

Sticky Cards ↔ GoHighLevel

Integration Report

Executive Summary

This document outlines the **two-way sync capabilities** between Sticky Cards and GoHighLevel (GHL). The integration enables real-time customer data synchronization via webhooks and API calls.

Integration Type: Bidirectional (2-Way Sync) - **Sticky Cards → GHL:** Real-time webhooks for customer events - **GHL → Sticky Cards:** API calls for push notifications and data updates

Key Capabilities: - ✓ Real-time webhook events (8 customer-focused events) - ✓ Push notification API (callable from GHL via middleware) - ✓ Customer data synchronization (create/update) - ✓ Points/loyalty balance updates - ✓ Customer tagging and segmentation - ✓ Bidirectional data flow between systems

1. Outbound Data (Sticky Cards → GoHighLevel)

What Sticky Cards Can Send

Sticky Cards emits real-time webhook events when customer actions occur. These events are sent via HTTP POST to your configured webhook endpoint.

✓ Available Event Types

Customer Action	Event Name	Supported	GHL Sync
Card installed or activated	CardInstalledEvent	✓ Yes	Create/Update Contact field: card_installed = "installed"
Card removed or deleted	CardRemovedEvent	✓ Yes	Update Contact (custom field: card_status = "removed")
Customer taps or opens	N/A	✗ No	-

card

				Update (last_act
Card scanned at POS	CardScannedEvent	✔ Yes		Note: St scanner not cus card op
Referral link shared	N/A	✘ No	-	
Referral completed	CustomerReferralCreatedEvent	✔ Yes		Update Add Tag ("Referr
Message sent (push, SMS, email)	N/A	✘ No	-	
Message opened or clicked	N/A	✘ No	-	
Loyalty milestone reached	CardBalanceUpdatedEvent	✔ Yes		Update (custom loyalty_
Customer engagement	CardScannedEvent, PaymentCompletedSuccessfulEvent, FeedbackCreatedEvent	✔ Yes		Update (last_act Create Note/Op
Inactivity threshold (30 days)	N/A	✘ No	-	
Tag or attribute updated	CustomerSegmentLinkedEvent	✔ Yes		Apply G
Customer unsubscribed	N/A	✘ No	-	

Complete Event Catalog

Event Type	Description	Priority	
CustomerCreatedEvent	New customer registered	High	Cr
CardInstalledEvent	Card activated by customer	High	Up (c
CardIssuedEvent	Card created/assigned to customer	Medium	Up (c

CardRemovedEvent	Customer deleted card	Medium	Up (c
CardScannedEvent	Card scanned by staff at POS (not customer opens)	Low	Up (la
CardBalanceUpdatedEvent	Points/balance changed	High	Up (lc
CardExpiredEvent	Card reached expiration	Medium	Up (c 'e2 Ta
CustomerReferralCreatedEvent	Referral completed	High	Ac re
CustomerSegmentLinkedEvent	Customer tagged/segmented	Medium	Af
FeedbackCreatedEvent	Customer submitted feedback	Medium	Cr
PaymentCompletedSuccessfulEvent	Payment successful	High	Cr Of
PaymentCompletedFailedEvent	Payment failed	High	Cr (fc
PaymentRefundedEvent	Payment refunded	Medium	Cr
RecurrentPaymentCompletedFailedEvent	Subscription payment failed	Medium	Cr

Webhook Event Payload Structure

Fields Included in All Events:

Field	Description	Supported	Used For
event_type	Event name (e.g., “CardInstalledEvent”)	✔ Yes	Route to correct handler
timestamp	Event occurrence date/time	✔ Yes	Track activity timeline
customer_id	Sticky Cards customer ID	✔ Yes	Entity mapping
card_id	Sticky Cards card ID	✔ Yes	Card tracking
email or phone_number	Customer contact info	✔ Yes	Match to GHL Contact
event_source	Channel/feature that triggered event	✘ No	-

status	Success/failure state	✗ No	-
metadata	Contextual info (referral code, milestone)	✗ No	-
campaign_id or message_id	Related campaign	✗ No	-
location	Geo-triggered location	✗ No	-
referral_count	Total referrals	✗ No	-

Customer Matching Strategy:

- Primary identifier: phone_number - Secondary identifier: email - Webhook payloads include contact info for matching across systems

Important Clarification: - **CardScannedEvent** tracks when staff scan the customer's card at the POS (in-store redemption/usage), NOT when a customer opens or views their card in their mobile wallet. - Sticky Cards does **not** provide tracking for customer card opens/views in the wallet app.

Webhook Configuration

- **Delivery Method:** HTTP POST to your endpoint
- **Delivery Timing:** Real-time (not batched)
- **Payload Customization:** Can select which event types to receive
- **Retry Logic:** Implement on your side (webhooks may fail)

1.5 Field-Level Mapping (Sticky Cards → GHL)

Following Phorest Integration Pattern: Two-phase sync via local database

Sticky Cards Field	Local DB Column	GHL Target	Integr.
Customer Identification			
customer_id	stickycardsId	stickycards_customer_id (custom)	Str:
email	email	Contact.email	Em
phone_number	mobile	Contact.phone	Pho
first_name	firstName	Contact.firstName	Str:

last_name	lastName	Contact.lastName	Str
Card Data			
card.id	cardId	card_id (custom)	Str
card.status	cardStatus	card_status (custom)	En
card.balance	loyaltyPoints	loyalty_points (custom)	Nu
card.installed_date	cardInstalledDate	card_installed_date (custom)	Dai
Engagement Tracking			
last_activity	lastActivity	last_activity (custom)	Dai
last_purchase_date	lastPurchaseDate	last_purchase_date (custom)	Dai
referral_count	referralCount	referral_count (custom)	Nu
Segmentation			
segments[]	segments (JSON)	Tags	Arr

Notes: - Primary matching: phone_number (normalized: strip + and spaces) - Secondary matching: email (case-insensitive) - Phone format normalization: +1-555-1234 → 5551234 for matching - If no match found: Create new GHL contact - Custom fields created on first sync (auto-provisioned by middleware)

2. Inbound Data (GoHighLevel → Sticky Cards)

What Sticky Cards Can Receive

Sticky Cards provides REST APIs that can be called via middleware to update customer data.

✗ Limitations

Sticky Cards CANNOT receive webhooks from external systems.

All inbound actions must be triggered via **API calls** initiated by your middleware or GHL.

Additional API Limitations: - ✗ Cannot schedule delayed messages (no time-based triggers) - ✗ Cannot start/stop automation sequences via API - ✗ Cannot pause/resume message flows programmatically - ✗ Cannot set

suppression periods via API - ✖ No API for message delivery/engagement metrics (dashboard only) - ✖ No API for customer activity history retrieval - ✖ Limited custom metadata tracking (see field mapping below)

✔ Available API Actions

Action	Supported	API Endpoint (Reference)	Use Case
Trigger push notification	✔ Yes	POST /v1/pushes/customer	Send promotional/transactional messages
Update customer record	✔ Yes	POST /v1/customers	Sync name, email, phone number
Apply/remove tags	✔ Yes	Customer attributes API	Segmentation
Update points balance	✔ Yes	Card balance API	Loyalty rewards sync
Deactivate card	✔ Yes	Card status API	Sync card status
Reactivate card	✔ Yes	Card status API	Sync card status
Update opt-in/opt-out	✔ Yes	Customer preferences API	Consent management
Initiate data refresh/resync	✔ Yes	Customer APIs	Manual sync

Inbound Action Details

Action	GHL Trigger	Sticky Cards Target	Method	Implementation
Send Push Notification	API call from GHL	Customer push endpoint	API POST	GHL HTTP request action
Update Customer Data	Contact field updated	Customer create/update	API POST	Sync via middleware
Update Points Balance	Custom field changed	Card balance update	API POST	Transaction processor
Apply Customer Tags	Tag added in GHL	Customer segments	API POST	Tag sync processor
Issue New Card	New contact created	Card issue endpoint	API POST	Initial sync

Deactivate Card	Tag applied	Card status endpoint	API PUT	Status sync
Reactivate Card	Tag applied	Card status endpoint	API PUT	Status sync

2.5 Field-Level Mapping (GHL → Sticky Cards)

Sync Triggers from GHL:

GHL Event	Middleware Action	Sticky Cards API Call		
Contact Created	Check if has stickycards_customer_id	POST /v1/customers	first_name, last_name, email, phone	
Contact Phone Updated	Find by stickycards_customer_id	POST /v1/customers	phone	
Contact Email Updated	Find by stickycards_customer_id	POST /v1/customers	email	
Tag Applied	Map GHL tag → SC segment	PUT /customers/{id}/segments	segment	
Custom Field Changed: bonus_points	Add points to card	PUT /cards/{id}/balance	balance (integer)	
Tag “churned” Applied	Deactivate card	PUT /cards/{id}/status	status: "Inactive"	
Tag “reactivated” Applied	Reactivate card	PUT /cards/{id}/status	status: "Active"	

Field Mapping Table (GHL → Sticky Cards):

GHL Field	Local DB Column	Sticky Cards Target	Direction
Contact.email	email	customers.email	GHL → SC
Contact.phone	mobile	customers.phone_number	GHL → SC
Contact.firstName	firstName	customers.first_name	GHL → SC
Contact.lastName	lastName	customers.last_name	GHL → SC

Tags	segments (JSON)	customers.segments[]	GHL → SC
loyalty_points (custom)	loyaltyPoints	cards.balance	GHL → SC
card_status (custom)	cardStatus	cards.status	GHL → SC
opt_in_status (custom)	optInStatus	customers.preferences	GHL → SC

✗ **NOT Supported Custom Metadata:** - review_submitted, referral_completed, last_purchase_date (not tracked by SC API) - inactive_days, campaign_stage, suppress_until (no automation control) - message_preference, geo_location, review_score (no API fields) - loyalty_tier (only for Membership card type, not general sync) - event_trigger_source, referred_friend_email (no custom fields)

Sync Strategy: - **Upsert Logic:** If customer exists (matched by email/phone), update fields. Otherwise, create new. - **ID Tracking:** Store stickycards_customer_id in GHL custom field for future lookups - **Conflict Resolution:** GHL is source of truth for contact info updates - **Rate Limiting:** Batch updates every 5 minutes (avoid hitting Sticky Cards API limits)

2.6 Data Access & Reporting Limitations

✗ **No API Access for Analytics/Metrics:**

Sticky Cards does **not** provide API endpoints for: - Message delivery status (delivered, failed, pending) - Message engagement (opens, clicks, timestamps) - Customer activity history (last push sent, last click, engagement frequency) - Campaign performance metrics - Referral tracking history - Card activation/usage analytics

All metrics are available in the Sticky Cards dashboard (UI only).

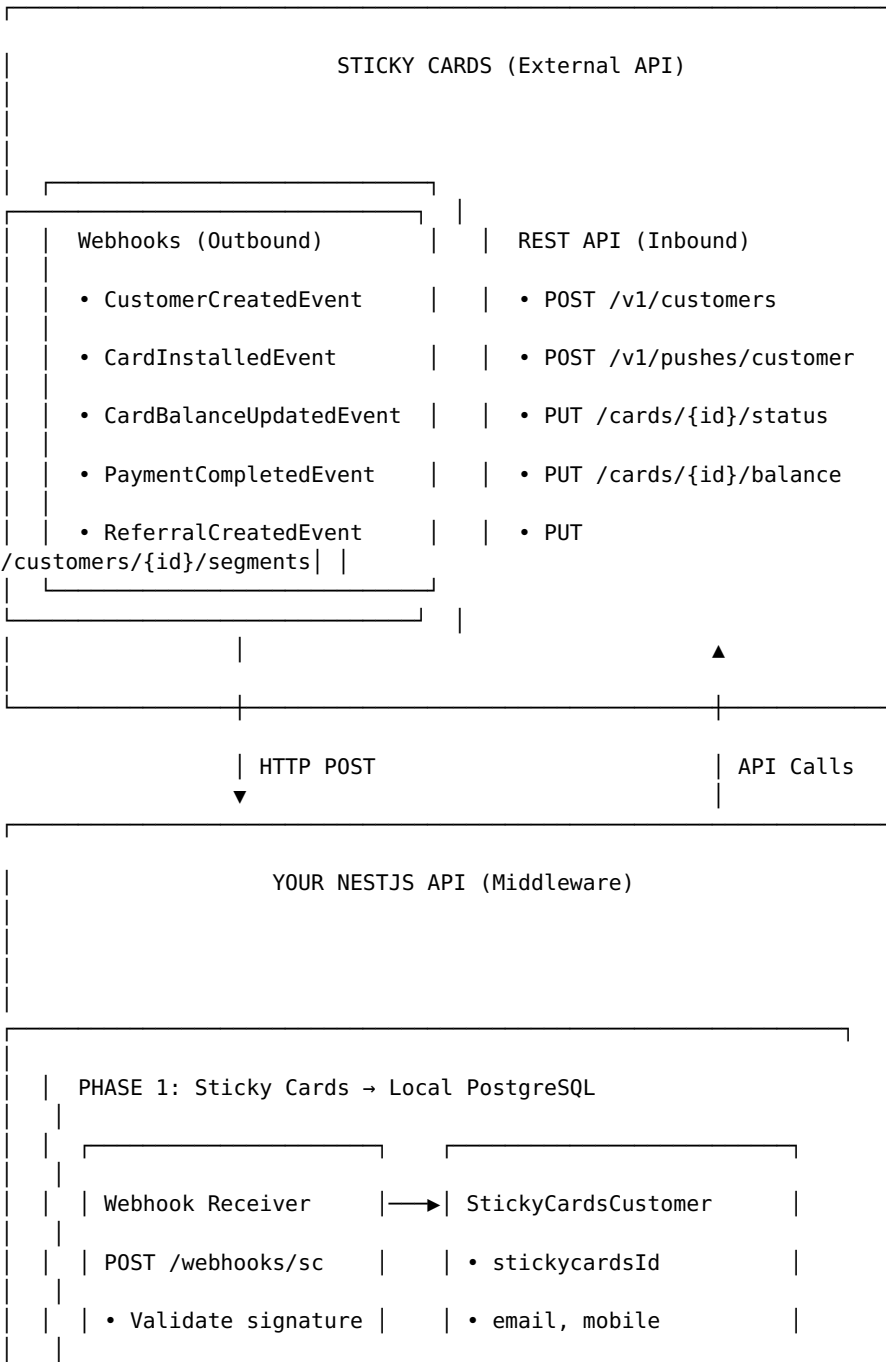
Workaround for Tracking: - Store engagement events in your local DB when webhooks fire - Track last_activity timestamp from webhook events - Calculate inactive_days in your middleware (not provided by SC) - Use GHL for campaign/message tracking instead of SC

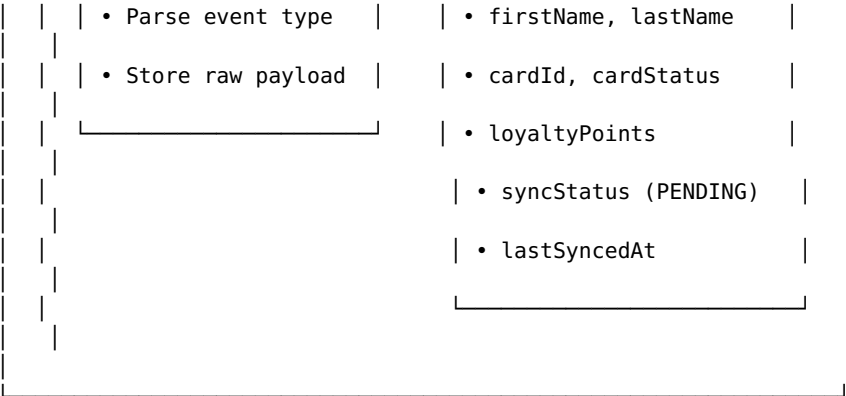
3. Two-Way Sync Architecture

Architecture Pattern: Two-Phase Sync (Following Phorest Integration)

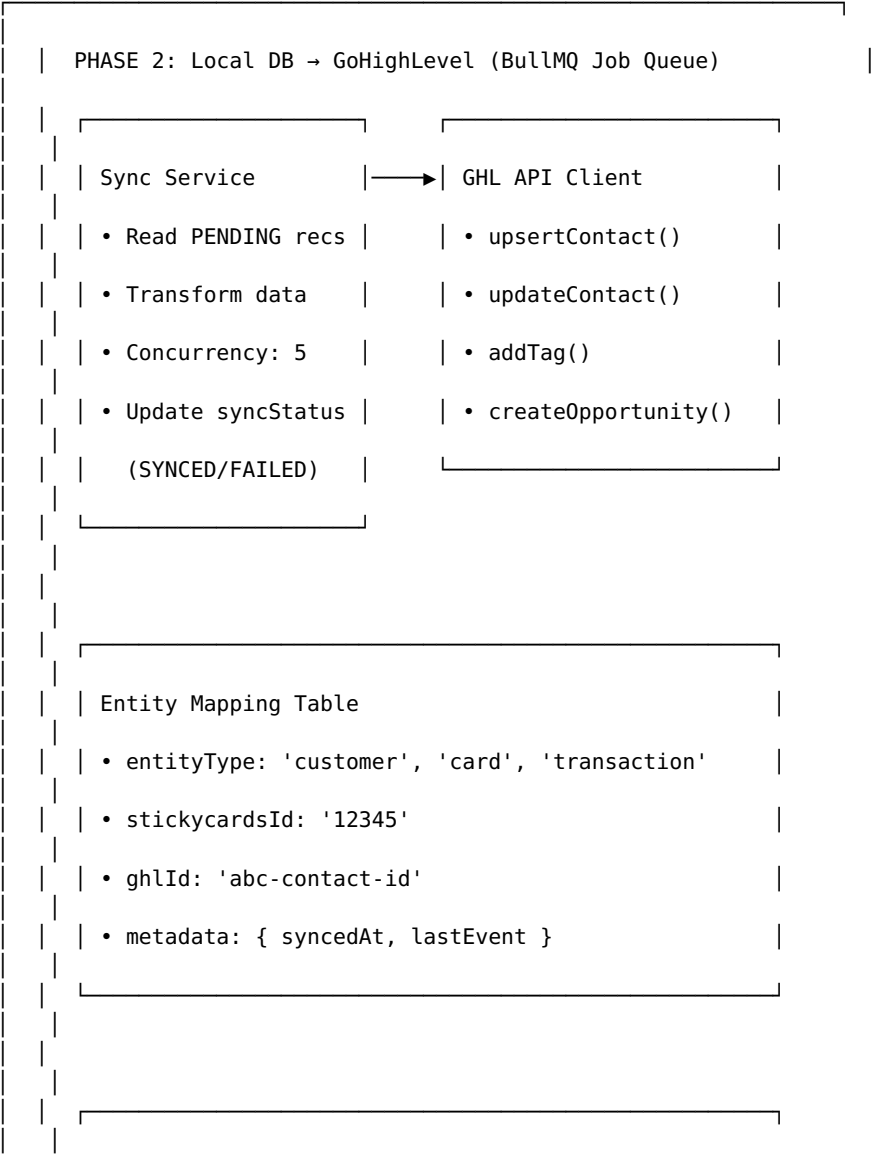
Why Two-Phase? 1. **Reliability:** Local DB acts as buffer if GHJ API is down 2. **Auditability:** Full history of received webhooks + sync attempts 3. **Retry Logic:** Failed syncs can be retried without re-fetching from Sticky Cards 4. **Performance:** Batch processing with concurrency control 5. **Consistency:** Single source of truth for sync status

Data Flow Diagram





|



	Sync Audit Logs	
	• SyncRunSummary (batch stats, run status)	
	• SyncLog (per-record success/fail, retry count)	

|



	REVERSE SYNC: GHL → Sticky Cards (On-Demand)	
	• Contact Updated → POST /v1/customers	
	• Tag Added → PUT /customers/{id}/segments	
	• Custom Field Changed → PUT /cards/{id}/balance	
	• API Call → POST /v1/pushes/customer (push notification)	

| GHL API



GOHIGHLEVEL (CRM)		
	Contact Records	
	• Contact.email, phone, firstName, lastName	
	• Custom Fields:	
	- stickycards_customer_id	
	- card_id, card_status, loyalty_points	
	- card_installed_date, last_activity	
	- last_purchase_date, referral_count	
	• Tags: "Card Holder", "Referrer", "VIP", etc.	
	• Opportunities: Payment transactions	
	• Notes: Feedback, failed payments	

