It will ONLY be switched, NO-sensor values read!

Available on devices of the V3 version from patch level 1008

Devices before the V2 version-NOT-Supported!

**Exceptions are:**

- ALL3073 from Version 2.00.1002
- ALL3075V2, ALL4075, ALL4076 from Version 2.00.1001

Description without Activated Basic authentication. If this is enable, you must pass the Authentication in URL.

(Basic Authentication: http:// USER: PASSWORD@192.168.0.100/xml/jsonswitch.php)

In this description is called Device IP is the IP address set in the factory state "192.168.0.100".

This must be replaced by the assigned address.

Call, http://192.168.0.100/xml/jsonswitch.php“.

**Parameter**

```
id={id}“ Number or name of the switching sensor / actuator
“set={0/1/toggle}“ Switch actuator off or on
“sensor={reset_count} {reset_today} {reset_absolute}“ Reset or Set a sensor value
“callback={objet}“ (optional) Values are returned as JSONP object
```

<table>
<thead>
<tr>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command (URL)</td>
</tr>
<tr>
<td>Explanation</td>
</tr>
<tr>
<td>Response</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
### set

**Command (URL)**  
http://192.168.0.100/xml/jsonswitch.php?id=2&set=1

**Explanation**  
(optional)  
with this parameter, the on / off switching command is set.  
It returns always the status before switching and the command themselves.

**Response**  
```
{
    "result": {
        "id": "2",
        "status": 0,
        "set": "1"
    }
}
```

### set toggle

**Command (URL)**  
http://192.168.0.100/xml/jsonswitch.php?id=2&set=toggle

**Explanation**  
(optional)  
with this parameter, the toggle command is set  
This command checks always the state of output and toggle the output.  
Output is ON -> Output is switched OFF  
Output is OFF->Output is switched ON  
It returns always the status before switching and the command themselves.

**Response**  
```
{
    "result": {
        "id": "2",
        "status": 0,
        "set": "1"
    }
}
```
sensor: `reset_count`

Command (URL): `http://192.168.0.100/xml?id=123&set=231&sensor=reset_count`

Explanation *(optional)*

Can be set to a counting sensor to set 0 or enforce any value.

Response

**success case:**

```json
{
    "result":{
        "id":"134",
        "sensor":"reset_count",
        "set":"234"
    }
}
```

**case of error:**

```json
{
    "result":{
        "id":"134",
        "sensor":"reset_count",
        "set":"234",
        "error":":given sensor is not a counter!"
    }
}
```
<table>
<thead>
<tr>
<th>sensor</th>
<th>reset_today</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command (URL)</td>
<td><code>http://192.168.0.100/xml?id=123&amp;set=0&amp;sensor=reset_today</code></td>
</tr>
<tr>
<td>Explanation</td>
<td><em>(optional)</em></td>
</tr>
<tr>
<td></td>
<td>Set to set or reset the Recorded MIN / MAX value of today. No value can be set!</td>
</tr>
<tr>
<td>Response</td>
<td>`{</td>
</tr>
<tr>
<td></td>
<td>&quot;result&quot;: {</td>
</tr>
<tr>
<td></td>
<td>&quot;id&quot;: &quot;123&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;sensor&quot;: &quot;reset_today&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;set&quot;: &quot;0&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sensor</th>
<th>reset_absolute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command (URL)</td>
<td><code>http://192.168.0.100/xml?id=123&amp;set=0&amp;sensor=reset_absolute</code></td>
</tr>
<tr>
<td>Explanation</td>
<td><em>(optional)</em></td>
</tr>
<tr>
<td></td>
<td>Set to reset the recorded MIN / MAX value from the beginning of the recording. No value can be set!</td>
</tr>
<tr>
<td>Response</td>
<td>`{</td>
</tr>
<tr>
<td></td>
<td>&quot;result&quot;: {</td>
</tr>
<tr>
<td></td>
<td>&quot;id&quot;: &quot;123&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;sensor&quot;: &quot;reset_absolute&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;set&quot;: &quot;0&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>
### callback

<table>
<thead>
<tr>
<th>Command (URL)</th>
<th><code>http://192.168.0.100/xml?id=2&amp;set=0&amp;callback=demo</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Explanation (optional)</td>
<td>can be set to get back the data as a JSONP object.</td>
</tr>
</tbody>
</table>
| Response            | `demo({
                          "result": {
                          "id": "2",
                          "status": 1,
                          "set": "0"
                      }
                })` |