

Ship & operate · Runbooks

# Runbook — launch-notice sign-up (Buttondown)

🕒 3 min read · Updated 25 September 2026 · Verified against [f25ab97](#) on 2026-09-25

*Turning on the launch-notice sign-up on the landing page with Buttondown, and the privacy rules it must keep.*

---

<https://docs.gurbanisoul.com/process/runbooks/newsletter/>

---

The site can show a quiet “know when it’s on the App Store” sign-up on the landing (/). It is a **launch notice**, not a marketing newsletter, and it is **env-gated**: it renders nothing (and the built HTML contains no Buttondown reference and no `<form>`) unless the build-time env var `PUBLIC_NEWSLETTER_FORM_URL` is set. This runbook turns it on.

We use **Buttondown**<sup>1</sup> because it supports double opt-in, has no trackers, allows a plain cross-origin form POST, and exports cleanly. The site sends **only** the subscriber’s email address, via `fetch(action, { mode: 'no-cors', body: { email } })` (see `frontend/src/scripts/newsletter.ts`).

## One-time setup (owner)

1. **Create the account** at [buttondown.com](https://buttondown.com) and verify the sending domain (or use Buttondown’s default sending domain to start).
2. **Enable double opt-in** — Settings → Subscribing → require confirmation. This is required: the on-page copy promises a confirmation email, and double opt-in is the GDPR-safe default.
3. **Disable open/click tracking** — Settings → so the promise “No tracking” stays true.
4. **Copy the embed form URL** — the endpoint the embeddable form POSTs to, of the shape `https://buttondown.com/api/emails/embed-subscribe/<username>` (Buttondown → Settings → Embedding shows the exact form action). This value is the env var; it is **not** a secret (it is a public POST endpoint) but it must **never be committed** to the repo — a repo gate (`NewsletterPrivacy`) fails if a Buttondown host literal appears in `frontend/src`.

## Turn it on

5. In the **Vercel project** → **Settings** → **Environment Variables**, add `PUBLIC_NEWSLETTER_FORM_URL` = the embed form URL, for **Production** and **Preview** (leave Development unset so local `npm run dev` keeps the section off). The `PUBLIC_` prefix is required — Astro/Vite only exposes `PUBLIC_`-prefixed vars to the built client.

6. Redeploy (a normal main/integration deploy). The landing then renders the sign-up; the native POST works even with JS disabled (Buttondown shows its own confirmation page).
7. Verify: on the deployed page, the section appears; submitting a test address lands you on Buttondown's confirmation flow and you receive one confirmation email.

## Export, rotation, GDPR

- **Export** subscribers any time from Buttondown → Subscribers → Export (CSV). Keep the export in the owner's storage, not the repo.
- **Rotation:** the form URL is public, not a secret, so there is nothing to rotate. To retire the feature, unset `PUBLIC_NEWSLETTER_FORM_URL` in Vercel and redeploy — the section disappears.
- **GDPR / privacy:** double opt-in is on; only the email address is collected, with explicit consent, for the single stated purpose (a launch notice). Every email carries an unsubscribe link (Buttondown adds it). Honour deletion requests via Buttondown → Subscribers → delete. The app and the rest of the site remain "Data Not Collected"; this optional sign-up is the only place the site collects anything, and only when the visitor types their address and confirms.

## How it fails safe

- Env var unset → no section, no form, no Buttondown host in the HTML (proved by NewsletterPrivacy and frontend/e2e/newsletter.spec.ts).
- Honeypot (hp\_name) filled → the submit is swallowed and the form action blanked (bot signal).
- Offline / rejected fetch → an inline hint pointing at support@gurbanisoul.com; nothing is lost because only the email field is ever sent and the native POST is the no-JS fallback.

## Links

1. <https://buttondown.com/>