

Introducing the New Globus CLI (alpha)



Stephen Rosen

April 20, 2016



What is this for? Isn't there already a CLI?



What is this for? Isn't there already a CLI?

The old CLI works well enough for today's needs, but it is limited. We're reaching those limits, and we need to move in a new direction.



There are problems inherent in the CLI being a hosted service which users SSH into.



There are problems inherent in the CLI being a hosted service which users SSH into.

- ▶ Additional infrastructure for us to run



There are problems inherent in the CLI being a hosted service which users SSH into.

- ▶ Additional infrastructure for us to run
- ▶ Closed source, can't leverage community contributions



There are problems inherent in the CLI being a hosted service which users SSH into.

- ▶ Additional infrastructure for us to run
- ▶ Closed source, can't leverage community contributions
- ▶ Mismatches between CLI and API functionality



There are problems inherent in the CLI being a hosted service which users SSH into.

- ▶ Additional infrastructure for us to run
- ▶ Closed source, can't leverage community contributions
- ▶ Mismatches between CLI and API functionality
- ▶ Difficult to transition from the CLI to the API



There are problems inherent in the CLI being a hosted service which users SSH into.

- ▶ Additional infrastructure for us to run
- ▶ Closed source, can't leverage community contributions
- ▶ Mismatches between CLI and API functionality
- ▶ Difficult to transition from the CLI to the API
- ▶ Commands are all piped over SSH, clunky shell ergonomics



The new CLI is a local command which communicates with our APIs, making Globus resources a native part of your shell environment.



The new CLI is a local command which communicates with our APIs, making Globus resources a native part of your shell environment.

- ▶ Open Source, and built on the Open Source SDK



The new CLI is a local command which communicates with our APIs, making Globus resources a native part of your shell environment.

- ▶ Open Source, and built on the Open Source SDK
- ▶ Translates down into API calls – clear transition path



The new CLI is a local command which communicates with our APIs, making Globus resources a native part of your shell environment.

- ▶ Open Source, and built on the Open Source SDK
- ▶ Translates down into API calls – clear transition path
- ▶ Improved scripting support



The new CLI is a local command which communicates with our APIs, making Globus resources a native part of your shell environment.

- ▶ Open Source, and built on the Open Source SDK
- ▶ Translates down into API calls – clear transition path
- ▶ Improved scripting support
- ▶ Better authentication model



The new CLI is a local command which communicates with our APIs, making Globus resources a native part of your shell environment.

- ▶ Open Source, and built on the Open Source SDK
- ▶ Translates down into API calls – clear transition path
- ▶ Improved scripting support
- ▶ Better authentication model
- ▶ Uniform interface to all of Globus



How do you get it?

Installation

It's a python package, so install with

```
$ pip install globus-cli
```




How do you get it?

Installation

It's a python package, so install with

```
$ pip install globus-cli
```

You now have the `globus` command at your disposal. Feel free to

```
$ globus --help
```



We are working on our long term solution for API keys and developer credentials, but the pieces aren't all in place yet. For now, you can get tokens from `tokens.globus.org`



We are working on our long term solution for API keys and developer credentials, but the pieces aren't all in place yet.

For now, you can get tokens from

`tokens.globus.org`

If you ever forget, you can always remind yourself with

```
$ globus login
```

which prints login help.



`tokens.globus.org` tokens are only good for 2 days, so don't try to build anything on this stuff (yet)!
Remember, the CLI is still in alpha.



You have the CLI installed, and you have credentials.
Make sure it's all setup right by firing a command at each of our services.



```
$ globus auth get-identities \  
  --usernames 'go@globusid.org' 'sirosen@xsede.org' | cut -d'|' -f1,2,3  
ID | Full Name | Username  
-----|-----|-----  
508e5ef6-cb9b-11e5-abe1-431ce3f42be1 | Stephen Isaac Rosen | sirosen@xsede.org  
c699d42e-d274-11e5-bf75-1fc5bf53bb24 | www.globus.org | go@globusid.org
```



```
$ globus auth get-identities \  
  --usernames 'go@globusid.org' 'sirosen@xsede.org' | cut -d'|' -f1,2,3  
ID | Full Name | Username  
-----|-----|-----  
508e5ef6-cb9b-11e5-abe1-431ce3f42be1 | Stephen Isaac Rosen | sirosen@xsede.org  
c699d42e-d274-11e5-bf75-1fc5bf53bb24 | www.globus.org | go@globusid.org
```

Hey, lookit that! It's cut friendly!



```
$ globus auth get-identities \  
  --usernames 'go@globusid.org' 'sirosen@xsede.org' | cut -d'|' -f1,2,3  
ID | Full Name | Username  
-----  
508e5ef6-cb9b-11e5-abe1-431ce3f42be1 | Stephen Isaac Rosen | sirosen@xsede.org  
c699d42e-d274-11e5-bf75-1fc5bf53bb24 | www.globus.org | go@globusid.org
```

Hey, lookit that! It's cut friendly!

We'll need the ID for go@globusid.org later – we could

```
$ globus ... | grep 'go@globusid.org' | cut  
  -d'|' -f1
```

and store it in a variable, but there is a better way.





What if you want something machine-parseable?



What if you want something machine-parseable?
All Globus CLI commands support JSON formatted output.

```
$ globus auth get-identities --format JSON --usernames 'sirosen@xsede.org'
{
  "identities": [
    {
      "username": "sirosen@xsede.org",
      "status": "used",
      "name": "Stephen Isaac Rosen",
      "id": "508e5ef6-cb9b-11e5-abe1-431ce3f42be1",
      "identity_provider": "36007761-2cf2-4e74-a068-7473afc1d054",
      "organization": "Globus",
      "email": "sirosen@uchicago.edu"
    }
  ]
}
```



How can we talk to the Transfer API with this?



How can we talk to the Transfer API with this?

Use `globus transfer`, rather than `globus auth`



How can we talk to the Transfer API with this?

Use `globus transfer`, rather than `globus auth`

Let's search for the Tutorial Endpoints (you may know them as "go#ep1" and "go#ep2")



```
$ globus transfer endpoint search \  
  --filter-owner-id 'c699d42e-d274-11e5-bf75-1fc5bf53bb24' \  
  --filter-fulltext 'Tutorial' | cut -d'|' -f2,3  
ID                                     | Display Name  
-----|-----  
cf9bcaa5-6d04-11e5-ba46-22000b92c6ec | Globus S3 Tutorial Endpoint  
ddb59aef-6d04-11e5-ba46-22000b92c6ec | Globus Tutorial Endpoint 1  
ddb59af0-6d04-11e5-ba46-22000b92c6ec | Globus Tutorial Endpoint 2
```

That UUID is the Identity ID for `go@globusid.org`



```
$ globus transfer endpoint search \  
  --filter-owner-id 'c699d42e-d274-11e5-bf75-1fc5bf53bb24' \  
  --filter-fulltext 'Tutorial' | cut -d'|' -f2,3  
ID | Display Name  
----- | -----  
cf9bcaa5-6d04-11e5-ba46-22000b92c6ec | Globus S3 Tutorial Endpoint  
ddb59aef-6d04-11e5-ba46-22000b92c6ec | Globus Tutorial Endpoint 1  
ddb59af0-6d04-11e5-ba46-22000b92c6ec | Globus Tutorial Endpoint 2
```

That UUID is the Identity ID for `go@globusid.org`
Little bit of a lie here – I trimmed whitespace in the above by piping to `sed -e 's/^\s*//g' -e 's/\s*$//g'`



It isn't pleasant for interactive use to have to lookup the ID of a username before using it. Wouldn't it be nice if the CLI translated those for you?



It isn't pleasant for interactive use to have to lookup the ID of a username before using it. Wouldn't it be nice if the CLI translated those for you?

It can!

```
$ globus transfer endpoint search \  
  --filter-owner-id 'go@globusid.org' \  
  --filter-fulltext 'Tutorial' | cut -d'|' -f2,3 | \  
  sed -e 's/~\s*//g' -e 's/\s*$//g'  
ID | Display Name  
----- | -----  
cf9bcaa5-6d04-11e5-ba46-22000b92c6ec | Globus S3 Tutorial Endpoint  
ddb59aef-6d04-11e5-ba46-22000b92c6ec | Globus Tutorial Endpoint 1  
ddb59af0-6d04-11e5-ba46-22000b92c6ec | Globus Tutorial Endpoint 2
```



It isn't pleasant for interactive use to have to lookup the ID of a username before using it. Wouldn't it be nice if the CLI translated those for you?

It can!

```
$ globus transfer endpoint search \  
  --filter-owner-id 'go@globusid.org' \  
  --filter-fulltext 'Tutorial' | cut -d'|' -f2,3 | \  
  sed -e 's/\s*//g' -e 's/\s*$//g'  
ID | Display Name  
----- | -----  
cf9bcaa5-6d04-11e5-ba46-22000b92c6ec | Globus S3 Tutorial Endpoint  
ddb59aef-6d04-11e5-ba46-22000b92c6ec | Globus Tutorial Endpoint 1  
ddb59af0-6d04-11e5-ba46-22000b92c6ec | Globus Tutorial Endpoint 2
```

Wherever the CLI takes an Identity ID, it can also take (and implicitly translate) an Identity Username.



Good errors means a good experience debugging, faster issue resolution, and happier engineers.



Good errors means a good experience debugging, faster issue resolution, and happier engineers.

Simple rules (at least for now):

- ▶ If it succeeds, exit with status 0
- ▶ If it fails, exit with status 1
- ▶ If arguments are bad, exit with status 2



Good errors means a good experience debugging, faster issue resolution, and happier engineers.

Simple rules (at least for now):

- ▶ If it succeeds, exit with status 0
- ▶ If it fails, exit with status 1
- ▶ If arguments are bad, exit with status 2

So far so good. But what about seeing what caused the error?



We want to show you the error that we got back from the API because that is the Truth. Any error translation we do client-side risks having an old CLI version + a new API version disagree about what an error means.



We want to show you the error that we got back from the API because that is the Truth. Any error translation we do client-side risks having an old CLI version + a new API version disagree about what an error means.

So, we try to show you what went wrong in a human-friendly format:

```
$ globus transfer endpoint search --filter-owner-id go@globusid.org
Globus CLI Error: A Transfer API Error Occurred.
HTTP status: 400
request_id: h0khd4VRK
code: BadRequest
message: fulltext filter is required for scope 'all'
$ echo $?
1
```




```
$ globus transfer endpoint search --filter-owner-id go@globusid.org
Globus CLI Error: A Transfer API Error Occurred.
HTTP status: 400
request_id: h0khd4VRK
code: BadRequest
message: fulltext filter is required for scope 'all'
```



```
$ globus transfer endpoint search --filter-owner-id go@globusid.org
Globus CLI Error: A Transfer API Error Occurred.
HTTP status: 400
request_id: h0khd4VRK
code: BadRequest
message: fulltext filter is required for scope 'all'
```

This is just pretty formatting of an API error. We try to be consistent in naming commandline arguments similarly to server-side parameters, so these should always make some kind of sense.



```
$ globus transfer endpoint search --filter-owner-id go@globusid.org
Globus CLI Error: A Transfer API Error Occurred.
HTTP status: 400
request_id: h0khd4VRK
code: BadRequest
message: fulltext filter is required for scope 'all'
```

This is just pretty formatting of an API error. We try to be consistent in naming commandline arguments similarly to server-side parameters, so these should always make some kind of sense.

Check out the help on the endpoint search command, and it should provide some guidance:

```
$ globus transfer endpoint search --help
```



```
$ globus list-commands
```

```
Globus CLI Commands
```

```
=====
```

```
globus list-commands
```

```
  Show a short description for all Globus CLI commands. For detailed  
  help text, run the specific command you want to know more about with  
  the `--help` option
```

```
globus login
```

```
  Get authentication credentials for the Globus CLI. Necessary before  
  any Globus CLI commands which require authentication will work
```

```
== AUTH ==
```

```
Interact with Globus Auth API. Inspect Tokens, Identities, and Identity Sets,  
consent to service terms, revoke and manage Consents, and manage OAuth Clients
```

```
globus auth get-identities
```

```
  Inspect Globus Auth Identities
```

```
globus auth token-introspect
```

```
  Inspect Globus Auth Tokens
```

```
== TRANSFER ==
```

```
Interact with Globus Transfer API. Transfer, Delete, List, and Rename files on  
Endpoints, manage your Endpoints and Shares, and monitor your ongoing Transfer  
Tasks
```





- ▶ Globus Auth View Identities



- ▶ Globus Auth View Identities
- ▶ Transfer async Tasks, ls, mkdir, and rename



- ▶ Globus Auth View Identities
- ▶ Transfer async Tasks, ls, mkdir, and rename
- ▶ Transfer Bookmark Management



- ▶ Globus Auth View Identities
- ▶ Transfer async Tasks, ls, mkdir, and rename
- ▶ Transfer Bookmark Management
- ▶ Transfer Endpoint Management + Search



- ▶ Globus Auth View Identities
- ▶ Transfer async Tasks, ls, mkdir, and rename
- ▶ Transfer Bookmark Management
- ▶ Transfer Endpoint Management + Search
- ▶ Transfer Task Management



- ▶ Globus Auth View Identities
- ▶ Transfer async Tasks, ls, mkdir, and rename
- ▶ Transfer Bookmark Management
- ▶ Transfer Endpoint Management + Search
- ▶ Transfer Task Management
- ▶ Transfer Endpoint Role + ACL Management



More Info

- ▶ The CLI is on GitHub at <https://github.com/globus/globus-cli>
- ▶ We have a small docs site at <https://globus.github.io/globus-cli> (will move to docs.globus.org)



More Info

- ▶ The CLI is on GitHub at <https://github.com/globus/globus-cli>
- ▶ We have a small docs site at <https://globus.github.io/globus-cli> (will move to docs.globus.org)

Be a Mensch! Give Us Your Feedback and Contributions!

- ▶ Open GitHub issues and Pull Requests (<https://github.com/globus/globus-cli/issues>)
- ▶ Help us determine good CLI behaviors on the Developer mailing list: developer-discuss@globus.org