

Your product. A clearer story.

Presagio Help Center · Integration candidate · Updated 11 September 2026

Updated integration candidate — requires matching SDK/runtime release; existing installations remain manual until enabled. This edition documents the Overview and Attribution candidates alongside the existing private-beta workflows. It does not confirm that those candidate features are deployed at presagio.in. Use the new options only with the matching release supplied by the Presagio team.

A practical guide for founders, administrators and the developers connecting their product. Start with the outcome you care about, then follow the evidence all the way to an individual journey.

Which experience am I looking at? The workspace at presagio.in/app uses your stored events. The [interactive demo](#) uses sample data. Screenshots show the actual Presagio interface with illustrative data. Your workspace uses your own collected events, including the same data-driven journey waves.

Inside the guide

1. What Presagio gives you

2. Start with one business question

3. Get access and create your workspace

4. Understand your workspace and dashboard

5. Configure a website source and collection key

6. Install the SDK: configuration checklist

7. Frontend integration: React, routed apps and website builders

8. Event naming, properties and SDK methods

9. Verify your first three events

10. Define meaningful events on the fly

11. Build and read your founder journey

12. How counting works

13. Privacy, identity and delivery behavior

14. Troubleshooting by symptom

15. Practical beta limits

16. A founder's operating routine

17. First-launch checklist

18. Glossary and next steps

1. What Presagio gives you

Presagio helps answer three practical questions: **Are people reaching value? Where does progress slow down? Which events support that conclusion?** The private beta connects deliberate website events to reusable definitions and ordered journeys. This integration candidate adds opt-in page views and a traffic Overview so you can understand arrivals before defining a custom journey.

You choose the moments that matter. For a SaaS product, that might be trial started → first useful result → subscription. For a collaboration tool, it might be workspace created → teammate invited → shared project completed. Track observable actions, not assumptions about satisfaction or intent.

Existing private-beta capability	What it helps you do
Secure Google sign-in and approved workspace access	Keep product data scoped to the people permitted to see it.
Projects, Production/Test environments and website sources	Separate products and testing activity before interpreting results.
Consent-aware browser SDK and scoped collection keys	Send deliberate events with explicit permission and exact website origins.
Durable event collection, retry deduplication and metering	Handle lost responses without counting the same event twice.
Live event inspector	Check what was actually stored before building reports.
Event definitions, previews and immutable versions	Turn raw signals into meaningful milestones, including rules created after capture.
Ordered journeys with clickable evidence	Measure progression and inspect the events behind a milestone.
Pending versus elapsed conversion windows	Avoid calling someone a drop-off while they still have time to convert.

Additional capabilities in this integration candidate: Overview shows browser visitors, observed sessions, page views and context rankings. Attribution shows first-touch, last-touch and last-non-direct conversion evidence. Both require the coordinated SDK/runtime release; neither is proof of verified revenue or causal impact.

What makes the approach distinctive: Presagio combines an outcome-oriented visual experience with traceable counting rules. The cinematic waves make a journey easy to scan; the milestone counts, pinned definitions and evidence timeline explain it. This describes the product's design, not a claim that no other analytics platform offers similar capabilities.

Not included in this beta: native iOS/Android SDKs, cross-device identity stitching, server-side identify APIs, automatic click capture, session replay, heatmaps, AI recommendations, email alerts, revenue attribution, advanced custom reports, self-service team invitations or subscription checkout. Some appear as concepts or planned features on the public site. They are not enabled workspace capabilities.

2. Start with one business question

A useful first journey is small enough to explain in a sentence. For example: “Of the visitors who start a trial, how many reach their first useful result and subscribe within seven days?”

1. Choose two to five observable milestones.
2. Give each underlying event one consistent name.
3. Decide which properties are necessary to distinguish meaningful activity.
4. Choose a conversion window that matches your product's buying cycle.
5. Test with synthetic activity before enabling your real production source.

Business moment	Raw event example	Deliberate properties	When to send
Trial begins	trial_started	plan: "founder"	After trial creation succeeds.
First useful result	value_reached	feature: "first_report"	After the useful result is successfully available.
Subscription confirmation	subscribed	plan: "founder"	After the frontend receives a successful confirmation.

A browser subscription event is an analytics signal. It is not an independently verified billing ledger. Don't infer revenue or successful payment from a button click or from the event name alone.

3. Get access and create your workspace

Open [the application](#) and choose **Continue securely**. Complete Google sign-in using the account approved for your beta invitation. Google sign-in and Presagio admission are separate steps: successful authentication can still leave your account awaiting approval.

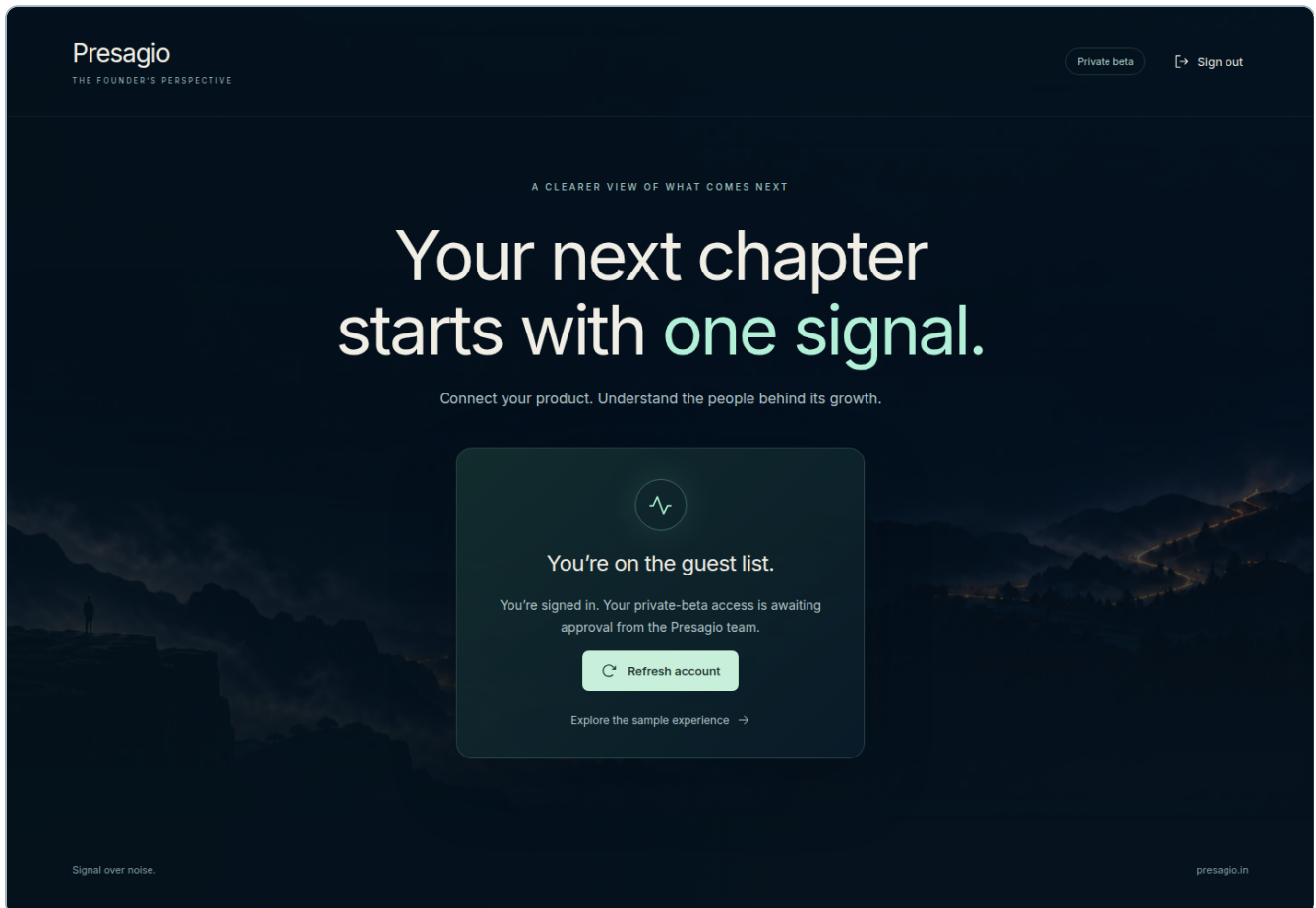


Figure 1. The real application's pending-access screen. “You’re on the guest list” means you have signed in; it does not yet grant workspace access.

If you see this screen, ask the Presagio team to approve your account, then choose **Refresh account**. Google is currently configured for a controlled test audience, so a new participant may also need to be added to the Google test-user list. Do not create repeated accounts to get around admission.

Once approved:

1. Enter your company or workspace name and choose **Create**.
2. Name your product under **What are you building?** and choose **Create**.
3. Select the product and its **Test** environment for initial integration work. Projects receive separate Test and Production environments.

4. Open **Connection & events** and add a website source, such as “Marketing website” or “Product app”. In the candidate, Overview also offers **Connect your website** when no source exists.
5. Ask the Presagio team to activate collection for the intended source. A source can be saved before it is ready to collect.

Two different gates: account approval lets you use the workspace. Source activation lets a website collect events. Creating a key does not activate a source.

4. Understand your workspace and dashboard

This candidate has five main tabs. **Overview** is the default traffic view; **Journeys** measures ordered business outcomes. **Connection & events** is the installation and verification area. **Attribution** is a separate candidate report, and **Event definitions** manages milestone rules. There is no live revenue, retention or AI executive dashboard in this release candidate.

Control or tab	What it changes	What to check
Workspace	Organization whose products you can access	Confirm the company before configuring anything.
Product	Project containing environments, definitions and journeys	Keep different products separate.
Environment	Test or Production activity used by reports	Test results do not establish production activity.
Website source	Source selected for key and connection setup	The inspector and journey reports are environment-scoped, not narrowed by this setup selector.
Overview — candidate	Browser visitors, observed sessions, page views and context rankings	Select all website sources or one source, period and display timezone.
Attribution — candidate	Observed campaign/referral credit for a conversion definition	Select one source, definition, period and model; inspect the supporting touches.
Connection & events	Source setup, keys, snippet and event inspector	Use this first when events are missing.
Event definitions	Rules that recognize business milestones	Preview a rule before relying on it.
Journeys	Ordered progression and evidence	Select a journey and revision, then reveal a period.

Overview fetches when its filters change; **Refresh overview** evaluates a fresh window. Other report controls fetch current state when requested. This is not a continuously streaming dashboard. Returning to a tab may refresh your account and reset the selected workspace or source. Recheck those selectors before taking action. This is a known beta usability limitation.

Read the Overview — integration candidate

After installing the matching candidate SDK and enabling web mode, select **Overview**. Choose the intended workspace, product and environment, then:

1. Choose **All website sources** or a single **Website source**. This filter belongs to Overview; the source selected in Connection & events does not silently narrow it.
2. Choose **Last 24 hours** or **Last 7 days**. These are rolling 24-hour and 168-hour windows ending at the query time, not complete calendar days.
3. Choose **UTC** or your browser's named **Local** timezone. The report displays the actual timezone and window. A browser already using UTC has no duplicate Local option.
4. Read Browser visitors, Sessions observed and Page views together. Inspect the dated trend and **View chart data** for exact bucket boundaries and counts.
5. Compare Pages, Countries, Arrivals, Devices, Browsers, Operating systems and Languages. Every ranking counts **page views**, not people or conversions.
6. Use **Refresh overview** for current stored evidence. Changing scope, period or timezone clears the old result while the new query runs; a failed query never substitutes sample activity.



Overview candidate. The actual application component with explicitly labeled local sample data, captured for this guide. These figures are not production traffic and do not establish a deployed capability.

Browser visitors counts distinct browser IDs within each source, over the entire period. It does not identify people or merge devices and sources. **Sessions observed** counts source-scoped browser/session combinations with a page view in the window; these are not necessarily sessions that began in that window. Neither total is a sum of daily unique counts. **Page views** counts stored qualifying `page_view` version-1 events, including older manual page views without context. General custom events do not become page views because they include a page name.

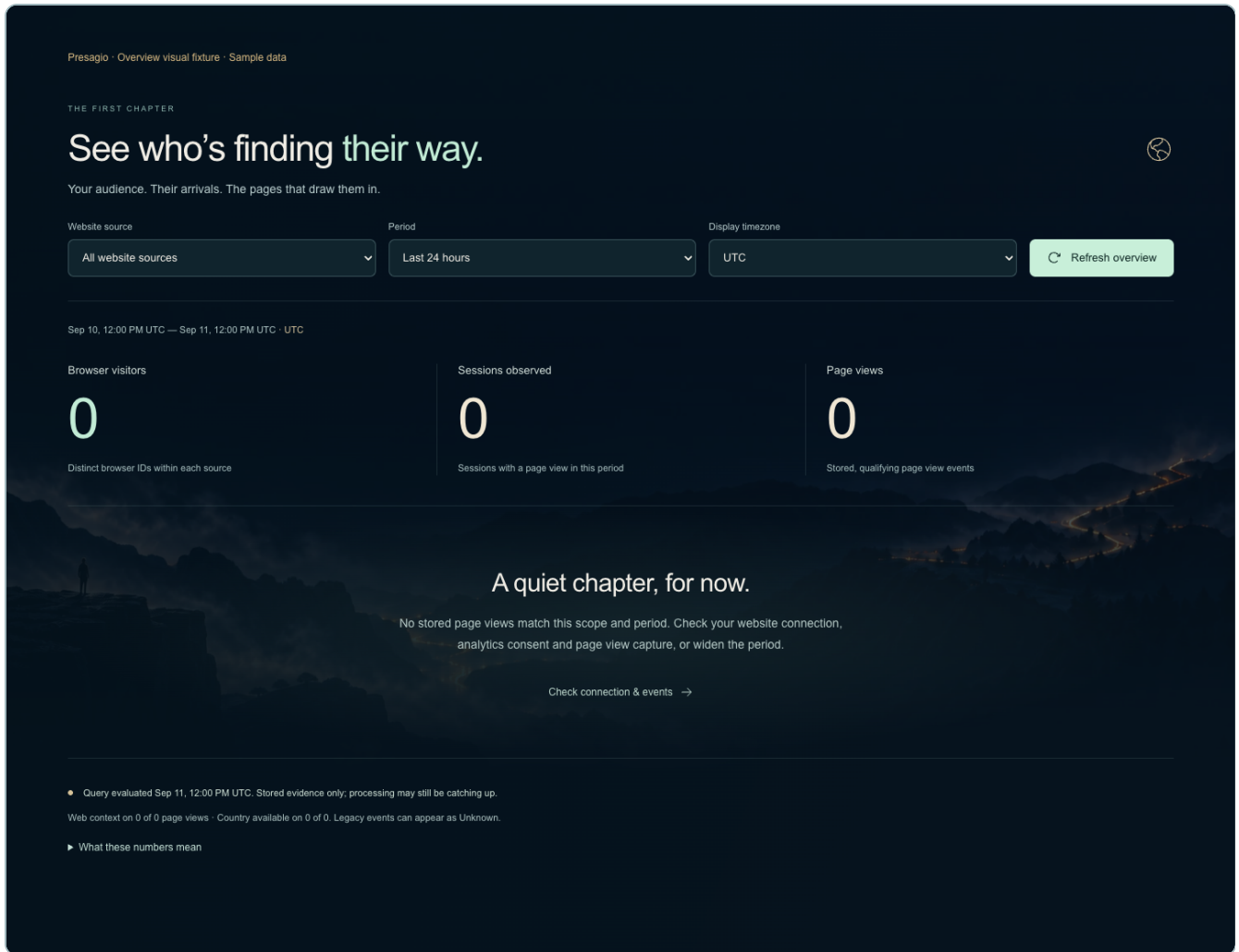
For the 24-hour view, chart buckets represent elapsed UTC hours, with labels displayed in your selected timezone. The seven-day view uses calendar-day boundaries in that timezone; daylight-saving transitions can make a local day 23 or 25 hours long. The rolling window clips its first and last buckets. Use the table's exact timestamps instead of assuming every displayed bucket covers a complete hour or day. Changing timezone changes grouping and labels, not the intended elapsed length of the reporting window; queries run at a new current time.

The header totals are exact within the report cap. Each ranking shows up to 20 groups, with omitted page views disclosed separately when applicable. An **OVERVIEW_LIMIT** error means more than 10,000 candidates matched: choose Last 24 hours or one website source. No partial headline count is presented as complete.

Countries use approximate ISO two-letter location codes, or **Unknown**, from a trusted country lookup. They are not nationality or an exact location. IP geolocation by [DB-IP](#), using Country Lite under [CC BY 4.0](#). A missing or unavailable lookup remains Unknown; raw IP is not included in the event or stored reporting context. Missing legacy/browser context also remains Unknown. The coverage line states how many page views contain web context and an available country.

Arrivals groups observed campaign/referral context and internal navigation. It is not a conversion-credit model. An empty referrer is not proof of direct traffic. The separate Attribution tab is described below.

The evaluated-at time indicates when the query ran, not that collection has caught up to that instant. Processing delay, late activity, retention and deletion can change results. Bots may be included; browser counts are not verified humans.



Overview empty state, captured from an isolated local fixture. No page views in the selected window is different from a disconnected source or a failed query. No illustrative values are inserted into the real report.

Attribution — separate integration candidate

With the matching release, configure `attribution: { enabled: true }` in the same SDK instance and call `captureArrival()` after analytics consent for an eligible entry. This is separate from web page-view context: web mode alone does not create attribution touches. Continue sending business events only from the successful business action, never merely because consent was granted.

```
// Add alongside web in your createPresagio configuration:
// attribution: { enabled: true }

function onAnalyticsConsentGranted() {
  analytics.setConsent('granted');
  analytics.captureArrival();
}

function onSubscriptionConfirmed() {
  analytics.track('subscribed', { plan: 'founder' });
}
```

Use this callback only after enabling attribution. The default arrival is remembered once per SDK instance. A later distinct external arrival in a long-lived application requires a fresh stable arrival ID; reuse it when retrying that same arrival. Do not call every internal route change or session timeout a new arrival. Never infer known direct traffic merely from an empty browser referrer. Keep campaign tags free of personal information.

Open **Attribution**, choose one website source, a published conversion definition and Last 24 hours or Last 7 days, then choose **Reveal attribution**. Compare **First touch**, **Last touch** and **Last non-direct**, and expand conversion evidence. Models use a seven-day lookback and source-local browser identity. Counts represent distinct matching conversion events, not customers or verified revenue. Capture history is unknown and results remain provisional. Touch events use the ordinary queue, retention, deletion and event-usage allowance; they are not free invisible tracking. This candidate cannot recover campaigns that were never captured.

Roles and administration

Owners and admins can configure the products they are authorized to manage, create or revoke keys, and publish definitions and journeys. Viewers have read-only access. Workspace membership and private-beta approval are operator-managed in this release. There is no self-service team invitation, ownership-transfer or billing screen.

Sign out using the header control. Sessions expire after 12 hours; if the app asks you to sign in again, do so before continuing. Losing access or changing membership can remove a previously visible workspace.

5. Configure a website source and collection key

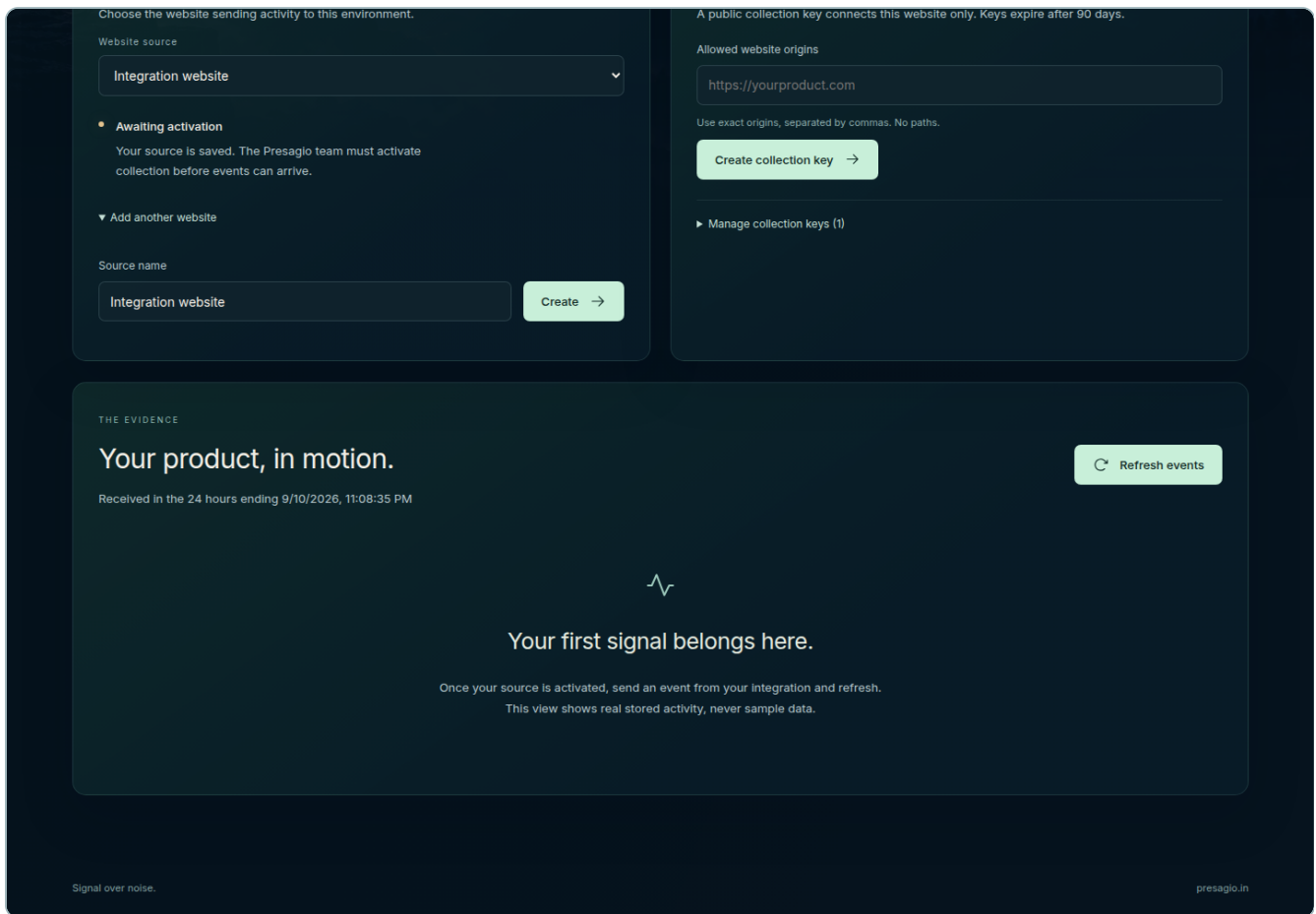


Figure 2. The source setup and empty inspector during initial verification. The “Awaiting activation” message is expected until the team enables collection.

In **Connection & events**, select the correct website source. In **Allowed website origins**, enter the exact origins allowed to send activity, separated by commas.

- Valid production example: `https://app.example.com` .
- Another host, such as `https://www.example.com` , must be listed separately.
- An origin contains the scheme, host and optional port. Do not include a path, query string or wildcard.
- Loopback testing can use an exact origin such as `http://localhost:5173` ; `localhost` and `127.0.0.1` are different origins.
- Production websites require HTTPS. Use a separate Test source and key for development.

Choose **Create collection key**. Copy the installation snippet while the key is visible, then choose **I've saved it — hide key**. The UI creates keys with a 90-day lifetime and shows the raw value only once. If it is lost, create a replacement and revoke the unused key.

A collection key is a **public write identifier** intended for frontend code. It is not a login credential, does not grant analytics read access and cannot approve users. Origin restrictions are one protection; they do not make a browser key secret or prove the sender's identity.

To rotate a key, create its replacement for the same source, update the frontend, verify delivery and then revoke the old key under **Manage collection keys**. Revocation is permanent. Rotating within the same source preserves that source's browser identity continuity.

6. Install the SDK: configuration checklist

You configure the tracking code in **your own product's frontend**, not inside the Presagio dashboard. Use the exact source ID and public key from the generated snippet.

Setting	Example	Required action
SDK module	<code>https://presagio.in/sdk/presagio.mjs</code>	Import in the browser, or vendor it into your frontend build.
Classic script	<code>https://presagio.in/sdk/presagio.js</code>	Alternative exposing <code>window.Presagio</code> . Choose one installation style.
<code>endpoint</code>	<code>https://presagio.in/v1/events/batch</code>	Use the collection URL from your snippet.
<code>key</code>	<code>YOUR_PUBLIC_COLLECTION_KEY</code>	Replace this placeholder with the generated write key.
<code>sourceId</code>	<code>YOUR_SOURCE_UUID</code>	Replace with the source's actual UUID.
Consent callbacks	Granted / denied	Wire to your consent manager; initialization alone sends nothing.
<code>web — candidate opt-in</code>	<code>{ enabled: true, pageViews: 'auto', spa: true }</code>	Enable consent-aware page views only with the matching SDK/runtime release.
Business event calls	<code>track()</code>	Send after the actual product action succeeds, not after consent alone.
Manual page views	<code>page()</code>	Use deliberately in manual mode; automatic mode already captures the initial page.
Content Security Policy	Script and connection permissions	Allow the SDK location and collection endpoint in your site's policy.

The examples below are for a new setup on the matching integration candidate. Do not assume the currently hosted file supports candidate options before the coordinated release. Existing installations remain manual unless web mode is explicitly enabled.

There is no published npm package in this beta. Do not invent an `npm install presagio` dependency. The hosted module is self-contained. If your bundler cannot import HTTPS modules, vendor the downloaded module under a reviewed, versioned filename in your application and import it locally.

Option A: browser module

Save this as a browser module such as `analytics.js` and load it with `<script type="module" src="/analytics.js"></script>` in your product. Replace both placeholders first.

```
import { createPresagio } from 'https://presagio.in/sdk/presagio.mjs';

export const analytics = createPresagio({
  endpoint: 'https://presagio.in/v1/events/batch',
  key: 'YOUR_PUBLIC_COLLECTION_KEY',
  sourceId: 'YOUR_SOURCE_UUID',
  web: { enabled: true, pageViews: 'auto', spa: true },
  onDiagnostic: ({ code, count }) => {
    // Optional development diagnostics. Do not log customer payloads.
    console.debug('Presagio', code, count);
  },
});

// Connect these functions to your actual consent manager.
export function onAnalyticsConsentGranted() {
  analytics.setConsent('granted');
}

export function onAnalyticsConsentWithdrawn() {
  analytics.setConsent('denied');
}

export function onTrialCreatedSuccessfully() {
  analytics.track('trial_started', { plan: 'founder' });
}
```

These functions are integration examples, not automatic browser hooks. Your application must call them at the appropriate time. If valid permission already exists on a new page load, your consent manager must communicate that grant to this SDK instance again.

Option B: classic script

Load the SDK before your own initialization script:

```
<script defer src="https://presagio.in/sdk/presagio.js"></script>
<script defer src="/analytics-init.js"></script>
```

In your own `analytics-init.js` :

```
const analytics = window.Presagio.createPresagio({
  endpoint: 'https://presagio.in/v1/events/batch',
  key: 'YOUR_PUBLIC_COLLECTION_KEY',
  sourceId: 'YOUR_SOURCE_UUID',
  web: { enabled: true, pageViews: 'auto', spa: true },
});

// Wire this to your consent manager's positive analytics decision.
function onAnalyticsConsentGranted() {
  analytics.setConsent('granted');
}

function onAnalyticsConsentWithdrawn() {
  analytics.setConsent('denied');
}

// Call after the product action succeeds, and only after permission.
function onFirstReportReady() {
  analytics.track('value_reached', { feature: 'first_report' });
}
```

Use one client per source in a page. Installing both options creates duplicate instrumentation risk. Two separate calls to `track()` create two distinct events even if the names and properties are identical; retry deduplication does not correct duplicate tracking calls.

7. Frontend integration: React, routed apps and website builders

React or another client-rendered app

Initialize once at the application boundary. Keep the client in a shared module or provider, not in every button or page component. For a React app, the following client wrapper is one option after you vendor the SDK into

`src/vendor/presagio.mjs` :

```
import { createPresagio } from './vendor/presagio.mjs';

let client;
export function getAnalytics() {
  if (typeof window === 'undefined') return null;
  client ??= createPresagio({
    endpoint: 'https://presagio.in/v1/events/batch',
    key: 'YOUR_PUBLIC_COLLECTION_KEY',
    sourceId: 'YOUR_SOURCE_UUID',
    web: { enabled: true, pageViews: 'auto', spa: true },
  });
  return client;
}
```

Then call `getAnalytics()?.setConsent('granted')` from your consent integration and `getAnalytics()?.track(...)` after a successful user action. Initialize only in a browser context. For server-rendered applications, keep SDK initialization and event calls inside client-side code; this SDK is not a server ingestion client.

Do not send business events during React render. An effect may run again in development or after remounting. Prefer the confirmed action handler, and ensure your application invokes it once for the business action. If a component owns its own SDK instance, destroy that instance during cleanup and recreate it on the next mount; never reuse a destroyed instance.

Single-page application routes

With candidate web mode enabled and `pageViews: 'auto'`, granting consent captures the current page once. With `spa: true`, pathname changes via `pushState`, `replaceState` and `popstate` capture the next page view. Do not add a duplicate initial `page()` call. Query-only and hash-only changes do not create page views. Pre-consent navigation is not replayed. A browser back/forward-cache restoration does not automatically create another page view on `pageshow`.

Choose `pageViews: 'manual'` in web mode if your router should call `analytics.page()` or `analytics.page('Pricing')` explicitly. Manual and automatic calls share a successfully queued route-visit marker in web mode. Existing integrations without `web.enabled` retain their original manual behavior. Custom `track()` events do not automatically receive web context.

Minimize page context before capture

Web mode reads browser context only after consent and attaches optional version-1 fields: minimized path, external referrer domain, device/browser/operating-system family, language, timezone and observed acquisition classification. It never sends the complete URL, query, fragment, raw user agent, IP or DOM contents as event context. There is no automatic click, form-value, network-request or replay capture.

Path masking catches obvious numeric IDs, UUIDs, email-like segments and long tokens. **It is heuristic minimization, not complete personal-data redaction.** Provide route templates and exclude sensitive pages where needed:

```
// Pass this as the web option in createPresagio({ ..., web }).
const web = {
  enabled: true,
  pageViews: 'auto',
  spa: true,
  mapPath: path => path.replace(/^\/customers\[^\]+\//, '/customers/:customer'),
  excludePath: path => path.startsWith('/private/')
};
```

These synchronous callbacks receive the pathname after consent; do not log it. Returning `null` from `mapPath` also excludes a page. Keep manually supplied page names and custom properties safe. Withdrawal and destruction remove navigation listeners and clear route state; consent withdrawal also follows the normal queue and identity cleanup rules.

WordPress, Webflow and similar websites

Use the classic script through the site's supported custom-code mechanism and put the initialization in your own script. Connect consent before tracking; wire product events to the builder's actual success callbacks or an application event you control. Presagio does not currently provide native plugins or a visual click-selection tool for these platforms.

Tag managers

A custom HTML/script installation may work if your tag manager and site's policy permit it, but there is no official Presagio tag-manager template in this beta. Configure consent gating, one-time initialization and action triggers explicitly. Test that the same trigger does not fire twice.

Frontend settings to verify

1. Choose the right Test or Production source and key at build or runtime configuration time.
2. Permit scripts from `https://presagio.in` if using hosted files. If vendored, permit your own asset origin.
3. Permit connections to `https://presagio.in` in `connect-src` if your site uses Content Security Policy. Extend your existing policy; do not replace it wholesale with a permissive one.
4. Use exact allowed origins in Presagio, including development ports.
5. Confirm analytics permission reaches the client on every initialization.
6. Ensure events are wired to success, not to every render or attempted click.
7. Check blockers, offline mode and network errors in your testing browser.
8. Keep OAuth client secrets, administrator tokens and database credentials out of the frontend. None are needed for tracking.

8. Event naming, properties and SDK methods

Use stable names such as `trial_started`, `project_created` and `value_reached`. Avoid names containing emails, names, order identifiers or variable text. Put only necessary, non-sensitive dimensions in properties.

```
analytics.track('value_reached', {
  feature: 'first_report',
  plan: 'founder',
  report_count: 1,
}, { schemaVersion: 1 });

// Only for deliberately manual page-view capture:
// analytics.page('Pricing', { experiment: 'layout_b' });
```

`page()` sends a `page_view` event and can include a deliberately chosen `page_name`. Without web mode, its behavior remains manual and does not read browser context automatically. With web mode enabled, page views attach the minimized context described above. Automatic mode already captures the current page after consent; the business-event example does not need another page call.

Method	Meaning
<code>setConsent('granted')</code>	Allow this instance to establish identity and send events.
<code>setConsent('denied')</code>	Clear buffered activity and identity/session storage; abort in-flight work where possible.

Method	Meaning
<code>track(name, properties, { schemaVersion })</code>	Validate and buffer a deliberate event. Schema version defaults to 1.
<code>page(name, properties)</code>	Buffer a deliberate page view; web mode adds optional context and route-visit deduplication.
<code>captureArrival(options)</code> — candidate	Capture eligible arrival evidence after consent when attribution is explicitly enabled; separate from automatic page views.
<code>flush()</code>	Perform or join one delivery attempt; useful during testing. It does not guarantee query visibility or wait through all retries.
<code>status()</code>	Inspect consent, queue size, attempts, queued/rejected/dropped counts and paused state for this instance.
<code>destroy()</code>	Stop this instance and discard its pending work. This is teardown, not consent withdrawal or server deletion.

A `track()` result of `buffered` means the SDK accepted the event into memory. A receipt of `queued` means the collector durably accepted it. Seeing it in the inspector confirms queryable storage. These are three different checkpoints.

Event names start with an ASCII letter and contain only letters, numbers, underscores or dots, up to 128 characters. Property keys start with an ASCII letter and contain only letters, numbers or underscores, up to 64 characters. Spaces and hyphens are rejected.

Properties must be JSON-compatible: strings, finite numbers, booleans, nulls, arrays and objects. Avoid undefined values, functions, dates as objects, cycles, non-finite numbers or deeply nested payloads. The contract limits one event to 32 KiB, a string to 8 KiB, depth to four, and keys/array sizes to bounded counts. Prefer a few short properties over a copy of an entire application object. Some credential-related property keys are rejected; validation cannot guarantee arbitrary free text contains no personal information.

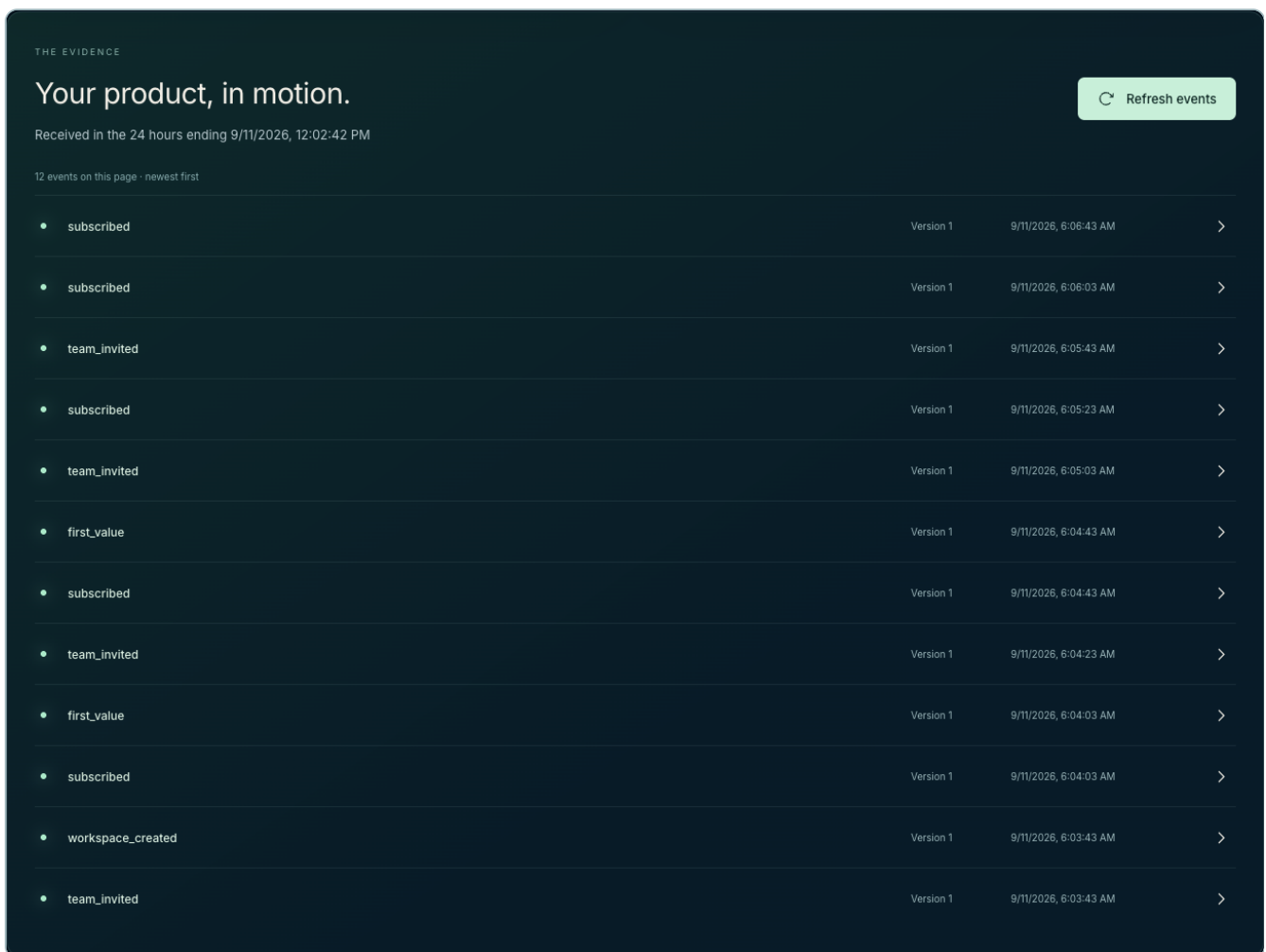
The SDK generates event IDs, occurrence timestamps, anonymous IDs and session IDs. You do not supply tenant IDs or authenticated person identities in a browser event. A property named `user_id` would not establish trusted identity linking.

9. Verify your first three events

Use a Test source first. After activation, load your own product with the SDK and grant analytics consent through your consent manager.

1. Complete the action that emits `trial_started`.

2. Complete the action that emits `value_reached`.
3. Complete the action that emits `subscribed`, using synthetic verification activity.
4. Optionally call `await analytics.flush()` and inspect `analytics.status()` in development.
5. Open Presagio, select the correct workspace, product and environment, open **Connection & events**, and choose **Refresh events**.
6. Open an event to check its name, schema version, timestamps and deliberately chosen properties.



THE EVIDENCE

Your product, in motion.

Received in the 24 hours ending 9/11/2026, 12:02:42 PM

12 events on this page · newest first

• subscribed	Version 1	9/11/2026, 6:06:43 AM	>
• subscribed	Version 1	9/11/2026, 6:06:03 AM	>
• team_invited	Version 1	9/11/2026, 6:05:43 AM	>
• subscribed	Version 1	9/11/2026, 6:05:23 AM	>
• team_invited	Version 1	9/11/2026, 6:05:03 AM	>
• first_value	Version 1	9/11/2026, 6:04:43 AM	>
• subscribed	Version 1	9/11/2026, 6:04:43 AM	>
• team_invited	Version 1	9/11/2026, 6:04:23 AM	>
• first_value	Version 1	9/11/2026, 6:04:03 AM	>
• subscribed	Version 1	9/11/2026, 6:04:03 AM	>
• workspace_created	Version 1	9/11/2026, 6:03:43 AM	>
• team_invited	Version 1	9/11/2026, 6:03:43 AM	>

Figure 3. An active product's event inspector: named milestones, schema versions and arrival times. Open a row to inspect its event details.

The inspector displays up to 50 events per page, newest received first, in the 24 hours ending at the displayed refresh time. Use its next-page control for older results in that same window. It is an event list, not a count of unique visitors. An enabled source indicates configuration state; stored events confirm that collection and processing succeeded.

For candidate web analytics, also verify one current-page view after consent and one eligible pathname change. Open Overview for the same scope and refresh. Browser and session totals need not equal page-view totals.

For a clear first journey, perform the milestones sequentially. If two events share the exact occurrence timestamp, Presagio uses event-ID order as a deterministic tie-breaker; IDs do not encode the human order of two simultaneous actions.

10. Define meaningful events on the fly

Raw events can arrive before you define their business meaning. **Event definitions** let you recognize meaningful moments in retained data without redeploying your frontend, provided the raw event and necessary properties were already captured. A definition cannot reconstruct activity that was never sent or has expired.

Open **Event definitions** and leave **Definition** on **Create a definition**:

1. Set **Definition key**, for example `first_value`. This is the stable identifier and cannot be edited after creation.
2. Set **Display label**, for example “First value”.
3. Set **Raw event name** to `value_reached`.
4. Set **Incoming schema version** to `1` if that is what your SDK sends.
5. Optionally choose **Add property rule**. For example, `feature` **Equals** text `first_report`.
6. Choose a **Preview window** and select **Preview matches**.
7. Inspect the matching-event count and up to five examples, then choose **Publish version 1**.

All property rules must match: the beta uses AND semantics, not arbitrary OR expressions. Supported comparisons are Equals, Does not equal, Exists, Greater than, At least, Less than and At most. Numeric comparisons require numeric values. Property paths such as `account.plan` address nested properties; array-index predicates and free-form query code are not supported. Missing values do not become matches merely because you selected “Does not equal”; Exists requires a present, non-null value.

Preview reports **matching events**, not unique people or journey entrants. The candidate limit is evaluated on the raw name and schema version before property filters. If more than 10,000 candidates fall in the window, shorten it to an hour, 15 minutes or a minute. Presagio refuses the oversized preview rather than presenting a partial count as complete.

Three versions that mean different things

Version	What it describes	Example
Incoming schema version	Shape/meaning of the event your frontend sends	<code>track('event_name', {}, { schemaVersion: 2 })</code> after an event-contract change.
Published definition version	A saved interpretation of that event	“First value” v2 adds <code>report_count >= 1</code> .
Published journey revision	The ordered milestones and their pinned definition versions	Journey v2 deliberately switches milestone 2 to First value v2.

Publishing a definition v2 leaves v1 unchanged. Existing journey revisions remain pinned to their selected definition versions; they do not silently adopt the new rule. Publish a new journey revision when you intentionally want the new meaning. A published rule is immutable, but its matching results can change as retained data arrives or is deleted.

Choose **Reload published version** before editing if another admin changed the definition. If publication could not be confirmed, refresh and inspect history before trying again. **Archive** prevents new versions while preserving published history for existing reports; it does not delete raw events or free the definition's capacity slot.

11. Build and read your founder journey

In **Journeys**, choose **Create a journey**, name it, and select two to five published definition versions as milestones. Add milestones as needed, then choose **Publish version 1**.

For the example in this guide, publish Trial started → First value → Subscribed. Next choose:

- **Published revision:** the exact saved journey you want to inspect.
- **Entrants in:** the period in which the first milestone occurred, from the last minute to the last seven days.
- **Time to complete:** one hour, 24 hours or seven days after each visitor's entry.

Choose **Reveal the journey**. Changing controls clears the previous report until you request a fresh result.



Figure 4. A five-milestone example: 480 entrants → 320 workspaces → 240 first-value moments → 150 team invitations → 96 subscriptions. Another 96 visitors still have time to complete. The waves and transition rows distinguish progression, pending visitors and elapsed windows.

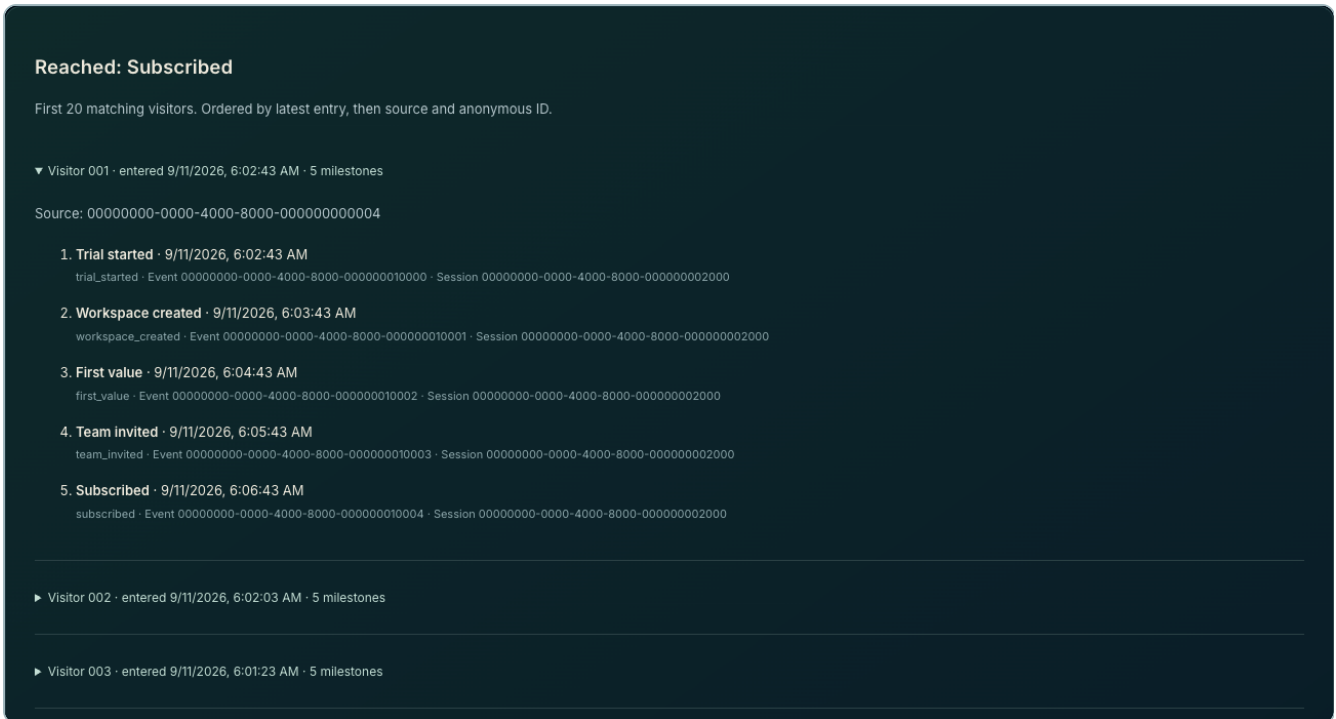


Figure 5. Clicking Subscribed opens the matching visitors; expanding one reveals its ordered milestones and event evidence. The same investigation is available for pending and elapsed groups.

Read the headlines first

Entered this journey counts distinct anonymous visitors within each source who matched the first milestone in the entrant period. **Reached the final milestone** counts those who completed the ordered path within the selected deadline. **Still within their window** counts incomplete entrants who still have time.

At each transition, Presagio shows how many progressed, how many still have time, and how many did not reach the next milestone before their window elapsed. “Elapsed” is a timing classification, not proof of churn, dissatisfaction or a lost customer.

What the waves mean

The strands spread through the center and converge at milestone points. **Peak spread represents a share of entrants. Exact milestone counts carry the quantitative meaning.** The glow is decorative; its brightness is not revenue, event volume, confidence or activity intensity.

Aquamarine represents progression, blue represents people still within their window, and copper represents an elapsed window. Read the labeled counts and transitions when comparing groups. An empty report displays an explicit empty state instead of invented waves.

Click through to the evidence

Choose a milestone's **Explore** control to see visitors who reached it, or a transition's **still have time / did not reach within window** control to inspect that group. Expand a visitor to see the milestone names, event IDs, session IDs and occurrence times supporting the result.

Evidence is limited to the first 20 matching visitors, ordered by latest entry. If more exist, the UI says so; do not treat the evidence list length as the total. Visitor labels are opaque and scoped. They are not names, emails or CRM profiles. Evidence clicks refresh the counts and evidence together; new processing or deletion can therefore change the totals.

12. How counting works

The counting rules are part of the product, not an implementation footnote:

- A visitor is a source-local anonymous identity. Presagio does not merge sources, browsers or devices in this beta.
- Different sessions for that same source-local identity can contribute to a journey. A journey is not necessarily one session.
- The earliest first milestone in the selected entrant period starts that entrant's clock.
- Each later milestone needs a later, distinct event. Repeated milestones cannot reuse one event twice.
- Equal occurrence timestamps use event-ID ordering. The entrant interval includes its start and excludes its end.
- The conversion window starts at each visitor's entry, not at midnight or the start of your dashboard period.
- Late processing and deletion can change later observations. Published definitions preserve meaning, not a frozen result set.
- No entrants means the completion rate is unavailable, not a misleading zero-percent conversion claim.

For example, if 100 visitors enter, 40 complete and 20 remain within their deadline, completion is 40%. The other 60 have not completed at this observation; only the 40 whose deadlines elapsed belong in an elapsed group. Do not lump the 20 pending visitors into confirmed failures.

13. Privacy, identity and delivery behavior

Consent starts as unknown for every SDK instance. Before an explicit grant, the SDK does not establish its analytics identity storage or transmit events. After grant, it uses source-scoped `localStorage` for anonymous identity and `sessionStorage` for the session. Sessions rotate after 30 minutes without a successfully buffered event; flushes and retries do not extend the session.

Storage denial degrades to in-memory identity and a diagnostic. Tabs cloned by the browser can inherit `sessionStorage`. Clearing storage or changing device may create another anonymous visitor. These constraints matter when comparing anonymous visitor counts to account counts from another system.

Withdrawal clears local buffers and identity/session storage and propagates a revocation signal to other same-origin instances when browser storage/signalling is available. If that mechanism is unavailable, your consent manager must coordinate the contexts. Aborting a request cannot retract an event already accepted by the server. Contact the Presagio team for server-side deletion; there is no self-service deletion screen yet.

Raw event payloads become eligible for expiration 30 days after server receipt. Cleanup runs asynchronously in bounded batches; this is not a guarantee of physical erasure at exactly day 30. The current inspector exposes a 24-hour window and journeys allow entrant periods up to seven days; stored retention and a screen's query range are different concepts. Privacy suppression blocks previously deleted subjects from being reintroduced through permitted retries. Operational backup integration is deferred; this beta does not promise host-loss recovery or high availability.

Delivery is bounded

The SDK batches up to 20 events and approximately 256 KiB per ordinary request, schedules a send after roughly two seconds, and permits one in-flight request. Its memory queue holds at most 200 events or 1 MiB. Unsent events may be lost when a page closes or crashes. Retries are bounded to eight attempts and 24 hours; there is no unlimited offline store or guaranteed unload delivery.

The service protects admission with rate, byte, concurrency and capacity limits. A retry of the same event ID and content does not create a second stored event or usage unit. A fresh `track()` call creates a new ID and is a new event. Admission attempts can consume collection allowance even when they do not become unique stored events. This is separate from metering; there is no active subscription checkout or billing-management UI.

14. Troubleshooting by symptom

What you see	What to do
Google refuses access	Verify the invited Google account and ask the team to check the test-user audience.
“You’re on the guest list”	Sign-in worked. Request Presagio admission, then choose Refresh account.
“Awaiting activation”	Ask the team to activate this exact project/environment/source. A key alone is insufficient.
SDK loads but nothing is sent	Check consent, source/key placeholders, success-handler wiring and duplicate installations. Inspect <code>status()</code> and browser network/CSP errors.
Events go to the wrong place	Recheck build environment, source ID, key and workspace/product/environment selectors.
Collection is refused	Check the exact website origin, key expiry/revocation and source activation. An auth refusal can pause the client; recreate it with a working configuration after correcting the cause.

What you see	What to do
Rate limit or temporary service-busy response	Let bounded SDK retries respect Retry-After. Do not add an aggressive retry loop. Ask the team to inspect capacity if it persists.
Device clock or old queued activity	Events more than seven days old or over five minutes ahead of server time are refused. Check the device clock and avoid replaying old activity as new.
<code>INVALID_EVENT</code> or rejected properties	Simplify the payload to a few safe JSON properties. Check the name, schema version, size, depth and forbidden credential-like fields.
Buffered or queued but no inspector event yet	Those statuses do not prove query visibility. Refresh the correct environment; if persistent, provide the event ID and approximate time to support.
Duplicate business actions	Ensure one SDK instance and one <code>track()</code> invocation per action. Retry deduplication does not merge separately generated IDs.
Overview is empty	Check the selected environment/source and period, source activation, consent and page-view configuration. Legacy custom business events alone do not populate Overview.
Overview country or context is Unknown	Some legacy events have no context; browser fields or country lookup may be unavailable. Inspect coverage. Do not replace missing values with a guessed country or direct arrival.
Overview exceeds its limit	Select Last 24 hours or one website source. No partial count is shown.
Chart days differ in duration	Seven-day charts follow selected local calendar boundaries, including daylight saving. Inspect exact bucket starts/ends; 24-hour charts use elapsed-hour buckets.
Attribution has no credit	Verify the matching candidate release, enabled attribution, eligible <code>captureArrival()</code> and conversion definition. Unknown entry history remains provisional; web context alone creates no touch.
Definition preview shows zero	Check exact event name/schema, property types, all AND rules, preview period and selected environment.
Preview or journey exceeds a limit	Choose a shorter time period. No partial count is shown. Property rules do not shrink the preview's candidate cap.
Journey counts differ from preview	Preview counts events; journeys count source-local visitors who meet ordered milestones and timing rules.
Journey appears empty	Verify first-milestone matches, published revision, environment and entrant interval.
Older revision did not adopt a new definition	Expected: versions are pinned. Publish a new journey revision intentionally.

What you see	What to do
“Cohort window has expired”	Choose Reveal the journey to refresh the period, then reopen evidence.
Save or publication was not confirmed	Refresh and inspect current metadata/history before another mutation. Avoid creating duplicate objects.
Workspace changed after returning to the tab	Re-select the intended workspace, product, environment and source before continuing.
Sign-in or permission disappears	Sign in again or ask an administrator to check current membership. Do not share session cookies.

For support, share the affected workspace/product/environment, action, approximate time and timezone, error code, and a sanitized screenshot or event ID. Do not send cookies, OAuth secrets, raw customer payloads or database credentials. Use the Presagio team contact that provided your invitation; an in-product support inbox is not part of this beta.

15. Practical beta limits

These are current product constraints, not premium pricing entitlements or guarantees of sustained throughput. Ask the team before designing a high-volume rollout.

Area	Current limit or behavior
Workspaces	Five across this beta deployment; one owned workspace per principal.
Projects / environments	Two projects per workspace; fixed Production and Test environments.
Sources / active keys	Five sources per environment; five active keys per source.
Event definitions	50 per project, 20 published versions each; archived definitions still count.
Property rules	Up to five AND conditions per definition.
Journeys	20 per project, 20 revisions each, two to five milestones; archived journeys still count.
Definition preview	Up to 10,000 candidates; up to five examples.

Area	Current limit or behavior
Journey query	Up to 10,000 candidate events; oversized queries are refused rather than partially counted.
Journey evidence	Up to 20 visitors for the selected group; the UI labels truncation.
Inspector	50 events per page in a 24-hour refreshed window.
Overview — candidate	Rolling 24/168-hour windows; 10,000 candidate page views; top 20 groups per ranking with omitted counts disclosed.
Attribution — candidate	One source, published conversion definition, 24-hour/7-day period and seven-day touch lookback; provisional source-local browser evidence.
Journeys	Entrant periods up to seven days; one-hour, 24-hour or seven-day conversion windows.
Payload retention	Expiration eligibility at 30 days from receipt, followed by bounded asynchronous cleanup; separate deduplication/suppression safeguards.

16. A founder's operating routine

After installation: on the matching web candidate, verify consent-aware page views and Overview first. Then verify three deliberate business events in Test; preview definitions; publish one journey; compare the evidence to what you actually did. Ask for Production activation only when the flow is understood.

During a product review: select a consistent environment, journey revision, entrant period and completion window. Read entrants, completed and pending together. Open evidence at the first weak transition. Treat a small sample as a small sample, not a reliable trend.

Before changing the product: write down a hypothesis, such as “People reach first value but cannot see the next action.” Inspect the recorded sequence and use separate customer research to understand why. A journey shows behavior, not causation.

After a release: compare equivalent periods and pinned rules. If you change event semantics, coordinate a new schema version and definition revision. If you change the business milestone, publish a new journey revision. Keep the old revision available for an honest comparison.

17. First-launch checklist

- ☐ Google account admitted and correct workspace selected.

- ☐ Product, Test/Production environment and website source confirmed.
- ☐ Source activated by the Presagio team.
- ☐ Exact website origins listed and current public key installed.
- ☐ One SDK instance initialized in the browser.
- ☐ Consent grant and withdrawal connected to the actual consent manager.
- ☐ Matching SDK/runtime release confirmed before enabling web or attribution candidate options.
- ☐ Candidate web mode explicitly configured; one current page after grant and eligible SPA navigation verified without a duplicate initial page call.
- ☐ Sensitive paths mapped/excluded and context fields checked.
- ☐ Overview scope, browser/session counting, timezone and Unknown coverage understood.
- ☐ Successful product actions emit deliberate, non-sensitive events once.
- ☐ Site script/connect policies allow the chosen SDK and endpoint.
- ☐ Stored events inspected in the correct environment.
- ☐ Definitions previewed and published with the intended schema versions.
- ☐ Journey revision published; entrant period and conversion window understood.
- ☐ Milestone evidence checked against the expected real sequence.
- ☐ Test keys revoked when no longer needed; production key rotation scheduled before expiry.

18. Glossary and next steps

Event: one recorded action. **Property:** a deliberate dimension attached to it. **Source:** a website integration boundary. **Environment:** Test or Production. **Definition:** a versioned matching rule. **Milestone:** a definition version used in a journey. **Entrant:** an anonymous visitor matching the first milestone in the selected period. **Conversion window:** time allowed after that entry. **Pending:** incomplete with time remaining. **Elapsed:** deadline passed without the next milestone. **Queued:** durably accepted for processing. **Stored:** persisted and available to authorized queries.

Start in [your workspace](#), or explore the [sample experience](#) before instrumenting your product. The sample experience is useful for orientation, but it is not evidence that your own integration is receiving data.