

amati.linea HTTP requests

<http://IP/cgi-bin/audio/request.cgi?request=cmd&cmd=CMD¶m=PARAM>

cmd `string`

param JSON-encoded `null`, `number`, `string`, `array`, `object`

cmd `play-single`

param `[uri, clear]`

plays single track or remote playlist, setting clear to `true` will clear the current playlist

cmd `vol-up`, `vol-down`

param `null`

changes volume by 2% step

cmd `db`

param `null`

returns current database of albums, artists, genres and playlists

cmd `play-pause`

param `null`

toggles between play and pause state

cmd `players`

param `null`

returns a list of other players in the same network

cmd `status`

param `null`

returns various information on current player status

cmd `playlistdelete`

param `[playlist, filename]`

removes all occurrences of `filename` in `playlist`

cmd `source`

param `"sourcename" (int, auto, ext1, ext2, ext3, ext4)`

selects current source (if hardware supports it)

cmd `zone`

param `zonenumber (1..4)`

toggles selected zone number on/off (if hardware supports it)

cmd `save`

param `playlistname`

saves current playlist as `playlistname`

cmd `action`

param `{ action: action, type: type, arg: arg }`

action `play`, `replace`, `add`

type `album`, `artist`, `genre`, `filesystem`, `radio`, `playlist`, `current`

arg `name`

adds, replaces, adds and plays track(s) or playlists matching search `name` criteria into current playlist

action `remove`
type `current`
arg `trackid`

removes track with id `trackid` from current playlist

action `remove`
type `playlist`
arg `playlistname`

deletes stored playlist named `playlistname`

action `track`
type `playlist`
arg `playlistname`

returns info on tracks in playlist `playlistname`

action `track`
type `album`, `artist`, `genre`
arg `name`

returns info on tracks matching search `name` criteria

cmd `MPD command`

param `[arg1, arg2, ...]`

sends a command to MPD server, returns JSON-encoded result

Helper function for LogicMachine

You can use `amaticmd(ip, cmd, param)` function to access player functions from LogicMachine

```
function amaticmd(ip, cmd, param)
  local url, res, err

  require('json')
  require('socket.http')
  require('socket.url')

  socket.http.TIMEOUT = 5

  if param == nil then
    param = json.null
  end

  param = json.encode(param)

  cmd = socket.url.escape(cmd)
  param = socket.url.escape(param)

  url = string.format('http://%s/cgi-bin/audio/request.cgi?request=cmd&cmd=%s&param=%s', ip, cmd, param)
  res, err = socket.http.request(url)

  if res then
    return json.pdecode(res)
  else
    return nil, err
  end
end
```