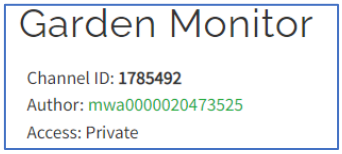
Challenge 1

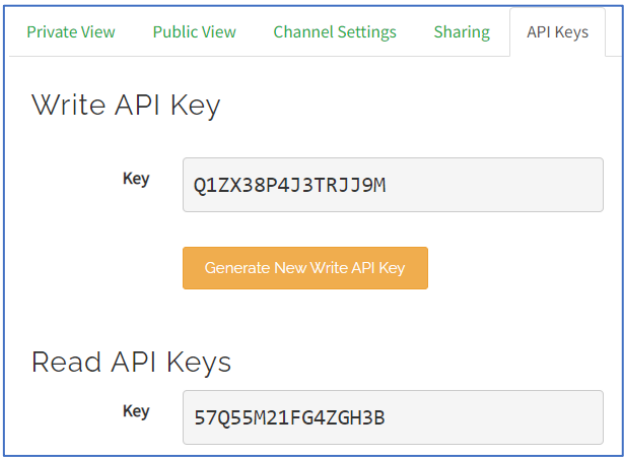
Setting Up Email Alerts

Step 1: Find your channel ID API keys

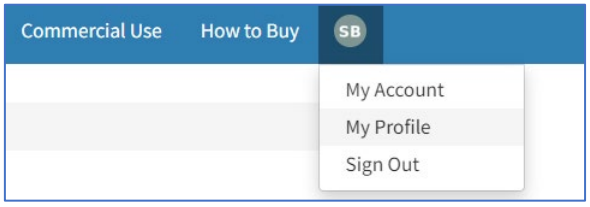
For this challenge activity, you will need your channel ID and the two keys that will help with reading data and sending emails. If you haven't already, copy down your channel ID:



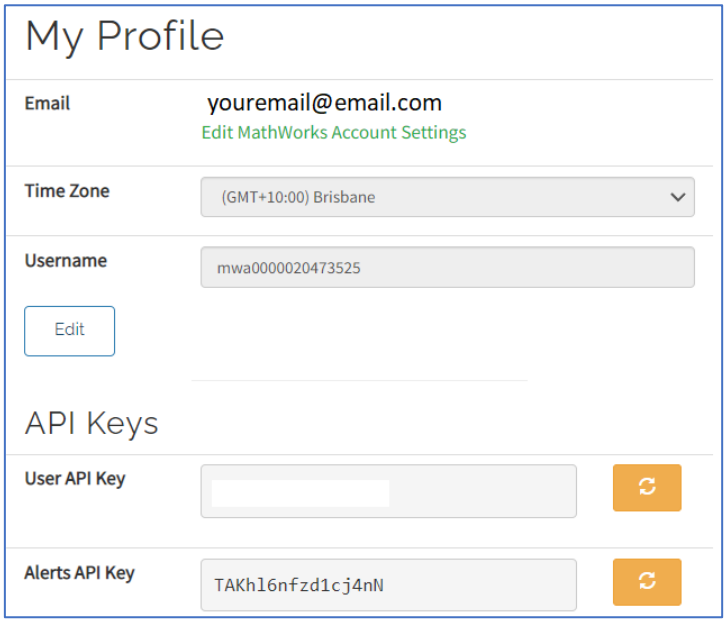
The first one is the Read Key for getting data from your channel. Navigate to the API Keys tab and make a note of the Read Key (I suggest copying it to a digital notebook)



The email alert key is a little harder to find, click on the little circle in the top right corner to navigate to 'My Profile'.

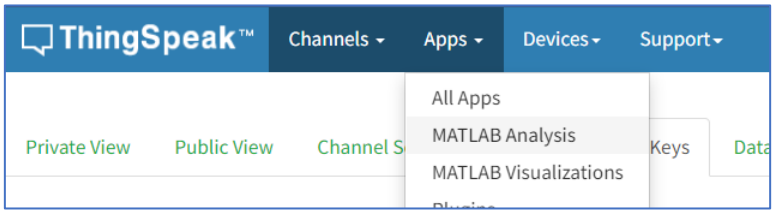


You should now see a screen that looks something like the following image. Copy the 'Alerts API Key' into your OneNote or Word file. If there is no 'Alerts API Key' when you open this page, just click the little orange refresh button next to it and a key should appear.

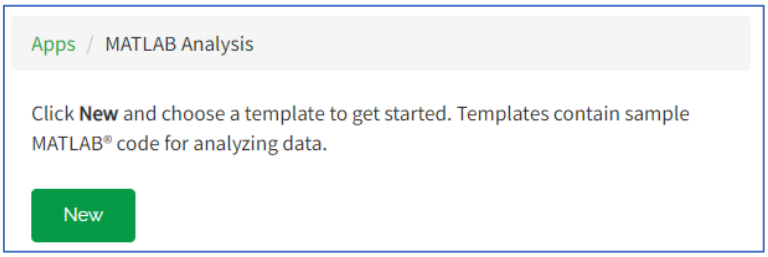


Step 2: Send a basic email

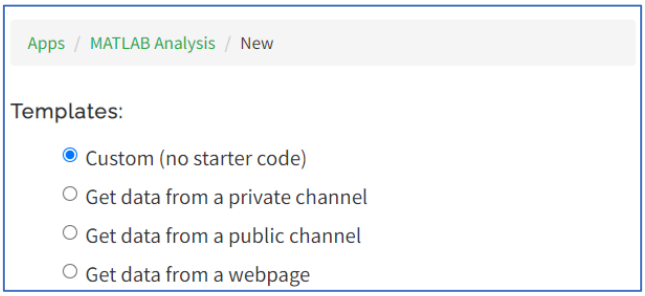
To send a basic email, we are going to write some code in a programming language called MATLAB. This language is widely used in engineering and is taught in first year at JCU. Start by using the top navigation bar to navigate to Apps > Matlab Analysis



Then click “New” to make a blank MATLAB file:



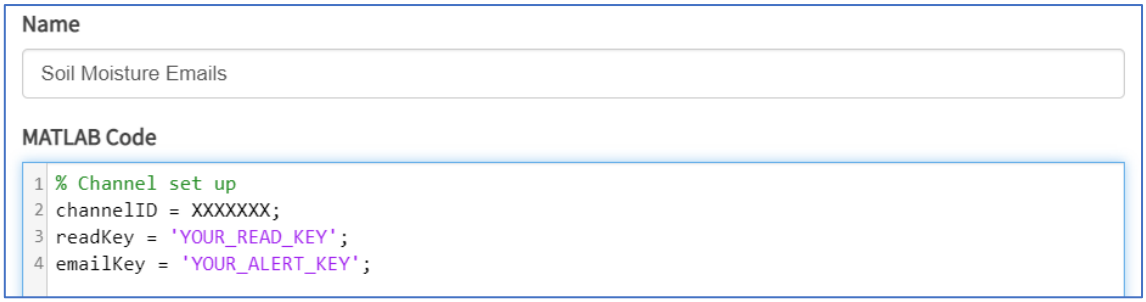
We like a challenge so select ‘Custom (no starter code)’ from the templates options, then scroll down and click ‘Create’.



This will open up a blank file for writing code. Give it a title, and create some variables to store your channel ID, read key, and alert key. Make sure you have copied your channel ID and keys correctly, otherwise you’ll run into problems.

A few notes about the Matlab language at this point:

- You can use the % sign to write comments
- Each line of code should end with a semicolon ; (otherwise it will print stuff out for every line!)



To read data from your channel, you will need to use the 'thingSpeakRead' function. Here's an example:

```
% Read in the soil moisture data
soilMoisture = thingSpeakRead(channelID, 'ReadKey', readKey, 'Fields', 1, 'NumPoints', 1);
```

Let's break this function down so that we can understand how it works.

- The first input is the **channelID**, which is your variable where you stored your channel ID earlier.
- After this first input, we have to explicitly tell the **thingSpeakRead** function what the rest of the inputs are. The second input '**ReadKey'** tells Matlab that the third input **readKey** is the key it should read data from.
- The fourth input '**Fields'** tells Matlab that the fifth input **1** is the field it should read from – so in this case, it should read the first field only (you will need to change the field number if you put your soil moisture in a different field).
- The sixth input '**NumPoints'** tells Matlab that the seventh input **1** is the number of soil moisture values it should collect. This means that Matlab will only gather the single most recent value. If you wanted to collect more values, you can simply increase this number.

To summarise: this line of code tells Matlab to read 1 data point from the soil moisture field, which is located at a particular channel with a particular read key.

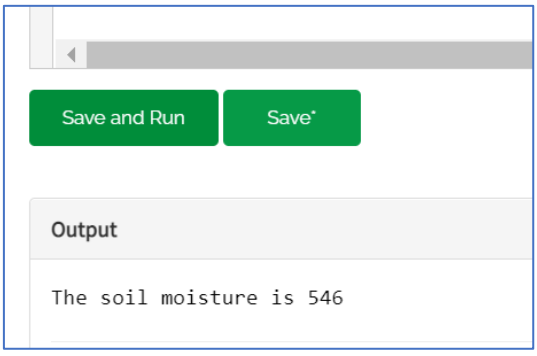
Now we want to test that our code is reading from the channel properly. To do this, you can use the following line of code:

```
% Display the soil moisture
fprintf('The soil moisture is %d\n', soilMoisture)
```

The **fprintf** function is similar to Serial.print, except it allows for fancier formatting. The **%d** part of the message tells Matlab that "a number will go here". After closing the message, the number 'soilMoisture' is inserted into the position held by the **%d**.

The **\n** simply moves to a new line after the message is done, which looks nicer if you want to print more messages later on.

At this point, click "Save and run". You should see the formatted message appear in the 'Output' box below:



Reading from the channel achieved! Now let's turn it into an email. Carefully copy the following code:

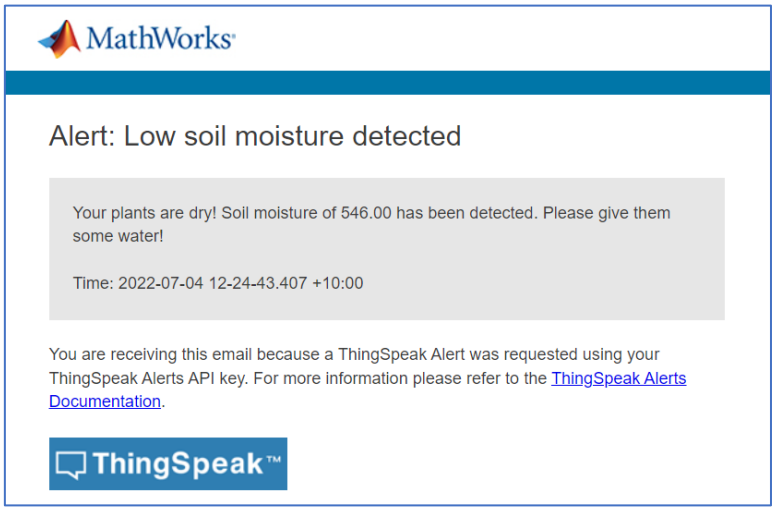
```
12 % Send an email
13 emailSubject = "Low soil moisture detected";
14 emailBody = sprintf("Your plants are dry! Soil moisture is %d.", soilMoisture);
15 alertUrl= "https://api.thingspeak.com/alerts/send";
16 options = weboptions("HeaderFields", {'Thingspeak-Alerts-API-Key', emailKey})
17 result = webwrite(alertUrl, 'subject', emailSubject, 'body', emailBody, options);
```

Now this code is quite complex, so let's briefly look at each line to understand what it's doing.

- Line 13 – we are creating the title for the email we are writing
- Line 14 – we are creating the message body for the email we are sending. **sprintf** works the same was as **fprintf**, except it stores it as a message rather than printing it to the output panel
- Line 15 – this is the URL that our message will be given to, so that it can then be emailed to us
- Line 16 – we are telling Matlab what our personal emailKey is (that's how it knows which email address to use)
- Line 19 – we are sending our message to the URL that handles ThingSpeak emails, along with some information about the type of message

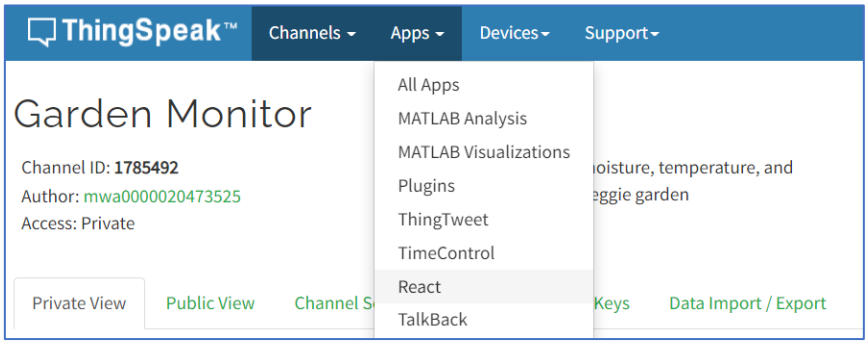
BE VERY CAREFUL THAT YOUR CODE IS CORRECT. ThingSpeak only lets you send an email request twice every 30 minutes. If you make a mistake, you could be waiting for a while to try again! **Check your code with an instructor to be super safe.** Once you have done this, you can click "Save and Run".

After a few moments, you should receive an email. It will go to the email address you used to set up your ThingSpeak account. If you don't receive it, check your Junk/Spam/Clutter folder.

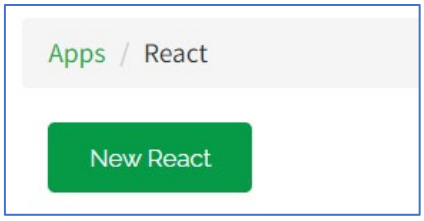


Step 3: Trigger the code when soil moisture is too low

At the moment we have to press a button to trigger an email. That's not overly helpful, so in this step we are going to set up a 'React' that will watch for low soil moisture and trigger our code only when low soil moisture occurs. To get started, navigate to Apps > React



Then click “New React”.

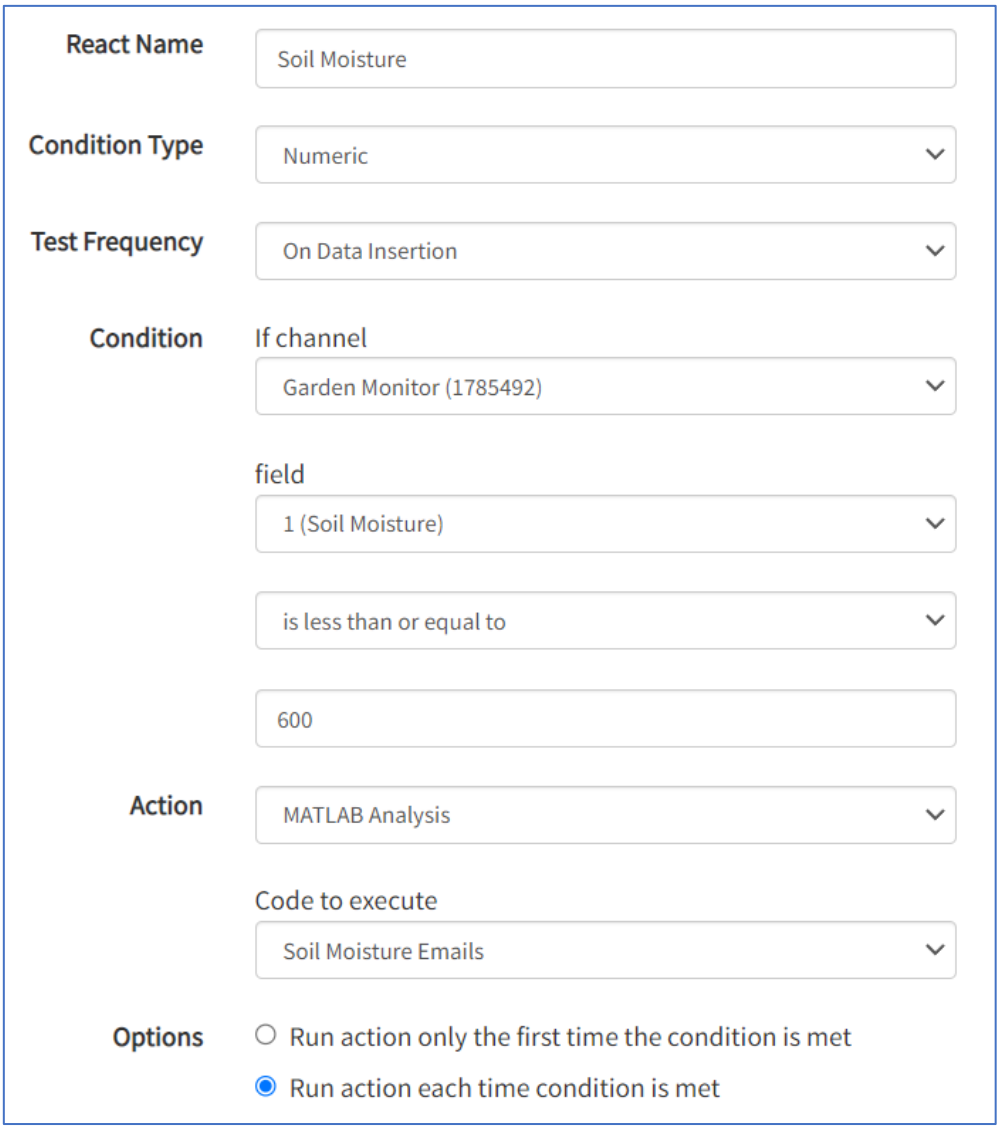


Now we have to set up our ‘React’. Give your React a name and set the “Condition Type” to numeric (because our soil moisture data is numbers).

For ‘Test Frequency’ – you can decide what you’d like! The “On Data Insertion” mode will run the code whenever new data appears in the channel. Alternatively, you can set it to run every 30 or 60 minutes.

Now go through and choose your channel, your field, your conditions, and the code to run. Use the following screenshot as a guide.

You will also want to select “Run action each time condition is met” – we want to get an email every time the soil moisture is too low!



Click “Save React” at the bottom of your screen.

The next time that your garden monitoring system sends new data to ThingSpeak, the 'React' will check the soil moisture level. If it's lower than the threshold that you set, it will trigger the code to send you an email. To test that it's working, stick the soil moisture probe into dry soil and wait for it to send new data to Matlab!

Step 4: Decreasing the frequency of updates

Once you know that your system is working, you will want to measure the soil moisture less often. The reason for this is that ThingSpeak can send a maximum of 2 emails every 30 minutes (and besides, soil moisture won't change too quickly).

Head back to your Arduino code and increase the delay or sleep period. You will probably want to set a minimum of 15 minutes between reads.

Additional Activities

Got everything working but still have time to spare? Here's some suggested extensions to add to your system:

- Set up email alerts for when the soil is too wet! You could do this in two main ways:
    - Write new code and add a new React
    - Modify your existing code and add a new React
- Set up email alerts for extreme temperatures.
- Modify your code so that no email is sent when it's raining (very high humidity indicates rain).
- Modify your code to show the last 5 soil moisture values.
- Modify your code to only send you an email if it hasn't done so for the last few hours (one way that you could do this is to look at the x most recent data points to determine whether an email would have already been sent previously).
- Include a weather report with your email.