Evaluation of Issue Thread Summaries by Large Language Models

Introduction

Thank you for participating in this research study. Your insights will provide invaluable data on the effectiveness of Large Language Models in summarizing software development issue threads.

Instructions

You will be presented with summaries generated by four different models. Please rate these summaries based on the following evaluation criteria.

Respondent Information

1. **Name:**
2. **Occupation:**
3. **Years of Experience in Software Development:**
4. **Have you previously worked with issue tracking systems?** Yes or No

---

Evaluation Criteria

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**

Summary Evaluation

Issue Thread 1 (Original Issue)

Issue Title: Allow starting the service without a notification
Issue description: # Description
Right now, even if you do not specify a notification with `TorServiceController.Builder`, when calling `TorServiceController.startTor()` a default one wlll be shown. I would like the ability to not show one at all. AFAICT this should be fine as long as there is a visible activity or another foreground service exists.
https://developer.android.com/about/versions/oreo/background#services
Context
<!--- Why is this change important to you? How would you use it? -->
<!--- How can it benefit other users? -->

This is important for me because I want my app to be integrated with Tor without explicitly telling the user via a notification (this is known and consented by the user inside the app). Tor is just one feature for me.
Possible Implementation
<!--- Not obligatory, but suggest an idea for implementing addition or change -->
I haven't looked at the code, but presumably it should be possible to just not create the notification if a flag is set.

Pull request title: Modify how the service is started as to not run in the Foreground.
Pull request Description: This PR contains the following: It now operates out of the Foreground by default, and only moves into the Foreground upon the implementing applications' task removal from the Recent Apps tray (required). bindService and startService are used in lieu of running in the Foreground such that notifications can be disabled by the Library user. Resolves #33 . The TorServiceController.Builder was modified by moving the NotificationBuilder to the ServiceNotification class to be instantiated from and used when initializing TorServiceController.Builder. NOTE: API breaking change Implements a queue for processing of actions submitted to TorService in a controlled, interrupt able manner for quickly reacting to user interaction and better reproducability using sealed class objects containing individual commands for the ServiceAction to be executed. Implements a BroadcastReceiver to execute NEW_ID, RESTART_TOR, and STOP ServiceActions, instead of piping them through startService
Various improvements to the notification (progress bar, better content text updates, etc)

Commit : clean up use of context as to not store it, add the ServiceAction string value to ActionCommand objects, add broadcasting of the ServiceAction strings while processing the ServiceActionObjects, add function for setting isAcceptingActions from an instantiated ServiceActionProcessor, add builder method for showNotification and remove startForegroundService startup, re-work ServiceNotification for stop/start foreground, progress bar, and track changes, implement ServiceNotification changes into ServiceEventBroadcaster, update documentation on showNotification method, add a destroy ServiceAction, modify get() method to throw exception, and add remove notification method

Comments: Hmm. Right now the service is being started by calling `startForegroundService` for API 26+, or `startService` for APIs 16-25. Utilization of `startForegroundService` requires a foreground notification to be present, otherwise it'll crash.

Also, every action called from the `TorServiceController.Companion` (start/stop/restart/newId) makes that `startForegroundService`/`startService` call, as it's cheaper (in terms of resources being used) and more secure than running a `BroadcastManager` to pass the Intents (Because you cannot use a LocalBroadcastManager and pending intents for notification actions).

What if an option existed during initialization to set the notification's importance level such that it will be in the user's notification tray, but will not be visible unless the tray is pulled down and then expanded?
, Actually, I'm re-thinking this whole process now to use Kotlin Flow. Will play with it this week to see what I can do ;-D.

- **Summary by Model 1:**

The software issue titled "Allow starting the service without a notification" revolves around the desire to initiate the `TorServiceController` service in an Android application without displaying a default notification. The issue was raised due to the need for more discreet operation of the Tor feature within an app, as discussed in the issue description. The user suggested a potential solution involving a flag to bypass notification creation. The subsequent pull request, "Modify how the service is started as to not run in the Foreground," addressed this by altering the service's operation mode. It proposed a method where the service operates out of the Foreground by default, switching to Foreground only upon task removal from the Recent Apps tray, thus enabling notification disablement. This involved significant changes, such as moving the NotificationBuilder to the ServiceNotification class, implementing a queue for action processing, and introducing a BroadcastReceiver for service actions. The commit further refined this approach by cleaning up context usage, revising the ServiceNotification for improved functionality, and updating documentation. However, comments in the thread highlighted concerns with the initial approach of using `startForegroundService` and `startService`, suggesting the need for an alternative method for starting the service that could circumvent the compulsory foreground notification for certain API levels. A possible shift towards utilizing Kotlin Flow was also mentioned for future exploration. Overall, while substantial progress has been made towards resolving the issue, it appears to remain unresolved, with ongoing discussions and potential revisions in the works.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**

- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - ○ **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - ○ **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - ○ **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - ○ **Rating scale: 1-10**

- **Summary by Model 2:**

The issue titled "Allow starting the service without a notification" was raised to address the default behavior in the TorServiceController library, where even if no notification was specified, a default notification would be displayed when starting the service. The user wanted the option to not show any notification, as long as there was a visible activity or another foreground service present. This change was important to the user as they wished to integrate Tor into their app without explicitly notifying the user via a notification. The suggested implementation was to add a flag that would prevent the creation of a notification. A pull request titled "Modify how the service is started as to not run in the Foreground" was submitted to address this issue. The pull request proposed changes to operate the service out of the foreground by default and move it into the foreground only when necessary, allowing users to disable notifications. The pull request also included various improvements to the notification system. As for the comments, there was a discussion about the usage of `startForegroundService` and `startService`, as well as the possibility of setting the notification's importance level. Additionally, the user mentioned reconsidering the process and exploring Kotlin Flow as an alternative solution. The issue has been resolved with the implementation of the pull request, which addresses the user's request to allow starting the service without a notification.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - ○ **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - ○ **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - ○ **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - ○ **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - ○ **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - ○ **Rating scale: 1-10**
    - ○

- **Summary by Model 3:**

In this GitHub issue thread, the developer wants to allow starting the Tor service without displaying a notification. Currently, a default notification is displayed whenever `TorServiceController.startTor()` is called, regardless of whether a notification object is specified in the `TorServiceController.Builder`. The developer suggests modifying the implementation so that no notification is created if a specific flag is set. However, they also note that utilizing `startForegroundService` for API levels 26+ requires a foreground notification to be present, which could cause crashes. They propose creating an option to set the notification's importance level such that it will be in the user's notification tray but won't be visible unless the tray is pulled down and expanded. There is currently no resolution to this issue.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**
- **Summary by Model 4:**

The issue titled "Allow starting the service without a notification" discusses the possibility of starting the Tor service without displaying a notification. The current implementation always shows a notification, even if none is specified, which the reporter finds unnecessary and intrusive. They propose adding a flag to suppress the notification. The discussion mentions that hiding the notification might not be feasible due to Android restrictions, but an alternative approach could be to create a custom notification that's not displayed by default. The associated pull request, "Modify how the service is started as to not run in the Foreground," addresses the issue by modifying the service to operate outside of the foreground by default. It utilizes the `bindService` and `startService` methods instead of `startForegroundService` to avoid showing a notification. Additionally, the pull request includes various improvements to the notification, such as a progress bar and better content text updates. In the comments, the discussion revolves around the challenges of working with Android's notification system, particularly the requirement for a foreground notification when using `startForegroundService`. One suggestion is to allow the notification's importance level to be set so that it's accessible in the notification tray but won't be visible until expanded. Another proposal involves leveraging Kotlin Flow for a potential solution. Overall, the issue remains open, with no clear resolution yet.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**

Issue Thread 2 (Original Issue)

Issue Title: Alarm is not triggerd: mobile number format.

Issue Description: Hi,

I have tried your App!

But I was not able to trigger the alarm.

1th Mobile: I didn't change the trigger word and entered my the Destination mobile number.

2nd Mobile: Unchanged triggerword. I entered the nr. from 1th: With no white Spacey and with Out the international prefix. Is that the right format?\

The SMS goes out and is recieved. But no alarm is triggered. I think the problem is the format of the mobile number.

Any advice? Testet on Pixel 3XL Android 10 and Galaxy S4 with Android 7.1.2

Comments: Ok, the Nr Format is +<countycode><number without white space>

I know the problem why the alarm is not triggered:

When an SMS arrives in content and the mobile number is checked.

1. Content check works.

2. But there is a bug in the code that checks the mobile number:

`if (contacts.contains(msg.getOriginatingAddress()) && msgs.contains(msgParts[0]))`

contacts is in such a format: +1XXXXXXXXXX;<Name>;General

but msg.getOriginatingAddress() returns only the mobile number, therefore the if-clause is always false and no alarm every triggered., Hello.

Thanks for your issue and PR #32. Unfortunately this project is currently unmaintained and will face a complete rewrite. Sadly I have to close your PR for now but I'm really grateful for it. If you are still interested at the time the rewrite will happen I'll happily review your upcoming contributions., any estimated time of when will the project be rewritten?

given the sparse dates among the messages above, i would assume it's still abandoned.

such a simple and wonderful idea, i'm willing to try and pick it up if you're going to keep it unmaintained.

Pull Request Title : Fixes the triggering

Pull Request Description: The actual version in this pull request fixes that the alarm is never triggered.

The problem is fixed by removing name and contact group before string compare

Commi message: fix compareson of the mobile number, fix compareson of the mobile number, Merge branch 'master' of github.com:Scyten/PanicTrigger into master.

- **Summary by Model 1:**

The issue titled "Alarm is not triggered: mobile number format" pertains to a bug in an app where alarms fail to activate upon receiving an SMS with a specific trigger word and mobile number. The user described their testing scenario using two mobile devices, highlighting that despite receiving the SMS, the alarm did not trigger. The issue was identified to be a formatting mismatch between the stored contact format and the format of the originating address of the received message. The stored contacts were in the format "+<countrycode><number without white space>;<Name>;General," while the `msg.getOriginatingAddress()` returned only the mobile number. This discrepancy led to the failure of a conditional check in the code, preventing the alarm from triggering. In response, a pull request titled "Fixes the triggering" was created, aiming to resolve this issue by adjusting the string comparison logic to ignore the name and contact group details. The commit associated with this pull request specifically addressed the mobile number comparison issue.

However, a comment in the thread revealed that the project was unmaintained at the time and faced a complete rewrite, leading to the temporary closure of the pull request. The contributor who raised the issue expressed willingness to assist in the project, indicating an ongoing interest in resolving the bug despite the project's status. As a result, while a technical solution was proposed and partially implemented, the issue remains unresolved due to the project's unmaintained status and pending complete rewrite.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**

- **Summary by Model 2:**

The issue titled "Alarm is not triggered: mobile number format" reported a problem with the app not triggering alarms when specific mobile numbers were used in SMS messages. After some investigation, it was discovered that the issue was related to the format of the mobile number, where the code was not correctly checking the number due to a discrepancy in how contacts were formatted versus the originating address from the SMS. The comments confirmed this problem and highlighted the project's unmaintained status, with the author expressing gratitude for a pull request (#32) but stating that the project would undergo a complete rewrite. The pull request titled "Fixes the triggering" aimed to address the issue by removing the name and contact group before comparing the mobile number, ensuring that alarms would be triggered correctly. However, the issue remains unresolved, as the project is currently unmaintained, and the pull request was closed in anticipation of a future rewrite, with no estimated timeline provided for the rewrite.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
    -

- **Summary by Model 3:**

The Panic Trigger app allows users to send an SMS to their emergency contacts to alert them of a panic situation. One user reported that the alarm was not triggered despite sending the correct SMS. After investigation, it was found that the issue was caused by a bug in the code that checked the mobile number format. The bug resulted in the comparison of the mobile number returning false, preventing the alarm from being triggered. To resolve the issue, the pull request proposed by the user removed names and contact groups before comparing the mobile numbers, allowing the alarm to be triggered correctly. However, due to lack of maintenance, the project is currently abandoned and may undergo a complete rewrite in the future.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
- **Summary by Model 4:**

The issue titled "Alarm is not triggered: mobile number format" describes a problem where the alarm is not triggered when an SMS is received due to a bug in the code that checks the mobile number. The bug causes the if-clause to always return false, thereby preventing the alarm from being triggered. The issue was reported on two different devices running Android 10 and 7.1.2. A pull request was submitted to fix the problem by removing the name and contact group from the mobile number comparison. However, the maintainer of the project closed the pull request citing that the project is currently unmaintained and will undergo a complete rewrite. The maintainer expressed gratitude for the contribution but did not provide an estimated timeline for the rewrite. Interestingly, someone else offered to take over the project since it seemed abandoned, suggesting they were willing to continue its development. Ultimately, the issue remains unresolved.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**

Issue Thread 3 (Original Issue)

Issue Title: Decide which builders fields are required to create Publisher and Subscriber objects.

Issue Description: We should decide which fields are required to build a Publisher and a Subscriber, and which can be optional. Issue is synchronized with this Jira Task by Unito.

Comments: I am looking at the Android version right now, and it looks to me like the set of fields is actually already correct for the Publisher: ably configuration map configuration routing profile context

On the subscriber side, however, I think we need to do the following changes based on the code sample from the Readme:

Required fields: connection configuration trackingID locations handler

Optional: resolution asset status handler @QuintinWillison wdyt?

Pull Request Title : Handle routing profile transportation mode in Publisher SDK.

Pull Request Description: Ive added the support for specifying a transportation mode routing profile for the DefaultPublisher. Ive also renamed the TransportationMode to RoutingProfile to be consistent with Mapbox terminology. Commit message: Handle setting transportation mode for publisher SDK, Rename TransportationMode to RoutingProfile, Remove inert assertion as routingProfile is never null, given it has a default value., Add documentation to routing profile, Handle setting routing profile with events queue, Merge remote-tracking branch origin/main into feature/publisher-mode-of-transport, Adjust changing routing profile to the new interface, Fix integration test after introducing the routing profile, Move Mapbox API usage to internal method., Remove pointless callback, simply changing the routingProfile property to look synchronous.

- **Summary by Model 1:**

The issue titled "Decide which builders fields are required to create Publisher and Subscriber objects" focuses on determining the essential and optional fields for building Publisher and Subscriber objects in a software development context. The discussion is synchronized with a Jira Task managed by Unito. Through analysis, it was identified that the Publisher's existing fields, including ably configuration, map configuration, routing profile, and context, were deemed appropriate. For the Subscriber, the required fields were proposed to be connection configuration, trackingID, locations handler, with resolution asset status handler being optional. The pull request titled "Handle routing profile transportation mode in Publisher SDK" contributed to this topic by adding support for specifying a transportation mode (renamed to RoutingProfile for consistency with Mapbox terminology) for the DefaultPublisher. The commit associated with this pull request included several changes, such as renaming TransportationMode to RoutingProfile, ensuring the routingProfile is never null by giving it a default value, adding documentation, and integrating the routing profile with events queue. Additionally, it involved internal method adjustments, interface changes, integration test fixes, and the removal of an inert assertion and a redundant callback to make the routingProfile property appear synchronous. This discussion and the pull request demonstrate an active effort in refining and specifying the necessary and optional fields for Publisher and Subscriber objects, indicating progress towards resolving the issue, albeit the final decision on field requirements remains implicit in the summary provided.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**

- **Summary by Model 2:**

The issue titled "Decide which builders fields are required to create Publisher and Subscriber objects" focused on determining which fields should be mandatory and which could be optional when building Publisher and Subscriber objects. In the comments, it was noted that for the Android version, the set of fields for the Publisher appeared to be correct, including ably configuration, map configuration, routing profile, and context. However, on the Subscriber side, changes were proposed, with certain fields like connection configuration, trackingID, locations, and handler being marked as required, while resolution, asset, status, and additional handler were considered optional. A related pull request titled "Handle routing profile transportation mode in Publisher SDK" was submitted to introduce support for specifying a transportation mode routing profile for the DefaultPublisher. The pull request aimed to make the routing profile more versatile and consistent with Mapbox terminology. However, the status of the issue is unclear from the provided information, as it doesn't explicitly state whether the changes have been accepted and implemented or if further discussions are needed.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
    -

- **Summary by Model 3:**

The issue revolves around determining which fields are necessary to construct both Publisher and Subscriber objects. The current Android version appears to have the proper field set for Publisher construction, whereas additional adjustments must be made for Subscribers. QuintinWillison suggested requiring "connection," "configuration," "tracking ID," "locations," and "handler" for Subscribers, with "resolution," "asset status," and "handler" being optional. A pull request proposes adding support for specifying a routing profile for the DefaultPublisher, renaming TransportationMode to RoutingProfile, handling routing profiles with event queues, and making various other modifications. The pull request also includes documentation and integration tests. The issue remains open until these suggestions are implemented and tested.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**

- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**
- **Summary by Model 4:**

The issue discussed is about deciding which fields are required to create Publisher and Subscriber objects. The Android version is examined, and it's determined that the current set of fields for Publisher is correct, but Subscriber needs modifications based on the code sample from the Readme. Required fields for Subscriber include connection, configuration, trackingID, locations, and handler. The pull request, "Handle routing profile transportation mode in Publisher SDK," adds support for specifying a transportation mode routing profile for the DefaultPublisher and renames TransportationMode to RoutingProfile for consistency with Mapbox terminology. The pull request also removes an inert assertion, adds documentation, and makes several other changes related to handling routing profiles. The issue remains open, awaiting further discussion and resolution.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**

Issue Thread 4 (Original Issue)

Issue Title: Improve fallback logic in resolution policys resolve method.

Issue Description: override fun resolve(request: TrackableResolutionRequest): Resolution

If the intention was to implement the logic from here: https//github.com/ably/ably-asset-tracking-android/pull/84/files

then this method is currently not doing that.

More specifically, this method right now is doing very simplistic fallback logic, where essentially if there is no resolution provided for given state, it will just disregard other resolution constraints. This is also related to the fact that I believe that DefaultResolutionSet class members should be optionals - user may provide some of them, may provide all of them, etc.

I think this is something we can address in the future, not necessarily in this PR

Originally posted by kavalerov in https//github.com/ably/ably-asset-tracking-android/pull/97#discussion_r544203902

Issue is synchronized with this Jira Task https//ably.atlassian.net/browse/ATS-150 by Unito.

Pull Request Title : Default resolution policy implementation.

Pull Request Description: Ive almost implemented the default resolution policy. Its missing tests (theyll be added in another PR) and temporal threshold (itll be added in upcoming commits) so its currently in the draft state. Probably we should define places in which the resolution should be refreshed because currently it happens only when the threshold is reached. Im implementing it in a hurry due to our deadline so please be extra cautious when checking this PR.

Commit message: Change TrackableResolutionRequest to a data class, Add method for getting correct resolution from DefaultResolutionSet, Assign levels to Accuracy for comparing them, Add battery data provider, Add helper functions for distance and time difference calculations, Add default resolution policy implementation, Make resolution policy factory a required builder param, Add resolution policy, hooks and methods to the publisher, Add refreshing resolution in Publisher, Use resolution policy hooks trackables methods

Comments: Clarification:

Right now there is a constructor that takes only one, default Resolution in the DefaultResolutionSet . This constructor sets the values for all elements of the set. However, the issue with this is that we now dont know from the outside whether certain values have been provided by user, which makes it not possible to implement custom fallback logic in the ResolutionPolicy.

If we want application developer to be able to define their own fallback logic as part of the resolve implementation - we need to provide them with knowledge on which resolutions were explicitly set by the user.

We could get away with baking in one fallback logic in the DefaultResolutionSet by providing multiple constructors, but I dont think it is the right option. , Kacper Kluka commented:

As Ivan mentioned, with our current implementation were unable to implement this fallback logic. We have at least 2 options here and their main difference is who is responsible for handling the fallback logic.

1. DefaultResolutionSet handling the fallback logic. We could make some or all of the resolutions optional (nullable) and then inside getResolution(isNear: Boolean, hasSubscriber: Boolean) we would have the fallback logic that would choose the correct resolution based on the params. This will probably require us to add some sort of defaultResolution for the given resolution set or mark one of those resolutions as the default one (that cant be null) so we will always return a resolution.
2. ResolutionPolicy handling the fallback logic. We could change the DefaultResolutionSet so it would accept nulls for its resolutions and then DefaultResolutionSet.getResolution() method would return null in some cases. Then we could use that information in the ResolutionPolicy.resolve() method to implement our custom fallback logic.

Both solutions would require us to change the DefaultResolutionSet structure and interface. I think we should discuss which approach is better for us., Ivan Kavalerov commented:

I think DefaultResoluitonSet (or DefaultResolutionPolicy) should handle the fallback logic, so that users could provide their own logic. Paddy Byers / Quintin Willison do you agree?, As weve discussed on todays call we will go with approach number 2 (putting the fallback logic into ResolutionPolicy).

- **Summary by Model 1:**

The issue titled "Improve fallback logic in resolution policy's resolve method" centers on enhancing the fallback mechanism in the `resolve` method of a resolution policy within a software application. The primary concern, as highlighted in the issue description and synchronized with a Jira Task (ATS-150), is that the current implementation exhibits simplistic fallback logic, ignoring other resolution constraints when a specific state lacks a resolution. The discussion suggests that `DefaultResolutionSet` class members should be optional, allowing flexibility in the resolutions provided by the user. This topic is further elaborated in the pull request "Default resolution policy implementation," which, while focusing on implementing the default resolution policy, lacks tests and a temporal threshold at its current stage. The commit messages indicate various enhancements, including changing `TrackableResolutionRequest` to a data class and adding a method for obtaining the correct resolution from `DefaultResolutionSet`.The comments section brings clarity to the issue: the current constructor in `DefaultResolutionSet` only allows a single default resolution, making it challenging to discern user-provided values and impeding the implementation of custom fallback logic in `ResolutionPolicy`. Two potential solutions are proposed: having `DefaultResolutionSet` handle the fallback logic by making resolutions optional and determining resolution based on parameters, or allowing `ResolutionPolicy` to manage the fallback logic by accepting nulls in `DefaultResolutionSet` and implementing custom logic in `ResolutionPolicy.resolve()`. After discussions, the decision was made to adopt the second approach, where the fallback logic would be incorporated into `ResolutionPolicy`. This decision indicates a move towards a more flexible and user-customizable resolution mechanism, addressing the initial concerns raised in the issue. However, the resolution of the issue is not explicitly stated, suggesting ongoing development and refinement in implementing this chosen approach.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**

- **Summary by Model 2:**

The issue titled "Improve fallback logic in resolution policy's resolve method" addressed a problem with the logic in the `resolve` method of the resolution policy. The issue discussed the need to implement more robust fallback logic and the fact that the current implementation didn't match the intended logic from a previous pull request. Specifically, the existing method had simplistic fallback logic, which would disregard other resolution constraints if there was no resolution provided for a given state. The issue also mentioned that certain members of the `DefaultResolutionSet` class should be optional to allow users more flexibility in providing resolution values. A related pull request titled "Default resolution policy implementation" aimed to address this issue by implementing the default resolution policy. However, the pull request was marked as in the draft state, and it lacked tests and a temporal threshold. The comments discussed different approaches to handle the fallback logic, including whether it should be in the `DefaultResolutionSet` or the `ResolutionPolicy`, ultimately opting for the latter. The issue appears to remain unresolved as the pull request is still in progress.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**
  -

- **Summary by Model 3:**

The issue concerns improving the fallback logic in the resolution policies' resolve method. Specifically, the current implementation does not take into account the possibility of different resolutions being provided by the user, making it difficult to implement custom fallback logic. Two potential approaches have been identified: having the DefaultResolutionSet handle the fallback logic or having the ResolutionPolicy handle it. After discussion, it was decided to put the fallback logic into the ResolutionPolicy, allowing users to provide their own logic. Additionally, the DefaultResolutionSet will accept nulls for its resolutions, and the ResolutionPolicy will use this information to implement custom fallback logic. These changes will require updating the DefaultResolutionSet structure and interface.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
- **Summary by Model 4:**

The issue discussed is about improving the fallback logic in the resolution policy's resolve method. The current implementation has a simplistic fallback mechanism, which doesn't take into account user-provided resolution constraints. The proposed solution is to make the resolutions in the DefaultResolutionSet optional, allowing the user to specify their own fallback logic. Two approaches are considered: either the DefaultResolutionSet handles the fallback logic or the ResolutionPolicy does. After a discussion, the team decides to go with approach number 2, putting the fallback logic into the ResolutionPolicy. The pull request is still in draft state, missing tests and a temporal threshold, and the commit message mentions several changes, including making the resolution policy factory a required builder parameter, adding resolution policy hooks, and refreshing resolution in the publisher. The issue remains open, waiting for further development and testing.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**

Issue Thread 5 (Original Issue)

Issue Title: Reduce amount of fields in DefaultPublisher.

Issue Description: We could think about reducing the number of fields (particularly maps and sets) that are currently in the DefaultPublisher https//github.com/ably/ably-asset-tracking-android/pull/97#discussion_r544195290 One solution could be to add Resolution to the Subscriber which should make some fields unnecessary.

Issue is synchronized with this Jira Task https//ably.atlassian.net/browse/ATS-151 by Unito.

Comments: Im going to close this issue as I think well resolve this under https//github.com/ably/ably-asset-tracking-android/pull/159.

Pull Request Title : Default resolution policy implementation.

Pull Request Description: Ive almost implemented the default resolution policy. Its missing tests (theyll be added in another PR) and temporal threshold (itll be added in upcoming commits) so its currently in the draft state. Probably we should define places in which the resolution should be refreshed because currently it happens only when the threshold is reached. Im implementing it in a hurry due to our deadline so please be extra cautious when checking this PR.

Commit message: Change TrackableResolutionRequest to a data class, Add method for getting correct resolution from DefaultResolutionSet, Assign levels to Accuracy for comparing them, Add battery data provider, Add helper functions for distance and time difference calculations, Add default resolution policy implementation, Make resolution policy factory a required builder param, Add resolution policy, hooks and methods to the publisher, Add refreshing resolution in Publisher, Use resolution policy hooks trackables methods.

- **Summary by Model 1:**

The issue titled "Reduce amount of fields in DefaultPublisher" aimed at simplifying the `DefaultPublisher` class in an Android asset tracking application by minimizing its fields, particularly maps and sets. The suggestion included adding a Resolution to the Subscriber to potentially make some fields redundant. This issue was tracked in conjunction with Jira Task ATS-151. However, it was decided to address this issue within the scope of another pull request, PR #159 titled "Default resolution policy implementation," leading to the closure of the original issue. The pull request in question involved implementing a default resolution policy, although it was noted to be in a draft state lacking tests and a temporal threshold. Key changes in the pull request included modifying `TrackableResolutionRequest` to a data class, adding a method for obtaining the correct resolution from `DefaultResolutionSet`, implementing levels to accuracy, adding a battery data provider, and integrating various helper functions for distance and time calculations. Additionally, the pull request emphasized the necessity of refreshing resolutions in the Publisher and integrating resolution policy hooks and methods. This comprehensive approach in the pull request suggests a strategic refactoring aimed at streamlining and enhancing the functionality of the `DefaultPublisher`, addressing the concerns raised in the original issue. Despite the closure of the initial issue, the summary does not explicitly confirm the resolution of the underlying concern about reducing fields in `DefaultPublisher`, implying that the matter may still be under development or pending final implementation.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**

- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**

- **Summary by Model 2:**

The issue titled "Reduce amount of fields in DefaultPublisher" discussed the possibility of reducing the number of fields, particularly maps and sets, in the `DefaultPublisher`. It was suggested that adding `Resolution` to the `Subscriber` might eliminate the need for some of these fields. The issue was marked to be resolved under a different pull request (#159). The related pull request titled "Default resolution policy implementation" aimed to address the issue by implementing the default resolution policy. However, the pull request was marked as a draft and lacked tests and a temporal threshold. The issue remains unresolved, as the changes proposed in the pull request are still in progress, and the issue itself was closed in favor of addressing it within the pull request #159.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
    -

- **Summary by Model 3:**

The issue titled "Reduce amount of fields in DefaultPublisher" is related to the Ably Asset Tracking Android library and focuses on reducing the number of fields in the DefaultPublisher class. The issue description mentions that some fields, particularly maps and sets, might be redundant and could be eliminated. The issue is linked to a Jira task (ATS-151) and has been closed in favor of resolving the issue under a separate pull request (#159). The pull request implements a default resolution policy and adds several features, including a data class for TrackableResolutionRequest, a method for obtaining the correct resolution from DefaultResolutionSet, and helper functions for calculating distances and time differences. The commit message lists the changes made in detail. Overall, the issue and pull request aim to improve the efficiency and functionality of the DefaultPublisher class in the Ably Asset Tracking Android library.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**

    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
- **Summary by Model 4:**

The issue addressed is about reducing the number of fields in the DefaultPublisher, particularly maps and sets. One solution proposed is to add Resolution to the Subscriber, which would make some fields unnecessary. The issue is synchronized with a Jira Task and is planned to be resolved under a separate pull request (https://github.com/ably/ably-asset-tracking-android/pull/159). The pull request for the default resolution policy implementation is mentioned, which includes various changes such as converting TrackableResolutionRequest to a data class, adding a method for getting the correct resolution from DefaultResolutionSet, and refreshing resolution in the publisher. The pull request is still in draft state, missing tests and a temporal threshold. The comment suggests that the team plans to define places where the resolution should be refreshed. The issue remains open, pending further development and testing.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**

Issue Thread 6 (Original Issue)

Issue Title: Dependency issues in Mac.

Issue Description: Probably the installer.py needs a reconstruct for Mac.

Comments: abhi-r3v0 This is what Im getting when I run python installer.py

```
[+] Installing necessary tools Password: sudo: apt-get: command not found

[+] Installation of dependencies complete

[+] Installing ARM dependencies sudo: apt-get: command not found sudo: apt-get: command not found sudo: apt-get: command not found

[+] Installation of ARM tools complete

[+] Installing Android debug tools sudo: apt-get: command not found
```

[+] Installation of Android tools complete

, It should be brew install instead of apt-get

from sys import platform if platform == linux or platform == linux2: linux elif platform == darwin: OS X elif platform == win32: Windows...

This might help differentiate the OS. abhi-r3v0 Please assign this issue to me. , eta on fix, 1 Day Max. , the1220 See 8 . Can you test it locally ?

Pull Request Title : Revamping Installer.py for Mac.

Pull Request Description: Installer.py fix for Mac

Revamped installer.py for Mac Removed sudo mode for Mac as homebrew does not support sudo mode installation. Mac Alternative for Android debug tools: android-tools-adb ARM dependencies: arm-gcc-bin binutils ncurses Fixes 4 TO-DO Installation candidate for win32.

Commit message: initialising the fix, Initial Code, Installation fo MAC, Fixed intendation, Removing sudo in Mac installation as sudo mode is not supported in homebrew, Re, Adding android-adb-tools for mac, Adding binutils for mac, Adding ncurses for mac, Adding arm-gcc-bin for Mac

- **Summary by Model 1:**

The issue titled "Dependency issues in Mac" addresses a problem encountered during the installation process of a software tool on macOS, as indicated by the error messages involving `apt-get` commands in the issue description. The primary concern was that the `installer.py` script, designed for installing dependencies and tools, was not compatible with macOS, which uses `brew install` instead of `apt-get` for package management. The issue discussion involved a user, abhi-r3v0, who experienced these errors and suggested modifying the installer script to differentiate between operating systems, offering to take on the task. The proposed timeline for a fix was one day. A pull request titled "Revamping Installer.py for Mac" was created to address this issue. The changes in this pull request included revamping `installer.py` specifically for Mac compatibility. Notably, the script was modified to remove the use of `sudo` mode, aligning with Homebrew's installation practices that do not support sudo mode. Additionally, the pull request introduced Mac alternatives for Android debug tools (`android-tools-adb`), ARM dependencies (`arm-gcc-bin`, `binutils`, `ncurses`), and prepared groundwork for future Windows support. The commit messages documented the progression of this fix, including the initial setup, code changes, and specific additions for Mac support. The issue appears to have been effectively addressed with these modifications, transitioning from a script suited for Linux environments to one also compatible with macOS. The addition of appropriate tools and dependencies for Mac indicates a successful resolution of the issue, although the final testing and verification of these changes may still be pending based on the last comment about local testing.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**

- **Summary by Model 2:**

The issue titled "Dependency issues in Mac" highlighted problems with the `installer.py` script on Mac. When running the script, it attempted to use `apt-get`, which is not available on Mac. A comment suggested using `brew install` instead. The issue was assigned to a user, and they estimated fixing it within one day. The related pull request titled "Revamping Installer.py for Mac" aimed to address the issue by revamping the installer script for Mac, removing `sudo` mode for Mac installations, and providing alternative dependencies for Mac. The pull request included various commits to fix the issue, such as adding necessary packages for Mac. The issue appears to have been resolved with the implementation of the pull request, which addressed the dependency issues on Mac by adapting the installer script accordingly.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**
  -

- **Summary by Model 3:**

The issue "Dependency Issues in Mac" refers to problems encountered while installing necessary tools using the Python script "installer.py." When running the script, the error "sudo: apt-get: command not found" appeared repeatedly. The root cause of the issue was determined to be the use of "apt-get" instead of "brew install" for MacOS. The issue was assigned to "abhi-r3v0" and had a target completion date of one day max. A pull request titled "Revamping Installer.py for Mac" was submitted, which addressed the issue by removing the use of "sudo" in the Mac installation process since Homebrew doesn't support sudo mode installation. The pull request included updates to the script to accommodate MacOS and Linux environments, as well as alternative tools for Android debugging and ARM dependencies. The revamped script was successfully merged into the repository.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**

    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
- **Summary by Model 4:**

The issue titled "Dependency issues in Mac" describes a problem with the installer.py script, which is failing to install necessary tools and dependencies for Mac. The comments suggest that the issue is likely caused by the use of `apt-get` commands, which are not compatible with Mac. The suggested solution is to replace `apt-get` with `brew install` commands. The pull request titled "Revamping Installer.py for Mac" implements this solution by revamping the installer.py script to remove sudo mode for Mac and adding alternative packages for Android debug tools, ARM dependencies, and other necessary libraries. The commit message lists the changes made in the patch, including the addition of `android-tools-adb`, `arm-gcc-bin`, `binutils`, `ncurses`, and `arm-gcc-bin` for Mac. The issue appears to be resolved through the pull request, which has been merged.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**

Issue 7:

Issue Title: Crash when starting playback from a playlist with only a single item.

Issue Description: The MediaRouterPlayService crashes when the playlist contains only a single item and the playback is started:

```
FATAL EXCEPTION: main Process: com.github.nutomic.controldlna, PID: 24026
java.lang.ArrayIndexOutOfBoundsException: length=10; index=-1 at java.util.ArrayList.get(ArrayList.java:413) at
com.github.nutomic.controldlna.mediarouter.MediaRouterPlayService.resume(MediaRouterPlayService.java:338) at
com.github.nutomic.controldlna.gui.RouteFragment.onClick(RouteFragment.java:486) at
android.view.View.performClick(View.java:5612) at android.view.View$PerformClick.run(View.java:22288) at
android.os.Handler.handleCallback(Handler.java:751) at android.os.Handler.dispatchMessage(Handler.java:95) at
android.os.Looper.loop(Looper.java:154) at android.app.ActivityThread.main(ActivityThread.java:6123) at
java.lang.reflect.Method.invoke(Native Method) at
com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:867) at
com.android.internal.os.ZygoteInit.main(ZygoteInit.java:757)
```

Comments: The crash could happen in any scenario where tracks were added to a previously unused playlist and then a route was selected and the resume button was pressed.

Pull Request Title : Fix crash when starting playback from a playlist with only a single item, fixes 13.

Pull Request Description: Fix crash when starting playback from a playlist with only a single item, fixes 13 Replaces 14 (forgot cherry pick branch there).

Commit message: Fix crash when starting playback from a playlist with only a single item, fixes 13, Improved fix for crash when starting playback from a playlist with only a single item, fixes 13

- **Summary by Model 1:**

The issue titled "Crash when starting playback from a playlist with only a single item" concerned a bug in the MediaRouterPlayService of the `com.github.nutomic.controldlna` application. The bug caused the service to crash when attempting to resume playback from a playlist containing just one item, as detailed in the issue description. This crash was triggered by an `ArrayIndexOutOfBoundsException`, indicating an attempt to access an invalid index in an array or list. This problem was not limited to a specific scenario but could occur anytime tracks were added to a previously unused playlist, followed by selecting a route and pressing the resume button. In response to this issue, a pull request titled "Fix crash when starting playback from a playlist with only a single item, fixes 13" was submitted. This pull request aimed to resolve the crash by addressing the root cause of the `ArrayIndexOutOfBoundsException`. The pull request description indicated that it was a replacement for a previous attempt (replaces 14), suggesting a refinement of the solution. The commit messages further confirmed the focus on fixing the crash by improving the handling of scenarios where the playlist contained only a single item. The pull request appears to address the crash issue effectively by modifying the playback start logic to handle single-item playlists correctly. The specific details of the fix were not provided in the summary, but the repeated emphasis on resolving this particular crash in both the pull request title and commit messages suggests that the issue was successfully resolved, pending testing and integration of the changes into the main codebase.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**

- **Summary by Model 2:**

The issue titled "Crash when starting playback from a playlist with only a single item" reported a crash in the `MediaRouterPlayService` when attempting to start playback from a playlist containing only one item. The crash was triggered by an `ArrayIndexOutOfBoundsException`. The problem was confirmed to occur when tracks were added to an empty playlist, a route was selected, and then the resume button was pressed. A pull request titled "Fix crash when starting playback from a playlist with only a single item" was submitted to address the issue, and it aimed to resolve the crash scenario. The issue appears to have been resolved with the implementation of this pull request, which fixed the crash by ensuring that playback from a playlist with a single item is handled correctly, thus preventing the `ArrayIndexOutOfBoundsException` and improving the overall stability of the application.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**

- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
    -

- **Summary by Model 3:**

The issue "Crash when starting playback from a playlist with only a single item" occurs when attempting to start playback from a playlist containing only one item. The service crashes with an ArrayIndexOutOfBoundsException caused by trying to access an element at index -1 in an ArrayList. The exception occurs in the `resume` function of the `MediaRouterPlayService` class, specifically in line 338. The issue has been fixed through a pull request that replaces the previous attempt at fixing the problem. The updated code checks if the list of items is empty before accessing its contents, avoiding the out-of-bounds exception. The fix addresses issue #13 and improves upon a previous attempt at solving the problem.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
- **Summary by Model 4:**

The issue titled "Crash when starting playback from a playlist with only a single item" describes a problem with the MediaRouterPlayService, which crashes when attempting to play back a playlist containing only one item. The error occurs at line 338 of the MediaRouterPlayService.java file and is caused by an ArrayIndexOutOfBoundsException. The issue can occur in any scenario where tracks are added to a previously unused playlist and then a route is selected and the resume button is pressed. A pull request labeled "Fix crash when starting playback from a playlist with only a single item" has been submitted to address this issue. The pull request includes a revised version of the MediaRouterPlayService.java file that resolves the crash. The commit message states that the fix improves the previous solution and correctly addresses the issue. Therefore, the issue appears to be resolved through the pull request.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**

    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**

Issue 8:

Issue Title: Inclusion in F-Droid.

Issue Description: Hi,

Since this app is Apache v2 licensed, are you considering the fact to add it to F-Droid? There is a strong community behind, that do not use Google Play nor anything else, except F-Droid.

You can find some documentation here.

Otherwise, if you dont want to work on the inclusion of this app by yourself, would it be ok for you if someone else do it for you?

Comments: Ive looked at using f-droid before but never managed to get it working properly. I will give it another look this weekend., Great The documentation is very helpful, if you need it. Otherwise ping me, maybe I could help you, even if Im not a master in Android development nor anything else, JAR files like this one here are problematic, they should be built from source or taken from a trusted maven repo if they are open-souce. Otherwise, if they are proprietary, they should be removed.

EDIT : I opened a Request For Packaging here., Ive gone through the docs and got stuck at running makebuildserver as the vagrant module wasnt found and I havent been able to install it properly.

Ive responded to the RFP, but if you wana drop me an email 7lpdwcaw@gmail.com with any details or things that need to be done then im happy to provide details.

Is a shame that they self sign the APK meaning people cant download from here and upgrade via f-droid, but thats fine, as long as its always built from source each time.

I have also imported the project into gitlab here https//gitlab.com/7lpdwcaw/GrowTracker-Android.

Pull Request Title : Update README.md.

Pull Request Description: Hi, This small PR adds a link to get your app on F-Droid close 60.

Commit message: Update README.md

- **Summary by Model 1:**

The issue titled "Inclusion in F-Droid" involves a user's inquiry about adding an Apache v2 licensed app to F-Droid, a platform preferred by a community that avoids Google Play. The user offered to assist if the app's developer was not willing to handle the inclusion process. The developer initially struggled with F-Droid's integration, citing difficulties in getting it to work and issues with running `makebuildserver` due to a missing vagrant module. A notable concern raised was the handling of JAR files, emphasizing the need for them to be either built from source or sourced from a trusted open-source maven repository. Proprietary JAR files were advised to be removed. Subsequently, the developer opened a Request For Packaging (RFP) for the app and imported the project into GitLab, indicating progress towards F-Droid inclusion. However, challenges like self-signing the APK, which prevents users from downloading and upgrading the app via F-Droid, were acknowledged. The developer was open to external help, as indicated by their willingness to provide details via email for necessary actions. The pull request titled "Update README.md" was a small yet significant step towards this integration, wherein the developer added a link to the app on F-Droid, hinting at successful progress in the

F-Droid inclusion process. This pull request, with the commit message "Update README.md," suggests that while the issue had complexities, it was moving towards a resolution, albeit with some unresolved technical challenges.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**

- **Summary by Model 2:**

The issue titled "Inclusion in F-Droid" discussed the possibility of adding the app to the F-Droid repository due to its Apache v2 licensing and the desire to make it available to users who do not use Google Play but rely on F-Droid. The discussion involved considering community contributions for inclusion. Some comments indicated challenges with F-Droid integration, and an external Request For Packaging (RFP) was opened. The issue remained unresolved as there were technical hurdles related to F-Droid integration. A related pull request titled "Update README.md" aimed to add a link to the app's inclusion process on F-Droid, facilitating the potential inclusion in the future. However, the core issue of adding the app to F-Droid was still pending due to technical complexities.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
    -

- **Summary by Model 3:**

The issue thread pertains to the inclusion of the Grow Tracker app in F-Droid, a free and open-source software distribution platform for Android devices. The app is currently available on Google Play Store, but the author of the issue wants to explore the possibility of adding it to F-Droid as well. They mention that there is a strong community of users who prefer to use F-Droid and would benefit from having the app available on the platform. The author suggests that the app could either be packaged manually or that someone else could volunteer to package it. The author also links to relevant documentation and offers to assist with the packaging process if needed. An update to the README file includes a link to download the app directly from F-Droid. While the issue is still open, it appears that the author has taken steps towards exploring the feasibility of adding the app to F-Droid.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
- **Summary by Model 4:**

The issue titled "Inclusion in F-Droid" discusses the possibility of adding the GrowTracker app to F-Droid, an open-source app store for Android devices. The original poster asks the maintainers if they plan to include the app in F-Droid, citing the app's Apache v2 license and the presence of a strong community that uses F-Droid exclusively. One of the maintainers expresses interest in exploring F-Droid again, while another offers assistance with the process. However, they also mention that JAR files like the ones used in the app are problematic and should be built from source or taken from a trusted Maven repository. The conversation continues with suggestions on how to proceed, including opening a Request For Packaging and importing the project into GitLab. Ultimately, a pull request is submitted to update the README.md file with a link to get the app on F-Droid. While progress has been made towards resolving the issue, it remains open until the app is successfully included in F-Droid.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**

Issue 9:

Issue Title: TCP transport support.

Issue Description: Hi! Thanks for great application!

Some time ago my sip provider changed settings, so that telephone stopped working correctly. Call is performed but there is absolute silence.

Same was on my android phone. They said to change Transport type to TCP it was UDP in advanced settings this made it working again.

Do you plan to add this option to telephone?

Comments: I would like to, but because of the lack of time, dont expect it to come in the near future. , Any news? , Ive just discovered that it is already possible to make use of the TCP transport. It is already enabled in the user agent which is not new but isnt chosen for the outgoing requests by default. To activate it for the outgoing requests, the proxy server should be configured with the addition transport=tcp in the advanced account settings. For example:

[x] Connect using proxy Server: proxy.example.com;transport=tcp

Pull Request Title : Explicit transport.

Pull Request Description: Set SIP transport explicitly closes 569. TCP transport support closes 107. Use radio buttons instead of checkbox to switch between IPv4 and IPv6 closes 571. Fix an issue where account configuration wasnt completely applied on account creation closes 570. Remove TransportPublicHost setting closes 576.

Commit message: Remove TransportPublicHost setting that is not expected to be used, Set transport explicitly, Read transport from user defaults, Remove duplication, Use the same account controller creation when new account is added

Fixes an issue when an account controller and its account are configured incompletely after a new account is added to the app., Add transport radio buttons to preferences, Radio buttons instead of checkbox to control IP version, Shorter outlet names, Separate user defaults key values for IP version, Add user defaults migration for the IP version

- **Summary by Model 1:**

The issue titled "TCP Transport Support" highlights a user's need for TCP transport in a SIP application after their provider changed settings, causing call issues. The user discovered that changing the transport type from UDP to TCP in advanced settings on another device resolved the problem and inquired about implementing this in the application in question. In response, a developer expressed interest in adding this feature but noted time constraints for its implementation. Further communication revealed that TCP transport was already possible through a workaround: configuring the proxy server with `transport=tcp` in the advanced account settings. Addressing this issue, a pull request titled "Explicit Transport" was created. This pull request, aimed at setting SIP transport explicitly, involved several changes: using radio buttons instead of a checkbox for IPv4 and IPv6 switching, fixing an incomplete account configuration upon creation, and removing the `TransportPublicHost` setting. The commit message detailed these changes, emphasizing the removal of `TransportPublicHost`, setting transport explicitly, and improvements in user defaults and account controller configuration. The pull request marked a substantial step towards resolving the issue by simplifying transport selection and enhancing account setup, indicating that while the core issue of TCP transport support was addressed, there were additional refinements and bug fixes included in the implementation.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**

- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**

- **Summary by Model 2:**

The issue titled "TCP transport support" raised a concern about a SIP provider's settings change, resulting in a loss of audio during calls. The user requested the addition of an option to change the transport type to TCP in the application's settings. Initially, the response mentioned a desire to implement this feature but noted that it might not happen soon due to time constraints. Later, it was discovered that TCP transport support was already available but not enabled by default. The user could activate it by configuring the proxy server with the addition of "transport=tcp" in advanced account settings. As a result, the issue was resolved, and there was no need for further changes. A related pull request titled "Explicit transport" was submitted to explicitly set the SIP transport and included various fixes and improvements.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
    -

- **Summary by Model 3:**

The issue revolves around supporting the Transmission Control Protocol (TCP) transport layer in the Telephone application. Some time ago, the OP's SIP provider modified their settings, causing calls to fail silently. However, changing the transport type to TCP in the advanced settings restored connectivity. The developer expressed interest in incorporating this feature but lacks sufficient time to address it promptly. Another commentator noted that TCP transport support is now available in the latest release, requiring no additional setup beyond enabling it in the user agent. Nonetheless, the developer created a pull request to enable explicit transport selection, remove obsolete settings, and streamline the user interface. The patch resolves five distinct issues and introduces various improvements, such as replacing checkboxes with radio buttons for controlling IP versions.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**

- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
- **Summary by Model 4:**

The issue titled "TCP transport support" discusses the need for the ability to select TCP as the transport protocol for outgoing requests in the SIP client. The original poster mentions that their SIP provider has changed their settings, causing calls to fail, and that changing the transport protocol to TCP fixed the issue on their Android device. Another commenter confirms that enabling TCP transport is already possible but requires manual configuration of the proxy server. A pull request is submitted to set the SIP transport explicitly, use radio buttons instead of checkboxes to switch between IPv4 and IPv6, fix an issue with incomplete account configuration, and remove unnecessary settings. The commit message indicates that the changes resolve several related issues and improve the app's functionality. The issue appears to be resolved through the pull request, ensuring that users can choose TCP transport for their outgoing requests.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
    - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
    - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**

Issue 10:

Issue Title: Problem with saving the state of recyclerView Issue Description: Hey guys

I have a problem with saving the state of Recycler View using Greedo Layout Manager. I get this exception when calling:

```
recyclerView.scrollToPosition(index);
```

in onResume()

```
03-06 12:35:55.287 20863-20863/com.artland E/AndroidRuntime: FATAL EXCEPTION: main Process: com.artland, PID:
20863 java.lang.RuntimeException: Invalid content width. Did you forget to set it? at
com.fivehundredpx.greedolayout.GreedoLayoutSizeCalculator.computeChildSizesUpToPosition(GreedoLayoutSizeCalcul
ator.java:101) at
com.fivehundredpx.greedolayout.GreedoLayoutSizeCalculator.getRowForChildPosition(GreedoLayoutSizeCalculator.java:
79) at com.fivehundredpx.greedolayout.GreedoLayoutManager.rowForChildPosition(GreedoLayoutManager.java:278) at
com.fivehundredpx.greedolayout.GreedoLayoutManager.scrollToPosition(GreedoLayoutManager.java:306)
```

Any idea on how to fix it? Weird thing is, that both setContentWidth() and setMaxRowHeight() are public void methods, but I can only use setMaxRowHeight() when configuring the GreedoLayoutManager. setContentWidth() is not even visible as a usable method.

Would highly appreciate any help! mck182 Comments: Could somebody from 500px, who was creating the GreedoLayout for android, please comment on this issue? It is quite urgent for me and would highly appreciate any help with this problem. JVillella mck182 , Hey there,

it took a while to get to this but weve now fixed the crash, feel free to reopen this ticket if it didnt resolve this issue for you.

Martin, 500px, Thanks ! I hope it is going to work now.

I must ask though - is the library version still 1.0.0 ?? , Hey mck182 !

Ive checked out the fixes you guys made to solving the problem with saving the state of GreedoManager. It does work in fact, however there is one thing that bothers me a lot, I wonder if it only happens for me or is a general issue:

1. open a fragment hosting a list of images with greedoManger
2. click on any to take you to another view displaying the full image
3. go back and see the position of the layout as you left it

Great, everything works as it was supposed to... however, the moment you scroll up (a fast scroll, like you would do to quickly get back to the top), the content is lost. Suddenly all the images disappear, although the presence of recyclerView is still there (you can see on the screen that it responds to scroll gestures). Only going to another fragment/activity and back resets the list of images.

I also cant help but notice, that the app stutters a lot when it enters the process of restoring the state of GreedoManager. There is a huge hiccup (a slight screen freeze + FPS drop)

Do you reckon its something on my end of implementation (or maybe the phone) ? Or is it a general issue?, New version of the library is up - 1.1.0, enjoy!

As for your issues, Ive seen the disappearance happen couple times, but Ive never really had a chance to investigate that.

Can you reliably reproduce that problem? Would you be able to produce a minimal reproducible code? Ie. a tiniest possible app that can reproduce this? I could then investigate.

As for slowness, we havent really seen that so cant comment on that.

Cheers, Martin from 500px Pull Request Title : Defer scrollToPosition() and fix IndexOutOfBounds crash Pull Request Description: This PR fixes two crashes related to scrollToPosition() calls. Defer scrollToPosition() after layouting is fully finished If scrollToPosition() is called before the postLayout pass, the LayoutManager has no knowledge about its own sizes yet and therefore the scroll operation cannot be completed and the safety checks in GreedoLayoutSizeCalculator will crash on that. So this defers the scrollToPosition() call to after the layout is finished. Should fix 31 Fix index-out-of-bounds crash With certain positions (usually those on the start of the row) this can crash with java.lang.IndexOutOfBoundsException. The problem is that were checking the position with mRowForChildPosition.size() and then computing all the child sizes up to the position (ie. not including the position). This can lead to crashes as can be seen in 20 Commit message: Defer scrollToPosition() after layouting is fully finished

If scrollToPosition() is called before the postLayout pass, the LayoutManager has no knowledge about its own sizes yet and therefore the scroll operation cannot be completed and the safety checks in GreedoLayoutSizeCalculator will crash on that.

So this defers the scrollToPosition() call to after the layout is finished., Fix index-out-of-bounds crash

With certain positions (usually those on the start of the row) this can crash with java.lang.IndexOutOfBoundsException. The problem is that were checking the position with mRowForChildPosition.size() and then computing all the child sizes *up to* the position (ie. not including the position). This can lead to crashes as can be seen in https://github.com/500px/greedo-layout-for-android/issues/20, Move the index increase inside computeChildSizesUpToPosition()

- **Summary by Model 1:**

The issue titled "Problem with saving the state of recyclerView" revolves around a user experiencing a crash when using the Greedo Layout Manager in a RecyclerView, specifically when calling `recyclerView.scrollToPosition(index)` in the `onResume()` method. The crash was caused by an "Invalid content width" exception, with the user noting an inability to access the `setContentWidth()` method, only `setMaxRowHeight()`. The urgency of the issue was heightened by its impact on the user's application functionality. Responding to the issue, a representative from 500px, the creators of GreedoLayout, acknowledged the problem and released a fix. However, the user later reported a new issue where images disappeared upon fast scrolling, alongside performance hiccups when restoring the state of GreedoManager. The user questioned whether this was a general issue or specific to their implementation or device. A new version of the library, 1.1.0, was released to address these issues. The pull request titled "Defer scrollToPosition() and fix IndexOutOfBounds crash" was submitted, aiming to resolve two specific crashes related to the `scrollToPosition()` calls. The solution involved deferring the `scrollToPosition()` call until after the layout was fully completed, addressing the crash caused by the LayoutManager's lack of size knowledge during the pre-layout pass. Additionally, a fix for an index-out-of-bounds crash was implemented, addressing issues with certain positions, particularly at the start of a row. The commit message detailed these changes, emphasizing the deferment of the `scrollToPosition()` call and the resolution of the index-out-of-bounds crash. This indicated that while the initial crash issue was resolved, subsequent updates were made to address new problems identified during the implementation process, demonstrating an ongoing effort to refine and stabilize the Greedo Layout Manager's functionality in various use cases.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**

- **Summary by Model 2:**

The issue titled "Problem with saving the state of recyclerView" reported a crash related to saving the state of a RecyclerView using the Greedo Layout Manager. The exception occurred when calling `recyclerView.scrollToPosition(index);` in the `onResume()` method. The user requested help on how to resolve this issue, mentioning difficulties in using the `setContentWidth()` method. Subsequently, a comment asked for assistance from someone at 500px, the creator of GreedoLayout. After some time, a response from 500px confirmed that they had fixed the crash and suggested reopening the ticket if the issue persisted. The user reported that the crash was fixed but mentioned another issue where, after scrolling up quickly, the content disappeared until navigating to another fragment or activity. A new library version, 1.1.0, was released, and a pull request titled "Defer scrollToPosition() and fix IndexOutOfBounds crash" was submitted to address crashes related to `scrollToPosition()`. The pull request proposed deferring `scrollToPosition()` after layouting and fixing an index-out-of-bounds crash. While the initial issue of the crash was resolved, the issue regarding disappearing content and performance concerns remained open for further investigation.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**

- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
    - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
    - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
    - **Rating scale: 1-10**
    -

- **Summary by Model 3:**

The issue thread discusses the possibility of reducing the number of fields in the DefaultPublisher, particularly maps and sets, by adding a Resolution to the Subscriber. This would make some fields unnecessary. The issue is linked to a Jira task and the comment suggests that it will be resolved under a different pull request (https://github.com/ably/ably-asset-tracking-android/pull/159). The pull request mentioned in the comment is titled "Default resolution policy implementation" and includes various changes such as making TrackableResolutionRequest a data class, adding a method for getting the correct resolution, assigning levels to Accuracy for comparison, adding a battery data provider, and more. However, the pull request is still in draft mode and lacks tests and a temporal threshold, so the author advises caution when reviewing it. Finally, the commit message lists the changes made in the pull request, including the addition of a resolution policy, hooks, and methods to the publisher, as well as the use of resolution policy hooks in trackables methods. The "Reduce amount of fields in DefaultPublisher" issue aimed to simplify theDefaultPublisher class by decreasing the number of fields, particularly maps and sets. A potential approach was proposed, involving adding a Resolution to the Subscriber, rendering some fields redundant. This idea was discussed in the context of the related Jira Task ATS-151. Later, the issue was closed since it was decided to address it through a different Pull Request, #159, which focused on implementing a default resolution policy. That Pull Request introduced various changes, including converting TrackableResolutionRequest to a data class, adding methods for obtaining the proper resolution and calculating distances and time differences, and incorporating battery data providers. Additionally, the commit included factories for creating resolution policies, linking them to the publisher, and periodically updating the resolution. Although the Pull Request was marked as "draft" due to outstanding tasks, such as adding tests and defining refresh rate locations, it signified progress toward resolving the original issue. Ultimately, the fate of the issue depends on the evaluation and approval of the associated Pull Request.

Issue Title: Reduce amount of fields in DefaultPublisher.

Issue Description: We could think about reducing the number of fields (particularly maps and sets) that are currently in the DefaultPublisher https//github.com/ably/ably-asset-tracking-android/pull/97#discussion_r544195290 One solution could be to add Resolution to the Subscriber which should make some fields unnecessary.

Issue is synchronized with this Jira Task https//ably.atlassian.net/browse/ATS-151 by Unito.

Comments: Im going to close this issue as I think well resolve this under https//github.com/ably/ably-asset-tracking-android/pull/159.

Pull Request Title : Default resolution policy implementation.

Pull Request Description: Ive almost implemented the default resolution policy. Its missing tests (theyll be added in another PR) and temporal threshold (itll be added in upcoming commits) so its currently in the draft state. Probably we should define places in which the resolution should be refreshed because currently it happens only when the threshold is reached. Im implementing it in a hurry due to our deadline so please be extra cautious when checking this PR.

Commit message: Change TrackableResolutionRequest to a data class, Add method for getting correct resolution from DefaultResolutionSet, Assign levels to Accuracy for comparing them, Add battery data provider, Add helper functions for distance and time difference calculations, Add default resolution policy implementation, Make resolution policy factory a required builder param, Add resolution policy, hooks and methods to the publisher, Add refreshing resolution in Publisher, Use resolution policy hooks trackables methods.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
    - **Rating scale: 1-10**

- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**

- **Summary by Model 4:**

The issue titled "Dependency issues in Mac" discusses problems encountered while running the `installer.py` script on a Mac system. Specifically, the script fails to install dependencies due to the absence of the `apt-get` command, which is not available on Macs. Instead, the suggested solution involves replacing `apt-get` with `brew install`. Assigned to abhi-r3v0, the issue has been addressed in a pull request that revamps `installer.py` for Mac, removing the use of sudo mode (unsupported by Homebrew) and adding alternative packages for Android debugging tools and ARM dependencies. The updated code includes changes for installing on Win32 systems. While some formatting issues remain, the pull request has been labeled as ready for review after resolving four todo items. Overall, the issue is currently in progress, awaiting further development before its eventual resolution.

For each summary, please rate the following:

- **Clarity: Is the summary easy to understand?**
  - **Rating scale: 1-10**
- **Conciseness: Is the summary to the point without unnecessary details?**
  - **Rating scale: 1-10**
- **Completeness: Does the summary cover all key points, including problem description, context, and possible implementation?**
  - **Rating scale: 1-10**
- **Technical Accuracy: Does the summary accurately reflect the technical issues and solutions?**
  - **Rating scale: 1-10**
- **Reproducibility: Can a developer understand how to reproduce the issue from the summary?**
  - **Rating scale: 1-10**
- **Actionability: Does the summary provide enough information to act upon?**
  - **Rating scale: 1-10**