rerobots Python client

Apr 25, 2021

Contents

This is the Python client library for the rerobots API. The corresponding source code repository is hosted at https://github.com/rerobots/py

CHAPTER 1

Introduction

1.1 Summary

Python client library for the rerobots API

Releases are available at PyPI.

Documentation of the current release is at https://rerobots-py.readthedocs.io/ or can be built from sources as described below.

1.2 Getting started

To install the current release, try

```
pip install rerobots
```

Check that the package is installed:

```
python -c 'import rerobots'
```

Most interesting interactions with rerobots require an API token, which can be provided through the environment variable `REROBOTS_API_TOKEN`.

For additional features, such as getting images from cameras as NumPy arrays,

```
pip install rerobots[extra]
```

1.3 Testing and development

All tests are in the directory `tests/`. If you have the `rerobots` package installed, then you can

```
make check
```

to run static analysis and tests that do not require a rerobots API token. Recent results on Travis CI are available at https://travis-ci.org/rerobots/py

Several other commands are available to run subsets of tests or create coverage reports. For example, to run tests that do not touch production servers:

```
make checklocal
```

and to measure code coverage: `make checklocalcover`. To view the coverage report, direct your Web browser at tests/cover/index.html

To build the User's Guide:

```
make doc
```

and direct your Web browser at doc/build/index.html

There are extra tests (not run during `make check`) that interact with production servers in a way that requires an API token and that may cause billing against the associated user account. These tests are only of interest if you plan to contribute internal changes to this Python package.

1.4 Participating

All participation must follow our code of conduct, elaborated in the file CODE_OF_CONDUCT.md in the same directory as this README.

1.4.1 Reporting errors, requesting features

Please first check for prior reports that are similar or related in the issue tracker at https://github.com/rerobots/py/issues If your observations are indeed new, please open a new issue

Reports of security flaws are given the highest priority and should be sent to <security@rerobots.net>, optionally encrypted with the public key available at https://rerobots.net/contact Please do so before opening a public issue to allow us an opportunity to find a fix.

1.4.2 Contributing changes or new code

Contributions are welcome! There is no formal declaration of code style. Just try to follow the style and structure currently in the repository.

Contributors, who are not rerobots employees, must agree to the Developer Certificate of Origin. Your agreement is indicated explicitly in commits by adding a Signed-off-by line with your real name. (This can be done automatically using `git commit --signoff`.)

1.5 License

This is free software, released under the Apache License, Version 2.0. You may obtain a copy of the License at https://www.apache.org/licenses/LICENSE-2.0

CHAPTER 2

Tutorial

Begin by getting an API token (from the Web UI). There are several ways to make it available to the client code. In this example, we assume that it is saved to a file named `jwt.txt`. Instantiate `APIClient` with this token:

```
import rerobots.api

with open('jwt.txt') as fp:
    apic = rerobots.api.APIClient(api_token=fp.read())
```

Get a list of all workspace deployments that involve "misty" (i.e., robots by Misty Robotics):

```
apic.get_wdeployments(query='misty')
```

yielding a list like

```
[{'id': '2c0873b5-1da1-46e6-9658-c40379774edf', 'type': 'fixed_misty2'},
 {'id': '3a65acd4-4aef-4ffc-b7f9-d50e48fc5541', 'type': 'basic_misty2fieldtrial'}]
```

The list you receive might be different, depending on availability of workspace deployments. To get more information about one of them, call `get_wdeployment_info()`, for example:

```
apic.get_wdeployment_info('3a65acd4-4aef-4ffc-b7f9-d50e48fc5541')
```

which will return a Python `dict` like

```
{'id': '3a65acd4-4aef-4ffc-b7f9-d50e48fc5541',
 'type': 'basic_misty2fieldtrial',
 'type_version': 1,
 'supported_addons': ['cam', 'mistyproxy', 'drive'],
 'desc': '',
 'region': 'us:cali',
 'icounter': 886,
 'created': '2019-07-28 23:26:16.983048',
 'queuelen': 0}
```

Notice that the field `supported_addons` includes `cam`. Later in this tutorial, the `cam` add-on is used to get images from cameras in the workspace.

The *Instance class* can be used to instantiate from this workspace deployment:

```
rri = rerobots.Instance(wdeployment_id='3a65acd4-4aef-4ffc-b7f9-d50e48fc5541',␣
↪apic=apic)
```

Then, methods on `rri` will affect the instance just created. For example, to get the status of the instance, call `rri.get_status()`, which usually begins with `'INIT'` (i.e., initializing). The instance is ready for action when `rri.get_status() == 'READY'`. For more information about it, call `rri.get_details()` to get a Python `dict` like

```
{'type': 'basic_misty2fieldtrial',
 'region': 'us:cali',
 'starttime': '2020-05-23 02:12:16.984534',
 'status': 'READY',
 'conn': {
   'type': 'sshtun',
   'ipv4': '147.75.70.51',
   'port': 2210,
   'hostkeys': ['ecdsa-sha2-nistp256␣
↪AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBOBfAaj/
↪HSSl7oJZ+CXnzxFsXnGQZjBh1Djdm8s7V1fdgdiyJn0JrBxzt0pSdcy50JZW+9qc1Msl34YXUjn0mwU=␣
↪root@newc247']}}
```

Notice that the connection type is `sshtun` and that the above host keys should be expected from hosts in the instance.

Recall from earlier in this tutorial that the `cam` add-on is supported by the workspace. Activate it by calling

```
rri.activate_addon_cam()
```

and waiting until `rri.status_addon_cam()` indicates that it is ready. In practice, activation is completed within several seconds. Then, use *get_snapshot_cam()* to get an image and save it in a NumPy ndarray, and display it with Matplotlib:

```
import matplotlib.pyplot as plt
import numpy as np

res = rri.get_snapshot_cam(dformat='ndarray')

plt.imshow(res['data'])
plt.show()
```

The resulting figure should open in a separate window.

Though not as powerful as dedicated `ssh` command-line programs, the *Instance class* provides methods for basic operations over SSH. To begin, start an ssh client:

```
rri.start_sshclient()
```

Then, arbitrary commands can be executed on the host in the instance via *`exec_ssh`*. For example,

```
rri.exec_ssh('pwd')
```

will return the default path from which commands are executed. Files can be uploaded and downloaded using *`put_file`*, and *`get_file`*, respectively. For example, to download the file `/etc/hosts` from the remote host:

```
rri.get_file('/etc/hosts', 'hosts')
```

Finally, to stop using the instance and delete your data from it,

```
rri.terminate()
```

CHAPTER 3

Command-line interface

MOVED TO https://rerobots-cli.readthedocs.io/en/latest/

source code at https://github.com/rerobots/cli

CHAPTER 4

API client objects

API client objects provide direct access to the rerobots API with several useful features like mapping returned data into other types.

4.1 Example

```
import rerobots.api

apic = rerobots.api.APIClient()

wdeployments = apic.get_wdeployments()
print(apic.get_wdeployment_info(wdeployments[0]['id']))
```

4.2 Create a new client object

class rerobots.api.**APIClient**(*api_token=None*, *headers=None*, *ignore_env=False*, *base_uri=None*, *verify=True*)

Instantiate API client.

api_token is some auth token obtained from https://rerobots.net/tokens In general this token has limited scope and might not be sufficient for some actions that this API client will try to do, leading to the exception WrongAuthToken.

headers is a dictionary of headers to add to every request made by this client object. This is only of interest in special use-cases.

ignore_env determines whether configuration data should be obtained from the process environment variable REROBOTS_API_TOKEN. Default (ignore_env=False) behavior is to try REROBOTS_API_TOKEN if *api_token* is not given.

base_uri is the string prefix used to create API requests. In general the default value works, but special cases might motivate changing this, e.g., to use an unofficial proxy.

verify determines whether the TLS certificates of the server are checked. Except possibly during testing, this should not be False.

4.3 Workspace deployments

`APIClient.`**`get_wdeployments`**`(`*query=None*, *maxlen=None*, *types=None*, *page=None*, *max_per_page=None*`)`

Get list of workspace deployments.

types, if given, should be a list of workspace types (str). The significance of parameters is described in the HTTP-based API documentation.

The parameters *page* and *max_per_page* can be used for pagination, which restricts the maximum number of items in the list of instances returned in any one response. Cf. documentation of the HTTP API.

`APIClient.`**`get_wdeployment_info`**`(`*wdeployment_id*`)`

Get details about a workspace deployment.

4.4 Instance creation and management

`APIClient` objects provide methods for working with instances. All operations are associated with an API token.

Note that classes presented in *Workspace instances* abstract some of the methods of `APIClient` and provide combined operations, e.g., copying a file to an instance via ssh.

`APIClient.`**`get_instances`**`(`*include_terminated=False*, *page=None*, *max_per_page=None*`)`

Get list of your instances.

The parameters *page* and *max_per_page* can be used for pagination, which restricts the maximum number of items in the list of instances returned in any one response. Cf. documentation of the HTTP API.

`APIClient.`**`get_instance_info`**`(`*instance_id*`)`

Get details about a workspace instance.

This operation requires sufficient permissions by the requesting user.

`APIClient.`**`request_instance`**`(`*type_or_wdeployment_id*, *sshkey=None*, *vpn=False*, *reserve=False*, *event_url=None*, *duration=None*`)`

Request new workspace instance.

If given, sshkey is the public key of the key pair with which the user can sign-in to the instance. Otherwise (default), a key pair is automatically generated.

If reserve=True, then create a reservation if the workspace deployment is not available at the time of this request.

`APIClient.`**`get_vpn_newclient`**`(`*instance_id*`)`

Create new OpenVPN client.

`APIClient.`**`terminate_instance`**`(`*instance_id*`)`

Terminate a workspace instance.

4.5 add-on: cam

The `cam` add-on provides access to cameras in the workspace through the rerobots API.

`APIClient.`**`get_snapshot_cam`**`(`*instance_id*, *camera_id=0*, *coding=None*, *dformat=None*`)`

Get image from camera via cam add-on.

If coding=None (default), then returned data are not encoded. The only coding supported is base64, which can be obtained with coding='base64'.

If dformat=None (default), then the image format is whatever the rerobots API provided. Currently, this can be 'jpeg' or 'ndarray' (i.e., ndarray type of NumPy).

Note that some coding and format combinations are not compatible. In particular, if dformat='ndarray', then coding must be None.

4.6 add-on: mistyproxy

The `mistyproxy` add-on provides proxies for the HTTP REST and WebSocket APIs of Misty robots.

`APIClient.`**`activate_addon_mistyproxy`**`(`*instance_id*`)`

Activate mistyproxy add-on.

Note that this add-on is unique to workspaces that involve Misty robots, e.g., https://help.rerobots.net/workspaces/fixed_misty2fieldtrial.html

When it is ready, proxy URLs can be obtained via status_addon_mistyproxy().

`APIClient.`**`status_addon_mistyproxy`**`(`*instance_id*`)`

Get status of mistyproxy add-on for this instance.

The response includes proxy URLs if any are defined.

`APIClient.`**`deactivate_addon_mistyproxy`**`(`*instance_id*`)`

Deactivate mistyproxy add-on.

Note that a cycle of deactivate-activate of the mistyproxy add-on will create new proxy URLs.

Note that calling this is not required if the workspace instance will be terminated.

CHAPTER 5

Workspace instances

Classes presented in this section have methods for working with instances. They are built on the rerobots API, but some methods do not correspond directly to rerobots API calls. In practice, `Instance` will provide everything needed for working with a single workspace instance, without need for raw calls from *API client objects*.

5.1 Example

```
import rerobots

inst = rerobots.Instance(['fixed_misty2'])
print(inst.get_status())
```

5.2 Instance class

class rerobots.**Instance**(*workspace_types=None*, *wdeployment_id=None*, *instance_id=None*, *api_token=None*, *headers=None*, *apic=None*)

client for a workspace instance

At least one of workspace_types or wdeployment_id must be given. If both are provided (not None), consistency is checked: the type of the workspace deployment of the given identifier is compared with the given type. If they differ, no instance is created, and ValueError is raised.

If instance_id is given, then attempt to attach this class to an existing instance. In this case, neither workspace_types or wdeployment_id is required. If they are provided, then consistency is checked.

The optional parameter *apic* is an instance of APIClient. If it is not given, then an APIClient object is instantiated internally from the parameters *api_token* etc., corresponding to parameters APIClient of the same name.

exec_ssh(*command*, *timeout=None*, *get_files=False*)

Execute command via SSH.

https://docs.paramiko.org/en/2.4/api/client.html#paramiko.client.SSHClient.exec_command

If get_files=True, then return files of stdin, stdout, and stderr.

get_file(*remotepath*, *localpath*)
Get file from remote host.

For the general case, the underlying Paramiko SFTP object is available from sftp_client().

put_file(*localpath*, *remotepath*)
Put local file onto remote host.

For the general case, the underlying Paramiko SFTP object is available from sftp_client().

sftp_client()
Get Paramiko SFTP client.

Note that methods put_file() and get_file() are small wrappers to put() and get() of this Paramiko class.

Read about it at https://docs.paramiko.org/en/2.4/api/sftp.html

start_sshclient()
Create SSH client to instance.

This method is a prerequisite to exec_ssh(), which executes remote terminal commands.

Index