Sync Diagnostics Feature Overview

StopTB application | Feature overview | 19 October 2026

I have added Sync Diagnostics to make local saves, upload results, downloaded records and camp connectivity easier to understand. This document explains the implemented features with screenshots and shows how logs and summaries can be shared for investigation. A connected hub or completed worker does not by itself prove that every record synced successfully.

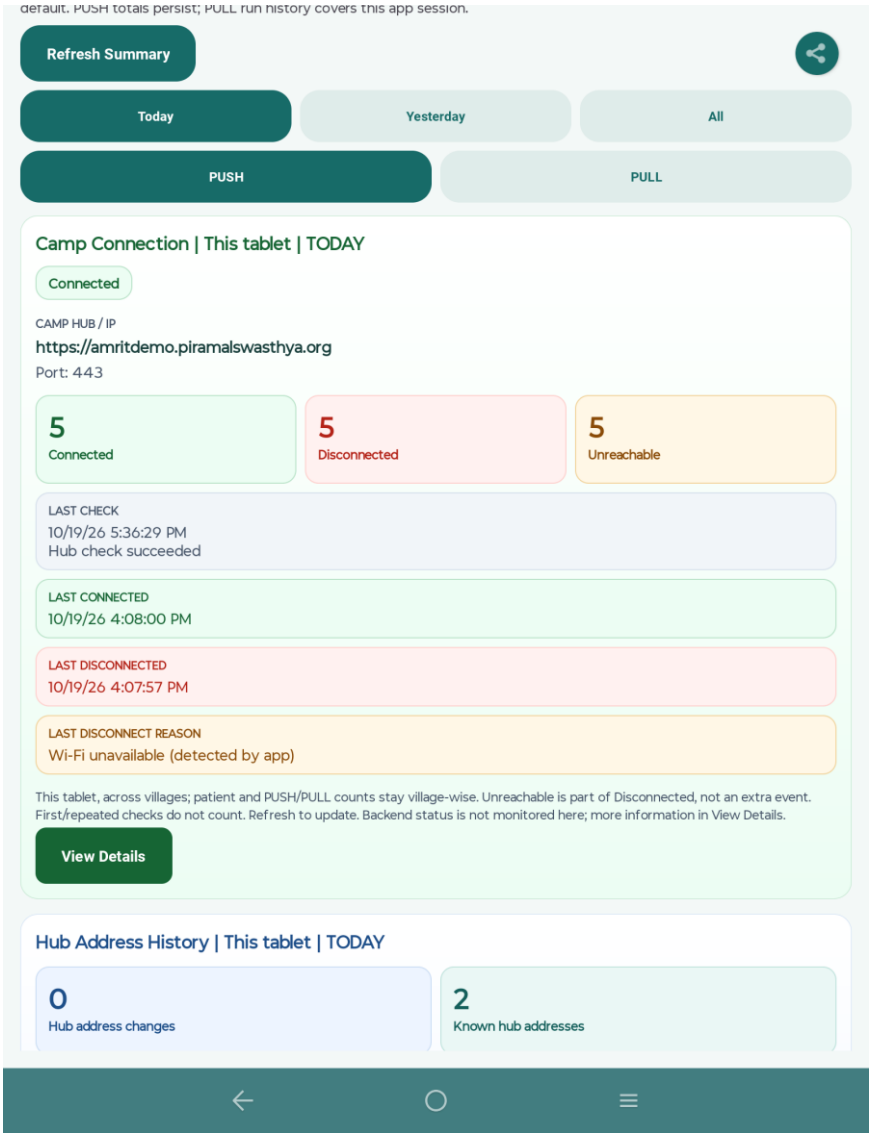


Figure 1 Dashboard tabs and Summary controls

How to read this screen

Open and choose a view

Open the navigation drawer and select Sync Dashboard. Use Status for existing record and worker progress, Logs for retained manual/direct activity, and Summary for counts and reasons.

View or section	Scope
Status	Existing worker and record status; do not equate it with daily local work.
PUSH and PULL	Selected account, service and village. Today is the default date filter.
Camp and hub history	This tablet across accounts and villages, for the selected dates.
Logs	Retained manual/direct events in this app process session; not the Summary date filter.

Refresh Summary updates the diagnostic snapshot only. Use the existing application refresh/sync action to actually upload or download data.

Logs and API request response details

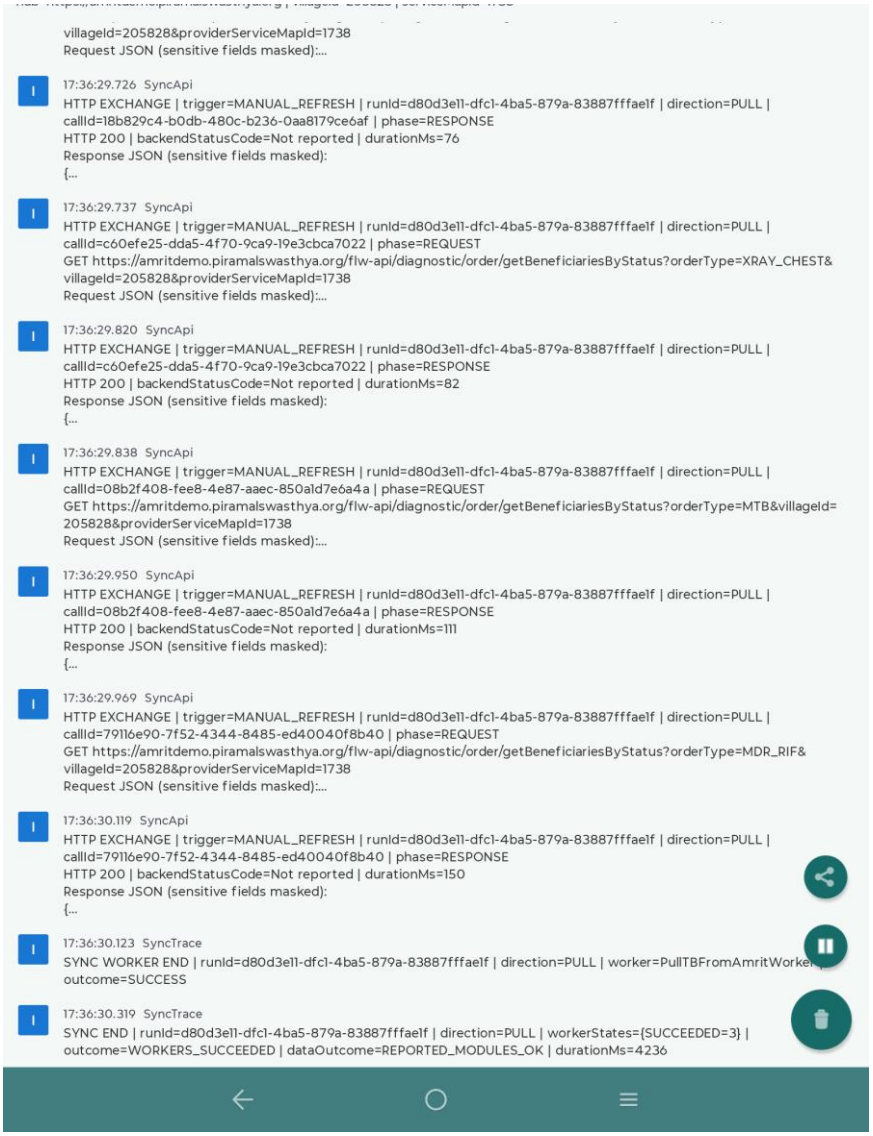


Figure 2 Manual refresh PULL request and response entries

How to read this screen

Read a matching exchange

- Check trigger and direction. MANUAL_REFRESH can include both PUSH and PULL calls; DIRECT_SAVE identifies an attributed direct action.
- Match REQUEST and RESPONSE using callId. runId groups related sync activity. Inspect the URL, method, HTTP result, recorded backend status and duration.
- Tap a row for its retained full entry. A preview ending with dots is not the full response; a missing backend status means Not reported, not success.
- A local save without a network request has no API response to show. LOCAL SAVE and successful upload are different events.

Controls and limits

Pause freezes the live view, not sync or capture. Clear empties session log buffers, not clinical data. The view shows up to 200 entries; sharing includes up to 500 retained manual/direct entries. Auto-sync/raw logs are excluded from this Logs export.

Recognized sensitive fields are masked. Body capture is limited to 32 KiB per body; larger or non-JSON bodies may be omitted. Review exported support data before sharing.

Camp connection counts and reasons

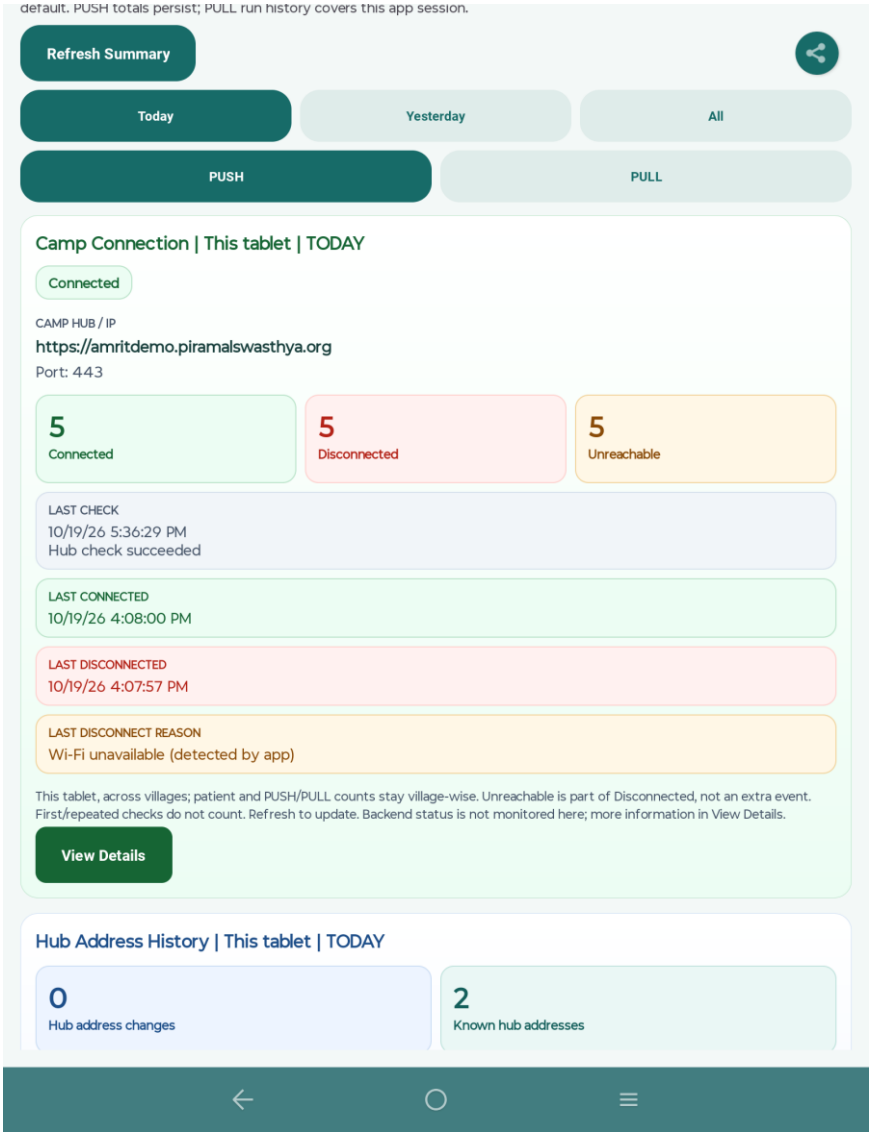


Figure 3 Tablet connection status with a separate Port label

How to read this screen

What this example means

The tablet is currently observed as Connected. The screenshot records 5 connected transitions and 5 disconnects. The 5 Unreachable events are part of those disconnects, not 5 additional outages.

- Last Check is the latest observed check. Last Connected and Last Disconnected are recorded transition times.
- The last shown reason is Wi-Fi unavailable. It does not establish that the backend server was down.
- First observations, app-restart baselines and repeated same-state checks do not increase counts. A hub-address change alone is not a disconnect.

After a connected baseline, one observed outage and its recovery are represented as one disconnect and one connected transition, rather than one event per repeated check.

Hub address history across villages

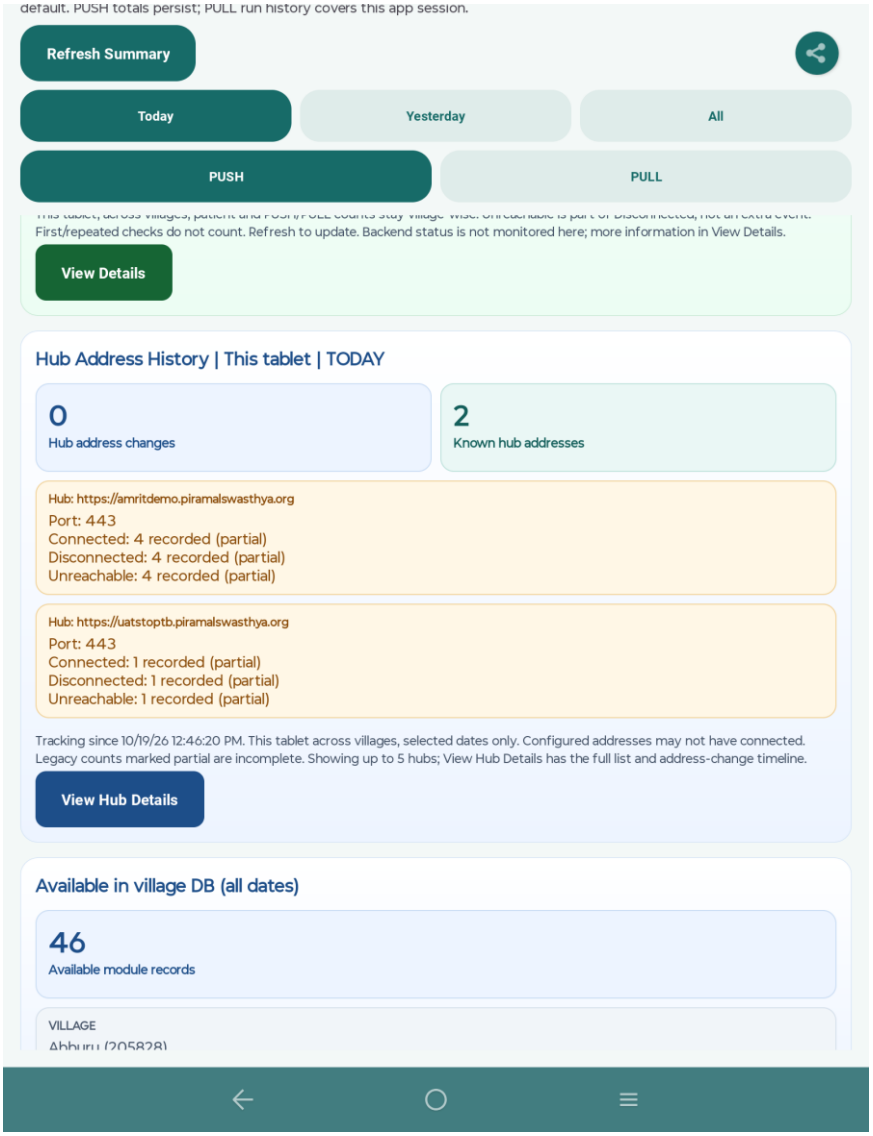


Figure 4 Two known endpoints and partial legacy per-hub counts

How to read this screen

Read endpoints separately from changes

Known hub addresses is 2, while Hub address changes is 0. This is valid when endpoints were observed or migrated but no address-change event was captured during tracking. The app does not invent older switches.

Label	Interpretation
Hub and Port	The endpoint hostname/IP and port are labelled separately. HTTPS commonly uses 443; a custom camp port such as 8080 remains visible.
Recorded partial	Legacy per-hub counts are incomplete. They are not a full historic audit.
View Hub Details	Open the retained endpoint list and old-to-new address-change timeline.
Tablet scope	Changing village must not hide this tablet connection history. Patient and module metrics remain village-wise.

Hub configuration changes are captured through the normal app flow, including pre-login configuration. The previous recorded endpoint remains in tablet history after moving to another hub.

PUSH local saves and village inventory

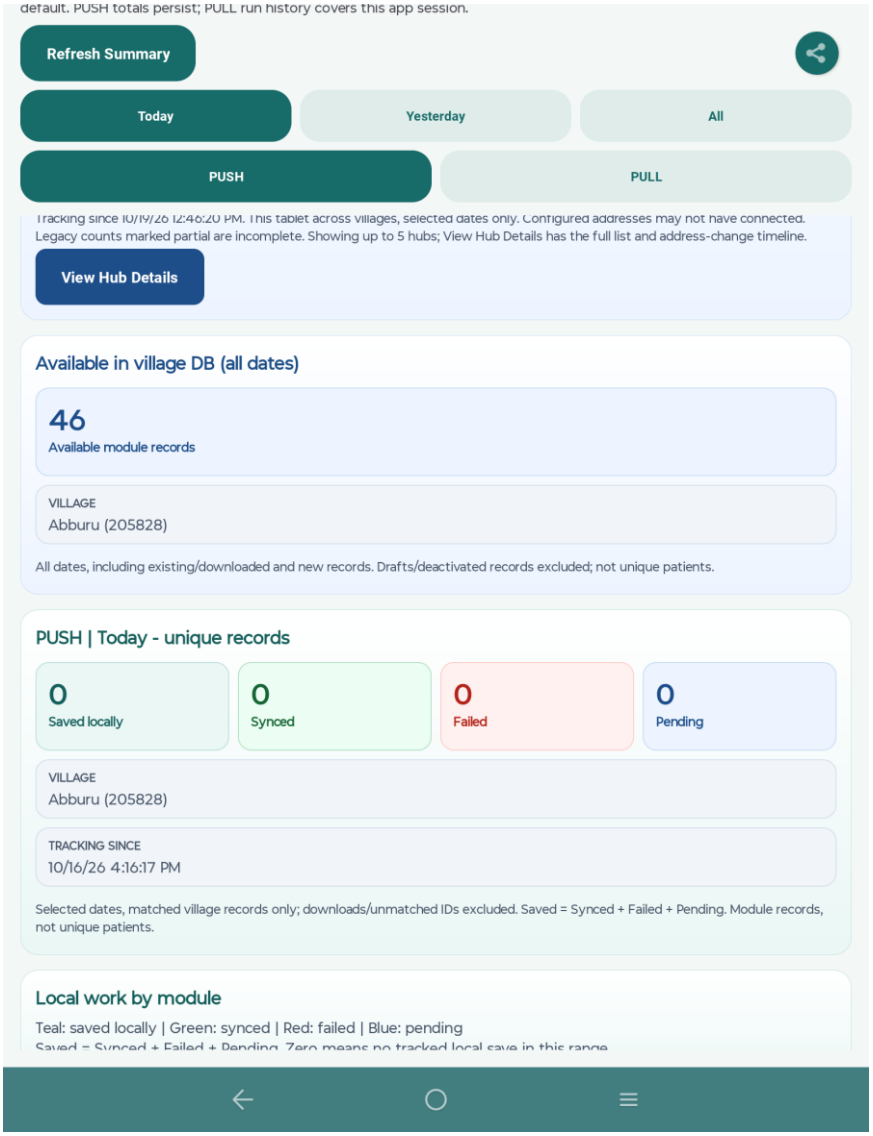


Figure 5 Existing village inventory is separate from today local activity

How to read this screen

Why 46 available can appear with 0 saved today

Available in village DB is the current all-date inventory, including existing and downloaded records. The example has 46 available module records, but no tracked local saves for Today. These values answer different questions.

PUSH metric	Meaning
Saved locally	Unique tracked module records saved locally in the selected date range; downloads excluded.
Synced	Those records with a qualifying successful upload after the latest local save.
Failed	Those records with a recorded latest applicable failure after the latest save.
Pending	Those records without a later qualifying success/failure. Pending is not automatically failed.

Saved locally = Synced + Failed + Pending. Counts are module records, not unique patients. Status is current for the selected local-save cohort, including later retries and edits.

Module graphs and direct failure access



Figure 6 Village inventory and current day graphs by module

How to read this screen

Check each affected module

The example shows 13 beneficiaries, 2 households and 9 TB screening records available in the village DB. The zero graphs mean no tracked local-save activity for the selected day, not an empty village.

- Teal is Saved locally, green is Synced, red is Failed and blue is Pending. Read the numeric label, not only the bar length.
- When recorded failures exist, open View Failures on that module. Available details include record ID, action, time, run and recorded reason.
- Switch village and refresh. Clinical inventory, local-save cohorts and PULL runs must belong to that village; camp history remains tablet-wide.

Example: for 3 new locally saved OPD records, if 2 upload and 1 has a recorded failure, the counts are Saved 3, Synced 2, Failed 1, Pending 0. After a successful retry they become Saved 3, Synced 3, Failed 0, Pending 0.

PULL saved records and blocking skips

Select PULL to inspect the latest matching account, service and village run within Today, Yesterday or All. These are run operation counts, not the PUSH local-save cohort and not a sum of every run that day.

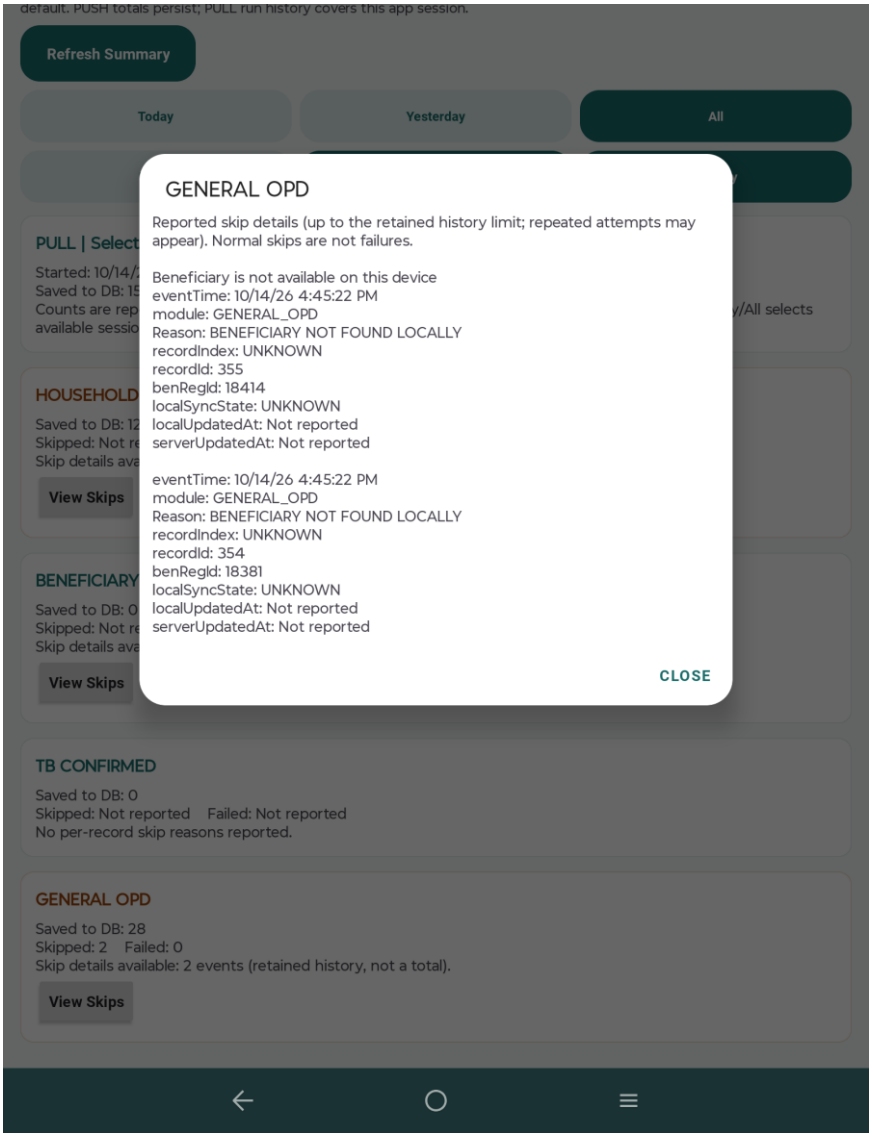


Figure 7 Earlier detail-dialog example showing a beneficiary missing locally

How to read this screen

This earlier dialog illustrates the recorded skip reason and identifiers. Its background uses the older layout; the current selector has PUSH and PULL only, without History.

Saved records and skip reasons

- Saved to DB is the count reported by that PULL run. Open View Skips on the affected module for retained blocking reasons.
- BENEFICIARY NOT FOUND LOCALLY means the module record could not be linked to a beneficiary on this tablet. Capture the module record ID and benRegId where reported.
- Normal unchanged/already-up-to-date and duplicate records are excluded from blocking skips. Protected pending local edits may be intentional, not a DB error.
- Not reported is not zero. Retained skip events may repeat and are not a complete unique-record total. Session PULL history can be empty after process restart.

Last failed sync and run timing

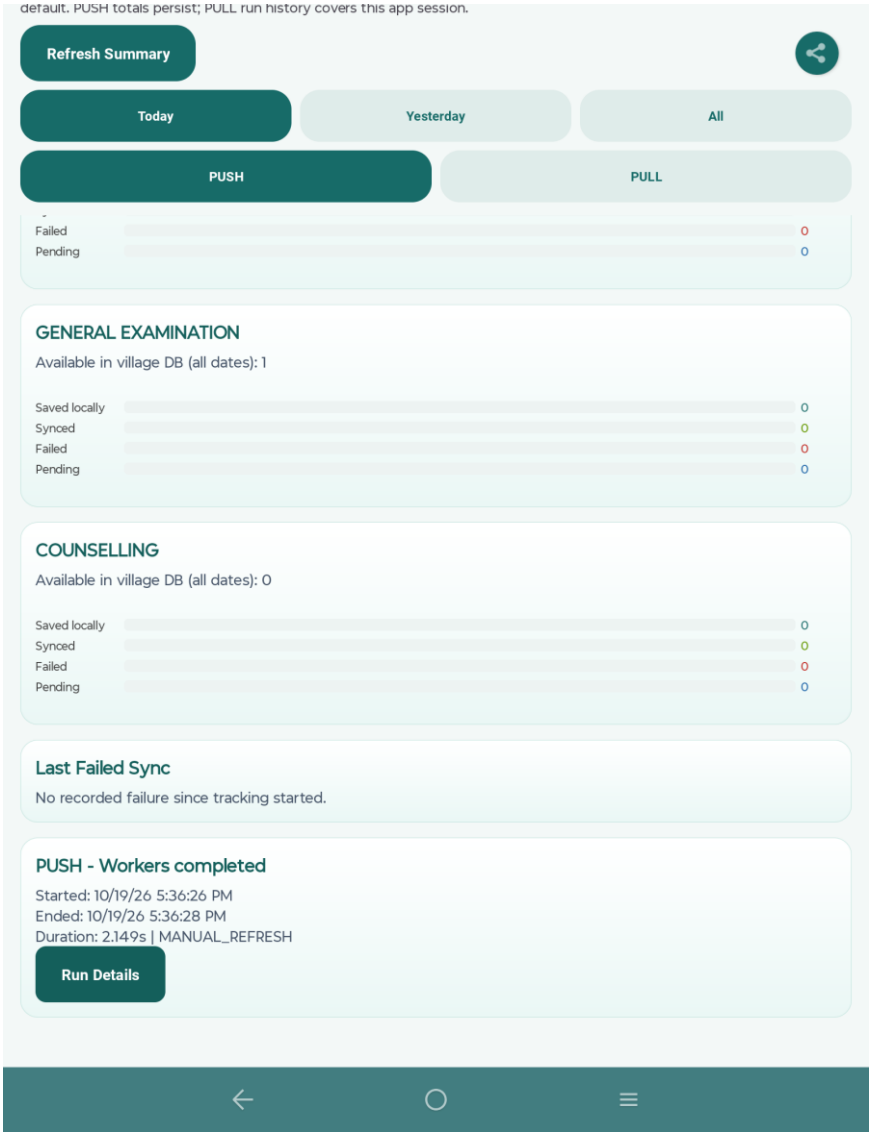


Figure 8 Historical failure section and manual PUSH run timing

How to read this screen

Separate current status from historical evidence

The screenshot says No recorded failure since tracking started. It does not prove that failures never happened before tracking. If a historical Last Failed Sync exists, it may remain after a successful retry; inspect current Failed counts to identify unresolved records.

The shown PUSH run started at 5:36:26 PM, ended at 5:36:28 PM and reported 2.149 seconds for MANUAL_REFRESH. Open Run Details for run identity and technical reports.

Reported state	What to do
Workers completed	Inspect module/record outcomes as well; worker completion alone is not record-level proof.
Running or completion not reported	An end result has not been captured. Refresh and collect evidence rather than assume success or failure.
Needs attention	Open affected module details and correlate the relevant request/response or worker reason.

Date filters

Today and Yesterday use the tablet local calendar dates. All includes retained tracked activity, not every historical save since installation. A yesterday save uploaded today can appear Synced in yesterday cohort. Inventory remains all-date and current.

Earlier beneficiary PUSH failure example

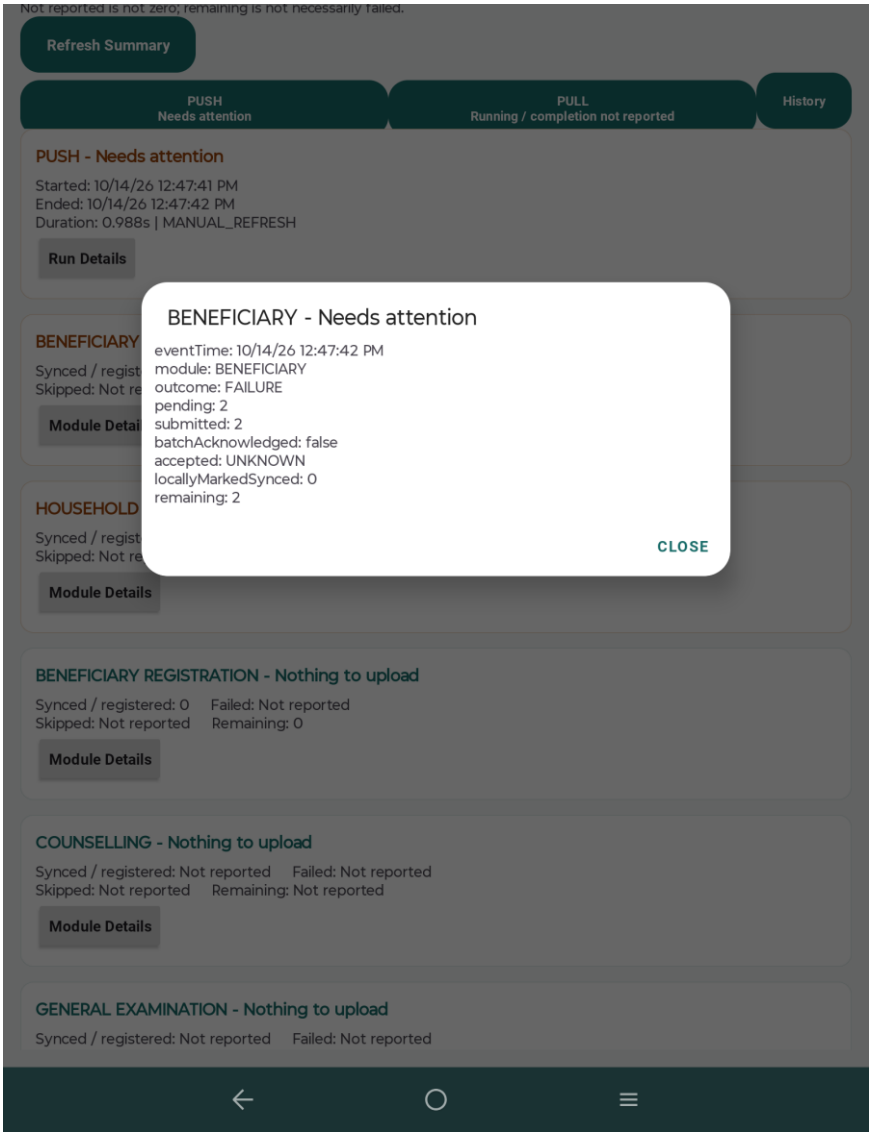


Figure 8A Earlier beneficiary module failure report not the Last Failed Sync card

How to read this screen

This earlier screenshot shows a BENEFICIARY module failure during a manual PUSH. It is not a screenshot of the current Last Failed Sync card. The older background includes History and Module Details; the current UI uses PUSH/PULL and module-specific actions.

Recorded field	What can be concluded
Event time	14 October 2026 at 12:47:42 PM is the timestamp reported in this screenshot.
outcome FAILURE	The module reported a failed operation. This does not supply a record-by-record rejection reason.
pending 2 and submitted 2	Two records were pending and two were submitted according to this report.
batchAcknowledged false	A successful batch acknowledgement was not recorded. The exact backend/network cause is not shown.
accepted UNKNOWN	The accepted-record count is unknown. Do not interpret it as confirmed zero or two confirmed rejections.
locallyMarkedSynced 0	No records were locally marked synced by this reported operation.
remaining 2	Two records remained according to the report. Remaining is not necessarily a confirmed failed-record count.

How to investigate with the current dashboard

- Refresh Summary for the relevant village. Check the affected module and open View Failures when recorded failure details are available.
- Inspect Last Failed Sync for the latest retained historical failure in this village, across any day. It can remain even after recovery; compare current Failed counts.
- Correlate recorded time/run/call identifiers with the actual masked request and response in Logs. If the exact reason was not captured, report it as unknown rather than guessing a server outage.

Keep this screenshot as a historical example only. It does not prove that this failure is still present in the current build or that the current Last Failed Sync card contains the same event.

Share Logs and Summary together

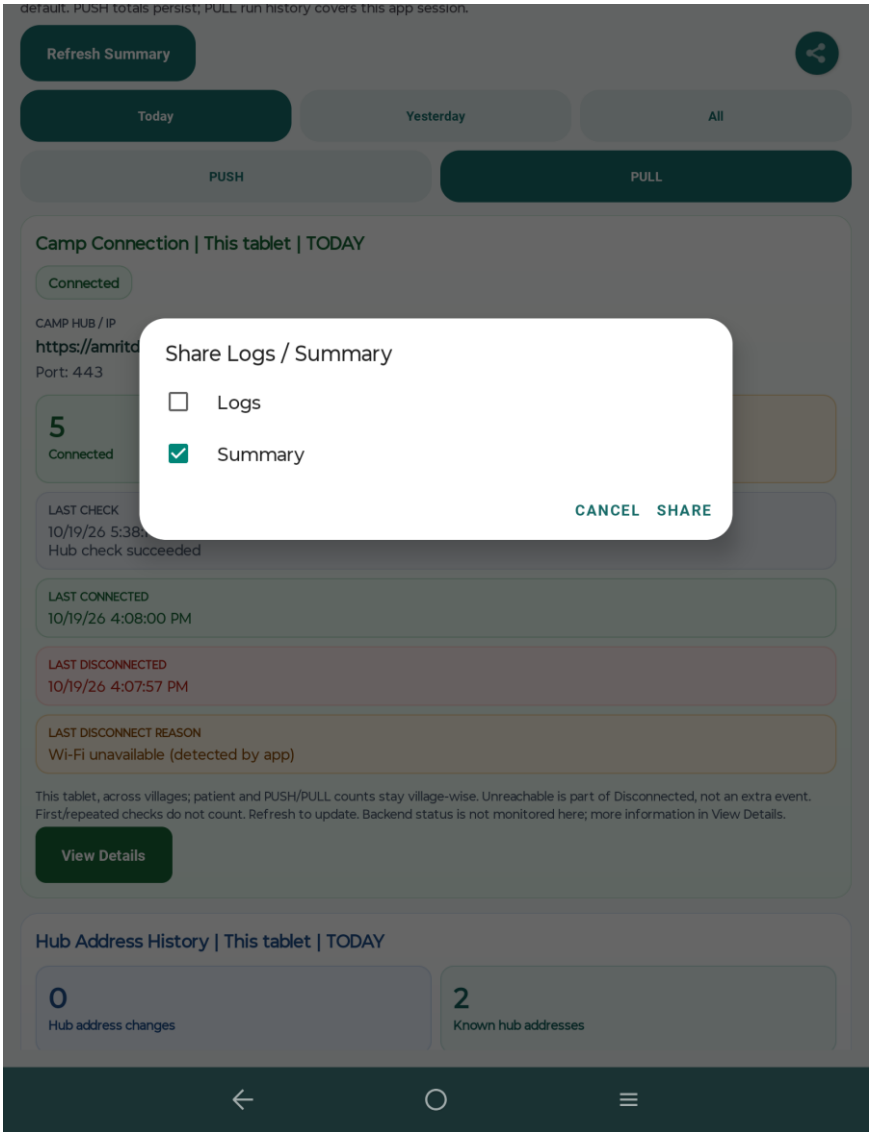


Figure 9 Share dialog opened from Summary with Summary selected

How to read this screen

In this example Summary is checked and Logs is unchecked. Tap Logs as well to send both sections in one ZIP; keep only Summary checked to send the rendered Summary alone.

Sharing steps

1. On Summary choose the required day and PUSH or PULL, then tap Refresh Summary.

2. Tap Share from Logs or Summary. Logs is selected by default from Logs; Summary is selected by default from Summary.

3. Check Logs only, Summary only, or both. With neither selected, Share is disabled. A requested summary must be ready.

4. Share the single generated ZIP through an approved support channel. Summary matches the rendered selection; Logs remain session/manual-direct scope.

ZIP file	Included content
sync-context.txt	Always present: current export context, timezone and selected sections.
manual-direct-sync.log	Only when Logs is selected. Up to 500 retained interactive entries; explicitly reports an empty log section.
sync-summary.txt	Only when Summary is selected. Rendered filter/direction, counts, cards and retained details.

Size and retention

- The old multi-GB raw log file is not attached by this exporter. There is no hard maximum ZIP size in MB; record the actual size when investigating.
- Daily PUSH diagnostics and camp history survive normal app restart. Logs and PULL run history are session-based. Data clear/uninstall is not covered by the persistence guarantee.
- Connection and address-change details each retain the latest 200 events; daily counts persist separately. Missing historical events cannot be reconstructed.

Attach useful evidence

Include build/device/timezone, selected village, day/direction, snapshot time, module and record ID, and run/call ID when available. For two-tablet problems collect both tablet ZIPs plus matching hub/backend evidence before clearing or restarting session logs.

Treat screenshots and exports as restricted support data. Do not attach real patient credentials or unreviewed sensitive records to public tickets.

Implementation summary

I have added the following capabilities to make sync activity and connection issues easier to investigate.

- Village-wise Summary with Today, Yesterday and All filters, and separate PUSH and PULL views.
- PUSH module graphs for locally saved, synced, failed and pending records. Downloads are excluded from local-save activity.
- Separate all-date village inventory, so existing data is not confused with newly saved records.
- PULL results for the latest matching run, with module-level access to retained blocking skip reasons.
- Current module failure details, a historical Last Failed Sync section, and reported run start, end and duration.
- Manual/direct request and response diagnostics with recognized sensitive fields masked. Automatic/raw logs are excluded from the interactive Logs export.
- Tablet-wide camp connection and endpoint history, recorded times/reasons, duplicate-check suppression, and separate Hub and Port labels.
- Selective sharing of Logs, Summary or both in one ZIP from either tab.
- Removal of duplicate TB save calls introduced during merge while retaining diagnostic logging and existing contact logic.

Scope and limitations

This feature improves diagnosis; it does not automatically fix every backend/network sync failure. A connected camp does not guarantee API or record-level success. Older uncaptured events cannot be reconstructed.