
Scope Agent

Sep 11, 2019

Contents:

1	API	1
2	CLI	3
2.1	Positional Arguments	3
2.2	Named Arguments	3
3	Indices and tables	5
	Index	7


```
class scopeagent.agent.agent.Agent (api_key=None, api_endpoint=None, repository=None,
                                     commit=None, service=None, branch=None,
                                     source_root=None, debug=False, dry_run=False, com-
                                     mand=None)
```

```
__init__ (api_key=None, api_endpoint=None, repository=None, commit=None, service=None,
          branch=None, source_root=None, debug=False, dry_run=False, command=None)
```

Creates a new Scope agent instance (without installing it). All parameters are optional and will be autodetected from the environment where possible.

Parameters

- **api_key** – the API key to use when sending data to the Scope backend
- **api_endpoint** – the API endpoint of the Scope instance where the data is going to be sent
- **repository** – the git repository URL of the service being instrumented
- **commit** – the commit hash being instrumented
- **service** – the name of the service being instrumented (defaults to *default*)
- **branch** – the branch being instrumented
- **source_root** – the absolute path to the root of the git repository in the local filesystem
- **debug** – if *true*, enables verbose debug logging
- **dry_run** – if *true*, avoids actually sending data to the backend
- **command** – command string used to launch the service being instrumented

```
install (testing_mode=False, set_global_tracer=True, autoinstrument=True)
```

Installs the tracer and instruments all libraries.

Parameters

- **testing_mode** – whether the command launched is for running tests

- **set_global_tracer** – if *true*, sets the Scope tracer as the OpenTracing global tracer
- **autoinstrument** – if *true*, patches all supported libraries to enable instrumentation

Returns None

Run an application with automatic Scope instrumentation

```
usage: scope-run [-h] [--apikey APIKEY] [--api-endpoint API_ENDPOINT]
                [--name SERVICE] [--commit COMMIT] [--repository REPOSITORY]
                [--branch BRANCH] [--root ROOT] [--debug] [--version]
                ...
```

2.1 Positional Arguments

command	command to run
----------------	----------------

2.2 Named Arguments

--apikey, -k	API key to use when sending data to Scope [\$SCOPE_APIKEY]
--api-endpoint, -e	API endpoint of the Scope installation to send data to [\$SCOPE_API_ENDPOINT]
--name, -n, --service	Service name to use when sending data to Scope [\$SCOPE_SERVICE]
--commit, -c	Override autodetected commit hash for this application [\$SCOPE_COMMIT_SHA]
--repository, -r	Override autodetected repository URL for this application [\$SCOPE_REPOSITORY]
--branch, -b	Override autodetected branch for this application [\$SCOPE_BRANCH]
--root	Override autodetected repository root path for this application [\$SCOPE_SOURCE_ROOT]
--debug, -D	Log debugging information to the console [\$SCOPE_DEBUG]

--version, -v Print version information and exit

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

Symbols

`__init__()` (*scopeagent.agent.agent.Agent method*), [1](#)

A

`Agent` (*class in scopeagent.agent.agent*), [1](#)

I

`install()` (*scopeagent.agent.agent.Agent method*), [1](#)