

Tukxi Ride - Bug #4825

Rider side not receiving location updates during trips

05/09/2025 02:35 AM - Anupama P

| | | | |
|------------------------|--|------------------------|-------------------------|
| Status: | QA Passed | Start date: | 05/13/2025 |
| Priority: | Normal | Due date: | 05/13/2025 |
| Assignee: | Sreedevi K S | % Done: | 0% |
| Category: | | Estimated time: | 0.00 hour |
| Target version: | 1.1.1 | Spent time: | 0.00 hour |
| Milestone: | Sprint 19 Tukxi Ride 2024 May 01 to May 31 | Branch name: | sprint-19_sreedevi_4825 |

Description

Particularly the problem of the rider side not receiving location updates during trips. Upon detailed analysis, We found that the root cause lies in the socket library currently used in our app.

We are using the SocketCluster library (io.github.sac:SocketclusterClientJava), which is outdated and not officially suitable for real-time location updates. Below are some recurring issues associated with this library:

Missing messages without any errors or disconnect indications.

Improper subscription timing or loss of connection state.

Inconsistent acknowledgements when multiple clients are subscribed to the same channel.

Socket not reconnecting properly despite showing as connected.

Failures due to Android clients being in background or Doze mode (affected by battery optimization settings).

While reviewing our implementation, I also found several code-level issues that may be compounding the problem:

Multiple subscriptions to the same channel are made repeatedly with each incoming location update.

Concurrent asynchronous subscriptions to different channels can lead to failure in message listening without triggering socket errors.

Channels are being disconnected right after the trip is accepted, even when some channels are still in a subscribed state.

Reconnection may sometimes fail, and even after reconnection, subscription and onMessage handling do not always work correctly.

Message acknowledgment failures occur occasionally, preventing reliable communication.

History

#1 - 05/09/2025 04:51 AM - Sreedevi K S

- Status changed from New to In Progress

- Branch name set to sprint-19_sreedevi_4825

#2 - 05/09/2025 07:50 AM - Sreedevi K S

- Status changed from In Progress to Resolved

I have resolved the bug and implemented the following fixes:

1. Implemented reconnection logic if the client is not connected at the time of a location update.
2. Ensured that each channel is subscribed only once by restricting repeated subscriptions to the same channel.
3. Resolved the issue where the socket disconnects immediately after the cab request is accepted, even though the channels are in a subscribed state. Also handled disconnection scenarios specifically during trip location updates.

4. Addressed the issue where the socket becomes inactive or unresponsive when the device is in an inactive state. The socket is now reactivated automatically when a cab request is accepted.

Note: There is still a delay of around 10–50 seconds in receiving location updates. Locations are received in bulk after some idle time. This delay is handled internally within the SocketCluster code and is not related to our app implementation.

#3 - 05/12/2025 02:50 AM - Anupama P

- Status changed from Resolved to QA Ready

#4 - 05/13/2025 05:18 AM - Shalu T S

- Due date set to 05/13/2025

- Status changed from QA Ready to QA In Progress

- Start date changed from 05/09/2025 to 05/13/2025

#5 - 05/13/2025 05:32 AM - Shalu T S

- Status changed from QA In Progress to QA Passed

No Issue found