

NodeRed Multiroom Audio with trivum

NodeRed Integration

1. General	1
1.1. Versions	1
1.2. License and Support	1
2. NodeRed package installation	2
3. trivum music server setup	2
4. NodeRed flow creation	2
4.1. trivum control object	2
4.2. trivum status object	2
4.3. trivum event object	3
4.4. technical background informations	4
4.4.1. trivum control	4
4.4.2. trivum status	4
4.4.3. trivum event	5
5. trivum Support	5

trivum technologies GmbH <info@trivum.com> v1.0, 2024-12-13 :title-logo-image: image::../images/trivum-logo.svg[pdfwidth=150,align=right]

1. General

NodeRed is a graphical wiring tool to integrate the Internet Of Things.

1.1. Versions

The current implementation version is 0.5.x

1.2. License and Support

trivum provides its NodeRed package free of charge. Within the scope of our support, we try to eliminate possible errors promptly.



The user has no claim to the correct function or a certain range of functions. trivum reserves the right at all times to extend, modify or remove functions between the versions. As far as it is possible for us to keep the package constant, develop it further and correct any errors. This is done without any obligation. If

you use the NodeRed package from trivum, you agree with this procedure.

2. NodeRed package installation

- search the node palette manager for trivum
- alternatively, you may use
`npm install node-red-contrib-trivum`

3. trivum music server setup

V9: in the Setup, select Automation / Add / NodeRed.

V10: in the Setup, enable Features / NodeRed.

The trivum NodeRed configuration comprises:

- **"server IP address"**
Enter here the IP address of your NodeRed server.
- **action mappings**
Create self-defined trivum event id's to send into the NodeRed system.

4. NodeRed flow creation

4.1. trivum control object

allows to control a trivum music server by messages from within the NodeRed system.

4.2. trivum status object

provides a zone status, either as JSON string, or an object. the most important fields are:

```
rows.system[0].softwareVersion
rows.system[0].softwareRevision
rows.system[0].uptimeInSeconds
rows.system[0].zones
rows.zone[0].groupID
rows.zone[0].basic[0].zoneID
rows.zone[0].basic[0].zoneDescription
rows.zone[0].status[0].volume
rows.zone[0].status[0].status
rows.zone[0].status[0].playingInfo
rows.zone[0].status[0].playingIcon
rows.keypad[0].display[0].line1
rows.keypad[0].display[0].line2
rows.source[0].basic[0].sourceMode
rows.source[0].status[0].artist
```

```
rows.source[0].status[0].album
rows.source[0].status[0].track
rows.source[0].status[0].trackNumber
rows.source[0].status[0].trackCount
```

text is encoded using underscores, so "01_20Codec_20Test" actually means "01 Codec Test".

a function node could extract the current display line 1 in decoded form like this:

```
var text = msg.payload.rows.keypad[0].display[0].line1;
text = ""+text;
var result = "";
var t = text.split("");
var i = 0, val = 0;
for (i=0; i<text.length; i++) {
  if (t[i] == '_' && i<text.length-2) {
    var num = t[i+1] + t[i+2];
    val = parseInt(num,16);
    i += 2;
  } else {
    val = text.charCodeAt(i);
  }
  result += String.fromCharCode(val);
}
msg.payload = result;
return msg;
```

4.3. trivum event object

sends a user defined event from a trivum music server into the NodeRed automation system.

at the trivum music server, go into

```
V9: automation / nodered +
V10: 3rd party / nodered
```

and enter the NodeRed server address where you installed the trivum nodes. then go into

```
V9: automation / nodered / actions / create a trivum event +
V10: 3rd party / nodered / actions / create a trivum event
```

and give the new event a short id like **lighton**. then define an action, for example under

```
V9: Configuration / zones / zone / actions / send lighton
V10: Customize / zones / zone / actions / send lighton
```

which will then appear in control units for the selected zones. pressing that action button will then send event 'lighton' into the nodered system.

4.4. technical background informations

4.4.1. trivum control

This object maps commands like "off" to HTTP API calls like

```
/xml/zone/runCommand.xml?zone=@0&command=1
```

and sends that to the configured trivum music server IP on port 80.

4.4.2. trivum status

This object sends a trivum HTTP API getChanges request like

```
/xml/zone/getChanges.xml?zone=@0&visuid=99&now
```

to the configured trivum music server on port 80, and internally retrieves an xml status reply like

```
<?xml version="1.0" encoding="UTF-8"?>
<rows>
  <info>
    ...
    <currentTimeAsInt>1564143686</currentTimeAsInt>
  </info>
  <system>
    <softwareVersion>946</softwareVersion>
    <softwareRevision>1234</softwareRevision>
    ...
  </system>
  <zone>
    <zoneID>0</zoneID>
    <groupID>255</groupID>
    <basic>
      <zoneID>0</zoneID>
      ...
    </basic>
    <status>
      <volume>15</volume>
      ...
    </status>
  </zone>
  <keypad>
    <keypadID>10</keypadID>
    <display>
```

```
<line1>26_2EJuly_20_20_20_2012_3A21</line1>
<line2>_20_20_20_20_20_20_20_20_20_20_20_20_20_20_20_20_20</line2>
<iconFile>/imgs/visuIconNoMusic_400px.png</iconFile>
</display>
</keypad>
...
</rows>
```

which is converted either to JSON or a JavaScript object. So for example

```
<rows>
  <system>
    <softwareVersion>946</softwareVersion>
```

is made accessible in the JavaScript object as

```
msg.payload.rows.system[0].softwareVersion
```

4.4.3. trivum event

This object listens for incoming UDP messages like

```
{"event":"myeventid","zoneid":"0"}
```

and either passes this on as a string (default), or converts it to a JavaScript object. The event id is accessible in the JavaScript object as `msg.payload.event`

For example, you may add a switch object, and within that you listen on property `msg.payload.event`, and add several `==` cases to check for the event id. If nothing happens check the event object properties and make sure that Output type **object** is selected.

5. trivum Support

For questions, you can use our support area at <http://service.trivum.com>.

If you have a problem / error then you can use our ticket system: [Create a support ticket](#)