Notifications forwarder

Let us face an easy example of a home made Internet of Things project: we want to change the color of a Phillip's HUE light bulb when we receive an instant message from whatever service. The first option which comes to mind is to use IFTTT.
But here comes the big problem with this platform: for example there are different “if” conditions related to Facebook, but none for message received. One option now would be not to use IFTT: to tweak a XMPP client but if there is some change in the API, or we want to reuse the code for some other project it might be difficult.
When we keep thinking, there is one point in common among all project similar to IFTTT: it is very likely that any “if” conditions we might think of, exists already as a notification on the android platform. We just need to forward this information in some way.
Let us recapitulate for a moment: if we are able to forward the notifications we are not just facing the use case of instant messaging, but absolutely anything where there is an app related to. This might be Google+ informing that the pictures are updated or whatever.

If we look at the Android's API we see the existence of a Notification Listener Service, which let us write some code to be executed whenever a notification is posted. This project will build upon that. We will let the user define actions to execute whenever notifications arise. The preferences will let the user choose which actions to perform for which apps. For the first version of the project the actions will be send an email (defining subject and address to send to), or performing an HTTP GET (defining the url). Most certainly the second one is more promising since it can be used to communicate with a REST api.

This project can be further enhanced by allowing more actions, or adding options: for example in the url to access in the case of the HTTP GET reuse information from the notification. But for this first version it is enough to be the building block to connect the Android device to a server (which then will have to use the Phillip's HUE API for example).

Note: the app keeps discovering the installed apps as they publish notifications: it will start with an empty list and as notifications are posted add the apps which publish them to the list. But before this can happen, the user has to select in the settings of notifications that the app might access this information. Additionally, if the user want the app to send emails, then she has to access the settings of the app and give credentials for a Gmail account (which might require accepting less-secure connections in the browser's settings window).

References:
http://www.javacodegeeks.com/2013/10/android-notificationlistenerservice-example.html
And very extensive use of Sunshine's code and the Android's documentation