

Thing Internet Protocol - “TIP”

Version 0.05

This protocol is designed to be a universal API for connected “smart” devices to the internet. This is built on HTTP RESTful APIs (defined in RFC 7231). All data is using plaintext. The protocol allows for device discovery, control, and data insight.

Concepts

Capitalization

In all instances, a lower camel case is used. Where the first letter of the first word is capitalized and the first letter of each subsequent word is also capitalized.

Devices

The Protocol allows endpoints to expose a list of devices. An unlimited number of devices are supported. These are meant as an organizational structure.

Types

Device features, such as states and commands, are represented and controlled using “types”. These are similar to data types in other programming languages. Each type supports a “readOnly” property that determines if the end application can change the value.

For each device, an unlimited number of types are supported, using a subtype.

Seventeen (17) types are supported, along with their commands/attributes.

type	Supported Commands/Attributes
switch	<ul style="list-style-type: none">• On• Off
level	<ul style="list-style-type: none">• Integer between 0 and 100

presence	<ul style="list-style-type: none"> • Present • Not present
alarmSensor	<ul style="list-style-type: none"> • None • Detected
Contact	<ul style="list-style-type: none"> • Open • Closed
Lock	<ul style="list-style-type: none"> • Lock • Unlock
thermostatMode	[heat, cool, auto]
temperature	Integer 0 to 500 <ul style="list-style-type: none"> • maxTemp • minTemp
fanMode	[auto, on, off]
rgbColor	Hexadecimal color value 000000-FFFFFF
temperatureUnit	["C", "F"]
button	<ul style="list-style-type: none"> • press • doublePress • hold
date	<ul style="list-style-type: none"> • Year (4 digit) • Month (1-12) • Day (1-31)
time	<ul style="list-style-type: none"> • Hour • Minute • Second • timeZoneName • timeZoneGMTOffset
powerUsage	<ul style="list-style-type: none"> • volts • watts
powerType	[battery, mains]
batteryLevel	Integer between 0 and 100
textData	String of data.
colorTemperature	2700-6500

Ports

It is advisable, although not required to run the server on HTTP port 2021.

Device Authentication

Thing Internet Protocol is designed to be an easy technology with moderate use of security. The Protocol used a 25 alphanumeric pre-shared key (“password”). This consists of the 26 letters of the alphabet in both capital and lowercase, as well as 0-9. [A-Za-z0-9]. The password is generated by the endpoint that exposes devices. The endpoint can either return an authorization key or the lowercase phrase “unauthorized”. It is requested to only allow a pairing code, which will allow control and reading devices. It is up to the device integrator to determine if an application will only have one authentication key or scoped permissions with multiple keys.

Device IDs

Each individual device that is part of integration has a unique id. Any values, with the exception of “names” may be used. Either an integer, mac address or IP address are common values.

Subtypes

If there are multiple types under a particular device, a “subtype” can be used to reference, via index, the individual types. Every type has one, a default value of 0.

Parameters

All parameters are passed through the URL, for example, to revoke an authorization key to the server, the URL GET command will be passed:

`http://{serverurl}/revokeAuth?authKey={AuthKey}`

Where {AuthKey} is the authorization key.

API Reference

Application

Endpoint: /applicationName

Method: GET

Return Type: text/plain

Returns: Friendly Name of Application

Endpoint: /applicationVersion

Method: GET

Return Type: text/plain

Returns: Version number of the Application

Authorization

Endpoint: /generateAuth

Method: GET

Return Type: text/plain

Returns: 25 character alphanumeric pairing code, or “unauthorized” if the server is not in a state to hand out authorization tokens.

Endpoint: /revokeAuth

Parameters: authKey, the 25 digit authorization key

Method: GET

Return Type: text/plain

Returns: “accepted” when the key has been removed, or “rejected” if an error has occurred.

Devices

Endpoint: /devices

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Returns: Semicolon separated list of device ids

Endpoint: /devices/names

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Returns: Semicolon separated list of device names

Endpoint: /devices/names
Method: GET
Parameters: authKey, the 25 digit authorization key
Return Type: text/plain
Returns: Semicolon separated list of device names

Device ID Commands

Endpoint: /devices/{id}
Method: GET
Parameters: authKey, the 25 digit authorization key
Return Type: text/plain
Returns: Semicolon separated list of types supported by devices

Subtype Commands

Endpoint: /devices/{id}/{type]
Method: GET
Parameters: authKey, the 25 digit authorization key
Return Type: text/plain
Returns: Semicolon separated list of subtypes that are supported. Default is "0".

Endpoint: /devices/{id}/{type}/{subtype}
Method: GET
Parameters: authKey, the 25 digit authorization key
Return Type: text/plain
Returns: current state of specified subtype.

Endpoint: /devices/{id}/{type}/{subtype}/readOnly
Method: GET
Parameters: authKey, the 25 digit authorization key
Return Type: text/plain
Returns: if subtype cannot be modified, "true" or "false"

Device Commands

switch

Endpoint: /devices/{id}/switch/{subtype}/on

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Turns switch value to “on”

Returns: New state of the device

Endpoint: /devices/{id}/switch/{subtype}/off

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Turns switch value to “off”

Returns: New state of the device

presence

Endpoint: /devices/{id}/switch/{subtype}/present

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets presence to present.

Returns: New state of the device

Endpoint: /devices/{id}/switch/{subtype}/notPresent

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets presence to not present.

Returns: New state of the device

alarmSensor

Endpoint: /devices/{id}/alarmSensor/{subtype}/none

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets alarmSensor to none detected.

Returns: New state of the device

Endpoint: /devices/{id}/alarmSensor/{subtype}/detected

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets alarmSensor to detected.

Returns: New state of the device

contact

Endpoint: /devices/{id}/contact/{subtype}/open

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets contact to open.

Returns: New state of the device

Endpoint: /devices/{id}/contact/{subtype}/closed

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets contact to closed.

Returns: New state of the device

lock

Endpoint: /devices/{id}/lock/{subtype}/lock

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets lock to locked.

Returns: New state of the device

Endpoint: /devices/{id}/unlock/{subtype}/unlock

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets contact lock to unlocked.

Returns: New state of the device

level

Endpoint: /devices/{id}/level/{subtype}/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets level to {newvalue}, 0-100 integer

Returns: New state of the device

temperature

Endpoint: /devices/{id}/temperature/{subtype}/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets temperature to {newvalue}

Returns: New state of the device

batteryLevel

Endpoint: /devices/{id}/batteryLevel/{subtype}/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets battery to {newvalue}

Returns: New state of the device

colorTemperature

Endpoint: /devices/{id}/colorTemperature/{subtype}/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets colorTemperature to {newvalue}, from 2700-6500

Returns: New state of the device

RGBColor

Endpoint: /devices/{id}/RGBColor/{subtype}/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets RGBColor to {newvalue}, from 000000-FFFFFF (Hexadecimal)

Returns: New state of the device

thermostatMode

Endpoint: /devices/{id}/thermostatMode/{subtype}/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets thermostatMode to {newvalue}, either "heat", "cool" or "auto"

Returns: New state of the device

powerType

Endpoint: /devices/{id}/powerType/{subtype}/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets powerType to {newvalue}, either "mains" or "battery"

Returns: New state of the device

button

Endpoint: /devices/{id}/button/{subtype}/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets button to {newvalue}, either "press", "doublePress"

Returns: New state of the device

date - Year

Endpoint: /devices/{id}/date/{subtype}/year/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets date's year to {newvalue}

Returns: New state of the device

date - Month

Endpoint: /devices/{id}/date/{subtype}/month/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets date's month to {newvalue}, 1-12

Returns: New state of the device

date - Day

Endpoint: /devices/{id}/date/{subtype}/day/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets date's day to {newvalue}, 1-31

Returns: New state of the device

time - Hour

Endpoint: /devices/{id}/time/{subtype}/hour/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets time's hour to {newvalue}, 0-23

Returns: New state of the device

time - Minute

Endpoint: /devices/{id}/time/{subtype}/minute/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets time's second to {newvalue}, 0-59

Returns: New state of the device

time - Second

Endpoint: /devices/{id}/time/{subtype}/second/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets time's second to {newvalue}, 0-59

Returns: New state of the device

time - timeZoneName

Endpoint: /devices/{id}/time/{subtype}/timeZoneName/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets time zone to {newvalue}, from tz database names, for example America/Chicago

Returns: New state of the device

time - timeZoneGMTOffset

Endpoint: /devices/{id}/time/{subtype}/timeZoneGMTOffset/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets time zone GMT Offset to {newvalue}, -12 to 14

Returns: New state of the device

textData

Endpoint: /devices/{id}/textData/{subtype}/{newvalue}

Method: GET

Parameters: authKey, the 25 digit authorization key

Return Type: text/plain

Function: Sets textData to {newValue}

Returns: New state of the device