

Chatwoot accounts, agents, teams, and bots

Use these 22 capabilities to read the configured account and manage its human agents, teams, team membership, and agent bots. The connector owns the account boundary; an Action cannot select another Chatwoot account through its input.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/chatwoot/capabilities/accounts-agents-teams-and-bots.md

This page documents the exact routes and the generic AIP request envelope frozen in the reviewed source. It does not invent provider payload fields that the Rust catalogue does not validate.

Family at a glance

Group	Operations	Reads	Mutations	High-risk deletes
Account	2	1	1	0
Agents	6	2	4	1
Teams and members	9	3	6	2
Agent bots	5	2	3	1
Total	22	8	14	4

All 22 capabilities are Tools. The eight `GET` operations are low-risk reads. The five `POST` and five `PATCH` operations are medium-risk mutations. The four `DELETE` operations are high-risk mutations.

Every mutation requires approval and an idempotency key. Reads do not require capability-level approval and advertise safe retry support.

Account and agent operations

Capability	Provider request	Purpose	Risk	Approval
cap:chatwoot:account.get	GET /api/v1/accounts/{account_id}	Read the configured account	Low	No
cap:chatwoot:account.update	PATCH /api/v1/accounts/{account_id}	Update the configured account	Medium	Required
cap:chatwoot:agent.list	GET /api/v1/accounts/{account_id}/agents	List account agents	Low	No
cap:chatwoot:agent.create	POST /api/v1/accounts/{account_id}/agents	Create an agent	Medium	Required
cap:chatwoot:agent.update	PATCH /api/v1/accounts/{account_id}/agents/{agent_id}	Update one agent	Medium	Required
cap:chatwoot:agent.delete	DELETE /api/v1/accounts/{account_id}/agents/{agent_id}	Delete one agent	High	Required
cap:chatwoot:agent.bulk_create	POST /api/v1/accounts/{account_id}/agents/bulk_create	Create agents in one bounded provider request	Medium	Required
cap:chatwoot:agent.assignable.list	GET /api/v1/accounts/{account_id}/assignable_agents	List agents available for assignment	Low	No

`agent.bulk_create` is one provider mutation. The catalogue does not define a per-item transaction, partial-success model, or compensating operation. Preserve the provider result before deciding how to recover a partial import.

Team and membership operations

Capability	Provider request	Purpose	Risk	Approval
cap:chatwoot:team.list	GET /api/v1/accounts/{account_id}/teams	List teams	Low	No
cap:chatwoot:team.create	POST /api/v1/accounts/{account_id}/teams	Create a team	Medium	Required
cap:chatwoot:team.get	GET /api/v1/accounts/{account_id}/teams/{team_id}	Read one team	Low	No
cap:chatwoot:team.update	PATCH /api/v1/accounts/{account_id}/teams/{team_id}	Update one team	Medium	Required
cap:chatwoot:team.delete	DELETE /api/v1/accounts/{account_id}/teams/{team_id}	Delete one team	High	Required
cap:chatwoot:team.member.list	GET /api/v1/accounts/{account_id}/teams/{team_id}/team_members	List team members	Low	No
cap:chatwoot:team.member.create	POST /api/v1/accounts/{account_id}/teams/{team_id}/team_members	Add team members	Medium	Required
cap:chatwoot:team.member.update	PATCH /api/v1/accounts/{account_id}/teams/{team_id}/team_members	Replace team members	Medium	Required
cap:chatwoot:team.member.delete	DELETE /api/v1/accounts/{account_id}/teams/{team_id}/team_members	Remove team members	High	Required

The three membership mutations share one collection route. Their provider methods distinguish add, replace, and remove behavior. Review the exact member set in the approved body, especially before `team.member.update`.

Agent-bot operations

Capability	Provider request	Purpose	Risk	Approval
cap:chatwoot:agent_bot.list	GET /api/v1/accounts/{account_id}/agent_bots	List agent bots	Low	No
cap:chatwoot:agent_bot.create	POST /api/v1/accounts/{account_id}/agent_bots	Create an agent bot	Medium	Required
cap:chatwoot:agent_bot.get	GET /api/v1/accounts/{account_id}/agent_bots/{agent_bot_id}	Read one agent bot	Low	No
cap:chatwoot:agent_bot.update	PATCH /api/v1/accounts/{account_id}/agent_bots/{agent_bot_id}	Update one agent bot	Medium	Required
cap:chatwoot:agent_bot.delete	DELETE /api/v1/accounts/{account_id}/agent_bots/{agent_bot_id}	Delete one agent bot	High	Required

Agent-bot records can influence automated handling in Chatwoot, but the generic contract declares only `write` and `external_network` side effects. Apply provider-aware policy without treating the generic markers as an exhaustive impact model.

Account and path inputs

The host inserts its configured `account_id` into every provider path. Do not put `account_id` in Action input; it is not present in the published input schema.

The remaining placeholders become required top-level strings:

Input	Required by
agent_id	agent.update and agent.delete
team_id	team.get, team.update, team.delete, and all four team.member.* operations
agent_bot_id	agent_bot.get, agent_bot.update, and agent_bot.delete

Each path value must be non-blank, contain no control character, and fit within 512 bytes. The URL builder inserts it as one encoded path segment. A caller cannot use a slash inside the value to select another route.

Generic catalogue input

Every operation accepts the same source-generated envelope around its path inputs:

Property	Shape	Connector boundary
Path inputs	Required strings for placeholders other than <code>account_id</code>	Non-blank; at most 512 bytes
query	Optional object of scalar values or scalar arrays	At most 128 keys and 256 values per array
body	Optional JSON object	At most 512 properties and 16 MiB when encoded
multipart	Optional fields-and-files object	Mutually exclusive with body

A null query value is omitted. String, number, integer, and Boolean values are encoded as query values. Nested query objects are rejected.

The connector does not freeze operation-specific body fields for this family. It validates only that `body` is an object within the generic bounds, then passes it to the pinned provider route. Use the provider schema admitted for the pinned Chatwoot revision. Do not infer that a schema-valid empty object is a valid provider mutation.

The same generic input technically permits multipart data. Use it only when the pinned provider operation explicitly accepts multipart content. These 22 catalogue entries publish no operation-specific multipart requirement.

Read an assignable-agent list

Select the capability from tenant discovery, then send only the provider query that the admitted route needs:

```
SH
aipctl action call \
  "$AIP_URL" \
  cap:chatwoot:agent.assignable.list \
  --action-id act_chatwoot_assignable_agents_001 \
  --input '{"query":{"page":1}}'
```

The connector inserts the account and calls:

```
TEXT
GET /api/v1/accounts/{configured_account_id}/assignable_agents?page=1
```

The read is safe to retry at the capability-contract level. A retry still requires the same tenant route, deadline, and policy context. It can return a different list if Chatwoot state changed between calls.

Update an agent

The pinned connector fixture demonstrates this exact generic input:

```
JSON
{
  "agent_id": "42",
  "body": {
    "name": "Operations"
  }
}
```

Submit it as one governed Action:

```
SH
aipctl action call \
  "$AIP_URL" \
  cap:chatwoot:agent.update \
  --action-id act_chatwoot_agent_update_001 \
  --idempotency-key agent-update-42-v1 \
  --input '{"agent_id":"42","body":{"name":"Operations"}}'
```

Expected first outcome: the policy path may hold the Action for approval. The approval must bind this capability, agent ID, body, Action identity, actor, tenant, and policy hash.

After authorization, the connector sends `PATCH` to the configured account, adds `X-AIP-Action-ID` and `Idempotency-Key`, and uses the configured `api_access_token`. Do not place the token in Action input or evidence.

Mutation safety

All 14 mutations share this AIP contract:

Contract field	Value
Approval	Required; tenant-policy selector
Approval validity	900,000 ms
Approval evidence	Reason, input snapshot, policy decision
Idempotency	Required; external-account scope; 86,400,000 ms published TTL
Collision behavior	Revalidate input hash
Connector retry support	No

Contract field	Value
Retry safety	Unsafe
Side effects	<code>write, external_network</code>
Data	Restricted; contains PII; redaction required
Execution	Synchronous; no async, streaming, or cancellation support
Compensation	Rollback not supported

An idempotency key binds one intent inside the external-account scope. The connector forwards that key to Chatwoot, but the contract does not prove that every pinned provider route deduplicates requests.

If dispatch may have occurred, retain the original Action and key. Inspect authoritative Chatwoot state before any corrective mutation. Do not create a new Action merely because the original response was lost.

Result contract

A successful catalogue call returns:

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

`http_status` contains the successful provider status. `body` contains parsed JSON, `null` for an empty response, or `{"text": "..."} for a non-JSON body. provider_request_id uses x-request-id when present and falls back to x-runtime.`

The default response bound is 16 MiB and configuration can raise it only to the connector's hard 128 MiB ceiling. A successful HTTP response that exceeds the active bound fails before an Action result is returned.

Provider response fields remain provider-owned. Preserve unknown fields and validate the specific object shape required by the application.

Failures and recovery

Failure code	Typical cause	Safe response
<code>connector.chatwoot.invalid_action</code>	Unknown, unadmitted, or malformed capability input	Refresh discovery or correct the path/query/body input
<code>connector.chatwoot.missing_idempotency_key</code>	Mutation has no valid key	Restore the key bound to the original intent before dispatch
<code>connector.chatwoot.authentication</code>	Provider returned 401 or 403	Verify account binding and credential revision
<code>connector.chatwoot.remote_rejected</code>	Provider rejected another client request	Correct provider state or payload before a new Action
<code>connector.chatwoot.remote_temporary</code>	Provider returned 429 or a server error	Retry a read within policy; reconcile a mutation
<code>connector.chatwoot.transport</code>	Provider exchange failed	Retry a read within policy; treat a mutation outcome as uncertain
<code>connector.chatwoot.response_too_large</code>	Response crossed the configured bound	Inspect provider output and the reviewed bound; reconcile a mutation
<code>connector.chatwoot.invalid_credential</code>	Configured token is not usable UTF-8	Rotate or replace the owner-only credential

The failure mapping sets `retryable` to false for every connector failure. That prevents an error from instructing a blind retry. The capability contract separately permits bounded retries for the eight reads.

Transport, temporary provider, and oversized-response failures mark an invocation outcome uncertain. For a mutation, uncertainty means the provider may already contain the change. Read authoritative state before deciding

on a separately approved correction.

Verify an integration

Require all of the following:

- tenant discovery returns only the admitted family operations;
- the Action contains no caller-selected account ID;
- each non-account path placeholder is present and bound to the intended provider object;
- every mutation has exact approval evidence and one retained idempotency key;
- the completed result belongs to the submitted Action and preserves provider status and request identity;
- a read-back confirms consequential writes when the provider offers one;
- restricted account, agent, team, and bot data follows tenant redaction and retention policy;
- uncertain mutations are reconciled instead of repeated.

These checks validate application handling. They do not establish live qualification of a Chatwoot version or permission to administer every object in the configured account.

Related documentation

- [Chatwoot capability index](#)
- [Authentication and account isolation](#)
- [Chatwoot connector configuration](#)
- [Approvals and policy](#)
- [Error reference](#)