

1. Overview

The iBoot-G2, iBoot-PoE and iBoot-PDU Family of products provides a simple RESTful API (URL Control) that allows users to control them easily from their own websites and portals. This, however, requires the knowledge of the IP Address of the iBoot(s) that need to be controlled. This is not always practical. Some internet connections do not provide static IP addresses making it impossible to use the RESTful API. The iBoot Cloud Services (IBCS) is a perfect solution to this problem.

iBoots make an outgoing connection to the iBCS for monitoring and control. Since the iBCS does have a static IP address as well as a domain name, this makes it the perfect solution to the aforementioned problem. This document provides a specification for the first stage of the iBCS RESTful API.

2. Specification

The iBCS implements a simple RESTful API that allows the user to post a simple JSON structure to the iBCS.

<https://iboot.co/services/control>

The posted data will contain the following JSON structure:

```
{
  "username": "",           //iBCS Account Username
  "password": "",         //iBCS Account Password
  "mac": "",              //MAC address of the iBoot
  "outlet": "",           //Output to control [0|1|2|3|4|5|6|7]
  "control": ""           //Action to take [cycle|on|off]
}
```

The iBCS will respond with a JSON structure:

```
{
  "success": "",          //Result code [true|false]
  "message": ""           //Message from the server
}
```

To retrieve outlet states, the following will be utilized:

<https://cloud.iboot.co/services/retrieve>

The posted data will contain the following JSON structure:

```
{
  "username": "",           //iBCS Account Username
  "password": "",         //iBCS Account Password
  "mac": ""               //MAC address of the iBoot
}
```

The iBCS will respond with a JSON structure for iBoot-G2/PoE:

```
{
  "success": "true",       //Result code [true|false]
  "message": "",          //Message from the server
  "status": {             //Status Information
    "Main": "ON",        //Main Outlet Status
    "AP-1": "Fail",     //Autoping-1 Status
    "AP-2": "OK"        //Autoping-2 Status
  },
  "triggerInfo": {       //Trigger Information
    "APT1": "0"         //Autoping Trigger Info
  }
}
```

To retrieve outlet states, the following will be utilized:

<https://cloud.iboot.co/services/retrieve>

The iBCS will respond with a JSON structure for iBoot-G2+/S:

```
{
  "success": "true",           //Result code [true/false]
  "message": "",             //Message from the server
  "status": {                //Status Information
    "Main": "ON",           //Main Outlet Status (Valid in expansion mode)
    "EXP-1": "OFF",         //EXP-1 Status (Valid in expansion mode)
    "EXP-2": "OFF",         //EXP-2 Status (Valid in expansion mode)
    "Main-2": "ON",         //Main-2 Outlet Status (Valid in Independent IO Mode)
    "Input-1": "Closed",    //Input-1 Status (Valid in Independent IO Mode)
    "Input-2": "Closed",    //Input-2 Status (Valid in Independent IO Mode)
    "Output-1": "Open",     //Output-1 Status
    "Output-2": "Open",     //Output-2 Status
    "AP-1A": "Inactive",    //Autoping-1A Status
    "AP-1B": "Inactive",    //Autoping-1B Status
    "AP-2": "Inactive",     //Autoping-2 Status
    "AP-3": "Inactive",     //Autoping-2 Status
    "HB": "OK"              //Heartbeat Status
  },
  "triggerInfo": {          //Trigger Information
    "APT1": "0",            //Autoping-1A & 1B Trigger Info
    "APT2": "0",            //Autoping-2 Trigger Info
    "APT3": "0",            //Autoping-3 Trigger Info
    "HBT1": "0"             //Heartbeat Trigger Info
  }
}
```

To retrieve outlet states, the following will be utilized:

<https://cloud.iboot.co/services/retrieve>

The iBCS will respond with a JSON structure for iBoot-PDU:

The response will display amount of outlets on the device.

```
{
  "message": "", //Message from the server
  "status": [ //Status Information
    {
      "Outlet-1": "On" //Outlet-1 Status
    },
    {
      "Outlet-2": "On" //Outlet-2 Status
    },
    {
      "Outlet-3": "On" //Outlet-3 Status
    },
    {
      "Outlet-4": "On" //Outlet-4 Status
    },
    {
      "Outlet-5": "On" //Outlet-5 Status
    },
    {
      "Outlet-6": "On" //Outlet-6 Status
    },
    {
      "Outlet-7": "On" //Outlet-7 Status
    },
    {
      "Outlet-8": "On" //Outlet-8 Status
    }
  ],
  "success": "true", //Result code [true/false]
  "triggerInfo": [] //Future Use
}
```

3. Examples

To control an iBoot-G2/PoE to Cycle via RestFul API:

```
curl -d '{"username":"myuser","password":"mypass","mac":"00:0d:ad:02:7d:d5","outlet":"0","control":"cycle"}' -X POST https://cloud.iboot.co/services/control
```

The command above will return a JSON structure as following:

```
{"success":"true","message":"Sent 'cycle' to 00-0d-ad-02-7d-d5 outlet 0"}
```

To query an iBoot-G2/PoE status via RestFul API:

```
curl -d '{"username":"myuser","password":"mypass","mac":"00:0d:ad:02:7d:d5"}' -X POST https://cloud.iboot.co/services/retrieve
```

The command above will return a JSON structure as following:

```
{"success":"true","message":"","status":{"Main":"ON","AP-1":"OK","AP-2":"OK"},"triggerInfo":{"APT1":"0"}}
```

To control an iBoot-G2+/S Main Outlet to Cycle via RestFul API:

```
curl -d '{"username":"myuser","password":"mypass","mac":"00-0d-ad-03-21-71","outlet":"0","control":"cycle"}' -X POST https://cloud.iboot.co/services/control
```

The command above will return a JSON structure as following:

```
{"success":"true","message":"Sent 'cycle' to 00-0d-ad-03-21-71 outlet 0"}
```

To query an iBoot-G2+/S status via RestFul API:

```
curl -d '{"username":"myuser","password":"mypass","mac":"00-0d-ad-03-21-71"}' -X POST https://dev.iboot.co/services/retrieve
```

The command above will return a JSON structure as following:

```
{"success":"true","message":"","status":{"Main":"ON","EXP-1":"ON","EXP-2":"ON","Main-2":"ON","Input-1":"Closed","Input-2":"Closed","Output-1":"Closed","Output-2":"Closed","AP-1A":"Inactive","AP-1B":"Inactive","AP-2":"Inactive","AP-3":"Inactive","HB":"OK"},"triggerInfo":{"APT1":"0","APT2":"0","APT3":"0","HBT1":"0"}}
```

To control an iBoot-PDU Outlet-1 to Cycle via RestFul API:

```
curl -d '{"username":"myuser","password":"mypass","mac":"68-9e-19-b3-fc-96","outlet":"0","control":"cycle"}' -X POST https://cloud.iboot.co/services/control
```

The command above will return a JSON structure as following:

```
{"success":"true","message":"Sent 'cycle' to 68-9e-19-b3-fc-96 outlet 0"}
```

To query an iBoot-PDU status via RestFul API:

```
curl -d '{"username":"myuser","password":"mypass","mac":"68-9e-19-b3-fc-96"}' -X POST https://dev.iboot.co/services/retrieve
```

The command above will return a JSON structure as following:

```
{"success":"true","message":"","status":{"Outlet-1":"On"},"triggerInfo":[]}
```