

# Slack Bot Runbook

## Service: Slack Bot

**Label:** com.john.slack-bot

**Tier:** P1 (Critical)

## What It Does

Claude-powered Slack bot that listens via Socket Mode and responds to messages. Uses an adapter registry (groq, claude-api, claude-cli, ollama) to select the best AI backend. Maintains conversation history per channel.

## Dependencies

- Node.js: `/opt/homebrew/bin/node`
- Source: `~/system/tools/slack-bot.js`
- Env vars (from plist):
  - `SLACK_BOT_TOKEN` (xoxb-...)
  - `SLACK_APP_TOKEN` (xapp-...)
  - Optional: `ANTHROPIC_API_KEY`, `GROQ_API_KEY`
- State file: `~/system/config/slack-bot-state.json`
- @slack/bolt (npm)
- MC tools (for task context in responses)
- HiveMind (for intel context)

## AI Tier-Routing — Which Model Answers

**Updated:** 2026-06-02 (MC #102825 fix)

The bot's reply engine is `~/system/tools/comms-responder.js` using an **adapter registry**. Adapters run by **priority** (lower number = tried first):

Priority	Adapter	Model	Purpose
5	groq	llama-3.1-8b-instant	Fast fallback for tool-less/voice/trivial messages (~100ms, free tier)
10	claude-api	Sonnet (with tools) or Haiku (tool-less)	Smart conversational messages, tool-use
20	claude-cli	Sonnet	CLI fallback
30	ollama	qwen	Local fallback

## THE RULE (Fix MC #102825, 2026-06-02)

**Groq adapter SKIPS** when `options.tools?.length > 0` (returns `null`), because Groq cannot use Claude tool-use format. Tool-bearing conversational messages therefore **fall through to claude-api** → **claude-sonnet-4-6**.

**Why this matters:** `slack-bot.js` passes `tools: TOOLS` for every real channel/DM/mention message (lines 980-984, 1088-1089). Before the fix, groq intercepted EVERY message first with llama-8b (priority 5) and never reached Sonnet → weak replies. After the fix, groq skips tool-bearing messages → Sonnet answers with live data.

## Routing Flow Examples

**Tool-bearing message** (e.g., "koliko otvorenih taskova imam?"):

1. Try groq (priority 5) → sees `options.tools.length > 0` → returns `null` (SKIP)
2. Try claude-api (priority 10) → sees `tools` present → uses `claude-sonnet-4-6` ✓

**Tool-less message** (e.g., voice, "ping"):

1. Try groq (priority 5) → no `tools` → executes with `llama-3.1-8b-instant` ✓ (fast path)

## Fallback Chain Integrity

- Tool-bearing: groq (skip) → **claude-api (Sonnet)** → claude-cli → ollama
- Tool-less: **groq (llama-8b)** → claude-api (Haiku) → claude-cli → ollama
- Image: groq (skip, no vision) → **claude-api** → claude-cli → ollama

All adapters remain registered. No adapter was removed.

## Cost & Latency Trade-off

- **Before fix:** Every message → llama-3.1-8b-instant (free, ~100ms)
- **After fix:**
  - Tool-bearing → claude-sonnet-4-6 (~\$3/M input tokens, ~500ms) — SMART with live tools
  - Tool-less/voice → llama-3.1-8b-instant (free, ~100ms) — fast path intact
- **Net cost:** ~\$0.10-0.75/day for typical ops channel (20-50 tool-bearing msgs/day)

## Verification

To verify which adapter answered a message:

```
tail -50 ~/system/logs/comms-responder.log | grep "success"

# Tool-bearing message should show:
# [claude-api] success | model: claude-sonnet-4-6

# Tool-less message should show:
# [groq] success | model: llama-3.1-8b-instant
```

## Related Files

- Adapter registry: `~/system/tools/adapters/index.js`
- Groq skip guard: `~/system/tools/adapters/groq.js` (lines 79-83)
- Claude-api model selection: `~/system/tools/adapters/claude-api.js` (lines 141-143)
- Bot tool-passing: `~/system/tools/slack-bot.js` (lines 980-984, 1088-1089)

## Common Failures & Fixes

### Failure 1: "Invalid token" or "not\_authed"

**Symptoms:** Bot fails to connect, log shows authentication error

**Cause:** SLACK\_BOT\_TOKEN or SLACK\_APP\_TOKEN expired or invalid

**Fix:**

```
# Check tokens in plist
cat ~/Library/LaunchAgents/com.john.slack-bot.plist | grep TOKEN

# Get new tokens from api.slack.com
# Update plist with new tokens
```

```
nano ~/Library/LaunchAgents/com.john.slack-bot.plist

# Reload plist
launchctl unload ~/Library/LaunchAgents/com.john.slack-bot.plist
launchctl load ~/Library/LaunchAgents/com.john.slack-bot.plist
```

## Failure 2: "WebSocket connection failed" or "Socket Mode error"

**Symptoms:** Bot starts but cannot receive messages

**Cause:** Slack Socket Mode disabled or network connectivity issue

**Fix:**

```
# Check Socket Mode is enabled in Slack app settings (api.slack.com)
# Verify app-level token has connections:write scope

# Test network
ping slack.com

# Check logs for reconnection attempts
tail -50 ~/system/logs/slack-bot-error.log | grep -i socket

# Bot auto-reconnects, wait 30s or restart
launchctl restart com.john.slack-bot
```

## Failure 3: "Claude API error" or "rate limit exceeded"

**Symptoms:** Bot receives messages but doesn't respond

**Cause:** Anthropic API down or rate limited

**Fix:**

```
# Check if ANTHROPIC_API_KEY is set
env | grep ANTHROPIC

# Test Claude API manually
curl -H "x-api-key: $ANTHROPIC_API_KEY" https://api.anthropic.com/v1/models
```

```
# Bot falls back to CLI if API fails
# Check if Claude CLI is available
which claude

# No immediate action needed, bot handles fallback automatically
```

## Failure 4: "state file corrupted"

**Symptoms:** Bot crashes on start with JSON parse error

**Cause:** Corrupted slack-bot-state.json

**Fix:**

```
# Check state file
cat ~/system/config/slack-bot-state.json

# If corrupted, reset state (loses conversation history)
echo '{"channels":{}}' > ~/system/config/slack-bot-state.json

# Restart
launchctl restart com.john.slack-bot
```

## Restart Procedure

```
launchctl unload ~/Library/LaunchAgents/com.john.slack-bot.plist
sleep 2
launchctl load ~/Library/LaunchAgents/com.john.slack-bot.plist
```

## Verification

```
# Check running
launchctl list | grep slack-bot

# Check logs for connection
tail -20 ~/system/logs/slack-bot.log | grep -i connected

# Test bot in Slack
```

```
# Send message to bot channel, should respond
```

```
# Test backend connection
```

```
node ~/system/tools/slack-bot.js --test
```

# Log Analysis

```
# Standard output (includes message activity, responses)
```

```
tail -50 ~/system/logs/slack-bot.log
```

```
# Errors (auth failures, API errors)
```

```
tail -50 ~/system/logs/slack-bot-error.log
```

```
# Look for connection status
```

```
grep -i "connected|authenticated" ~/system/logs/slack-bot.log | tail -5
```

```
# Look for API errors
```

```
grep -i "api error|rate limit" ~/system/logs/slack-bot-error.log | tail -10
```

# Escalation

If restart doesn't fix:

1. Verify Slack app tokens are valid ([api.slack.com](https://api.slack.com))
2. Check Socket Mode is enabled in app settings
3. Test Anthropic API key if using API backend
4. Verify @slack/bolt npm package is installed
5. Review state file for corruption
6. Check Slack workspace status ([status.slack.com](https://status.slack.com))

---

Revision #1

Created 2026-06-02 21:32:09 UTC by John

Updated 2026-06-02 21:32:09 UTC by John