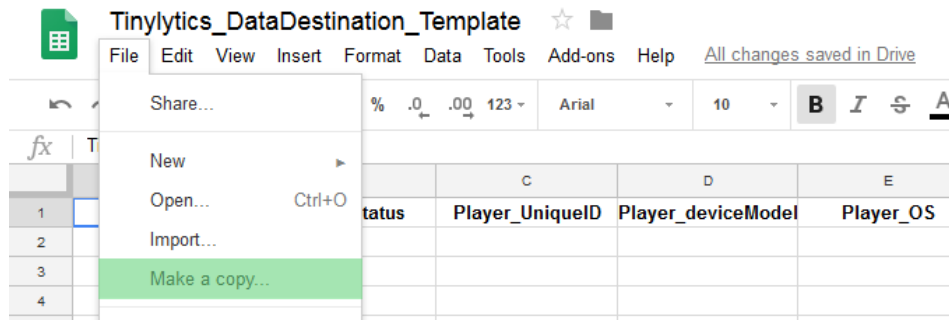


Tinylytics User Guide v0.1

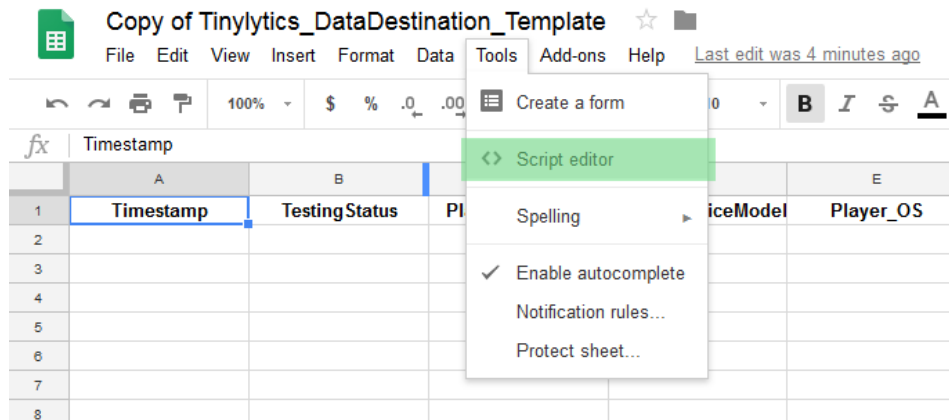
Initial Setup Guide

1. Go to this template, make a copy. Name it whatever you'd like (and you can rename it whenever you want later).

https://docs.google.com/spreadsheets/d/1afYRzPYwN3HHg_G63SF9IM1i2gaFDpsbTZCkur6cVnk

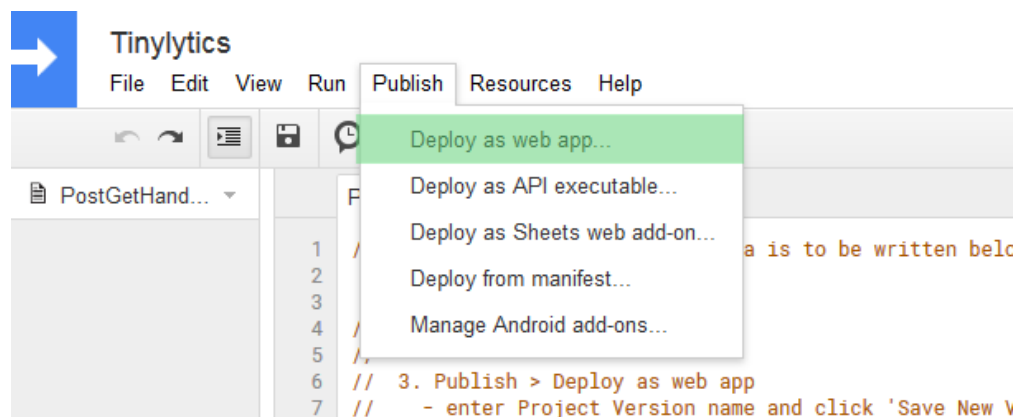


2. Go to Tools > Script Editor

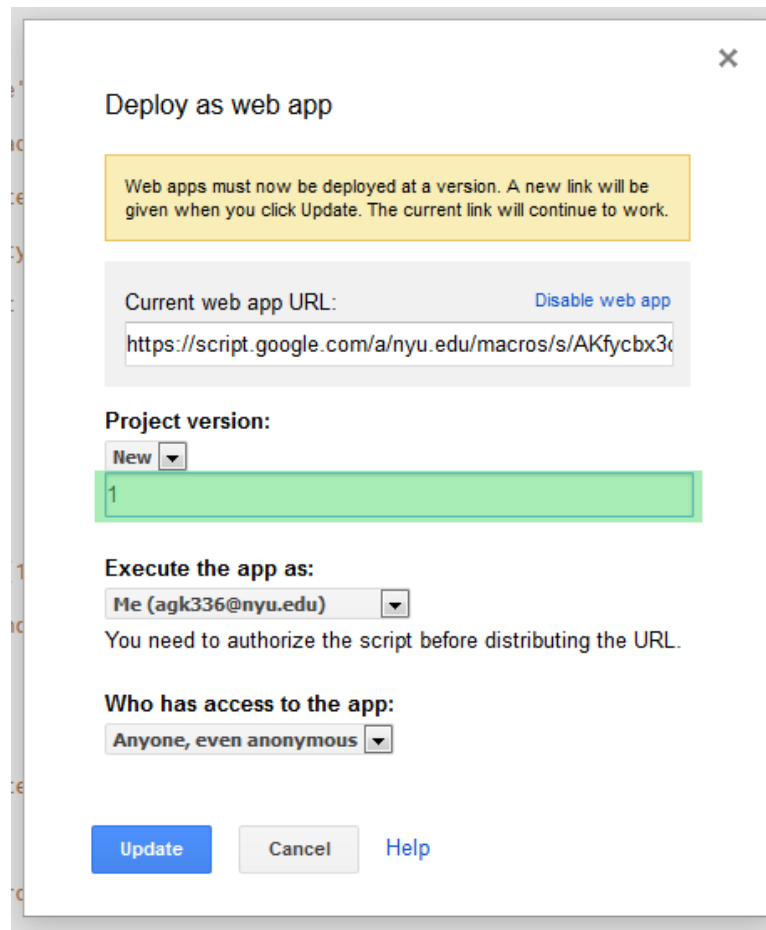


This will open a new tab, a project named Tinylytics_Instance, with a single script called PostGetHandling.gs

3. Go to Publish > Deploy as web app...



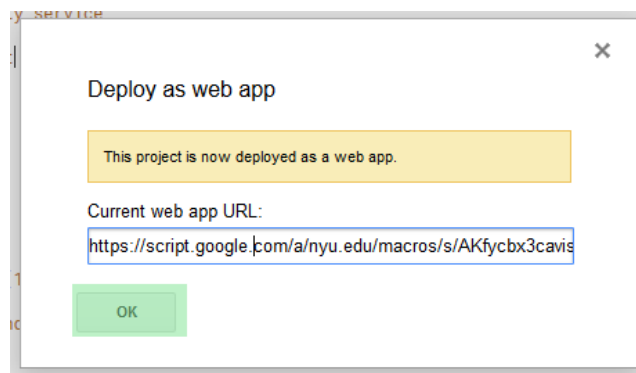
4. That will bring up a prompt like the following. Don't edit the "web app URL". Check that "Execute the App as" is you, and that access is "Anyone, even anonymous". You must enter in a project version (for some reason), just put "1" as below. Then click "Update".



The screenshot shows a dialog box titled "Deploy as web app" with a close button (X) in the top right corner. Inside the dialog, there is a yellow informational box stating: "Web apps must now be deployed at a version. A new link will be given when you click Update. The current link will continue to work." Below this, the "Current web app URL:" is displayed as "https://script.google.com/a/nyu.edu/macros/s/AKfycbx3c...". To the right of the URL is a link that says "Disable web app". Under the heading "Project version:", there is a dropdown menu currently set to "New" and a text input field containing the number "1". Below this, under "Execute the app as:", a dropdown menu shows "Me (agk336@nyu.edu)". A note below states: "You need to authorize the script before distributing the URL." Under the heading "Who has access to the app:", a dropdown menu is set to "Anyone, even anonymous". At the bottom of the dialog are three buttons: "Update" (blue), "Cancel" (gray), and "Help" (blue).

5. An Authorization Required prompt might appear. If so, click "Review Permissions", and "Allow". Note: This is only affecting the view/edit permissions for this project copy. No one else has access to your data.

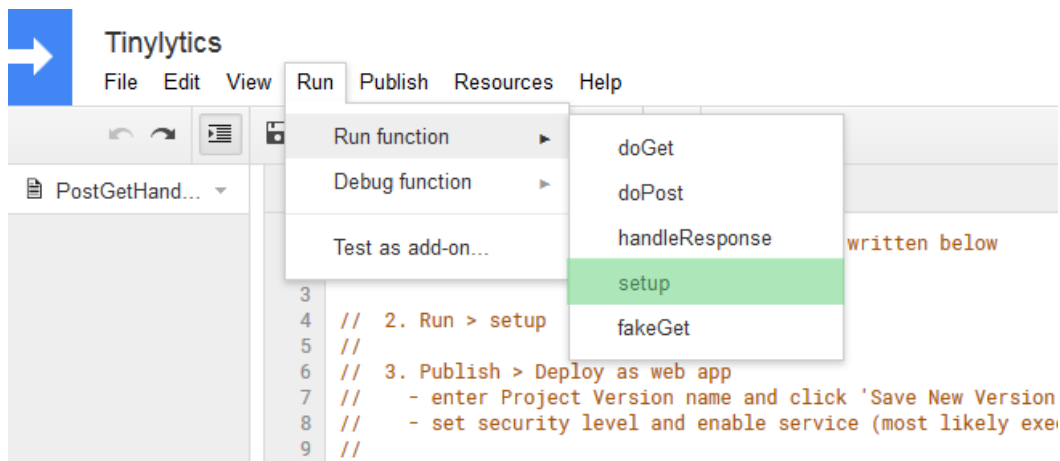
6. A prompt will tell you the project is displayed. Do *not* copy the 'web app url link'. Just Click OK.



This screenshot shows the same "Deploy as web app" dialog box, but now with a yellow success message box at the top that says: "This project is now deployed as a web app." The "Current web app URL:" is still "https://script.google.com/a/nyu.edu/macros/s/AKfycbx3cavis...". At the bottom, there is a single green button labeled "OK".

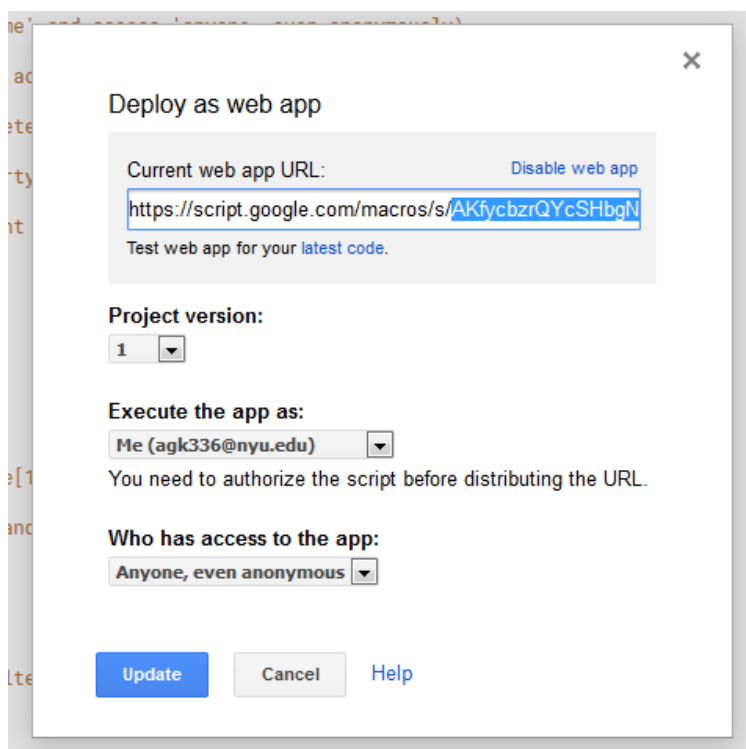
7. Go to Run > Run function > setup

If the authorization prompt didn't appear earlier, it should now.



8. After running the setup function, navigate again to Publish > Deploy as web app...

You'll see a screen like this:



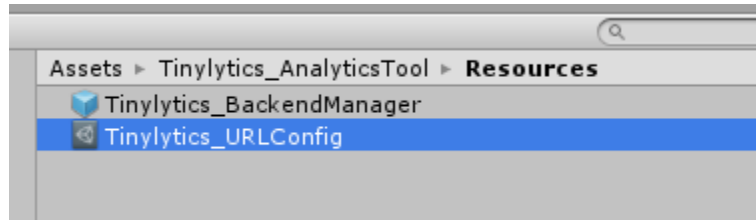
Copy out the "Current web app URL" It should look like this:

<https://script.google.com/macros/s/AKfycbZrQYcSHbgNmKNwYNDM-I2EmsrhK-2BriQPXgbF7TbaTkKZ9a4u/exec>

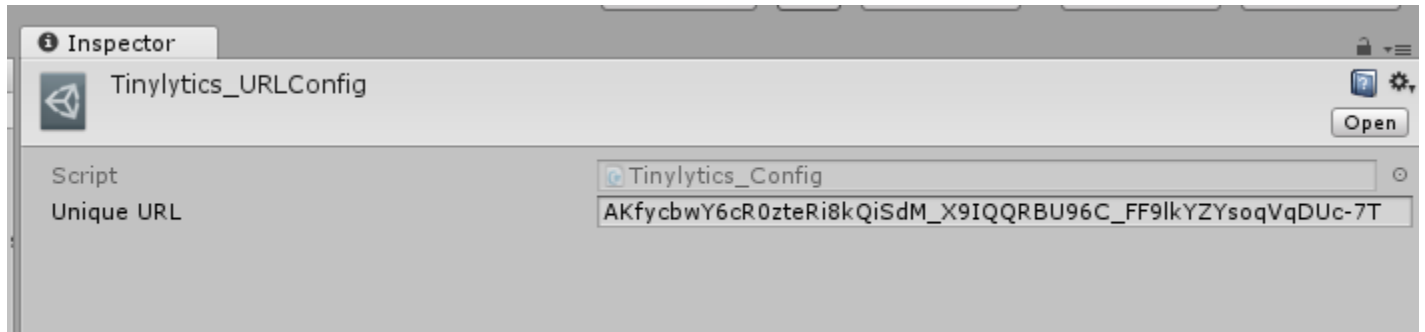
We just want the code portion between "s/" and "/exec" that looks like random numbers and letters. Like for this one, it's "AKfycbZrQYcSHbgNmKNwYNDM-I2EmsrhK-2BriQPXgbF7TbaTkKZ9a4u"

Copy it.

9. Back in Unity, navigate to Assets / Tinylytics_AnalyticsTool / Resources, and click on the Tinylytics_URLConfig asset.



10. Paste the code from step 8 into the unique URL field, like so:

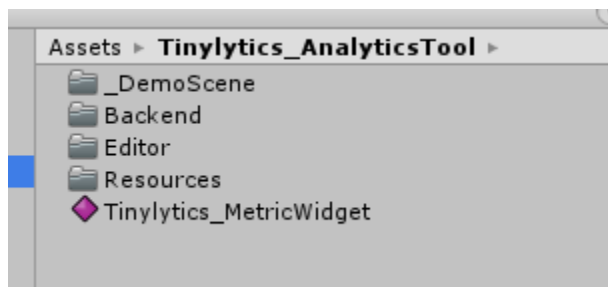


11. To check if it's working, go to Assets / Tinylytics_AnalyticsTool / _DemoScene

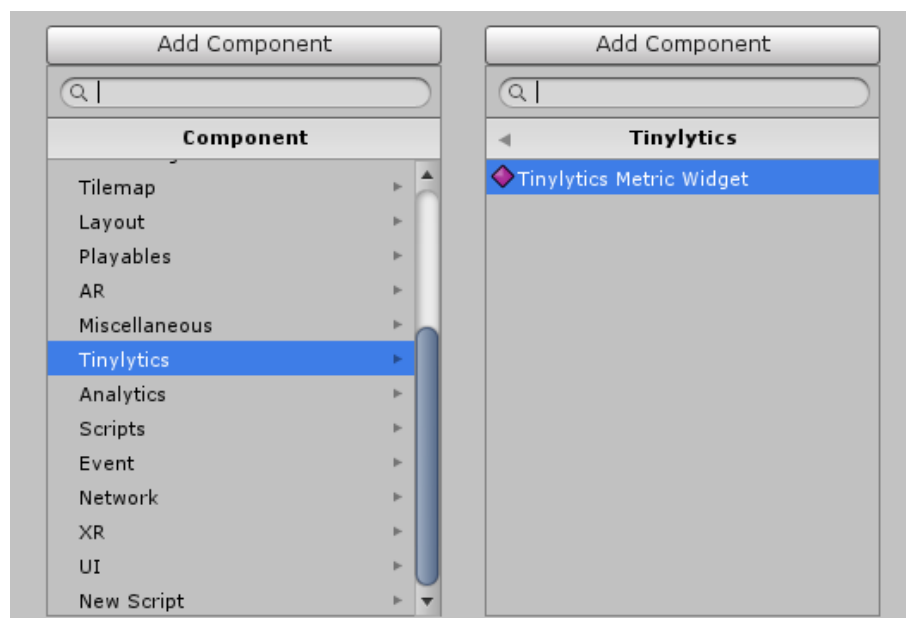
Click on "ExampleScene" and hit play. Now go back to your data sheet. If you did everything correctly, you should see data populated!

How to Track Data

Option 1: Using the Tinylytics Metric Widget



The MetricWidget is an easily customizable component you can add to objects in your game. You add to objects like any other component. You can drag it onto things from the Asset folder, or add via Add Component > Tinylytics



As a component, the Metric Widget looks like this:



Here you can set:

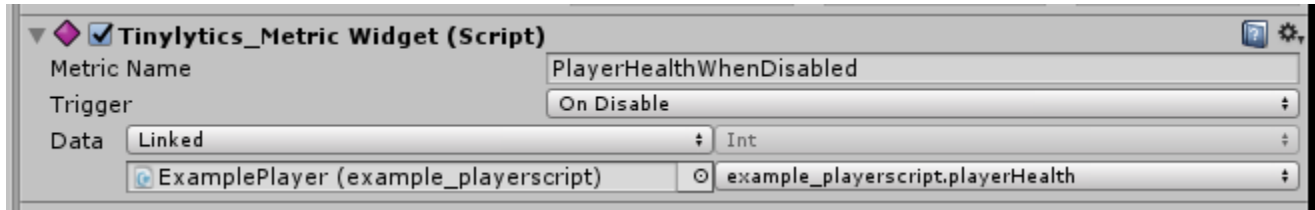
Metric Name: What the metric will be called when it hits your spreadsheet database. Good names are things like “PlayerHealthAtGameEnd” or “SecondsRemainingWhenPlayerDied”.

Trigger: This is what will tell this widget to log data. If it’s “Start” or “Awake” it will do so when the object it’s on is first loaded into a scene. “On Enable” and “On Disable” fire when the object is enabled or disabled. If you set the trigger to “Custom Trigger Call”, the widget will only fire when another script calls its “OnCustomTrigger” method is called. You still set the data to send in the inspector here, but you can control when it fires using another script.

Data: This is what the widget will send when the trigger condition is met.

If it's "Custom" you can have it send a string, int, float or bool that you enter. For example, you could the widget on a scene's GameManager, set the trigger to Start, and have it send the Level Number.

If you change "Custom" to "Linked", you can drag any object into the field to grab public variables from it (just like how UI Buttons' OnClick method is set up).



Option 2: Call using code

The Metric Widget allows you to track data without code. But you can also send data directly using code. In any of your scripts, you can log a data event by calling `Tinylytics.AnalyticsManager.LogCustomMetric`. You need to supply the metric name, and the data, which must be formatted as a string (if it is a int or float etc, use the `.ToString()` method when passing the data).

In this manner, you can embed the analytics tracking directly into your gameplay scripts.

Caution: The game can only send so much data at once. Do not place this in the `Update()` loop!

```
Tinylytics.AnalyticsManager.LogCustomMetric("Metric Name", "Data to send (as a string)");
```

Option 3: Customize the Analytics Manager

If you're an advanced user, you can customize the analytics manager yourself. It's located in Assets > Tinylytics_AlayticsTool > Backend, open `AnalyticsManager.cs`

You could write your own tracking commands here. "LogSessionPlaytime" is an example of how the manager could track playtime, and report it whenever called. In this way, you can extend Option 2 by consolidating your reporting code here.

Lastly to disable analytics:

Navigate to Assets/Tinylytics_AnalyticsTool/Resources/

Click on the `Tinylytics_BackendManager` prefab, and Uncheck "Analytics_Enabled". If this is unchecked, no data will be sent.