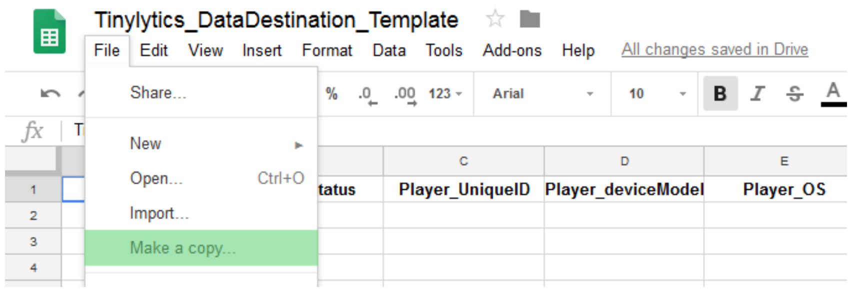
Tinylytics User Guide v0.1

Initial Setup Guide

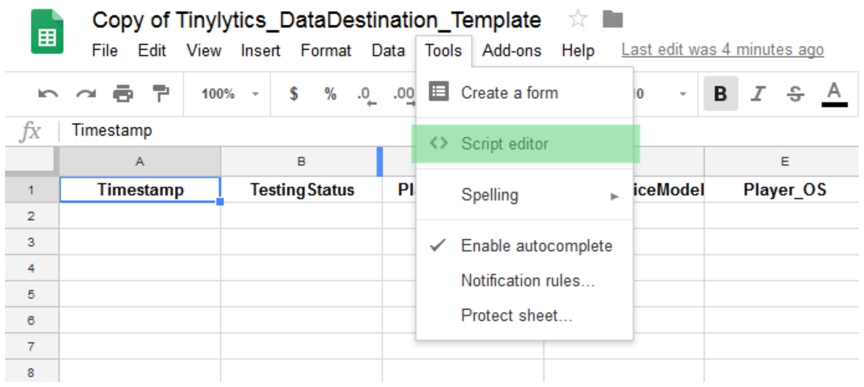
1. Navigate to this template sheet and make a copy. Name it whatever you'd like (and you can rename it whenever you want later too).

https://docs.google.com/spreadsheets/d/1afYRzPYwN3HHg_G63SF9lM1i2gaFDpsbTZCkur6cVnk

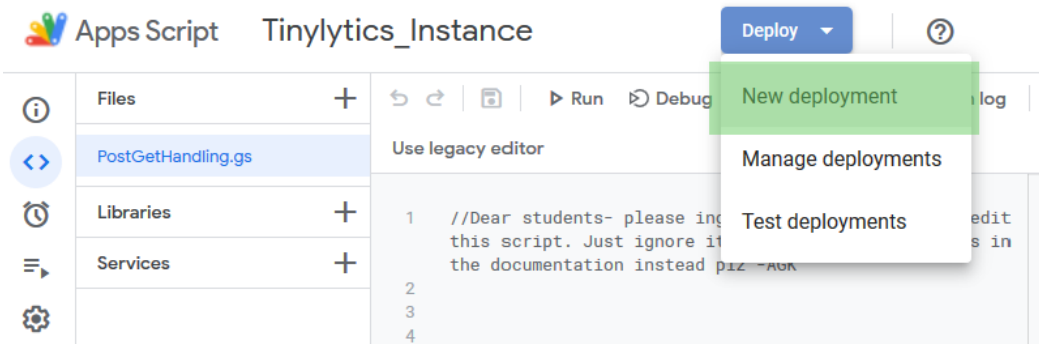

Tinylytics_DataDestination_Template
File Edit View Insert Format Data Tools Add-ons Help
All changes saved in Drive
Share…
New
Open… Ctrl+O
Import…
Make a copy…
Arial
10
tatus
Player_UniqueID
Player_deviceModel
Player_OS

2. Go to Tools > Script Editor

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



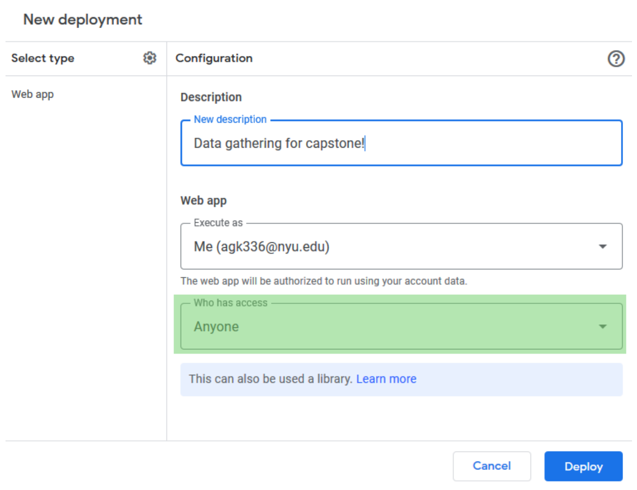
3. Go to Deploy > New Deployment…



```
//Dear students- please ing                 edit
this script. Just ignore it                s in
the documentation instead plz -AGK
```

4. That will bring up a prompt like the following. Enter a quick description for what you're using this for. Change "Who has Access" to "Anyone".

Then click "Deploy"


New deployment

Select type

Web app

Configuration

Description

New description: Data gathering for capstone!

Web app

Execute as: Me (agk336@nyu.edu)

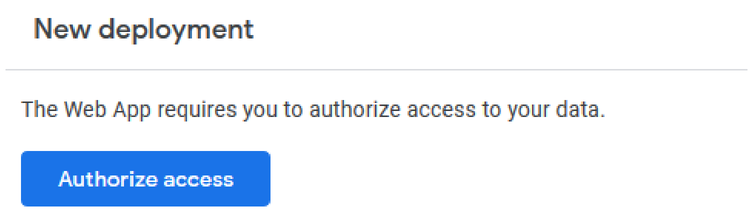
The web app will be authorized to run using your account data.

Who has access: Anyone

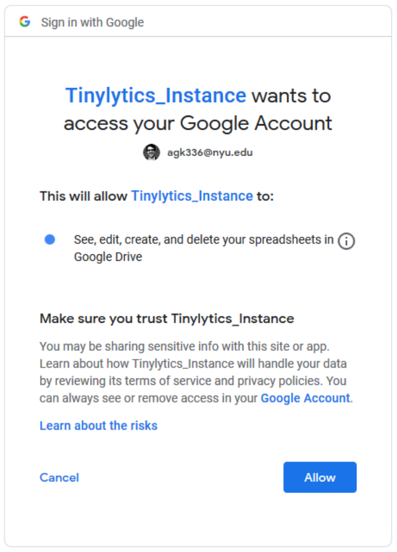
This can also be used a library. Learn more

5. An Authorization Required prompt might appear. If so, click "Authorize Access" and then "Allow"

Note: This is only affecting the view/edit permissions for this particular workbook. No one else has access to your data.


New deployment

The Web App requires you to authorize access to your data.


Sign in with Google

Tinylytics_Instance wants to access your Google Account

agk336@nyu.edu

This will allow Tinylytics_Instance to:

- See, edit, create, and delete your spreadsheets in Google Drive

Make sure you trust Tinylytics_Instance

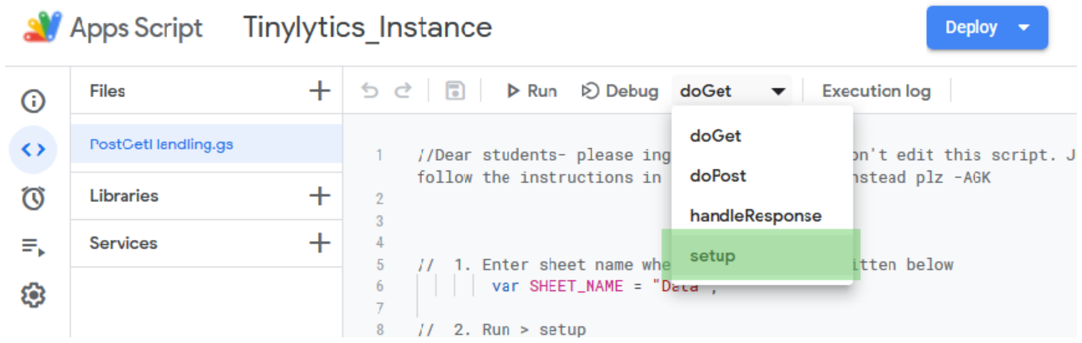
You may be sharing sensitive info with this site or app. Learn about how Tinylytics_Instance will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your Google Account.

Learn about the risks

6. After a short delay, the project is displayed. Click "Done" to close the deployments window.

7. Click the dropdown at the top of the code editor, and select the “Setup” function. Then click “Run”.

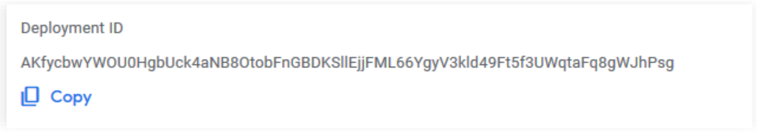
If the authorization prompt didn’t appear earlier, it should now.



```
//Dear students- please ing                    on't edit this script. J
follow the instructions in                      nstead plz -AGK



//  1. Enter sheet name whe                    itten below
          var SHEET_NAME = "Data";

//  2. Run > setup
```

8. After running the setup function, navigate again to “Deploy”, and “Manage Deployments”.

We need the “Deployment ID”, it’s a long code of letters and numbers. Copy it.

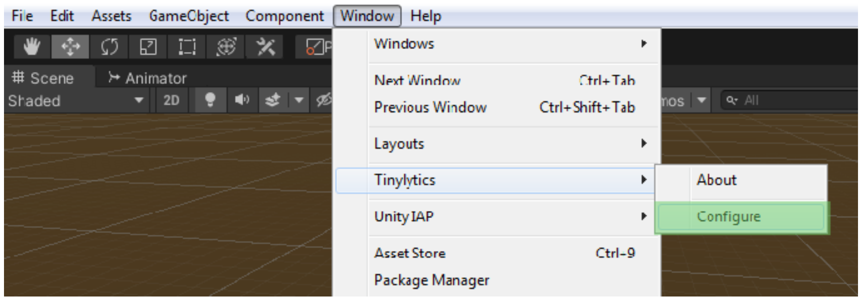


Deployment ID

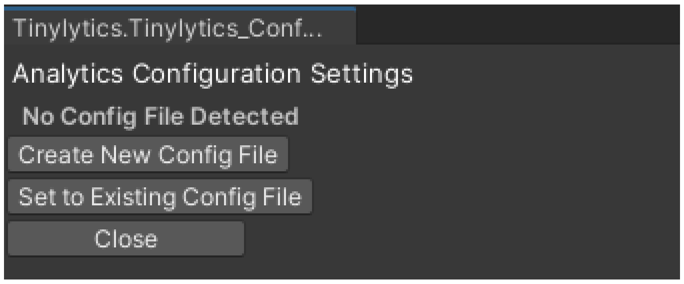
AKfycbwYWOU0HgbUck4aNB8OtobFnGBDKSlIEjjFML66YgyV3kld49Ft5f3UWqtaFq8gWJhPsg

Copy

9. Back in Unity, click Windows > Tinylytics > Configure, to bring up the config window.



10. Click “Create New Config File”



Analytics Configuration Settings

No Config File Detected

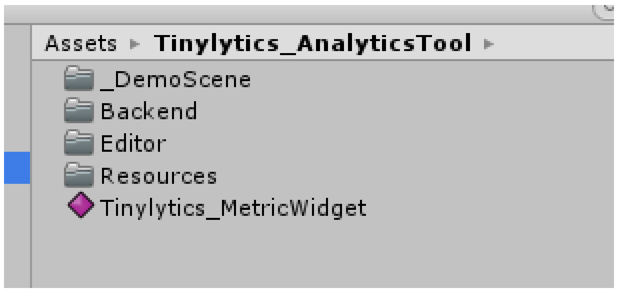
10. Paste the code from step 8 into the Deployment ID field. Close this window.

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

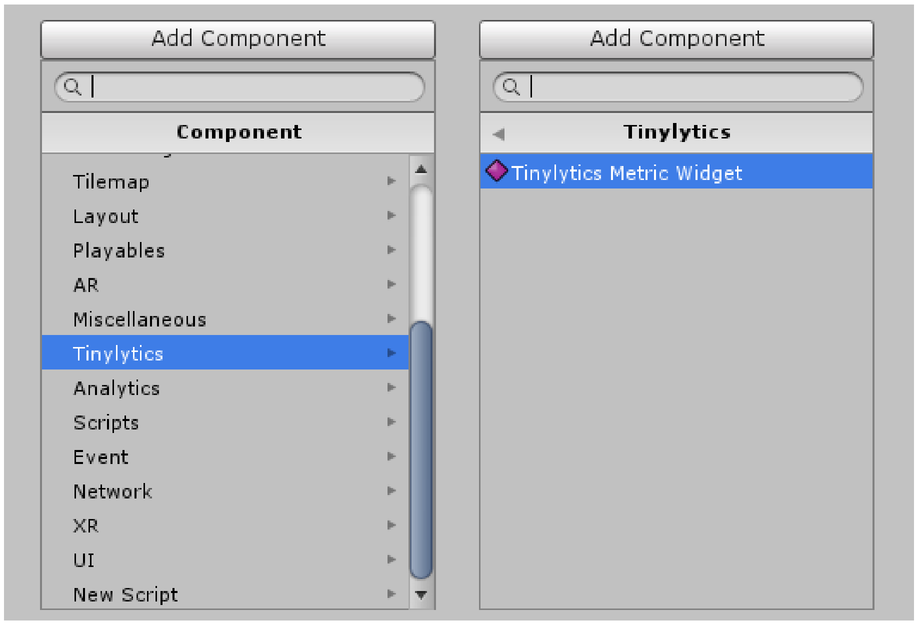
Click on “ExampleScene” and hit play. Now go back to your data sheet. If you did everything correctly, you should see data populated!

How to Track Data

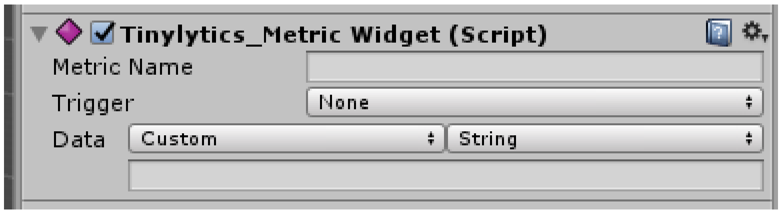
Option 1: Using the Tinylytics Metric Widget



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:

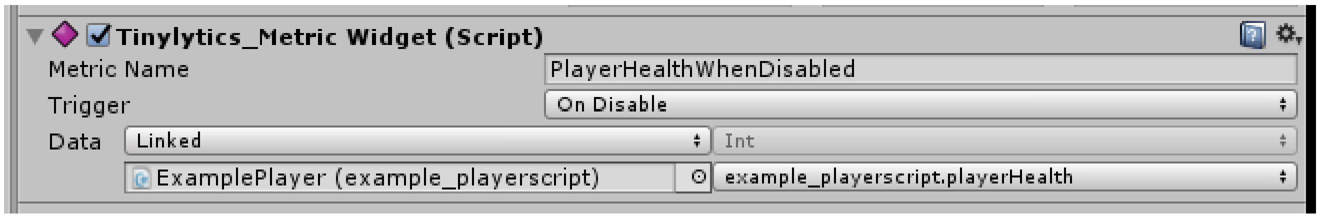
Metric Name: What the metric will be called when it hits your spreadsheet database. Good names are things like “PlayerHealthAtGameEnd” or “SecondsRemainingWhenPlayerDied”.

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).


Tinylytics_Metric Widget (Script)

Metric Name: PlayerHealthWhenDisabled

Trigger: On Disable

Data: Linked | Int

ExamplePlayer (example_playerscript) | example_playerscript.playerHealth

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.