Bose SoundTouch Webservices API

Bose Corporation, Home Entertainment Product Division

Version 1.0.1

Contents

1 Document Version History

Version	*Release Date*	*Description of Changes*
1.0.0	December 5, 2014	• Initial Release
1.0.1	December 17, 2014	• Section 3 updated with a link to the License Agreement • Updated incorrect variable names to remove errant "\" in sections: 6.8, 6.9, 6.11, 6.12, 6.15, 7.5 • Corrected WebSockets port to 8080 (previous version incorrectly listed 8090) • Title/description corrections for section 7.2

2 Acronyms and Definitions

Acronyms	*Expanded Term*	*Definition*
API	Application Programming Interface	A definition for how to interact with and use a software component
REST	Representational State Transfer	A common type of web service API that is modeled around resources
WS API	Webservices API	An API made available by a web server
SSDP	Simple Services Discovery Protocol	A discovery protocol that uses unicast and multicast over UDP
MDNS	Multicast Domain Name System	A type of discovery protocol that requires zero configuration
	Bonjour	Apple's implementation of MDNS

3 Contact Info/Legal

For any questions, comments, or suggestions for improvements please email us at SoundTouchAPI@bose.com

Use of this API material is subject to the API License Agreement, which can be found at developers.bose.com/SoundTouch-API-License

4 Overview

These commands are the primary interface to command and control a Bose SoundTouch. They are sent over HTTP on port 8090 to the SoundTouch device you would like to connect to using the GET and POST methods.

4.1 Special types used by the SoundTouch WSAPI

```
ART_STATUS {
   INVALID
   SHOW_DEFAULT_IMAGE
   DOWNLOADING
   IMAGE_PRESENT
}

BOOL: "true" or "false"

INT: a 32-bit integer
```

```
IPADDR: an IP address, represented as a string

KEY_VALUE {
    PLAY
    PAUSE
    STOP
    PREV_TRACK
    NEXT_TRACK
    THUMBS_UP
    THUMBS_DOWN
    BOOKMARK
    POWER
    MUTE
    VOLUME_UP
    VOLUME_DOWN
    PRESET_1
    PRESET_2
    PRESET_3
    PRESET_4
    PRESET_5
    PRESET_6
    AUX_INPUT
    SHUFFLE_OFF
    SHUFFLE_ON
    REPEAT_OFF
    REPEAT_ONE
    REPEAT_ALL
    PLAY_PAUSE
    ADD_FAVORITE
    REMOVE_FAVORITE
    INVALID_KEY
}

MACADDR: a MAC address, upcased, represented as a string

PLAY_STATUS {
    PLAY_STATE
    PAUSE_STATE
    STOP_STATE
    BUFFERING_STATE
    INVALID_PLAY_STATUS
}

PRESET_ID: An integer, 1 through 6 inclusive

SOURCE {
    INVALID_SOURCE
    SLAVE_SOURCE
    INTERNET_RADIO
    PANDORA
    AIRPLAY
    STORED_MUSIC
    AUX
    OFF_SOURCE
    CURRATED_RADIO
    STANDBY
    UPDATE
    DEEZER
    SPOTIFY
    IHEART
}

SOURCE_STATUS {
    UNAVAILABLE
    READY
}

STRING: any valid XML-escaped string

UINT: a 32-bit unsigned integer

UINT64: a 64-bit unsigned integer

URL: a URL, encoded as a string

Any get* command results in a HTTP GET command
Any set* command results in a HTTP POST command, i.e. requires a payload
```

5 General Status and Errors

For calls that do not have a special return payload, the default response is:

```
<status>$STRING</status>
```

For calls that can produce errors, the error response is:

```
<errors deviceID="$STRING">
    <error value="$INT" name="$STRING" severity="$STRING">$STRING</error>
    ...
</errors>
```

For malformed requests, i.e., wrong value the response is:

```
<application_octet-stream/>
```

6 API Methods/URLs

6.1 /bass

Description: Sets or gets the current bass setting for a particular speaker. This may or may not be a supported capability, use the /bassCapabilities to find out whether a speaker supports bass configuration

Get or set bass

GET:

```
<bass deviceID="$MACADDR">
    <targetbass>$INT</targetbass>
    <actualbass>$INT<actualbass>
</bass>target<bass>
```

POST:

```
<bass deviceID="$MACADDR">
<targetbass>$INT</targetbass> <actualbass><ITacs>aarget<bass>3</targetbass>
    <actualbass>-3</actualbass> </</bass>
```

6.2 /bassCapabilities

Description: Some speakers do not support the ability to customize the bass levels, use this to find out whether bass customization is supported

Get or set bass Capabilities

GET:

```
<bassCapabilities deviceID="$MACADDR">
<bassAvailable>true</bassAvailable>
<bassMin>-9</bassMin>
<bassMax>0</bassMax>
<bassDefault>0</bassDefault>
</bassCapabilities>
```

POST:

```
<bassCapabilities deviceID="$MACADDR">
<bassAvailable>true</bassAvailable>
<bassMin>-9</bassMin>
<bassMax>0</bassMax>
<bassDefault>0</bassDefault>
</bassCapabilities>
```

6.3 /getZone

Description:

Gets the current state of the multi-room zone from particular device

GET:

```
<zone master="$MACADDR">
<member ipaddress="$IPADDR">"$MACADDR"</member>
<member ipaddress="slave1 $IPADDR">"slave1 MACADDR"</member>
...
</zone>
```

6.4 /select

Description:

Plays a particular content item (which is any playable thing, e.g. a preset)

GET:
N/A
POST:

```
<ContentItem source="$SOURCE" sourceAccount="$STRING" location="$STRING">
    <itemName>$STRING</itemName>
</ContentItem>
```

6.5 /setZone

Description:

```
Creates a multi-room zone
```

GET:

N/A

POST:

```
<zone master="$MACADDR" senderIPAddress="$IPADDR">
    <member ipaddress="$IPADDR">$MACADDR</member>
    ...
</zone>
```

6.6 /addZoneSlave

Description:

```
Add a slave to a "play everywhere" zone
```

GET:

N/A

POST:

```
<zone master="$MACADDR">
    <member ipaddress="$IPADDR">$MACADDR</member>
    ...
</zone>
```

6.7 /removeZoneSlave

Description:

```
Take a slave out of a "play everywhere" zone
```

GET:

N/A

POST:

```
<zone master="$MACADDR">
    <member ipaddress="$IPADDR">$MACADDR</member>
    ...
</zone>
```

6.8 /now_playing

Description:

```
Gets all info about the currently playing media
```

GET:

```
<nowPlaying deviceID="$MACADDR" source="$SOURCE">
     <ContentItem source="$SOURCE" location="$STRING" sourceAccount="$STRING" isPresetable="$BOOL">
        <itemName>$STRING</itemName>
    </ContentItem>
    <track>$STRING</track>
   <artist>$STRING</artist>
   <album>$STRING</album>
   <stationName>$STRING</stationName>
   <art artImageStatus="$ART_STATUS">$URL</art>
   <playStatus>$PLAY_STATUS</playStatus>
   <description>$STRING</description>
   <stationLocation>$STRING</stationLocation>
</nowPlaying>
```

POST:
N/A

6.9 /trackInfo

Description:

Get track information

GET:

```
<nowPlaying deviceID="$MACADDR" source="$SOURCE">
   <ContentItem source="$SOURCE" location="$STRING" sourceAccount="$STRING" isPresetable="$BOOL">
   <itemName>$STRING</itemName>
   </ContentItem>
   <track>$STRING</track>
   <artist>$STRING</artist>
   <album>$STRING</album>
   <stationName>$STRING</stationName>
   <art artImageStatus="$ART_STATUS">$URL</art>
   <playStatus>$PLAY_STATUS</playStatus>
   <description>$STRING</description>
   <stationLocation>$STRING</stationLocation>
   </nowPlaying>
```

POST:
N/A

6.10 /volume

Description:

Get or Set the volume and mute status for this SoundTouch device

Volume ranges between 0, 100 inclusive

GET:

```
<volume deviceID="$MACADDR">
   <targetvolume>$INT</targetvolume>
   <actualvolume>$INT</actualvolume>
   <muteenabled>$BOOL</muteenabled>
</volume>
```

POST:

```
<volume>$INT</volume>
```

6.11 /sources

Description:

List all available content sources

GET:

```
<sources deviceID="$MACADDR">
   <sourceItem source="$SOURCE" sourceAccount="$STRNG"
       status="$SOURCE_STATUS">$STRING</sourceItem>
   ...
</sources>
```

POST:

N/A

6.12 /presets

Description: Presets are a core part of the SoundTouch ecosystem. A preset is used to set and recall a specific music stream supported by the SoundTouch speaker

List of current Presets

GET:

```
<presets>
   <preset id="$PRESET_ID" createdOn="$UINT64" updateOn="$UINT64">
      <ContentItem source="$SOURCE" location="$STRING" sourceAccount="$STRING"
          isPresetable="$BOOL">
         <itemName>$STRING</itemName>
      </ContentItem>
   </preset>
   ...
</presets>
```

POST:

N/A

6.13 /info

Description:

Get device information; mostly static device info such as device id, type, IP address (per component if applicable), cloud account ID, software version, product version and component type and version

GET:

```
<info deviceID="$MACADDR">
   <name>$STRING</name>
   <type>$STRING</type>
   <margeAccountUUID>$STRING</margeAccountUUID>
   <components>
      <component>
         <componentCategory>$STRING</componentCategory>
         <softwareVersion>$STRING</softwareVersion>
         <serialNumber>$STRING</serialNumber>
      </component>
      ...
   </components>
   <margeURL>$URL</margeURL>
   <networkInfo type="$STRING">
      <macAddress>$MACADDR</macAddress>
      <ipAddress>$IPADDR</ipAddress>
   </networkInfo>
   ...
</info>
```

POST:
N/A

6.14 /name

Description:

Set the device name

GET:
N/A
POST:

```
<name>$STRING</name>
```

6.15 /key

Description: Keys are used as a simple means to interact with the SoundTouch speaker.

Send a remote button press to the device

GET:

N/A

POST:

```
<key state="$KEY_STATE" sender="Gabbo">$KEY_VALUE</key>
```

In general, it is a good practice to send 2 discrete HTTP POST calls, the first using “press” as the key_state, and the second using “release” as the key_state. Doing so simulates both the press and release action of clicking a key. Possible values for “$KEY_STATE” are “press” or “release”

The back to back message bodies will look like the following:

```
<key state="press" sender="Gabbo">$KEY_VALUE</key>
<key state="release" sender="Gabbo">$KEY_VALUE</key>
```

7 WebSockets Notifications

Notifications are server initiated WebSocket messages which inform client(s) of changes in SoundTouch device. They serve to keep clients in sync w/the server. They are sent over HTTP on port 8080.

7.1 PresetsChangedNotifyUI

Description: When a preset is changed in any way like added, cleared, or modified the SoundTouch speaker will send this asynchronous notification. This is a signal for the WS API client to request the new list of presets via the /presets API

```
<updates deviceID="$MACADDR">
   <presetsUpdated>
      <presets>
          <preset id="$INT">
             <ContentItem source="$SOURCE" location="$STRING" sourceAccount=""
                 isPresetable="$BOOL">
               <itemName>$STRING</itemName>
            </ContentItem>
         </preset>
         <preset id="$INT">
            <ContentItem source="$SOURCE" location="$STRING" sourceAccount="$STRING"
                 isPresetable="$BOOL">
               <itemName>STRING</itemName>
            </ContentItem>
         </preset>
         <preset id="$INT">
            <ContentItem source="$SOURCE" location="$STRING" sourceAccount="STRING"
                 isPresetable="$BOOL">
               <itemName>$STRING</itemName>
            </ContentItem>
         </preset>
         <preset id="$INT" createdOn="$UINT64" updatedOn="$UINT64">
            <ContentItem source="$SOURCE" location="$STRING" sourceAccount="" isPresetable="$BOOL">
               <itemName>$STRING</itemName>
            </ContentItem>
         </preset>
      </presets>
   </presetsUpdated>
</updates>
```

7.2 RecentsUpdatedNotifyUI

Description: When the recents list is changed in any way like a recent is added, removed, or moved within the list, the SoundTouch speaker will send this asynchronous notification. This is a signal for the WS API client to request the new list of recents via the /recents API

```
<updates deviceID=’$MACADDR’>
   <recentsUpdated>
      <recents>
         <recent deviceID="$MACADDR" utcTime="$UINT64">
         <contentItem source="$SOURCE" location="$STRING" sourceAccount="$STRING"
              isPresetable="$BOOL">
            <itemName>$STRING</itemName>
         </contentItem>
         </recent>
         <recent deviceID="$MACADDR" utcTime="$UINT64">
         <contentItem source="$SOURCE" location="$STRING" sourceAccount="" isPresetable="$BOOL">
            <itemName>$STRING</itemName>
         </contentItem>
         </recent>
         <recent deviceID="$MACADDR" utcTime="$UINT64">
         <contentItem source="$SOURCE" location="$STRING" sourceAccount="" isPresetable="$BOOL">
            <itemName>$STRING</itemName>
         </contentItem>
         </recent>
      </recents>
   </recentsUpdated>
</updates>
```

7.3 AcctModeChangedNotifyUI

Description: When the SoundTouch speaker's association with a cloud account changes then this asynchronous notification will be sent

```
<updates deviceID=’$MACADDR’>
   <acctModeUpdated>
   </acctModeUpdated>
</updates>"
```

7.4 ErrorNotification

```
ErrorNotification
```

7.5 NowPlayingChange

```
<updates deviceID="$MACADDR">
   <nowPlayingUpdated/>
</updates>
<updates deviceID="$MACADDR">
   <nowPlayingUpdated><nowPlaying deviceID="$MACADDR" source="$SOURCE">
      <ContentItem source="$SOURCE" location="$STRING" sourceAccount="" isPresetable="$BOOL">
         <itemName>$STRING</itemName>
      </ContentItem>
      <track/>
      <artist/>
      <album/>
      <stationName>$STRING</stationName>
      <art artImageStatus="$ART_STATUS">$URL</art>
      <playStatus>$PLAY_STATUS</playStatus>
      <description>$STRING</description>
      <stationLocation>$STRING</stationLocation>
      </nowPlaying>
   </nowPlayingUpdated>
</updates>
```

7.6 VolumeChange

```
<updates deviceID="$MACADDR">
   <volumeUpdated/>
</updates>
```

7.7 BassChange

```
<updates deviceID="$MACADDR">
   <bassUpdated/>
</updates>
```

7.8 ZoneMapChange

```
<updates deviceID="$MACADDR">
   <zoneUpdated/>
</updates>
```

* Slave device joining a zone

```
<updates deviceID="slave $MACADDR">
   <zoneUpdated/>
</updates>
<updates deviceID="slave $MACADDR">
   <volumeUpdated/>
</updates>
<updates deviceID="slave $MACADDR">
   <volumeUpdated/>
</updates>
<updates deviceID="slave $MACADDR">
   <nowPlayingUpdated/>
</updates>
```

* Slave device leaving a zone

```
<updates deviceID="slave $MACADDR">
   <zoneUpdated/>
</updates>
<updates deviceID="slave $MACADDR">
   <nowPlayingUpdated/>
</updates>
```

* Master device notifies any time a slave device joins its zone

```
<updates deviceID="slave $MACADDR">
   <zoneUpdated/>
</updates>
<updates deviceID="slave $MACADDR">
   <nowPlayingUpdated/>
</updates>
```

* Master device notifies any time a slave device leaves its zone

```
<updates deviceID="$MACADDR">
   <zoneUpdated/>
</updates>
<updates deviceID="$MACADDR">
   <zoneUpdated/>
</updates>
```

7.9 SWUpdateStatusChange

```
<updates deviceID="$MACADDR">
   <swUpdateStatusUpdated/>
</updates>
```

7.10 SiteSurveyResultsChange

```
<updates deviceID="$MACADDR">
   <siteSurveyResultsUpdated/>
</updates>
```

7.11 SourcesChange

```
<updates deviceID="$MACADDR">
   <sourcesUpdated/>
</updates>
```

7.12 NowSelectionChange

```
<updates deviceID="$MACADDR">
   <nowSelectionUpdated>
      <preset id="$INT">
         <ContentItem source="$SOURCE" location="$STRING" sourceAccount="$STRING"
              isPresetable="$BOOL">
            <itemName>$STRING</itemName>
         </ContentItem>
      </preset>
   </nowSelectionUpdated>
</updates>
```

7.13 NetworkConnectionStatus

```
<updates deviceID="$MACADDR">
   <connectionStateUpdated/>
</updates>
```

7.14 InfoChange, e.g., the device name changed in Homer

```
<updates deviceID="$MACADDR">
   <infoUpdated/>
</updates>
```