rapidsms-multimodem Documentation

Release 0.0.1

Caktus Consulting Group, LLC

May 05, 2015

Contents

Getting Started and Setup

Below are the basic steps need to get rapidsms-multimodem integrated into your RapidSMS project.

1.1 MultiModem Setup

Log into the MultiModem Web Management system and:

- Enable HTTP API Status under `SMS Services (top nav) > SMS API (sidebar nav) > HTTP API Configuration`. Make sure you click Save.
- Add a new SMS user under `SMS Services (top nav) > Users (sidebar nav)`.
- **Enable Non Polling Receive API Status under `SMS Services (top nav) > SMS API (sidebar nav) > Non I`**

    - **Server:** Server URI or hosname. For local development, this will most likely just be your IP address, e.g. `192.168.1.100`.
    - **Port:** Either `8000` for local development or `80` for production.
    - **Server Default Page:** You backend URL as defined below, e.g. `backend/multimodem/isms-lebanon/`

1.2 RapidSMS Setup

Install `rapidsms-multimodem`:

```
pip install rapidsms-multimodem
```

Add `rapidsms_multimodem` to your `INSTALLED_APPS` in your `settings.py` file:

```
INSTALLED_APPS = (
    # other apps
    'rapidsms_multimodem',
)
```

Add the following to your existing `INSTALLED_BACKENDS` configuration in your `settings.py` file:

```
INSTALLED_BACKENDS = {
    # ...
    # other backends, if any
    "isms-lebanon-1": {
```

```
        "ENGINE":  "rapidsms_multimodem.outgoing.MultiModemBackend",
        "sendsms_url": "http://<multimodem-ip-address>:81/sendmsg",
        "sendsms_user": "<username>",
        "sendsms_pass": "<password>",
        "modem_port": 1,
        "server_slug": "isms-lebanon",
    },
}
```

Single port modems only have 1 port, but it should still be specified.

The `server_slug` parameter serves 2 purposes. It uniquely identifies the iSMS server, so that RapidSMS doesn't get confused by 2 different servers having the same port number (since those are restricted to be integers from 1 to 8). It's also used to create the RapidSMS URL to which the iSMS server will send messages.

Next, you need to add an endpoint to your `urls.py` for the newly created backend. You can do this like so:

```python
from django.conf.urls import url
from rapidsms_multimodem.views import receive_multimodem_message

urlpatterns = [
    url(r"^backend/multimodem/(?P<server_slug>[\w_-]+)/$",
        receive_multimodem_message, name='multimodem-backend'),
]
```

Now inbound MultiModem messages can be received at `<your-server>/backend/multimodem/isms-lebanon/` and outbound messages will be sent via the MultiModem backend.

Additional modems on the same iSMS server will need additional entries in `INSTALLED_BACKENDS`. The only parameter that will be different than above will be the `modem_port`.

If you have more than one iSMS server, you'll create additional entries in `INSTALLED_BACKENDS`, making sure that `server_slug` is unique for each iSMS server. You will NOT need to add additional patterns to `urls.py`. The regular expression will catch the `server_slug` and match messages to the proper backend.

Indices and tables

- genindex
- modindex
- search