

Twenty capabilities

Use this index to choose the Twenty operation family that owns a task and interpret its AIP contract before invocation. The reviewed connector contains 22 fixed operations across records, workspace metadata, and OpenAPI documents.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/twenty/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
Provider mapping source	96a24563674313a3071d359bfccaf33d5e130ab8
Connector profile	aip.connector.twenty.v1
Capability prefix	cap:twenty:
Frozen operations	22
Reads and mutations	8 and 14
Low, high, and critical risk	8, 9, and 5
Approval required	14
Idempotency key required	14
Connector retry support	11

An operations file can admit any non-empty subset. The full source catalogue count is therefore not the capability count of every running manifest.

Read a capability ID

Every capability appends its exact operation suffix:

```
TEXT
cap:twenty:record.list
■      ■      ■ operation
■      ■■■■■ resource family
■■■■■■■■■■■■■■■■ connector namespace
```

Other examples are:

```
TEXT
cap:twenty:record.create_many
cap:twenty:metadata.update
cap:twenty:openapi.core
```

Do not derive a capability from a provider route or GraphQL operation. Use the admitted manifest or the source-derived snapshot.

Choose a family

Family	Focused page	Total	Read	Mutation	Primary use
Record query and aggregation	record-query-and-aggregation.md	4	4	0	List, get, detect duplicates, and group records
Record create, update, and batch upsert	record-create-update-and-batch-upsert.md	4	0	4	Create or update standard and custom records
Record delete, restore, and merge	record-delete-restore-and-merge.md	7	0	7	Change record lifecycle or consolidate duplicates
Workspace metadata	workspace-metadata.md	5	2	3	Read and change twelve fixed metadata resource types
OpenAPI documents	openapi-documents.md	2	2	0	Read current core and metadata descriptions
Total		22	8	14	

Select the focused page before constructing input. Similar HTTP methods do not make operation schemas or safety contracts interchangeable.

Compare all operations

Operation	Method	Kind	Risk	Approval	Key	Retry
<code>record.list</code>	GET	Tool	Low	No	Optional	Yes
<code>record.get</code>	GET	Tool	Low	No	Optional	Yes
<code>record.find_duplicates</code>	POST	Tool	Low	No	Optional	Yes
<code>record.group_by</code>	GET	Tool	Low	No	Optional	Yes
<code>record.create</code>	POST	Tool	High	Yes	Required	Yes
<code>record.create_many</code>	POST	Tool	High	Yes	Required	Yes
<code>record.update</code>	PATCH	Tool	High	Yes	Required	Yes
<code>record.update_many</code>	PATCH	Tool	High	Yes	Required	No
<code>record.soft_delete</code>	DELETE	Tool	High	Yes	Required	No
<code>record.soft_delete_many</code>	DELETE	Tool	High	Yes	Required	No
<code>record.destroy</code>	DELETE	Tool	Critical	Yes	Required	No
<code>record.destroy_many</code>	DELETE	Tool	Critical	Yes	Required	No
<code>record.restore</code>	PATCH	Tool	High	Yes	Required	No
<code>record.restore_many</code>	PATCH	Tool	High	Yes	Required	No
<code>record.merge</code>	PATCH	Tool	High	Yes	Required	No
<code>metadata.list</code>	GET	Tool	Low	No	Optional	Yes
<code>metadata.get</code>	GET	Tool	Low	No	Optional	Yes
<code>metadata.create</code>	POST	Tool	Critical	Yes	Required	No
<code>metadata.update</code>	PATCH	Tool	Critical	Yes	Required	No
<code>metadata.delete</code>	DELETE	Tool	Critical	Yes	Required	No
<code>openapi.core</code>	GET	Resource	Low	No	Optional	Yes
<code>openapi.metadata</code>	GET	Resource	Low	No	Optional	Yes

`record.find_duplicates` is a read despite its provider POST method. `record.update` advertises safe retry with its required idempotency key, while the filtered `record.update_many` operation does not.

`Retry` describes connector contract support. It does not authorize an application to repeat a call without checking Action identity, terminal state, input hash, idempotency ownership, and provider outcome.

Interpret operation kinds

The two OpenAPI operations are AIP Resources. The remaining 20 operations are Tools, including record and metadata reads.

The native AIP caller still submits Resource operations through the Action path implemented by the connector host. Kind describes capability semantics; it does not expose a caller-selected provider URL.

Apply approval and idempotency

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

Contract	Read	Mutation
Approval	No	Required
Idempotency key	Optional	Required
Key scope	External account	External account
Collision behavior	Revalidate input hash	Revalidate input hash
Published key TTL	604800000 ms	604800000 ms

The connector sends a mutation key to Twenty as `Idempotency-Key`. Record create also derives missing IDs from workspace, key, and record index and forces provider upsert.

An idempotency key identifies one intent. It does not prove that every provider endpoint deduplicates requests or make an uncertain mutation safe to repeat under a new identity.

Interpret execution and retry

All 22 operations support synchronous execution and advertise a 120,000 ms timeout hint. None supports connector-managed asynchronous execution, streaming, or cancellation.

All eight reads advertise retry support. Three mutations also advertise it:

- `record.create`;
- `record.create_many`;
- `record.update`.

Those mutations are safe only with the original idempotency key and unchanged canonical input. The remaining eleven mutations advertise unsafe retry.

The published expected latency, queue delay, and 99.9% availability value are contract metadata. They are not measured deployment SLOs or qualification results.

Apply data and side-effect rules

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

Read operations declare `read` and `external_network`. Create, update, restore, merge, and metadata mutations declare `write` and `external_network`. Delete operations declare `delete` and `external_network`.

Provider responses remain provider-shaped. Error projection retains only a bounded safe-field allowlist, and webhook metadata results recursively remove fields named `secret`.

Understand compensation and uncertainty

All 14 mutations declare `rollback_not_supported` and name no compensation capability. The connector also exposes no transaction or reconciliation operation.

Approval and idempotency do not make a mutation reversible. Preserve original Action, key, provider request identity, output, and receipts whenever a result is incomplete or uncertain.

The connector marks temporary provider failures and many post-dispatch transport or response failures as uncertain for mutations. Recovery must read actual provider state before another governed write.

Understand discovery projection

The host starts with all 22 operations or a non-empty configured subset. Only that set enters its manifest and invocation dispatcher.

Runtime discovery also depends on tenant, connector admission, route eligibility, instance lifecycle, artifact identity, and manifest limits. A well-formed capability ID does not authorize a call.

An operations file is process configuration. Replacing the file without restarting and re-admitting the host does not change the published surface.

Use the machine snapshot

The JSON asset contains source identities, aggregate counts, contract defaults, and one ordered record for every operation. Each record includes its capability ID, family, provider method and path template, kind, risk, mutation, approval, idempotency, execution, input summary, side effects, data, and compensation.

Its reviewed SHA-256 is:

```
TEXT
35315598e1e392eeeb3d3a0a194d37bfedd86d90da702a7bdc676762318fd465
```

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

Choose the next page

- Use `record-query-and-aggregation.md` for reads and duplicate or grouping input.
- Use `record-create-update-and-batch-upsert.md` for deterministic upsert and update behavior.
- Use `record-delete-restore-and-merge.md` for destructive lifecycle changes.
- Use `workspace-metadata.md` for the twelve supported metadata resource types.
- Use `openapi-documents.md` for current provider descriptions and drift use.
- Use the workflow guides for procedures rather than field lookup.

Related documentation

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