

CrewAI capabilities

Use this index to choose the CrewAI operation that owns a task and interpret its AIP contract before invocation. The reviewed connector has ten frozen operations, but each configured crew publishes only its admitted subset.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/crewai/capabilities/README.md

The source-derived machine snapshot owns the ordered inventory. Focused pages own input details and task guidance; neither artifact replaces authenticated tenant discovery.

Catalogue identity

Property	Reviewed value
AIP source	97be86e9efedf07ecf1783b03800f683f107fb04
CrewAI source	bfa652a7be8637562cc9b0833f75d927a64552d1
CrewAI package in default product image	1.15.5
Connector profile	aip.connector.crewai.v1
Frozen operations	10
Low, medium, and high risk	3, 5, and 2
Mutations requiring approval	7
Operations requiring an idempotency key	8
Streaming operations	6
Operations advertising cancellation support	5

The catalogue count is not a manifest count. One fully enabled crew publishes ten capabilities, while a descriptor using compatibility defaults publishes four.

Read capability IDs

The `run` operation keeps the historical base capability:

```
TEXT
cap:crewai:<crew_id>
```

Every other operation appends its frozen suffix:

```
TEXT
cap:crewai:<crew_id>:<operation>
```

For a crew named `support-primary`, examples include:

```
TEXT
cap:crewai:support-primary
cap:crewai:support-primary:status
cap:crewai:support-primary:batch_run
cap:crewai:support-primary:memory_reset
```

Do not append `:run` to the base ID. Do not derive IDs from Python module names, CrewAI object names, sidecar URLs, or provider credentials.

Choose a family

Family	Focused page	Operations	Primary use
Run lifecycle and replay	<code>run-lifecycle-and-replay.md</code>	<code>run</code> , <code>status</code> , <code>events</code> , <code>cancel</code> , <code>batch_run</code> , <code>replay</code>	Execute work and control or inspect its durable job
Training and testing	<code>training-and-testing.md</code>	<code>train</code> , <code>test</code>	Create a training artifact or evaluate a crew
Knowledge and memory	<code>knowledge-and-memory.md</code>	<code>knowledge_query</code> , <code>memory_reset</code>	Read indexed knowledge or reset an explicit memory domain

Select the focused page before constructing input. Operations sharing the same sidecar job route still have different schemas and safety contracts.

Compare all operations

Operation	Kind	Risk	Mutation	Approval	Key	Stream	Cancel	Retry
<code>run</code>	Agent	Medium	Yes	Yes	Required	Yes	Yes	No
<code>status</code>	Tool	Low	No	No	Optional	No	No	Yes
<code>events</code>	Tool	Low	No	No	Optional	Yes	No	Yes
<code>cancel</code>	Tool	Medium	Yes	Yes	Required	No	No	No
<code>batch_run</code>	Agent	Medium	Yes	Yes	Required	Yes	Yes	No
<code>replay</code>	Agent	Medium	Yes	Yes	Required	Yes	Yes	No
<code>train</code>	Workflow	High	Yes	Yes	Required	Yes	Yes	No
<code>test</code>	Workflow	Medium	Yes	Yes	Required	Yes	Yes	No
<code>knowledge_query</code>	Tool	Low	No	No	Required	No	No	Yes
<code>memory_reset</code>	Tool	High	Yes	Yes	Required	No	No	No

`Cancel` in the table reflects published connector implementation support. It does not assert that a CrewAI provider method or every resulting side effect stopped.

The explicit `cancel` capability requests cancellation of another run. The cancel request itself is not a cancellable long-running job.

Streaming support means the connector can publish ordered job events and a terminal result. It does not mean every underlying CrewAI method can be interrupted.

Understand discovery projection

The host starts from a public descriptor for each crew. Its `allowed_operations` set controls which capability objects enter the manifest.

Descriptor policy	Published surface
Explicit subset	Exactly that validated subset
Missing legacy field	<code>run</code> , <code>status</code> , <code>events</code> , and <code>cancel</code>
Empty set	Startup failure
Unknown operation	Descriptor deserialization failure
Crew absent from sidecar health	Host remains unready

Descriptor policy	Published surface
Descriptor operation absent from sidecar health	Host remains unready

Runtime discovery is also scoped by tenant, connector admission, route eligibility, instance state, artifact identity, and manifest limits. A well-formed capability ID alone does not authorize a call.

Interpret methods and fixed sidecar paths

Operation	Sidecar method	Sidecar path
<code>run</code>	POST	<code>/jobs</code>
<code>status</code>	GET	<code>/jobs/{run_action_id}</code>
<code>events</code>	GET	<code>/jobs/{run_action_id}/events</code>
<code>cancel</code>	POST	<code>/jobs/{run_action_id}/cancel</code>
Remaining startable operations	POST	<code>/jobs</code>

Streaming startable operations use `/jobs/stream`. Event replay uses its own event path with an SSE `Accept` header.

At this revision, published binding metadata still assigns `/jobs/stream` to every streaming operation, including `events`. Runtime code ignores that value for event replay and calls `/jobs/{run_action_id}/events`; the machine snapshot retains the published metadata.

These paths are private bindings between the Rust host and the frozen sidecar. AIP callers invoke capability IDs through the admitted host route.

Apply approval and idempotency

Every mutation requires tenant-policy approval. The published approval window is 900,000 milliseconds and requires a reason, input snapshot, and policy decision.

Contract	Read-only status or events	Knowledge query	Mutation
Approval	No	No	Required
Idempotency key	Optional	Required	Required
Key scope	Tenant	Tenant	Tenant
Collision behavior	Revalidate input hash	Revalidate input hash	Revalidate input hash
Published key TTL	86,400,000 ms	86,400,000 ms	86,400,000 ms
Connector retry support	Yes	Yes	No

`knowledge_query` is read-only for provider state but starts a durable sidecar job. That coordinator boundary requires a key.

The reviewed failure mapper still sets `retryable: false` on every concrete connector failure. The table describes published support and safety, not an automatic retry instruction.

An idempotency key identifies one intent. It does not prove exactly-once model, tool, memory, knowledge, or external-provider effects.

Interpret execution and cancellation

`run`, `batch_run`, `replay`, `train`, and `test` support blocking and streaming completion. `events` supports blocking JSON replay and SSE replay.

All five streamed mutations advertise sidecar cancellation. A stream failure can therefore trigger a remote cancellation attempt for any of them.

Replay, training, and testing execute synchronous CrewAI methods in worker threads. Their common cancellation route cancels the asyncio waiter, records `cancelled`, and can return `remote_stop_confirmed: true` while provider work continues.

Cancelling an `events` stream only detaches the read. It does not change the target job.

Apply data and side-effect rules

Every capability declares confidential data, possible personal information, and required redaction.

Read-only operations declare read and external-network side effects. Mutations declare read, write, external-network, and code-execution effects.

Every mutation publishes `rollback_not_supported`. The connector exposes no transaction, reconciliation, compensation capability, or provider rollback.

Crew factories can invoke deployment-authored Python, models, tools, storage, and networks. Capability risk does not replace workload isolation or least- authority provider credentials.

Use the machine snapshot

The JSON asset contains source identities, aggregate counts, contract defaults, and one ordered record for each operation. Each record includes its capability template, family, method, private route, kind, risk, mutation, approval, idempotency, execution, input summary, and side effects.

Its reviewed SHA-256 is:

```
TEXT
d004e0dd77115304dfdf105817efaf630138d6b6d383147518271c14e61f359c
```

The snapshot is a source-derived review artifact. It does not prove host admission, tenant authorization, artifact qualification, or live provider compatibility.

Choose the next page

- Use `run-lifecycle-and-replay.md` for execution, status, event, cancellation, batch, and task-replay inputs.
- Use `training-and-testing.md` for bounded iterations, artifact names, and evaluator selection.
- Use `knowledge-and-memory.md` for queries, thresholds, result limits, and destructive reset domains.
- Use the run and recovery guides for procedures rather than field lookup.

Related documentation

- [Source-derived capability snapshot](#)
- [CrewAI connector overview](#)
- [CrewAI connector configuration](#)
- [Capabilities and contracts](#)
- [Approvals and policy](#)