

Troubleshoot the CrewAI connector

Use this index to move from one observed CrewAI connector symptom to a bounded check and safe response. Preserve the original Action, host runtime, and sidecar journal before restarting, retrying, archiving, or replacing state.

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

The connector marks every reviewed `ConnectorFailure` as non-retryable. A temporary category or HTTP status does not authorize automatic replay.

Capture the incident identity first

Record:

- UTC start time and first observed symptom;
- tenant, connector type, version, instance, and replica;
- host and sidecar image digests;
- AIP source and CrewAI method-verification revisions;
- configuration and credential revisions;
- crew ID, capability ID, operation, and Action ID;
- sidecar status, event sequence, and request hash when available;
- host `/ready`, metrics, registry lease, and route assignment;
- provider, model, tool, and artifact identifiers;
- every operator action already taken.

Do not copy Bearer tokens, signing seeds, model credentials, prompts, or governed output into an incident ticket.

Use the symptom index

Symptom	First check	Containment	Next section
Sidecar will not start	Process error and mounted paths	Keep host out of routing	Startup failures
Host is alive but not ready	Structured host /ready	Stop new assignments	Readiness failures
Sidecar returns 401	Host and sidecar token identity	Quarantine the route	Authentication failures
Capability is absent	Descriptor and sidecar operation policy	Do not widen policy during incident	Discovery failures
Start returns 409	Original Action and request hash	Preserve existing record	Identity collisions
Start returns 507	Journal count, bytes, and persistence	Stop new job admission	Journal failures
Stream fails or closes early	Event sequence, byte limits, terminal event	Reconcile admitted work	Stream failures
Action times out or is cancelled	Execution mechanism and provider evidence	Assume effects may continue	Uncertain outcomes
Terminal job failed	Sidecar event and restricted logs	Preserve traceback securely	Provider failures

Diagnose startup failures

Message family	Source-backed cause	Safe correction
Registry reference is required or malformed	Missing value or not <code>module:attribute</code>	Mount the intended registry module and exact attribute
Registry missing, empty, or invalid	Attribute does not exist, is not a mapping, or has invalid factories	Correct code or registry identity; do not substitute an unrelated crew
Durable state file is required	Non-loopback deployment lacks <code>AIP_CREWAI_STATE_FILE</code>	Mount the protected persistent volume and configure the file
Journal already owned	Another process holds the companion lock	Find the active owner; never delete the lock while it runs
Journal path, mode, or size invalid	Symlink, non-regular file, mode other than <code>0600</code> , or configured bound exceeded	Preserve bytes, repair ownership or restore from verified state
Training directory required or unsafe	<code>train</code> enabled without protected storage	Provide a non-symlink directory without group or world write permission
Bearer authentication required	No valid token outside explicit loopback development	Project one owner-controlled token into host and sidecar
Runtime bound invalid	Concurrency, run, event, checkpoint, or state setting is outside code bounds	Restore a reviewed in-range configuration

If journal decoding or validation fails, preserve the file before repair. Starting with an empty journal can remove duplicate-work protection.

Diagnose host construction failures

Connector error	Meaning	Correction
<code>NoCrews</code>	Host descriptor array is empty	Provide one through 1,000 reviewed descriptors
<code>InvalidCrew</code>	Descriptor ID, name, capability ID, or allowlist is invalid	Correct the descriptor source and repeat admission review
<code>InvalidUrl</code>	Sidecar URL cannot be parsed	Supply one absolute valid URL
<code>InvalidBaseUrl</code>	Credentials, query, fragment, or insecure non-loopback HTTP violates policy	Remove forbidden URL parts and use private HTTPS
<code>InvalidClient</code>	Hardened HTTP client construction failed	Inspect TLS and client configuration at startup
<code>InvalidCredential</code>	Host token material is empty, contains controls, or is not UTF-8	Rotate the projected token file and restart deliberately

The host validates descriptors before registration. A startup correction can change the manifest digest, so re-run admission when capability material changes.

Diagnose discovery and readiness failures

The CrewAI connector readiness probe fails when:

- authenticated sidecar health cannot be read;
- sidecar `status` is not `ready`;
- `durable_journal` is not true;
- a configured crew is absent from sidecar health;
- a descriptor operation is absent from sidecar policy.

Compare the host crew file with the sidecar registry and `AIP_CREWAI_ALLOWED_OPERATIONS`. Keep the narrower policy during diagnosis. Widen it only through a reviewed configuration and admission change.

Host `/ready` can also fail because the replica drains, its lease expires, or PostgreSQL is not readable and writable. Diagnose those fields before changing CrewAI configuration.

Diagnose authentication failures

Sidecar HTTP 401 maps to `connector.crewai.authentication`. Confirm:

1. host and sidecar read the intended token projections;
2. neither projection changed only on one process;
3. the token is non-empty UTF-8 within 16 KiB;
4. the private ingress preserves the `Authorization` header;
5. old and new token windows follow the deployment rotation procedure.

Do not print the header or token. Compare secret versions or digests through an approved secret-management channel.

Sidecar HTTP 403 means the startable operation is disabled by sidecar policy. It maps to the same auth category at the Rust boundary, but token rotation will not repair it.

Diagnose configuration or input rejection

`connector.crewai.configuration_or_input` covers several local failures:

- capability ID is not configured;
- operation is disabled for the crew or execution path;
- `run_action_id` is missing or invalid;
- required idempotency metadata is absent;
- descriptor, URL, or client configuration is invalid.

Sidecar HTTP 400, 404, 409, 413, 416, 422, and 428 map to `connector.crewai.remote_rejected`. Inspect the retained sidecar response body and classify it with this table:

Status	Typical correction
400	Correct an invalid key or mismatched path and body Action ID
404	Use an admitted crew or an existing target Action ID

Status	Typical correction
409	Reuse exact request material or preserve the existing terminal record
413	Reduce encoded operation input below 1 MiB
416	Resume from a cursor between zero and current event count
422	Match the exact operation-specific input contract
428	Supply the required idempotency key or archive confirmation

Do not replace an Action ID to make a 409 disappear. The collision protects the original provider-effect boundary.

Diagnose journal failures

HTTP 507 occurs when retained jobs reach the configured count or the sidecar cannot persist the admission fence. It maps to `connector.crewai.remote_temporary` but remains non-retryable.

For capacity exhaustion:

1. stop new assignments;
2. record retained and maximum run counts;
3. identify only terminal records beyond the AIP replay-retention window;
4. retain their evidence externally;
5. archive each through exact Action confirmation;
6. verify the journal and health before restoring traffic.

For persistence failure, inspect volume capacity, inode capacity, ownership, mode, filesystem errors, temporary-file creation, replacement, and directory `fsync`. Preserve the last valid journal.

Never truncate or edit `runs.json` in place while the sidecar owns its lock.

Diagnose stream failures

`connector.crewai.invalid_stream` covers:

- transport chunks that are not valid SSE framing;
- non-UTF-8 frames or invalid JSON data;
- missing event name or sequence;
- a sequence gap or duplicate;
- more than 10,000 accepted events;
- an SSE buffer over 1 MiB;
- stream closure before a terminal event;
- downstream chunk-emission failure.

`connector.crewai.response_too_large` means a response or complete stream exceeded the host-configured byte bound.

After admitted streamed work fails, the connector attempts sidecar cancellation for five cancellable operations. A successful attempt preserves the original stream error; a failed attempt becomes `connector.crewai.cancelled_unconfirmed`.

Reconcile the target Action through its status and events. Do not create a new Action merely because the client lost the stream.

Diagnose timeout and cancellation

Sidecar HTTP 504 means the blocking route exceeded its wait duration. HTTP 502 means the durable job reached failed state before the blocking response.

Both map to a temporary connector category for invocation, with uncertainty set for 5xx. The connector still sets `retryable` to false.

For `cancelled_unconfirmed`, preserve:

- whether an HTTP response arrived before local cancellation;
- whether the operation used native async work or a worker thread;
- the sidecar record and terminal event;
- model, tool, training, or evaluator effects;
- any artifact created after the cancellation timestamp.

Replay, train, test, memory reset, and fallback paths can continue in worker threads after their asyncio waiter ends. A successful cancel route does not prove those calls stopped.

Diagnose terminal provider failure

A sidecar execution exception appends a `failed` event with:

- `status: failed`;
- Python exception class in `error_type`;
- a generic message directing the operator to logs by Action ID.

The blocking route returns HTTP 502. A stream returns a failed AIP result with code `connector.crewai.execution_failed` and the event details.

Review restricted sidecar logs for the matching Action ID. The log includes the provider exception and traceback, so treat it as confidential.

Do not infer retry safety from the exception class alone. First determine whether models, tools, memory, training artifacts, or downstream systems received effects.

Escalate with complete evidence

Escalate when identity cannot be reconciled, the journal is corrupted, a thread-backed call continues after containment, a stream violates sequence, or provider effects remain unknown.

Include:

- the incident identity captured at the start;
- exact AIP and CrewAI revisions;
- redacted host readiness and sidecar health;
- relevant metric deltas;
- sidecar record and event slice;
- restricted log location, not copied secret-bearing lines;
- provider and downstream evidence;
- containment and recovery decisions;
- a clear statement of what remains unproved.

Related documentation

- [CrewAI health, observability, and recovery](#)
- [CrewAI sidecar contract and lockfile](#)
- [CrewAI idempotency, cancellation, and process fencing](#)
- [CrewAI capability index](#)
- [Errors and retry decisions](#)