

[Slack to Telegram Integration](#)

# Slack to Telegram Integration Checklist

A setup checklist for routing the right alerts to Telegram - so teams don't miss updates and don't drown in noise.

[Open this template](#)[Copy checklist](#)

## What you'll get

- Decide what deserves an instant alert vs a daily digest.
- Route key events (mentions, task assigned, overdue, new lead) to the right Telegram chat.
- Set quiet hours and escalation rules so alerts help instead of distract.
- Test the flow end-to-end before rolling it out.

**Important:** This checklist is UI-agnostic. Where you see placeholders like *[Corcava screen]*, replace them with your actual Corcava navigation labels.

## Setup overview

Use this checklist to design the alerting system first, then connect tools, then refine noise controls.

### 1) Pick your destinations

Choose 1-3 Telegram chats (e.g., Team Updates, Sales Leads, On-call Alerts). Define who owns each.

### 2) Define routing rules

Start with 4 event types: mentions, task assigned, overdue, new lead. Add more only after week 1.

### 3) Connect and test

Connect Slack and Telegram, send test events, then run a 48-hour pilot before broad rollout.

## Pre-flight checklist

- Decide the primary goal: reduce missed updates, speed up response, or improve visibility.
- List the Slack sources you care about (channels, keywords, mentions, apps).
- Create or choose Telegram destinations (groups/channels) and name them clearly.
- Assign an owner for each destination (who maintains rules and noise controls).
- Decide which alerts should bypass quiet hours (rare).

## Connection checklist (generic steps)

- Confirm you have the right permissions to connect Slack (admin approval if required).
- Add the Telegram bot/integration to the target Telegram chat(s) and confirm it can post.
- Connect Slack and Telegram in your integration tool (or Corcava).
- Verify message formatting: sender, channel, link back to Slack context, and timestamps.
- Create a test rule and confirm Telegram receives it within 1 minute.
- Add a fallback: if posting fails, notify an owner (email or alternate chat).

### Corcava placeholders

Replace these with your actual navigation labels:

- *[Corcava screen]* = where you configure Slack/Telegram routing
- *[Automation rule]* = the rule that triggers a Telegram message
- *[Destination]* = Telegram chat or channel

## Routing rules starter set

Start small. The goal is to build trust in alerts. Too many rules too early creates mute behavior.

| Event          | Slack signal                               | Telegram destination | Delivery                       |
|----------------|--|----------------------|--------------------------------|
| Mentions       | Message includes @you, @team, or a keyword | Team Updates         | Immediate                      |
| Task assigned  | New assignment created in your CRM/PM tool | Ops / Delivery       | Immediate                      |
| Overdue task   | Task due date passed                       | Ops / Delivery       | Daily digest + escalation      |
| New lead       | New lead created or moved to qualified     | Sales Leads          | Immediate (business hours)     |
| Incident/alert | #alerts message or monitoring webhook      | On-call Alerts       | Immediate (bypass quiet hours) |

### Noise controls (recommended defaults)

- [ ] Turn on quiet hours for non-urgent events (e.g., 7pm-8am local time).
- [ ] Use digests for overdue items (once per day) instead of one message per task.
- [ ] Add rate limiting: cap to N messages per hour per destination.
- [ ] Avoid routing entire high-volume channels. Prefer keywords, mentions, or specific event webhooks.
- [ ] Include a clear next action in each message (who owns it and by when).

### Escalation rules (keep it human)

- [ ] If an alert is not acknowledged within X minutes/hours, escalate to a smaller group or an owner.
- [ ] For truly critical alerts, use a second channel (e.g., Telegram + email) rather than spamming everyone.
- [ ] Define what 'acknowledged' means (emoji reaction, reply keyword, or status change).

## Test plan (15 minutes)

### Run these tests before rollout:

- Send a message that matches your 'mentions' rule and verify it arrives with a link back to context.
- Create a sample task assignment and verify the Telegram message includes owner + due date.
- Trigger an overdue example (temporarily set due date in the past) and confirm digest behavior.
- Trigger a 'new lead' event and confirm business-hours behavior.
- Turn on quiet hours and confirm non-urgent alerts are delayed or suppressed.

## How to roll out without chaos

- Pilot with one team for 48 hours. Adjust rules based on what gets ignored.
- Publish a short 'rules of the road' message: what gets routed, when, and who owns updates.
- Review weekly for the first month: remove noisy rules, promote useful ones to 'default'.

## FAQ

### How many Telegram destinations should we start with?

One to three. Start with one general updates chat and add a sales or on-call chat only if needed.

### Should we mirror Slack channels into Telegram?

Usually no. Mirror only high-signal events. Full channel mirroring tends to create alert fatigue.

### What should bypass quiet hours?

Only truly urgent alerts (incidents, on-call). Task assignments and leads typically should not.

### What makes a good alert message?

A one-line summary, the owner, the due time, and a link back to the source context.

## Related templates

- Telegram Task Bot Commands Cheat Sheet
- Telegram Daily Standup Template
- Outreach Sequence Tracker Template