[www.particle.io](http://www.particle.io)

**2/22/22, 8/11/22**

<https://docs.particle.io/reference/device-cloud/api/#get-a-variable-value>

**REST API 🡪 browser🡪regular IQfact+ automation**

**All the information is**

How does server control Device through Rest API:

1. Read Device info:

* Device info is stored in database.
* Test flow in txt/json file, which can be converted to json files(test flow).

1. Control device

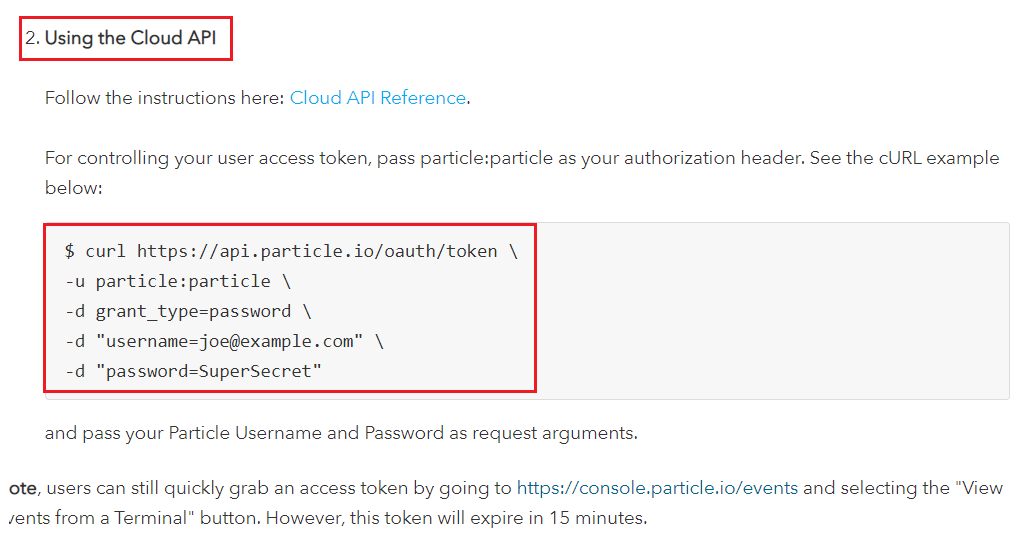
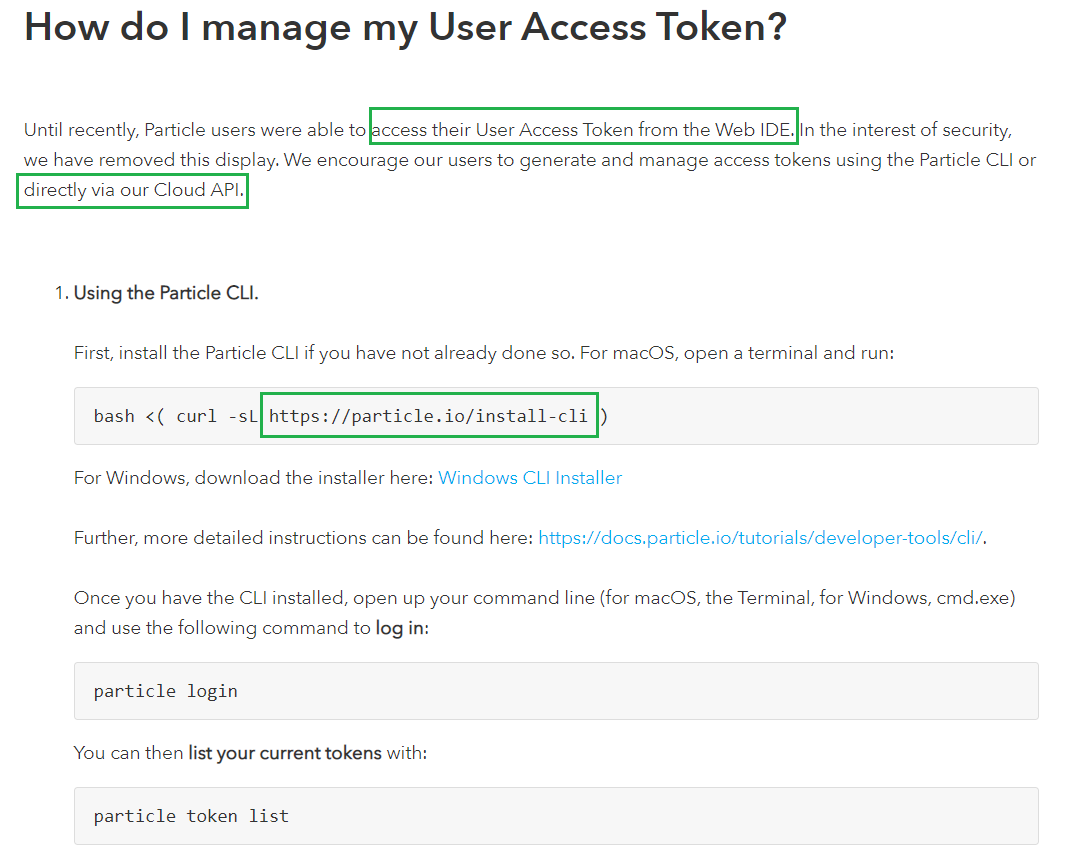
* Add/update Device a flow
* Device to run a flow through API or UI (there is a run button)
* API to run a test: Create a new test

POST ../v1/device/test\_name

Dev name/id/ip, test\_flow\_name, test\_name

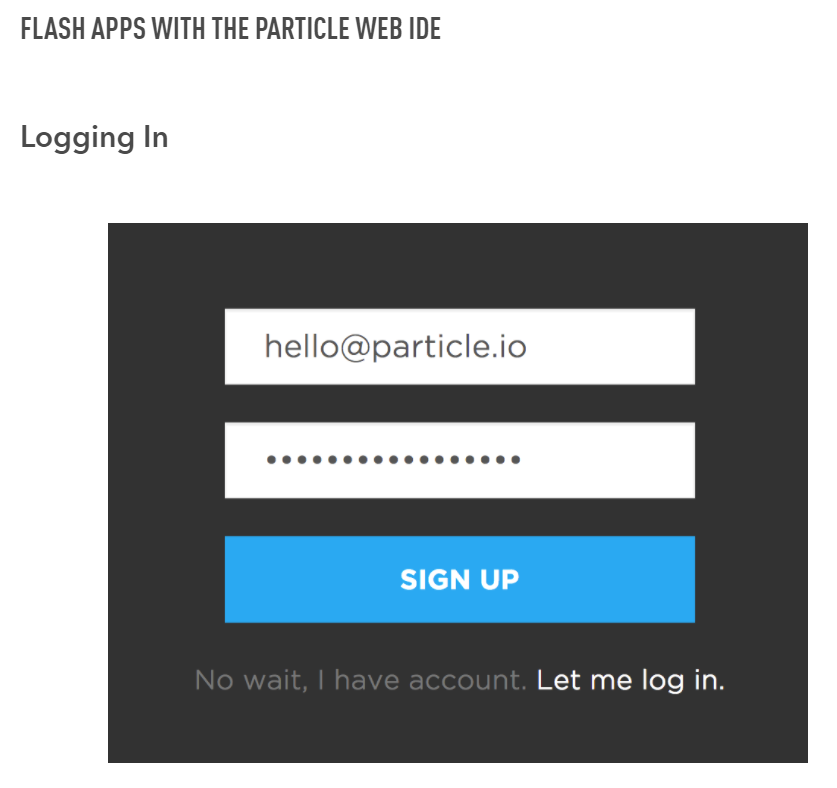
* Once server receive the request, it will update status of device and pass the demand to agent; agent will initiate test program with all the info passed by server.
* Question? How the test program gets work-flow. So agent should have a functionality that can trigger server to download workflow to device PC. Or test flow can be retrieved from local PC; API needs to pass the “path to the workflow”.

1. Manage work flow /

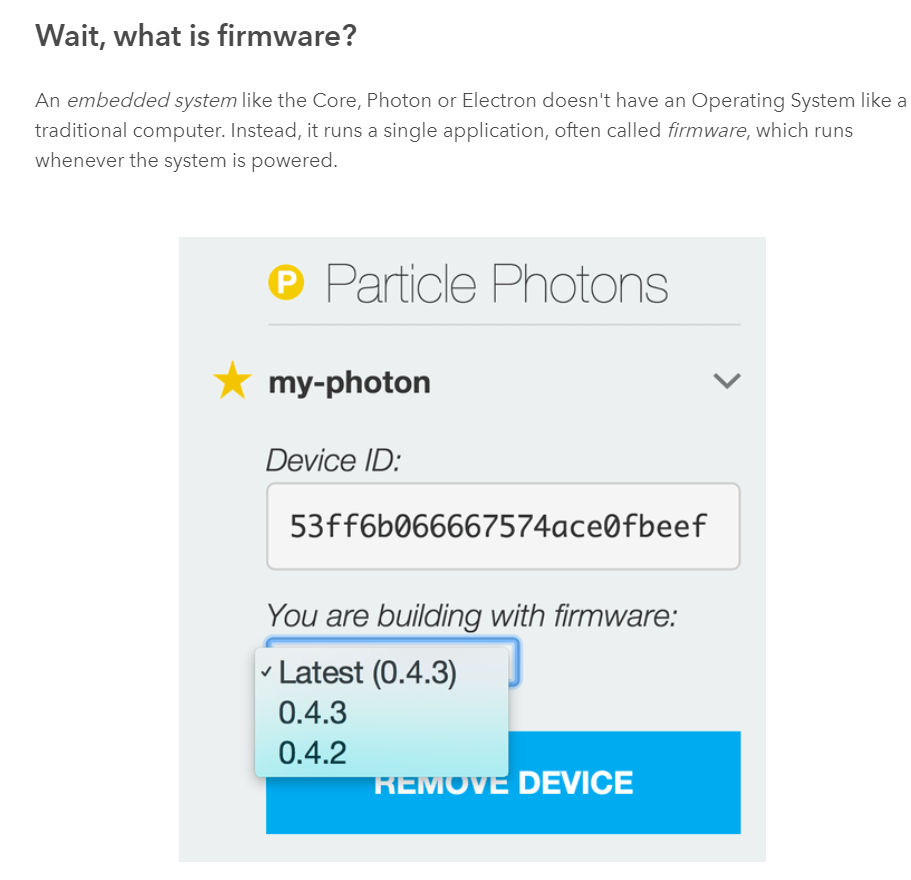
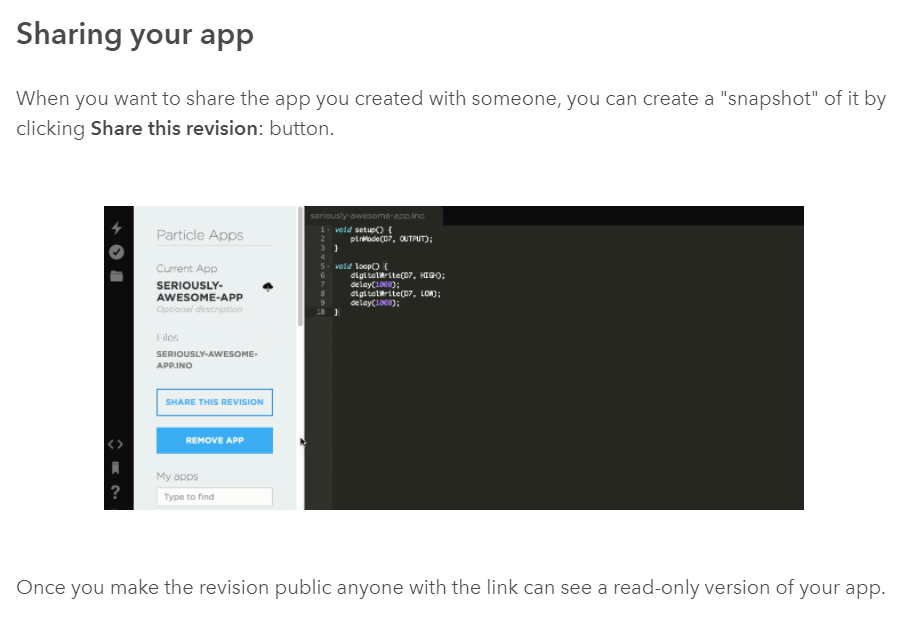
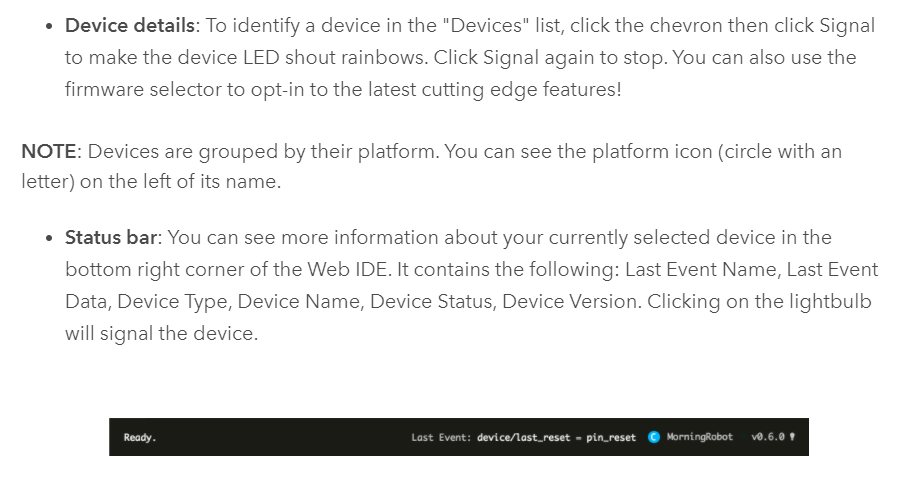
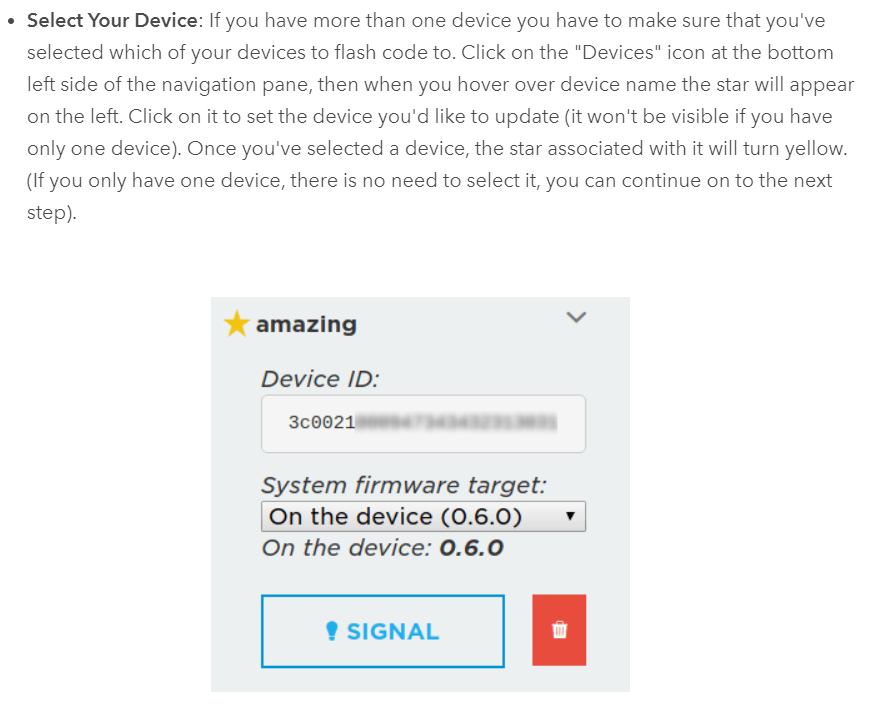
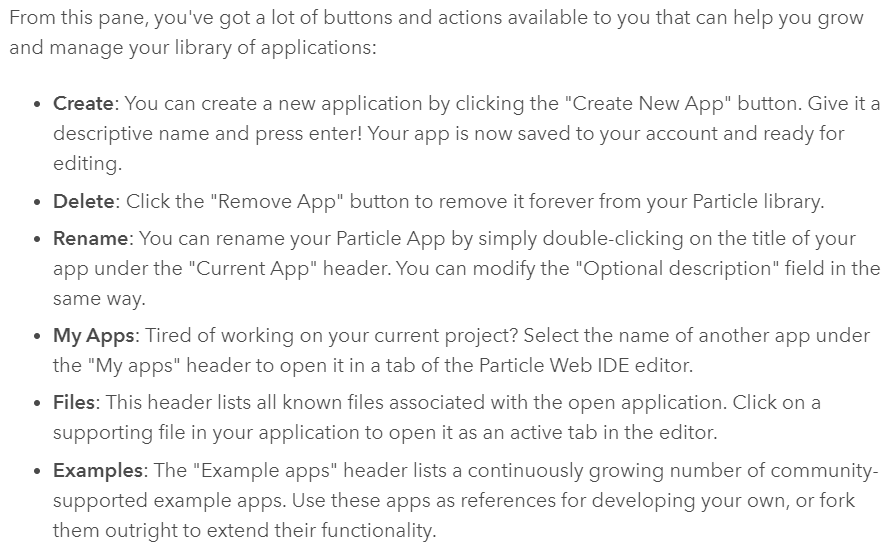
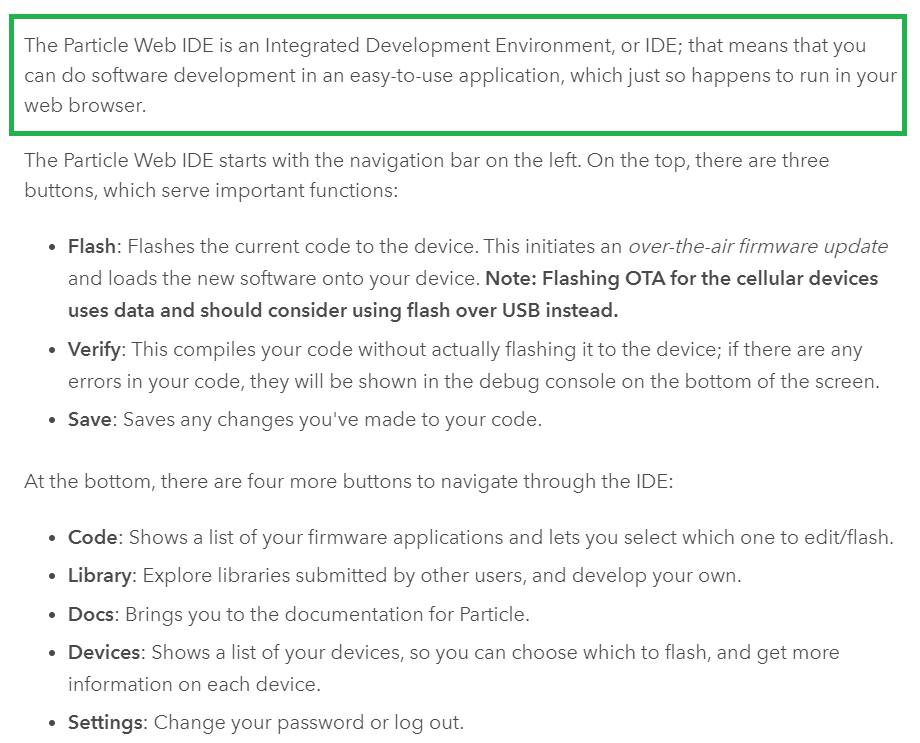
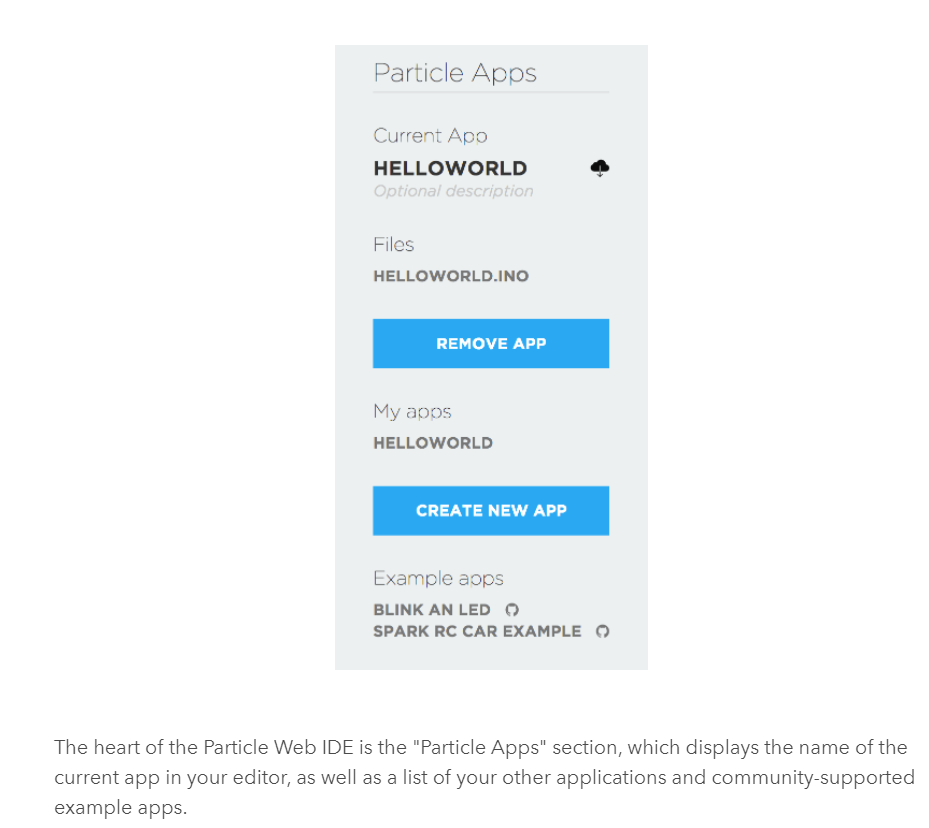
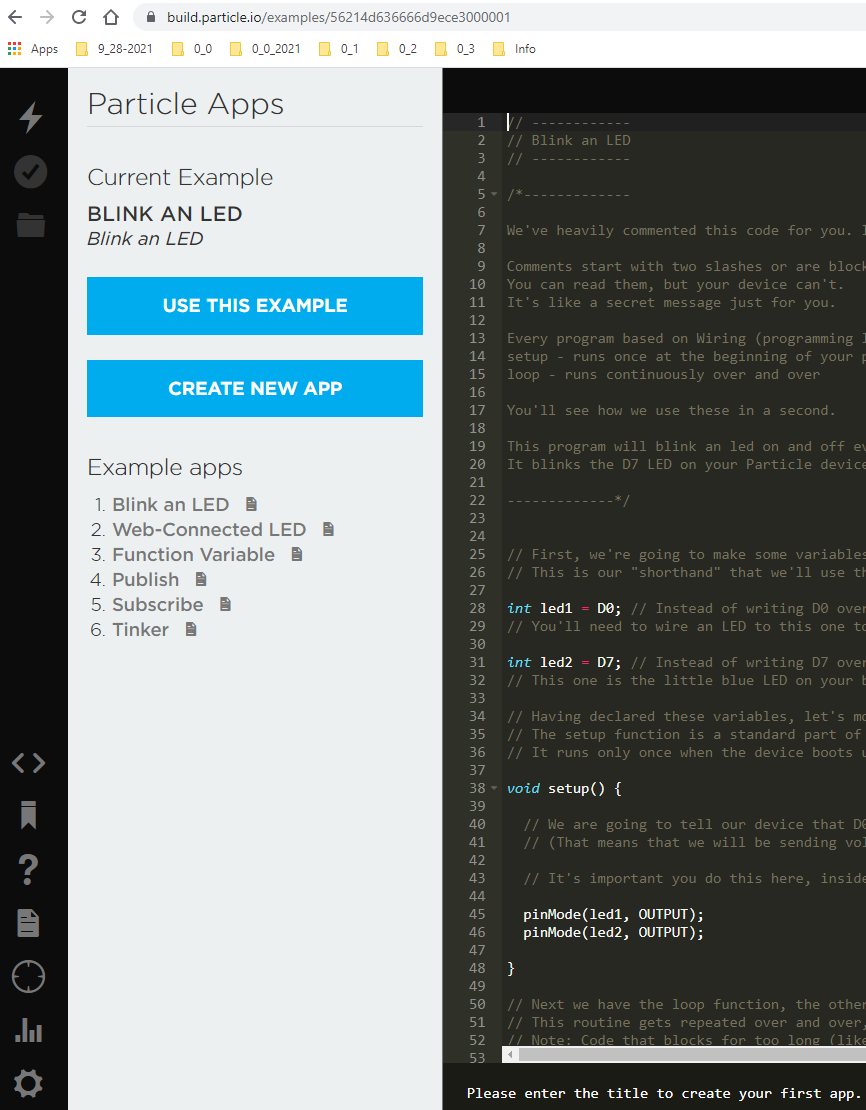
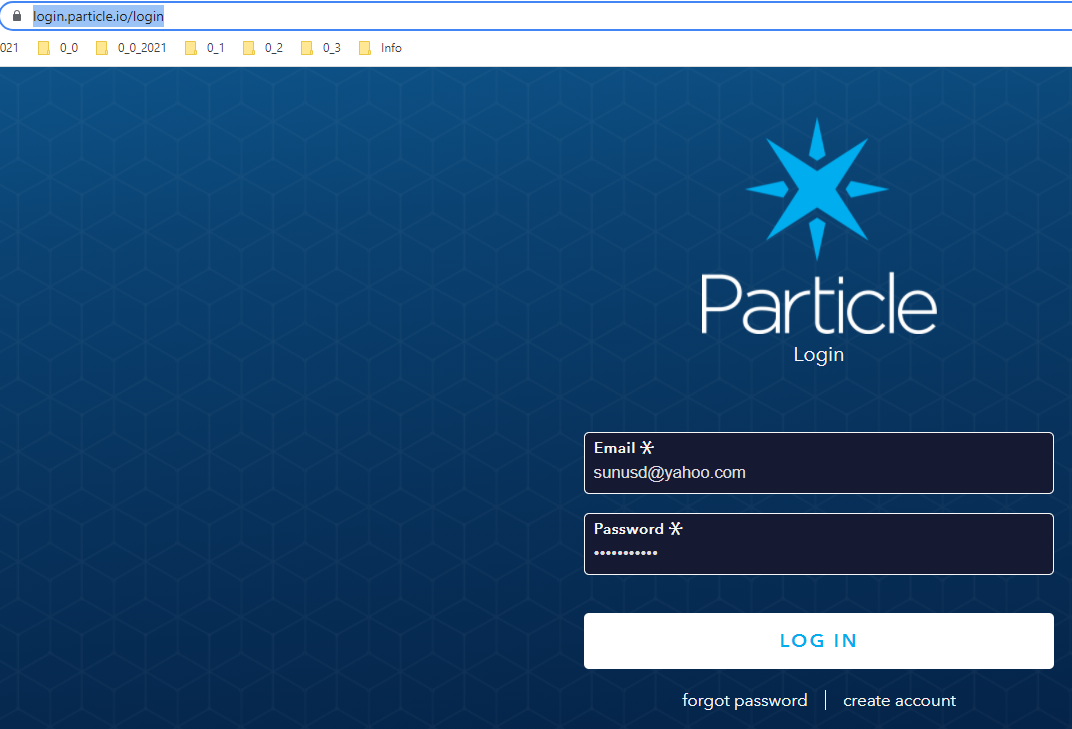


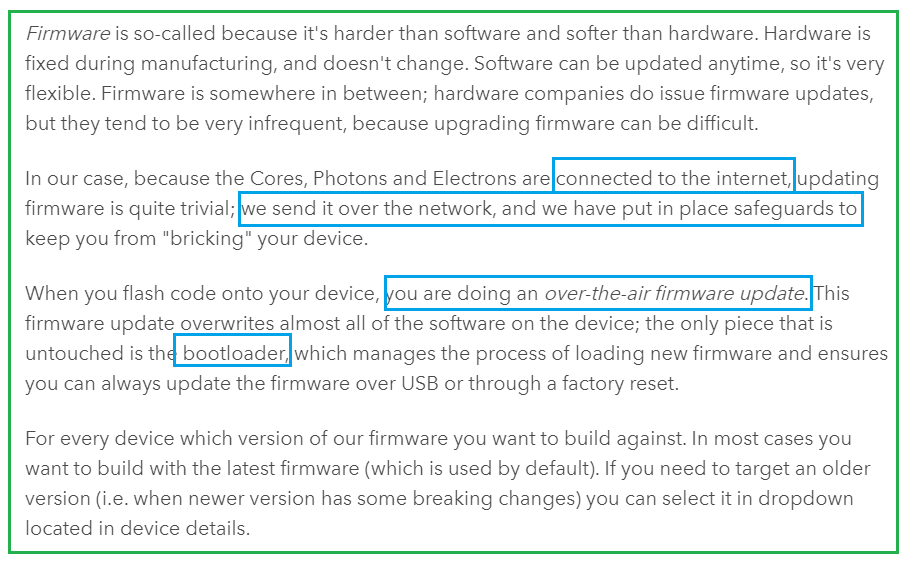
[**https://docs.particle.io/tutorials/developer-tools/build/**](https://docs.particle.io/tutorials/developer-tools/build/)

The Particle Web IDE makes it easy to write code, compile, and flash devices over-the-air (OTA) right from a web browser!

****

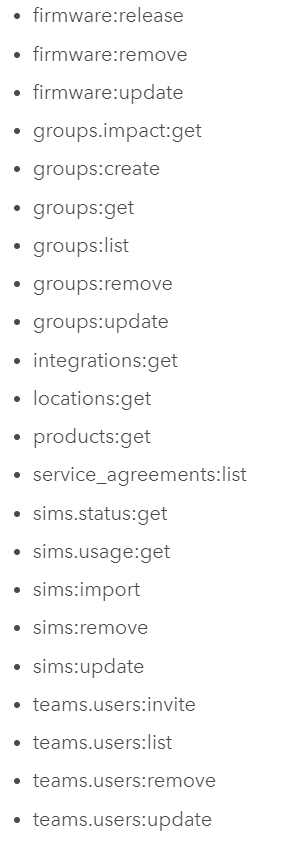
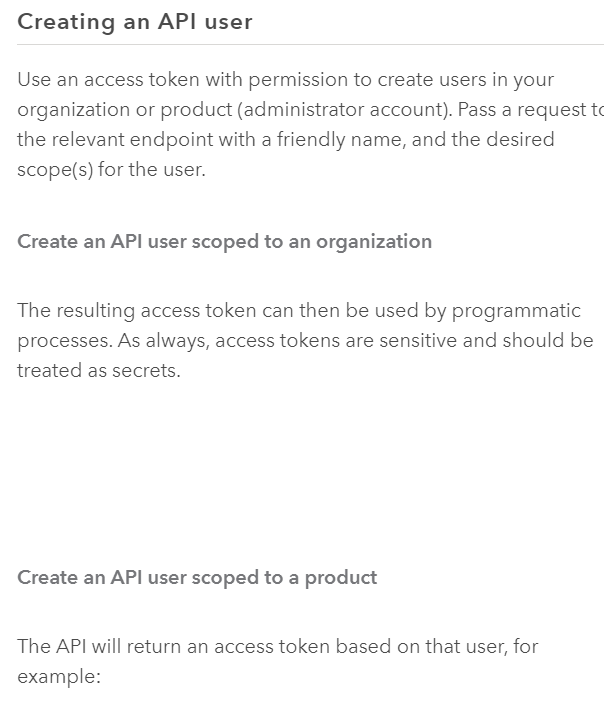
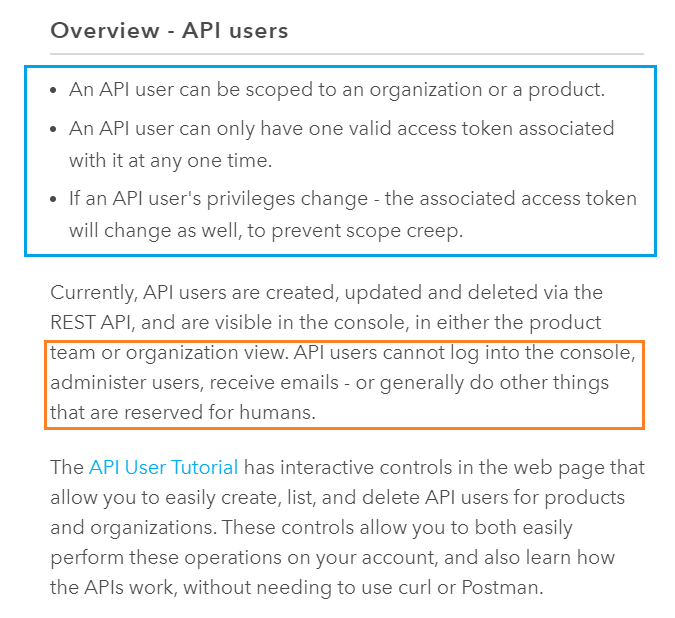
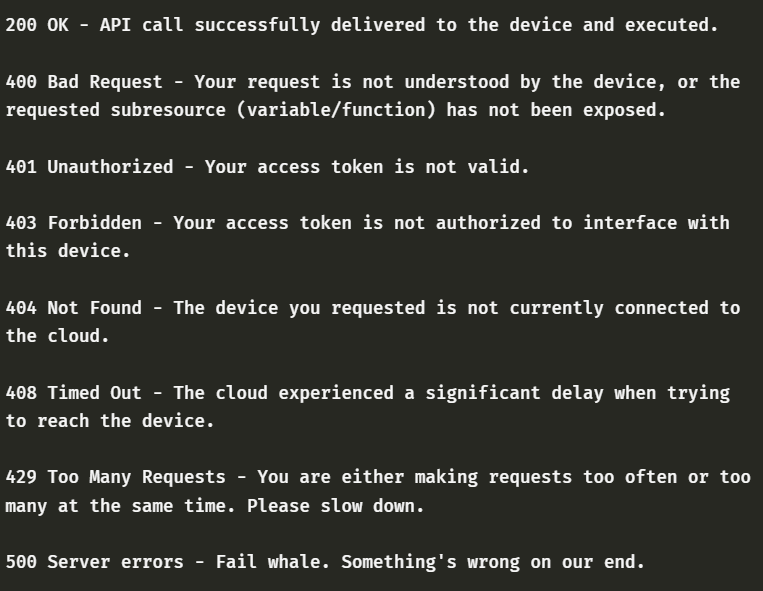
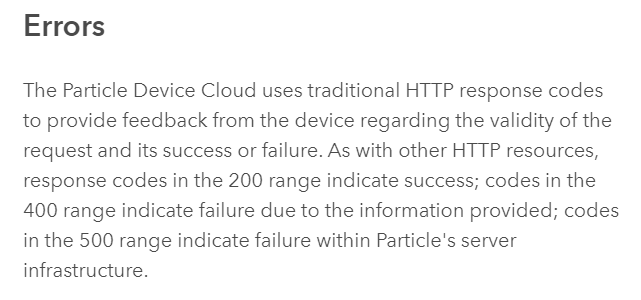
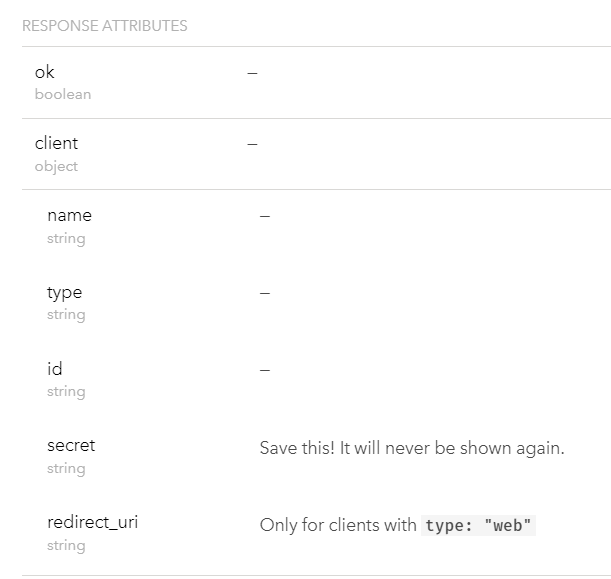
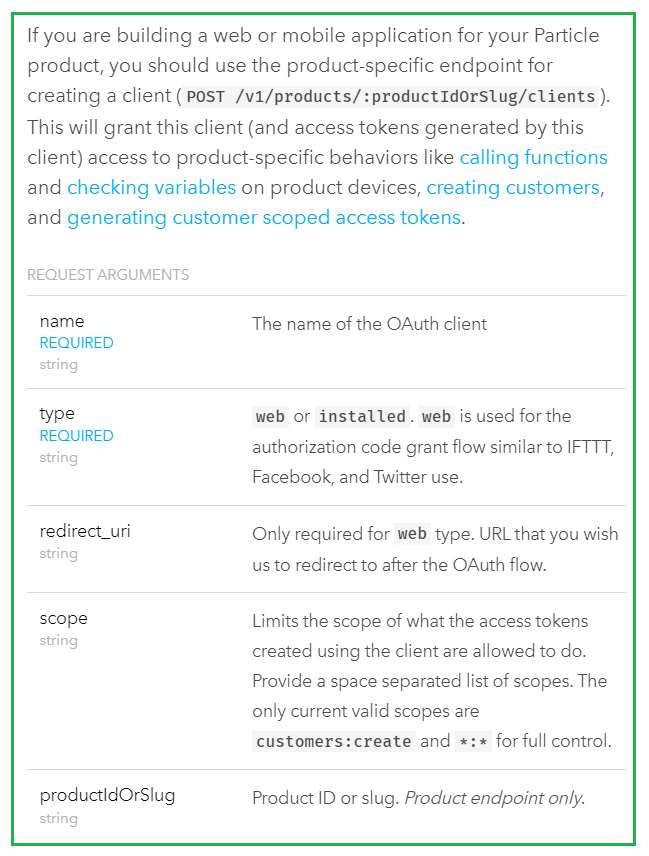
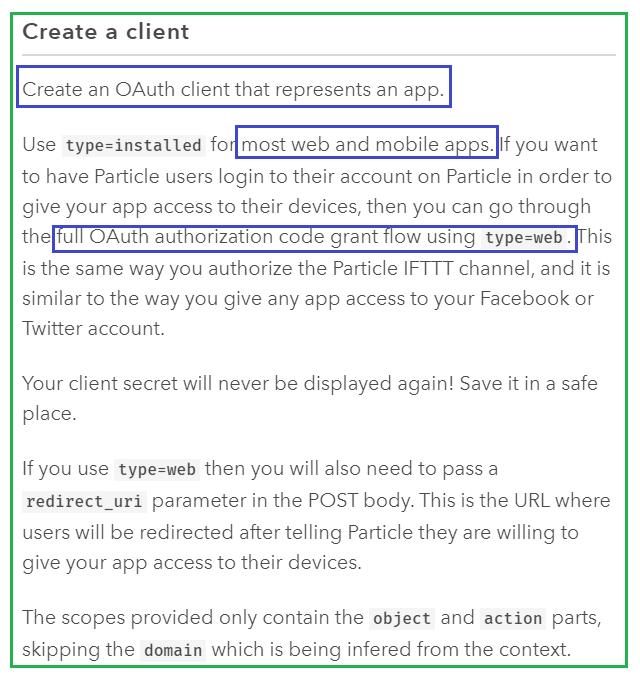
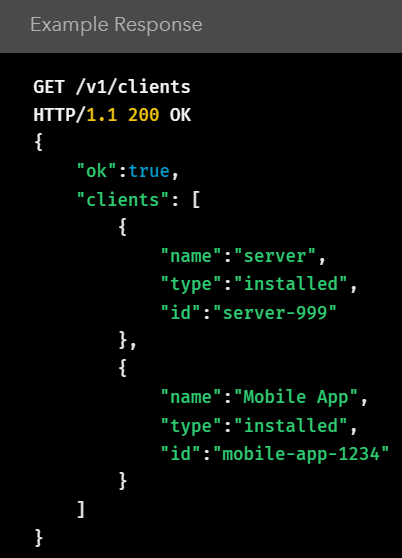
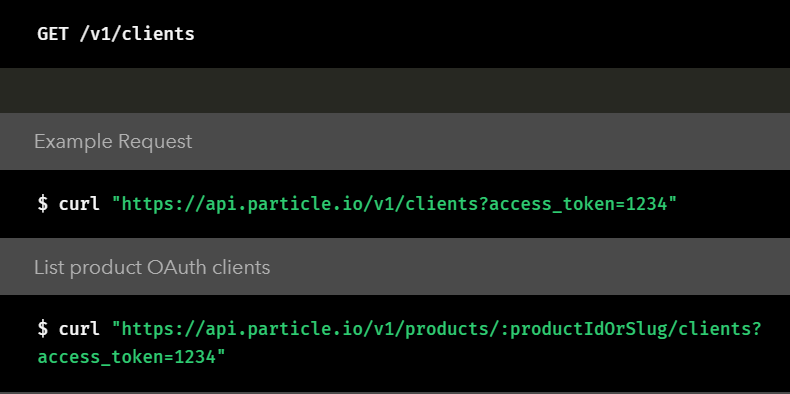
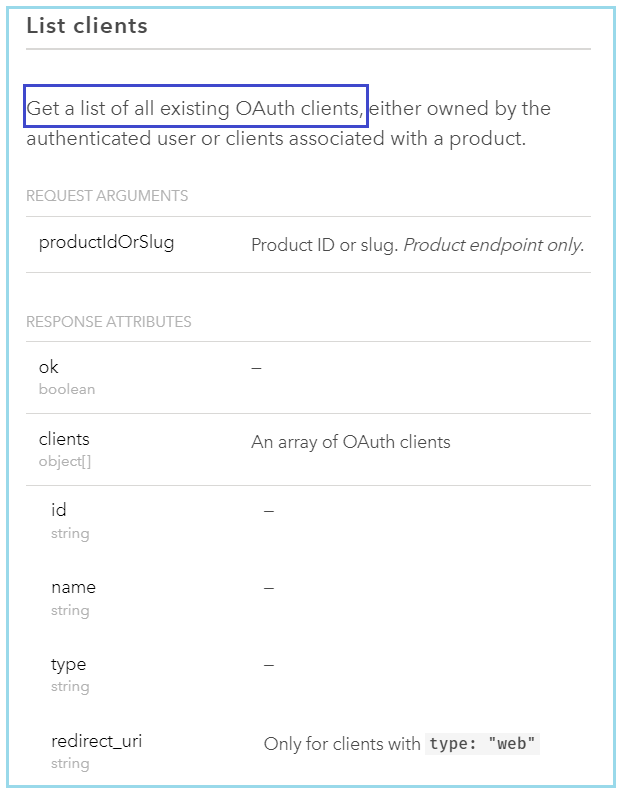
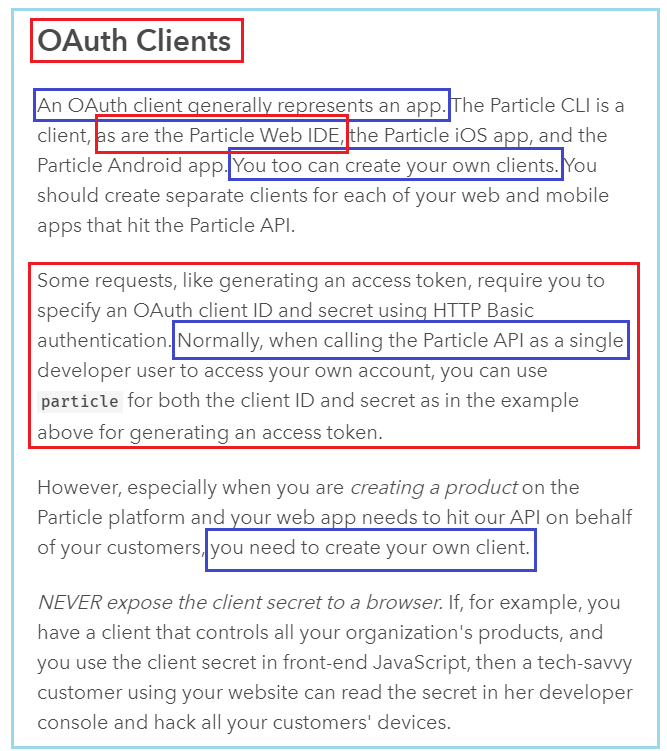
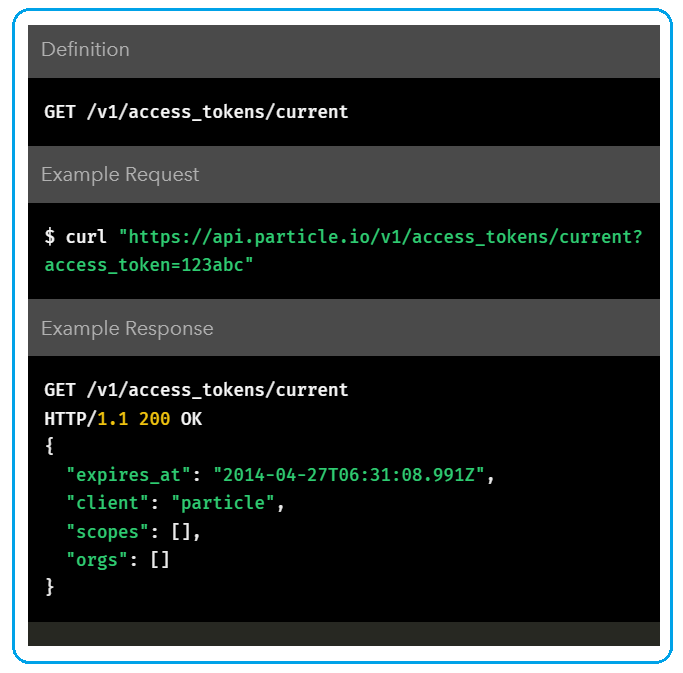
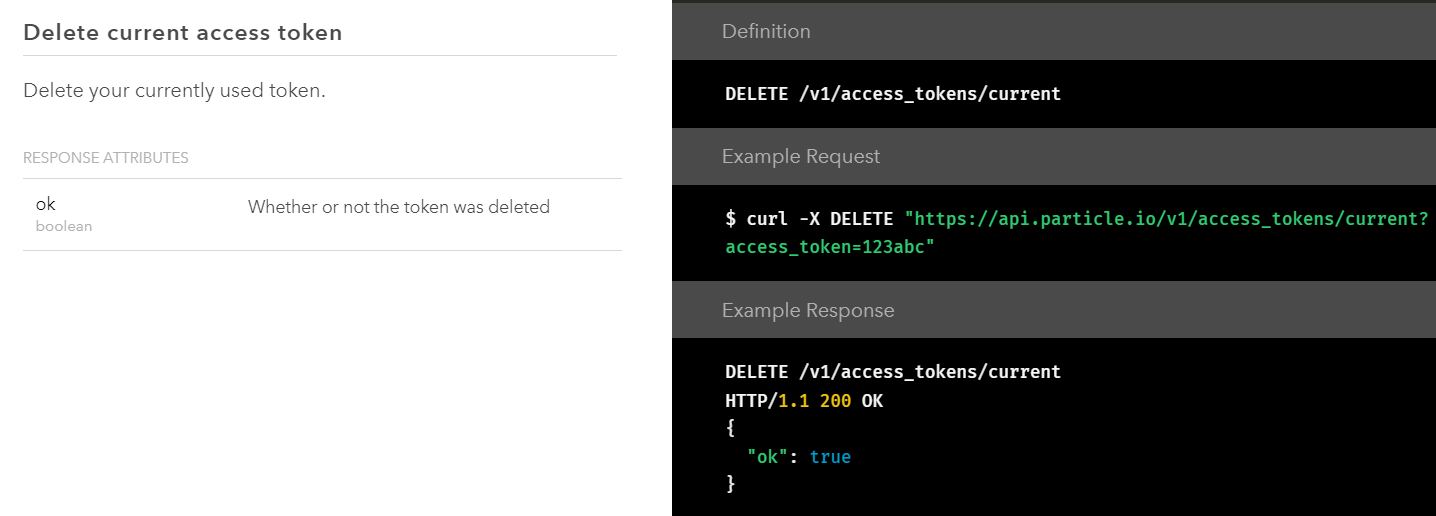
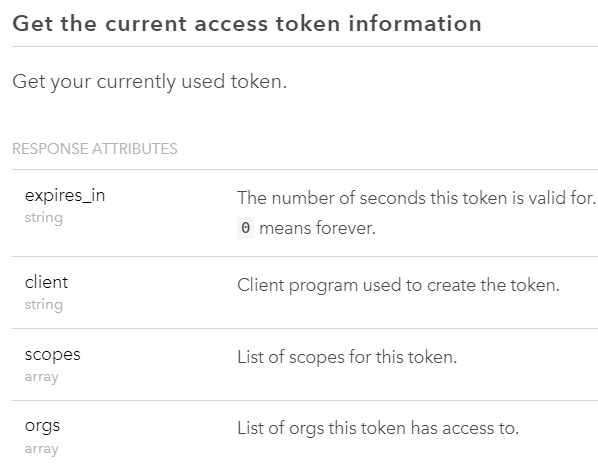
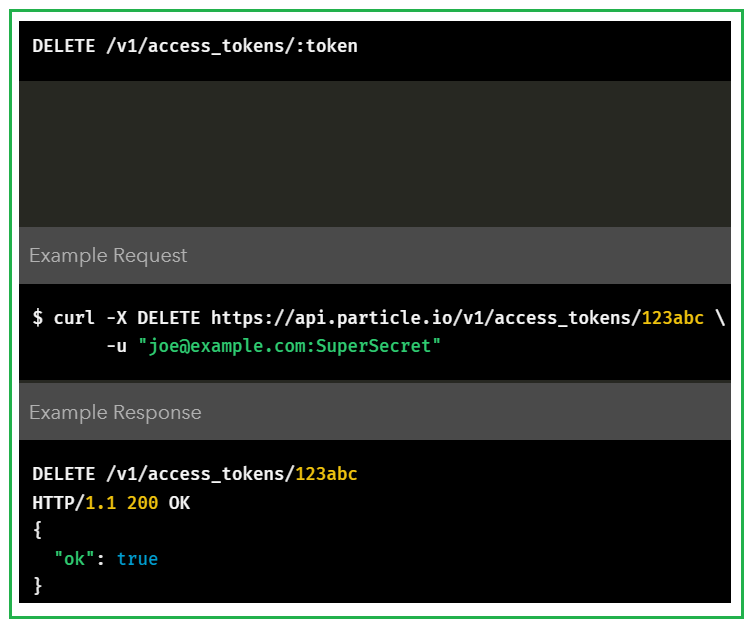
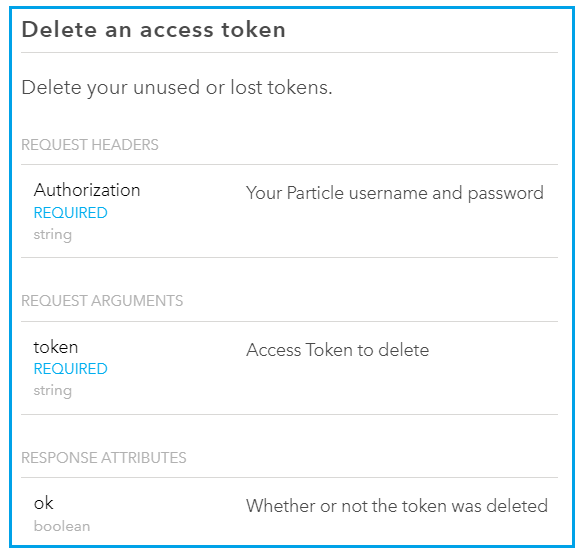
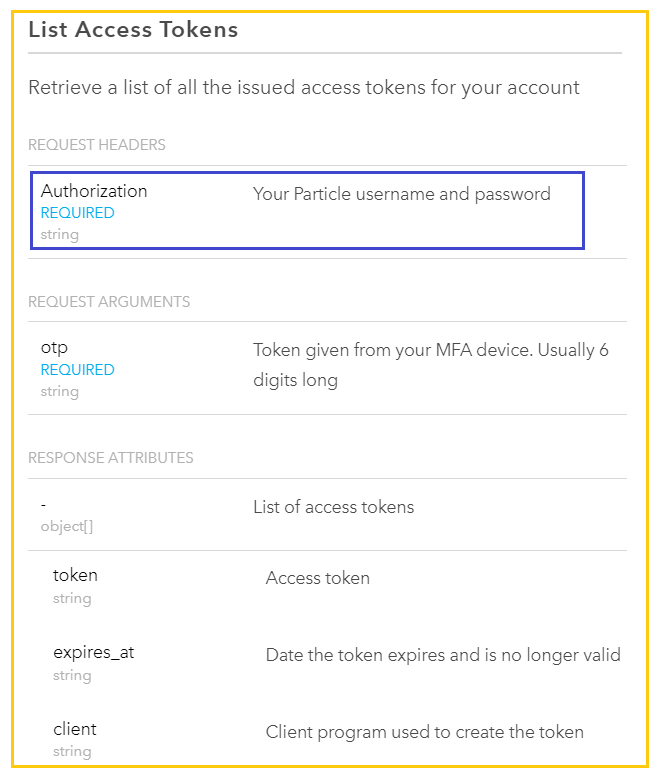
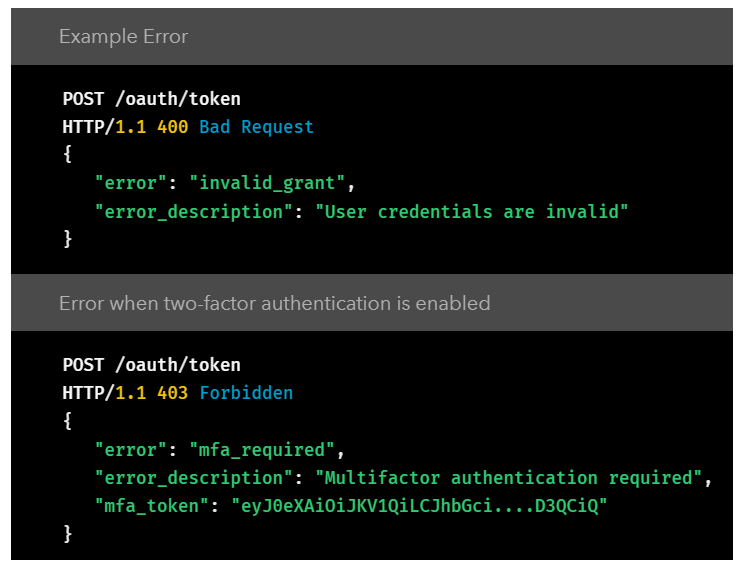
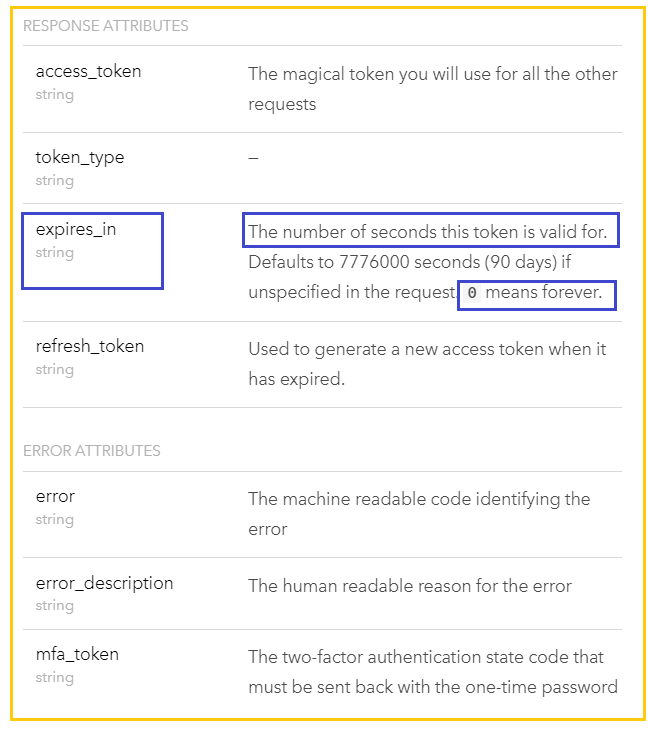
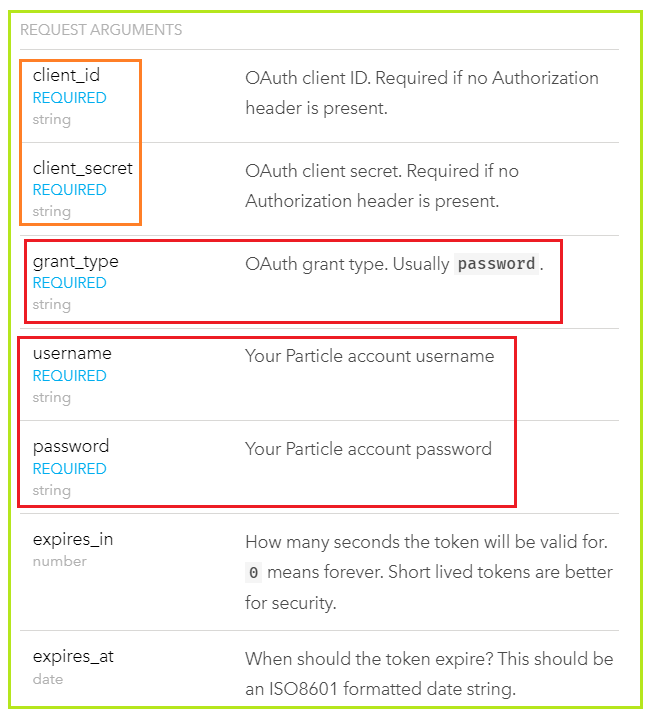
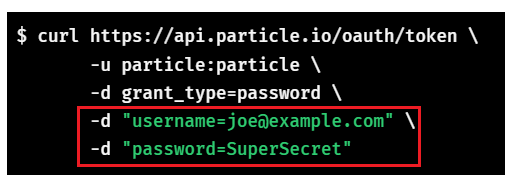
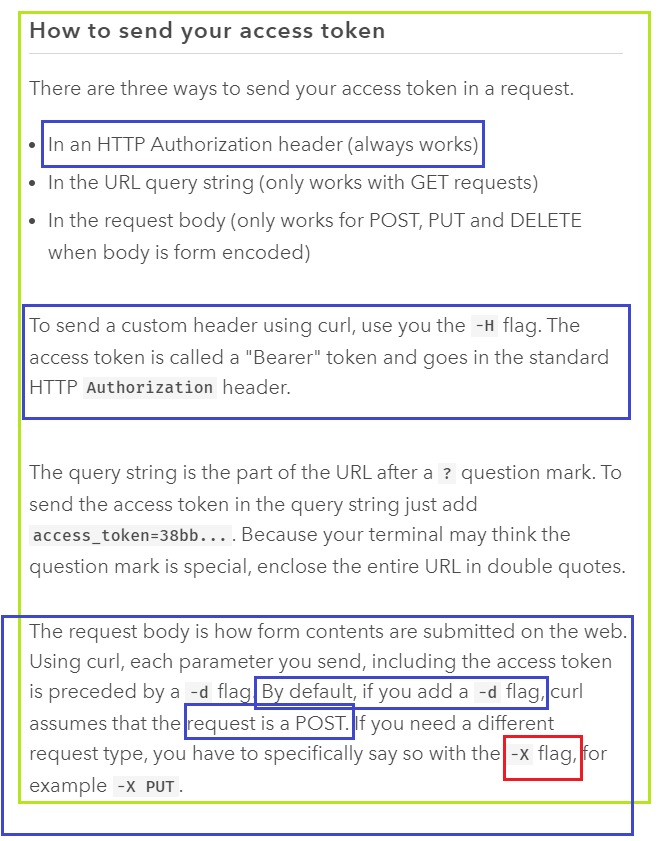
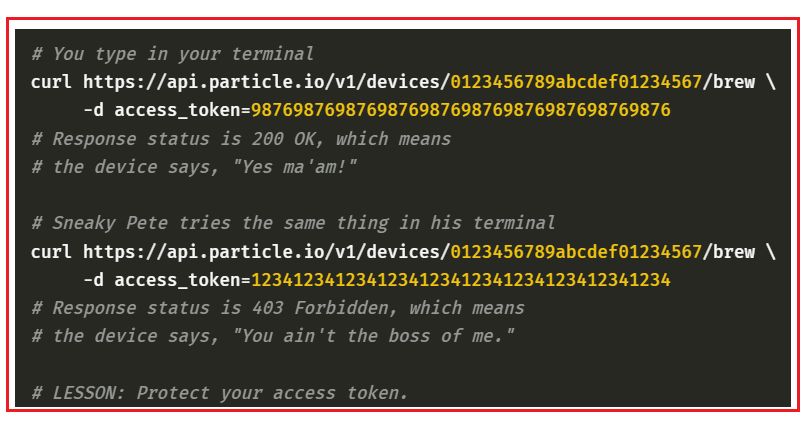
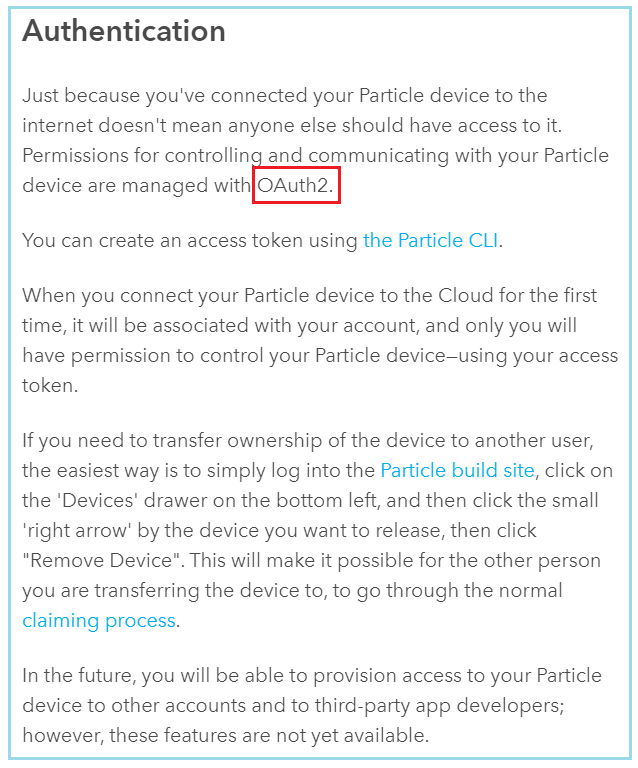
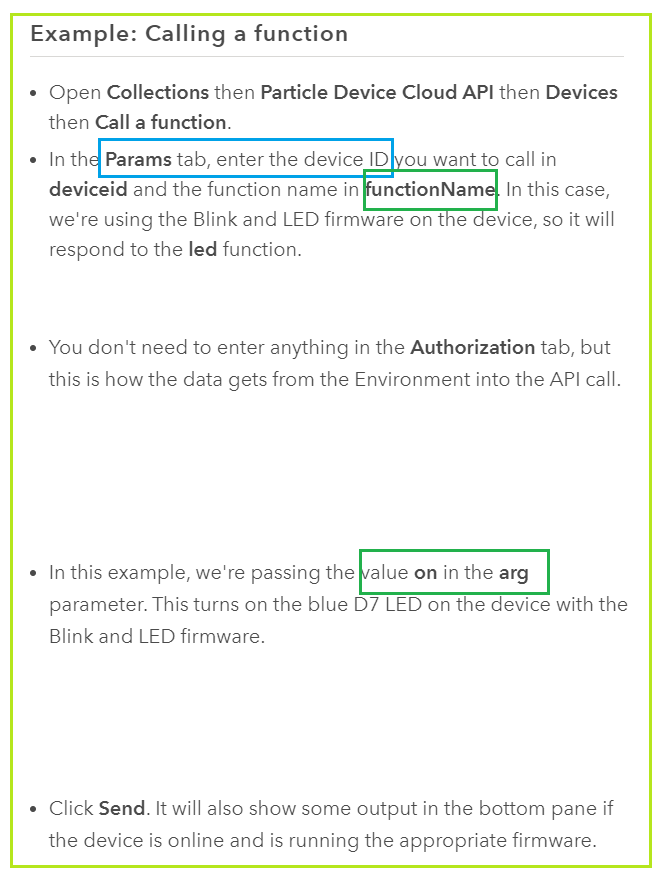
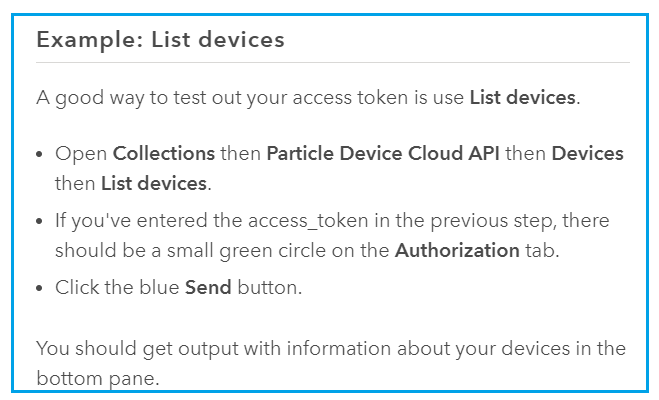
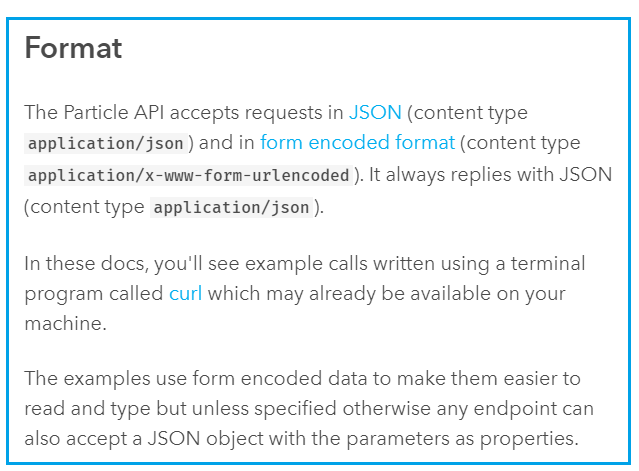
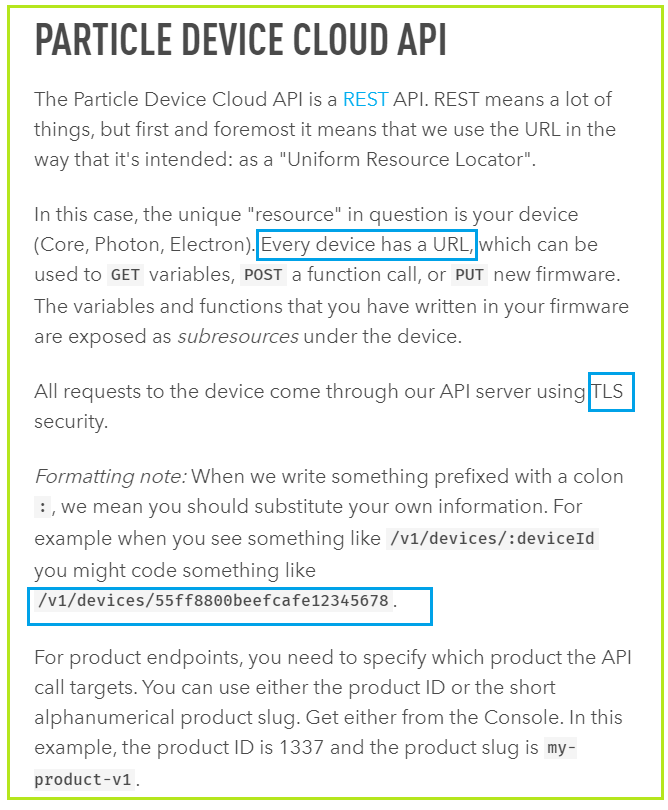
[**https://login.particle.io/login**](https://login.particle.io/login)

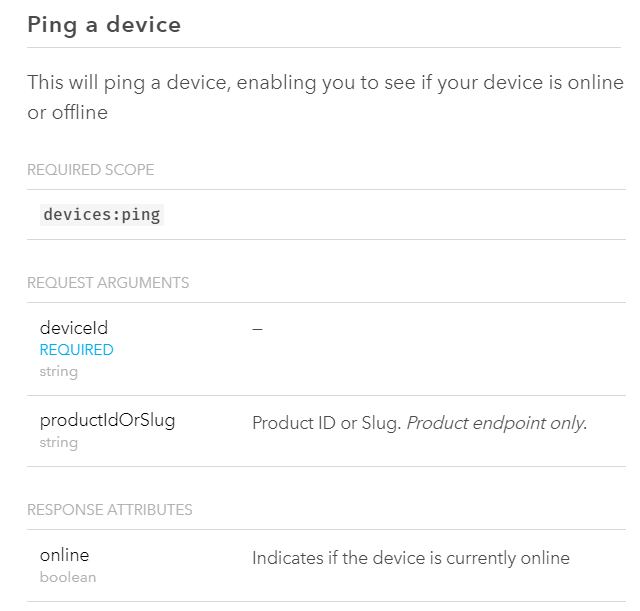
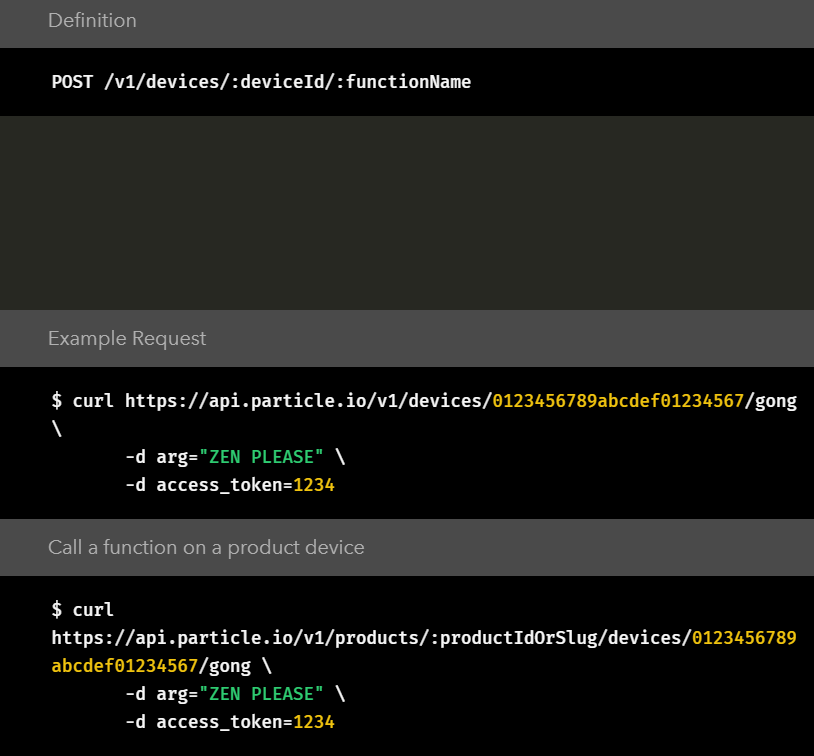
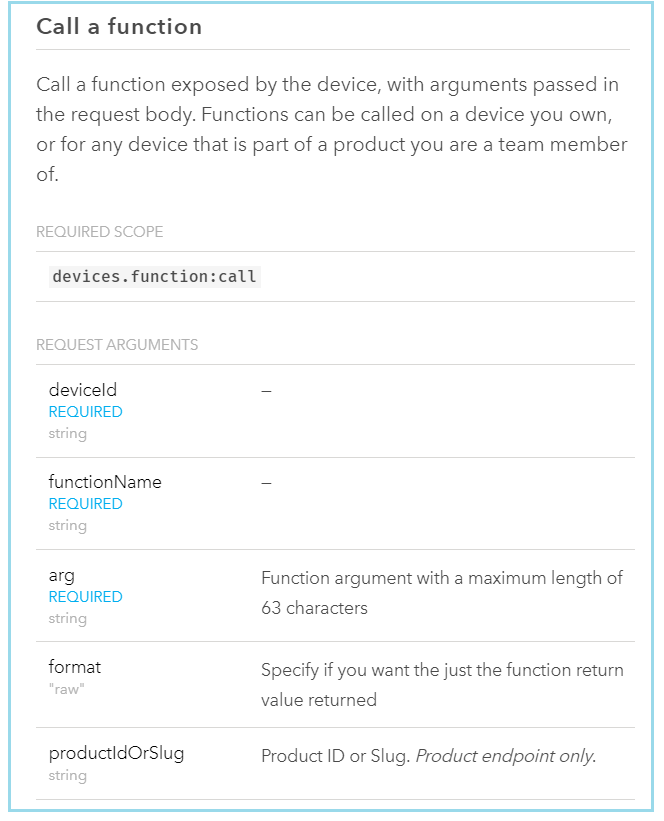
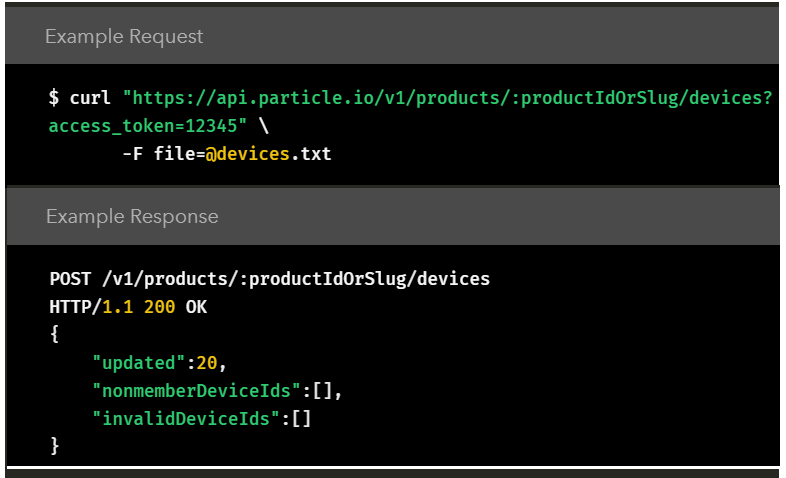
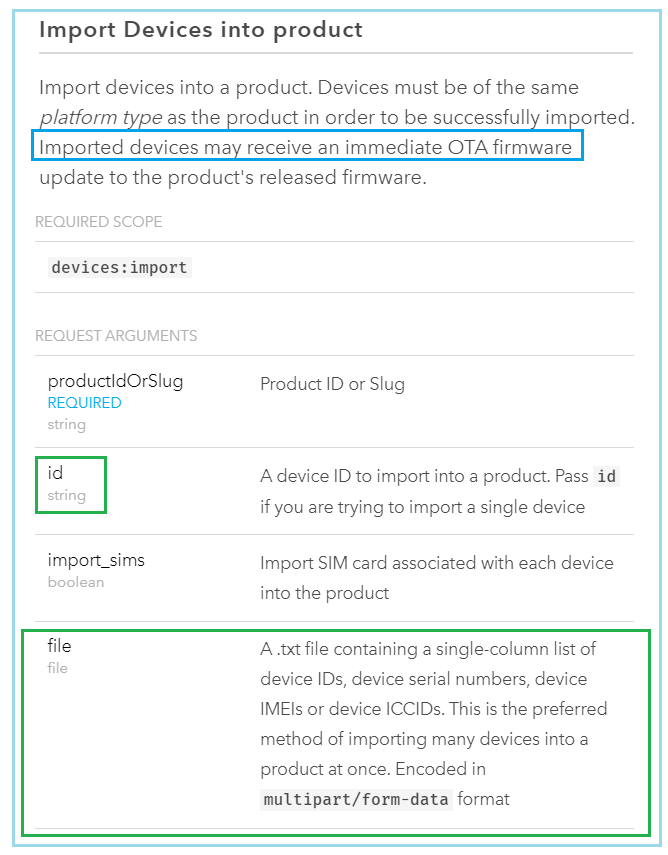
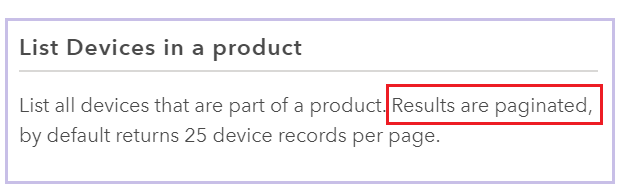
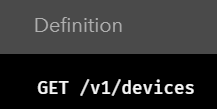
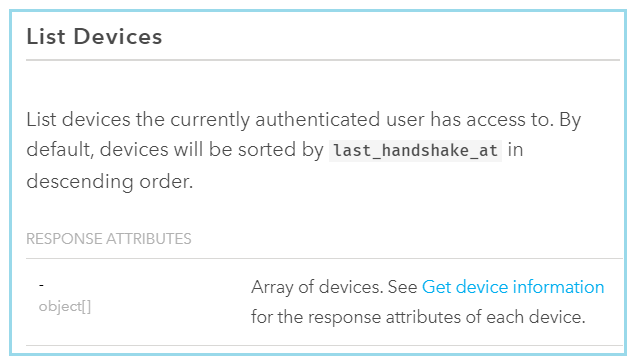
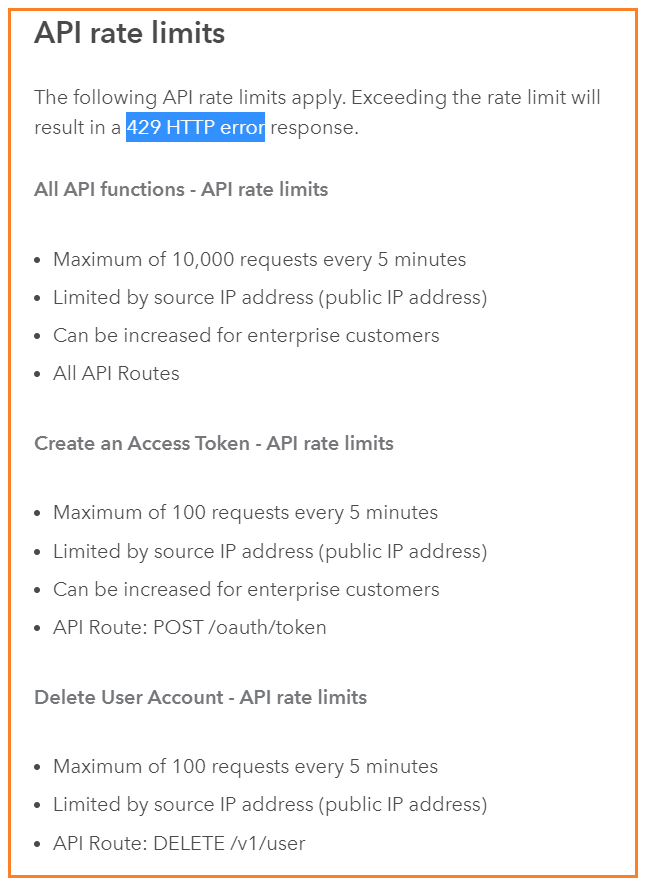
****

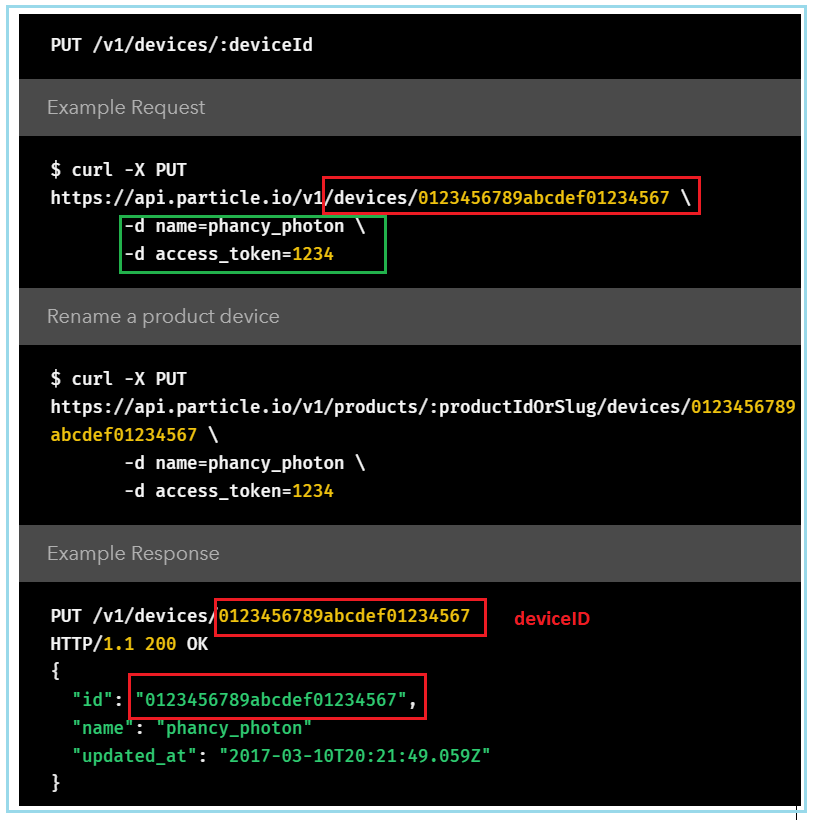
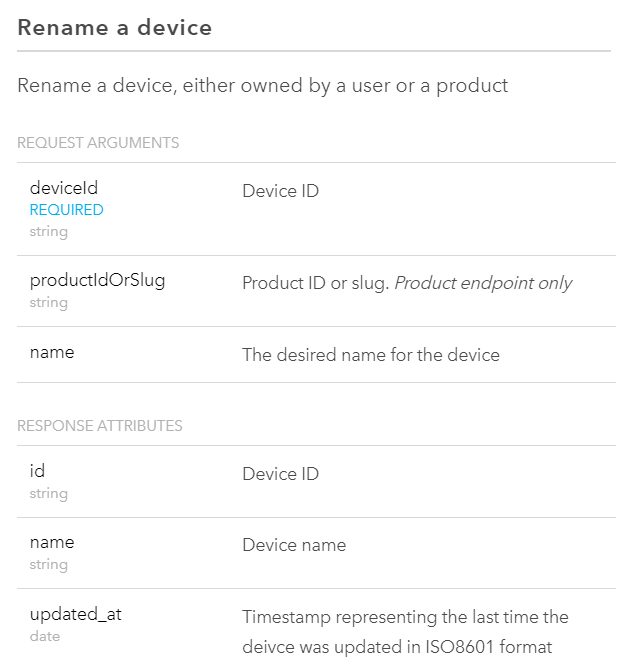
****

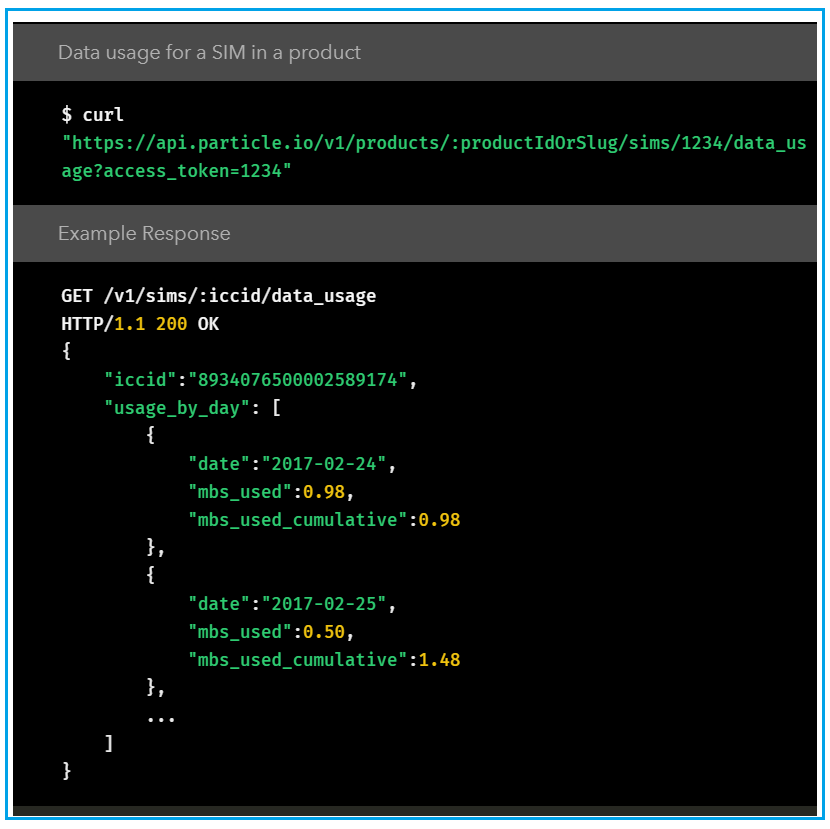
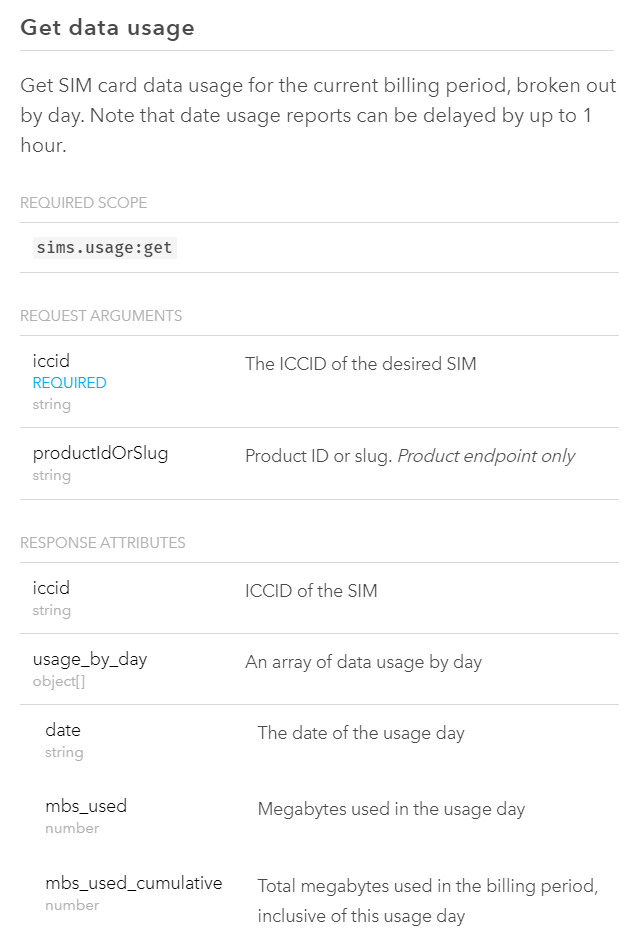
[**https://docs.particle.io/reference/device-cloud/api/#generate-an-access-token**](https://docs.particle.io/reference/device-cloud/api/#generate-an-access-token)

**PARTICLE DEVICE CLOUD API**

****

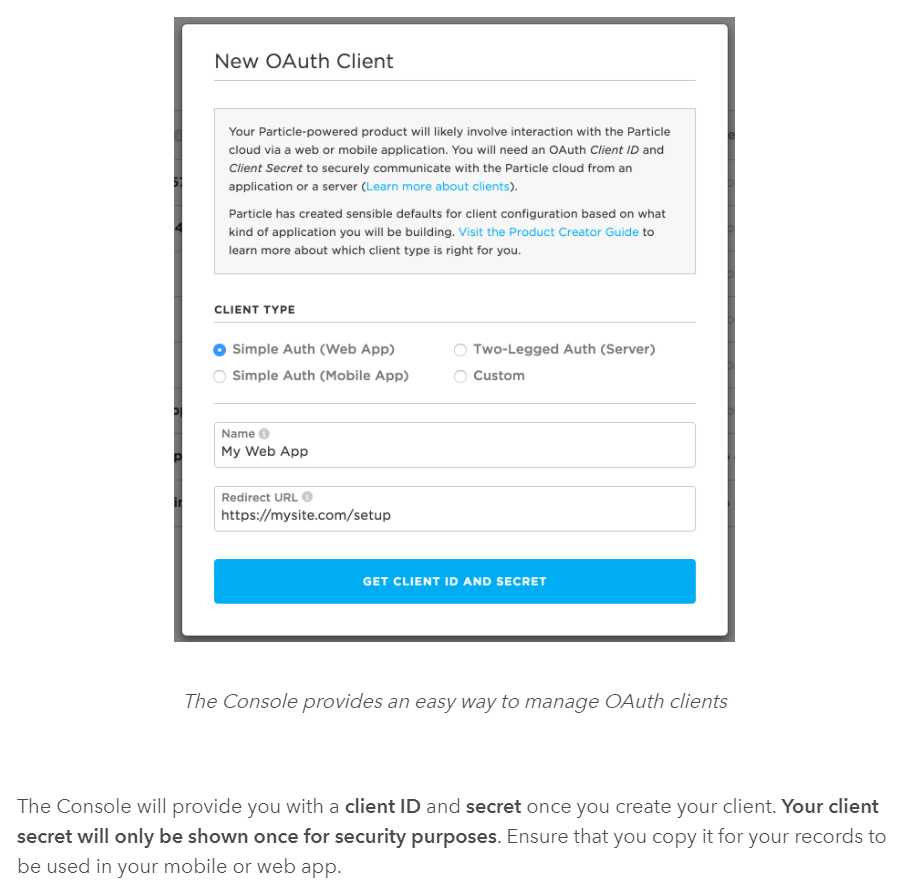
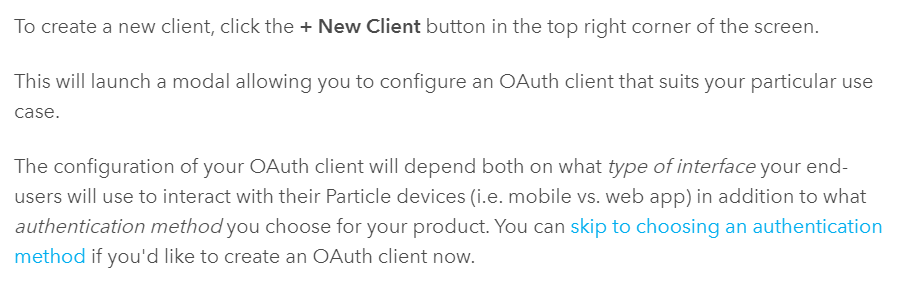
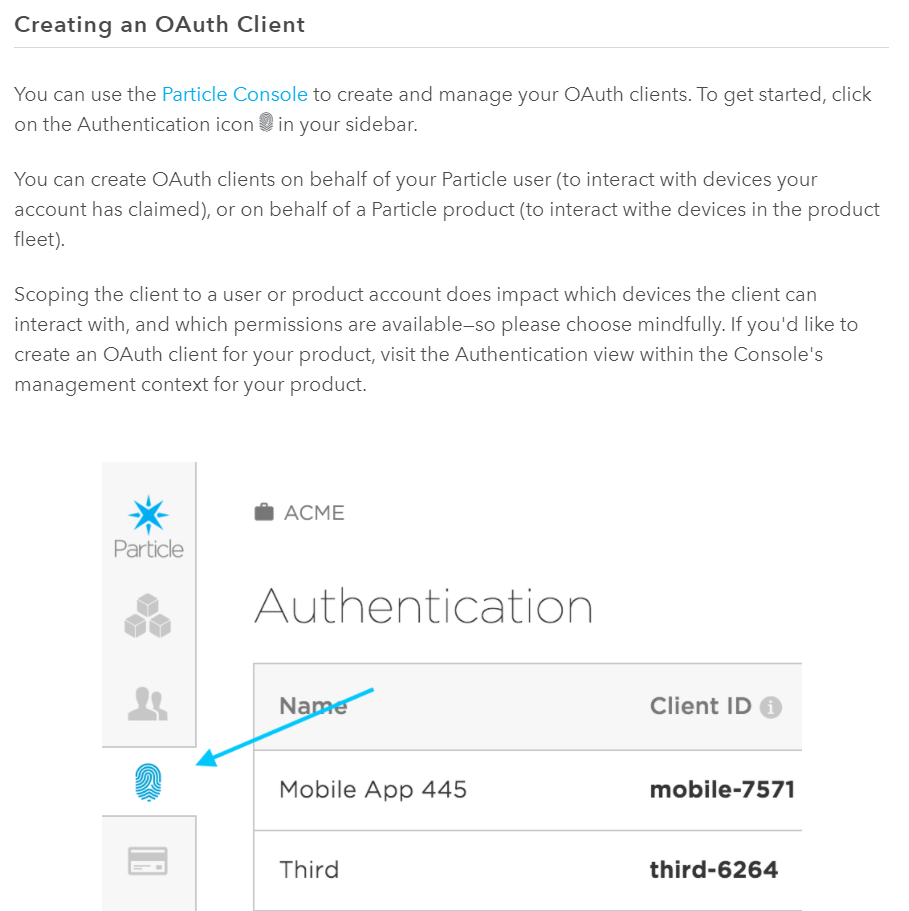
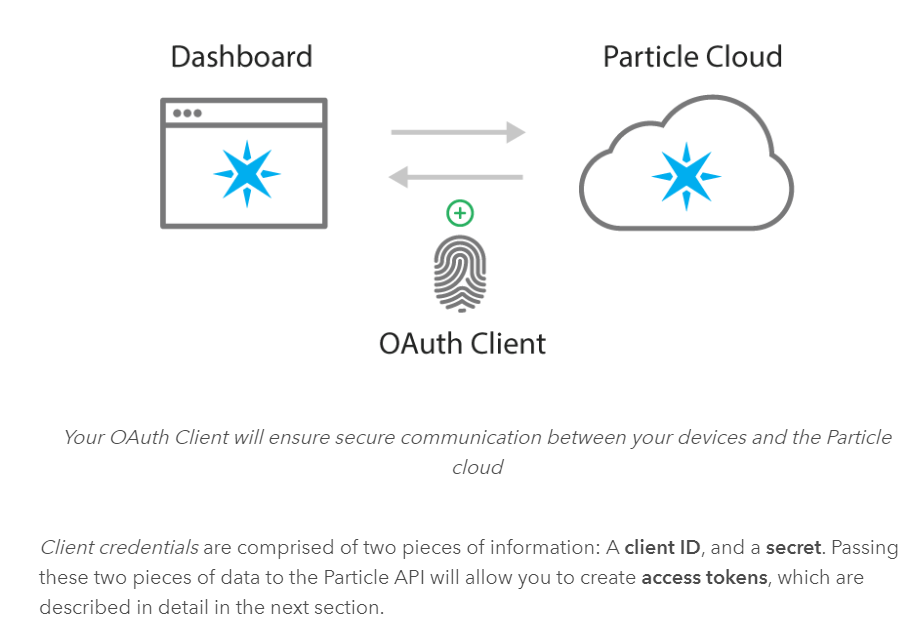
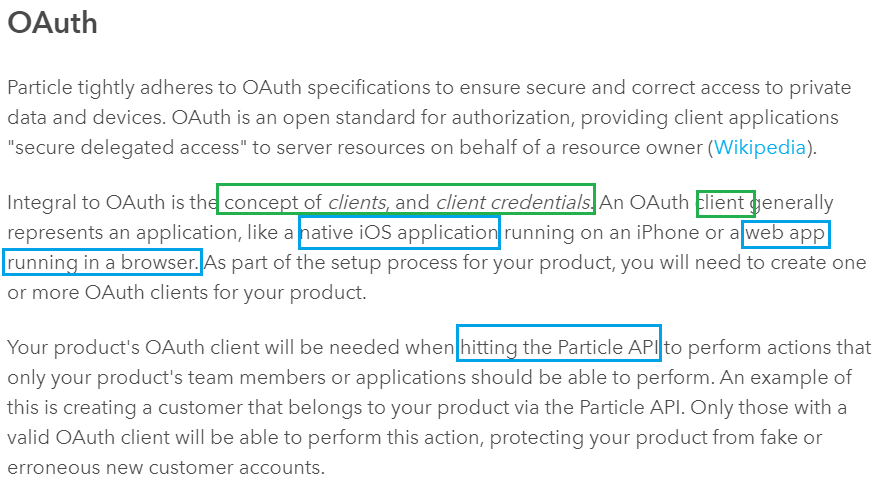
****

****

****

[**https://support.particle.io/hc/en-us/categories/360002801914-Knowledge-Base**](https://support.particle.io/hc/en-us/categories/360002801914-Knowledge-Base)

[**https://docs.particle.io/tutorials/device-cloud/authentication/**](https://docs.particle.io/tutorials/device-cloud/authentication/)

**Never expose your client credentials, especially if the client has full permissions**. Credentials are sensitive pieces of information that, if exposed, could allow unauthorized people or applications to interact with your product's data and devices.

Note: You may also manage OAuth clients programmatically, using the [*Device Cloud REST API*](https://docs.particle.io/reference/device-cloud/api/#oauth-clients).

## Choosing an Authentication Method

Take a deep breath. We've covered a lot so far and you're picking things up quick! This section will help you determine the best place for you to go next.

As described [earlier](https://docs.particle.io/tutorials/device-cloud/console/#devices-vs-product-devices), end-users of your product are referred to as customers in the Particle ecosystem. As a product creator, you will need to make a strategic decision about how you will want to handle authentication for your customers. Your choice will likely be influenced by **how much you would like to hide Particle from your customers**, as well as how you would like your product to function.

There are three ways to manage authentication for your customers.

* [**Simple Authentication**](https://docs.particle.io/tutorials/device-cloud/authentication/#simple-authentication): Customers are created and managed directly by Particle. This is the simplest and fastest way to get your connected product app working.
* [**Two-legged Authentication**](https://docs.particle.io/tutorials/device-cloud/authentication/#two-legged-authentication): You create and manage accounts for your customers yourself, and request scoped access to control customers' devices from Particle. This provides the maximum amount of flexibility and security for your application.

Typically you will use an iOS or Android mobile app to allow your customers to authenticate, setup their devices, and interact with their product. [More info](https://docs.particle.io/tutorials/device-cloud/authentication/)

Other techniques such as a web-browser based setup and on-device setup are possible, as well.

When you're ready, click on the authentication method that makes most sense to you.

### 

### **Advantages** of Simple Auth

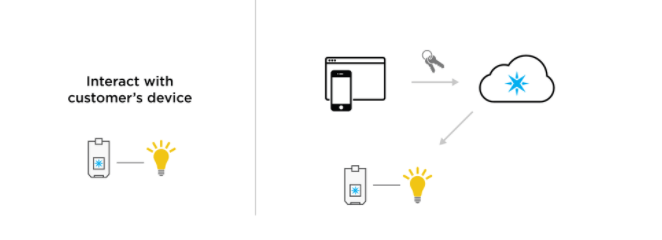
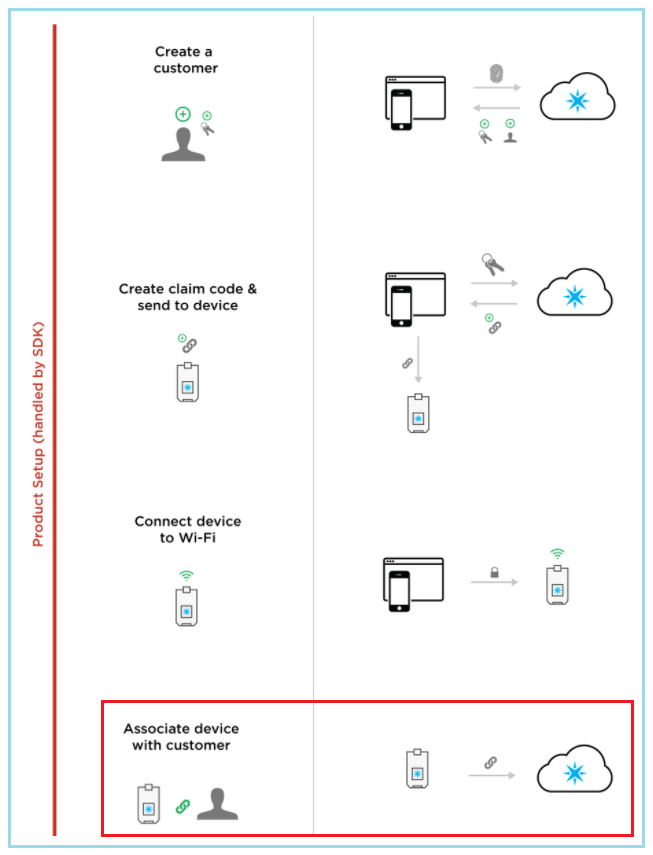
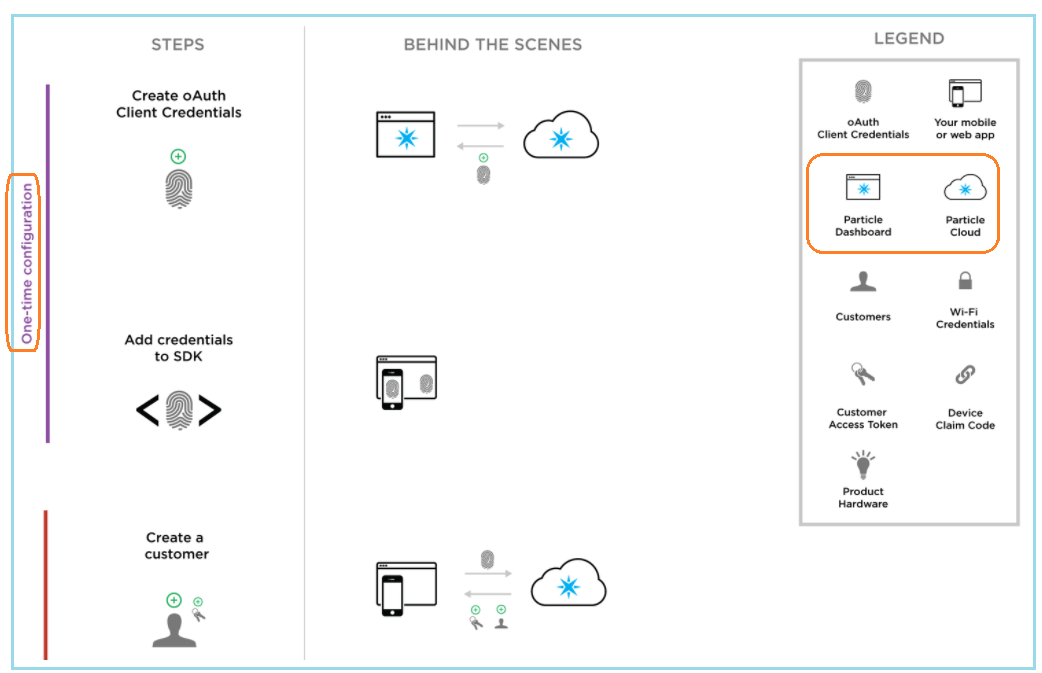
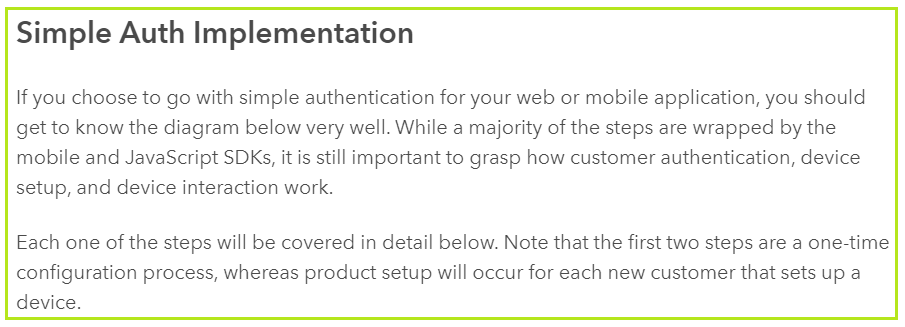
**Simple auth is ideal for getting a Particle product up-and-running quickly. Without needing to build your own back-end**, development time to creating an app to work with a Particle device is greatly reduced. There are less moving parts and opportunities to introduce bugs. In addition, Particle's [mobile SDKs](https://docs.particle.io/reference/SDKs/ios/) and [JavaScript SDK](https://docs.particle.io/reference/SDKs/javascript/) will handle much of the heavy lifting for you when it comes to session management and device interaction. In short, simple auth is...simple.

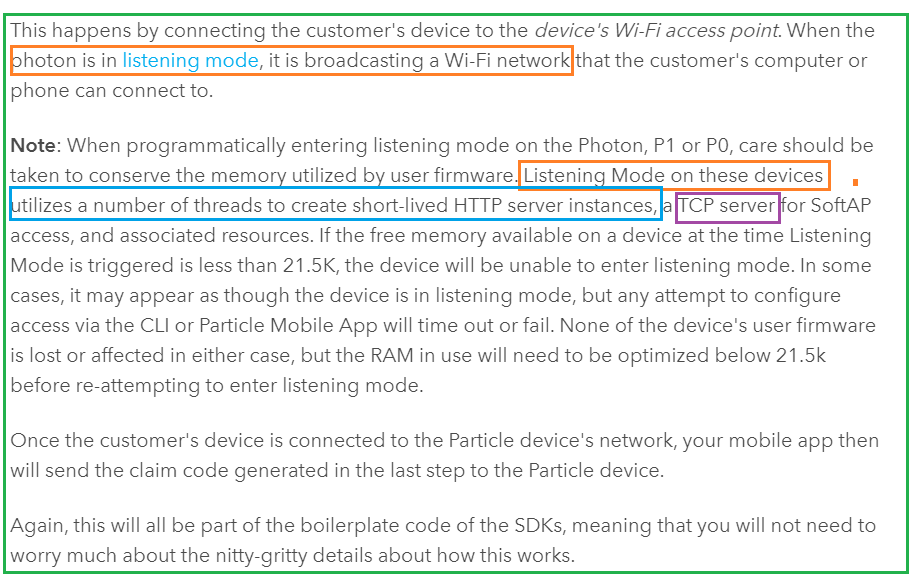
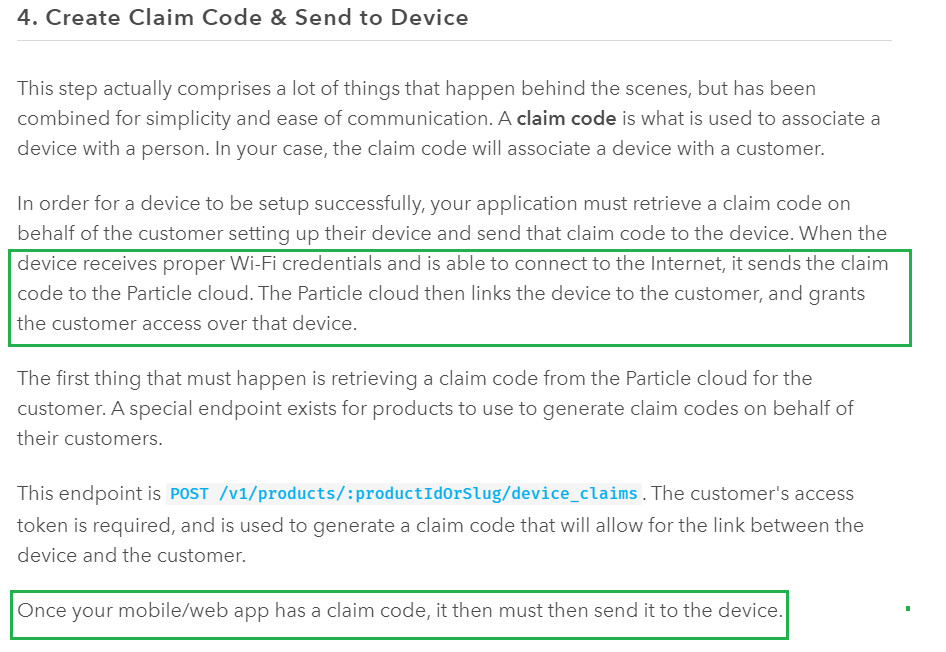
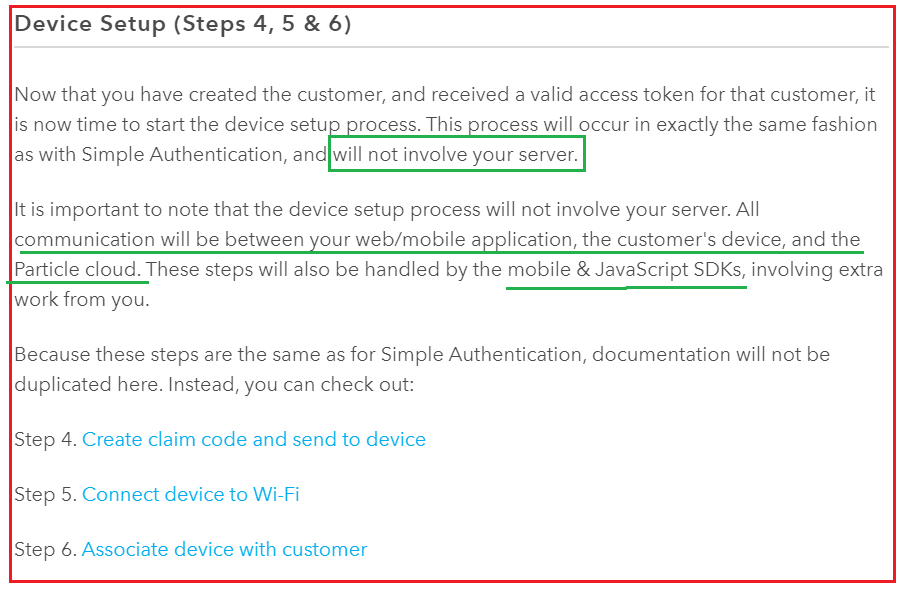
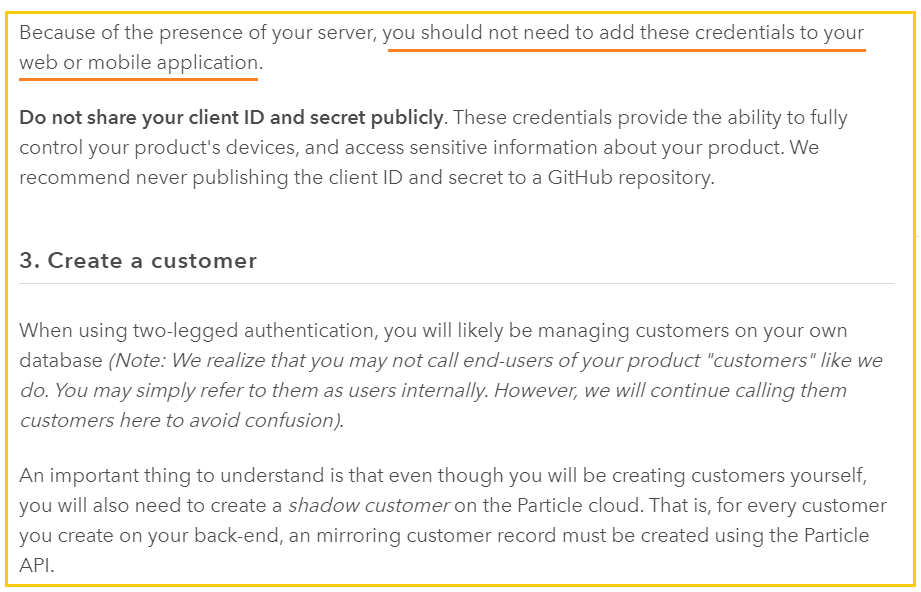
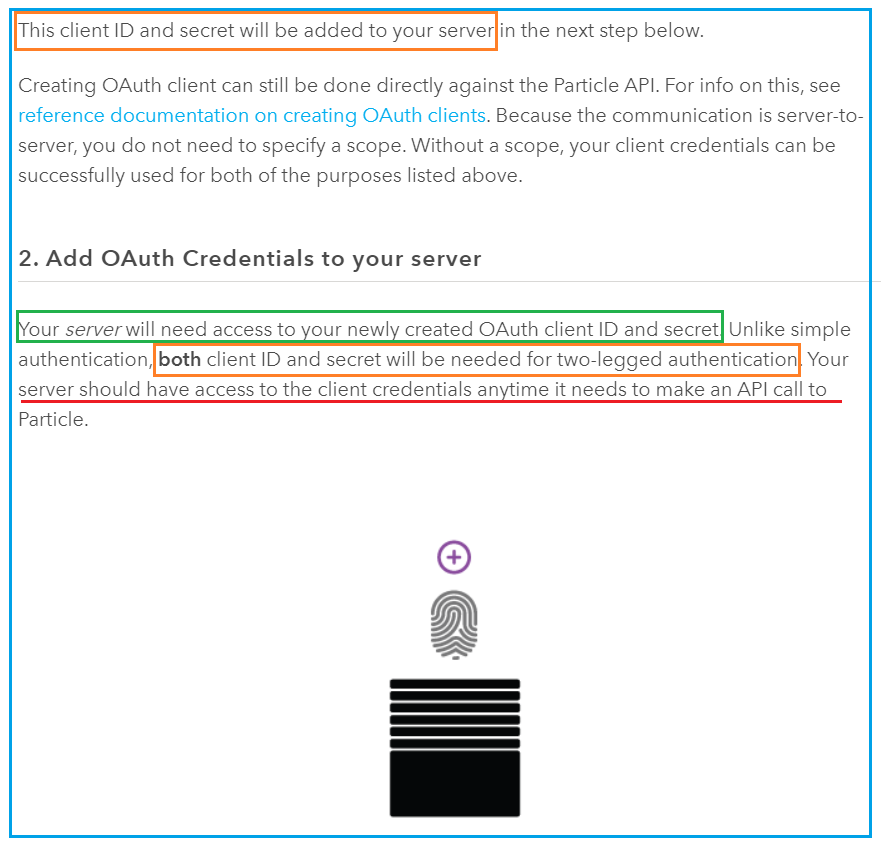
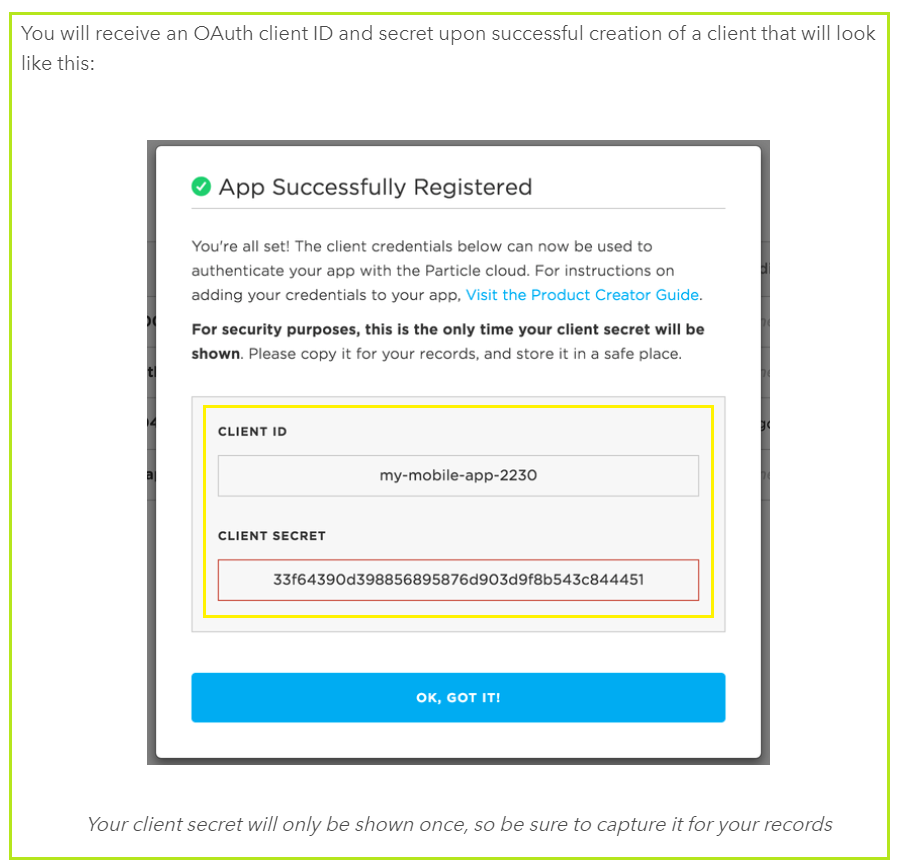
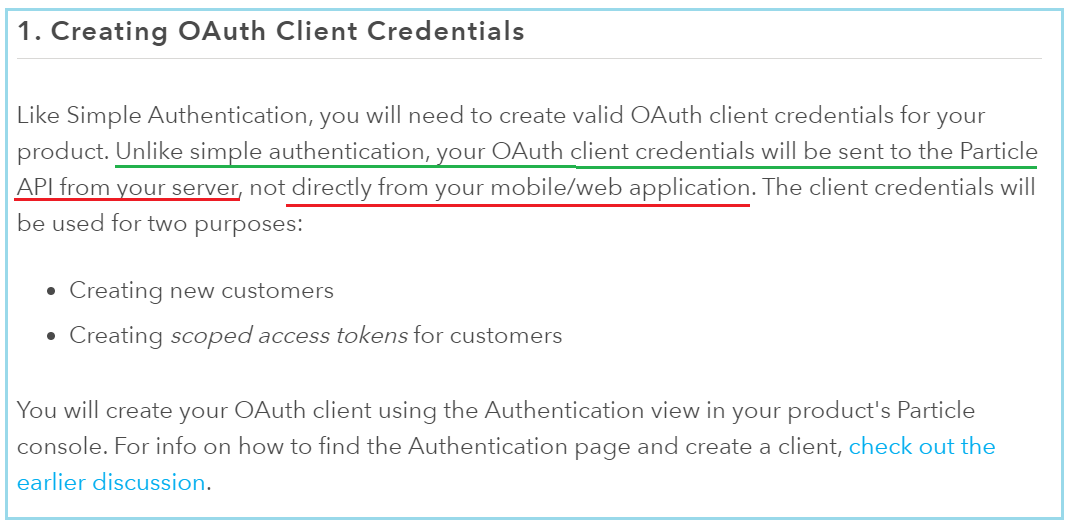
Another advantage of simple authentication is the ability to **hide Particle from your customers.** The SDKs allow for [front-end skinning and customization](https://docs.particle.io/reference/SDKs/ios/#customization) that will allow you to create your own brand experience for customers of your app. All interaction with Particle will happen behind the scenes, hidden from your customers (unless they are tech savvy enough to monitor the network traffic to and from your app).

### Disadvantages of Simple Auth

**Without your own server, you lose some level of flexibility and ability to customize in your application**. For instance, if you wanted to **store custom information** about your customer specific to your application like their name or their favorite pizza topping, this would not be currently supported with simple auth.

In addition, using simple auth would make it more difficult to capture and use historical data about devices and customers' behavior. **With your own server and database, you could store data about what time a customer turns on their lights,** for example. Using simple auth, this would not be supported.

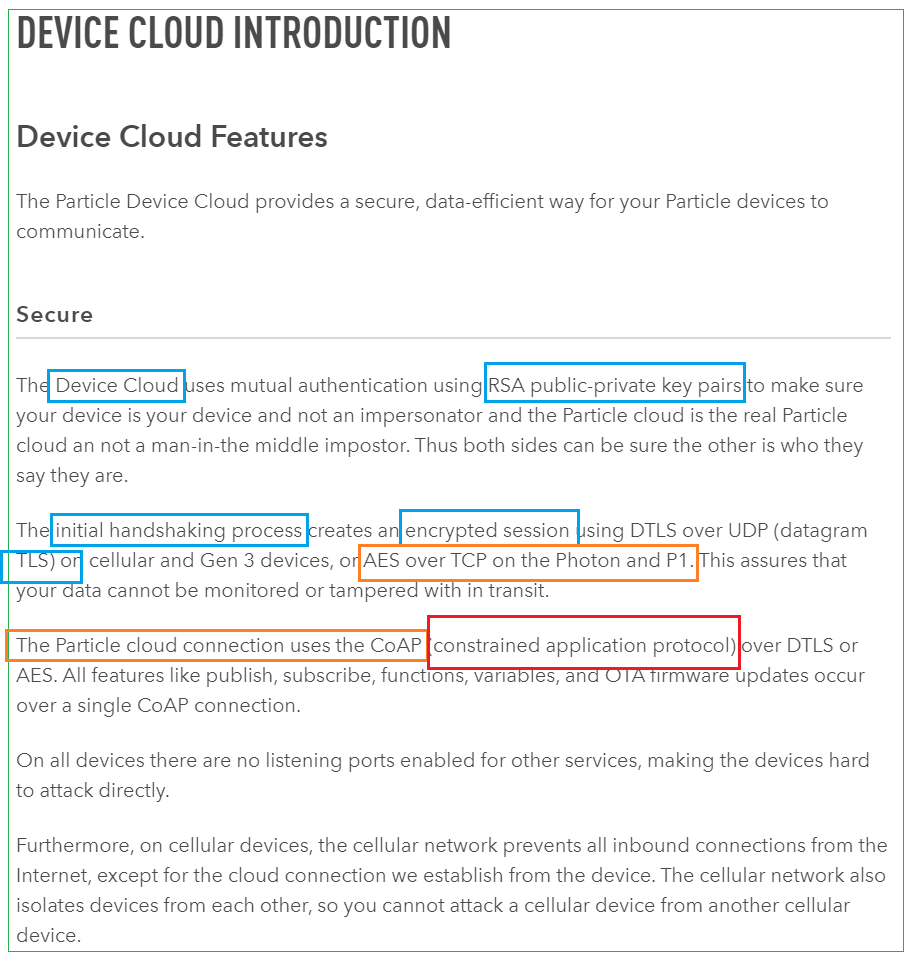
****

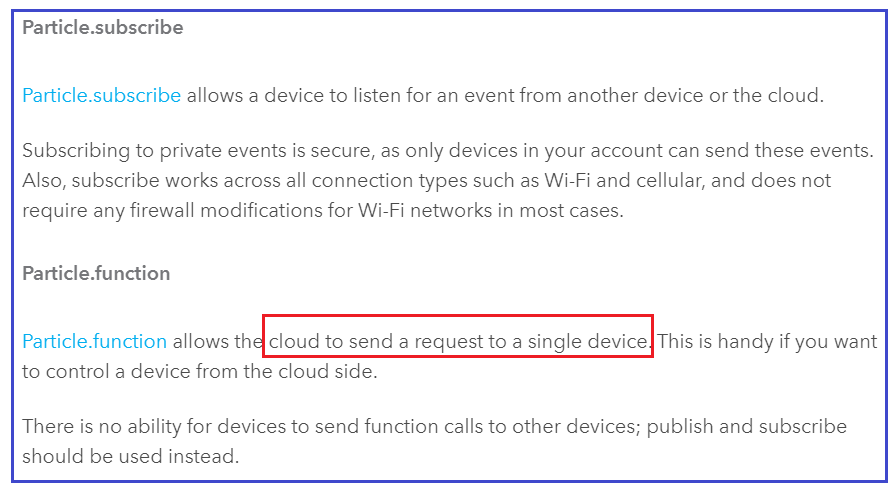
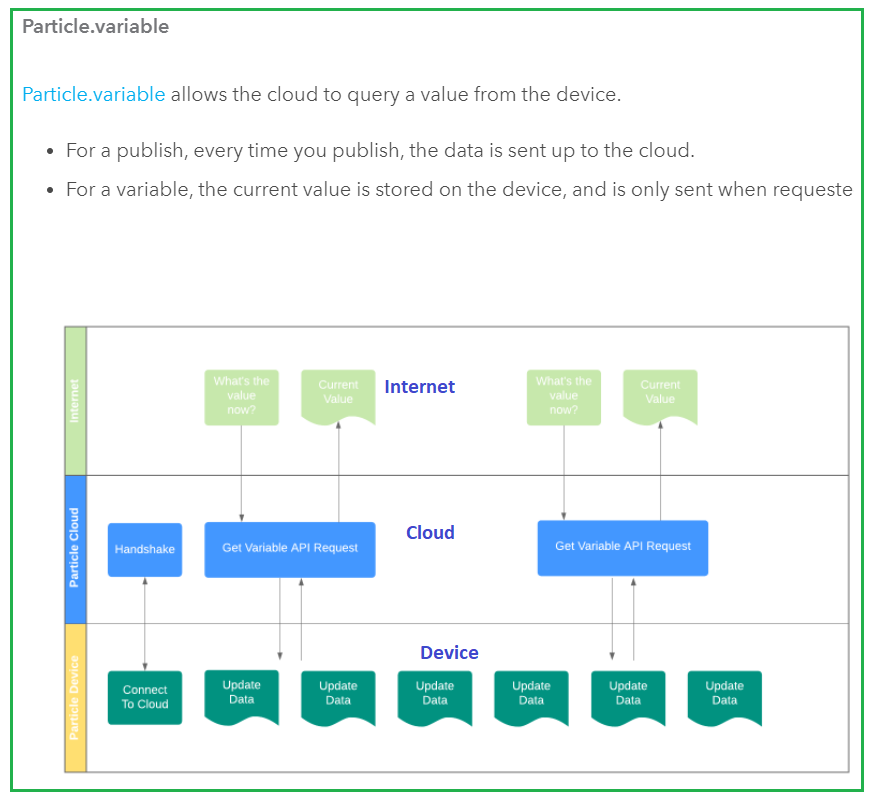
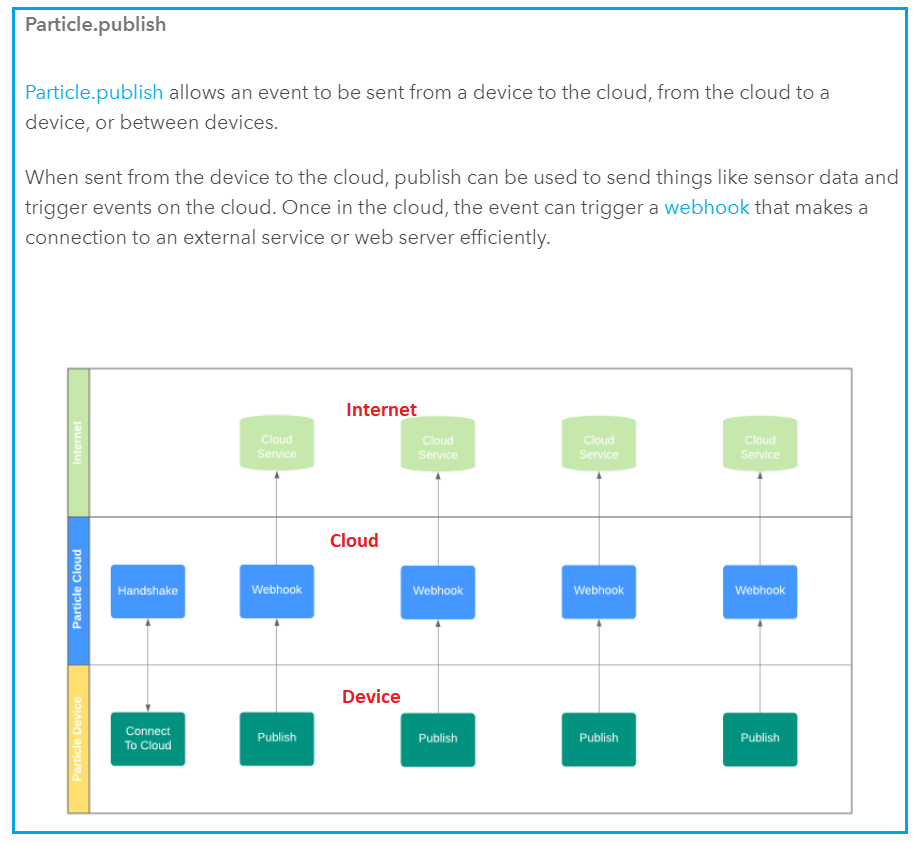
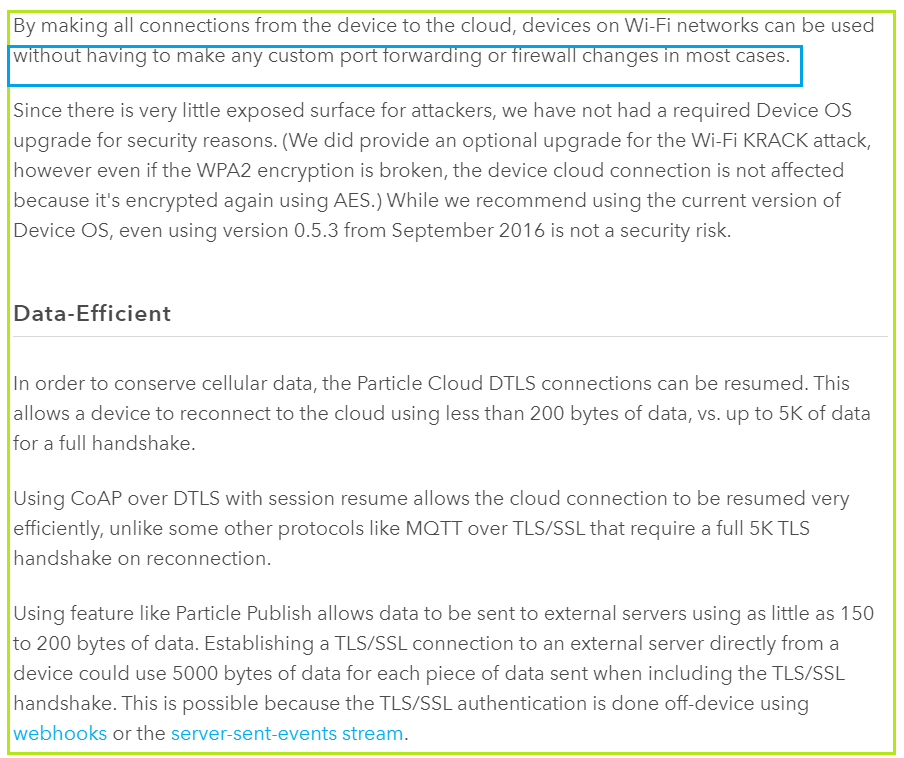
****

[**https://docs.particle.io/tutorials/device-os/led/photon/#**](https://docs.particle.io/tutorials/device-os/led/photon/)

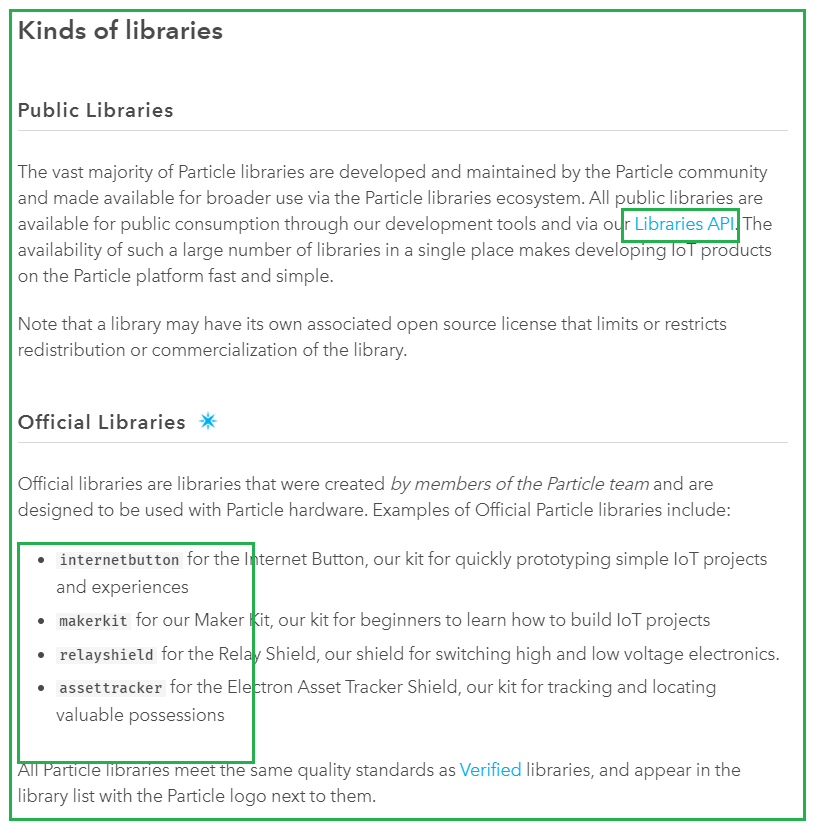
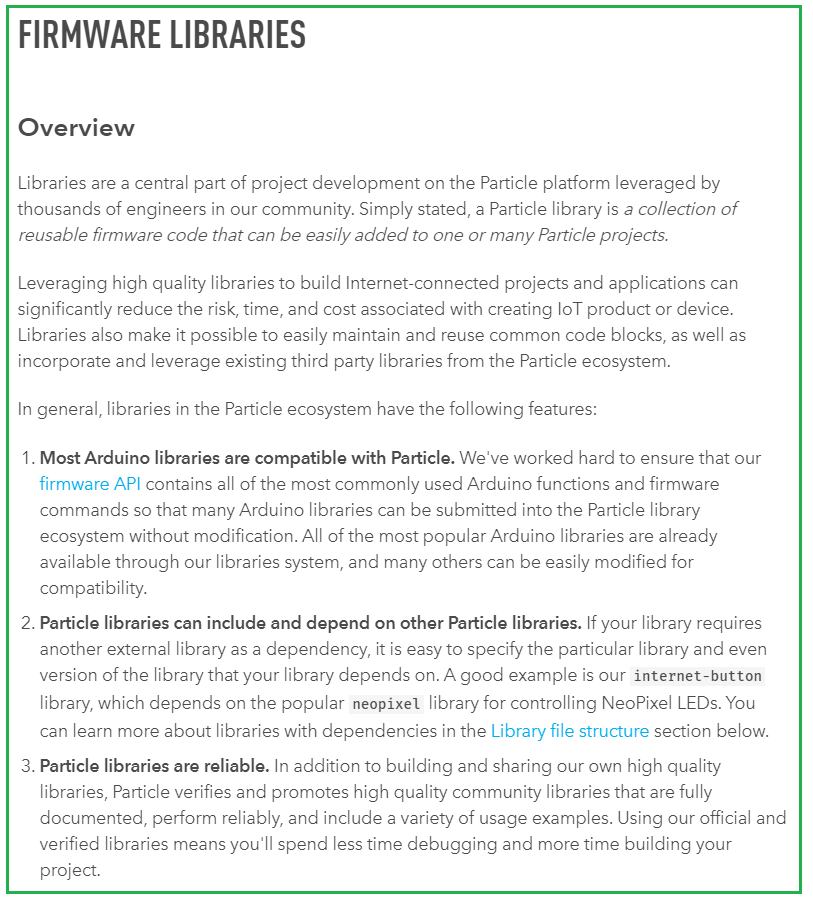
[**https://docs.particle.io/tutorials/device-os/cloud-communication/**](https://docs.particle.io/tutorials/device-os/cloud-communication/)

[**https://docs.particle.io/tutorials/device-cloud/introduction/**](https://docs.particle.io/tutorials/device-cloud/introduction/)

****

****

[**https://docs.particle.io/tutorials/device-os/libraries/**](https://docs.particle.io/tutorials/device-os/libraries/)

****