



MIT App Inventor and Particle.IO

Created by Richard Albritton

The screenshot shows the MIT App Inventor interface for a project named "Particle". The interface is divided into several sections:

- Blocks:** A palette on the left containing categories like Built-in, Control, Logic, Math, Text, Lists, Colors, Variables, Procedures, and Screen1. Under Screen1, there are sub-categories like VerticalArrangement1, ButtonOn, ButtonOff, and Web1.
- Viewer:** The main workspace where code blocks are assembled. It shows a sequence of "initialize global" blocks for variables like serverURL, deviceId, accessToken, Fun, Var, HTTP_Reply, and PINstate. Below these are two "when ButtonOn/Off Click" blocks that set the global Var to "on" or "off" and call a "SendToParticle" function.
- SendToParticle Function:** A detailed view of the function block, showing a "do" loop with "set Web1.Uri" and a "call Web1.PostText" block. The URL is constructed using "join" blocks to concatenate global variables and a "params=" parameter.
- Media:** A section at the bottom left with an "Upload File ..." button.
- Warnings:** A "Show Warnings" button at the bottom left of the viewer area, showing 0 warnings.

<https://learn.adafruit.com/mit-app-inventor-and-particle-io>

Last updated on 2023-08-29 03:08:11 PM EDT

Table of Contents

What is App Inventor?	3
The Design View	3
• The Designer	
Code Blocks	4
Set Global Variables	5
The Send Procedure	5
The Button Click	6
What Next?	7

What is App Inventor?



Making custom Applications for Android OS can be difficult unless you have a deep understanding of Java. MIT's App Inventor removes that barrier by providing an easy to use drag and drop style of coding to App development.

Most IoT projects need a user interface and while IFTTT and DO Buttons work well, sometimes you want something a bit more versatile.

This guide will take you through the steps to create a dedicated App for your IoT project using any of the Particle devices. We will focus on the simple Web Connected LED sketch from Particle, then point out the critical parts of the App Inventor code that will let you send and receive any information to and from the Particle device.

To get started with MIT App Inventor, go to the following link;

<http://appinventor.mit.edu/> ()

1. Once there, click on the [Create apps! \(\)](#) button in the top right of the page.
2. App Inventor uses your Google account for verification and will ask you to grant access to it. Don't worry, they won't sell off your information.
3. Click the Start new project button and we are all set.

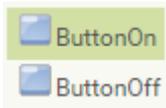
The Design View

This is set up to work with the Web Connected LED example app for the Particle Spark, Photon, or Electron.

App Inventor is an easy way to make Apps for the Android platform. We will go through how to use App Inventor for communicating with any Particle device.

The Designer

You will want to make two buttons. The button names do not matter, but we will just call them ButtonOn and ButtonOff.



Next we will add the Web component with the following attributes.



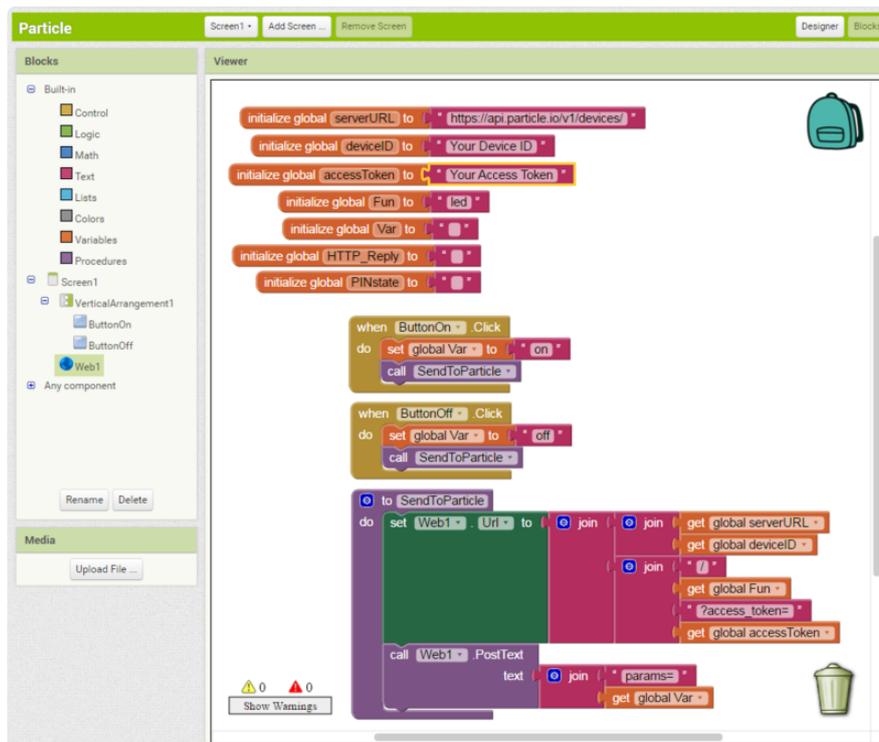
1. Check AllowCookies.
2. Enter `https://api.particle.io/v1/devices/` in the Url field.

Code Blocks

Now we get into the coding. The thing to keep in mind is that a Particle function can only be called with an HTTP POST request. App Inventor makes it easy to send GET data so we will have to jump through a few hoops to get there.

This green button below links to the file which may be imported directly into App Inventor.

Particle.aia on the Adafruit GitHub repository



Set Global Variables

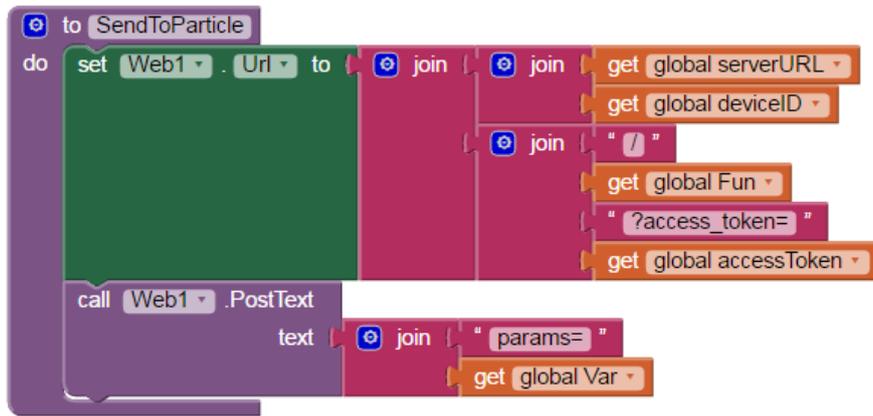
First, let's set up some global variables. These can be found under Variables and initialize globally. To this we will add a Text value to start with.



1. Make sure the serverURL is set to "https://api.particle.io/v1/devices/".
2. Copy the Device ID and use that for the deviceId variable.
3. Locate the Access Token from your Particle account and use that for the accessToken.
4. Fun should be set to the function name from the Particle code. For this instance, we are calling "led".
5. Var is a placeholder for the value we will be sending to the function. It is blank.
6. HTTP_Reply can be used to get a confirmation message from the Particle device.
7. PINstate can be used later to toggle the On/Off button.

The Send Procedure

In App Inventor, a Procedure is a block of code that can be called on to run at any time. This saves time and space when you need to run the same code over and over again.



1. From the Procedures blocks, add the Do block.
2. Name this procedure "SendToParticle".
3. Add the Set Web1 block from the Web1 section to the new procedure.
4. Now we will assemble a string for the set Web1.Url by using the Join block from Text.
5. All together we will be joining the serverURL, deviceId, "/", Fun, "?access_token=", and accessToken.
6. Under the Set.Url block, we will add a Call.PostText block from the Web1 section.
7. To this we will add a Join block for sending to our Particle function via POST request.
8. We will join "params=" and the Var value.

When ever this procedure is called Val will be sent via POST to:

https://api.particle.io/v1/devices/your-device-id/led?access_token=your-access-token

The Button Click

For now we will just use one button for ON and one for OFF. So we want to grab the when.click from the ButtonOn and ButtonOff blocks.



1. To that we will add a Set block for theVar variable. Also add the value of “on”to this.
2. Next we will be calling on the SendToParticle Procedure.

That is it!

Once this App is loaded onto your phone it will turn the built-in LED, attached to Pin 7 on the Particle device, on or off.

What Next?

Well this is a good place to get you started. From here you can try to make the button toggle the LED on or off.



Change the button into an image to make things look a little nicer.

Add a Relay to Pin 7 on the Particle device and you can turn an actual light on or off.

The possibilities are nearly endless and this will help get you well on the road to making it easy to use the next big IoT thing.