



SAMSARA MACHINE VISION

VS2 API Integrations Guide



Introduction

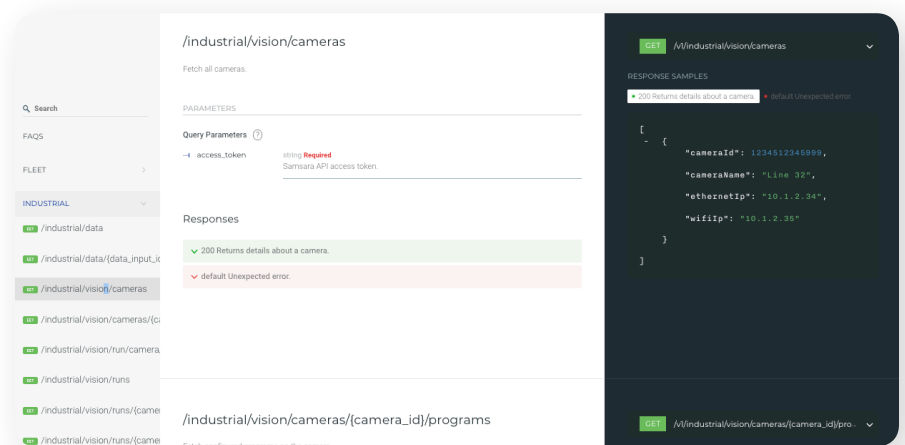
Production lines require systems that perform different functions in order to run properly. Many times, the systems in place need to communicate with one another so that tasks like rejecting defects are possible. While it is possible to physically wire connections between the VS2 camera and third party systems, like break-beam triggers and rejection systems, some integrations require a programmatic approach. In these cases, our Application Programming Interface (API) can be used to connect VS2 functionality into workflows triggered from external systems like a Manufacturing Execution System (MES).

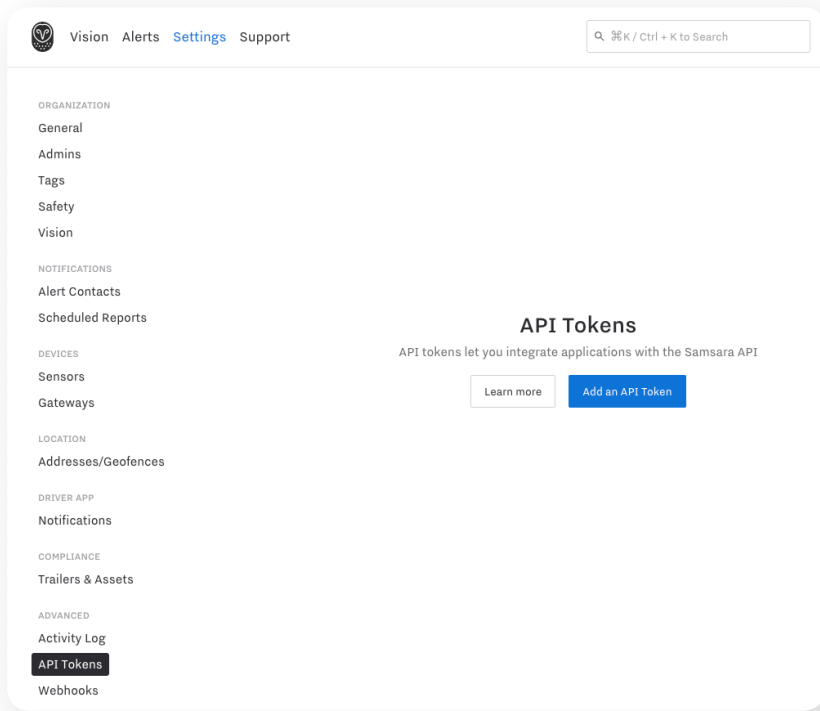
This guide will walk through how to use the VS2 APIs to build automated workflows that can be executed externally. The VS2 APIs can enable workflows such as starting runs and querying for results. However, the system needs to be configured and programmed ahead of time in order to leverage this functionality. For more information, please refer to the VS2 deployment guide.

There are two types of API for the Samsara VS2 product: Samsara Cloud APIs and VS2 Controller APIs. They call into different endpoints and have different use cases. Both APIs are REST APIs that can be called through standard programmatic means.

Samsara Cloud API

The Samsara Cloud APIs are for more information-centric use cases like getting camera details, program details, and results. For full documentation on the Samsara Cloud APIs for VS2, please refer to the online API documentation at www.samsara.com/api. VS2 APIs can be found under the Industrial section and begin with “/industrial/vision”.





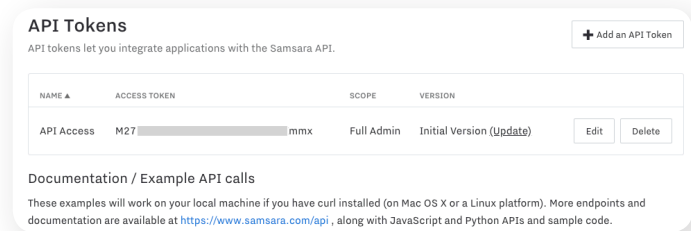
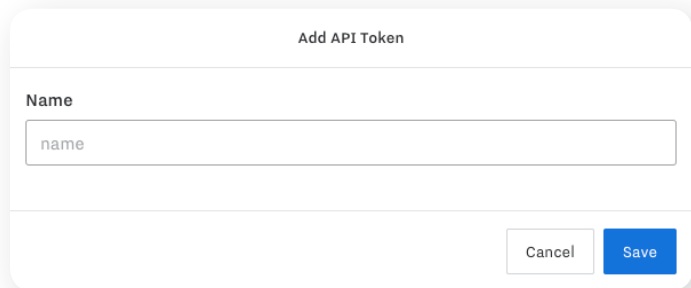
Setup - Creating an API Token

To communicate with the Samsara Cloud APIs, you will need to first create an API token to use. You can find this by going to Settings > API Tokens.

After clicking “Add an API Token”, you will be presented with a pop up module to name your API token. You can choose any name you like, but for logging purposes the best practice would be to name related to the application.

Once you have created your API token, the settings screen should now show it listed. From this point on, you can keep track of and manage your API tokens from this screen.

To call any of the Samsara Cloud APIs, you will need to include the Access Token listed above as a parameter.



Using Cloud APIs

In this section we will go through several examples of retrieving data from the Samsara Cloud. The most basic example will be to retrieve the cameras associated with your organization. To do this, you will need to call:

```
GET https://api.samsara.com/v1/industrial/vision/cameras?access_token={ACCESS_TOKEN}
```

This will return a JSON file with cameras currently linked to your organization in the format as described on <https://www.samsara.com/api#operation/getCameras>

Some commonly used APIs are:

- Get Programs for a given camera: `/industrial/vision/cameras/{camera_id}/programs`
- Get Run Results High Level Statistics: `/industrial/vision/runs`
- Get Detailed Run Results: `/industrial/vision/runs/{camera_id}/{program_id}/{started_at_ms}`

Some helpful tips with using the cloud APIs:

- All the time/date inputs and outputs use UNIX epoch time in milliseconds. For more information and help with conversion, you can use the website <https://currentmillis.com/>
- You should call our GET APIs to retrieve the appropriate ID of VS2 objects, such as program or camera.

Samsara VS2 Controller API

The Samsara VS2 Controller APIs can be used to programmatically control the local device to enable workflows such as starting and stopping runs during changeovers. The API endpoint will be a local IP address of the VS2 Controller so the system that will integrate with the VS2 Controller will need to have the ability to reach the IP address. To determine the IP address of the Controller, navigate to Settings > Gateways.

The screenshot shows the 'Gateways' management page in the Samsara VS2 Controller interface. The page has a sidebar on the left with navigation options: ORGANIZATION (General, Admins, Tags, Safety, Vision), NOTIFICATIONS (Alert Contacts, Scheduled Reports), DEVICES (Sensors, Gateways), and LOCATION. The main content area is titled 'Gateways' and includes a search bar, a 'Tags' dropdown, and a table of gateway devices. Each device row has 'Settings' and 'Remove' buttons. At the top right of the main area are '+ Activate Devices' and 'More Actions' buttons.

NAME	PRODUCT	ID	IP ADDRESS	MAC ADDRESS	ACTIONS
Line_1	VS2	GDH3-8A2-RHJ	10.6.32.9 (wifi)	c2:d3:91:e6:a9:4b (wifi), c0:d3:91:e6:a9:4b (eth)	Settings Remove
Line_4-Master	VS2	GUZW-EGY-7EE	10.6.32.48 (wifi)	c2:d3:91:e6:a9:3f (wifi), c0:d3:91:e6:a9:3f (eth)	Settings Remove
Line_5	VS2	G6VP-TJ6-J2G	10.6.32.40 (wifi)	c2:d3:91:e6:a9:37 (wifi), c0:d3:91:e6:a9:37 (eth)	Settings Remove

The controller APIs that are available to use are detailed below:

Start a Run

This API starts a run of a program that has been configured on the VS2 device. It expects a JSON parameter that includes the program ID or program name to start and expected results to read.

Endpoint	/v1/runs/start
Server	VS2 Controller IP or Domain Name
Port	3040
Protocol	HTTP
Method	POST
Example URL	http://10.1.1.100:3040/v1/runs/start
Post body (as raw JSON)	<pre>{“program_name”: “New Program”, “program_id”: 1, “environment_ tags”:{“barcode_match_criteria”:”<EXPECTED_BARCODE>”}}</pre>
	Notes: Only one of program_name and program_id is required Currently, we only support passing in barcode match criteria
Response — Success	200
Sample Success Body	<pre>{ “program_id”: 1, “program_name”: “Label Match”, “started_at_ms”: 1553728412880 }</pre>
	The “started_at_ms” field will return the start time of the program in unix epoch time (https://currentmillis.com/)
Response — Failure	500 - Error starting run. This error indicates that there is already a Run in progress. Stop the current Run and try again.

Current Run

This API returns the current run in progress. The return will include a JSON that includes basic run details.

Endpoint	/v1/runs/current
Server	VS2 Controller IP or Domain Name
Port	3040
Protocol	HTTP
Method	GET
Example URL	http://10.1.1.100:3040/v1/runs/current
Response — Success	200
Sample Success Body	<pre>{ "program_id": 1, "program_name": "Label Match", "started_at_ms": 1553728412880 }</pre>
	The "started_at_ms" field will return the start time of the program in unix epoch time (https://currentmillis.com/)
Response — Failure	500 Will return a 200 - null when no run is in progress.

Trigger Camera

This API call will trigger the camera whether a run is currently in progress or not.

Endpoint	<code>/v1/camera/trigger</code>
Server	VS2 Controller IP or Domain Name
Port	3040
Protocol	HTTP
Method	POST
Example URL	<code>http://10.1.1.100:3040/v1/camera/trigger</code>
Response — Success	200 - Check HMI or camera to verify result was captured.
Response — Failure	500 Will return a 200 when no run is in progress.

End a Run

This API will stop a run if one is in progress. If no run is in progress, you will receive a null return. If a run was stopped, you will receive the run details.

Endpoint	/v1/runs/end
Server	VS2 Controller IP or Domain Name
Port	3040
Protocol	HTTP
Method	POST
Example URL	http://10.1.1.100:3040/v1/runs/end
Response — Success	200
Sample Success Body	<pre>{ "program_id": 1, "program_name": "Label Match", "started_at_ms": 1553728412880 }</pre>
	The "started_at_ms" field will return the start time of the program in unix epoch time (https://currentmillis.com/)
Response — Failure	500 Will return a 200 - null when no run is in progress.

Retrieve Local Results

This API will return the last 1000 results from the camera. For historical results beyond the previous 1000 results, please use the Cloud API: <https://www.samsara.com/api#operation/getVisionRunsByCamera>

Endpoint	/v1/results
Server	VS2 Controller IP or Domain Name
Port	3040
Protocol	HTTP
Method	GET
Parameters	Limit: Integer number from 1 to 1000. Limits the Run Results. Default value is 1000 Include: String value that accepts pass reject no_read. Filters the Run Results
Example URL	http://10.1.1.100:3040/v1/results?limit=2&include=pass
Response — Success	200 - Check HMI or camera to verify result was captured.
Sample Success Body	<pre>[{ "programId": 23, "programName": "try to crash", "stepResults": [{ "name": "Count Shapes", "result": "PASS", "findShapes": { "minCount": 1, "maxCount": 1, "foundCount": 1 } }], "result": "PASS", "capturedAtMs": 1562641162866 ... }]</pre>

```

    },
    {
      "programId": 23,
      "programName": "try to crash",
      "stepResults": [
        {
          "name": "Count Shapes",
          "result": "PASS",
          "findShapes": {
            "minCount": 1,
            "maxCount": 1,
            "foundCount": 1
          }
        }
      ],
      "result": "PASS",
      "capturedAtMs": 1562641161868
    }
  ]

```

For full "StepResults" response template refer to response sample in the Cloud API <https://www.samsara.com/api#operation/getVisionRunsByCameraAndProgram>

Response — 500
Failure Will return a 200 - null when no run is in progress.

Results API Webhook

Another way to receive results data from the local controller is to set up an API webhook. This differs from the previous API in that it follows a "push model" rather than "pull model". When setting up a program, you have the option to configure when to send results outbound to a target URL endpoint:

When any of the conditions are met, we will automatically push a results API message to the URL destination. It will follow the same format as the results API output, as seen below.

Output

Webhook ▾

Conditions

1

consecutive rejects

1

consecutive no-reads

1

consecutive passes

Webhook Settings

Delay (ms)

0

URL

```
Sample Success    {
Body              "programId": 18,
                    "triggerType": "MANUAL",
                    "programName": "rubix cubes",
                    "stepResults": [
                      {
                        "name": "Count Shapes",
                        "result": "REJECT",
                        "findShapes": {
                          "minCount": 48,
                          "maxCount": 57,
                          "foundCount": 0
                        }
                      },
                      {
                        "name": "Color Match",
                        "result": "PASS",
                        "presenceAbsence": {
                          "score": 100,
                          "threshold": 70
                        }
                      }
                    ],
                    "status": "REJECT",
                    "happenedAtMs": 1562049417461
                  }
```

For full "StepResults" response template refer to response sample in the Cloud API <https://www.samsara.com/api#operation/getVisionRunsByCameraAndProgram>



samsara

samsara.com/industrial