

GesySense Rest Interface Documentation



 **GeSySense**

Gesytec GmbH
Pascalstr. 6
D 52076 Aachen

Tel. +(49) 24 08 / 9 44-0
FAX +(49) 24 08 / 9 44-100
e-mail: info@gesytec.de
www.gesytec.de

Dokument / File	Version	Release	Author
GesySense Rest Interface Documentation	01.00	9.11.2017	Phani Gangula

Preliminary Remarks

This document describes the REST interface of GesySense wireless sensor system.

Revision Status

Version	Section	Description of changes	Date/Sign
01.00	All	English version	09.11.2017/PG

Reference Documents

Abbreviations

Table of contents

1	Introduction	5
2	Security.....	6
2.1	Encryption.....	6
2.2	Authentication	6
3	Sensor Data	7
3.1	URL	7
3.2	JSON Object	7
3.3	Reponse	7
3.4	Number of objects per POST.....	8
3.5	Error Handling	8
4	Life-Sign	9
4.1	URL	9
4.2	JSON.....	9
4.3	Response.....	9
4.4	Status Cycle	10
4.5	Error Handling	10
5	Download	11
5.1	Timing	11
5.2	Error Handling	11
6	Limitation.....	12

1 Introduction

This document describes the REST interface of GesySense wireless sensor system.

The REST based service provides following functions

- Sends sensor data to the server.
- Sends status of the receiver to the server.
- Download files, e.g. Firmware Update.

2 Security

2.1 Encryption

All the received/sent data from the receiver is Transport Layer Security (TLS) encrypted. The protocol version TLS 1.2 is used for this purpose.

The REST interface supports:

- The key for the CA-certificate:
 - **ECDSA-Key**
- Encryption and signature scheme (Cipher-Suites):
 - **ECDHE-ECDSA-AES128-SHA256**

2.2 Authentication

The interface is adequately protected as it uses *Basic Access Authentication in combination with SSL / TLS over https*.

The username and password for the REST client on the receiver are entered over receiver's web interface.

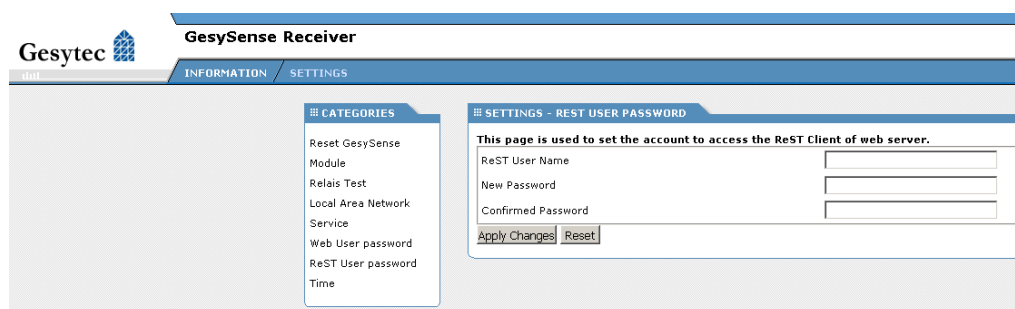


Image 1 Web Interface

3 Sensor Data

3.1 URL

The URL of REST server to which the sensor data is sent is located on the receiver's configuration file *GRC.URL*. The file is stored on receiver's SD-card.

3.2 JSON Object

The receiver sends this JSON-Object to the server.

```
{
  "receiver": "8000000012",
  "data": [
    {
      "sn": "0000015441",
      "value": "22.18",
      "type": "Temperature",
      "rssi": "78",
      "quality": "100",
      "battery": "100",
      "timestamp": "2017-09-26T14:43:04Z"
    },
    {
      "sn": "0000007989",
      "value": "-19.57",
      "type": "Temperature",
      "rssi": "66",
      "quality": "100",
      "battery": "99",
      "timestamp": "2017-09-26T14:43:06Z"
    }
  ]
}
```

The JSON-Object contains the receiver's ID and the current data of the respective sensors as well as serial number, measured value, type, signal strength, signal quality, battery percentage and the current time (ISO 8601) of the sensor.

The data is sent using HTTP *POST* method.

3.3 Reponse

The following response is expected on a successful request:

```
{
  "response": {
    "status": "SUCCESS"
  }
}
```

3.4 Number of objects per POST

The maximum number of **32** items are sent per POST.

3.5 Error Handling

The following response is expected on an unsuccessful request:

```
{
  "response": {
    "status": "ERROR"
  }
}
```

If the connection between the receiver and the server is interrupted, the sensor data is buffered and sent once the connection is restored.

4 Life-Sign

4.1 URL

The URL of REST server to which the status data is sent is located on the receiver's configuration file *UPDATE.URL*. The file is stored on receiver's SD-card.

4.2 JSON

The receiver sends this object cyclically to the server:

```
{
  "GesysenseID": "8.000000000",
  "FWversion": "0.134",
  "BLversion": "1.4",
  "Repeaters": "3",
  "Sensors": "21",
  "SD Card Storage Total": "3942645760",
  "SD Card Storage Free": "1968078848",
  "Bootcnt": "502",
  "timestamp": "2017-10-04T08:47:30Z"
}
```

The server receives the ID, firmware version, bootloader-version and other diagnostic information such as number of known repeaters and sensors in the system, SD card storage information, boot count and current time (**ISO 8601**) of the receiver.

The data is sent using HTTP *POST* method.

4.3 Response

A bootloader update or a firmware update can be carried out using REST interface.

First, the server compares the bootloader version of the client to the current version and returns the following Fwupdate object in JSON notation if the version is not up to date:

```
{
  "fwupdate": {
    "URL": "https://rest1.gesysense.de/update/ttr4_bl_v1_04.bin"
  },
  "response": {
    "status": "SUCCESS"
  }
}
```


Similarly, if the firmware version of the client is not up to date then server sends the following *fwupdate* JSON notation:

```
{
  "fwupdate": {
    "URL": "https://rest1.gesysense.de/update/TTR4_v0_132.bin"
  },
  "response": {
    "status": "SUCCESS"
  }
}
```

The server sends the URL of the file for the firmware update. It is important for the receiver that the filename begins with *TTR4_v*.

4.4 Status Cycle

The receiver sends status information every **15 Minutes**.

4.5 Error Handling

The following response is expected on an unsuccessful request:

```
{
  "response": {
    "status": "ERROR"
  }
}
```

5 Download

The update files (e.g. firmware update) are downloaded to the receiver using HTTP *GET* method.

5.1 Timing

The receiver evaluates the response of the HTTP server. If the receiver receives the URL successfully then it **immediately** try's to download the file and updates itself.

If the update has to be on a specific time then do not place the file on the server until the specified time.

5.2 Error Handling

The download process is stopped after three unsuccessful attempts and resumed again in next cycle (**15 Min**).

6 Limitation

The character length of the URL should be less than 128 characters and the JSON object should be less than 512 characters.