

Dify annotations and annotation reply

Use this family to list and maintain question-and-answer annotations for one Dify app. It also enables or disables annotation reply and exposes the provider background job that changes its search index.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/dify/capabilities/annotations.md

Annotation text and job errors can contain confidential data. Approval must bind every mutation to the selected app, exact content, intended index configuration, and retained recovery evidence.

Choose an operation

Operation suffix	Method and route	Purpose	Policy
<code>annotation.list</code>	<code>GET /v1/apps/annotations</code>	List or search app annotations	Low-risk read; retry supported
<code>annotation.create</code>	<code>POST /v1/apps/annotations</code>	Create one question-and-answer pair	Medium mutation; approval and key required
<code>annotation.update</code>	<code>PUT /v1/apps/annotations/{annotation_id}</code>	Replace both fields of one annotation	Medium mutation; approval and key required
<code>annotation.delete</code>	<code>DELETE /v1/apps/annotations/{annotation_id}</code>	Delete an annotation and its hit history	High mutation; approval and key required
<code>annotation_reply.toggle</code>	<code>POST /v1/apps/annotation-reply/{action}</code>	Start enable or disable work	High mutation; approval and key required
<code>annotation_reply.status</code>	<code>GET /v1/apps/annotation-reply/{action}/status/{job_id}</code>	Read provider job state	Low-risk read; retry supported

All six Tool operations are available in `workflow`, `completion`, `chat`, `advanced-chat`, `agent-chat`, and `agent` modes. They use the configured app Bearer credential and have no Dify end-user field.

The capability template is:

```
TEXT
cap:dify:<app_id>:<operation>
```

Use tenant discovery to select the exact app capability. Annotation and job identifiers never select a different credential or app.

List app annotations

`annotation.list` accepts `page`, `limit`, and `keyword` through Action query. Page and limit default to 1 and 20. The pinned provider requires each to be at least 1 and declares no smaller route-specific maximum.

```
JSON
{
  "query": {
    "page": 1,
    "limit": 20,
    "keyword": "refund policy"
  }
}
```

The response contains `data`, `has_more`, `limit`, `total`, and `page`. Each record can contain `id`, `question`, `answer`, `hit_count`, and a Unix `created_at` timestamp.

The connector permits at most 128 query keys and 256 scalar values per array. The provider performs keyword matching within the selected app.

Create or replace an annotation

Create and update both require a JSON `body` with `question` and `answer`. Update is a complete two-field replacement, not a partial patch.

```
JSON
{
  "body": {
    "question": "When can a refund be requested?",
    "answer": "Submit the request within the approved service window."
  }
}
```

Use `annotation_id` for update. The connector accepts a bounded path string, while the pinned provider route requires a UUID and edit permission.

Create returns status 201 with the annotation record. Update returns status 200 with the replaced record. The connector wraps either response in its generic JSON result.

Delete an annotation

`annotation.delete` requires `annotation_id`. The pinned provider checks edit permission, deletes the annotation and associated hit history, then returns an empty 204 response.

```
JSON
{
  "annotation_id": "0190c42f-2d5a-7000-8000-000000000301"
}
```

The connector represents the empty provider body as `null` inside the generic result. Deletion has no rollback contract, so preserve authorization and the approved record identity before dispatch.

Enable or disable annotation reply

Set `action` to `enable` or `disable`. The pinned provider validates the same three body fields for both actions:

```
JSON
{
  "action": "enable",
  "body": {
    "score_threshold": 0.82,
    "embedding_provider_name": "provider-name",
    "embedding_model_name": "embedding-model-name"
  }
}
```

The score threshold is a floating-point similarity boundary. Provider and model names identify the embedding configuration used to index annotation questions.

The disable implementation ignores those embedding values, but request validation still requires them at the pinned provider revision. Supply the known approved configuration instead of invented placeholders.

A successful toggle response contains `job_id` and `job_status`. The initial state is normally `waiting`. A repeated provider request during active work can return `processing` with the existing job ID.

The connector operation itself is synchronous because it receives this JSON acknowledgement. The provider indexing or deletion job continues in the background and has no connector cancellation capability.

Track the provider job

Call `annotation_reply.status` with the exact `action` and UUID `job_id` returned by the toggle:

```
JSON
{
  "action": "enable",
  "job_id": "0190c42f-2d5a-7000-8000-000000000302"
}
```

Known provider states are `waiting`, `processing`, `completed`, and `error`. An error response field can contain `error_msg`.

The status lookup key is built from action and job ID. Bind both values to the selected app in the caller's evidence rather than treating the job ID as an app-scoped authorization token.

At the pinned revision, completed and error status entries use a 600-second Redis retention period. Poll promptly and persist the terminal evidence needed by the caller. Do not treat a later missing job as proof of failure.

Apply mutation governance

The four mutations require a non-blank Action idempotency key of at most 512 bytes. AIP retains its tenant-scoped reservation for 24 hours and revalidates the input hash on collision.

That reservation does not prove provider deduplication. None of the mutations publishes safe retry, transaction, reconciliation, cancellation, or compensation support.

Treat questions, answers, hit counts, embedding choices, job identifiers, and error details as confidential PII-bearing data. Redact them from general logs and retain only the evidence required by policy.

Recover safely

Observation	Decision
Query, UUID, action, or body is rejected before dispatch	Correct the bounded request without claiming a provider effect
Create or update loses its response after dispatch	Treat the outcome as uncertain; inspect the annotation list before another mutation
Delete returns no usable response	Verify the record through a read before considering any replacement
Toggle returns a job ID	Record the exact action and job ID, then poll status
Toggle response is lost	Inspect provider configuration and retained evidence; do not replay automatically
Job remains waiting or processing	Continue bounded polling; no connector cancellation is available
Job reports error	Preserve the redacted error, repair configuration, and authorize a new Action
Job status disappears after terminal retention	Use retained terminal evidence and current provider configuration
Any record crosses an expected app boundary	Stop traffic and treat the event as an isolation incident

Preserve the Action ID, input digest, approval, idempotency key, app identity, annotation ID, action, job ID, and terminal provider evidence used to settle an ambiguous outcome.

Related documentation

- [Dify capability index](#)
- [Dify connector overview](#)
- [Dify connector configuration](#)
- [Authentication and credential scopes](#)

- Errors and retry decisions