Installation Manual for Sendinel

Sending SMS and Phone Calls to Patients
http://www.sendinel.org

Potsdam, May 2010

[This page intentionally left blank]

Contents

1 What is Sendinel for?

Sendinel aims at improving the communication between clinics and patients. It allows clinics to reach the patients on their cell phones. For example the clinic staff can inform patients if their lab results or medicine have arrived. This quick information transfer avoids useless trips to the clinic for the patients and improves the treatment. The clinic staff can also regularly send messages to a group of people.For example, they can inform all subscribers to "Information about the Clinic" if a specialist is coming the next day.

2 Setting up Sendinel

2.1 Requirements

First, please make sure that the basic system requirements are met:

- a running Linux installation - for the automated installer Ubuntu or Debian is required. In this linux distribution at least Asterisk 1.6 and Python 2.5 have to be supported
- If the asterisk telephony server is not present yet, at least 150MB of free disk space are needed and 50MB have to be downloaded
- a working network connection
- an internet connection if the packages are to be installed from a web resource (at least 60MB have to be downloaded)
- a free USB port to connect the 3G Stick
- a Huawei E169 (also known as Vodafone K3520) - other devices will not work!

2.2 Preparation

Please make sure that you have inserted a working SIM-Card *without PIN protection* into the 3G stick. Connect the stick to the system.

3 The Automatic Installation Script

To use the automatic installation script your system has to run either Debian or Ubuntu. A internet connection is required - as a lot of data has to be downloaded a broadband connection is recommended. On Debian at least 50 MB have to be downloaded.

3.1 Getting the installation package

Either go to http://github.com/Sendinel/Sendinel/ and choose *Download Source* to get the Sendinel archive or use the direct link *http://github.com/Sendinel/Sendinel/tarball/master*.

To download the file you may use your preferred browser, like Firefox. You may also use the command line tool wget.

To do that, you first have to open a *terminal*, sometimes also called *command line*. On Debian it can be found in the *upper menu bar* → *Applications* → *Accessories* → *Terminal*

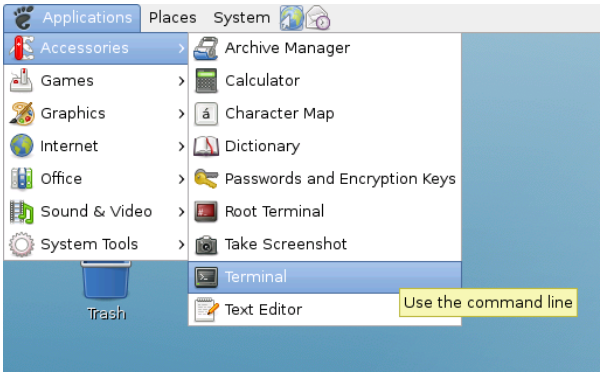

Figure 1: opening a terminal on Debian

After the terminal has been opened, you may create a new folder with *mkdir [Foldername]* or you may just use your home directory.

To download the Sendinel core files with wget, enter

```
wget http://github.com/Sendinel/Sendinel/tarball/master
```

and press the *Return* key on your keyboard.

The download progress will be displayed, as seen in this screenshot.

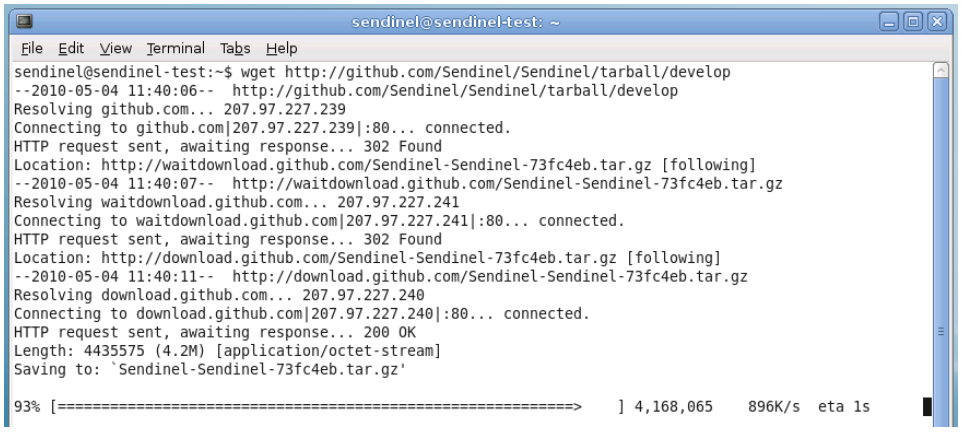

Figure 2: downloading Sendinel with wget

After the file has been downloaded, it has to be extracted. This is to be done with the command

```
tar -xzf [Name of the downloaded file]
```

You may also just enter *tar -xzf* and the first letters of the filename (S in this case - please pay attention to Uppercase/Lowercase) and press the *TAB*-key to let the terminal enter the right name. Afterwards use *cd* to change to the directory where you have just extracted the files. Also here you may use the auto completion feature of the terminal.

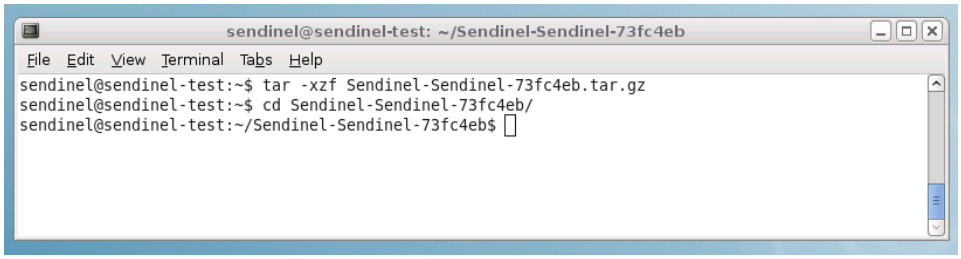

Figure 3: extracting the archive and changing the directory

If your are using debian, use *su* to become the superuser (root), which is equivalent to the *Administrator* on Windows.

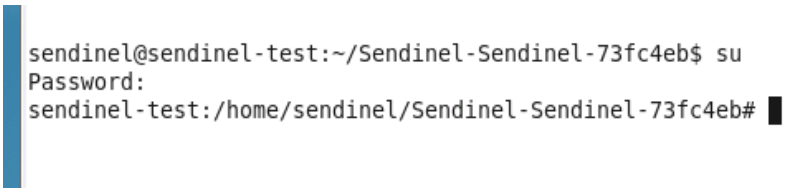

Figure 4: extracting the archive and changing the directory

3.2 Installing Sendinel

To start the installation enter

```
./install.sh
```

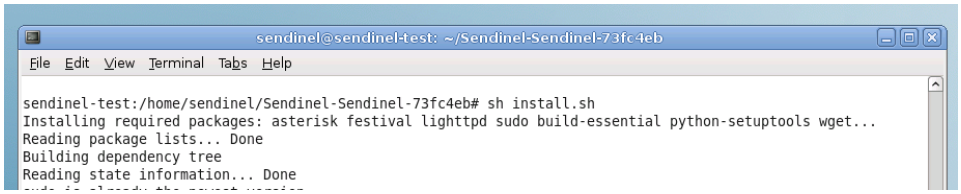

Figure 5: starting the Sendinel installer

After some seconds (this may also take a little bit longer), the system displays how much has to be downloaded and asks for confirmation. If it is okay, simply press the *Return* key and start the download.

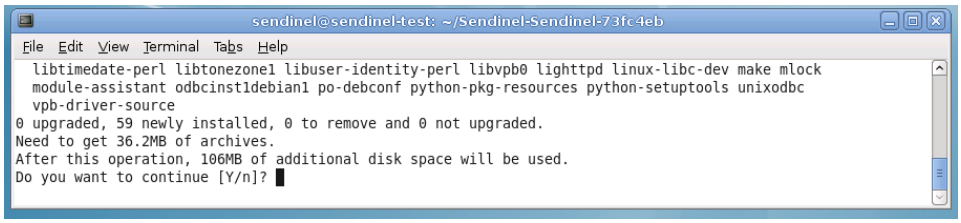

Figure 6: Download Confirmation

3.2.1 Individual Configuration

During the setup you will be asked if you want to create an *administrative user*. It is recommended to do so. Enter name, password and e-mail address.

Later, you will be asked a few questions concerning the individual configuration:

"Do you want the patient to authenticate against the system via ringing a mobile phone number (True/False)? "

- If you enter "True", every time patients want to register for a group or waiting list or want to enter an appointment, they have to call the system. This way Sendinel ensures, that the phone number really belongs to the patient. Then you will be asked: "Please enter the phone number of the SIM card in the 3G stick: ". Please do so.
- If you enter "False", phone numbers are not checked for correctness.

If you want to change your decisions, you can do that in the local_settings.py as explained in the section "Possible settings for Sendinel". Answering the first question will enter the settings variable ”AUTHENTICATION_ENABLED”, the second question fills ”AUTH_NUMBER”.

”Please enter the clinic’s name: ” Here you have to enter the name of the clinic. If you want to change the name later on, you can do that via the *admin interface* (see the corresponding section).

”Please enter your country’s calling code prefixed with two zeros (e.g. 0049 for Germany): ” This questions asks you for your country code (COUNTRY_CODE_PHONE): ’. If you don’t know it, it can be found on Wikipedia:
http://en.wikipedia.org/wiki/List_of_country_calling_codes. Do not enter the + but only the digits.

"Do you want to enable the bluetooth functionality - this requires Bluetooth hardware on client computers (True/False)? " This is the last question . If you enable bluetooth, the transfer of calendar entries via bluetooth to the patients’ mobile phones is added as way of communication. So when appointments are made and ’bluetooth’ is selected, the patient will be reminded by a calendar entry and not by an sms or a phonecall. If you want to change your decision, you need to use the *admin interface* (see the corresponding section).

The setup process may take some time and will tell you if errors occur. In that case, you will be asked if you want to continue with the installation. You should only continue, if you understood what went wrong.

4 (manual) off-line installation

If you are either not able to install from online package sources or you want to control the installation by yourself (maybe because you are using another Linux distribution), you should follow this guide.

4.1 installing the needed packages

Please make sure the following packages are installed on your system; on Ubuntu/Debian you may use *apt-get install* to install them

- asterisk
- festival
- lighttpd
- sudo
- build-essential
- python-setuptools
- wget

You should make sure you have asterisk 1.6.x installed.
The following python packages also have to be installed. For that you may use

```
easy\_install Django python-daemon lockfile}
```

If you get an error like lockfile-0.9 can't be found, use the following command: *easy_install Django python-daemon==1.5.5 lockfile*

4.2 Setting up festival text-to-speech synthesis

To get a working festival installation, installing the package should be enough. If you like to change the default voice, please refer to the festival documentation.

4.3 Configuring the Asterisk Telephony Server

4.3.1 the datacard channel

Download the latest revision of chan datacard from github:

```
http://github.com/thomasklingbeil/chan\_datacard
```

You can either use git or download a tarball of the source code from

```
http://github.com/thomasklingbeil/chan\_datacard/tarball/master
```

If you have downloaded the tarball you've got to untar it, using

```
tar -xvzf [name of the file]
```

Change the directory to the folder where you extracted the files and run the following commands make make install

If you want to configure the datacard by yourself, you can use the datacard.conf from the repository as a template. Else, you can also use the version from the Sendinel repository. In each case the file has to be copied to */etc/asterisk/datacard.conf*

4.3.2 Asterisk configuration files

If you have a plain Asterisk installation it is recommended to use the *extensions.conf* file from the Sendinel repository. It includes all necessary settings for the Asterisk server to run Sendinel.

Manual configuration: If authentication is to be used, an extension in the default call context has to be created which runs a special AGI script for each incoming call; This script created an entry into the Sendinel call log. If such an entry is already existing, extend it to run the AGI script.

```
[default]
...
exten => s,1,AGI(call_log.agi)
exten => s,n,Wait(20)
...
```

For sending outbound short messages, add the following extension into the default context:

```
exten => 2000,1,Answer()
exten => 2000,n,DatacardSendSMS(datacard0,${SmsNumber},${Text})
exten => 2000,n,Hangup()
```

To be able to send automatic outbound calls, the context *"outbound call"* has to be created. Here, you may also adapt the waiting times between the messages.

```
[outbound-call]
exten => s,1,Wait(2)
exten => s,n,Playback(${Salutation})
exten => s,n,Wait(1)
exten => s,n,Playback(${PassedInfo})
exten => s,n,Wait(1)
exten => s,n,Playback(${PassedInfo})
exten => s,n,Hangup()
```

To ensure the system does not hang when there is an incoming SMS on the data card, add the following context. Incoming short messages will be saved to */var/log/asterisk/sms.txt*. You may also change this path.

```
[datacard-incoming]
exten => sms,1,Verbose(Incoming SMS from ${SMSSRC} ${SMSTXT})
exten => sms,n,System(echo '${STRFTIME(${EPOCH},,%Y-%m-%d %H:%M:%S)} -
        ${CHANNEL} - ${SMSSRC}: ${SMSTXT}' >> /var/log/asterisk/sms.txt)

exten => sms,n,Hangup()
```

```
exten => cusd,1,Verbose(Incoming CUSD: ${CUSDTXT})
exten => cusd,n,System(echo '${STRFTIME(${EPOCH},,%Y-%m-%d %H:%M:%S)} -
        ${CHANNEL}: ${CUSDTXT}' >> /var/log/asterisk/cusd.txt)

exten => cusd,n,Hangup()
```

4.3.3 Other

Please ensure that the user the system is running as, is able to write to the asterisk spool directory (by default *ASTERISK_SPOOL_DIR* is configured as /var/spool/asterisk/outgoing). This can either be achieved by adding the user sendinel to the group *asterisk* or by granting *any* user the right of writen to that directory (chmod 777). The latter is usually a bad idea.

4.4 Installing Sendinel

Download the source code package from the project page. Extract it and move/copy the contained folder Sendinel to */opt/*, so that the file *manage.py* should reside in */opt/sendinel/manage.py*

Change the directory to */opt/sendinel* and run

```
python manage.py syncdb
```

You will be asked if you want to create an *administrative user*. It is recommended to do so. Enter name, password and e-mail address. After this step you may load some basic data into the system by running

```
python manage.py loaddata backend
```

4.5 The lighttpd server

An example configuration for the lighttpd server can be found in the folder *configs/lighttpd*. You can either copy those files to the corresponding folder (*/etc/lighttpd/*) or adapt your existing configuration.

At least you should replace the following text blocks in *10-sendinel.conf*:

- *%mediaPath%* with */opt/sendinel/sendinel/media*
- *%adminMediaPath%* with *DJANGO_DIR/contrib/admin/media* where DJANGO_DIR is the directory where django is installed

Now you should restart lighttpd:

```
/etc/init.d/lighttpd restart
```

4.6 Init-Scripts

Copy the init scripts *sendinel* and *sendinel-scheduler* from *configs/init-scripts/* to */etc/init.d/*.

Please adapt the scripts and replace *%targetDir%* with */opt/sendinel* and *%user%* with user sendinel.

run

```
update-rc.d sendinel defaults
```

and

```
update-rc.d sendinel-scheduler defaults
```

5 Bluetooth

Sendinel supports sending SMS and VoiceCalls by default. As an additional feature Bluetooth is also available for reminders but has to be turned on manually.

5.1 Enabling Bluetooth in Sendinel

5.2 Installing the Java-Client

For using Bluetooth a Bluetooth Server must be runnning on the client computer. To run this Bluetooth Server you have to download the installer first.

Before installation of the bluetooth client, make sure your bluetooth device at the client is working correctly.

1. Go to "'http://SENDINELSERVERIP/media/bluetooth/installbluetooth.exe'" by typing it in a browser and download the file (SENDINELSERVERIP must be replaced with the IP address you got at the Sendinel Server Installation
2. Open the file with a double click
3. Accept any security questions
4. If the intallation is done press the "'close'" button
5. After installing the downloaded file, restart the computer
6. If you can see a Security Question from the windows firewall at startup, press the "'do not block'" button to enable the bluetooth communication

Now the Bluetooth Server is running at the startup of the computer

If the bluetooth is not working correctly, this could be, because the bluetooth server is not started at the client. To do this manually, double click on the shortcut at the Desktop with the name "'SendinelBluetoothServer'"

6 Possible settings for Sendinel

To configure the system, create the file *local_settings.py* in the sendinel directory.

6.1 Country Settings

COUNTRY_CODE_PHONE Prefix of the country the system is going to be used in

START_MOBILE_PHONE Beginning of mobile phone numbers for validation purposes.

6.2 Asterisk Settings

ASTERISK_USER and ASTERISK_GROUP Group and user asterisk is running as. In most cases both is *asterisk*

ASTERISK_SPOOL_DIR The directory asterisk uses for scheduling calls. default: /var/spool/asterisk/outgoing/

ASTERISK_DATACARD select whether to use an external datacard for conducting the calls and the short messages. If False is selected, another SIP account has to be configured in the asterisk server manually. Sending short messages is only possible with a connected data card.
possible values: True/False (pay attention to the Upcase first letter) default: True

ASTERISK_EXTENSION asterisk extension to be used for outbound calls
default: *s*

ASTERISK_SIP_ACCOUNT SIP account to be used for asterisk for datacard use, this should be set to *ext-sip-account*, corresponding to the setting in *datacard.conf*

6.3 Text-to-Speech settings

FESTIVAL_CACHE Directory used for caching the speech sound files For use on a plug computer with Debian this should be set to */lib/init/rw*
default: *tmp*

6.4 Sendinel Settings

ENABLE_AUTHENTICATION Enable/Disable authentication for entered phone numbers. If used, also *AUTH NUMBER* has to be set.
possible values: True, False
default: False

AUTH_NUMBER This setting is only required if authentication is enabled. This entry has to be set to the number of the used SIM-Card/SIP-Account. It is used for display purposes only.

AUTHENTICATION_CALL_TIMEOUT This setting is only required if authentication is enabled. Time after which the authentication request gets invalid.
default: timedelta(minutes = 3)

CALL_SALUTATION This text is played before the information when a user is called.

SMS_SALUTATION This text is put in front of each short message sent through the system.

7 Administration Backend

To provide better control over the data that is saved by Sendinel, a administration backend (further called "admin") is provided. Following explained is the usage of the admin and some of the tasks one can accomplish with it.

7.1 Logging into Admin

To login, you have to open a web browser and enter the address of the admin in the address field. The address is serveradress/*admin* and usually looks something like this:

- http://localhost/admin
- http://192.168.1.1/admin

The address of the server is displayed at the end of the installation as well.

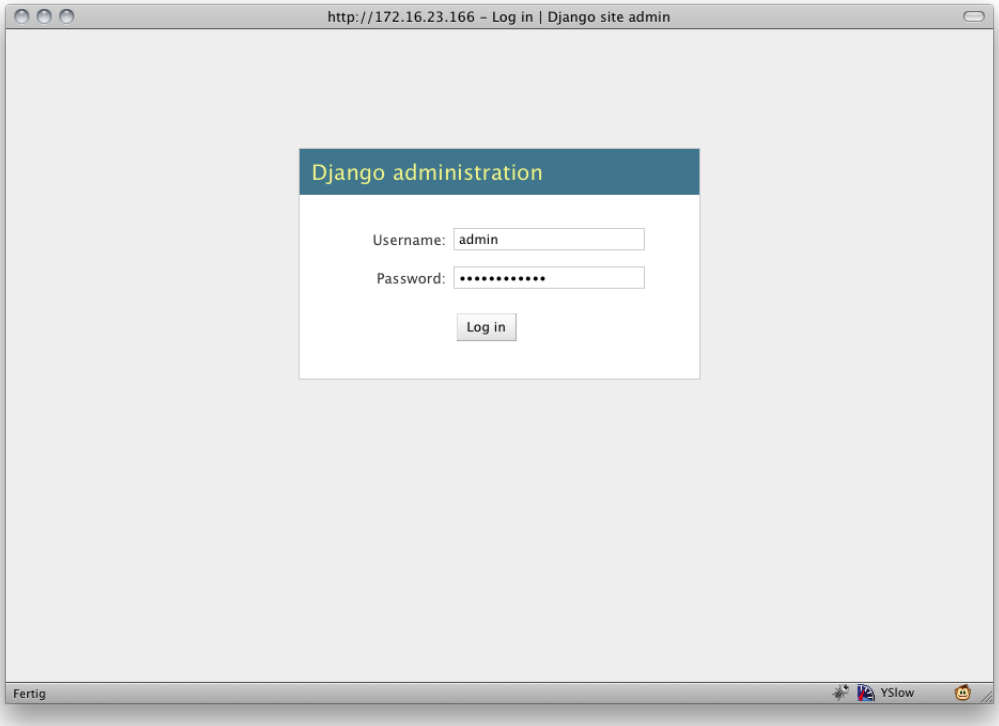

Figure 7: This screen should be displayed once you have accessed the admin.

To log-in you use the e-mail and password you have chosen during the installation.

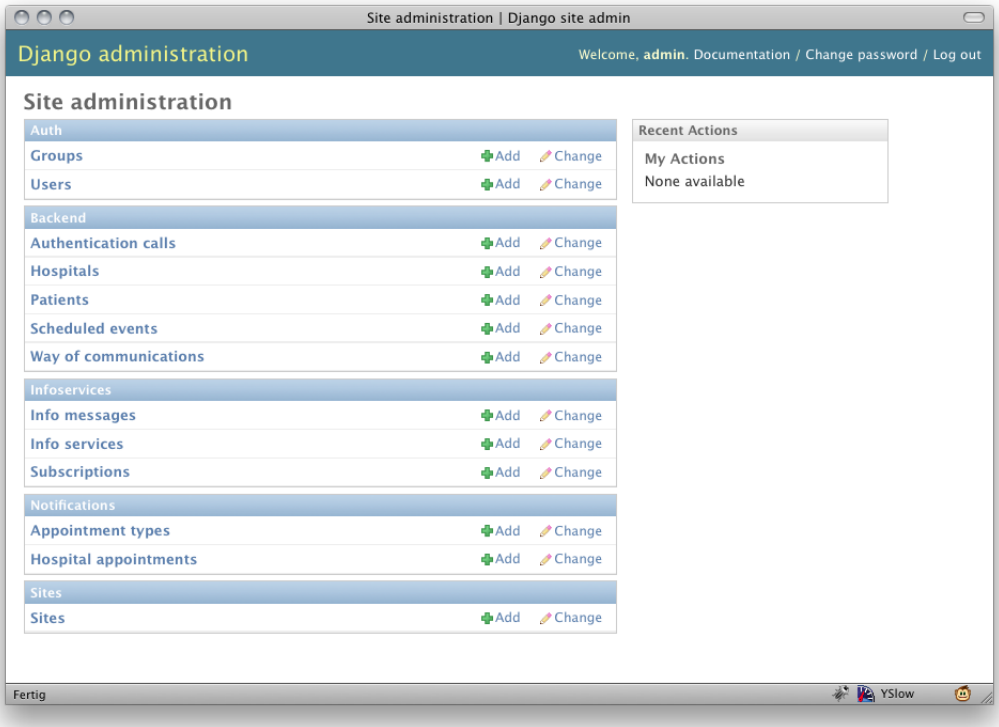

Figure 8: Once you are logged in, the screen should look like this.

You can see a list of data categories. Click on a name and you can get in, add new data or change and delete existing one. Please be careful since it is possible to break the entire system here, if you do something wrong.

Next are some common tasks, one can do in the admin:

7.2 Change Hospital Name

To change the hospitals name, click on "Hospitals" in the main menu. You then should see the following screen, possibly with another name than "your hospital". To change the name, click on the hospital and change the name in the following form. Click on "Save" in the right bottom corner and you are done.

If no hospital is displayed, click on "Add hospital" in the upper right corner. Enter the hospitals name. make sure to click "Current hospital" and click "Save".

Note: Best is if you always only have one hospital here. And this one hospital should always have "Current hospital" checked.

7.3 Add an appointment type

Sendinel offers the possibility to remind people of appointments in the hospital. There are different things patients should be reminded of. Sendinel already delivers three of the: "Remind of Follow-Up Consultation", "Remind of Vaccination" and "Inform about Lab Results". Each one is having it's own point in the main menu, it's own text that is send to the patient.

It is possible to add new types. In this example we will add "Remind of Dentist Consultation" to the appointment reminders.

Go to the main menu. Click on "Appointment types". There you can see a list of all currently available appointment reminder types. By default this list looks like this:

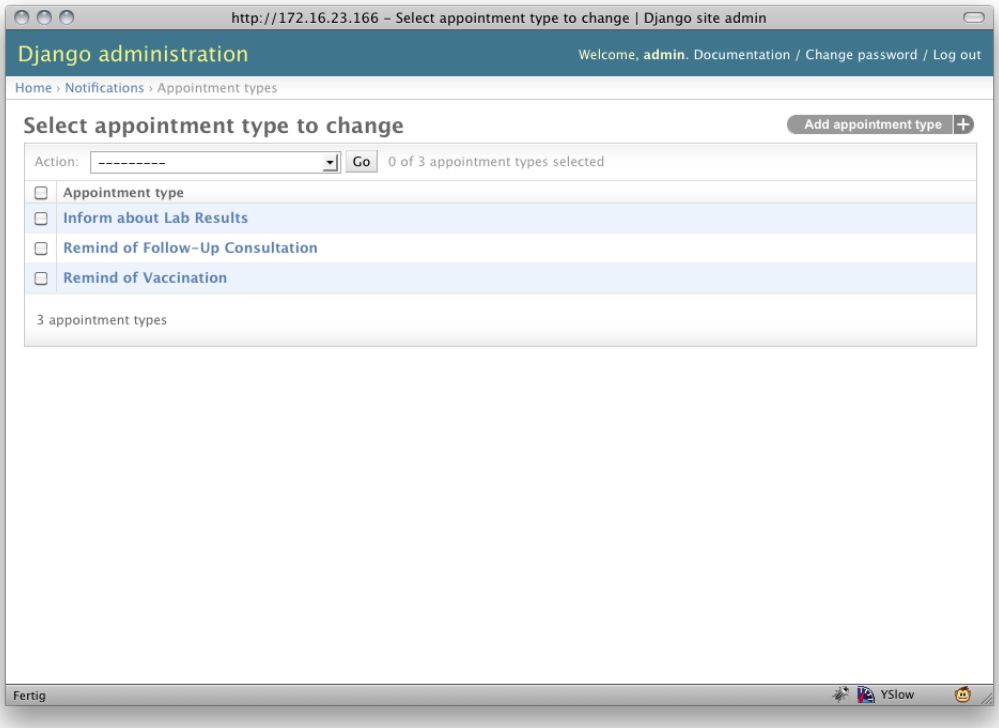

Figure 9: List of all default appointment types.

To add a new appointment type click on the button labeled "Add appointment type" in the upper right corner.

You can now have to enter three fields:

- *name*: a short word describing the type
- *verbose name*: the title which will be displayed in the main menu
- *template*: the text which will be sent via sms, voicecall or be displayed in the title of the bluetooth calendar entry

The template has two special words:

- *$hospital* will be replaced with the hospital name
- *$date* will be replaced with the date of the appointment

Be sure to include both special words. Also keep in mind, that the overall lenth of the used templates with the replaced special words shall not exceed 160 characters.

- *name*: dentist
- *verbose name*:Remind of Dentist Consultation
- *template*: please remember your dentist consultation at the $hospital on $date

In our example we add the following:

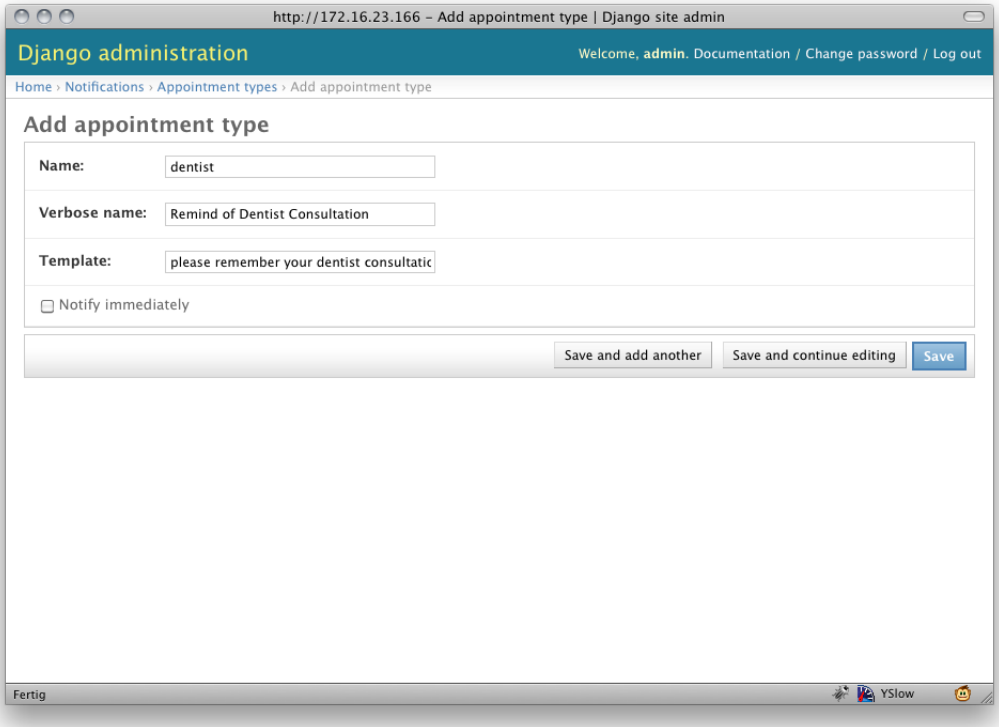

Figure 10: Adding a new appointment type.

The new appointment type now shows up in the Sendinel Main Menu as seen here:

Figure 11: The new appointment type "Remind of Dentist" appears in the main menu.

Of course you can also change the name and template text of existing appointment types or delete them in the appointment type menu in the administration backend.

7.4 Enabling/Disabling Bluetooth/SMS/Voice Call

Bluetooth is one way to remind patients of appointments. You can enable or disable the Bluetooth functionality in the admin.

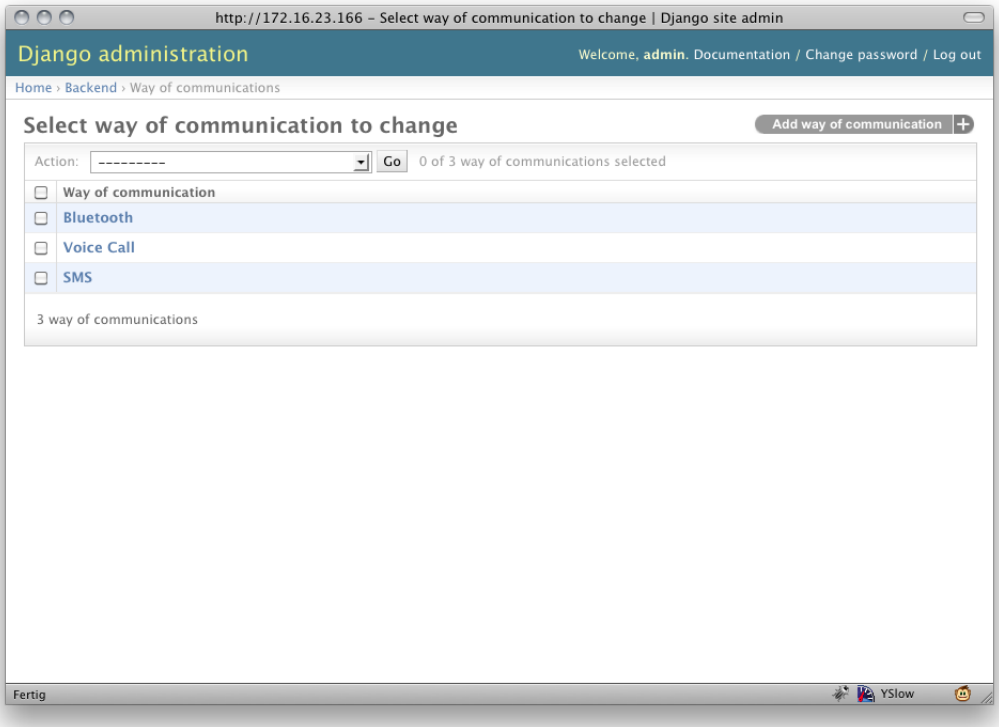

Figure 12: You can see all ways of communications in this list.

Click on "Way of communication" in the main menu. Click on Bluetooth afterwards. In the next menu you can enable or disable Bluetooth by toggling the box "Enabled". Click on Save afterwards.

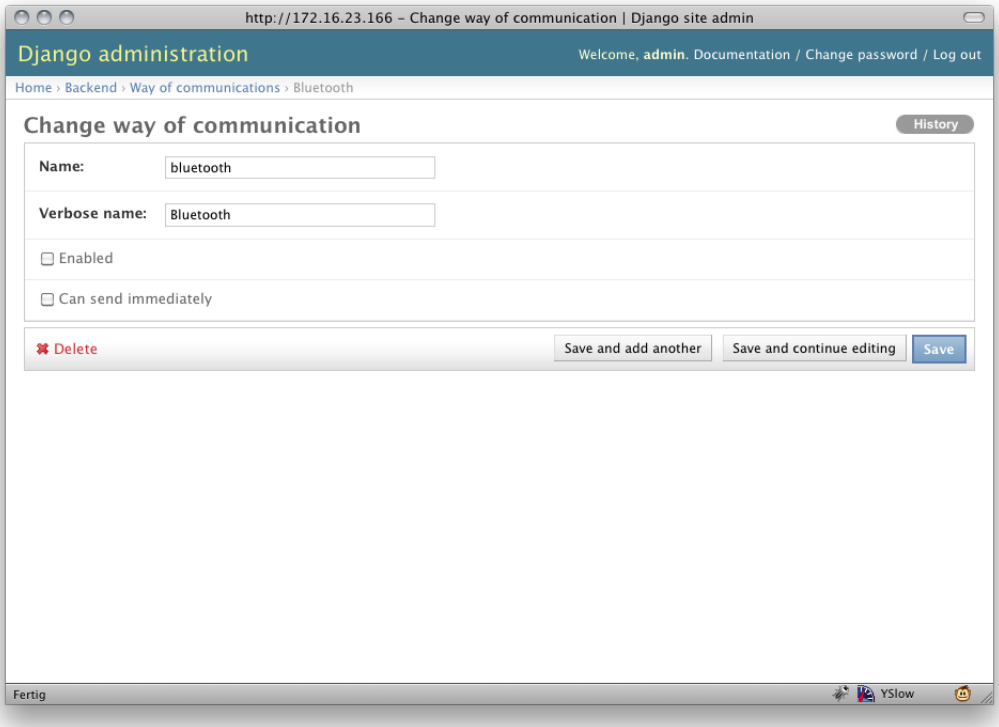

Figure 13: You can change the Bluetooth settings here.

The following data can be added:

- *Name*: a short word describing the way of communication
- *Verbose name*: the title of the way of communication, which will be displayed in the user interface
- *Enabled* If this checkbox is on, the way of communication is enabled
- *Can send immediately* If this checkbox is on, this way of communication can always be sent without the need of a patient present on-site

The same like with Bluetooth can be done for SMS and Voicecalls. In this way, you have a good control over what options for sending are available.