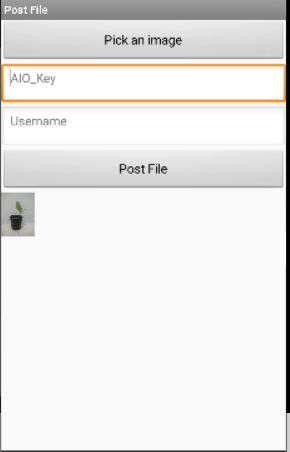
Posting Plant image to AdafruitIO:

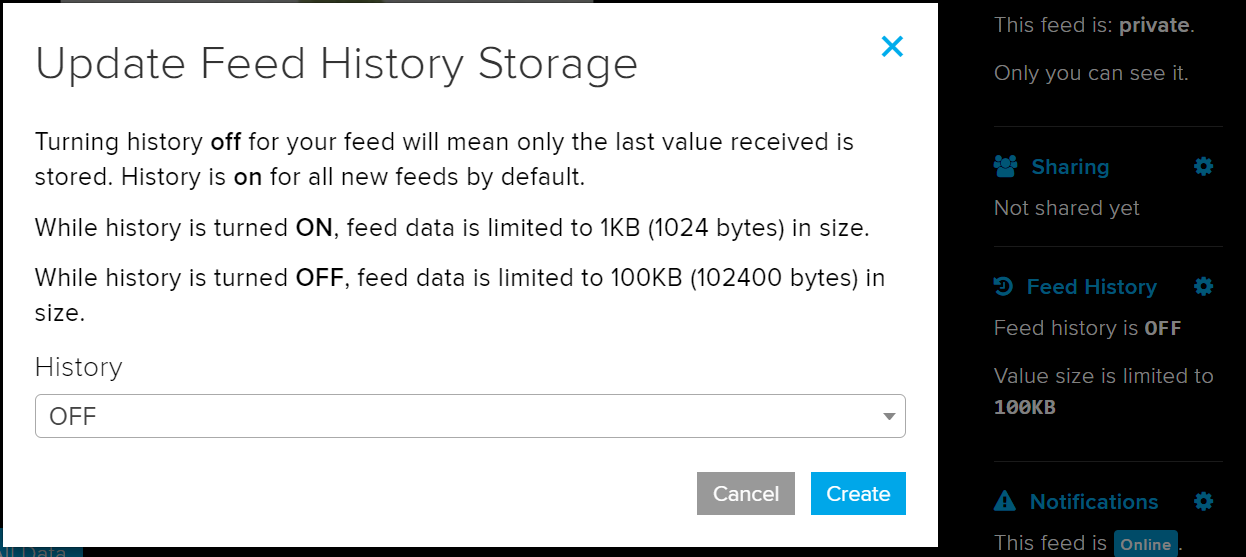
Plant image block App screenshot

We can create an android App in MIT’s APP Inventor platform to select an image we have captured from the camera and upload it the image feed in AdafruitIO.

Using an Image Block on an Adafruit IO, you can automatically display a Base64 image data string on your dashboard by sending a Base64 image data string to an Adafruit IO feed.

For a detailed description on base64 checkout: https://stackabuse.com/encoding-and-decoding-base64-strings-in-python/

Normal feeds are limited to 1KB of data, or about 1024 bytes, for publishing. Turning off feed history from the feed settings page allows publishing up to 100KB, or 102400 bytes, of data. 1KB is too small for an image, so let us create a image feed and then click on it, on the right there is option to turn OFF feed history so we publish data upto 100KB.

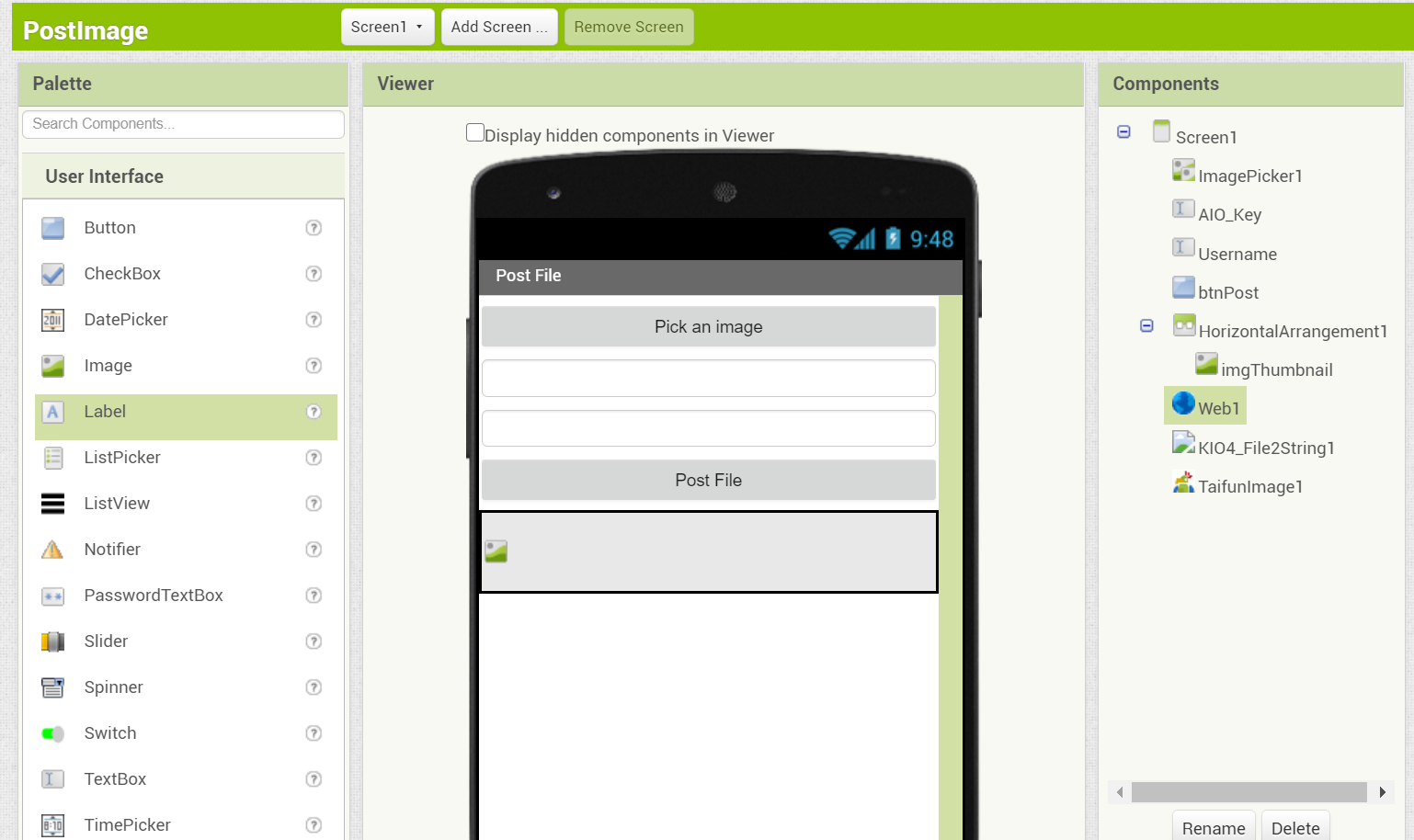


Mit APP inventor let us make an android APP with block based functions instead of codes, thus this can help us quickly make a simple APP.

Go to <https://appinventor.mit.edu/> and select “create APPs”, login or register.

The APP inventor block based coding environment can be easily be understood by someone who has prior experience in Python or C++. Nevertheless you still can still have look at <https://appinventor.mit.edu/explore/ai2/tutorials> for tutorials.

UI Elements



In the above application there is an image picker element to select the image to be uploaded from the gallery.

Text boxes to input the AIO key and Username.

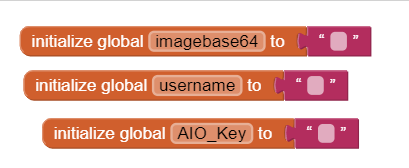
A button to post the Image to AdafruitIO and an Image element which displays the image we have selected.

From the connectivity tab we can add the Web element which lets us perform POST request to POST data to adafruitIO.

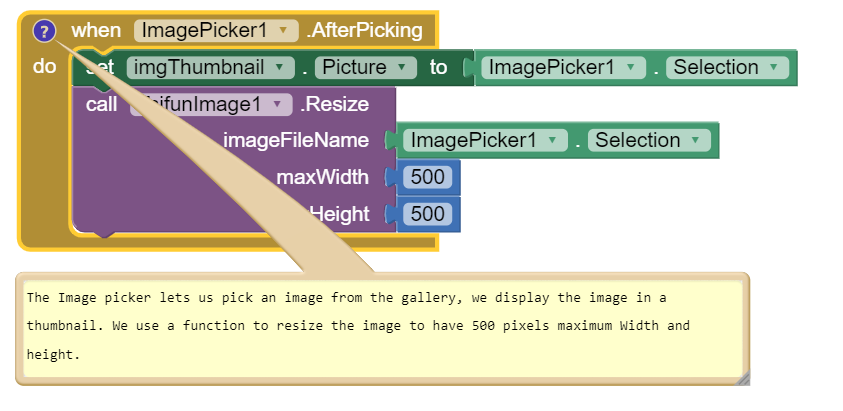
There are also two additional elements which are not present in APP inventor by default, these can be added by going to the import extensions tab.

File2String is used to convert the image to a base64 string and TaifunImage is used to resize the selected image so that it is less than 100KB in size.

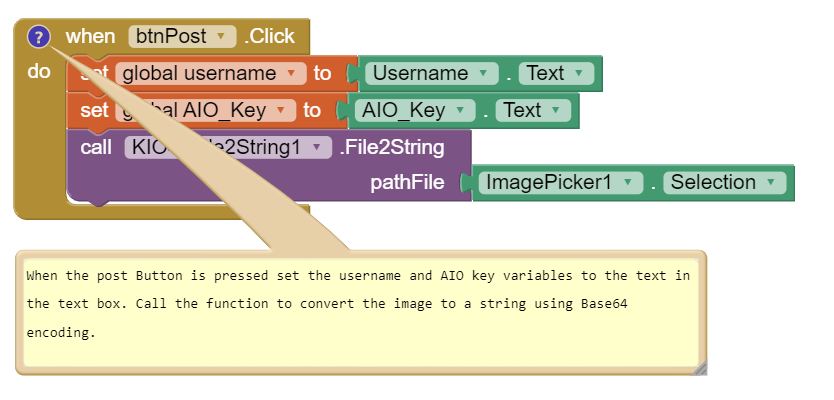
In the Block Section:



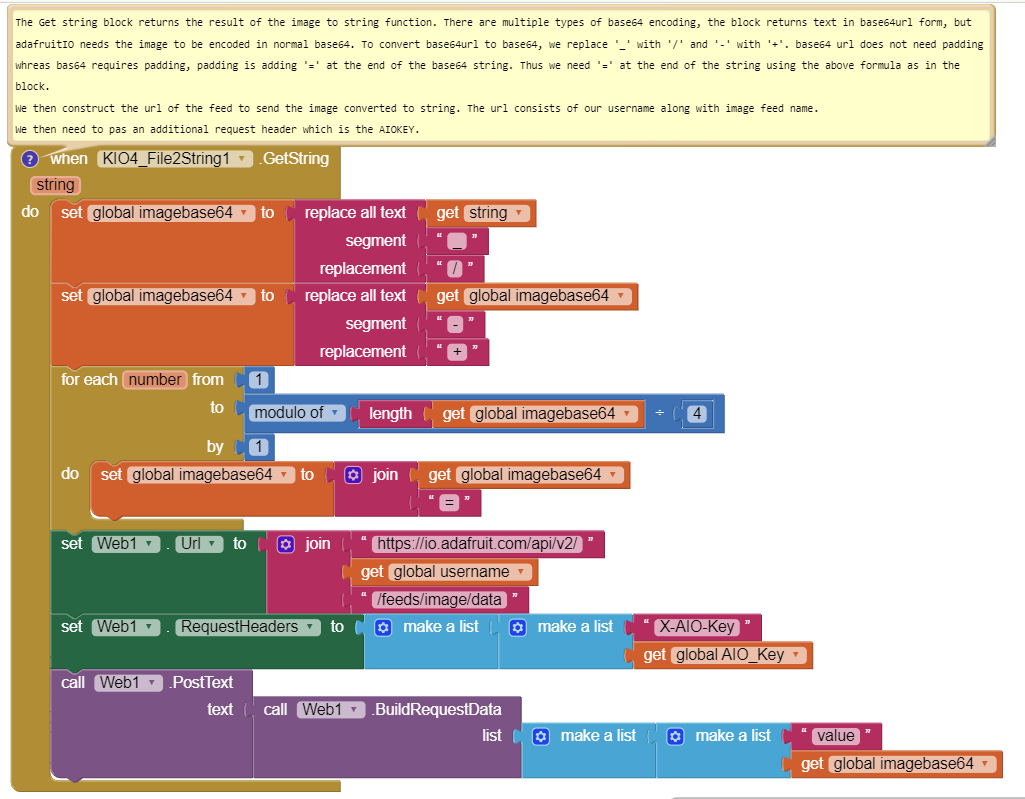
We create 3 empty variables for the base64image, username and the AIO\_KEY



From the image picker element, we can select the after picking block to update the image in the UI with selected image from the gallery. We also resize the image.



From the button element, we select the button click block. The variables can be set to the username and AIO Key values instead of the textbox text so that the username and AIO Key do not have to be entered each time the App is launched.



The Get string block returns the result of the image to string function. There are multiple types of base64 encoding, the block returns text in base64url form, but adafruitIO needs the image to be encoded in normal base64. To convert base64url to base64, we replace '\_' with '/' and '-' with '+'. base64url does not need padding whereas bas64 requires padding, padding is adding '=' at the end of the base64 string. Thus we need '=' at the end of the string using the above formula as in the block.

We then construct the url of the feed to send the image converted to string. The url consists of our username along with image feed name.

We then need to pass an additional request header which is the AIOKEY.

From the build tab the app can be downloaded as an apk or you can download the app inventor companion app on your phone to view a preview of the app. The app can be now used to post images to AdafruitIO

Files