

CrewAI knowledge and memory capabilities

Use `knowledgequery` to retrieve bounded results from a configured crew knowledge store without running its task graph. Use `memoryreset` to clear one explicit CrewAI memory domain or all available domains.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/crewai/capabilities/knowledge-and-memory.md

The two operations have opposite safety profiles. Query is a retry-safe read; reset is a high-risk mutation without rollback.

Compare knowledge query and memory reset

Property	knowledge_query	memory_reset
Capability	cap:crewai:<crew_id>:knowledge_query	cap:crewai:<crew_id>:memory_reset
Kind	Tool	Tool
Risk	Low	High
Provider mutation	No	Yes
Approval	No	Required
Idempotency key	Required	Required
Streaming	No	No
Cooperative cancellation	No	No
Connector retry	Yes	No
Compensation	None	Rollback not supported

Both operations use a durable sidecar job. This is why the read still requires an idempotency key.

The retry row reflects published implementation support. The reviewed concrete failure mapper still reports every connector failure as non-retryable.

Query crew knowledge

The input requires one through 100 query strings:

```
JSON
{
  "query": [
    "refund policy for annual subscriptions",
    "approved exception process"
  ],
  "results_limit": 5,
  "score_threshold": 0.4
}
```

Field	Required	Constraint	Default
<code>query</code>	Yes	Array of 1 through 100 non-empty strings	None
Query item	Yes	At most 4,096 UTF-8 bytes and no control characters	None
<code>results_limit</code>	No	Integer from 1 through 100; booleans rejected	3
<code>score_threshold</code>	No	Number from 0 through 1; booleans rejected	0.35

Unknown fields are rejected. The complete encoded input may not exceed 1 MiB.

The operation catalogue records source revision `bfa652a7` as the method- verification baseline. The default product image installs CrewAI `1.15.5`.

The sidecar prefers:

```
TEXT
crew.aquery_knowledge(query, results_limit=..., score_threshold=...)
```

It falls back to `crew.query_knowledge(...)` in a worker thread when the `async` method is absent. CrewAI queries only the crew-level knowledge object; it returns `None` when that object is not configured.

The sidecar serializes returned search results and normalizes `None` to an empty array:

```
JSON
{
  "results": []
}
```

An empty array can mean no configured crew knowledge, no match above the threshold, or no provider result. The connector does not add a reason field.

Interpret knowledge results safely

Result structure is owned by the pinned CrewAI knowledge implementation and its configured storage backend. The connector converts model objects, mappings, sequences, primitives, and unsupported values into bounded JSON.

Do not compare scores across different embedding models, indexes, backends, or crew revisions as though they share one calibration. Record those identities with any evaluation evidence.

The read can expose confidential or personal data. Apply tenant routing, provider access control, query authorization, output filtering, and redaction before returning content to an application.

The connector does not accept a knowledge store, namespace, embedding model, or provider credential in Action input. Those values belong to the admitted crew factory and deployment.

Preserve the durable query identity

`knowledge_query` is read-only at the CrewAI boundary, but the sidecar creates a journaled job before calling the provider. Use one stable Action ID and key for a logical query.

The persisted request hash includes crew, Action, operation, complete input, timeout, and key. An exact repeat returns the existing record; changed material under the same Action ID returns a conflict.

An ordinary provider failure event contains an exception class and generic message, but no `uncertain_outcome` field. The read-only contract informs operator classification; the missing field is not a false value.

If local cancellation wins while the blocking request is pending, the AIP result is cancelled. The provider operation is read-only, but its completion is unobserved and the connector does not claim a remote stop.

At this revision, that result also says `dispatched: false`. The select branch does not prove whether the HTTP request already reached the sidecar, so do not use that field as transport evidence.

Reset an explicit memory domain

The input contains exactly one field:

```
JSON
{
  "command_type": "kickoff_outputs"
}
```

Accepted values are:

Value	Pinned CrewAI target
memory	Crew memory object
knowledge	Crew knowledge and every configured agent knowledge store
agent_knowledge	Configured agent knowledge stores only
kickoff_outputs	Task-output handler storage
all	Every available target above, in CrewAI's iteration order

The sidecar calls `crew.reset_memories(command_type)` in a worker thread. A specific target that is not initialized causes the pinned CrewAI method to fail.

An `all` reset skips unavailable systems and resets available systems one by one. Failure after an earlier reset can therefore leave a partial result.

The completed result contains only:

```
JSON
{
  "status": "completed",
  "command_type": "kickoff_outputs"
}
```

It does not enumerate deleted records, prove secure erasure, or provide a restoration point.

Review a reset before approval

Memory reset requires tenant-policy approval, a stable idempotency key, and the complete input snapshot. Review:

1. admitted crew, tenant, instance, sidecar state, and provider identities;
2. the exact domain selected by `command_type`;
3. data retention, legal hold, audit, and incident requirements;
4. downstream behavior that depends on knowledge or memory continuity;
5. a deployment-owned backup or export when restoration is required;
6. possible partial completion across the selected targets;
7. a verification method that does not repopulate or mutate unrelated data.

Choose a specific domain instead of `all` when the narrower operation satisfies the task. Approval for a query does not authorize a reset.

Understand the irreversible boundary

The connector publishes no transaction, reconciliation, compensation, or provider rollback for memory reset. An idempotency fence prevents a different request from reusing the same Action ID, but it does not restore deleted state.

The operation is not cooperatively cancellable. Cancelling its waiter or reaching the timeout can leave the worker thread and storage effects continuing.

Local cancellation can return `dispatched: false` without proving that the sidecar did not admit the job. Continue status and provider reconciliation under the original Action ID.

An ordinary provider failure records `failed` without an uncertainty field. Treat reset effects as uncertain because this is a worker-thread mutation, not because the event says so.

A sidecar restart converts a restored running reset into `failed` with `uncertain_outcome: true` rather than repeating it.

Recover after an uncertain reset

1. preserve the Action ID, key, crew, command type, approval, and input hash;
2. read durable status and events without starting a replacement reset;
3. inspect each selected CrewAI storage target through an approved read path;
4. classify targets as unchanged, reset, repopulated, unavailable, or unknown;
5. restore from an authorized backup only under deployment policy;
6. create a new mutation only after the partial-effect boundary is known.

Do not use `knowledge_query` alone to prove the state of agent knowledge, general memory, or kickoff-output storage. It reads the crew knowledge query surface only.

Diagnose capability failures

Symptom	Decision
Capability is absent	Review descriptor and sidecar allowlists before enabling it
Query item is rejected	Remove control characters and stay within the item and array bounds
Result limit or threshold is rejected	Use an integer limit and numeric threshold inside the exact ranges
Query returns no results	Distinguish no knowledge object, no match, and provider behavior outside the connector
Query exposes unexpected data	Disable the route and reconcile tenant, crew, store, and credential boundaries
Reset target is not initialized	Verify the admitted crew configuration; do not broaden automatically to <code>all</code>
Reset fails after starting	Treat every selected domain as potentially changed
Reset waiter is cancelled	Ignore <code>dispatched: false</code> as proof and observe durable and provider state
Repeat conflicts	Preserve the Action ID and restore the exact original input and key
Output exceeds 4 MiB	Narrow the query or result limit; do not increase unrelated transport bounds

Related documentation

- [CrewAI capability index](#)
- [CrewAI connector overview](#)
- [Authentication and admitted crews](#)

- Approvals and policy
- Errors and retry decisions