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 contributions: Tim Slagle, Alex Malikov

Documentation contributions:

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 using the Echo. 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 easily allows.

**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 and 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. There are three types of virtual switches we will discuss when using Alexa Helper: A Simulated Switch, a Momentary Tile and a Virtual dimmer.

**Simulated Switch**-Just like an on/off switch on the wall, when you turn on a simulated switch it remains on, and when you turn it off, it remains off. This type of virtual switch is great when are performing a function that has two states of action. For example, when you go to bed you can say *<“Alexa, turn on Night Mode”>* and a set of functions can occur. When you say *<”Alexa, turn off Night Mode”>* another set of actions can occur with just this one switch.

**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 and when pressed it comes on momentarily and the turns itself off. This type of device is good for items that just need an on trigger, such as changing modes or routines.

**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. This type of device is appropriate for controlling the volume on a speaker, or setting different temperatures on a thermostat.

**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”>*

**Keyboard Shortcuts:** *CTRL+C, CTR+V*

**Warnings/Notices:**

|  |
| --- |
| **Notice** |
| This is a box you will see to give you notices and warnings on things to watch out for during the usage of these products. |

**SmartThings Mobile Application Navigation**: MarketPlace>>>My Home>>>SmartApps

**SmartApps Names:** **Alexa Helper**, **Alexa Helper-Cloud Interface**

**Installation:**

While there are a total of four components of Alexa Helper, only the Parent/Child(scenario) apps are required for basic functionality. 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*).

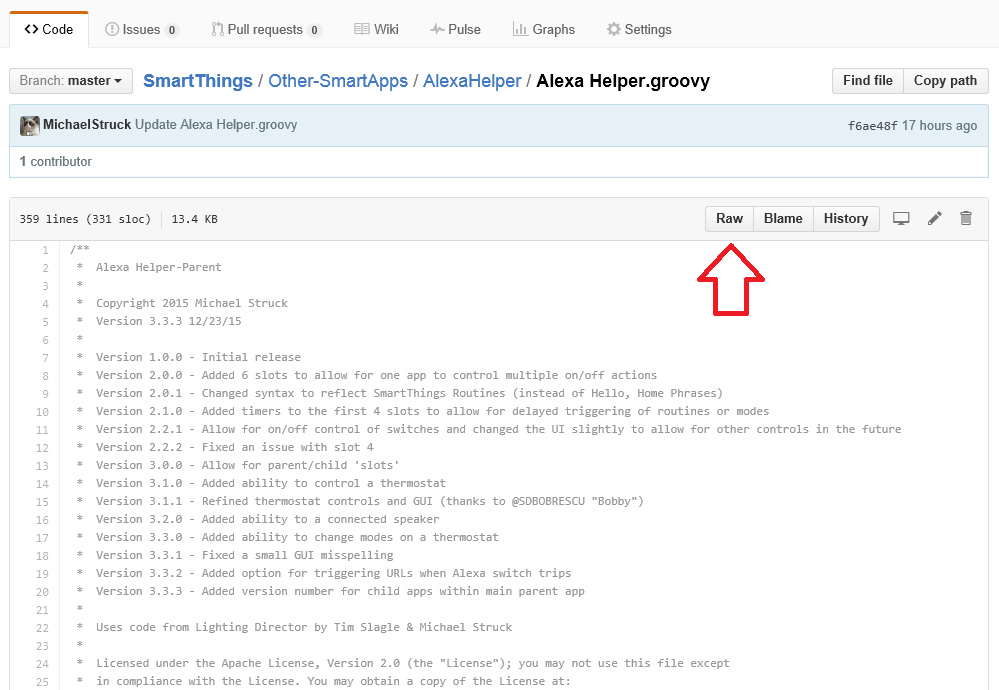


Figure - GitHub "Raw" Button

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

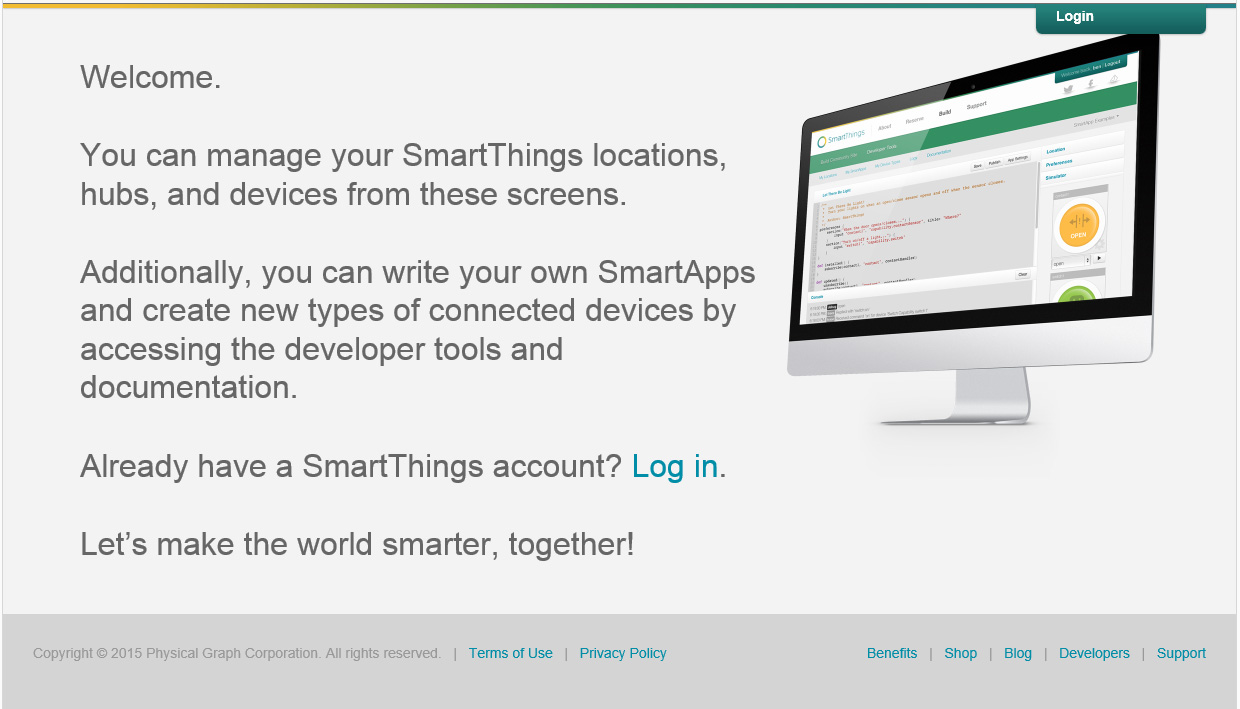


Figure - SmartThings IDE Login

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

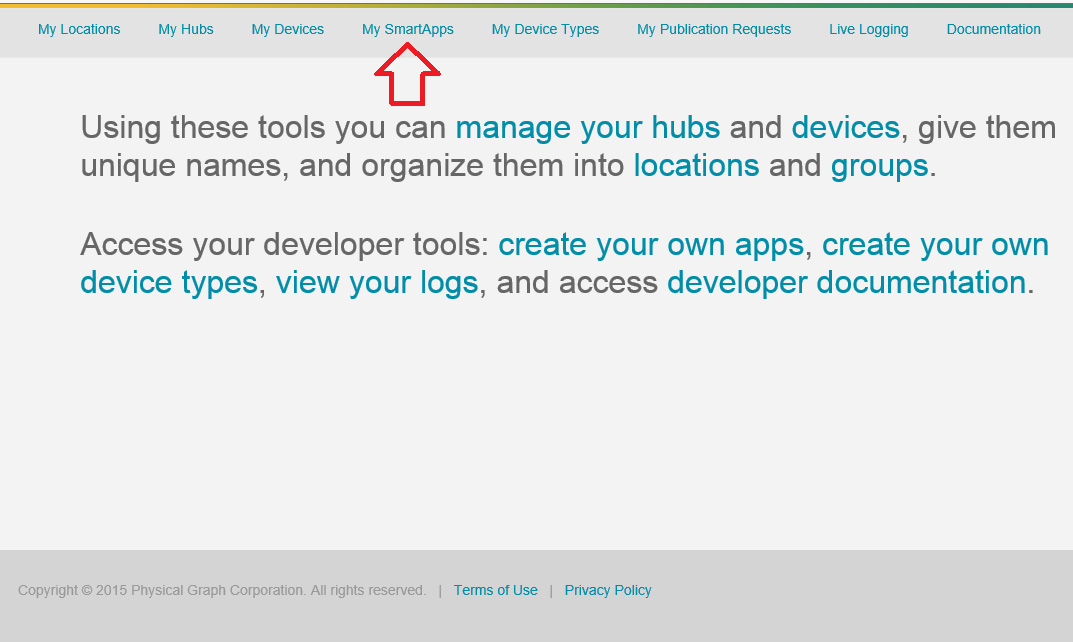


Figure - SmartThings IDE My SmartApps

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 and log in
* 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 and log in
* 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
* 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.

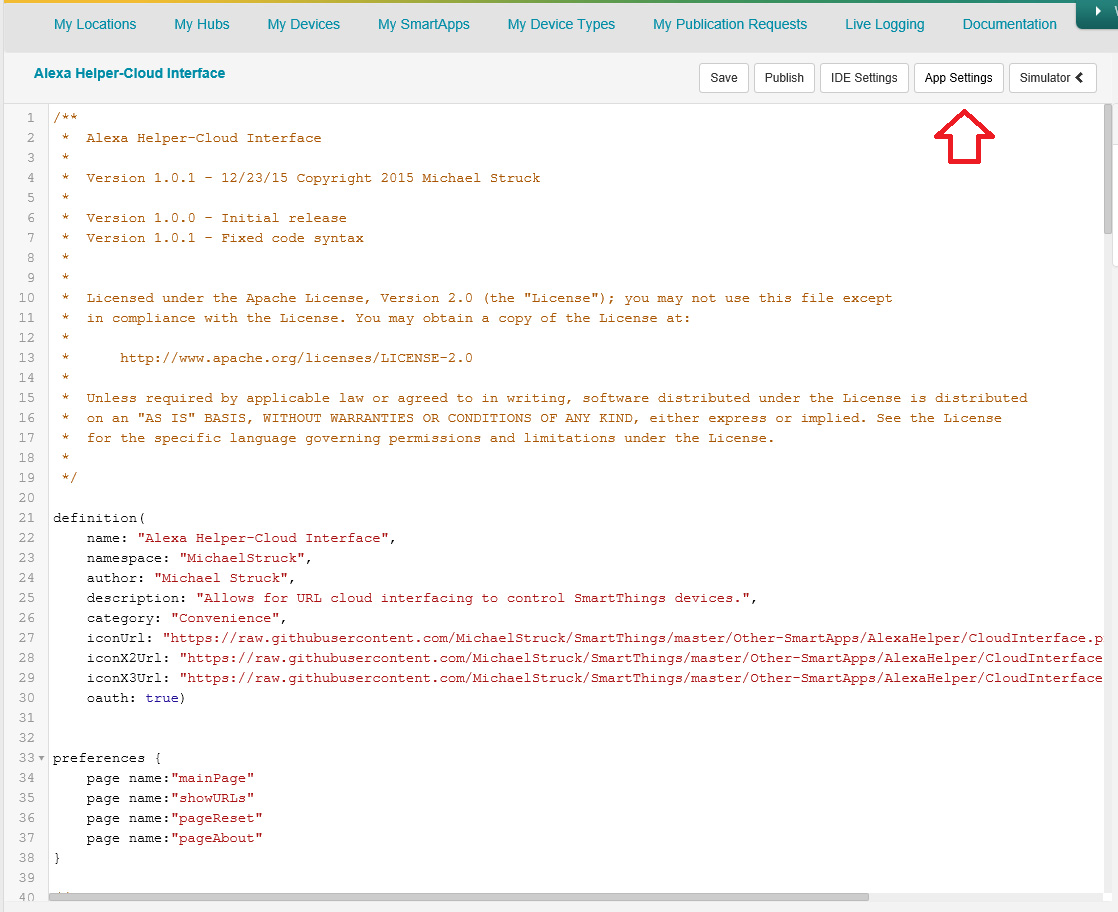


Figure 4-SmartThings IDE App Settings

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

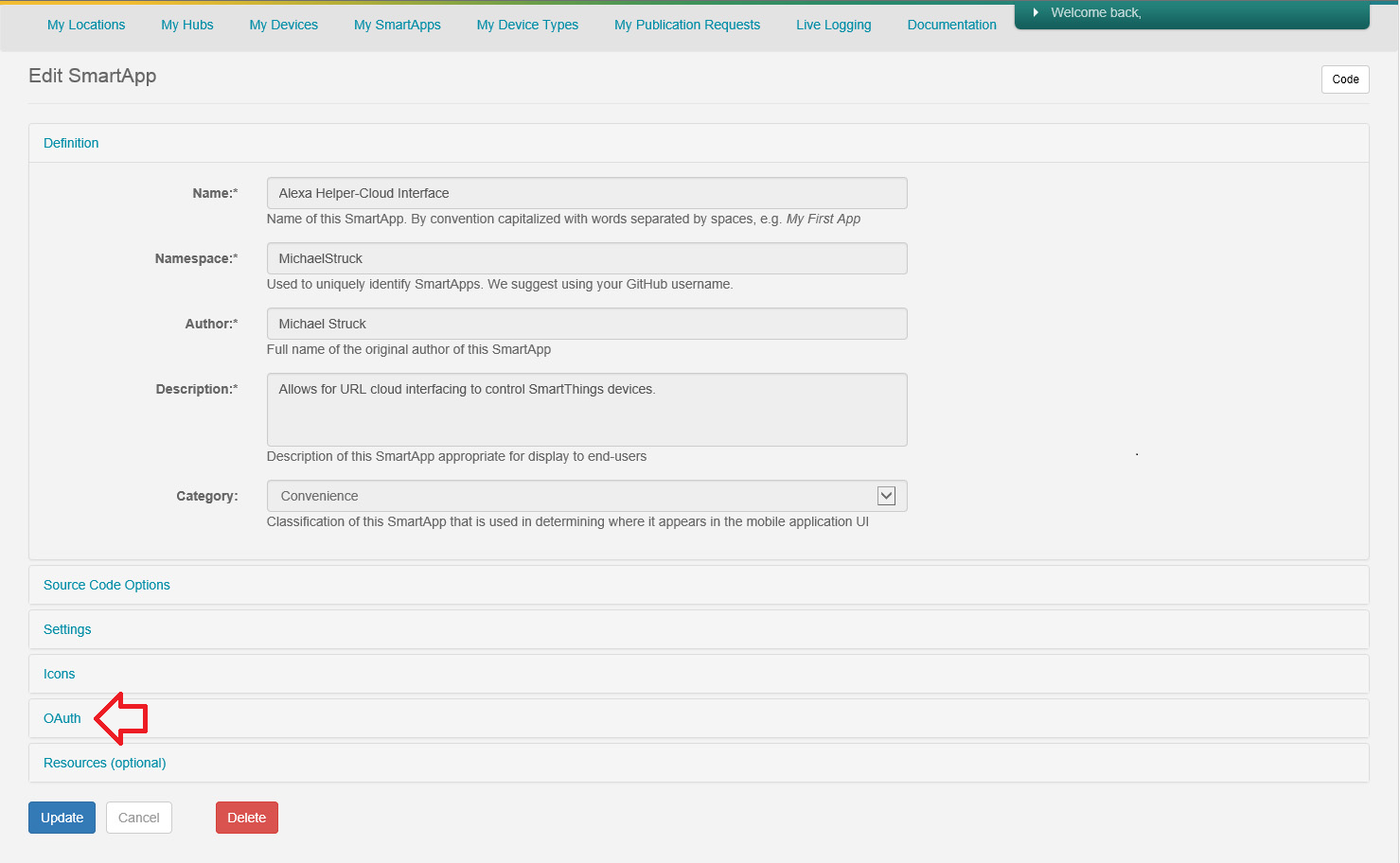


Figure 5- SmartThings IDE OAuth Settings

Clicking the OAuth link will reveal a button labeled 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 down these codes or add any other information to this page; OAuth simply needs to be enabled for **Alexa Helper-Cloud Interface** to generate your unique URLs.

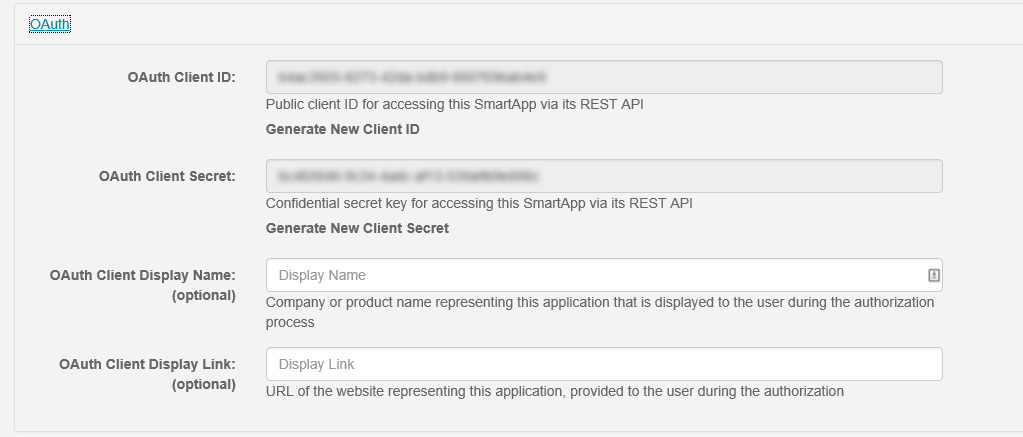


Figure 6- SmartThings IDE-OAuth Settings

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 ensure you can turn on and off the switches associated with your account

If any of the conditions above are not true, the Alexa Helper SmartApps will not function in your environment. Please refer to your Amazon Echo documentation, use the Amazon Echo App, 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 Setup:**

While not required, virtual switches allow for better control of your SmartThings environment than using the physical switches in your house. This allows a non-physical switch to be “tripped” to make other events happens, some of which the Amazon Echo cannot perform natively. This includes changing security modes, or running SmartThings routines, which can include a lot of functions that need to happen at the same time. For example, having a virtual switch named “Night Mode” can be tied to a “Good Night” Routine, which could shut off the lights, lock the door, turn off the heat, set the mode to “Asleep” and set the Smart Home Monitor to “Armed:Stay”. All with saying *<”Alexa, turn on Night Mode”>*

Virtual switches are not present by default in your SmartThings Mobile Application; you must gain access to the SmartThings Developer IDE to create these switches. However, the process is rather simple.

Just like adding the code for the Alexa Helper SmartApp, go to the IDE at <http://ide.smartthings.com>.

When you have logged into the site, find the section labeled My Devices. Click this link.

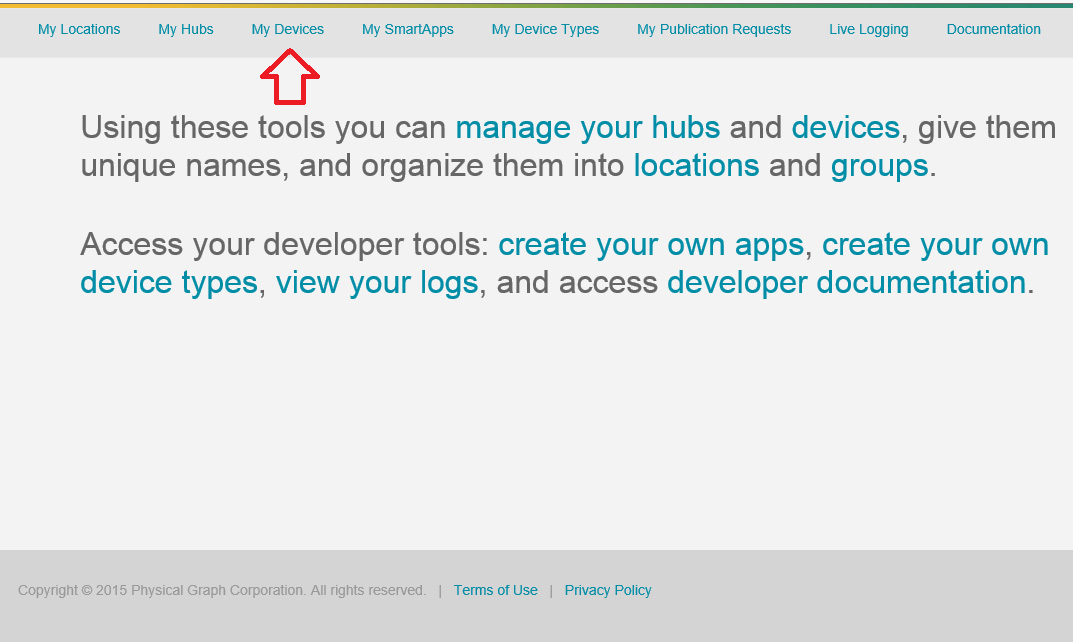


Figure - SmartThings IDE My Devices

On this next page you will be presented a list of all of the devices within your SmartThings account. These are typically the physical devices within your account. From here, find the button in the upper right corner of the page labeled +New Device. Click this link and you will be presented a number of fields to fill in.

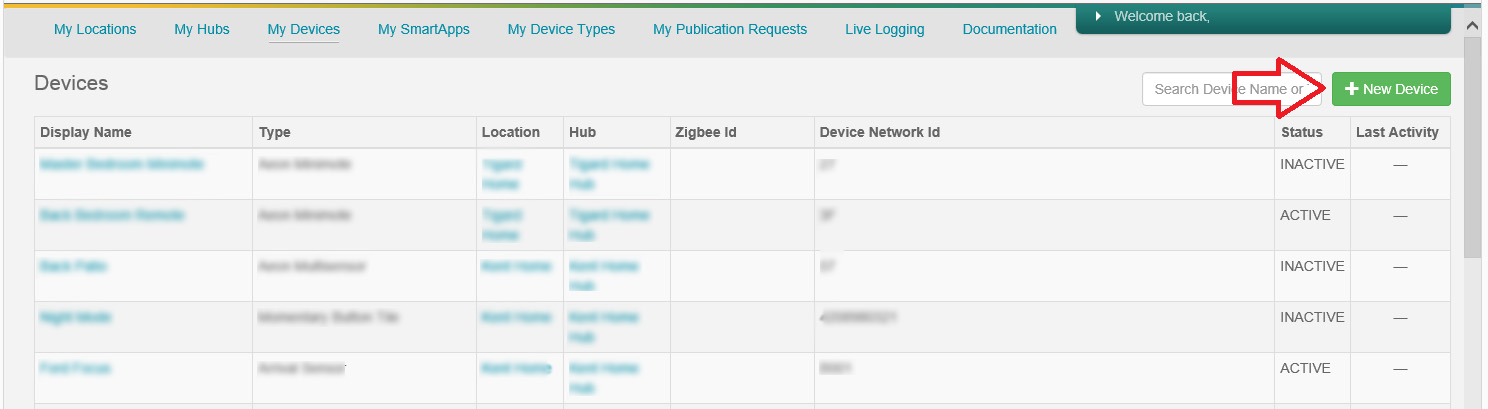


Figure - SmartThings IDE New Device

|  |  |
| --- | --- |
| Starting at the top of the fields, fill out the name of the device. You may want to create a sort of naming convention with starting everything with a ‘v’ such as vThermostat. However, if you plan to use these names with Alexa you should use a name that Alexa can recognize (Like “Night Mode”, “Day Mode”, etc.). You can always change this name later.  Once you fill out the name, simply use the same name used above for the label.  Since this is a virtual device, leave the Zigbee Id field blank.  For the field labeled Device Network Id, fill in a number that is unique to your environment (you may need to look at the list from the previous screen to determine what is unique.) This field is required, however, the number is not used anywhere else, so a simple “1234567890” could work well for this. Any subsequent devices created could build off of this (for example: “223456790”).  In the next field labeled Type, choose either a “Momentary Button Tile” for a momentary switch, or “Simulated Switch” that has two unique states (on/off). | C:\Users\Michael\Desktop\ClipArt-AlexaHelper\createDevice.jpg  Figure -SmartThings IDE Create New Device |

The Version field should remain as “Published”, and both the Location and Hub should have your location’s hub listed. Ensure both of these fields are not empty, however Group can remain empty. When you are happy with your entries, click the Create button. You will be taken back to the My Devices page, which should now include your newly created virtual switch. You can always click on the switch again and edit it should you need to, or you can edit some of the attributes (such as the name) in the SmartThings mobile application on your phone or tablet.

**Virtual Dimmer Installation:**

Setting up a virtual dimmer is slightly different than setting up a simulated switch or momentary tile. Both of those switches are native to SmartThings, however a virtual dimmer is not. However, code is provided as part of the Alexa Helper suite that will allow you to create virtual dimmers as easy as the other virtual switches.

First, copy the code from the following GitHub location:

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

Just as with the code for the SmartApps, be sure to click the Raw link, and then select all of the code (*CTRL+A*), and then copy it (*CTRL+C*) to your computer’s clipboard.

Then go to the SmartThings IDE at <http://ide.smartthings.com>. Log into the site and proceed to the link at the top labeled My Device Types. Click on this link.

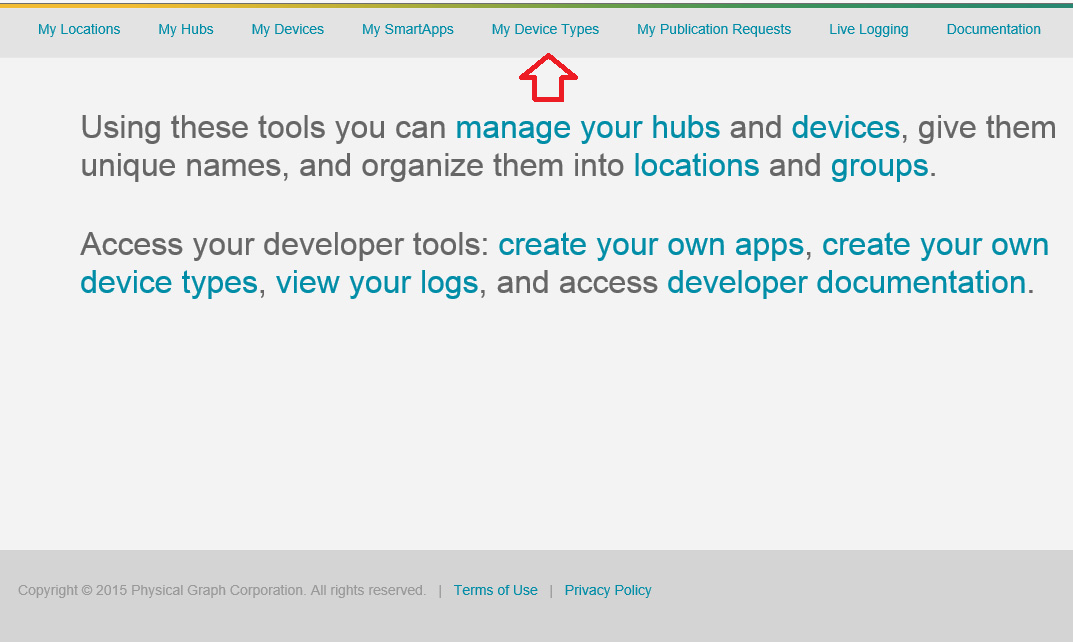
****

Figure -SmartThings IDE My Device Types

This will present you with a list of custom device types that you may have within your account. If you do not yet have the virtual dimmer listed, click the link +New Device Type in the upper right-hand corner of the screen.

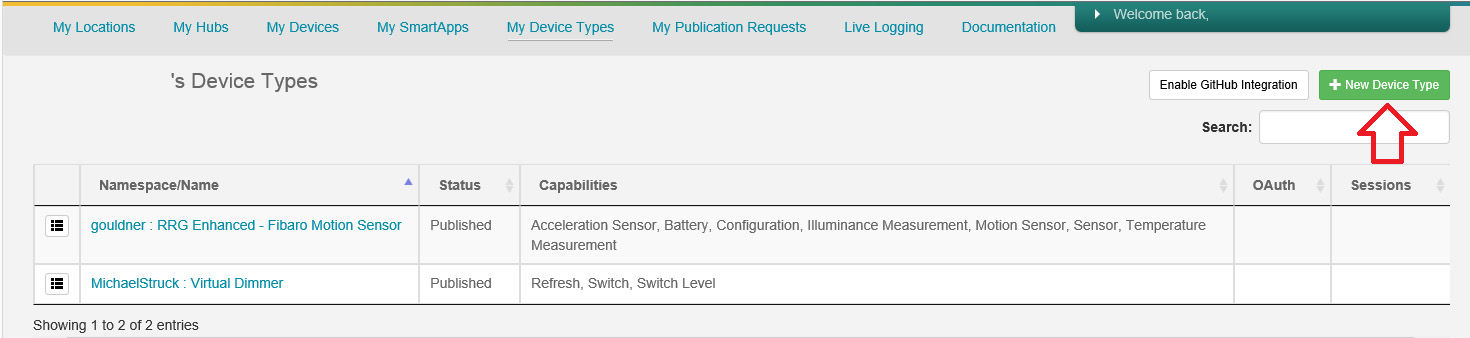
****

Figure - SmartThings IDE - My Device Type

There will be three tabs at the top, which represent the different ways to add a device. Choose From Code and you will be presented an empty area where the code you copied earlier can be placed. In the open area in the middle paste (*CTRL+V*) this code.

From here, simply Save and Publish this code, exactly as you did when creating the SmartApps.

****

Figure - SmartThings IDE – My Device Type Code

|  |
| --- |
| Notice |
| The code for the virtual dimmer only needs to be installed in this manner once; you can create multiple virtual dimmers now as long as this code is in place. |

The remaining steps for creating the virtual dimmer are the same as the other virtual switches:

* Find the section labeled My Devices at the top of <http://ide.smartthings.com>. Click this link.
* Find the button in the upper right corner of the page labeled +New Device. Click this link.

|  |  |
| --- | --- |
| * Fill out the fields as you did for the virtual switch, being mindful of what you use for the Name/Label if you plan to speak it to the Amazon Echo. * For the field labeled Device Network Id, ensure the number you choose is unique to your SmartThings environment. * In the field labeled Type use the pull down and scroll to bottom of the list where you will find a new device called “Virtual Dimmer”. Choose this device type. If this device does not appear in the list, go back to the code installation section above to ensure you properly published the code. * The Version field should read “Published”. * Both the Location and Hub should have your location’s hub listed. * Group can remain empty. * Click the Create button.   You will be taken back to the My Devices page, which should now include your newly created virtual dimmer switch.  From here you should have all of the components to finally install and operate Alexa Helper. | **C:\Users\Michael\Desktop\ClipArt-AlexaHelper\VirtualDimmerSelection.jpg**  Figure 13- SmartThings New Device-Virtual Dimmer |

**Alexa Helper Installation:**

If you installed the code properly, you should find both the Alexa Helper and Alexa Helper-Cloud Interface SmartApps located in the SmartThings Mobile Application under *Market Place>>>SmartApps>>>+MyApps*. If you just need to control lights, routines, modes, etc. simply install the **Alexa Helper** Smart App. If you require more advanced functionality, also install the **Alexa Helper-Cloud Interface** Smart App. If the **Alexa Helper-Scenario** SmartApp shows up in the list, this indicates you may have inadvertently published it. Go back to My SmartApps, choose the edit button for **Alexa Helper-Scenario**, select all of the code (*CTRL+A*) and copy it (*CTRL+C*). Then, click the App Settings link. On this page, delete the code. Then, follow the process to create a new SmartApp, ensuring when you are done that you DO NOT publish the code…only save it.

Once you complete these steps, you should see the new SmartApps on the My Home>>>SmartApps section of the SmartThings Mobile Application.

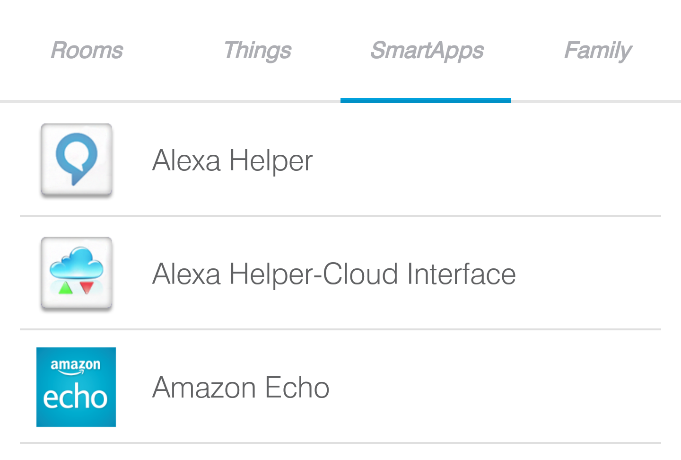


Figure 14- SmartThing SmartApps

**Scenario Setup:**

|  |  |
| --- | --- |
| **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 15-Alexa Helper-Cloud Interface | C:\Users\Michael\Desktop\ClipArt-AlexaHelper\Screenshot_20151224-120822.png  Figure 16- Cloud Interface-Show URLs |

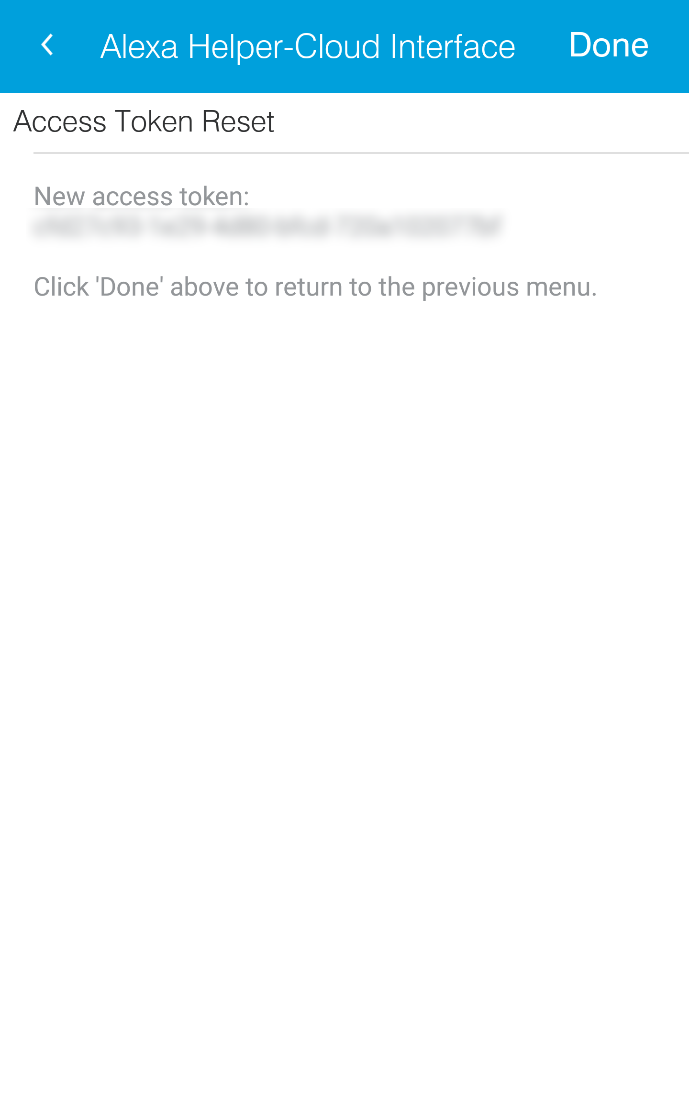
**

Figure 17- Cloud Interface-Access Token Reset

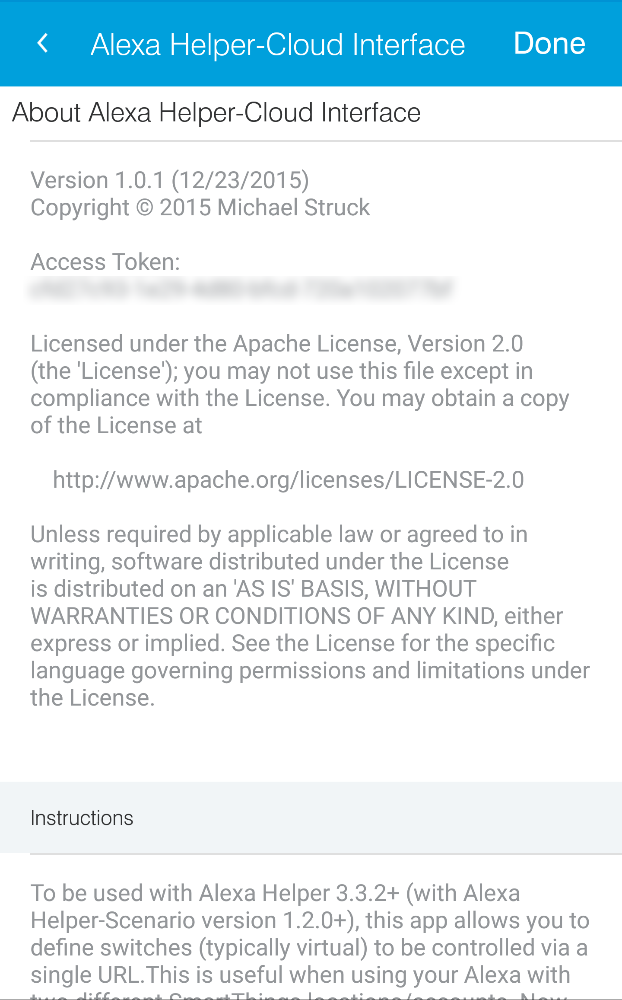


Figure 18- Cloud Interface-About