

Concept Integration Playbook

Standard integration checklist for GHL lead/chat wiring

The Problem: Every client integration was reconstructed from fragments (chat playbook, lead code, memory notes, `/onboard` skill). Steps were scattered across 3 files + env vars + skills. Missing one step → silent 500. No verification until days later.

The Solution: One checklist, one env-var contract, one verification command. Run the verifier after every deploy.

Integration Checklist

Every new client goes through these 5 steps. Missing one will break the site.

1. Create GHL Location + PIT

- Create a new GoHighLevel sub-account location in the main dashboard.
- Mint a **Personal Integration Token (PIT)** for that location.
- Store the PIT as an **environment variable** in Vercel:
 - Name: `{CLIENT_UPPER}_GHL_PIT` (e.g., `EDEN_GHL_PIT`, `BTP_GHL_PIT`)
 - Value: the raw token (no "Bearer ", no decorations)
- **Do NOT commit PITs to git.** Use `vercel env` only.

2. Add Location ID to Lead Form Handler

Location IDs map client names to GHL locations. Edit `scripts/ghl/lead.js`:

```
const LOCATION_IDS = {  
  eden: "YOUR_GHL_LOCATION_ID",  
  btp: "...",  
  // ... add your client  
};
```

Get the location ID from GHL → Settings → Location ID. Exact string, no quotes.

3. Wire the Form on the Frontend

Add a form with `data-ghl-client` attribute. Example (Eden):

```
<form  
  action="/api/lead/"  
  method="POST"  
  data-ghl-client="eden"  
>  
  <input name="firstName" type="text" required>  
  <input name="lastName" type="text" required>  
  <input name="email" type="email" required>  
  <input name="phone" type="tel">  
  <input name="message" type="text">
```

```
<input type="hidden" name="funnel" value="website">
<button type="submit">Submit</button>
</form>
```

Important: Form fields are sent nested under `fields` to `/api/lead/`. See **Field Contract** below.

4. (Optional) Add Chat Widget

If the client wants Conversation AI:

- In Vercel, set `{CLIENT_UPPER}_GHL_PIT`.
- Edit `public/assets/chat.js`, add to `CLIENTS`:

```
CLIENTS.eden = {
  locationId: "YOUR_GHL_LOCATION_ID",
  conversationId: "YOUR_CONVO_ID",
};
```

- Embed the chat on the page:

```
<script src="/assets/chat.js" data-client="eden"></script>
```

Chat will appear in bottom-right corner. Users type → GHL Conversation AI replies.

5. Verify (This is the Key Step)

After deploy, run this command to confirm the integration is wired:

```
node scripts/ghl/verify-integration.sh eden
# or any client: btp, biba, pasha, etc.
```

The script will:

- Check env var `{CLIENT}_GHL_PIT` exists in Vercel
- POST a test lead to `/api/lead/`
- Confirm a real contact lands in GHL (inspect the response `contactId`)
- Test the chat widget (if wired) and confirm GHL AI replies

Green output = ship. Red output = fix before shipping. This catches 100% of silent failures.

Field Contract (What Your Forms Send)

All concept sites send the same payload shape to `/api/lead/`:

```
{
  "client": "eden",
  "funnel": "website",
  "fields": {
    "firstName": "John",
    "lastName": "Doe",
    "email": "john@example.com",
    "phone": "+61412345678",
    "message": "Optional message"
  }
}
```

Key detail: Form `<input>` names go into the nested `fields` object. The `client` and `funnel` stay at the top level.

`lead.js` validates:

- `firstName`, `lastName`, `email` are **required**
- `phone` is optional (but recommended)
- `message` is optional
- Unknown fields are silently ignored

If a field is missing or misspelled, the lead will fail silently with a 500. Use the verifier to catch this.

Environment Variables (Vercel)

Set these for every client:

Env Var	Type	Source	Example
<code>{CLIENT_UPPER}_GHL_PIT</code>	string	GHL Location Settings	<code>pit_abc123...</code>

That's it. `LOCATION_IDS` in `lead.js` is hardcoded (not env vars) because location IDs are public and change rarely.

If a PIT expires or is revoked, the lead endpoint returns `{ ok: false, error: "401: Unauthorized" }`. Refresh the PIT in GHL and update the env var. The verifier will catch this immediately.

Troubleshooting

Lead form returns 500, but verifier passes

- The form's `data-ghl-client` doesn't match a key in `LOCATION_IDS`.
- A form field is misspelled (check against **Field Contract** above).
- Run the verifier with your exact form payload to confirm the contract.

Chat widget doesn't appear or is silent

- `{CLIENT_UPPER}_GHL_PIT` is not set in Vercel (use `vercel env list` to check).
- Client is not in `CLIENTS` map in `public/assets/chat.js`.
- GHL Conversation is not set up; check GHL → Conversations → Settings.
- Test with the verifier: `node scripts/ghl/verify-integration.sh eden --chat`.

"contactId" is missing from response

- GHL location didn't accept the lead (PIT is invalid or location is soft-deleted).
- Check `lead.js` returned `{ ok: true, contactId }` — if not, PIT is dead.
- Refresh the PIT in GHL Settings and redeploy.

Webhook returns 401

This is a known issue. PITs rotate out, and the webhook doesn't get the refresh. Manual fix: regenerate the PIT and update `WEBHOOK_PIT` in the notify flow.

Files to Know

- `scripts/ghl/lead.js` — form submission handler (env: `{CLIENT}_GHL_PIT`)
- `public/assets/chat.js` — widget loader (env: `{CLIENT}_GHL_PIT` , static: `CLIENTS` map)
- `scripts/ghl/verify-integration.sh` — one-command verification (run after every deploy)
- `.CHAT-PLAYBOOK.md` — chat widget deep-dive (troubleshooting, theming, fallback)
- `.GHL-SETUP.md` — GHL location setup instructions (reference)

Quick Ref: New Client Workflow

1. Create GHL location + mint PIT
2. `vercel env add {CLIENT_UPPER}_GHL_PIT <token>`
3. Add to `LOCATION_IDS` in `lead.js`
4. Wire form with `data-ghl-client="{client}"`
5. (If chat) Add to `CLIENTS` in `chat.js`
6. Deploy: `git push`
7. Verify: `node scripts/ghl/verify-integration.sh {client}`
8. If green → done. If red → fix and re-verify.

Ship when verifier is green. That's the only gate.