Alexa Helper

A SmartThings Suite of SmartApps for controlling your home with your voice



Written by: Michael Struck

Documentation Version 1.0.0

Contributions:

Code: Tim Slagle, Alex Malikov

Documentation:

Licensed under the Apache License, Version 2.0 (the "License"); you may not use Alexa Helper code except in compliance with the License. You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

All product and company names are trademarks™ or registered® trademarks of their respective holders. Use of them does not imply any affiliation with or endorsement by them.

**What is Alexa Helper:**

The suite of Alexa Helper SmartApps are add-ins to the SmartThings mobile application that allow for additional control over your SmartThings devices using the Amazon Echo. The Amazon Echo (“Alexa as sometimes referred to) already has control over switches and dimmers within the SmartThings environment; however, it cannot natively control other SmartThings devices such as locks, speakers or thermostats. In addition, there is no way to natively trigger SmartThings modes, routines or security states. With the Alexa Helper Suite, almost all items within a SmartThings home can be controlled, along with the ability to control other locations, a function not even the SmartThings infrastructure allows easily.

**Definitions:**

**SmartApp-**Add-in to the SmartThings mobile application that allows for functionality outside of what is provided by default. These SmartApps are not stand alone, but only operate within the SmartThings mobile application. SmartApps are open source and written in a form of the Java programming language called “groovy”.

**Parent/Child SmartApp**-A code structure within the SmartThings development environment where a single instance of a parent SmartApp can have multiple child SmartApps associated with it. Alexa Helper uses this structure; there can only be one instance of the Alexa Helper Parent SmartApp. However, each scenario created within the parent SmartApp is actually an instance of the child SmartApp. This structure cuts down on repeated code within a single SmartApp.

**Virtual Switch**-A device within the SmartThings environment that appears to be a regular switch. However, the device is ‘virtual’ and not tied to a physical device.

**Momentary Tile**-A type of virtual switch that behaves like a spring loaded physical switch. In other words, it is in a normally closed (off) state. When pressed, it comes on momentarily and the turns itself off.

**Virtual Dimmer**-Another type of virtual switch that acts as a dimmer. You can not only turn this virtual device on or off but you can also set different levels (0 to 100) of ‘brightness’. Again, this is just a virtual device that is not tied to any physical device in the SmartThings environment.

**IDE**-Abbreviation for “Integrated Development Environment”. In the context of the SmartThings environment, this is the web site located at <http://ide.smartthings.com> that allows for advanced functionality within the SmartThings environment. This allows you to set up custom devices and applications that appear within your SmartThings mobile application.

**GitHub**-A web-based repository that allows for distribution revision control and source code management functionality.

**OAuth**-Provides client applications a secure method for accessing resources on behalf of the resource owner. Within the context of SmartThings, certain custom applications that allow for remote access (such as remotely turning on or off a light) will require additional authentication. As the owner of the resource, you can authorize remote access to an application by enabling OAuth.

**Access Token**-A security identity that is linked to OAuth and in the context of SmartThings, allows for remote control of a locations devices.

**URL**-Uniform Resource Locator, or basically the address of a web site or online resource, such as <http://www.smartthings.com>

**Conventions:**

The following conventions are used throughout this manual.

**Web sites/URLs** - <http://www.smartthings.com>

**Website links**-“Save”, “Publish”

**Spoken Commands** - *<”Alexa, turn on the front porch”>*

**Installation:**

While there are a total of four components of Alexa Helper, only the Parent/Child(scenario) apps are required. If you are wanting to add a virtual dimmer for speaker or thermostat control, you will also install the virtual dimmer code. Finally, if you desire to generate URLs for use in Alexa Helper, you will also need the Cloud Interface code as well.

*Alexa Helper-Parent*

The code for the main parent app is found of the GitHub site:

<https://github.com/MichaelStruck/SmartThings/blob/master/Other-SmartApps/AlexaHelper/Alexa%20Helper.groovy>

While on this site, find the “Raw” button and click it. This will bring up a blank page with just the code present. Select all of the code (typically Ctrl+A) and copy It (Typically Ctrl+C).

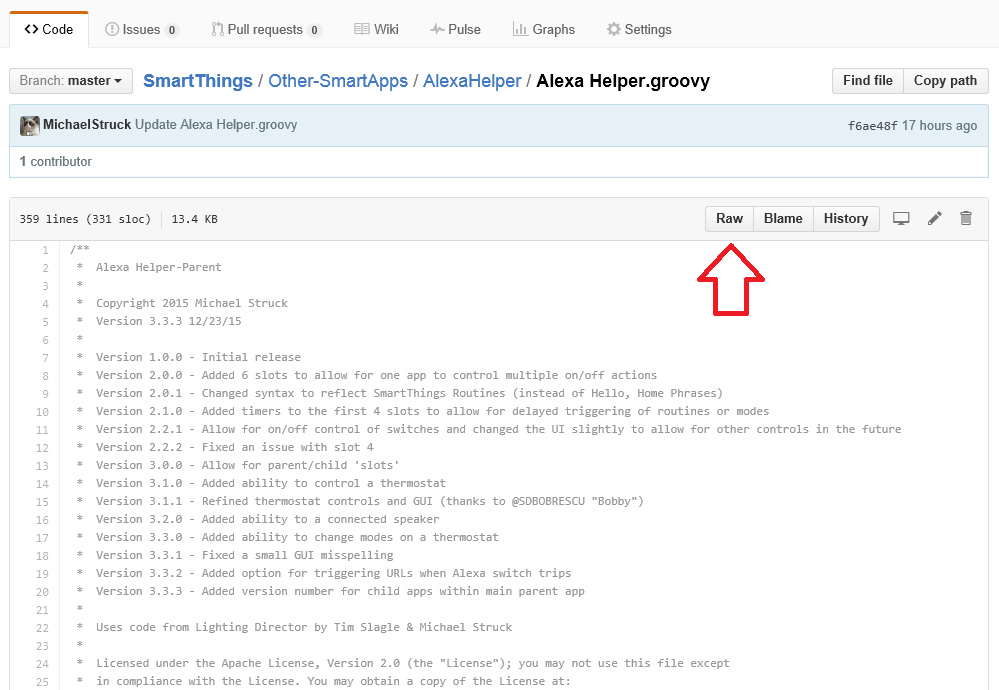


Figure - GitHub "Raw" Button

Next, point your browser to you SmartThings IDE (<http://ide.smartthings.com>) and log in.

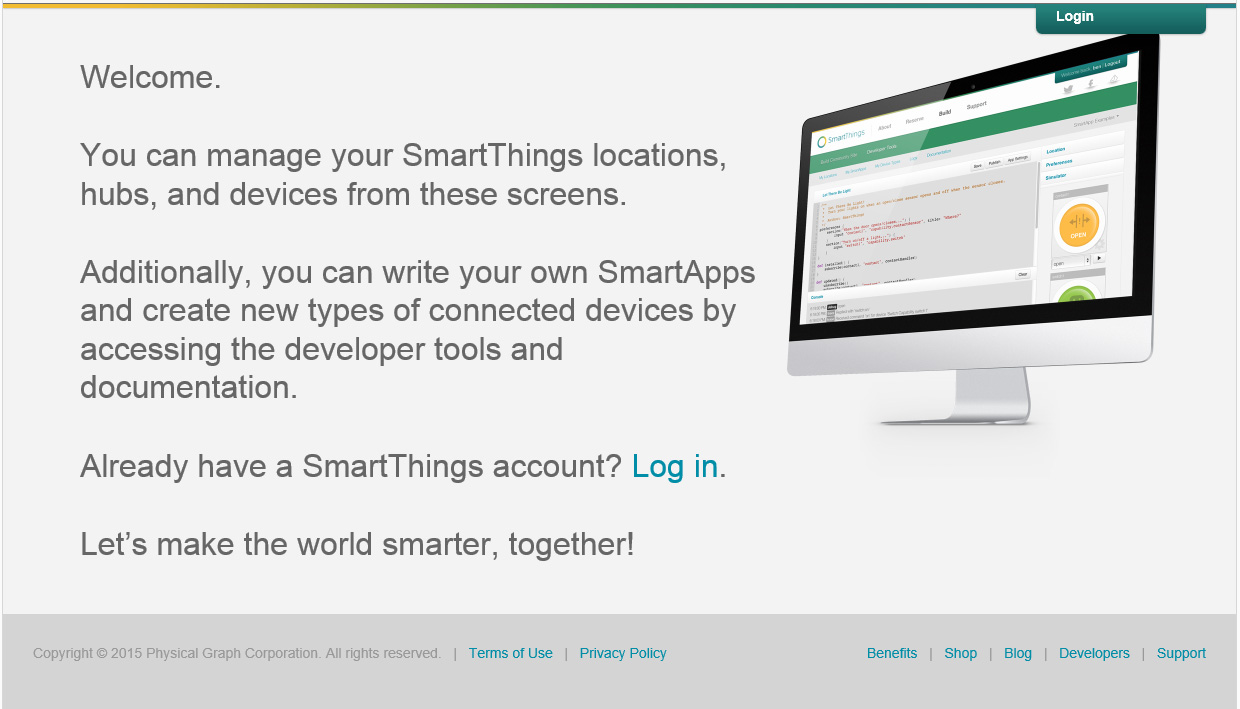


Figure - SmartThings IDE Login

Once you are logged in, find the “My SmartApps” link on the top of the page.

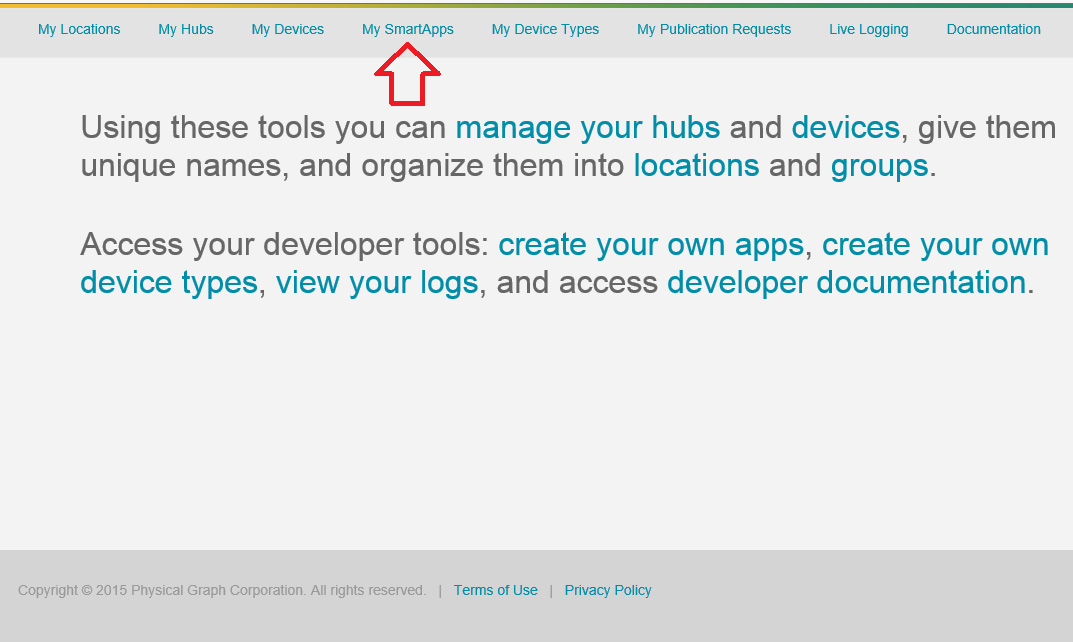


Figure - SmartThings IDE My SmartApps

Clicking this will allow you to produce a new SmartApp. Find the button called “+New SmartApp” and click it. Since you already have the code in your computer’s clipboard, find the tab along the top section called “From Code”. In the area provided, paste (typically Ctrl+V) the code you copied from GitHub. Click “Create” in the bottom left portion of the page. This will bring up another page, with the code now formatted within the IDE. If the code was copied correctly, there are no other steps except to save and publish the code. In the upper right corner of the page, find “Save”. Clicking it should give you confirmation the code was saved. Now, click publish, and you should receive a confirmation of the publishing the code.

Next, you will need to load up the child (scenario app). This code is located here:

<https://github.com/MichaelStruck/SmartThings/blob/master/Other-SmartApps/AlexaHelper/Alexa%20Helper-Scenario.groovy>

The steps for installation are almost exactly the same as above:

* Copy the raw code from the GitHub link above
* Go to the SmartThings IDE page
* Click the “My SmartApps” link
* Click the “+New SmartApp” link
* Click the “From Code” tab
* Paste the GitHub code in the open area
* Click “Create” from the bottom left corner
* Save the code using the button in the upper right hand corner of the page
* DO NOT publish this code. The child apps do not need to be published as they should only be accessed via the main (parent) app.

If you are an advanced user, you may also want to control two different SmartThings locations using one Echo. This is typically not possible within any SmartThings SmartApp; most apps only control the devices in one location. However, Alexa Helper can use a URL to send an “HTTP GET” to another SmartThings hub you set up, allowing control of multiple locations from a centralized Echo.

To accomplish this, you will need to install the Alexa Helper-Cloud Interface code in the remote location you want to control. The code for this SmartApp is located here:

<https://github.com/MichaelStruck/SmartThings/blob/master/Other-SmartApps/AlexaHelper/CloudInterface.groovy>

The steps for installation are similar to the Alexa App Parent Application:

* Copy the raw code from the GitHub link above
* Go to the SmartThings IDE page
* Click the “My SmartApps” link
* Click the “+New SmartApp” link
* Click the “From Code” tab
* Paste the GitHub code in the open area
* Click “Create” from the bottom left corner
* Save the code using the button in the upper right hand corner of the pag
* Publish the code using the button in the upper right hand corner of the page

Before leaving this page, you will need to ensure you enable OAuth for this particular SmartApp. This OAuth allows for secure remote access of the SmartThings devices. To enable OAuth, first find the “App Settings” button in the upper right corner of the page.

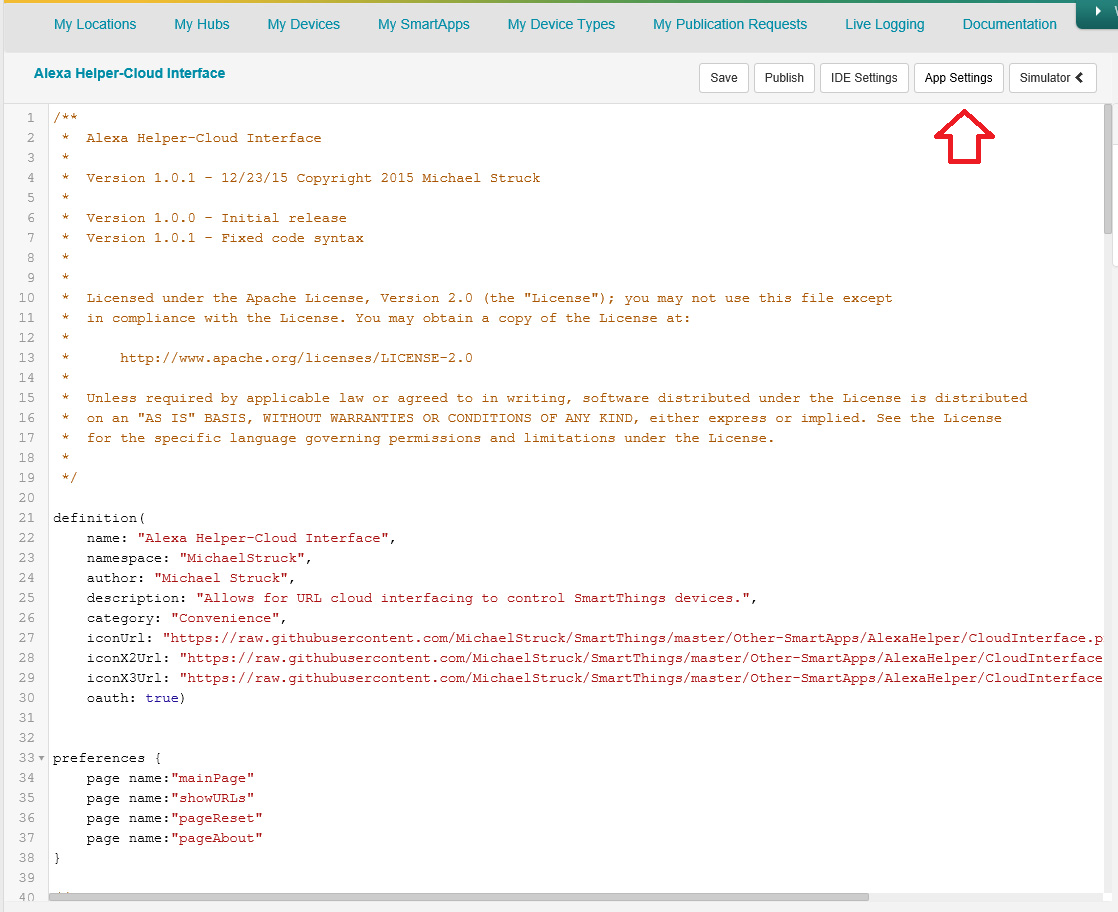


Figure 4-SmartThings IDE App Settings

From here, find the OAuth section toward the bottom of the page.

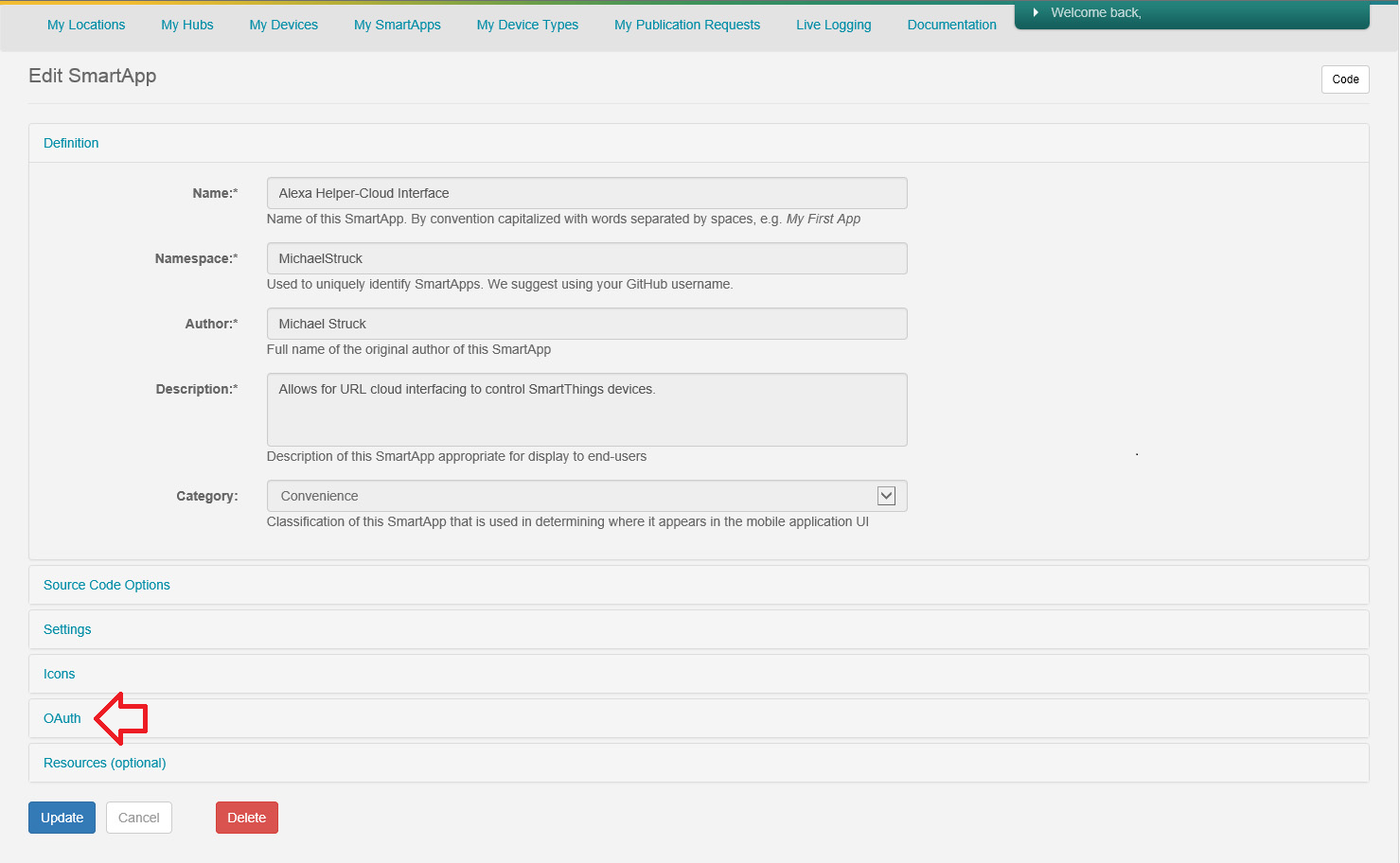


Figure 5- SmartThings IDE OAuth Settings

Clicking the “OAuth” link will reveal a button labled “Enable OAuth in Smart App”. Click this button. The screen will change, giving you a unique code for your Client ID and Client Secret. These are the foundations of the security of your app and should be kept secret. You will not need to memorize or write these down or add any other information to this page; OAuth simply needs to be enabled for Alexa Helper-Cloud Interface to generate your unique URLs.

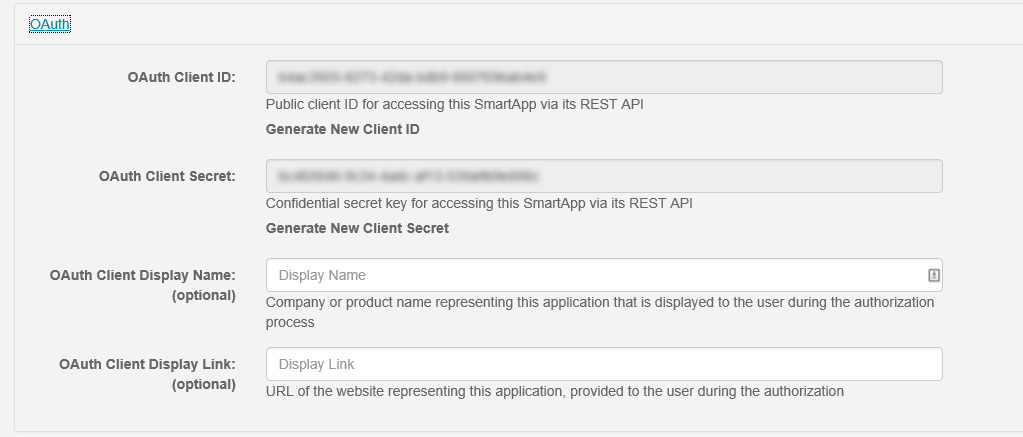


Figure 6- SmartThings IDE-OAuth Settings

The final step is to go back to your code using the button in the top left of the page and Save, then publish the SmartApp again.

**Set up of your Amazon Echo:**

The setup of your Amazon Echo within the SmartThings environment is outside the scope of this document. To use Alexa Helper, it is assumed you have done the following:

* Configured your Echo on your home’s Wifi and that it is functional
* Configured the Echo to work with your local SmartThings devices, and that you can turn on and off the switches associated with your account.

If either of the conditions above are not true, Alexa Helper will not function in your environment. Please refer to your Amazon Echo documentation or visit <http://alexa.amazon.com> to set up your device.

Once you have a functioning Echo, and you have confirmed it can control your SmartThings devices, it is recommended to set up some virtual tiles and switches to be used by Alexa Helper.

**Virtual Switch/Momentary Tile Setup:**

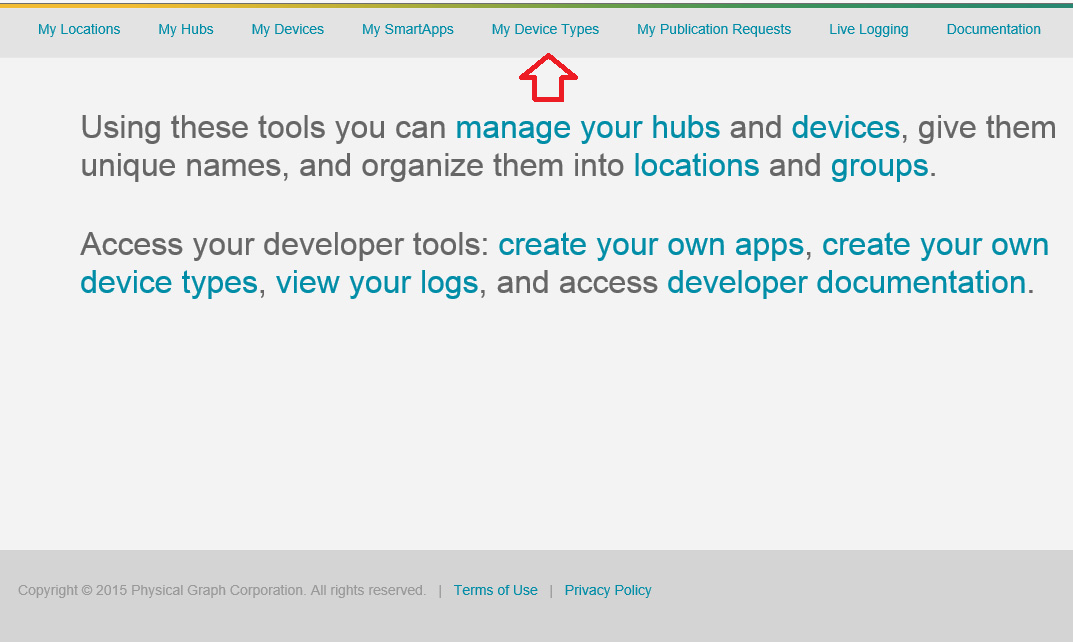
****

Figure 7-SmartThings IDE My Device Types

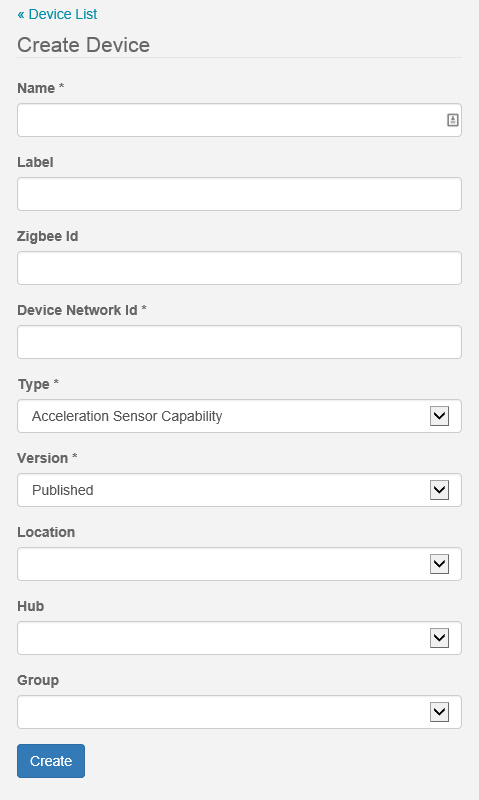


Figure 8-SmartThings IDE Create New Device

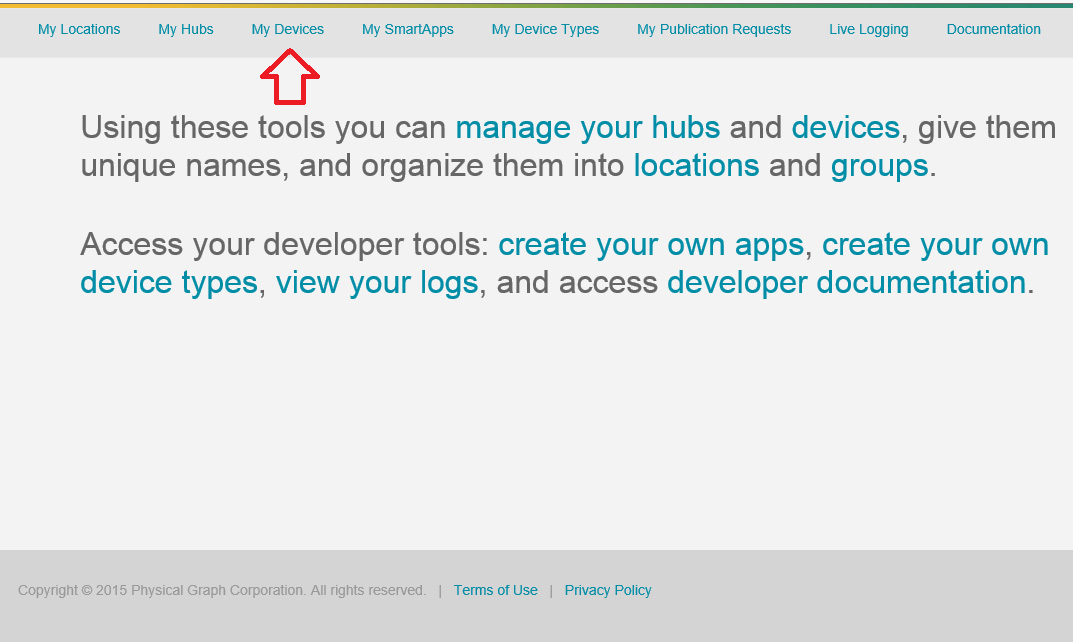


Figure 9- SmartThings IDE My Devices

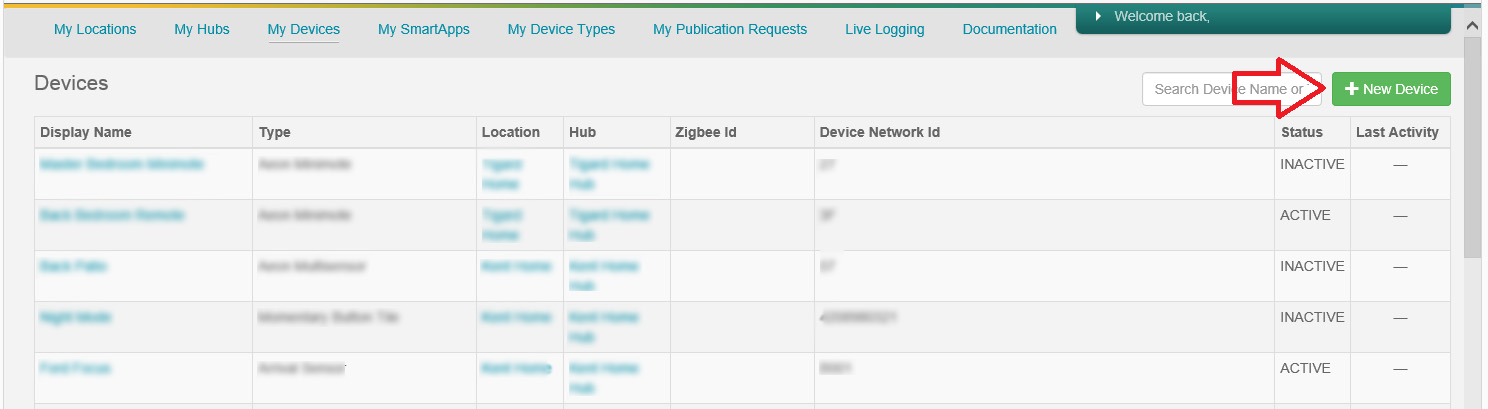


Figure 10- SmartThings IDE New Device

**Virtual Dimmer Installation:**

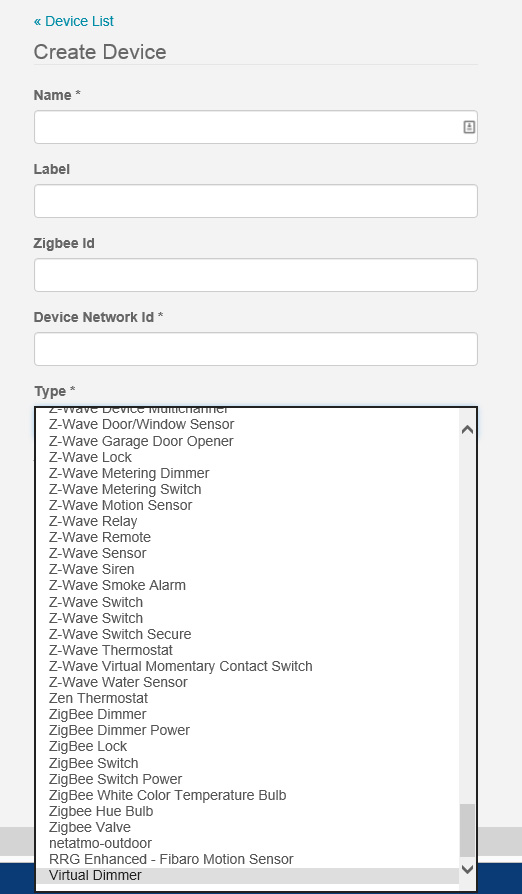
****

Figure 11- SmartThings New Device-Virtual Dimmer

**Alexa Helper Usage:**

**Scenario Setup:**

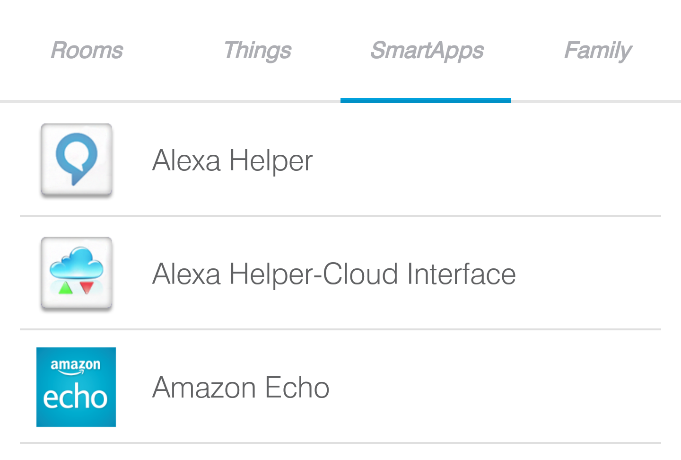


Figure 12- SmartThing SmartApps

|  |  |
| --- | --- |
| **Scenario Name**  **Alexa Switch**  **Is this a momentary switch?**  **When Switch is on:**  Perform this routine  Change to this mode  Change Smart Home Monitor to  Turn on these switches  Run this HTTP request  Delay in minutes  **Restrictions** | C:\Users\Michael\Desktop\ClipArt-AlexaHelper\FullInterface-Scenario.jpg |

**Thermostat Control:**

**Speaker Control:**

**Cloud Interface Setup:**

|  |  |
| --- | --- |
| **C:\Users\Michael\Desktop\ClipArt-AlexaHelper\Screenshot_20151224-120724.png**  Figure 13-Alexa Helper-Cloud Interface | C:\Users\Michael\Desktop\ClipArt-AlexaHelper\Screenshot_20151224-120822.png  Figure 14- Cloud Interface-Show URLs |

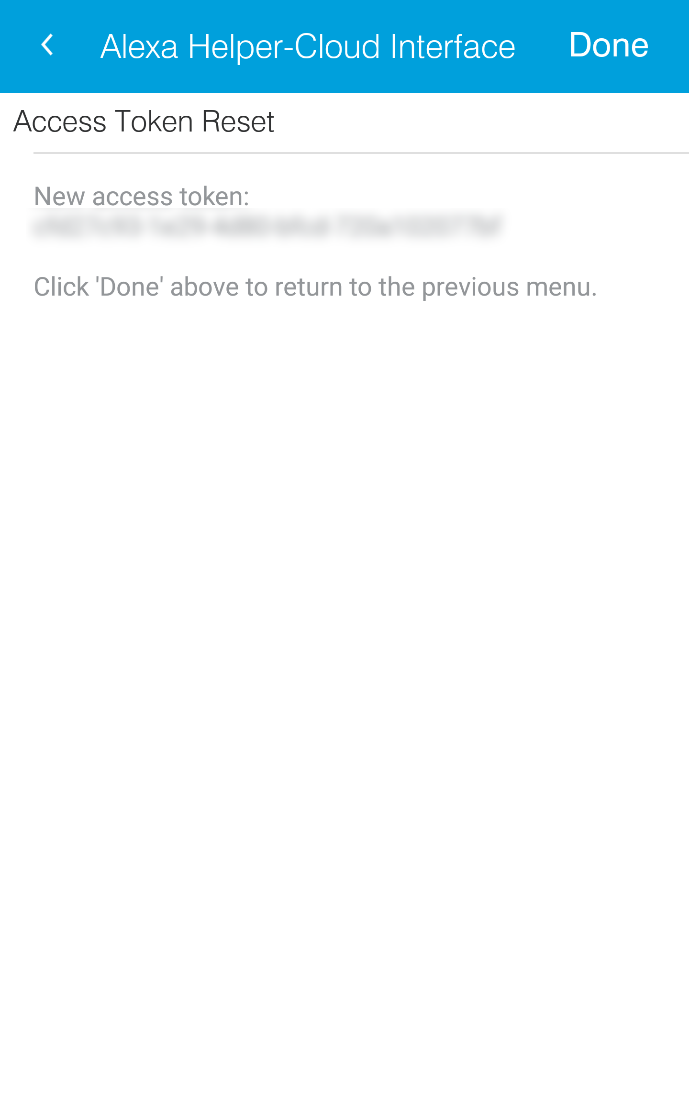
**

Figure 15- Cloud Interface-Access Token Reset

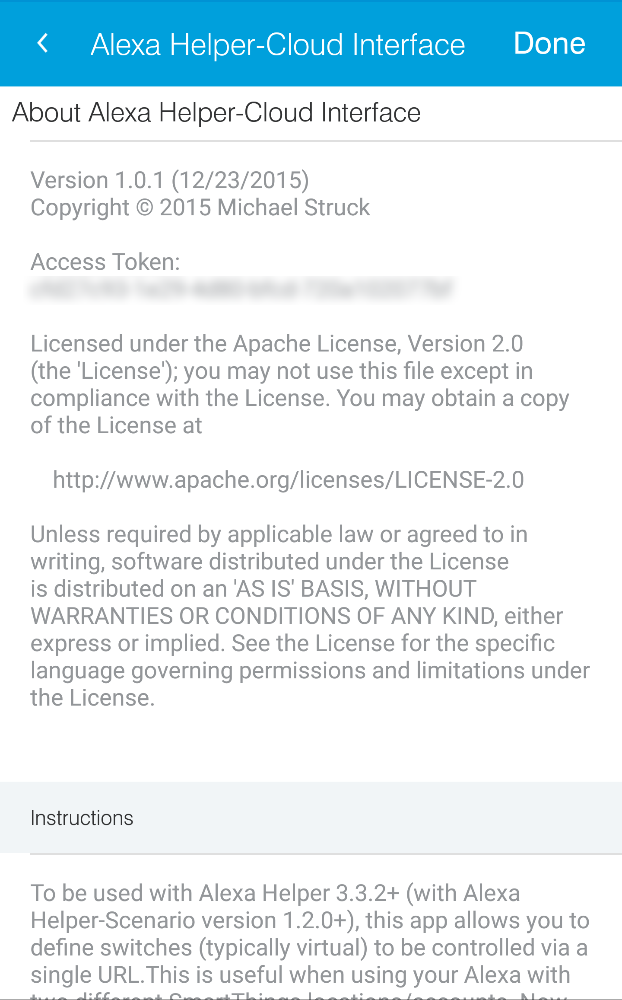


Figure 16- Cloud Interface-About