

# A Guide to the Smart by Bond Ceiling Fan API for Control System Programmers

Bond Home

19 Feb 2022

Only a small subset of the Bond Local API (<http://docs-local.appbond.com/>) is required for integration of smart fans into control systems. The following is a brief self-contained guide to discovery, control, and state feedback.

## Discovery

There are two discovery mechanisms. mDNS may be already supported by libraries on the control system, but is somewhat slow to resolve and overall quite complex. BPUP is a simple proprietary discovery protocol: it requires fewer system resources and resolves faster than mDNS.

### mDNS discovery

Smart by Bond fans advertise their HTTP API via mDNS service `_bond._tcp`.

To discover on MacOS, do:

```
dns-sd -Z _bond._tcp .
```

Or on Linux:

```
avahi-browse -a | grep bond
```

The mDNS reply contains a serial number, for example:

```
KSMJBCE72801
```

Note that mDNS discovery should be run with several retries to ensure that all Smart by Bond fans are discovered.

### BPUP discovery

An alternative discovery technique uses a simple Bond-proprietary protocol called BPUP. Here's a quick rundown:

## Check Device Type

---

Control system sends a broadcast UDP packet with destination port 30007 and data containing a single newline character. On a POSIX system, this can be done using:

```
echo | socat -t1 STDIO UDP4-DATAGRAM:255.255.255.255:30007,broadcast
```

All Smart by Bond units on the subnet will reply directly to the control system with a newline-terminated JSON object containing their serial number and firmware version. Here's example output from the above `socat` command:

```
{"B": "PPCTAXX88799", "d": 0, "v": "v3.0.6-beta"}  
{"B": "KSMJBCE72801", "d": 0, "v": "v3.0.9-alpha"}  
{"B": "BD25020", "d": 0, "v": "v2.29.2.1-beta"}  
{"B": "TWCTAXX88792", "d": 0, "v": "v3.0.9-alpha"}  
{"B": "TWCTBXX88806", "d": 0, "v": "v3.0.9-alpha"}
```

## Check Device Type

The device type may be determined by querying the API. However, it is far simpler to check the first letter of the serial number.

All Smart by Bond fans have serial numbers starting with K.

## Obtaining API Token

In order to access most API endpoints, a token string is required.

The token may be obtained from the Device Settings screen in the Bond Home app, or alternatively, it may be obtained automatically by instructing the user to hold the “Stop or Power” button on the fan remote control for 10 seconds while the control system queries the `token` endpoint:

```
curl -i http://192.168.86.241/v2/token
```

After 5 seconds of receiving the Power signal from the remote, the token will become “unlocked” and is sent in the reply:

```
{"locked": 0, "token": "2f6eaf536ae6779c", ...}
```

NOTE: The time required for holding the Power/Stop button is actually 5 seconds, but we instruct users to hold for 10 seconds to ensure they do not release too early.

After the control system has memorized the token, it is recommended that it relock the token endpoint:

```
curl -i http://192.168.86.241/v2/token -X PATCH -d '{"locked": 1}'
```

## Determining Device Configuration

To determine which features may be controlled on a Smart by Bond fan, the control system should query the `state` endpoint:

```
curl -iH "Bond-Token: 2f6eaf536ae6779c"
      http://192.168.86.241/v2/devices/1/state
```

A typical reply is:

```
{"power":0,
 "speed":4,
 "light":0,
 "brightness":100,
 "brightness_cycle_phase":-1,
 "timer":0,
 "breeze":[0,50,50],
 "direction":-1,
 "_":"cc897346",
 "__":"cc897346"}
```

A brief description of each important feature is as follows:

- **power**: on/off state of fan motor
- **speed**: fan motor speed (integral speed number)
- **light**: on/off state of fan light
- **brightness**: light brightness (percentage)
- **direction**: summer (1) or winter (-1) direction of fan motor

All fans will have **power** and **speed**. Some fans have no light. Others have non-dimmable lights, and so have **light** but no **brightness**. Most fans have dimmable lights, and so have both **light** and **brightness**. The **direction** feature is available on select AC fans and all DC fans.

To determine the maximum fan speed, the control system must query the **properties** endpoint and check the **max\_speed** variable:

```
curl -iH "Bond-Token: 2f6eaf536ae6779c"
      http://192.168.86.241/v2/devices/1/properties
```

```
{"max_speed":6,
 "feature_light":true,
 "feature_brightness":true,
 "breeze_max_speed":6,
 "breeze_min_speed":1,
 "breeze_period":30,
 "_":"3569eb3a",
 "__":"3569eb3a"}
```

## Fan and Light Control

The fan is controlled by sending PUT requests with a JSON object as body to the desired **actions** endpoint. Certain actions require an **argument** to be specified, otherwise the JSON object will be empty. Here are the essentials:

To turn the light off:

```
curl -iH "Bond-Token: 2f6eaf536ae6779c"  
  http://192.168.86.241/v2/devices/1/actions/TurnLightOff  
  -X PUT -d {}
```

To turn the light on, replace TurnLightOff with TurnLightOn.

To set brightness to 42%:

```
curl -iH "Bond-Token: 2f6eaf536ae6779c"  
  http://192.168.86.241/v2/devices/1/actions/SetBrightness  
  -X PUT -d '{"argument":42}'
```

To set the fan to Speed #5:

```
curl -iH "Bond-Token: 2f6eaf536ae6779c"  
  http://192.168.86.241/v2/devices/1/actions/SetSpeed  
  -X PUT -d '{"argument":5}'
```

To turn fan motor on (remembering previous speed), and to turn it off, use TurnOn and TurnOff actions respectively.

To set fan to winter direction:

```
curl -iH "Bond-Token: 2f6eaf536ae6779c"  
  http://192.168.86.241/v2/devices/1/actions/SetDirection  
  -X PUT -d '{"argument":-1}'
```

For summer direction, use an argument of 1 rather than -1.

When integrating with a toggle button UI that does not provide state feedback to the user, it is recommended to use the **ToggleLight** or **TogglePower** actions.

## State Feedback

The fan may be also operated via the Bond Home app or the fan's remote control. Therefore some state feedback mechanism to the control system is desirable.

The control system may simply poll the **state** endpoint.

Alternatively, the control system may obtain state feedback via the BPUP protocol. To do this, the control system need only continue to send a single broadcast datagram on port 30007 once every 60 seconds. The immediate replies from the fans can serve as an ongoing discoverability and reachability mechanism. Furthermore, whenever the state of the fan changes, the control system will receive a BPUP reply containing the new state data:

```
{"B": "ZZBL12345",  
  "d": 0,  
  "v": "v2.18.2",  
  "t": "devices/1/state",  
  "i": "00112233bbefeeff",
```

```
"s":200,  
"m":0,  
"f":255,  
"b":{"_":"ab9284ef","power":1,"speed":2}}\n
```

The control system should ensure that serial number (B) matches an integrated device, and furthermore that the topic `t` is `devices/1/state`. The state information is available in the body (b) object. The other key-value pairs may be safely ignored.

Here is a one-liner which will collect state feedback from all Smart by Bond devices on the network with a single command:

```
while true; do echo; sleep 60; done |  
  socat -t1 STDIO UDP4-DATAGRAM:255.255.255.255:30007,broadcast
```