EcomLabs

STEP-BY-STEP GUIDE

Zero-result searches in your store, tracked in GA4

The Shopify custom pixel that sends searches with no products to GA4, the dimension and exploration for the weekly report, a Node script that merges and ranks the terms, and the template for deciding what to fix every Monday.

FOR	Shopify stores that measure with GA4 and want to know what customers search for and do not find
TESTED WITH	Node 22.22.3 · the pixel in node:vm with a fake analytics object · illustrative CSVs in GA4 export format
UPDATED	November 19, 2026
DOCS	shopify.dev/docs/api/web-pixels-api/standard-events/search_submitted
ARTICLE	ecomlabs.dev/g/busqueda-en
AUTHOR	Luis Morales · @luismoralesdev

Contents

Step 01	The event: a custom pixel
Step 02	The report in GA4
Step 03	The top-ten script
Every Monday	Weekly template

STEP 01

The event: a custom pixel

`view_search_results` does not say how many results there were; Shopify's `search_submitted` does. In Shopify: *Settings > Customer events > Add custom pixel*. Paste the code, replace `G-XXXXXXXXXX` with your measurement ID, save and click *Connect*.

```
// Shopify admin > Settings > Customer events > Add custom pixel
const GA_ID = "G-XXXXXXXXXX"; // your GA4 measurement ID

const script = document.createElement("script");
script.setAttribute("src", `https://www.googletagmanager.com/gtag/js?id=${GA_ID}`);
script.setAttribute("async", "");
document.head.appendChild(script);

window.dataLayer = window.dataLayer || [];
function gtag() { dataLayer.push(arguments); }
gtag("js", new Date());
gtag("config", GA_ID, { send_page_view: false });

analytics.subscribe("search_submitted", (event) => {
  const result = event.data?.searchResult;
  if (!result || result.productVariants.length > 0) return; // only searches with zero products
  let term = (result.query || "").trim().toLowerCase().replace(/\s+/g, " ");
  if (!term) return;
  if (term.includes("@") || term.replace(/\D/g, "").length >= 7) term = "(redacted)"; // no emails or phone
numbers
  const { origin, pathname } = event.context.window.location;
  gtag("event", "search_no_results", {
    zero_result_term: term.slice(0, 100),
    page_location: origin + pathname, // the store URL without ?q=, which holds the raw term
  });
});
```

- It only sends with zero products. The term is cleaned, cut to 100 characters (the GA4 maximum) and sent as `(redacted)` if it has `@` or seven digits: Google asks for no personal data.
- `page_location` has no `?q=`, and `send_page_view: false` prevents a second `page_view`. The pixel only runs with the permissions in its configuration (Marketing and Analytics by default): match them to your banner.

Test it with the Pixel Helper and DebugView: a search that exists sends nothing; `zzzz test` sends one event.

STEP 02

The report in GA4

The dimension. *Admin > Data display > Custom definitions > Create custom dimensions* (Editor role required):

- *Dimension name*: `Zero result term` (no hyphens: GA4 rejects them).

- *Scope*: *Event*. *Event parameter*: `zero_result_term`. Neither can be changed later.
- It takes 24 to 48 hours to show up. Do not register `search_term`: it is already predefined.

The exploration. Build it once, open it every Monday:

1. *Explore*, blank exploration (free-form table).
2. Under *Variables*, import *Zero result term*, *Event name* and the *Event count* metric.
3. *Zero result term* in rows, *Event count* in values.
4. Filter: *Event name exactly matches* `search_no_results`.
5. Range: the closed week, Monday to Sunday; processing takes up to 48 hours.
6. *Export data > CSV*: it includes every row, not only the visible ones.

STEP 03

The top-ten script

GA4 groups by exact text (`café` and `cafe` are two rows). `top-terms.mjs` merges them and prints the top ten; tested with Node 22:

```
// Usage: node top-terms.mjs ga4-export.csv [catalog.txt]
// ga4-export.csv: the exploration exported as CSV (term column, then Event count).
// catalog.txt (optional): words from your product titles, one per line.
import { readFileSync } from "node:fs";

const [csvFile, catalogFile] = process.argv.slice(2);

// One CSV line into cells; handles "quoted, cells" and "" inside quotes.
function cells(line) {
  const out = [];
  let cell = "", quoted = false;
  for (let i = 0; i < line.length; i++) {
    const c = line[i];
    if (quoted && c === '"' && line[i + 1] === '"') { cell += '"'; i++; }
    else if (c === '"') quoted = !quoted;
    else if (c === "," && !quoted) { out.push(cell); cell = ""; }
    else cell += c;
  }
  return [...out, cell];
}

// "Tenis  Blancos" and "tenis blancos" count as the same search.
const norm = (s) => s.normalize("NFD").replace(/\p{M}/gu, "").toLowerCase().replace(/\s+/g, " ").trim();

// Letters you have to change, add or remove to turn a into b.
function distance(a, b) {
  let prev = Array.from({ length: b.length + 1 }, (_, j) => j);
  for (let i = 1; i <= a.length; i++) {
    const row = [i];
    for (let j = 1; j <= b.length; j++) {
      row[j] = Math.min(prev[j] + 1, row[j - 1] + 1, prev[j - 1] + (a[i - 1] === b[j - 1] ? 0 : 1));
```

```
    }
    prev = row;
  }
  return prev[b.length];
}

// Skip the # lines GA4 puts before the header, the header itself and any totals row.
const lines = readFileSync(csvFile, "utf8").replace(/^/, "").split(/\r?\n/);
const rows = lines.filter((l) => l.trim() && !l.startsWith("#")).slice(1).map(cells);
const totals = new Map();
for (const [term, count] of rows) {
  const n = Number(String(count).replace(/,/g, ""));
  const key = norm(term ?? "");
  if (!key || !Number.isFinite(n) || /^(grand )?totals?$/.test(key)) continue;
  totals.set(key, (totals.get(key) ?? 0) + n);
}

const catalog = catalogFile ? readFileSync(catalogFile, "utf8").split(/\r?\n/).map(norm).filter(Boolean) : [];
function hint(term) {
  for (const word of term.split(/[^\p{L}\p{N}]+/u)) {
    if (word.length < 5 || catalog.includes(word)) continue; // short words give false hints
    const near = catalog.find((w) => distance(word, w) <= 2);
    if (near) return `typo? ${word} -> ${near}`;
  }
  return "";
}

const top = [...totals].sort((a, b) => b[1] - a[1] || a[0].localeCompare(b[0])).slice(0, 10);
console.log("  #  searches  term".padEnd(39) + "hint");
top.forEach(([term, n], i) => {
  console.log(`${String(i + 1).padStart(3)}  ${String(n).padStart(8)}  ${term.padEnd(24)}${hint(term)}`);
});
```

Run `node top-terms.mjs ga4-export.csv catalog.txt`. `catalog.txt`, optional, holds one word per line from your titles and tags; with it, the script flags likely typos.

EVERY MONDAY

Weekly template

WEEK	TERM	SEARCHES	CAUSE	FIX	GONE?
Nov 9–15 (illustrative)	leather neklase	22	Typo	Tag "neklase" on necklaces	Nov 24

1. Export the closed week, run the script, paste the ten terms and give each a cause and a fix: synonym or typo (the word as a tag), field search does not read (move it to title, type or tags), missing product (log it) or noise.
2. Fix the most searched, cheapest ones first. Next Monday, mark which fixed terms no longer appear: if one is still there, the fix did not work.