

Laboratory – The BroadcastReceiver Class

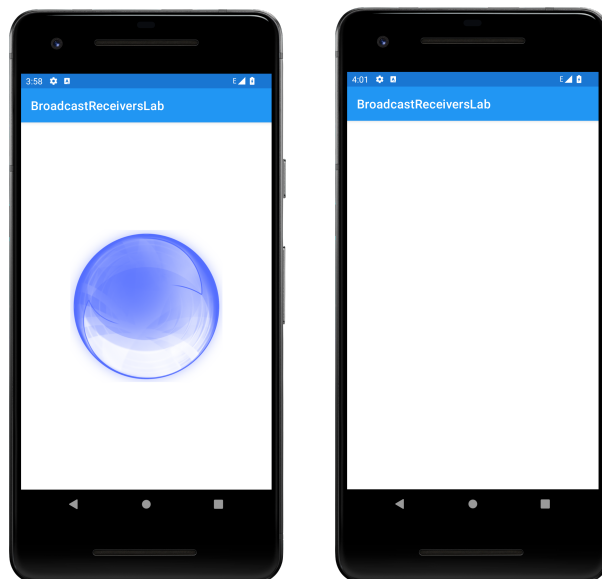
Learn about the BroadcastReceiver Class

Objectives

Familiarize yourself with the BroadcastReceiver class. Create an application that displays a pulsing image on the display (the image's visibility toggles on and off in a periodic fashion). When the device is plugged in to an AC power source the app toggles rapidly. Otherwise, the device toggles slowly. To do this your app will have a dynamically registered BroadcastReceiver instance that is listening for specific changes in the device's charging status. Once you've completed this Lab you should understand how to create BroadcastReceivers that respond to changes in device status.

The BroadcastReceiver Class

This lab involves an app called BroadcastReceiversLab. When it runs, the app displays a user interface like that shown below. The visibility of this image will toggle its visibility status four times per second when device is plugged in to an AC power source. Otherwise, it will toggle its visibility once per second.



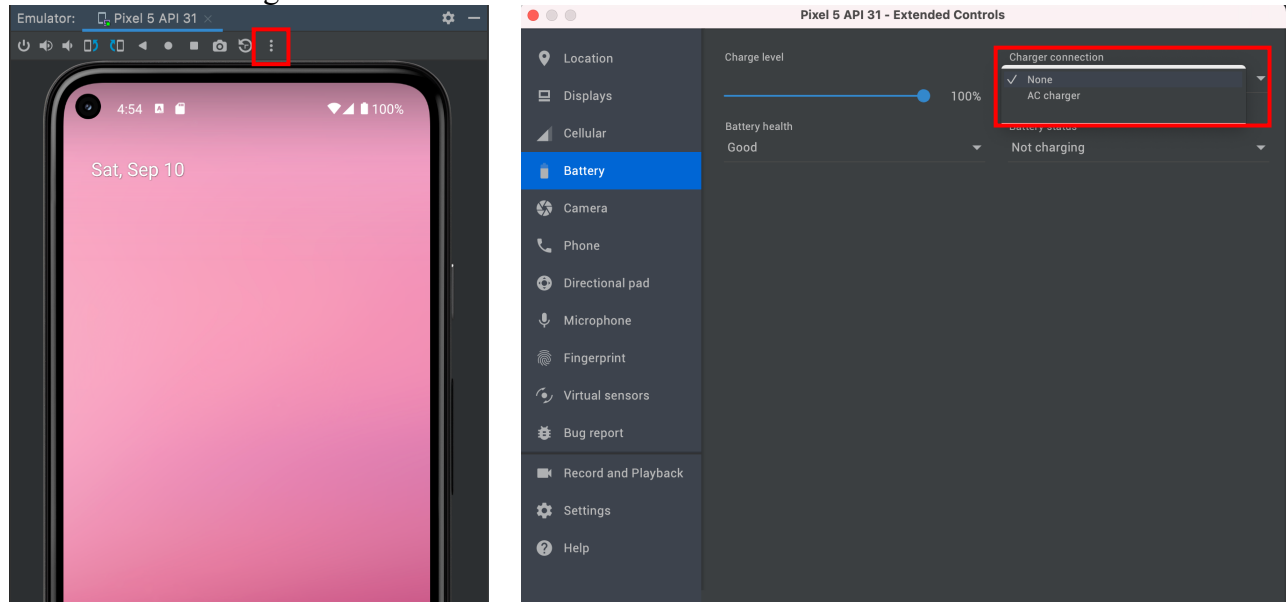
This application includes a single Activity called “MainActivity.” MainActivity displays the image and periodically toggles its visibility. The application also includes a ViewModel called PowerStatusViewModel that manages and publishes the device's current power connection status as LiveData. Finally, the application includes a BroadcastReceiver called PowerConnectionStatusReceiver that listens for changes to power connection status and updates the ViewModel when that status changes.

See the screencast, LabBroadcastReceivers.mp4, that's included in the Lab directory.

Testing

There are two test cases with several evaluation points. Each evaluation occurs at a step labelled “evaluation point”

* How to set the Charger Connection in Extended Controls



The first test case operates as follows:

1. Start the emulator. Open its Extended Controls, select the Battery tab, and set the Charger Connection to “None”
2. Start the BroadcastReceiver1Lab app in portrait mode
3. Check that the image’s visibility toggles once per second (evaluation point 1)
4. Go back to the Extended Controls and set the Charger Connection to “AC Charger”
5. Check that the image’s visibility toggles four times per second (evaluation point 2)

The second test case operates as follows:

1. Start the emulator. Open its Extended Controls, select the Battery tab, and set the Charger Connection to “None”
2. Start the BroadcastReceiversLab app in portrait mode
3. Check that the image’s visibility toggles once per second (evaluation point 1)
4. Put the app into the background (by hitting the home button at the bottom middle of the device)
5. Go back to the Extended Controls and set the Charger Connection to “AC Charger”
6. Bring the BroadcastReceiversLab app back into the foreground (using the Task Switcher button at the bottom right of the device)
7. Check that the image’s visibility toggles four times per second (evaluation point 2)

After completing your solution, you will record a screencast while performing the manual test. Afterward, you will submit your code and the screencast via git.

[Note] For this submission, you need to include the “Extended Controls” window in your screencast to show how you set the charger connection state. Do not use the Logcat screen recorder this time since it only records the screen of the device. Use an external recording tool to record both the device screen and the extended controls window as the instruction screencast shows.

Submission

When you are ready just commit your solution to your repo on GitLab by running the following commands:

```
% git add path/to/changed/files  
% git commit -m “completed Lab7_BroadcastReceiversLab”  
% git push origin main
```

Note: if you have not already pushed this branch to your repo on GitLab you will need to make a slight modification for this first time and run this instead:

```
git push -u origin main
```

This sets up tracking between your local branch and a branch with the same name on your repo in GitLab.

Some Implementation Notes

We are providing template code and layout resources for this application. Only modify the areas marked with the word `TODO`.

We have done our testing on an emulator using a Pixel 5 AVD with API level 31. To limit configuration problems, you should test your app against a similar AVD.