

GSoC'17 - Improve the Hawkular-Android client to support full alert setup and display.

Overview

Hawkular is a time-series database with alerting capability. It can monitor anything that can send formatted data to the Hawkular server. The project provides REST services for all kinds of monitoring needs from collecting rain sensors data and send an email notification to monitoring docker containers or application performance monitoring, it provides generic solutions to the common problems.

Powerful alerting is a must have in any monitoring system. In general, alerting can be used to notify people or to execute operations(restart a machine, autoscale an environment).

Present condition of the app

Hawkular Android client is a fully featured application with the following features:

- Authorization, which is done using the Hawkular server.
- Authentication is done using Aerogear Auth which really simplifies the scoping of access to the Hawkular services.
- Displays resources, metrics, metrics data in a chart(list).
- Alerts are also displayed in a list and have the feature to sort the alerts on the basis of time including hours, day, week, month and year.
- Option is provided to show and hide the alerts which are already resolved.
- Push notifications are available using Aerogear Push service.
- Personas are fully supported within an account.

Ideas to be implemented

Improve the support for Alerts to be triggered by implementing the following:

- Display of fired alerts (this already exists but extended data and markers in the graphs need to be displayed).
- Set up of new Alert triggers on individual metrics and groups of metrics. This can be done in a very UI-way (with on/off sliders etc).

Implementation

1.Setting up of new alert triggers for individual metrics and group of metrics in a good UI way including on/off sliders etc.

- Injecting the metrics in the Hawkular server which could be easily done using the metrics REST API. Could use stats for smaller intervals of time to send data to the server. Alerts would be triggered based on some specific conditions. Testing the alert triggers created on the basis of some sample data send to the server. An alert will be fired if the conditions for firing the alerts are met and the administrator/maintainer will be notified via email. Details for this process is provided [here](#).

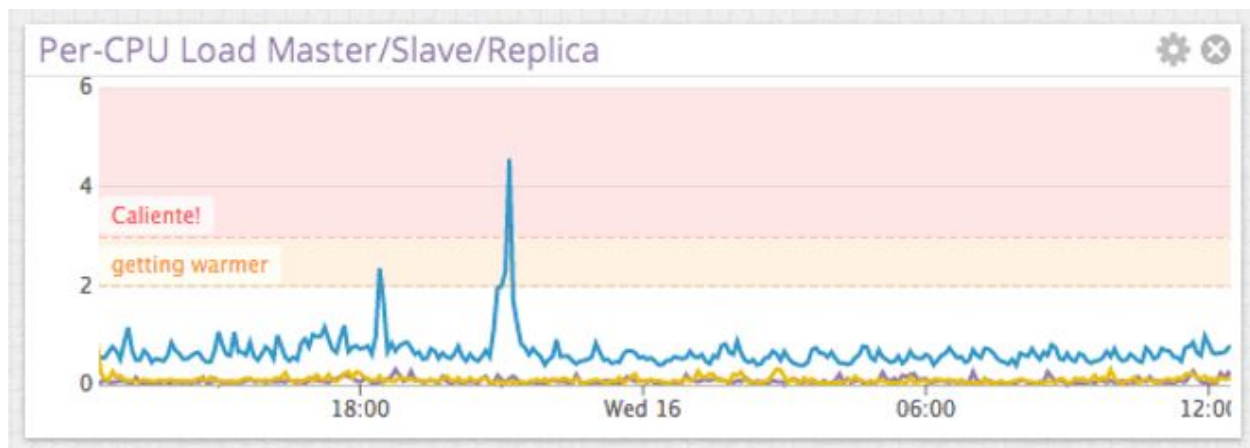
The data will be fetched in the form of a listview by connecting the android application to the remote web interface using the [Aerogear pipe module](#). The code snippet would be something like [this](#).

Subsequently, displaying of the alert triggers created on the server in a simple and elegant listview with the on/off sliders to present with a better UI would increase the user-friendliness and would be similar to the alerts displaying using the Aerogear pipe library/module. The display will look similar to the present UI but would have better UI orientation. Enabling the push notify service for

the application using the [Aerogear push services](#) along with [firebase push](#) notification service will be done thereafter.

2. Display of fired alerts (this already exists but extended data and markers in the graphs need to be displayed)

- The fired alerts display currently does not contain extended data of the alerts and markers in the graphs. When we receive alerts from the server that part of what makes them more useful than just an email full of exclamation points is the graph snapshot with shaded regions and lines(markers) showing the alert threshold and triggering value for the alerts, so the graph markers will thus make it easier to interpret the state of the system. The graphs for the metrics would look like the one below. Here markers will point the threshold and the triggering value. Alerts will be sent via email or push notification when the threshold value is crossed.



The [android graph library](#) and this [site](#) would be useful for making the amendments in the graph and attaining this state eventually. Data will be displayed in the details activity of the fired alerts specifying the threshold value and the value at the time of alert firing.

Timeline

April 10 - April 25(15 days)	<ul style="list-style-type: none">• An in-depth check of the code of the Hawkular-Android project.• Go through various components of the Hawkular project.
April 26 - May 8(13 days)	<ul style="list-style-type: none">• An in-depth learning of Aerogear modules including Aerogear pipe, Aerogear push notification service.• Going through firebase push service.
May 9 - May 28(20 days)	<ul style="list-style-type: none">• End semester college examinations.
May 30	<ul style="list-style-type: none">• Officially Coding period starts.
May 30 - June 12(14 days)	<ul style="list-style-type: none">• Create new alert triggers for individual and group of metrics.• Displaying of triggered alerts using Aerogear pipe module.
June 13 - June 21(7 days)	<ul style="list-style-type: none">• Enabling push notifications using Aerogear push and Firebase push services.• Sample curl calls to keep the code free from discrepancies.
June 21 - June 30(10 days)	<ul style="list-style-type: none">• Resolve issues if any regarding alerting and notifying process.• UI enhancement for the application.
July 1 - July 13(14 days)	<ul style="list-style-type: none">• Work on displaying the extended data in the fired alerts.• Improve the UI of the fired alerts activity.

July 14 - July 23(10 days)	<ul style="list-style-type: none"> • Displaying markers in the graphs to indicate the threshold limit and triggering value. • Improve the graph quality.
July 24 - August 1(9 days)	<ul style="list-style-type: none"> • Ensure the firing of respective alerts if the graphs cross the threshold value or both threshold and an excessive limit value.
August 2 - August 10(10 days)	<ul style="list-style-type: none"> • Buffer of 10 days has been kept for unpredictable problems.
August 11 - August 20(10 days)	<ul style="list-style-type: none"> • Review the work done till now. • Make sure everything works as expected. • Final submission of code.

Contributions to the Hawkular Android Project

- Shows message when the favourite triggers list is empty.[#92](#)
- placed the plus action in alerts fragment in overflow and enabled title change according to state.[#105](#)
- Added license header formatting and checking in gradle.[#114](#)
- Contents are properly shown on rotating the screen to landscape mode.[#151](#)
- Updated constructors annotated with `@VisibleForTesting`(still some work to be done).[#158](#)

About me

- I am a 1st year undergraduate student pursuing my degree in Computer Science Engineering from Amrita School of Engineering, Kerala, India. Apart from academics, I am avidly interested in contributing to Open Source software. Also, I am an active member of [FOSS@Amrita](#) which is an Open Source club of my university. My special interest lies in developing for android and was really fascinated by the idea of taking part in Google Summer Of Code, so I went through some Android projects and found this project really interesting to work on. I have worked with firebase and have basic knowledge of integrating with RESTFUL Web services in android. Currently learning Aerogear libraries which are the main libraries used in this project. I have also done some contributions to the project and plan to do more. I am familiar with the code base of the project would like to work on it to make it even better. I am an ever learner and keen to learn and explore new technologies.
- I would love to work and contribute to JBoss even after the summer.

Personal details

- Email id: sauravvishal8797@gmail.com
- Phone no: +917004512427
- Github handle: <https://github.com/sauravvishal8797>
- Telegram handle: @sauravvishal8797
(<https://t.me/sauravvishal8797>)

Location (City, Country and/or Time Zone): Kollam, India,
UTC+5:30

References:

- [1]
<https://aerogear.org/docs/guides/aerogear-android/pipe/>
- [2]
<https://aerogear.org/push/>
- [3]
<https://firebase.google.com/docs/notifications/>
- [4]
<http://www.hawkular.org/>
- [5]
<http://www.amritafoss.in/>
- [6]
<https://www.datadoghq.com/blog/customize-graphs-dashboards-graph-markers/#>
- [7]
<http://www.android-graphview.org/>
- [8]
<https://github.com/hawkular/hawkular-android-client>