
rapidsms-appointments Documentation

Release 0.1.0

Cactus Consulting Group

March 21, 2016

1	Dependencies	3
2	Documentation	5
3	Translations	7
4	Running the Tests	9
5	License	11
6	Contributing	13
7	Contents	15
7.1	Getting Started	15
7.2	Project Overview	16
7.3	Release History	18

rapidsms-appointments is a reusable RapidSMS application for sending appointment reminders. Users can be subscribed to a timeline of milestones for future appointments. Reminders are sent to the patient or staff to remind them of the appointment. Appointments can be confirmed or rescheduled by patient or staff. It also tracks the history of confirmed notifications and missed/made appointments.

Dependencies

rapidsms-appointments currently runs on Python 2.6 and 2.7 and requires the following Python packages:

- Django >= 1.3
- RapidSMS >= 0.11.0
- Celery >= 3.0.13

Documentation

Documentation on using rapidsms-appointments is available on [Read The Docs](#).

Translations

The translations for rapidsms-appointment are managed on our [Transifex project](#). If you are interested in translating rapidsms-appointments into your native language you can join the project and add your language.

Running the Tests

With all of the dependencies installed, you can quickly run the tests with via:

```
python setup.py test
```

or:

```
python runtests.py
```

To test rapidsms-appointment in multiple supported environments you can make use of the [tox](#) configuration.:

```
# You must have tox installed
pip install tox
# Build default set of environments
tox
# Build a single environment
tox -e py26-1.4.X
```

License

rapidsms-appointments is released under the BSD License. See the [LICENSE](#) file for more details.

Contributing

If you think you've found a bug or are interested in contributing to this project check out [rapidsms-appointments](#) on [Github](#).

Development sponsored by [Cactus Consulting Group, LLC](#).

Contents

7.1 Getting Started

Here you will find the necessary steps to install and initial configure the rapidsms-appointments application.

7.1.1 Requirements

rapidsms-appointments requires Python 2.6 or 2.7. Python 3 is not currently supported but is planned in the future as Django and RapidSMS support for Python 3 increases. It also requires the following packages:

- Django >= 1.3
- RapidSMS >= 0.11
- Celery >= 3.0

7.1.2 Installation

Stable releases of rapidsms-appointments can be found on [PyPi](#) and [pip](#) is the recommended method for installing the package:

```
pip install rapidsms-appointments
```

Once installed you should add `appointments` to your `INSTALLED_APPS` setting:

```
INSTALLED_APPS = (  
    # Required apps  
    'rapidsms',  
    'rapidsms.contrib.handlers',  
    # Other apps go here  
    'appointments',  
)
```

To create the necessary tables for the appointment data via:

```
python manage.py syncdb
```

rapidsms-appointments is compatible with [South](#) and if you are using South to handle schemamigrations then instead you should run:

```
python manage.py migrate appointments
```

7.1.3 Configuration

As noted in the above requirements, rapidsms-appointments uses Celery to manage periodic tasks such as sending reminders for upcoming appointments. There are two tasks which you should add to your `CELERYBEAT_SCHEDULE` configuration:

```
from celery.schedules import crontab

CELERYBEAT_SCHEDULE = {
    'generate-appointments': {
        'task': 'appointments.tasks.generate_appointments',
        'schedule': crontab(hour=0, minute=0, day_of_week=1), # Every monday at midnight
    },
    'send-notifications': {
        'task': 'appointments.tasks.send_appointment_notifications',
        'schedule': crontab(hour=12, minute=0), # Every day at noon
    },
}
```

These schedules are for example purposes and can be customized for your deployments needs. For a complete reference on integrating Celery in a Django project you should see the [Celery documentation](#).

7.1.4 Next Steps

Continue reading to the [project overview](#) to learn about the terminology used and how you can define the set of appointments for your project.

7.2 Project Overview

rapidsms-appointments allows users to subscribe to a timeline of future appointments. These timelines and their related milestones are defined in the database. Appointments are scheduled relative to the user joining the timeline and they are sent a reminder just prior to the appointment.

7.2.1 Defining Appointments

Below is a description of the terminology used in this project and a high level overview of the underlying data model. There are two primary models that you need to manage in the admin for defining your set of appointments: *Timeline* and *Milestone*. The other models are noted here but are primarily managed through the application workflow.

Timeline

A timeline is the top-level structure for appointments. Each timeline has a set of keywords that the user can use to subscribe to the timeline.

Milestone

A milestone is a point on a timeline which requires an appointment. Milestones are relative to a user joining the timeline. For instance you might have a timeline to track pregnancies. The milestones would correspond to appointments throughout the pregnancy (6 weeks, 12 weeks, 18 weeks, etc).

TimelineSubscription

A TimelineSubscription attaches a User to a particular Timeline, and is the basis for generating Appointments based on the Timeline's Milestones.

Appointment

An appointment is a record of a user who is subscribed to a timeline hitting a particular milestone. Appointments might be made or missed by the user. They could also be rescheduled.

Notification

When a user has an upcoming appointment they will receive one or more notifications to serve as a reminder. A user can confirm they received the notification or not.

7.2.2 Workflow

Below describes a sample workflow for creating the timeline and milestone data and the using the SMS interactions. This example will track a set of appointments for a new born baby.

First a site admin will create a timeline with keyword 'birth|bir|postnatal|pnc'. This timeline allows for the keyword 'birth' along with the alias keywords 'bir', 'postnatal' and 'pnc'. Each of the actions can use any of these keywords to reference this timeline. For our example we'll subscribe users with the 'birth' keyword and switch to the 'pnc' shorthand for future interactions. Associated with that timeline they will create milestones at 1 week (7 days), 2 weeks (14 days), 1 month (30 days), 3 months (90 days), 6 months (180 days), and 1 year (365 days).

```
from appointments.models import Timeline, Milestone

# Creating the necessary timeline/milestone data. This could also be done in the admin

# Create the timeline
timeline = Timeline.objects.create(
    name='New Birth/Postnatal Care Visits', slug='birth|bir|postnatal|pnc'
)

# Create each milestone in the timeline
for offset in [7, 14, 30, 90, 180, 365]:
    milestone = Milestone.objects.create(
        name='{0} day appointment'.format(offset), timeline=timeline, offset=offset
    )
```

To subscribe to these notifications a user will send in an SMS with the message:

```
APPT NEW BIRTH Joe
```

where Joe is the name of the child. In place of the name they might also use a patient identifier such as ID123. This name/id will be used for future confirmations.

One week later the user will have hit the first milestone and they will be sent a reminder over SMS. Once they have received they can confirm they read it by sending back:

```
APPT CONFIRM PNC Joe
```

Again Joe might be replaced with another identifier, whichever was used for the original subscription.

Once the appointment has passed the user can mark that the appointment was made or missed:

```
APPT STATUS PNC Joe SAW
APPT STATUS PNC Joe MISSED
```

Appointments can also be rescheduled via:

```
APPT MOVE PNC Joe <Date>
```

where <DATE> denotes the new appointment date. A new reminder will be sent for the new appointment.

A user can be unsubscribed from a TimelineSubscription, and will no longer receive notifications for said subscription:

```
APPT QUIT PNC Joe <Date>
```

where <DATE> denotes the day on which the subscription should end. This field is optional and defaults to the date which the QUIT message was sent.

7.3 Release History

Release and change history for rapidsms-appointments

7.3.1 v0.1.0 (Released 2013-03-13)

- Initial public release