

---

# Globus Automate Client

**Globus**

**Mar 16, 2023**



## GETTING STARTED

<b>1</b>	<b>Quick Start</b>	<b>3</b>
<b>2</b>	<b>Using the CLI</b>	<b>5</b>
<b>3</b>	<b>License</b>	<b>15</b>
<b>4</b>	<b>Running a flow automatically</b>	<b>17</b>
<b>5</b>	<b>CLI Reference</b>	<b>21</b>
<b>6</b>	<b>globus-automate</b>	<b>23</b>
<b>7</b>	<b>Python SDK Reference</b>	<b>47</b>
<b>8</b>	<b>Example Flows</b>	<b>65</b>
<b>9</b>	<b>CHANGELOG</b>	<b>67</b>
	<b>Python Module Index</b>	<b>79</b>
	<b>Index</b>	<b>81</b>



This is the command line interface and Python package for working with the Globus Flows service and services that implement the Globus Action Provider interface.



## QUICK START

### 1.1 Installation

The Globus Automate CLI and SDK contains both a command-line tool `globus-automate` and a client library for interacting with actions and flows. It requires [Python 3.6+](#). If a supported version of Python is not already installed on your system, see this [Python installation guide](#).

For new users and users who want to primarily use the CLI interface into *Globus Flows*, we highly recommend installing the package using `pipx`:

```
pipx install globus-automate-client
```

For users interested in programmatic access to *Globus Flows*, the SDK interface will be required. Simply install the `globus-automate-client` library using your choice of package manager. For example, if using the `pip` package manager:

```
pip install globus-automate-client
```

### 1.2 Basic Usage

Once installed, run the CLI with the help flag to receive a summary of its functionality:

```
globus-automate --help
```

Or, if programmatic access to the *Flows* service is required, simply import the SDK in a Python script:

```
import globus_automate_client
help(globus_automate_client)
```

---

**Note:** Most operations available on the CLI are also available in the SDK. For users getting familiar with the *Flows* service, we recommend starting off with the CLI interface.

---

## 1.3 Authentication

All CLI and SDK operations invoking Globus automation services require authentication.

When using the SDK, all operations require that authentication be provided in the form of [Globus Authorizers](#).

Upon first interaction with any Globus automation service via the CLI, a text prompt will appear directing the user to a web URL where they can proceed through an authentication process using Globus Auth to consent to the service they are using the CLI to interact with. This typically only needs to be done the first time a particular service is invoked. Subsequently, the cached authentication information will be used. Authentication information is cached in the file `~/.globus_automate_tokens.json`. It is recommended that this file be protected. The file is in json format with section names based on the scope and it may be edited to remove particular scopes if done with care.

If re-authentication or re-consent is needed, the entire authentication cache may be deleted with the following command:

```
globus-automate session logout
```

Or, to invalidate the authentication cache's contents first and then remove it:

```
globus-automate session revoke
```

---

**Note:** These commands will remove locally cached authentication data for all Globus Automate services.

---



## USING THE CLI

For many, the primary way of interacting with Globus automation services will be via the CLI.

### 2.1 Configuration

It is recommended to install `globus-automate` command line completion to support use and discovery of the available commands and options. Currently only the `bash`, `fish`, and `zsh` shells are supported.

```
globus-automate --install-completion
```

---

**Note:** For users of `zsh`, make sure the `zsh` completion system is correctly initialized by running `compinit` near the end of your `.zshrc` file.

---

### 2.2 General Usage

Each command support a `--verbose` option which can be used to see the underlying HTTP request information for what each command is doing. This can be useful for reusing tokens, debugging purposes, or for getting more familiar with the Globus Automate services' APIs.

Each command also supports a `--help` option which provides concise information on the command and documents its expected inputs.

In almost all cases, the output from each command will be in JSON format. The CLI will format the output to try to improve readability, however, you may wish to filter the output using pipes and other command line tools such as `grep` or `jq`.

Many of the commands involve specifying JSON formatted input. Anywhere a command makes use of JSON, it can be specified in one of two ways:

- **Directly on the command line**

Typically by using single quote characters surrounding the content to protect it from interpretation by the shell:

```
--option '{"property": "Value"}'
```

where `--option` is the option requiring a JSON formatted parameter.

- **In a file**

Place the JSON object in a file and refer to the file instead:

```
--option /path/to/your/input_file.json
```

where `--option` is the option requiring a JSON formatted parameter.

Commands which can run or monitor an action or run support a `--watch` flag which can be used to stream updates on an activity's execution state.

Many operations require a principal ID to delegate access. This value represents a single Globus user identity or a Globus Group. The format for a principal is the user ID or Group ID, prefixed with `urn:globus:auth:identity:` if it is a user, or `urn:globus:groups:id:` if it is a Group.

## 2.3 Using the CLI with Actions

In Globus automation services, *actions* represent a single unit of work. Actions are created by invoking *action providers*. The CLI provides a means of interacting with action providers to create and manage actions, however in many cases, the fine grain nature of the actions means they are not often interacted with directly.

To interact with an action, you must know the URL for the action provider that created (or will create) it. This URL is specified using the `--action-url` option to the action subcommands.

Wherever an `--action-url` option is present, the `--action-scope` option may also be provided. The *scope string* is the Globus Auth scope registered for interacting with this action provider. This *scope string* is published as part of the action provider's description (see [Introspection](#)) and the CLI will automatically retrieve this value when it is not specified by the user. In some cases, even retrieving the *scope string* may require authentication, and in these cases, introspecting the action provider is not possible without providing the `--action-scope` option.

All of the action providers operated by the Globus team are described in the [hosted action providers](#) documentation, which includes their URL and tips on using the CLI for interactions with these action providers directly. As these action providers are publicly viewable, there is no need to provide the `--action-scope` option when working with them from the CLI – the CLI will look up the *scope string* automatically.

As an example, we will work through the operations on the Hello World Action Provider at the URL `https://actions.globus.org/hello_world`.

### 2.3.1 Introspection

The first step to learning more about an action provider is using the introspect operation to get a description of the action provider:

```
globus-automate action introspect --action-url https://actions.globus.org/hello_world
```

```
{
  "admin_contact": "support@globus.org",
  "administered_by": [],
  "api_version": "1.0",
  "description": null,
  "event_types": null,
  "globus_auth_scope": "https://auth.globus.org/scopes/actions.globus.org/hello_world",
  "input_schema": {
    "additionalProperties": false,
    "properties": {
      "echo_string": {
        "type": "string"
      }
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

    },
    "required_dependent_scope": {
      "type": "string"
    },
    "sleep_time": {
      "type": "integer"
    },
    "type": "object"
  },
  "keywords": null,
  "log_supported": false,
  "maximum_deadline": "P30D",
  "runnable_by": [
    "all_authenticated_users"
  ],
  "subtitle": "An action responding Hello to an input value",
  "synchronous": false,
  "title": "Hello World",
  "types": [
    "ACTION"
  ],
  "visible_to": [
    "public"
  ]
}

```

From this introspection response we can see that the *scope string* for this action provider is the value of the `globus_auth_scope` field, `https://auth.globus.org/scopes/actions.globus.org/hello_world`. We can also see that the `admin_contact` is Globus.

For information on what this action provider does, it is useful to examine the `title`, `subtitle`, and `description` fields. We can also see that the action provider is `visible_to public`, meaning that anyone can make unauthenticated requests to the introspection endpoint. Similarly, it is `runnable_by all_authenticated_users`, meaning that any user with valid Globus Auth credentials may use this action provider to create actions.

The most important information for our next step is the `input_schema` element as it provides a description of the input we need to form for running an action on this action provider. The `input_schema` element is in [JSON Schema](#) format. This schema defines three properties: `echo_string`, `sleep_time`, and `required_dependent_scope`. We will use this information in the next section on running an action.

### 2.3.2 Running

The first step to prepare for running an action is to create a file containing the input to the action. We'll call the file `hello_input.json` and it contains the following:

```

{
  "echo_string": "Welcome to Globus Automate!",
  "sleep_time": 60
}

```

This input conforms to the `input_schema` from the *Introspection* call, and specifies that we will have the action echo a message back to us and that it will “sleep” for 60 seconds until the action is complete. We'll use this sleep time to

demonstrate monitoring the state of an action below.

With our input in place, run the action using the following command:

```
globus-automate action run --action-url https://actions.globus.org/hello_world --body_↵  
↵hello_input.json
```

---

**Note:** If this is your first time running the Hello World Action Provider you will see text and a prompt appear on your terminal window. Follow the instructions to authenticate to Globus Auth to run this action. This will only appear on the first time you interact with an action provider.

---

The resulting output will look like:

```
{  
  "action_id": "CBOXB3fUdKr0",  
  "completion_time": null,  
  "creator_id": "urn:globus:auth:identity:06a24bef-940e-418a-97bc-48229c64cc99",  
  "details": {  
    "Hello": "World",  
    "hello": "Welcome to Globus Automate!"  
  },  
  "display_status": "ACTIVE",  
  "label": null,  
  "manage_by": [  
    "urn:globus:auth:identity:6f8c1345-33c6-4235-86c6-90fbadbf4d35",  
    "urn:globus:auth:identity:06a24bef-940e-418a-97bc-48229c64cc99"  
  ],  
  "monitor_by": [  
    "urn:globus:auth:identity:6f8c1345-33c6-4235-86c6-90fbadbf4d35",  
    "urn:globus:auth:identity:06a24bef-940e-418a-97bc-48229c64cc99"  
  ],  
  "release_after": null,  
  "start_time": "2021-04-29 23:21:47.763653+00:00",  
  "status": "ACTIVE"  
}
```

This output is referred to as an Action Status document and all output from working with actions will follow this format.

The `action_id` is an identifier associated with this action provider invocation and is used to track this action's lifecycle.

The status value of `ACTIVE` indicates that the action is in the process of executing. The possible values for `status` are:

- **ACTIVE**  
The action is running and making progress towards completion.
- **INACTIVE**  
The action has not yet completed and it is not making progress. Commonly, some intervention is necessary to help it continue to make progress.
- **SUCCEEDED**  
The action is complete and the completion was considered to be normal.
- **FAILED**  
The action has stopped running due to some error condition. It cannot make progress towards a successful

completion.

Each action can be provided a `label` to help identity the purpose for which it was run.

The `details` field format is specific to every action provider and is the output or result of running the action. It will often contain information about why an action has reached the state it is in.

The `release_after` field is an ISO8601 format time duration value that indicates how long after completion the action provider will retain a record of the action's execution. Until then, the record will persist and can be looked up.

`monitor_by` represents delegated read-only access to the action's execution state, meaning that principals in an action's `monitor_by` field will be able to retrieve the action's execution state (see [Retrieving Status](#)). Principals may be either a Globus Auth user or a Globus Auth group. The format for a Globus Auth user is `urn:globus:auth:identity:<UUID>` and for a Globus Auth group is `urn:globus:groups:id:<UUID>`.

`manage_by` represents delegated write access to the action's execution state, meaning that principals in an action's `manage_by` field will have the ability to change the alter the state it is in (see [Canceling and Releasing](#)). Principals may be either a Globus Auth user or a Globus Auth group. The format for a Globus Auth user is `urn:globus:auth:identity:<UUID>` and for a Globus Auth group is `urn:globus:groups:id:<UUID>`.

Since the action has already been run, we cannot change any of these fields. If we wanted to run another action with updated values for any of the fields, we would pass those as command line options. For information on how to use the options, run the command with `--help`:

```
globus-automate action run --help
```

### Tip

You can specify each of the `--monitor-by` and `--manage-by` flags multiple times to provide multiple principals with read or write access on the action.

## 2.3.3 Retrieving Status

Once an action has been run, the user who initiated the action or anyone in the action's `monitor_by` field can monitor or retrieve its status as follows:

```
globus-automate action status --action-url https://actions.globus.org/hello_world
↪<action_id>
```

where the `action_id` is the value returned from the `action run` command from above. The output will be an Action Status document. When the action is completed, the `completion_time` field will be present indicating when the action reached its final state. You can continue requesting the action's status as long as the action exists on the action provider.

In our example, we asked the action to “sleep” for 60 seconds. Therefore, the action will remain in an `ACTIVE` state until 60 seconds have passed, at which point the status should be `SUCCEEDED`.

### 2.3.4 Canceling and Releasing

An action which is running, but which is no longer needed, may be canceled (or released) by the user who initiated the action execution or anyone in the action's `manage_by` field using a command of the form:

```
globus-automate action cancel --action-url https://actions.globus.org/hello_world  
↪<action_id>
```

The cancel operation is considered to be an advisory request from the user. actions may not be cancelled immediately, or they may not be canceled at all. A request to cancel an action which has reached a final state of either `SUCCEEDED` or `FAILED` will result in an error return.

To remove an action's state from the action provider, the user who initiated the action execution or anyone in the action's `manage_by` field can use the release subcommand:

```
globus-automate action release --action-url https://actions.globus.org/hello_world  
↪<action_id>
```

Release may only be performed on actions which have reached a final state. If the action is in either the `ACTIVE` or `INACTIVE` state, the release will fail.

Once released, the action state is forever removed from the action provider and all attempts to access it will fail. Action providers use the `maximum_deadline` field to advertise how long they will keep a record of an action after it reaches a completed state. The time at which this will happen is equal to the `completion_time` plus the `release_after` values in the Action Status document.

## 2.4 Using the CLI with Flows

As described in the [Flows overview](#), a *flow* combines actions and other operations into a more complex operation. When a flow is invoked, it creates a *run* and the run's interface is very much like an action's; it has `run`, `status`, `cancel` and `release` operations defined. Because of this similarity, we sometimes refer to runs as actions in the documentation, CLI and SDK.

The CLI contains commands for creating, defining, and managing flow definitions and commands for running, monitoring, and managing flow runs (also known as *actions*).

---

**Note:** This section does not provide details on writing flows. That is covered in greater detail in the [Authoring Flows](#) documentation.

---

### 2.4.1 Finding and Displaying Flows

When a flow is deployed to Automate, the creator can specify which identities the flow should be visible to and which identities the flow should be runnable by. As the names suggest, users in a flow's `visible_to` field will be able to query the service to view a flow's definition and metadata. Users in a flow's `runnable_by` field will be able to run an instance of the flow.

The following command will list the flows you have created:

```
globus-automate flow list
```

To view flows which are visible or runnable by you as well, run the following command:

```
globus-automate flow list --role created_by --role visible_to --role runnable_by
```

This outputs a list of flows, where the description of each flow carries the same fields as the output from `globus-automate action introspect` described above. This emphasizes again the similarity between flows and actions. The `title` and `description` fields may be helpful in determining what a flow does and what its purpose is. Like actions, the `input_schema` may define what is required of the input when running the flow. However, not all flows are required to define an `input_schema` as a convenience to flow authors who may not be familiar with creating JSON Schema specifications. Importantly, each entry in the list of flows will also contain a value for `id` which we refer to as the “flow id” and denote as `flow_id` below. This value will be used for further interacting with a particular flow.

To display information about a single flow you may use:

```
globus-automate flow display <flow_id>
```

Or, to visualize the flow:

```
globus-automate flow display <flow_id> --format image
```

When focusing on one flow, it is also useful to notice the `field_definition`. This is the actual encoding of the flow as it was created and deployed by the flow’s author. Looking at this value may give further information about how the flow works. This can be useful both to determine if a flow performs the function you desire, but also as a method to see how other flows have been defined if you are interested in creating new flows.

## 2.4.2 Executing and Monitoring Flows

Execution and monitoring of flows follows the same pattern as actions: the `run/status/cancel/release` pattern is the same.

When initiating a flow run, you can delegate access to the flow instance to other Globus Auth identities. By providing the `monitor-by` option, you can delegate read-only access to other users or groups, allowing them to retrieve its execution state. By providing the `manage-by` option, you delegate write access to other users or groups, allowing them to alter its execution state. In the example below, we show how to run an instance of a flow and delegate monitor access to a Globus Group:

```
globus-automate flow run <flow_id> --flow-input input.json \
  --monitor-by urn:globus:groups:id:00000000-0000-0000-0000-000000000000
```

**Note:** If no `manage_by` or `monitor_by` values are specified, only the identity instantiating the flow run is allowed to monitor or manage a flow’s running state.

This acts like `globus-automate action run` with the flow id rather than the `action_url` specifying the “name” of the action to be run. The output, like for actions, will be an action status document including an `action_id` which is used in the following commands:

```
globus-automate flow action-status --flow-id <flow_id> <action_id>
```

```
globus-automate flow action-cancel --flow-id <flow_id> <action_id>
```

```
globus-automate flow action-release --flow-id <flow_id> <action_id>
```

For each of these, the `details` provides information about the most recent, potentially final, state executed by the flow. However, as the flow may execute many states, it is useful to be able to see what states have been executed and what their input and output have been. This can be seen via the “log” of the flow execution as follows:

```
globus-automate flow action-log --flow-id <flow_id> <action_id>
```

The log may have a large number of entries. You can request more entries be returned using the option `-limit N` where `N` is the number of log entries to return. The default value is 10.

## 2.4.3 Creating and managing Flows

Many users will only ever use flows created by others, so they may not necessarily need to understand how to create flows including the commands listed in this section. For those that have created a flow, the first step is to deploy a flow as follows:

```
globus-automate flow deploy --title <title> \
  --definition <flow definition JSON> --input-schema <Input schema JSON> \
  --visible-to <urn of user or group which can see this flow> \
  --runnable-by <urn of user or group which can run this flow> \
  --administered-by <urn of user or group who can maintain this flow>
```

When deployed this way, only the identity that deployed the flow will be able to view the flow and only they will be able to run an instance of the flow. When deploying, it's possible to specify who should be able to see and run the flow. Using the `visible_to` flag, you can indicate which Globus identities can view the deployed flow, or set it to `public`, which creates a flow viewable by anyone. Using the `runnable_by` flag, you can indicate which Globus identities can run an instance of the deployed flow, or set a value of `all_authenticated_users` which allows any authenticated user to run an instance of the flow.

Below, we demonstrate how to deploy a flow that is `visible_to` a single Globus group and `runnable_by` any authenticated user:

```
globus-automate flow deploy --title <title> \
  --definition <flow definition JSON> \
  --input-schema <Input schema JSON> \
  --visible-to urn:globus:groups:id:000000000-0000-0000-0000-000000000000 \
  --runnable-by all_authenticated_users
```

Once deployed, the output will be the flow description as displayed by the `flow display` command above. These command line options provide the values for the similarly named fields in the flow description. Of these, only `title` and `definition` are required. To aid users in using your flow, we highly recommend the use of `input-schema` as it provides them both a form of documentation and assurance at run-time that the input they provide is correct for executing the flow. By providing a value or values to `administered-by` you grant rights to others for updating or eventually removing the flow you have deployed. Commands for updating and removing flows are as follows.

```
globus-automate flow update --title <title> \
  --definition <flow definition JSON> --input-schema <Input schema JSON> \
  --visible-to <urn of user or group which can see this flow> \
  --runnable-by <urn of user or group which can run this flow> \
  --administered-by <urn of user or group who can maintain this flow> \
  <flow_id>
```

This will update any of the fields or description of the flow, including the flow definition itself. Note the `flow_id` field is present at the end of the command line.

Deleting a flow is done via:

```
globus-automate flow delete <flow_id>
```



Care should be taken when issuing this command. There is no further prompting to ensure the flow should really be deleted. After deletion, no record of the flow definition or its execution history (i.e. the `flow action-*` commands) is maintained.

The bulk of the effort in creating flows is in authoring their definition which is covered in the [Authoring Flows](#) documentation.



**LICENSE**

Copyright 2020-2023 University of Chicago

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.



## RUNNING A FLOW AUTOMATICALLY

It is frequently desirable to run a flow when new data becomes available. This document provides examples for how to accomplish this using `watchdog` to monitor for filesystem events and using the Globus Automate CLI to run already-defined flows.

### 4.1 Shell scripting

`watchdog` provides a CLI tool named `watchmedo`. If you're comfortable with shell scripting then you can use the following command to run a script named `runner.sh`.

Listing 1: `watchmedo.sh` [download]

```
watchmedo shell-command \  
  --command 'bash runner.sh "${watch_event_type}" "${watch_src_path}"' \  
  --recursive \  
  .
```

The `watchmedo` command currently has no way to filter filesystem events. On Linux, `runner.sh` will be run when:

- a file is created (but has not had data written yet)
- data is written to the file (but the file has not been closed yet)
- the file is closed

This will likely result in your flow running far more often than expected. To avoid this, `runner.sh` must filter the incoming filesystem events. Here is an example script that will run a flow with custom input when the filesystem event type is "closed":

Listing 2: `runner.sh` [download]

```
set -eu  
  
# Only run a flow when a new file is created.  
if [ "$1" != "closed" ]; then  
  exit  
fi  
  
FLOW_ID="your-flow-id-here"  
FLOW_INPUT=$(cat << EOF  
{  
  "event": "$1",
```

(continues on next page)

(continued from previous page)

```

    "filename": "$2"
}
EOF
)

globus-automate flow run \
  --label "File change: $1" \
  --flow-input "${FLOW_INPUT}" \
  "${FLOW_ID}"

```

**Note:** Filesystem events are less granular on Windows platforms. Notably, there is no "closed" event type to signal all data have been written to a file. There is only a "modified" event type, which may fire multiple times if large files are written and flushed to disk in multiple chunks.

Users on Windows may want to use the Python script below.

## 4.2 Python scripting

If you know that files will be created or modified in large batches, you may need to write a script to monitor for filesystem events and wait some amount of time for filesystem events to taper off. One way to do this is using the `watchdog` package.

The script below will monitor for filesystem events of all types. It will only run a flow 60 seconds after the most recent filesystem event is encountered.

Listing 3: `runner.py` [\[download\]](#)

```

import logging
import os
import pathlib
import queue
import sys

# The flow to run.
FLOW_ID = "your-flow-id-here"

# The flow will be run X seconds after the most recent filesystem event is received.
# If no filesystem events are ever received, the flow will not be run.
COOL_OFF_TIME_S = 60

logging.basicConfig(
    level=logging.WARNING, # Eliminate INFO messages from the Globus SDK.
    format="%(asctime)s - %(message)s",
    datefmt="%Y-%m-%d %H:%M:%S",
)
log = logging.getLogger(__name__)

try:
    from globus_automate_client import create_flows_client

```

(continues on next page)

(continued from previous page)

```

except ImportError:
    log.error(
        "The globus_automate_client package is not installed."
        " (Do you need to activate a virtual environment?)"
    )
    sys.exit(1)

try:
    import watchdog
    import watchdog.events
    import watchdog.observers
except ImportError:
    log.error(
        "The watchdog package is not installed."
        " (Do you need to activate a virtual environment?)"
    )
    sys.exit(1)

class Handler(watchdog.events.FileSystemEventHandler):
    def __init__(self, events: queue.Queue):
        self.events = events

    def dispatch(self, event):
        """Put all filesystem events in a queue."""

        self.events.put(event)

def main():
    try:
        path = pathlib.Path(sys.argv[1]).absolute()
    except IndexError:
        path = pathlib.Path(os.getcwd()).absolute()
    log.warning(f"Monitoring {path}")
    log.warning("Press CTRL-C to exit (on Windows, press CTRL-BREAK)")

    event_queue = queue.Queue(maxsize=-1)
    handler = Handler(event_queue)
    observer = watchdog.observers.Observer()
    observer.schedule(handler, str(path), recursive=True)
    observer.start()

    flows_client = create_flows_client()

    try:
        timeout = None
        files = set()
        while True:
            try:
                event = event_queue.get(block=True, timeout=timeout)
            except queue.Empty:

```

(continues on next page)

(continued from previous page)

```
# .get() timed out.
# It's now been COOL_OFF_TIME_S seconds since the last filesystem event.
# Reset the timeout for the next batch of files and run the flow.
timeout = None
log.warning(f"Running the flow ({len(files)} paths were modified)")
flows_client.run_flow(
    flow_id=FLOW_ID,
    flow_scope=None,
    flow_input={
        "count": len(files),
    },
    label=f"[AUTO] File system changes detected ({len(files)} paths)",
)
files = set()
else:
    # .get() returned a filesystem event.
    # Make sure the next .get() call times out after COOL_OFF_TIME_S.
    timeout = COOL_OFF_TIME_S
    files.add(event.src_path)
    event_queue.task_done()
except KeyboardInterrupt:
    pass
finally:
    observer.stop()
    observer.join()

if __name__ == "__main__":
    main()
```



**CLI REFERENCE**



## GLOBUS-AUTOMATE

A command-line client for the Globus Flows service

By default, this CLI keeps all its config and cached tokens in `.globus_automate_tokens.json` in the user's home directory.

**Usage:**

```
$ globus-automate [OPTIONS] COMMAND [ARGS]...
```

**Options:**

- `-V, --version`: Print CLI version number and exit
- `--install-completion`: Install completion for the current shell.
- `--show-completion`: Show completion for the current shell, to copy it or customize the installation.
- `--help`: Show this message and exit.

**Commands:**

- `action`: Manage runs (actions)
- `flow`: Manage flows
- `session`: Manage your session with the Automate command line client

### 6.1 globus-automate action

**Usage:**

```
$ globus-automate action [OPTIONS] COMMAND [ARGS]...
```

**Options:**

- `--help`: Show this message and exit.

**Commands:**

- `cancel`: Terminate a running Action by its ACTION\_ID.
- `introspect`: Introspect an Action Provider's schema.
- `release`: Remove an Action's execution history by its...
- `resume`: Resume an inactive Action by its ACTION\_ID.
- `run`: Launch an Action.
- `status`: Query an Action's status by its ACTION\_ID.

### 6.1.1 globus-automate action cancel

Terminate a running Action by its ACTION\_ID.

Usage:

```
$ globus-automate action cancel [OPTIONS] ACTION_ID
```

Arguments:

- ACTION\_ID: [required]

Options:

- --action-url TEXT: The url at which the target Action Provider is located. [required]
- --action-scope TEXT: The scope this Action Provider uses to authenticate requests.
- -v, --verbose: Run with increased verbosity
- -f, --format [json|yaml]: Output display format. [default: json]
- --help: Show this message and exit.

### 6.1.2 globus-automate action introspect

Introspect an Action Provider's schema.

Usage:

```
$ globus-automate action introspect [OPTIONS]
```

Options:

- --action-url TEXT: The url at which the target Action Provider is located. [required]
- --action-scope TEXT: The scope this Action Provider uses to authenticate requests.
- -v, --verbose: Run with increased verbosity
- -f, --format [json|yaml]: Output display format. [default: json]
- --help: Show this message and exit.

### 6.1.3 globus-automate action release

Remove an Action's execution history by its ACTION\_ID.

Usage:

```
$ globus-automate action release [OPTIONS] ACTION_ID
```

Arguments:

- ACTION\_ID: [required]

Options:

- --action-url TEXT: The url at which the target Action Provider is located. [required]
- --action-scope TEXT: The scope this Action Provider uses to authenticate requests.
- -v, --verbose: Run with increased verbosity

- `-f, --format [json|yaml]`: Output display format. [default: json]
- `--help`: Show this message and exit.

### 6.1.4 globus-automate action resume

Resume an inactive Action by its ACTION\_ID.

Usage:

```
$ globus-automate action resume [OPTIONS] ACTION_ID
```

Arguments:

- ACTION\_ID: [required]

Options:

- `--action-url TEXT`: The url at which the target Action Provider is located. [required]
- `--action-scope TEXT`: The scope this Action Provider uses to authenticate requests.
- `--query-for-inactive-reason / --no-query-for-inactive-reason`: Should the Action first be queried to determine the reason for the resume, and prompt for additional consent if needed. [default: True]
- `-v, --verbose`: Run with increased verbosity
- `-f, --format [json|yaml]`: Output display format. [default: json]
- `-w, --watch`: Continuously poll this Action until it reaches a completed state. [default: False]
- `--help`: Show this message and exit.

### 6.1.5 globus-automate action run

Launch an Action.

Usage:

```
$ globus-automate action run [OPTIONS]
```

Options:

- `--action-url TEXT`: The url at which the target Action Provider is located. [required]
- `--action-scope TEXT`: The scope this Action Provider uses to authenticate requests.
- `-b, --body TEXT`: The body to supply to the Action Provider. Can be a filename or raw JSON string. [required]
- `--request-id TEXT`: An identifier to associate with this Action invocation request
- `--manage-by TEXT`: A principal which may change the execution of the Action. The principal is the user's or group's UUID prefixed with either `'urn:globus:groups:id:'` or `'urn:globus:auth:identity:'` [repeatable]
- `--monitor-by TEXT`: A principal which may view the state of the Action. The principal is the user's or group's UUID prefixed with either `'urn:globus:groups:id:'` or `'urn:globus:auth:identity:'` [repeatable]
- `-v, --verbose`: Run with increased verbosity
- `-f, --format [json|yaml]`: Output display format. [default: json]
- `-l, --label TEXT`: Optional label to mark this execution of the action.
- `-w, --watch`: Continuously poll this Action until it reaches a completed state. [default: False]

- `--help`: Show this message and exit.

### 6.1.6 globus-automate action status

Query an Action's status by its ACTION\_ID.

Usage:

```
$ globus-automate action status [OPTIONS] ACTION_ID
```

Arguments:

- ACTION\_ID: [required]

Options:

- `--action-url` TEXT: The url at which the target Action Provider is located. [required]
- `--action-scope` TEXT: The scope this Action Provider uses to authenticate requests.
- `-v`, `--verbose`: Run with increased verbosity
- `-f`, `--format` [json|yaml]: Output display format. [default: json]
- `-w`, `--watch`: Continuously poll this Action until it reaches a completed state. [default: False]
- `--help`: Show this message and exit.

## 6.2 globus-automate flow

Manage Globus Automate Flows

To target a different Flows service endpoint, export the GLOBUS\_AUTOMATE\_FLOWS\_ENDPOINT environment variable.

Usage:

```
$ globus-automate flow [OPTIONS] COMMAND [ARGS]...
```

Options:

- `--help`: Show this message and exit.

Commands:

- `action-cancel`: Cancel an active execution for a particular...
- `action-enumerate`: Retrieve all Flow Runs you have access to...
- `action-list`: List a Flow definition's discrete...
- `action-log`: Get a log of the steps executed by a Flow...
- `action-release`: Remove execution history for a particular...
- `action-resume`: Resume a Flow in the INACTIVE state.
- `action-status`: Display the status for a Flow definition's...
- `action-update`: Update a Run on the Flows service.
- `batch-run-update`: Update metadata and permissions on one or...

- **delete:** Delete a Flow.
- **deploy:** Deploy a new Flow.
- **display:** Visualize a local or deployed Flow...
- **get:** Get a Flow's definition as it exists on the...
- **lint:** Parse and validate a Flow definition by...
- **list:** List Flows for which you have access.
- **run:** Run an instance of a Flow.
- **run-cancel:** Cancel an active execution for a particular...
- **run-enumerate:** Retrieve all Flow Runs you have access to...
- **run-list:** List a Flow definition's discrete...
- **run-log:** Get a log of the steps executed by a Flow...
- **run-release:** Remove execution history for a particular...
- **run-resume:** Resume a Flow in the INACTIVE state.
- **run-status:** Display the status for a Flow definition's...
- **run-update:** Update a Run on the Flows service.
- **update:** Update a Flow.

### 6.2.1 globus-automate flow action-cancel

Cancel an active execution for a particular Flow definition's invocation.

**Usage:**

```
$ globus-automate flow action-cancel [OPTIONS] ACTION_ID
```

**Arguments:**

- **ACTION\_ID:** [required]

**Options:**

- **--flow-id TEXT:** The ID for the Flow which triggered the Action. [required]
- **--flow-scope TEXT:** The scope this Flow uses to authenticate requests.
- **-f, --format [json|yaml]:** Output display format. [default: json]
- **-v, --verbose:** Run with increased verbosity
- **--help:** Show this message and exit.

## 6.2.2 globus-automate flow action-enumerate

Retrieve all Flow Runs you have access to view.

Usage:

```
$ globus-automate flow action-enumerate [OPTIONS]
```

Options:

- `--role [run_monitor|run_manager|run_owner|created_by|monitor_by|manage_by]`: Display Actions/Runs where you have at least the selected role. Precedence of roles is: run\_monitor, run\_manager, run\_owner. Thus, by specifying, for example, run\_manager, all flows for which you have run\_manager or run\_owner roles will be displayed. Values monitored\_by, managed\_by and created\_by are deprecated. [repeatable use deprecated as the lowest precedence value provided will determine the Actions/Runs displayed.] [default: ActionRole.run\_owner]
- `--status [SUCCEEDED|FAILED|ACTIVE|INACTIVE]`: Display Actions with the selected status. [repeatable] [default: ]
- `-m, --marker TEXT`: A pagination token for iterating through returned data.
- `-p, --per-page INTEGER RANGE`: The page size to return. Only valid when used without providing a marker.
- `--filter TEXT`: A filtering criteria in the form 'key=value' to apply to the resulting Action listing. The key indicates the filter, the value indicates the pattern to match. Multiple patterns for a single key may be specified as a comma separated string, the results for which will represent a logical OR. If multiple filters are applied, the returned data will be the result of a logical AND between them. [repeatable]
- `--orderby TEXT`: An ordering criteria in the form 'key=value' to apply to the resulting Flow listing. The key indicates the field to order on, and the value is either ASC, for ascending order, or DESC, for descending order. The first ordering criteria will be used to sort the data, subsequent ordering criteria will further sort ties. [repeatable]
- `-w, --watch`: Continuously poll for new Actions. [default: False]
- `-f, --format [json|yaml|table]`: Output display format. [default: table]
- `-v, --verbose`: Run with increased verbosity
- `--help`: Show this message and exit.

## 6.2.3 globus-automate flow action-list

List a Flow definition's discrete invocations.

Usage:

```
$ globus-automate flow action-list [OPTIONS]
```

Options:

- `--flow-id TEXT`: The ID for the Flow which triggered the Action. If not present runs from all Flows will be displayed.
- `--flow-scope TEXT`: The scope this Flow uses to authenticate requests.
- `--role [run_monitor|run_manager|run_owner|created_by|monitor_by|manage_by]`: Display Actions/Runs where you have at least the selected role. Precedence of roles is: run\_monitor, run\_manager, run\_owner. Thus, by specifying, for example, run\_manager, all runs for which you have run\_manager or run\_owner roles will be displayed.



run\_owner roles will be displayed. [repeatable use deprecated as the lowest precedence value provided will determine the flows displayed.]

- `--status` [SUCCEEDED|FAILED|ACTIVE|INACTIVE]: Display Actions with the selected status. [repeatable] [default: ]
- `-m, --marker` TEXT: A pagination token for iterating through returned data.
- `-p, --per-page` INTEGER RANGE: The page size to return. Only valid when used without providing a marker.
- `--filter` TEXT: A filtering criteria in the form 'key=value' to apply to the resulting Action listing. The key indicates the filter, the value indicates the pattern to match. Multiple patterns for a single key may be specified as a comma separated string, the results for which will represent a logical OR. If multiple filters are applied, the returned data will be the result of a logical AND between them. [repeatable]
- `--orderby` TEXT: An ordering criteria in the form 'key=value' to apply to the resulting Flow listing. The key indicates the field to order on, and the value is either ASC, for ascending order, or DESC, for descending order. The first ordering criteria will be used to sort the data, subsequent ordering criteria will further sort ties. [repeatable]
- `-v, --verbose`: Run with increased verbosity
- `-w, --watch`: Continuously poll for new Actions. [default: False]
- `-f, --format` [json|yaml|table]: Output display format. [default: table]
- `--help`: Show this message and exit.

## 6.2.4 globus-automate flow action-log

Get a log of the steps executed by a Flow definition's invocation.

Usage:

```
$ globus-automate flow action-log [OPTIONS] ACTION_ID
```

Arguments:

- ACTION\_ID: [required]

Options:

- `--flow-id` TEXT: The ID for the Flow which triggered the Action. [required]
- `--flow-scope` TEXT: The scope this Flow uses to authenticate requests.
- `--reverse`: Display logs starting from most recent and proceeding in reverse chronological order [default: False]
- `--limit` INTEGER RANGE: Set a maximum number of events from the log to return
- `-m, --marker` TEXT: A pagination token for iterating through returned data.
- `-p, --per-page` INTEGER RANGE: The page size to return. Only valid when used without providing a marker.
- `-f, --format` [json|yaml|table|image|graphiz]: Output display format. [default: table]
- `-w, --watch`: Continuously poll this Action until it reaches a completed state. Using this option will report only the latest state available. [default: False]
- `-v, --verbose`: Run with increased verbosity
- `--help`: Show this message and exit.

### 6.2.5 globus-automate flow action-release

Remove execution history for a particular Flow definition's invocation. After this, no further information about the run can be accessed.

Usage:

```
$ globus-automate flow action-release [OPTIONS] ACTION_ID
```

Arguments:

- ACTION\_ID: [required]

Options:

- --flow-id TEXT: The ID for the Flow which triggered the Action. [required]
- --flow-scope TEXT: The scope this Flow uses to authenticate requests.
- -f, --format [json|yaml]: Output display format. [default: json]
- -v, --verbose: Run with increased verbosity
- --help: Show this message and exit.

### 6.2.6 globus-automate flow action-resume

Resume a Flow in the INACTIVE state. If query-for-inactive-reason is set, and the Flow Action is in an INACTIVE state due to requiring additional Consent, the required Consent will be determined and you may be prompted to allow Consent using the Globus Auth web interface.

Usage:

```
$ globus-automate flow action-resume [OPTIONS] ACTION_ID
```

Arguments:

- ACTION\_ID: [required]

Options:

- --flow-id TEXT: The ID for the Flow which triggered the Action. [required]
- --flow-scope TEXT: The scope this Flow uses to authenticate requests.
- --query-for-inactive-reason / --no-query-for-inactive-reason: Should the Action first be queried to determine the reason for the resume, and prompt for additional consent if needed. [default: True]
- -f, --format [json|yaml]: Output display format. [default: json]
- -w, --watch: Continuously poll this Action until it reaches a completed state. [default: False]
- -v, --verbose: Run with increased verbosity
- --help: Show this message and exit.

### 6.2.7 globus-automate flow action-status

Display the status for a Flow definition's particular invocation.

Usage:

```
$ globus-automate flow action-status [OPTIONS] ACTION_ID
```

Arguments:

- ACTION\_ID: [required]

Options:

- --flow-id UUID: The ID for the Flow which triggered the Action. [required]
- --flow-scope TEXT: The scope this Flow uses to authenticate requests.
- -w, --watch: Continuously poll this Action until it reaches a completed state. [default: False]
- -f, --format [json|yaml]: Output display format. [default: json]
- -v, --verbose: Run with increased verbosity
- --help: Show this message and exit.

### 6.2.8 globus-automate flow action-update

Update a Run on the Flows service.

Usage:

```
$ globus-automate flow action-update [OPTIONS] RUN_ID
```

Arguments:

- RUN\_ID: [required]

Options:

- --run-manager TEXT: A principal which may change the execution of the Run. The principal value is the user's Globus Auth username or their identity UUID in the form `urn:globus:auth:identity:`. A Globus Group may also be used using the form `urn:globus:groups:id:`. Specify an empty string once to erase all Run managers. [repeatable]
- --run-monitor TEXT: A principal which may monitor the execution of the Run. The principal value is the user's Globus Auth username or their identity UUID in the form `urn:globus:auth:identity:`. A Globus Group may also be used using the form `urn:globus:groups:id:`. [repeatable]
- --tag TEXT: A tag to associate with the Run. If specified, the existing tags on the Run will be replaced with the list of tags specified here. Specify an empty string once to erase all tags. [repeatable]
- --label TEXT: A label to associate with the Run.
- -v, --verbose: Run with increased verbosity
- -f, --format [json|yaml]: Output display format. [default: json]
- --help: Show this message and exit.

## 6.2.9 globus-automate flow batch-run-update

Update metadata and permissions on one or more Runs.

Modifying lists of values =====

Most options support set, add, and remove operations.

The “add” option variants will add the specified value to whatever is set on each affected Run. For example, if one Run has a “star” tag and another has a “circle” tag, `--add-tag square` will result in a Run with “star” and “square” tags, and the other Run will have “circle” and “square” tags.

The “remove” option variants will remove the specified value from whatever is set on each affected Run. There will not be an error if the value is not set on a Run. For example, if one Run has a “star” tag and another has a “circle” tag, `--remove-tag star` will result in a Run with no tags while the other still has a “circle” tag.

The “set” option variants will overwrite the metadata and permissions currently set on all affected Runs. For example, `--set-tag example` will standardize all affected Runs so that they have just one tag: “example”.

To remove all values on all affected Runs, use the “set” variant of an option with an empty string. For example, to erase all Run monitors, use `--set-run-monitors ""`.

All options with “set”, “add”, and “remove” variants can be used multiple times. However, only one variation of an option can be specified at a time. For example, `--set-tag` and `--add-tag` cannot be combined in the same command, and `--set-run-manager` and `--add-run-manager` cannot be combined. It is fine to combine `--add-tag` and `--remove-run-manager`.

Modifying roles =====

Run managers and monitors must be specified in one of these forms:

\* A user’s Globus Auth username \* A user’s identity UUID in the form `urn:globus:auth:identity:` \* A group’s identity UUID in the form `urn:globus:groups:id:`

Usage:

```
$ globus-automate flow batch-run-update [OPTIONS] RUN_IDS...
```

Arguments:

- `RUN_IDS...`: [required]

Options:

- `--set-run-manager TEXT`: Set a principal on affected Runs that can change the Run execution.
- `--add-run-manager TEXT`: Add a principal to affected Runs that can change the Run execution.
- `--remove-run-manager TEXT`: Remove a principal from affected Runs that can change the Run execution.
- `--set-run-monitor TEXT`: Set a principal on affected Runs that can monitor Run execution.
- `--add-run-monitor TEXT`: Add a principal to affected Runs that can monitor Run execution.
- `--remove-run-monitor TEXT`: Remove a principal from affected Runs that can monitor Run execution.
- `--set-tag TEXT`: A tag to set on the specified Runs.
- `--add-tag TEXT`: A tag to add to the affected Runs.
- `--remove-tag TEXT`: A tag to remove from the affected Runs.
- `--status TEXT`: Set the status of the affected Runs.

Currently, “cancel” is the only valid value. \* `-v`, `--verbose`: Run with increased verbosity \* `-f`, `--format [json|yaml]`: Output display format. [default: json] \* `--help`: Show this message and exit.

### 6.2.10 globus-automate flow delete

Delete a Flow. You must be in the Flow's "flow\_administrators" list.

Usage:

```
$ globus-automate flow delete [OPTIONS] FLOW_ID
```

Arguments:

- FLOW\_ID: [required]

Options:

- -f, --format [json|yaml]: Output display format. [default: json]
- -v, --verbose: Run with increased verbosity
- --help: Show this message and exit.

### 6.2.11 globus-automate flow deploy

Deploy a new Flow.

Usage:

```
$ globus-automate flow deploy [OPTIONS]
```

Options:

- --title TEXT: The Flow's title. [required]
- --definition TEXT: JSON or YAML representation of the Flow to deploy. May be provided as a filename or a raw string representing a JSON object or YAML definition. [required]
- --subtitle TEXT: A subtitle for the Flow providing additional, brief description.
- --description TEXT: A long form description of the Flow's purpose or usage.
- --input-schema TEXT: A JSON or YAML representation of a JSON Schema which will be used to validate the input to the deployed Flow when it is run. If not provided, no validation will be performed on Flow input. May be provided as a filename or a raw string.
- --keyword TEXT: A keyword which may categorize or help discover the Flow. [repeatable]
- --flow-viewer TEXT: A principal which may view this Flow. The principal value is the user's Globus Auth username or their identity UUID in the form `urn:globus:auth:identity:.`. A Globus Group may also be used using the form `urn:globus:groups:id:.`. The special value of 'public' may be used to indicate that any user can view this Flow. [repeatable]
- --flow-starter TEXT: A principal which may run an instance of the deployed Flow. The principal value is the user's Globus Auth username or their identity UUID in the form `urn:globus:auth:identity:.`. A Globus Group may also be used using the form `urn:globus:groups:id:.`. The special value of 'all\_authenticated\_users' may be used to indicate that any authenticated user can invoke this flow. [repeatable]
- --flow-administrator TEXT: A principal which may update the deployed Flow. The principal value is the user's Globus Auth username or their identity UUID in the form `urn:globus:auth:identity:.`. A Globus Group may also be used using the form `urn:globus:groups:id:.[{}]` [repeatable]
- --subscription-id TEXT: The Id of the Globus Subscription which will be used to make this flow managed.

- `--validate / --no-validate`: (EXPERIMENTAL) Perform rudimentary validation of the flow definition. [default: True]
- `-v, --verbose`: Run with increased verbosity
- `-f, --format [json|yaml]`: Output display format. [default: json]
- `--dry-run`: Do a dry run of deploying the flow to test your definition without actually making changes. [default: False]
- `--help`: Show this message and exit.

### 6.2.12 globus-automate flow display

Visualize a local or deployed Flow definition. If providing a Flow's ID, You must have either created the Flow or be present in the Flow's "flow\_viewers" list to view it.

Usage:

```
$ globus-automate flow display [OPTIONS] [FLOW_ID]
```

Arguments:

- [FLOW\_ID]

Options:

- `--flow-definition TEXT`: JSON or YAML representation of the Flow to display. May be provided as a filename or a raw string representing a JSON object or YAML definition.
- `-f, --format [json|yaml|image|graphviz]`: Output display format. [default: json]
- `--help`: Show this message and exit.

### 6.2.13 globus-automate flow get

Get a Flow's definition as it exists on the Flows service.

Usage:

```
$ globus-automate flow get [OPTIONS] FLOW_ID
```

Arguments:

- FLOW\_ID: A deployed Flow's ID [required]

Options:

- `-f, --format [json|yaml]`: Output display format. [default: json]
- `-v, --verbose`: Run with increased verbosity
- `--help`: Show this message and exit.

### 6.2.14 globus-automate flow lint

Parse and validate a Flow definition by providing visual output.

Usage:

```
$ globus-automate flow lint [OPTIONS]
```

Options:

- `--definition TEXT`: JSON or YAML representation of the Flow to deploy. May be provided as a filename or a raw string. [required]
- `--help`: Show this message and exit.

### 6.2.15 globus-automate flow list

List Flows for which you have access.

Usage:

```
$ globus-automate flow list [OPTIONS]
```

Options:

- `-r, --role [flow_viewer|flow_starter|flow_administrator|flow_owner|created_by|visible_to|runnable_by]`: Display Flows where you have at least the selected role. Precedence of roles is: flow\_viewer, flow\_starter, flow\_administrator, flow\_owner. Thus, by specifying, for example, flow\_starter, all flows for which you have flow\_starter, flow\_administrator, or flow\_owner roles will be displayed. Values visible\_to, runnable\_by, administered\_by and created\_by are deprecated. [repeatable use deprecated as the lowest precedence value provided will determine the flows displayed.] [default: FlowRole.flow\_owner]
- `-m, --marker TEXT`: A pagination token for iterating through returned data.
- `-p, --per-page INTEGER RANGE`: The page size to return. Only valid when used without providing a marker.
- `--filter TEXT`: A filtering criteria in the form 'key=value' to apply to the resulting Flow listing. The key indicates the filter, the value indicates the pattern to match. Multiple patterns for a single key may be specified as a comma separated string, the results for which will represent a logical OR. If multiple filters are applied, the returned data will be the result of a logical AND between them. [repeatable]
- `--orderby TEXT`: An ordering criteria in the form 'key=value' to apply to the resulting Flow listing. The key indicates the field to order on, and the value is either ASC, for ascending order, or DESC, for descending order. The first ordering criteria will be used to sort the data, subsequent ordering criteria will further sort ties. [repeatable]
- `-v, --verbose`: Run with increased verbosity
- `-f, --format [json|yaml|table]`: Output display format. [default: table]
- `-w, --watch`: Continuously poll for new Flows. [default: False]
- `--help`: Show this message and exit.

## 6.2.16 globus-automate flow run

Run an instance of a Flow. The argument provides the initial state of the Flow. You must be in the Flow’s “flow\_starters” list.

Usage:

```
$ globus-automate flow run [OPTIONS] FLOW_ID
```

Arguments:

- FLOW\_ID: [required]

Options:

- `--flow-input TEXT`: JSON or YAML formatted input to the Flow. May be provided as a filename or a raw string. [required]
- `--flow-scope TEXT`: The scope this Flow uses to authenticate requests.
- `--run-manager TEXT`: A principal which may change the execution of the Flow instance. The principal value is the user’s Globus Auth username or their identity UUID in the form `urn:globus:auth:identity:.`. A Globus Group may also be used using the form `urn:globus:groups:id:.` [repeatable]
- `--run-monitor TEXT`: A principal which may monitor the execution of the Flow instance. The principal value is the user’s Globus Auth username or their identity UUID in the form `urn:globus:auth:identity:.`. A Globus Group may also be used using the form `urn:globus:groups:id:.` [repeatable]
- `-v, --verbose`: Run with increased verbosity
- `-f, --format [json|yaml|table]`: Output display format. If `--watch` is enabled then the default is ‘table’, otherwise ‘json’ is the default.
- `-l, --label TEXT`: Label to mark this run. [required]
- `--tag TEXT`: A tag to associate with this Run.

This option can be used multiple times. The full collection of tags will associated with the Run. \* `-w, --watch`: Continuously poll this Action until it reaches a completed state. If enabled the default output format is ‘table’. [default: False] \* `--dry-run`: Do a dry run with your input to this flow to test the input without actually running anything. [default: False] \* `--help`: Show this message and exit.

## 6.2.17 globus-automate flow run-cancel

Cancel an active execution for a particular Flow definition’s invocation.

Usage:

```
$ globus-automate flow run-cancel [OPTIONS] ACTION_ID
```

Arguments:

- ACTION\_ID: [required]

Options:

- `--flow-id TEXT`: The ID for the Flow which triggered the Action. [required]
- `--flow-scope TEXT`: The scope this Flow uses to authenticate requests.
- `-f, --format [json|yaml]`: Output display format. [default: json]
- `-v, --verbose`: Run with increased verbosity



- `--help`: Show this message and exit.

## 6.2.18 globus-automate flow run-enumerate

Retrieve all Flow Runs you have access to view.

Usage:

```
$ globus-automate flow run-enumerate [OPTIONS]
```

Options:

- `--role [run_monitor|run_manager|run_owner|created_by|monitor_by|manage_by]`: Display Actions/Runs where you have at least the selected role. Precedence of roles is: `run_monitor`, `run_manager`, `run_owner`. Thus, by specifying, for example, `run_manager`, all flows for which you have `run_manager` or `run_owner` roles will be displayed. Values `monitored_by`, `managed_by` and `created_by` are deprecated. [repeatable use deprecated as the lowest precedence value provided will determine the Actions/Runs displayed.] [default: `ActionRole.run_owner`]
- `--status [SUCCEEDED|FAILED|ACTIVE|INACTIVE]`: Display Actions with the selected status. [repeatable] [default: ]
- `-m, --marker TEXT`: A pagination token for iterating through returned data.
- `-p, --per-page INTEGER RANGE`: The page size to return. Only valid when used without providing a marker.
- `--filter TEXT`: A filtering criteria in the form 'key=value' to apply to the resulting Action listing. The key indicates the filter, the value indicates the pattern to match. Multiple patterns for a single key may be specified as a comma separated string, the results for which will represent a logical OR. If multiple filters are applied, the returned data will be the result of a logical AND between them. [repeatable]
- `--orderby TEXT`: An ordering criteria in the form 'key=value' to apply to the resulting Flow listing. The key indicates the field to order on, and the value is either `ASC`, for ascending order, or `DESC`, for descending order. The first ordering criteria will be used to sort the data, subsequent ordering criteria will further sort ties. [repeatable]
- `-w, --watch`: Continuously poll for new Actions. [default: `False`]
- `-f, --format [json|yaml|table]`: Output display format. [default: `table`]
- `-v, --verbose`: Run with increased verbosity
- `--help`: Show this message and exit.

## 6.2.19 globus-automate flow run-list

List a Flow definition's discrete invocations.

Usage:

```
$ globus-automate flow run-list [OPTIONS]
```

Options:

- `--flow-id TEXT`: The ID for the Flow which triggered the Action. If not present runs from all Flows will be displayed.
- `--flow-scope TEXT`: The scope this Flow uses to authenticate requests.

- `--role [run_monitor|run_manager|run_owner|created_by|monitor_by|manage_by]`: Display Actions/Runs where you have at least the selected role. Precedence of roles is: run\_monitor, run\_manager, run\_owner. Thus, by specifying, for example, run\_manager, all runs for which you have run\_manager or run\_owner roles will be displayed. [repeatable use deprecated as the lowest precedence value provided will determine the flows displayed.]
- `--status [SUCCEEDED|FAILED|ACTIVE|INACTIVE]`: Display Actions with the selected status. [repeatable] [default: ]
- `-m, --marker TEXT`: A pagination token for iterating through returned data.
- `-p, --per-page INTEGER RANGE`: The page size to return. Only valid when used without providing a marker.
- `--filter TEXT`: A filtering criteria in the form 'key=value' to apply to the resulting Action listing. The key indicates the filter, the value indicates the pattern to match. Multiple patterns for a single key may be specified as a comma separated string, the results for which will represent a logical OR. If multiple filters are applied, the returned data will be the result of a logical AND between them. [repeatable]
- `--orderby TEXT`: An ordering criteria in the form 'key=value' to apply to the resulting Flow listing. The key indicates the field to order on, and the value is either ASC, for ascending order, or DESC, for descending order. The first ordering criteria will be used to sort the data, subsequent ordering criteria will further sort ties. [repeatable]
- `-v, --verbose`: Run with increased verbosity
- `-w, --watch`: Continuously poll for new Actions. [default: False]
- `-f, --format [json|yaml|table]`: Output display format. [default: table]
- `--help`: Show this message and exit.

## 6.2.20 globus-automate flow run-log

Get a log of the steps executed by a Flow definition's invocation.

Usage:

```
$ globus-automate flow run-log [OPTIONS] ACTION_ID
```

Arguments:

- ACTION\_ID: [required]

Options:

- `--flow-id TEXT`: The ID for the Flow which triggered the Action. [required]
- `--flow-scope TEXT`: The scope this Flow uses to authenticate requests.
- `--reverse`: Display logs starting from most recent and proceeding in reverse chronological order [default: False]
- `--limit INTEGER RANGE`: Set a maximum number of events from the log to return
- `-m, --marker TEXT`: A pagination token for iterating through returned data.
- `-p, --per-page INTEGER RANGE`: The page size to return. Only valid when used without providing a marker.
- `-f, --format [json|yaml|table|image|graphiz]`: Output display format. [default: table]
- `-w, --watch`: Continuously poll this Action until it reaches a completed state. Using this option will report only the latest state available. [default: False]
- `-v, --verbose`: Run with increased verbosity

- `--help`: Show this message and exit.

### 6.2.21 globus-automate flow run-release

Remove execution history for a particular Flow definition's invocation. After this, no further information about the run can be accessed.

**Usage:**

```
$ globus-automate flow run-release [OPTIONS] ACTION_ID
```

**Arguments:**

- `ACTION_ID`: [required]

**Options:**

- `--flow-id TEXT`: The ID for the Flow which triggered the Action. [required]
- `--flow-scope TEXT`: The scope this Flow uses to authenticate requests.
- `-f, --format [json|yaml]`: Output display format. [default: json]
- `-v, --verbose`: Run with increased verbosity
- `--help`: Show this message and exit.

### 6.2.22 globus-automate flow run-resume

Resume a Flow in the INACTIVE state. If `query-for-inactive-reason` is set, and the Flow Action is in an INACTIVE state due to requiring additional Consent, the required Consent will be determined and you may be prompted to allow Consent using the Globus Auth web interface.

**Usage:**

```
$ globus-automate flow run-resume [OPTIONS] ACTION_ID
```

**Arguments:**

- `ACTION_ID`: [required]

**Options:**

- `--flow-id TEXT`: The ID for the Flow which triggered the Action. [required]
- `--flow-scope TEXT`: The scope this Flow uses to authenticate requests.
- `--query-for-inactive-reason / --no-query-for-inactive-reason`: Should the Action first be queried to determine the reason for the resume, and prompt for additional consent if needed. [default: True]
- `-f, --format [json|yaml]`: Output display format. [default: json]
- `-w, --watch`: Continuously poll this Action until it reaches a completed state. [default: False]
- `-v, --verbose`: Run with increased verbosity
- `--help`: Show this message and exit.

### 6.2.23 globus-automate flow run-status

Display the status for a Flow definition's particular invocation.

Usage:

```
$ globus-automate flow run-status [OPTIONS] ACTION_ID
```

Arguments:

- ACTION\_ID: [required]

Options:

- --flow-id UUID: The ID for the Flow which triggered the Action. [required]
- --flow-scope TEXT: The scope this Flow uses to authenticate requests.
- -w, --watch: Continuously poll this Action until it reaches a completed state. [default: False]
- -f, --format [json|yaml]: Output display format. [default: json]
- -v, --verbose: Run with increased verbosity
- --help: Show this message and exit.

### 6.2.24 globus-automate flow run-update

Update a Run on the Flows service.

Usage:

```
$ globus-automate flow run-update [OPTIONS] RUN_ID
```

Arguments:

- RUN\_ID: [required]

Options:

- --run-manager TEXT: A principal which may change the execution of the Run. The principal value is the user's Globus Auth username or their identity UUID in the form `urn:globus:auth:identity:`. A Globus Group may also be used using the form `urn:globus:groups:id:`. Specify an empty string once to erase all Run managers. [repeatable]
- --run-monitor TEXT: A principal which may monitor the execution of the Run. The principal value is the user's Globus Auth username or their identity UUID in the form `urn:globus:auth:identity:`. A Globus Group may also be used using the form `urn:globus:groups:id:`. [repeatable]
- --tag TEXT: A tag to associate with the Run. If specified, the existing tags on the Run will be replaced with the list of tags specified here. Specify an empty string once to erase all tags. [repeatable]
- --label TEXT: A label to associate with the Run.
- -v, --verbose: Run with increased verbosity
- -f, --format [json|yaml]: Output display format. [default: json]
- --help: Show this message and exit.

## 6.2.25 globus-automate flow update

Update a Flow.

Usage:

```
$ globus-automate flow update [OPTIONS] FLOW_ID
```

Arguments:

- FLOW\_ID: [required]

Options:

- `--title TEXT`: The Flow's title.
- `--definition TEXT`: JSON or YAML representation of the Flow to update. May be provided as a filename or a raw string.
- `--subtitle TEXT`: A subtitle for the Flow providing additional, brief description.
- `--description TEXT`: A long form description of the Flow's purpose or usage.
- `--input-schema TEXT`: A JSON or YAML representation of a JSON Schema which will be used to validate the input to the deployed Flow when it is run. If not provided, no validation will be performed on Flow input. May be provided as a filename or a raw string.
- `--keyword TEXT`: A keyword which may categorize or help discover the Flow. [repeatable]
- `--flow-viewer TEXT`: A principal which may view this Flow. The principal value is the user's Globus Auth username or their identity UUID in the form `urn:globus:auth:identity:`. A Globus Group may also be used using the form `urn:globus:groups:id:`. The special value of 'public' may be used to indicate that any user can view this Flow. [repeatable]
- `--flow-starter TEXT`: A principal which may run an instance of the deployed Flow. The principal value is the user's Globus Auth username or their identity UUID in the form `urn:globus:auth:identity:`. A Globus Group may also be used using the form `urn:globus:groups:id:`. The special value of 'all\_authenticated\_users' may be used to indicate that any authenticated user can invoke this flow. [repeatable]
- `--flow-administrator TEXT`: A principal which may update the deployed Flow. The principal value is the user's Globus Auth username or their identity UUID in the form `urn:globus:auth:identity:`. A Globus Group may also be used using the form `urn:globus:groups:id:[]repeatable`
- `--assume-ownership`: Assume the ownership of the Flow. This can only be performed by user's in the `flow_administrators` role. [default: False]
- `--subscription-id TEXT`: The Globus Subscription which will be used to make this flow managed.
- `--validate / --no-validate`: (EXPERIMENTAL) Perform rudimentary validation of the flow definition. [default: True]
- `-v, --verbose`: Run with increased verbosity
- `-f, --format [json|yaml]`: Output display format. [default: json]
- `--help`: Show this message and exit.

## 6.3 globus-automate queue

Usage:

```
$ globus-automate queue [OPTIONS] COMMAND [ARGS]...
```

Options:

- `--help`: Show this message and exit.

Commands:

- `create`: Create a new Queue.
- `delete`: Delete a Queue based on its id.
- `display`: Display the description of a Queue based on...
- `list`: List Queues for which you have access.
- `message-delete`: Notify a Queue that a message has been...
- `message-receive`: Receive a message from a Queue.
- `message-send`: Send a message to a Queue.
- `update`: Update a Queue's properties.

### 6.3.1 globus-automate queue create

Create a new Queue.

Usage:

```
$ globus-automate queue create [OPTIONS]
```

Options:

- `--label TEXT`: A convenient name to identify the new Queue. [required]
- `--admin TEXT`: The Principal URNs allowed to administer the Queue. [repeatable] [required]
- `--sender TEXT`: The Principal URNs allowed to send to the Queue. [repeatable] [required]
- `--receiver TEXT`: The Principal URNs allowed to receive from the Queue. [repeatable] [required]
- `--delivery-timeout INTEGER RANGE`: The minimum amount of time (in seconds) that the Queue Service should wait for a message-delete request after delivering a message before making the message visible for receiving by other consumers once again. If used in conjunction with 'receiver\_url' this value represents the minimum amount of time (in seconds) that the Queue Service should attempt to retry delivery of messages to the 'receiver\_url' if delivery is not initially successful [default: 60]
- `-f, --format [json|yaml]`: Output display format. [default: json]
- `-v, --verbose`: Run with increased verbosity
- `--help`: Show this message and exit.

### 6.3.2 globus-automate queue delete

Delete a Queue based on its id. You must have either created the Queue or have a role defined on the Queue.

Usage:

```
$ globus-automate queue delete [OPTIONS] QUEUE_ID
```

Arguments:

- QUEUE\_ID: [required]

Options:

- -f, --format [json|yaml]: Output display format. [default: json]
- -v, --verbose: Run with increased verbosity
- --help: Show this message and exit.

### 6.3.3 globus-automate queue display

Display the description of a Queue based on its id.

Usage:

```
$ globus-automate queue display [OPTIONS] QUEUE_ID
```

Arguments:

- QUEUE\_ID: [required]

Options:

- -f, --format [json|yaml]: Output display format. [default: json]
- -v, --verbose: Run with increased verbosity
- --help: Show this message and exit.

### 6.3.4 globus-automate queue list

List Queues for which you have access.

Usage:

```
$ globus-automate queue list [OPTIONS]
```

Options:

- -r, --role [admin|sender|receiver]: Display Queues where you have the selected role. [repeatable] [default: QueueRole.admin]
- -f, --format [json|yaml]: Output display format. [default: json]
- -v, --verbose: Run with increased verbosity
- --help: Show this message and exit.

### 6.3.5 globus-automate queue message-delete

Notify a Queue that a message has been processed.

Usage:

```
$ globus-automate queue message-delete [OPTIONS] QUEUE_ID
```

Arguments:

- QUEUE\_ID: [required]

Options:

- --receipt-handle TEXT: A receipt\_handle value returned by a previous call to receive message. [repeatable] [required]
- -f, --format [json|yaml]: Output display format. [default: json]
- -v, --verbose: Run with increased verbosity
- --help: Show this message and exit.

### 6.3.6 globus-automate queue message-receive

Receive a message from a Queue. You must have the “receiver” role on the Queue to perform this action.

Usage:

```
$ globus-automate queue message-receive [OPTIONS] QUEUE_ID
```

Arguments:

- QUEUE\_ID: [required]

Options:

- --max-messages INTEGER RANGE: The maximum number of messages to retrieve from the Queue
- -f, --format [json|yaml]: Output display format. [default: json]
- -v, --verbose: Run with increased verbosity
- --help: Show this message and exit.

### 6.3.7 globus-automate queue message-send

Send a message to a Queue. You must have the “sender” role on the Queue to perform this action.

Usage:

```
$ globus-automate queue message-send [OPTIONS] QUEUE_ID
```

Arguments:

- QUEUE\_ID: [required]

Options:

- -m, --message TEXT: Text of the message to send. Files may also be referenced. [required]
- -f, --format [json|yaml]: Output display format. [default: json]



- `-v`, `--verbose`: Run with increased verbosity
- `--help`: Show this message and exit.

### 6.3.8 globus-automate queue update

Update a Queue's properties. Requires the admin role on the Queue.

Usage:

```
$ globus-automate queue update [OPTIONS] QUEUE_ID
```

Arguments:

- `QUEUE_ID`: [required]

Options:

- `--label TEXT`: A convenient name to identify the new Queue. [required]
- `--admin TEXT`: The Principal URNs allowed to administer the Queue. [repeatable] [required]
- `--sender TEXT`: The Principal URNs allowed to send to the Queue. [repeatable] [required]
- `--receiver TEXT`: The Principal URNs allowed to receive from the Queue. [repeatable] [required]
- `--delivery-timeout INTEGER RANGE`: The minimum amount of time (in seconds) that the Queue Service should wait for a message-delete request after delivering a message before making the message visible for receiving by other consumers once again. If used in conjunction with 'receiver\_url' this value represents the minimum amount of time (in seconds) that the Queue Service should attempt to retry delivery of messages to the 'receiver\_url' if delivery is not initially successful [required]
- `--visibility-timeout INTEGER RANGE`: [default: 30]
- `-f`, `--format [json|yaml]`: Output display format. [default: json]
- `-v`, `--verbose`: Run with increased verbosity
- `--help`: Show this message and exit.

## 6.4 globus-automate session

Usage:

```
$ globus-automate session [OPTIONS] COMMAND [ARGS]...
```

Options:

- `--help`: Show this message and exit.

Commands:

- `logout`: Remove all locally cached Globus Automate...
- `revoke`: Remove all locally cached Globus Automate...
- `whoami`: Determine the username for the identity...

### 6.4.1 globus-automate session logout

Remove all locally cached Globus Automate authentication information.

Usage:

```
$ globus-automate session logout [OPTIONS]
```

Options:

- `--help`: Show this message and exit.

### 6.4.2 globus-automate session revoke

Remove all locally cached Globus Automate authentication information and invalidate all locally cached access or refresh tokens. These tokens can no longer be used elsewhere.

Usage:

```
$ globus-automate session revoke [OPTIONS]
```

Options:

- `--help`: Show this message and exit.

### 6.4.3 globus-automate session whoami

Determine the username for the identity logged in to Globus Auth. If run with increased verbosity, the caller's full user information is displayed.

Usage:

```
$ globus-automate session whoami [OPTIONS]
```

Options:

- `-v`, `--verbose`: Run with increased verbosity
- `--help`: Show this message and exit.

## PYTHON SDK REFERENCE

The Globus Automate Client package provides a Python SDK for interfacing with and invoking Globus actions and flows. Below, we provide information on helper functions available for creating authenticated action and flow clients as well as documentation on the methods available to those clients.

### 7.1 Actions Client

**class** globus\_automate\_client.action\_client.**ActionClient**(\*args, \*\*kwargs)

**Parameters**

- **args** (*Any*) –
- **kwargs** (*Any*) –

**Return type**

*Any*

**property** action\_scope: **str**

This property can be used to determine an **ActionClient**'s **action\_scope**. Internally, this property will introspect the Action Provider at the URL for which the **ActionClient** was created. If the Action Provider is not public, a valid Globus Authorizer will have to have been provided on initialization to the **ActionClient**. Otherwise, this call will fail.

**introspect**(\*\*\_)

Introspect the details of an Action Provider to discover information such as its expected **action\_scope**, its **input\_schema**, and who to contact when there's trouble.

**Return type**

globus\_sdk.GlobusHTTPResponse

**run**(body, request\_id=None, manage\_by=None, monitor\_by=None, label=None, tags=None, force\_path=None, \*\*kwargs)

Invoke the Action Provider to execute an Action with the given parameters.

**Parameters**

- **body** (*Mapping[str, Any]*) – The Action Provider specific input required to execute an Action payload
- **request\_id** (*Optional[str]*) – An optional identifier that serves to de-duplicate requests to the Action Provider

- **manage\_by** (*Optional[Iterable[str]]*) – A series of Globus identities which may alter this Action’s execution. The principal value is the user’s or group’s UUID prefixed with either ‘urn:globus:groups:id:’ or ‘urn:globus:auth:identity:’
- **monitor\_by** (*Optional[Iterable[str]]*) – A series of Globus identities which may view the state of this Action. The principal value is the user’s or group’s UUID prefixed with either ‘urn:globus:groups:id:’ or ‘urn:globus:auth:identity:’
- **force\_path** (*Optional[str]*) – A URL to use for running this action, ignoring any previous configuration
- **label** (*Optional[str]*) – Set a label for the Action that is run.
- **tags** (*Optional[List[str]]*) – A list of tags to associate with the Run.
- **run\_monitors** – May be used as an alias for **monitor\_by**
- **run\_managers** – May be used as an alias for **manage\_by**

**Return type**

globus\_sdk.GlobusHTTPResponse

**status**(*action\_id*)

Query the Action Provider for the status of executed Action

**Parameters**

**action\_id** (*str*) – An identifier that uniquely identifies an Action executed on this Action Provider.

**Return type**

globus\_sdk.GlobusHTTPResponse

**get\_definition**(*action\_id*)

Get the flow definition for a given run ID.

**Parameters**

**action\_id** (*str*) – An identifier that uniquely identifies an Action executed on this Action Provider.

**Return type**

globus\_sdk.GlobusHTTPResponse

**resume**(*action\_id*)

Resume an INACTIVE action. Corrective action must have been taken prior to invoking this method, including the possibility of consenting to additional permissions and using tokens issued by those consents when creating this client. These consents would commonly be required when an Action is INACTIVE and shows the code `ConsentRequired`.

**Parameters**

**action\_id** (*str*) – An identifier that uniquely identifies an Action executed on this Action Provider.

**Return type**

globus\_sdk.GlobusHTTPResponse

**cancel**(*action\_id*)

Cancel a currently executing Action on an Action Provider

**Parameters**

**action\_id** (*str*) – An identifier that uniquely identifies an Action executed on this Action Provider.

**Return type**

globus\_sdk.GlobusHTTPResponse

**release**(*action\_id*)

Remove the history of an Action's execution from an Action Provider

**Parameters**

**action\_id** (*str*) – An identifier that uniquely identifies an Action executed on this Action Provider.

**Return type**

globus\_sdk.GlobusHTTPResponse

**log**(*action\_id*, *limit=10*, *reverse\_order=False*, *marker=None*, *per\_page=None*)

Retrieve an Action's execution log history. Not all Action Providers support this operation.

**Parameters**

- **action\_id** (*str*) – An identifier that uniquely identifies an Action executed on this Action Provider.
- **limit** (*int*) – A integer specifying how many log records to return
- **reverse\_order** (*bool*) – Display the Action states in reverse- chronological order
- **marker** (*Optional[str]*) – A pagination\_token indicating the page of results to return and how many entries to return. Not all ActionProviders will support this parameter.
- **per\_page** (*Optional[int]*) – The number of results to return per page. If supplied a pagination\_token, this parameter has no effect. Not all ActionProviders will support this parameter.

**Return type**

globus\_sdk.GlobusHTTPResponse

**classmethod new\_client**(*action\_url*, *authorizer*, *http\_timeout=10*)

Classmethod to simplify creating an ActionClient. Use this method when attempting to create an Action-Client with pre-existing credentials or authorizers.

**Parameters**

- **action\_url** (*str*) – The url at which the target Action Provider is located.
- **authorizer** (*Optional[globus\_sdk.authorizers.GlobusAuthorizer]*) – The authorizer to use for validating requests to the Action Provider.
- **http\_timeout** (*int*) – The amount of time to wait for connections to the Action Provider to be made.

**Return type**

*\_ActionClient*

**Examples**

```
>>> auth = ...
>>> url = "https://actions.globus.org/hello_world"
>>> ac = ActionClient.new_client(url, auth)
>>> print(ac.run({"echo_string": "Hello from SDK"}))
```

## 7.2 Flows Client

**class** globus\_automate\_client.flows\_client.**FlowsClient**(\*args, \*\*kwargs)

This is a specialized type of the Globus Auth service's `BaseClient` used to interact with the Globus Flows service. Any keyword arguments given are passed through to the `BaseClient` constructor.

### Parameters

- **args** (*Any*) –
- **kwargs** (*Any*) –

### Return type

*Any*

**use\_temporary\_authorizer**(*authorizer*)

Temporarily swap out the authorizer instance variable.

This is a context manager. Use it like this:

```
authorizer = self._get_authorizer_for_flow(...)
with self.alternate_authorizer(authorizer):
    ...
```

**deploy\_flow**(*flow\_definition*, *title*, *subtitle=None*, *description=None*, *keywords=()*, *flow\_viewers=()*, *flow\_starters=()*, *flow\_administrators=()*, *subscription\_id=None*, *input\_schema=None*, *validate\_definition=True*, *validate\_schema=True*, *dry\_run=False*, \*\*kwargs)

Deploys a Flow definition to the Flows service, making the Flow available for execution on the Globus Automate Flows Service.

### Parameters

- **flow\_definition** (*Mapping[str, Any]*) – A mapping corresponding to a Globus Flows definition.
- **title** (*str*) – A simple, human-readable title for the deployed Flow
- **subtitle** (*Optional[str]*) – A longer, more verbose title for the deployed Flow
- **description** (*Optional[str]*) – A long form explanation of the Flow's purpose or usage
- **keywords** (*Iterable[str]*) – A series of words which may help categorize or make the Flow discoverable
- **flow\_viewers** (*Iterable[str]*) – A series of Globus identities which may discover and view the Flow definition
- **flow\_starters** (*Iterable[str]*) – A series of Globus identities which may run an instance of this Flow
- **flow\_administrators** (*Iterable[str]*) – A series of Globus identities which may update this Flow's definition
- **subscription\_id** (*Optional[str]*) – The Globus Subscription which will be used to make this flow managed.
- **input\_schema** (*Optional[Mapping[str, Any]]*) – A mapping representing the JSONSchema used to validate input to this Flow. If not supplied, no validation will be done on input to this Flow.

- **validate\_definition** (*bool*) – Set to True to validate the provided *flow\_definition* before attempting to deploy the Flow.
- **validate\_schema** (*bool*) – Set to True to validate the provided *input\_schema* before attempting to deploy the Flow.
- **dry\_run** (*bool*) – Set to True to test whether the Flow can be deployed successfully.

**Return type**

globus\_sdk.GlobusHTTPResponse

**update\_flow**(*flow\_id*, *flow\_definition=None*, *title=None*, *subtitle=None*, *description=None*, *keywords=None*, *flow\_viewers=None*, *flow\_starters=None*, *flow\_administrators=None*, *subscription\_id=None*, *input\_schema=None*, *validate\_definition=True*, *validate\_schema=True*, *\*\*kwargs*)

Updates a deployed Flow's definition or metadata. This method will preserve the existing Flow's values for fields which are not submitted as part of the update.

**Parameters**

- **flow\_id** (*str*) – The UUID for the Flow that will be updated
- **flow\_definition** (*Optional[Mapping[str, Any]]*) – A mapping corresponding to a Globus Flows definition
- **title** (*Optional[str]*) – A simple, human-readable title for the deployed Flow
- **subtitle** (*Optional[str]*) – A longer, more verbose title for the deployed Flow
- **description** (*Optional[str]*) – A long form explanation of the Flow's purpose or usage
- **keywords** (*Optional[Iterable[str]]*) – A series of words which may help categorize or make the Flow discoverable
- **flow\_viewers** (*Optional[Iterable[str]]*) – A series of Globus identities which may discover and view the Flow definition
- **flow\_starters** (*Optional[Iterable[str]]*) – A series of Globus identities which may run an instance of this Flow
- **flow\_administrators** (*Optional[Iterable[str]]*) – A series of Globus identities which may update this Flow's definition
- **subscription\_id** (*Optional[str]*) – The Globus Subscription which will be used to make this flow managed.
- **input\_schema** (*Optional[Mapping[str, Any]]*) – A mapping representing the JSONSchema used to validate input to this Flow. If not supplied, no validation will be done on input to this Flow.
- **validate\_definition** (*bool*) – Set to True to validate the provided *flow\_definition* before attempting to update the Flow.
- **validate\_schema** (*bool*) – Set to True to validate the provided *input\_schema* before attempting to update the Flow.

**Return type**

globus\_sdk.GlobusHTTPResponse

**get\_flow**(*flow\_id*, *\*\*kwargs*)

Retrieve a deployed Flow's definition and metadata

**Parameters**

- **flow\_id** (*str*) – The UUID identifying the Flow for which to retrieve details

**Return type**

globus\_sdk.GlobusHTTPResponse

**list\_flows**(*roles=None, marker=None, per\_page=None, filters=None, orderings=None, role=None, \*\*kwargs*)

Display all deployed Flows for which you have the selected role(s)

**Parameters**

- **roles** (*Optional[Iterable[str]]*) – Deprecated since version 0.12: Use **role** instead  
See description for **role** parameter. Providing multiple roles behaves as if only a single **role** value is provided and displays the equivalent of the most permissive role.
- **role** (*Optional[str]*) – A role value specifying the minimum role-level permission which will be displayed based on the follow precedence of role values:
  - flow\_viewer
  - flow\_starter
  - flow\_administrators
  - flow\_owner
 Thus, if, for example, **flow\_starter** is specified, flows for which the user has the **flow\_starter**, **flow\_administrator** or **flow\_owner** roles will be returned.
- **marker** (*Optional[str]*) – A pagination\_token indicating the page of results to return and how many entries to return. This is created by the Flows service and returned by operations that support pagination.
- **per\_page** (*Optional[int]*) – The number of results to return per page. If supplied a pagination\_token, this parameter has no effect.
- **filters** (*Optional[dict]*) – A filtering criteria to apply to the resulting Flow listing. The keys indicate the filter, the values indicate the pattern to match. The returned data will be the result of a logical AND between the filters. Patterns may be comma separated to produce the result of a logical OR.
- **orderings** (*Optional[dict]*) – An ordering criteria to apply to the resulting Flow listing. The keys indicate the field to order on, and the value can be either ASC, for ascending order, or DESC, for descending order. The first ordering criteria will be used to sort the data, subsequent ordering criteria will be applied for ties. Note: To ensure orderings are applied in the correct order, use an OrderedDict if trying to apply multiple orderings.

**Return type**

globus\_sdk.GlobusHTTPResponse

**delete\_flow**(*flow\_id, \*\*kwargs*)

Remove a Flow definition and its metadata from the Flows service

**Parameters**

**flow\_id** (*str*) – The UUID identifying the Flow to delete

**Return type**

globus\_sdk.GlobusHTTPResponse

**scope\_for\_flow**(*flow\_id*)

Returns the scope associated with a particular Flow

**Parameters**

**flow\_id** (*str*) – The UUID identifying the Flow's scope to lookup



### Return type

str

**run\_flow**(*flow\_id*, *flow\_scope*, *flow\_input*, *run\_managers*=None, *run\_monitors*=None, *dry\_run*=False, *label*=None, *tags*=None, *\*\*kwargs*)

Run an instance of a deployed Flow with the given input.

### Parameters

- **flow\_id** (*str*) – The UUID identifying the Flow to run
- **flow\_scope** (*Optional[str]*) – The scope associated with the Flow *flow\_id*. If not provided, the SDK will attempt to perform an introspection on the Flow to determine its scope automatically
- **flow\_input** (*Mapping*) – A Flow-specific dictionary specifying the input required for the Flow to run.
- **run\_managers** (*Optional[Iterable[str]]*) – A series of Globus identities which may alter this Flow instance’s execution. The principal value is the user’s or group’s UUID prefixed with either ‘[urn:globus:groups:id:](#)’ or ‘[urn:globus:auth:identity:](#)’
- **run\_monitors** (*Optional[Iterable[str]]*) – A series of Globus identities which may view this Flow instance’s execution state. The principal value is the user’s or group’s UUID prefixed with either ‘[urn:globus:groups:id:](#)’ or ‘[urn:globus:auth:identity:](#)’
- **label** (*Optional[str]*) – An optional label which can be used to identify this run
- **tags** (*Optional[List[str]]*) – Tags that will be associated with this Run.
- **kwargs** – Any additional kwargs passed into this method are passed onto the Globus Base-Client. If there exists an “authorizer” keyword argument, that gets used to run the Flow operation. Otherwise, the *authorizer\_callback* defined for the *FlowsClient* will be used.
- **dry\_run** (*bool*) – Set to True to test what will happen if the Flow is run without actually running the Flow.

### Return type

globus\_sdk.GlobusHTTPResponse

**flow\_action\_status**(*flow\_id*, *flow\_scope*, *flow\_action\_id*, *\*\*kwargs*)

Determine the status for an Action that was launched by a Flow

### Parameters

- **flow\_id** (*str*) – The UUID identifying the Flow which triggered the Action
- **flow\_scope** (*Optional[str]*) – The scope associated with the Flow *flow\_id*. If not provided, the SDK will attempt to perform an introspection on the Flow to determine its scope automatically
- **flow\_action\_id** (*str*) – The ID specifying which Action’s status to query
- **kwargs** – Any additional kwargs passed into this method are passed onto the Globus Base-Client. If there exists an “authorizer” keyword argument, that gets used to run the Flow operation. Otherwise, the *authorizer\_callback* defined for the *FlowsClient* will be used.

### Return type

globus\_sdk.GlobusHTTPResponse

**get\_flow\_definition\_for\_run**(*flow\_id*, *flow\_scope*, *flow\_action\_id*, *\*\*kwargs*)

Determine the status for an Action that was launched by a Flow

### Parameters

- **flow\_id** (*str*) – The UUID identifying the Flow which triggered the Action
- **flow\_scope** (*Optional[str]*) – The scope associated with the Flow *flow\_id*. If not provided, the SDK will attempt to perform an introspection on the Flow to determine its scope automatically
- **flow\_action\_id** (*str*) – The ID specifying which Action’s status to query
- **kwargs** – Any additional kwargs passed into this method are passed onto the Globus Base-Client. If there exists an “authorizer” keyword argument, that gets used to run the Flow operation. Otherwise, the *authorizer\_callback* defined for the *FlowsClient* will be used.

**Return type**

`globus_sdk.GlobusHTTPResponse`

**flow\_action\_resume**(*flow\_id, flow\_scope, flow\_action\_id, \*\*kwargs*)

Resume a Flow Action which is in an INACTIVE state.

**Parameters**

- **flow\_id** (*str*) – The UUID identifying the Flow which triggered the Action
- **flow\_scope** (*Optional[str]*) – The scope associated with the Flow *flow\_id*. If not provided, the SDK will attempt to perform an introspection on the Flow to determine its scope automatically.
- **flow\_action\_id** (*str*) – The ID specifying the Action with an INACTIVE status we want to resume.
- **kwargs** – Any additional kwargs passed into this method are passed onto the Globus Base-Client. If there exists an “authorizer” keyword argument, that gets used to run the Flow operation. Otherwise, the *authorizer\_callback* defined for the *FlowsClient* will be used.

**Return type**

`globus_sdk.GlobusHTTPResponse`

**flow\_action\_release**(*flow\_id, flow\_scope, flow\_action\_id, \*\*kwargs*)

Remove the execution history for an Action that was launched by a Flow

**Parameters**

- **flow\_id** (*str*) – The UUID identifying the Flow which launched the Action
- **flow\_scope** (*Optional[str]*) – The scope associated with the Flow *flow\_id*. If not provided, the SDK will attempt to perform an introspection on the Flow to determine its scope automatically
- **flow\_action\_id** (*str*) – The ID specifying the Action to release
- **kwargs** – Any additional kwargs passed into this method are passed onto the Globus Base-Client. If there exists an “authorizer” keyword argument, that gets used to run the Flow operation. Otherwise, the *authorizer\_callback* defined for the *FlowsClient* will be used.

**Return type**

`globus_sdk.GlobusHTTPResponse`

**flow\_action\_cancel**(*flow\_id, flow\_scope, flow\_action\_id, \*\*kwargs*)

Cancel the execution of an Action that was launched by a Flow

**Parameters**

- **flow\_id** (*str*) – The UUID identifying the Flow which launched the Action

- **flow\_scope** (*Optional[str]*) – The scope associated with the Flow `flow_id`. If not provided, the SDK will attempt to perform an introspection on the Flow to determine its scope automatically
- **flow\_action\_id** (*str*) – The ID specifying the Action we want to cancel
- **kwargs** – Any additional kwargs passed into this method are passed onto the Globus Base-Client. If there exists an “authorizer” keyword argument, that gets used to run the Flow operation. Otherwise, the `authorizer_callback` defined for the `FlowsClient` will be used.

#### Return type

`globus_sdk.GlobusHTTPResponse`

**enumerate\_runs** (*roles=None, statuses=None, marker=None, per\_page=None, filters=None, orderings=None, role=None, \*\*kwargs*)

Retrieve a listing of Runs the caller has access to. This operation requires the supplied Authorizer to have the `RUN_STATUS_SCOPE`.

#### Parameters

- **statuses** (*Optional[Iterable[str]]*) – A list of statuses used to filter the Actions that are returned by the listing. Returned Actions are guaranteed to have one of the specified statuses. Valid values are:
  - `SUCCEEDED`
  - `FAILED`
  - `ACTIVE`
  - `INACTIVE`
- **roles** (*Optional[Iterable[str]]*) – Deprecated since version 0.12: Use `role` instead  
See description for `role` parameter. Providing multiple roles behaves as if only a single `role` value is provided and displays the equivalent of the most permissive role.
- **role** (*Optional[str]*) – A role value specifying the minimum role-level permission on the runs which will be returned based on the follow precedence of role values:
  - `run_monitor`
  - `run_manager`
  - `run_owner`

Thus, if, for example, `run_manager` is specified, runs for which the user has the `run_manager`, or `run_owner` roles will be returned.

- **marker** (*Optional[str]*) – A `pagination_token` indicating the page of results to return and how many entries to return. This is created by the Flows service and returned by operations that support pagination.
- **per\_page** (*Optional[int]*) – The number of results to return per page. If supplied a `pagination_token`, this parameter has no effect.
- **filters** (*Optional[dict]*) – A filtering criteria to apply to the resulting Action listing. The keys indicate the filter, the values indicate the pattern to match. The returned data will be the result of a logical AND between the filters. Patterns may be comma separated to produce the result of a logical OR.
- **orderings** (*Optional[dict]*) – An ordering criteria to apply to the resulting Action listing. The keys indicate the field to order on, and the value can be either `ASC`, for ascending order, or `DESC`, for descending order. The first ordering criteria will be used to sort the

data, subsequent ordering criteria will be applied for ties. Note: To ensure orderings are applied in the correct order, use an OrderedDict if trying to apply multiple orderings.

**Return type**

globus\_sdk.GlobusHTTPResponse

**enumerate\_actions**(*\*\*kwargs*)

An alias for `enumerate_runs`

**Return type**

globus\_sdk.GlobusHTTPResponse

**list\_flow\_actions**(*\*\*kwargs*)

An alias for `list_flow_runs`

**Return type**

globus\_sdk.GlobusHTTPResponse

**list\_flow\_runs**(*flow\_id=None, flow\_scope=None, statuses=None, roles=None, marker=None, per\_page=None, filters=None, orderings=None, role=None, \*\*kwargs*)

List all Runs for a particular Flow.

If no `flow_id` is provided, all runs for all Flows will be returned.

**Parameters**

- **flow\_id** (*Optional[str]*) – The UUID identifying the Flow which launched the Run. If not provided, all runs will be returned regardless of which Flow was used to start the Run (equivalent to `enumerate_runs`).
- **flow\_scope** (*Optional[str]*) – The scope associated with the Flow `flow_id`. If not provided, the SDK will attempt to perform an introspection on the Flow to determine its scope automatically
- **statuses** (*Optional[Iterable[str]]*) – The same as in `enumerate_runs`.
- **roles** (*Optional[Iterable[str]]*) – Deprecated since version 0.12: Use `role` instead. The same as in `enumerate_runs`.
- **marker** (*Optional[str]*) – The same as in `enumerate_runs`.
- **per\_page** (*Optional[int]*) – The same as in `enumerate_runs`.
- **filters** (*Optional[dict]*) – The same as in `enumerate_runs`.
- **orderings** (*Optional[dict]*) – The same as in `enumerate_runs`.
- **role** (*Optional[str]*) – The same as in `enumerate_runs`.
- **kwargs** – Any additional kwargs passed into this method are passed onto the Globus Base-Client. If there exists an “authorizer” keyword argument, that gets used to run the Flow operation. Otherwise, the `authorizer_callback` defined for the `FlowsClient` will be used.

**Return type**

globus\_sdk.GlobusHTTPResponse

**flow\_action\_update**(*action\_id, run\_managers=None, run\_monitors=None, tags=None, label=None, \*\*kwargs*)

Update a Flow Action.

**Parameters**

- **action\_id** (*str*) – The UUID identifying the Action to update

- **run\_managers** (*Optional[Sequence[str]]*) – A list of Globus Auth URNs which will have the ability to alter the execution of the Action. Supplying an empty list will remove all existing managers.
- **run\_monitors** (*Optional[Sequence[str]]*) – A list of Globus Auth URNs which will have the ability to view the execution of the Action. Supplying an empty list will remove all existing monitors.
- **kwargs** – Any additional kwargs passed into this method are passed onto the Globus Base-Client. If there exists an “authorizer” keyword argument, that gets used to run the Flow operation. Otherwise, the `authorizer_callback` defined for the `FlowsClient` will be used.
- **tags** (*Optional[Sequence[str]]*) – A list of tags to apply to the Run.
- **label** (*Optional[str]*) – A label to apply to the Run.

#### Return type

`globus_sdk.GlobusHTTPResponse`

**update\_runs**(*run\_ids, add\_tags=None, remove\_tags=None, set\_tags=None, add\_run\_managers=None, remove\_run\_managers=None, set\_run\_managers=None, add\_run\_monitors=None, remove\_run\_monitors=None, set\_run\_monitors=None, status=None, \*\*kwargs*)

Update a Flow Action.

#### Parameters

- **run\_ids** (*Iterable[str]*) – A list of Run ID’s to query.
- **set\_tags** (*Optional[Iterable[str]]*) – A list of tags to set on the specified Run ID’s. If the list is empty, all tags will be deleted from the specified Run ID’s.

---

**Note:** The `set_tags`, `add_tags`, and `remove_tags` arguments are mutually exclusive.

---

- **add\_tags** (*Optional[Iterable[str]]*) – A list of tags to add to each of the specified Run ID’s.

---

**Note:** The `set_tags`, `add_tags`, and `remove_tags` arguments are mutually exclusive.

---

- **remove\_tags** (*Optional[Iterable[str]]*) – A list of tags to remove from each of the specified Run ID’s.

---

**Note:** The `set_tags`, `add_tags`, and `remove_tags` arguments are mutually exclusive.

---

- **set\_run\_managers** (*Optional[Iterable[str]]*) – A list of Globus Auth URN’s to set on the specified Run ID’s.

If the list is empty, all Run managers will be deleted from the specified Run ID’s.

---

**Note:** The `set_run_managers`, `add_run_managers`, and `remove_run_managers` arguments are mutually exclusive.

---

- **add\_run\_managers** (*Optional[Iterable[str]]*) – A list of Globus Auth URN’s to add to each of the specified Run ID’s.

---

**Note:** The `set_run_managers`, `add_run_managers`, and `remove_run_managers` arguments are mutually exclusive.

---

- **remove\_run\_managers** (*Optional[Iterable[str]]*) – A list of Globus Auth URN's to remove from each of the specified Run ID's.

---

**Note:** The `set_run_managers`, `add_run_managers`, and `remove_run_managers` arguments are mutually exclusive.

---

- **set\_run\_monitors** (*Optional[Iterable[str]]*) – A list of Globus Auth URN's to set on the specified Run ID's.

If the list is empty, all Run monitors will be deleted from the specified Run ID's.

---

**Note:** The `set_run_monitors`, `add_run_monitors`, and `remove_run_monitors` arguments are mutually exclusive.

---

- **add\_run\_monitors** (*Optional[Iterable[str]]*) – A list of Globus Auth URN's to add to each of the specified Run ID's.

---

**Note:** The `set_run_monitors`, `add_run_monitors`, and `remove_run_monitors` arguments are mutually exclusive.

---

- **remove\_run\_monitors** (*Optional[Iterable[str]]*) – A list of Globus Auth URN's to remove from each of the specified Run ID's.

---

**Note:** The `set_run_monitors`, `add_run_monitors`, and `remove_run_monitors` arguments are mutually exclusive.

---

- **status** (*Optional[str]*) – A status to set for all specified Run ID's.
- **kwargs** – Any additional keyword arguments passed to this method are passed to the Globus BaseClient.

If an “authorizer” keyword argument is passed, it will be used to authorize the Flow operation. Otherwise, the `authorizer_callback` defined for the `FlowsClient` will be used.

#### Raises

**ValueError** – If more than one mutually-exclusive argument is provided. For example, if `set_tags` and `add_tags` are both specified, or if `add_run_managers` and `remove_run_managers` are both specified.

#### Return type

`globus_sdk.GlobusHTTPResponse`

**flow\_action\_log**(*flow\_id, flow\_scope, flow\_action\_id, limit=10, reverse\_order=False, marker=None, per\_page=None, \*\*kwargs*)

Retrieve an Action's execution log history for an Action that was launched by a specific Flow.

#### Parameters

- **flow\_id** (*str*) – The UUID identifying the Flow which launched the Action

- **flow\_scope** (*str*) – The scope associated with the Flow *flow\_id*. If not provided, the SDK will attempt to perform an introspection on the Flow to determine its scope automatically
- **flow\_action\_id** (*str*) – The ID specifying which Action’s history to query
- **limit** (*int*) – An integer specifying the maximum number of records for the Action’s execution history to return.
- **reverse\_order** (*bool*) – An indicator for whether to retrieve the records in reverse-chronological order.
- **marker** (*Optional[str]*) – A pagination\_token indicating the page of results to return and how many entries to return. This is created by the Flows service and returned by operations that support pagination.
- **per\_page** (*Optional[int]*) – The number of results to return per page. If supplied a pagination\_token, this parameter has no effect.
- **kwargs** – Any additional kwargs passed into this method are passed onto the Globus Base-Client. If there exists an “authorizer” keyword argument, that gets used to run the Flow operation. Otherwise, the *authorizer\_callback* defined for the *FlowsClient* will be used.

#### Return type

`globus_sdk.GlobusHTTPResponse`

**classmethod new\_client**(*client\_id*, *authorizer\_callback*, *authorizer*, *base\_url=None*, *http\_timeout=10*)

Classmethod to simplify creating an *FlowsClient*. Use this method when attempting to create a *FlowsClient* with pre-existing credentials or authorizers. This method is useful when creating a *FlowClient* for interacting with a Flow without wanting to launch an interactive login process.

#### Parameters

- **client\_id** (*str*) – The *client\_id* to associate with this *FlowsClient*.
- **authorizer\_callback** (*Callable[[...], Union[globus\_sdk.AccessTokenAuthorizer, globus\_sdk.RefreshTokenAuthorizer, globus\_sdk.ClientCredentialsAuthorizer]]*) – A callable which is capable of returning an authorizer for a particular Flow. The callback should accept three keyword-args: *flow\_url*, *flow\_scope*, *client\_id*. Using some, all, or none of these args, the callback should return a *GlobusAuthorizer* which provides access to the targetted Flow.
- **authorizer** (*Optional[globus\_sdk.authorizers.GlobusAuthorizer]*) – The authorizer to use for validating requests to the Flows service. This authorizer is used when interacting with the Flow’s service, it is not used for interactive with a particular flow. Therefore, this authorizer should grant consent to the *MANAGE\_FLOWS\_SCOPE*. For interacting with a particular flow, set the *authorizer\_callback* parameter.
- **base\_url** (*Optional[str]*) – The url at which the target Action Provider is located.
- **http\_timeout** (*int*) – The amount of time to wait for connections to the Action Provider to be made.

#### Return type

`_FlowsClient`

#### Examples

```
>>> def cli_authorizer_callback(**kwargs):
    flow_url = kwargs["flow_url"]
    flow_scope = kwargs["flow_scope"]
```

(continues on next page)

(continued from previous page)

```

        client_id = kwargs["client_id"]
        return get_cli_authorizer(flow_url, flow_scope, client_id)
>>> action_url = "https://actions.globus.org/hello_world"
>>> client_id = "00000000-0000-0000-0000-000000000000"
>>> auth = ...
>>> fc = FlowsClient.new_client(client_id, cli_authorizer_callback, auth)
>>> print(fc.list_flows())

```

## 7.3 Helpers

If you're running the SDK locally in a secure location, you can use the provided convenience functions, `create_action_client` and `create_flows_client`, to create authenticated `ActionClient` and `FlowsClient` instances. These helpers share functionality with the `globus-automate` CLI to handle loading and storing Globus Auth tokens on the local filesystem. They will also trigger interactive logins when additional consents are required to interact with actions or flows.

```

globus_automate_client.create_action_client(action_url, action_scope=None,
                                           client_id='e6c75d97-532a-4c88-b031-8584a319fa3e')

```

A helper function to handle creating a properly authenticated `ActionClient` which can operate against *Globus Action Provider Interface* compliant Action Providers. This helper will create an `ActionClient` by searching for and storing tokens on the local filesystem, potentially triggering a log-in flow if the requested tokens are not found locally.

Given the `action_url` for a specific `ActionProvider`, this function will attempt to create a valid `ActionClient` for interacting with that `ActionProvider`. If the `action_scope` is not provided, this function will attempt to discover the `action_scope` by querying the target Action Provider's introspection endpoint. If the Action Provider is not configured to allow public, unauthenticated access to its introspection endpoint, the `action_scope` will be non-discoverable and authentication will fail.

With the `action_scope` available, the function will search for a valid token in the local filesystem cache. In the event that tokens for the scope cannot be loaded, an interactive login will be triggered. Once tokens have been loaded, an Authorizer is created and used to instantiate the `ActionClient` which can be used for operations against that Action Provider.

### Parameters

- **action\_url** (*str*) – The URL address at which the target Action Provider exists
- **action\_scope** (*Optional[str]*) – The target Action Provider's Globus Auth Scope used for authenticating access to it
- **client\_id** (*str*) – The ID for the Native App Auth Client which will be triggering the login flow for this `ActionClient`

### Return type

`ActionClient`

### Examples

```

>>> from globus_automate_client import create_action_client
>>> # Create an ActionClient for the HelloWorld Action
>>> ac = create_action_client("https://actions.globus.org/hello_world")
>>> # Run an Action and check its results

```

(continues on next page)



(continued from previous page)

```
>>> resp = ac.run({"echo_string": "Hello from SDK"})
>>> assert resp.data["status"] == "SUCCEEDED"
```

```
globus_automate_client.create_flows_client(client_id='e6c75d97-532a-4c88-b031-8584a319fa3e',
                                           base_url='https://flows.globus.org',
                                           scope='https://auth.globus.org/scopes/eec9b274-0c81-4334-bdc2-54e90e689b9a/manage_flows', *, authorizer=None,
                                           authorizer_callback=<function cli_authorizer_callback>,
                                           http_timeout=10)
```

A helper function to handle creating a properly authenticated `FlowsClient` which can operate against the Globus Automate Flows service. This function will attempt to load tokens for the `MANAGE_FLOWS_SCOPE` from the local filesystem, triggering a log-in if the requested tokens are not found locally. Once tokens have been loaded, an `Authorizer` is created and used to instantiate the `FlowsClient`. Attempts to interact with a specific Flow will similarly search for valid tokens in the local cache, triggering an interactive log-in if they cannot be found.

#### Parameters

- **scope** (*str*) – The Globus Auth scope to which the `FlowsClient` should be created with consents to
- **client\_id** (*str*) – The Globus ID to associate with this instance of the `FlowsClient`
- **base\_url** (*str*) – The URL at which the Globus Automate Flows service is located
- **authorizer** (*Optional[globus\_sdk.authorizers.GlobusAuthorizer]*) – An authorizer providing access to the Flows service. If not provided, it will be created using the `authorizer_callback`
- **authorizer\_callback** (*Callable*) – A callback used to dynamically return `GlobusAuthorizers`. If not provided, the Globus Automate CLI callback will be used which triggers interactive logins and stores tokens locally
- **http\_timeout** (*int*) – Close any requests taking longer than this parameter's value

#### Return type

`FlowsClient`

#### Examples

```
>>> from globus_automate_client import create_flows_client
>>> # Create an authenticated FlowsClient that can run operations against the Flows
>>> # service
>>> fc = create_flows_client()
>>> # Get a listing of runnable, deployed flows
>>> available_flows = fc.list_flows(["runnable_by"])
>>> for flow in available_flows.data["flows"]:
>>>     print(flow)
```

## 7.4 Examples

### 7.4.1 Non-interactive Operations

There are plenty of scenarios in which the `create_action_client` and `create_flows_client` helper functions should *NOT* be used, included in those are:

- The SDK is being run on a platform without access to the local filesystem (or it is preferable not to write to the local filesystem).
- You do not want to (or cannot) trigger interactive logins during execution.
- The SDK is being used as part of another service or in a pipeline

In this case, you'll need to create the clients using the `new_client` classmethod attached to each client. When creating an `ActionClient`, you'll need to create and supply the appropriate `GlobusAuthorizer` for the action.

When creating a `FlowsClient`, you'll need to create a `GlobusAuthorizer` with access to the `MANAGE_FLOWS_SCOPE`. This authorizer will be used to authenticate interactions against the `Flows` API, such as creating a new Flow, updating an existing Flow, or deleting an old Flow. Additionally, you'll need to create a callback that accepts three keyword-arguments, `flow_url`, `flow_scope` and `client_id` and returns a `GlobusAuthorizer` that will be used to provide access to specific Flows. This authorizer allows the SDK to run operations against the Flow, such as running a Flow and checking an execution's status.

The example below shows an example defining the callback used to create a `GlobusAuthorizer` by pulling tokens from environment variables.

```
import os

from globus_sdk import AccessTokenAuthorizer

from globus_automate_client import FlowsClient
from globus_automate_client.cli.auth import CLIENT_ID
from globus_automate_client.flows_client import AllowedAuthorizersType


def authorizer_retriever(
    flow_url: str, flow_scope: str, client_id: str
) -> AllowedAuthorizersType:
    """
    This callback will be called when attempting to interact with a
    specific Flow. The callback will receive the Flow url, Flow scope, and
    client_id and can choose to use some, all or none of the kwargs. This is
    expected to return an Authorizer which can be used to make authenticated
    calls to the Flow.

    The method used to acquire valid credentials is up to the user. Here, we
    naively create an Authorizer using the same token everytime.
    """
    flow_token = os.environ.get("MY_ACCESS_TOKEN", "")
    return AccessTokenAuthorizer(flow_token)


# Create an AccessTokenAuthorizer using a token that has consents to the
# MANAGE_FLOWS_SCOPE. This lets the FlowsClient perform operations against the
```

(continues on next page)

(continued from previous page)

```
# Flow's service i.e. create flow, update a flow, delete a flow
flows_service_token = os.environ.get("MANAGE_FLOWS_SCOPED_TOKEN", "")
flows_service_authorizer = AccessTokenAuthorizer(flows_service_token)

fc = FlowsClient.new_client(
    client_id=CLIENT_ID,
    authorizer=flows_service_authorizer,
    authorizer_callback=authorizer_retriever,
)

my_flows = fc.list_flows()
print(my_flows)

# When running a specific Flow, the authorizer_retriever callback is called
# internally to make the authenticated call to the Flow
running_flow = fc.run_flow(
    "1e6b4406-ee3d-4bc5-9198-74128e108111", None, {"echo_string": "hey"}
)
print(running_flow)

# It's possible to create an Authorizer and pass it as a kwarg to the flow
# operation. This usage will not use the authorizer_callback:
running_flow_2 = fc.run_flow(
    "1e6b4406-ee3d-4bc5-9198-74128e108111",
    None,
    {"echo_string": "hey"},
    authorizer=AccessTokenAuthorizer("..."),
)
print(running_flow_2)
```



## EXAMPLE FLOWS

For examples of flows, input schemas, and inputs see the [examples in the Globus Automate Client repository](#).



## CHANGELOG

### 9.1 Unreleased changes

Unreleased changes are documented in files in the `changelog.d` directory.

### 9.2 0.17.0 — 2023-03-16

#### 9.2.1 Features

- Add official support for python3.11
- Support retrieval of the flow definition and input schema used to start a run.

This is supported in the new `flow run-definition` subcommand.

#### 9.2.2 Bugfixes

- Fix a bug which could lead to improper handling of token refreshes
- [sc-19291] Prevent passive flow permission updates.

Previously, permissions were erased during updates if no permissions were specified. Now, permissions will only be erased if an empty string is passed. For example:

```
# Erase all viewer permissions.
globus-automate flow update $UUID --flow-viewer=""
```

#### 9.2.3 Documentation

- Fix example flow definition `web-option-with-parameters`
- Improve documentation by normalizing capitalization and emphasis, as well as clarifying usage of “Globus Automate” wherever it occurred.
- Add simplified versions of our production flows as inline examples on the “Authoring Flows” page.

## 9.3 0.16.1.post1 — 2022-07-15

### 9.3.1 Documentation

- [sc-16711] Document how to automatically run a flow based on filesystem change events.

## 9.4 0.16.1 — 2022-06-23

### 9.4.1 Bugfixes

- Fix a bug in the flow administrator/starter/runner CLI alias validation which prevents successfully running the flow update subcommand.

## 9.5 0.16.0 — 2022-06-23

### 9.5.1 Changes

- [sc-13892] Standardize flow viewer, starter, and administrator arguments – and their aliases – throughout the CLI and API.

Although this change fixes bugs that prevented documented CLI options from working, it also introduces breaking changes:

- `--flow-viewer`, `--flow-starter`, and `--flow-administrator` are the canonical CLI options.
- The CLI option aliases will continue to work, but the alias behavior has changed:
  - \* If an alias is used, a warning is written to `STDERR`.
  - \* If the canonical option name is combined with one of its aliases, an error will be written to `STDERR` and the Automate client will exit. The exit code will be 1.
  - \* If more than one of the acceptable aliases is used (like `--starter` and `--runnable-by`), an error will be written to `STDERR` and the Automate client will exit. The exit code will be 1.

In addition, there are breaking changes in the API:

- `flow_viewers`, `flow_starters`, and `flow_administrators` are the canonical API argument names.
- The API argument aliases will continue to work, but the alias behavior has changed:
  - \* If an alias is used, a Python warning will be issued. Applications can control the warning behavior using Python's `warnings` module.
  - \* If the canonical option name is combined with one of its aliases, a `ValueError` will be raised.
  - \* If more than one of the acceptable aliases is used (like `starters` and `runnable_by`), a `ValueError` will be raised.



## 9.5.2 Bugfixes

- Update pyjwt to version 2.4.0 to address [CVE-2022-29217](#).

## 9.6 0.15.2 — 2022-06-06

- [\[sc-16137\]](#) Fix flows run dry-run URL path

## 9.7 0.15.1 — 2022-05-09

### 9.7.1 Bugfixes

- Fix a missing dependency when running on Python 3.10.

This was fixed by adding typing-extensions as an explicit dependency. Note that pip may need to be upgraded due to changes in its dependency resolver.

## 9.8 0.15.0.post1 — 2022-05-09

### NOTE:

This release contains no changes from 0.15.0. It adds text to the changelog regarding an update to the rich package dependency.

### 9.8.1 Documentation

- [\[sc-13642\]](#) Provide two examples of looping in flow definitions:
  - How to loop a set number of times
  - How to perform batch processing over an unknown quantity of items

### 9.8.2 Development

- Update click to version 8.0.4. This resolves a security issue.
- Update typer to version 0.4.1.
- Update scriv to version 0.14.0. scriv is a development dependency.
- Update rich to 0.12.3.

This resolves a dependency conflict between the Globus CLI and the Globus Automate CLI. Both command line clients can now be installed in the same environment.

- Temporarily remove typer-cli as a listed development dependency. It is still needed when generating the CLI documentation.
- Add safety as a test environment for local and CI testing.
- Test against Python 3.10 in CI.
- [\[sc-14485\]](#) Disable SSL verification when interacting with a local development server.

## 9.9 0.15.0 — 2022-04-29

### 9.9.1 Documentation

- [\[sc-13642\]](#) Provide two examples of looping in flow definitions:
  - How to loop a set number of times
  - How to perform batch processing over an unknown quantity of items

### 9.9.2 Development

- Update click to version 8.0.4. This resolves a security issue.
- Update typer to version 0.4.1.
- Update scriv to version 0.14.0. scriv is a development dependency.
- Temporarily remove typer-cli as a listed development dependency. It is still needed when generating the CLI documentation.
- Add safety as a test environment for local and CI testing.
- Test against Python 3.10 in CI.
- [\[sc-14485\]](#) Disable SSL verification when interacting with a local development server.

## 9.10 0.14.1 — 2022-04-13

### 9.10.1 Bugfixes

- Changed text on improper token cache file condition so that it doesn't reference the Timer CLI

## 9.11 0.14.0 — 2022-03-25

### 9.11.1 Features

- [\[sc-13426\]](#) Support setting tags when using the `flow run` subcommand.
- Support batch updates of one or more Runs.
- Support updating tags and labels using the `flow run-update` subcommand.
- Support erasing the list of Run managers and Run monitors using the `flow run-update` subcommand. This can be done by specifying an empty string for the value of the `--run-manager` and `--run-monitor` options.

### 9.11.2 Bugfixes

- [sc-13664] Fix tabular `run-list` output.
- [sc-14109] Mark the `run-status` subcommand's `--flow-id` option as a mandatory UUID.
- [sc-14127] Prevent a validation error that occurs when an input schema is not provided to the `flow deploy` subcommand.

## 9.12 0.13.1 — 2022-03-02

### 9.12.1 Bugfixes

- Output login prompts to `STDERR`. This protects serialized output to `STDOUT` so it can be piped to tools like *jq*.

### 9.12.2 Documentation

- Documentation and examples for the `globus-collection` input schema format.

## 9.13 0.13.0 — 2022-02-11

### 9.13.1 Documentation

- Add the `"notify_on_*`" parameters to the transfer action provider JSON example.
- The description of the Action polling policy has been updated and a discussion of how caching of token validation checks may impact users who invalidate their tokens has been added.
- Adds an input schema for the example single-transfer Flow definition.
- Add documentation for *globus-collection-id* and *globus-collection-path* formats

## 9.14 0.13.0b2 — 2021-12-09

### 9.14.1 Bugfixes

- Fix a `KeyError` crash that occurs when enabling verbose output using the `-v` argument. (#111)
- Fix a `ValueError` crash that occurs when displaying a flow. (#110)

## 9.15 0.13.0b1 — 2021-12-09

### 9.15.1 Features

- Upgrade to Globus SDK v3.

### 9.15.2 Bugfixes

- Fixes a bug in the SDK that prevented Flow updates from removing all `flow_administrators`, `flow_viewers`, and `flow_starters`. This bug also prevented updates from setting text fields to empty strings.
- Fix a bug that could allow the Flows authorizer to be lost if an exception was raised. (Authorizer swaps are now handled using a context manager.)
- Support strings (and tuples/sets containing strings) as argument values when running, deploying, or updating an action or a flow and specifying a keyword argument alias like `visible_to` or `runnable_by`.

### 9.15.3 Other

- Add code linting, documentation build testing, and a bunch of unit tests.
- Add GitHub Actions to run on push and pull requests.
- Add a pre-commit configuration file to increase overall code quality.

## 9.16 0.12.3 — 2021-11-10

### 9.16.1 Bugfixes

- Fix a bug that prevented the Flows client from properly validating flow definition states in lists.
- Prevent empty values from appearing in query parameters or JSON submissions.
- Fix a bug that prevented the input schema of an existing Flow from being set to an all-permissive JSON schema.
- Fix a bug that prevented a custom authorizer from being used if attempting to list all runs of a specific flow without specifying the flow ID.
- Fix a bug that introduced duplicate forward slashes in some API URL's.

### 9.16.2 Documentation

- Add a CHANGELOG and include it in the documentation.
- Use scriv for CHANGELOG management.
- Added documentation for the new Action Providers for: - Make a directory via Globus Transfer (`mkdir`) - Get collection information via Globus Transfer (`collection_info`)
- Added documentation for new feature of the list directory Action Provider to get information only about the path provided as input
- Added documentation related to use of built-in functions in expressions. Documented the new functions `pathsplit`, `is_present` and `getattr`.

## 9.17 0.12.2 — 2021-10-19

### 9.17.1 Features

- The output of `globus-automate flow list` is modified to ensure that the Flow ID is always visible. The new output search is similar to the output of `globus endpoint search`.
- The output when watching the results of a `globus-automate flow run now` defaults to table view.

### 9.17.2 Bugfixes

- Fixes an infinite loop when watching the output of `flow action-log/flow run-log` with the `--reverse` flag.
- Fixes the limit minimum level from 0 to 1 when doing `flow action-log/flow run-log` to prevent server errors.
- Fixes a bug where an unsafe indexing operation was being made during `flow action-list/flow run-list`

### 9.17.3 Documentation

- CLI documentation is updated to more precisely reflect that `--label` is a required property when running a Flow.

## 9.18 0.12.1 — 2021-09-14

### 9.18.1 Features

- CLI commands which output lists of data now display a subset of the data fields as a table. For access to the full data or to access data in JSON or YAML format, the `-f json | yaml` option may be used. The tabular output is on the following commands: `- globus-automate flow list - globus-automate flow action-list ... - globus-automate flow action-enumerate ... - globus-automate flow action-log ...`
- File inputs to CLI commands may now be in either JSON or YAML formatting without the need to specify the input file format.

### 9.18.2 Bugfixes

- Fixed an error where the output of the `globus-automate flow update` command displayed unformatted JSON

### 9.18.3 Documentation

- Added text explaining that the Fail state is a supported state type and can be used in Flows. A simple example using the Fail state is included in the examples directory for the repository.

## 9.19 0.12.0 — 2021-08-16

### 9.19.1 Features

- CLI and SDK support for updating user roles on new and existing Runs
- Wherever identities are referenced on the CLI we now support supplying Globus Auth usernames instead.
- Updates to CLI and SDK arguments to more closely reflect RBAC updates in the Flows service.

### 9.19.2 Bugfixes

- The Run enumeration CLI and SDK methods would attempt to use the Flow manage authorizer to authenticate its calls. This method has been updated to instead look up or create an authorizer for the RUN\_STATUS scope

### 9.19.3 Documentation

- The RBAC system for the Flows service has been updated to follow a subset model rather than the previously existing separate permissions model. The documentation has been updated with [a description of the new behavior](#).

## 9.20 0.11.5 — 2021-06-17

### 9.20.1 Features

- Adds SDK and CLI support for dry running a Flow deploy or Flow run
- Adds SDK + CLI commands for enumerating Actions and sorting/filtering through results
- Adds a CLI command to retrieve a single Flow definition and its metadata: `globus-automate flow get <id>`
- Expands the use of the `create_flows_client` function to allow specifying an authorizer, an authorizer callback, and a `http_timeout`.

### 9.20.2 Bugfixes

- Fixes a regression where Flow deploy results via the CLI were unformatted
- Adds license to output of `pip show globus-automate-client`

### 9.20.3 Documentation

- Fixes an issue where `FlowsClient` and `ActionClient` auto-generated docs were not getting generated
- Adds references to exemplar Flows and their inputs
- Adds input examples to Action Provider reference page
- Adds a hosted CLI reference

## 9.21 0.11.4 — 2021-05-10

### 9.21.1 Features

- The CLI and SDK now allow Subscription IDs to be associated with Flows

### 9.21.2 Bugfixes

- The Flow List CLI and SDK operations were sending malformed query arguments to the API, which produced incorrect results when trying to filter based on role. This release corrects the behavior.

## 9.22 0.11.3 — 2021-05-04

### 9.22.1 Bugfixes

- Reformats verbose output to make the separation between request information and request results more obvious
- Verbose output writes output to `stderr` to allow output to be parsed as JSON
- Empty query arguments are not sent as part of the Flows API request

### 9.22.2 Documentation

- Typo fixes

## 9.23 0.11.1 — 2021-04-08

### 9.23.1 Features

- `flow display` can now visualize local Flow definitions and deployed Flows.

### 9.23.2 Bugfixes

- Fixes an issue where the Globus Auth login link was being rendered as a non-clickable link.
- Fixes an issue where the prompt for inputting the Globus Auth auth code was disappearing.

### 9.23.3 Documentation

- Adds explanation and examples for how to use `manage_by` and `monitor_by` values on Actions and Flow runs to delegate access to other identities.
- Clarifies the expected format for provided identities.
- Explicitly adds `manage_by` and `monitor_by` as parameters to the `FlowsClient.run` method.

## 9.24 0.11.0 — 2021-03-29

### 9.24.1 Features

- Export the `validate_flow_definition` function which can be used to perform a local JSONSchema based validation of a Flow definition.
- Using `create_flows_client` no longer requires the use of a `CLIENT_ID`.
- The `action run`, `action status`, `flow run`, `flow status`, and `flow log` CLI commands implement a new `--watch` flag which lets you stream an Action's status updates.
- CLI and SDK level support for filtering and ordering Flow Listing and Flow Action Enumerations endpoints [preview].
- New CLI commands to facilitate the following Globus Auth operations: - `session whoami` - determine the caller's user information as it exists in Auth - `session logout` - remove locally cached auth state - `session revoke` - invalidate local tokens and remove locally cached auth state.

### 9.24.2 Documentation

- Various typo fixes.

## 9.25 0.10.7 — 2021-02-11

### 9.25.1 Features

- Improved error handling on CLI operations so that users receive formatted output instead of `GlobusAPIError` tracebacks.
- Added CLI and SDK level support for using `labels` to launch Flows and Actions.



## 9.25.2 Documentation

- Removes references to `ActionScope` from example Flow definitions because the Flows service handles the scope lookups.

## 9.25.3 Bugfixes

- The Flows CLI interface would attempt to load empty arguments, resulting in `NoneType` errors. Empty arguments are now ignored.
- When using the CLI with the `--verbose` flag, the results of the verbosity are printed to `stderr`, allowing the commands outputs to still be parsed by other tools, such as `jq`.
- Fixes a `NameTooLong` exception that was thrown when the CLI attempted to parse long JSON strings as filenames.

## 9.26 0.10.6 — 2021-01-27

### 9.26.1 Features

- Adds support for YAML formatted input when defining Flows, input schemas, and inputs via the CLI.

### 9.26.2 Documentation

- Improves documentation around manually creating authorizers and how to use them to create `ActionClients` and `FlowsClient`: [https://globus-automate-client.readthedocs.io/en/latest/python\\_sdk.html#sdk-the-hard-way](https://globus-automate-client.readthedocs.io/en/latest/python_sdk.html#sdk-the-hard-way)
- Adds examples for Flow definitions as YAML: <https://github.com/globus/globus-automate-client/tree/main/examples/flows/hello-world-yaml>

## 9.27 0.10.5 — 2020-12-11

### 9.27.1 Features

- Removes custom SSH session detection in favor of using fair-research native-login's SSH session detection
- Adds Flows pagination support to CLI and SDK layers
- Fully decouples the SDK from the CLI. SDK users can now opt to supply their own authorizers for Flow operations, either as a kwarg to the operation or as a callback to the `FlowsClient` which should be used to lookup the appropriate authorizer.

### 9.27.2 Documentation

- Fixes typos in Flow’s documentation where `Private_Parameters` were incorrectly referenced as `Private_Properties`
- Publishes a new example Flow for performing a multi-step Transfer & delete, along with error checking

## 9.28 0.10.4 — 2020-10-01

### 9.28.1 Features

- Added support for deleting messages off a Globus Queue to the CLI and SDK
- Adds example action bodies to the repository for running an action on the new Search Delete Action Provider
- Updated docs and example action bodies for running an action on the Set Permissions Action Provider
- Updates the schema validation for the Pass State to make `Parameters` and `InputPath` optional.

### 9.28.2 Bugfixes

- Corrected an issue in CLI option validation where “public” and “all\_authenticated\_users” were not being accepted
- Corrected an issue where the SDK’s `ActionClient` was setting `monitor_by` and `manage_by` to `None` by default, thus failing Action Provider schema validation.

## 9.29 0.10 — 2020-08-24

This release is the first based on the public `globus-automate-client` repository. Compared to previous PyPi releases, this release contains:

- A more complete set of documentation which is also published to `readthedocs`
- A set of examples under the `examples` directory
- Client side validation of flow definitions based on a `jsonschema`. This is somewhat experimental at this point, and feedback is welcome on experience both with the accuracy and the helpfulness of the reported errors. Validation is turned on by default when deploying or linting a flow, but can be turned off with the SDK parameter `validate_definition` and the CLI `--validate/no-validate` flags.

## PYTHON MODULE INDEX

### g

globus\_automate\_client, [46](#)



## INDEX

### A

`action_scope` (`globus_automate_client.action_client.ActionClient` property), 47  
`ActionClient` (class in `FlowsClient` in `globus_automate_client.action_client`), 47  
`flow_action_update()` (`globus_automate_client.flows_client.FlowsClient` method), 56

### C

`cancel()` (`globus_automate_client.action_client.ActionClient` method), 48  
`create_action_client()` (in module `globus_automate_client`), 60  
`create_flows_client()` (in module `globus_automate_client`), 61

### D

`delete_flow()` (`globus_automate_client.flows_client.FlowsClient` method), 52  
`deploy_flow()` (`globus_automate_client.flows_client.FlowsClient` method), 50

### E

`enumerate_actions()` (`globus_automate_client.flows_client.FlowsClient` method), 56  
`enumerate_runs()` (`globus_automate_client.flows_client.FlowsClient` method), 55

### F

`flow_action_cancel()` (`globus_automate_client.flows_client.FlowsClient` method), 54  
`flow_action_log()` (`globus_automate_client.flows_client.FlowsClient` method), 58  
`flow_action_release()` (`globus_automate_client.flows_client.FlowsClient` method), 54  
`flow_action_resume()` (`globus_automate_client.flows_client.FlowsClient` method), 54  
`flow_action_status()` (`globus_automate_client.flows_client.FlowsClient` method), 53

### G

`get_definition()` (`globus_automate_client.action_client.ActionClient` method), 48  
`get_flow()` (`globus_automate_client.flows_client.FlowsClient` method), 51  
`get_flow_definition_for_run()` (`globus_automate_client.flows_client.FlowsClient` method), 53  
`globus_automate_client` module, 46  
`introspect()` (`globus_automate_client.action_client.ActionClient` method), 47

### L

`list_flow_actions()` (`globus_automate_client.flows_client.FlowsClient` method), 56  
`list_flow_runs()` (`globus_automate_client.flows_client.FlowsClient` method), 56  
`list_flows()` (`globus_automate_client.flows_client.FlowsClient` method), 52  
`log()` (`globus_automate_client.action_client.ActionClient` method), 49

### M

module `globus_automate_client`, 46

### N

`new_client()` (`globus_automate_client.action_client.ActionClient` class method), 49  
`new_client()` (`globus_automate_client.flows_client.FlowsClient` class method), 59

### R

`release()` (*globus\_automate\_client.action\_client.ActionClient*  
*method*), 49

`resume()` (*globus\_automate\_client.action\_client.ActionClient*  
*method*), 48

`run()` (*globus\_automate\_client.action\_client.ActionClient*  
*method*), 47

`run_flow()` (*globus\_automate\_client.flows\_client.FlowsClient*  
*method*), 53

### S

`scope_for_flow()` (*globus\_automate\_client.flows\_client.FlowsClient*  
*method*), 52

`status()` (*globus\_automate\_client.action\_client.ActionClient*  
*method*), 48

### U

`update_flow()` (*globus\_automate\_client.flows\_client.FlowsClient*  
*method*), 51

`update_runs()` (*globus\_automate\_client.flows\_client.FlowsClient*  
*method*), 57

`use_temporary_authorizer()`  
(*globus\_automate\_client.flows\_client.FlowsClient*  
*method*), 50