Glitch Chat — Tenant Web Interface

Last updated: 2026-05-05 **URL:** https://chat.realtalkha.com/ **Hosted on:** DigitalOcean droplet `142.93.60.203`, PM2 process `glitch-chat` on port 3001

The third channel sub-account owners use to talk to Glitch (alongside Telegram and GHL Conversations). A standalone Next.js 14 app that plugs into the same per-tenant infrastructure as the rest of the Glitch stack.

---

1. How a tenant signs in

1. Owner visits `https://chat.realtalkha.com/`
2. Enters their email → POST `/api/auth/request`
3. Backend looks up email via GHL agency-level `/locations/search?companyId=…`
4. Signs HMAC-SHA256 JWT (Web Crypto, Edge-runtime compatible) with `{loc, email, exp}`
5. POSTs `{email, link, businessName}` to n8n webhook `glitch-magic-link` (workflow id `YmauA2EXSM7kvK7g`)
6. n8n upserts contact in BizDev, sends email via GHL Conversations API
7. Owner clicks link → `/login/<token>` → cookie set → redirected to `/<locationId>/`

Magic link URLs use `/login/<token>` path pattern (not `/api/auth/verify?token=`) to avoid Chrome Safe Browsing phishing-pattern false positives.

Session: HttpOnly Secure cookie, SameSite=Lax, 30-day expiry. Cross-tenant access enforced by Edge middleware which verifies the cookie and injects `x-glitch-loc` header on every `/api/*` request.

2. Per-tenant OpenRouter key

`/api/chat` resolves the tenant's OpenRouter child API key on every request via `getTenantOpenRouterKey(locationId)` — fetches from the master BizDev location's custom values where each tenant's key is stored as a custom value named

`or_key_<locationId>`. 5-minute in-memory cache. Falls back to shared key if missing (dev only).

This is the same child-key system minted by the `Glitch Subaccount Onboarding → Auto OpenRouter Key` n8n workflow (id `EU9jr1lVoSP05T9C`) when a tenant is provisioned. Same key, same cap, same usage attribution — the chat just reads from the master custom-values store rather than the tenant's own customValues.

3. Models

All routing through OpenRouter (no Claude Max for tenants). Intent classifier picks one of:

Intent	Model	Notes
greeting / simple_qa	`openai/gpt-5-mini`	Cheap, fast
crm_action	`openai/gpt-5.4-mini`	Structured ops
creative	`anthropic/claude-sonnet-4.6`	Default fallback (vision + tools)
reasoning	`arcee-ai/trinity-large-thinking`	Free tier, no tools
web_search	`perplexity/sonar-pro`	Specialized search, no tools
reporting	`openai/gpt-5.4`	Analytics
multimodal	`google/gemini-3.1-pro-preview`	Image analysis

When attachments are present, the route is forced to `anthropic/claude-sonnet-4.6` (vision + tools).

When the routed model doesn't support tools (Perplexity, Arcee Trinity), tools are dropped from the streamText call — the model itself becomes the tool.

4. Tools available to Glitch per tenant

Every tool is bound to the verified locationId at request time so the model cannot call another tenant's services.

Tool	Description
`busybee_list_tools(filter?)`	List the 588 GHL operations available from BusyBee
`busybee_call(name, arguments)`	Execute any GHL operation (contacts, opportunities, calendars, conversations, products, payments, etc.)
`arcee_think(question)`	Escalate a hard reasoning task to Trinity-Large (free tier on OpenRouter, still through tenant key)
`web_search(query)`	Live web research via Perplexity Sonar Pro
`apify_scrape(source, target)`	Scrape website / Google Maps / Instagram / LinkedIn / Facebook
`replicate_run(model, inputs)`	Generate image / video / audio. Output URLs persisted locally so they don't expire (Replicate's `replicate.delivery` URLs expire ~1 hour)

Multi-step tool calling: `streamText({ tools, maxSteps: 5 })` — Glitch can chain `busybee_list_tools → busybee_call → busybee_call` in one turn.

5. File upload

Paperclip button next to the input. Supports:

- Images: jpg, png, gif, webp
- Documents: PDF
- Text: txt, md, csv, json (extracted up to 200 KB)

Up to 4 attachments per turn, 10 MB each.

When attachments are present in `/api/chat` body, the route forces Claude Sonnet 4.6 and builds multimodal content blocks: images become `{type:'image', image:'data:...;base64,...'}`, PDFs become `{type:'file', data, mimeType}`, text files inline into the prompt with delimiters.

6. UI

- Glitch mascot photo in chat header, empty-state hero, every assistant avatar, login page, browser favicon

- `ToolCallCard` renders inline for each tool invocation: icon + status (spinner → ✓ / ×) + collapsed arg summary, click to expand the result
- Replicate output cards render images/videos/audio inline with `<img>` / `<video controls>` / `<audio controls>`
- Conversation history sidebar (clicking the History icon in header) — backed by `/api/chat/threads`, locationId from middleware header
- User menu: avatar circle (first letter of email) → click for email + Sign out
- Theme toggle (dark/light)
- Empty-state quick actions: "Show me my pipeline", "Draft a follow-up email", etc.

7. n8n integration

Workflow	id	Purpose
Glitch Magic Link Email	`YmauA2EXSM7kvK7g`	Webhook `/glitch-magic-link` → upsert contact in BizDev → send email via GHL Conversations
`_glitch_busybee_call_sub`	`K2XVksu7VR0mlML5`	Existing alternative path for tool calls (not currently wired into chat — chat uses direct localhost calls)
`_glitch_busybee_list_tools_sub`	`JrbBTyX6FQro3O7b`	Same as above, list flavor

Required n8n cloud variables: - `GHL_AGENCY_PIT` (location custom-values write access) - `GHL_BIZDEV_PIT_local` (BizDev contacts upsert + Conversations email send) - `OPENROUTER_API_KEY`, `OPENROUTER_MGMT_KEY` - `STRIPE_WEBHOOK_SECRET` - `TELEGRAM_BOT_TOKEN`, `TELEGRAM_OWNER_CHAT_ID`

8. Common ops

SSH in:

```
ssh -i ~/.ssh/ghlcrew_ed25519 root@142.93.60.203
```

View logs:

```
pm2 logs glitch-chat --lines 100
pm2 logs glitch-chat --err --lines 60 --nostream
```

Restart with new env:

```
cd /opt/claudebotz/workspace/glitch-chat
set -a && source /opt/claudebotz/.env && set +a
pm2 delete glitch-chat
pm2 start ecosystem.config.js
pm2 save
```

Rebuild after code change:

```
cd /opt/claudebotz/workspace/glitch-chat
npm run build
cp -r .next/static .next/standalone/.next/static
cp -r public .next/standalone/public
pm2 restart glitch-chat
```

(The `cp -r .next/static …` step is required because Next.js standalone output mode does not include static assets in the standalone bundle automatically.)

Verify SSL renewal:

```
certbot certificates
```

Auto-renews via cron (Let's Encrypt cert expires 2026-08-03).

Generate session cookie for testing without going through magic link:

```
node -e "(async()=>{const fs=require('fs'); const env=fs.readFileSync('/opt/claudebotz/.env',
```

9. nginx + DNS + SSL

DNS managed in **GoDaddy**. Records:

```
chat.realtalkha.com  A  142.93.60.203  (TTL 1 hour)
```

nginx config: `/etc/nginx/sites-enabled/glitch-chat` — listens on 443, proxies to `localhost:3001`. Certificate from `/etc/letsencrypt/live/chat.realtalkha.com/`.

The same config also has `glitch.runmycrm.com` listed as an alternate `server_name`, but that subdomain currently points at a different host (Cloudflare → GHL Funnel) and is not used for tenant chat.

10. Files of interest

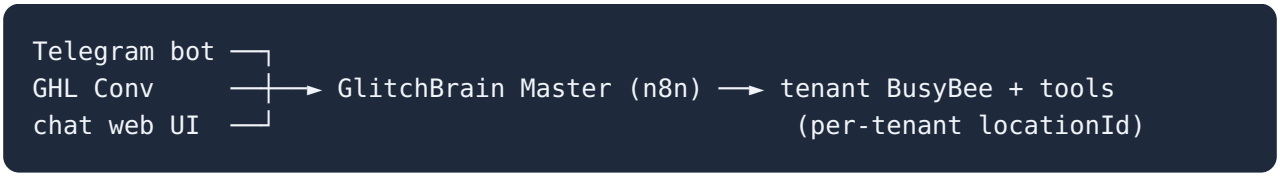
```
/opt/claudebotz/workspace/glitch-chat/                  # repo root
  app/page.tsx                                          # login page
  app/[locationId]/page.tsx                             # chat page
  app/api/chat/route.ts                                 # main chat handler (streamText with
  app/api/upload/route.ts                               # multipart upload handler
  app/api/auth/{request,verify,logout,me}/route.ts      # auth endpoints
  app/login/[token]/route.ts                            # magic-link verify (Safe-Browsing-fr
  app/api/chat/threads/route.ts                         # thread list/get/delete
  app/lib/session.ts                                    # JWT sign/verify (Web Crypto)
  app/lib/tenant-key.ts                                 # OR child key resolver
  app/lib/location-lookup.ts                            # email → locationId (GHL agency)
  app/lib/magic-link-sender.ts                          # n8n webhook + Telegram fallback
  app/lib/busybee.ts                                    # BusyBee MCP client
  app/lib/chat-tools.ts                                 # busybee_call / busybee_list_tools
  app/lib/extra-tools.ts                                # arcee/web_search/apify/replicate
  app/components/ToolCallCard.tsx                       # inline tool render
  app/components/AttachmentBar.tsx                      # paperclip + chips
  middleware.ts                                         # session enforcement + tenant scoping
  ecosystem.config.js                                   # PM2 config + env mapping
```

11. Open follow-ups

Item	Status
Submit Chrome Safe Browsing review for old `/api/auth/verify` URL false positive	Owner action: https://safebrowsing.google.com/safebrowsing/report_error/
Auto-provision BusyBee on tenant signup	Currently manual ~20 sec/tenant via `/glitch/finalize` page. Either modify BusyBee to use shared v2 JWT

Item	Status
	(eliminates per-tenant PIT requirement) or wait for OAuth flow
Replicate polling for video gen	`Prefer: wait=60` is too short for 1–3 min video models. Need fire-and-poll: `replicate_run` returns prediction id, separate `replicate_status(id)` tool checks back
Email pre-fill on login	Add 5 lines to read `?email=` query param from URL and pre-fill the form. Speeds up GHL custom-menu-link UX
Persistent file storage	Currently base64 in-memory per-turn. Uploads not retained across messages
.docx / .xlsx upload support	Need mammoth + xlsx libs. Currently rejects with 415
Model-router classifier	Returns non-JSON in some cases, always falls back to creative. Cosmetic — works

12. Where this fits



The chat web UI bypasses GlitchBrain Master and calls `/api/chat` on glitch-chat directly, but the underlying tools (`busybee_call` , etc.) hit the same per-tenant BusyBee MCP server at `localhost:<port>` on the droplet. Tool routing is the same; the chat is just a different ingress.