ENDURANCE EVENT HUB

User manual

Nightly endurance-event data from RunSignup, ACTIVE and Race Roster — how to browse it, download it, run it and connect it to your directory.

Live site: endurance-event-hub.pages.dev

Prepared by: Poornima Ramakrishnan

Edition: Proof of concept · November 2026

Contents

1. What this is, in one minute
2. A quick tour of the website
3. Finding races
4. Reading a race record
5. Getting the data files
6. How the nightly update works
7. Everyday tasks (no coding)
8. When something goes wrong
9. Connecting your Next.js site
10. Security & credentials
11. Frequently asked questions
12. Glossary

HELP & USER MANUAL

Everything you need, in plain English

No technical background needed. Read it start to finish in about ten minutes, or jump to the part you need.

1. What this is, in one minute

Your race directory needs fresh, trustworthy information about endurance events: dates, distances, registration status, prices, deadlines and capacity. This system collects that information **every night at 2:00 a.m. Eastern** from the official, documented APIs of the race-registration providers, tidies it into one consistent format, and delivers it in three ways:

- **A searchable database** your website can read from directly.
- **Data files** (JSON and CSV) saved to a private cloud folder each night.
- **This website**, which lets you browse and check everything without any tools.

> **The golden rule:** we only use what each provider officially documents and permits. If a provider doesn't offer a piece of information, the record says so plainly — it is never guessed or filled in from somewhere else.

Where each provider stands today

- **RunSignup** — live. Every upcoming US race is collected nightly.
- **ACTIVE** — ready. Needs a free API key; then it switches on automatically.
- **Race Roster** — waiting on partner approval from Race Roster. Nothing is collected until they grant access.

2. A quick tour of the website

- **Overview** — live totals, a map by state, and tonight's countdown.
- **Explorer** — search and filter every race distance; click one for full details.
- **Sources** — what each provider allows, their limits, and a field-by-field availability table.
- **Data model** — what every field means, with a real example record.
- **Pipeline** — history of nightly runs, recent price/status changes, and a runtime calculator.

- **Downloads** — the latest data files.
- **Milestones** — each project requirement and where to see it working.

3. Finding races

1. Open Explorer. You'll see upcoming race distances, soonest first.
2. Type in the search box — a race name, a city or a ZIP code all work.
3. Narrow down with the filters: **state**, **dates**, **status** (for example only “Open”), **distance** (5K, 10K, half, marathon, ultra) or a **maximum price**.
4. Click any card to open its details: prices by registration period, deadlines, capacity, the other distances at the same event, and its change history.
5. Use **Export view** to save what you're looking at as JSON or CSV.

Tip: the page address updates as you filter, so you can bookmark or share a search — for example “open marathons in Texas”.

4. Reading a race record

Each record describes **one distance** at one event. A marathon weekend with a 5K, a half and a full marathon produces three records, because each can sell out or close at different times.

Registration status

STATUS	WHAT IT MEANS
• OPEN	You can register right now.
• OPENS SOON	Registration has a published opening date that hasn't arrived yet.
• FULL	The provider reports the race (or this distance) as sold out.
• CLOSED	The registration deadline has passed or the organizer closed it.
• EXPIRED	The race date itself is in the past.
• ~~CANCELLED~~	The provider marks the event as cancelled.
• UNKNOWN	The provider didn't give enough information to decide — we never guess.

Why some fields are empty

Every price, date and capacity value comes with a small label explaining where it came from — or why it's missing:

LABEL	MEANING
available	The provider returned this value.
not provided	The provider supports the field, but this race left it blank.
not retrieved	Available with an extra per-race lookup. Add the race to the target list to get it.
unavailable at source	The provider's public API doesn't offer this for directory use.

Capacity, explained

Maximum participants (the cap) is kept separate from **registered** and **spots remaining**. A race can have a known cap while the number of people registered stays private — so “cap 500” never implies “500 spots left”.

Countdowns

If a provider says something like “registration closes in 3 days”, the record keeps that original wording *and* converts it to a calendar date based on the exact time the data was collected.

5. Getting the data files

Each nightly run saves fresh files. Open Downloads and choose:

- **events.csv** — opens in Excel or Google Sheets. One row per race distance.
- **events.json** — the complete data for your website or developers.
- **run_report.json** — a summary of what last night's run did.

Opening the CSV in Google Sheets

1. Download `events.csv`.
2. In Google Sheets choose **File** → **Import** → **Upload** and pick the file.
3. Choose “Replace current sheet” and click **Import data**.

Every file also has a SHA-256 checksum on the Downloads page, so you can prove a file hasn't been altered.

6. How the nightly update works

- **2:00 a.m. Eastern** — the run starts automatically. It adjusts for daylight-saving time on its own.
- **Plan** — it works out how many requests it will need and how long that should take, and records the estimate.
- **Collect, politely** — one request at a time, at least **60 seconds apart**. If a provider asks it to slow down, it waits longer.
- **Tidy** — every race distance becomes one consistent record. Problems with individual records are logged without stopping the run.
- **Deliver** — the database is updated, price and status changes are kept as history, and new files are saved to the private cloud folder.
- **Report** — the run is marked *succeeded*, *degraded* (finished, but some things failed) or *failed*, and you're alerted if it wasn't a success.

7. Everyday tasks (no coding)

All settings live in your private GitHub repository. You only need a web browser.

Run the update right now

- Open the repository on GitHub and click the **Actions** tab.
- Choose **Nightly pipeline** on the left, then **Run workflow**.
- Pick *staging* or *production* and click the green **Run workflow** button. You'll see progress live.

Add or remove races in the target list

- In the repository, open `config/targets.toml` and click the pencil (edit) icon.
- Under `targets = [ … ]`, add the RunSignup race ID (the number in the race's RunSignup address), one per line, followed by a comma.
- Click **Commit changes**. The next run uses the new list — no code changes needed.

Switch on ACTIVE

- Register for a free key at developer.active.com.
- In GitHub go to **Settings** → **Secrets and variables** → **Actions** → **New repository secret**.

. Name it `ACTIVE_API_KEY`, paste the key, and save. ACTIVE races appear after the next run.

Change the request spacing

In `config/targets.toml`, change `request_interval_sec = 60`. Values below 60 are not recommended and the nightly schedule always enforces at least 60 seconds.

8. When something goes wrong

You don't need to watch anything overnight. If a run fails or is degraded:

- A **GitHub issue** is opened automatically with the reason — GitHub emails it to you.
- If you've connected Slack or Microsoft Teams, a message is posted there too.
- The previous night's data stays in place, so your directory never goes blank.

What to do

. Read the reason in the issue — for example “RunSignup returned errors for 3 requests”.
. Most problems are temporary on the provider's side. Simply wait for the next night, or run it again manually (section 7).
. If the same error repeats for two nights, forward the issue to your developer — the run report contains everything needed to diagnose it.

9. Connecting your Next.js site

For your developer: the Data model page has the JSON Schema, generated TypeScript types and a ready-to-paste Next.js App Router example. Two integration options:

- **Files** — read tonight's `events.json` from the private bucket using a short-lived signed URL, and revalidate hourly.
- **Database** — query the directory tables or the provided search functions (status, state, distance, dates, price, ZIP) directly from server components.

Use each record's `registration.url` as the “Register” button link. Records keep the same `record_id` from night to night, so favourites and bookmarks stay valid.

10. Security & credentials

- API keys and passwords are stored only as encrypted secrets in GitHub and Cloudflare — never in the code, the files, or this website.
- The cloud folder holding the exports is private.
- The database is locked down: the public site can only call specific read-only search functions.
- Each provider's terms were reviewed, and only their documented APIs are used.

11. Frequently asked questions

Why do some races show “Price not listed”?

Prices come from the registration periods the organizer sets up. If no period is active right now (for example, registration hasn't opened yet), there is no current price — but upcoming tier prices still appear in the record's registration periods.

Why is “spots remaining” usually empty?

RunSignup only shares registration counts with each race's own director. Your directory isn't authorized for that, so the field is honestly marked as unavailable instead of being estimated.

Why aren't ACTIVE races showing yet?

ACTIVE requires a free API key. The connector is already built and tested — once a key is added (see “Everyday tasks”), ACTIVE races appear after the next nightly run, with the attribution ACTIVE requires.

Why aren't Race Roster races showing?

Race Roster only issues API access to approved partners, and its terms forbid other access methods. The pipeline makes no requests and reports this every night until Race Roster grants access.

Can the update run more than once a night?

Yes — anyone with access can start a run from GitHub with one click. Scheduled runs stay at 2:00 a.m. Eastern so the providers are contacted at a quiet time.

How long does a nightly run take?

About one minute per request. The current setup makes roughly 60 requests, so about an hour. The Pipeline page has a calculator that shows the effect of adding more races.

Is my data public?

The exported files live in a private storage bucket. This demo website streams them through a secure proxy for convenience; in production you decide who can access them.

What happens if a provider is down?

The run waits and retries politely, then pauses that provider after three failures in a row so others still update. You get an alert, and yesterday's data stays in place.

12. Glossary

API	A documented, official way for one computer system to ask another for data.
Event	A race page on a provider, e.g. “City Marathon Weekend”.
Sub-event / distance	One race within an event, e.g. the 5K or the half marathon. Each gets its own record.
Record	One row of data describing a single race distance.
Nightly run	The automatic update that starts at 2:00 a.m. Eastern every night.
Target list	Specific races you want looked up in extra detail each night (adds capacity data).
Rate limit	The maximum speed a provider allows. We stay far below it: one request per minute.
JSON / CSV	File formats. JSON is for software; CSV opens in Excel or Google Sheets.
Schema	The agreed shape of a record — which fields exist and what they mean.
Secret	A password or key. Stored only in secure settings, never in code or this website.

Questions not covered here? Use the feedback button on any page — it reaches us directly.