

MAVSDK client for Python. <https://mavsdk.mavlink.io>

231 commits	3 branches	0 packages	9 releases	9 contributors	View license
Branch: master New pull request Create new file Upload files Find file Clone or download					
julianoes and JonasVautherin examples: remove comment which is no longer true Latest commit 409a253 17 days ago					
examples	examples: remove comment which is no longer true				17 days ago
mavsdk	Update to v0.23.0 server and set default args				4 months ago
other	add support for repeated type in return of requests				4 months ago
proto @ 9ed9854	Proto submodule rev updated				6 months ago
tests	Tests are obsolete atm				2 years ago
.gitignore	travis: add upload for Linux				4 months ago
.gitmodules	Added renamed proto repository				2 years ago
.travis.yml	travis-ci: only upload one file				4 months ago
LICENSE.txt	Support embedding of the grpc server binary (backend_bin)				12 months ago
MANIFEST.in	rename to mavsdk				10 months ago
MAVSDK_SERVER_VERSION	Update to v0.23.0 server and set default args				4 months ago
README.md	Update readme				4 months ago
requirements-dev.txt	Updated test framework to use Dockerized ubuntu 18.04				2 years ago
requirements-test.txt	Updated test framework to use Dockerized ubuntu 18.04				2 years ago
requirements.txt	Support embedding of the grpc server binary (backend_bin)				12 months ago
setup.cfg	Restructured python package				2 years ago
setup.py	setup.py: add ttype of long_description				4 months ago
tox.ini	Support for Python >= 3.6				2 years ago

README.md

MAVSDK-Python

build passing

This is the Python wrapper for MAVSDK.

The Python wrapper is based on a gRPC client communicating with the gRPC server written in C++. To use the Python wrapper the gRPC server called "backend" needs to be running on the same system. The wrapper is essentially auto-generated from the message definitions ([proto files](#)).

Important Notes

- Python 3.6+ is required (because the wrapper is based on [asyncio](#)).
- You may need to run `pip3` instead of `pip` and `python3` instead of `python`, depending of your system defaults.
- Auterion has a [Getting started with MAVSDK-Python] (<https://auterion.com/getting-started-with-mavsdk-python/>) guide if you're a beginner and not sure where to start.

Install using pip from PyPi

To install mavsdk-python, simply run:

```
pip3 install mavsdk
```

The package contains `mavsdk_server` already (previously called "backend"), which is started automatically when connecting (e.g. `await drone.connect()`). Have a look at the examples to see it used in practice. It will be something like:

```
from mavsdk import System
```

```
...
drone = System()
await drone.connect(system_address="udp://:14540")
```

Run the examples

Once the package has been installed, the examples can be run:

```
examples/takeoff_and_land.py
```

Build and run from sources

Note: this is more involved and targetted to contributors.

Get the Python wrapper

Clone this repo and recursively update submodules:

```
git clone https://github.com/mavlink/MAVSDK-Python --recursive
cd MAVSDK-Python
```

Install prerequisites

First install the protoc plugin (`protoc-gen-dcsdk`):

```
cd proto/pb_plugins
pip3 install -r requirements.txt
pip3 install -e .
```

You can check that the plugin was installed with `$ which protoc-gen-dcsdk`, as it should now be in the PATH.

Then go back to the root of the repo and install the dependencies of the SDK:

```
cd ../../
pip3 install -r requirements.txt -r requirements-dev.txt
```

Generate wrapper

Run the following helper script. It will generate the Python wrappers for each plugin. If the submodules are not initialized already, the script will take care of it.

```
./other/tools/run_protoc.sh
```

Install the package locally

After generating the wrapper you can install a development (editable) version of the package using:

```
pip3 install -e .
```

Build mavsdk_server

MAVSDK-Python runs the `mavsdk_server` when `await drone.connect()` is called. Under the hood, this will run `mavsdk/bin/mavsdk_server`, which has to be built separately from `MAVSDK` and copied there.

For more help on this step, check the [docs on how to build from source](#).

