SensFloor® Care API

The SensFloor® Care API offers you an easy way to integrate SensFloor® Care with your own application, as well as with nurse call systems. The API provides events from all rooms equipped with SensFloor®, in a condensed and easy to parse format.

Please note that, although SensFloor® Care provides alarm events and a simplified acknowledgment interface, the system does NOT replace a certified nurse call system.

There are three different kinds of events that can be consumed:

- **Alarm events**: emitted when one or more alarms becomes active or inactive
- **Acknowledge events**: emitted when a past alarm is acknowledged by the care givers
- **State events**: emitted when there is a change in the state of a given room

Furthermore, you can use the API to acknowledge (claim) alarms for a room. This works in the same way as acknowledging an alarm through the SensFloor® Care Station Web Interface.

App Version

SensFloor Station App - **v3.1.0**

How to use

The API is reachable via web sockets, namely via socket.io (see https://socket.io/). Alternatively, you can receive and send event string messages via the serial interface, by connecting a USB-to-Serial adapter to the SensFloor® Care Station Terminal.

Socket.io clients exist out of the box for Node.js and for the browser, in JavaScript. Implementations in other languages are also available (see https://github.com/socketio/socket.io).

To use the API, you need to know the local network's IP of the SensFloor® Care Station Terminal (e.g., 192.168.100.10). Using JavaScript on the browser as an example, you can connect to the API, and print event data to the console, as well as claim alarms, as follows:

```
<script src="http://192.168.100.10:10000/socket.io/socket.io.js"></script>
<script>
        var client = io.connect('http://192.168.100.10:10000');
        client.on('connect', function() {
                console.log('Connected to SensFloor® Care');
        });
        client.on('alarm-new', function(newAlarm) {
                console.log(newAlarm);
        });
        client.on('alarm-ack', function(ackInfo) {
                console.log(ackInfo);
        });
        client.on('state-new', function(newState) {
                console.log(newState);
        });

        // Acknowledge the most recent alarm in room 2
        client.emit(claim-recent-alarm', 2);
        // Acknowledge all alarms in room 13
        client.emit(claim-all-alarms', 13);
</script>
```

Events (consume)

To use data from each event emitted by the SensFloor® Care API, simply register a listener for the event name and retrieve the data.

'alarm-new'

Description

This event is triggered when at least one alarm for a given room becomes active or inactive. Please note that transition alarms (e.g., 'Room Out', 'Toilet In', etc.) are active for a fixed number of seconds.

Data

An object containing information about the alarm

Format

```
alarm: {
        roomNumber,
        roomName,
        alarmNumber,
        type,
        state,
        time
}
```

.roomNumber [number] – room number (From 1 to N rooms for this installation)
.roomName [string] – room name displayed on the SensFloor® Care Terminal
.alarmNumber [number] – alarm number (From 1 to N defined alarms for this room)
.type [string] – alarm type [“fall”, “presence”, “room In” “toilet in”, etc.]
.state [boolean] – alarm state (NOT RELEVANT FOR TRANSITION ALARMS)
.time [string] – when the alarm state changed [“YYYY-MM-DD-HH:MM:SS:SSS”]

Data examples

- *A fall happened in room number 3*

```
{
    roomNumber: 3,
    roomName: 'Room 3',
    alarmNumber: 1,
    type: 'fall',
    state: true,
    time: '2018-08-16-15:50:53:140'
}
```
- *Someone went to the toilet in room number 7*
```
{
    roomNumber: 7,
    roomName: 'Room 7',
    alarmNumber: 3,
    type: 'toilet in',
    state: true, (NOT RELEVANT)
    time: '2018-08-17-23:42:12:935'
}
```

'alarm-ack'

Description

This event is triggered when a care giver acknowledges an alarm through the SensFloor® Care Terminal.
You can use this event to track response times to different rooms and alarms.

Data

An object containing information about the alarm acknowledged.

Format

```
ackInfo: {
        roomNumber,
        roomName,
        alarmNumber,
        type,
        time
}
```

.roomNumber [number] – room number (From 1 to N rooms for this installation)
.roomName [string] – room name displayed on the SensFloor® Care Terminal
.alarmNumber [number] – alarm number (From 1 to N defined alarms for this room)
.type [string] – alarm type [“fall”, “presence”, “room In” “toilet in”, etc.]
.time [string] – when the alarm first became active [“YYYY-MM-DD-HH:MM:SS:SSS”]

Data examples

- *A care giver acknowledges a bed out alarm in room number 12*
```
{
    roomNumber: 12,
    roomName: '1.12',
    alarmNumber: 4,
    type: 'bed out',
    time: '2018-05-24-02:33:01:232'
}
```

'state-new'

Description

This event is triggered when at least one room’s state changes. It provides condensed information about the room terminals each time something changes. Please note that only rooms that change state will generate an event.

Data

An object containing state information for the room

Format

```
state: {
        roomNumber,
        roomName,
        connected,
        enabled,
        tempDisabled,
        sensfloor,
        recalibrating,
        activity,
        alarm
}
```

.roomNumber [number] – room number (From 1 to N rooms for this installation)
.roomName [string] – room name displayed on the SensFloor® Care Terminal
.connected [boolean] – whether the room terminal is connected to the SensFloor® Care Terminal
.enabled [boolean] – whether the room is enabled in the SensFloor® Care Terminal
.tempDisabled [boolean] – whether the room is temporarily disabled (e.g. to wet clean the floor)
.sensfloor [boolean] – whether the room terminal is receiving messages from SensFloor®
.recalibrating [boolean] – whether SensFloor® is currently recalibrating (e.g. after cleaning)
.activity [boolean] – whether there is activity in the room
.alarm [Boolean] – whether there is an active alarm

Data examples

- *There is activity in room number 1*

```
    roomNumber: 1,
    roomName: 'Mr. Smith',
    connected: true,
    enabled: true,
    tempDisabled: false,
    sensfloor: true,
    recalibrating: false,
    activity: true,
    alarm: false
}
```

- *Room number 2 was temporarily disabled to clean*

```
{
    roomNumber: 2,
    roomName: 'Mrs. Jane',
    connected: true,
    enabled: true,
    tempDisabled: true,
    sensfloor: true,
    recalibrating: false,
    activity: false,
    alarm: false,
}
```

Events (emit)

To send data to the SensFloor® Care API, simply emit the desired event and payload.

'claim-recent-alarm'

Description

Emit this event every time you want to acknowledge the most recent alarm for a certain room. This mimics touching on an unattended alarm icon on the SensFloor® Care Station Web Interface.

Data

The API expects the room number to claim. Please note that the room number starts with 1 and increases with the number of rooms. Furthermore, note that here the room number should be used, and NOT the room name.

Format

```
socket.emit('claim-recent-alarm', roomNumber)
```
roomNumber [number] – room number (From 1 to N rooms for this installation)

Data examples

- Claim the most recent unattended alarm in room 2
```
socket.emit('claim-recent-alarm', 2);
```

'claim-all-alarms'

Description

Emit this event every time you want to acknowledge all the alarms for a certain room. This mimics touching on the unattended alarm icon for a certain room on the SensFloor® Care Station Web Interface multiple times, until all the alarms are acknowledged.

Data

The API expects the room number to claim. Please note that the room number starts with 1 and increases with the number of rooms. Furthermore, note that here the room number should be used, and NOT the room name.

Format

```
socket.emit('claim-all-alarms', roomNumber)
```
roomNumber [number] – room number (From 1 to N rooms for this installation)

Data examples

- Claim all unattended alarms in room 13
```
socket.emit('claim-all-alarms', 13);
```

Serial Port

You can consume and send serial string messages via the API by connecting a usb-to-serial device to the SensFloor® Care Terminal.

The serial port on your system must be configured as follows:

- **Baud rate**: 115200
- **Parity**: none
- **Data bits**: 8
- **Stop bits**: 1
- **HW Flow Control**: none

Events (consume)

Each event generates a serial telegram that ends with a new line character [‘\n’] [10] [0x0A]. Fields in the serial telegram are separated by a comma-space sequence [,], where the first field identifies the type of telegram.

‘alarms-new’

Description
This telegram starts with ALARM

Format
ALARM, roomNumber, roomName, alarmNumber, type, state, time*n*

Serial string examples
- A fall happened in room number 3
“ALARM, 3, Room 3, 1, fall, on, 2018-08-16-15:50:53:140\n”

- Someone went to the toilet in room number 7
“ALARM, 7, Room 7, 3, toilet in, on, 2018-08-17-23:42:12:935\n”

‘alarm-ack’

Description
This telegram starts with ACK

Format
ACK, roomNumber, roomName, alarmNumber, type, alarmTime*n*

Serial string examples
- A care giver acknowledges a bed out alarm in room number 12
“ALARM, 12, 1.12, 4, bed out, 2018-05-24-02:33:01:232\n”

‘state-new’

Description
This telegram starts with STATE

Format
STATE, roomNumber, roomName, *connection* connetionState, *enable* enableState, *temp disable* tempDisableState, *floor* floorState, *recal* recalibrationState, *activity* activeState, *alarm* alarmState*n*

Serial string examples

- There is activity in room number 1

"STATE, 1, Mr. Smith, connection on, enable on, temp disable off, floor on, recal off, activity on, alarm off\n"

- Room number 2 was temporarily disabled to clean

"STATE, 2, Mrs. Jane, connection on, enable on, temp disable on, floor on, recal off, activity off, alarm off\n"

Events (emit)

To send events to the API, you need to start the serial telegram with 'SENSFLOOR', and end with a new line character ['\n'] [10] [0x0A]. Similarly, fields in the serial telegram must be separated by a comma-space sequence [,], where the first field identifies the type of telegram.

'claim-recent-alarm'

Description

This telegram must start with "SENSFLOOR, CLAIM RECENT"

Format

SENSFLOOR, CLAIM RECENT, roomNumber*n*

Serial string examples

- Claim the most recent unattended alarm in room 2

"SENSFLOOR, CLAIM RECENT, 2\n"

'claim-all-alarms'

Description

This telegram must start with "SENSFLOOR, CLAIM ALL"

Format

SENSFLOOR, CLAIM ALL, roomNumber*n*

Serial string examples

- Claim all unattended alarms in room 13

"SENSFLOOR, CLAIM ALL, 13\n"