

Cal.diy conferencing

Use these capabilities to list installed conferencing applications, read or change the default, and connect or disconnect an app supported by the published business-operation surface. This page owns all five conferencing operations.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/cal-diy/capabilities/conferencing.md

All five use provider API version `2024-08-13`. Browser OAuth authorization and callback routes remain in the operator control plane and are not agent-callable capabilities.

Choose an operation

Capability	Provider request	Risk	Approval
<code>cap:cal_diy:conferencing.list</code>	<code>GET /v2/conferencing</code>	Low	No
<code>cap:cal_diy:conferencing.default.get</code>	<code>GET /v2/conferencing/default</code>	Low	No
<code>cap:cal_diy:conferencing.default.set</code>	<code>POST /v2/conferencing/{app}/default</code>	Medium	Required
<code>cap:cal_diy:conferencing.connect</code>	<code>POST /v2/conferencing/{app}/connect</code>	Medium	Required
<code>cap:cal_diy:conferencing.disconnect</code>	<code>DELETE /v2/conferencing/{app}/disconnect</code>	High	Required

The allowed `app` values differ by mutation:

App	Set default	Connect	Disconnect
<code>google-meet</code>	Yes	Yes	Yes
<code>zoom</code>	Yes	No	Yes
<code>msteams</code>	Yes	No	Yes
<code>daily-video</code>	Yes	No	No

Do not infer one operation's app set from another operation.

Discover installed apps and the default

Both reads accept an empty object:

```
JSON
{}
```

Use `conferencing.list` to inspect installed conferencing applications. Use `conferencing.default.get` to read the current default. Both are low-risk, approval-free reads at the capability level.

An installed entry or default selection does not prove that provider credentials are current, that OAuth has completed, or that meeting creation is healthy. The connector does not publish the provider response field shape.

Change the default app

Set the default by passing exactly one supported app:

```
JSON
{
  "app": "zoom"
}
```

The exact enum is `google-meet`, `zoom`, `msteams`, or `daily-video`. Additional input fields are rejected. `app` becomes the provider path segment, so the request has no JSON body after routing fields are removed.

The connector lists installed apps before dispatch. A successful preflight does not prove that the selected app appears in the list or can become the default; Cal.diy still validates the mutation.

Default set is medium risk. It requires approval and an idempotency key, does not require a plan, and declares no rollback capability.

Connect the supported non-OAuth app

The published connect operation accepts only:

```
JSON
{
  "app": "google-meet"
}
```

The connector describes this as connecting a supported non-OAuth conferencing application. `zoom`, `msteams`, and `daily-video` fail the connect input schema even though some appear in the default or disconnect enums.

OAuth authorization URL and callback routes are intentionally excluded because they establish browser sessions or credentials. Complete those flows through the deployment's operator-owned process, not an AIP action.

Connect first lists installed apps. It is medium risk and names `conferencing.disconnect` as best-effort compensation. Compensation requires a separate approval and is not invoked automatically.

Disconnect an app

Disconnect accepts `google-meet`, `zoom`, or `msteams`:

```
JSON
{
  "app": "msteams"
}
```

The connector lists installed apps before sending `DELETE` to the app-specific path. `daily-video` is not accepted by this capability.

Disconnect is high risk and destructive. It has no rollback contract. A later connect or OAuth flow is a new provider action and does not restore prior credential state or meeting history.

Shared read contract

The two reads are synchronous, retry-safe, and eligible for connector retry. Their idempotency key is optional, action-scoped, and collision-checked against the input hash. No capability-level human approval is required.

Shared mutation contract

All three mutations require:

Contract field	Value
Human approval	Required
Approval selector	Tenant policy
Approval validity window	900,000 ms
Approval evidence	Reason, input snapshot, policy decision
Idempotency	Required; external-account scope
Collision behavior	Revalidate input hash
Connector retry	Not supported
Retry safety	Unsafe
Execution	Synchronous only
Transaction modes	dry_run, plan, commit, reconcile
Plan before commit	Not required
Dry-run fidelity	Policy and schema only

Default set and connect are medium risk. Disconnect is high risk, destructive, and uses the destructive approval reason. None of the operations carries the `send_message` side-effect marker.

Only connect declares best-effort compensation. Default set and disconnect declare rollback as not supported.

Data and result contract

The family is classified as confidential. Its contract does not declare PII and does not require redaction. Deployment policy may still impose stricter handling based on provider output or tenant context.

The connector returns the validated provider JSON body. A non-empty string `status` is required. `data`, `pagination`, and additional provider fields are optional and provider-owned.

The common response scrubber recursively replaces secret-, key-, and token-named values with `[REDACTED]` even though redaction is not required by this family's data contract. Do not treat that targeted scrubber as a general confidential-data filter.

Failures and recovery

Failure	Relevant cause	Safe response
<code>connector.cal_diy.invalid_action</code>	Unsupported app or unknown input field	Choose an app from the exact operation enum
<code>connector.cal_diy.policy</code>	Mutation omitted its required idempotency key	Preserve approved input and add the missing key
<code>connector.cal_diy.idempotency_collision</code>	One key owns different mutation input	Stop and inspect the original action
<code>connector.cal_diy.in_flight</code>	A matching conferencing mutation is executing	Read durable state and wait
<code>connector.cal_diy.outcome_unknown</code>	Prior provider dispatch cannot be proved	Reconcile; do not create a replacement key
<code>connector.cal_diy.authentication</code>	Provider returned 401 or 403	Verify account binding and provider credentials
<code>connector.cal_diy.state_conflict</code>	Provider returned 409 or 412	Refresh installed and default state before review
<code>connector.cal_diy.rate_limited</code>	Provider returned 429	Retry reads within policy; reconcile mutations
<code>connector.cal_diy.remote_temporary</code>	Provider returned a server error	Retry reads within policy; reconcile mutations
<code>connector.cal_diy.transport</code>	Provider exchange failed	Retry reads only; treat dispatched mutations as uncertain
<code>connector.cal_diy.invalid_provider_output</code>	Successful response lacked valid <code>status</code>	Preserve evidence and reconcile a mutation

Permanent rejection, invalid credentials, oversized responses, and durable settlement failures use the common Cal.diy error family. Gateway and lifecycle errors are described in the [global error reference](#).

Verify conferencing state

For a read, retain the capability, actor, tenant, external account, response status, and provider request id when present.

For a mutation, also retain the exact app, approval evidence, and original idempotency key. Wait for a completed durable result, then list apps and read the current default. For uncertain dispatch, reconcile the original operation before another mutation.

These checks validate application handling. They do not prove OAuth completion, provider credential health, successful meeting creation, invitation delivery, live-provider qualification, or production readiness.

Related documentation

- [Cal.diy capability index](#)
- [Booking attendees and artifacts](#)
- [Authentication and tenant routing](#)
- [Approvals and policy](#)
- [Error reference](#)