
pygsm-gateway Documentation

Release 0.0.1

Cactus Consulting Group, LLC

October 24, 2016

1	Using pygsm-gateway	3
1.1	Caveats and Incompatibilities	3
1.2	Configuring and Running	3
1.3	Using with rapidsms-threadless-router	4
2	Indices and tables	5

pygsm-gateway is a [RapidSMS](#) backend or “gateway” that wraps [PyGSM](#) with a basic HTTP server, to separate it from the route process and simplify development. It works seamlessly with [rapidsms-threadless-router](#) to help make RapidSMS communicate to gateways purely by HTTP, while still allowing the use of PyGSM.

Contents:

Using pygsm-gateway

1.1 Caveats and Incompatibilities

pygsm-gateway is a new GSM backend or “gateway” for RapidSMS projects. It connects to the modem the same way the `rapidsms.backends.gsm` backend does, but instead of communicating directly with RapidSMS via Python code, it uses HTTP to send and receive messages to and from RapidSMS.

Because of this, *pygsm-gateway* cannot be used as a backend in the typical sense of the word, but must be used in conjunction with `rapidsms-threadless-router` or `rapidsms-httprouter`. In theory it could also be used with the `rapidsms.backends.http` backend, but this has not been tested and this backend may be phased out in a future release of RapidSMS.

1.2 Configuring and Running

To configure and run *pygsm-gateway*, complete the following steps:

- Customize modem configuration and message handler URL in `bin/pygsm-gateway.py`:

```
args = {
    'url': 'http://localhost:8000/backend/pygsm-gateway/',
    'url_args': {},
    'modem_args': {
        'port': '/dev/ttyACM0',
        'baudrate': '115200',
        'rtscts': '1',
        'timeout': '10',
    }
}
```

The format is similar to that for the `gsm` gateway in the old `INSTALLED_BACKENDS`, but has been reorganized slightly to improve usability. The `url` and `url_args` parameters tell *pygsm-gateway* where to deliver inbound messages from the modem. `url_args` can be left empty unless you need to pass additional POST variables, such as a username or password, to the receiving URL. The `modem_args` parameter tells *pygsm-gateway* what arguments to pass directly to the PyGSM modem.

- After customizing the configuration, create a virtual environment containing the necessary requirements and start the gateway:

```
mkvirtualenv --distribute pygsm-gateway
pip install -r requirements.txt
```

```
python setup.py install
bin/pygsm-gateway.py
```

- *pygsm-gateway* will boot the modem, spawn a thread to poll the modem, and then start up a single-threaded HTTP server to receive outbound messages from RapidSMS.

1.3 Using with *rapidsms-threadless-router*

The *simple-http* backend in *rapidsms-threadless-router* provides the foundation for building http-powered services and works seamlessly with *pygsm-gateway*.

simple-http Setup

- Add *http* app to `INSTALLED_APPS`:

```
INSTALLED_APPS = [
    # ...
    "threadless_router.backends.http",
    # ...
]
```

- Add a *simple-http* backend for *pygsm-gateway* to `INSTALLED_BACKENDS`:

```
INSTALLED_BACKENDS = {
    # ...
    "pygsm-gateway": {
        "ENGINE": "threadless_router.backends.http.outgoing",
        "outgoing_url": 'http://localhost:8080/',
    },
    # ...
}
```

- Add http urls:

```
urlpatterns = patterns('',
    # ...
    (r'^backend/', include('threadless_router.backends.http.urls')),
    # ...
)
```

- Now incoming requests for `/backend/pygsm-gateway/` will be handled by *rapidsms-threadless-router*.

Indices and tables

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