event that it can use to trigger the device ping. The “Refresh” button can always be used to ensure that the device is pinged for its current status.

<u>“Device Status” area</u>: After the currently selected Particle device has been “pinged” over the Internet, its status (online, offline) displayed in this area. If the device is successfully pinged and online, the label will be green. If the device is offline or otherwise not responding to the ping, this label will be red. While the ping is in progress and the status is yet to be determined, this label will be yellow. It is perfectly ok to tap “Setup” when this label is yellow, e.g. if the user realizes that they selected the wrong device and wants to go immediately to the “Device Setup Screen” to change the device selection. The user need not wait for a ping to time out an offline or unavailable device.

2.2. “Device Setup Screen” Visual Components.

The entire “Device Setup Screen” and all of its components are reserved for the Template and should not be altered by an app developer.

<u>“User Login” textbox</u>: The user types their Particle account login in this textbox in order to log in to the Particle Cloud. Initially, this textbox is blank. However, the first time that a user logs into the Particle Cloud using this login, the login text is stored in the TinyDB component associated with the app and thereafter this textbox defaults to the last login used. The user can always type over this default, of course, but remembering the last login saves typing in most cases.

<u>“User Password” password-textbox</u>: The user types their Particle account password in this password-textbox in order to log in to the Particle Cloud. Initially, this textbox is blank. However, the first time that a user logs into the Particle Cloud using their password, the password is stored in the TinyDB component associated with the app and thereafter this textbox defaults to the last password used. The actual text of the password is obscured – a series of dots shows in place of the actual text. The user can always type over this default, of course, but remembering the last password saves typing in most cases.

<u>“Request Device List” button</u>: Tapping this button attempts to log the user in to the Particle Cloud (over the Internet) using the “User Login” and “User Password” supplied. The result of this login attempt will be displayed in the “Request Status” area of the screen, as described below.

<u>“Clear” button</u>: Tapping this button clears out the “Request Status” area of the screen. This may be useful if the user is unsure if the information displayed in the “Request Status” area of the screen is the result of an actual inquiry of the Particle Cloud or if it is just the residual result of some earlier login to the Particle Cloud.

<u>“Request Status” area</u>: Information obtained as a result of the user logging into the Particle Cloud is displayed on this area of the screen. If the login was successful, information similar to that shown on Figure 1 will be displayed. Specifically, the user’s Particle

“access_token” will be displayed next to “user: “, “Response: “ will show “OK” or whatever the Particle Cloud returns as its response, and the “Response Code: “ will show 200, or whatever the Particle Cloud returns.

A response code of 200 (or anything in the 200’s) is the standard HTTP response for success. The “User Login”, “User Password” and “Particle User Access Token” will be stored in the TinyDB at this time and will become the new default for these values.

Other response codes, particularly codes in the 400’s, represent an error in logging the user into the Particle Cloud. Such an error is generally the result of typing an incorrect Login or Password, but may also be the result of a problem with the Particle Cloud itself. Try *carefully* retyping the Login and Password to see if the problem was a typo. If this doesn’t work, open a web browser on your phone or computer and try to access the Particle Cloud this way in order to see if the Cloud is up and running correctly.

“Select Device” button: Whenever the “Device Setup Screen” is entered, this button displays the text “No devices” and tapping it brings up a blank screen (which may be dismissed using the phone’s “back” button). Upon successful “Request Device List”, this button text will change to “Select device” and tapping this button will open up a pick list of the user’s devices, as returned by the Particle Cloud. Tapping on any device in the pick list will select that device for use by the app. The “Selected Device Info” area of the screen will display the name and particle device ID of the selected device. The selected device name and ID will also be saved in the TinyDB at this time and will become the new default values.

“Selected Device Info” area: When a device is successfully selected, the Particle “Device Name” and Particle “Device ID” will be displayed here.

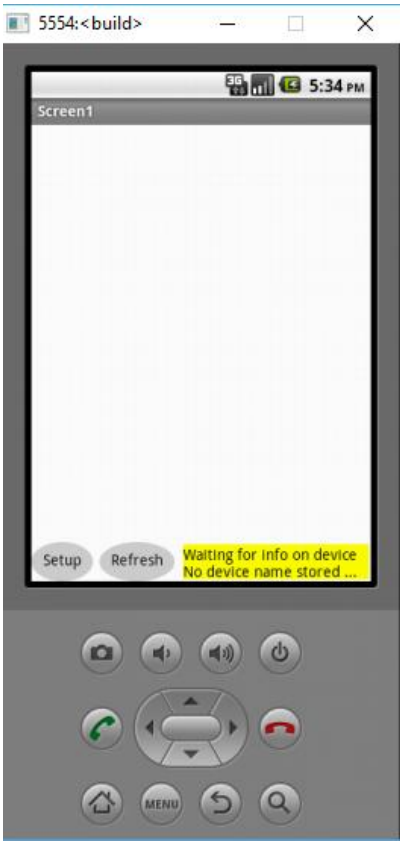
“Exit Setup” button: Tapping this button will exit the “Device Setup Screen” and transfer the user back to “Screen1”. “Screen1” will read the Particle User “access-token”, Particle “Device ID” and Particle “Device Name” from the Tiny DB and use this information to ping the device for its current status. The Particle User “access-token”, Particle “Device ID” and Particle “Device Name” will also be stored in “Screen1” global variables for use by the developer of an app based upon this Template.

3. Launching the App for the First Time.

When an app that is based upon this Template is launched for the first time, the app’s opening screen should look something like the picture in figure 2. Of course, figure 2 is a screenshot of the Template only. A real app will have other visible components in the “User App Area” of “Screen1”.

The important point for this discussion is the yellow “Device Status” area. As can be seen by the message in this area of the screen, this is because no device has previously been selected. Note that it is also possible to have this area show an error message and be colored red, indicating an error in trying to communicate with the Particle Cloud. Either way, this is because basic setup information has not yet been stored in the app’s TinyDB.

Therefore, the app user should ignore the “Device Status” area for now. Rather, the user should tap the “Setup” button to go to the “Device Setup Screen”. The “Device Setup Screen” will, at this point in time, look like the photo in figure 3.



Waiting for info on device
No device name stored ...

Figure 2. First Time Opening of the Newly Installed App.

Note the following in the photo in figure 3:

- The Particle “User ID” textbox is <u>not populated</u>.
- The Particle “User PW” password-textbox is <u>not populated</u>.
- The “Request Status” and “Selected Device Info” areas are blank.
- The “Select Device” button says “No devices”.

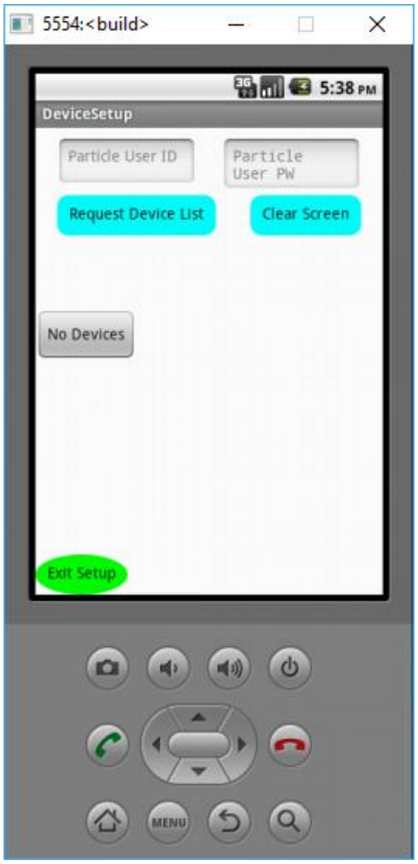
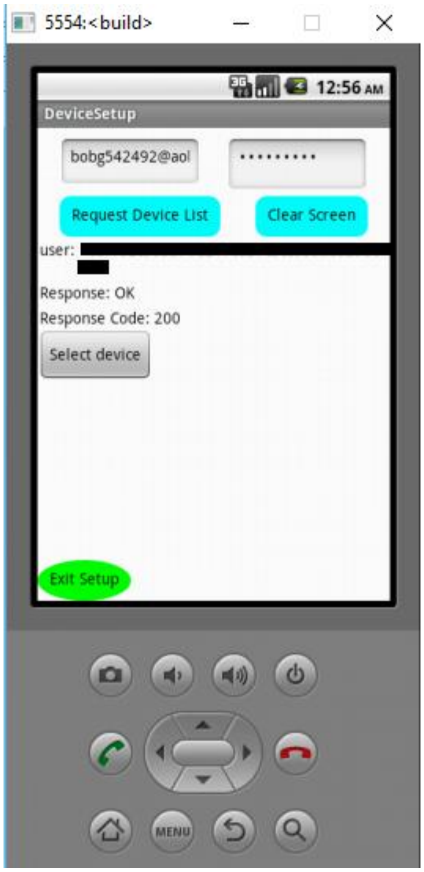


Figure 3. Device Setup Screen for Newly Installed App.

The user should now enter their Particle “User ID” and “User Password” in the appropriate text boxes. The user should take care to type these in correctly; otherwise, the user will get an error message in response to tapping the “Request Device List” button.

Tapping the “Request Device List” should result in a successful response from the Particle Cloud, similar to that shown in figure 4. Note that the “No devices” button text has changed to “Select device”. The Particle Cloud has returned a list of the user’s registered devices to the Template.



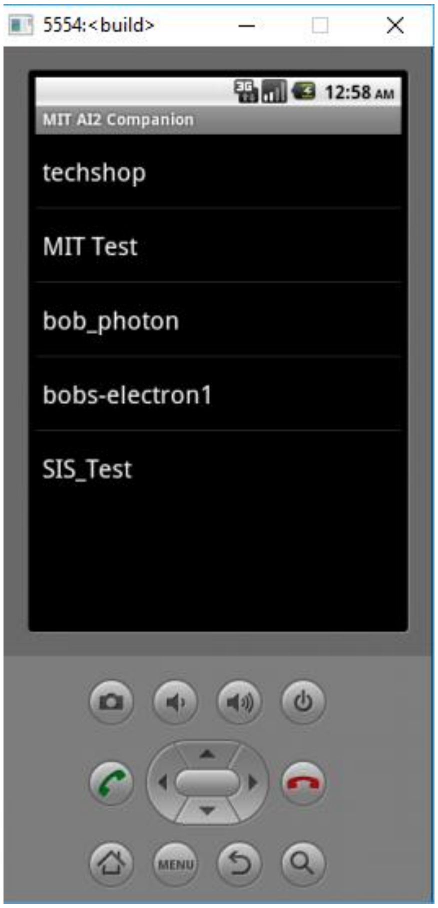
bobg542492@aol

••••••••••

```
user: [redacted]
Response: OK
Response Code: 200
```

Figure 4. Request Device List

The user may now tap the “Select device” button. A pick list of the devices in the user’s Particle account will appear, similar to the photo in figure 5.



techshop

MIT Test

bob_photon

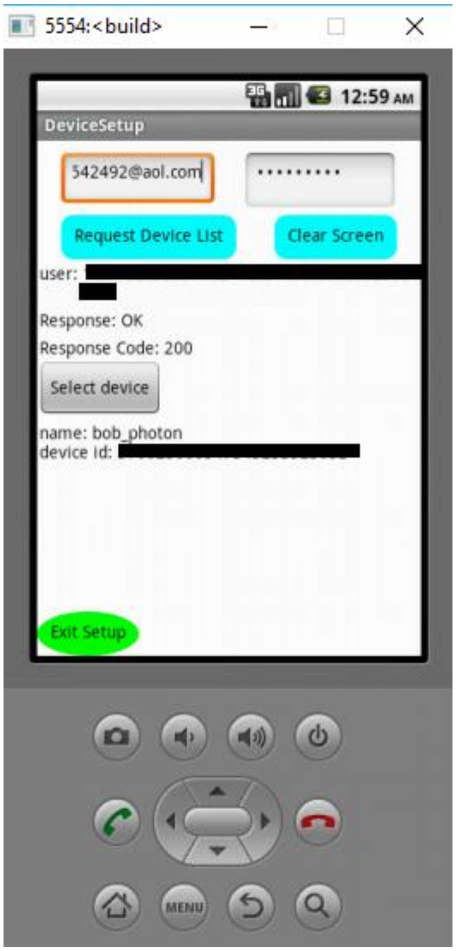
bobs-electron1

SIS_Test

Figure 5. Device Pick List.

The devices in this pick list will be those registered to the logged-in Particle account.

The user can now tap on the device that the user wants to use with this app. The pick list will disappear and the “Device Status Screen” will reappear with the selected device information displayed, similar to the photo in figure 6.



5554:<build>

12:59 AM

DeviceSetup

542492@aol.com

•••••••••

user: [redacted]

Response: OK

Response Code: 200

name: bob_photon
device id: [redacted]

Figure 6. Device has been selected.

Note in figure 6 that the Particle device “name” and the Particle “device ID” are displayed in the “Device Info” area of the screen.

At this point in time, the following information has been stored in the app’s TinyDB:

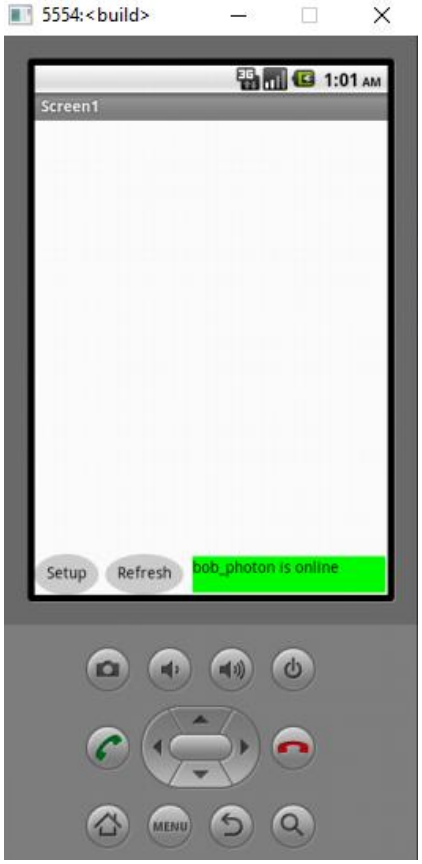
- The Particle “User Name”, as entered in the “User Login” textbox.
- The Particle “User Password”, as entered in the “User Password” password-textbox.
- The Particle User “access_token”, as shown next to “user: “ in the “Request Status” area of the screen.
- The Particle “Device Name” of the selected device, as shown next to “name: “ in the “Device Info” area of the screen.
- The Particle “Device ID” of the selected device, as shown next to “device id: “ in the “Device Info” area of the screen.

The app’s TinyDB data is persistent across all screens of the app and is also persistent between opening and closing sessions of the app. The last selected values will always be available in the app’s TinyDB unless the app is uninstalled and reinstalled (or unless code is added to remove the data in the TinyDB).

The user may now tap the “Exit Setup” button to return to “Screen1”. The Template will automatically “ping” the selected device using the Particle User “access_token” and the Particle “device ID” that has been stored in the TinyDB.

4. Subsequent Launching of the App.

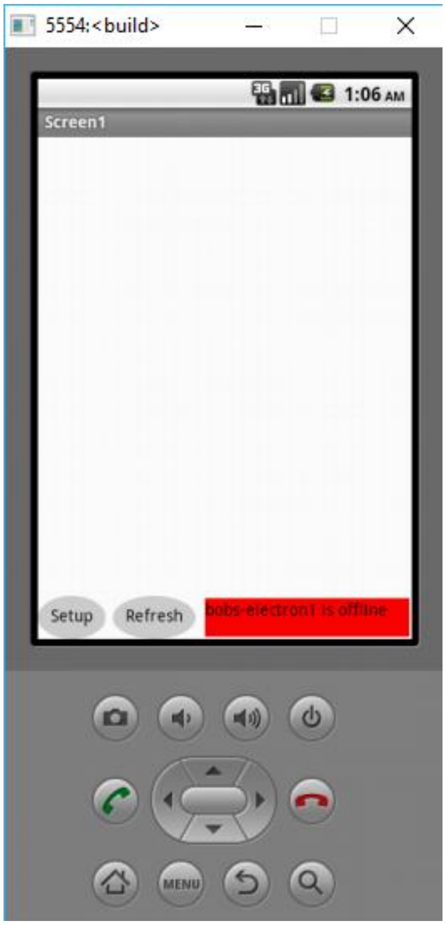
After exiting the “Device Setup Screen” or upon re-launching the app after closing it, “Screen1” will show, similar to the photo in figure 7.



bob_photon is online

Figure 7. Opening the App after initial setup.

Switching back from the “Device Setup Screen” to “Screen1” will always result in pinging the Particle Cloud for the current status of the selected device. If the selected device is online, the “Device Status” area on “Screen1” will turn yellow (during the ping) and then turn either green or red when the ping response is returned. If the selected device is online, the ping will be rapid (usually take just a second or two). However, if the device is offline, the ping will take a long time (about 30 seconds) to time out. This means that the “Device Status” area on Screen1 will be yellow during this time, and will eventually turn red, similar to the photo in figure 8.



bobs-electron1 is offline

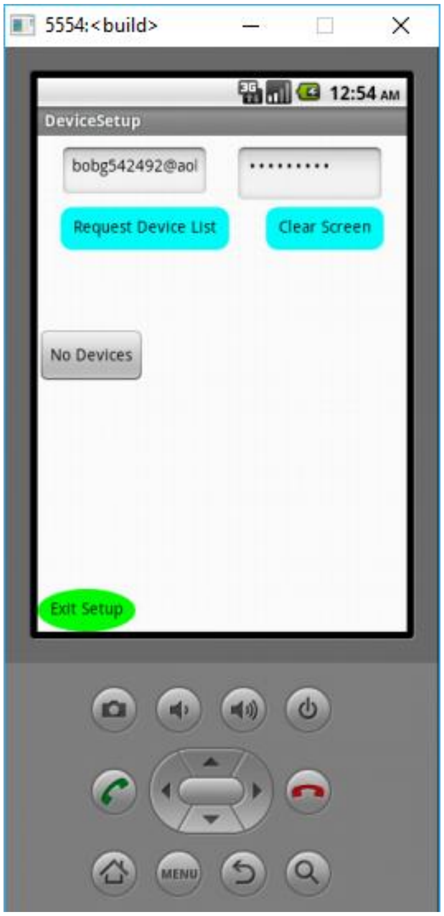
Figure 8. An offline device was selected.

If the “Device Status” area on “Screen1” is yellow for longer than a few seconds and the user realizes that they selected the wrong device, the user does not need to wait for this area to turn red. The user can immediately tap the “Setup” button to return to the “Device Setup Screen” and select another device, as described below.

A similar process <u>should</u> happen whenever the user opens the app. However, it <u>cannot be guaranteed</u> that opening the app will cause the previously selected device to be pinged. This is because the Android OS caches screens and sometimes uses a cached screen in lieu of re-opening the “Screen1” via code in the app. When this occurs, the app cannot know that is was re-launched and cannot therefore know to fetch the stored user and device information from the TinyDB and use the information to ping the device. However, the user can always tap the “Refresh” button to ping the currently selected device at any time.

5. Subsequent Device Setup.

After setting up the app the first time, subsequent visits to the “Device Setup Screen” will result in a display similar to the photo in figure 9.



5554:<build>

DeviceSetup

bobg542492@aol

••••••••••

Figure 9. Subsequent “Device Setup” screen.

The “User Login” and “User Password” fields will automatically be filled in using the data from the TinyDB. This is the last Login and Password data that was used to query the Particle Cloud for a list of the user’s registered devices. Usually, then, it is only necessary to click the “Request Device List” button to obtain the list of the user’s devices and move to the state shown previously in figure 4. Of course, the user can always change the Login and/or password; e.g. if using the app with another person’s Particle project.

Note that the Template does not persistently store the list of user devices. We believe it advisable to always login to the Particle Cloud and request the most current device list from it. There is little overhead in doing this and it helps to ensure that the user can only pick from the list of devices that Particle recognizes as belonging to the user’s account.