Newfies-Dialer Documentation

Release 2.20.0

Arezqui Belaid

December 04, 2015

Contents

Version 2.20

Release 2.20.0

Date December 04, 2015

Contents:

CHAPTER 1

Getting Started

Web http://www.newfies-dialer.org/

Download http://www.newfies-dialer.org/download/

Source https://github.com/newfies-dialer/newfies-dialer/

Keywords dialer, voip, freeswitch, django, asynchronous, rabbitmq, redis, python, distributed

–

Newfies is an open source VoIP Dialer and SMS broadcasting system based on distributed message passing. It has been built to support cloud based servers and can also work on standalone servers. It uses Freeswitch (VoIP Server) to outbound calls.

The platform is focused on real-time operations and task call distributions to clustered brokers and workers.

Newfies-Dialer is written in Python, using the Django Framework. It also operates with message brokers such as RabbitMQ, Redis but support for Beanstalk, MongoDB, CouchDB and DBMS is also available.

Newfies-Dialer provides an extensive set of APIs to easily integrate with third-party applications. Virtually every feature on the UI can be managed via the API's. There is also an API explorer to test different features.

Using very simple steps, Newfies-Dialer will help you create campaigns, add phonebooks, contacts, build audio messages and create complex telephony applications. Once the campaigns are ready to start, the messages will be dispatched and delivered.

1.1 Overview

Newfies-Dialer can be installed and used by anyone who has a need for SMS broadcasting, mass outbound calling, voice broadcasting or providing outbound IVR Some of the potential uses for Newfies-Dialer are listed below.

The system may be installed and used by either companies who wish to make calls on their own behalf, or by SaaS (Software as a Service) companies that want to provide bulk dialling and SMS broadcasting facilities to their own customers.

Newfies-Dialer can be configured on a single server, or for really high capacity or high speed systems, Newfies-Dialer can be configured across multiple servers.

1.2 Utility

Newfies-Dialer is loaded up with a list of telephone numbers that can be dialled sequentially at very high rates of calling depending on carrier capacity and hardware, potentially delivering many millions of calls per day.

When the called party answers the call, Newfies-Dialer passes the call to a telephony application that is custom designed to provide the desired behaviour.

Some examples of where Newfies-Dialer may be used follow:

- **Telecasting**: Broadcast marketing or informational messages to customers and clients.
- **Phone Polling, Surveys and Voting**: Ring large numbers of people and present IVR options for either polling their opinions, interactive surveys, or taking their vote and record the results.
- **Debt Control**: Customers can be automatically reminded at intervals that they owe money, and an IVR menu presented to talk to the finance department or passed to a credit card capture IVR to pay over the phone.
- **Appointment Reminders**: Doctors, Dentists, and other organisations that make appointments for their clients can integrate Newfies-Dialer into their appointment systems to pass a message reminding clients of an upcoming appointment.
- **Dissemination of information via phone**: Newfies-Dialer was originally designed to call large numbers of people and disseminate medical and health advice via cellphone in 3rd world countries where often, literacy levels are low. On a local scale, it can be used to disseminate information about forthcoming community events.
- **Mass Emergency broadcast**: Where there is a necessity to warn large numbers of people in a short space of time, such as weather warnings, evacuation notices and crime prevention.
- **Subscription Reminders and Renewals**: Where a company sells an annual subscription for a product or service, Newfies-Dialer can be configured to dial the customer, remind them that the subscription is due.

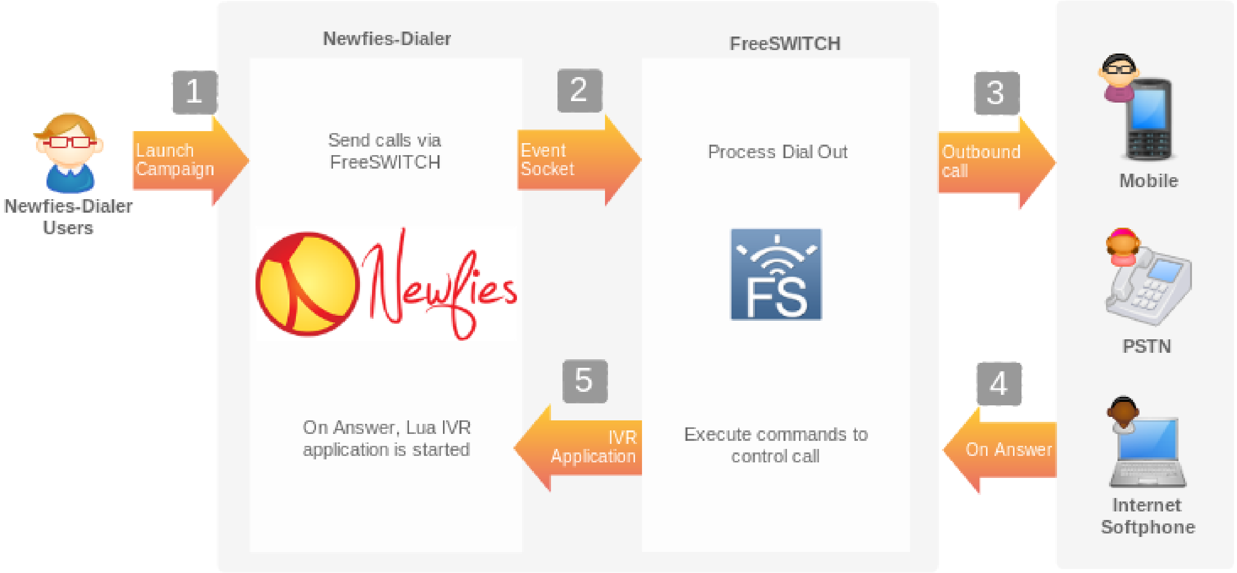
1.3 Features

Broadcasting	Web-based SMS and Voice Broadcasting application.
Distributed	Runs on one or more machines. Supports broker *clustering* and *HA* when used in combination with *RabbitMQ*. You can set up new workers without central configuration.
Concurrency	Throttle Concurrent Calls.
Scheduling	Supports recurring tasks like cron, or specifying an exact date or countdown for when the task should be executed. Can re-try failed calls at a later time.
IVR support	Accommodates multiple IVR scripts with options to connect the user to another IVR/phone number on pressing a key.
Web Interface	Newfies-Dialer can be managed via a Web interface. Realtime web-based reports for call details and current calls.
Import Contact	Import contact details from a .csv file
Multi-tenant	It provies different roles for end-users, staff and administrators. With Appointment reminders module, it also provides Calendar Users.
Real-Time Control	Control the speed of campaigns in realtime, as well as start, stop and pause buttons to control the campaigns
Surveys	IVR designer application enable the easy creation of survey application used. Survey reports can be consulted in real-time.
Audio file	Support multiple audio file formats : wav, mp3, ogg, gsm, etc...
Text2Speech	Supports powerful text2speech engines : Flite, Acapela & Cepstral. Acapela covers a wide range languages.
DNC	Support Do Not Call List. Several DNC lists can be managed per campaign and per user.
SMS	SMS delivery, SMS Gateway support, SMS campaign.

1.4 Extra Features

AMD	Answering Machine Detection module is available. For this module see the section Add-ons on website
Rebranding & Whitelabelling	Rebranding and design services to match your corporate imagee. See the Add-ons section on website

1.5 Architecture



- User selects contacts, phonebooks and campaigns, then chooses a voice application to use. The campaign is then launched
- **Newfies-Dialer** spools the outbound calls to **FreeSWITCH** via **ESL**.
- **FreeSWITCH** dials the contact via the configured telephony gateways.
- Contact picks up the call, and the answer event is received in **FreeSWITCH** and passed back to the Lua IVR Application.
- **Newfies-Dialer** is notified that the call is answered, then renders the appropriate IVR.
- The application is delivered to the contact by **FreeSWITCH**.

CHAPTER 2

Beginner's Guide

2.1 Introduction

This is a simple step-by-step guide for installation of the necessary components to install Newfies-Dialer. It covers the installation of the operating system, Freeswitch and Newfies-Dialer along with all the dependencies, followed by setting off the first Newfies-Dialer campaign, and can be achieved in under one hour, given suitably quick broadband.

2.2 Server Specification for Newfies-Dialer

As a minimum, we recommend a server with 2Gb of RAM and at least 20Gb of hard drive space with a 64 bit architecture. This will be sufficient to test Newfies-Dialer. However, for best performance a highly specified server would be better with good IO suggesting SSD, RAID 0 or RAID 10, plenty of RAM and multiple processors.

The capacity of the system is not limited by the software, but by the hardware dedicated to Newfies-Dialer, and the way that the software is tuned to take best advantage of the given hardware. We offer a tuning and optimisation service(http://www.newfies-dialer.org/add-ons/acceleration-and-optimisation-service/) which can make some dramatic improvements to performance.

Factors affecting performance include, but are not limited to:

- Duration of call
- Number of people who answer
- RAM
- Processor
- IO Speeds
- Complexity of the survey
- Answering Machine Detection
- Beep Detection
- Other processes in the OS

The capacity of your Newfies-Dialer server can only be determined accurately with benchmarking.

As a rule of thumb, if you are looking for more than 1000 concurrent calls, then we would recommend a multi-server installation consisting of one database and web server and multiple Freeswitch systems to make the calls. See http://www.newfies-dialer.org/pricing/high-capacity-multi-server-system/ for more details.

2.3 Operating System Installation

We recommend that you use Debian 7.X or 8.x 64 bit server, with only SSH installed; This document is written with this in mind. It is available at http://www.debian.org/distrib/

Please note that the Newfies-Dialer install scripts are only designed to work on Debian 7.X and 8.x (64 bit)

Burn the image to CD, and install onto your hardware. The installation of Debian is straight forward and simple. The only package to install is SSH Server so that you can manage the server from your desktop PC or Mac. All other packages will be installed by Star2Billing's Newfies-Dialer installation scripts.

When Debian is installed, it maybe worthwhile installing the latest patches and updates. This is achieved with the following commands:

```
sudo apt-get update
sudo apt-get upgrade
```

A reboot after major updates is recommended, type "reboot" at the command line.

2.4 Complete Installation

We have provided automated scripts that installs Newfies-Dialer on your Server and includes all the dependencies such as Freeswitch, Celery and Redis.

- Freeswitch is the telephony engine used by Newfies-Dialer to make calls, as well as play voice applications
- Celery is the task queuing system.
- Redis is the task messaging system

Once Debian 7.x is installed, log in as root, then download and run the install-all.sh installation script, by executing the following command:

```
wget --no-check-certificate http://bit.ly/newfies-dialer-installer -O install-all.sh
bash install-all.sh
```

The installation will prompt you to install Freeswitch, press enter. this will download and install Freeswitch with the modules appropriate for Newfies-Dialer. The installation will take some time, but does not require your interaction once started.

Finally, the Newfies-Dialer selection menu will appear.

2.5 Newfies-Dialer Installation

Newfies-Dialer is the management and control system providing a web interface for admin and customers alike, to manage the platform, subscribers and voice broadcast campaigns.

The installation script will prompt you as to what you want to install. For this guide, we chose the option to install all.

At points during the installation, you will be prompted to press enter to continue, with a short explanation of what is happening next. Just press enter for the defaults.

You will be prompted to create a superuser. Accept root, enter your email address, and your chosen password, twice. This will be the username and password you use to log on to the Newfies-Dialer web interface.

Once installed, the system will then prompt you to continue with the installation and install the backend processes. Press enter to continue. On completion, the script will return to the original Newfies-Dialer install menu. Exit the menu.

We can now reboot to ensure that, on startup, all required services are running. Type “reboot” at the command line. We don’t need to type “sudo reboot” as mentioned earlier, because we are already logged in as root.

On reboot log into the new system via your web browser to check that everything is working. Type http://<<IP-ADDRESS>>:8008 into your web browser’s address bar where <<IP-ADDRESS>> is the IP address or hostname of your Newfies-Dialer platform. If you have followed the instructions above, you should be able to login using the username and password you created during the installation.

The scripts are well tested, and there is no reason for them to fail provided you have followed the instructions and you have Internet access. If things have not gone well, and you cannot log on, re-read the instructions and check you have internet access and DNS resolution.

2.6 Freeswitch Trunk configuration

NB: See the video at the top of the page for an overview of the initial configuration of Newfies-Dialer.

In order for Newfies-Dialer to make outbound calls to its subscribers, you will need a SIP trunk. The Freeswitch wiki can provide more information on configuring trunks. However creating a trunk simply for Newfies-Dialer is straightforward.

Trunks or gateways, as they are known in Freeswitch, are configured using XML syntax, so using your favourite text editor, while logged in as root “sudo su -” create an XML file in /etc/freeswitch/sip_profiles/external/ and give it an identifiable name, e.g. call-labs.xml, and place the following lines in the file.

```
<include>
<gateway name="ip address or hostname of carrier">
<!--/// account username *required* ///-->
<param name="username" value="your username provided by carrier"/>
<!--/// auth realm: *optional* same as gateway name, if blank ///-->
<!--<param name="realm" value="asterlink.com"/>-->
```

```
<!--/// username to use in from: *optional* same as username, if blank ///-->
<param name="from-user" value="your username provided by carrier"/>
<!--/// domain to use in from: *optional* same as realm, if blank ///-->
<!--param name="from-domain" value=""/-->
<!--/// account password *required* ///-->
<param name="password" value="your password supplied by carrier"/>
<!--/// extension for inbound calls: *optional* same as username, if blank ///-->
<!--<param name="extension" value="cluecon"/>-->
<!--/// proxy host: *optional* same as realm, if blank ///-->
<!--<param name="proxy" value="asterlink.com"/>-->
<!--/// send register to this proxy: *optional* same as proxy, if blank ///-->
<!--<param name="register-proxy" value="mysbc.com"/>-->
<!--/// expire in seconds: *optional* 3600, if blank ///-->
<!--<param name="expire-seconds" value="60"/>-->
<!--/// do not register ///-->
<param name="register" value="true"/>
<!-- which transport to use for register -->
<!--<param name="register-transport" value="udp"/>-->
<!--How many seconds before a retry when a failure or timeout occurs -->
<!--<param name="retry-seconds" value="30"/>-->
<!--Use the callerid of an inbound call in the from field on outbound calls via this gateway -->
<!--<param name="caller-id-in-from" value="false"/>-->
<!--extra sip params to send in the contact-->
<!--<param name="contact-params" value="tport=tcp"/>-->
<!--send an options ping every x seconds, failure will unregister and/or mark it down-->
<!--<param name="ping" value="25"/>-->
</gateway>
</include>
```

The lines in bold are almost certainly required by your carrier and Freeswitch, the remaining parameters can be uncommented and used, if required by your carrier. The XML syntax for comments are denoted by lines that begin ” <!– ” and end in “–> ”.

Finally we need to load the new configuration, and check the trunk is registered. The Freeswitch console is accessed by typing “fs_cli” at the command prompt. The logging level can be increased or decreased from level 0 to level 7. to switch off logging, type /log 0 at the Freeswitch console, followed by enter. For the greatest verbosity, type /log 7 followed by enter.

Type “fs_cli” followed by enter.

You should now see the Freeswitch CLI, so now reload the Freeswitch configuration with the following command: (tip; Tab auto-completes)

```
sofia profile external restart reloadxml
```

When complete, check the trunk has registered with the following command.

```
sofia status
```

Against the name of the trunk you configured in the XML file, you should see REGED (registered) at the end of the line. Take a note of the trunk name, we are going to need it for telling Newfies-Dialer that it can use this trunk.

To exit the Freeswitch CLI, do CTRL D, or /exit

Freeswitch configuration is now complete.

2.7 Newfies-Dialer Configuration

For demonstration purposes, we are going to configure one standard voice application supplied with Newfies-Dialer, using the root user, to call only one person.

We are going to use the admin interface to tell Newfies-Dialer about the gateway we have configured in Freeswitch and set the maximum parameters for a customer such as their allowed dial rate, maximum number of campaigns, etc.

2.7.1 Set Hostname

Some of the features of Newfies-Dialer are dependent on the hostname or IP address, so this has to be set correctly in site address.

In the Admin dashboard, locate the "Sites" link and click change. By default, there will be an entry of "example.com". Edit this setting to reflect the hostname of the server, or if you do not have this set up, the IP address. e.g. http://www.domain.tld:8008 or in the case of an IP address, something like http://192.168.1.200:8008

2.7.2 Newfies-Dialer's Gateway

Log into the Newfies-Dialer interface using your root username and password previously created in these instructions.

Click Admin, and locate and enter the "Dialer Gateways" panel on the dashboard, then add a dialer gateway. Only the fields in bold need to be filled in. That is to say, the name of gateway, for identification, and the Gateway's field, e.g.

```
sofia/gateway/myprovider/
```

"myprovider" is the name of the gateway which you can take from the gateway name when you typed "sofia status" when configuring the Freeswitch Gateway. Finally, check the status is active, and save. The trunk should now appear in the gateway list.

2.7.3 Dialer Settings

Each customer, including the root user, needs to have Dialer Settings and trunks associated with their account. These set the limit to which each customer can utilise Newfies-Dialer in terms of trunks, calls per minute, call duration, subscribers and campaigns, as well as blacklisting or whitelisting phone numbers.

In the Admin Dashboard, locate the Dialer Settings, and click Add, in the name, put "DS1" and, for the moment, leave the default settings as they are, then save them.

The Dialer Settings have to be associated with the customer's account. For the purposes of this demonstration, we are using the root user. In production, you would apply these settings to the customers.

In the Admin dashboard, click Admins, then root. Scroll to the bottom of the page.

- Optionally add an accountcode, which can be used for billing in an external billing application to identify the customer from the CDR. The accountcode must be numerical.
- Select which Dialer Settings apply to this customer, in this case "DS1"
- Select which gateway(s) this customer is allowed to use.

Finally, click save.

NB: Everything above this is covered at the video at the top of the page, from configuring FreeSWITCH to apply Dialer Settings.

2.8 Customer Portal

Now we are going to use the customer's portal to create phonebooks containing subscribers to call, configure a survey, add campaign, and start the outbound voice broadcast. These should always be done from the Customer Portal and not from the Admin Interface. To proceed, click "Customer Panel" at the top of Admin interface.

2.8.1 Create Phonebook

The phonebook is where lists of subscribers are grouped. Click Customer Panel on the top menu in the Admin Dashboard, then click Phonebook, and add a new phonebook.

Create a new phonebook called, for the purposes of the exercise, PB1, and give it a description.

2.8.2 Add Contacts

Click contact, and add a new contact. As a minimum add a phone number to call in the Contact field, and for the purposes of this demonstration, enter your own telephone number in the format that your carrier expects it. Click Submit.

Repeat as necessary until you have your test numbers added.

2.8.3 Add Survey

Next we are going to configure the survey that will be executed when Newfies-Dialer calls your phone and you answer. Click Modules, Survey, then click add. As is traditional with first steps, we will call this survey "Hello World".

The aim is that Newfies-Dialer will call you, and when you answer, Newfies-Dialer will broadcast "Hello World" to you using the text to speech engine in Freeswitch.

Click "Add Section" and select "Play Message", enter "Hello World" in the section title. This is the speech that will be read out by the TTS engine. It can be edited later. When done, click save.

Best practice is that we should hangup, and mark the survey as complete, so click "Add Section" and select Hangup as the node type. Type "Goodbye" in the section title, and tick "Survey complete"

Next, set up the node branching. On the play message panel, click the branch icon, and select "Goodbye" as the next destination.

Finally, click Close Survey, and move on to configuring the campaign.

2.8.4 Configure Campaign

We now have all the components to create and run our campaign, so click campaign, and click add.

Give the Campaign a name, say "CPN Hello World", optionally, a description, set a caller ID to pass the called party, select the A-Leg Gateway you configured earlier, the application, Hello World, and highlight the phonebook PB1, and submit. You will be returned to the list of campaigns.

2.8.5 Start The Campaign

We are now ready to launch the campaign, simply press the Play button against the CPN Hello World campaign.

In a few seconds, your campaign will launch, your phone will ring, and you will hear "Hello World" followed by "Goodbye" broadcast to you over the phone.

2.9 Conclusion

This is a brief step by step instruction set of installing Newfies-Dialer, and making the first call. Please refer to the advanced documentation on the Newfies-Dialer and Freeswitch websites to cover more advanced topics, and begin implementing your own Voice Broadcast applications.

Star2Billing's Support Team (newfies-dialer@star2billing.com) are on hand to provide one to one installation, support and tuition where required, and Star2Billing's Development Team can be commissioned to build voice applications and integrate them with third party software and systems.

Note: As with all telephony systems, they are a valuable target on the internet, so before exposing any telephone system to the Internet, do ensure that you have done a security audit. Also note that in some countries, telemarketing, phone polling, and automated dialling is under regulatory control, and advice should be sought as to how best to remain within the limits of the law. First points of reference may be Ofcom in the UK and the Federal Trade Commission (FTC) in the USA.

CHAPTER 3

Installation

Contents:

3.1 Overview

3.1.1 Install requirements

A Requirements file gives you a way to create an environment where you can put all optional dependencies which are needed for your Project/Application.

To get started with Newfies-Dialer you must have the following installed:

- python >= 2.7 (programming language)
- HTTP server with WSGI modules (Apache, Nginx)
- Django Framework >= 1.6.1 (Python based Web framework)
- Celery >= 3.0 (Asynchronous task queue/job queue based on distributed message passing)
- MySQL-python >= 1.2.3 (MySQL for python language)
- Werkzeug >= 0.6.2 (Collection of various utilities for WSGI applications)
- amqplib >= 0.6.1 (Client library for AMQP)
- anyjson >= 0.3 (Loads the fastest JSON module)
- django-celery >= 3.1.1 (Celery integration for Django)
- django-extensions >= 1.0.3 (Collection of global custom management extensions for Django)
- django-jsonfield >= 0.9.12 (Reusable django field that can use inside models)
- django-pagination >= 1.0.7 (Utilities for creating robust pagination tools throughout a django application)
- django-picklefield >= 0.3.1 (Implementation of a pickled object field)
- django-threaded-multihost >= 1.4-0 (Provides multi-host utilities to Django projects)
- django-urlauth = 0.1.8 (Allows to build links with authentication effect)
- django-uuidfield >= 0.4 (Provides a UUIDField for your Django models)
- django-reusableapps >= 0.1.1 (Python module to enable Django to load reusable, pluggable and egg-based applications)
- docutils >= 0.10 (Text processing system for processing plaintext documentation into useful formats)

- importlib >= 1.0.2 (Implementation of the *import* statement)
- kombu >= 3.0.7 (An AMQP - Advanced Message Queuing Protocol messaging framework for Python)
- pyparsing >= 1.5.6 (A general parsing module for Python)
- python-dateutil >= 1.5 (Extensions to the standard datetime module)
- redis >= 2.8 (Redis Python Client)
- uuid >= 1.30 (UUID object and generation functions)
- wsgiref >= 0.1.2 (Validation support for WSGI)
- django-lets-go >= 2.8.5 (Common library that are reused by Star2Billing)
- django-country-dialcode >= 0.4.8 (Provide Prefix and Country information)
- djangorestframework == 2.3.9 (Creating delicious APIs for Django)
- BeautifulSoup >= 3.2.1 (HTML parser optimized for screen-scraping)
- Pygments >= 1.6 (A generic syntax highlighter)
- django-admin-tools (Collection of tools for the django administration)
- python-memcached >= 1.48 (Python based API for communicating with the memcached distributed memory object cache daemon)
- django-memcache-status >= 1.0.1 (Displays statistics about memcached instances)
- django-notification >= 0.2 (User notification management for the Django web framework)
- identicon (identicon python implementation)
- django-sentry >= 1.8.6.2 (Real-time logging / error tracing for Django)
- django-qsstats-magic >= 0.7.2 (A django microframework that eases the generation of aggregate data for querysets)
- django-frontend-notification >= 0.1.8 (Django application to display quickly on frontend the list of notification and run some basic actions such as view all notifications, delete notifications, etc...)

Use PIP to install all the requirements,:

```
$ pip install -r requirements.txt
```

3.1.2 Installation Script

You can install Newfies-Dialer manually or using the shell script provided.

To install Newfies-Dialer using this script,:

```
$ chmod +x install/install-newfies.sh

$ ./install/install-newfies.sh
```

3.1.3 Running Newfies-Dialer

Inside Newfies-Dialer directory you should run:

```
$ python manage.py syncdb

$ python manage.py collectstatic

$ python manage.py runserver
```

`syncdb` will create a database named test.db in `database` folder of the Newfies-Dialer directory. We have configured Newfies-Dialer to do this, but you can change this simply by modifying settings.py where DATABASES dictionary is constructed. You can find more information about this in the Django documentation.

`collectstatic` will fetch all necessary media files and put them into `static` folder defined in the settings module.

`runserver` runs an embedded webserver to test your site. By default it will run on http://localhost:8000. This is configurable and more information can be found on `runserver` in Django documentation.

3.1.4 Caching System

When a User requests a page, the Web server makes calculations for business logic and to create the page that your visitor sees. It creates a processing overhead higher than a standard read-a-file-off-the-filesystem server arrangement.

This is where caching comes in.

To cache something is to save the result of an expensive calculation so that you don't have to perform the calculation next time:

```
$ mkdir /usr/share/django_cache
```

3.2 Broker Installation

This document describes the installation of two different Brokers. One is `Redis` and second is `Rabbitmq`. You can install either to work with Newfies-Dialer.

3.2.1 Redis

To install Redis-Server

On Debian 7.X, you can easily install Redis as follow:

```
$ apt-get install redis-server
```

Start Redis Server

```
$ /etc/init.d/redis-server start
```

3.2.2 Rabbitmq

RabbitMQ is a complex and sophisticated product. If you don't need this level of robustness, then you might want to take a look at Redis - it installs easily, runs relatively lean, and can be monitored and maintained without a lot of fuss.

See Installing RabbitMQ over at RabbitMQ's website.

Note: If you're getting *nodedown* errors after installing and using **rabbitmqctl** then this blog post can help you identify the source of the problem:

> http://somic.org/2009/02/19/on-rabbitmqctl-and-badrpcnodedown/

Download Source

http://www.rabbitmq.com/server.html

Debian APT repository

To make use of the RabbitMQ APT repository,

1. Add the following line to your /etc/apt/sources.list:

```
    deb http://www.rabbitmq.com/debian/ testing main
```

Note: The word **testing** in the above line refers to the state of the release of RabbitMQ, not any particular Debian distribution. You can use it with Debian stable, testing or unstable, as well as with Ubuntu. In the future there will be a stable release of RabbitMQ in the repository.

2. (optional) To avoid warnings about unsigned packages, add RabbitMQ's public key to your trusted key list using apt-key(8):

```
    $ wget http://www.rabbitmq.com/rabbitmq-signing-key-public.asc
```

   $ sudo apt-key add rabbitmq-signing-key-public.asc

3. Run apt-get update.

4. Install packages as usual; for instance:

```
    $ sudo apt-get install rabbitmq-server
```

Setting up RabbitMQ

To use celery we need to create a RabbitMQ user, a virtual host and allow that user access to that virtual host:

```
$ rabbitmqctl add_user myuser mypassword

$ rabbitmqctl add_vhost myvhost

$ rabbitmqctl set_permissions -p myvhost myuser ".*" ".*" ".*"
```

See the RabbitMQ Admin Guide for more information about access control.

Starting/Stopping the RabbitMQ server

To start the server:

```
$ sudo rabbitmq-server
```

you can also run it in the background by adding the `-detached` option (note: only one dash):

```
$ sudo rabbitmq-server -detached
```

Never use **kill** to stop the RabbitMQ server, but rather use the **rabbitmqctl** command:

```
$ sudo rabbitmqctl stop
```

When the server is running, you can continue reading *Setting up RabbitMQ*.

3.3 Celery Installation

3.3.1 Celery

Celery is an asynchronous task queue/job queue based on distributed message passing. It is focused on real-time operation, but supports scheduling as well.

Celery communicates via messages using a broker to mediate between clients and workers. To initiate a task a client puts a message on the queue, the broker then delivers the message to a worker.

You can visit Celery Project webpage to find further information : http://celeryproject.org/

You can install Celery either via the Python Package Index (PyPI) or from source.

To install using pip

```
$ pip install Celery
```

Downloading and installing from source

To Download the latest version click here.

You can install it by doing the following:

```
$ tar xvfz celery-*****.tar.gz

$ cd celery-*****

$ python setup.py build

$ python setup.py install
```

CHAPTER 4

Configuration and Defaults

Contents:

4.1 Sample Configuration

This is a sample configuration to get you started. It should contain all you need to create a basic set-up.

Newfies-Dialer configuration file is by default located at /usr/share/newfies-dialer/newfies_dialer/settings_local.py

4.1.1 The Configuration Module

Some of the more important parts of the configuration module for the Newfies, `settings.py`, are explained below:

```
import os.path
APPLICATION_DIR = os.path.dirname(globals()['__file__'])
```

`APPLICATION_DIR` now contains the full path of your project folder and can be used elsewhere in the `settings.py` module so that your project may be moved around the system without you having to worry about changing any troublesome hard-coded paths.

```
DEBUG = True
```

turns on debug mode allowing the browser user to see project settings and temporary variables.

```
ADMINS = ( ('xyz', 'xyz@abc.com') )
```

sends all errors from the production server to the admin’s email address.

```
DATABASE_ENGINE = 'django.db.backends.postgresql_psycopg2'
DATABASE_NAME = 'newfies_dialer_db'
DATABASE_USER = 'user'
DATABASE_PASSWORD = 'password'
DATABASE_HOST = 'db-host'
DATABASE_PORT = '5432'
```

sets up the options required for Django to connect to your database. Only Postgresql is supported.

```
MEDIA_ROOT = os.path.join(APPLICATION_DIR, 'static')
```

tells Django where to find your media files such as images that the `HTML templates` might use.

```
ROOT_URLCONF = 'urls'
```

tells Django to start finding URL matches at in the `urls.py` module in the `newfies` project folder.

```
TEMPLATE_DIRS = ( os.path.join(APPLICATION_DIR, 'templates'), )
```

tells Django where to find your HTML template files.

```
 INSTALLED_APPS = (
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.sites',
'django.contrib.admin',
...
'dialer_gateway',
'dialer_campaign',
'dialer_cdr',
'dialer_settings',
'user_profile',
'survey',
...
)
```

tells Django which applications (custom and external) to use in your project. The custom applications, `dialer_gateway`, `dialer_campaign` etc. are stored in the project folder along with these custom applications.

4.1.2 The URLs modules

The defined URL patterns for the CPI Pilot project are divided into URL patterns specific to the project and URL patterns specific to the applications. For more information on how the pattern matching syntax work or how to write your own url patterns please consult Django's URL Dispatcher documentation.

Project specific URL patterns

The URL patterns specific to the project are applied in the `urls.py` file that is stored in the project directory `newfies`. The code segments that add these URL patterns aren't lengthy and are shown below:

```
urlpatterns = patterns('',
  # redirect
  ('^$', 'django.views.generic.simple.redirect_to', {'url': '/dialer_campaign/'}),
  (r'^admin/', include(admin.site.urls)),
  (r'^api/dialer_campaign/', include('dialer_campaign.api.urls')),
  (r'^dialer_campaign/', include('dialer_campaign.urls')),
  (r'^static/(?P<path>.*)$', 'django.views.static.serve',
      {'document_root': settings.STATIC_ROOT}),
)
```

Application specific URL patterns

The URL patterns specific to the dialer_campaign application are applied in the `/dialer_campaign/urls.py` file in the dialer_campaign application folder. The code segment that adds these URL patterns isn't lengthy either and is shown below:

```
urlpatterns = patterns('dialer_campaign.views',
   (r'^phonebook/$', 'phonebook_list'),
   (r'^phonebook/add/$', 'phonebook_add'),
   (r'^phonebook/(.+)/$', 'phonebook_change'),
)
```

4.1.3 The Views module

The functions defined in `views.py` represent the logic behind the webpages. The view functions (called through the URL matching) decide which data structures need to be constructed and sent through to the HTML templates. To do this, each view function uses Django's object relational model (ORM) to query the database picking out what is needed for any particular page.

```
@login_required
def phonebook_add(request):
    """
    Add new Phonebook
    """
    form = PhonebookForm()
    if request.method == 'POST':
        form = PhonebookForm(request.POST)
        if form.is_valid():
            obj = form.save(commit=False)
            obj.user = User.objects.get(username=request.user)
            obj.save()
            request.session["msg"] = _('"%s" is added.' %\
            request.POST['name'])
            return HttpResponseRedirect('/dialer_campaign/phonebook/')
    template = 'dialer_campaign/phonebook/change.html'
    data = {
       'form': form,
       'action': 'add',
    }
    return render_to_response(template, data,
           context_instance=RequestContext(request))
```

4.1.4 The Admin Module

The classes defined in `admin.py` tell Django what attributes are visible and modifiable from the admin site.

Example:

```
def get_urls(self):
    urls = super(VoiceAppAdmin, self).get_urls()
    my_urls = patterns('',
        (r'^add/$', self.admin_site.admin_view(self.add_view)),
    )
    return my_urls + urls

def add_view(self, request, extra_context=None):
    ctx = {
        'app_label': _('Voice'),
        'title': _('Add Voice'),
    }
    return super(VoiceAppAdmin, self)\
           .add_view(request, extra_context=ctx)
```

4.2 Celery Configuration

4.2.1 After installing Broker (Redis or Rabbitmq)

1. Redis Settings

This is a configuration example for Redis.

```
# Redis Settings
CARROT_BACKEND = "ghettoq.taproot.Redis"

BROKER_HOST = "localhost"  # Maps to redis host.
BROKER_PORT = 6379         # Maps to redis port.
BROKER_VHOST = "0"         # Maps to database number.

CELERY_RESULT_BACKEND = "redis"
REDIS_HOST = "localhost"
REDIS_PORT = 6379
REDIS_DB = 0
#REDIS_CONNECT_RETRY = True
```

2. Rabbitmq Settings

This is a configuration example for Rabbitmq.

```
BROKER_HOST = "localhost"
BROKER_PORT = 5672
BROKER_USER = "root"
BROKER_PASSWORD = "root"
BROKER_VHOST = "localhost"

CELERY_RESULT_BACKEND = "amqp"
```

4.2.2 Launch celery/celerybeat in debug mode

If you don't want to run celeryd and celerybeat as a daemon then

To run celeryd

```
$ python manage.py celeryd -E -l debug
```

To run celerybeat

```
$ python manage.py celerybeat --schedule=/var/run/celerybeat-schedule
```

To run both

```
$ python manage.py celeryd -E -B -l debug
```

4.2.3 Running celeryd/celerybeat as a daemon (Debian/Ubuntu)

To configure celeryd you will need to tell it where to change directory to, when it starts in order to find your celeryconfig.

```
$ cd install/celery-init/etc/default/
```

1. Open celeryd in text editor & change the following variables

   Configuration file: /etc/default/celeryd

   Init script: celeryd.

   Usage : /etc/init.d/celeryd {start|stop|force-reload|restart|try-restart|status}:

```
    # Where to chdir at start
    CELERYD_CHDIR="/path/to/newfies/"

    # Path to celeryd
    CELERYD="/path/to/newfies/manage.py celeryd"

    # Extra arguments to celeryd
    CELERYD_OPTS="--time-limit=300"

    # Name of the celery config module.
    CELERY_CONFIG_MODULE="celeryconfig"

    # Extra Available options
    # %n will be replaced with the nodename.
    # Full path to the PID file. Default is /var/run/celeryd.pid.
    CELERYD_PID_FILE="/var/run/celery/%n.pid"

    # Full path to the celeryd log file. Default is /var/log/celeryd.log
    CELERYD_LOG_FILE="/var/log/celery/%n.log"

    # User/Group to run celeryd as. Default is current user.
    # Workers should run as an unprivileged user.
    CELERYD_USER="celery"
    CELERYD_GROUP="celery"
```

2. Open celeryd (for periodic task) in text editor & add the following variables

   Configuration file: /etc/default/celerybeat or /etc/default/celeryd

   Init script: celerybeat

   Usage: /etc/init.d/celerybeat {start|stop|force-reload|restart|try-restart|status}:

```
    # Path to celerybeat
    CELERYBEAT="/path/to/newfies/manage.py celerybeat"

    # Extra arguments to celerybeat
    CELERYBEAT_OPTS="--schedule=/var/run/celerybeat-schedule"
```

3. Copy the configuration file & init scripts to /etc dir:

```
    $ cp etc/default/celeryd /etc/default/

    $ cp etc/init.d/celeryd /etc/init.d/

    $ cp etc/init.d/celerybeat /etc/init.d/
```

4. Run/Start or Stop celery as a daemon:

```
    $ /etc/init.d/celeryd start or stop

    $ /etc/init.d/celerybeat start or stop
```

4.2.4 Troubleshooting

If you can’t get the celeryd as a daemon to work, you should try running them in verbose mode:

```
$ sh -x /etc/init.d/celeryd start

$ sh -x /etc/init.d/celerybeat start
```

CHAPTER 5

FreeSwitch

5.1 Installation and configuration

Freeswitch is the telephony engine used by Newfies-Dialer to make calls, as well as broadcast voice applications

Newfies-Dialer communicates with Freeswitch though the Event-Socket. Communication is made via the Event Socket Library ESL. We have some Freeswitch dependencies, therefore the following modules will need to be installed

```
mod_curl, asr_tts, mod_flite, asr_tts, mod_shout, mod_dingaling, mod_shell_stream, mod_xml_cdr
```

A script for Freeswitch Installation which will install Freeswitch with the required modules and configure it for you is available.

Download and run the Freeswitch installation script.

Once logged in as root, execute the following command:

```
wget https://raw.github.com/newfies-dialer/newfies-dialer/master/install/install-freeswitch.sh
```

The above commmand download the installation script. We can then execute the script with the following command:

```
bash install-freeswitch.sh
```

This will download and install Freeswitch with the modules appropriate for Newfies-Dialer. The installation will take some time, but does not require your interaction once started.

5.2 Trunk configuration

In order for Newfies-Dialer to make outbound calls to its subscribers, you will need a SIP trunk. The Freeswitch wiki can provide more information on configuring trunks. However creating a trunk simply for Newfies-Dialer is straightforward.

Trunks or gateways, as they are known in Freeswitch, are configured using XML syntax, so using your favourite text editor, while logged in as root “sudo su -” create an XML file in /usr/local/freeswitch/conf/sip_profiles/external/ and give it an identifiable name, e.g. call-labs.xml, and place the following lines in the file:

```
<include>
<gateway name="ip address or hostname of carrier">
<!--/// account username *required* ///-->
<param name="username" value="your username provided by carrier"/>
<!--/// auth realm: *optional* same as gateway name, if blank ///-->
<!--<param name="realm" value="asterlink.com"/>-->
```

```
<!--/// username to use in from: *optional* same as username, if blank ///-->
<param name="from-user" value="your username provided by carrier"/>
<!--/// domain to use in from: *optional* same as realm, if blank ///-->
<!--param name="from-domain" value=""/-->
<!--/// account password *required* ///-->
<param name="password" value="your password supplied by carrier"/>
<!--/// extension for inbound calls: *optional* same as username, if blank ///-->
<!--<param name="extension" value="cluecon"/>-->
<!--/// proxy host: *optional* same as realm, if blank ///-->
<!--<param name="proxy" value="asterlink.com"/>-->
<!--/// send register to this proxy: *optional* same as proxy, if blank ///-->
<!--<param name="register-proxy" value="mysbc.com"/>-->
<!--/// expire in seconds: *optional* 3600, if blank ///-->
<!--<param name="expire-seconds" value="60"/>-->
<!--/// do not register ///-->
<param name="register" value="true"/>
<!-- which transport to use for register -->
<!--<param name="register-transport" value="udp"/>-->
<!--How many seconds before a retry when a failure or timeout occurs -->
<!--<param name="retry-seconds" value="30"/>-->
<!--Use the callerid of an inbound call in the from field on outbound calls via this gateway -->
<!--<param name="caller-id-in-from" value="false"/>-->
<!--extra sip params to send in the contact-->
<!--<param name="contact-params" value="tport=tcp"/>-->
<!--send an options ping every x seconds, failure will unregister and/or mark it down-->
<!--<param name="ping" value="25"/>-->
</gateway>
</include>
```

The uncommented lines are almost certainly required by your carrier and Freeswitch, the remaining parameters can be uncommented and used, if required by your carrier. The XML syntax for comments are denoted by lines that begin “ <!– “ and end in “–> “.

Finally we need to load the new configuration, and check the trunk is registered. Enter the Freeswitch CLI (Command Line Interface) from the console:

```
/usr/local/freeswitch/bin/fs_cli
```

You should now see the Freeswitch CLI, so now reload the Freeswitch configuration with the following command: (tip; Tab auto-completes):

```
sofia profile external restart reloadxml
```

When complete, check the trunk has registered with the command:

```
sofia status
```

Against the name of the trunk you configured in the XML file, you should see REGED (registered) at the end of the line. Take a note of the trunk name, we are going to need it for telling Newfies-Dialer that it can use this trunk.

To exit the Freeswitch CLI, do CTRL D, or /exit

Freeswitch configuration is now complete.

SMS Configuration

6.1 Configuration

SMS braodcasting is supported as standard by Newfies-Dialer, allowing Newfies-Dialer to send out thousands or millions of SMS messages to contacts.

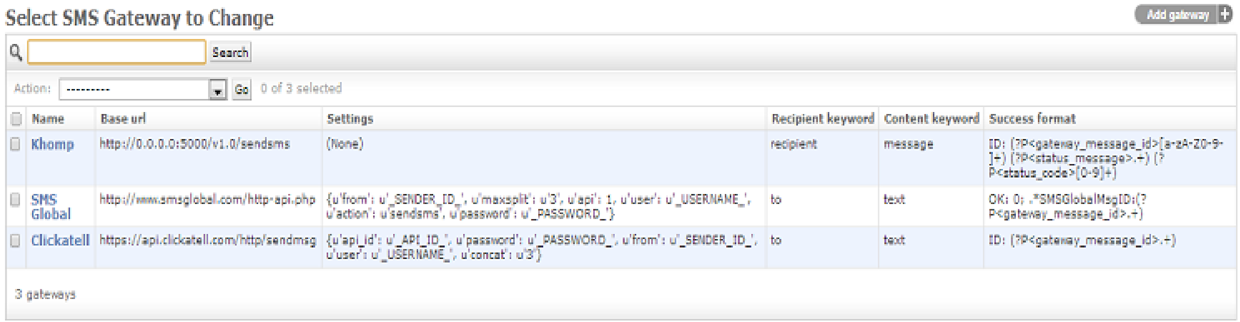
The SMS gateway supports the following gateways, Clickatell (http://www.clickatell.com/) and SMSGlobal (http://www.smsglobal.com). Additionally, Khomp hardware is supported to allow the use of SIM cards in house. See http://www.khomp.com.br/ for more details.

Go to the Admin interface and identify the SMS Gateway section, and click on Gateways. Listed by default are three gateways, Clickatell, SMSGlobal and Khomp. Those that you don't want to be configured can be deleted.

Click the gateway to be configured and edit the pre-filled settings and populate the fields with the credentials supplied by your SMS provider:

URL:

- http://localhost:8000/admin/sms/gateway/
- http://localhost:8000/admin/sms/gateway/add/

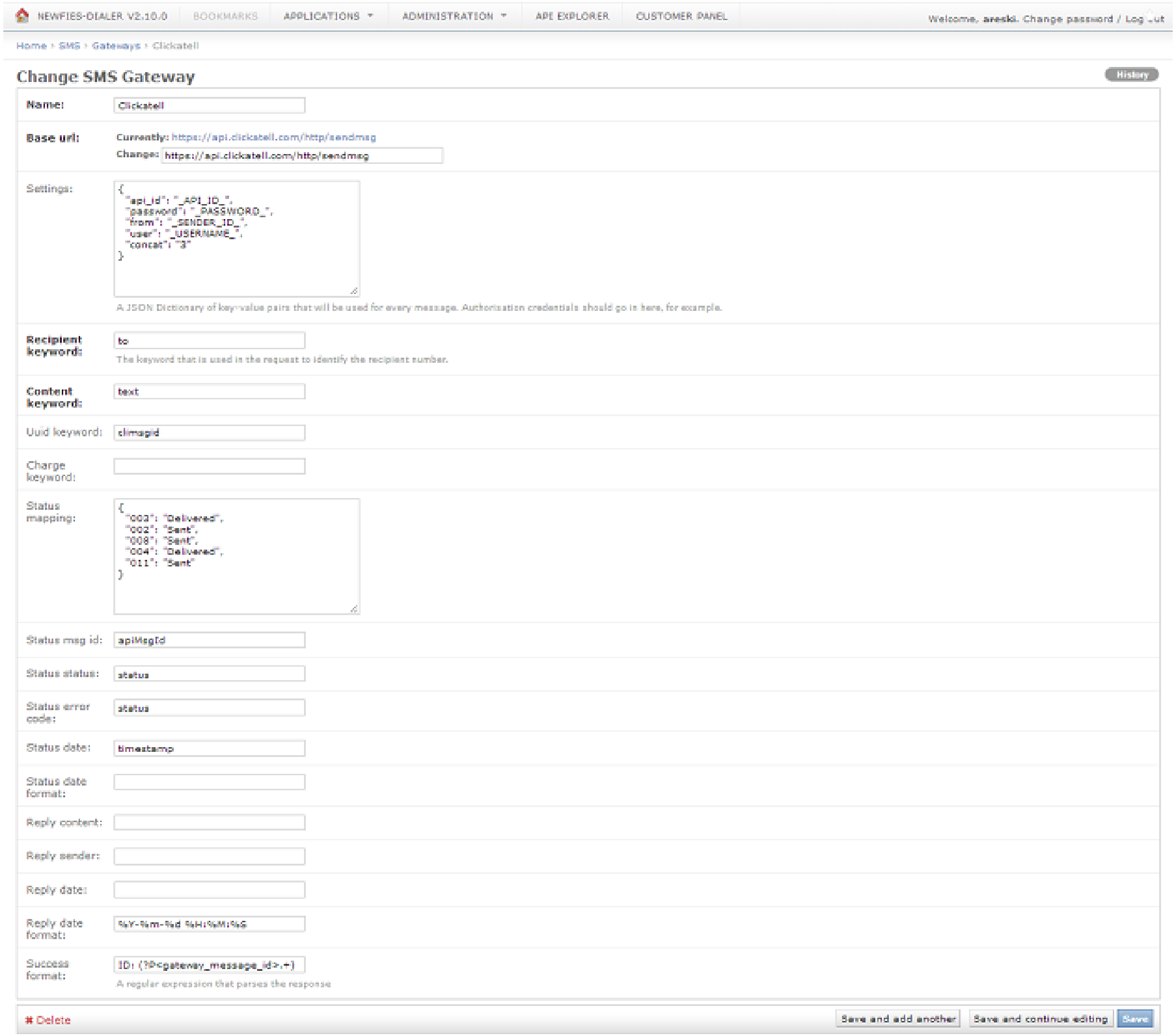


Select SMS Gateway to Change

0 of 3 selected

□	Name	Base url	Settings	Recipient keyword	Content keyword	Success format
□	Khomp	http://0.0.0.0:5000/v1.0/sendsms	(None)	recipient	message	ID: (?P<gateway_message_id>[a-zA-Z0-9-]+) (?P<status_message>.+) (?P<status_code>[0-9]+)
□	SMS Global	http://www.smsglobal.com/http-api.php	{u'from': u'_SENDER_ID_', u'maxsplit': u'3', u'api': 1, u'user': u'_USERNAME_', u'action': u'sendsms', u'password': u'_PASSWORD_'}	to	text	OK: 0; .*SMSGlobalMsgID:(?P<gateway_message_id>.+)
□	Clickatell	https://api.clickatell.com/http/sendmsg	{u'api_id': u'_API_ID_', u'password': u'_PASSWORD_', u'from': u'_SENDER_ID_', u'user': u'_USERNAME_', u'concat': u'3'}	to	text	ID: (?P<gateway_message_id>.+)

3 gateways



Change SMS Gateway

Name:	Clickatell
Base url:	Currently: https://api.clickatell.com/http/sendmsg Change: https://api.clickatell.com/http/sendmsg

Settings:	{
	"api_id": "_API_ID_",
	"password": "_PASSWORD_",
	"from": "_SENDER_ID_",
	"user": "_USERNAME_",
	"concat": "3"
	}
	A JSON Dictionary of key-value pairs that will be used for every message. Authorisation credentials should go in here, for example.

Recipient keyword:	to The keyword that is used in the request to identify the recipient number.
Content keyword:	text
Uuid keyword:	climsgid
Charge keyword:	

Status mapping:	{
	"003": "Delivered",
	"002": "Sent",
	"008": "Sent",
	"004": "Delivered",
	"011": "Sent"
	}

Status msg id:	apiMsgId
Status status:	status
Status error code:	status
Status date:	timestamp
Status date format:	
Reply content:	
Reply sender:	
Reply date:	
Reply date format:	%Y-%m-%d %H:%M:%S
Success format:	ID: (?P<gateway_message_id>.+) A regular expression that parses the response

6.1.1 Clickatell

Clickatell settings:

```
{
  "api_id": "_API_ID_",
  "password": "_PASSWORD_",
  "from": "_SENDER_ID_",
  "user": "_USERNAME_",
  "concat": "3"
}
```

Edit _API_ID_, _Sender_ID_, _USERNAME_ and _PASSWORD_ to the credentials provided by Clickatell and leave everything else as it is, and click save.

6.1.2 SMSGlobal

The default settings are as follows:

```
{
  "from": "_SENDER_ID_",
  "maxsplit": "3",
  "api": 1,
  "user": "_USERNAME_",
  "action": "sendsms",
  "password": "_PASSWORD_"
}
```

Edit _Sender_ID_, _USERNAME_ and _PASSWORD_ to match the credetials issued by SMSGlobal, leave everything else unchanged and save.

6.1.3 Vitelity

Vitelity: http://www.vitelity.com/services_sms/

You will need to turn on API access for whatever IP your Newfies-Dialer installation is hosted on, create a shortcode keyword, and set a default response to that keyword.

Here it's how you should configure the new SMS Gateway to support Vitelity:

Name:

```
SMSvitelity
```

Base URL:

```
http://smsout-api.vitelity.net/api.php
```

Set the *Settings* as follow:

```
{
  "xml": "yes",
  "src": "99629",
  "login": "YOUR_VITELITY_API_LOGIN",
  "cmd": "sendshort",
  "pass": "YOUR_VITELITY_API_PASSWORD"
}
```

Recipient keyword:

```
dst
```

Content keyword:

```
msg
```

Status mapping:

```
{
  "fail": "Error",
  "ok": "Sent"
}
```

Success format:

```
.*\n*.*\n*.*<status>(?P<status_message>.+)</status>\n*<response>(?P<status_code>.+)</response>(?P<gat
```

6.1.4 Khomp

Khomp hardware is supported to all the use of SIM cards in house. See http://www.khomp.com.br/ for more details. The hardware needs to be configured for Freeswitch, see http://wiki.freeswitch.org/wiki/Khomp for more details.

Additionally, the SMS Khomp API needs to be installed to link Newfies-Dialer and Khomp hardware which is available at https://github.com/areski/sms-khomp-api.

Contact us (http://www.newfies-dialer.org/about-us/contact/) for support and assistance.

6.2 Messages

The messages section in the admin screens shows the status of the messages, e.g. success or failure as well as the status message.

6.3 Providers

This area is not operational at the moment and is as a place-holder for future development.

6.4 Replies

This area is not operational at the moment and is as a place-holder for future development

CHAPTER 7

User Guide

Contents:

7.1 Overview

Newfies-Dialer is an open source VoIP Dialer based on distributed message passing. It has been built to support cloud servers and also works on standalone servers. It uses Freeswitch (VoIP Server) to outbound calls, but support for other VoIP Servers such as Asterisk could be easily added in the future. The platform is focused on real-time operations and task call distributions to clustered brokers and workers.

Newfies-Dialer is a computerised system that automatically dials a group of telephone numbers for connection to assigned campaigns.

Features

- Restful-API based to easily integrate the platform dialer with third-party applications
- Web-based administrative/customer interfaces
- Lower operating costs
- Calls are made through Internet VoIP. No need for telephony hardware
- Call reports and Statistics

7.2 How to use Newfies-Dialer

7.2.1 Freeswitch Set-Up

Configure trunks and gateways in Freeswitch by creating an XML file in /usr/local/freeswitch/conf/sip_profiles/external/ and give it an identifiable name, e.g. call-labs.xml, and place the following lines in the file, edited to suit your provider:

```
<include>
<gateway name="ip address or hostname of carrier">
<!--/// account username *required* ///-->
<param name="username" value="your username provided by carrier"/>
<!--/// auth realm: *optional* same as gateway name, if blank ///-->
<!--<param name="realm" value="asterlink.com"/>-->
<!--/// username to use in from: *optional* same as username, if blank ///-->
<param name="from-user" value="your username provided by carrier"/>
<!--/// domain to use in from: *optional* same as realm, if blank ///-->
<!--param name="from-domain" value=""/-->
<!--/// account password *required* ///-->
<param name="password" value="your password supplied by carrier"/>
<!--/// extension for inbound calls: *optional* same as username, if blank ///-->
<!--<param name="extension" value="cluecon"/>-->
<!--/// proxy host: *optional* same as realm, if blank ///-->
<!--<param name="proxy" value="asterlink.com"/>-->
<!--/// send register to this proxy: *optional* same as proxy, if blank ///-->
<!--<param name="register-proxy" value="mysbc.com"/>-->
<!--/// expire in seconds: *optional* 3600, if blank ///-->
<!--<param name="expire-seconds" value="60"/>-->
<!--/// do not register ///-->
<param name="register" value="true"/>
<!-- which transport to use for register -->
<!--<param name="register-transport" value="udp"/>-->
<!--How many seconds before a retry when a failure or timeout occurs -->
<!--<param name="retry-seconds" value="30"/>-->
<!--Use the callerid of an inbound call in the from field on outbound calls via this gateway -->
<!--<param name="caller-id-in-from" value="false"/>-->
<!--extra sip params to send in the contact-->
<!--<param name="contact-params" value="tport=tcp"/>-->
<!--send an options ping every x seconds, failure will unregister and/or mark it down-->
<!--<param name="ping" value="25"/>-->
</gateway>
</include>
```

Then in the Freeswitch CLI (fs_cli) "sofia profile external restart reloadxml". When the command is complete, check the gateway has registered with the command "sofia status".

7.2.2 Create Gateway

Having created the gateway in Freeswitch, Newfies-Dialer has to be told that it can use it. In admin, add a new dialer gateway, e.g. sofia/gateway/myprovider/ (The final / is important) where "myprovider" is the gateway name setting used in above xml file in Freeswitch.

Only the fields in bold are compulsory.

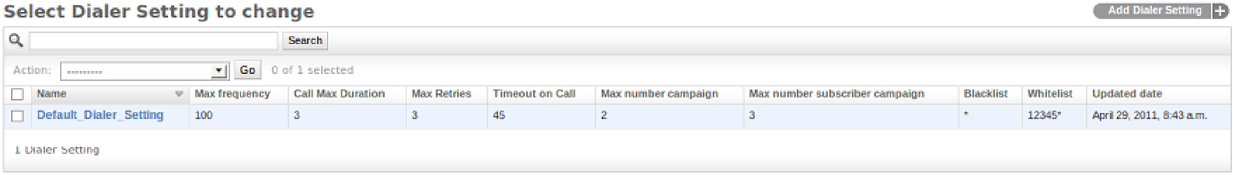
7.2.3 Dialer Settings

Dialer settings has to be mapped with system users who are going to create campaigns & contacts. If dialer setting is not mapped with system users, notification & mail has been sent to Super admin user.

To create restrictions (like the Max. no of campaign, Max no of contacts etc.) for system User, Click on `Add` in dialer settings list of admin side. Add numeric values for the limit.

To apply the dialer settings limit on a system user, click on `Customers` or `Admins` in admin UI, select the user to update, & apply the settings from the dialer settings list.

Dialer settings:



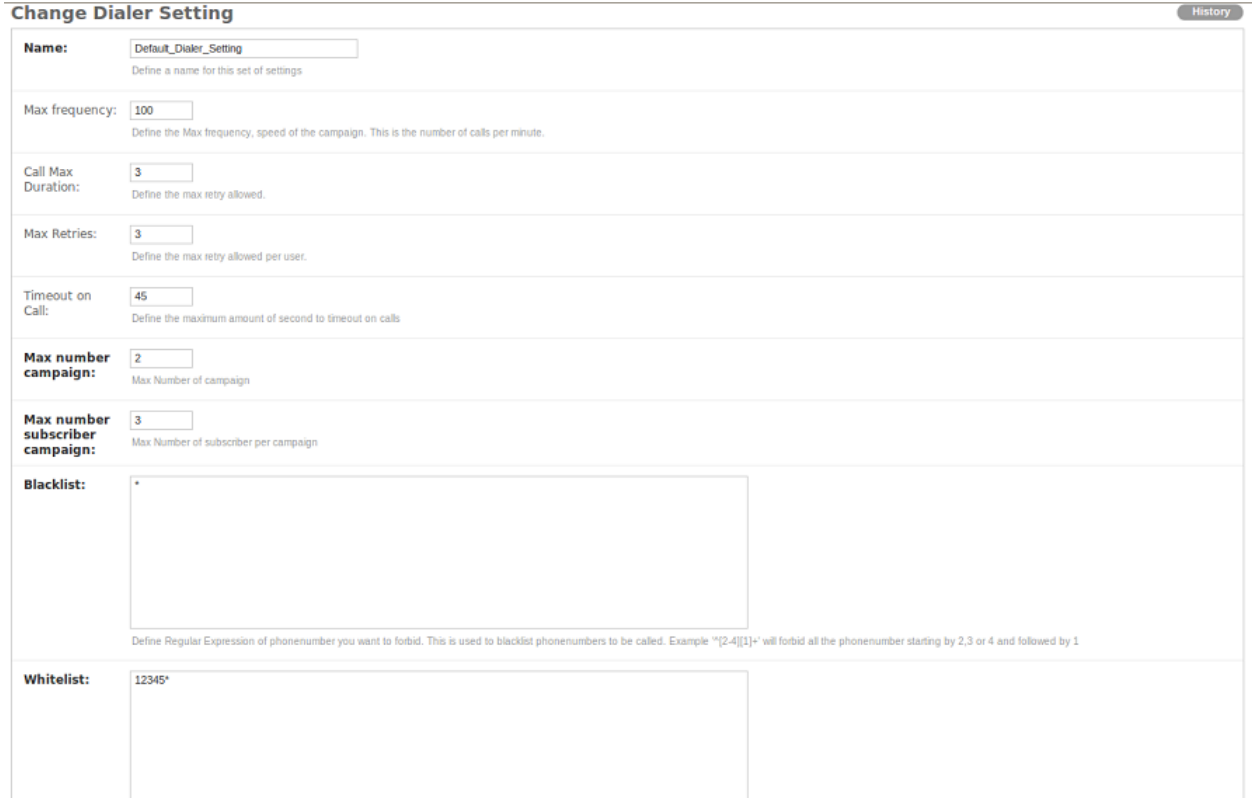
Select Dialer Setting to change

0 of 1 selected

□	Name	Max frequency	Call Max Duration	Max Retries	Timeout on Call	Max number campaign	Max number subscriber campaign	Blacklist	Whitelist	Updated date
□	Default_Dialer_Setting	100	3	3	45	2	3	*	12345*	April 29, 2011, 8:43 a.m.

1 Dialer Setting

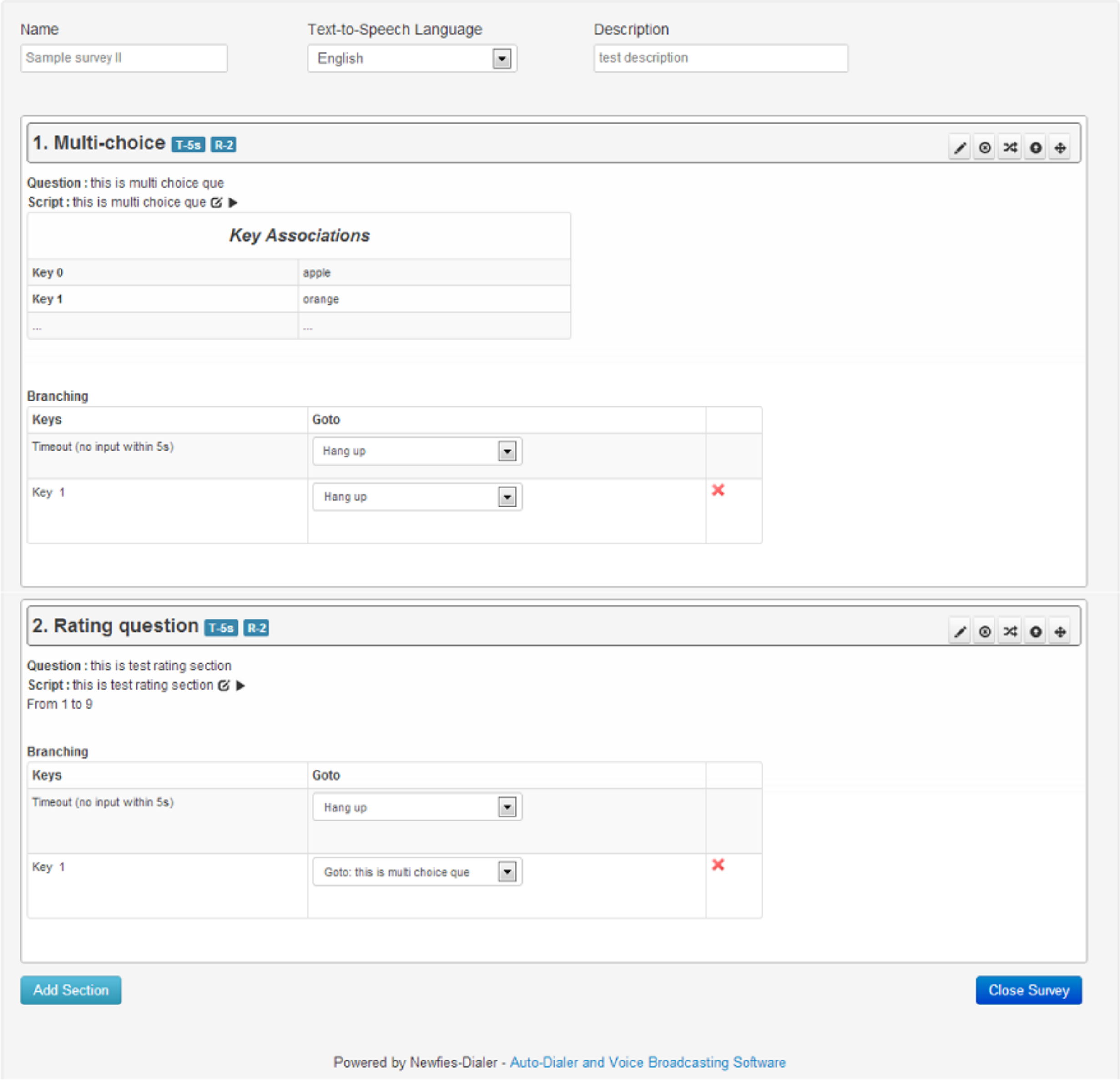
Update dialer settings:



Change Dialer Setting

Name:	Default_Dialer_Setting Define a name for this set of settings
Max frequency:	100 Define the Max frequency, speed of the campaign. This is the number of calls per minute.
Call Max Duration:	3 Define the max retry allowed.
Max Retries:	3 Define the max retry allowed per user.
Timeout on Call:	45 Define the maximum amount of second to timeout on calls
Max number campaign:	2 Max Number of campaign
Max number subscriber campaign:	3 Max Number of subscriber per campaign
Blacklist:	* Define Regular Expression of phonenumber you want to forbid. This is used to blacklist phonenumbers to be called. Example '^[2-4][1]+' will forbid all the phonenumber starting by 2,3 or 4 and followed by 1
Whitelist:	12345*

7.2.4 Create Survey



Name	Text-to-Speech Language	Description
Sample survey II	English	test description

1. Multi-choice T-5s R-2

Question : this is multi choice que
Script : this is multi choice que

Key Associations	
Key 0	apple
Key 1	orange
...	...

Branching

Keys	Goto	
Timeout (no input within 5s)	Hang up	
Key 1	Hang up	✗

2. Rating question T-5s R-2

Question : this is test rating section
Script : this is test rating section
From 1 to 9

Branching

Keys	Goto	
Timeout (no input within 5s)	Hang up	
Key 1	Goto: this is multi choice que	✗

Add Section
Close Survey
Powered by Newfies-Dialer - Auto-Dialer and Voice Broadcasting Software

The survey application for Newfies-Dialer allows polls and surveys to be taken over the phone.

Each contact is called, and then played a sound file. After the sound file is heard, the user can enter their answer through the phone keypad using keys 0 to 9.

Therefore before creating the survey, the first job is to upload the audio for the survey. One audio file is required for each question.

Uploading Audio Files:

Select Audio Files from the top menu then click add.

Enter a name to describe the audio, then click chose file, select the file you require from your computer, then click submit.

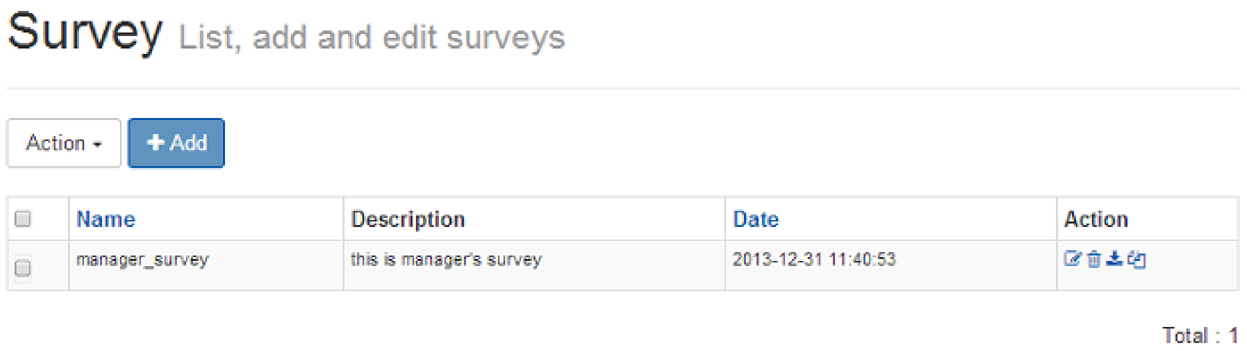
Note that only mp3, Wav and ogg formats are supported.

The audio file will then be uploaded and renamed so that it is unique.

Create the Survey:

URL:

- http://localhost:8000/survey/
- http://localhost:8000/survey/add/
- http://localhost:8000/survey/1/



Survey List, add and edit surveys

□	Name	Description	Date	Action
□	manager_survey	this is manager's survey	2013-12-31 11:40:53	

Total : 1

Select Modules from the top menu, then Survey.

Click the add button, then give the survey a name and description, then click Submit.

A button will now appear to add a question.

In the question field, put in some text to describe the question - e.g "What is 1+1"; select the audio file pertaining to the question which was uploaded in the previous step.

If no audio file is selected, then the system will automatically play the text in the question field using the text to speech engine.

Then click Add Response. A further two fields will appear named Key Digit and Key Value. In key Digit put a number from 0 to 9 which should be pressed for this answer. In the example "What is 1+1", "2" should be placed in the Key Digit Field"

The Key Value field is used in the survey reports, and so in this case, you would put "Correct" as 1+1=2. You may chose to add responses 0,1 and 3 to 9 as key digits, with key values of "Wrong" as these answers will then be summed up in the Survey Reports.

You can then go on to add another question, and its associated responses. On completion, click Update Survey.

To use the Survey in a campaign, simply create a campaign as normal, and select the Survey name in the Application drop-down.

7.2.5 Survey Nodes

Most survey nodes have similar attributes that include:

- Section Title, the title of the section and becomes the TTS (text To Speech)
- Audio File, The sound file to be played for this node.
- Retries, if no valid input, then the question can be asked again.

- Timeout, the amount of time before it is considered that no input has been received.
- Check Validity, whether the answer is deemed valid.
- Audio Invalid Input, The audio to play when invalid input is received.
- Survey Complete, check this field to mark the survey is complete for reporting & retry purposes.

There is many type of survey nodes which allow to create complex IVR applications. On each node you will be able to perform call transfer, capture digits, go to conference room, add a contact to the DNC list and much more, see the list of survey nodes *Survey Nodes*.

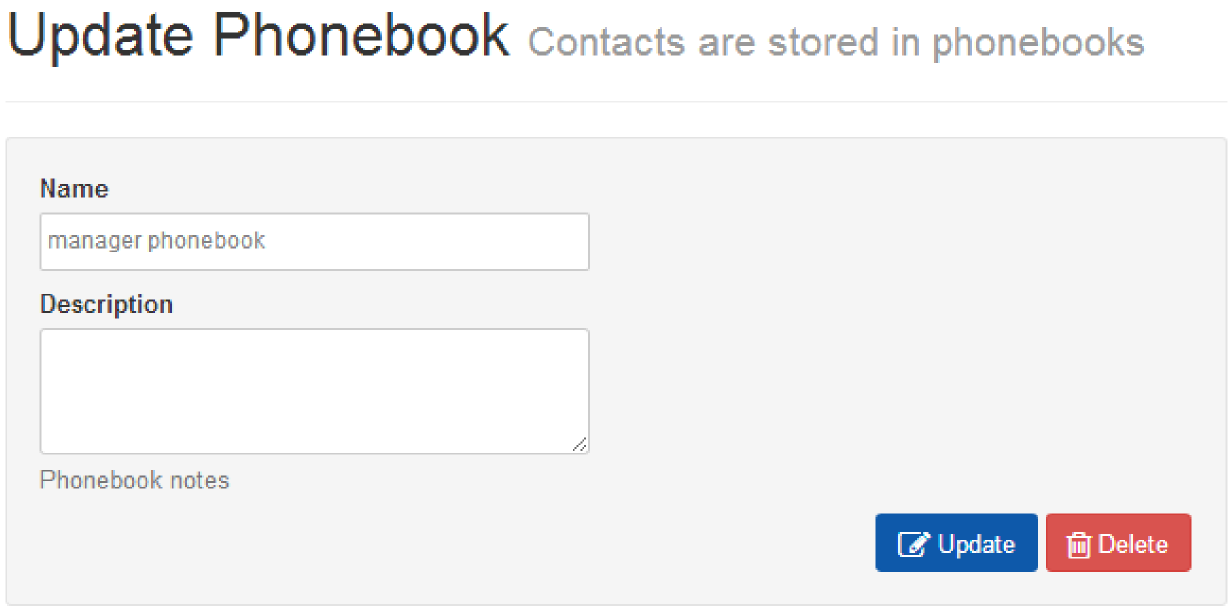
7.2.6 Survey Branching

The flow of the survey is determined by Survey Branching, which will pass the contact onto the next node depending on the parameters you set for instance, invalid entry, any key or a specified key.

7.2.7 Create call list

To create a call list, click on `Add` in Phonebook list, add name of phonebook & its description. Click on `Contacts` and add phone numbers in the contact list. You can also import your call list from csv files, via clicking on `Import contact`.

Update Phonebook:



Update Phonebook Contacts are stored in phonebooks

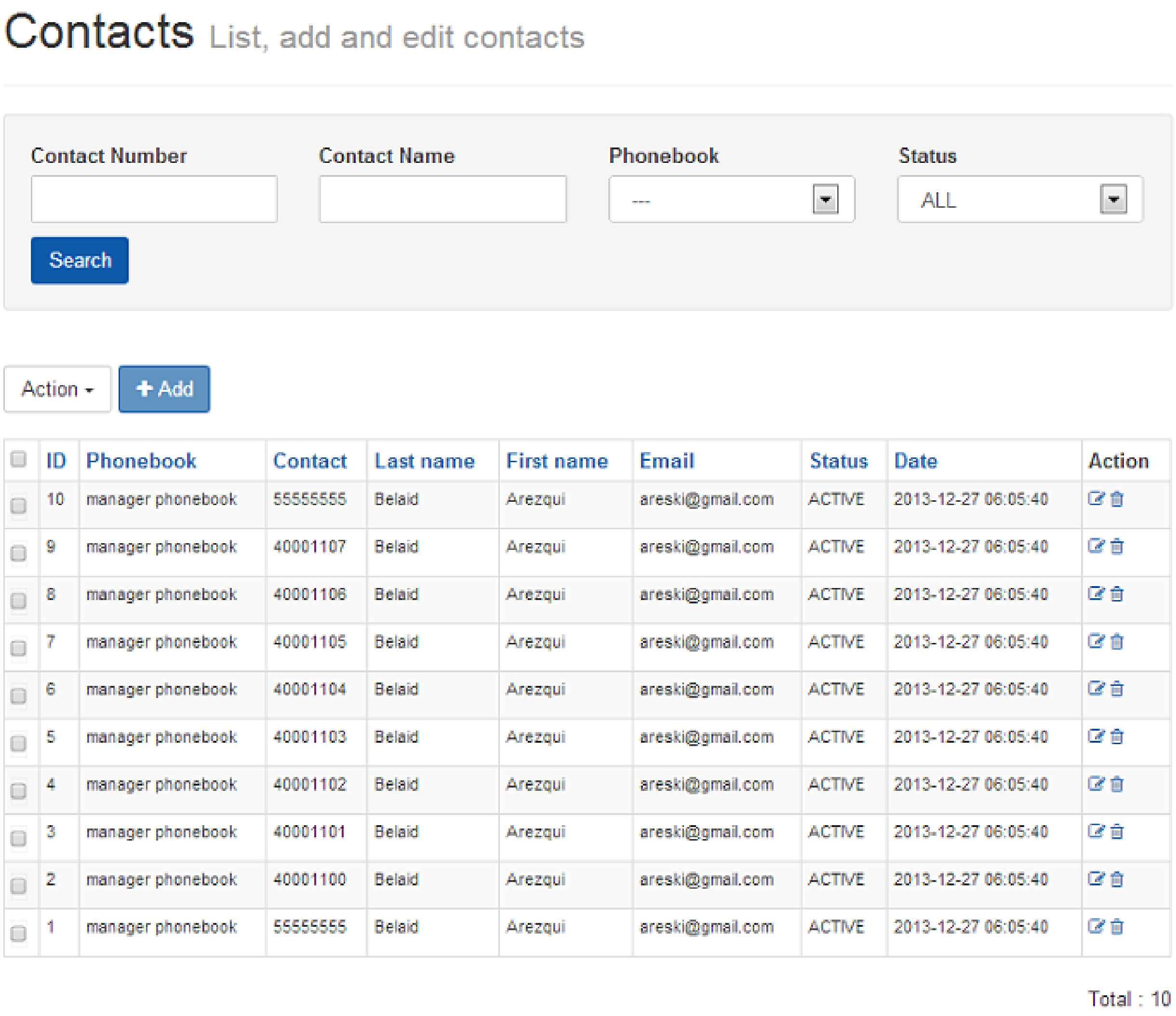
Name

manager phonebook

Description

Phonebook notes

Contact List:



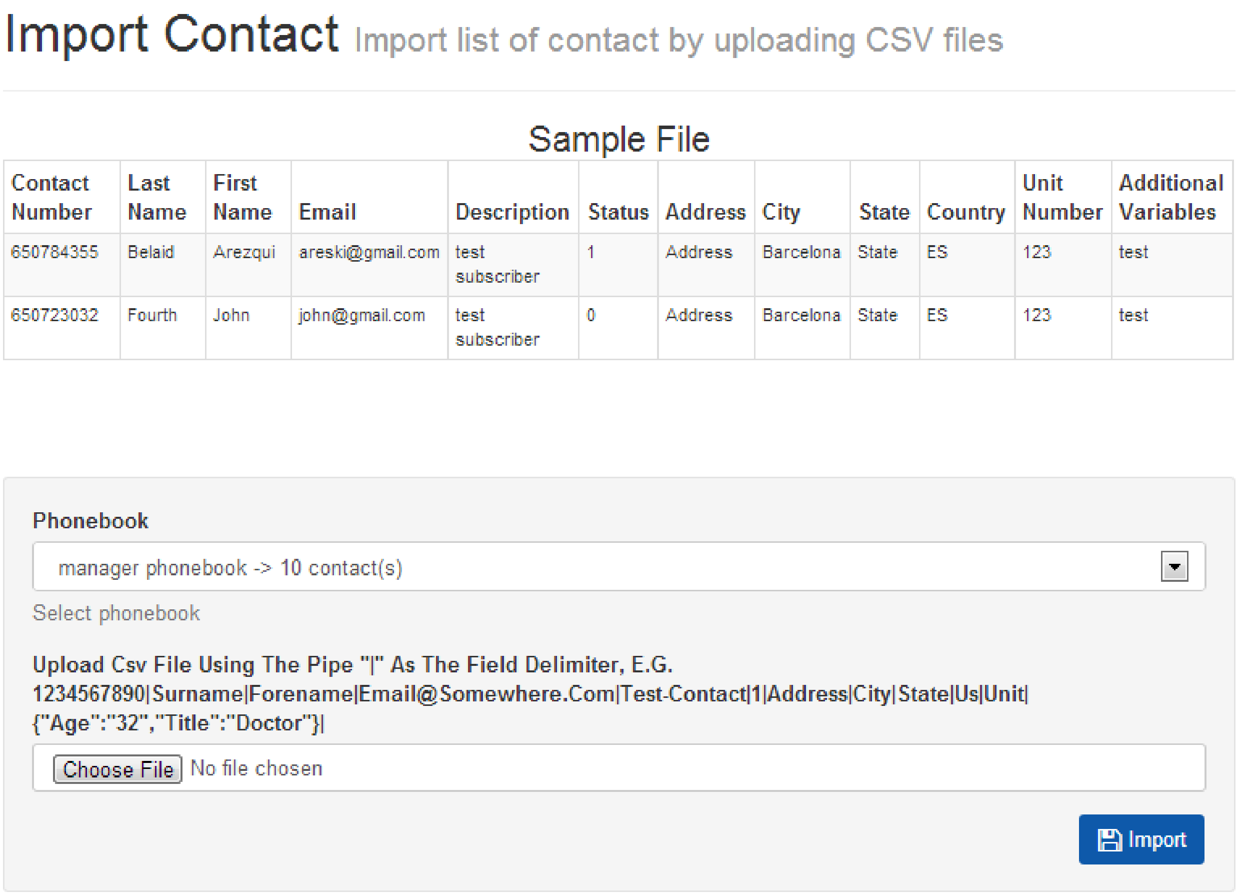
Contacts List, add and edit contacts

Contact Number	Contact Name	Phonebook	Status
		---	ALL

□	ID	Phonebook	Contact	Last name	First name	Email	Status	Date	Action
□	10	manager phonebook	55555555	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	9	manager phonebook	40001107	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	8	manager phonebook	40001106	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	7	manager phonebook	40001105	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	6	manager phonebook	40001104	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	5	manager phonebook	40001103	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	4	manager phonebook	40001102	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	3	manager phonebook	40001101	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	2	manager phonebook	40001100	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	1	manager phonebook	55555555	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑

Total : 10

Import contact:



Import Contact Import list of contact by uploading CSV files

Sample File

Contact Number	Last Name	First Name	Email	Description	Status	Address	City	State	Country	Unit Number	Additional Variables
650784355	Belaid	Arezqui	areski@gmail.com	test subscriber	1	Address	Barcelona	State	ES	123	test
650723032	Fourth	John	john@gmail.com	test subscriber	0	Address	Barcelona	State	ES	123	test

Phonebook

manager phonebook -> 10 contact(s)

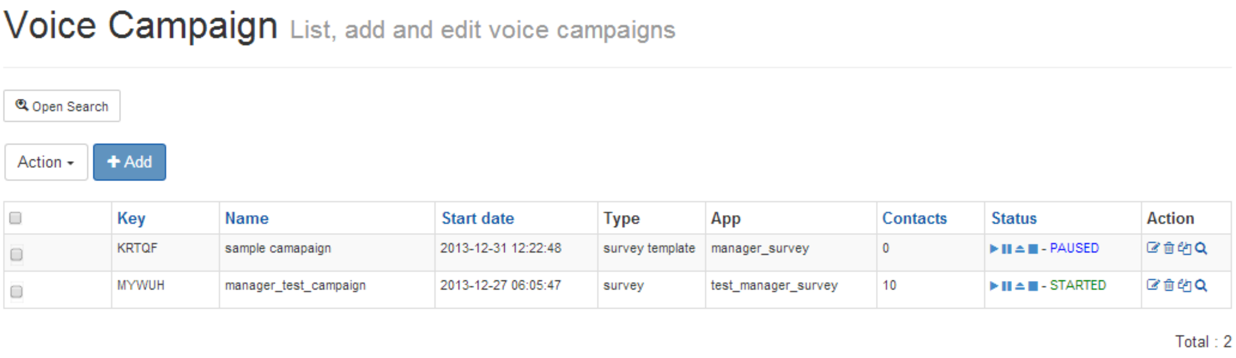
Select phonebook

Upload Csv File Using The Pipe "|" As The Field Delimiter, E.G. 1234567890|Surname|Forename|Email@Somewhere.Com|Test-Contact|1|Address|City|State|Us|Unit|{"Age":"32","Title":"Doctor"}|

7.2.8 Create campaign

To create a campaign, click on `Add` in campaign list, add details for the campaign. Important: Add the campaign's start and end dates with times & week-day exceptions. Select the gateway through which calls will be routed & the phonebook(s) linked with the contacts.

Campaign list:

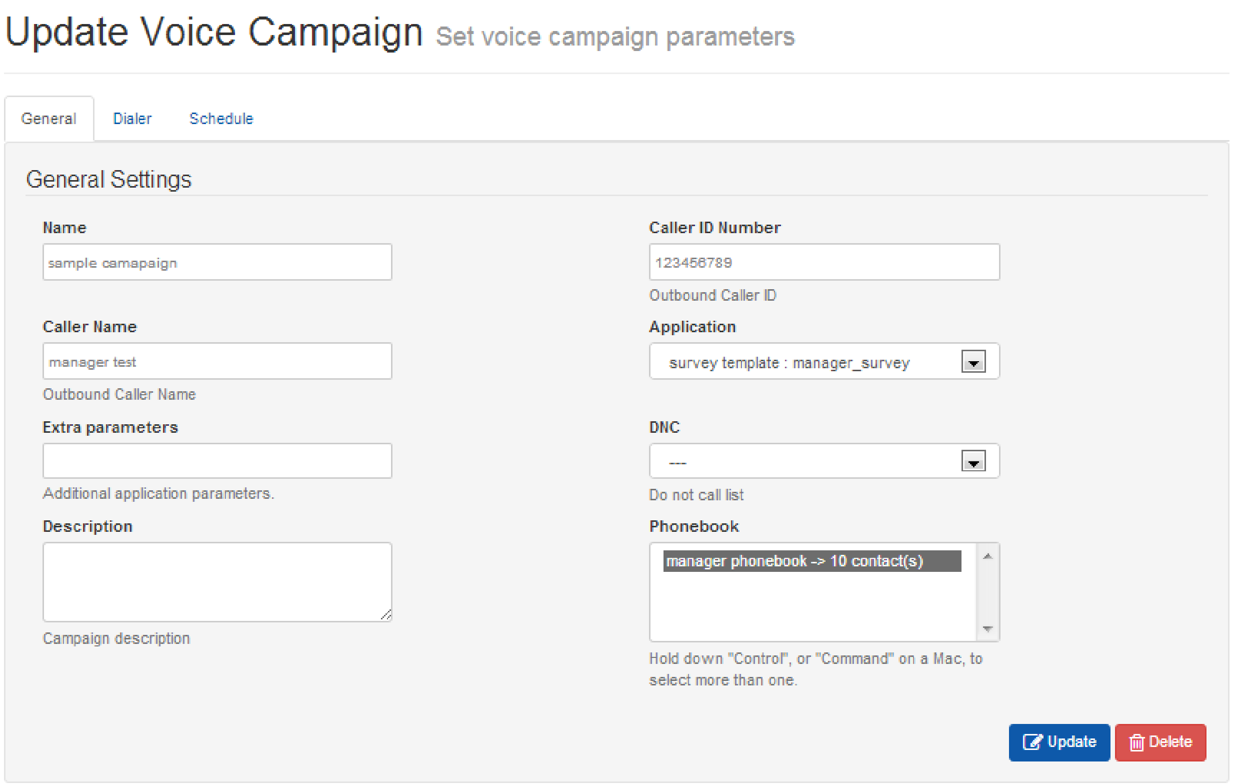


Voice Campaign List, add and edit voice campaigns

□	Key	Name	Start date	Type	App	Contacts	Status	Action
□	KRTQF	sample camapaign	2013-12-31 12:22:48	survey template	manager_survey	0	▶ ❚❚ ⏏ ■ - PAUSED	
□	MYWUH	manager_test_campaign	2013-12-27 06:05:47	survey	test_manager_survey	10	▶ ❚❚ ⏏ ■ - STARTED	

Total : 2

Update Campaign:



Update Voice Campaign Set voice campaign parameters

General Settings

Name: sample camapaign

Caller Name: manager test

Outbound Caller Name

Extra parameters:

Additional application parameters.

Description:

Campaign description

Caller ID Number: 123456789

Outbound Caller ID

Application: survey template : manager_survey

DNC: ---

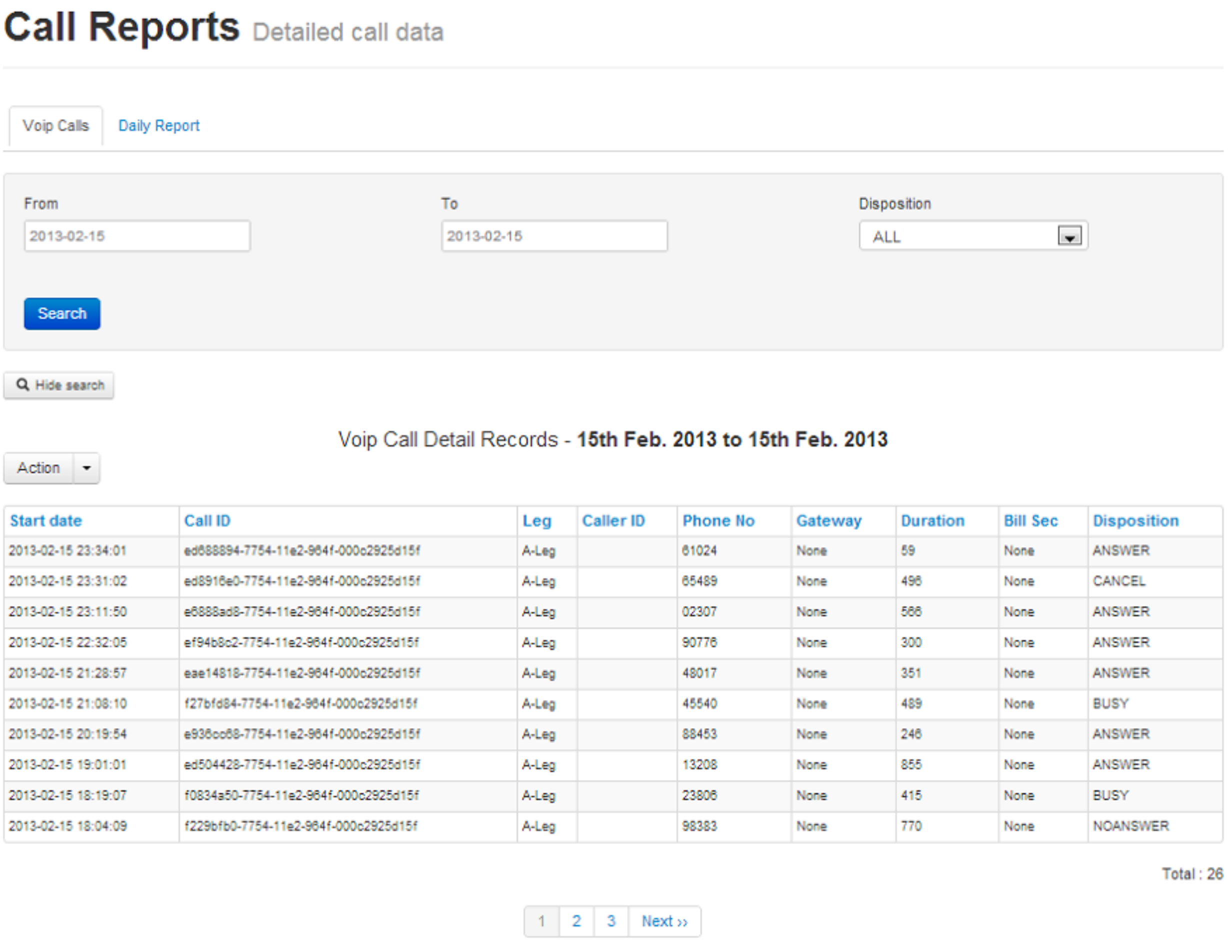
Do not call list

Phonebook: manager phonebook -> 10 contact(s)

Hold down "Control", or "Command" on a Mac, to select more than one.

7.2.9 VoIP Call Report

As per the status of a call-request, it will be stored in the VoIP call records. This gives information of all the calls & call statistics made with the call-request and also you can search for records on the basis of date range. You can export the VoIP call report into a csv file.



Call Reports Detailed call data

From: 2013-02-15

To: 2013-02-15

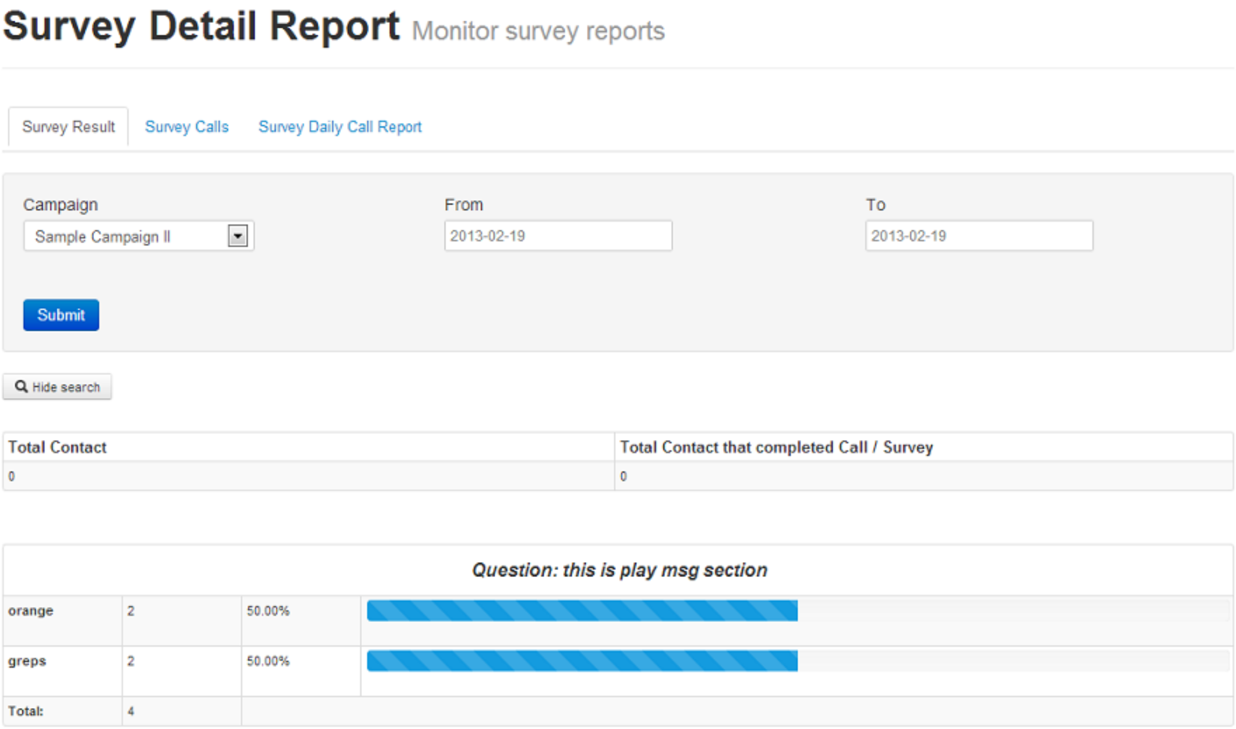
Disposition: ALL

Voip Call Detail Records - **15th Feb. 2013 to 15th Feb. 2013**

Start date	Call ID	Leg	Caller ID	Phone No	Gateway	Duration	Bill Sec	Disposition
2013-02-15 23:34:01	ed688894-7754-11e2-964f-000c2925d15f	A-Leg		61024	None	59	None	ANSWER
2013-02-15 23:31:02	ed8916e0-7754-11e2-964f-000c2925d15f	A-Leg		65489	None	496	None	CANCEL
2013-02-15 23:11:50	e6888ad8-7754-11e2-964f-000c2925d15f	A-Leg		02307	None	566	None	ANSWER
2013-02-15 22:32:05	ef94b8c2-7754-11e2-964f-000c2925d15f	A-Leg		90776	None	300	None	ANSWER
2013-02-15 21:28:57	eae14818-7754-11e2-964f-000c2925d15f	A-Leg		48017	None	351	None	ANSWER
2013-02-15 21:08:10	f27bfd84-7754-11e2-964f-000c2925d15f	A-Leg		45540	None	489	None	BUSY
2013-02-15 20:19:54	e936cc68-7754-11e2-964f-000c2925d15f	A-Leg		88453	None	246	None	ANSWER
2013-02-15 19:01:01	ed504428-7754-11e2-964f-000c2925d15f	A-Leg		13208	None	855	None	ANSWER
2013-02-15 18:19:07	f0834a50-7754-11e2-964f-000c2925d15f	A-Leg		23806	None	415	None	BUSY
2013-02-15 18:04:09	f229bfb0-7754-11e2-964f-000c2925d15f	A-Leg		98383	None	770	None	NOANSWER

Total : 26

7.2.10 Survey Report



Campaign: Sample Campaign II

From: 2013-02-19

To: 2013-02-19

Total Contact	Total Contact that completed Call / Survey
0	0

Question: this is play msg section

orange	2	50.00%	
greps	2	50.00%	
Total:	4		

When the survey is complete, the survey results can be inspected by clicking Reporting on the top and selecting Survey Results from the drop-down.

7.3 Survey Nodes

Most survey nodes have similar attributes that include:

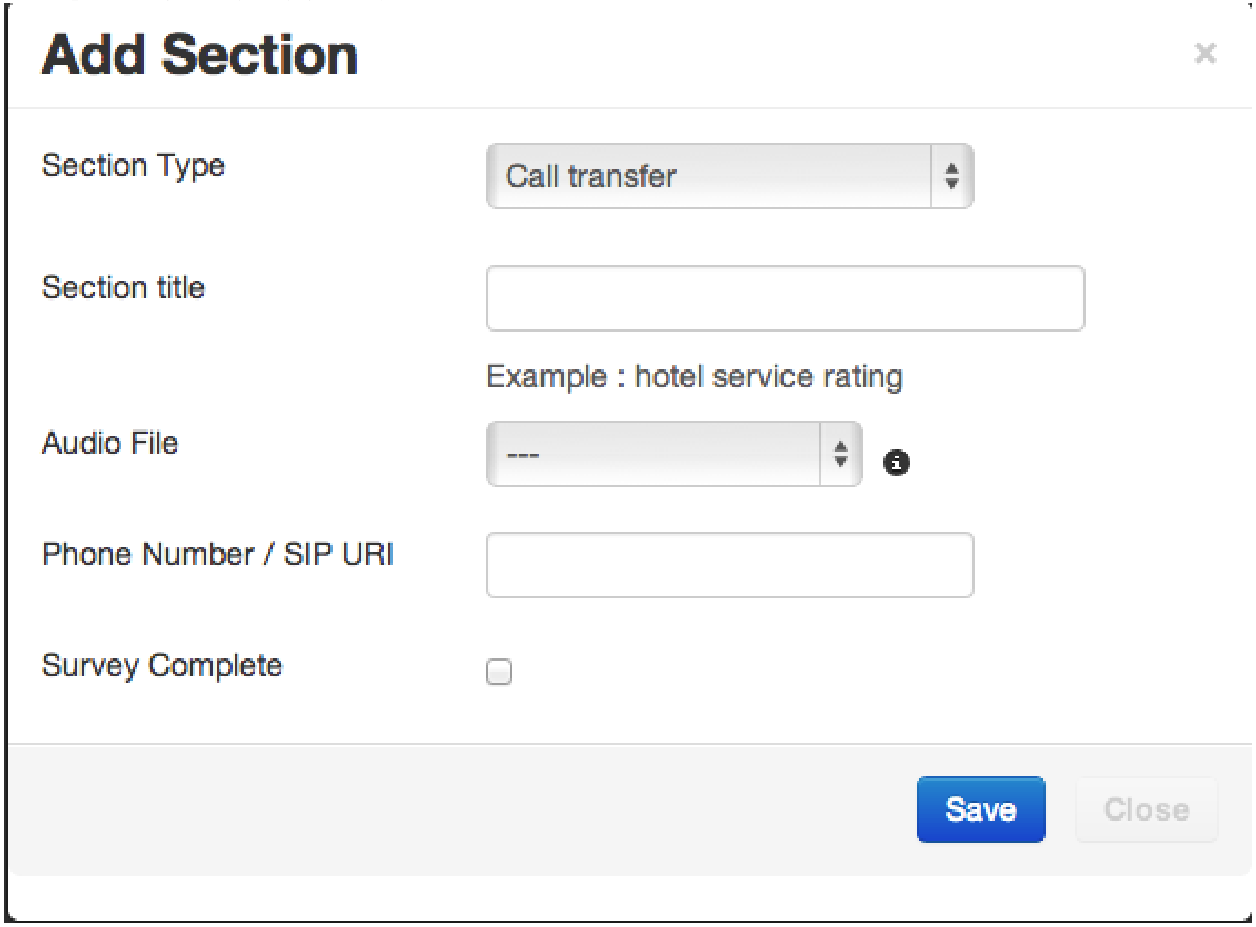
- Section Title, the title of the section and becomes the TTS (text To Speech)
- Audio File, The sound file to be played for this node.
- Retries, if no valid input, then the question can be asked again.
- Timeout, the amount of time before it is considered that no input has been received.
- Check Validity, whether the answer is deemed valid.

- Audio Invalid Input, The audio to play when invalid input is received.
- Survey Complete, check this field to mark the survey is complete for reporting & retry purposes.

7.3.1 Call Transfer

Used for "press one" campaigns and live lead generation. This node bridges the call to the number in the Phone Number field when the contact answers.

The field can have a telephone number or something like sofia/gateway/my.gateway/12345 which will direct the call via a specified gateway (my.gateway) to number 12345.



Add Section

Section Type: Call transfer

Section title:

Example : hotel service rating

Audio File: ---

Phone Number / SIP URI:

Survey Complete □

In order to support screen-pops and deliver other data, there has been some data included in the SIP message

By default the following data is included in the SIP header on call transfer: P-CallRequest-ID is the Call request ID P-Contact-ID is is the Contact ID.

It is envisaged that these ID numbers can be used to do database look-ups on the Newfies-Dialer database.

Furthermore, there is a optional SIP header that can be added.

P-Contact-Transfer-Ref:

This can be added against the Contact in the "Additional Parmeters (JSON)" field. Simply add the "transfer_ref" keyword and string to send in the SIP message as follows:

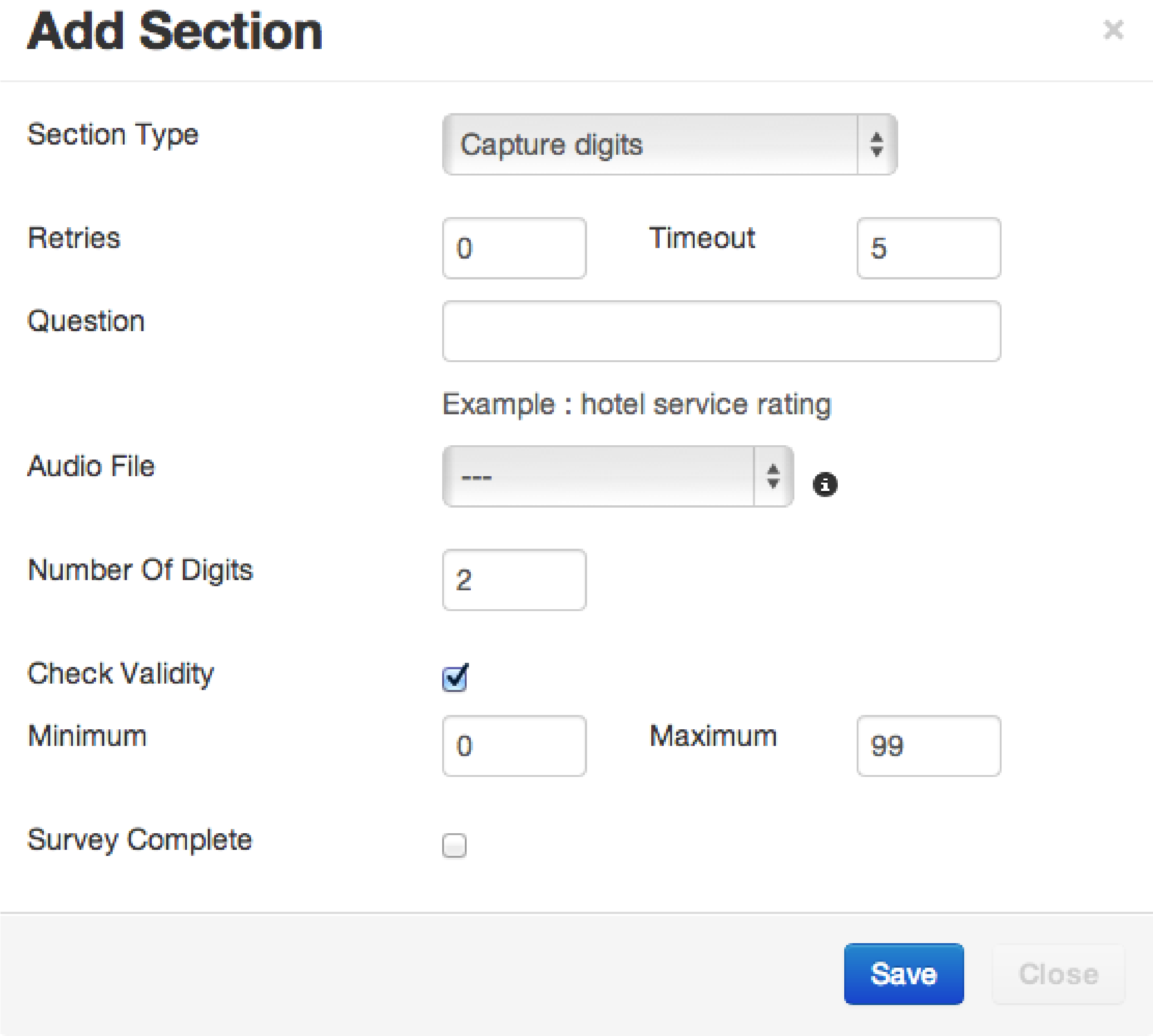
{"transfer_ref": "My-Unique-Ref-Number"}

In the SIP headers, you will see:

P-CallRequest-ID: 3 P-Contact-ID: 1 P-Contact-Transfer-Ref: My-Unique-Ref-Number

7.3.2 Capture Digits

Captures a series of digits, e.g. a telephone number or account number and stores it in the reporting. The number of digits and the minimum and maximum values can be set.

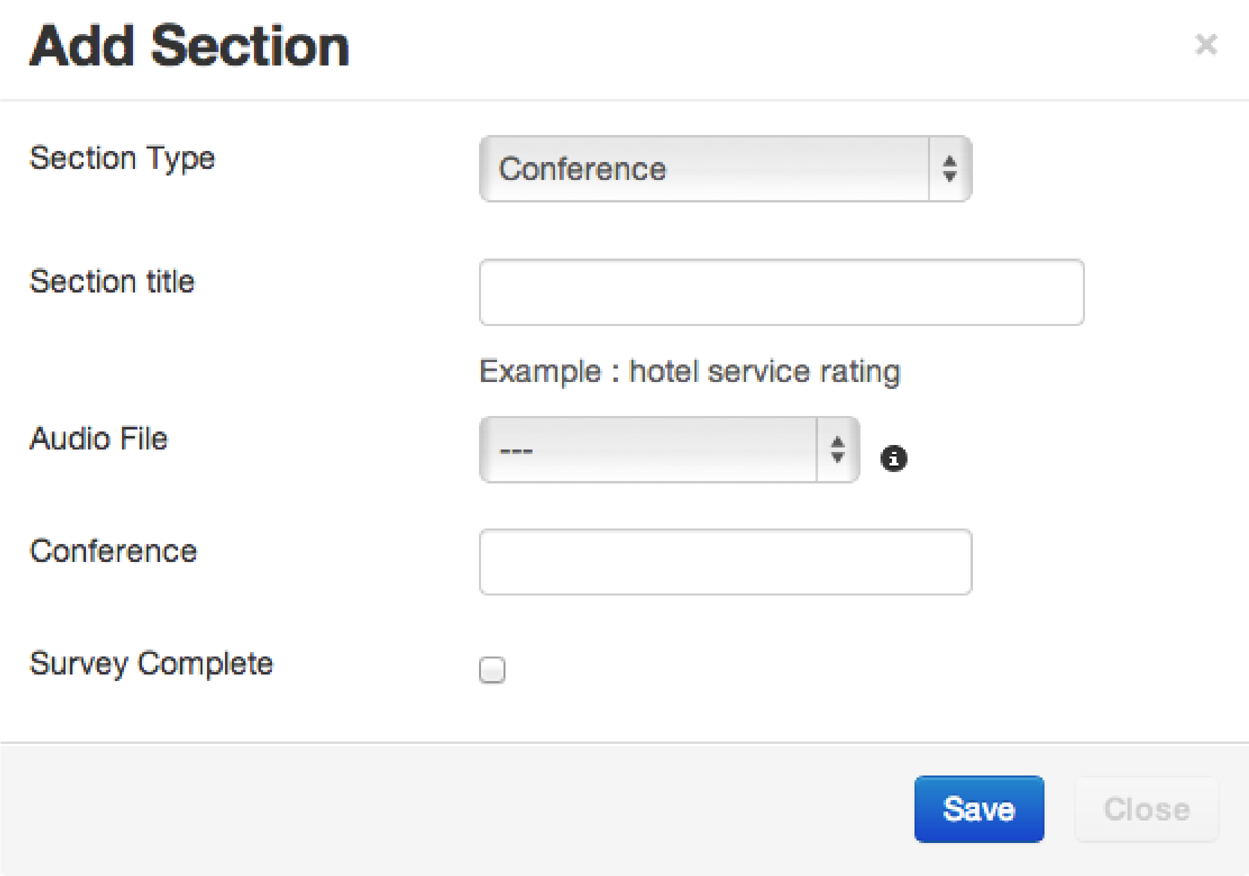


Add Section

Section Type	Capture digits		
Retries	0	Timeout	5
Question			
	Example : hotel service rating		
Audio File	---		
Number Of Digits	2		
Check Validity	☑		
Minimum	0	Maximum	99
Survey Complete	□		

7.3.3 Conference

Set up a conference with outbound calls. The default conference number in Freeswitch is 9888. The Freeswitch dialplan can be adjusted to add more conferences.



Add Section

Section Type: Conference

Section title:

Example : hotel service rating

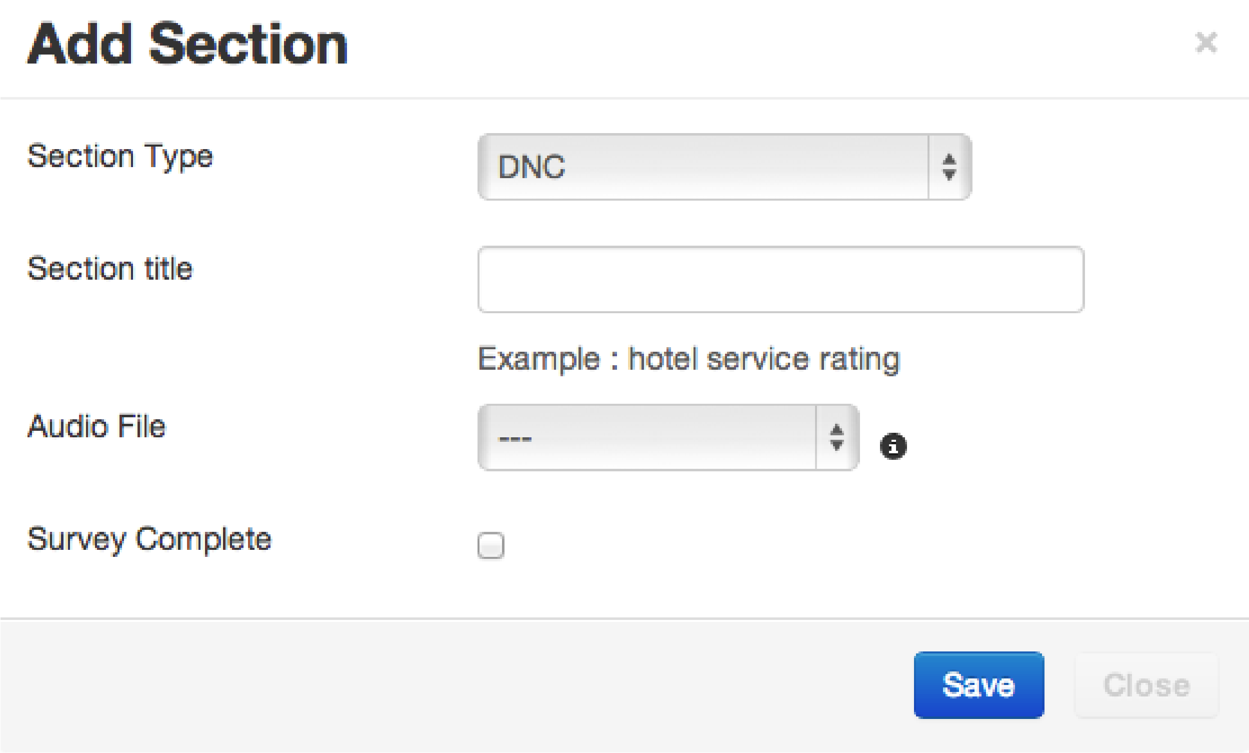
Audio File: ---

Conference:

Survey Complete □

7.3.4 DNC

Do Not Call node, which will add the called contact to the DNC list configured in the campaign.



Add Section

Section Type: DNC

Section title:

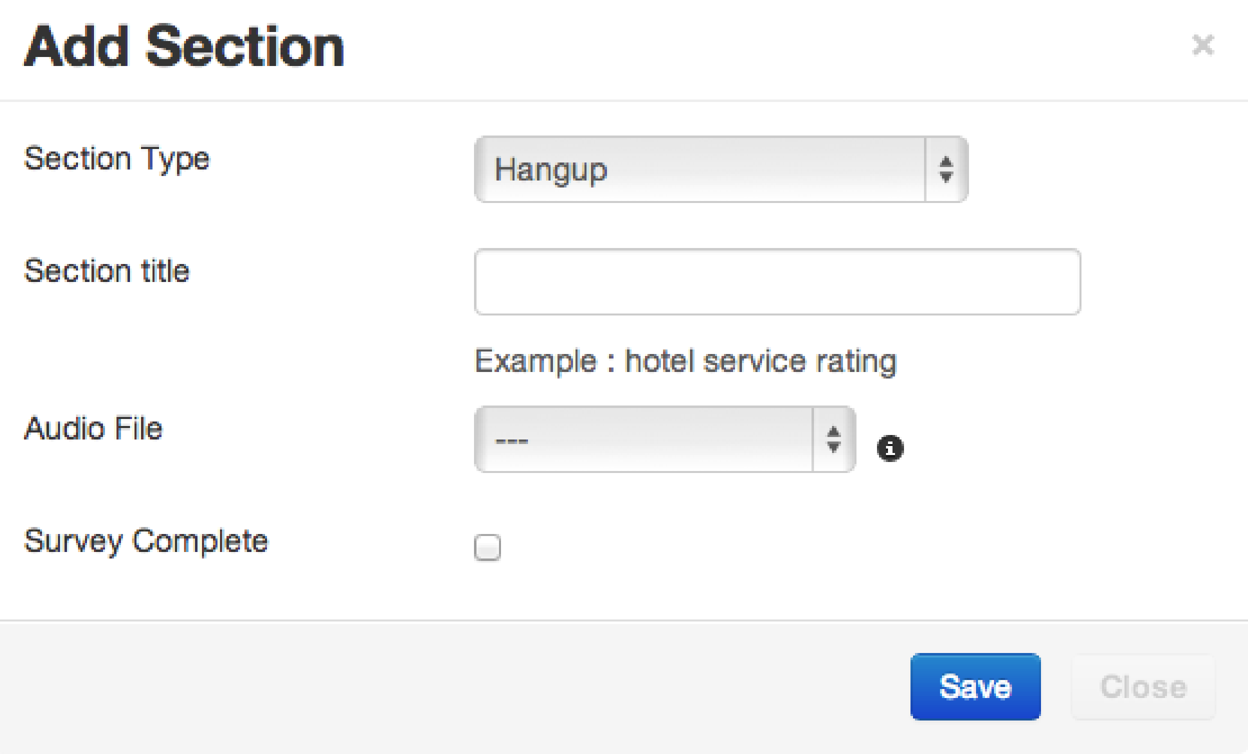
Example : hotel service rating

Audio File: ---

Survey Complete □

7.3.5 Hangup

Hang up the call at the end of the survey.



Add Section

Section Type: Hangup

Section title:

Example : hotel service rating

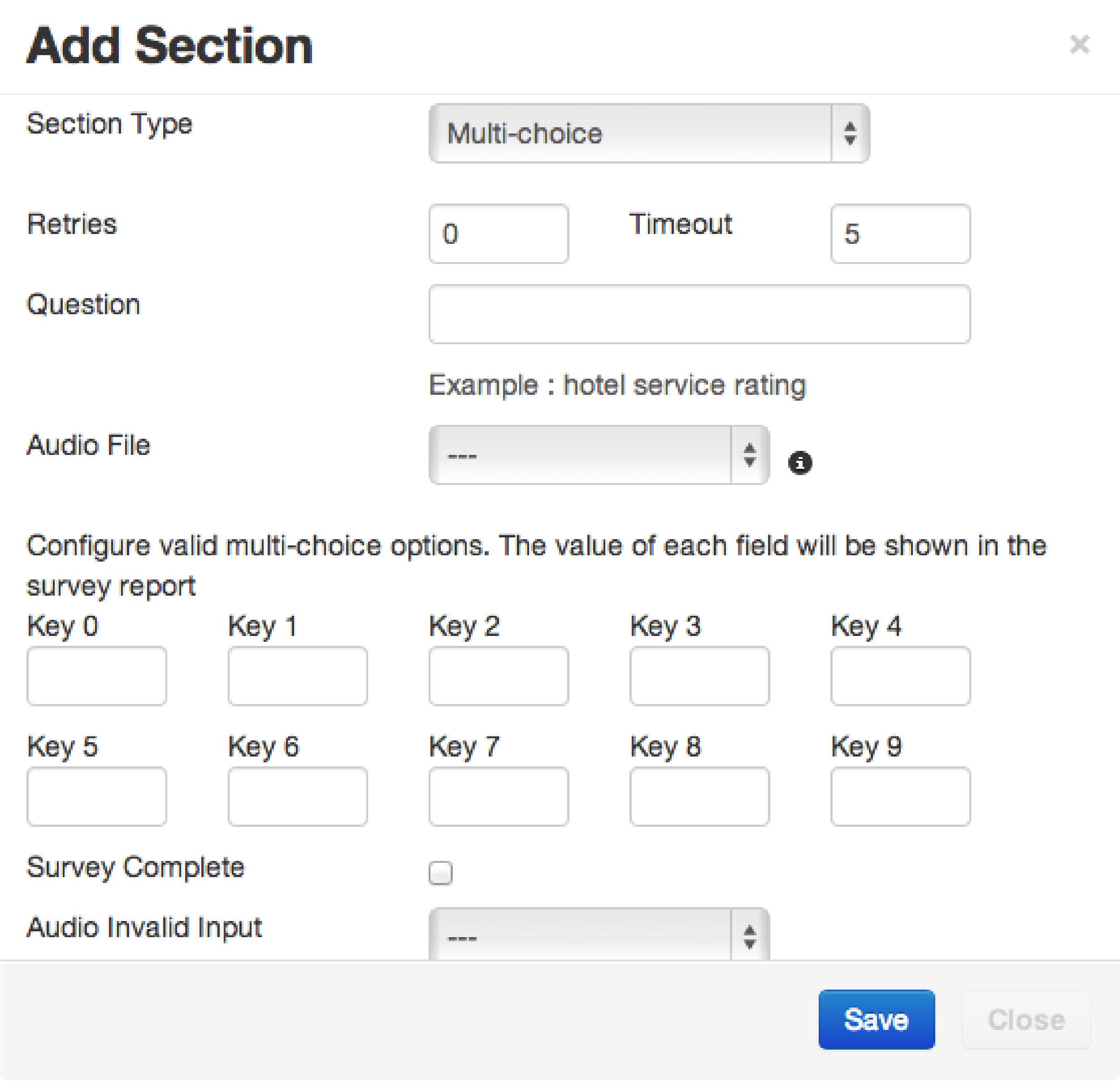
Audio File: ---

Survey Complete: □

7.3.6 Multi-Choice

Multi-Choice offering options 0 to 9. The value placed in the “Key X” fields appears in the survey reports.

Survey branching can be used to control the flow of the IVR depending on the key pressed.



Add Section

Section Type: Multi-choice

Retries: 0 Timeout: 5

Question:

Example : hotel service rating

Audio File: ---

Configure valid multi-choice options. The value of each field will be shown in the survey report

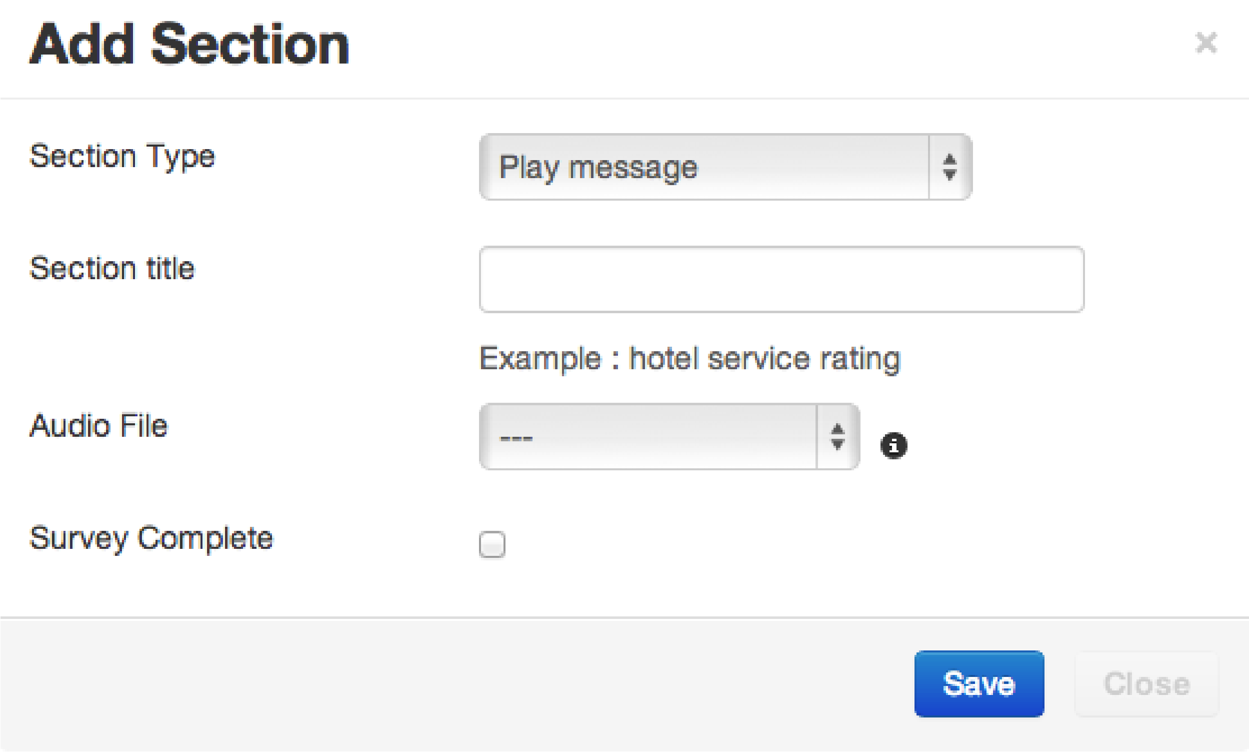
Key 0	Key 1	Key 2	Key 3	Key 4
Key 5	Key 6	Key 7	Key 8	Key 9

Survey Complete □

Audio Invalid Input: ---

7.3.7 Play Message

Play message is simply to play a message, either with TTS or pre-recorded audio.



Add Section

Section Type: Play message

Section title:

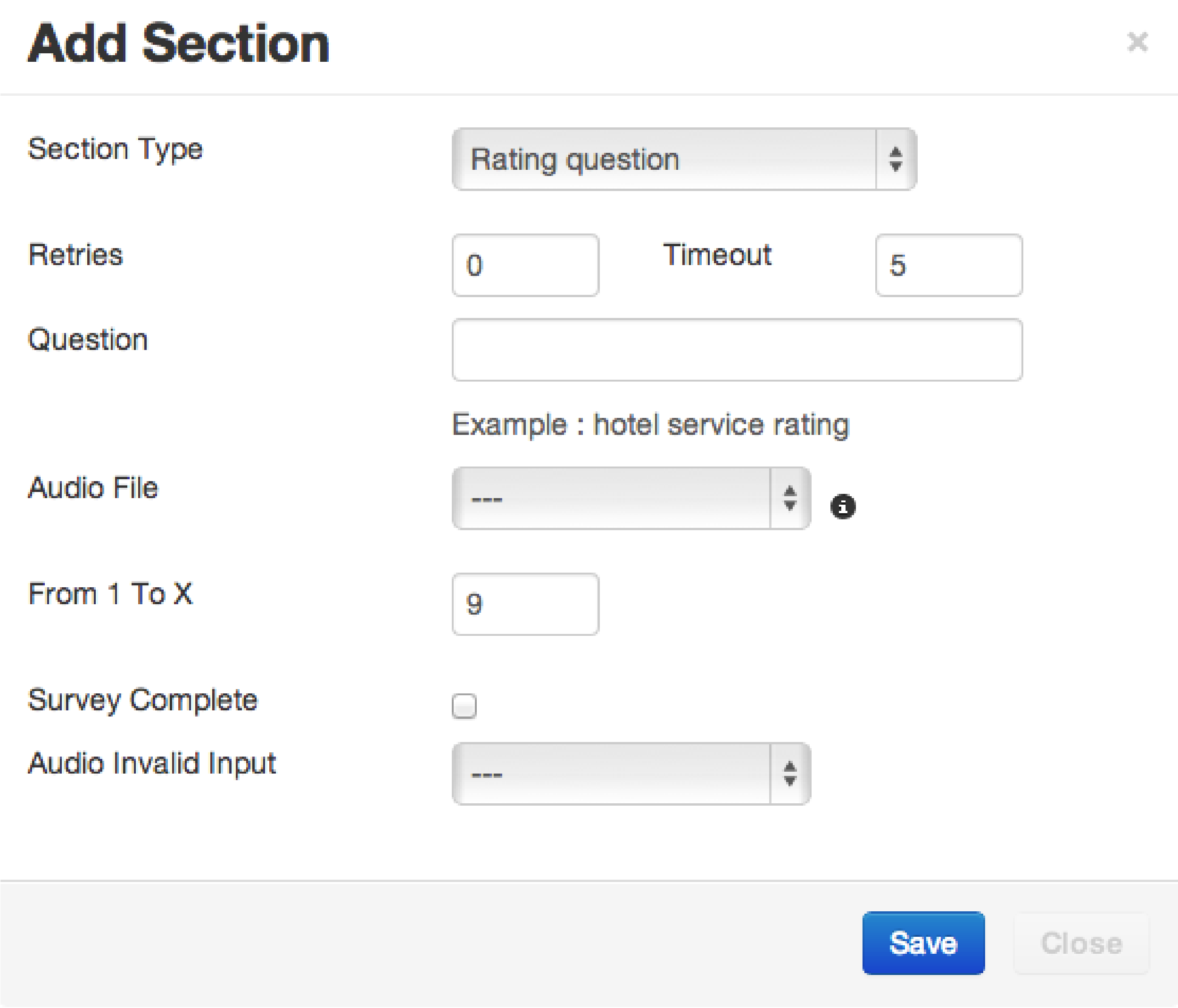
Example : hotel service rating

Audio File: ---

Survey Complete □

7.3.8 Rating Question

Rating allows the entry of a digit or digits from 1 to X, where X is a number you select.



Add Section

Section Type: Rating question

Retries: 0

Timeout: 5

Question:

Example : hotel service rating

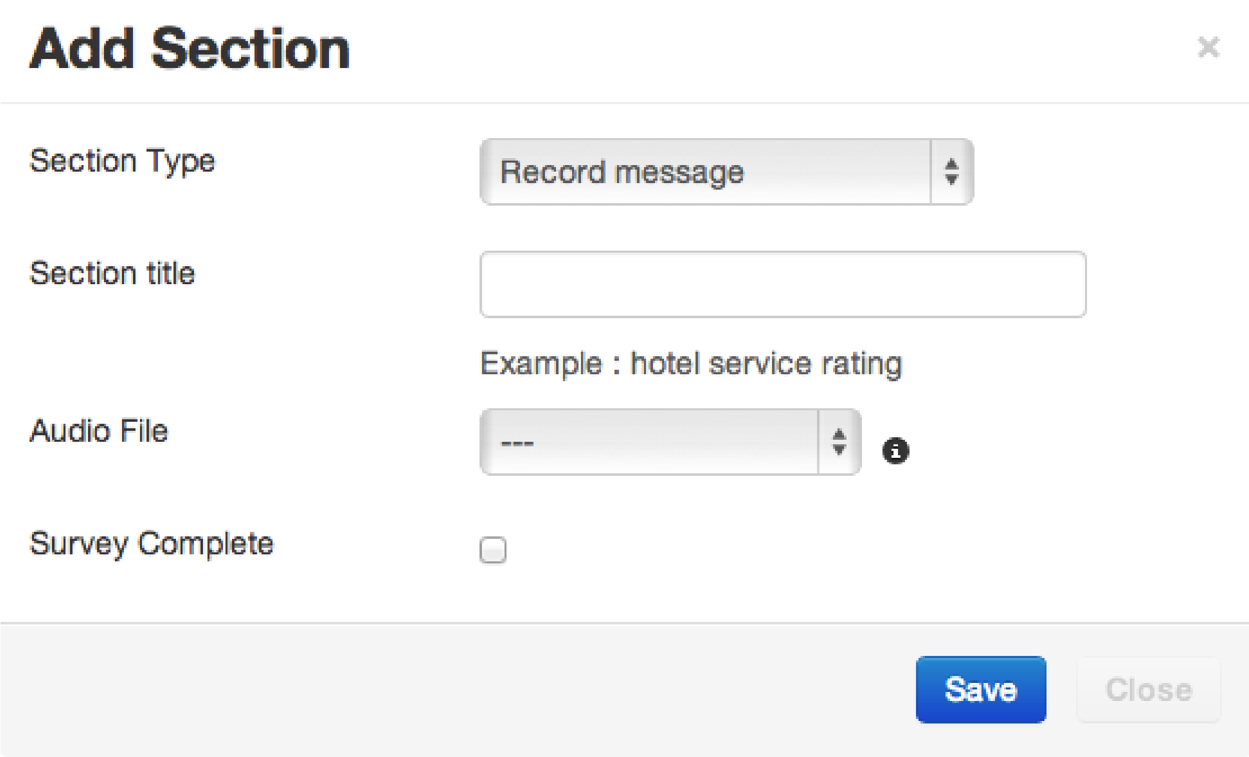
Audio File: ---

From 1 To X: 9

Survey Complete ☐

Audio Invalid Input: ---

7.3.9 Record Message



Add Section

Section Type: Record message

Section title:

Example : hotel service rating

Audio File: ---

Survey Complete: □

Record a message, the system stops recording after 3 seconds of silence, or by pressing the # key. The IVR flow will then continue.

Recordings can be listened to in the survey reports menu.

7.3.10 SMS Message

Play a message to the customer, either via TTS or audio file, then send an SMS message to the customer.

Note that when the campaign is created, an SMS gateway must be selected under the Dialer tab.

Reports on SMS messages can be be viewed under the Reporting section.

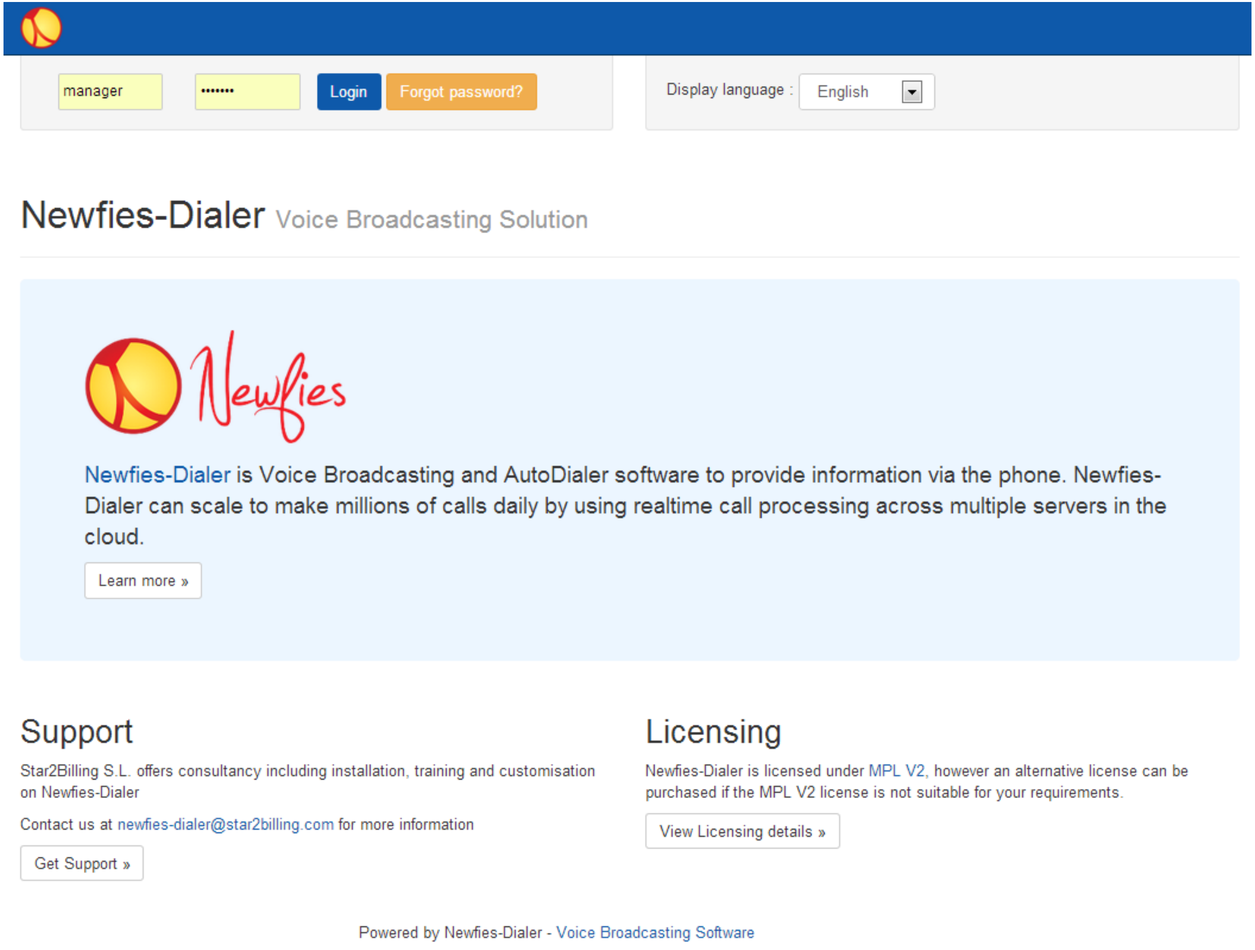
7.4 Customer Panel

Customer Panel URL : http://localhost:8000/admin/

This application provides a user interface for restricted management of the User's Campaigns, Phonebooks and Subscribers. It also provides detailed reporting of calls and message delivery.

7.4.1 Login Page

Index page for the customer interface after successful login with user credentials



Newfies-Dialer Voice Broadcasting Solution

Newfies-Dialer is Voice Broadcasting and AutoDialer software to provide information via the phone. Newfies-Dialer can scale to make millions of calls daily by using realtime call processing across multiple servers in the cloud.

Support

Star2Billing S.L. offers consultancy including installation, training and customisation on Newfies-Dialer

Contact us at newfies-dialer@star2billing.com for more information

Licensing

Newfies-Dialer is licensed under MPL V2, however an alternative license can be purchased if the MPL V2 license is not suitable for your requirements.

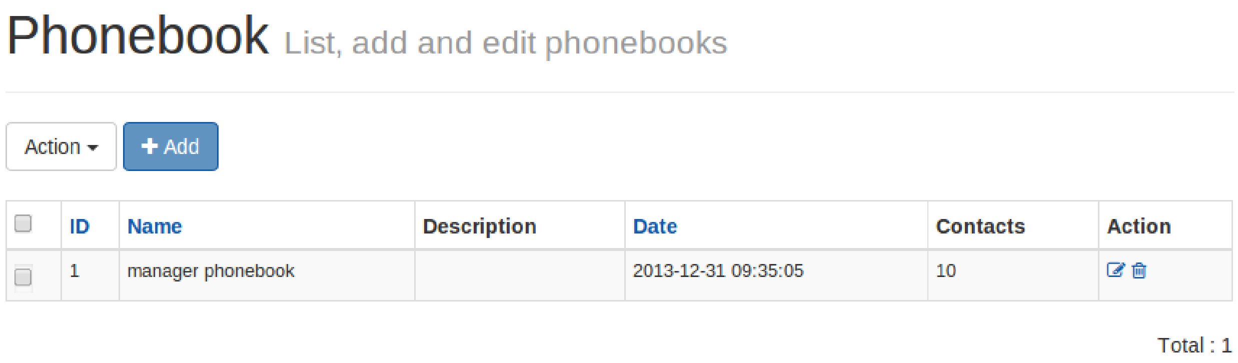
Powered by Newfies-Dialer - Voice Broadcasting Software

7.4.2 Phonebook

The phonebook list will be displayed from the following URL. You can add a new phonebook by clicking `Add phonebook` and add the name of a phonebook and its description. Also from the phonebook list, click on the phonebook to update.

URL:

- http://localhost:8000/phonebook/



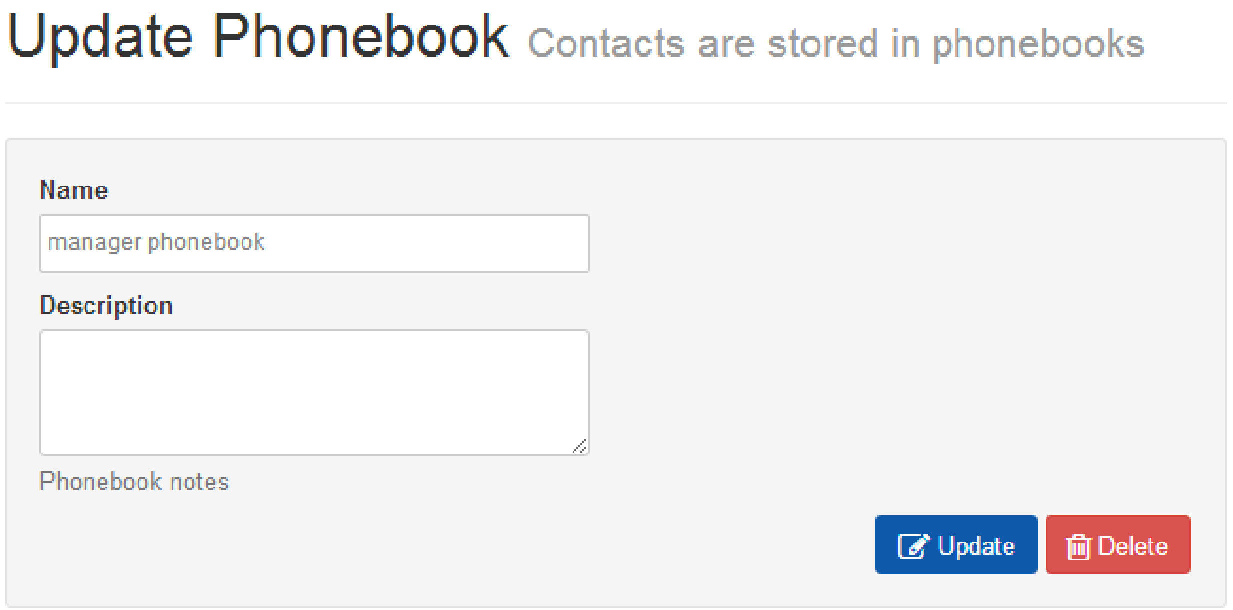
□	ID	Name	Description	Date	Contacts	Action
□	1	manager phonebook		2013-12-31 09:35:05	10	

Total : 1

To Add/Update a Phonebook for a logged in user

URL:

- http://localhost:8000/phonebook/add/
- http://localhost:8000/phonebook/1/



Name: manager phonebook

Description:

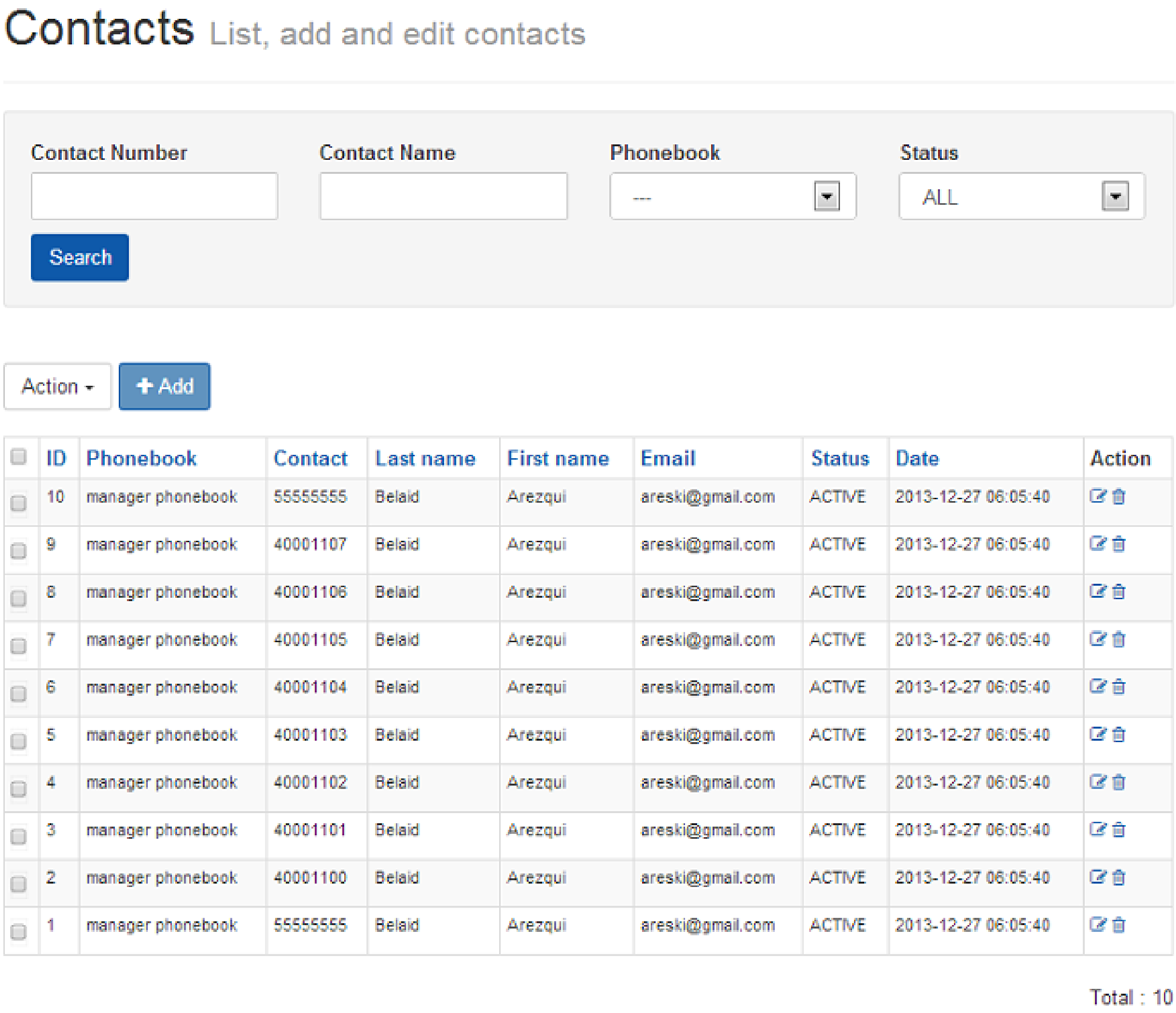
Phonebook notes

7.4.3 Contact

The contact list will be displayed from following the URL. You can add a new contact by clicking `Add contact` & adding the contact details (i.e. phone number, name, description about contact, contact status) under the logged in user's phonebook from the phonebook list. On the contact list, click on the contact to update.

URL:

- http://localhost:8000/contact/



Contacts List, add and edit contacts

Contact Number

Contact Name

Phonebook: ---

Status: ALL

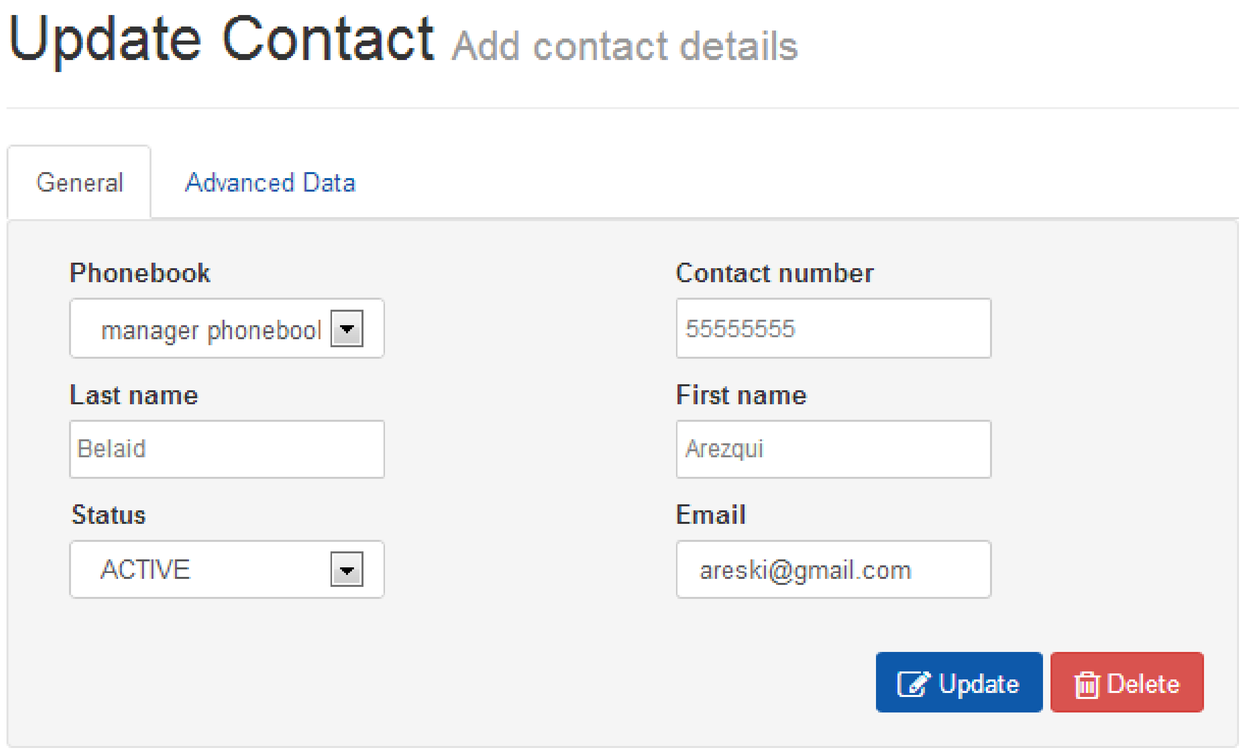
□	ID	Phonebook	Contact	Last name	First name	Email	Status	Date	Action
□	10	manager phonebook	55555555	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	9	manager phonebook	40001107	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	8	manager phonebook	40001106	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	7	manager phonebook	40001105	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	6	manager phonebook	40001104	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	5	manager phonebook	40001103	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	4	manager phonebook	40001102	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	3	manager phonebook	40001101	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	2	manager phonebook	40001100	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑
□	1	manager phonebook	55555555	Belaid	Arezqui	areski@gmail.com	ACTIVE	2013-12-27 06:05:40	✎ 🗑

Total : 10

To Add/Update a contact in a phonebook

URL:

- http://localhost:8000/contact/add/
- http://localhost:8000/contact/1/



Update Contact Add contact details

Phonebook: manager phonebool

Contact number: 55555555

Last name: Belaid

First name: Arezqui

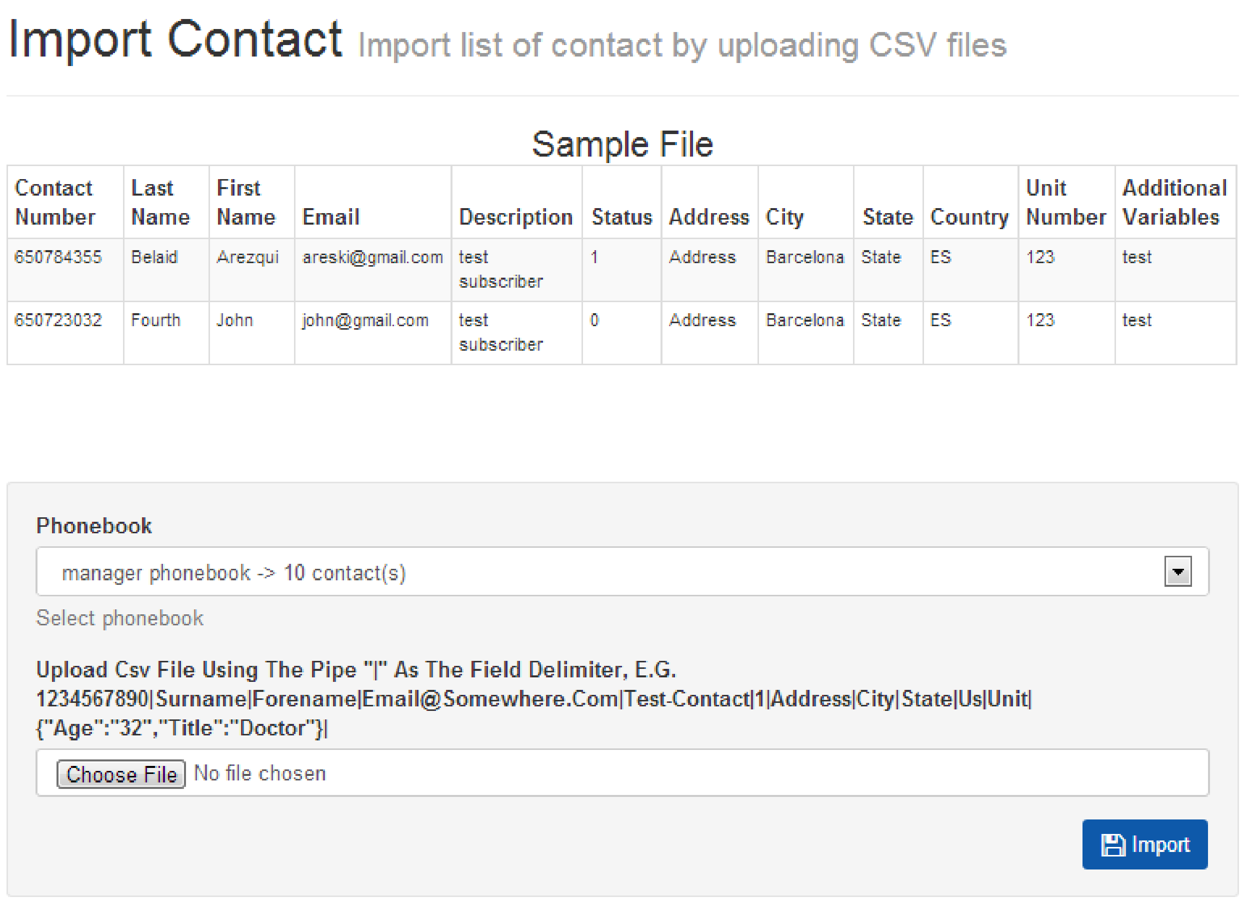
Status: ACTIVE

Email: areski@gmail.com

To import bulk contacts into a phonebook, click on `Import.` where you can upload contacts via a CSV file under a logged in user's phonebook.

URL:

- http://localhost:8000/contact_import/



Import Contact Import list of contact by uploading CSV files

Sample File

Contact Number	Last Name	First Name	Email	Description	Status	Address	City	State	Country	Unit Number	Additional Variables
650784355	Belaid	Arezqui	areski@gmail.com	test subscriber	1	Address	Barcelona	State	ES	123	test
650723032	Fourth	John	john@gmail.com	test subscriber	0	Address	Barcelona	State	ES	123	test

Phonebook

manager phonebook -> 10 contact(s)

Select phonebook

Upload Csv File Using The Pipe "|" As The Field Delimiter, E.G. 1234567890|Surname|Forename|Email@Somewhere.Com|Test-Contact|1|Address|City|State|Us|Unit|{"Age":"32","Title":"Doctor"}|

7.4.4 Survey

The survey application for Newfies-Dialer allows polls and surveys to be taken over the phone.

Each contact is called, and then played a sound file. After the sound file is heard, the user can enter their answer through the phone keypad using keys 0 to 9.

Therefore before creating the survey, the first job is to upload the audio for the survey. One audio file is required for each question.

Uploading Audio Files:

Select Audio Files from the top menu then click add.

Enter a name to describe the audio, then click chose file, select the file you require from your computer, then click submit.

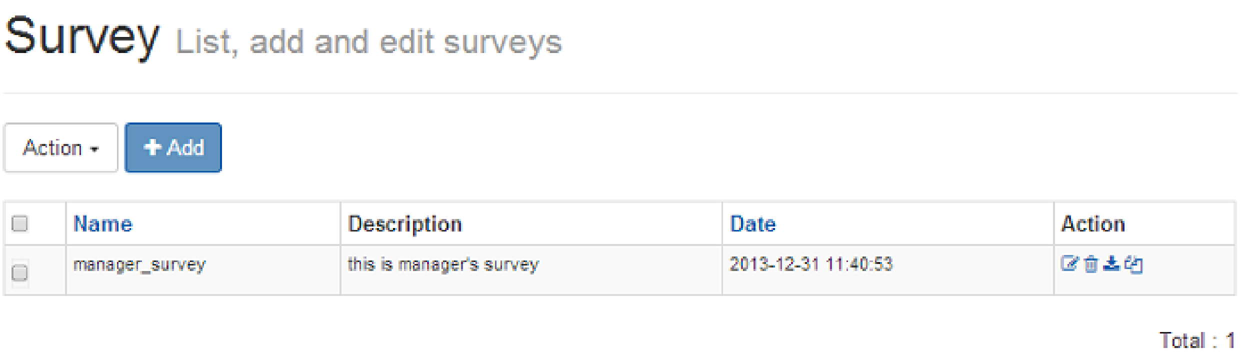
Note that only mp3, Wav and ogg formats are supported.

The audio file will then be uploaded and renamed so that it is unique.

Create the Survey:

URL:

- http://localhost:8000/module/survey/
- http://localhost:8000/module/survey/add/
- http://localhost:8000/module/survey/1/



Survey List, add and edit surveys

□	Name	Description	Date	Action
□	manager_survey	this is manager's survey	2013-12-31 11:40:53	

Total : 1

Select Modules from the top menu, then Survey.

Click the add button, then give the survey a name and description, then click Submit.

A button will now appear to add a question.

In the question field, put in some text to describe the question - e.g "What is 1+1"; select the audio file pertaining to the question which was uploaded in the previous step.

If no audio file is selected, then the system will automatically play the text in the question field using the text to speech engine.

Then click Add Response. A further two fields will appear named Key Digit and Key Value. In key Digit put a number from 0 to 9 which should be pressed for this answer. In the example "What is 1+1", "2" should be placed in the Key Digit Field"

The Key Value field is used in the survey reports, and so in this case, you would put "Correct" as 1+1=2. You may chose to add responses 0,1 and 3 to 9 as key digits, with key values of "Wrong" as these answers will then be summed up in the Survey Reports.

You can then go on to add another question, and its associated responses. On completion, click Update Survey.

To use the Survey in a campaign, simply create a campaign as normal, and select the Survey name in the Application drop-down.

Survey Results

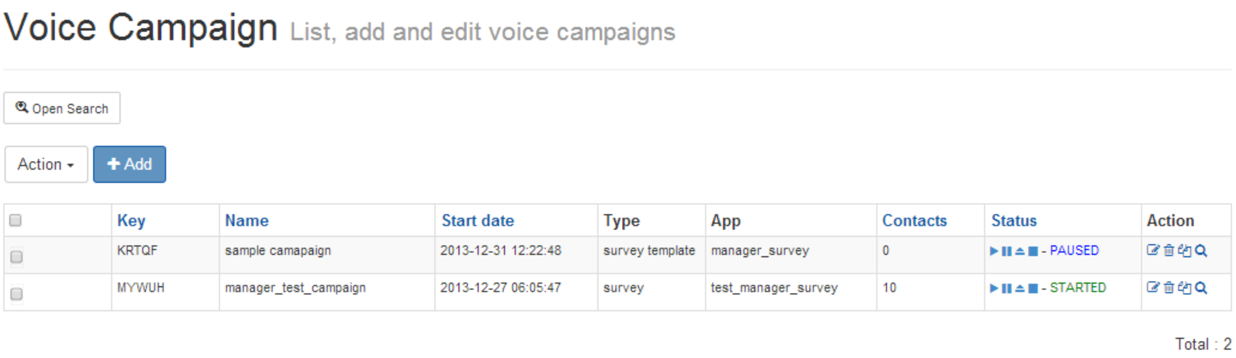
When the survey is complete, the survey results can be inspected by clicking Reporting on the top and selecting Survey Results from the drop-down.

7.4.5 Campaign

The campaign list will be displayed from the following URL. You can add a new campaign for the logged in user by clicking `Add campaign`. When adding a campaign, it is important to add the campaign's start and end dates with time & week-day exceptions. Select the gateway through which calls will be routed & phonebook(s) that are linked with contacts from the campaign list, click on campaign to update.

URL:

- http://localhost:8000/campaign/



Voice Campaign List, add and edit voice campaigns

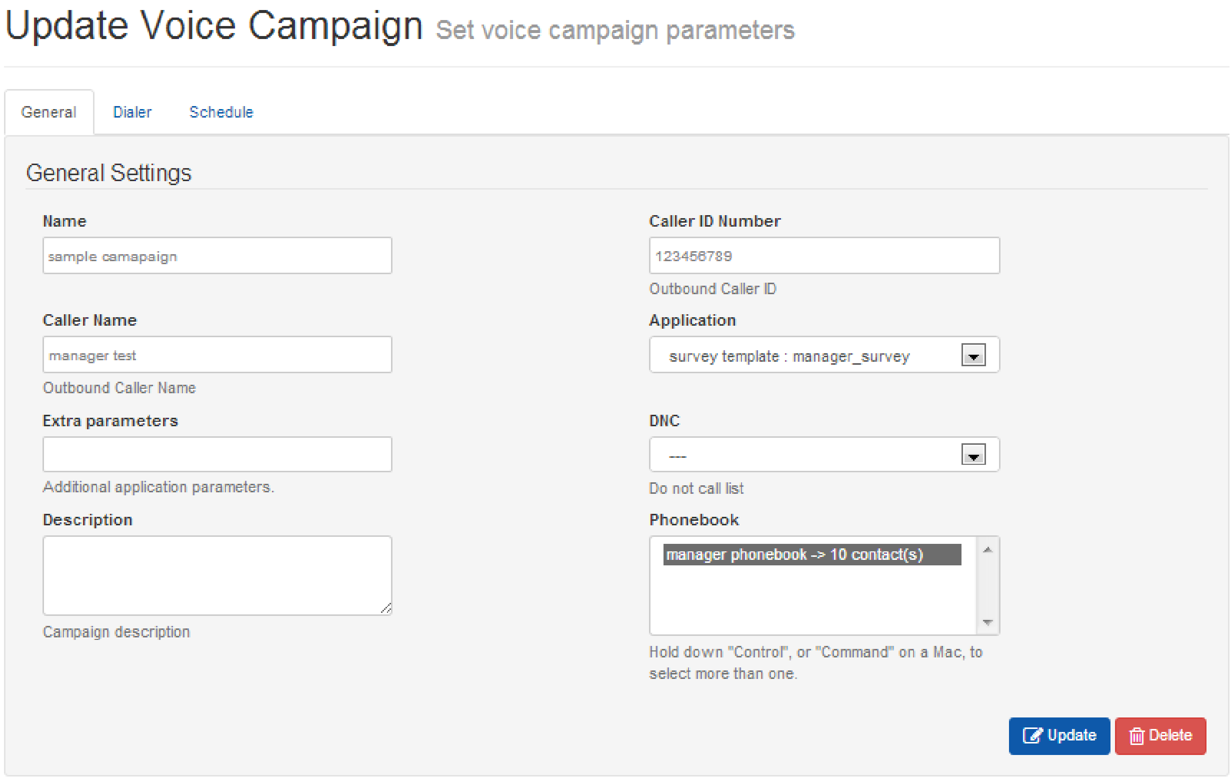
□	Key	Name	Start date	Type	App	Contacts	Status	Action
□	KRTQF	sample camapaign	2013-12-31 12:22:48	survey template	manager_survey	0	- PAUSED	
□	MYWUH	manager_test_campaign	2013-12-27 06:05:47	survey	test_manager_survey	10	- STARTED	

Total : 2

To Add/Update a Campaign for a logged in user

URL:

- http://localhost:8000/campaign/add/
- http://localhost:8000/campaign/1/



Update Voice Campaign Set voice campaign parameters

General Settings

Name: sample camapaign

Caller ID Number: 123456789
Outbound Caller ID

Caller Name: manager test
Outbound Caller Name

Application: survey template : manager_survey

Extra parameters:
Additional application parameters.

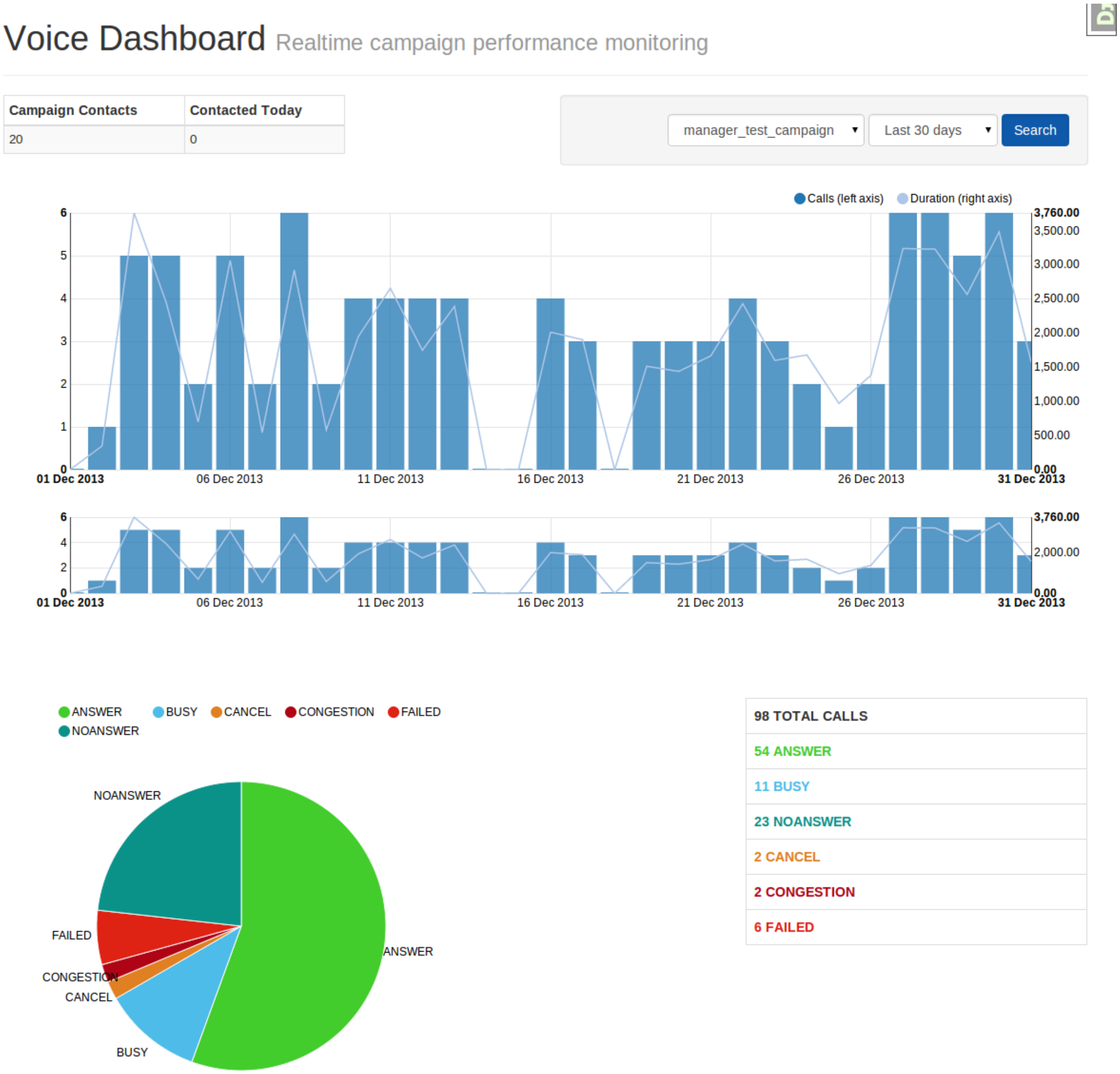
DNC: ---
Do not call list

Description:
Campaign description

Phonebook: manager phonebook -> 10 contact(s)
Hold down "Control", or "Command" on a Mac, to select more than one.

7.4.6 Dashboard

Dashboard provides a contact and call reporting for the running campaign.



Voice Dashboard Realtime campaign performance monitoring

Campaign Contacts	Contacted Today
20	0

98 TOTAL CALLS

54 ANSWER

11 BUSY

23 NOANSWER

2 CANCEL

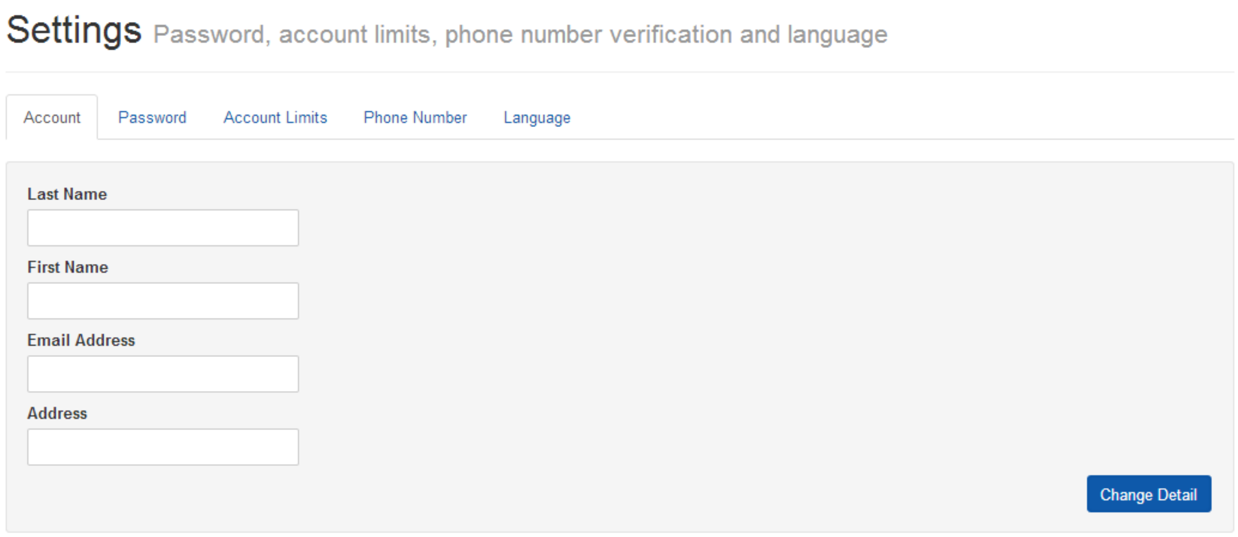
2 CONGESTION

6 FAILED

7.4.7 User Settings

The settings page provides a number of functions:

- Account - Change the detail of the account.
- Password - Change the password.
- Notifications - Display system notifications.
- Limitation - Displays the parameters of the dialer settings.
- Authorized - Facitiy to check that a number is not blacklisted.



Settings Password, account limits, phone number verification and language

Last Name

First Name

Email Address

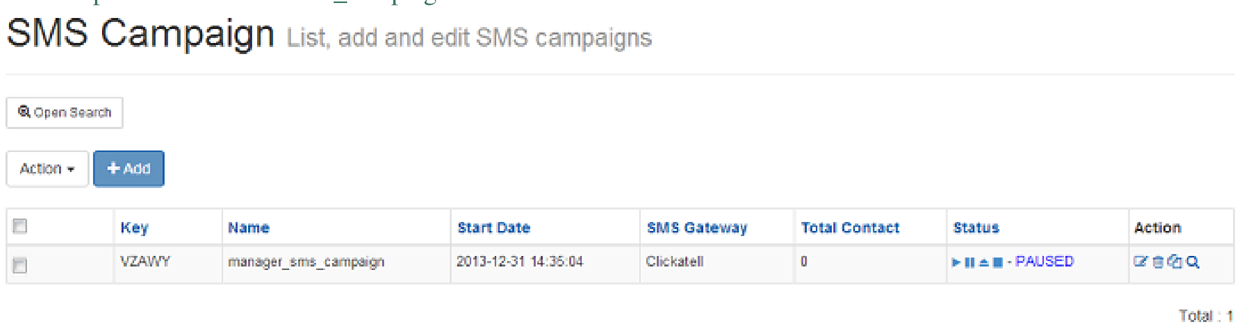
Address

7.4.8 SMSCampaign

The sms campaign list will be displayed from the following URL. You can add a new sms campaign for the logged in user by clicking `Add sms campaign`. When adding a sms campaign, it is important to add the campaign's start and end dates with time & week-day exceptions. Select the sms gateway through which smses will be routed & phonebook(s) that are linked with contacts from the sms campaign list, click on sms campaign to update.

URL:

- http://localhost:8000/sms_campaign/



SMS Campaign List, add and edit SMS campaigns

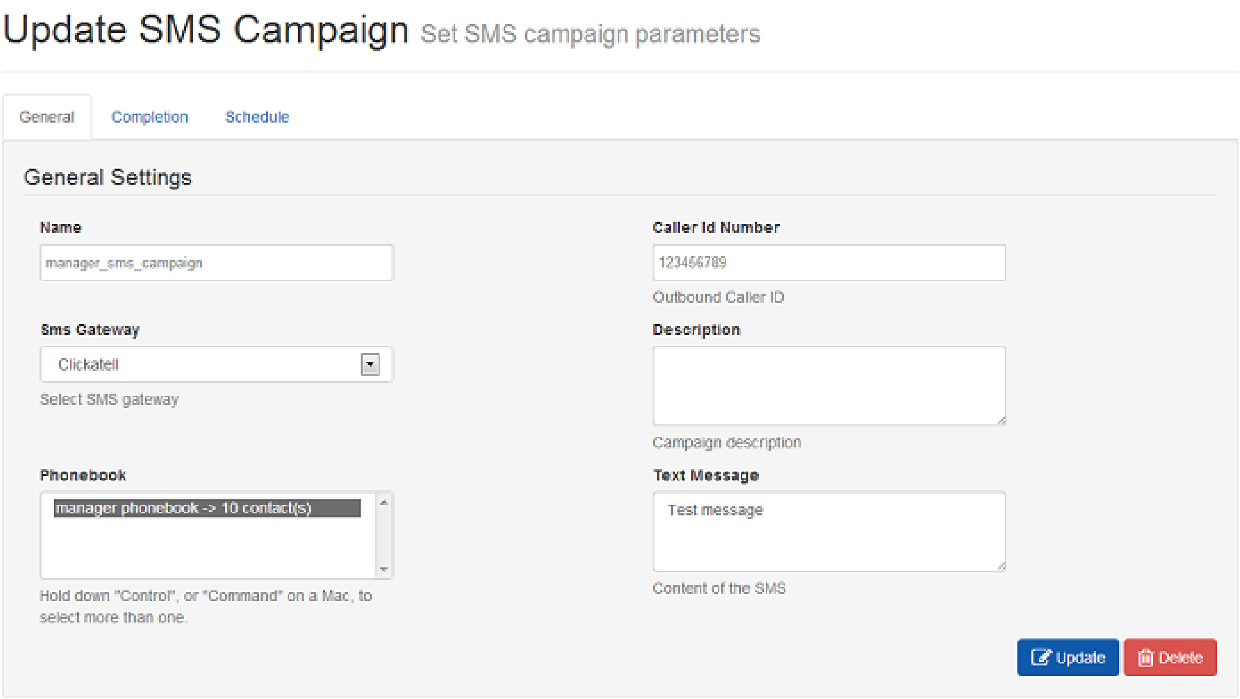
□	Key	Name	Start Date	SMS Gateway	Total Contact	Status	Action
□	VZAWY	manager_sms_campaign	2013-12-31 14:35:04	Clickatell	0	▶ ❚❚ ⏏ ■ - PAUSED	

Total : 1

To Add/Update a SMS Campaign for a logged in user

URL:

- http://localhost:8000/sms_campaign/add/
- http://localhost:8000/sms_campaign/1/



General Settings

Name: manager_sms_campaign

Caller Id Number: 123456789

Outbound Caller ID

Sms Gateway: Clickatell

Select SMS gateway

Description:

Campaign description

Phonebook: manager phonebook -> 10 contact(s)

Hold down "Control", or "Command" on a Mac, to select more than one.

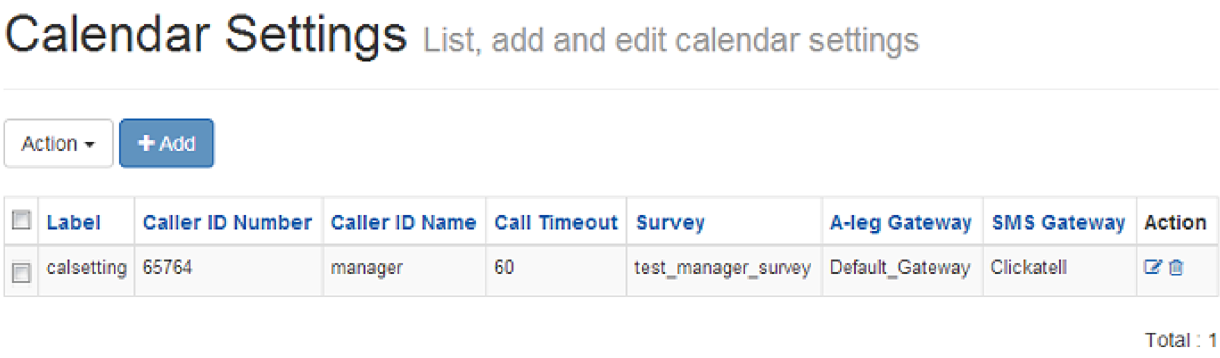
Text Message: Test message

Content of the SMS

7.4.9 Calendar Setting

URL:

- http://localhost:8000/module/calendar_setting/



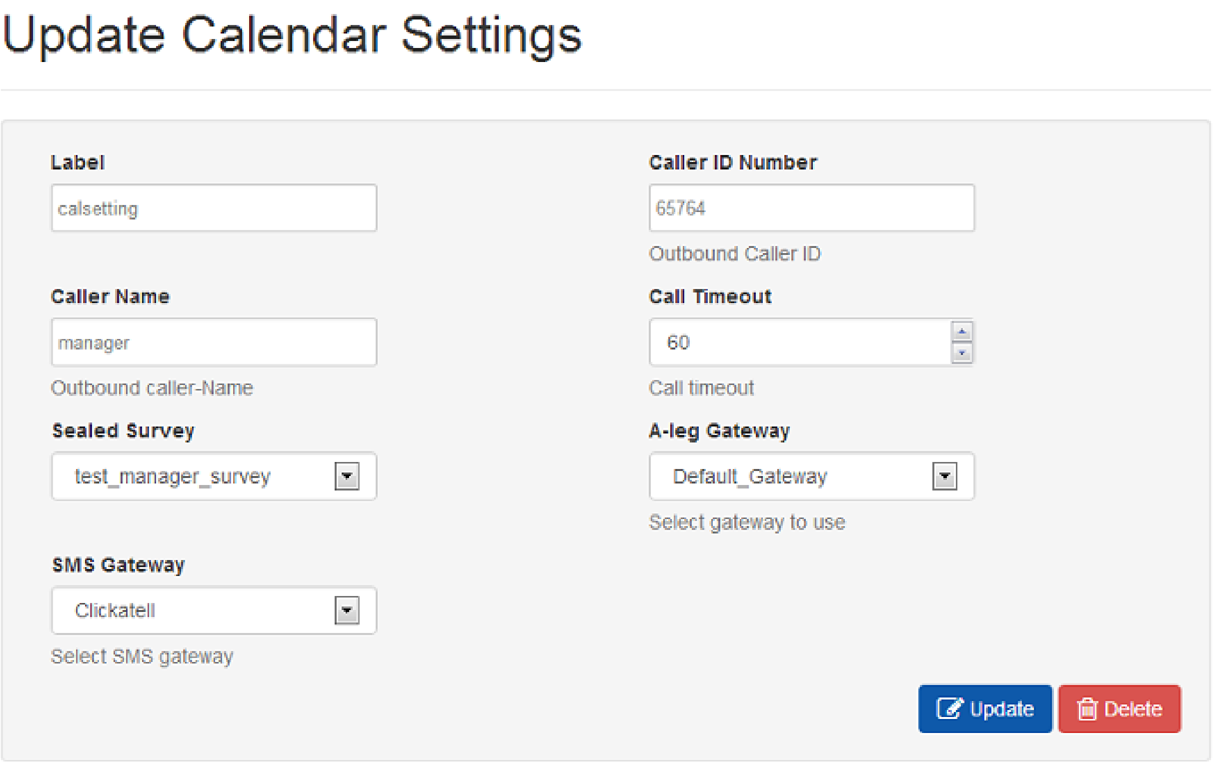
□	Label	Caller ID Number	Caller ID Name	Call Timeout	Survey	A-leg Gateway	SMS Gateway	Action
□	calsetting	65764	manager	60	test_manager_survey	Default_Gateway	Clickatell	

Total : 1

To Add/Update a calendar setting for a logged in user

URL:

- http://localhost:8000/module/calendar_setting/add/
- http://localhost:8000/module/calendar_setting/1/



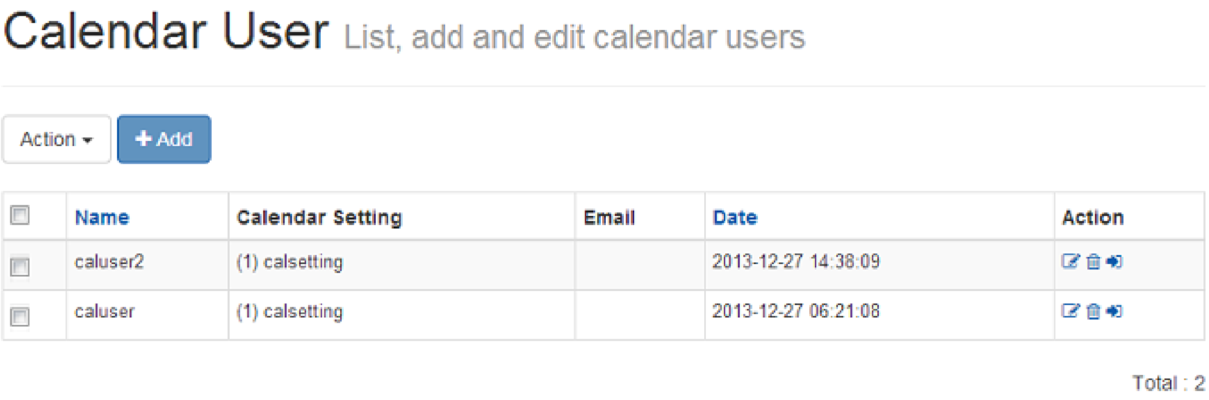
Update Calendar Settings

Label calsetting	Caller ID Number 65764 Outbound Caller ID
Caller Name manager Outbound caller-Name	Call Timeout 60 Call timeout
Sealed Survey test_manager_survey	A-leg Gateway Default_Gateway Select gateway to use
SMS Gateway Clickatell Select SMS gateway	

7.4.10 Calendar User

URL:

- http://localhost:8000/module/calendar_user/



Calendar User List, add and edit calendar users

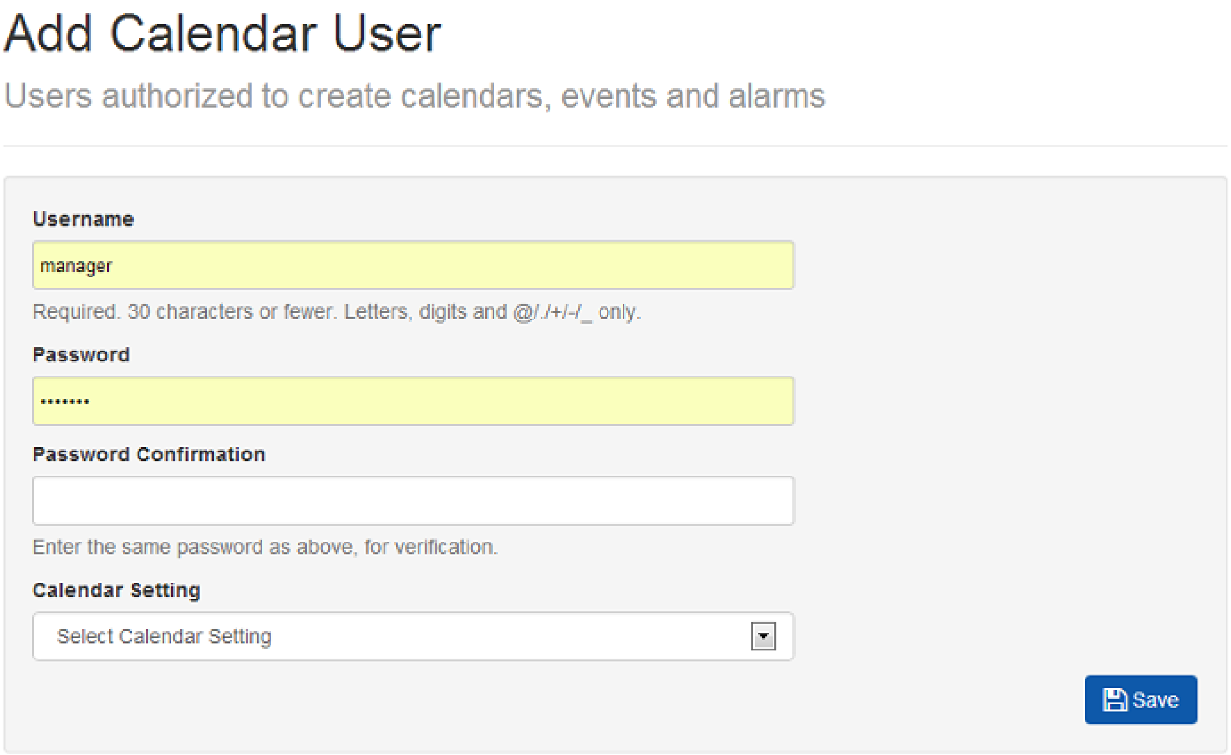
□	Name	Calendar Setting	Email	Date	Action
□	caluser2	(1) calsetting		2013-12-27 14:38:09	
□	caluser	(1) calsetting		2013-12-27 06:21:08	

Total : 2

To Add/Update a calendar user for a logged in user

URL:

- http://localhost:8000/module/calendar_user/add/
- http://localhost:8000/module/calendar_user/4/



Add Calendar User

Users authorized to create calendars, events and alarms

Username

manager

Required. 30 characters or fewer. Letters, digits and @/./+/-/_ only.

Password

•••••••

Password Confirmation

Enter the same password as above, for verification.

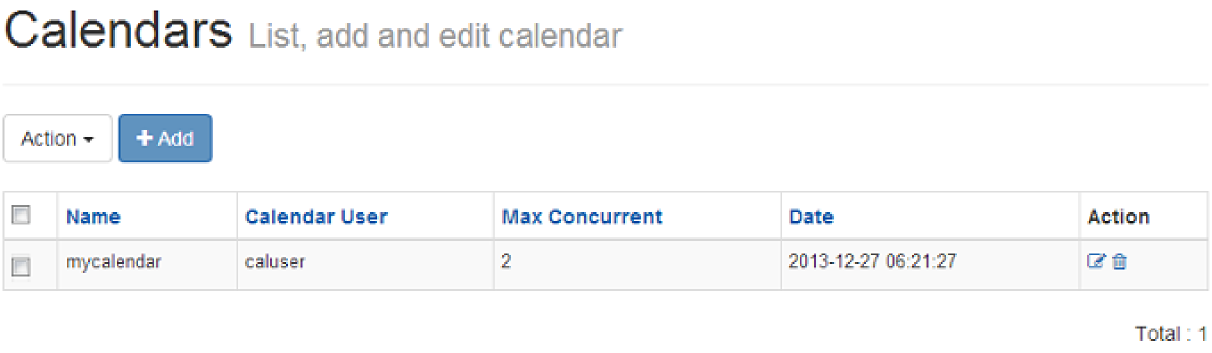
Calendar Setting

Select Calendar Setting

7.4.11 Calendar

URL:

- http://localhost:8000/module/calendar/



Calendars List, add and edit calendar

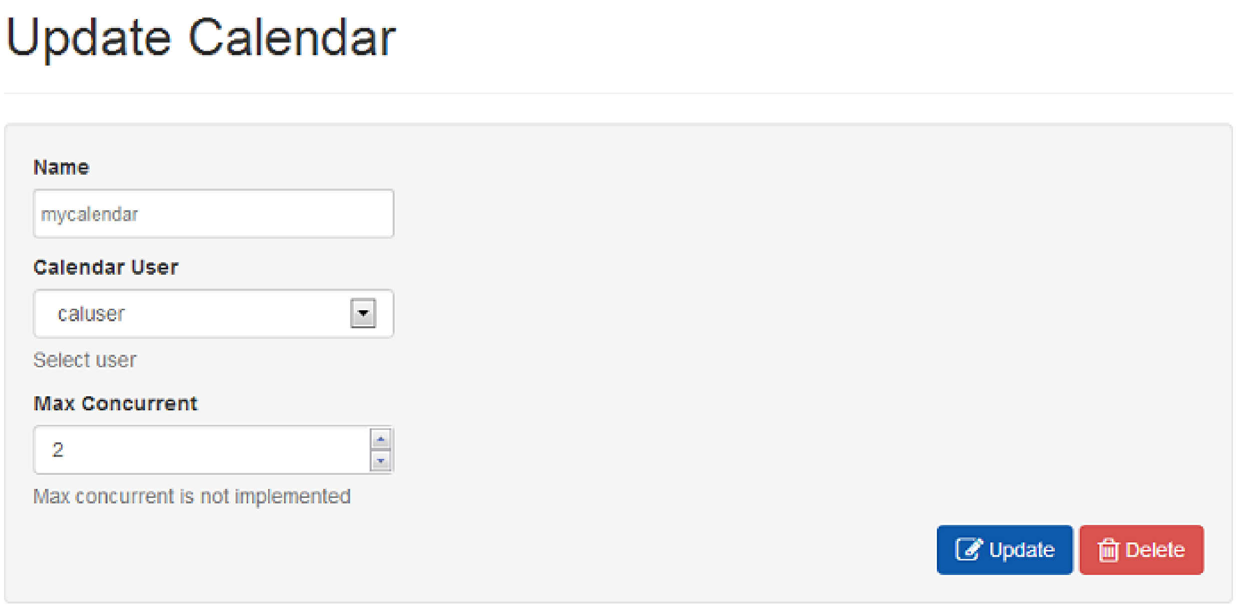
□	Name	Calendar User	Max Concurrent	Date	Action
□	mycalendar	caluser	2	2013-12-27 06:21:27	✎ 🗑

Total : 1

To Add/Update a calendar for a logged in user

URL:

- http://localhost:8000/module/calendar/add/
- http://localhost:8000/module/calendar/4/



Update Calendar

Name

mycalendar

Calendar User

caluser

Select user

Max Concurrent

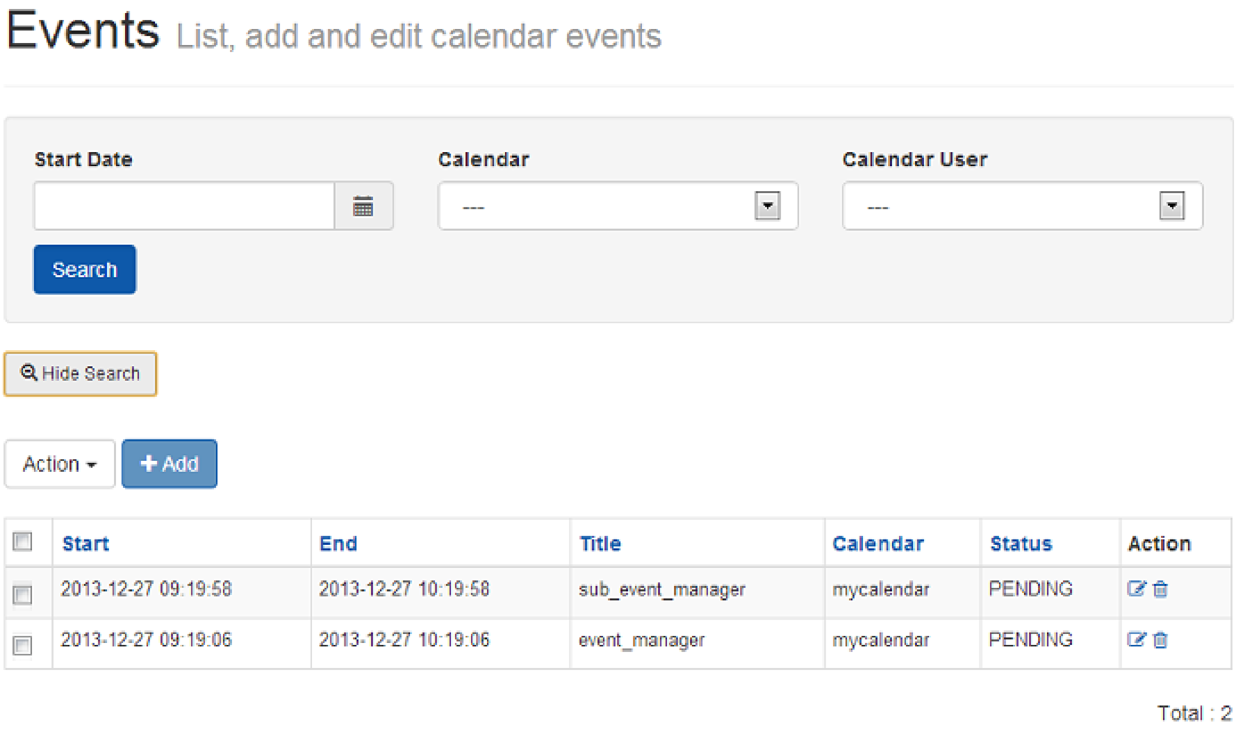
2

Max concurrent is not implemented

7.4.12 Event

URL:

- http://localhost:8000/module/event/



Events List, add and edit calendar events

Start Date

Calendar

Calendar User

□	Start	End	Title	Calendar	Status	Action
□	2013-12-27 09:19:58	2013-12-27 10:19:58	sub_event_manager	mycalendar	PENDING	
□	2013-12-27 09:19:06	2013-12-27 10:19:06	event_manager	mycalendar	PENDING	

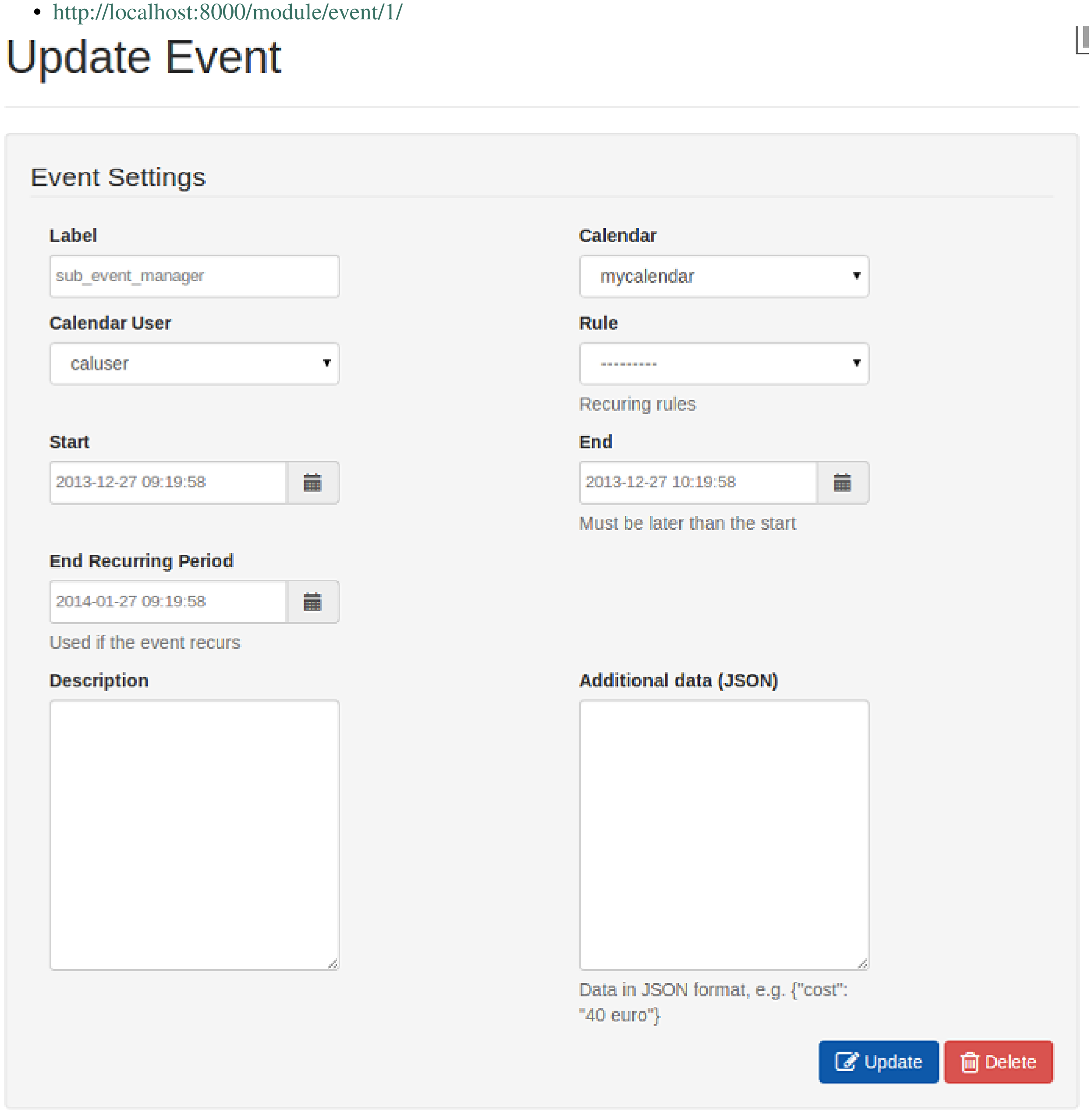
Total : 2

To Add/Update a event for a logged in user

URL:

- http://localhost:8000/module/event/add/

- http://localhost:8000/module/event/1/



Update Event

Event Settings

Label: sub_event_manager

Calendar: mycalendar

Calendar User: caluser

Rule: ---------

Recuring rules

Start: 2013-12-27 09:19:58

End: 2013-12-27 10:19:58

Must be later than the start

End Recurring Period: 2014-01-27 09:19:58

Used if the event recurs

Description

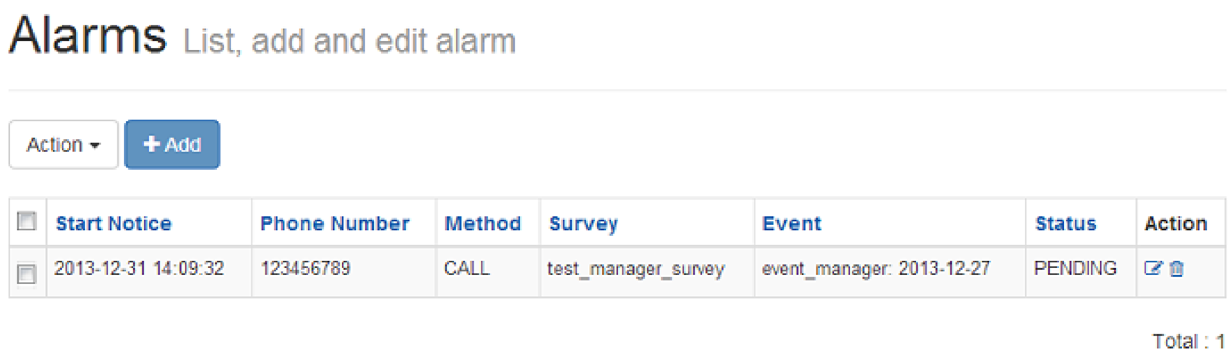
Additional data (JSON)

Data in JSON format, e.g. {"cost": "40 euro"}

7.4.13 Alarm

URL:

- http://localhost:8000/module/alarm/



Alarms List, add and edit alarm

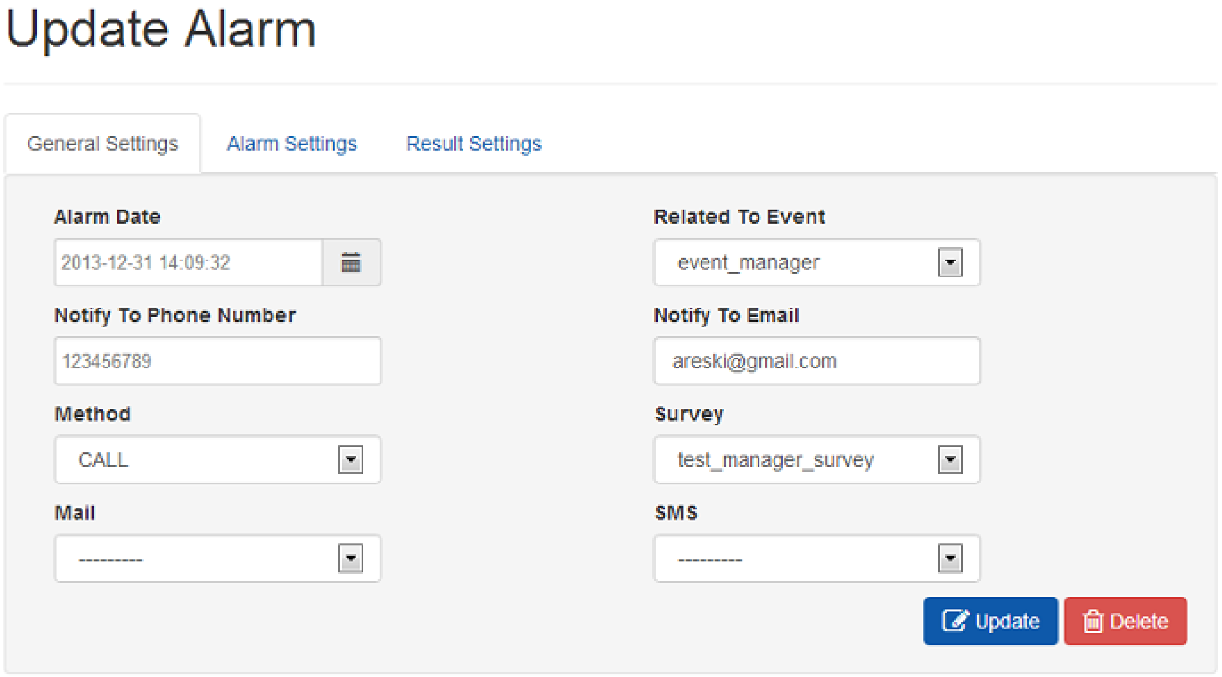
□	Start Notice	Phone Number	Method	Survey	Event	Status	Action
□	2013-12-31 14:09:32	123456789	CALL	test_manager_survey	event_manager: 2013-12-27	PENDING	✎ 🗑

Total : 1

To Add/Update a alarm for a logged in user

URL:

- http://localhost:8000/module/alarm/add/
- http://localhost:8000/module/alarm/1/



Update Alarm

Alarm Date: 2013-12-31 14:09:32

Related To Event: event_manager

Notify To Phone Number: 123456789

Notify To Email: areski@gmail.com

Method: CALL

Survey: test_manager_survey

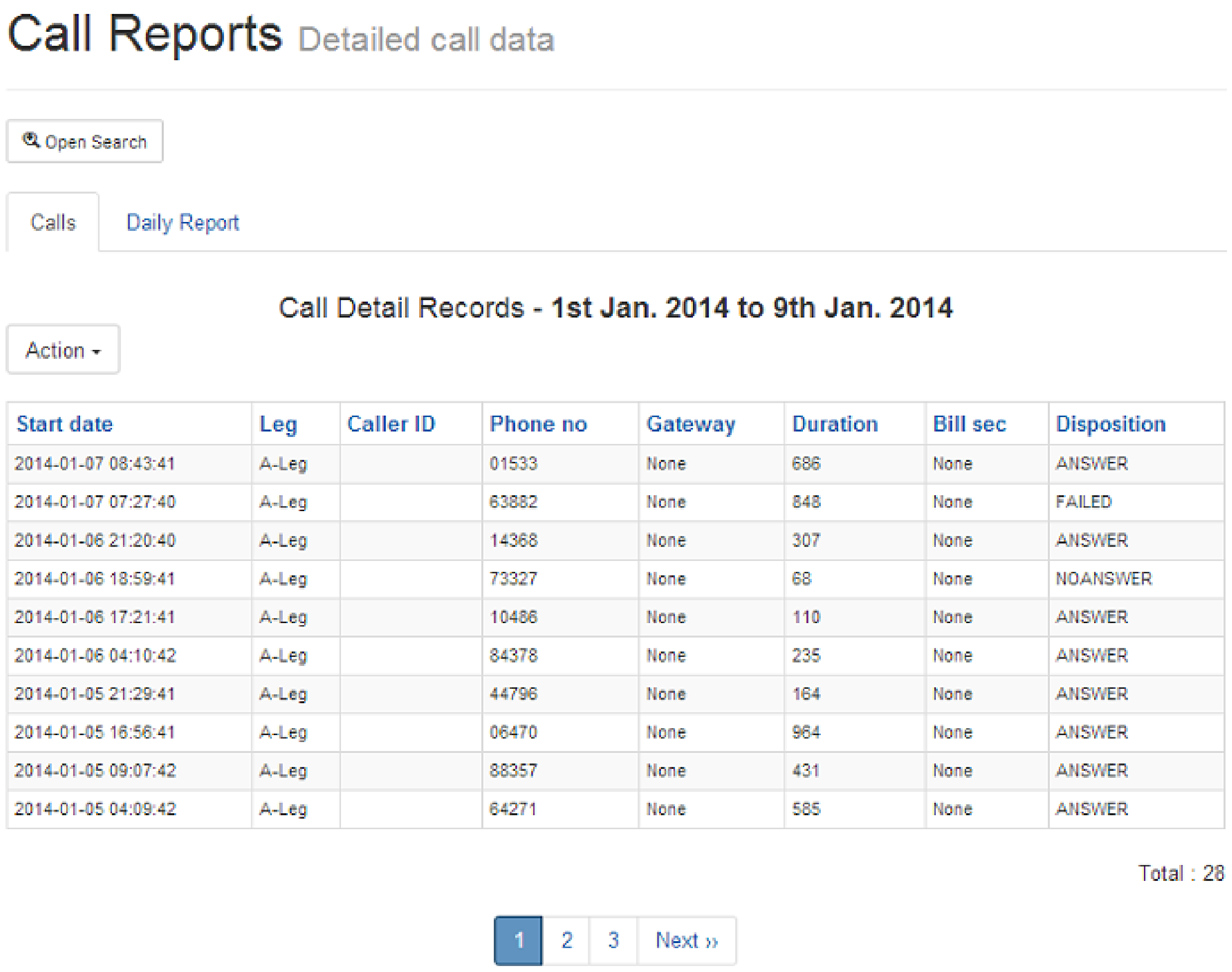
Mail: ---------

SMS: ---------

7.4.14 Reports

URL:

- http://localhost:8000/voipcall_report/

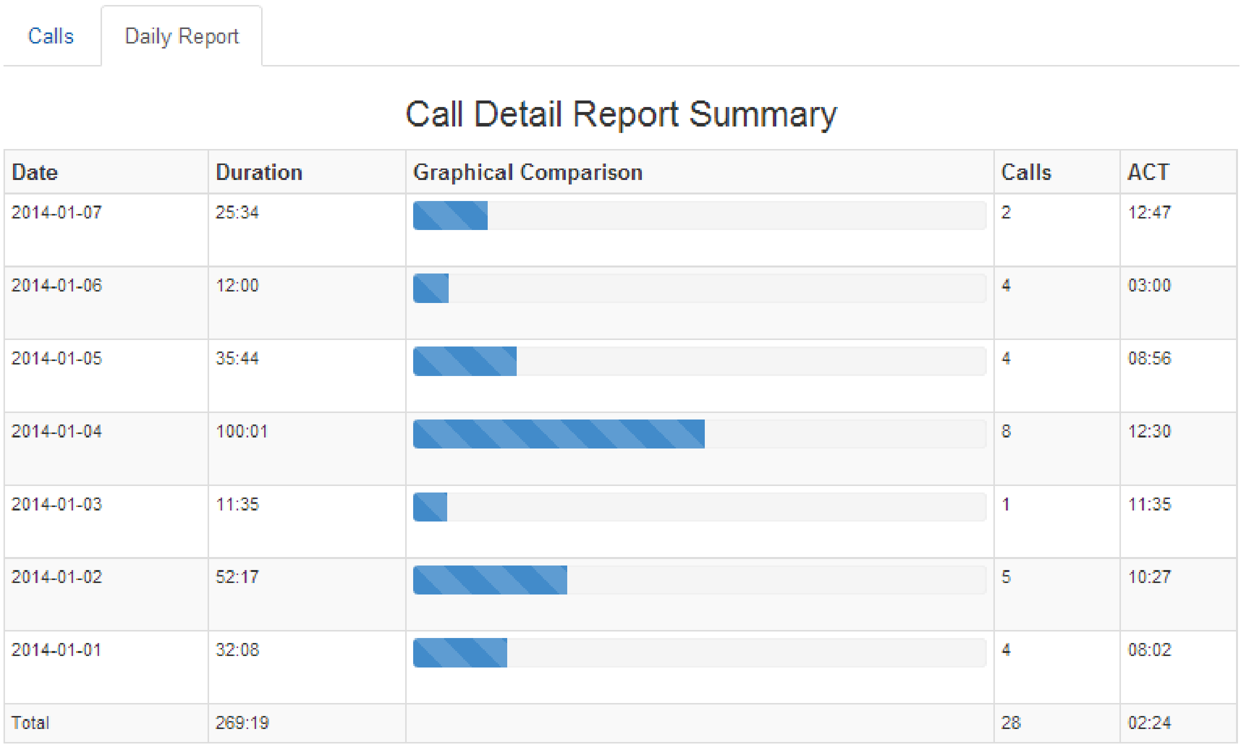


Call Reports Detailed call data

Call Detail Records - 1st Jan. 2014 to 9th Jan. 2014

Start date	Leg	Caller ID	Phone no	Gateway	Duration	Bill sec	Disposition
2014-01-07 08:43:41	A-Leg		01533	None	686	None	ANSWER
2014-01-07 07:27:40	A-Leg		63882	None	848	None	FAILED
2014-01-06 21:20:40	A-Leg		14368	None	307	None	ANSWER
2014-01-06 18:59:41	A-Leg		73327	None	68	None	NOANSWER
2014-01-06 17:21:41	A-Leg		10486	None	110	None	ANSWER
2014-01-06 04:10:42	A-Leg		84378	None	235	None	ANSWER
2014-01-05 21:29:41	A-Leg		44796	None	164	None	ANSWER
2014-01-05 16:56:41	A-Leg		06470	None	964	None	ANSWER
2014-01-05 09:07:42	A-Leg		88357	None	431	None	ANSWER
2014-01-05 04:09:42	A-Leg		64271	None	585	None	ANSWER

Total : 28

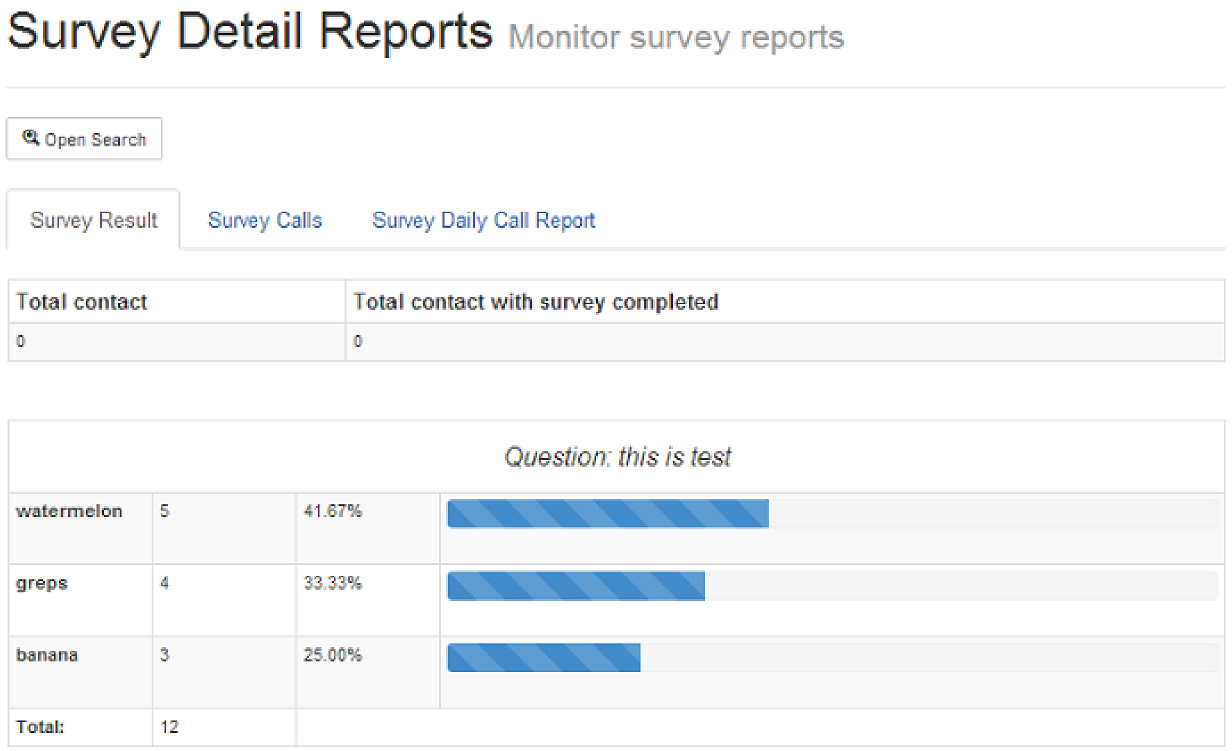


Call Detail Report Summary

Date	Duration	Graphical Comparison	Calls	ACT
2014-01-07	25:34		2	12:47
2014-01-06	12:00		4	03:00
2014-01-05	35:44		4	08:56
2014-01-04	100:01		8	12:30
2014-01-03	11:35		1	11:35
2014-01-02	52:17		5	10:27
2014-01-01	32:08		4	08:02
Total	269:19		28	02:24

URL:

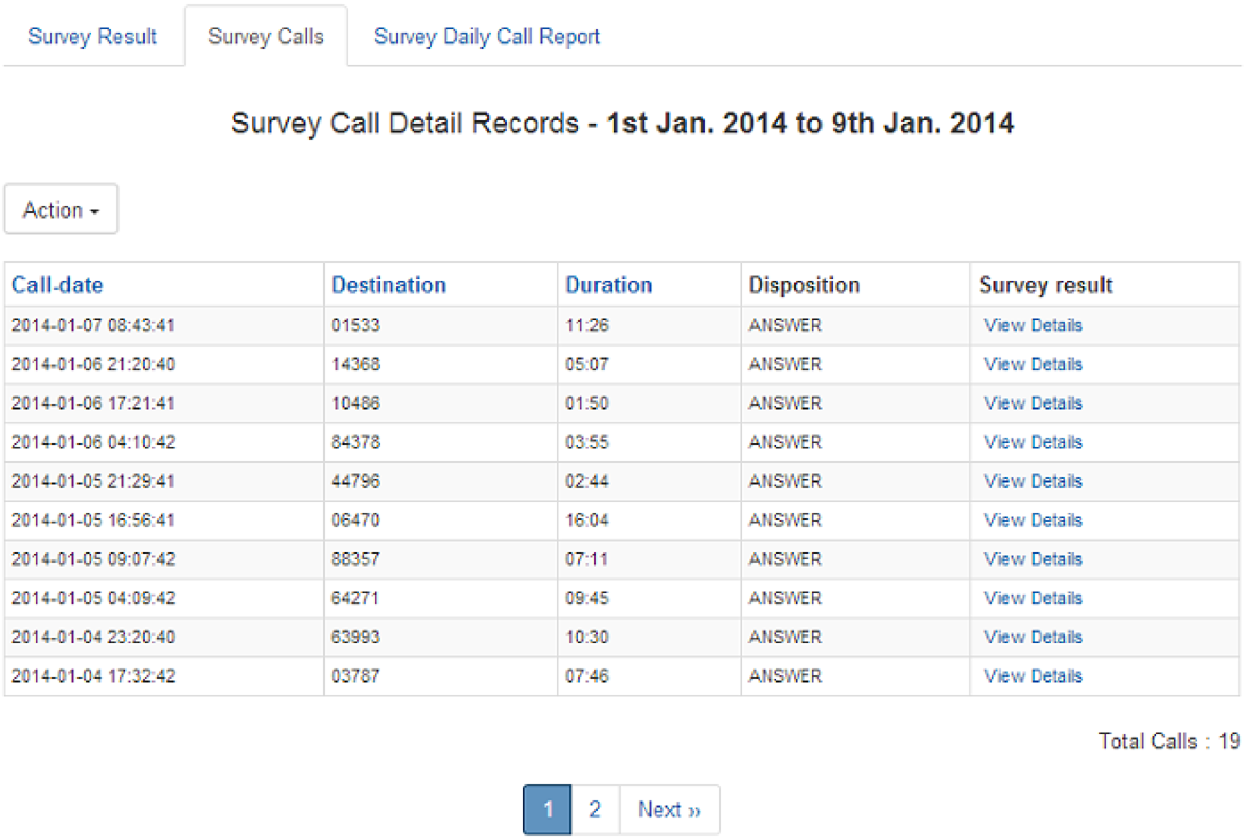
- http://localhost:8000/survey_report/


Survey Detail Reports Monitor survey reports

Total contact	Total contact with survey completed
0	0

Question: this is test

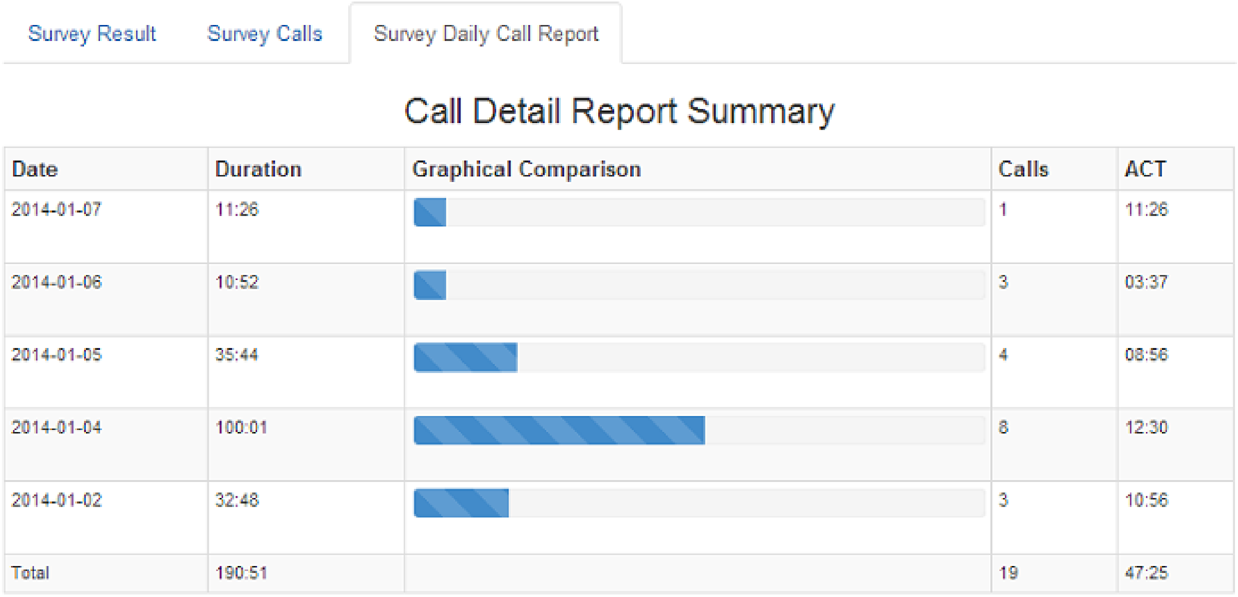
watermelon	5	41.67%	
greps	4	33.33%	
banana	3	25.00%	
Total:	12		



Survey Call Detail Records - 1st Jan. 2014 to 9th Jan. 2014

Call-date	Destination	Duration	Disposition	Survey result
2014-01-07 08:43:41	01533	11:26	ANSWER	View Details
2014-01-06 21:20:40	14368	05:07	ANSWER	View Details
2014-01-06 17:21:41	10486	01:50	ANSWER	View Details
2014-01-06 04:10:42	84378	03:55	ANSWER	View Details
2014-01-05 21:29:41	44796	02:44	ANSWER	View Details
2014-01-05 16:56:41	06470	16:04	ANSWER	View Details
2014-01-05 09:07:42	88357	07:11	ANSWER	View Details
2014-01-05 04:09:42	64271	09:45	ANSWER	View Details
2014-01-04 23:20:40	63993	10:30	ANSWER	View Details
2014-01-04 17:32:42	03787	07:46	ANSWER	View Details

Total Calls : 19

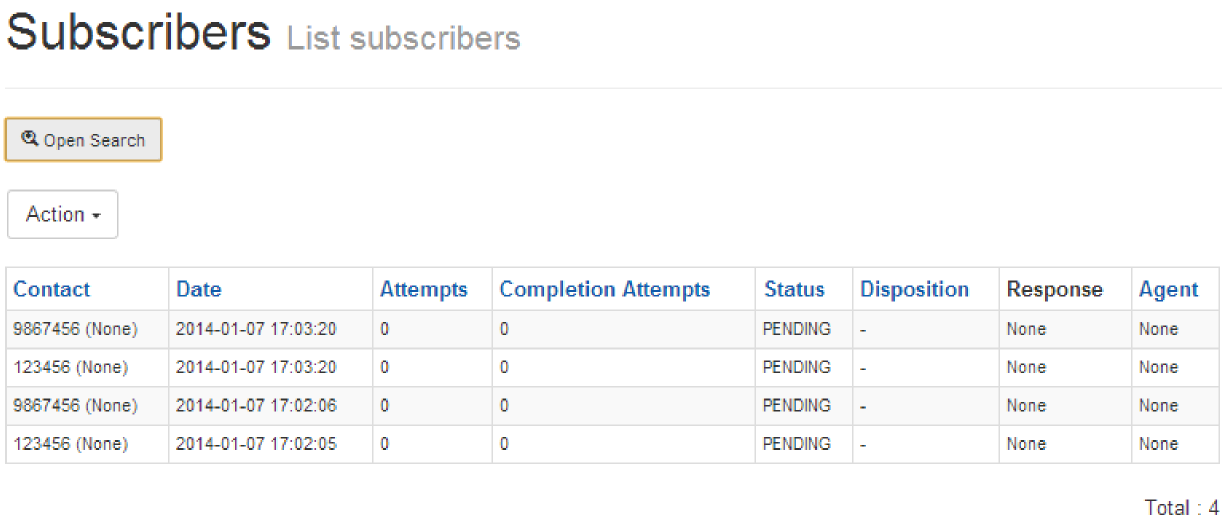


Call Detail Report Summary

Date	Duration	Graphical Comparison	Calls	ACT
2014-01-07	11:26		1	11:26
2014-01-06	10:52		3	03:37
2014-01-05	35:44		4	08:56
2014-01-04	100:01		8	12:30
2014-01-02	32:48		3	10:56
Total	190:51		19	47:25

URL:

- http://localhost:8000/subscribers/



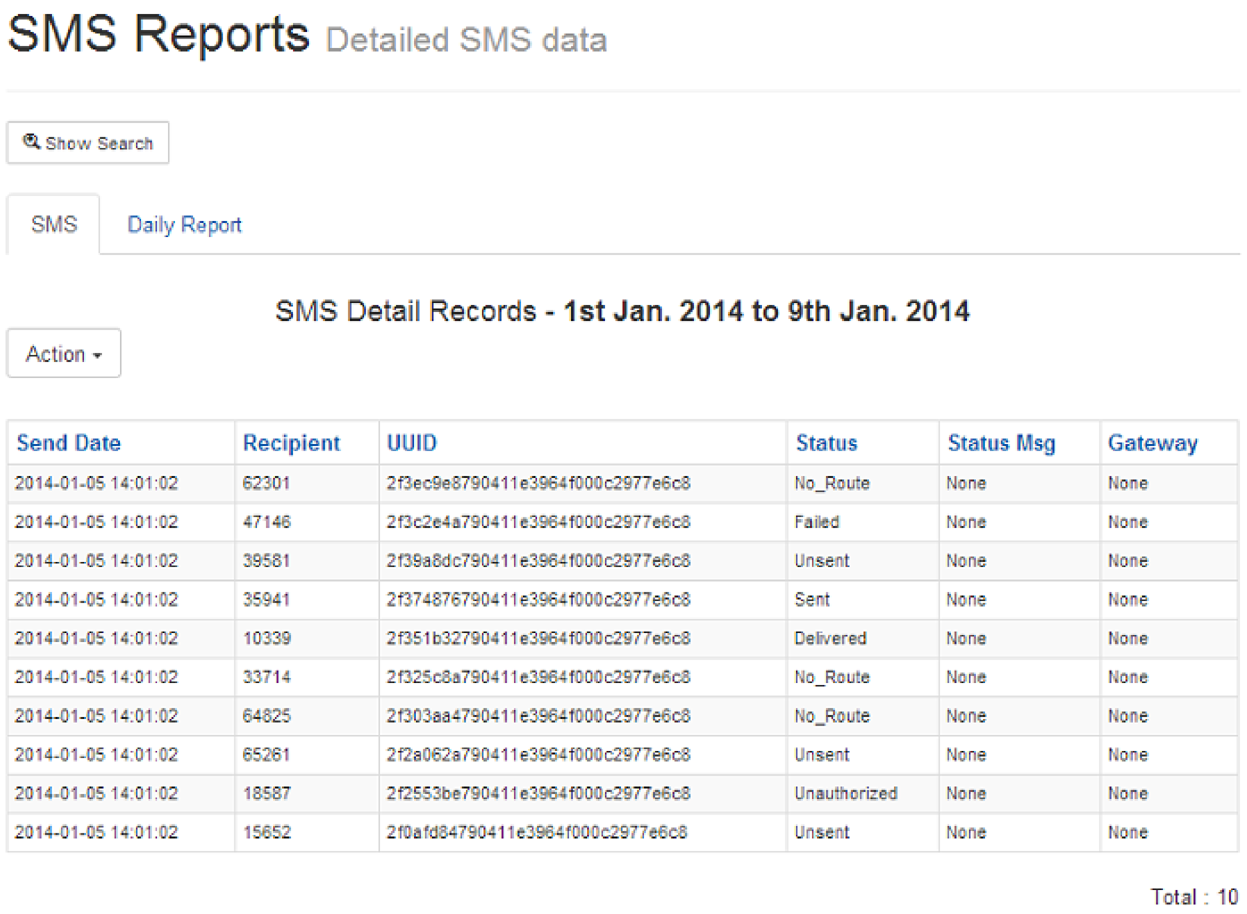
Subscribers List subscribers

Contact	Date	Attempts	Completion Attempts	Status	Disposition	Response	Agent
9867456 (None)	2014-01-07 17:03:20	0	0	PENDING	-	None	None
123456 (None)	2014-01-07 17:03:20	0	0	PENDING	-	None	None
9867456 (None)	2014-01-07 17:02:06	0	0	PENDING	-	None	None
123456 (None)	2014-01-07 17:02:05	0	0	PENDING	-	None	None

Total : 4

URL:

- http://localhost:8000/sms_report/

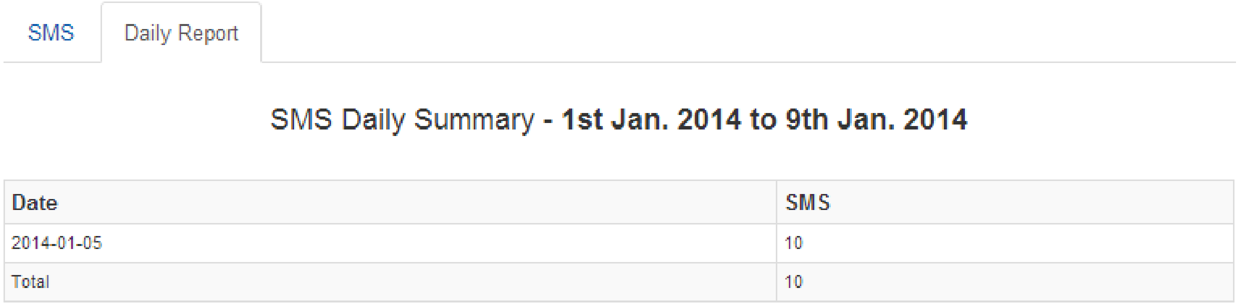


SMS Reports Detailed SMS data

SMS Detail Records - 1st Jan. 2014 to 9th Jan. 2014

Send Date	Recipient	UUID	Status	Status Msg	Gateway
2014-01-05 14:01:02	62301	2f3ec9e8790411e3964f000c2977e6c8	No_Route	None	None
2014-01-05 14:01:02	47146	2f3c2e4a790411e3964f000c2977e6c8	Failed	None	None
2014-01-05 14:01:02	39581	2f39a8dc790411e3964f000c2977e6c8	Unsent	None	None
2014-01-05 14:01:02	35941	2f374876790411e3964f000c2977e6c8	Sent	None	None
2014-01-05 14:01:02	10339	2f351b32790411e3964f000c2977e6c8	Delivered	None	None
2014-01-05 14:01:02	33714	2f325c8a790411e3964f000c2977e6c8	No_Route	None	None
2014-01-05 14:01:02	64825	2f303aa4790411e3964f000c2977e6c8	No_Route	None	None
2014-01-05 14:01:02	65261	2f2a062a790411e3964f000c2977e6c8	Unsent	None	None
2014-01-05 14:01:02	18587	2f2553be790411e3964f000c2977e6c8	Unauthorized	None	None
2014-01-05 14:01:02	15652	2f0afd84790411e3964f000c2977e6c8	Unsent	None	None

Total : 10



SMS Daily Summary - **1st Jan. 2014 to 9th Jan. 2014**

Date	SMS
2014-01-05	10
Total	10

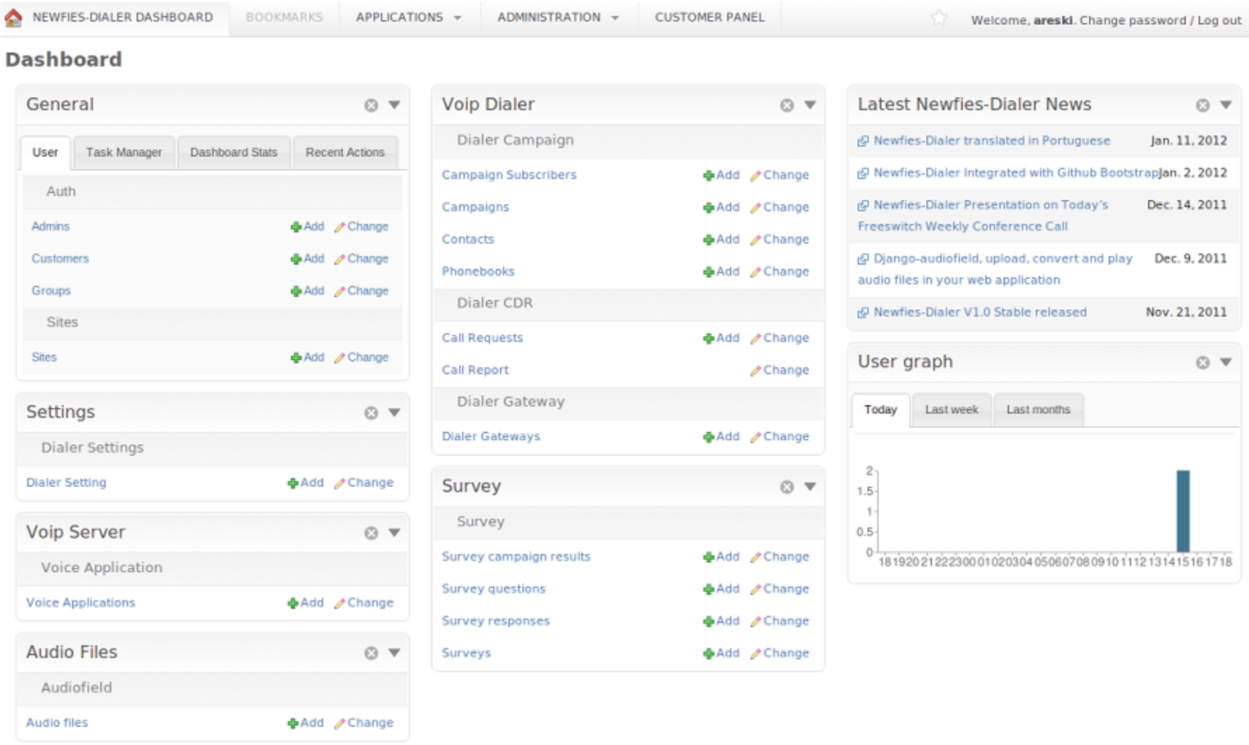
7.5 Admin Panel

Admin URL : http://localhost:8000/admin/

This interface provides user (ACL) management, full control of all Campaigns, Phonebooks, Subscribers, Gateways and configuration of the Audio Application.

7.5.1 Dashboard

Dashboard page for the admin interface after successful login with superuser credentials



Newfies-Dialer translated in Portuguese Jan. 11, 2012

Newfies-Dialer Integrated with Github Bootstrap Jan. 2, 2012

Newfies-Dialer Presentation on Today's Freeswitch Weekly Conference Call Dec. 14, 2011

Django-audiofield, upload, convert and play audio files in your web application Dec. 9, 2011

Newfies-Dialer V1.0 Stable released Nov. 21, 2011

7.5.2 Admin User Management

The Admin section allows you to create administrators who have access the admin screens. Levels of access can be set. If the administrator is to be allowed to make calls, then a gateway and dialer- settings must be set for that administrator.

7.5.3 Customers

Customers only have access to the customer panel. Before a customer can make calls, they must have both dialer-settings and a gateway attached to their account.

7.5.4 Sites

Some of the features of Newfies-Dialer are dependent on the hostname or IP address, so this has to be set correctly in site address.

In the Admin dashboard, locate the "Sites" link and click change. By default, there will be an entry of "example.com". Edit this setting to reflect the hostname of the server, or if you do not have this set up, the IP address. e.g. http://www.domain.tld:8008 or in the case of an IP address, http://192.168.1.200:8008

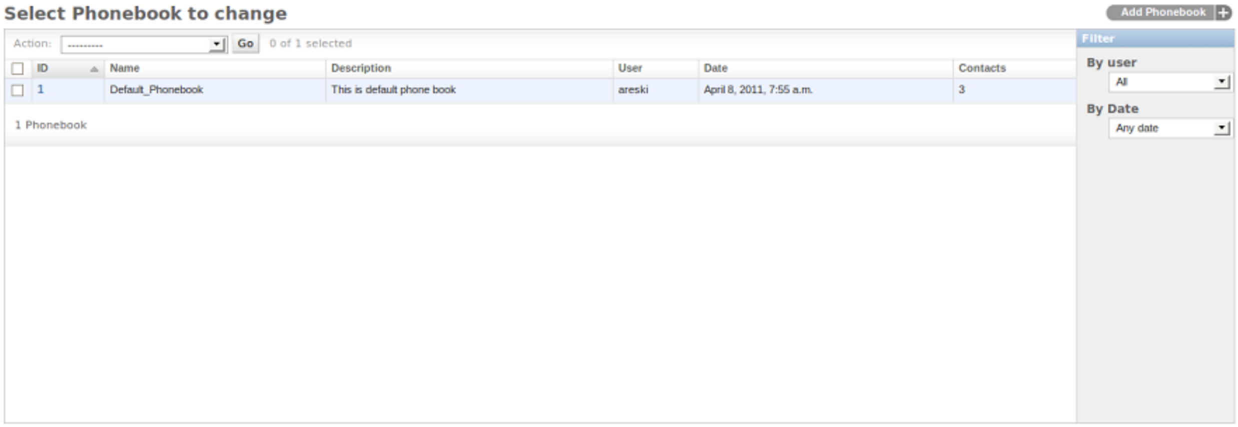
- http://localhost:8000/admin/sites/

7.5.5 Phonebook

The phonebook list will be displayed from the following URL. You can add a new phonebook by clicking `Add phonebook` and adding the name of the phonebook and its description, Also from the phonebook list, click on the phonebook that you want to update.

URL:

- http://localhost:8000/admin/dialer_campaign/phonebook/



Select Phonebook to change

Action: --------- Go 0 of 1 selected

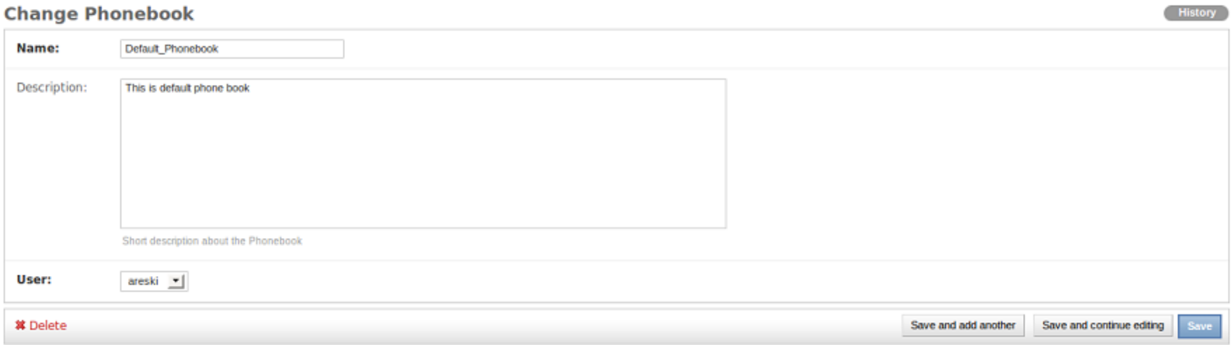
□	ID	Name	Description	User	Date	Contacts
□	1	Default_Phonebook	This is default phone book	areski	April 8, 2011, 7:55 a.m.	3

1 Phonebook

To Add/Update phonebook for a user

URL:

- http://localhost:8000/admin/dialer_campaign/phonebook/add/
- http://localhost:8000/admin/dialer_campaign/phonebook/1/



Change Phonebook

Name: Default_Phonebook

Description: This is default phone book

Short description about the Phonebook

User: areski

7.5.6 Contact

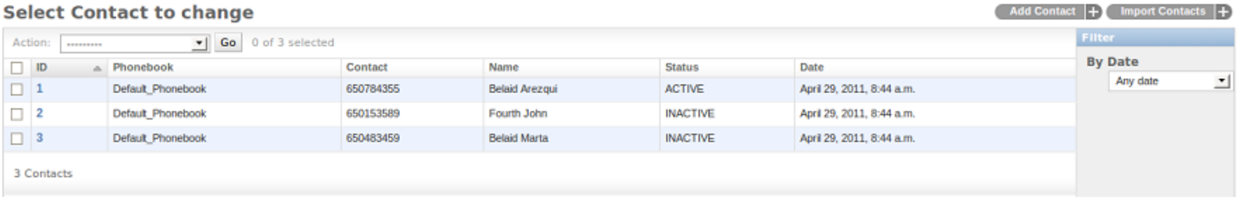
The contact list will be displayed from the following URL and you can add a new contact by clicking `Add contact` & adding the contact details (i.e. phone number, name, description about contact, contact status) to one phonebook from the phonebook list.

If the contact is active and the linked phonebook is also attached to a running campaign, then the contact will be added into subscriber.

From the contact list, click on the contact that you want to update.

URL:

- http://localhost:8000/admin/dialer_campaign/contact/



Select Contact to change

Action: --------- Go 0 of 3 selected

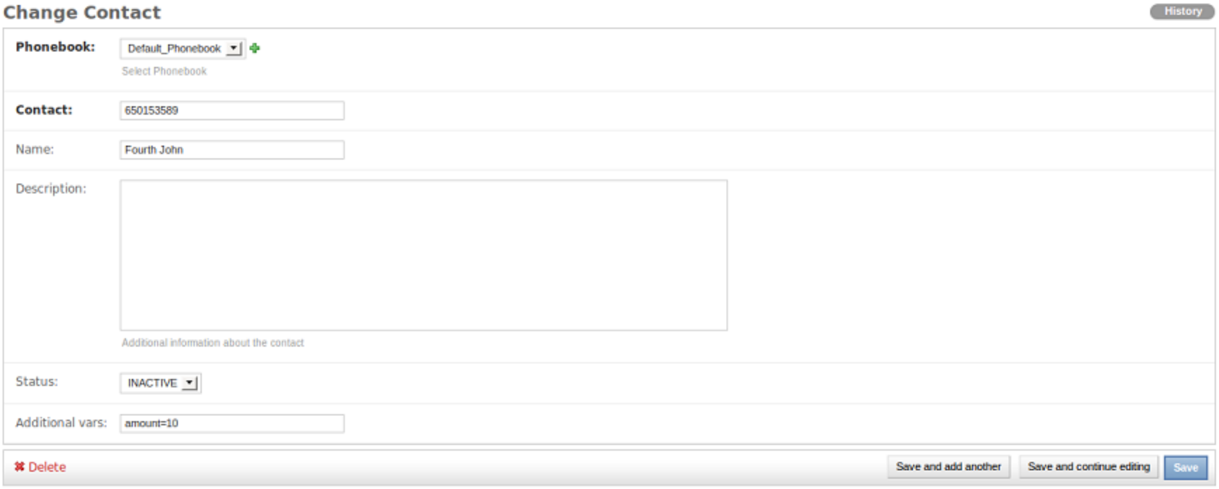
□	ID	Phonebook	Contact	Name	Status	Date
□	1	Default_Phonebook	650784355	Belaid Arezqui	ACTIVE	April 29, 2011, 8:44 a.m.
□	2	Default_Phonebook	650153589	Fourth John	INACTIVE	April 29, 2011, 8:44 a.m.
□	3	Default_Phonebook	650483459	Belaid Marta	INACTIVE	April 29, 2011, 8:44 a.m.

3 Contacts

To Add/Update a contact

URL:

- http://localhost:8000/admin/dialer_campaign/contact/add/
- http://localhost:8000/admin/dialer_campaign/contact/1/



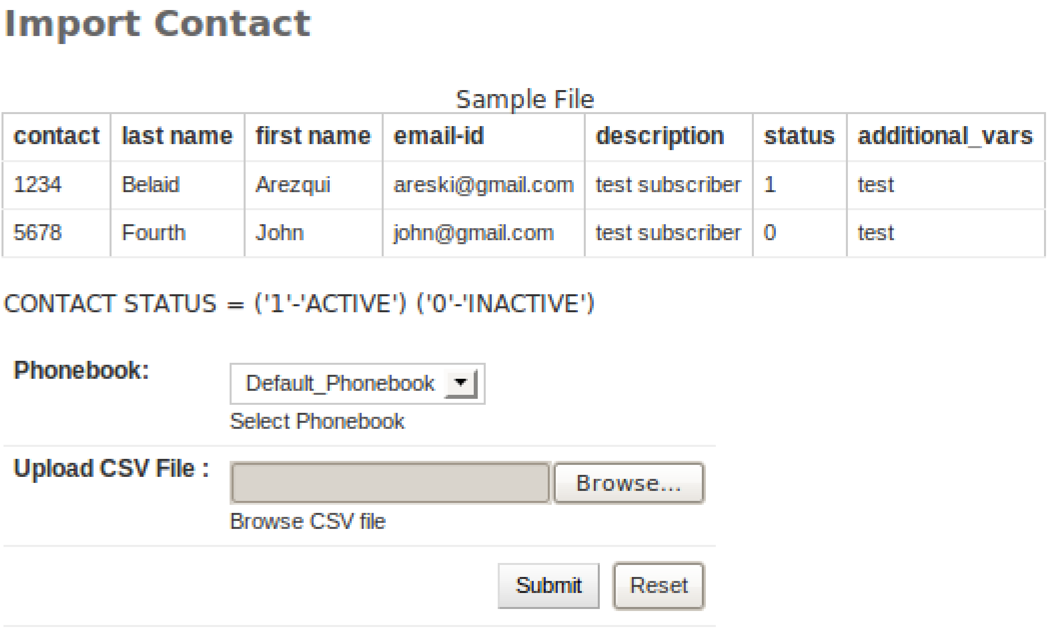
Change Contact

Phonebook:	Default_Phonebook Select Phonebook
Contact:	650153589
Name:	Fourth John
Description:	 Additional information about the contact
Status:	INACTIVE
Additional vars:	amount=10

To import bulk contacts into a phonebook, click on `Import contacts`. where you can upload the contacts via a CSV file in to one phonebook.

URL:

- http://localhost:8000/admin/dialer_campaign/contact/import_contact/



Import Contact

Sample File

contact	last name	first name	email-id	description	status	additional_vars
1234	Belaid	Arezqui	areski@gmail.com	test subscriber	1	test
5678	Fourth	John	john@gmail.com	test subscriber	0	test

CONTACT STATUS = ('1'-'ACTIVE') ('0'-'INACTIVE')

Phonebook:	Default_Phonebook Select Phonebook
Upload CSV File :	Browse CSV file

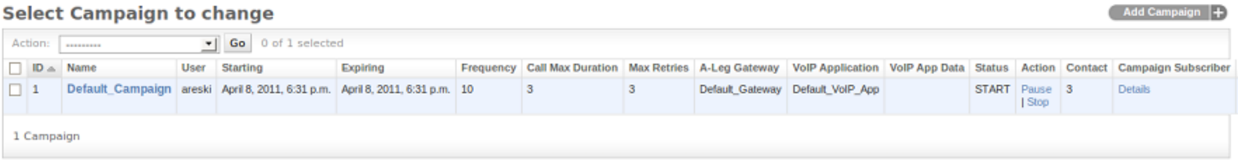
7.5.7 Campaign

The campaign list will be displayed from the following URL. You can add a new campaign by clicking `Add campaign`. While adding a campaign, it is important to add campaign's start and end dates with time & week-day exceptions. Also select the gateway through which calls will be routed & the phonebook(s) linked with contacts.

From the campaign list, click on the campaign that you want to update.

URL:

- http://localhost:8000/admin/dialer_campaign/campaign/



Select Campaign to change

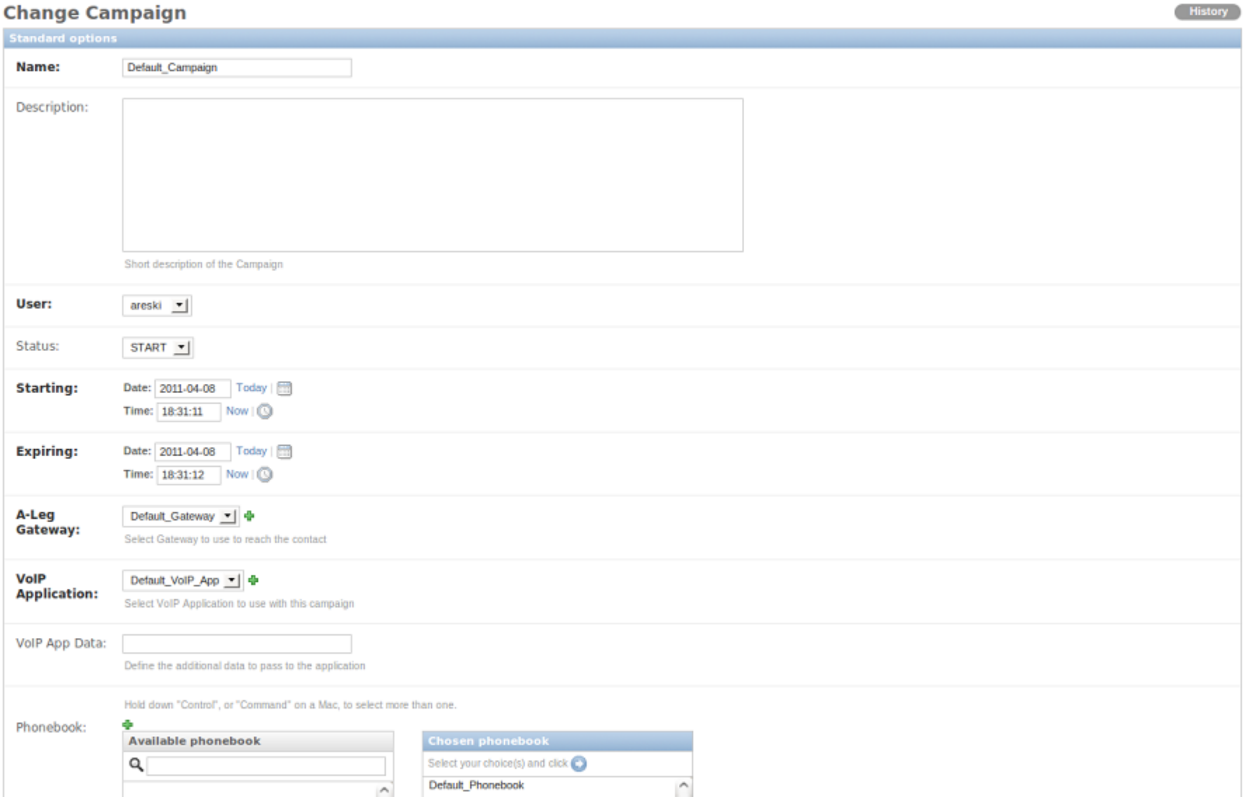
□	ID	Name	User	Starting	Expiring	Frequency	Call Max Duration	Max Retries	A-Leg Gateway	VoIP Application	VoIP App Data	Status	Action	Contact	Campaign Subscriber
□	1	Default_Campaign	areski	April 8, 2011, 6:31 p.m.	April 8, 2011, 6:31 p.m.	10	3	3	Default_Gateway	Default_VoIP_App		START	Pause | Stop	3	Details

1 Campaign

To Add/Update Campaign for user

URL:

- http://localhost:8000/admin/dialer_campaign/campaign/add/
- http://localhost:8000/admin/dialer_campaign/campaign/1/



Change Campaign

Name: Default_Campaign

Description:

Short description of the Campaign

User: areski

Status: START

Starting: Date: 2011-04-08 Today | Time: 18:31:11 Now |

Expiring: Date: 2011-04-08 Today | Time: 18:31:12 Now |

A-Leg Gateway: Default_Gateway

Select Gateway to use to reach the contact

VoIP Application: Default_VoIP_App

Select VoIP Application to use with this campaign

VoIP App Data:

Define the additional data to pass to the application

Phonebook:

Hold down "Control", or "Command" on a Mac, to select more than one.

Default_Phonebook

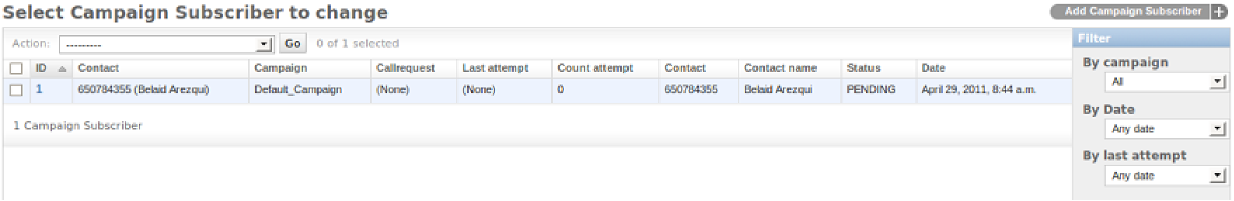
7.5.8 Subscriber

The Subscriber list will be displayed from the following URL. You can add a new subscriber by clicking `Add Subscriber`. Also from the campaign subscriber list, click on the subscriber to update.

While creating a contact, if its linked phonebook is also attached to a running campaign, then the contact will be added into the subscriber.

URL:

- http://localhost:8000/admin/dialer_campaign/campaignsubscriber/



Select Campaign Subscriber to change

Action: --------- 0 of 1 selected

□	ID	Contact	Campaign	Callrequest	Last attempt	Count attempt	Contact	Contact name	Status	Date
□	1	650784355 (Belaid Arezqui)	Default_Campaign	(None)	(None)	0	650784355	Belaid Arezqui	PENDING	April 29, 2011, 8:44 a.m.

1 Campaign Subscriber

By campaign: All

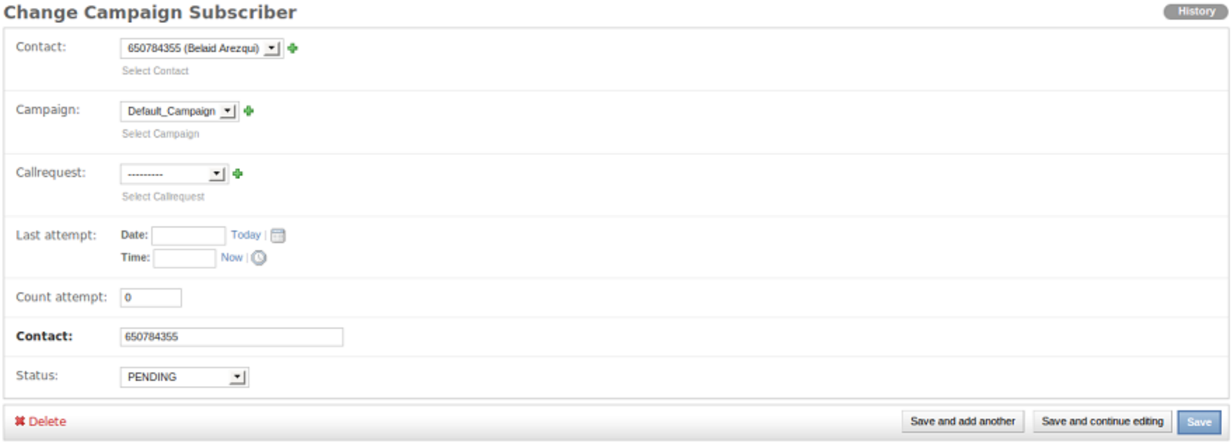
By Date: Any date

By last attempt: Any date

To Add/Update Subscriber

URL:

- http://localhost:8000/admin/dialer_campaign/campaignsubscriber/add/
- http://localhost:8000/admin/dialer_campaign/campaignsubscriber/1/



Change Campaign Subscriber

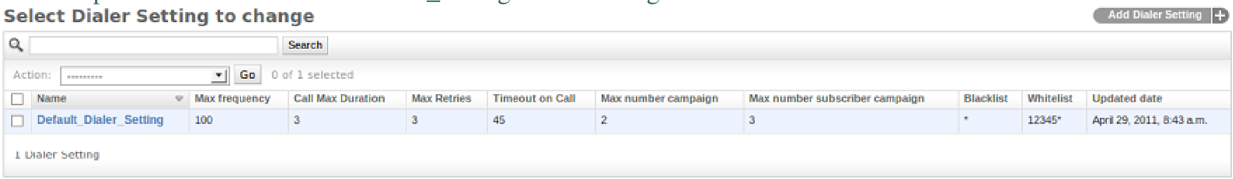
Contact:	650784355 (Belaid Arezqui) Select Contact
Campaign:	Default_Campaign Select Campaign
Callrequest:	--------- Select Callrequest
Last attempt:	Date: Time:
Count attempt:	0
Contact:	650784355
Status:	PENDING

7.5.9 Dialer Settings

The dialer settings list will be displayed from the following URL. The Dialer settings list is applied to a system User. You can add a new setting by clicking `Add Dialer Settings` and add numeric values for the limit. Also from the dialer settings list, click on the setting to update.

URL:

- http://localhost:8000/admin/dialer_settings/dialersetting/



Select Dialer Setting to change

Action: --------- 0 of 1 selected

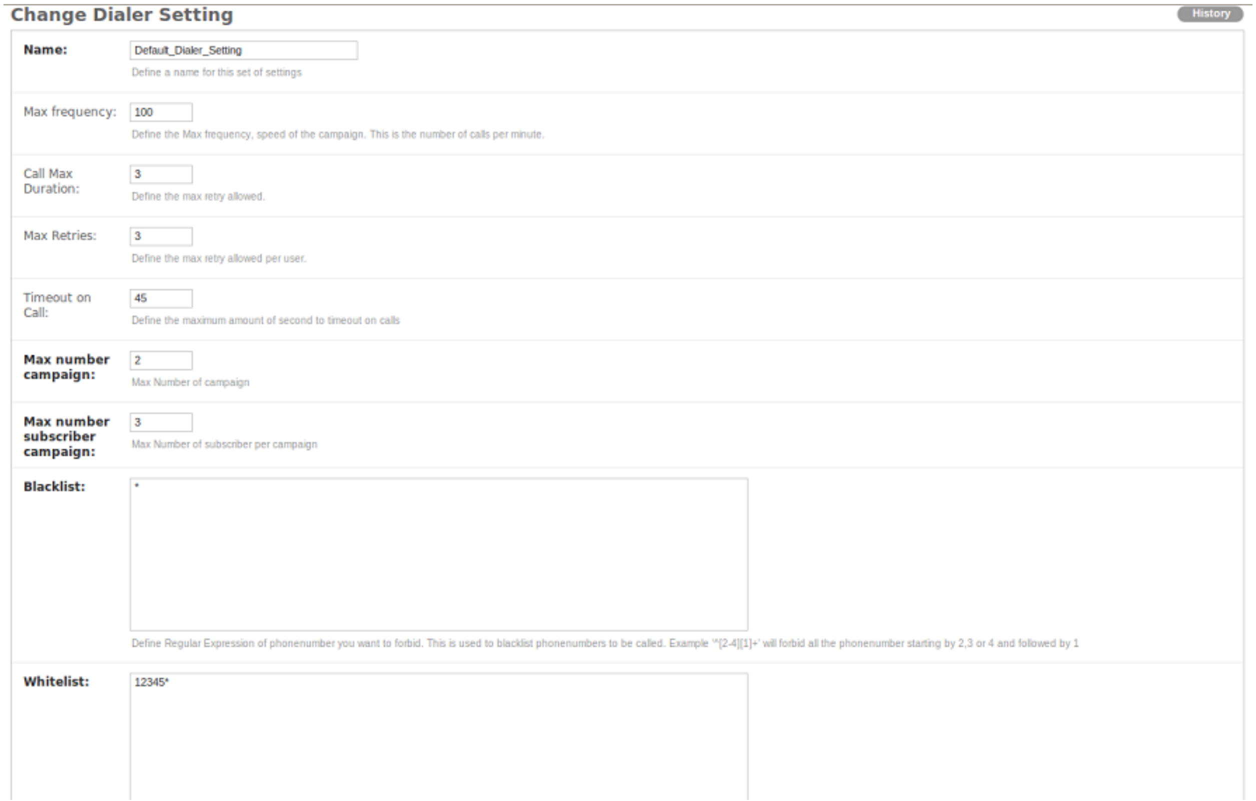
□	Name	Max frequency	Call Max Duration	Max Retries	Timeout on Call	Max number campaign	Max number subscriber campaign	Blacklist	Whitelist	Updated date
□	Default_Dialer_Setting	100	3	3	45	2	3	*	12345*	April 29, 2011, 8:43 a.m.

1 Dialer Setting

To Add/Update dialer settings for a Newfies-Dialer user

URL:

- http://localhost:8000/admin/dialer_settings/dialersetting/add/
- http://localhost:8000/admin/dialer_settings/dialersetting/1/



Change Dialer Setting

Name: Default_Dialer_Setting
Define a name for this set of settings

Max frequency: 100
Define the Max frequency, speed of the campaign. This is the number of calls per minute.

Call Max Duration: 3
Define the max retry allowed.

Max Retries: 3
Define the max retry allowed per user.

Timeout on Call: 45
Define the maximum amount of second to timeout on calls

Max number campaign: 2
Max Number of campaign

Max number subscriber campaign: 3
Max Number of subscriber per campaign

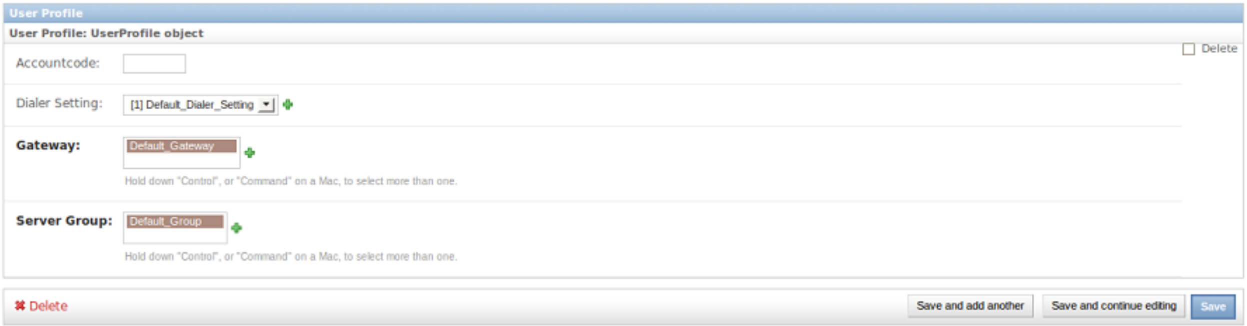
Blacklist: *
Define Regular Expression of phonenumber you want to forbid. This is used to blacklist phonenumbers to be called. Example '^[2-4][1]+' will forbid all the phonenumber starting by 2,3 or 4 and followed by 1

Whitelist: 12345*

To apply dialer settings limit to a User, click on `Customers` or `Admins`, select the user to be updated & apply settings from the dialer settings list.

URL:

- http://localhost:8000/admin/auth/staff/1/



User Profile: UserProfile object

Accountcode:

Dialer Setting: [1] Default_Dialer_Setting

Gateway: Default_Gateway
Hold down "Control", or "Command" on a Mac, to select more than one.

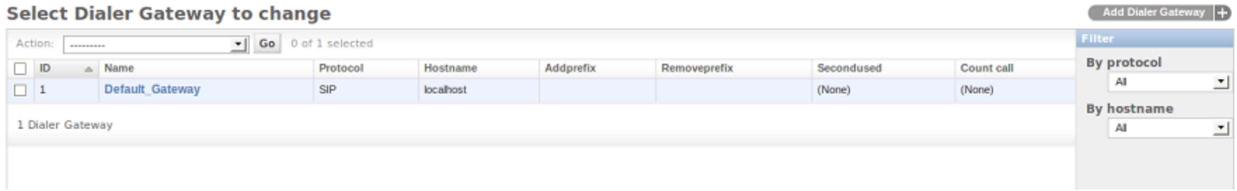
Server Group: Default_Group
Hold down "Control", or "Command" on a Mac, to select more than one.

7.5.10 Dialer Gateway

The Dialer Gateway list will be displayed from the following URL. You can add a new gateway by clicking `Add Dialer Gateway` and adding the details (e.g. gateway name, hostname, protocol etc.). Also from the gateway list, click on the gateway that you want to update.

URL:

- http://localhost:8000/admin/dialer_gateway/gateway/



Select Dialer Gateway to change

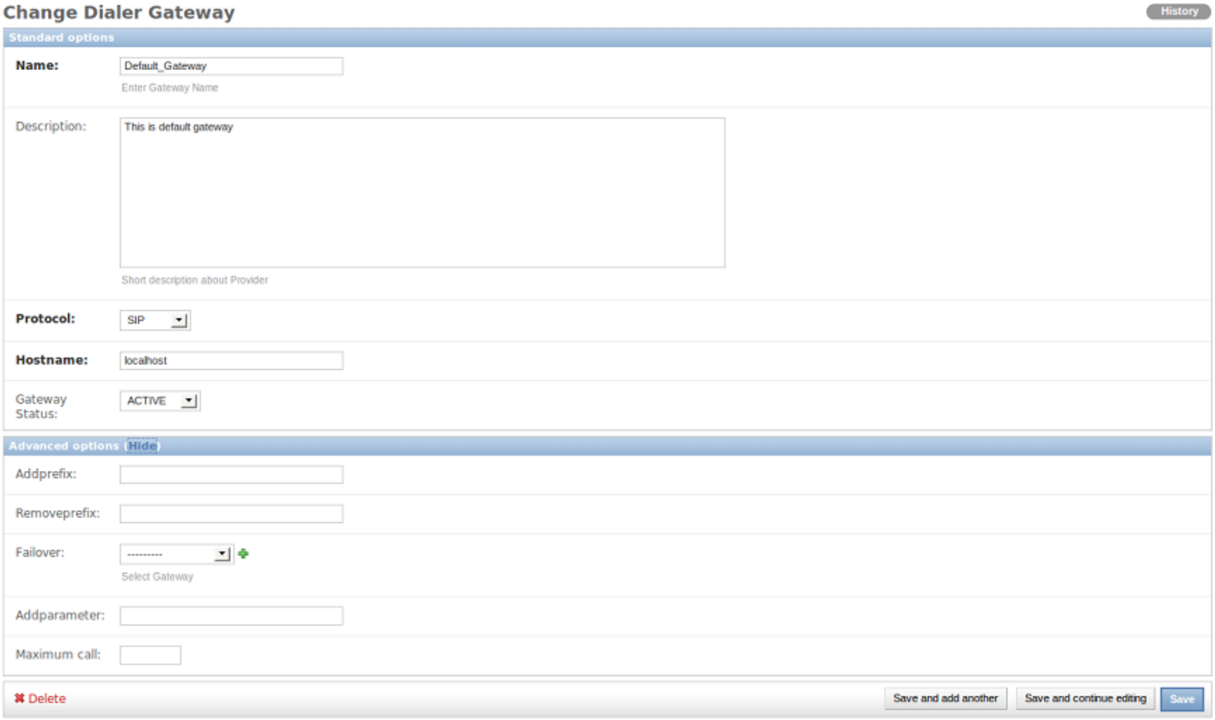
☐	ID	Name	Protocol	Hostname	Addprefix	Removeprefix	Secondused	Count call
☐	1	Default_Gateway	SIP	localhost			(None)	(None)

1 Dialer Gateway

To Add/Update a dialer gateway

URL:

- http://localhost:8000/admin/dialer_gateway/gateway/add/
- http://localhost:8000/admin/dialer_gateway/gateway/1/



Change Dialer Gateway

Name: Default_Gateway
Enter Gateway Name

Description: This is default gateway
Short description about Provider

Protocol: SIP

Hostname: localhost

Gateway Status: ACTIVE

Addprefix:

Removeprefix:

Failover: ---------
Select Gateway

Addparameter:

Maximum call:

7.5.11 Survey

The survey application for Newfies-Dialer allows polls and surveys to be taken over the phone.

Each contact is called, and then played a sound file. After the sound file is heard, the user can enter their answer through the phone keypad using keys 0 to 9.

Therefore before creating the survey, the first job is to upload the audio for the survey. One audio file is required for each question.

Add Audio File

CLick Audio Files, then click Add.

Give the audio file a descriptive name, chose the file to upload.

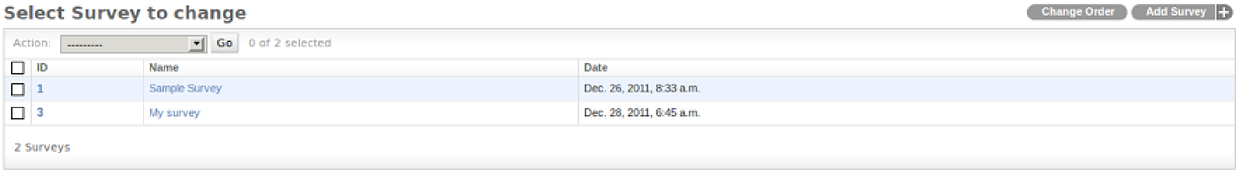
In the Admin screens, there are options to convert to other formats, change the number of channels, e.g. change from stereo to mono, and down-sample the frequency. Finally, the user can be assigned.

Survey Admin Functions

From the admin screen, Surveys can be added, deleted and edited. Additionally, survey questions can be re-ordered.

URL:

- http://localhost:8000/admin/survey/surveyapp/



Select Survey to change

Action: --------- Go 0 of 2 selected

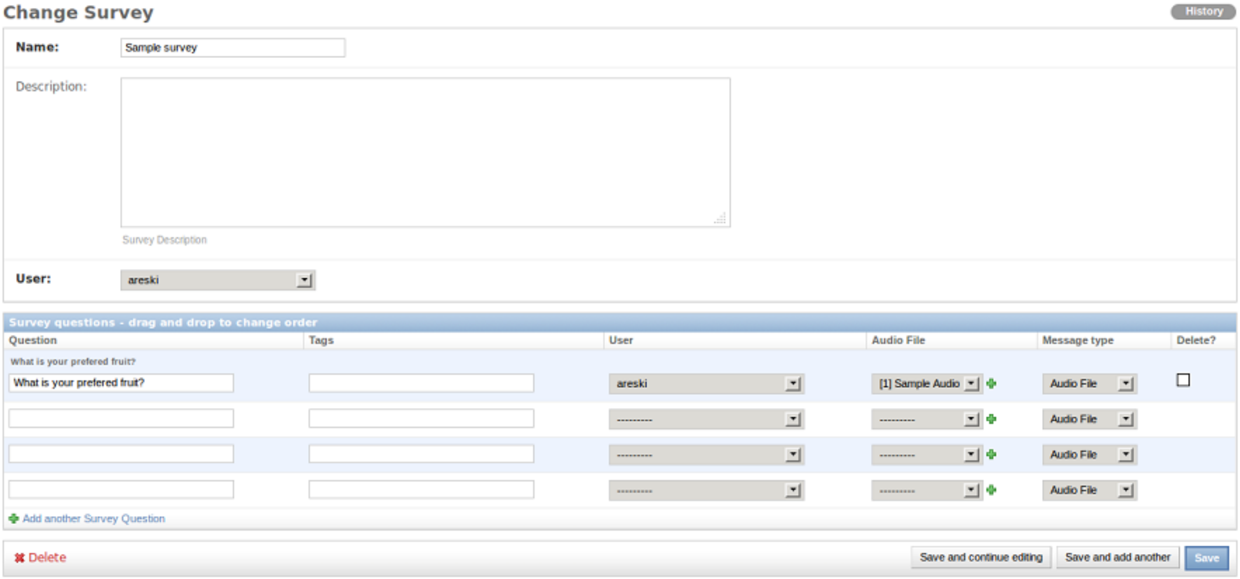
□	ID	Name	Date
□	1	Sample Survey	Dec. 26, 2011, 8:33 a.m.
□	3	My survey	Dec. 28, 2011, 6:45 a.m.

2 Surveys

To Add/Update a survey

URL:

- http://localhost:8000/admin/survey/surveyapp/add/
- http://localhost:8000/admin/survey/surveyapp/1/



Change Survey

Name: Sample survey

Description:

Survey Description

User: areski

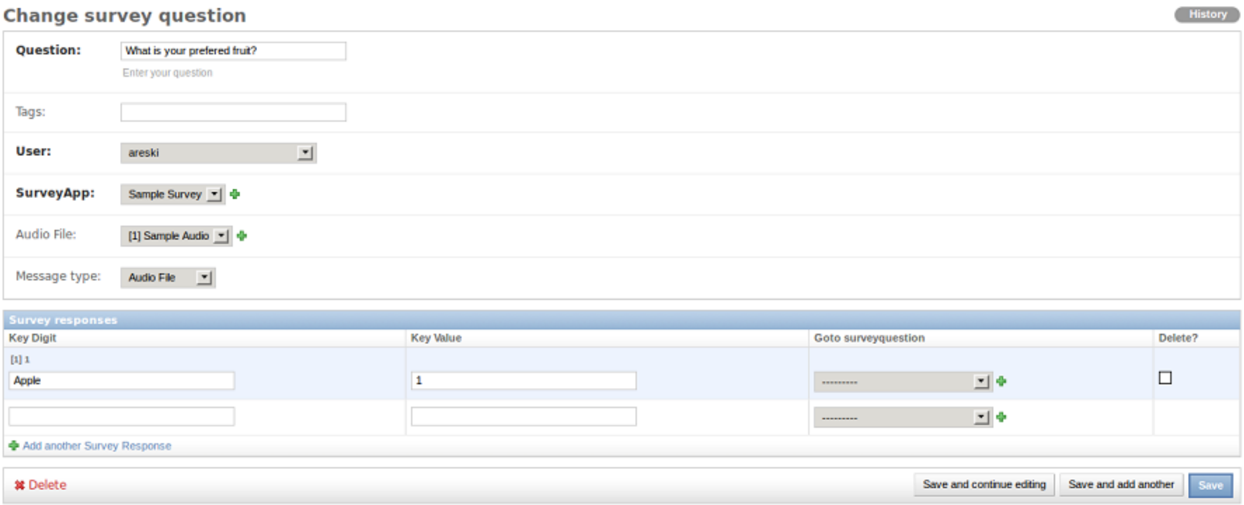
Survey questions - drag and drop to change order

Question	Tags	User	Audio File	Message type	Delete?
What is your prefered fruit? What is your prefered fruit?		areski	[1] Sample Audio	Audio File	□
		---------	---------	Audio File	
		---------	---------	Audio File	
		---------	---------	Audio File	

Add another Survey Question

To Add/Update a survey question

- http://localhost:8000/admin/survey/surveyquestion/add/
- http://localhost:8000/admin/survey/surveyquestion/1/



Change survey question

Question:	What is your prefered fruit? Enter your question
Tags:	
User:	areski
SurveyApp:	Sample Survey
Audio File:	[1] Sample Audio
Message type:	Audio File

Survey responses

Key Digit	Key Value	Goto surveyquestion	Delete?
[1] 1 Apple	1	---------	□

When the survey is complete, the results can be inspected.

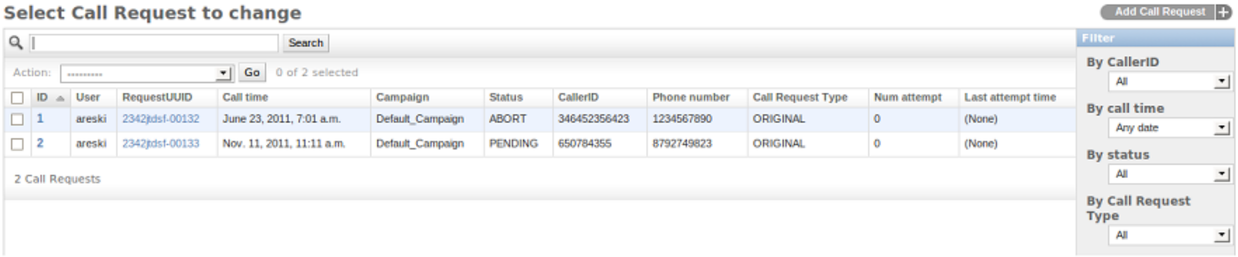
- http://localhost:8000/admin/survey/surveycampaignresult/

7.5.12 Call Request

The call request list will be displayed from the following URL. You can add a new call request by clicking `Add Call Request`. Also from the call request list, click on the request to update.

URL:

- http://localhost:8000/admin/dialer_cdr/callrequest/



Select Call Request to change

□	ID	User	RequestUUID	Call time	Campaign	Status	CallerID	Phone number	Call Request Type	Num attempt	Last attempt time
□	1	areski	2342jtdsf-00132	June 23, 2011, 7:01 a.m.	Default_Campaign	ABORT	346452356423	1234567890	ORIGINAL	0	(None)
□	2	areski	2342jtdsf-00133	Nov. 11, 2011, 11:11 a.m.	Default_Campaign	PENDING	650784355	8792749823	ORIGINAL	0	(None)

2 Call Requests

Filter

By CallerID: All

By call time: Any date

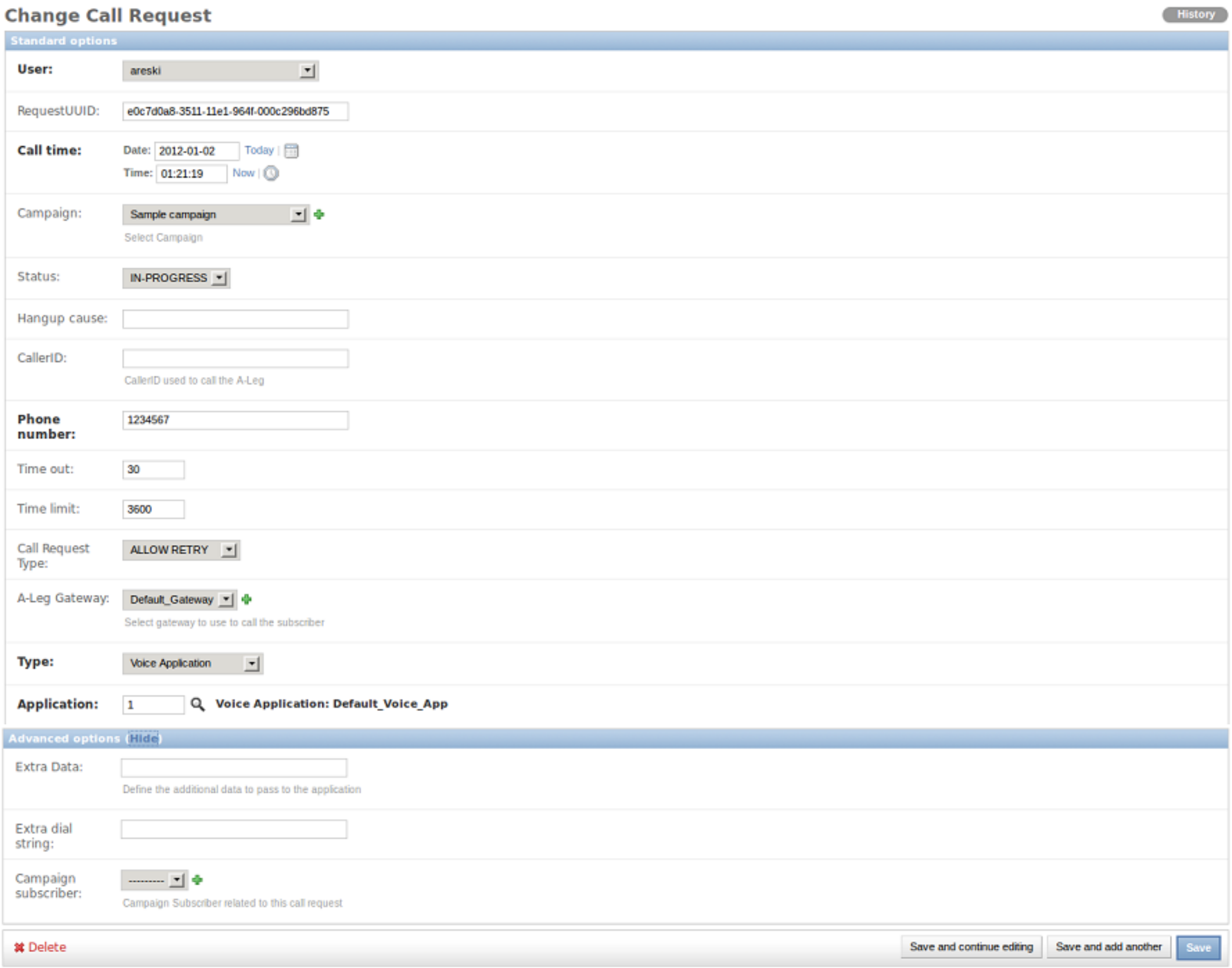
By status: All

By Call Request Type: All

To Add/Update a Call Request

URL:

- http://localhost:8000/admin/dialer_cdr/callrequest/add/
- http://localhost:8000/admin/dialer_cdr/callrequest/1/



Change Call Request

Standard options

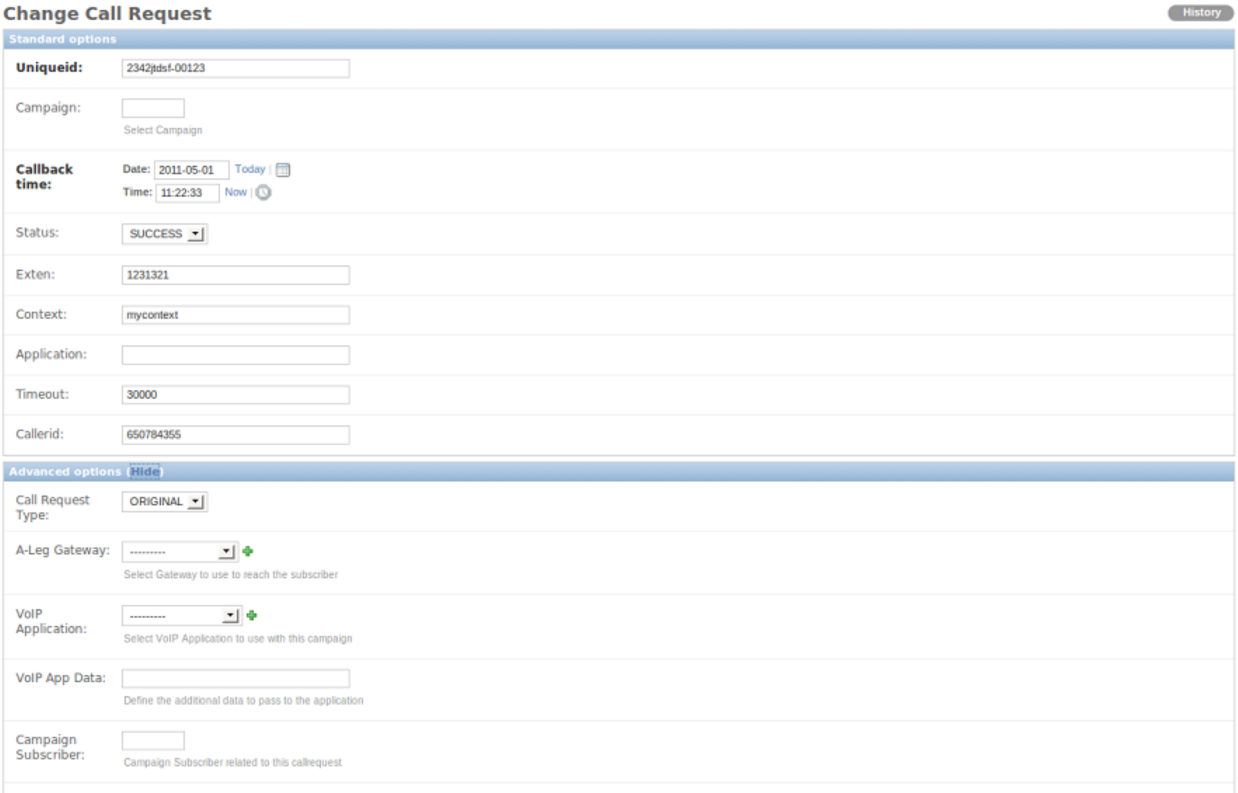
User:	areski
RequestUUID:	e0c7d0a8-3511-11e1-964f-000c296bd875
Call time:	Date: 2012-01-02 Time: 01:21:19
Campaign:	Sample campaign Select Campaign
Status:	IN-PROGRESS
Hangup cause:	
CallerID:	CallerID used to call the A-Leg
Phone number:	1234567
Time out:	30
Time limit:	3600
Call Request Type:	ALLOW RETRY
A-Leg Gateway:	Default_Gateway Select gateway to use to call the subscriber
Type:	Voice Application
Application:	1 Voice Application: Default_Voice_App

Advanced options (Hide)

Extra Data:	Define the additional data to pass to the application
Extra dial string:	
Campaign subscriber:	--------- Campaign Subscriber related to this call request

7.5.13 Call contacts/subscribers

A call-request will spool a call directly from the platform using a dialer gateway and update the call-request status after receiving a response from the gateway.



Change Call Request

Standard options

Uniqueid:	2342jtdsf-00123
Campaign:	Select Campaign
Callback time:	Date: 2011-05-01 Today Time: 11:22:33 Now
Status:	SUCCESS
Exten:	1231321
Context:	mycontext
Application:	
Timeout:	30000
Callerid:	650784355

Advanced options (Hide)

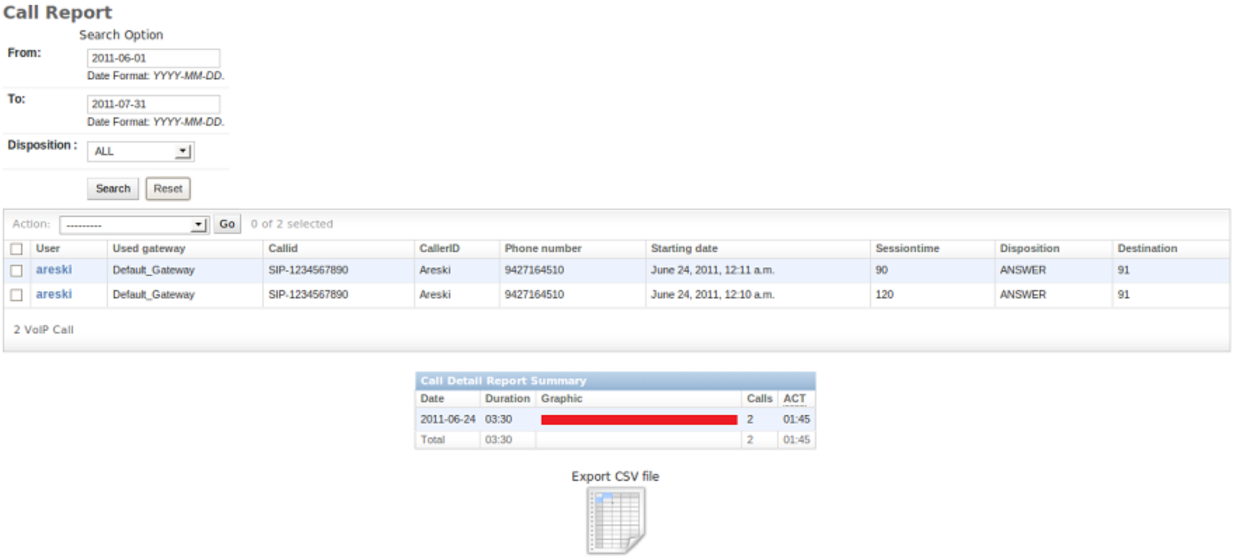
Call Request Type:	ORIGINAL
A-Leg Gateway:	--------- Select Gateway to use to reach the subscriber
VoIP Application:	--------- Select VoIP Application to use with this campaign
VoIP App Data:	Define the additional data to pass to the application
Campaign Subscriber:	Campaign Subscriber related to this callrequest

7.5.14 VoIP Call Report

A VoIP Call list will be displayed from following URL. You **can not** add new call reports.

URL:

- http://localhost:8000/admin/dialer_cdr/voipcall/



Call Report

Search Option

From:	2011-06-01 Date Format: *YYYY-MM-DD.*
To:	2011-07-31 Date Format: *YYYY-MM-DD.*
Disposition :	ALL

Action: --------- Go 0 of 2 selected

□	User	Used gateway	Callid	CallerID	Phone number	Starting date	Sessiontime	Disposition	Destination
□	areski	Default_Gateway	SIP-1234567890	Areski	9427164510	June 24, 2011, 12:11 a.m.	90	ANSWER	91
□	areski	Default_Gateway	SIP-1234567890	Areski	9427164510	June 24, 2011, 12:10 a.m.	120	ANSWER	91

2 VoIP Call

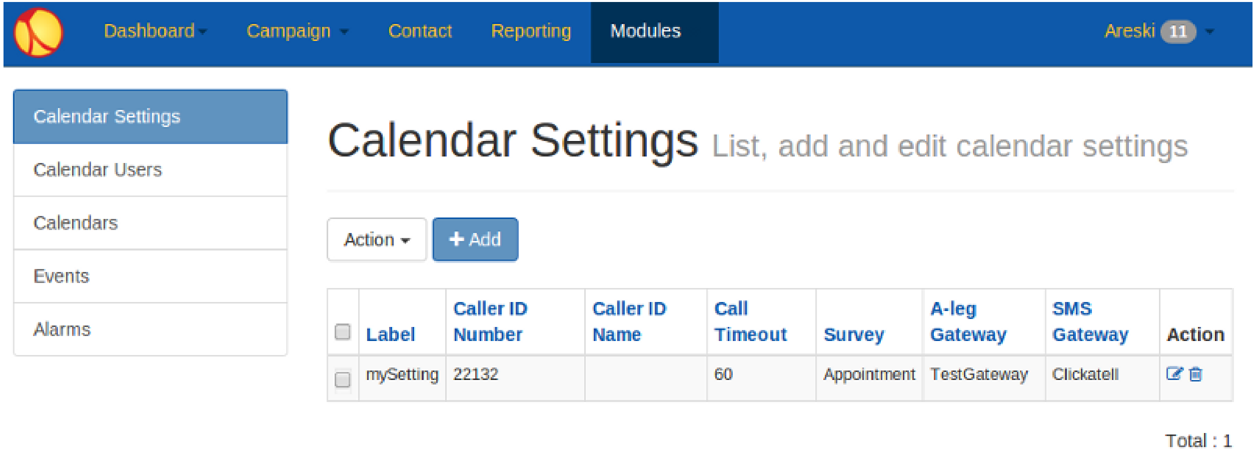
Call Detail Report Summary

Date	Duration	Graphic	Calls	ACT
2011-06-24	03:30	🟥	2	01:45
Total	03:30		2	01:45

Export CSV file

7.6 Appointment Module

From version 2.10, Newfies-Dialer includes an appointments module supporting appointment reminders. Newfies-Dialer was originally designed to provide powerful voice broadcasting, making thousands or even millions of calls via gateways to different audiences. One of the major limitations of voice broadcasting was that it is difficult, or even impossible to target a call to a specific contact at a *programmed or preset time*. The appointments module addresses this limitation allowing calls to be sent at a preset time.



Calendar Settings List, add and edit calendar settings

□	Label	Caller ID Number	Caller ID Name	Call Timeout	Survey	A-leg Gateway	SMS Gateway	Action
□	mySetting	22132		60	Appointment	TestGateway	Clickatell	

Total : 1

7.6.1 Structure of the appointment module

The appointments module has the following components and are described in this section.

A survey is created with the required messages and actions. Once the Calendar settings and users are configured, an event is created attached to a calendar, then an alarm is triggered by the Event and sent out via voice call, email or SMS. In the case of a Voice Call, the survey selected will be executed.

Calendar User

The Calendar-User is a sub-user of the Newfies-Dialer User. The Calendar-User will be the customer who is creating the reminders or alarms. For instance, if you create a appointment reminders for a Doctor's surgery, then the calendar user may be an individual doctor or secretary. The rights of a Calendar-User can be configured via the Calendar-User's settings.

Note that a Calendar-User does not log into the interface, alarms are created and configured via the Newfies-Dialer login.

Calendar

Calendar is an entity that helps to group and collect events.

Event

An Event sets the time, date and duration of when the alarms are to be sent out. An event is linked to a calendar. The Event also includes custom rules that can be set up to trigger an event every day, week or month, as well as create more complex rules such as; repeat every Tuesday at a given time.

A data field is provided and used to pass extra variables You can pass custom variables by using the "Additional Parameters (JSON)" field which allows you to add custom key-values that will be used in the survey and replaced by the Text2Speech processor.

For example; set "Additional Parameters (JSON)" to:

```
{"doctor_name": "Herbert", "apt_date" : "10 January at 5pm"}
```

You can use "sms_phonenumber" as one of the "Additional Parameters", in this case it will overwrite the sms destination number when sending SMS during an IVR call:

```
{"sms_phonenumber": "6004891138"}
```

When a survey is created with a node that plays TTS, you can replace the key-values in the text:

```
"We are calling you on behalf of {doctor_name}, we want to remind you of your appointment at {ap
```

Rule

Newfies-Dialer comes with a set of predefined rules which can be used for the events, e.g. the Daily rule, which, if applied to an event, will make the event recur every day.

The Rule is based on the rrule python module. The rrule module offers a small, complete, and very fast implementation of the recurrence rules documented in the iCalendar RFC : http://www.ietf.org/rfc/rfc2445.txt

More info about rrule: http://labix.org/python-dateutil#head-470fa22b2db72000d7abe698a5783a46b0731b57

Events happen at a specified time and you can also program the event to recur daily or follow recurring rules, In order to get notified about an event it is necessary to create Alarms related to the event.

Alarm

An Alarm notifies by SMS, voice call or email that an event is occuring. Other methods could be added. Alarms have a number of settings which offer a great deal of flexibility.

Settings of Alarms

- Method: Set the method to use for an Alarm (Voice Call, SMS or email)
- Date Start: Date and time to trigger the alarm which generally will be similar to the event date and time.
- Alarm Phone Number: Set the phone number to be called by the alarm.
- Alarm Email: Set the email address to send the alarm email.

- Daily Start & Daily Stop: Configure when alarms are allowed to triggered during the day, for instance, prevent alarms before 6AM and 10PM.
- Advance Notice: How long before the alarm occurs to start the notification.
- Max Retry: Applies to a voice call, and sets the number of retries in the event of a failed call.
- Retry Delay: Interval between each call retry
- Phone Number SMS Failure: In the case of a failed voice call, an SMS is triggered to this number
- URL Cancel / URL Confirm / Phone Number Transfer: A feature that will be implemented in a future version.

Alarm Result

A mechanism is provided to store the results from Alarms. For instance, imagine that a calendar user is calling all clients at the beginning of the week to tell them about their forthcoming appointment and ask them if they want to confirm, cancel or re-schedule.

For this scenario, a Survey application is built with the 3 options:

```
- Press 1 to Confirm,
- Press 2 to Cancel
- or Press 4 to Re-schedule your appointment
```

You will simply need to create a **Rating Question**, as this section type will behave differently with the appointment module and the result of the question will be stored in the Alarm result. This will be practical for future reporting and will allow the calendar user to see all the Events that need to be cancelled or re-scheduled.

Alarm Requests

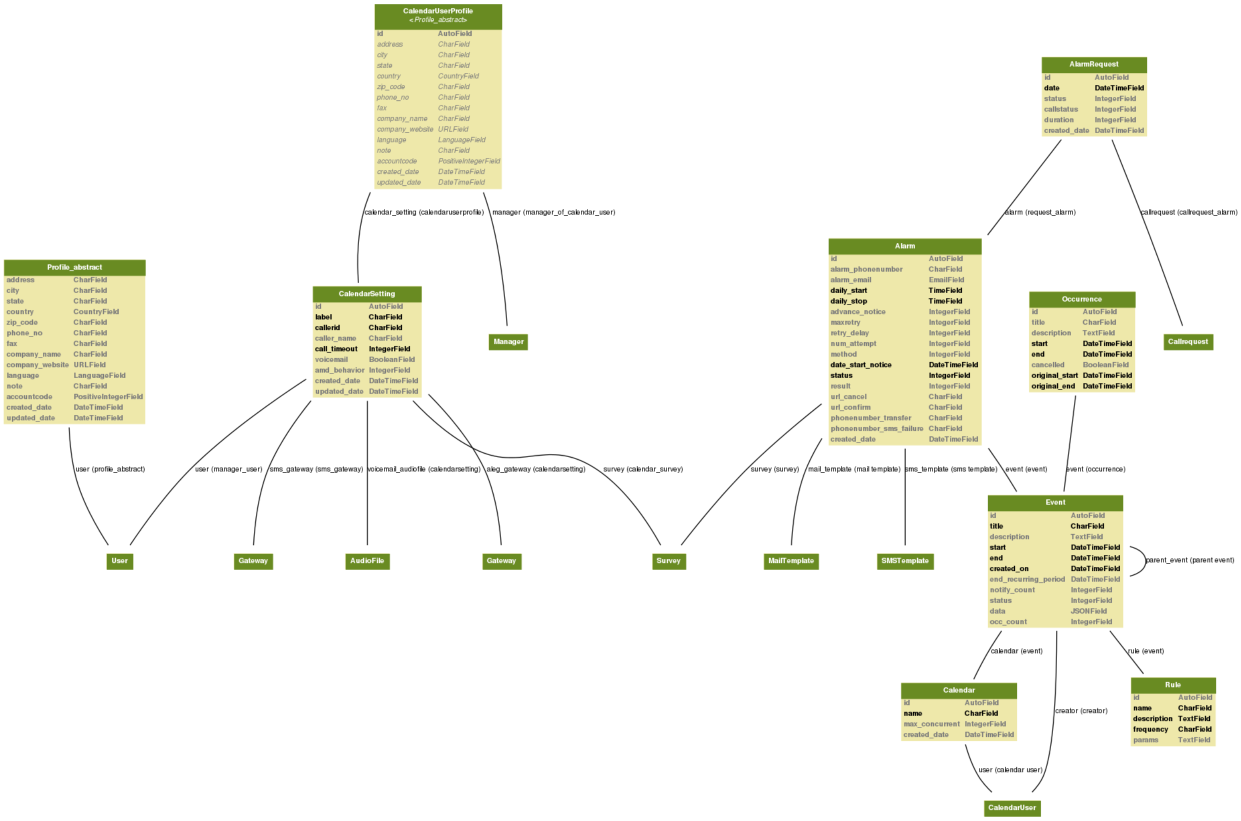
Keeps track of the alarms attempts, it’s a useful resource to provide detailed logs.

Calendar User Settings

The Calendar Settings determine the Caller ID name and number to be delivered to the called party, the timeout and the voice and SMS gateway to use. There is also an option to set the AMD (if installed) settings.

7.6.2 Models description

This diagram of the appointment models should help developers and integrators to understand the system.



Explore the appointment models.

7.6.3 User cases

The Appointment Module has been built with flexibility in mind, trying to cover several scenarios required by different applications and software to perform complex appointment reminder applications or alarm systems.

Modules you could build with the Newfies-Dialer Appointment Module may include:

Wake up call alarm

> Each of the calendar users will have a calendar in which they create an wake up event, decide when the event will be triggered, how many times, delay between retries, etc...
>
> The voice application will then play something like "Good morning, it's 7am and it's time to wake up and get ready for work"

Appointment Reminder

> In a common appointment reminder scenario, your user could be doctors surgery with a need to call their patient 24 hours before each appointment and offer an an IVR menu that will call their patient and say "Hello, you have an appointment tomorrow with Doctor {doc_name} at {apt_date}, please press 1 to confirm, press 2 to cancel or press 3 to reschedule"
>
> When the user presses any key during a Rating type of Node on the IVR application, this is considered as a result and will be stored in the alarm result field. If the patient presses 2, it will be seen in the results

and could be displayed to the doctor's receptionist, if the user presses 3, the call could be transferred to reception to re-arrange the appointment.

The Appointment Module supports Voice Calls, SMS and email, so the system could be configured to send a passive SMS as an extra reminder one hour before the appointment.

Calendar Alarm event

Some users might want to simply remember important dates and events, such as a meeting, birthday or to pickup their child from music class. Each Calendar User can create as many Calendars as they want, for instance they could have a personal calendar and a work calendar so as not to mix personal and professional events.

7.6.4 Process flow

To be able to setup and receive alarm there is a process to follow and is described below:

1. Create and Configure Voice Gateway and SMS gateway.

   This is done via the admin panel: http://127.0.0.1:8000/admin

2. Create an IVR application (Survey) that will be played to the user receiving the calls

   Go to the survey module and create an application with several IVR nodes: http://127.0.0.1:8000/module/survey/

3. Seal the Survey. This prevents the survey being modified and is important to ensure accurate and consistant reporting on each survey node.

   There is a button in the Action column against the survey to seal the survey and prevent further editing: http://127.0.0.1:8000/module/survey/

4. Create Calendar User Settings, define a callerID and configure the gateway to use.

   Add new Calendar Settings at: http://127.0.0.1:8000/module/calendar_setting/

5. Create Calendar User, set credentials and permissions for this user and assign them to Calendar Setting.

   Create a user with a username & password: http://127.0.0.1:8000/module/calendar_user/

6. Create Calendar and give it a name:

   http://127.0.0.1:8000/module/calendar/

7. Create Event, for instance an event happening today at 11am, define when the event start and finish, add an recurring rule if the event is to recur.

   Create Event can be done by click on Add button : http://127.0.0.1:8000/module/event/

8. Create Alarm to receive notification when Events are happening. e.g. Create an alarm of type "Call", link it to the previously created event, add a date and time for the alarm and the phone number top be called, then configure the rest of the alarm settings as required.

   Create Alarm can be done by click on Add button : http://127.0.0.1:8000/module/alarm/

9. Access results of Alarm Requests.

   Access detailed logs of the Alarm by browsing to alarm request at : http://127.0.0.1:8000/admin/appointment/alarmrequest/

7.6.5 Work Flow with API

One of the most powerful features of The Appointment Module are the API's that allow developers to build an application on top of Newfies-Dialer or integrate Newfies-Dialer into third party software such as CRM systems.

Described below is an example of work flow with the API's.

1) Create Calendar Setting

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data '{"l
```

Result:

```
HTTP/1.0 201 CREATED
Date: Mon, 16 Dec 2013 11:19:30 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Language: en
Content-Type: application/json; charset=utf-8
Location: http://localhost:8000/rest-api/calendar-setting/3/
Allow: GET, POST, HEAD, OPTIONS

{
  "user": "manager",
  "sms_gateway": "http://localhost:8000/rest-api/sms-gateway/1/",
  "url": "http://localhost:8000/rest-api/calendar-setting/3/",
  "label": "cal_setting",
  "callerid": "123456",
  "caller_name": "xyz",
  "call_timeout": 60,
  "survey": "http://localhost:8000/rest-api/sealed-survey/1/",
  "aleg_gateway": "http://localhost:8000/rest-api/gateway/1/",
  "voicemail": false,
  "amd_behavior": 1,
  "voicemail_audiofile": null,
  "created_date": "2013-12-16T11:19:29.994Z",
  "updated_date": "2013-12-16T11:19:29.994Z"
}
```

2) Create Calendar User

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data '{"u
```

Result:

```
HTTP/1.0 201 CREATED
Date: Mon, 16 Dec 2013 11:20:33 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Language: en
Content-Type: application/json; charset=utf-8
Location: http://localhost:8000/rest-api/calendar-user/6/
Allow: GET, POST, HEAD, OPTIONS
```

```
{
  "url": "http://localhost:8000/rest-api/calendar-user/6/",
  "username": "caluser3",
  "password": "pbkdf2_sha256$12000$Rb78UOwQeL2T$YWWy02zcxtFTIDG0ac4lJ7i9jtUbK7FCG1IkgARDVAE=",
  "last_name": "",
  "first_name": "",
  "email": "caluser3@gmail.com",
  "groups": []
}
```

3) Update Calendard User Profile

We will need to use the previously created Calendar Setting.

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X PATCH --data '{
```

Result:

```
HTTP/1.0 200 OK
Date: Mon, 16 Dec 2013 11:23:44 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json; charset=utf-8
Content-Language: en
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS

{
  "manager": "manager",
  "id": 4,
  "user": 6,
  "address": null,
  "city": null,
  "state": null,
  "country": "",
  "zip_code": null,
  "phone_no": null,
  "fax": null,
  "company_name": null,
  "company_website": null,
  "language": null,
  "note": null,
  "accountcode": 35365532,
  "created_date": "2013-12-16T11:20:33.456Z",
  "updated_date": "2013-12-16T11:23:44.342Z",
  "calendar_setting": 3
}
```

4) Create a new Calendar

We will call the new calendar “myCalendar”

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data '{"n
```

Result:

```
HTTP/1.0 201 CREATED
Date: Mon, 16 Dec 2013 11:25:01 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Language: en
Content-Type: application/json; charset=utf-8
Location: http://localhost:8000/rest-api/calendar/4/
Allow: GET, POST, HEAD, OPTIONS

{
  "url": "http://localhost:8000/rest-api/calendar/4/",
  "name": "mynewcalendar",
  "user": "http://localhost:8000/rest-api/calendar-user/6/",
  "max_concurrent": 1,
  "created_date": "2013-12-16T11:25:01.378Z"
}
```

5) Create a new Event

We will create a new event in the previous created Calendar “myCalendar”.

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data '{"t
```

Result:

```
HTTP/1.0 201 CREATED
Date: Mon, 16 Dec 2013 11:26:56 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Language: en
Content-Type: application/json; charset=utf-8
Location: http://localhost:8000/rest-api/event/3/
Allow: GET, POST, HEAD, OPTIONS

{
  "url": "http://localhost:8000/rest-api/event/3/",
  "title": "event_with_new_title",
  "description": null,
  "start": "2013-12-10T12:34:43",
  "end": "2013-12-15T14:43:32",
  "creator": "http://localhost:8000/rest-api/calendar-user/6/",
  "created_on": "2013-12-16T11:26:56.056Z",
  "end_recurring_period": "2013-12-27T12:23:34",
  "rule": null,
  "calendar": "http://localhost:8000/rest-api/calendar/4/",
  "notify_count": 0,
  "status": 1,
  "data": null,
  "parent_event": null,
  "occ_count": 0
}
```

6) Get list of child events from a main event

Events, occurring over time, will have a tail of sub-events linked to the parent event. Usually the systems integrator using API's will keep track of the parent event and at some stage may need to retrieve information about the sub-events.

This is achieved by using the API /get_list_child/

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/event/5/g
```

Result:

```
HTTP/1.0 200 OK
Date: Thu, 19 Dec 2013 16:37:22 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json
Content-Language: en
Allow: GET, HEAD, OPTIONS
Set-Cookie:  sessionid=lklh4z9cop3jhh48n9i9ioe2wkp8simx; expires=Thu, 02-Jan-2014 16:37:22 GMT;

{
  "url": "http://localhost:8000/rest-api/event/11/",
  "start": "2013-12-17 16:26:07+00:00",
  "end": "2013-12-17 17:26:07+00:00",
  "description": "",
  "title": "EventTitle"
}
```

7) Update the last child event status to Paused

In some scenarios it may be necessary to pause an alarm being triggered. As an alarm is linked to all occurence of an event created, it's not necessary to update the status of the parent event or all the child events, only the last event needs updated.

This is achieved by using the API /update_last_child_status/

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X PATCH --data '{
```

Result:

```
HTTP/1.0 200 OK
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json
Content-Language: en
Allow: PATCH, OPTIONS
Set-Cookie:  sessionid=dmzjzwt3b6l0eru6vq27vt9ixd9z84ei; expires=Tue, 31-Dec-2013 10:52:26 GMT;
Connection: close
Server: Werkzeug/0.8.3 Python/2.7.3
Date: Tue, 17 Dec 2013 10:52:26 GMT

{
  "status": "event status has been updated"
}
```

8) Get detailed log from alarm request for a event

In order to provide logs or stats to your customers/users, information can be retrieved from the AlarmRequest for a given event.

The result is a nested JSON structure which gives the Event-ID with the Alarm-ID related to the event, plus the list alarm-request-ID for each of those Alarms.

This is achieved by using the API /get_nested_alarm_request/

CURL Usage:

```
curl -u username:password --dump-header - -H 'Accept: application/json' http://localhost:8000/re
```

Result:

```
HTTP/1.0 200 OK
Date: Thu, 19 Dec 2013 16:41:22 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json
Content-Language: en
Allow: GET, HEAD, OPTIONS
Set-Cookie:  sessionid=w7ze05sob1esrsykp94e0hi8gg1tq0kv; expires=Thu, 02-Jan-2014 16:41:22 GMT;

{
  "event-url": "http://localhost:8000/rest-api/event/5/",
  "event-5": {
    "alarm-23": {
      "url": "http://localhost:8000/rest-api/alarm/23/",
      "alarm-request-48": {
        "status": "4",
        "url": "http://localhost:8000/rest-api/alarm-request/48/",
        "alarm-callrequest": "http://localhost:8000/rest-api/callrequest/15731/",
        "duration": "0",
        "date": "2013-12-18 17:19:23.368534+00:00",
        "callstatus": "0"
      },
      "alarm-request-49": {
        "status": "5",
        "url": "http://localhost:8000/rest-api/alarm-request/49/",
        "alarm-callrequest": "http://localhost:8000/rest-api/callrequest/15732/",
        "duration": "13",
        "date": "2013-12-18 17:20:05.062474+00:00",
        "callstatus": "0"
      }
    },
    "alarm-21": {
      "url": "http://localhost:8000/rest-api/alarm/21/",
      "alarm-request-40": {
        "status": "5",
        "url": "http://localhost:8000/rest-api/alarm-request/40/",
        "alarm-callrequest": "http://localhost:8000/rest-api/callrequest/15722/",
        "duration": "13",
        "date": "2013-12-16 17:20:27.849068+00:00",
        "callstatus": "0"
      },
    }
  }
}
```

7.6.6 API explorer for Appointment module

Some APIs can be explored and tested easily via the API-Explorer. This is the best way to understand and read about all the APIs provided by Newfies-Dialer.

To access the API-Explorer go to http://127.0.0.1:8000/rest-api/

7.7 SMS Campaign

Creating an SMS campaign is straight-forward in Newfies-Dialer and is usually done via the customer interface.

7.7.1 Create a Phonebook

The first step is to create a phone book consisting of the SMS contacts to be called. Click Contact at the top, then Phonebooks on the side menu, add a new Phonebook and give it a name, for instance SMS-PB1

7.7.2 Add or Import Contacts

Manually add or import contacts to the phone book created in the previous step. Note that most SMS providers require you to properly format the numbers into IETF format, that is to say, +1234567890 rather than 001234567890 or 0111234567890.

7.7.3 Add SMS Campaign

Click Campaign on the top menu, then select SMS Campaign from the dropdown.

Click Add and fill in the fields:

- *Name:* A descriptive name for the Campaign
- *Caller ID Number:* The number to deliver as the caller ID
- *SMS Gateway:* Select the gateway to use for the campaign
- *Description:* Notes on the campaign
- *Phonebook:* Select one or more phone books containing the contacts to call
- *Text Message:* Enter the message to be sent, with no more than 160 characters.

> Tag Replacement can be used here to personalise the message, e.g. Hello {first_name} {last_name} will replace whatever you have in the first and last name fields against the contact.

At this point the campaign could be saved and sent, however there are a number of other parameters that can be adjusted:

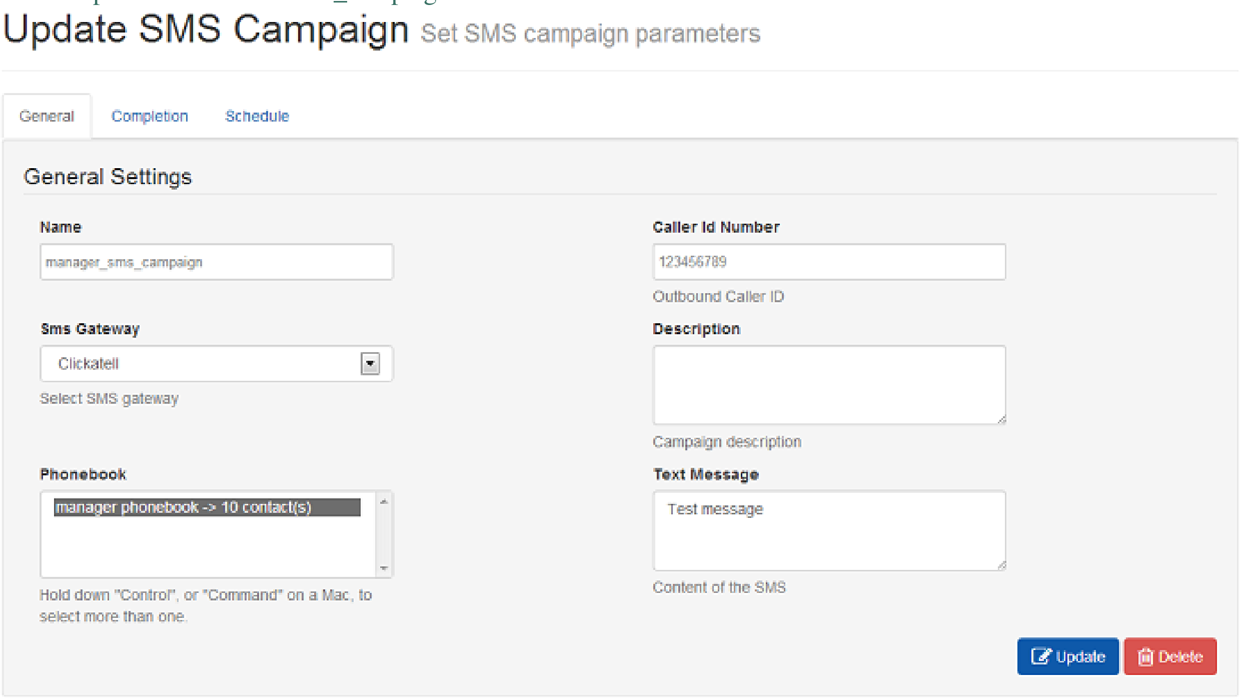
Completion

Under the completion tab:

- *Frequency:* Number of SMS's to send per minute.
- *Time Between Retries:* If a message delivery fails, wait this number of seconds before retrying.
- *Max Retries:* Number of times to re-attempt message delivery.

URL:

- http://localhost:8000/sms_campaign/add/
- http://localhost:8000/sms_campaign/1/



Update SMS Campaign Set SMS campaign parameters

General Settings

Name: manager_sms_campaign

Caller Id Number: 123456789

Outbound Caller ID

Sms Gateway: Clickatell

Select SMS gateway

Description:

Campaign description

Phonebook: manager phonebook -> 10 contact(s)

Hold down "Control", or "Command" on a Mac, to select more than one.

Text Message: Test message

Content of the SMS

Schedule

Under the Schedule Tab

It may not be desirable to send SMS messages during weekends or out of hours, so the daily start and stop time can be set as well as the days of the week to send messages.

There is also a start and stop date as to when it is valid to send the campaign.

When all the settings are as required, then the survey can be saved and started by clicking the play button.

7.7.4 Reporting

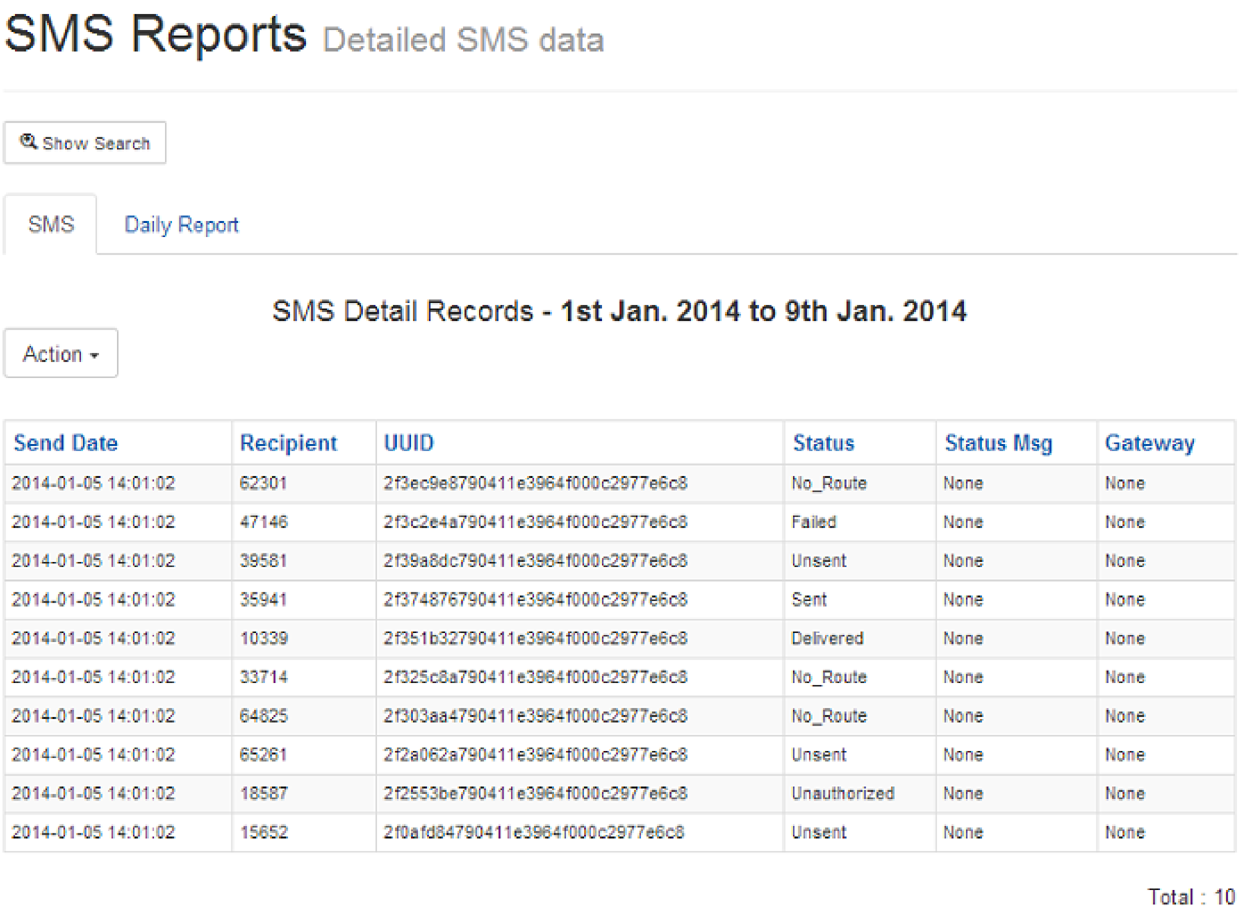
To view the SMS reports, click reporting then SMS reports on the side menu. The call data records for SMS messages are listed under the SMS tab, while a daily report shows the number of SMS's sent by day.

Records can be filtered on date, status and campaign.

SMS CDR can be exported in CSV, XML and Excel formats.

URL:

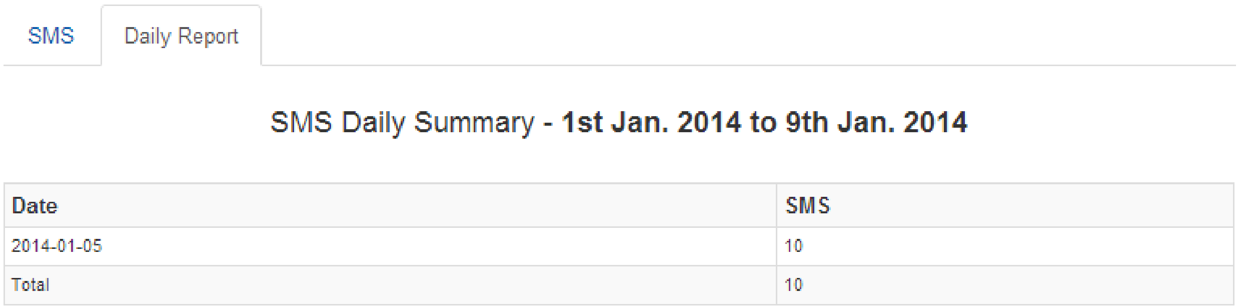
- http://localhost:8000/sms_report/



SMS Detail Records - **1st Jan. 2014 to 9th Jan. 2014**

Send Date	Recipient	UUID	Status	Status Msg	Gateway
2014-01-05 14:01:02	62301	2f3ec9e8790411e3964f000c2977e6c8	No_Route	None	None
2014-01-05 14:01:02	47146	2f3c2e4a790411e3964f000c2977e6c8	Failed	None	None
2014-01-05 14:01:02	39581	2f39a8dc790411e3964f000c2977e6c8	Unsent	None	None
2014-01-05 14:01:02	35941	2f374876790411e3964f000c2977e6c8	Sent	None	None
2014-01-05 14:01:02	10339	2f351b32790411e3964f000c2977e6c8	Delivered	None	None
2014-01-05 14:01:02	33714	2f325c8a790411e3964f000c2977e6c8	No_Route	None	None
2014-01-05 14:01:02	64825	2f303aa4790411e3964f000c2977e6c8	No_Route	None	None
2014-01-05 14:01:02	65261	2f2a062a790411e3964f000c2977e6c8	Unsent	None	None
2014-01-05 14:01:02	18587	2f2553be790411e3964f000c2977e6c8	Unauthorized	None	None
2014-01-05 14:01:02	15652	2f0afd84790411e3964f000c2977e6c8	Unsent	None	None

Total : 10



SMS Daily Summary - **1st Jan. 2014 to 9th Jan. 2014**

Date	SMS
2014-01-05	10
Total	10

CHAPTER 8

Developer doc

Contents:

8.1 Prerequisites

To fully understand this project, developers will need to have a advanced knowledge of:

- Django : http://www.djangoproject.com/
- Celery : http://celeryproject.org/
- Python : http://www.python.org/
- Freeswitch : http://freeswitch.org/
- Freeswitch : Event_Sockets (wiki.freeswitch.org/wiki/Mod_event_socket)

8.2 Coding Style & Structure

8.2.1 Style

Coding follows the PEP 8 Style Guide for Python Code.

8.2.2 Structure

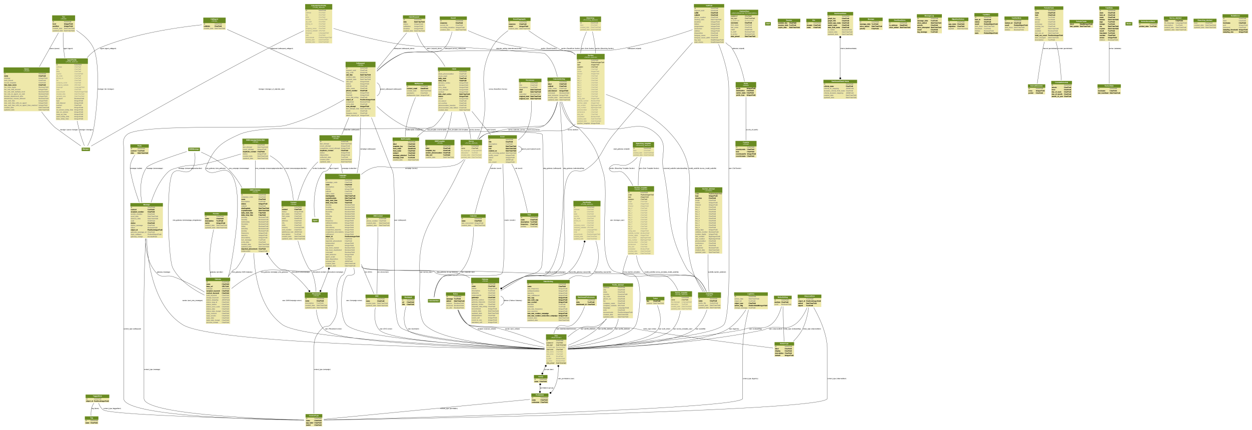
The newfies directory:

```
|-- custom_admin_tools - The code for admin dashboard/menu
|-- api                - The code for APIs
|-- dialer_contact     - The code for phonebook, contact
|   `-- fixtures
|-- dialer_campaign    - The code for dialer campaign
|   `-- fixtures
|-- dialer_cdr         - This defines the call request & its information
|   `-- fixtures
|-- dialer_gateway     - This defines the trunk to deliver the VoIP Calls
|   `-- fixtures
|-- dialer_settings    - This defines sets of settings to apply on user
|-- survey             - The code for survey module
```

```
|   `-- fixtures
|-- frontend            - The code for dashboard, forgot password
|-- static
|   |-- newfies
|   |     |-- css
|   |     |-- js
|   |     |-- icons
|   |     `-- images
|-- user_profile        - The code for user profile to extend auth model of Django
|-- resources           - This area is used to hold media files
|-- usermedia           - This folder is used to upload audio files
`-- templates           - This area is used to override templates
    |-- admin
    |   |-- dialer_contact
    |   `-- dialer_cdr
    |-- admin_tools
    |-- registration
    |-- memcache_status
    `-- frontend
```

8.3 Database Models

The current database schema is shown below:



Explore the database models.

8.4 Objects Description

8.4.1 `Phonebook`

class `dialer_contact.models.`**`Phonebook`**(**args*, ***kwargs*)

This defines the Phonebook

Attributes:

- `name` - phonebook name.
- `description` - description about the phonebook.

Relationships:

- `user` - Foreign key relationship to the User model. Each phonebook is assigned to a User

Name of DB table: dialer_phonebook

`phonebook_contacts``()`

This will return a count of the contacts in the phonebook

8.4.2 Contact

class `dialer_contact.models.`**`Contact`**`(`**args*, ***kwargs*`)`

This defines the Contact

Attributes:

- `contact` - Contact no
- `last_name` - Contact's last name
- `first_name` - Contact's first name
- `email` - Contact's e-mail address
- `city` - city name
- `description` - description about a Contact
- `status` - contact status
- `additional_vars` - Additional variables

Relationships:

- `phonebook` - Foreign key relationship to the Phonebook model. Each contact mapped with a phonebook
- `country` - Foreign key relationship to the Country model. Each contact mapped with a country

Name of DB table: dialer_contact

`contact_name``()`

Return Contact Name

`replace_tag``(`*text*`)`

Replace tag by contact values. This function will replace all the following tags:

> {last_name} {first_name} {email} {country} {city} {phone_number}

as well as, get additional_vars, and replace json tags

8.4.3 CampaignManager

class `dialer_campaign.models.`**`CampaignManager`**

Campaign Manager

`get_expired_campaign``()`

Return all the campaigns which are expired or going to expire based on the expiry date but status is not 'END'

`get_running_campaign``()`

Return all the active campaigns which will be running based on the expiry date, the daily start/stop time and days of the week

8.4.4 Campaign

class dialer_campaign.models.**Campaign**(**args*, ***kwargs*)

This defines the Campaign

Attributes:

- campaign_code - Auto-generated campaign code to identify the campaign
- name - Campaign name
- description - Description about the Campaign
- status - Campaign status
- callerid - Caller ID
- startingdate - Starting date of the Campaign
- expirationdate - Expiry date of the Campaign
- daily_start_time - Start time
- daily_stop_time - End time
- week_day_setting (monday, tuesday, wednesday, thursday, friday, saturday, sunday)
- frequency - Frequency, speed of the campaign. number of calls/min
- callmaxduration - Max call duration allowed
- maxretry - Max retry allowed per user
- intervalretry - Time to wait between retries in seconds
- completion_maxretry - Number of retries until a contact completes survey
- completion_intervalretry - Time delay in seconds before retrying contact to complete survey
- calltimeout - Number of seconds to timeout on dialing
- aleg_gateway - Gateway to use to reach the contact
- extra_data - Additional data to pass to the application
- totalcontact - Total Contact for this campaign
- completed - Total Contact that completed Call / Survey
- has_been_started - campaign started flag
- has_been_duplicated - campaign duplicated flag
- voicemail - Enable Voicemail Detection
- amd_behavior - Detection Behaviour
- sms_gateway - Gateway to transport the SMS

Relationships:

- content_type - Defines the application (survey) to use when the call is established on the A-Leg
- object_id - Defines the object of content_type application
- content_object - Used to define the Voice App or the Survey with generic ForeignKey
- phonebook - Many-To-Many relationship to the Phonebook model.
- user - Foreign key relationship to the a User model. Each campaign assigned to a User

- `voicemail_audiofile` - Foreign key relationship to the a AudioFile model.
- `dnc` - Foreign key relationship to the a DNC model.

Name of DB table: dialer_campaign

`get_active_callmaxduration``()`
Get the active call max duration

`get_active_contact``()`
Get all the active Contacts from the phonebook

`get_active_max_frequency``()`
Get the active max frequency

`get_campaign_type``()`
Get campaign type

`get_pending_subscriber_update``(`**args*, ***kwargs*`)`
Get all the pending subscribers from the campaign

`is_authorized_contact``(`*dialersetting*, *str_contact*`)`
Check if a contact is authorized

`progress_bar``()`
Progress bar generated based on no of contacts

`subscriber_detail``()`
This will link to subscribers who are associated with the campaign

`update_campaign_status``()`
Update the campaign's status

For example, If campaign is active, you can change status to 'Pause' or 'Stop'

8.4.5 Subscriber

class `dialer_campaign.models.`**`Subscriber`**`(`**args*, ***kwargs*`)`
This defines the Contact imported to a Campaign

Attributes:

- `last_attempt` - last call attempt date
- `count_attempt` - Count the amount of call attempt
- `completion_count_attempt` - Count the amount of attempt to call in order to achieve completion
- `duplicate_contact` - copy of the contact phonenumber
- `status` - subscriber status

Relationships:

- `contact` - Foreign key relationship to the Contact model.
- `campaign` - Foreign key relationship to the Campaign model.

Name of DB table: dialer_subscriber

8.4.6 Callrequest

class dialer_cdr.models.**Callrequest**(**args*, ***kwargs*)

This defines the call request, the dialer will read any new request and attempt to deliver the call.

Attributes:

- request_uuid - Unique id
- call_time - Total call time
- call_type - Call type
- status - Call request status
- callerid - Caller ID
- last_attempt_time -
- result –
- timeout -
- timelimit -
- extra_dial_string -
- phone_number -
- parent_callrequest -
- extra_data -
- num_attempt -
- hangup_cause -

Relationships:

- user - Foreign key relationship to the User model. Each campaign assigned to a User
- content_type - Defines the application (voip_app or survey) to use when the call is established on the A-Leg
- object_id - Defines the object of content_type application
- content_object - Used to define the VoIP App or the Survey with generic ForeignKey
- aleg_gateway - Foreign key relationship to the Gateway model. Gateway to use to call the subscriber
- subscriber - Foreign key relationship to the Subscriber Model.
- campaign - Foreign key relationship to the Campaign model.

Name of DB table: dialer_callrequest

8.4.7 VoIPCall

class dialer_cdr.models.**VoIPCall**(**args*, ***kwargs*)

This gives information of all the calls made with the carrier charges and revenue of each call.

Attributes:

- callid - callid of the phonecall
- callerid - CallerID used to call out

- phone_number - Phone number contacted
- dialcode - Dialcode of the phonenumber
- starting_date - Starting date of the call
- duration - Duration of the call
- billsec -
- progresssec -
- answersec -
- waitsec -
- disposition - Disposition of the call
- hangup_cause -
- hangup_cause_q850 -

Relationships:

- user - Foreign key relationship to the User model.
- used_gateway - Foreign key relationship to the Gateway model.
- callrequest - Foreign key relationship to the Callrequest model.

Name of DB table: dialer_cdr

destination_name()
Return Recipient dialcode

min_duration()
Return duration in min & sec

8.4.8 Gateway

class dialer_gateway.models.**Gateway**(**args*, ***kwargs*)
This defines the trunk to deliver the Voip Calls. Each of the Gateways are routes that support different protocols and sets of rules to alter the dialed number.

Attributes:

- name - Gateway name.
- description - Description about the Gateway.
- addprefix - Add prefix.
- removeprefix - Remove prefix.
- gateways - “user/,user/”, # Gateway string to try dialing separated by comma. First in the list will be tried first
- gateway_codecs - “‘PCMA,PCMU’,’PCMA,PCMU”’, # Codec string as needed by FS for each gateway separated by comma
- gateway_timeouts - “10,10”, # Seconds to timeout in string for each gateway separated by comma
- gateway_retries - “2,1”, # Retry String for Gateways separated by comma, on how many times each gateway should be retried
- originate_dial_string - originate_dial_string

- `secondused` -
- `failover` -
- `addparameter` -
- `count_call` -
- `count_in_use` -
- `maximum_call` -
- `status` - Gateway status

Name of DB table: dialer_gateway

8.4.9 `DialerSetting`

class `dialer_settings.models.`**`DialerSetting`**`(`**args*, ***kwargs*`)`

This defines the settings to apply to a user

Attributes:

- `name` - Settings name.
- `max_frequency` - Max frequency, speed of the campaign. This is the number of calls per minute.
- `callmaxduration` - Max retries allowed
- `maxretry` - Max retries allowed per user
- `max_calltimeout` - Maximum number of seconds to timeout on calls
- `max_cpg` - Max Number of campaigns
- `max_subr_cpg` - Max Number of subscriber
- `blacklist` - Used to blacklist phone numbers to be called
- `whitelist` - Used to whitelist phone numbers to be called

Name of DB table: dialer_setting

8.4.10 `Survey_abstract`

class `survey.models.`**`Survey_abstract`**`(`**args*, ***kwargs*`)`

This defines the Survey template

Attributes:

- `name` - survey name.
- `description` - description about the survey.

Relationships:

- `user` - Foreign key relationship to the User model. Each survey is assigned to a User

Name of DB table: survey

8.4.11 Survey_template

class survey.models.**Survey_template**(**args*, ***kwargs*)
This defines the Survey template

copy_survey_template(*campaign_id=None*)
copy survey template to survey when starting campaign

8.4.12 Survey

class survey.models.**Survey**(**args*, ***kwargs*)
This defines the Survey

create_duplicate_survey(*campaign_obj*, *new_campaign*)
create duplicate survey

8.4.13 Section_abstract

class survey.models.**Section_abstract**(**args*, ***kwargs*)
This defines the question for survey

Attributes:

- type - section type
- question - question
- script - text that will be used for TTS
- audiofile - audio file to be use instead of TTS
- invalid_audiofile - audio to play when input is invalid
- retries - amount of time to retry to get a valid input
- timeout - time to wait for user input
- key_0 - on multi choice section, text for result on key 0
- key_1 - on multi choice section, text for result on key 1
- key_2 - on multi choice section, text for result on key 2
- key_3 - on multi choice section, text for result on key 3
- key_4 - on multi choice section, text for result on key 4
- key_5 - on multi choice section, text for result on key 5
- key_6 - on multi choice section, text for result on key 6
- key_7 - on multi choice section, text for result on key 7
- key_8 - on multi choice section, text for result on key 8
- key_9 - on multi choice section, text for result on key 9
- rating_laps - From 1 to X, value to accept rating input
- validate_number - check if we want to valid the input on Enter Number section
- number_digits - Number of digits to wait for on Enter Number section
- min_number - if validate_number the minimum number accepted

- `max_number` - if validate_number the maximum number accepted
- `phonenumber` - phonenumber to dialout / call transfer
- `confirm_script` - script that will be play to the called part on transfer
- `confirm_key` - key to confirm the call t
- `completed` - reaching this section will mark the subscriber as completed
- `conference` - Conference Room
- `sms_text` - text to send via SMS

Relationships:

- `survey` - Foreign key relationship to the Survey model. Each survey question is assigned to a Survey
- `audio_message` - Foreign key relationship to the AudioFile model.

Name of DB table: survey_question

`build_dtmf_filter`()
Build the dtmf filter to capture digits

`get_branching_count_per_section`()
Get branching count per section

`sortable_by`
alias of *`Survey`*

8.4.14 `Section_template`

class `survey.models.`**`Section_template`**(**args*, ***kwargs*)
This defines the question for survey section template

`copy_section_branching_template`(*section*, *new_survey_obj*)
copy section template to section when starting campaign

`copy_section_template`(*new_survey_obj*)
copy section template to section when starting campaign

8.4.15 `Section`

class `survey.models.`**`Section`**(**args*, ***kwargs*)
This defines the question for survey section

8.4.16 `Branching_abstract`

class `survey.models.`**`Branching_abstract`**(**args*, ***kwargs*)
This defines the response of the survey section

Attributes:

- `keys` - Key digit (DTMF entered by the calling party)

Relationships:

- `section` - Foreign key relationship to the Section. Each response is assigned to a Section

8.4.17 Branching_template

class survey.models.**Branching_template**(**args*, ***kwargs*)

This defines the response of the survey section

copy_branching_template(*new_section*, *new_survey_obj*)

copy branching template to branching when starting campaign

8.4.18 Branching

class survey.models.**Branching**(**args*, ***kwargs*)

This defines the response of the survey section

8.4.19 Result

class survey.models.**Result**(**args*, ***kwargs*)

This gives survey result

That will be difficult to scale for reporting One big issue is when the user update the survey in time, we need to keep an history somehow of the question/response

Ideally we can try to build 2 other table, survey_track_question (id, question_text), survey_track_response (id, response_text) Where question_text / response_text is unique

Attributes:

- callrequest - Call Request
- section - survey question
- response - survey question's response

Relationships:

- campaign - Foreign key relationship to the Campaign model. Each survey result is belonged to a Campaign
- survey - Foreign key relationship to the Survey model. Each survey question is assigned to a Survey
- section - Foreign key relationship to the Section model. Each result is assigned to a Section

Name of DB table: result

8.4.20 ResultAggregate

class survey.models.**ResultAggregate**(**args*, ***kwargs*)

This gives survey result aggregate, used to display survey result in a more efficient way

Name of DB table: result_aggregate

8.4.21 UserProfile

class user_profile.models.**UserProfile**(**args*, ***kwargs*)

This defines extra features for the user

Relationships:

- userprofile_gateway - ManyToMany

- `dialersetting` - Foreign key relationship to the DialerSetting model.

Name of DB table: user_profile

8.4.22 `CalendarSetting`

class `user_profile.models.`**`CalendarSetting`**(**args*, ***kwargs*)

This defines the Calender settings to apply to a ar_user

Attributes:

- `label` - Label for the Calendar Setting
- `callerid` - CallerID number
- `caller_name` - Caller name
- `call_timeout` - call timeout
- `survey` - Foreign key relationship to the Survey
- `aleg_gateway` - Foreign key relationship to the Gateway model. Gateway to use to call the subscriber
- `sms_gateway` - Gateway to transport the SMS
- `voicemail` - Enable Voicemail Detection
- `amd_behavior` - Detection Behaviour

Relationships:

- `user` - Foreign key relationship to the a User model. CalendarSetting are assigned to a User
- `voicemail_audiofile` - Foreign key relationship to the a AudioFile model.

Name of DB table: calendar_setting

8.4.23 `CalendarUser`

class `user_profile.models.`**`CalendarUser`**(**args*, ***kwargs*)

Calendar User Model

8.4.24 `CalendarUserProfile`

class `user_profile.models.`**`CalendarUserProfile`**(**args*, ***kwargs*)

This defines extra features for the AR_user

Attributes:

- `calendar_setting` - appointment reminder settings

Name of DB table: calendar_user_profile

8.4.25 `Calendar`

class `appointment.models.calendars.`**`Calendar`**(**args*, ***kwargs*)

This is for grouping events so that batch relations can be made to all events. An example would be a project calendar.

name: the name of the calendar events: all the events contained within the calendar. >>> calendar = Calendar(name = 'Test Calendar') >>> calendar.save()

get_recent (*amount=5*, *in_datetime=<built-in method now of type object>*, *tzinfo=<UTC>*)
This shortcut function allows you to get events that have started recently.

amount is the amount of events you want in the queryset. The default is 5.

in_datetime is the datetime you want to check against. It defaults to datetime.datetime.now

8.4.26 Event

class appointment.models.events.**Event** (**args*, ***kwargs*)
This model stores meta data for a event

copy_event (*next_occurrence*)
create new event with next occurrence

get_list_alarm ()
we will list alarm of an event

get_list_child ()
we will list childs of an event

get_next_occurrence ()
TODO: implement this

```
>>> rule = Rule(frequency="MONTHLY", name="Monthly")
>>> rule.save()
>>> event = Event(rule=rule, start=datetime.datetime(2008,1,1,tzinfo=pytz.utc), end=datetime
>>> event.rule
<Rule: Monthly>
>>> event.get_next_occurrence()
2008-02-02 00:00:00+00:00
```

get_occurrences (*start*, *end*)

```
>>> rule = Rule(frequency="MONTHLY", name="Monthly")
>>> rule.save()
>>> event = Event(rule=rule, start=datetime.datetime(2008,1,1,tzinfo=pytz.utc), end=datetime
>>> event.rule
<Rule: Monthly>
>>> occurrences = event.get_occurrences(datetime.datetime(2008,1,24), datetime.datetime(2008
>>> ["%s to %s" %(o.start, o.end) for o in occurrences]
['2008-02-01 00:00:00+00:00 to 2008-02-02 00:00:00+00:00',
 '2008-03-01 00:00:00+00:00 to 2008-03-02 00:00:00+00:00']
```

Ensure that if an event has no rule, that it appears only once.

```
>>> event = Event(start=datetime.datetime(2008,1,1,8,0), end=datetime.datetime(2008,1,1,9,0)
>>> occurrences = event.get_occurrences(datetime.datetime(2013,1,24), datetime.datetime(2014
>>> ["%s to %s" %(o.start, o.end) for o in occurrences]
[]
```

occurrences_after (*after=None*)
returns a generator that produces occurrences after the datetime `after`. Includes all of the persisted Occurrences.

update_last_child_status (*status*)
we will search for the last created child of an event and update his status to the status value

Note for the integrators: We can pause an event for 12hours but after that we will have to stop and create the new event.

8.4.27 Alarm

class appointment.models.alarms.**Alarm**(**args*, ***kwargs*)

This is for Alarms / Reminders on events models.

copy_alarm(*new_event*, *next_occurrence*)

Create a copy of the Alarm

retry_alarm()

Task to check if Alarm needs to be respooled after it failed

8.4.28 SMSCampaign

class mod_sms.models.**SMSCampaign**(**args*, ***kwargs*)

This defines the SMSCampaign

Attributes:

- campaign_code - Auto-generated campaign code to identify the campaign
- name - Campaign name
- description - Description about the Campaign
- status - Campaign status
- callerid - Caller ID
- startingdate - Starting date of the Campaign
- expirationdate - Expiry date of the Campaign
- daily_start_time - Start time
- daily_stop_time - End time
- week_day_setting (monday, tuesday, wednesday, thursday, friday, saturday, sunday)
- frequency - Frequency, speed of the campaign. number of calls/min
- maxretry - Max retry allowed per user
- intervalretry - Time to wait between retries in seconds
- sms_gateway - Gateway to transport the SMS
- extra_data - Additional data to pass to the application

Relationships:

- content_type - Defines the application (voice_app or survey) to use when the call is established on the A-Leg
- object_id - Defines the object of content_type application
- content_object - Used to define the Voice App or the Survey with generic ForeignKey
- phonebook - Many-To-Many relationship to the Phonebook model.
- user - Foreign key relationship to the a User model. Each campaign assigned to a User

Name of DB table: sms_campaign

common_sms_campaign_status(*status*)
SMS Campaign Status (e.g. start | stop | abort | pause) needs to be changed. It is a common function for the admin and customer UI's

Attributes:

- `status` - selected status for the sms campaign record

Logic Description:

- Selected SMS Campaign's status needs to be changed. Changed status can be start, stop or pause.
- This function is used by `update_sms_campaign_status_admin()` & `update_sms_campaign_status_cust()`

count_contact_of_phonebook(*status=None*)
Count the no. of Contacts in a phonebook

get_active_contact()
Get all the active Contacts from the phonebook

get_active_contact_no_subscriber()
List of active contacts that do not exist in Campaign Subscriber

get_active_max_frequency()
Get the active max frequency

get_pending_subscriber(*limit=1000*)
Get all the pending subscribers from the sms_campaign

get_pending_subscriber_update(*limit=1000*, *status=6*)
Get all the pending subscribers from the campaign

is_authorized_contact(*str_contact*)
Check if a contact is authorized

progress_bar()
Progress bar generated based on no of contacts

sms_campaignsubscriber_detail()
This will link to sms_campaign subscribers who are associated with the sms_campaign

update_sms_campaign_status()
Update the sms_campaign's status

For example, If campaign is active, you can change status to 'Pause' or 'Stop'

8.4.29 SMSCampaignSubscriber

class `mod_sms.models.`**SMSCampaignSubscriber**(**args*, ***kwargs*)
This defines the Contact imported to a SMSCampaign

Attributes:

- `last_attempt` -
- `count_attempt` -
- `duplicate_contact` -
- `status` -

Relationships:

- `contact` - Foreign key relationship to the Contact model.

- `campaign` - Foreign key relationship to the Campaign model.

Name of DB table: sms_campaign_subscriber

8.4.30 `SMSMessage`

class `mod_sms.models.`**`SMSMessage`**`(`**args*, ***kwargs*`)`

extension on Message

Attributes:

Relationships:

- `message` - One to one relationship to the Message model.
- `sms_campaign` - Foreign key relationship to the SMSCampaign model.

Name of DB table: message_smscampaign

8.4.31 `SMSTemplate`

class `mod_sms.models.`**`SMSTemplate`**`(`**args*, ***kwargs*`)`

This table store the SMS Template

8.5 Newfies-Dialer Views

8.5.1 `index`

`frontend.views.`**`index`**`(`*request*`)`

Index view of the Customer Interface

Attributes:

- `form` - LoginForm
- `template` - frontend/index.html

8.5.2 `customer_dashboard`

`frontend.views.`**`customer_dashboard`**`(`*request*, **args*, ***kwargs*`)`

Customer dashboard gives the following information

- Total Campaigns contacts
- Amount of contact reached today
- Disposition of calls via pie chart
- Call records & Duration of calls are shown on graph by days/hours

Attributes:

- `template` - frontend/dashboard.html
- `form` - DashboardForm

8.5.3 login_view

frontend.views.**login_view**(*request*)

Check User credentials

Attributes:

- `form` - LoginForm
- `template` - frontend/index.html

Logic Description:

- Submitted user credentials need to be checked. If it is not valid then the system will redirect to the login page.
- If submitted user credentials are valid then system will redirect to the dashboard.

8.5.4 cust_password_reset

8.5.5 cust_password_reset_done

8.5.6 cust_password_reset_confirm

8.5.7 phonebook_list

dialer_contact.views.**phonebook_list**(*request*, **args*, ***kwargs*)

Phonebook list for the logged in user

Attributes:

- `template` - dialer_contact/phonebook/list.html

Logic Description:

- List all phonebooks which belong to the logged in user.

8.5.8 phonebook_add

dialer_contact.views.**phonebook_add**(*request*, **args*, ***kwargs*)

Add new Phonebook for the logged in user

Attributes:

- `form` - PhonebookForm
- `template` - dialer_contact/phonebook/change.html

Logic Description:

- Add a new phonebook which will belong to the logged in user via the phonebookForm & get redirected to the phonebook list

8.5.9 phonebook_change

`dialer_contact.views.`**`phonebook_change`**`(`*request*, **args*, ***kwargs*`)`

Update/Delete Phonebook for the logged in user

Attributes:

- `object_id` - Selected phonebook object
- `form` - PhonebookForm
- `template` - dialer_contact/phonebook/change.html

Logic Description:

- Update/delete selected phonebook from the phonebook list via PhonebookForm & get redirected to phonebook list

8.5.10 contact_list

`dialer_contact.views.`**`contact_list`**`(`*request*, **args*, ***kwargs*`)`

Contact list for the logged in user

Attributes:

- `template` - dialer_contact/contact/list.html
- `form` - ContactSearchForm

Logic Description:

- List all contacts from phonebooks belonging to the logged in user

8.5.11 contact_add

`dialer_contact.views.`**`contact_add`**`(`*request*, **args*, ***kwargs*`)`

Add a new contact into the selected phonebook for the logged in user

Attributes:

- `form` - ContactForm
- `template` - dialer_contact/contact/change.html

Logic Description:

- Before adding a contact, check dialer setting limit if applicable to the user.
- Add new contact belonging to the logged in user via ContactForm & get redirected to the contact list

8.5.12 contact_change

`dialer_contact.views.`**`contact_change`**`(`*request*, **args*, ***kwargs*`)`

Update/Delete contact for the logged in user

Attributes:

- `object_id` - Selected contact object
- `form` - ContactForm
- `template` - dialer_contact/contact/change.html

Logic Description:

- Update/delete selected contact from the contact list via ContactForm & get redirected to the contact list

8.5.13 `contact_import`

`dialer_contact.views.`**`contact_import`**`(`*request*, **args*, ***kwargs*`)`

Import CSV file of Contacts for the logged in user

Attributes:

- `form` - Contact_fileImport
- `template` - dialer_contact/contact/import_contact.html

Logic Description:

- Before adding contacts, check dialer setting limit if applicable to the user.
- Add new contacts which will belong to the logged in user via csv file & get the result (upload success and failure statistics)

Important variable:

- total_rows - Total no. of records in the CSV file
- retail_record_count - No. of records imported from the CSV file

8.5.14 `campaign_list`

`dialer_campaign.views.`**`campaign_list`**`(`*request*, **args*, ***kwargs*`)`

List all campaigns for the logged in user

Attributes:

- `template` - dialer_campaign/campaign/list.html

Logic Description:

- List all campaigns belonging to the logged in user

8.5.15 `campaign_add`

`dialer_campaign.views.`**`campaign_add`**`(`*request*, **args*, ***kwargs*`)`

Add a new campaign for the logged in user

Attributes:

- `form` - CampaignForm
- `template` - dialer_campaign/campaign/change.html

Logic Description:

- Before adding a campaign, check dialer setting limit if applicable to the user.

- Add the new campaign which will belong to the logged in user via CampaignForm & get redirected to campaign list

8.5.16 `campaign_change`

`dialer_campaign.views.`**`campaign_change`**`(`*request*, **args*, ***kwargs*`)`

Update/Delete campaign for the logged in user

Attributes:

- `object_id` - Selected campaign object
- `form` - CampaignForm
- `template` - dialer_campaign/campaign/change.html

Logic Description:

- Update/delete selected campaign from the campaign list via CampaignForm & get redirected to the campaign list

8.5.17 `survey_list`

`survey.views.`**`survey_list`**`(`*request*, **args*, ***kwargs*`)`

SurveyApp list for the logged in user

Attributes:

- `template` - survey/list.html

Logic Description:

- List all surveys which belong to the logged in user.

8.5.18 `survey_add`

`survey.views.`**`survey_add`**`(`*request*, **args*, ***kwargs*`)`

Add new Survey for the logged in user

Attributes:

- `form` - SurveyForm
- `template` - survey/change.html

Logic Description:

- Add a new survey which will belong to the logged in user via the SurveyForm & get redirected to the survey list

8.5.19 `survey_change`

`survey.views.`**`survey_change`**`(`*request*, **args*, ***kwargs*`)`

Update/Delete Survey for the logged in user

Attributes:

- `object_id` - Selected survey object

- `form` - SurveyForm
- `template` - survey/change.html

Logic Description:

- Update/delete selected survey from the survey list via SurveyForm & get redirected to survey list

8.5.20 `survey_report`

`survey.views.`**`survey_report`**`(`*request*, **args*, ***kwargs*`)`

Survey detail report for the logged in user

Attributes:

- `template` - survey/survey_report.html
- `form` - SurveyDetailReportForm

Logic Description:

- List all survey_report which belong to the logged in user.

8.5.21 `audio_list`

`dialer_audio.views.`**`audio_list`**`(`*request*, **args*, ***kwargs*`)`

AudioFile list for the logged in user

Attributes:

- `template` - dialer_audio/audio_list.html

Logic Description:

- List all audios which belong to the logged in user.

8.5.22 `audio_add`

`dialer_audio.views.`**`audio_add`**`(`*request*, **args*, ***kwargs*`)`

Add new Audio for the logged in user

Attributes:

- `form` - SurveyCustomerAudioFileForm
- `template` - dialer_audio/audio_change.html

Logic Description:

- Add a new audio which will belong to the logged in user via the CustomerAudioFileForm & get redirected to the audio list

8.5.23 `audio_change`

`dialer_audio.views.`**`audio_change`**`(`*request*, **args*, ***kwargs*`)`

Update Audio for the logged in user

Attributes:

- `form` - SurveyCustomerAudioFileForm
- `template` - dialer_audio/audio_change.html

Logic Description:

- Update audio which is belong to the logged in user via the CustomerAudioFileForm & get redirected to the audio list

8.5.24 `calendar_setting_list`

`appointment.views.`**`calendar_setting_list`**`(`*request*, **args*, ***kwargs*`)`

Calendar setting list for the logged in user

Attributes:

- `template` - appointment/calendar_setting/list.html

Logic Description:

- List all calendar settings which belong to the logged in user.

8.5.25 `calendar_setting_add`

`appointment.views.`**`calendar_setting_add`**`(`*request*, **args*, ***kwargs*`)`

Add a new calendar setting for the logged in user

Attributes:

- `form` - CalendarSettingForm
- `template` - appointment/calendar_setting/change.html

Logic Description:

- Add new calendar_setting belonging to the logged in user via ContactSettingForm & get redirected to the calendar_setting list

8.5.26 `calendar_setting_del`

`appointment.views.`**`calendar_setting_del`**`(`*request*, **args*, ***kwargs*`)`

Delete calendar_setting for the logged in user

Attributes:

- `object_id` - Selected calendar_setting object
- `object_list` - Selected calendar_setting objects

Logic Description:

- Delete selected calendar_setting from the calendar_setting list

8.5.27 `calendar_setting_change`

`appointment.views.`**`calendar_setting_change`**`(`*request*, **args*, ***kwargs*`)`

Update/Delete calendar_setting for the logged in user

Attributes:

- `object_id` - Selected calendar_setting object
- `form` - CalendarSettingForm
- `template` - appointment/calendar_setting/change.html

Logic Description:

- Update/delete selected calendar_setting from the calendar_setting list via CalendarSettingForm & get redirected to the calendar_setting list

8.5.28 `calendar_user_add`

`appointment.views.`**`calendar_user_add`**(*request*, **args*, ***kwargs*)

Add new calendar user for the logged in manager

Attributes:

- `form` - CalendarUserCreationForm
- `template` - appointment/calendar_user/change.html

Logic Description:

- Add a new calendar user which will belong to the logged in manager via the UserCreationForm & get redirected to the calendar user list

8.5.29 `calendar_user_change`

`appointment.views.`**`calendar_user_change`**(*request*, **args*, ***kwargs*)

Update/Delete calendar user for the logged in manager

Attributes:

- `object_id` - Selected calendar_user object
- `form` - CalendarUserChangeDetailExtendForm, CalendarUserNameChangeForm
- `template` - appointment/calendar_user/change.html

Logic Description:

- Update/delete selected calendar user from the calendar_user list via CalendarUserChangeDetailExtendForm & get redirected to calendar_user list

8.5.30 `calendar_user_del`

`appointment.views.`**`calendar_user_del`**(*request*, **args*, ***kwargs*)

Delete a calendar_user for a logged in manager

Attributes:

- `object_id` - Selected calendar_user object
- `object_list` - Selected calendar_user objects

Logic Description:

- Delete calendar_user from a calendar_user list.

8.5.31 calendar_user_change_pw

appointment.views.**calendar_user_change_pw**(*request*, **args*, ***kwargs*)

CalendarUser Detail change

Attributes:

- form - CalendarUserPasswordChangeForm
- **template - 'appointment/calendar_user/change_password.html',** 'frontend/registration/user_detail_change.html'

Logic Description:

- Reset calendar_user password.

8.5.32 calendar_list

appointment.views.**calendar_list**(*request*, **args*, ***kwargs*)

Calendar list for the logged in user

Attributes:

- template - appointment/calendar/list.html

Logic Description:

- List all calendars which belong to the logged in user.

8.5.33 calendar_add

appointment.views.**calendar_add**(*request*, **args*, ***kwargs*)

Add a new calendar for the logged in user

Attributes:

- form - CalendarForm
- template - appointment/calendar/change.html

Logic Description:

- Add new contact belonging to the logged in user via ContactForm & get redirected to the contact list

8.5.34 calendar_del

appointment.views.**calendar_del**(*request*, **args*, ***kwargs*)

Delete calendar for the logged in user

Attributes:

- object_id - Selected calendar object
- object_list - Selected calendar objects

Logic Description:

- Delete selected calendar from the calendar list

8.5.35 `calendar_change`

`appointment.views.`**`calendar_change`**`(`*request*, **args*, ***kwargs*`)`

Update/Delete calendar for the logged in user

Attributes:

- `object_id` - Selected calendar object
- `form` - CalendarForm
- `template` - appointment/calendar/change.html

Logic Description:

- Update/delete selected calendar from the calendar list via CalendarForm & get redirected to the calendar list

8.5.36 `event_list`

`appointment.views.`**`event_list`**`(`*request*, **args*, ***kwargs*`)`

Event list for the logged in user

Attributes:

- `template` - appointment/event/list.html

Logic Description:

- List all events which belong to the logged in user.

8.5.37 `event_add`

`appointment.views.`**`event_add`**`(`*request*, **args*, ***kwargs*`)`

Add a new event for the logged in user

Attributes:

- `form` - EventForm
- `template` - appointment/event/change.html

Logic Description:

- Add new event belonging to the logged in user via EventForm & get redirected to the event list

8.5.38 `event_del`

`appointment.views.`**`event_del`**`(`*request*, **args*, ***kwargs*`)`

Delete event for the logged in user

Attributes:

- `object_id` - Selected event object
- `object_list` - Selected event objects

Logic Description:

- Delete selected event from the event list

8.5.39 event_change

appointment.views.**event_change**(*request*, **args*, ***kwargs*)

Update/Delete event for the logged in user

Attributes:

- object_id - Selected event object
- form - EventForm
- template - appointment/event/change.html

Logic Description:

- Update/delete selected event from the event list via EventForm & get redirected to the event list

8.5.40 alarm_list

appointment.views.**alarm_list**(*request*, **args*, ***kwargs*)

Alarm list for the logged in user

Attributes:

- template - appointment/alarm/list.html

Logic Description:

- List all alarms which belong to the logged in user.

8.5.41 alarm_add

appointment.views.**alarm_add**(*request*, **args*, ***kwargs*)

Add a new alarm for the logged in user

Attributes:

- form - AlarmForm
- template - appointment/alarm/change.html

Logic Description:

- Add new alarm belonging to the logged in user via AlarmForm & get redirected to the alarm list

8.5.42 alarm_del

appointment.views.**alarm_del**(*request*, **args*, ***kwargs*)

Delete alarm for the logged in user

Attributes:

- object_id - Selected alarm object
- object_list - Selected alarm objects

Logic Description:

- Delete selected alarm from the alarm list

8.5.43 alarm_change

appointment.views.**alarm_change**(*request*, **args*, ***kwargs*)

Update/Delete alarm for the logged in user

Attributes:

- object_id - Selected alarm object
- form - AlarmForm
- template - appointment/alarm/change.html

Logic Description:

- Update/delete selected alarm from the alarm list via AlarmForm & get redirected to the alarm list

8.5.44 sms_campaign_list

mod_sms.views.**sms_campaign_list**(*request*, **args*, ***kwargs*)

List all sms campaigns for the logged in user

Attributes:

- template - mod_sms/list.html

Logic Description:

- List all sms campaigns belonging to the logged in user

8.5.45 sms_campaign_add

mod_sms.views.**sms_campaign_add**(*request*, **args*, ***kwargs*)

Add a new sms campaign for the logged in user

Attributes:

- form - SMSCampaignForm
- template - mod_sms/change.html

Logic Description:

- Before adding a sms campaign, check dialer setting limit if applicable to the user.
- Add the new sms campaign which will belong to the logged in user via SMSCampaignForm & get redirected to sms campaign list

8.5.46 sms_campaign_del

mod_sms.views.**sms_campaign_del**(*request*, **args*, ***kwargs*)

Delete/Stop sms campaign for the logged in user

Attributes:

- object_id - Selected sms campaign object
- object_list - Selected sms campaign objects

Logic Description:

- Delete/Stop the selected sms campaign from the sms campaign list

8.5.47 `sms_campaign_duplicate`

`mod_sms.views.`**`sms_campaign_duplicate`**(*request*, **args*, ***kwargs*)

Duplicate sms campaign via DuplicateSMSCampaignForm

Attributes:

- `id` - Selected sms campaign object
- `form` - DuplicateSMSCampaignForm
- `template` - mod_sms/sms_campaign_duplicate.html

8.5.48 `sms_campaign_text_message`

`mod_sms.views.`**`sms_campaign_text_message`**(*request*, **args*, ***kwargs*)

Get sms campaign's text message

Attributes:

- `object_id` - Selected sms campaign object
- `template` - mod_sms/sms_campaign_text_message.html

8.5.49 `update_sms_campaign_status_cust`

`mod_sms.views.`**`update_sms_campaign_status_cust`**(*request*, **args*, ***kwargs*)

SMS Campaign Status (e.g. start|stop|pause|abort) can be changed from customer interface (via sms campaign list)

8.5.50 `sms_campaign_change`

`mod_sms.views.`**`sms_campaign_change`**(*request*, **args*, ***kwargs*)

Update/Delete sms campaign for the logged in user

Attributes:

- `object_id` - Selected campaign object
- `form` - SMSCampaignForm
- `template` - mod_sms/change.html

Logic Description:

- Update/delete selected sms campaign from the sms campaign list via SMSCampaignForm & get redirected to the sms campaign list

8.5.51 `sms_dashboard`

`mod_sms.views.`**`sms_dashboard`**(*request*, **args*, ***kwargs*)

SMS dashboard gives the following information

- No of SMSCampaigns for logged in user

- Total phonebook contacts
- Total SMSCampaigns contacts
- Amount of contact reached today
- Disposition of sms via pie chart
- SMS count shown on graph by days/hours

Attributes:

- `template` - mod_sms/sms_dashboard.html
- `form` - SMSDashboardForm

8.5.52 `sms_report`

`mod_sms.views.`**`sms_report`**`(`*request*, **args*, ***kwargs*`)`

SMS Report

Attributes:

- `form` - SMSSearchForm
- `template` - mod_sms/sms_report.html

Logic Description:

- Get SMS list according to search parameters for logged-in user

Important variable:

- `request.session['sms_record_kwargs']` - stores sms kwargs

8.5.53 `export_sms_report`

`mod_sms.views.`**`export_sms_report`**`(`*request*, **args*, ***kwargs*`)`

Export CSV file of SMS record

Important variable:

- `request.session['sms_record_kwargs']` - stores sms query set

Exported fields: ['sender', 'recipient_number', 'send_date', 'uuid', 'status', 'status_message', 'gateway']

8.6 Newfies-Dialer Admin Views

8.6.1 `CampaignAdmin`

class `dialer_campaign.admin.`**`CampaignAdmin`**`(`*model*, *admin_site*`)`

Allows the administrator to view and modify certain attributes of a Campaign.

`add_view``(`*request*, *extra_context=None*`)`

Override django add_view method for checking the dialer setting limit

Logic Description:

- Before adding campaign, check dialer setting limit if applicable to the user, if matched then the user will be redirected to the campaign list

8.6.2 `SubscriberAdmin`

class `dialer_campaign.admin.`**`SubscriberAdmin`**(*model*, *admin_site*)

Allows the administrator to view and modify certain attributes of a Subscriber.

`form`

alias of `SubscriberAdminForm`

`subscriber_report`(*request*)

Get subscriber report

Attributes:

- `form` - SubscriberReportForm
- `template` - admin/dialer_campaign/subscriber/subscriber_report.html

8.6.3 `PhonebookAdmin`

class `dialer_contact.admin.`**`PhonebookAdmin`**(*model*, *admin_site*)

Allows the administrator to view and modify certain attributes of a Phonebook.

8.6.4 `ContactAdmin`

class `dialer_contact.admin.`**`ContactAdmin`**(*model*, *admin_site*)

Allows the administrator to view and modify certain attributes of a Contact.

`add_view`(*request*, *extra_context=None*)

Override django admin add_view method for checking the dialer setting limit

Logic Description:

- Before adding a contact, check the dialer setting limit if applicable to the user. If matched, the user will be redirected to the contact list

`import_contact`(*request*)

Add custom method in django admin view to import CSV file of Contacts

Attributes:

- `form` - Contact_fileImport
- `template` - admin/dialer_campaign/contact/import_contact.html

Logic Description:

- Before adding contact, check the dialer setting limit if applicable to the user.
- Add a new contact which will belong to the logged in user via csv file & get the result (Upload success & failure statistics)

Important variable:

- total_rows - Total no. of records in the CSV file
- retail_record_count - No. of records which are imported from The CSV file

8.6.5 `CallrequestAdmin`

class `dialer_cdr.admin.`**`CallrequestAdmin`**(*model*, *admin_site*)

Allows the administrator to view and modify certain attributes of a Callrequest.

8.6.6 `VoIPCallAdmin`

class `dialer_cdr.admin.`**`VoIPCallAdmin`**(*model*, *admin_site*)

Allows the administrator to view and modify certain attributes of a VoIPCall.

`changelist_view`(*request*, *extra_context=None*)

Override changelist_view method of django-admin for search parameters

Attributes:

- `form` - AdminVoipSearchForm
- `template` - admin/dialer_cdr/voipcall/change_list.html

Logic Description:

- VoIP report Record Listing with search option & Daily Call Report search Parameters: by date, by status and by billed.

`export_voip_report`(*request*)

Export a CSV file of VoIP call records

Important variable:

- request.session['admin_voipcall_record_kwargs'] - stores voipcall kwargs

Exported fields: [user, callid, callerid, phone_number, starting_date, duration, disposition, used_gateway]

`has_add_permission`(*request*)

Remove add permission on VoIP Call Report model

Logic Description:

- Override django admin has_add_permission method to remove add permission on VoIP Call Report model

`used_gateway_link`(*obj*)

Used gateway link to edit gateway detail

`user_link`(*obj*)

User link to user profile

8.6.7 `GatewayAdmin`

class `dialer_gateway.admin.`**`GatewayAdmin`**(*model*, *admin_site*)

Allows the administrator to view and modify certain attributes of a Gateway.

8.6.8 `DialerSettingAdmin`

class `dialer_settings.admin.`**`DialerSettingAdmin`**(*model*, *admin_site*)

Allows the administrator to view and modify certain attributes of a DialerSetting.

8.6.9 SurveyAdmin

class survey.admin.**SurveyAdmin**(*model*, *admin_site*)
Allows the administrator to view and modify survey.

8.6.10 SectionAdmin

class survey.admin.**SectionAdmin**(*model*, *admin_site*)
Allows the administrator to view and modify survey question.

8.6.11 BranchingAdmin

class survey.admin.**BranchingAdmin**(*model*, *admin_site*)
Allows the administrator to view and modify branching.

8.6.12 ResultAdmin

class survey.admin.**ResultAdmin**(*model*, *admin_site*)
Allows the administrator to view and modify survey results.

8.6.13 ResultAggregate

class survey.admin.**ResultAggregate**(**args*, ***kwargs*)
This gives survey result aggregate, used to display survey result in a more efficient way

Name of DB table: result_aggregate

8.6.14 CalendarSettingAdmin

class calendar_settings.admin.**CalendarSettingAdmin**(*model*, *admin_site*)

8.6.15 CalendarUserAdmin

class appointment.admin.**CalendarUserAdmin**(*model*, *admin_site*)

8.6.16 CalendarAdmin

class appointment.admin.**CalendarAdmin**(*model*, *admin_site*)

8.6.17 EventAdmin

class appointment.admin.**EventAdmin**(*model*, *admin_site*)

form
alias of EventAdminForm

8.6.18 AlarmAdmin

class appointment.admin.**AlarmAdmin**(*model*, *admin_site*)

8.7 Newfies Tasks

dialer_cdr.**tasks**
alias of *dialer_cdr.tasks*

8.8 Newfies Signals

8.8.1 post_save_add_contact

dialer_campaign.models.**post_save_add_contact**(*sender*, ***kwargs*)
A post_save signal is sent by the Contact model instance whenever it is going to save.

Logic Description:

- When new contact is added into Contact model, active the campaign list will be checked with the contact status.
- If the active campaign list count is more than one & the contact is active, the contact will be added into Subscriber model.

8.9 Test Case Descriptions

8.9.1 Requirement

Run/Start Celery:

```
$ /etc/init.d/celery start
```

or:

```
$ python manage.py celeryd -l info
```

Run/Start Redis:

```
$ /etc/init.d/redis-server start
```

8.9.2 How to run test

1. Run Full Test Suit:

```
$ python manage.py test
```

2. Run Individual Test:

```
$ python manage.py test api

$ python manage.py test dialer_campaign

$ python manage.py test dialer_cdr

$ python manage.py test dialer_audio

$ python manage.py test dialer_gateway

$ python manage.py test dialer_settings

$ python manage.py test frontend

$ python manage.py test survey

$ python manage.py test user_profile

$ python manage.py test appointment

$ python manage.py test mod_sms
```

Model Test Case

class `dialer_contact.tests.`**`DialerContactModel`**(*methodName=’runTest’*)
Test Phonebook, Contact models

class `dialer_campaign.tests.`**`DialerCampaignModel`**(*methodName=’runTest’*)
Test Campaign, Subscriber models

class `dialer_cdr.tests.`**`DialerCdrModel`**(*methodName=’runTest’*)
Test Callrequest, VoIPCall models

class `dialer_settings.tests.`**`DialerSettingModel`**(*methodName=’runTest’*)
Test DialerSetting model

class `dialer_gateway.tests.`**`GatewayModel`**(*methodName=’runTest’*)
Test Gateway model

class `survey.tests.`**`SurveyModel`**(*methodName=’runTest’*)
Test Survey, Section, Branching, Result, ResultAggregate Model

8.10 Javascript Files

- jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is designed to change the way that you write JavaScript.
- Bootstrap is sleek, intuitive, and powerful front-end framework for faster and easier web development.
- Bootbox is a small JavaScript library which allows you to create programmatic dialog boxes using Twitter’s Bootstrap modals, without having to worry about creating, managing or removing any of the required DOM elements or JS event handlers.
- Bootstrap-datepicker Datepicker for Bootstrap

CHAPTER 9

API Reference

Contents:

9.1 *AlarmRequestSerializer*

9.2 *AlarmSerializer*

9.3 *AudioFileSerializer*

class `apirest.audiofile_serializers.`**`AudioFileSerializer`**(*instance=None, data=None, files=None, context=None, partial=False, many=False, allow_add_remove=False, **kwargs*)

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/audio

curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/audio
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/rest-api/audio-files/1/",
            "name": "Sample audio",
            "audio_file": "upload/audiofiles/audio-file-SODXT-1669906647.wav",
            "user": "http://127.0.0.1:8000/rest-api/users/1/",
            "created_date": "2013-06-14T18:56:58.550",
            "updated_date": "2013-06-14T18:56:58.969"
        }
    ]
}
```

9.4 *BranchingTemplateSerializer*

class `apirest.branching_template_serializers.`**`BranchingTemplateSerializer`**(*instance=None, data=None, files=None, context=None, partial=False, many=False, allow_add_remove=False, **kwargs*)

Create:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data
```

Response:

```
HTTP/1.0 201 CREATED
Date: Fri, 14 Jun 2013 09:52:27 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json; charset=utf-8
Content-Language: en-us
Location: http://localhost:8000/rest-api/survey/1/
Allow: GET, POST, HEAD, OPTIONS
```

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/branc

    or

curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/branc
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/rest-api/branching-template/1/",
            "keys": "0",
            "created_date": "2013-06-13T12:42:28.531",
            "updated_date": "2013-06-13T12:42:28.531",
            "section": "http://127.0.0.1:8000/rest-api/section-template/1/",
            "goto": null
        }
    ]
}
```

Update:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X PATCH --dat
```

Response:

```
HTTP/1.0 202 NO CONTENT
Date: Fri, 23 Sep 2011 06:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.1+
Vary: Accept-Language, Cookie
Content-Length: 0
Content-Type: text/html; charset=utf-8
Content-Language: en-us
```

9.5 *CalendarSerializer*

9.6 *CalendarSettingSerializer*

9.7 *CalendarUserSerializer*

9.8 *CalendarUserProfileSerializer*

9.9 *CampaignSerializer*

class `apirest.campaign_serializers.`**`CampaignSerializer`**(*instance=None*, *data=None*, *files=None*, *context=None*, *partial=False*, *many=False*, *allow_add_remove=False*, ***kwargs*)

Create:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data
```

CURL Usage (2nd):

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data
```

Response:

```
HTTP/1.0 201 CREATED
Date: Fri, 14 Jun 2013 09:52:27 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json; charset=utf-8
Content-Language: en-us
Allow: GET, POST, HEAD, OPTIONS

{"id": 1, "campaign_code": "JDQBG", "name": "mycampaign1", "description": "", "callerid": "1
```

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/campa

curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/campa
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 2,
            "campaign_code": "BXTWX",
            "name": "Sample survey campaign",
            "description": "",
            "callerid": "",
            "phonebook": [
                "http://127.0.0.1:8000/rest-api/phonebook/1/"
            ],
            "startingdate": "2011-12-27T14:35:46",
            "expirationdate": "2011-12-28T14:35:46",
            "aleg_gateway": "http://127.0.0.1:8000/rest-api/gateway/1/",
            "user": "http://127.0.0.1:8000/rest-api/users/1/",
            "status": 2,
            "content_type": "http://127.0.0.1:8000/rest-api/content-type/49/",
            "object_id": 1,
            "extra_data": "",
            "dnc": "http://127.0.0.1:8000/rest-api/dnc/1/",
            "voicemail": false,
            "amd_behavior": null,
            "voicemail_audiofile": null,
            "frequency": 10,
            "callmaxduration": 1800,
            "maxretry": 0,
            "intervalretry": 300,
            "calltimeout": 45,
            "daily_start_time": "00:00:00",
            "daily_stop_time": "23:59:59",
            "monday": true,
            "tuesday": true,
            "wednesday": true,
            "thursday": true,
            "friday": true,
            "saturday": true,
            "sunday": true,
            "completion_maxretry": 0,
            "completion_intervalretry": 900
        }
    ]
}
```

Update:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X PATCH --dat
```

Response:

```
HTTP/1.0 202 NO CONTENT
Date: Fri, 23 Sep 2011 06:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.1+
Vary: Accept-Language, Cookie
Content-Length: 0
Content-Type: text/html; charset=utf-8
Content-Language: en-us

{"id": 1, "campaign_code": "JDQBG", "name": "mylittlecampaign243", "description": "", "calle
```

CURL Usage (Update Campaign Status==Started):

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X PATCH --dat
```

Delete:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X DELETE  htt
```

Response:

```
{
    "data": "campaign deleted"
}
```

get_fields(**args*, ***kwargs*)
filter content_type field

validate(*attrs*)
Validate campaign form

9.10 *ContactSerializer*

class apirest.contact_serializers.**ContactSerializer**(*instance=None*, *data=None*, *files=None*, *context=None*, *partial=False*, *many=False*, *allow_add_remove=False*, ***kwargs*)

Create:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data
```

Response:

```
HTTP/1.0 201 CREATED
Date: Fri, 14 Jun 2013 09:52:27 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json; charset=utf-8
Content-Language: en-us
Allow: GET, POST, HEAD, OPTIONS
```

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/conta

curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/conta
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/rest-api/contact/1/",
            "phonebook": "http://127.0.0.1:8000/rest-api/phonebook/1/",
            "contact": "55555555",
            "status": 1,
            "last_name": "Belaid",
            "first_name": "Arezqui",
            "email": "areski@gmail.com",
            "address": "Address",
            "city": "Barcelona",
            "state": "state",
            "country": "ES",
            "unit_number": "123",
            "additional_vars": "{"facility":"hurron","debt":10,"address":"Sant Lluis street
            "description": "test subscriber",
            "created_date": "2013-06-27T19:48:45.118",
            "updated_date": "2013-06-27T19:48:45.118"
        },
    ]
}
```

Update:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X PATCH --dat
```

Response:

```
HTTP/1.0 202 NO CONTENT
Date: Fri, 23 Sep 2011 06:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.1+
Vary: Accept-Language, Cookie
Content-Length: 0
Content-Type: text/html; charset=utf-8
Content-Language: en-us
```

get_fields(**args*, ***kwargs*)

filter survey field

9.11 *ContentTypeSerializer*

class `apirest.content_type_serializers.`**`ContentTypeSerializer`**(*instance=None, data=None, files=None, context=None, partial=False, many=False, allow_add_remove=False, **kwargs*)

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/conte
```

9.12 *DNCSerializer*

class `apirest.dnc_serializers.`**`DNCSerializer`**(*instance=None, data=None, files=None, context=None, partial=False, many=False, allow_add_remove=False, **kwargs*)

Create:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data
```

Response:

```
HTTP/1.0 201 CREATED
Date: Fri, 14 Jun 2013 09:52:27 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json; charset=utf-8
Content-Language: en-us
Location: http://localhost:8000/rest-api/dnc/1/
Allow: GET, POST, HEAD, OPTIONS
```

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/dnc-l
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/rest-api/dnc/1/",
            "name": "sample dnc",
            "user": "http://127.0.0.1:8000/rest-api/users/1/",
            "created_date": "2013-03-15T18:28:30.208",
            "updated_date": "2013-03-15T18:28:30.208"
        }
    ]
}
```

Update:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X PATCH --dat
```

Response:

```
HTTP/1.0 202 NO CONTENT
Date: Fri, 23 Sep 2011 06:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.1+
Vary: Accept-Language, Cookie
Content-Length: 0
Content-Type: text/html; charset=utf-8
Content-Language: en-us
```

9.13 *DNCContactSerializer*

class `apirest.dnc_contact_serializers.`**`DNCContactSerializer`**(*instance=None*, *data=None*, *files=None*, *context=None*, *partial=False*, *many=False*, *allow_add_remove=False*, ***kwargs*)

Create:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data
```

Response:

```
HTTP/1.0 201 CREATED
Date: Fri, 14 Jun 2013 09:52:27 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json; charset=utf-8
Content-Language: en-us
Location: http://localhost:8000/rest-api/dnc_contact/1/
Allow: GET, POST, HEAD, OPTIONS
```

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/dnc-c
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/rest-api/dnc-contact/1/",
            "dnc": "http://127.0.0.1:8000/rest-api/dnc-list/1/",
            "phone_number": "12345"
        }
    ]
}
```

Update:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X PUT --data
```

Response:

```
HTTP/1.0 202 NO CONTENT
Date: Fri, 23 Sep 2011 06:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.1+
Vary: Accept-Language, Cookie
Content-Length: 0
Content-Type: text/html; charset=utf-8
Content-Language: en-us
```

get_fields(**args*, ***kwargs*)
filter survey field

9.14 *EventSerializer*

9.15 *GatewaySerializer*

class apirest.gateway_serializers.**GatewaySerializer**(*instance=None*, *data=None*, *files=None*, *context=None*, *partial=False*, *many=False*, *allow_add_remove=False*, ***kwargs*)

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/gatew

curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/gatew
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/rest-api/gateway/1/",
            "name": "Default_Gateway",
            "status": 1,
            "description": "",
            "addprefix": "",
            "removeprefix": "",
            "gateways": "user/,user",
            "gateway_codecs": "PCMA,PCMU",
            "gateway_timeouts": "10,10",
            "gateway_retries": "2,1",
            "originate_dial_string": "",
            "secondused": null,
            "created_date": "2011-06-15T00:28:52",
            "updated_date": "2013-06-14T17:54:24.130",
            "failover": null,
            "addparameter": "",
            "count_call": 1,
            "count_in_use": null,
            "maximum_call": null
        }
    ]
}
```

9.16 *MailTemplateSerializer*

class `apirest.mail_template_serializers.`**`MailTemplateSerializer`**(*instance=None*, *data=None*, *files=None*, *context=None*, *partial=False*, *many=False*, *allow_add_remove=False*, ***kwargs*)

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/mail-
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/rest-api/mail-template/1/",
            "label": "test_mail",
            "template_key": "test_mail",
            "from_email": "xyz@gmail.com",
            "from_name": "xyz",
            "subject": "hi",
            "message_plaintext": "test message",
            "message_html": "<h1>Hi</h1>",
            "created_date": "2013-12-16T06:37:42.938Z"
        }
    ]
}
```

9.17 *PhonebookSerializer*

class `apirest.phonebook_serializers.PhonebookSerializer`(*instance=None*, *data=None*, *files=None*, *context=None*, *partial=False*, *many=False*, *allow_add_remove=False*, `**kwargs`)

Create:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data
```

Response:

```
HTTP/1.0 201 CREATED
Date: Fri, 14 Jun 2013 09:52:27 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json; charset=utf-8
Content-Language: en-us
Allow: GET, POST, HEAD, OPTIONS
```

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/phone

curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/phone
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/rest-api/phonebook/1/",
            "name": "Default_Phonebook",
            "description": "",
            "user": "http://127.0.0.1:8000/rest-api/users/1/",
            "created_date": "2011-04-08T07:55:05",
            "updated_date": "2011-04-08T07:55:05"
        }
    ]
}
```

Update:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X PATCH --dat
```

Response:

```
HTTP/1.0 200 NO CONTENT
Date: Fri, 23 Sep 2011 06:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.1+
Vary: Accept-Language, Cookie
Content-Length: 0
Content-Type: text/html; charset=utf-8
Content-Language: en-us
```

Delete:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X DELETE  htt
```

9.18 *RuleSerializer*

9.19 *SectionTemplateSerializer*

class apirest.section_template_serializers.**SectionTemplateSerializer**(*instance=None*, *data=None*, *files=None*, *context=None*, *partial=False*, *many=False*, *allow_add_remove=False*, ***kwargs*)

Create:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data
```

Response:

```
HTTP/1.0 201 CREATED
Date: Fri, 14 Jun 2013 09:52:27 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json; charset=utf-8
Content-Language: en-us
Location: http://localhost:8000/rest-api/section-template/1/
Allow: GET, POST, HEAD, OPTIONS
```

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/secti

    or

curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/secti
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/rest-api/section-template/1/",
            "order": 1,
            "type": 1,
            "question": "this is test question",
            "script": "this is test question",
            "audiofile": null,
            "retries": null,
            "timeout": 5,
            "key_0": null,
            "key_1": null,
            "key_2": null,
            "key_3": null,
            "key_4": null,
            "key_5": null,
            "key_6": null,
            "key_7": null,
            "key_8": null,
            "key_9": null,
            "rating_laps": 9,
            "validate_number": true,
            "number_digits": 2,
            "min_number": 0,
            "max_number": 99,
            "phonenumber": null,
            "conference": null,
            "completed": false,
            "queue": null,
            "created_date": "2013-06-13T12:42:28.457",
            "updated_date": "2013-06-13T12:42:28.511",
            "survey": "http://127.0.0.1:8000/rest-api/survey-template/1/",
            "invalid_audiofile": null
        }
    ]
}
```

Update:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X PATCH --dat
```

Response:

```
HTTP/1.0 202 NO CONTENT
Date: Fri, 23 Sep 2011 06:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.1+
Vary: Accept-Language, Cookie
Content-Length: 0
Content-Type: text/html; charset=utf-8
Content-Language: en-us
```

9.20 *SMSCampaignSerializer*

class `apirest.sms_campaign_serializers.`**`SMSCampaignSerializer`**(*instance=None*, *data=None*, *files=None*, *context=None*, *partial=False*, *many=False*, *allow_add_remove=False*, ***kwargs*)

Create:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data
```

Response:

```
HTTP/1.0 201 CREATED
Date: Fri, 14 Jun 2013 09:52:27 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json; charset=utf-8
Content-Language: en-us
Allow: GET, POST, HEAD, OPTIONS

{"id": 1, "sms_campaign_code": "JDQBG", "name": "mysmscampaign1", "description": "", "caller
```

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/sms-c

curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/sms-c
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 2,
            "sms_campaign_code": "BXTWX",
            "name": "Sample SMS campaign",
            "description": "",
            "callerid": "",
            "phonebook": [
                "http://127.0.0.1:8000/rest-api/phonebook/1/"
            ],
            "startingdate": "2011-12-27T14:35:46",
            "expirationdate": "2011-12-28T14:35:46",
            "sms_gateway": "http://127.0.0.1:8000/rest-api/sms-gateway/1/",
            "user": "http://127.0.0.1:8000/rest-api/users/1/",
            "status": 2,
            "frequency": 10,
            "callmaxduration": 1800,
            "maxretry": 0,
            "daily_start_time": "00:00:00",
            "daily_stop_time": "23:59:59",
            "monday": true,
            "tuesday": true,
            "wednesday": true,
            "thursday": true,
            "friday": true,
            "saturday": true,
            "sunday": true,
        }
    ]
}
```

Update:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X PATCH --dat
```

Response:

```
HTTP/1.0 202 NO CONTENT
Date: Fri, 23 Sep 2011 06:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.1+
Vary: Accept-Language, Cookie
Content-Length: 0
Content-Type: text/html; charset=utf-8
Content-Language: en-us
```

Delete:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X DELETE  htt
```

Response:

```
{
    "data": "sms campaign deleted"
}
```

get_fields(**args*, ***kwargs*)

filter content_type field

validate(*attrs*)

Validate sms campaign form

9.21 *SMSGatewaySerializer*

class `apirest.sms_gateway_serializers.`**SMSGatewaySerializer**(*instance=None*, *data=None*, *files=None*, *context=None*, *partial=False*, *many=False*, *allow_add_remove=False*, ***kwargs*)

Create:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data
```

Response:

```
HTTP/1.0 201 CREATED
Date: Fri, 14 Jun 2013 09:52:27 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json; charset=utf-8
Content-Language: en-us
Allow: GET, POST, HEAD, OPTIONS
```

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/gatew

curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/gatew
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/rest-api/gateway/1/",
            "name": "Default_Gateway",
            "status": 1,
            "description": "",
            "addprefix": "",
            "removeprefix": "",
            "gateways": "user/,user",
            "gateway_codecs": "PCMA,PCMU",
            "gateway_timeouts": "10,10",
            "gateway_retries": "2,1",
            "originate_dial_string": "",
            "secondused": null,
            "created_date": "2011-06-15T00:28:52",
            "updated_date": "2013-06-14T17:54:24.130",
            "failover": null,
            "addparameter": "",
            "count_call": 1,
            "count_in_use": null,
            "maximum_call": null
        }
    ]
}
```

9.22 *SMSTemplateSerializer*

class `apirest.sms_template_serializers.`**`SMSTemplateSerializer`**(*instance=None, data=None, files=None, context=None, partial=False, many=False, allow_add_remove=False, **kwargs*)

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/sms-t
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/rest-api/sms-template/1/",
            "label": "sms_test",
            "template_key": "sms_test",
            "sender_phonenumber": "9427164510",
            "sms_text": "hello test",
            "created_date": "2013-12-16T06:43:29.475Z"
        }
    ]
}
```

9.23 *SubscriberListSerializer*

class `apirest.subscriber_list_serializers.`**`SubscriberListSerializer`**(*instance=None, data=None, files=None, context=None, partial=False, many=False, allow_add_remove=False, **kwargs*)

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/subsc
```

Response:

```
        [
            {
                "id": 1,
                "contact": "/rest-api/contact/11/",
                "campaign": "/rest-api/campaigns/3/",
                "last_attempt": null,
                "count_attempt": 0,
                "completion_count_attempt": 0,
                "duplicate_contact": "34235464",
                "status": 1
            },
            {
                "id": 2,
                "contact": "/rest-api/contact/12/",
                "campaign": "/rest-api/campaigns/3/",
                "last_attempt": null,
                "count_attempt": 0,
                "completion_count_attempt": 0,
                "duplicate_contact": "34235464",
                "status": 1
            }
        ]
```

9.24 *SurveySerializer*

class `apirest.survey_serializers.`**`SurveySerializer`**(*instance=None*, *data=None*, *files=None*, *context=None*, *partial=False*, *many=False*, *allow_add_remove=False*, ***kwargs*)

Read:

CURL Usage:

```
        curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/seale

            or

        curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/seale
```

Response:

```
        {
            "count": 1,
            "next": null,
            "previous": null,
            "results": [
                {
                    "url": "http://127.0.0.1:8000/rest-api/sealed-survey/1/",
                    "name": "Sample survey campaign",
                    "tts_language": "en",
                    "description": "ok",
                    "created_date": "2013-06-13T12:42:18.148",
                    "updated_date": "2013-06-13T12:42:31.527",
                    "user": "http://127.0.0.1:8000/rest-api/users/1/"
                }
            ]
        }
```

9.25 *SurveyTemplateSerializer*

class `apirest.survey_template_serializers.`**`SurveyTemplateSerializer`**(*instance=None, data=None, files=None, context=None, partial=False, many=False, allow_add_remove=False, **kwargs*)

Create:

CURL Usage:

```
curl -u username:password --dump-header - -H "Content-Type:application/json" -X POST --data
```

Response:

```
HTTP/1.0 201 CREATED
Date: Fri, 14 Jun 2013 09:52:27 GMT
Server: WSGIServer/0.1 Python/2.7.3
Vary: Accept, Accept-Language, Cookie
Content-Type: application/json; charset=utf-8
Content-Language: en-us
Location: http://localhost:8000/rest-api/survey-template/1/
Allow: GET, POST, HEAD, OPTIONS
```

Read:

CURL Usage:

```
curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/surve

    or

curl -u username:password -H 'Accept: application/json' http://localhost:8000/rest-api/surve
```

Response:

```
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "url": "http://127.0.0.1:8000/rest-api/survey-template/1/",
            "name": "Sample survey campaign",
            "tts_language": "en",
            "description": "ok",
            "created_date": "2013-06-13T12:42:18.148",
            "updated_date": "2013-06-13T12:42:31.527",
            "user": "http://127.0.0.1:8000/rest-api/users/1/"
```

```
                }
            ]
        }
```

Update:

CURL Usage:

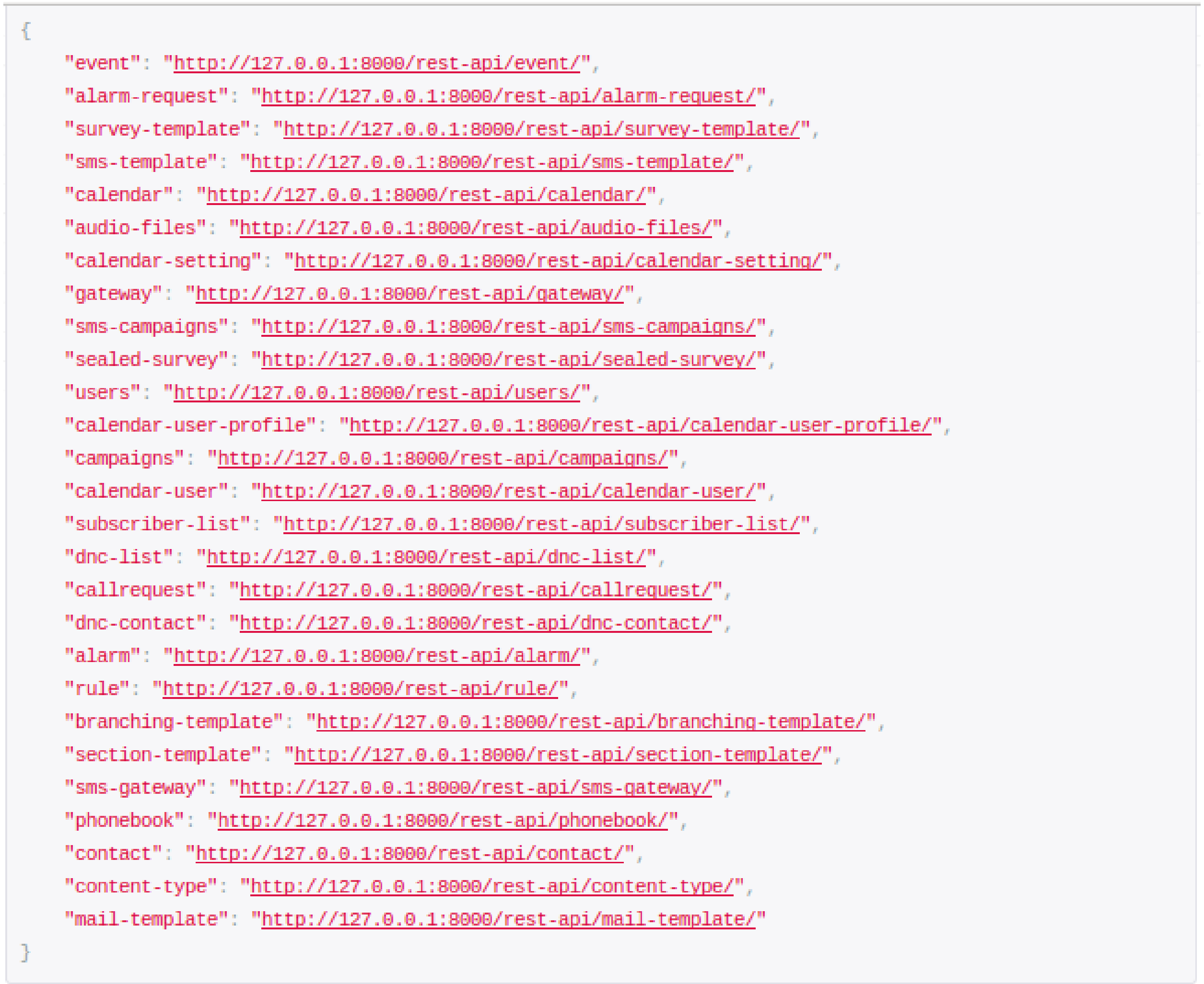
```
curl -u username:password --dump-header - -H "Content-Type: application/json" -X PATCH --dat
```

Response:

```
HTTP/1.0 202 NO CONTENT
Date: Fri, 23 Sep 2011 06:46:12 GMT
Server: WSGIServer/0.1 Python/2.7.1+
Vary: Accept-Language, Cookie
Content-Length: 0
Content-Type: text/html; charset=utf-8
Content-Language: en-us
```

Testing console of APIs:

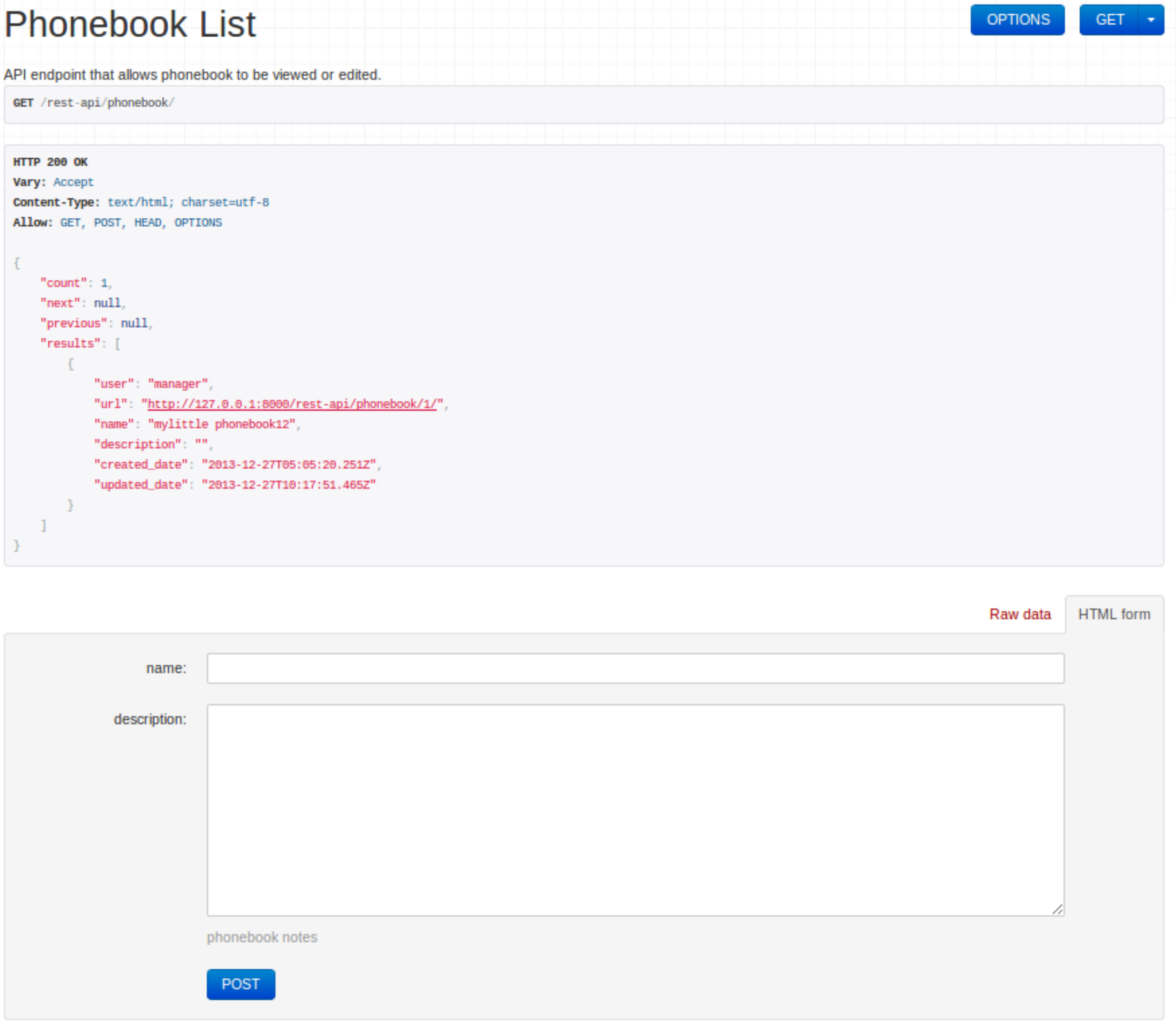
URL : http://127.0.0.1:8000/api-explorer/



```
{
    "event": "http://127.0.0.1:8000/rest-api/event/",
    "alarm-request": "http://127.0.0.1:8000/rest-api/alarm-request/",
    "survey-template": "http://127.0.0.1:8000/rest-api/survey-template/",
    "sms-template": "http://127.0.0.1:8000/rest-api/sms-template/",
    "calendar": "http://127.0.0.1:8000/rest-api/calendar/",
    "audio-files": "http://127.0.0.1:8000/rest-api/audio-files/",
    "calendar-setting": "http://127.0.0.1:8000/rest-api/calendar-setting/",
    "gateway": "http://127.0.0.1:8000/rest-api/gateway/",
    "sms-campaigns": "http://127.0.0.1:8000/rest-api/sms-campaigns/",
    "sealed-survey": "http://127.0.0.1:8000/rest-api/sealed-survey/",
    "users": "http://127.0.0.1:8000/rest-api/users/",
    "calendar-user-profile": "http://127.0.0.1:8000/rest-api/calendar-user-profile/",
    "campaigns": "http://127.0.0.1:8000/rest-api/campaigns/",
    "calendar-user": "http://127.0.0.1:8000/rest-api/calendar-user/",
    "subscriber-list": "http://127.0.0.1:8000/rest-api/subscriber-list/",
    "dnc-list": "http://127.0.0.1:8000/rest-api/dnc-list/",
    "callrequest": "http://127.0.0.1:8000/rest-api/callrequest/",
    "dnc-contact": "http://127.0.0.1:8000/rest-api/dnc-contact/",
    "alarm": "http://127.0.0.1:8000/rest-api/alarm/",
    "rule": "http://127.0.0.1:8000/rest-api/rule/",
    "branching-template": "http://127.0.0.1:8000/rest-api/branching-template/",
    "section-template": "http://127.0.0.1:8000/rest-api/section-template/",
    "sms-gateway": "http://127.0.0.1:8000/rest-api/sms-gateway/",
    "phonebook": "http://127.0.0.1:8000/rest-api/phonebook/",
    "contact": "http://127.0.0.1:8000/rest-api/contact/",
    "content-type": "http://127.0.0.1:8000/rest-api/content-type/",
    "mail-template": "http://127.0.0.1:8000/rest-api/mail-template/"
}
```

To test individual api, click on one api from above api list and you will get like following screen.

URL : http://127.0.0.1:8000/api-explorer/switch/



Phonebook List

API endpoint that allows phonebook to be viewed or edited.

```
GET /rest-api/phonebook/
```

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, POST, HEAD, OPTIONS

{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "user": "manager",
            "url": "http://127.0.0.1:8000/rest-api/phonebook/1/",
            "name": "mylittle phonebook12",
            "description": "",
            "created_date": "2013-12-27T05:05:20.251Z",
            "updated_date": "2013-12-27T10:17:51.465Z"
        }
    ]
}
```

name:

description:

phonebook notes

CHAPTER 10

Troubleshooting

10.1 Where to find help

10.1.1 Documentation:

http://www.newfies-dialer.org/documentation/

10.1.2 Mailing list:

We have set up a mailing list at http://groups.google.com/group/newfies-dialer

10.1.3 Forum:

We have a forum at http://forum.newfies-dialer.org/

10.1.4 Support:

Star2Billing S.L. offers consultancy including installation, training and customisation. Please email us at newfies-dialer@star2billing.com for more information.

10.2 Where to find the log files

All the logs are centralized into one single directory **/var/log/newfies/**

newfies-django-db.log : This contains all the Database queries performed by the UI

newfies-django.log : All the logger events from Django

err-apache-newfies.log : Any apache errors pertaining to Newfies-Dialer

celery-newfies-node1.log : This contains celery activity

newfieslua_logs_%%.log.log : This contains the phone system logs

10.3 Where are located the config files

Newfies-Dialer configuration file is located by default at /usr/share/newfies-dialer/newfies_dialer/settings_local.py

settings_local.py contains all the local changes specific to your installation, this file is inherited from /usr/share/newfies-dialer/newfies_dialer/settings.py, therefore you can copy settings to settings_local.py and overwrite the default configuration.

10.4 Enable the debug

When you get a message "Server Error, You've encountered an error!", you might want to enable the debugger to get a full traceback of the error.

To enable the debugger, edit the file /usr/share/newfies-dialer/newfies_dialer/settings_local.py and set those settings as follow:

```
DEBUG = True
TEMPLATE_DEBUG = DEBUG
```

Note that you should be able to get the traceback in **/var/log/newfies/err-apache-newfies.log**

10.5 How to run a quick test call

Go on the admin panel and check if there is any call request that has been spooled.

- http://your-ip:8008/admin/dialer_cdr/callrequest/

If there are no calls queued, this means that the campaign is not properly configured.

You should:

1. Check if the campaign is started that the "Start time", "Finish Time" and server time are correct.
2. Make sure that you configured a Dialer Setting for the user running the campaign, although there will be a warning for this on the Customer UI : http://your-ip:8008/admin/dialer_settings/dialersetting/

If there is an existing Call Request, check the status, and check the Celery log stored in /var/log/newfies

10.6 Run in debug mode

Make sure you stop the services first:

```
$ /etc/init.d/newfies-celeryd stop
```

Then run in debug mode:

```
$ workon newfies-dialer
$ cd /usr/share/newfies/
$ python manage.py celeryd -EB --loglevel=DEBUG
```

10.7 Run on Amazon EC2

Although installing and running Newfies-Dialer on EC2 should work flawlessly, you will need to update the settings ALLOWED_HOSTS manually as the install script won't be able to detect the public IP address of your EC2 Instance.

Edit /usr/share/newfies/settings_local.py and ensure that your public IP is set in ALLOWED_HOSTS, ie. ALLOWED_HOSTS = ['XXX.YYY.ZZZ']

Further information about ALLOWED_HOSTS: https://docs.djangoproject.com/en/1.5/ref/settings/#allowed-hosts

10.8 Celerymon

- https://github.com/ask/celerymon

Running the monitor :

Start celery with the –events option on, so celery sends events for celerymon to capture:: $ workon newfies-dialer $ cd /usr/share/newfies/ $ python manage.py celeryd -E

Run the monitor server:

```
$ workon newfies-dialer
$ cd /usr/share/newfies/
$ python manage.py celerymon
```

However, in production you probably want to run the monitor in the background, as a daemon:

```
$ workon newfies-dialer
$ cd /usr/share/newfies/
$ python manage.py celerymon --detach
```

For a complete listing of the command line arguments available, with a short description, you can use the help command:

```
$ workon newfies-dialer
$ cd /usr/share/newfies/
$ python manage.py help celerymon
```

Now you can visit the webserver celerymon starts by going to: http://localhost:8989

10.9 How to discard all pending tasks

You can use the celery purge command to purge all configured task queues. http://docs.celeryproject.org/en/latest/faq.html?highlight=purge#how-do-i-purge-all-waiting-tasks

10.10 Checking Freeswitch

Entering the Freeswitch CLI shold indicate whether it is running by typing fs_cli at the console. Once logged in, you can check the trunk registration by typing sofia status at the Freeswitch CLI. CTRL-D exits the Freeswitch CLI.

If the Freeswitch CLI cannot be launched, then the status of freeswitch can be checked with:

```
$ ps aux | grep freeswitch
or
$ /etc/init.d/freeswitch status
```

If Freeswitch is not running, then it can be started with:

```
$ /etc/init.d/freeswitch start
```

10.11 Step By Step Checklist

The step by step checklist below should be used to validate that all components of the platform are running.

User interface :

- 1. Dialer Gateway matching a configured trunk is set up in the UI
- 2. Dialer Settings configured and attached to the appropriate user
- 3. Phonebook Created with contacts attached to the phonebook
- 4. Configured voice application
- 5. Campaign created, and started, with a phone book attached, and the campaign schedule current

Backend :

- 1. Celery Monitor Running
- 2. Freeswitch running

If there are still problems, then raise a support question on the mailing-list http://groups.google.com/group/newfies-dialer or our forum, http://forum.newfies-dialer.org/, alternatively, contact newfies-dialer@star2billing.com for commercial support.

CHAPTER 11

Contributing

This document is highly inspired from the Celery documentation.

Welcome to Newfies-Dialer!

This document is fairly extensive and you are not really expected to study this in detail for small contributions;

> The most important rule is that contributing must be easy and that the community is friendly and not nitpicking on details such as coding style.

If you're reporting a bug you should read the Reporting bugs section below to ensure that your bug report contains enough information to successfully diagnose the issue, and if you're contributing code you should try to mimic the conventions you see surrounding the code you are working on, but in the end all patches will be cleaned up by the person merging the changes so don't worry too much.

11.1 Community Code of Conduct

The goal is to maintain a diverse community that is pleasant for everyone. That is why we would greatly appreciate it if everyone contributing to and interacting with the community also followed this Code of Conduct.

The Code of Conduct covers our behavior as members of the community, in any forum, mailing list, wiki, website, Internet relay chat (IRC), public meeting or private correspondence.

The Code of Conduct is heavily based on the Ubuntu Code of Conduct, Celery Code of Conduct, and the Pylons Code of Conduct.

11.1.1 Be considerate.

Your work will be used by other people, and you in turn will depend on the work of others. Any decision you take will affect users and colleagues, and we expect you to take those consequences into account when making decisions. Even if it's not obvious at the time, our contributions to Newfies will impact the work of others. For example, changes to code, infrastructure, policy, documentation and translations during a release may negatively impact others work.

11.1.2 Be respectful.

The Newfies community and its members treat one another with respect. Everyone can make a valuable contribution to Newfies. We may not always agree, but disagreement is no excuse for poor behavior and poor manners. We might all experience some frustration now and then, but we cannot allow that frustration to turn into a personal attack. It's important to remember that a community where people feel uncomfortable or threatened is not a productive one. We expect members of the Newfies community to be respectful when dealing with other contributors as well as with people outside the Newfies project and with users of Newfies.

11.1.3 Be collaborative.

Collaboration is central to Newfies and to the larger free software community. We should always be open to collaboration. Your work should be done transparently and patches from Newfies should be given back to the community when they are made, not just when the distribution releases. If you wish to work on new code for existing upstream projects, at least keep those projects informed of your ideas and progress. It many not be possible to get consensus from upstream, or even from your colleagues about the correct implementation for an idea, so don't feel obliged to have that agreement before you begin, but at least keep the outside world informed of your work, and publish your work in a way that allows outsiders to test, discuss and contribute to your efforts.

11.1.4 When you disagree, consult others.

Disagreements, both political and technical, happen all the time and the Newfies community is no exception. It is important that we resolve disagreements and differing views constructively and with the help of the community and community process. If you really want to go a different way, then we encourage you to make a derivative distribution or alternate set of packages that still build on the work we've done to utilize as common of a core as possible.

11.1.5 When you are unsure, ask for help.

Nobody knows everything, and nobody is expected to be perfect. Asking questions avoids many problems down the road, and so questions are encouraged. Those who are asked questions should be responsive and helpful. However, when asking a question, care must be taken to do so in an appropriate forum.

11.1.6 Step down considerately.

Developers on every project come and go and Newfies is no different. When you leave or disengage from the project, in whole or in part, we ask that you do so in a way that minimizes disruption to the project. This means you should tell people you are leaving and take the proper steps to ensure that others can pick up where you leave off.

11.2 Reporting Bugs

11.2.1 Bugs

Bugs can always be described to the *Mailing list*, but the best way to report an issue and to ensure a timely response is to use the issue tracker.

1. **Create a GitHub account.**

You need to create a GitHub account to be able to create new issues and participate in the discussion.

2. **Determine if your bug is really a bug.**

You should not file a bug if you are requesting support. For that you can use the *Mailing list*, or *IRC*.

3. **Make sure your bug hasn't already been reported.**

Search through the appropriate Issue tracker. If a bug like yours was found, check if you have new information that could be reported to help the developers fix the bug.

4. **Check if you're using the latest version.**

A bug could be fixed by some other improvements and fixes - it might not have an existing report in the bug tracker. Make sure you're using the latest version.

5. **Collect information about the bug.**

To have the best chance of having a bug fixed, we need to be able to easily reproduce the conditions that caused it. Most of the time this information will be from a Python traceback message, though some bugs might be in design, spelling or other errors on the website/docs/code.

1. If the error is from a Python traceback, include it in the bug report.
2. We also need to know what platform you're running (Windows, OS X, Linux, etc.), the version of your Python interpreter, and the version of related packages that you were running when the bug occurred.

6. **Submit the bug.**

By default GitHub will email you to let you know when new comments have been made on your bug. In the event you've turned this feature off, you should check back on occasion to ensure you don't miss any questions a developer trying to fix the bug might ask.

11.2.2 Issue Trackers

Bugs for a package in the Newfies-Dialer ecosystem should be reported to the relevant issue tracker.

- Newfies-Dialer Core: https://github.com/newfies-dialer/newfies-dialer/issues/
- Python-Acapela: https://github.com/newfies-dialer/python-acapela/issues
- Lua-Acapela: https://github.com/newfies-dialer/lua-acapela/issues
- Python-NVD3: https://github.com/areski/python-nvd3/issues

If you are unsure of the origin of the bug you can ask the *Mailing list*, or just use the Newfies-Dialer issue tracker.

11.3 Versions

Version numbers consists of a major version, minor version and a release number. We use the versioning semantics described by semver: http://semver.org.

Stable releases are published at PyPI while development releases are only available in the GitHub git repository as tags. All version tags starts with "v", so version 0.8.0 is the tag v0.8.0.

11.4 Branches

Current active version branches:

- master (http://github.com/newfies-dialer/newfies-dialer/tree/master)
- 2.19.10 (http://github.com/newfies-dialer/newfies-dialer/tree/v2.19.10)

You can see the state of any branch by looking at the Changelog:

https://github.com/newfies-dialer/newfies-dialer/blob/master/Changelog

11.4.1 Feature branches

Major new features are worked on in dedicated branches. There is no strict naming requirement for these branches.

Feature branches are removed once they have been merged into a release branch.

11.5 Tags

Tags are used exclusively for tagging releases. A release tag is named with the format `vX.Y.Z`, e.g. `v2.3.1`. Experimental releases contain an additional identifier `vX.Y.Z-id`, e.g. `v3.0.0-rc1`. Experimental tags may be removed after the official release.

11.6 Working on Features & Patches

Note: Contributing to Newfies-Dialer should be as simple as possible, so none of these steps should be considered mandatory.

You can even send in patches by email if that is your preferred work method. We won't like you any less, any contribution you make is always appreciated!

However following these steps may make maintainers life easier, and may mean that your changes will be accepted sooner.

11.6.1 Forking and setting up the repository

First you need to fork the repository, a good introduction to this is in the Github Guide: Fork a Repo.

After you have cloned the repository you should checkout your copy to a directory on your machine:

```
$ git clone git@github.com:username/newfies-dialer.git
```

When the repository is cloned enter the directory to set up easy access to upstream changes:

```
$ cd newfies-dialer
$ git remote add upstream git://github.com/newfies-dialer/newfies-dialer.git
$ git fetch upstream
```

If you need to pull in new changes from upstream you should always use the `--rebase` option to `git pull`:

```
$ git pull --rebase upstream master
```

With this option you don't clutter the history with merging commit notes. See Rebasing merge commits in git. If you want to learn more about rebasing see the Rebase section in the Github guides.

If you need to work on a different branch than `master` you can fetch and checkout a remote branch like this:

```
$ git checkout --track -b 3.0-devel origin/3.0-devel
```

11.6.2 Running the unit test suite

To run the Newfies-Dialer test suite you need to install a few dependencies. A complete list of the dependencies needed are located in `requirements/test.txt`.

Installing the test requirements:

```
$ pip install -U -r requirements/test.txt
```

When installation of dependencies is complete you can execute the test suite by calling `py.test`:

```
$ py.test
```

Some useful options to **`py.test`** are:

- `-x`

  Exit instantly on first error or failed test.

- `--ipdb`

  Starts the interactive IPython debugger on errors.

- `-k EXPRESSION`

  Only run tests which match the given substring expression.

- `-v`

  Increase verbose.

If you want to run the tests for a single test file only you can do so like this:

```
$ py.test appointment./tests.py
```

11.6.3 Creating pull requests

When your feature/bugfix is complete you may want to submit a pull requests so that it can be reviewed by the maintainers.

Creating pull requests is easy, and also let you track the progress of your contribution. Read the Pull Requests section in the Github Guide to learn how this is done.

You can also attach pull requests to existing issues by following the steps outlined here: http://bit.ly/koJoso

Calculating test coverage

To calculate test coverage you must first install the `coverage` module.

Installing the `coverage` module:

```
$ pip install -U coverage
```

Code coverage in HTML:

```
$ nosetests --with-coverage --cover-html
```

The coverage output will then be located at `newfies-dialer/tests/cover/index.html`.

Code coverage in XML (Cobertura-style):

```
$ nosetests --with-coverage --cover-xml --cover-xml-file=coverage.xml
```

The coverage XML output will then be located at `coverage.xml`

Running the tests on all supported Python versions

There is a `tox` configuration file in the top directory of the distribution.

To run the tests for all supported Python versions simply execute:

```
$ tox
```

If you only want to test specific Python versions use the `-e` option:

```
$ tox -e py27
```

11.6.4 Building the documentation

To build the documentation you need to install the dependencies listed in `requirements/docs.txt`:

```
$ pip install -U -r requirements/docs.txt
```

After these dependencies are installed you should be able to build the docs by running:

```
$ cd docs
$ rm -rf .build
$ make html
```

Make sure there are no errors or warnings in the build output. After building succeeds the documentation is available at `.build/html`.

11.6.5 Verifying your contribution

To use these tools you need to install a few dependencies. These dependencies can be found in `requirements/pkgutils.txt`.

Installing the dependencies:

```
$ pip install -U -r requirements/pkgutils.txt
```

pyflakes & PEP8

To ensure that your changes conform to PEP8 and to run pyflakes execute:

```
$ flake8 newfies
```

11.7 Coding Style

You should probably be able to pick up the coding style from surrounding code, but it is a good idea to be aware of the following conventions.

- All Python code must follow the PEP-8 guidelines.

pep8.py is an utility you can use to verify that your code is following the conventions.

- Docstrings must follow the PEP-257 conventions, and use the following style.

  Do this:

  ```
  def method(self, arg):
      """Short description.

      More details.

      """
  ```

  or:

  ```
  def method(self, arg):
      """Short description."""
  ```

  but not this:

  ```
  def method(self, arg):
      """
      Short description.
      """
  ```

- Lines should not exceed 78 columns.

  You can enforce this in **vim** by setting the `textwidth` option:

  ```
  set textwidth=78
  ```

  If adhering to this limit makes the code less readable, you have one more character to go on, which means 78 is a soft limit, and 79 is the hard limit :)

- Import order

  - Python standard library (*import xxx*)
  - Python standard library ('from xxx import')
  - Third party packages.
  - Other modules from the current package.

  or in case of code using Django:

  - Python standard library (*import xxx*)
  - Python standard library ('from xxx import')
  - Third party packages.
  - Django packages.
  - Other modules from the current package.

  Within these sections the imports should be sorted by module name.

  Example:

  ```
  import threading
  import time

  from collections import deque
  from Queue import Queue, Empty
  ```

```
from .datastructures import TokenBucket
from .five import zip_longest, items, range
from .utils import timeutils
```

- Wildcard imports must not be used (*from xxx import* *).
- For distributions where Python 2.5 is the oldest support version additional rules apply:
  - Absolute imports must be enabled at the top of every module:

```
from __future__ import absolute_import
```

  - If the module uses the with statement and must be compatible with Python 2.5 then it must also enable that:

```
from __future__ import with_statement
```

  - Every future import must be on its own line, as older Python 2.5 releases did not support importing multiple features on the same future import line:

```
# Good
from __future__ import absolute_import
from __future__ import with_statement

# Bad
from __future__ import absolute_import, with_statement
```

  (Note that this rule does not apply if the package does not include support for Python 2.5)

- Note that we use “new-style‘ relative imports when the distribution does not support Python versions below 2.5

  This requires Python 2.5 or later:

```
from . import submodule
```

11.8 Contacts

This is a list of people that can be contacted for questions regarding the official git repositories, PyPI packages Read the Docs pages.

If the issue is not an emergency then it is better to *report an issue*.

11.8.1 Committers

Areski Belaid

github https://github.com/areski

twitter http://twitter.com/#!/areskib

11.8.2 Website

The Newfies-Dialer Project is run and maintained by

Star2Billing

website http://star2billing.com/

twitter https://twitter.com/#!/star2billing

11.9 Release Procedure

11.9.1 Updating the version number

The version number must be updated one place:

- newfies/newfies_dialer/__init__.py

After you have changed these files you must render the README files. There is a script to convert sphinx syntax to generic reStructured Text syntax, and the make target *readme* does this for you:

```
$ make readme
```

Now commit the changes:

```
$ git commit -a -m "Bumps version to X.Y.Z"
```

and make a new version tag:

```
$ git tag vX.Y.Z
$ git push --tags
```

11.9.2 Releasing

Commands to make a new public stable release:

```
$ make distcheck  # checks pep8, autodoc index, runs tests and more
$ make dist  # NOTE: Runs git clean -xdf and removes files not in the repo.
$ python setup.py sdist bdist_wheel upload  # Upload package to PyPI
```

If this is a new release series then you also need to do the following:

- **Go to the Read The Docs management interface at:** http://readthedocs.org/projects/newfies-dialer/?fromdocs=newfies-dialer
- Enter “Edit project”

  Change default branch to the branch of this series, e.g. 2.4 for series 2.4.
- Also add the previous version under the “versions” tab.

CHAPTER 12

Resources

12.1 Getting Help

12.1.1 Mailing list

For discussions about the usage, development, and future of Newfies-Dialer, please join the Newfies-Dialer mailing list.

12.1.2 IRC

Come chat with us on IRC. The **#newfies-dialer** channel is located at the Freenode network.

12.2 Bug tracker

If you have any suggestions, bug reports or annoyances please report them to our issue tracker at https://github.com/newfies-dialer/newfies-dialer/issues/

12.3 Wiki

https://github.com/newfies-dialer/newfies-dialer/wiki/

12.4 Contributing

Development of *Newfies-Dialer* happens at Github: https://github.com/newfies-dialer/newfies-dialer

You are highly encouraged to participate in the development of *Newfies-Dialer*. If you would prefer not to use Github, you are welcome to send us regular patches

Be sure to also read the *Contributing* section in the documentation.

12.5 License

This software is licensed under the *MPL 2.0 License*. See the `LICENSE` file in the top distribution directory for the full license text.

CHAPTER 13

Frequently Asked Questions

13.1 General

13.1.1 What is Newfies-Dialer

Answer:

Newfies-Dialer is a voice broadcast application designed and built to automate the delivery of interactive phone calls to contacts, clients and the general public.

13.1.2 Why should I use Newfies-Dialer

Answer:

Below are some examples of some of the uses that Newfies-Dialer can be put to. There are more details and examples at http://www.newfies-dialer.org/solutions/

- **Telecasting**:

  Broadcast marketing or informational messages to customers and clients.

- **Phone Polling, Surveys and Voting**:

Ring large numbers of people and present IVR options for either polling their opinions, interactive surveys, or taking their vote and record the results.

- **Debt Control**:

  Customers can be automatically reminded at intervals that they owe money, and an IVR menu presented to talk to the finance department or passed to a credit card capture IVR to pay over the phone.

- **Appointment reminders**:

  Doctors, Dentists, and other organisations that make appointments for their clients can integrate Newfies-Dialer into their appointment systems to pass a message reminding them of an upcoming appointment.

- **Dissemination of Information by Phone**:

  Newfies-Dialer was originally designed to call large numbers of people and disseminate medical and health advice via the ubiquitous cellphone in 3rd world countries where often, literacy levels are low.

- **Mass Emergency Broadcasting**:

  Where there is a necessity to warn large numbers of people in a short space of time, such as weather warnings.

- **Subscription Reminders and Renewals**:

  Where a company sells an annual subscription for a product or service, Newfies-Dialer can be configured to dial the customer, remind them that the subscription is due, and optionally pass the call into a call centre or into a credit card payment IVR.

13.1.3 What's the history behind Newfies-Dialer

Answer:

Newfies-Dialer is a bulk dialer application which was commissioned by a charity named Kubatana (http://www.kubatana.net) based in Zimbabwe, which sponsors the Freedomfone project (http://www.freedomfone.org/) dedicated to providing information via phone technology.

In less economically developed countries, Internet is often limited, but there is usually comprehensive mobile phone coverage. Freedomfone uses Newfies-Dialer to dial up people's phones and offer health information on Cholera, Malaria and so many other avoidable health issues in the third world, which may be alleviated by education. Newfies-Dialer was so named after the Newfoundland Dog nicknamed Newfies and used by sea rescue services around the world.

13.1.4 How to change your IP address

During the installation Newfies-Dialer Web interface are bind to the detected IP, if you change the IP or want to change the interface to which is bind you then need change the settings ALLOWED_HOSTS. Please find some documentation about this settings at https://docs.djangoproject.com/en/dev/ref/settings/#allowed-hosts

The configuration file on your server is located at /usr/share/newfies-dialer/newfies_dialer/settings_local.py

After changes in settings_local.py you will need to restart supervisord:

> /etc/init.d/supervisor stop

Then to start it back:

> /etc/init.d/supervisor start

13.1.5 How to translate the frontend application

First check if there is an existing translation directory in: https://github.com/newfies-dialer/newfies-dialer/tree/develop/newfies/locale Each language is represented by a 2 character ISO code (http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)

If there is no existing directory for your language code, then you can download the English PO file and use it as a base: https://raw.githubusercontent.com/Star2Billing/newfies-dialer/develop/newfies/locale/en/LC_MESSAGES/django.po

If there is a file for the language you want to translate, you will need to download that PO file and completed the missing translations. For instance, there is a PO file for Japanese: https://raw.githubusercontent.com/Star2Billing/newfies-dialer/develop/newfies/locale/ja/LC_MESSAGES/django.po

Working with an existing PO File: it's likely that you will see missing translations and/or translations that need to be reviewed, they are usually marked by *fuzzy*. The translation has to be corrected before removing the *fuzzy* line.

Here an example: #: appointment/models/events.py:32 #, fuzzy msgid "created on" msgstr ""

If the translation is good, then just remove the line with "#, fuzzy" If there is an error, correct it and remove the line with "#, fuzzy"

Once the PO is completed, please send it over to our team at newfies-dialer@star2billing.com, so that it can be included in the next release.

13.1.6 How often do I need to upgrade / How do I upgrade?

It is not necessary to upgrade to each and every release, and generally we would advise against it, unless we issue an advisory via mailing list to the contrary, e.g. major bug or security issue. Generally one can stay with a single version for a long time and decide to upgrade when there are new features or bug fixes that affect you.

The database has to be migrated with Django migration which is very well documented here: https://docs.djangoproject.

Newfies-Dialer has several components that will need to be reinstalled / upgraded: the code of the Django project, the dependencies in virtualenv, the lua code base for the IVR and the dependencies for Lua.

In some versions, OS packages need to be upgraded too, so it might be wise to keep an eye on what is needed.

The Newfies-Dialer team provide an upgrade service as described at http://www.newfies-dialer.org/ where we try to limit your downtime. Do note that it may not be possible to upgrade from very old instances of Newfies-Dialer due to major architecture changes.

13.2 Text2Speech

13.2.1 How does the tag substitution work with the TTS engine

Answer:

This is the list of standard tags that will be automatically replaced: {last_name} {first_name} {email} {country} {city} {contact} // This is the phone number

If you need more flexibility, you can use the "Additional Parameters (JSON)" field which allow you to add custom key-values that will be replaced.

For example, let's add this in "Additional Parameters (JSON)": {"company_name": "Canonical", "bonus" : "200", "currency" : "euro"}

When you create a survey with a node that plays TTS, you can easily replace the key-values in the text. Text example:

> "We are calling you on behalf of {company_name}, you receive a bonus of {bonus} {currency}"

13.2.2 How does Newfies-Dialer provide TTS in multiple languages

Answer:

By default the TTS engine used by newfies-Dialer is Flite (http://www.speech.cs.cmu.edu/flite/) which only supports English. If you want to use another language you will need another TTS engine.

We have integrated Acapela: http://acapela-vaas.com/ and in order to use Acapela, the only thing you have to do is to sign in and enable Acapela on Newfies-dialer.

13.2.3 How to enable Acapela on Newfies-Dialer

Answer:

First you will have to sign-up and register an account with Acapela : http://acapela-vaas.com/ Once you signed up you will receive a login, an application login and an application password, you will need those to configure Acapela on Newfies-Dialer.

Acapela needs to be configured in 2 places:

1. On the Web interface

Edit the file /usr/share/newfies-dialer/newfies_dialer/settings_local.py and find:

```
#TEXT-TO-SPEECH
#==============
TTS_ENGINE = 'FLITE'  # FLITE, CEPSTRAL, ACAPELA

ACCOUNT_LOGIN = 'EVAL_XXXX'
APPLICATION_LOGIN = 'EVAL_XXXXXXX'
APPLICATION_PASSWORD = 'XXXXXXXX'

SERVICE_URL = 'http://vaas.acapela-group.com/Services/Synthesizer'
QUALITY = '22k'  # 22k, 8k, 8ka, 8kmu
ACAPELA_GENDER = 'W'
ACAPELA_INTONATION = 'NORMAL'

You will have to change the value of the settings : TTS_ENGINE, ACCOUNT_LOGIN, APPLICATION_LOGIN and
```

2. On the IVR application

Create a new file /usr/share/newfies-lua/libs/acapela_config.lua and add the following:

```
TTS_ENGINE = 'acapela'

ACCOUNT_LOGIN = 'EVAL_VAAS'
APPLICATION_LOGIN = 'EVAL_XXXXXX'
APPLICATION_PASSWORD = 'XXXXXX'

SERVICE_URL = 'http://vaas.acapela-group.com/Services/Synthesizer'
QUALITY = '22k'  -- 22k, 8k, 8ka, 8kmu
ACAPELA_GENDER = 'M'
ACAPELA_INTONATION = 'NORMAL'
ACAPELA_LANG = 'EN'


Change the value of the settings : ACCOUNT_LOGIN, APPLICATION_LOGIN, APPLICATION_PASSWORD and optiona
```

Finally restart the web UI::

```
/etc/init.d/supervisor stop
and
/etc/init.d/supervisor start
```

13.3 Dialer Logic

13.3.1 How does the dial retry logic works

Answer:

There are 2 systems available to retry calls:

Basic Retry:

> Basic retry checks to see if the call has been answered, even for a very short duration, this is regarded as a successful call. However if the call is not answered, busy or unreachable, it will be considered as a failed call, and will be retried as defined in the "Dialer Settings" section in the campaign.

Completion Retry:

> This feature works with the survey editor. When a survey node is created, the "Survey Complete" checkbox can be ticked so that it is clear that the contact reached a certain point in the survey and did not abandon the call early. Usually, one of the last nodes in the survey is ticked as "Survey Complete".
>
> The Completion Retry interval and number of times to retry is set in the campaign under the "Dialer Completion Settings" section.

CHAPTER 14

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index