digitalSTROM Server JSON

digitalSTROM

Version: tag: v1.1-branch*

June 19, 2014

*Revision: b9175b4666c1b95435a7813d996eb6d96cad7ddc

digitalSTROM Alliance
Brandstrasse 33
CH-8952 Schlieren-Zürich
Switzerland

Contents

1 Introduction **7**

2 Apartment **8**
2.1 Name . . . 8
2.1.1 getName . . . 8
2.1.2 setName . . . 8
2.2 Scene . . . 9
2.2.1 callScene . . . 9
2.2.2 saveScene . . . 9
2.2.3 undoScene . . . 10
2.2.4 getLockedScenes . . . 11
2.3 Value . . . 11
2.3.1 Set Device Output Value . . . 11
2.4 Groups . . . 12
2.4.1 getReachableGroups . . . 12
2.5 Structure . . . 13
2.5.1 getStructure . . . 13
2.5.2 getDevices . . . 20
2.5.3 getCircuits . . . 23
2.5.4 removeMeter . . . 23

3 Zone **25**
3.1 Common . . . 25
3.2 Name . . . 25
3.2.1 getName . . . 25
3.2.2 setName . . . 26
3.3 Scene . . . 26
3.3.1 callScene . . . 26
3.3.2 saveScene . . . 27
3.3.3 undoScene . . . 28
3.3.4 sceneGetName . . . 28
3.3.5 sceneSetName . . . 29
3.3.6 getReachableScenes . . . 30
3.3.7 getLastCalledScene . . . 31
3.4 Value . . . 31
3.4.1 Set Output Value . . . 31
3.4.2 Blink . . . 32

4 Device **34**
4.1 Common . . . 34
4.2 Name . . . 34
4.2.1 getName . . . 34
4.2.2 setName . . . 35
4.2.3 getSpec . . . 35
4.3 Groups . . . 36
4.3.1 getGroups . . . 36
4.4 Scene . . . 37
4.4.1 callScene . . . 37
4.4.2 saveScene . . . 37
4.4.3 undoScene . . . 38
4.4.4 turnOn . . . 38
4.4.5 turnOff . . . 39
4.4.6 increaseValue . . . 39
4.4.7 decreaseValue . . . 40
4.5 Value . . . 40
4.5.1 Set Value . . . 40
4.5.2 Set Output Value . . . 41
4.5.3 Get Output Value . . . 42
4.5.4 Get Scene Value . . . 43
4.5.5 Set Scene Value . . . 43
4.5.6 Blink . . . 44
4.6 Configuration . . . 44
4.6.1 setButtonID . . . 44
4.6.2 setButtonInputMode . . . 45
4.6.3 setOutputMode . . . 45
4.6.4 setJokerGroup . . . 46
4.6.5 setButtonActiveGroup . . . 46
4.6.6 getSceneMode . . . 47
4.6.7 setSceneMode . . . 48
4.6.8 getTransitionTime . . . 49
4.6.9 setTransitionTime . . . 49
4.6.10 setConfig . . . 50
4.6.11 getConfig . . . 51
4.6.12 getConfigWord . . . 52
4.7 Sensor . . . 53
4.7.1 Get Sensor Value . . . 53
4.7.2 Get Sensor Type . . . 53
4.7.3 getSensorEventTableEntry . . . 54
4.7.4 setSensorEventTableEntry . . . 55
4.8 Programming . . . 55
4.8.1 Set Programming Mode . . . 55
4.8.2 Add To Area . . . 56

4.8.3 Remove From Area . . . 56
4.8.4 Get Transmission Quality . . . 57

5 Circuit **58**
5.1 Common . . . 58
5.2 Name . . . 58
5.2.1 getName . . . 58
5.2.2 setName . . . 59
5.3 Energy Meter . . . 59
5.3.1 getConsumption . . . 59
5.3.2 getEnergyMeterValue . . . 60

6 Structure **61**
6.1 Zone . . . 61
6.1.1 addZone . . . 61
6.1.2 removeZone . . . 61
6.2 Group . . . 62
6.2.1 addGroup . . . 62
6.2.2 removeGroup . . . 63
6.2.3 groupAddDevice . . . 63
6.2.4 groupSetName . . . 64
6.2.5 groupSetColor . . . 64
6.3 Device . . . 65
6.3.1 zoneAddDevice . . . 65
6.3.2 removeDevice . . . 66

7 Event **67**
7.1 Raise . . . 67
7.1.1 raise . . . 67
7.2 Subscription . . . 67
7.2.1 subscribe . . . 67
7.2.2 unsubscribe . . . 68
7.2.3 get . . . 69

8 Metering **71**
8.1 Metering . . . 71
8.1.1 getResolution . . . 71
8.1.2 getSeries . . . 71
8.1.3 getValues . . . 72
8.1.4 getLatest . . . 74

9 System **76**
9.1 System Information 76
9.1.1 version 76
9.1.2 time 76
9.1.3 getDSID 77
9.2 Authentication 77
9.2.1 login 77
9.2.2 logout 78
9.2.3 loggedInUser 78
9.2.4 setPassword 79
9.2.5 requestApplicationToken 79
9.2.6 enableToken 80
9.2.7 revokeToken 80
9.2.8 loginApplication 81

10 Property Tree **82**
10.1 Basic Property Tree Operations 82
10.1.1 getString 82
10.1.2 setString 82
10.1.3 getInteger 83
10.1.4 setInteger 83
10.1.5 getBoolean 84
10.1.6 setBoolean 84
10.1.7 getChildren 85
10.1.8 getType 86
10.1.9 getFlags 86
10.1.10 setFlag 87
10.1.11 remove 87
10.2 Property Query 88
10.2.1 query 88

1 Introduction

All requests are sent using HTTP GET and parameters added to the query string url like:

```
/json/apartment/setName?name="My digitalStrom Server"&username=dssadmin&password=secret
```

If not properly authenticated the HTTP Status 403 is returned and the error response contains:

```
{
   "ok":false,
   "message":"not logged in"
}
```

If an unknown method is requested the error message "Unhandled Function" is returned:

```
{
    "ok": false,
    "message": "Unhandled function"
}
```

If a request has been successfully processed the JSON answer contains an "ok" and an optional "result" field. The result array is explained in the particular sections.

ok	true
result	array of result values

Where Group Names are allowed the following table lists the possible names.

Name	Group Id	Description
yellow	1	Light
gray	2	Light
blue	3	Climate

2 Apartment

2.1 Name

2.1.1 getName

Returns the user defined name of the installation.

Synopsis
HTTP GET /json/apartment/getName

Parameter
None

Response
HTTP Status 200

name	identifier string for the installation

Sample

```
GET /json/apartment/getName
{
    "ok":true,
    "result" :
    {
        "name" : "digitalStrom Installation Hans Mustermann"
    }
}
```

2.1.2 setName

Sets the installation name.

Synopsis
HTTP GET /json/apartment/setName

Parameter	Description	Remarks
newName	identifier string for the installation	Mandatory

Parameter

Response
HTTP Status 200

ok	true

Sample

```
GET /json/apartment/setName?newName="My dSS"
{
    "ok":true
}
```

2.2 Scene

2.2.1 callScene

Excutes the scene *sceneNumber* on a group of devices.

Synopsis
HTTP GET /json/apartment/callScene

Parameter

Parameter	Description	Remarks
sceneNumber	Numerical value	Mandatory
groupID	Number of the target group	Optional
groupName	Name of the target group	Optional
force	Boolean value, if set a forced scene call is issued	Optional

If the group parameters are omitted the command is sent as broadcast to all zones and all devices.

Response
HTTP Status 200

ok	true

Sample

```
GET /json/apartment/callScene?sceneNumber=65
{
    "ok":true
}
```

2.2.2 saveScene

Tells devices to store their current output values as a default for the scene *sceneNumber*.

Synopsis
HTTP GET /json/apartment/saveScene

Parameter

Parameter	Description	Remarks
sceneNumber	Numerical value	Mandatory
groupID	Number of the target group	Optional
groupName	Name of the target group	Optional

If the group parameters are omitted the command is sent as broadcast to all zones and all devices.

Response
HTTP Status 200

ok	true

Sample

```
GET /json/apartment/saveScene?sceneNumber=65
{
    "ok":true
}
```

2.2.3 undoScene

Tells devices to restore their output values to the previous state if the current scene matches the *sceneNumber*.

Synopsis
HTTP GET /json/apartment/undoScene

Parameter

Parameter	Description	Remarks
sceneNumber	Numerical value	Mandatory
groupID	Number of the target group	Optional
groupName	Name of the target group	Optional

If the group parameters are omitted the command is sent as broadcast to all zones and all devices.

Response
HTTP Status 200

ok	true

Sample

```
GET /json/apartment/undoScene?sceneNumber=65
{
    "ok":true
}
```

2.2.4 getLockedScenes

Retrieves scene numbers of scenes that are currently locked because of an update of device scene tables.

Synopsis
HTTP GET /json/apartment/getLockedScenes

Parameter None

Response
HTTP Status 200

result.lockedScenes[]	array of scene numbers that are currently locked

Sample

```
GET /json/apartment/getLockedScenes
{
    "ok" : true,
    "result" :
    {
        "lockedScenes" : []
    }
}
```

2.3 Value

2.3.1 Set Device Output Value

Set the output value of a group of devices to a given value.

> **Notice** Setting output values directly bypasses the group state machine and is unrecommended.

Synopsis
HTTP GET /json/apartment/setValue

Parameter

Parameter	Description	Remarks
value	Numerical value	Mandatory
groupID	Number of the target group	Optional
groupName	Name of the target group	Optional

If the group parameters are omitted the command is sent as broadcast to all devices.

> **Notice** Setting output values without a group identification is strongly unrecommended.

Response
HTTP Status 200

ok	true

Sample

```
GET /json/apartment/setValue?value=0&groupID=2
{
    "ok":true,
}
```

2.4 Groups

2.4.1 getReachableGroups

Returns a list of groups for which are actuators actually present in the installation.

Synopsis
HTTP GET /json/apartment/getReachableGroups

Parameter
None

Response
HTTP Status 200

result.zones	array of zones in the installation
result.zones[].groups	array of groups in a zone

Sample

```
GET /json/apartment/getReachableGroups
{

    "ok": true,
    "result": {
        "zones": [
            {
                "zoneID": 1223,
                "name": "Wohnen",
                "groups": [
                    1,
                    2,
                    7
                ]
            },
            {
                "zoneID": 1241,
                "name": "Schlafen",
                "groups": [
                    1,
                    5,
                    7
                ]
            },
            {
                "zoneID": 1237,
                "name": "Essen",
                "groups": [
                    1,
                    6
                ]
            }
        ]
    }
}
```

2.5 Structure

2.5.1 getStructure

Returns an object containing the structure of the apartment. This includes detailed information about all zones, groups and devices.

Synopsis
HTTP GET /json/apartment/getStructure

Parameter
None

Response
HTTP Status 200

result.apartment.zones	array of zone information
result.apartment.zones[].devices	array of device information in each zone
result.apartment.zones[].devices[].groups	group membership of each device
result.apartment.zones[].groups	array of group information in each zone
result.apartment.zones[].groups[].devices	array of devices per group in a zone

Sample

```
GET /json/apartment/getStructure
{
    "ok": true,
    "result": {
        "apartment": {
            "zones": [
                {
                    "id": 0,
                    "name": "",
                    "isPresent": false,
                    "devices": [
                        {
                            "id": "3504175fe0000000000182f6",
                            "name": "Regalleuchte",
                            "functionID": 4152,
                            "productRevision": 49955,
                            "productID": 6344,
                            "hwInfo": "GE-SDS200",
                            "meterDSID": "3504175fe0000010000003dd",
                            "busID": 97,
                            "zoneID": 989,
                            "isPresent": false,
                            "lastDiscovered": "2012-10-24 11:17:29",
                            "firstSeen": "2012-10-22 16:22:02",
                            "inactiveSince": "2012-10-22 16:22:02",
                            "outputMode": 22,
                            "buttonID": 0,
                            "buttonActiveGroup": 1,
                            "buttonInputMode": 0,
                            "buttonInputIndex": 0,
                            "buttonInputCount": 1,
                            "groups": [
                                "1"
                            ]
                        },
                        {
                            "id": "3504175fe00000000000439c",
                            "name": "Stehlampe",
                            "functionID": 4152,
                            "productRevision": 789,
                            "productID": 200,
                            "hwInfo": "GE-KM200",
                            "meterDSID": "3504175fe0000010000003dd",
                            "busID": 153,
                            "zoneID": 989,
                            "isPresent": false,
                            "lastDiscovered": "2012-10-24 11:17:29",
                            "firstSeen": "2012-10-22 16:22:02",
                            "inactiveSince": "2012-10-22 16:22:02",
                            "outputMode": 22,
                            "buttonID": 0,
                            "buttonActiveGroup": 1,
                            "buttonInputMode": 0,
                            "buttonInputIndex": 0,
                            "buttonInputCount": 1,
                            "groups": [
                                "1"
                            ]
                        },
                        {
                            "id": "3504175fe0000000000151fd",
                            "name": "Fernseher",
                            "functionID": 33041,
                            "productRevision": 41761,
                            "productID": 5320,
                            "hwInfo": "SW-ZWS200",
                            "meterDSID": "3504175fe0000010000003dd",
```

```
    "busID": 693,
    "zoneID": 989,
    "isPresent": false,
    "lastDiscovered": "2012-10-24 11:17:29",
    "firstSeen": "2012-10-22 16:22:02",
    "inactiveSince": "2012-10-22 16:22:02",
    "outputMode": 39,
    "buttonID": 0,
    "buttonActiveGroup": 5,
    "buttonInputMode": 0,
    "buttonInputIndex": 0,
    "buttonInputCount": 1,
    "groups": [
        "5",
        "8"
    ]
},
{
    "id": "3504175fe000000000001234",
    "name": "Wandlampe",
    "functionID": 4144,
    "productRevision": 789,
    "productID": 1234,
    "hwInfo": "GE-TKM210",
    "meterDSID": "3504175fe0000010000003dd",
    "busID": 782,
    "zoneID": 989,
    "isPresent": false,
    "lastDiscovered": "2012-10-24 11:17:29",
    "firstSeen": "2012-10-22 16:22:02",
    "inactiveSince": "2012-10-22 16:22:02",
    "outputMode": 22,
    "buttonID": 4,
    "buttonActiveGroup": 1,
    "buttonInputMode": 0,
    "buttonInputIndex": 0,
    "buttonInputCount": 1,
    "groups": [
        "1"
    ]
},
{
    "id": "3504175fe00000000043a7",
    "name": "Deckenlicht",
    "functionID": 4152,
    "productRevision": 789,
    "productID": 200,
    "hwInfo": "GE-KM200",
    "meterDSID": "3504175fe0000010000003dd",
    "busID": 784,
    "zoneID": 1038,
    "isPresent": true,
    "lastDiscovered": "2012-10-26 15:36:30",
    "firstSeen": "2012-10-22 16:22:02",
    "inactiveSince": "1970-01-01 01:00:00",
    "outputMode": 22,
    "buttonID": 5,
    "buttonActiveGroup": 1,
    "buttonInputMode": 0,
    "buttonInputIndex": 0,
    "buttonInputCount": 1,
    "groups": [
        "1"
    ]
},
{
    "id": "3504175fe00000000042dc",
    "name": "Paniktaster",
    "functionID": 24896,
    "productRevision": 790,
    "productID": 1225,
    "hwInfo": "RT-TKM201",
    "meterDSID": "3504175fe0000010000003dd",
    "busID": 785,
    "zoneID": 989,
    "isPresent": false,
    "lastDiscovered": "2012-10-24 11:17:29",
    "firstSeen": "2012-10-23 16:23:38",
    "inactiveSince": "2012-10-24 11:01:40",
    "outputMode": 0,
    "buttonID": 17,
    "buttonActiveGroup": 154,
    "buttonInputMode": 20,
    "buttonInputIndex": 0,
```

```
            "buttonInputCount": 0,
            "groups": [
                "6"
            ]
        }
    ],
    "groups": [
        {
            "id": 0,
            "name": "broadcast",
            "isPresent": false,
            "devices": [
                "3504175fe00000000000182f6",
                "3504175fe00000000000439c",
                "3504175fe00000000000151fd",
                "3504175fe00000000000001234",
                "3504175fe00000000000043a7",
                "3504175fe00000000000042dc"
            ]
        },
        {
            "id": 1,
            "name": "yellow",
            "isPresent": true,
            "devices": [
                "3504175fe00000000000182f6",
                "3504175fe00000000000439c",
                "3504175fe00000000000001234",
                "3504175fe00000000000043a7"
            ]
        },
        {
            "id": 2,
            "name": "gray",
            "isPresent": true,
            "devices": [ ]
        },
        {
            "id": 3,
            "name": "blue",
            "isPresent": true,
            "devices": [ ]
        },
        {
            "id": 4,
            "name": "cyan",
            "isPresent": true,
            "devices": [ ]
        },
        {
            "id": 5,
            "name": "magenta",
            "isPresent": true,
            "devices": [
                "3504175fe00000000000151fd"
            ]
        },
        {
            "id": 6,
            "name": "red",
            "isPresent": true,
            "devices": [
                "3504175fe00000000000042dc"
            ]
        },
        {
            "id": 7,
            "name": "green",
            "isPresent": true,
            "devices": [ ]
        },
        {
            "id": 8,
            "name": "black",
            "isPresent": true,
            "devices": [
                "3504175fe00000000000151fd"
            ]
        },
        {
            "id": 9,
            "name": "white",
            "isPresent": true,
            "devices": [ ]
```

```
        },
        {
            "id": 10,
            "name": "display",
            "isPresent": false,
            "devices": [ ]
        }
    ]
},
{
    "id": 989,
    "name": "Wohnen",
    "isPresent": true,
    "devices": [
        {
            "id": "3504175fe0000000000182f6",
            "name": "Regalleuchte",
            "functionID": 4152,
            "productRevision": 49955,
            "productID": 6344,
            "hwInfo": "GE_SDS200",
            "meterDSID": "3504175fe0000010000003dd",
            "busID": 97,
            "zoneID": 989,
            "isPresent": false,
            "lastDiscovered": "2012-10-24 11:17:29",
            "firstSeen": "2012-10-22 16:22:02",
            "inactiveSince": "2012-10-22 16:22:02",
            "outputMode": 22,
            "buttonID": 0,
            "buttonActiveGroup": 1,
            "buttonInputMode": 0,
            "buttonInputIndex": 0,
            "buttonInputCount": 1,
            "groups": [
                "1"
            ]
        },
        {
            "id": "3504175fe00000000000439c",
            "name": "Stehlampe",
            "functionID": 4152,
            "productRevision": 789,
            "productID": 200,
            "hwInfo": "GE_KM200",
            "meterDSID": "3504175fe0000010000003dd",
            "busID": 153,
            "zoneID": 989,
            "isPresent": false,
            "lastDiscovered": "2012-10-24 11:17:29",
            "firstSeen": "2012-10-22 16:22:02",
            "inactiveSince": "2012-10-22 16:22:02",
            "outputMode": 22,
            "buttonID": 0,
            "buttonActiveGroup": 1,
            "buttonInputMode": 0,
            "buttonInputIndex": 0,
            "buttonInputCount": 1,
            "groups": [
                "1"
            ]
        },
        {
            "id": "3504175fe0000000000151fd",
            "name": "Fernseher",
            "functionID": 33041,
            "productRevision": 41761,
            "productID": 5320,
            "hwInfo": "SW_ZWS200",
            "meterDSID": "3504175fe0000010000003dd",
            "busID": 693,
            "zoneID": 989,
            "isPresent": false,
            "lastDiscovered": "2012-10-24 11:17:29",
            "firstSeen": "2012-10-22 16:22:02",
            "inactiveSince": "2012-10-22 16:22:02",
            "outputMode": 39,
            "buttonID": 0,
            "buttonActiveGroup": 5,
            "buttonInputMode": 0,
            "buttonInputIndex": 0,
            "buttonInputCount": 1,
            "groups": [
                "5",
```

```
                "8"
            ]
        },
        {
            "id": "3504175fe000000000001234",
            "name": "Wandlampe",
            "functionID": 4144,
            "productRevision": 789,
            "productID": 1234,
            "hwInfo": "GE-TKM210",
            "meterDSID": "3504175fe0000010000003dd",
            "busID": 782,
            "zoneID": 989,
            "isPresent": false,
            "lastDiscovered": "2012-10-24 11:17:29",
            "firstSeen": "2012-10-22 16:22:02",
            "inactiveSince": "2012-10-22 16:22:02",
            "outputMode": 22,
            "buttonID": 4,
            "buttonActiveGroup": 1,
            "buttonInputMode": 0,
            "buttonInputIndex": 0,
            "buttonInputCount": 1,
            "groups": [
                "1"
            ]
        },
        {
            "id": "3504175fe00000000000042dc",
            "name": "Paniktaster",
            "functionID": 24896,
            "productRevision": 790,
            "productID": 1225,
            "hwInfo": "RT-TKM201",
            "meterDSID": "3504175fe0000010000003dd",
            "busID": 785,
            "zoneID": 989,
            "isPresent": false,
            "lastDiscovered": "2012-10-24 11:17:29",
            "firstSeen": "2012-10-23 16:23:38",
            "inactiveSince": "2012-10-24 11:01:40",
            "outputMode": 0,
            "buttonID": 17,
            "buttonActiveGroup": 154,
            "buttonInputMode": 20,
            "buttonInputIndex": 0,
            "buttonInputCount": 0,
            "groups": [
                "6"
            ]
        }
    ],
    "groups": [
        {
            "id": 0,
            "name": "broadcast",
            "isPresent": false,
            "devices": [
                "3504175fe00000000000182f6",
                "3504175fe000000000000439c",
                "3504175fe00000000000151fd",
                "3504175fe000000000001234",
                "3504175fe00000000000042dc"
            ]
        },
        {
            "id": 1,
            "name": "yellow",
            "isPresent": true,
            "devices": [
                "3504175fe00000000000182f6",
                "3504175fe000000000000439c",
                "3504175fe000000000001234"
            ]
        },
        {
            "id": 2,
            "name": "gray",
            "isPresent": true,
            "devices": [ ]
        },
        {
            "id": 3,
            "name": "blue",
```

```
                "isPresent": true,
                "devices": [ ]
            },
            {
                "id": 4,
                "name": "cyan",
                "isPresent": true,
                "devices": [ ]
            },
            {
                "id": 5,
                "name": "magenta",
                "isPresent": true,
                "devices": [
                    "3504175fe0000000000151fd"
                ]
            },
            {
                "id": 6,
                "name": "red",
                "isPresent": true,
                "devices": [
                    "3504175fe00000000000042dc"
                ]
            },
            {
                "id": 7,
                "name": "green",
                "isPresent": true,
                "devices": [ ]
            },
            {
                "id": 8,
                "name": "black",
                "isPresent": true,
                "devices": [
                    "3504175fe0000000000151fd"
                ]
            },
            {
                "id": 9,
                "name": "white",
                "isPresent": true,
                "devices": [ ]
            },
            {
                "id": 10,
                "name": "display",
                "isPresent": false,
                "devices": [ ]
            }
        ]
    },
    {
        "id": 1038,
        "name": "Schlafen",
        "isPresent": true,
        "devices": [
            {
                "id": "3504175fe0000000000043a7",
                "name": "Deckenlicht",
                "functionID": 4152,
                "productRevision": 789,
                "productID": 200,
                "hwInfo": "GE-KM200",
                "meterDSID": "3504175fe0000010000003dd",
                "busID": 784,
                "zoneID": 1038,
                "isPresent": true,
                "lastDiscovered": "2012-10-26 15:36:30",
                "firstSeen": "2012-10-22 16:22:02",
                "inactiveSince": "1970-01-01 01:00:00",
                "outputMode": 22,
                "buttonID": 5,
                "buttonActiveGroup": 1,
                "buttonInputMode": 0,
                "buttonInputIndex": 0,
                "buttonInputCount": 1,
                "groups": [
                    "1"
                ]
            }
        ],
        "groups": [
```

```
                    {
                        "id": 0,
                        "name": "broadcast",
                        "isPresent": true,
                        "devices": [
                            "3504175fe0000000000043a7"
                        ]
                    },
                    {
                        "id": 1,
                        "name": "yellow",
                        "isPresent": true,
                        "devices": [
                            "3504175fe0000000000043a7"
                        ]
                    },
                    {
                        "id": 2,
                        "name": "gray",
                        "isPresent": true,
                        "devices": [ ]
                    },
                    {
                        "id": 3,
                        "name": "blue",
                        "isPresent": true,
                        "devices": [ ]
                    },
                    {
                        "id": 4,
                        "name": "cyan",
                        "isPresent": true,
                        "devices": [ ]
                    },
                    {
                        "id": 5,
                        "name": "magenta",
                        "isPresent": true,
                        "devices": [ ]
                    },
                    {
                        "id": 6,
                        "name": "red",
                        "isPresent": true,
                        "devices": [ ]
                    },
                    {
                        "id": 7,
                        "name": "green",
                        "isPresent": true,
                        "devices": [ ]
                    },
                    {
                        "id": 8,
                        "name": "black",
                        "isPresent": true,
                        "devices": [ ]
                    },
                    {
                        "id": 9,
                        "name": "white",
                        "isPresent": true,
                        "devices": [ ]
                    },
                    {
                        "id": 10,
                        "name": "display",
                        "isPresent": false,
                        "devices": [ ]
                    }
                ]
            }
        ]
    }
}
```


2.5.2 getDevices

Returns an array containing all devices of the apartment.

Synopsis
HTTP GET /json/apartment/getDevices

Parameter
None

Response
HTTP Status 200

result	array of devices

Sample


```
GET /json/apartment/getDevices
{

    "ok": true,
    "result": [
        {
            "id": "3504175fe0000000000182f6",
            "name": "Regalleuchte",
            "functionID": 4152,
            "productRevision": 49955,
            "productID": 6344,
            "hwInfo": "GE-SDS200",
            "meterDSID": "3504175fe0000010000003dd",
            "busID": 97,
            "zoneID": 989,
            "isPresent": false,
            "lastDiscovered": "2012-10-24 11:17:29",
            "firstSeen": "2012-10-22 16:22:02",
            "inactiveSince": "2012-10-22 16:22:02",
            "outputMode": 22,
            "buttonID": 0,
            "buttonActiveGroup": 1,
            "buttonInputMode": 0,
            "buttonInputIndex": 0,
            "buttonInputCount": 1,
            "groups": [
                "1"
            ]
        },
        {
            "id": "3504175fe00000000000439c",
            "name": "Stehlampe",
            "functionID": 4152,
            "productRevision": 789,
            "productID": 200,
            "hwInfo": "GE-KM200",
            "meterDSID": "3504175fe0000010000003dd",
            "busID": 153,
            "zoneID": 989,
            "isPresent": false,
            "lastDiscovered": "2012-10-24 11:17:29",
            "firstSeen": "2012-10-22 16:22:02",
            "inactiveSince": "2012-10-22 16:22:02",
            "outputMode": 22,
            "buttonID": 0,
            "buttonActiveGroup": 1,
            "buttonInputMode": 0,
            "buttonInputIndex": 0,
            "buttonInputCount": 1,
            "groups": [
                "1"
            ]
        },
        {
            "id": "3504175fe0000000000151fd",
            "name": "Fernseher",
            "functionID": 33041,
            "productRevision": 41761,
            "productID": 5320,
```

```
    "hwInfo": "SW-ZWS200",
    "meterDSID": "3504175fe0000010000003dd",
    "busID": 693,
    "zoneID": 989,
    "isPresent": false,
    "lastDiscovered": "2012-10-24 11:17:29",
    "firstSeen": "2012-10-22 16:22:02",
    "inactiveSince": "2012-10-22 16:22:02",
    "outputMode": 39,
    "buttonID": 0,
    "buttonActiveGroup": 5,
    "buttonInputMode": 0,
    "buttonInputIndex": 0,
    "buttonInputCount": 1,
    "groups": [
        "5",
        "8"
    ]
},
{
    "id": "3504175fe000000000001234",
    "name": "Wandlampe",
    "functionID": 4144,
    "productRevision": 789,
    "productID": 1234,
    "hwInfo": "GE-TKM210",
    "meterDSID": "3504175fe0000010000003dd",
    "busID": 782,
    "zoneID": 989,
    "isPresent": false,
    "lastDiscovered": "2012-10-24 11:17:29",
    "firstSeen": "2012-10-22 16:22:02",
    "inactiveSince": "2012-10-22 16:22:02",
    "outputMode": 22,
    "buttonID": 4,
    "buttonActiveGroup": 1,
    "buttonInputMode": 0,
    "buttonInputIndex": 0,
    "buttonInputCount": 1,
    "groups": [
        "1"
    ]
},
{
    "id": "3504175fe0000000000043a7",
    "name": "Deckenlicht",
    "functionID": 4152,
    "productRevision": 789,
    "productID": 200,
    "hwInfo": "GE-KM200",
    "meterDSID": "3504175fe0000010000003dd",
    "busID": 784,
    "zoneID": 1038,
    "isPresent": true,
    "lastDiscovered": "2012-10-26 15:36:30",
    "firstSeen": "2012-10-22 16:22:02",
    "inactiveSince": "1970-01-01 01:00:00",
    "outputMode": 22,
    "buttonID": 5,
    "buttonActiveGroup": 1,
    "buttonInputMode": 0,
    "buttonInputIndex": 0,
    "buttonInputCount": 1,
    "groups": [
        "1"
    ]
},
{
    "id": "3504175fe0000000000042dc",
    "name": "Paniktaster",
    "functionID": 24896,
    "productRevision": 790,
    "productID": 1225,
    "hwInfo": "RT-TKM201",
    "meterDSID": "3504175fe0000010000003dd",
    "busID": 785,
    "zoneID": 989,
    "isPresent": false,
    "lastDiscovered": "2012-10-24 11:17:29",
    "firstSeen": "2012-10-23 16:23:38",
    "inactiveSince": "2012-10-24 11:01:40",
    "outputMode": 0,
    "buttonID": 17,
    "buttonActiveGroup": 154,
```

```
            "buttonInputMode": 20,
            "buttonInputIndex": 0,
            "buttonInputCount": 0,
            "groups": [
                "6"
            ]
        }
    ]
}
```


2.5.3 getCircuits

Returns an array containing all digitalSTROM Meters of the apartment.

Synopsis
HTTP GET /json/apartment/getCircuits

Parameter
None

Response
HTTP Status 200

result.circuits	array of digitalSTROM Meters

Sample


```
GET /json/apartment/getCircuits
{

    "ok": true,
    "result": {
        "circuits": [
            {
                "name": "dSM03DD—#1",
                "dsid": "3504175fe0000010000003dd",
                "hwVersion": 721409,
                "armSwVersion": 17498112,
                "dspSwVersion": 16908800,
                "apiVersion": 517,
                "hwName": "",
                "isPresent": true,
                "isValid": true
            },
            {
                "name": "dSM040E—#2",
                "dsid": "3504175fe0000010000040e",
                "hwVersion": 721409,
                "armSwVersion": 17498112,
                "dspSwVersion": 16908800,
                "apiVersion": 517,
                "hwName": "",
                "isPresent": true,
                "isValid": true
            }
        ]
    }

}
```


2.5.4 removeMeter

Removes an inactive digitalSTROM Meter object from the installation.

Synopsis
HTTP GET /json/apartment/removeMeter

Parameter

Parameter	Description	Remarks
dsid	dSID of the digitalSTROM Meter	Mandatory

Response
HTTP Status 200

result	array of digitalSTROM Meters

Sample

```
GET /json/apartment/removeMeter?dsid=3504175fe00000100000040e
{
    "ok" : true
}
```

3 Zone

3.1 Common

Every /json/zone/ function uses a common selection scheme for the zone to which the command refers to. Either the parameter "id" or "name" must be given to identify the zone. The special value zero for the "id" maybe used to send the command as broadcast to all zones.

Parameter	Description	Remarks
id	Zone Number	Optional
name	Zone Name	Optional

A missing zone identifier result in the following error message to be returned.

```
{
   "ok": false,
   "message": "Need parameter name or id to identify zone"

}
```

If a zone identifier does not match any actually known zone in the installation the following error message is returned.

```
{
    "ok": false,
    "message": "Could not find zone with id '1250'"
}
```

3.2 Name

3.2.1 getName

Returns the user defined name of the zone.

Synopsis
HTTP GET /json/zone/getName

Parameter
None

Response
HTTP Status 200

name	identifier string for the zone

Sample

```
GET /json/zone/getName?id=1237
{
    "ok":true,
    "result" :
    {
        "name" : "Wohnen"
    }
}
```

3.2.2 setName

Sets the zone name.

Synopsis
HTTP GET /json/zone/setName

Parameter	Description	Remarks
newName	identifier string for the zone	Mandatory

Parameter

Response
HTTP Status 200

ok	true

Sample

```
GET /json/zone/setName?id=1237&newName="Wohnen"
{
    "ok":true
}
```

3.3 Scene

3.3.1 callScene

Excutes the scene *sceneNumber* in a zone for a group of devices.

Synopsis
HTTP GET /json/zone/callScene

Parameter

Parameter	Description	Remarks
sceneNumber	Numerical value	Mandatory
groupID	Number of the target group	Optional
groupName	Name of the target group	Optional
force	Boolean value, if set a forced scene call is issued	Optional

If the group parameters are omitted the command is sent as broadcast to all devices in a zone.

Response
HTTP Status 200

ok	true

Sample

```
GET /json/zone/callScene?id=1237&groupNumber=1&sceneNumber=5&force=true
{
    "ok":true
}
```

3.3.2 saveScene

Tells devices to store their current output values as a default for the scene *sceneNumber*.

Synopsis
HTTP GET /json/zone/saveScene

Parameter

Parameter	Description	Remarks
sceneNumber	Numerical value	Mandatory
groupID	Number of the target group	Optional
groupName	Name of the target group	Optional

If the group parameters are omitted the command is sent as broadcast to all devices in a zone.

Response
HTTP Status 200

ok	true

Sample

```
GET /json/zone/saveScene?id=1237&groupNumber=2&sceneNumber=17
{
    "ok":true
}
```

3.3.3 undoScene

Tells devices to restore their output values to the previous state if the current scene matches the *sceneNumber*.

Synopsis
HTTP GET /json/zone/undoScene

Parameter

Parameter	Description	Remarks
sceneNumber	Numerical value	Mandatory
groupID	Number of the target group	Optional
groupName	Name of the target group	Optional

If the group parameters are omitted the command is sent as broadcast to all devices in the zone.

Response
HTTP Status 200

ok	true

Sample

```
GET /json/zone/undoScene?id=1237&sceneNumber=65
{
    "ok":true
}
```

3.3.4 sceneGetName

Get the user defined name for a scene *sceneNumber* within a group of a zone.

Synopsis
HTTP GET /json/zone/sceneGetName

Parameter

Parameter	Description	Remarks
sceneNumber	Numerical value	Mandatory
groupID	Number of the target group	M/O
groupName	Name of the target group	M/O

Either groupID or groupName must be supplied to this request.

Response
HTTP Status 200

result.name	the user defined name of the scene

Sample

```
GET /json/zone/sceneGetName?id=1237&sceneNumber=19&groupID=1
{
    "ok":true
    result": {
        "name": "Fernsehen"
    }
}
```

3.3.5 sceneSetName

Sets a user defined name for a scene *sceneNumber* within a group of a zone. This name is stored on the digitalSTROM Server only.

Synopsis
HTTP GET /json/zone/sceneSetName

Parameter

Parameter	Description	Remarks
newName	User defined name of the scene	Mandatory
sceneNumber	Numerical value	Mandatory
groupID	Number of the target group	M/O
groupName	Name of the target group	M/O

Either groupID or groupName must be supplied to this request.

Response
HTTP Status 200

ok	true

Sample

```
GET /json/zone/sceneSetName?id=1237&sceneNumber=17&groupID=2&newName="Fernsehen"
{
    "ok":true
}
```

3.3.6 getReachableScenes

Returns a list of groups which can be controlled by pushbuttons which are actually present in the zone.

Synopsis
HTTP GET /json/zone/getReachableScenes

Parameter

Parameter	Description	Remarks
groupID	Number of the target group	Optional
groupName	Name of the target group	Optional

If groupID or groupName are omitted the combined list for all groups is returned.

Response
HTTP Status 200

result.reachableScens	array of scene numbers

Sample

```
GET /json/zone/getReachableScenes?id=1237&groupID=1
{
    "ok": true,
    "result": {
        "reachableScenes": [
            0,
            1,
            5,
            6,
            17,
            18,
            19,
            29,
            30,
            31,
            38,
            39
        ]
    }
}
```

3.3.7 getLastCalledScene

Returns the *sceneNumber* which has been executed last for a group in a zone.

Synopsis
HTTP GET /json/zone/getLastCalledScene

Parameter

Parameter	Description	Remarks
groupID	Number of the target group	Optional
groupName	Name of the target group	Optional

Response
HTTP Status 200

result.scene	the number of the last called scene

Sample

```
GET /json/zone/getLastCalledScene?id=1237&groupID=1
{
    "ok": true,
    "result": {
        "scene": 0
    }
}
```

3.4 Value

3.4.1 Set Output Value

Set the output value of a group of devices in a zone to a given value.

> **Notice** Setting output values directly bypasses the group state machine and is unrecommended.

Synopsis
HTTP GET /json/zone/setValue

Parameter

Parameter	Description	Remarks
value	Numerical value	Mandatory
groupID	Number of the target group	Optional
groupName	Name of the target group	Optional

If the group parameters are omitted the command is sent as broadcast to all devices in the selected zone.

Notice Setting output values without a group identification is strongly unrecommended.

Response
HTTP Status 200

ok	true

Sample

```
GET /json/zone/setValue?id=1237&value=0&groupID=2
{
    "ok":true,
}
```

3.4.2 Blink

Executes the "blink" function on a group of devices in a zone for identification purposes.

Synopsis
HTTP GET /json/zone/blink

Parameter

Parameter	Description	Remarks
groupID	Number of the target group	Optional
groupName	Name of the target group	Optional

Response

HTTP Status 200

ok	true

Sample

```
GET /json/zone/blink?id=1237&groupID=1
{
    "ok":true,
}
```

4 Device

4.1 Common

Every /json/device/ function uses a common selection scheme for the device to which the command refers to. Either the parameter "dsid" or "name" must be given to identify the device.

Parameter	Description	Remarks
dsid	Device dSID String	Optional
name	Device Name	Optional
category	Request Category	Optional

A missing device identifier result in the following error message to be returned.

```
{ "ok": false, "message": "Need parameter name or dsid to identify device" }
```

If a device identifier does not match any actually known device in the installation the following error message is returned.

```
{ "ok": false, "message": "Could not find device named 'Wandlampe am Eingang'" }
```

The category parameter has an influence on how particular requests are treated, the goal is to prevent scene calls from automated scripts in certain situations. Currently supported categories are:

- manual
- timer
- algoirthm

A missing category parameter is currently treated as manual category, this compatibility will be removed in release 1.8.

4.2 Name

4.2.1 getName

Returns the user defined name of a device.

Synopsis
HTTP GET /json/device/getName

Parameter
None

Response
HTTP Status 200

result.name	identifier string for the device

Sample

```
GET /json/device/getName?dsid=3504175fe000000000017ef3
{
    "ok":true,
    "result" :
    {
        "name" : "App-Taster"
    }
}
```

4.2.2 setName

Sets the device name.

Synopsis
HTTP GET /json/device/setName

Parameter	Description	Remarks
newName	identifier string for the device	Mandatory

Parameter

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/setName?id=3504175fe000000000017ef3&newName="Wohnen"
{
    "ok":true
}
```

4.2.3 getSpec

Retrieves device and product information.

Synopsis
HTTP GET /json/device/getSpec

Parameter
None

Response
HTTP Status 200

result.functionID	Function ID of the device
result.productID	Product ID of the device
result.revisionID	Revision ID of the device

Sample

```
GET /json/device/getName?dsid=3504175fe000000000017ef3
{
    "ok": true,
    "result": {
        "functionID": 33027,
        "productID": 1224,
        "revisionID": 834
    }

}
```

4.3 Groups

4.3.1 getGroups

Returns a list of groups the device is registered in.

Synopsis
HTTP GET /json/device/getGroups

Parameter
None

Response
HTTP Status 200

result.groups	array of groups of the device

Sample

```
GET /json/device/getGroups
{
    "ok": true,
    "result": {
        "groups": [
            {
                "id": 3,
                "name": "blue"
            },
            {
                "id": 8,
                "name": "black"
```

```
            }
        ]
    }
}
```

4.4 Scene

4.4.1 callScene

Excutes the scene *sceneNumber* on a devices.

Synopsis
HTTP GET /json/device/callScene

Parameter

Parameter	Description	Remarks
sceneNumber	Numerical value	Mandatory
force	Boolean value, if set a forced scene call is issued	Optional

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/callScene?dsid=3504175fe000000000017ef3&sceneNumber=13
{
    "ok":true
}
```

4.4.2 saveScene

Tells the device to store the current output values as a default for the scene *sceneNumber*.

Synopsis
HTTP GET /json/device/saveScene

Parameter

Parameter	Description	Remarks
sceneNumber	Numerical value	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/saveScene?dsid=3504175fe000000000017ef3&sceneNumber=5
{
    "ok":true
}
```

4.4.3 undoScene

Tells devices to restore the output values to the previous state if the current scene matches the *sceneNumber*.

Synopsis
HTTP GET /json/device/undoScene

Parameter

Parameter	Description	Remarks
sceneNumber	Numerical value	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/undoScene?dsid=3504175fe000000000017ef3&sceneNumber=18
{
    "ok":true
}
```

4.4.4 turnOn

Tells devices to execute the scene MAX.

Synopsis
HTTP GET /json/device/turnOn

Parameter
None

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/turnOn?dsid=3504175fe000000000017ef3
{
    "ok":true
}
```

4.4.5 turnOff

Tells devices to execute the scene MIN.

Synopsis
HTTP GET /json/device/turnOff

Parameter
None

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/turnOff?dsid=3504175fe000000000017ef3
{
    "ok":true
}
```

4.4.6 increaseValue

Tells devices to execute the scene INC.

Synopsis
HTTP GET /json/device/increaseValue

Parameter
None

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/increaseValue?dsid=3504175fe000000000017ef3
{
    "ok":true
}
```

4.4.7 decreaseValue

Tells devices to execute the scene DEC.

Synopsis
HTTP GET /json/device/decreaseValue

Parameter
None

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/decreaseValue?dsid=3504175fe000000000017ef3
{
    "ok":true
}
```

4.5 Value

4.5.1 Set Value

Set the primary output value of a device to a given value.

> **Notice** Setting output values directly bypasses the group state machine and is unrecommended.

Synopsis
HTTP GET /json/device/setValue

Parameter

Parameter	Description	Remarks
value	Numerical 8 bit value, in the range from 0 to 255	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/setValue?dsid=3504175fe000000000017ef3&value=127
{
    "ok":true
}
```

4.5.2 Set Output Value

Set a output channel value of a device to a given value. The available output parameter ranges and channels depend on the feature of the hardware components.

> **Notice** Setting output values directly bypasses the group state machine and is unrecommended.

Synopsis
HTTP GET /json/device/setOutputValue

Parameter

Parameter	Description	Remarks
value	Numerical Value	Mandatory
offset	Output Channel Offset	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/setOutputValue?dsid=3504175fe000000000017ef3&value=5345&offset=0
{
    "ok":true
}
```

4.5.3 Get Output Value

Get the current output channel status of a device. The available output channels depend on the feature of the hardware components.

> **Notice** Getting output values directly from the device takes a noticeable amount of time. This request is subject of limitations in the systems certification rules.

Synopsis
HTTP GET /json/device/getOutputValue

Parameter

Parameter	Description	Remarks
offset	Output Channel Offset	Mandatory

Response
HTTP Status 200

result.offset	the given offset from the request
result.value	Numerical value of the selected output channel queried from the device

Sample

```
GET /json/device/getOutputValue?dsid=3504175fe000000000017ef3&offset=0
{
    "ok": true, "result": { "offset": 0, "value": 5345 }
}
```

4.5.4 Get Scene Value

Retrieves the device value of the given scene.

Synopsis
HTTP GET /json/device/getSceneValue

Parameter

Parameter	Description	Remarks
sceneID	Numerical value	Mandatory

Response
HTTP Status 200

result.value	value of the device
result.angle	angle value of the device, field available only for for GR-KL

Sample

```
GET /json/device/getSceneValue?dsid=3504175fe000000000017ef3&sceneID=72
{
    "ok":true,"result":{"value":65535,"angle":255}
}
```

4.5.5 Set Scene Value

Retrieves the device value of the given scene.

Synopsis
HTTP GET /json/device/setSceneValue

Parameter

Parameter	Description	Remarks
sceneID	Numerical value	Mandatory
value	Numerical value	Mandatory
angle	Numerical value, only applicable for GR-KL	Optional

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/setSceneValue?dsid=3504175fe000000000016c4f&sceneID=72&value=26987
{
    "ok":true
}
```

4.5.6 Blink

Executes the "blink" function on a device for identification purposes.

Synopsis
HTTP GET /json/device/blink

Parameter
None

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/blink?dsid=3504175fe000000000017ef3
{
    "ok":true
}
```

4.6 Configuration

4.6.1 setButtonID

Sets the button ID of a device. For details about the push button configuration see the ds-basics reference document.

Synopsis
HTTP GET /json/device/setButtonID

Parameter

Parameter	Description	Remarks
buttonID	Button number to set	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/setButtonID?dsid=3504175fe000000000016be7&buttonID=5
{
    "ok": true
}
```

4.6.2 setButtonInputMode

Sets the button input mode of a device. For details about the push button configuration see the ds-basics reference document.

Synopsis
HTTP GET /json/device/setButtonInputMode

Parameter

Parameter	Description	Remarks
modeID	Numerical value of the button mode to set	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/setButtonInputMode?dsid=3504175fe000000000016be7&modeID=0
{
    "ok": true
}
```

4.6.3 setOutputMode

Sets the output mode of a device.

Synopsis
HTTP GET /json/device/setOutputMode

Parameter

Parameter	Description	Remarks
modelID	Numerical value of the output mode to set	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/setOutputMode?dsid=3504175fe000000000016be7&modelID=0
{
    "ok": true
}
```

4.6.4 setJokerGroup

Sets the color group of a Joker device.

Synopsis
HTTP GET /json/device/setJokerGroup

Parameter

Parameter	Description	Remarks
groupID	Group number to set	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/setJokerGroup?dsid=3504175fe000000000016be7&groupID=2
{
    "ok": true
}
```

4.6.5 setButtonActiveGroup

Sets the user group of a push button device.

Synopsis
HTTP GET /json/device/setButtonActiveGroup

Parameter

Parameter	Description	Remarks
groupID	Group number to set	Mandatory, value range between 0 and 63,use 0xff to reset

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/setButtonActiveGroup?dsid=3504175fe000000000016be7&groupID=20
{
    "ok": true
}
```

4.6.6 getSceneMode

Reads the device configuration flags for a given *sceneID*. For details about the scene configuration see the ds-basics reference document.

Synopsis
HTTP GET /json/device/getSceneMode

Parameter

Parameter	Description	Remarks
groupID	Group number to set	Mandatory

Response
HTTP Status 200

sceneID	Scene number which has been requested
dontCare	Don't Care Flag
localPrio	Local Prio Flag
specialMode	Special Mode Flag
flashMode	Flashing Mode Flag
ledconIndex	Index of the LED configuration register
dimmTimeIndex	Index of the transition configuration register

Sample

```
GET /json/device/getSceneMode?dsid=3504175fe000000000016be7&sceneID=5
{
    "ok": true,
    "result":
    {
        "sceneID": 5,
        "dontCare": false,
        "localPrio": false,
        "specialMode": false,
        "flashMode": false,
        "ledconIndex": 1,
        "dimtimeIndex": 1
    }
}
```

4.6.7 setSceneMode

Sets the device configuration flags for a given *sceneID*. For details about the scene configuration see the ds-basics reference document.

Synopsis
HTTP GET /json/device/setSceneMode

Parameter

Parameter	Description	Remarks
sceneID	Scene number which has been requested	Mandatory
dontCare	Don't Care Flag	Optional
localPrio	Local Prio Flag	Optional
specialMode	Special Mode Flag	Optional
flashMode	Flashing Mode Flag	Optional
ledconIndex	Index of the LED configuration register	Optional
dimmtimeIndex	Index of the transition configuration register	Optional

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/setSceneMode?dsid=3504175fe000000000016be7&sceneID=5&dimtimeIndex=2&dontCare=true
     { "ok": true }
```

4.6.8 getTransitionTime

Reads the device transition time configuration for a given register set. For details about the transition time configuration see the ds-basics reference document.

Synopsis
HTTP GET /json/device/getTransitionTime

Parameter

Parameter	Description	Remarks
dimtimeIndex	Index of the transition configuration register	Mandatory

Response
HTTP Status 200

dimmtimeIndex	Index of the transition configuration register
up	Ramptime up in Milliseconds
down	Ramptime down in Milliseconds

Sample

```
GET /json/device/getTransitionTime?dsid=3504175fe000000000016be7&dimtimeIndex=2
{
    "ok": true,
    "result":
    {
        "dimtimeIndex": 2,
        "up": 600,
        "down": 55
    }
}
```

4.6.9 setTransitionTime

Sets the device transition time configuration for a given register set. For details about the transition time configuration see the ds-basics reference document.

Synopsis
HTTP GET /json/device/setTransitionTime

Parameter

Parameter	Description	Remarks
dimtimeIndex	Index of the transition configuration register	Mandatory
up	Ramptime up in Milliseconds	Mandatory
down	Ramptime down in Milliseconds	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/setTransitionTime?dsid=3504175fe000000000016be7&dimtimeIndex=2&up=600&down=600
{
    "ok": true
}
```

4.6.10 setConfig

Write a configuration value of a config class parameter to the device.

> **Notice** Writing configuration parameters directly to the device may lead to malfunctions including complete failure of the whole device. Do not write parameters or values unless you are sure that the device supports it.

Synopsis
HTTP GET /json/device/setConfig

Parameter

Parameter	Description	Remarks
class	Configuration Class	Mandatory
index	Parameter Index	Mandatory
value	Parameter Value	Mandatory

Response
HTTP Status 200

class	the class parameter from the request
index	the index parameter from the request
value	parameter value

Sample

```
GET /json/device/setConfig?dsid=3504175fe000000000016be7&class=3&index=0&value=33
{
    "ok": true
}
```

4.6.11 getConfig

Reads a 8 bit parameter value of a config class from the device.

> **Notice** Getting parameter values directly from the device takes a noticeable amount of time. This request is subject of limitations in the systems certification rules.

Synopsis
HTTP GET /json/device/getConfig

Parameter

Parameter	Description	Remarks
class	Configuration class	Mandatory
index	Parameter index	Mandatory

Response
HTTP Status 200

class	the class parameter from the request
index	the index parameter from the request
value	parameter value

Sample

```
GET /json/device/getConfig?dsid=3504175fe000000000016be7&class=1&index=2
{
    "ok": true,
    "result":
```

```
    {
        "class": 1,
        "index": 2,
        "value": 231
    }
}
```

4.6.12 getConfigWord

Reads a 16 bit parameter value of a config class from the device.

> **Notice** Getting parameter values directly from the device takes a noticeable amount of time. This request is subject of limitations in the systems certification rules.

Synopsis
HTTP GET /json/device/getConfigWord

Parameter

Parameter	Description	Remarks
class	Configuration class	Mandatory
index	Parameter index, even	Mandatory

Response
HTTP Status 200

class	the class parameter from the request
index	the index parameter from the request
value	parameter value

Sample

```
GET /json/device/getConfigWord?dsid=3504175fe000000000016be7&class=3&index=2
{
    "ok": true,
    "result":
    {
        "class": 3,
        "index": 2,
        "value": 65280
    }
}
```

4.7 Sensor

4.7.1 Get Sensor Value

Ready a sensor measurement from a device.

Synopsis
HTTP GET /json/device/getSensorValue

Parameter

Parameter	Description	Remarks
sensorIndex	Numerical value, in the range from 0 to 14	Mandatory

Response
HTTP Status 200

sensorIndex	the index parameter from the request
sensorValue	the actual measurement read from the device

Sample

```
GET /json/device/getSensorValue?dsid=3504175fe000000000017ef3&sensorIndex=4
{
    "ok": true,
    "result": {
        "sensorIndex": 4,
        "sensorValue": 0
    }
}
```

4.7.2 Get Sensor Type

Ready the sensor type description from a device. For details about sensor types see the ds-basics reference document.

Synopsis
HTTP GET /json/device/getSensorType

Parameter

Parameter	Description	Remarks
sensorIndex	Numerical value, in the range from 0 to 14	Mandatory

Response
HTTP Status 200

sensorIndex	the index parameter from the request
sensorType	the sensor type read from the device

Sample

```
GET /json/device/getSensorType?dsid=3504175fe000000000017ef3&sensorIndex=4
{
    "ok": true,
    "result": {
        "sensorIndex": 4,
        "sensorType": 6
    }
}
```

4.7.3 getSensorEventTableEntry

Reads the device event configuration for a given index. For details about the event table configuration see the ds-basics reference document.

Synopsis
HTTP GET /json/device/getSensorEventTableEntry

Parameter

Parameter	Description	Remarks
eventIndex	Index of the event configuration entry	Mandatory

Response
HTTP Status 200

eventIndex	Index of the event configuration register
eventName	User defined name of this event
sensorIndex	Sensor index on which this entry operates
action	Action value
value	Threshold value
test	Comparison operator
hysteresis	Hysteresis value
validity	Enabled Flag

Sample

```
GET /json/device/getSensorEventTableEntry?dsid=3504175fe00000000001540c&eventIndex=0
{
    "ok": true,
    "result": {
        "eventIndex": 0,
        "eventName": "",
        "sensorIndex": 2,
        "test": 2,
```

```
        "action": 0,
        "value": 35,
        "hysteresis": 0,
        "validity": 2
    }
}
```


4.7.4 setSensorEventTableEntry

Sets the device event configuration for a given index. For details about the event table configuration see the ds-basics reference document.

Synopsis
HTTP GET /json/device/setSensorEventTableEntry

Parameter

Parameter	Description	Remarks
eventIndex	Index of the event configuration register	Mandatory
eventName	User defined name of this event	Mandatory
sensorIndex	Sensor index on which this entry operates	Mandatory
action	Action value	Mandatory
value	Threshold value	Mandatory
test	Comparison operator	Mandatory
hysteresis	Hysteresis value	Mandatory
validity	Enabled Flag	Mandatory

Response
HTTP Status 200

ok	true

Sample


```
GET /json/device/setSensorEventTableEntry?dsid=3504175fe00000000001540c&eventIndex=0&eventName="TV
      turned on"&sensorIndex=2&test=2&action=0&value=50&hysteresis=25&validity=2
{
    "ok": true
}
```


4.8 Programming

4.8.1 Set Programming Mode

Enabled or disabled the programming mode on a device.

Synopsis
HTTP GET /json/device/setProgMode

Parameter

Parameter	Description	Remarks
mode	mode value, either enabled or disabled	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/setProgMode?dsid=3504175fe000000000017ef3&mode=disabled
{
    "ok": true
}
```

4.8.2 Add To Area

Modify the device scene table configuration and activate the area scene.

Synopsis
HTTP GET /json/device/addToArea

Parameter

Parameter	Description	Remarks
areaScene	either the area-on or area-off scenes	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/addToArea?dsid=3504175fe000000000017ef3&areaScene=7
{
    "ok": true
}
```

4.8.3 Remove From Area

Modify the device scene table configuration and deactivate the area scene.

Synopsis
HTTP GET /json/device/removeFromArea

Parameter

Parameter	Description	Remarks
areaScene	either the area-on or area-off scenes	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/device/removeFromArea?dsid=3504175fe000000000017ef3&areaScene=7
{
    "ok": true
}
```

4.8.4 Get Transmission Quality

Sends test commands to a device to evaluate the actual transmission quality.

Synopsis
HTTP GET /json/device/getTransmissionQuality

Parameter
None

Response
HTTP Status 200

upstream	a numerical value in the range of 0 to 62, 62 meaning best quality
downstream	a numerical value in the range 0 to 6, 0 meaning best quality

Sample

```
GET /json/device/getTransmissionQuality?dsid=3504175fe000000000017ef3
{
    "ok": true,
    "result": {
        "upstream": 61,
        "downstream": 0
    }
}
```

5 Circuit

5.1 Common

Every /json/circuit/ function uses a common selection scheme for the digitalSTROM Meter to which the command refers to. The parameter "id" must be given to identify the dSM which is a string value of the dSID.

Parameter	Description	Remarks
id	dSID Number of the digitalSTROM Meter	Mandatory

A missing dsid identifier result in the following error message to be returned.

```
{
   "ok": false,
   "message": "Missing parameter id"

}
```

If a dSID identifier does not match any actually known digitalSTROM Meter in the installation the following error message is returned.

```
{
    "ok": false,
    "message": "Could not find dSMeter with given dsid"
}
```

5.2 Name

5.2.1 getName

Returns the user defined name of the zone.

Synopsis
HTTP GET /json/circuit/getName

Parameter
None

Response
HTTP Status 200

name	identifier string for the digitalSTROM Meter

Sample

```
GET /json/circuit/getName?id=3504175fe0000010000004d5
{
    "ok":true,
    "result" : {
        "name" : "Wohnen/Flur/Eingang"
    }
}
```

5.2.2 setName

Sets the zone name.

Synopsis
HTTP GET /json/circuit/setName

Parameter	Description	Remarks
newName	identifier string for the digitalSTROM Meter	Mandatory

Parameter

Response
HTTP Status 200

ok	true

Sample

```
GET /json/circuit/setName?id=3504175fe0000010000004d5&newName="Wohnen"
{
    "ok":true
}
```

5.3 Energy Meter

5.3.1 getConsumption

Returns the current measurent of the power consumption on this circuit.

Synopsis
HTTP GET /json/circuit/getConsumption

Parameter
None

Response
HTTP Status 200

consumption	Current power consumption [W]

Sample

```
GET /json/circuit/getConsumption?id=3504175fe0000010000004d5
{
    "ok": true,
    "result" : {
        "consumption": 725
    }
}
```

5.3.2 getEnergyMeterValue

Returns the current measurent of the power consumption on this circuit.

Synopsis
HTTP GET /json/circuit/getEnergyMeterValue

Parameter
None

Response
HTTP Status 200

meterValue	Energy Meter Value [Ws]

Sample

```
GET /json/circuit/getEnergyMeterValue?id=3504175fe000001000004d5
{
    "ok": true,
    "result": {
        "meterValue": 1438467
    }
}
```

6 Structure

6.1 Zone

6.1.1 addZone

Adds a zone with the given Id. The zone is added to the digitalSTROM Server data model only and initially does not have any devices associated.

Synopsis
HTTP GET /json/structure/addZone

Parameter

Parameter	Description	Remarks
zoneID	unique numerical identifier for the new zone	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/structure/addZone?zoneID=1
{
    "ok":true
}
```

6.1.2 removeZone

Removes the zone with the give Id from the installation. A zone can only be removed if it has no associated devices.

Synopsis
HTTP GET /json/structure/removeZone

Parameter	Description	Remarks
zoneID	unique numerical identifier	Mandatory

Parameter

Response
HTTP Status 200

ok	true

Sample

```
GET /json/structure/removeZone?zoneID=1234
{
    "ok":true
}
```

6.2 Group

6.2.1 addGroup

Adds a user group to the zone with the given Id.

Synopsis
HTTP GET /json/structure/addGroup

Parameter

Parameter	Description	Remarks
zoneID	unique numerical identifier for the new zone	Mandatory
groupID	numerical identifier for the new group	Mandatory if groupAutoSelect is
groupAutoSelect	group type 'user' or 'global'	Mandatory if groupID is not give

Response
HTTP Status 200

result.groupID	numeric identifier for the new group
result.zoneID	numeric identifier for the zone
result.groupName	string, name of the new group
result.groupColor	numeric identifier, color of the new group

Sample

```
GET /json/structure/addGroup?zoneID=0&groupAutoSelect=global&groupColor=5&groupName=test
{
    "ok":true,
    "result":
    {
        "groupID":18,
        "zoneID":0,
        "groupName":"test",
        "groupColor":5
    }
}
```

6.2.2 removeGroup

Removes a user group to the zone with the given Id.

Synopsis
HTTP GET /json/structure/removeGroup

Parameter

Parameter	Description	Remarks
zoneID	unique numerical identifier for the zone	Mandatory
groupID	numerical identifier for the group	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/structure/removeGroup?zoneID=0&groupID=18
{
    "ok":true
}
```

6.2.3 groupAddDevice

Adds a device to the user group. Only active devices can be added to additional groups.

Synopsis
HTTP GET /json/structure/groupAddDevice

Parameter	Description	Remarks
deviceID	DSID of the device	Mandatory
groupID	unique numerical group identifier	Mandatory

Parameter

Response
HTTP Status 200

ok	true

Sample

```
GET /json/structure/groupAddDevice?deviceID=3504175fe000000000005854&groupID=16
{
    "ok":true
}
```

6.2.4 groupSetName

Rename a group.

Synopsis
HTTP GET /json/structure/groupSetName

Parameter

Parameter	Description	Remarks
zoneID	unique numerical identifier for the zone	Mandatory
groupID	numerical identifier for the group	Mandatory
newName	string, new name for the group	Mandatory

Response
HTTP Status 200

ok	true

Sample

```
GET /json/structure/groupSetName?zoneID=0&groupID=18&newName=test
{
    "ok":true
}
```

6.2.5 groupSetColor

Change color of the group.

Synopsis
HTTP GET /json/structure/groupSetColor

Parameter	Description	Remarks
zoneID	unique numerical identifier for the zone	Mandatory
groupID	numerical identifier for the group	Mandatory
newColor	numerical identifier of the color for the group	Mandatory

Parameter

Response
HTTP Status 200

ok	true

Sample

```
GET /json/structure/groupSetColor?zoneID=0&groupID=18&newColor=4
{
    "ok":true
}
```

6.3 Device

6.3.1 zoneAddDevice

Associates a device with a new zone. A device is automatically removed from the old zone. Only active devices can be moved to a new zone because the zone configuration has to be synchronized with the device itself.

Synopsis
HTTP GET /json/structure/zoneAddDevice

Parameter

Parameter	Description	Remarks
deviceID	DSID of the device to move	Mandatory
zone	unique numerical identifier for the new zone	Mandatory

Response
HTTP Status 200

movedDevices	array of devices which have been moved

In the case of a failure various different error messages may occur.

Sample

```
GET /json/structure/zoneAddDevice?deviceID= &zone=1
{
    ok: true
    result: {
        movedDevices: [
            {
                id: "3504175fe000000000005854"
                name: ""
                functionID: 4144
                productRevision: 788
                productID: 1234
                hwInfo: "GE-TKM210"
                meterDSID: "3504175fe0000010000004d9"
                busID: 241
                zoneID: 1
                isPresent: true
                lastDiscovered: "2012-11-22 10:35:05"
                firstSeen: "2012-11-19 14:34:02"
                inactiveSince: "1970-01-01 01:00:00"
                outputMode: 16
                buttonID: 12
                buttonActiveGroup: 1
                buttonInputMode: 0
                buttonInputIndex: 0
                buttonInputCount: 1
                groups: [
                    "1"
                ]
            }
        ]
    }
}
```

6.3.2 removeDevice

Removes a device from the data model. Only inactive devices can be removed.

Synopsis
HTTP GET /json/structure/removeDevice

Parameter	Description	Remarks
deviceID	DSID of the device to be removed	Mandatory

Parameter

Response
HTTP Status 200

ok	true

Sample

```
GET /json/structure/removeDevice?deviceID=3504175fe000000000005854
{
    ok: false
    message: "Cannot remove present device"
}
```

7 Event

7.1 Raise

7.1.1 raise

Raises an event and appends it to the digitalSTROM Server event queue. Details of the digitalSTROM Server event processing can be found in the system-interfaces document.

> **Notice** System events should be treated as reserved and must not be raised by external applications. In this term system events are events which originate from the digitalSTROM system lower layers.

Synopsis
HTTP GET /json/event/raise

Parameter

Parameter	Description	Remarks
name	identifier string for event	Mandatory
parameter	list of key=value pairs, seperated with semicolons	Optional

Response
HTTP Status 200

ok	true

Sample

```
GET /json/event/raise?name=highlevelevent&parameter=id=1026;value=0;index=1
{
    "ok":true
}
```

7.2 Subscription

7.2.1 subscribe

Subscribe to an event with the given name and registers the callers subscriptionId. A unique subscriptionId can be selected by the subscriber. It is possible to subscribe to several events reusing the same subscriptionId.

Synopsis
HTTP GET /json/event/subscribe

Parameter	Description	Remarks
name	identifier string for the event	Mandatory
subscriptionID	numerical unique value	Mandatory

Parameter

Response
HTTP Status 200

ok	true

Sample

```
GET /json/event/subscribe?name=deviceSensorEvent&subscriptionID=42
{
    "ok":true
}
```

7.2.2 unsubscribe

Unsubscribes for the previously registered events by giving the event name and the unique subscriptionId.

Synopsis
HTTP GET /json/event/unsubscribe

Parameter	Description	Remarks
name	identifier string for the event	Mandatory
subscriptionID	numerical unique value	Mandatory

Parameter

Response
HTTP Status 200

ok	true

If there is no registered session for the given event name the following error message is returned.

```
{
    ok: false
    message: "Event "callScene" is not subscribed in this session"
}
```

If the subscriptionId is unknown to the digitalSTROM Server the following error message is returned.

```
{
    ok: false
    message: "Token not found!"
}
```

Sample

```
GET /json/event/unsubscribe?name=callScene&subscriptionID=42
{
    ok: true
}
```

7.2.3 get

Get event and context information for an event subscription. All events subscribed with the given Id will be handled by this call. An optional timeout value in milliseconds can be specified and will block the call until either an event or the timeout occurs. If the timeout value is zero or missing the call will not timeout.

Synopsis
HTTP GET /json/event/get

Parameter	Description	Remarks
subscriptionID	numerical unique value	Mandatory
timeout	numerical value, timeout in milli seconds	Optional

Parameter

Response
HTTP Status 200

events	array of events

Sample

```
GET /json/event/get?subscriptionID=42&timeout=60000
{
    ok: true
    result: {
        events: [ ]
    }
}
```

```
GET /json/event/get?subscriptionID=42&timeout=60000
{
    ok: true
    result: {
        events: [
            {
                name: "callScene"
                properties: {
                    groupID: "1"
                    sceneID: "8"
                    zoneID: "1241"
                    originDeviceID: "3504175fe000000000005854"
                }
            }
        ]
    }
}
```

8 Metering

8.1 Metering

8.1.1 getResolution

Returns a list of time-series metering data resolutions stored on the digitalSTROM Server.

Synopsis
HTTP GET /json/metering/getResolutions

Parameter
None

Response
HTTP Status 200

Parameter	Description
ok	boolean result of the call
result.resolutions	a list of supported resolutions
result.resolutions[...].resolution	step size in thei resolution in seconds

Sample

```
GET /json/metering/getResolutions
{
    "ok": true,
    "result": {
        "resolutions": [
            {
                "resolution": 1
            },
            {
                "resolution": 60
            },
            {
                "resolution": 900
            },
            {
                "resolution": 86400
            },
            {
                "resolution": 604800
            },
            {
                "resolution": 2592000
            }
        ]
    }
}
```

8.1.2 getSeries

Returns a list of all metering series stored on the digitalSTROM Server.
Three types of series are available:

energy An energy meter counter.

energyDelta The total energy consumed during the previous time slot.

consumption The average power used during the previous time slot.

Synopsis
HTTP GET /json/metering/getSeries

Parameter
None

Response
HTTP Status 200

Parameter	Description
ok	boolean result of the call
result.series	a list of available time series
result.series[...].dsid	dSID of the digitalSTROM Meter for this series
result.series[...].type	the series type

Sample

```
GET /json/metering/getSeries
{
    "ok": true,
    "result": {
        "series": [
            {
                "dsid": "3504175fe00000100000053e",
                "type": "energy"
            },
            {
                "dsid": "3504175fe0000010000006b4",
                "type": "energyDelta"
            },
            {
                "dsid": "3504175fe0000010000008a5",
                "type": "consumption"
            },
        ]
    }
}
```

8.1.3 getValues

Returns a time series of metering values with the specified properties.

All times are integers that represent UNIX timestamps (seconds since 1970-01-01).

The (optional) window selection parameters can be used in different combinations. Only two of the three options can be used together in a call. The

following table details the available combinations:

startTime return all available values starting at startTime until now.

endTime return all available values from the oldest available until endTime.

valueCount return the valueCount newest values

startTime and valueCount return valueCount values starting from startTime.

endTime and valueCount return valueCount values ending at endTime

startTime and endTime return the values between startTime and endTime.

Synopsis
HTTP GET /json/metering/getValues

Parameter

Parameter	Description	Remarks
dsid	request the data for this digitalSTROM Meter	Mandatory
type	series type (according to the getSeries call)	Mandatory
resolution	series resolution (the digitalSTROM Server will adjust the resolution to the closest multiple of the available resolutions according to the getResolutions call)	Mandatory
unit	(only relevant for types "energy" and "energyDelta") unit of the returned metering values. Options are "Wh" and "Ws". Defaults to "Wh".	Optional
startTime	start time (UNIX timestamp)	Optional
endTime	ent time (UNIX timestamp)	Optional
valueCount	number of values (UNIX timestamp)	Optional

Response
HTTP Status 200

Parameter	Description
ok	boolean result of the call
result.meterID	dSID of the digitalSTROM Meter
result.type	same as Request
result.resolution	actual resolution of the data, might differ from the requested resolution if it was not available.
result.values	array of time-value pairs

Sample

```
GET /json/metering/getValues?dsid=3504175fe00000100000063a&type=energy&resolution=60&unit=Ws&
     valueCount=5
{
    "ok": true,
    "result": {
        "meterID": "3504175fe00000100000063a",
        "type": "energy",
        "unit": "Ws",
        "resolution": "60",
        "values": [
            [
                1352906040,
                47562600
            ],
            [
                1352906100,
                47562600
            ],
            [
                1352906160,
                47562600
            ],
            [
                1352906220,
                47562600
            ],
            [
                1352906280,
                47562600
            ]
        ]
    }
}
```

8.1.4 getLatest

Returns the latest available metering values.

Synopsis
HTTP GET /json/metering/getLatest

Parameter

Parameter	Description	Remarks
from	the dSID of the requested digitalSTROM Meters. It uses a Set-Syntax: ".meters(dsid1,dsid2,...)" and ".meters(all)"	Mandatory
type	series type (according to the getSeries call)	Mandatory
unit	(only relevant for types "energy" and "energyDelta") unit of the returned metering values. Options are "Wh" and "Ws". Defaults to "Wh".	Optional

Response
HTTP Status 200

Parameter	Description
ok	boolean result of the call
result.values	array of results
result.values[...].dsid	dSID of the digitalSTROM Meter
result.values[...].value	the latest metering value
result.values[...].date	date and time when the latest metering value was recorded

Sample

```
GET /json/metering/getLatest?from=.meters(3504175fe00000100000063a,3504175fe0000010000008c4)&type=
    energy&unit=Ws
{
    "ok": true,
    "result": {
        "values": [
            {
                "dsid": "3504175fe00000100000063a",
                "value": 49414887,
                "date": "2012-11-19 13:49:41"
            },
            {
                "dsid": "3504175fe0000010000008c4",
                "value": 151215631,
                "date": "2012-11-19 10:29:29"
            }
        ]
    }
}
```

9 System

9.1 System Information

9.1.1 version

Returns the version of the digitalSTROM Server software.

Synopsis
HTTP GET /json/system/version

Parameter
None

Response
HTTP Status 200

result.version	identifier string for the dSS version

Sample

```
GET /json/systme/version
{  "ok":true, "result" : { "version" : "dSS v1.17.3 (1.17.3) (oebuild@oe-build)" } }
```

9.1.2 time

Sets the installation name.

Synopsis
HTTP GET /json/system/time

Parameter
None

Response
HTTP Status 200

result.time	number of seconds since the Epoch, 1970-01-01 00:00:00 +0000 (UTC)

Sample

```
GET /json/system/time
{ "ok":true, "result" : { "time" : 1353510820 } }
```

9.1.3 getDSID

Returns the dSID of the digitalSTROM Server.

Synopsis
HTTP GET /json/system/getDSID

Parameter
None

Response
HTTP Status 200

result.dsid	dSID of the dSS

Sample

```
GET /json/systme/getDSID
{  "ok":true, "result" : { "dSID" : ""3504175FEFF0120345678901"" } }
```

9.2 Authentication

9.2.1 login

Creates a new session using the provided credentials.

Synopsis
HTTP GET /json/system/login

Parameter

Parameter	Description	Remarks
user	user name string	Mandatory
password	password string	Mandatory

Response
HTTP Status 200

result.token	session token as string

Sample

```
GET /json/system/login?user=dssadmin\&password=dssadmin
```

```
{
  "ok" : true,
  "result" : { "token" : "cea026b6f9d69e57e030736076285da77dbf117d24dbec69e349b2fb4ab7425e" }
}
```

9.2.2 logout

Destroys the session and signs out the user.

Synopsis
HTTP GET /json/system/logout

Parameter
None

Response
HTTP Status 200

Sample

```
GET /json/system/logout { "ok" : true }
```

9.2.3 loggedInUser

Returns the name of the currently logged in user.

Synopsis
HTTP GET /json/system/loggedInUser

Parameter
None

Response
HTTP Status 200

result.name	name of the currently logged in user

Note: if noone is currently logged in, the result will be empty, i.e. name will be missing.

Sample

```
GET /json/system/loggedInUser
{ "ok" : true, "result" : { "name" : "dssadmin" } }
```

9.2.4 setPassword

Changes the password of the currently logged in user.

Synopsis
HTTP GET /json/system/setPassword

Parameter	Description	Remarks
password	new password	Mandatory

Parameter

Response
HTTP Status 200

Sample

```
GET /json/system/setPassword
{ "ok" : true, "message" : "Password changed, have a nice day" }
```

9.2.5 requestApplicationToken

Returns a token for paswordless login. The token will need to be approved by a user first, the caller must not be logged in.

Synopsis
HTTP GET /json/system/requestAppplicationToken

Parameter	Description	Remarks
applicationName	name of the application that requests the token	Mandatory

Parameter

Response
HTTP Status 200

result.applicationToken	application token as string

Sample

```
GET /json/system/requestApplicationToken?applicationName=Example
```

```
{
    "ok" : true,
    "result" :
    {
        "applicationToken" : "4fa07386c77d7f32260066c83b58aece5814698376bd03f0e3b5764e58f0ec1a"
    }
}
```

9.2.6 enableToken

Enables an application token, caller must be logged in.

Synopsis
HTTP GET /json/system/enableToken

Parameter	Description	Remarks
applicationToken	application token as string	Mandatory

Parameter

Response
HTTP Status 200

Sample

```
GET /json/system/enableToken?applicationToken=4
      fa07386c77d7f32260066c83b58aece5814698376bd03f0e3b5764e58f0ec1a
```

```
{ "ok" : true }
```

9.2.7 revokeToken

Revokes an application token, caller must be logged in.

Synopsis
HTTP GET /json/system/revokeToken

Parameter	Description	Remarks
applicationToken	application token as string	Mandatory

Parameter

Response
HTTP Status 200

Sample

```
GET /json/system/revokeToken?applicationToken=4
      fa07386c77d7f32260066c83b58aece5814698376bd03f0e3b5764e58f0ec1a

{ "ok" : true }
```

9.2.8 loginApplication

Creates a new session using the registered application token.

Synopsis
HTTP GET /json/system/loginApplication

Parameter	Description	Remarks
loginToken	application token as string	Mandatory

Response
HTTP Status 200

result.token	session token as string

Sample

```
GET /json/system/loginApplication?loginToken=4
      fa07386c77d7f32260066c83b58aece5814698376bd03f0e3b5764e58f0ec1a

{
    "ok" : true,
    "result" : { "token" : "a84bfd1219512078d5537a4a5cd1c78084e6c4d3f8b0ef2ae3a2c81dff638822" }
}
```

10 Property Tree

10.1 Basic Property Tree Operations

10.1.1 getString

Returns the string value of the property, this call will fail if the property is not of type 'string'.

Synopsis
HTTP GET /json/property/getString

Parameter

Parameter	Description	Remarks
path	path of the property	Mandatory

Response
HTTP Status 200

result.value	string value of the property

Sample

```
GET /json/property/getString?path=/system/version/version
{ "ok" : true, "result" : { "value" : "1.17.3" } }
```

10.1.2 setString

Sets the string value of the property, this call will fail if the property is not of type 'string'.

Synopsis
HTTP GET /json/property/setString

Parameter

Parameter	Description	Remarks
path	path of the property	Mandatory
value	string value to set	Mandatory

Response
HTTP Status 200

Sample

```
GET /json/property/setString?path=/testpath/teststring\&value=testvalue
```

```
{ "ok" : true }
```

10.1.3 getInteger

Returns the integer value of the property, this call will fail if the property is not of type 'integer'.

Synopsis
HTTP GET /json/property/getInteger

Parameter

Parameter	Description	Remarks
path	path of the property	Mandatory

Response
HTTP Status 200

result.value	integer value of the property

Sample

```
GET /json/property/getInteger?path=/system/uptime
```

```
{ "ok" : true, "result" : { "value" : 7539 } }
```

10.1.4 setInteger

Sets the integer value of the property, this call will fail if the property is not of type 'integer'.

Synopsis
HTTP GET /json/property/setInteger

Parameter

Parameter	Description	Remarks
path	path of the property	Mandatory
value	integer value of the property	Mandatory

Response
HTTP Status 200

Sample

```
GET /json/property/setInteger?path=/testpath/testint\&value=1
{ "ok" : true }
```

10.1.5 getBoolean

Returns the boolean value of the property, this call will fail if the property is not of type 'boolean'.

Synopsis
HTTP GET /json/property/getBoolean

Parameter

Parameter	Description	Remarks
path	path of the property	Mandatory

Response
HTTP Status 200

result.value	boolean value of the property

Sample

```
GET /json/property/getBoolean?path=/config/subsystems/Metering/enabled
{ "ok" : true, "result" : { "value" : true } }
```

10.1.6 setBoolean

Returns the boolean value of the property, this call will fail if the property is not of type 'boolean'.

Synopsis
HTTP GET /json/property/setBoolean

Parameter

Parameter	Description	Remarks
path	path of the property	Mandatory
value	boolean value of the property	Mandatory

Response
HTTP Status 200

Sample

```
GET /json/property/setBoolean?path=/testpath/testbool\&value=true
```

```
{ "ok" : true }
```

10.1.7 getChildren

Returns an array of child nodes.

Synopsis
HTTP GET /json/property/getChildren

Parameter

Parameter	Description	Remarks
path	path of the node	Mandatory

Response
HTTP Status 200

result[]	result is an array of child nodes

Sample

```
GET /json/property/getChildren?path=/system/host/interfaces/lo
```

```
{
    "ok" : true,
    "result":
```

```
    [
        { "name" : "mac", "type" : "string"},
        { "name" : "ip", "type" : "string"},
        { "name" : "netmask", "type" : "string"}
    ]
}
```

10.1.8 getType

Returns the type of the property, this can be ``none'', ``string'', ``integer'' or ``boolean''.

Synopsis
HTTP GET /json/property/getType

Parameter

Parameter	Description	Remarks
path	path of the property	Mandatory

Response
HTTP Status 200

result.type	type of the property

Sample

```
GET /json/property/getType?path=/system/host/interfaces/lo/mac
```

```
{ "ok" : true, "result" : { "type" : "string" } }
```

10.1.9 getFlags

Returns the flag values of a property.

Synopsis
HTTP GET /json/property/getFlags

Parameter

Parameter	Description	Remarks
path	path of the property	Mandatory

Response
HTTP Status 200

result.READABLE	information about the READABLE flag
result.WRITEABLE	information about the WRITEABLE flag
result.ARCHIVE	information about the ARCHIVE flag

Sample

```
GET /json/property/getFlags?path=/system/host/interfaces/lo/mac
{ "ok" : true, "result" : { "READABLE" : true, "WRITEABLE" : true, "ARCHIVE" : false } }
```

10.1.10 setFlag

Sets a given flag of a property.

Synopsis
HTTP GET /json/property/setFlag

Parameter

Parameter	Description	Remarks
path	path of the property	Mandatory
flag	flag identifier	Mandatory
value	boolean flag value	Mandatory

Response
HTTP Status 200

Sample

```
GET /json/property/setFlag?path=/system/host/interfaces/lo/mac\&flag=WRITEABLE\&value=true
{ "ok" : true }
```

10.1.11 remove

Removes a property node.

Synopsis
HTTP GET /json/property/remove

Parameter

Parameter	Description	Remarks
path	path of the property	Mandatory

Response
HTTP Status 200

Sample

```
GET /json/property/remove?path=/testpath
```

```
{ "ok" : true }
```

10.2 Property Query

10.2.1 query

Returns a part of the tree specified by query. All queries start from the root.
The properties to be included have to be put in parentheses. A query to get
all device from zone4 would look like this: '/apartment/zones/zone4/*(ZoneID,name)'.
More complex combinations (see example below) are also possible.

Synopsis
HTTP GET /json/property/query

Parameter

Parameter	Description	Remarks
query	query string	Mandatory

Response
HTTP Status 200

result.value	string value of the property

Sample

```
GET /json/property/query?query=/apartment/zones/{*}(ZoneID,scenes)/groups/{*}(group,name)/scenes
    /{*}(scene,name)
```

```
{
```

```
    "ok":true,
    "result":
    {
        "zones":
        [
            {
                "ZoneID":3663,
                "groups":
                [
                    {
                        "group":1,
                        "name":"yellow",
                        "scenes":
                        [
                            {
                                "scene":5,
                                "name":"demo scene"
                            }
                        ]
                    },
                    {
                        "group":2,
                        "name":"gray",
                        "scenes":[]
                    }
                ]
            }
        ]
    }
}
```