

This document contains the specifications for the Manager software.

## HTTP protocol

**url:** requests sent to the manager should have a url of the following format

**http://<manager name | IP>:<port>/manager.php?action=<action>[&f1=v1[&...]]**

**port:** The port that manager listens on is undefined. As part of the response to a device's advertize broadcast the manager must include a port.

**events:** The manager must handle a minimal set of actions. The action type is encoded into the GET or POST query string in the standard way

(eg ?action=eventtype&field=val1&field2=val2...).

All managers must implement the following actions:

?action=	parameters	POST data	response
store	<b>uuid=&lt;device uuid&gt;<sup>2</sup></b>	text formatted data <1kb	201 Created 400 Bad request (missing uuid) 413 Request Entity Too Large 503 Service Unavailable
storeBig	<b>uuid=&lt;device uuid&gt;<sup>2</sup></b>	binary data <1GB <sup>1</sup>	201 Created 400 Bad request (missing uuid) 413 Request Entity Too Large 503 Service Unavailable
retrieve	<b>uuid=&lt;device uuid&gt;<sup>2</sup></b> since=<epoch time latest>	- none -	200 ok 400 bad request (missing uuid) 503 Service Unavailable (db error) body: All data fields, separated by new lines ('\n').
<del>getCode<sup>3</sup></del>	<del><b>uuid=&lt;device uuid&gt;<sup>2</sup></b></del>	<del>=none=</del>	<del>200 ok 400 bad request missing/unknown uuid 503 Service unavailable (device error) the JS code for App.</del>
<del>getHTML<sup>3</sup></del>	<del><b>uuid=&lt;device uuid&gt;<sup>2</sup></b></del>	<del>=none=</del>	<del>same as get code but returns app HTML</del>
list	- none -	- none -	200 ok body: the device list (see below)
forward	<b>uuid=&lt;device uuid&gt;<sup>2</sup></b> <b>cmd=&lt;device command&gt;<sup>2</sup></b>	depends on cmd	forwards the request to the device with specified uuid

1: if stored as a mysql blob (recommended) the default max size is 16mb see max\_allowed\_packet in mysql docs

2: required

3: getCode and getHTML have been replaced with forward.

**store:** description here:

**retrieve:** gets the data associated with uuid. the since parameter may either be an epoch time value (integer) or the string 'latest', which indicates that only the most recent dataum should be retrieved .

**list:** fill in descr

**forward:** forwards the query to the device specified with uuid. Also passes on any other queries.

- 

### Device List format:

The device list is a javascript object with one property for every known device. Each property is named after the devices uuid and refers to an object that has *at least* the same fields as the device info object (ie uuid, status, state, name) but may contain more fields.

The manager must also serve the dashboard html (dash.html) on the same port and any other files required for the dash to run.

## Multicast listener

The manager must listen on multicast address 224.250.67.238 port 17768 for a UDP message announcing the existence of a new device. Once a device is discovered the manager will send a configuration "cmd" to the device on the http port specified

**Multicast packet:** The multicast packet sent by the device should contain only a text string containing formatted as shown below.

keystr (64chars)	port (5char)
obqQm3gtDFZdaYlENpIYiKzl+/qARDQRmiWbYhDW9wreM/APut73nnxCBJ8a7Pww	<dev port>

Where the key string must be reproduced exactly as above and is case sensitive. This is to avoid confusing in the extremely unlikely case that another protocol is using the same multicast address and port. And the port string is a zero padded port number IP address should be inferred from the packet header information.