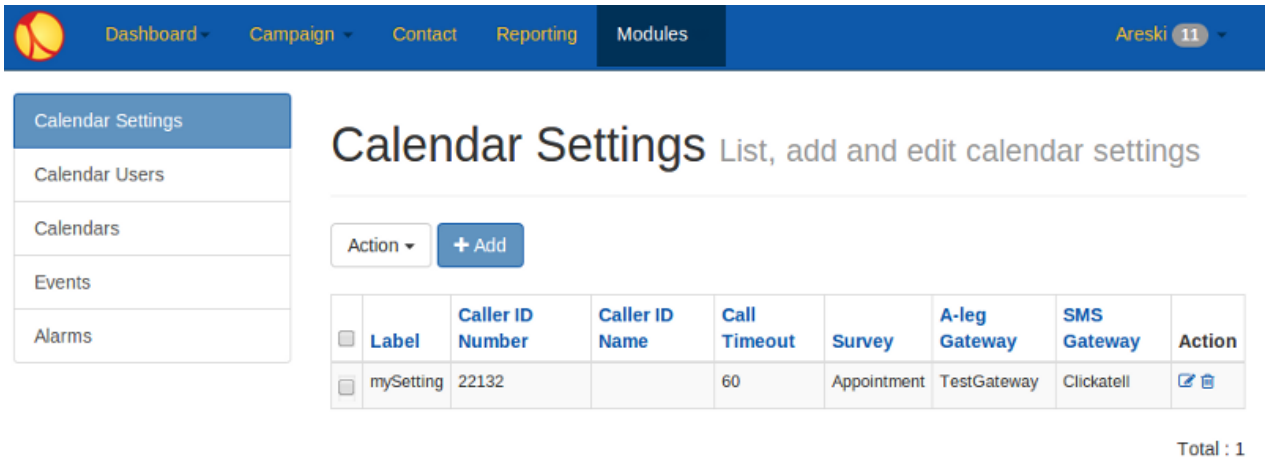
Appointment Module

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.



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 the customer who is creating the reminders or alarms. For instance, if you create a appointment reminders to 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.

TODO: add documentation on data fields (similar to additionals_data)

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

TODO: Explain how alarm result works!

Alarm Requests

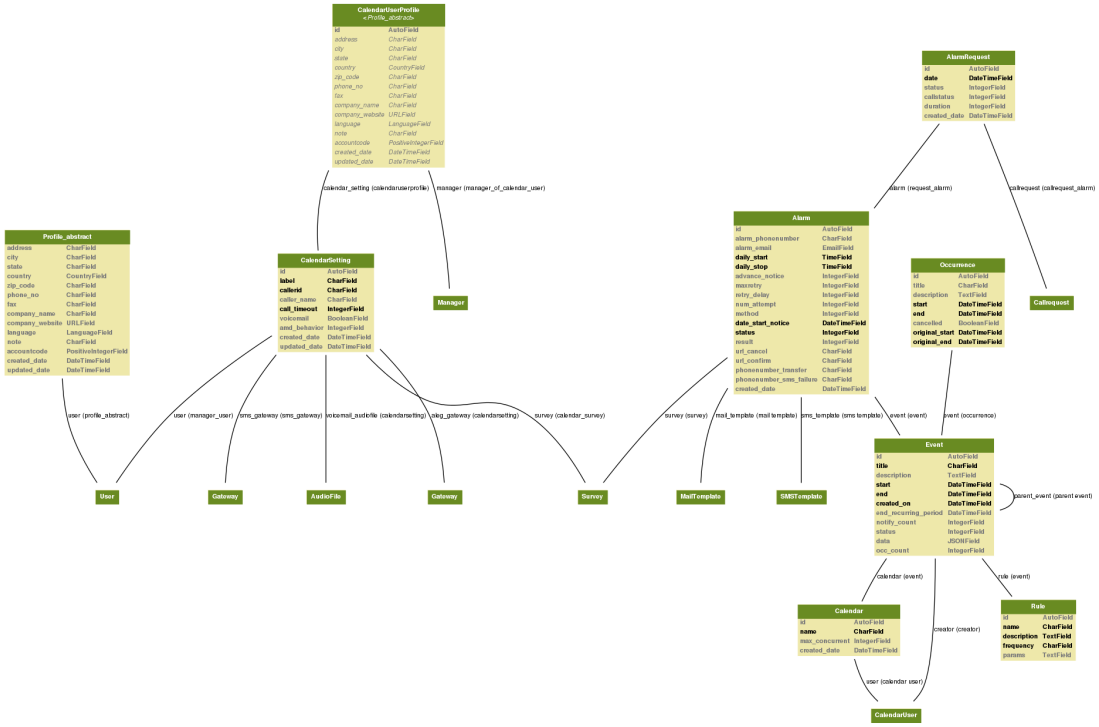
Keeps track of the alarms attempts, it's a useful resource to provide detailed logs.

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.

Models description

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



Explore the appointment models.

User cases

The appointment modules has been build with flexibility in mind, and we tried to cover several scenarios needed by different application/software willing to perform complex Appointment reminder application or Alarm system.

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.

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/

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
'{"label": "cal_setting", "callerid": "123456", "caller_name": "xyz", "user":
"http://127.0.0.1:8000/rest-api/user/2/", "survey": "http://127.0.0.1:8000/rest-api/sealed-
survey/1/", "aleg_gateway": "http://127.0.0.1:8000/rest-api/gateway/1/", "sms_gateway":
"http://127.0.0.1:8000/rest-api/sms-gateway/1/"}' http://localhost:8000/rest-api/calendar-
setting/
```

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
'{"username": "caluser3", "password": "caluser3", "email": "caluser3@gmail.com"}'
http://localhost:8000/rest-api/calendar-user/
```

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 '{"accountcode": "35365532", "calendar_setting": "3"}' http://localhost:8000/rest-
api/calendar-user-profile/6/
```

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
'{"name": "mynewcalendar", "max_concurrent": "1", "user": "http://127.0.0.1:8000/rest-
api/calendar-user/6/"}' http://localhost:8000/rest-api/calendar/
```

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
'{"title": "event_with_new_title", "start": "2013-12-10 12:34:43", "end": "2013-12-15
14:43:32", "creator": "http://127.0.0.1:8000/rest-api/calendar-user/6/",
"end_recurring_period": "2013-12-27 12:23:34", "calendar": "http://127.0.0.1:8000/rest-
api/calendar/4/", "status": "1"}' http://localhost:8000/rest-api/event/
```

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/get_list_child/
```

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; httponly; Max-Age=1209600; Path=/

{
  "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 '{"status": "3"}' http://127.0.0.1:8000/rest-api/event/5/update_last_child_status/
```

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; httponly; Max-Age=1209600; Path=/
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/rest-api/alarm-request/5/get_nested_alarm_request/
```

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; httponly; Max-Age=1209600; Path=/

{
  "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"
      },
    }
  }
}
```


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/