
RESEARCH BRIEF 01 / OCTOBER 2026

Where website enquiries disappear

A diagnostic guide from click to the sales inbox

Trace a controlled enquiry through the saved record, notification and intended inbox. Diagnose the first stage that cannot be verified.

IN THIS BRIEF

- 01 Evidence and diagnosis
- 02 Retries and analytics
- 03 A reusable diagnostic record

A prospect submits an enquiry. The website displays a confirmation, the analytics dashboard records a conversion, and the sales team says nothing arrived. Those observations can all be true. Each describes a different stage of the same process.

Before changing the page or buying more traffic, trace one controlled enquiry through the system. The objective is to locate the first stage that cannot be verified.

1. Define what success means at each stage

The following is a proposed Layerly diagnostic model. The state names are not HTTP or Google Analytics standards.

Stage	Evidence to collect	What remains unproven
Submitted	The browser sent the agreed payload	The server stored it
Stored	The record can be read back using its reference	A notification was sent
Notification accepted	The sending service accepted the message	The intended inbox contains it
Inbox confirmed	The recipient located the same reference	A person owns the follow-up
Assigned	The request has a responsible owner	A reply or sale occurred

An HTTP response is evidence about the request. A 200 response reports success according to the endpoint's behaviour; it is not an independent audit of the application. A 202 response explicitly permits processing to remain unfinished. Check the response body and the application record before translating either into a business outcome.

Source: [1] RFC 9110: HTTP response semantics

2. Use one traceable test

On a site you own or are authorised to test, agree a receiving person and a short test window. Submit harmless, clearly labelled details once. Use a reference such as TEST-ENQUIRY-001 and record the time, page, browser and expected destination.

Ask the administrator to locate the saved record. Ask the recipient to locate the corresponding notification. Check any acknowledgement to the visitor separately: the visitor receiving a receipt does not establish that the sales team received its message.

If access to a stage is unavailable, mark it unverified. Do not infer a failed database from a missing email, or a working inbox from a confirmation banner.

3. Diagnose the first missing link

Observation	Next investigation	Premature conclusion to avoid
Confirmation shown, no stored record found	Check endpoint response, validation and persistence for that reference	The email provider lost it
Record stored, no notification attempt	Check the job or integration that should create the notification	The website needs redesigning
Sending service accepted, inbox not confirmed	Check recipient routing, delivery events and the mailbox with its owner	The recipient read it
Inbox confirmed, no assignment	Check routing rules and the responsible team	More traffic will solve the problem

SMTP acceptance matters, but it is a transport-stage acknowledgement. RFC 5321 distinguishes acceptance for delivery or relay from subsequent delivery handling. A log from your sending relay does not show where the message appeared in the recipient's mailbox or whether anyone read it.

Source: [\[2\] RFC 5321: mail acceptance and delivery](#)

4. Test ambiguous retries before launch

A request may be stored even when the browser times out before receiving confirmation. Blindly submitting again can create duplicates.

A suitable implementation can associate retries of the same intended submission with one idempotency key. The server must enforce that relationship atomically, verify that the payload matches and return the existing result or an explicit conflict. Merely adding an ID field does not prevent duplicates. A genuinely new enquiry needs its own identifier.

Stripe documents an implementation of idempotent API requests. It is a useful engineering reference, not evidence that a particular website form already supports the same behaviour.

Source: [\[3\] Stripe: idempotent API requests](#)

5. Reconcile analytics with operational records

Google Analytics defines `generate_lead` as an event for a generated lead, including through a form. The event name cannot verify the website's implementation. If it fires on a button click or before storage is confirmed, its count describes that trigger.

Source: [\[4\] Google Analytics: recommended lead events](#)

Our proposed measurement contract is to record a successful enquiry after the application confirms the intended durable record. Keep detailed internal references out of public reports. Use comparable cohorts and time windows when reconciling systems: a consent-limited analytics count is not automatically the denominator for every backend enquiry.

For an illustrative controlled test of ten unique submissions, ten confirmations, nine saved records and eight inbox-confirmed notifications locate two gaps worth investigating. They do not establish how many historical customers or sales were lost.

Keep this diagnostic record

Field	What to record
Page	URL and action under review.
test reference	Harmless identifier linking the same test across stages.
time	Actual observation time and time zone.
stored record	Readback evidence, or unverified.
notification attempt	Attempt identifier and sending-service response.
inbox evidence	Confirmation from the intended recipient.
owner	Person responsible for the next action.
correction	Specific agreed edit.
retest	Repeated action, date and observed result.

The completion test should repeat the original path, verify the same stages and check the retry case. A correction is ready to hand over when another person can follow the evidence from the visitor's action to the agreed receiving point.

All test identifiers and counts in this article are illustrative. The state model and measurement contract are Layerly proposals. No live customer test or revenue result is asserted.

Technical sources

Primary documentation reviewed 30 October 2026. The diagnostic frameworks and illustrative examples are identified in the article.

[1] RFC 9110: HTTP response semantics

<https://www.rfc-editor.org/rfc/rfc9110.html#section-15.3>

[2] RFC 5321: mail acceptance and delivery

<https://www.rfc-editor.org/rfc/rfc5321.html#section-6.1>

[3] Stripe: idempotent API requests

https://docs.stripe.com/api/idempotent_requests

[4] Google Analytics: recommended lead events

https://developers.google.com/analytics/devguides/collection/ga4/reference/events#generate_lead

