POWERPBX

PlaySMS Install Guide for Debian 11

playSMS [(https://playsms.org/)](https://playsms.org/)

PlaySMS is an open source SMS management system. It can be used as an SMS gateway, bulk SMS provider, personal messaging system, and for corporate and group communications.

It has built in gateway connectivity to various providers such as Twilio, Nexmo, BulkSMS, Clickatell, Infobip, RouteSMS etc. It also has a generic plugin that can be adapted to work with other providers.

Our changes from the original repository are shown at the following link.
https://github.com/playsms/playsms/compare/master...powerpbx:playsms:php81_fix
[(https://github.com/playsms/playsms/compare/master...powerpbx:playsms:php81_fix)](https://github.com/playsms/playsms/compare/master...powerpbx:playsms:php81_fix)

We have created a [custom gateway for Telnyx](https://github.com/powerpbx/plugin-telnyx)
[(https://github.com/powerpbx/plugin-telnyx)](https://github.com/powerpbx/plugin-telnyx) .

Tested using the following software:

> Debian 11 (Bullseye) x64 minimal install
> PlaySMS 1.5
> Apache 2.4
> PHP 8.1
> MariaDB 10

Prerequisites

Verify locale is set to `C.UTF-8` or `en US.UTF-8` .

```
locale
```

If it is not then set it now. You may also set your own UTF-8 locale.

```
# Select  C.UTF-8 UTF-8
apt update && apt -y install locales && dpkg-reconfigure locales
```

Log out/in or close/open shell for changes to take effect.

Install some basics

Ensure a basic server install with intial dependencies to start with.

```
apt update && apt -y upgrade
```

```
apt -y install nano git dbus sudo wget curl dirmngr openssl ntp
dnsutils mariadb-client mariadb-server postfix gettext apache2
```

Postfix

If a postfix configuration wizard pops up you can select the default
`Internet Site` and also the default mail name. These settings can
be manually changed later in `/etc/postfix/main.cf` .

Set Timezone

```
## FIND YOUR TIMEZONE
tzselect
```

```
## SET TIMEZONE EXAMPLE
timedatectl set-timezone America/Vancouver
```

```
systemctl restart rsyslog
timedatectl status
```

Install repository PHP

<u>FOR UBUNTU:</u> LC_ALL=C.UTF-8 sudo add-apt-repository ppa:ondrej/php

while for <u>Debian use:</u>

```
# install this section one line at a time.
apt -y install gnupg2 apt-transport-https ca-certificates
software-properties-common
wget -O /etc/apt/trusted.gpg.d/php.gpg
https://packages.sury.org/php/apt.gpg
echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main"
> /etc/apt/sources.list.d/php.list
```

Install PHP v8.1

```
apt update && apt -y install php8.1 php8.1-opcache php8.1-cli
php8.1-mysqli php8.1-gd php8.1-mbstring php8.1-xml php8.1-curl
php8.1-zip php8.1-fpm
```

```
update-alternatives --set php /usr/bin/php8.1
```

Configure Apache

```
cat >> /etc/apache2/sites-available/playsms.conf << EOF
<VirtualHost *:443>

  ServerAdmin webmaster@localhost
  ServerName playsms.somedomain.com
  # ServerAlias www.playsms.somedomain.com
  DocumentRoot /var/www/playsms

  SSLEngine on
  SSLCertificateFile    /etc/ssl/certs/ssl-cert-snakeoil.pem
  SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

</VirtualHost>

<Directory /var/www/playsms>
  Options SymLinksIfOwnerMatch Indexes
  DirectoryIndex index.php
  Require all granted
</Directory>

<Directory /var/www/storage>
  Require all denied
</Directory>
EOF
```

Enable website

```
a2dissite 000-default.conf
a2dissite default-ssl.conf
a2ensite playsms.conf
a2enmod proxy_fcgi ssl
a2enconf php8.1-fpm
systemctl restart php8.1-fpm
systemctl restart apache2
```

Install PlaySMS

This is using our fork which includes updates for PHP 8.1.

```
cd /usr/src
git clone -b php81_fix https://github.com/powerpbx/playsms.git
(https://github.com/powerpbx/playsms/tree/php81_fix)
cd playsms
```

```
DBPASS=somepassword
```

```
mysqladmin create playsms
mysql -e "GRANT ALL PRIVILEGES ON playsms.* TO
'playsms'@'localhost' IDENTIFIED BY '${DBPASS}';"
mysql -e "flush privileges;"
```

```
SERVER_IP=$(ifconfig | sed -En 's/127.0.0.*//;s/.*inet (addr:)?
(([0-9]*\.){3}[0-9]*).*/\2/p' | head -n1 | cut -d " " -f1)
```

```
cat >> /usr/src/playsms/install.conf << EOF
ADMINPASSWORD="admin"
DBUSER="playsms"
DBPASS="${DBPASS}"
DBNAME="playsms"
DBHOST="localhost"
DBPORT="3306"
URLWEB="https://${SERVER_IP}"
PATHSRC="/usr/src/playsms"
PATHWEB="/var/www/playsms"
PATHBIN="/usr/local/bin"
PATHLOG="/var/log/playsms"
PATHSTR="/var/www/storage"
PATHSRC="$(pwd)"
EOF
```

Check configuration before running script and modify as necessary.

```
nano /usr/src/playsms/install.conf
```

Run PlaySMS install script and answer `y` to all questions

```
/usr/src/playsms/install.sh
```

Create systemd file

```
cat >> /etc/systemd/system/playsms.service << EOF
[Unit]
Description=playsms
After=mariadb.service

[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/usr/local/bin/playsmsd start
ExecStop=/usr/local/bin/playsmsd stop

User=www-data
Group=www-data

[Install]
WantedBy=multi-user.target
EOF
```

```
chmod 755 /usr/local/bin/playsmsd
systemctl daemon-reload
```

```
systemctl enable playsms
systemctl restart playsms
systemctl status playsms
```

Post Install

Set ownership and permissions

Run this any time there are any changes/moves/adds/upgrades or if experiencing problems.

```
# Set owner and group to www-data
chown -R www-data. /var/log/playsms /var/www

# Directory permissions to 755 (u=rwx,g=rx,o='rx')
find /var/log/playsms -type d -exec chmod 755 {} \;
find /var/www -type d -exec chmod 755 {} \;

# File permissions to 644 (u=rw,g=r,o=r)
find /var/log/playsms -type f -exec chmod 644 {} \;
find /var/www -type f -exec chmod 644 {} \;
```

Firewall

```
apt -y install firewalld
```

```
firewall-cmd --permanent --zone=public --add-service={http,https}
firewall-cmd --reload
firewall-cmd --list-all
```
```
nano /usr/local/vin/playsmsd
modify this line adding:
$PLAYSMS_WEB = '/var/www/playsms';
save.
```
```
mysql_secure_installation
```

Answer `Y` to everything.

Login

browse to `https://x.x.x.x`

username: `admin`
password: `admin`

Browse to `My account > Preferences` and change password

Optional

Optimize PHP-FPM

```
sed -i "s/;request_terminate_timeout =
0/request_terminate_timeout = 300/"
/etc/php/8.1/fpm/pool.d/www.conf
sed -i "s/max_execution_time = 30/max_execution_time = 60/"
/etc/php/8.1/fpm/php.ini
sed -i "s/upload_max_filesize = 2M/upload_max_filesize = 20M/"
/etc/php/8.1/fpm/php.ini
sed -i "s/post_max_size = 8M/post_max_size = 20M/"
/etc/php/8.1/fpm/php.ini
sed -i "s/memory_limit = 128M/memory_limit = 512M/"
/etc/php/8.1/fpm/php.ini
```

```
systemctl restart php8.1-fpm
```

Setup SSL Certificate

This assumes you already have some domain or subdomain pointed at the server IP address. For example, `playsms.somedomain.com`.

Edit the `playsms.conf` to remove the port 443 virtualhost and add a port 80 virtualhost. So the new `playsms.conf` should look like the following

```
cat >> /etc/apache2/sites-available/playsms.conf << EOF
<VirtualHost *:80>
ServerName playsms.somedomain.com
DocumentRoot /var/www/playsms
</VirtualHost>

<Directory /var/www/playsms>
  Options SymLinksIfOwnerMatch Indexes
  DirectoryIndex index.php
  Require all granted
</Directory>

<Directory /var/www/storage>
  Require all denied
</Directory>
EOF
```

```
systemctl reload apache2
```

Install certbot and get letsencrypt certificate

```
apt install certbot python3-certbot-apache
certbot --apache -d playsms.somedomain.com
systemctl reload apache2
```

Update the web url if necessary

```
nano /var/www/playsms/appsetup.php
```

```
$core_config['http_path']['base'] =
'https://playsms.somedomain.com';
```

Test automatic renewal.

```
certbot renew --dry-run
```

Set up the automatic renewal routine to run weekly.

```
crontab -e
```

```
0 0 * * 0 root /usr/bin/certbot renew >/dev/null 2>&1
```

Disable Captcha

Captcha can be disabled at
`storage/application/plugin/core/auth/config.php` .

Adding Additional Gateways

Most gateways are not installed by default and are located in
`/usr/src/playsms/storage/application/plugin/gateway`
The generic gateway can be used as a basis for creating a new
gateway for some other provider.

Troubleshooting

To show all errors in `/var/log/apache2/error.log` , go to
`storage/custom/application/configs/config.php` and uncomment

```
error_reporting(E_ALL ^ (E_NOTICE | E_WARNING | E_DEPRECATED));
```

For verbose run time logging to `/var/log/playsms/playsms.log` ,
change the following setting just below the above one.

```
$core_config['logstate'] = 4;
```

Sections:

[Other](/category/sections/install/other) (/category/sections/install/other)

Install smstools by compiled:

sudo apt install smstools

and then

sudo update-rc.d -f smstools disable

sudo update-rc.d -f smstools remove

go on my repository and download as zip the fix:

provide to set for all folders/subfolder/files, 0777 permission and as user/group: www-data, and at end execute:

sudo systemctl daemon-reload

sudo update-rc.d -f sms3 defaults

OR better: Install smstools and comipile for debian 10/11/12 – Ubuntu 20/22/24

apt-get install build-essential libusb-1.0 libusb-1.0-0-dev build-essential manpages-dev

sudo apt-get update & sudo apt-get install usb-modeswitch

cd /tmp
git clone --depth=1 https://github.com/pappicio/smstools3
cd smstools3
make
make install
systemctl restart sms3

5.8. compiled smstools, now configure:

mkdir -p /var/log/sms/stats

mkdir -p /var/spool/sms/ {checked,failed,incoming,outgoing,sent}

mkdir /var/spool/sms/modem1

chown www-data:www-data -R /var/spool/sms

chmod 777 -R /var/spool/sms

mv /etc/smsd.conf /etc/smsd.conf.dist

configure your own, my is:

```
devices = modem1
loglevel = 7
# logfiles
stats = /var/log/sms/stats
logfile = /var/log/sms/smsd.log
# Default queue directory = /var/spool/sms
outgoing = /var/spool/sms/outgoing
checked = /var/spool/sms/checked
failed = /var/spool/sms/failed
incoming = /var/spool/sms/incoming
sent = /var/spool/sms/sent
delaytime = 2
errorsleeptime = 10
blocktime = 180
autosplit = 3
# Queue configurations

[queues]
modem1 = /var/spool/sms/modem1

[modem1]
device = /dev/ttyUSB1
init = AT^CURC=0
###init2 = AT+CPMS="ME","ME","ME"
#pin  = 1234
report = yes
incoming = yes
queues = modem1
# mode = new
smsc = 393770001016
baudrate = 115200 ###19200
memory_start = 0
decode_unicode_text = yes
#cs_convert = yes
report_device_details = no
```

...and at end...

A PERSISTENT DEVICE NAME, STEP BY STEP.
One FAQ on Gammu is Device name always changes on Linux, how to solve that? The answer given there is not very adequate. It took me quite some googling and trial & error to get it working. Hence this text.

We're using udev, Linux' device manager. It supports persistent device naming and it runs by default on any Linux distro. Just to verify:

ps -A | grep udev

should respond with something like

156 ? 00:00:00 udevd

Yup, it's running, process ID 156.

Find out a unique enough identifier set for the device.

lsusb

lists all USB devices. This is the output in my case:

Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 005: ID 12d1:1003 Huawei Technologies Co., Ltd. E220 HSDPA Modem / E230/
E270/E870 HSDPA/HSUPA Modem
Bus 001 Device 004: ID 7392:7811 Edimax Technology Co., Ltd EW-7811Un 802.11n Wireless
Adapter [Realtek RTL8188CUS]

OK, so the modem is connected and it's USB device 005 (this time). Note the numbers. 12d1 is the Vendor ID (Huawei Technologies) and 1003 is the Product ID, this particular model modem that they make. In this case, the combination Vendor ID + Product ID is already a unique enough identifier set. (So you could skip the next two steps.)

In other cases, things may be different. For instance, lots of devices use FTDI's USB-to-serial converter, all with the same Vendor and Product ID and you may want to have them all connected to you RPi. So you'll need some more information to distinguish one from the other(s).

To find that extra bit of information, enter

dmesg | grep usb

the response might be a long list. Somewhere in there, you'll find something like this:

[523.057081] usb 1-1.2: new high-speed USB device number 5 using dwc_otg
[523.168297] usb 1-1.2: New USB device found, idVendor=12d1, idProduct=1003
[523.168330] usb 1-1.2: New USB device strings: Mfr=2, Product=1, SerialNumber=0
[523.168346] usb 1-1.2: Product: HUAWEI Mobile
[523.168362] usb 1-1.2: Manufacturer: HUAWEI Technology

Note idVendor and idProduct. We'll use them later on. Note also SerialNumber=0. I don't trust serialnumbers that are zero. With other devices you should see a non-zero serial number, usually on a line of its own.

dmesg | grep ttyUSB

will list all ttyUSB devices, like so:

[6.546558] usb 1-1.2: GSM modem (1-port) converter now attached to ttyUSB0
[6.571094] usb 1-1.2: GSM modem (1-port) converter now attached to ttyUSB1

My modem occupies 2 ttyUSB's. By trial and error, I found out that sending and receiving works on ttyUSB1. I don't know why that is.

To get an idea of all the attributes of a USB-device, try:

udevadm info --name=/dev/ttyUSB1 --attribute-walk

This walks along a branch of USB devices and lists, for each device, all possible attributes in the udev rules key format.[2]. Try it! If your device has a serial number, you will see it there, and you will also see what it's called. That's important, because you'll have to call it by that name in the next step. Also, I always find it reassuring when different sources report the same information.

Almost done.

Create a file /etc/udev/rules.d/99-usb-serial.rules (yes you'll need root rights) with something like this line in it:
SUBSYSTEM=="tty", ATTRS{idVendor}=="12d1", ATTRS{idProduct}=="1003", SYMLINK +="kpn_dongel"

with, of course, your values for idVendor, idProduct and the symlink. You may need to add ATTR{serial}==blahblah to make it unique. Add a similar line for each device you want a symlink for. Note the comma's to separate each item. Forget one, and your rule won't work.

Now load the new rule. Rebooting is the crass way, this is much more elegant:

sudo udevadm trigger

Finally, test it. Three useful commands:

ls -l /dev/kpn_dongel

will give something like:

lrwxrwxrwx 1 root root 7 Jan 17 21:48 /dev/kpn_dongel -> ttyUSB1

ls -l /dev/ttyUSB1

Will give something similar to:

crw-rw---T 1 root dialout 188, 1 Jan 17 21:48 /dev/ttyUSB1

Note the ownerships and group memberships. Software will have to be aware of this, chown and chgrp may come in handy. But that's a different subject.

Finally, the output of this should point to the appropriate ttyUSB, and you'll recognise vendor name in there too.

udevadm test -a -p $(udevadm info -q path -n /dev/kpn_dongel)

Other way to assign ttyUSBX to modem:

The following script updates the modem device path in /etc/gammu-smsdrc whenever needed. You can configure it as cron task every 15 minutes or so

```bash
#!/usr/bin/env bash

log_file="/var/log/modem_update.log"
gammu_config_file="/etc/gammu-smsdrc"
modem=""
# Optional command when modem port change detected
optional_cmd="systemctl restart gammu-smsd"

function log {
    line="${1}"

    echo "${line}" >> "${log_file}"
    echo "${line}"
}

for tty in $(ls /dev/ttyUSB*); do
    log "Checking $tty"
    picocom -qrX -b 9600 $tty
    sleep 1
    result=$(echo "AT&F" | picocom -qrix 1000 $tty)
    if [ "${result}" = "AT&F" ]; then
        log "Found AT compatible modem at $tty"
        modem="${tty}"
    else
        log "No AT compatible modem at $tty"
    fi
done

curr_gammu_modem=$(cat "${gammu_config_file}" | grep "device = " | awk -F'='
'{gsub(/ /,""); print $2}')
if [ "${modem}" != "" ]; then
    if [ "${modem}" != "${curr_gammu_modem}" ]; then
        log "Updating modem from ${curr_gammu_modem} to ${modem}"
        sed -i "s%${curr_gammu_modem}%${modem}%" "${gammu_config_file}"
        if [ "${optional_cmd}" != "" ]; then
            log "Running ${optional_cmd}"
            ${optional_cmd} >> "${log_file}"
            if [ $? -ne 0 ]; then
                log "Could not restart service, exitcode: $?"
            fi
        fi
    fi
fi
```

MANAGE ACL per menu limitati per gli users/subusers:
per gli USERS:

inc=feature_phonebook,
inc=core_user&route=subuser_mgmnt,
inc=core_user&route=user_pref&op=user_pref,
inc=feature_queuelog&op=queuelog_list

o meglio ancora:

inc=feature_phonebook,
inc=core_user&route=subuser_mgmnt,
inc=core_user&route=user_pref&op=user_pref,
inc=feature_report&route=user_inbox&op=user_inbox,
inc=feature_queuelog&op=queuelog_list

per i subuser:

inc=core_sendsms,
inc=feature_report&route=user,
inc=feature_schedule,
inc=feature_msgtemplate,
inc=core_user&route=user_pref&op=user_pref,
inc=feature_queuelog

o meglio ancora:

inc=core_sendsms,
inc=feature_report&route=user,
inc=feature_schedule,
inc=core_user&route=user_pref&op=user_pref,
inc=feature_report&route=user_inbox&op=user_inbox,
inc=feature_queuelog

SUBUSERS CON PHONEBOOK:

inc=feature_phonebook,

inc=core_sendsms,

inc=feature_report&route=user,

inc=feature_schedule,

inc=feature_msgtemplate,

inc=core_user&route=user_pref&op=user_pref,

inc=feature_queuelog

o meglio ancora:

inc=feature_phonebook,

inc=core_sendsms,

inc=feature_report&route=user,

inc=feature_schedule,

inc=core_user&route=user_pref&op=user_pref,

inc=feature_report&route=user_inbox&op=user_inbox,

inc=feature_queuelog

Playsms trick on php-html files/functions

/plugin/core/user/fn.php

// commentando questa funzione si possono aggiungere quanti subuser (addetti all'INVIO SMS) si vuole, con lo stesso numero telefonico (magari quello del centralino COC).

//che è il minimo comun denominatore per vedere la rubrica che crea l'utente del compartimento!!!

Quindi basta aggiungere un contatto nella rubrica tipo "f_a_k_e_n_a_p_o_l_i_u_s_e_r" con lo stesso numero telefonico (081000000) e tutti gli utenti addetti all'invio SMS che saranno creati dall'user di turno, avranno lo stesso numero telefonico personale (esempio: 081000000), cosi inserito "fakenapoliuser" in rubrica e nel gruppo NAPOLI, potranno vedere la rubruca!!!

E seguendo questo discorso logico (aggiiungendo un nuovo gruppo con nuovi contatti e nuovi subuser / utente con stesso numero, esempio f_a_k_e_p_g_n_a_p_o_l_i (081111111)) si puo creare quante ribriche si vuole e ognuno vedra solo il feuppo cui fa parte (con il fakenumebr, diciamo!)

```
                // check mobile, must check for duplication only when filled
//              if ($ret['status'] && $data['mobile']) {
//                  if (dba_isexists(_DB_PREF_ . '_tblUser', array(
//                      'flag_deleted' => 0,
//                      'mobile' => $data['mobile']
//                  ), 'AND')) {
//                      if ($data['mobile'] != $existing['mobile']) {
//                          $ret['error_string'] = _('Account with this mobile already exists') . " (" . _('mobile') . ": " . $data['mobile'] . ")";
//                          $ret['status'] = false;
//                      }
//                  }
//              }
        }
```

Per dare tutti i gruppi imitati (ACL) anche ai subusers cambiare le seguenti righe di codice:

in: /plugin/core/user/subuser_mgmnt.php

```
//////$option_acl = _select('add_acl_id', array_flip(acl_getallbyuid($user_config['uid'])));
$option_acl = _select('add_acl_id', array_flip(acl_getall()));
```

e

in: /plugin/core/user/user_config.php

```
if ($user_edited['status'] == 4) {
                    $parent_id = user_getparentbyuid($user_edited['uid']);
                    if ($parent_id == $user_config['uid']) {
/////////$c_option_acl = array_flip(acl_getallbyuid($user_config['uid']));
        $c_option_acl = array_flip(acl_getall());
```

FACOLTATIVI (hidden by ACL)

per rendere readonly il telefono di un subuser (che ricordiamo deve essere fisso e uguale al fake-contatto in ribrica e nel gruppo, senno addio invio sms con ricerca….)

qui: /plugin/core/user/user_prefs.php

```
<tr>
<td>{{ Mobile }}</td>
<td><input type=text maxlength=20 name=up_mobile value="{{ mobile }}"
readonly></td>
</tr>
```

In pratica aggiungere "readonly"

per rendere readonly la "firma" del subuser qui:

in: /plugin/core/user/template/user_config.html

<tr>

<td>{{ Default message footer }}</td>

<td><input type=text maxlength=30 name=up_footer value="{{ footer }}" readonly> {{ HINT_MAX_ALPHANUMERIC }}

</td>

</tr>

In pratica aggiungere "readonly"

Per rendere la "firma" readonly anche su invio sms qui:

/plugin/core/semdsms/templates/sendsms.html

<label for="msg_footer">{{ Message footer }}</label>

<p>

<input type="text" name="sms_footer" id="msg_footer"

style="width: 100%" value="{{ sms_footer }}" readonly>

</p>

In pratica aggiungere "readonly".

Install Adminer project homepage

Newer versions are offered here, such as 4.2.1.

sudo mkdir /usr/share/adminer

sudo wget "http://www.adminer.org/latest.php" -O /usr/share/adminer/latest.php

sudo ln -s /usr/share/adminer/latest.php /usr/share/adminer/adminer.php

echo "Alias /adminer /usr/share/adminer/adminer.php" | sudo tee /etc/apache2/conf-available/adminer.conf

sudo a2enconf adminer.conf

Once the installation completes, restart Apache.

sudo service apache2 restart

At this point, the setup is complete. You can access Adminer at either of the following addresses.

http://[SERVER_IP]/adminer

or:

http://[SERVER_IP]/adminer.php