

Dify segments and child chunks

Use this family to create, inspect, update, or delete document segments and their child chunks. The nine operations resolve every object through its workspace, dataset, document, and parent ownership chain.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/dify/capabilities/knowledge-segments-and-child-chunks.md

Chunk content, answers, attachments, summaries, and errors can contain confidential data. Approval must bind each mutation to the complete route identity and intended index change.

Choose an operation

Operation suffix	Method and route	Purpose	Policy
<code>segment.create</code>	POST /v1/datasets/{dataset_id}/documents/{document_id}/segments	Create one or more segments	Medium mutation; approval and key required
<code>segment.list</code>	GET /v1/datasets/{dataset_id}/documents/{document_id}/segments	List and filter segments	Low-risk read; retry supported
<code>segment.get</code>	GET /v1/datasets/{dataset_id}/documents/{document_id}/segments/{segment_id}	Read one segment	Low-risk read; retry supported
<code>segment.update</code>	POST /v1/datasets/{dataset_id}/documents/{document_id}/segments/{segment_id}	Update content, index, or availability	Medium mutation; approval and key required
<code>segment.delete</code>	DELETE /v1/datasets/{dataset_id}/documents/{document_id}/segments/{segment_id}	Delete one segment	High mutation; approval and key required
<code>child_chunk.create</code>	POST /v1/datasets/{dataset_id}/documents/{document_id}/segments/{segment_id}/child_chunks	Create one child chunk	Medium mutation; approval and key required
<code>child_chunk.list</code>	GET /v1/datasets/{dataset_id}/documents/{document_id}/segments/{segment_id}/child_chunks	List child chunks	Low-risk read; retry supported
<code>child_chunk.update</code>	PATCH /v1/datasets/{dataset_id}/documents/{document_id}/segments/{segment_id}/child_chunks/{child_chunk_id}	Replace child content	Medium mutation; approval and key required
<code>child_chunk.delete</code>	DELETE /v1/datasets/{dataset_id}/documents/{document_id}/segments/{segment_id}/child_chunks/{child_chunk_id}	Delete one child chunk	High mutation; approval and key required

These Tool operations have knowledge scope and no app-mode projection. The capability template is:

```
TEXT
cap:dify:knowledge:<credential_id>:<operation>
```

The provider requires UUID route IDs. It resolves a segment through the dataset and document, then resolves a child through that segment. Preserve the complete chain in approval and audit evidence.

Create document segments

The document must exist, be enabled, and have completed indexing. A high-quality dataset also requires its configured embedding model.

`segment.create` requires a non-empty `body.segments` array. Each item requires non-blank `content` and can include `answer`, `keywords`, and `attachment_ids`.

```
JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000601",
  "document_id": "0190c42f-2d5a-7000-8000-000000000602",
  "body": {
    "segments": [
      {
        "content": "Refund requests must arrive within thirty days.",
        "keywords": ["refund", "window"],
        "attachment_ids": ["attachment-0190"]
      }
    ]
  }
}
```

A `qa_model` document also requires a non-blank `answer` for every item. The provider attachment limit is configurable and defaults to ten per segment.

The provider setting `DATASET_MAX_SEGMENTS_PER_REQUEST` controls the item count. Its pinned default is zero, meaning no provider-specific count ceiling. Apply a tenant policy bound before dispatch.

Success returns `data` with created segments and the document `doc_form`. Segment fields include IDs, content, answer, keywords, token and word counts, index state, child chunks, attachments, and optional summary.

List or inspect segments

`segment.list` accepts `page`, `limit`, repeated `status`, and `keyword` through Action `query`. Page and limit must be at least 1; provider pagination caps the limit at 100.

```
JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000601",
  "document_id": "0190c42f-2d5a-7000-8000-000000000602",
  "query": {
    "page": 1,
    "limit": 20,
    "status": ["completed", "error"],
    "keyword": "refund"
  }
}
```

The list result contains `data`, `doc_form`, `total`, `has_more`, `limit`, and `page`. Results are ordered by segment position and ID.

Use `segment.get` with the full route chain for one record. Its result contains `data` and `doc_form`, including the summary when one exists.

Update one segment

`segment.update` wraps optional changes inside `body.segment`:

```
JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000601",
  "document_id": "0190c42f-2d5a-7000-8000-000000000602",
  "body": {
    "segment": {
      "content": "Refund requests must arrive within thirty days.",
      "keywords": ["refund", "window"],
      "attachment_ids": ["attachment-0190"]
    }
  }
}
```

```

"segment_id": "0190c42f-2d5a-7000-8000-000000000603",
"body": {
  "segment": {
    "content": "Refund requests must arrive within thirty calendar days.",
    "keywords": ["refund", "calendar-days"],
    "enabled": true,
    "regenerate_child_chunks": false,
    "attachment_ids": [],
    "summary": "Thirty-day refund request window."
  }
}
}

```

The object can contain `content`, `answer`, `keywords`, `enabled`, `regenerate_child_chunks`, `attachment_ids`, and `summary`. QA content changes should preserve the intended answer.

Setting `enabled` to false disables the segment and schedules index removal. A disabled segment cannot accept ordinary edits; enable it through an approved update first.

`regenerate_child_chunks` applies to a parent-child document and requires high-quality indexing. Content, keywords, attachments, and summary changes can also update vector or summary index state.

The pinned provider can catch an indexing failure and return a segment marked `error`. Inspect `data.status` and `data.error`; HTTP success alone does not prove a usable index.

Delete one segment

`segment.delete` requires the complete dataset, document, and segment chain. It removes the segment and its associated index state and returns empty status 204.

The provider prevents overlapping delete work with a short-lived cache key. The connector represents an empty successful body as `null` and provides no rollback contract.

Maintain child chunks

Use child operations for the explicit children below one known parent segment. Route ownership does not by itself prove that the document uses a parent-child processing form.

Create requires `body.content`:

```

JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000601",
  "document_id": "0190c42f-2d5a-7000-8000-000000000602",
  "segment_id": "0190c42f-2d5a-7000-8000-000000000603",
  "body": {
    "content": "Submit the request through the approved support channel."
  }
}

```

The provider creates vector state and returns one child record. An indexing failure rolls back that provider transaction and returns an error.

`child_chunk.list` accepts `page`, `limit`, and `keyword`. Page and limit must be at least 1, and limit is capped at 100. The result contains `data`, `total`, `total_pages`, `page`, and `limit`.

Update requires the full route chain and `body.content`. It replaces content, word count, update identity, and vector state. Delete removes the record and its vector entry, then returns empty status 204.

```

JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000601",
  "document_id": "0190c42f-2d5a-7000-8000-000000000602",
  "segment_id": "0190c42f-2d5a-7000-8000-000000000603",
  "child_chunk_id": "0190c42f-2d5a-7000-8000-000000000604",
  "body": {

```

```
    "content": "Submit the request through the verified support portal."
  }
}
```

Apply mutation governance

All six 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.

The key does not prove provider deduplication. These operations publish no automatic retry, transaction, reconciliation, cancellation, or compensation support.

Treat content, answers, keywords, attachments, summaries, vector identity, statuses, and errors as confidential PII-bearing data. Redact them from general logs.

Recover safely

Observation	Decision
Any route ID or body is rejected before dispatch	Correct the bounded request without claiming a provider effect
Document is incomplete or disabled	Finish indexing or authorize the intended enable transition first
High-quality model is unavailable	Repair the pinned workspace model before another mutation
Segment create response is lost	List segments and compare content before another create
Segment update returns status error	Preserve the error and inspect content and index state before repair
Child vector mutation fails	Read the same child route before authorizing another mutation
Delete response is lost	Read the exact ownership chain before considering reconstruction
List pagination reaches the cap	Continue by page without widening connector bounds
Returned content crosses an ownership boundary	Stop traffic and treat the event as an isolation incident

Preserve the Action ID, digest, approval, idempotency key, credential, complete route chain, returned status, and provider evidence needed to settle an ambiguous outcome.

Related documentation

- [Dify capability index](#)
- [Dify knowledge documents and indexing](#)
- [Dify connector configuration](#)
- [Authentication and credential scopes](#)
- [Errors and retry decisions](#)