Run workflows from *code* and agents

Run workflows and Apps through the API and the Scenario MCP: find the input names, quote with a dry run, start the job, wait, read results and answer approvals.

GUIDE Step 2 · Repeat

IN SHORT

- Run any workflow or App from a script or an AI agent by sending the same inputs a person would fill in.
- Quote every run with a dry run first, and sum the child jobs to report what a run cost.
- Limits: 100 requests per minute, 10 concurrent workflow jobs and 50 nodes per workflow.

You need ✓ An API key and secret, or the Scenario MCP connected to your agent ✓ A project ID ✓ The ID of the workflow or App

The workflow built in the editor does not change: your script, pipeline or agent fills the same inputs and gets a job back.

As of October 2026

The models, settings and screens in this article are the ones Scenario offered in October 2026. New models and updates arrive often, so check the app for newer options before you choose.

Run a workflow from code

1 **Find inputs** workflow ID and input names → 2 **Upload** files become asset IDs → 3 **Dry run** a quote before running → 4 **Run** one workflow job → 5 **Wait** poll, webhook or `jobs_wait` → 6 **Answer reviews** approve, reject or select

Quick version

1. Copy the workflow ID (**...** > **Copy Workflow ID**) and read its input names with `GET /v1/workflows/{id}`.
2. Upload each file input and keep its `asset_...` ID.

3. Send `PUT /v1/workflows/{id}/run` with `dryRun=true` and read the quote.
4. Send the same request without `dryRun`, then poll `GET /v1/jobs/{jobId}` until it ends.
5. Sum the child jobs in `metadata.flow` to know what the run cost.

An App is a workflow with status `ready`. Platform Apps are public; your own workflows and Apps are private to their project. Both run the same way.

The same five steps through the API or the Scenario MCP:

Step	API	Scenario MCP
1. Get the workflow ID and input names (such as `image1`); upload files, since file inputs take asset IDs	`GET /v1/workflows/{id}`, `POST /v1/assets`	`workflow_get`, `upload_asset`
2. Quote with a dry run	`PUT .../run` with `dryRun=true` (status `269` with the quote)	`workflow_run` with `dry_run: true`
3. Run with the same body	`PUT /v1/workflows/{id}/run`	`workflow_run`
4. Wait until success or failure	`GET /v1/jobs/{jobId}` or a webhook	`jobs_wait`
5. Answer reviews, only if the run pauses	`PUT` user-approval or user-selection	`workflow_approve`, `workflow_reject`

The parent job's `cuCost` reads 0: the cost is the sum of the child jobs in `metadata.flow`.

1. Get the workflow ID and input names

- **Your workflows:** in the editor, open **...** and choose **Copy Workflow ID**.
- **Public workflows** have IDs too. Angle Multiplier is `wflow_angle-multiplier`.

Each input has a name, such as `image1` or `text1`, which is what you send. Read them from the workflow definition:

Shell

```
curl -s "https://api.cloud.scenario.com/v1/workflows/wflow_angle-multiplier?projectId=" \
  -u ":"
```

Through the Scenario MCP, ask your agent to call `workflow_get`; it returns the inputs definition and the editor graph. For Angle Multiplier there is one required input: `image1` ("Upload Product", an image).

File inputs take an asset ID, so upload your file first (`POST /v1/assets` , or `upload_asset` in the Scenario MCP) and keep the `asset_...` ID.

2. Quote it with a dry run

A dry run returns a quote for the run without running it. On Enterprise, an IP Detection check requested in the same call is still charged.

Shell

```
curl -s -X PUT "https://api.cloud.scenario.com/v1/workflows/wflow_angle-multiplier/run?projectId=&dryRun=true" \
  -u ":" \
  -H "Content-Type: application/json" \
  -d '{ "image1": "asset_..." }'
```

A dry run answers with HTTP status `269` and the cost in Compute Units: the status means "estimate", not an error, so do not treat it as a failure. `dryRun=api` estimates at the API tier.

Through the Scenario MCP, the same quote looked like this on October 12, 2026:

JSON

```
// workflow_run  { "workflow_id": "wflow_angle-multiplier",
//                 "inputs": { "image1": "asset_SstohPMc..." }, "dry_run": true }
{ "creativeUnitsCost": 13.75, "creativeUnitsDiscount": 0, "job": {} }
```

The quote is in Compute Units (CU), the unit credits are metered in. Budget with it.

3. Run it

Send the same request without `dryRun` . Inputs go directly in the body, keyed by input name.

Shell

```
curl -s -X PUT "https://api.cloud.scenario.com/v1/workflows/wflow_angle-multiplier/run?projectId=" \
  -u ":" \
  -H "Content-Type: application/json" \
  -d '{ "image1": "asset_..." }'
```

The response is a workflow job. Its `metadata.flow` lists each node with its own status and, for model nodes, its own child job. Trimmed from the real Scenario MCP run:

JSON


```
{ "job": { "jobId": "job_BfM1ywYj...", "jobType": "workflow",
  "metadata": { "flow": [
    { "id": "imageGenerator1", "type": "custom-model", "status": "processing",
      "modelId": "model_openai-gpt-image-2-5-sunburst", "jobId": "job_HEmqpWhU..." },
    { "id": "text1", "type": "transform", "status": "success",
      "output": "Generate multiple viewing angles of the product shown in this image..." } ],
    "input": { "image1": "asset_SstohPMc..." } } } }
```


4. Wait for the result

Poll `GET /v1/jobs/{jobId}` or use webhooks (see Jobs, polling and webhooks).

Job statuses

1	2	3	4	5	6
pending accepted	**queued** waiting for capacity	**warming-up** model loading	**in-progress** generating	**finalizing** saving assets	**success** or `failure`, `canceled`

In the Scenario MCP, `jobs_wait` does the polling:

JSON


```
// jobs_wait  { "job_ids": ["job_BfM1ywYj..."] }
{ "status": "completed", "jobs": [ { "jobId": "job_BfM1ywYj...", "jobType": "workflow",
  "status": "success", "assetIds": ["asset_kygugJ6u..."], "cuCost": 0 } ] }
```


The run took about 23 seconds. The output asset is the six-angle grid shown in Platform workflows and templates.

> **Note**
>
> The workflow job's own `cuCost` reads 0. The charge sits on the child jobs listed in `metadata.flow`: here the Image Generator's job carried the whole charge. To report what a run cost, sum the child jobs.

Through an AI agent

With the Scenario MCP connected, an agent can do all of the above from a plain request:

Prompt

```
Run the Scenario App "Angle Multiplier" on the lantern photo I uploaded. Quote the cost
first and wait for my go. Then show me the result and tell me what it cost.
```

Step	Scenario MCP tool
Find it (Apps are ready workflows)	`workflows_list`
Read the inputs	`workflow_get`
Quote	`workflow_run` with `dry_run`
Start	`workflow_run`
Wait	`jobs_wait`
Answer approval nodes	Catalog tools `workflow_approve` and `workflow_reject`
Copy a workflow into your project	`workflow_copy`

The `scenario-workflows` skill teaches agents the run procedure; `scenario-workflow-authoring` covers building graphs. See Scenario MCP tools reference and Scenario's skills.

Answer review steps from code

A run that reaches an **Approval** node or **Slice Items** in Choose mode waits until someone answers.

Node	Endpoint	Body
Approval	`PUT /v1/workflows/{workflowId}/user-approval`	`workflowJobId`, `nodeId`, `action`: `approve` (default) or `reject`
Slice Items, Choose	`PUT /v1/workflows/{workflowId}/user-selection`	`workflowJobId`, `nodeId`, `action`: `select` with `selectedIndices` (zero-based), or `reject`

A rejected stand-alone node cancels the run. A rejection inside a Loop iteration cascades to the surrounding loop.

Limits

As documented for the API:

Limit	Value
Requests	100 per minute per project

Limit	Value
Concurrent workflow jobs	10 per project
Nodes	50 per workflow

For large batches, queue your inputs and keep at most 10 runs in flight.

Check before you accept

Accept when

the dry run quote matches the editor's badge for the same inputs; the job ends in `success` with the expected number of assets; the summed child-job cost is at or under the quote; outputs land in the project you passed.

Reject when

you started a batch without a dry run; an input name in your body does not match the definition, so the node ran empty; your script treats the dry-run status `269` as a failure; you report the parent job's `cuCost` of 0 as the cost.

If it breaks

Symptom	Cause and fix
400 Bad Request	An input name is wrong or a required input is missing. Compare your body with the inputs from `workflow_get`
401 or 403	The key is missing, wrong, or lacks access to the project. See API keys and authentication
404	The workflow ID is wrong, or the workflow belongs to another project
The workflow cannot run	Workflows in `draft` status do not execute; it must be `ready`
The job waits forever	It is paused at a review node. Answer it with the endpoints above
Runs queue up	You hit the 10-concurrent limit. Throttle your client

Related

What the Scenario MCP is, and how to connect it

Platform workflows and templates

Approval and review steps

For Developer Page 14 of 14 in Workflows and Apps

Verified in Scenario on Oct 12, 2026.