
Libsoundtouch Documentation

Release 0.5.0

Charles Blonde

Feb 11, 2018

Contents

1	Features	3
2	Usage	5
2.1	Installation	5
2.2	Discovery	5
2.3	Basic Usage	5
2.4	Multi-room	7
2.5	Websocket	7
3	API Documentation	9
3.1	Developer Interface	9
4	TODO	17
5	Releases	19
5.1	Versions	19
6	About Libsoundtouch	23
7	Contributors	25
	Python Module Index	27

This Python 2.7+/3.4+ library allows you to control [Bose Soundtouch devices](#) .

CHAPTER 1

Features

- Discovery
- power on/power off
- play/pause
- next/previous track
- volume setting (mute/set volume/volume up/volume down)
- repeat one/all/off
- shuffle on/off
- select AUX/Bluetooth inputs
- select preset (bookmark)
- playback selected music
- allow snapshot and restore playing content
- play HTTP URL (not HTTPS)
- Multi room (zones)
- Websocket notifications

2.1 Installation

```
pip install libsoundtouch
```

2.2 Discovery

Soundtouch devices support mDNS discovery protocol.

```
from libsoundtouch import discover_devices

devices = discover_devices(timeout=2)  # Default timeout is 5 seconds

for device in devices:
    print(device.config.name + " - " + device.config.type)
```

2.3 Basic Usage

```
from libsoundtouch import soundtouch_device
from libsoundtouch.utils import Source, Type

device = soundtouch_device('192.168.1.1')  # Manual configuration
device.power_on()

# Config object
print(device.config.name)

# Status object
```

```
# device.status() will do an HTTP request.
# Try to cache this value if needed.
status = device.status()
print(status.source)
print(status.artist+ " - " + status.track)
device.pause()
device.next_track()
device.play()

# Media Playback
# device.play_media(source, location, account, media_type)
# account and media_type are optionals

# Radio
device.play_media(Source.INTERNET_RADIO, '4712') # Studio Brussel

# Spotify
spot_user_id = '' # Should be filled in with your Spotify userID
# This userID can be found by playing Spotify on the
# connected SoundTouch speaker, and calling
# device.status().content_item.source_account
device.play_media(Source.SPOTIFY,
    'spotify:track:5J59VOgvclrhLDYUoH5OaW',
    spot_user_id) # Bazart - Goud

# Local music (Windows media player, Itunes)
# Account ID can be found by playing local music on the
# connected Soundtouch speaker, and calling
# device.status().content_item.source_account
account_id = device.status().content_item.source_account
device.play_media(Source.LOCAL_MUSIC,
    'album:1',
    account_id,
    Type.ALBUM)

# Snapshot current playing
device.snapshot()

# Select AUX input
device.select_source_aux()
# Select Bluetooth input
device.select_source_bluetooth()

# Restore previous snapshot
device.restore()

# Play an HTTP URL (not HTTPS)
device.play_url('http://fqdn/file.mp3')

# Volume object
# device.volume() will do an HTTP request.
# Try to cache this value if needed.
volume = device.volume()
print(volume.actual)
print(volume.muted)
device.set_volume(30) # 0..100

# Presets object
```

```
# device.presets() will do an HTTP request.
# Try to cache this value if needed.
presets = device.presets()
print(presets[0].name)
print(presets[0].source)
# Play preset 0
device.select_preset(presets[0])

# ZoneStatus object
# device.zone_status() will do an HTTP request.
# Try to cache this value if needed.
zone_status = device.zone_status()
print(zone_status.master_id)
print(len(zone_status.slaves))
```

2.4 Multi-room

Soundtouch devices supports multi-room features called zones.

```
from libsoundtouch import soundtouch_device

master = soundtouch_device('192.168.18.1')
slave1 = soundtouch_device('192.168.18.2')
slave2 = soundtouch_device('192.168.18.3')

# Create a new zone
master.create_zone([slave1, slave2])

# Remove a slave
master.remove_zone_slave([slave2])

# Add a slave
master.add_zone_slave([slave2])
```

2.5 Websocket

Soundtouch devices support Websocket notifications in order to prevent pulling and to get immediate updates.

```
from libsoundtouch import soundtouch_device
import time

# Events listeners

# Volume updated
def volume_listener(volume):
    print(volume.actual)

# Status updated
def status_listener(status):
    print(status.track)

# Presets updated
def preset_listener(presets):
```

```
    for preset in presets:
        print(preset.name)

# Zone updated
def zone_status_listener(zone_status):
    if zone_status:
        print(zone_status.master_id)
    else:
        print('no Zone')

device = soundtouch_device('192.168.18.1')

device.add_volume_listener(volume_listener)
device.add_status_listener(status_listener)
device.add_presets_listener(preset_listener)
device.add_zone_status_listener(zone_status_listener)

# Start websocket thread. Not started by default
device.start_notification()

time.sleep(600) # Wait for events
```

If you are looking for information on a specific function, class, or method, this part of the documentation is for you.

3.1 Developer Interface

This part of the documentation covers all the interfaces of Libsoundtouch.

3.1.1 Main Interface

`libsoundtouch.soundtouch_device` (*host*, *port=8090*)

Create a new Soundtouch device.

Parameters

- **host** – Host of the device
- **port** – Port of the device. Default 8090

`libsoundtouch.discover_devices` (*timeout=5*)

Discover devices on the local network.

Parameters **timeout** – Max time to wait in seconds. Default 5

3.1.2 Classes

Bose Soundtouch Device.

class `libsoundtouch.device.SoundTouchDevice` (*host*, *port=8090*, *ws_port=8080*,
dlna_port=8091)

Bose SoundTouch Device.

add_device_info_listener (*listener*)

Add a new device info updated listener.

add_presets_listener (*listener*)
Add a new presets updated listener.

add_status_listener (*listener*)
Add a new status updated listener.

add_volume_listener (*listener*)
Add a new volume updated listener.

add_zone_slave (*slaves*)
Add slave(s) to an existing zone (multi-room).
Zone must already exist and slaves array can not be empty.

Parameters slaves – List of slaves. Can not be empty

add_zone_status_listener (*listener*)
Add a new zone status updated listener.

clear_device_info_listeners ()
Clear device info updated listener..

clear_presets_listeners ()
Clear presets updated listeners.

clear_status_listener ()
Clear status updated listeners.

clear_volume_listeners ()
Clear volume updated listeners.

clear_zone_status_listeners ()
Clear zone status updated listeners.

config
Get config object.

create_zone (*slaves*)
Create a zone (multi-room) on a master and play on specified slaves.

Parameters slaves – List of slaves. Can not be empty

device_info_updated_listeners
Return Device Info Updated listeners.

dlna_port
Return DLNA port.

host
Host of the device.

mute ()
Mute/Un-mute volume.

next_track ()
Switch to next track.

pause ()
Pause.

play ()
Play.

play_media (*source, location, source_acc=None, media_type=<Type.URI: 'uri'>*)
Start music playback from a chosen source.

Parameters

- **source** – Source from which to play. Elements of Source enum.
- **location** – A unique uri or identifier. Represents the requested music from the source.
- **source_acc** – Source account. Imperative for some sources. For Spotify, this can be found by playing Spotify on the connected SoundTouch speaker, and calling: `device.status().content_item.source_account`
- **media_type** – Type of the requested music. Typical values are: “uri”, “track”, “album”, “playlist”. This can be found in `device.status().content_item.type`

play_pause()

Toggle play status.

play_url(url)

Start music playback from an HTTP URL.

Warning: HTTPS is not supported.

Parameters url – HTTP URL to play.

port

Return API port of the device.

power_off()

Power off device.

power_on()

Power on device.

presets(refresh=True)

Presets.

Parameters refresh – Force refresh, else return old data.

presets_updated_listeners

Return Presets Updated listeners.

previous_track()

Switch to previous track.

refresh_presets()

Refresh presets.

refresh_status()

Refresh status state.

refresh_volume()

Refresh volume state.

refresh_zone_status()

Refresh Zone Status.

remove_device_info_listener(listener)

Remove a new device info updated listener.

remove_presets_listener(listener)

Remove a new presets updated listener.

remove_status_listener(listener)

Remove a new status updated listener.

remove_volume_listener (*listener*)

Remove a new volume updated listener.

remove_zone_slave (*slaves*)

Remove slave(s) from an existing zone (multi-room).

Zone must already exist and slaves list can not be empty. Note: If removing last slave, the zone will be deleted and you'll have to create a new one. You will not be able to add a new slave anymore.

Parameters slaves – List of slaves to remove

remove_zone_status_listener (*listener*)

Remove a new zone status updated listener.

repeat_all ()

Repeat all.

repeat_off ()

Turn off repeat.

repeat_one ()

Repeat one. Doesn't work.

restore ()

Restore last snapshot.

select_content_item (*source*, *source_account=None*, *location=None*, *media_type=None*)

Select specified content.

:param source The source :param source_account The source account :param location The location :param media_type The media type

select_preset (*preset*)

Play selected preset.

:param preset Selected preset.

select_source_aux ()

Select AUX source.

select_source_bluetooth ()

Select BLUETOOTH source.

set_volume (*level*)

Set volume level: from 0 to 100.

shuffle (*shuffle*)

Shuffle on/off.

Parameters shuffle – Boolean on/off

snapshot ()

Snapshot current playing media.

start_notification ()

Start Websocket connection.

status (*refresh=True*)

Get status object.

Parameters refresh – Force refresh, else return old data.

status_updated_listeners

Return Status Updated listeners.

volume (*refresh=True*)

Get volume object.

Parameters **refresh** – Force refresh, else return old data.

volume_down ()

Volume down.

volume_up ()

Volume up.

volume_updated_listeners

Return Volume Updated listeners.

ws_port

Return Web Socket port.

zone_status (*refresh=True*)

Get Zone Status.

Parameters **refresh** – Force refresh, else return old data.

zone_status_updated_listeners

Return Zone Status Updated listeners.

class libsoundtouch.device.**Config** (*xml_dom*)

Soundtouch device configuration.

account_uuid

Account UUID.

components

Components.

country_code

Country code.

device_id

Device ID.

device_ip

Ip.

mac_address

Mac address.

module_type

Return module type.

name

Device name.

networks

Network.

region_code

Region code.

type

Device type.

variant

Variant.

variant_mode
Variant mode.

class libsoundtouch.device.**Network** (*network_dom*)
Soundtouch network configuration.

ip_address
IP Address.

mac_address
Mac Address.

type
Type.

class libsoundtouch.device.**Component** (*component_dom*)
Soundtouch component.

category
Category.

serial_number
Return serial number.

software_version
Software version.

class libsoundtouch.device.**Status** (*xml_dom*)
Soundtouch device status.

album
Album name.

artist
Artist.

content_item
Content item.

description
Description.

duration
Duration.

image
Image URL.

play_status
Status.

position
Position.

repeat_setting
Repeat setting.

shuffle_setting
Shuffle setting.

source
Source.

station_location

Station location.

station_name

Station name.

stream_type

Stream type.

track

Track.

track_id

Track id.

class libsoundtouch.device.**ContentItem**(*xml_dom*)

Content item.

is_presetable

Return true if presetable.

location

Location.

name

Name.

source

Source.

source_account

Source account.

type

Type.

class libsoundtouch.device.**Volume**(*xml_dom*)

Volume configuration.

actual

Actual volume level.

muted

Return True if volume is muted.

target

Target volume level.

class libsoundtouch.device.**Preset**(*preset_dom*)

Preset.

is_presetable

Return True if is presetable.

location

Location.

name

Name.

preset_id

Id.

source
Source.

source_account
Source account.

source_xml
XML source.

type
Type.

class libsoundtouch.device.**ZoneStatus** (*zone_dom*)
Zone Status.

is_master
Return True if current device is the zone master.

master_id
Master id.

master_ip
Master ip.

slaves
Zone slaves.

class libsoundtouch.device.**ZoneSlave** (*member_dom*)
Zone Slave.

device_ip
Slave ip.

role
Slave role.

3.1.3 Exceptions

exception libsoundtouch.device.**SoundtouchException**
Parent Soundtouch Exception.

exception libsoundtouch.device.**NoExistingZoneException**
Exception while trying to add slave(s) without existing zone.

exception libsoundtouch.device.**NoSlavesException**
Exception while managing multi-room actions without valid slaves.

CHAPTER 4

TODO

The following features are not yet implemented:

- Better error management
- Bass configuration

5.1 Versions

5.1.1 Version 0.8.0

Date 2018/02/11

- Fix: New API content with latest firmware
- Fix: Device names with UTF-8 characters
- Allow to select AUX/Bluetooth inputs
- Add snapshotting/restore feature

5.1.2 Version 0.7.2

Date 2017/07/05

- Fix: Add missing template

5.1.3 Version 0.7.1

Date 2017/07/05

- Fix: remove debug

5.1.4 Version 0.7.0

Date 2017/07/05

- Add play_url to play an HTTP URL (not HTTPS)

5.1.5 Version 0.6.2

Date 2017/06/21

- Fix: websocket source status in messages

5.1.6 Version 0.6.1

Date 2017/06/19

- Fix: Use enum-compat instead of enum34 directly

5.1.7 Version 0.6.0

Date 2017/06/17

- Add discovery (mDNS) support
- Official Python 3.6 support

5.1.8 Version 0.5.0

Date 2017/05/28

- Add Websocket support

5.1.9 Version 0.4.0

Date 2017/05/21

- Add Bluetooth source

5.1.10 Version 0.3.0

Date 2017/04/09

- Allow playing local computer media
- Fix issue with non ASCII characters

5.1.11 Version 0.2.2

Date 2017/02/07

- Fix status with non ascii characters in Python 2.7

5.1.12 Version 0.2.1

Date 2017/02/05

- Fix dependencies

5.1.13 Version 0.2.0

Date 2017/02/05

- Add *play_media* support

5.1.14 Version 0.1.0

Date 2016/11/20

- Initial release

CHAPTER 6

About Libsoundtouch

This library has been created in order to create a component for the [Home Assistant](#) project but is totally independent.

CHAPTER 7

Contributors

Thanks to all the wonderful folks who have contributed to Libsoundtouch:

- jeanregisser <<https://github.com/jeanregisser>> (Use enum-compat instead of enum34 directly)
- Tyzer34 <<https://github.com/Tyzer34>> (add *play_media* support)
- wanderor <<https://github.com/wanderor>> (add local computer media support)
- obadz <<https://github.com/obadz>> (add Bluetooth source)
- luca-angemi <<https://github.com/luca-angemi>> (Fix new firmware error)
- vanto <<https://github.com/vanto>> (Fix device names with UTF-8 characters)

I

`libsoundtouch`, 9

`libsoundtouch.device`, 9

A

account_uuid (libsoundtouch.device.Config attribute), 13
 actual (libsoundtouch.device.Volume attribute), 15
 add_device_info_listener() (libsoundtouch.device.SoundTouchDevice method), 9
 add_presets_listener() (libsoundtouch.device.SoundTouchDevice method), 9
 add_status_listener() (libsoundtouch.device.SoundTouchDevice method), 10
 add_volume_listener() (libsoundtouch.device.SoundTouchDevice method), 10
 add_zone_slave() (libsoundtouch.device.SoundTouchDevice method), 10
 add_zone_status_listener() (libsoundtouch.device.SoundTouchDevice method), 10
 album (libsoundtouch.device.Status attribute), 14
 artist (libsoundtouch.device.Status attribute), 14

C

category (libsoundtouch.device.Component attribute), 14
 clear_device_info_listeners() (libsoundtouch.device.SoundTouchDevice method), 10
 clear_presets_listeners() (libsoundtouch.device.SoundTouchDevice method), 10
 clear_status_listener() (libsoundtouch.device.SoundTouchDevice method), 10
 clear_volume_listeners() (libsoundtouch.device.SoundTouchDevice method), 10
 clear_zone_status_listeners() (libsound-

touch.device.SoundTouchDevice method), 10

Component (class in libsoundtouch.device), 14
 components (libsoundtouch.device.Config attribute), 13
 Config (class in libsoundtouch.device), 13
 config (libsoundtouch.device.SoundTouchDevice attribute), 10
 content_item (libsoundtouch.device.Status attribute), 14
 ContentItem (class in libsoundtouch.device), 15
 country_code (libsoundtouch.device.Config attribute), 13
 create_zone() (libsoundtouch.device.SoundTouchDevice method), 10

D

description (libsoundtouch.device.Status attribute), 14
 device_id (libsoundtouch.device.Config attribute), 13
 device_info_updated_listeners (libsoundtouch.device.SoundTouchDevice attribute), 10
 device_ip (libsoundtouch.device.Config attribute), 13
 device_ip (libsoundtouch.device.ZoneSlave attribute), 16
 discover_devices() (in module libsoundtouch), 9
 dlina_port (libsoundtouch.device.SoundTouchDevice attribute), 10
 duration (libsoundtouch.device.Status attribute), 14

H

host (libsoundtouch.device.SoundTouchDevice attribute), 10

I

image (libsoundtouch.device.Status attribute), 14
 ip_address (libsoundtouch.device.Network attribute), 14
 is_master (libsoundtouch.device.ZoneStatus attribute), 16
 is_presetable (libsoundtouch.device.ContentItem attribute), 15
 is_presetable (libsoundtouch.device.Preset attribute), 15

L

libsoundtouch (module), 9

libsoundtouch.device (module), 9
location (libsoundtouch.device.ContentItem attribute), 15
location (libsoundtouch.device.Preset attribute), 15

M

mac_address (libsoundtouch.device.Config attribute), 13
mac_address (libsoundtouch.device.Network attribute), 14
master_id (libsoundtouch.device.ZoneStatus attribute), 16
master_ip (libsoundtouch.device.ZoneStatus attribute), 16
module_type (libsoundtouch.device.Config attribute), 13
mute() (libsoundtouch.device.SoundTouchDevice method), 10
muted (libsoundtouch.device.Volume attribute), 15

N

name (libsoundtouch.device.Config attribute), 13
name (libsoundtouch.device.ContentItem attribute), 15
name (libsoundtouch.device.Preset attribute), 15
Network (class in libsoundtouch.device), 14
networks (libsoundtouch.device.Config attribute), 13
next_track() (libsoundtouch.device.SoundTouchDevice method), 10
NoExistingZoneException, 16
NoSlavesException, 16

P

pause() (libsoundtouch.device.SoundTouchDevice method), 10
play() (libsoundtouch.device.SoundTouchDevice method), 10
play_media() (libsoundtouch.device.SoundTouchDevice method), 10
play_pause() (libsoundtouch.device.SoundTouchDevice method), 11
play_status (libsoundtouch.device.Status attribute), 14
play_url() (libsoundtouch.device.SoundTouchDevice method), 11
port (libsoundtouch.device.SoundTouchDevice attribute), 11
position (libsoundtouch.device.Status attribute), 14
power_off() (libsoundtouch.device.SoundTouchDevice method), 11
power_on() (libsoundtouch.device.SoundTouchDevice method), 11
Preset (class in libsoundtouch.device), 15
preset_id (libsoundtouch.device.Preset attribute), 15
presets() (libsoundtouch.device.SoundTouchDevice method), 11
presets_updated_listeners (libsoundtouch.device.SoundTouchDevice attribute), 11

previous_track() (libsoundtouch.device.SoundTouchDevice method), 11

R

refresh_presets() (libsoundtouch.device.SoundTouchDevice method), 11
refresh_status() (libsoundtouch.device.SoundTouchDevice method), 11
refresh_volume() (libsoundtouch.device.SoundTouchDevice method), 11
refresh_zone_status() (libsoundtouch.device.SoundTouchDevice method), 11
region_code (libsoundtouch.device.Config attribute), 13
remove_device_info_listener() (libsoundtouch.device.SoundTouchDevice method), 11
remove_presets_listener() (libsoundtouch.device.SoundTouchDevice method), 11
remove_status_listener() (libsoundtouch.device.SoundTouchDevice method), 11
remove_volume_listener() (libsoundtouch.device.SoundTouchDevice method), 11
remove_zone_slave() (libsoundtouch.device.SoundTouchDevice method), 12
remove_zone_status_listener() (libsoundtouch.device.SoundTouchDevice method), 12
repeat_all() (libsoundtouch.device.SoundTouchDevice method), 12
repeat_off() (libsoundtouch.device.SoundTouchDevice method), 12
repeat_one() (libsoundtouch.device.SoundTouchDevice method), 12
repeat_setting (libsoundtouch.device.Status attribute), 14
restore() (libsoundtouch.device.SoundTouchDevice method), 12
role (libsoundtouch.device.ZoneSlave attribute), 16

S

select_content_item() (libsoundtouch.device.SoundTouchDevice method), 12
select_preset() (libsoundtouch.device.SoundTouchDevice method), 12

[select_source_aux\(\)](#) (libsoundtouch.device.SoundTouchDevice method), [12](#)
[select_source_bluetooth\(\)](#) (libsoundtouch.device.SoundTouchDevice method), [12](#)
[serial_number](#) (libsoundtouch.device.Component attribute), [14](#)
[set_volume\(\)](#) (libsoundtouch.device.SoundTouchDevice method), [12](#)
[shuffle\(\)](#) (libsoundtouch.device.SoundTouchDevice method), [12](#)
[shuffle_setting](#) (libsoundtouch.device.Status attribute), [14](#)
[slaves](#) (libsoundtouch.device.ZoneStatus attribute), [16](#)
[snapshot\(\)](#) (libsoundtouch.device.SoundTouchDevice method), [12](#)
[software_version](#) (libsoundtouch.device.Component attribute), [14](#)
[soundtouch_device\(\)](#) (in module libsoundtouch), [9](#)
[SoundTouchDevice](#) (class in libsoundtouch.device), [9](#)
[SoundtouchException](#), [16](#)
[source](#) (libsoundtouch.device.ContentItem attribute), [15](#)
[source](#) (libsoundtouch.device.Preset attribute), [15](#)
[source](#) (libsoundtouch.device.Status attribute), [14](#)
[source_account](#) (libsoundtouch.device.ContentItem attribute), [15](#)
[source_account](#) (libsoundtouch.device.Preset attribute), [16](#)
[source_xml](#) (libsoundtouch.device.Preset attribute), [16](#)
[start_notification\(\)](#) (libsoundtouch.device.SoundTouchDevice method), [12](#)
[station_location](#) (libsoundtouch.device.Status attribute), [14](#)
[station_name](#) (libsoundtouch.device.Status attribute), [15](#)
[Status](#) (class in libsoundtouch.device), [14](#)
[status\(\)](#) (libsoundtouch.device.SoundTouchDevice method), [12](#)
[status_updated_listeners](#) (libsoundtouch.device.SoundTouchDevice attribute), [12](#)
[stream_type](#) (libsoundtouch.device.Status attribute), [15](#)

T

[target](#) (libsoundtouch.device.Volume attribute), [15](#)
[track](#) (libsoundtouch.device.Status attribute), [15](#)
[track_id](#) (libsoundtouch.device.Status attribute), [15](#)
[type](#) (libsoundtouch.device.Config attribute), [13](#)
[type](#) (libsoundtouch.device.ContentItem attribute), [15](#)
[type](#) (libsoundtouch.device.Network attribute), [14](#)
[type](#) (libsoundtouch.device.Preset attribute), [16](#)

V

[variant](#) (libsoundtouch.device.Config attribute), [13](#)

[variant_mode](#) (libsoundtouch.device.Config attribute), [13](#)
[Volume](#) (class in libsoundtouch.device), [15](#)
[volume\(\)](#) (libsoundtouch.device.SoundTouchDevice method), [12](#)
[volume_down\(\)](#) (libsoundtouch.device.SoundTouchDevice method), [13](#)
[volume_up\(\)](#) (libsoundtouch.device.SoundTouchDevice method), [13](#)
[volume_updated_listeners](#) (libsoundtouch.device.SoundTouchDevice attribute), [13](#)

W

[ws_port](#) (libsoundtouch.device.SoundTouchDevice attribute), [13](#)

Z

[zone_status\(\)](#) (libsoundtouch.device.SoundTouchDevice method), [13](#)
[zone_status_updated_listeners](#) (libsoundtouch.device.SoundTouchDevice attribute), [13](#)
[ZoneSlave](#) (class in libsoundtouch.device), [16](#)
[ZoneStatus](#) (class in libsoundtouch.device), [16](#)