

General Description

This document details the specifications and usage of the RESTful API for the iBoot-PDU family of power distribution units. It allows users to get outlet status and control outlets using HTTP and HTTPS.

The system relies on an authentication token to provide additional security.

Authorization Token

An authorization token is required prior to any further communication. The token is valid until there is 5 minutes of inactivity between the client and the iBoot-PDU. Each time a token is issued this 5 minute timeout will reset.

Acquire the token from the iBoot-PDU using

`https://<ipaddress>/services/auth/` or
`http://<ipaddress>/services/auth/` where <ipaddress> is the IP Address of the iBoot-PDU

The authorization request requires a username and password that is valid for the iBoot-PDU being accessed. The authorization request uses the following JSON structure:

```
{
  "username": "<username>",           //PDU Username
  "password": "<password>"          //PDU Password
}
```

Upon receiving an authorization request, the iBoot-PDU will issue a response with the following JSON structure:

```
{
  "success": "<true|false>",         //Result code true or false
  "message": "<message>",           //Error Message from the server. See appendix
  "token": "#####-#####-#####-#####" //64 bit randomly generated token
}
```

success returns true or false based on the successful receipt of the authorization request

message is an error message to assist in debugging when success is false

token is the 64 bit code to allow further communication with the iBoot-PDU

Example – using CURL Authorization Request:

```
curl --data '{"username": "Jeff", "password": "jeffpass"}' -X POST http://10.10.10.30/services/auth/
```

Authorization Response:

```
{"success": "true", "message": "", "token": "7594-235E-E439-69D3"}
```

Retrieve Status

Get the status of one or more outlets, or any predefined group of outlets using

`https://<ipaddress>/services/services/retrieve/` or
`http://<ipaddress>/services/services/retrieve/` where <ipaddress> is the IP Address of the iBoot-PDU

The JSON structure for Retrieve is

```
{
  "token": "####-####-####-####", //token received from authorization request
  "outlets": [<array>], //array of outlets to retrieve
  "groups": [<array>] //array of groups to retrieve
  "names": [<array>] //array of name types to retrieve
  "analog": [<array>] //array of analog sensor names to retrieve
}
```

token is a 64 bit value retrieved from `http(s)://<ipaddress>/services/auth/`

outlets is a JSON array of the outlets to be retrieved.

outlets accepts outlet numbers, outlet names, or a combination of both.

groups is a JSON array of the groups to be retrieved.

groups accepts one or more group names to retrieve.

names is a JSON array of the name types to retrieve.

names accepts one or both of the following: outlets, groups.

analog is a JSON array of the sensor names to retrieve.

analog accepts one or multiple of the following: LV1,LC1,LV2,LC2,T0,T1.

Note: T0 and T1 will display temperature only if external probe is connected.(F° or C°)

Note: LV2 and LC2 will display Voltage and Current only if a second line cord is present.

The API will return the following JSON structure upon receipt of a retrieve request

```
{
  "success": "[true|false]", //Result code
  "message": "[errormessage]" //Message from the server, if status false
  "outlets": {
    "<outletid1>": "[on|off]", //the status of the first requested outlet
    "<outletid2>": "[on|off]", //the status of the second requested outlet
    "<outletidn>": "[on|off]" //the status of the last requested outlet
  }
  "groups": {
    "<groupid1>": "[on|off|mixed]", //the status of the first requested group
    "<groupid2>": "[on|off|mixed]", //the status of the second requested group
    "<groupidn>": "[on|off|mixed]" //the status of the last requested group
  }
  "names": {
    "outletNames": {
      "1": "<outlet name>", //the name of the 1st outlet
      "2": "<outlet name>", //the name of the 2nd outlet
      "3": "<outlet name>", //the name of the 3rd outlet
      "n": "<outlet name>" //the name of the nth outlet
    }
    "groupNames": {
      "1": "<group name>", //the name of the 1st group
      "2": "<group name>", //the name of the 2nd group
      "3": "<group name>", //the name of the 3rd group
      "n": "<group name>", //the name of the nth group
    }
  }
  "analog": {
    "Line1Voltage": "115.50"
    "Line1Current": "1.50"
    "Line2Voltage": "115.50"
    "Line2Current": "0.00"
    "T0": "72.29"
    "T1-Probe": "101.25"
  }
}
```

success is true if the command was executed correctly.

success is false if the command could not be executed.

message is a string containing an error message when **success** is false.

outlets is a substructure containing JSON variable value pairs where the variable is the outlet name and the value is the outlet status.

groups is a substructure containing JSON variable value pairs where the variable is the group name and the value is the group status. (On, Off or Mixed)

outletNames is a JSON array of the outlet names.

groupNames is a JSON array of the group names.

analog is a substructure containing JSON variable value pairs where the variable is the one requested analog sensor and the value is appropriate value for that sensor.

LV1 – Line Voltage Cord 1 : **LC1** – Line Current Cord 1

LV2 – Line Voltage Cord 2 : **LC2** – Line Current Cord 2 (LV2 and LC2 will show error in no line cord 2)

T0 – Temperature for external sensor within the USB device

T1 – Temperature for external sensor probe on end of cable connected to USB device

Example – using CURL Retrieve:

```
curl-d '{"token":"7594-235E-E439-69D3","outlets":["1","2","5"],"groups":["Group1","Group2","group3"]}'-X POST http://10.10.10.30/services/retrieve/
```

Authorization Response:

```
{"success":"true","message":"","outlets":{"1":"Off","2":"Off","5":"On"},"groups":{"Group1":"Mixed","Group2":"Mixed","group3":"Mixed"}}
```

Control

Control any outlet or group using

<https://<ipaddress>/services/control/> or
<http://<ipaddress>/services/control/>

The JSON Structure for Control is

```
{
  "token": "####-####-####-####",           //token received form auth
  "control": "[outlet|group|sequence]",     //object to be controlled
  "command": "[on|off|cycle|run|stop]",    //command to be executed
  "outlets": [<array>],                   //array of outlets to control
  "sequence": "<sequencename>",           //name of the sequence
  "group": "<groupname>"                  //name of the group to be controlled
}
```

token is a 64 bit value retrieved from <https://pdu ip address/services/auth/>

control will accept outlet, group or sequence.

command will accept on, off, or cycle when control is outlet.

command will accept run or stop when control is sequence.

outlets will only be required when command is outlet.

outlets is ignored when command is sequence.

outlets is a JSON array of the outlets to be controlled.

outlets will accept outlet numbers, outlet names, or a combination of both.

sequence is only required when control is sequence.

sequence accepts sequence names only.

group is only required when control is group.

group only accepts group names.

Example 1 – using CURL Control:

```
curl -d '{"token":"CFB0-4354-3008-386C","control":"sequence","command":"run","sequence":"Jeff1"}' -X POST http://10.10.10.30/services/control/
```

Response:

```
{"success":"true","message":""}
```

Example 2 – using CURL Control:

```
curl -d '{"token":"C773-A44D-6A91-D08E","control":"group","command":"off","group":"Group2"}' -X POST http://10.10.10.30/services/control/
```

Response:

```
{"success":"true","message":""}
```

Example 3 – using CURL Control:

```
curl -d '{"token":"3638-A8FC-CE55-01AC","control":"outlet","command":"off","outlets":["0","1","5"]}' -X POST http://10.10.10.30/services/control/
```

Response:

```
{"success":"true","message":""}
```