

## Ketra API 1.0 Overview

March 19, 2018

### Introduction

The Ketra N4 Hub is a bridge device that exposes a JSON-based REST API that can be used to interact with the Ketra ecosystem of lighting and control products. The 1.0 version of the N4 API exposes basic control functionality. All system configuration is performed using Ketra's Design Studio desktop software.

This document provides a high-level overview of the API functionality and setup. The API documentation can be found [here](#)<sup>1</sup> and example code can be found [here](#)<sup>2</sup>.

### API Functionality

Version 1.0 of the Ketra API exposes the following functionality:

- Ability to enumerate and control Ketra X1 and X2 keypads, including
  - Query the button names and activation states
  - Activate or deactivate any keypad button, mimicking a button push on a physical keypad to activate or deactivate a scene
  - Query and set the brightness level for the activated scene
  - *Note: these functions simulate a user physically performing these actions at a keypad*
  - *Note: only one activate, deactivate, or set brightness command may be sent every 1.5 seconds*
- Ability to enumerate groups of lights and control them one group at a time
  - Ability to control brightness, color (either in CCT or xy), vibrancy, and/or fade time
  - *Note: only one group control command may be sent every 1.5 seconds*
- Support for bidirectional feedback through a websocket connection. Please see [here](#)<sup>3</sup> for more details.

### Hardware and Setup

- 1) Ketra wireless network, groups and scenes must first be setup in Ketra's Design Studio software
- 2) API is RESTful interface with JSON data format
- 3) Commands are sent as TCP/IP over ethernet to N4 Hubs

---

<sup>1</sup> <https://s3.amazonaws.com/ketra-software/KetraMobileAPI/v1/dist/index.html>

<sup>2</sup> <https://s3.amazonaws.com/ketra-software/KetraMobileAPI/v1/KetraN4APIExample.py>

<sup>3</sup> <https://s3.amazonaws.com/ketra-software/KetraMobileAPI/v1/KetraN4WebsocketProtocol.pdf>

# K E T R A

---

- 4) The N4 Hub will require an internet connection during setup so that the database can synchronize and user rights can be established for security purposes. A live internet connection thereafter is not required but is highly recommended.
- 5) Remote API access is available as of hub firmware version 1.14 and Design Studio 1.8. This can be enabled and used as follows:
  - a) After updating to Design Studio 1.8 and updating your hubs to the latest firmware using Ketra Tech Tool, you can enable remote access on a per-installation basis by opening the installation in Design Studio and selecting “enable remote access” by opening the menu using the button at the upper left.
  - b) Find the installation id for your installation by logging into <https://my.goketra.com> and finding your installation in the list and going to the “Details” page for your installation. The installation id is displayed in the URL of this page, for example a URL of <https://my.goketra.com/installations/0fbcada7-b318-4d29-858c-1ea3ac1fd5cb> would indicate an installation id of “0fbcada7-b318-4d29-858c-1ea3ac1fd5cb”
  - c) The N4 API is accessible from the base URL [https://api.goketra.com/\[inst-id\]/\[hub-serial\]/webAPI](https://api.goketra.com/[inst-id]/[hub-serial]/webAPI) where [inst-id] is your installation id and [hub-serial] is your hub’s serial number. For example: <https://api.goketra.com/0fbcada7-b318-4d29-858c-1ea3ac1fd5cb/KP00001485/webAPI/api/v1/keypads> would fetch the keypads for installation 0fbcada7-b318-4d29-858c-1ea3ac1fd5cb and hub serial KP00001485.
  - d) Note that with remote access you must send credentials as a basic authentication header and not through URL query parameters.

## Encryption

The N4 API uses SSL encryption to establish a secure communication channel between the client and the N4. Each N4 device contains a unique SSL certificate signed by a Ketra self-signed certificate authority. The public certificate of the CA is available [here](#)<sup>4</sup>.

## Authentication

The N4 API uses HTTP basic authentication to ensure that the client has permissions to access the device. The N4 uses an OAuth-style mechanism to validate the credentials passed through the API. API clients must obtain a valid OAuth token from the Ketra server using the Ketra account credentials that are used in Design Studio software. A document detailing how to authenticate be found [here](#)<sup>5</sup>. Please contact Ketra to get a Client ID and Client Secret needed for authentication.

---

<sup>4</sup> <https://s3.amazonaws.com/ketra-software/KetraMobileAPI/ketra-ca.pem>

<sup>5</sup> <https://s3.amazonaws.com/ketra-software/KetraMobileAPI/v1/KetraOAuthExample.pdf>

# K E T R A

---

## Response Envelope

Every response from the N4 is returned in a JSON object with the following schema:

```
{
  "Success": (true or false)
  "Error" : error code, or N4Status_NoStatus if successful
  "Content" : (requested content)
}
```

## LAN Discovery

The N4 hubs support three different mechanisms for discovery on a LAN:

- An HTTP GET request to <https://my.goketra.com/api/n4/v1/query> will return JSON data containing the IP addresses of the hubs on your LAN. Note that the N4 hubs must have internet access in order for this to work, and their public IP address must be the same as the client that makes the GET request to <https://my.goketra.com/api/n4/v1/query>
- If your client is on the same LAN subnet as the N4 hubs, you can send a UDP broadcast packet containing the character '\*' (hex 0x2A) and the hubs will respond and indicate their serial number and other information. This technique is illustrated in the example code.
- The N4 hubs also support the SSDP/UPnP discovery protocol. The hubs will respond to the "M-SEARCH \* HTTP/1.1" message sent over UDP to multicast group 239.255.255.250, port 1900. See section 1.3 of <http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf> for implementation details. This method works across LAN subnets if your LAN is configured to route multicast traffic across subnets.

## New and Improved Behaviors with Firmware Version 1.15 and Design Studio 2.0

Version 1.15 of the N4 hub firmware introduces new behaviors that are enabled when using Design Studio 2.0 and all devices in your installation have firmware version 1.15 or later. This will be indicated through the SupportsZoneKeypads property on the Hub (see here).

### Group Tracking and Hierarchy

Previously, the N4 could not keep track of the state of the lamp groups inside an installation if the groups were modified by a keypad button or show. With firmware 1.15, if the SupportsZoneKeypads property is true, the N4 hub will keep track of the state of each lamp group and return the current state regardless of how the state was modified. In addition, the N4 hub will return the group hierarchy (group parent/child relationships) as defined in Design Studio 2.0. The websocket API also contains a new event type that indicates when a group state is changed.

# K E T R A

---

## **New Group State Properties**

The following group state properties are new in firmware version 1.15.

- **MasterBrightness and NetBrightness:** previously, only a single brightness channel ("brightness") was provided. However, scenes are usually defined for a set of groups, and the brightness level can be different for each group. It's therefore convenient to have a master brightness channel that can be applied to an entire group tree to adjust the overall brightness of the scene. The MasterBrightness property can vary the brightness independently of the Brightness property, and the resulting level is the product of the two levels, reflected in NetBrightness.
- **DimCurve and DimCurveActive:** In Design Studio, one of the predefined content types is a Dimming Curve, which is a special type of content intended to emulate the dim-to-warm effect of an incandescent light. If a scene containing dimming curve content is activated on a group, the DimCurveActive property will be true and the DimCurve property will indicate which dim curve type is active.
- **ActiveShows:** This property will indicate which show groups are active for a particular group. Although this is an array property, only one show can normally be active at a time.
- **TransitionComplete:** This property will return false if the lamp group is currently in the middle of a transition (fade) from one state to another. The property will return true if the transition is complete.
- **UpdatedAt:** This property will return the time at which the group was last updated.

## **Scene Enumeration and Activation**

The N4 hub with firmware 1.15 can now return the list of scenes defined in Design Studio 2.0. A scene can be activated through the API, and optionally a group id or level can be specified in order to activate the scene on a particular group or at a specific brightness level.