

Chatwoot reports, audit, and events

Use these 12 read-only capabilities to inspect Chatwoot audit records, reporting events, and metrics for the configured account. Keep each provider observation separate from AIP's own Action, approval, routing, Event, and outbox evidence.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/chatwoot/capabilities/reports-audit-and-events.md

Every operation uses `GET`. None changes provider state through its frozen route, but every result remains restricted and potentially contains PII.

Family at a glance

Group	Operations	Provider API	Risk	Approval	Retry
Account audit log	1	v1	Low	No	Safe
Account reporting events	1	v1	Low	No	Safe
Reports and summaries	10	v2	Low	No	Safe
Total	12		Low	No	Safe

All 12 capabilities are Tools. They execute synchronously and advertise no async, streaming, or cancellation support.

Capability-level approval is not required. The authenticated actor, tenant, admitted connector route, credential revision, and data policy still govern access.

Keep evidence sources separate

Evidence source	What it can show	What it does not prove
<code>audit_log.list</code>	Provider account audit records returned by Chatwoot	AIP approval, routing, dispatch, or durable result
<code>reporting_event.list</code>	Provider account reporting events	AIP Event publication or outbox delivery
<code>report.*</code>	Provider metrics for a selected query and time	Correctness of AIP lifecycle evidence
AIP Action and result	Governed request identity and terminal execution state	Complete provider business history
AIP approval record	Actor, input, policy, and decision binding	Provider-side effect by itself
AIP Event and outbox evidence	Durable event identity and publication progress	Provider metric semantics

Correlate sources through retained identities and timestamps when investigating an incident. Do not replace one evidence class with another because names such as `audit` or `event` appear in both products.

Version-one audit and event operations

Capability	Provider request	Purpose
<code>cap:chatwoot:audit_log.list</code>	GET /api/v1/accounts/{account_id}/audit_logs	List account audit-log entries
<code>cap:chatwoot:reporting_event.list</code>	GET /api/v1/accounts/{account_id}/reporting_events	List account reporting events

`reporting_event.list` is account-scoped. The separate `conversation.reporting_event.list` capability reads one conversation's enterprise reporting events through a conversation-qualified route.

Neither operation is the connector host's webhook ingress or AIP Event API.

Version-two report operations

Capability	Provider request	Purpose
<code>cap:chatwoot:report.list</code>	GET /api/v2/accounts/{account_id}/reports	Read the account report series
<code>cap:chatwoot:report.summary</code>	GET /api/v2/accounts/{account_id}/reports/summary	Read a report summary
<code>cap:chatwoot:report.conversation</code>	GET /api/v2/accounts/{account_id}/reports/conversations	Read conversation metrics
<code>cap:chatwoot:report.first_response_distribution</code>	GET /api/v2/accounts/{account_id}/reports/first_response_time_distribution	Read first-response-time distribution metrics
<code>cap:chatwoot:report.inbox_label_matrix</code>	GET /api/v2/accounts/{account_id}/reports/inbox_label_matrix	Read the inbox-label metric matrix
<code>cap:chatwoot:report.outgoing_message_count</code>	GET /api/v2/accounts/{account_id}/reports/outgoing_messages_count	Read outgoing-message counts
<code>cap:chatwoot:report.agent_summary</code>	GET /api/v2/accounts/{account_id}/summary_reports/agent	Read agent summary metrics
<code>cap:chatwoot:report.team_summary</code>	GET /api/v2/accounts/{account_id}/summary_reports/team	Read team summary metrics
<code>cap:chatwoot:report.inbox_summary</code>	GET /api/v2/accounts/{account_id}/summary_reports/inbox	Read inbox summary metrics
<code>cap:chatwoot:report.channel_summary</code>	GET /api/v2/accounts/{account_id}/summary_reports/channel	Read channel summary metrics

The v2 segment is part of each frozen provider path. It does not indicate AIP protocol version 2, a second connector profile, or a different capability-ID namespace.

Account and query inputs

The connector inserts its configured `account_id` into every route. This family has no other path placeholder.

Each operation accepts the generic optional `query` object:

Query boundary	Value
Keys	At most 128
Key length	1–256 bytes without control characters
Values	String, number, integer, Boolean, null, or bounded arrays of those scalars
Array length	At most 256 values
Null handling	Omitted from the provider query

The connector does not freeze date ranges, grouping dimensions, timezone, inbox, agent, team, channel, pagination, or other provider query names. Validate the query against the exact pinned Chatwoot route.

The generic schema also exposes `body` and `multipart`, but no operation in this family publishes a provider-specific request body. Do not add a body merely because the generic envelope permits one.

Read account audit records

This call demonstrates the source-owned route without assuming provider query fields:

```
SH
aipctl action call \
  "$AIP_URL" \
  cap:chatwoot:audit_log.list \
  --action-id act_chatwoot_audit_log_001 \
  --input '{}'
```

The connector sends:

```
TEXT
GET /api/v1/accounts/{configured_account_id}/audit_logs
```

Chatwoot can still require provider state or query values not expressed by the generic connector schema. A completed result proves only that the selected provider request succeeded.

Shared read contract

All 12 capabilities declare:

Contract field	Value
Kind	Tool
Risk	Low
Approval	Not required by the capability
Idempotency	Optional; external-account scope; 86,400,000 ms published TTL
Connector retry	Supported; safe
Side effects	<code>read</code> , <code>external_network</code>
Data	Restricted; contains PII; redaction required
Execution	Synchronous; no async, streaming, or cancellation support
Compensation	Not required for the read

Safe retry means another governed provider read is permitted after a suitable failure. It does not promise an identical metric snapshot; Chatwoot state and the query's time window can change.

Result contract

Every successful operation returns:

```
JSON
{
  "http_status": 200,
  "body": {},
  "provider_request_id": null
}
```

`body` contains parsed JSON, `null` for an empty response, or a `text` wrapper for non-JSON bytes. `provider_request_id` uses `x-request-id` first and `x-runtime` second.

The output schema does not freeze provider rows, metric names, units, dimensions, pagination, or aggregation behavior. Validate the result shape and metric semantics required by the application.

Failures and retry

Failure code	Typical cause	Safe response
<code>connector.chatwoot.invalid_action</code>	Unknown or unadmitted capability or malformed query	Refresh discovery or correct the query
<code>connector.chatwoot.authentication</code>	Provider returned 401 or 403	Verify account binding and credential revision
<code>connector.chatwoot.remote_rejected</code>	Provider rejected query or current state	Correct the provider request before retrying
<code>connector.chatwoot.remote_temporary</code>	Provider returned 429 or a server error	Retry with bounded backoff inside policy and deadline
<code>connector.chatwoot.transport</code>	Provider exchange failed	Retry the same read contract within policy
<code>connector.chatwoot.response_too_large</code>	Response crossed the active bound	Narrow the provider query or review the configured bound

Connector failures set `retryable: false`, so an error never directs a blind repeat. The capability contract separately marks these reads `retry-safe`.

Preserve the original query and observation time when retrying. A later successful response is a new provider observation, not proof of the missing response's contents.

Handle reporting data

Apply restricted-data controls to:

- actor or customer identifiers in audit records;
- conversation, inbox, label, agent, team, and channel dimensions;
- row-level event detail;
- query strings and retained provider responses;
- exported evidence used for incident or performance analysis.

Redaction-required does not mean the connector redacts every provider field in this generic result. The application and evidence pipeline must enforce the tenant's field, access, retention, and disclosure policy.

Verify an integration

Require all of the following:

- discovery admits the exact report or audit capability;
- the route version and account binding match the intended provider surface;
- query names and metric semantics come from the pinned Chatwoot API;
- the completed result belongs to the submitted Action;
- provider request identity and observation time are retained;
- retry results are treated as new snapshots;
- restricted audit, event, and metric data is redacted and retained under

tenant policy;

- provider observations are correlated with, not substituted for, AIP

execution evidence.

These checks validate observation handling. They do not establish metric accuracy, completeness of provider history, or live qualification.

Related documentation

- [Chatwoot capability index](#)
- [Conversations and messages](#)
- [Actions and sessions](#)
- [Observe and recover](#)
- [Error reference](#)