



AUTOMATION CASE STUDY

Automated Discord Community Onboarding & Recurring Membership Workflow

A client-ready GoHighLevel + Make.com + Discord integration for a paid trading community

GOHIGHLEVEL

CRM, form, recurring product & email

MAKE.COM

Webhook orchestration & data handoff

DISCORD

Fresh community invite generation

CLIENT PROFILE

Paid Trading Community — Client Anonymized

Objective: turn a website form submission into a paid member onboarding flow with automated Discord access, CRM synchronization and recurring monthly billing.

Business Problem & Solution

A practical membership automation designed to remove manual onboarding steps while keeping GoHighLevel as the operational CRM.

Executive summary

A visitor joins the client's trading community through a form on the website. The submission enters GoHighLevel, triggers a webhook to Make.com, generates a fresh Discord channel invite, updates the member's GHL contact record, and then triggers a second GHL workflow that emails the invitation to the member. The community is also configured as a recurring product in the GHL form/checkout so active members can be charged monthly.

The challenge

A paid Discord community needs more than a simple sign-up form. New members must be captured in the CRM, granted community access quickly, and kept connected to the recurring membership process. Handling each invite and contact update manually creates delays, inconsistent records and unnecessary administrative work.

Solution objectives

- Capture every new community member through the website/GHL form.
- Generate a fresh Discord invitation automatically after the form submission.
- Write the invite/status back to GoHighLevel so the CRM remains synchronized.
- Send the invite to the member by email without manual intervention.
- Support the paid community model with recurring monthly billing configured as a GHL product.

Technology stack

Platform	Role in the solution	Why it matters
GoHighLevel	Website form/checkout, CRM contact, workflow triggers, tag logic and email delivery.	Central system of record and customer-facing automation.
Make.com	Receives the webhook, creates the Discord invite and updates the GHL contact.	Connects GHL and Discord while controlling the integration logic.
Discord	Creates the member invitation to the trading community channel/server.	Provides controlled access to the paid community.

Reporting note: No conversion-rate or time-saved measurements were provided, so this case study describes verified workflow behavior and qualitative business impact rather than inventing performance percentages.

End-to-End Automation Design

The integration is intentionally split between GHL for CRM/customer communication and Make.com for the Discord API handoff.

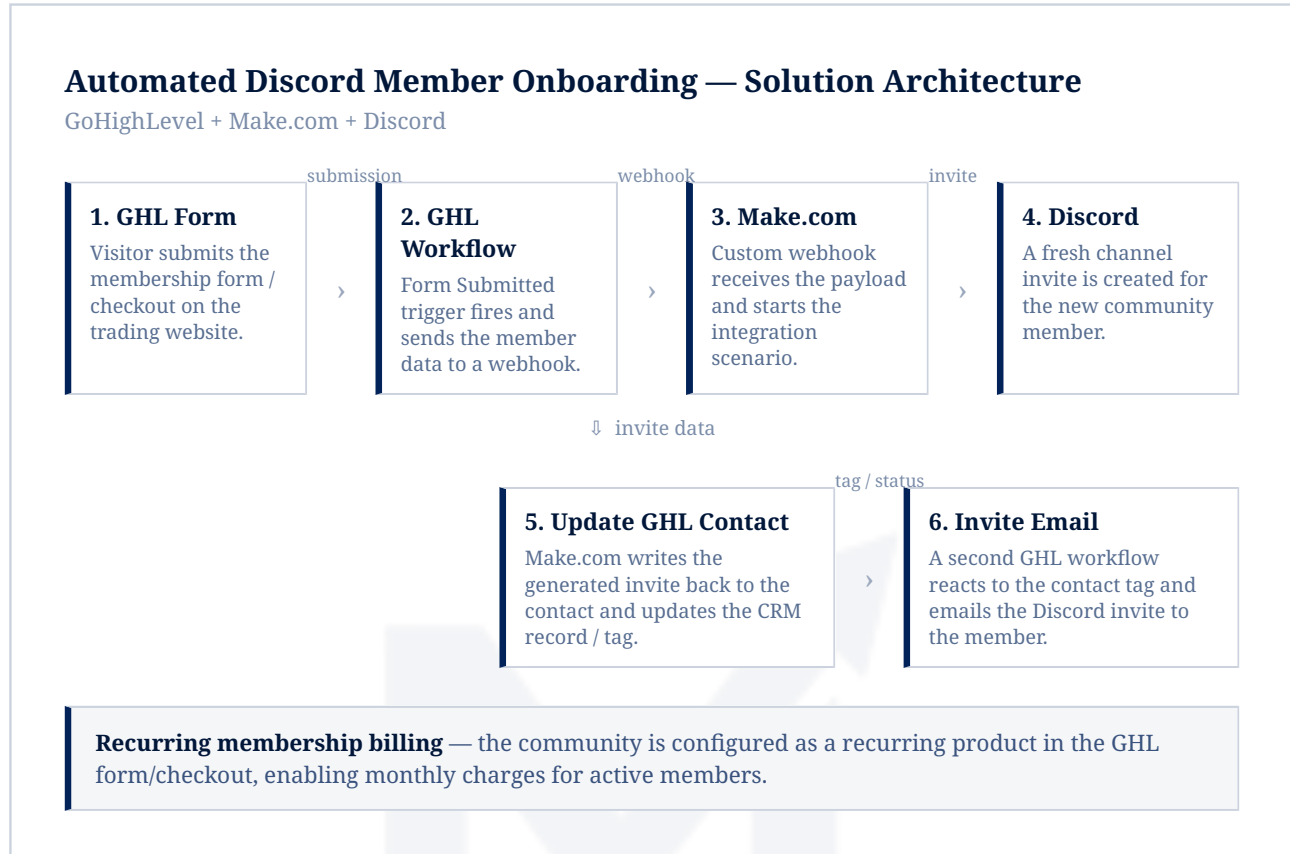


Figure 1. End-to-end member onboarding architecture.

Workflow at a glance

- 01 Member submits the paid community form on the client website.
- 02 GHL Workflow 1 detects the form submission and fires a webhook.
- 03 Make.com receives the payload through its custom webhook.
- 04 Make.com creates a new Discord channel invite for the member.
- 05 The generated invite data is written back to the matching GHL contact and the contact is updated/tagged.
- 06 GHL Workflow 2 detects the invite-related tag/status and sends the member an email containing the Discord invite.
- 07 Recurring membership billing continues through the community product configured in the GHL form/checkout.

Stage 1 — GHL Form Submission to Webhook

The first GHL workflow is the entry point for the integration and hands the new member to Make.com.

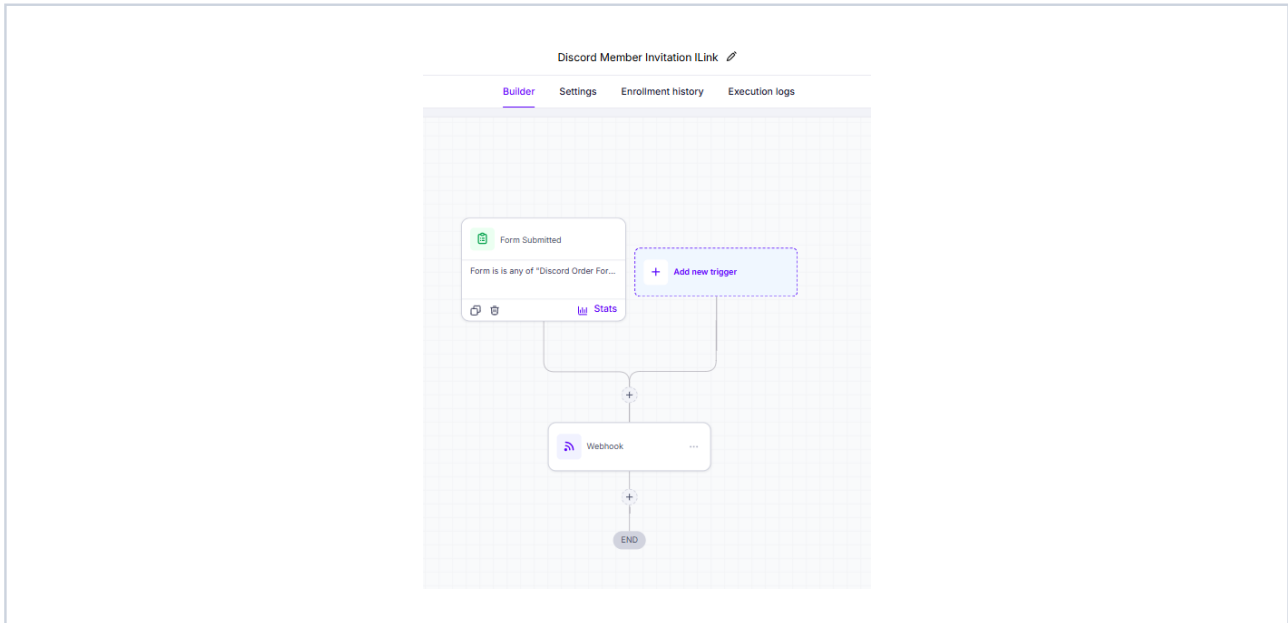


Figure 2. GoHighLevel workflow: Form Submitted trigger → Webhook action.

What this workflow does

- The workflow enrolls a contact when the selected Discord/community order form is submitted.
- The webhook action forwards the relevant member/contact data to the Make.com scenario.
- GHL retains the original lead/customer record, so the integration does not create a disconnected onboarding system.
- The community product is configured in the form/checkout as a recurring membership, which supports monthly charging as part of the same customer journey.

Design choice

Using a webhook keeps the GHL workflow lightweight. GHL handles enrollment and customer state, while Make.com handles the cross-platform operation that needs to talk to Discord.

Stage 2 — Make.com Creates the Discord Invite

Make.com is the orchestration layer that receives the GHL webhook, generates community access, and synchronizes the result back to the CRM.

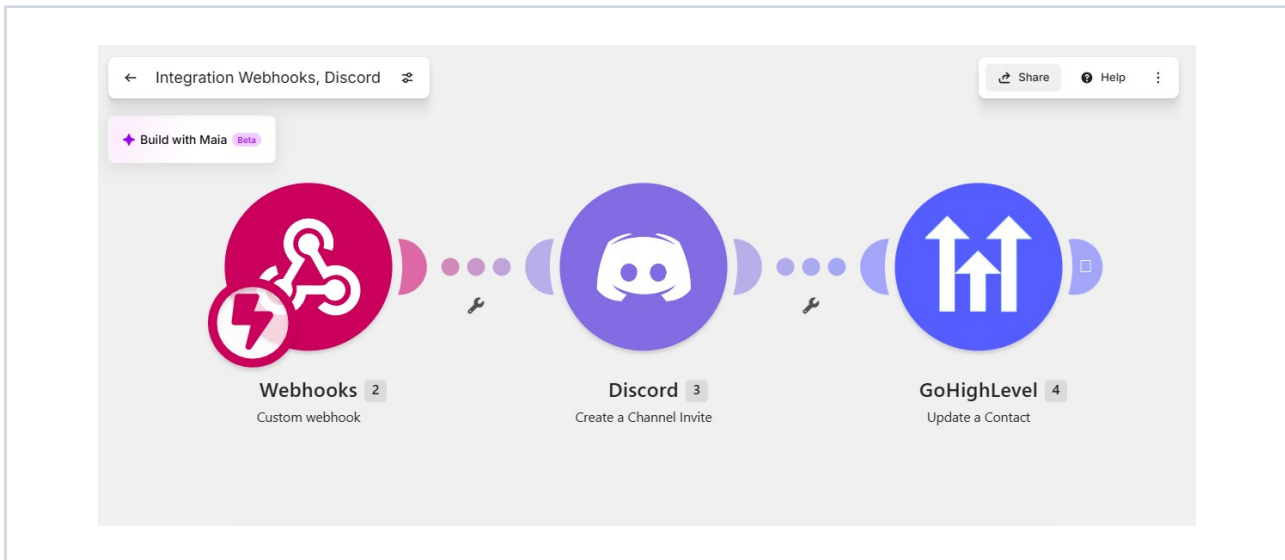


Figure 3. Make.com scenario: Custom Webhook → Discord Create a Channel Invite → GoHighLevel Update a Contact.

Scenario logic

- Custom Webhook: receives the member payload from the first GHL workflow.
- Discord — Create a Channel Invite: creates a fresh invite URL for the trading community.
- GoHighLevel — Update a Contact: writes the generated invite/status back to the member's CRM record and enables downstream tag-based automation.
- The updated contact becomes the handoff point back into GoHighLevel for the email-delivery workflow.

Why this is scalable

The same integration pattern can process each new submission independently. The automation does not rely on an operator manually creating Discord links or copying them into CRM records.

Stage 3 — GHL Sends the Discord Invite by Email

After Make.com updates the contact, a second GHL workflow handles the customer-facing message.

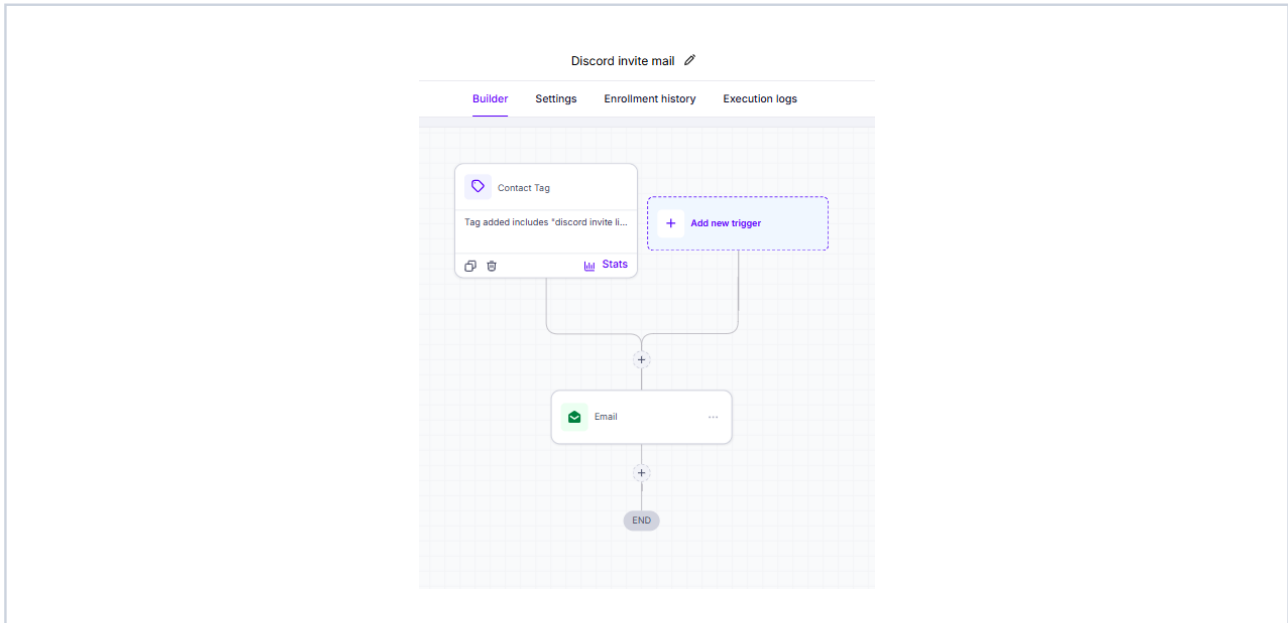


Figure 4. GoHighLevel workflow: invite-related Contact Tag trigger → Email action.

Delivery logic

- The second workflow starts when the invite-related contact tag/status is added.
- The email action sends the newly generated Discord invite to the member's email address.
- Because the invite data is synchronized to GHL first, the member record and communication history remain centralized in the CRM.
- Separating invite generation from email delivery makes the system easier to troubleshoot and allows the email step to be changed without rebuilding the Discord integration.

Member experience

From the customer's perspective, the journey is simple: submit the community form, complete the paid membership flow, and receive the Discord access link automatically by email.

Operational Benefits, Controls & Next Steps

The automation connects acquisition, access delivery, CRM management and recurring membership into one repeatable onboarding process.

Before vs. after

Without this automation	With the implemented automation
Discord invite may need to be created and sent manually.	Fresh invite generation is triggered automatically from the member journey.
CRM records and access delivery can become separate tasks.	Invite/status is written back to the GHL contact before delivery.
Member onboarding speed depends on staff availability.	Email delivery is triggered automatically after the integration completes.
Billing and community access can require separate handling.	The community is sold as a recurring GHL product within the same onboarding flow.

Business benefits

- Faster member access with fewer manual handoffs.
- Consistent onboarding experience for every new community subscriber.
- Cleaner CRM data because the access information is returned to the correct GHL contact.
- Reduced operational dependency on staff for Discord invite creation and email sending.
- A foundation that can scale with the trading community as membership volume grows.
- Recurring monthly revenue collection supported by the GHL product configuration.

Recommended next enhancements

- Gate invite creation behind confirmed successful payment status, if that validation is not already part of the form/checkout configuration.
- Configure Discord invites with an appropriate expiration and/or maximum-use policy for stronger access control.
- Add failure alerts for webhook, Discord or GHL update errors so an admin is notified when onboarding cannot complete.
- Automate access removal or a cancellation workflow when a recurring membership is cancelled or payment fails.
- Track operational KPIs such as onboarding completion rate, invite-email delivery rate and failed automation runs.

Conclusion

This solution turns a paid-community sign-up into a connected automation: GHL captures and bills the member, Make.com generates Discord access, GHL stores the result, and the member receives the invite by email. The architecture reduces repetitive admin work while keeping the CRM at the center of the membership lifecycle.