

Dify models and knowledge metadata

Use this family to discover configured workspace models, define dataset metadata, enable built-in fields, or assign values to documents. All eight operations use one workspace-scoped knowledge credential.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/dify/capabilities/knowledge-metadata-and-models.md

Metadata can expose document identity, upload history, source, and business attributes. Approval must bind each mutation to exact field identities, documents, replacement semantics, and readback evidence.

Choose an operation

Operation suffix	Method and route	Purpose	Policy
<code>model.list</code>	GET /v1/workspaces/current/models/model-types/{model_type}	List configured models by type	Low-risk read; retry supported
<code>metadata.create</code>	POST /v1/datasets/{dataset_id}/metadata	Create one custom field	Medium mutation; approval and key required
<code>metadata.list</code>	GET /v1/datasets/{dataset_id}/metadata	List fields and usage counts	Low-risk read; retry supported
<code>metadata.update</code>	PATCH /v1/datasets/{dataset_id}/metadata/{metadata_id}	Rename one custom field	Medium mutation; approval and key required
<code>metadata.delete</code>	DELETE /v1/datasets/{dataset_id}/metadata/{metadata_id}	Delete a field and its values	High mutation; approval and key required
<code>metadata.builtin.list</code>	GET /v1/datasets/{dataset_id}/metadata/built-in	List built-in field definitions	Low-risk read; retry supported
<code>metadata.builtin.update</code>	POST /v1/datasets/{dataset_id}/metadata/built-in/{action}	Enable or disable built-ins	Medium mutation; approval and key required
<code>document.metadata.update</code>	POST /v1/datasets/{dataset_id}/documents/metadata	Update metadata for document entries	Medium 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 credential selects the workspace. Dataset and metadata route IDs must be provider UUIDs inside that workspace.

Discover available models

`model.list` accepts one of these `model_type` values:

- `text-embedding`
- `rerank`

- `llm`
- `tts`
- `speech2text`
- `moderation`

```
JSON
{
  "model_type": "text-embedding"
}
```

The result contains `data`, grouped by provider. Provider entries expose identity, localized labels, configuration status, and model records.

Use the returned provider and model identifiers when configuring knowledge bases or indexing. Discovery proves visibility at read time, not future quota, availability, quality, or successful invocation.

Create and list custom fields

`metadata.create` requires `body.type` and `body.name`. Type accepts `string`, `number`, or `time`.

```
JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000701",
  "body": {
    "type": "string",
    "name": "policy_region"
  }
}
```

The pinned service limits names to 255 characters. A name must be unique in the dataset and cannot equal a built-in field name. Apply a caller policy that also rejects blank or ambiguous names.

Creation returns status 201 with `id`, `type`, and `name`. Persist the returned ID because document binding uses field identity, not only its visible name.

`metadata.list` returns `doc_metadata` and `built_in_field_enabled`. Each custom entry contains `id`, `name`, `type`, and the count of bound documents.

Rename or delete a custom field

Rename requires `metadata_id` and a body `name`:

```
JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000701",
  "metadata_id": "0190c42f-2d5a-7000-8000-000000000702",
  "body": {
    "name": "approved_region"
  }
}
```

The provider migrates the metadata key across bound documents and keeps its value. The new name follows the same length, uniqueness, and built-in-name rules.

`metadata.delete` removes the field, its bindings, and its values from related documents. Success returns empty status 204 and has no rollback contract.

The pinned rename and delete helpers catch internal exceptions. A nominal response can therefore be insufficient evidence of the intended state. Always read `metadata.list` and affected documents after either mutation.

Manage built-in fields

`metadata.builtin.list` returns five static definitions:

Name	Type
document_name	string
uploader	string
upload_date	time
last_update_date	time
source	string

The definitions do not prove that built-in fields are enabled for the selected dataset. Read `metadata.list` for `built_in_field_enabled`.

Set `action` to `enable` or `disable`:

```
JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000701",
  "action": "enable",
  "body": {}
}
```

Enable backfills those five values on working documents. Disable removes the keys and marks the dataset setting false.

The pinned provider helper catches internal failures before returning `result: success`. Confirm the flag and sample affected documents after either action.

Update document metadata

`document.metadata.update` accepts `body.operation_data`. Each entry names one document, its field values, and whether the change is partial.

```
JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000701",
  "body": {
    "operation_data": [
      {
        "document_id": "0190c42f-2d5a-7000-8000-000000000703",
        "metadata_list": [
          {
            "id": "0190c42f-2d5a-7000-8000-000000000702",
            "name": "approved_region",
            "value": "emea"
          }
        ],
        "partial_update": true
      }
    ]
  }
}
```

Use the exact ID, name, and type from `metadata.list`. The provider uses the ID for its binding and the name for the stored document key. Mismatched identities can create misleading data.

With `partial_update: true`, Dify preserves unspecified metadata and bindings. With false, the default, it starts from an empty custom map and replaces the document's custom bindings with the supplied list.

When built-ins are enabled, the provider restores their current values after either mode. A custom update cannot remove those five system fields.

The provider commits each document entry separately. A later failure can leave earlier entries changed. The operation is not an atomic transaction across the whole array.

Apply a tenant-policy count bound because the provider schema declares no route-specific maximum for `operation_data`. Read every intended document after the call before reporting batch success.

Apply mutation governance

All five 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, cross-document transaction, reconciliation, cancellation, or compensation support.

Treat model inventory, field definitions, document values, upload identities, sources, counts, and provider errors as confidential PII-bearing data. Redact them from general logs.

Recover safely

Observation	Decision
Route ID, model type, or body is rejected before dispatch	Correct the bounded request without claiming a provider effect
Model is absent from discovery	Repair workspace provider configuration before selecting it
Create response is lost	List fields by exact name and type before another create
Rename returns no trustworthy record	Read the field list and affected document keys
Delete returns 204	Confirm the field and values are absent before closing the action
Built-in toggle returns success	Confirm the dataset flag and sample document values
Multi-document update fails partway	Read every intended document and reconcile only through new approvals
Field ID and name no longer match	Refresh metadata discovery before another document mutation
Returned data crosses a workspace or dataset boundary	Stop traffic and treat the event as an isolation incident

Preserve the Action ID, digest, approval, idempotency key, credential, dataset, field identities, document set, mode, and readback 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](#)