# Table Tennis Information Tracker Maintenance Documentation

Table of Contents

[Table Tennis Information Tracker Maintenance Documentation 1](#_Toc87385558)

[Overall Note 3](#_Toc87385559)

[Heroku 3](#_Toc87385560)

[What is Heroku and why it significant? 3](#_Toc87385561)

[Access Application 3](#_Toc87385562)

[Get Link to Application 5](#_Toc87385563)

[Application Specifics 7](#_Toc87385564)

[Application Deployment 7](#_Toc87385565)

[Add Access 9](#_Toc87385566)

[Environment Variables 10](#_Toc87385567)

[Transfer Application 11](#_Toc87385568)

[Google API 11](#_Toc87385569)

[What is Google API and why it significant? 11](#_Toc87385570)

[Azure 13](#_Toc87385571)

[What is Azure and why it significant? 13](#_Toc87385572)

# Overall Note

Everything in this document has been set up and there is no need to change anything. This document is only to give you the basics of the tools used to run the Table Tennis Information Tracker Application. If you plan on updating the application and adding in new feature, please reach out to us so we can give you a more comprehensive run-down of the specifics needed to do you want to do.

**IMPORTANT NOTE:**

This application **WILL** go dormant if not used for an extended period of time (like long breaks such as Winter or Summer). This will cause the whole app to basically be erased. The easiest way to mitigate this is to have one person log in at least once every few weeks. A single use will make the app come back to life and not erase its data due to lack to use.

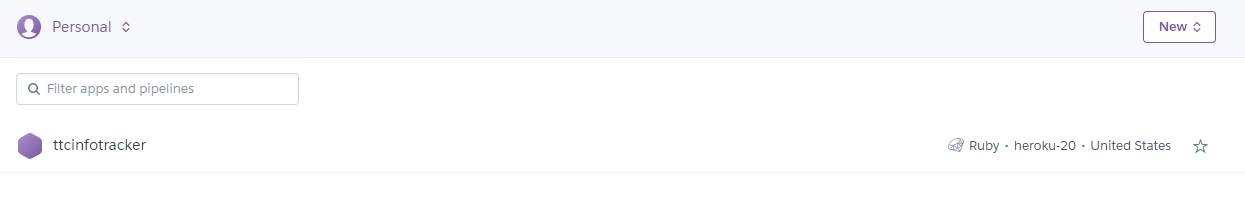
# Heroku

## What is Heroku and why it significant?

Heroku is a service used to host the Table Tennis Information Tracker application. It is the centralized location which brings all components required to run the application together. This is the **most** important service that keeps this application running for the Table Tennis Club.

## Access Application

1. Visit <https://id.heroku.com/login>
2. Sign in with credentials provided during the application transfer process
3. Click on ‘***ttcinfotracker***’



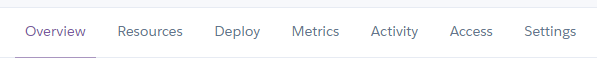
You should be directed to the page below:

Graphical user interface, text, application, email, Teams

Description automatically generated

### Get Link to Application

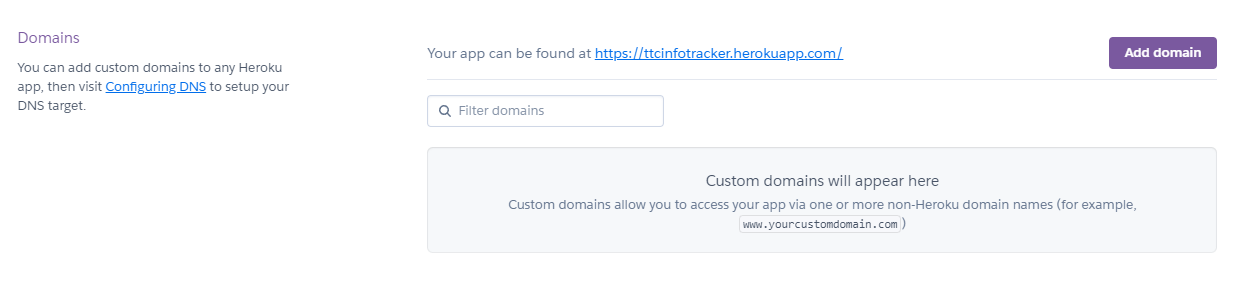
1. Go to the app on Heroku
2. There are 2 ways to get the link:
   1. In Heroku itself
      1. Go to the ‘***Settings***’ tab



You should be directed to the page below:

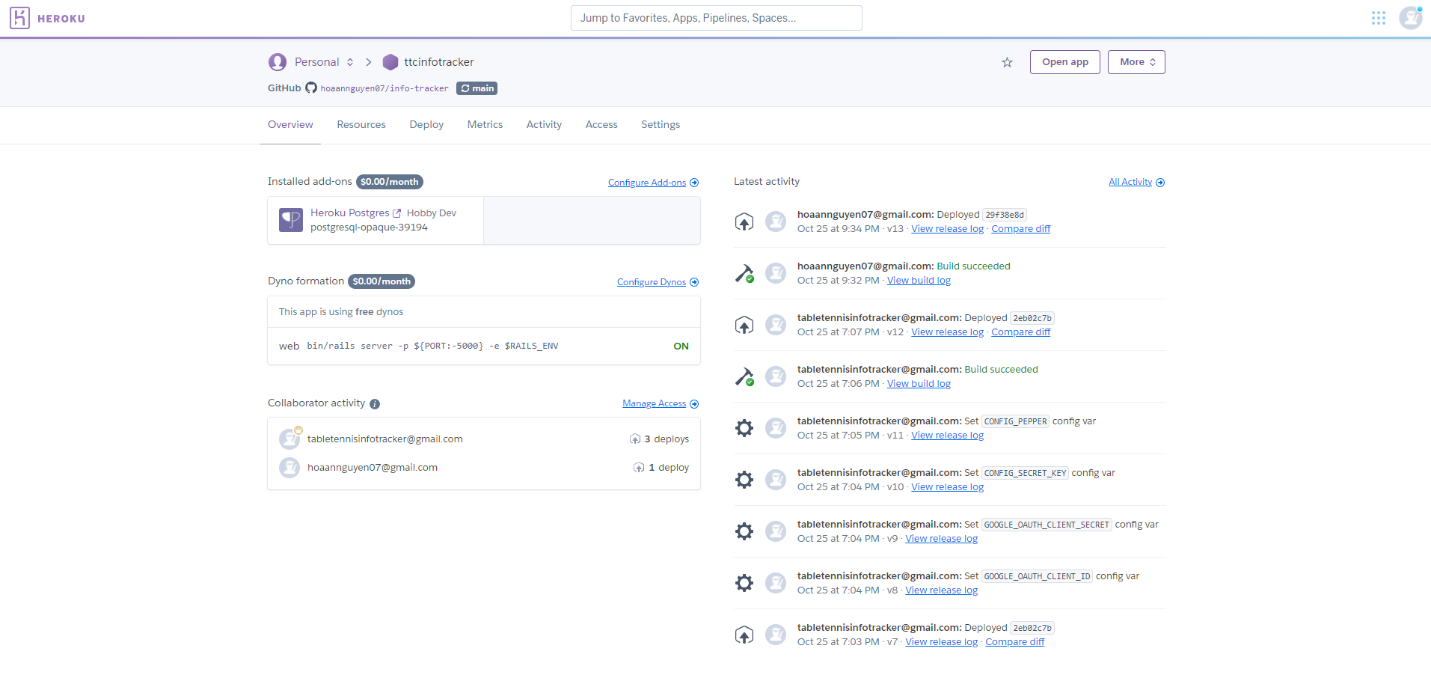
* + 1. Graphical user interface, application, Teams

       Description automatically generatedScroll to the bottom until you can see the ‘***Domains***’ section



Website Link

* 1. Open the Application itself and get the URL
     1. Click on ‘***Open App****’* button (can be accessed from anywhere in the app in Heroku



* + 1. You will be directed to the Table Tennis Information Tracker application on a new tab and then you can copy the URL

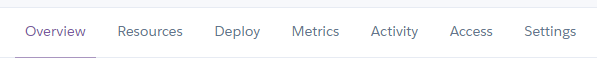
## Application Specifics

Below are specifics on how the application is ran and all the different topics we, as developers, think you should be aware of in case you want to work on the application in the future. That being said, everything has been set up by us before the transfer of ownership of this application and additional work needs to be done. If there are additional questions in the future, you can reach out to us for help.

### Application Deployment

We have set up the application to automatically deploy whenever the ‘main’ branch of the repository has been updated (through pull requests on GitHub). However, if there is ever a need to manually re-deploy the application, then follow the steps bellow:

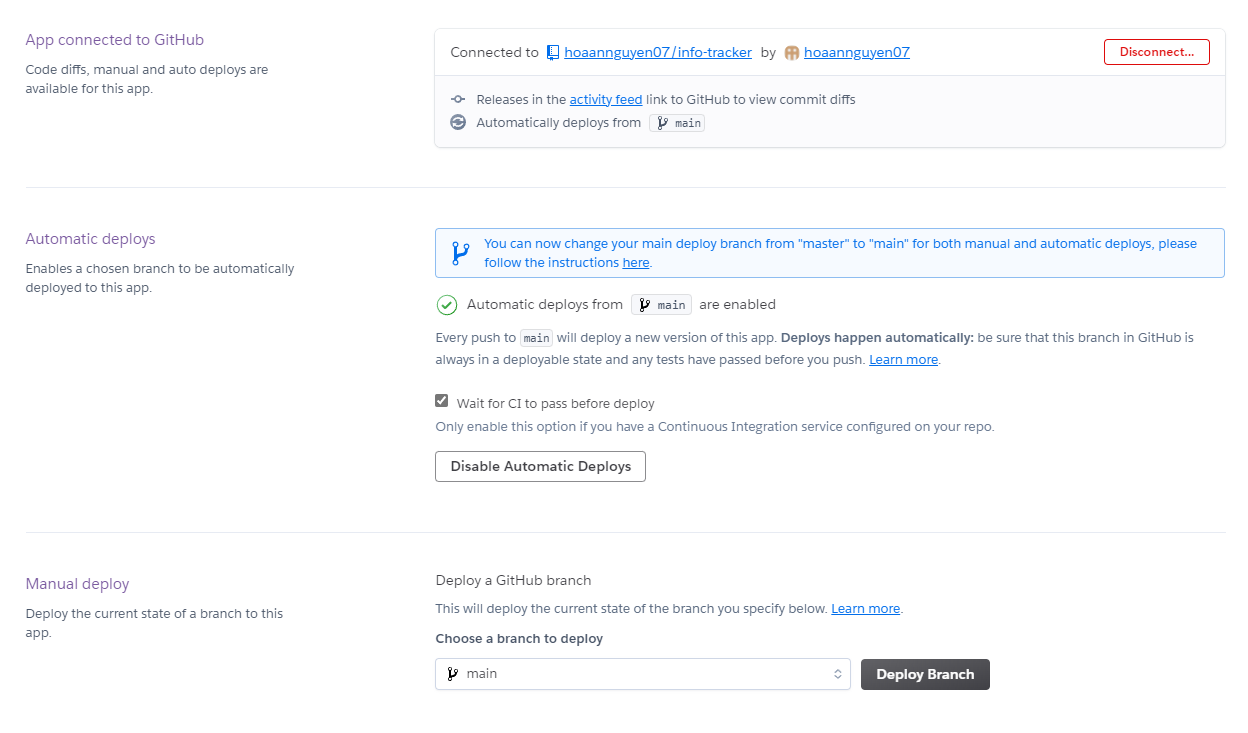
1. Click on the ‘***Deploy***’ tab



Graphical user interface, text, application, email

Description automatically generatedYou should be directed to the page below:

1. Scroll to the bottom and click on ‘***Deploy Branch***’



1. Application will start building and get ready for re-deployment

Graphical user interface, text, application, email

Description automatically generated

You can click on ‘***View build log***’ to see what it is actually doing but this will mostly be for future developers to use

1. When build is done and re-deployment is successful, you should see something like this

Graphical user interface, text, application, Teams

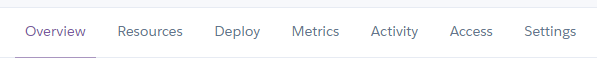
Description automatically generated

1. If the build failed, you could go to the ‘***Activity***’ tab and view the logs of the build and re-deployment to see what went wrong
   1. There are times that the building phase would succeed but the deployment phase would fail

### Add Access

If you have people you want to give access to in order to further develop this project, you can give their Heroku accounts access to the ‘***ttcinfotracker***’. This will allow them to view and make changes to this app on their Heroku account.

1. Go to the ‘*Access*’ tab



You should see a list of names who have access to this app on Heroku. It should look similar to this image:

Graphical user interface, text, application, email

Description automatically generated

1. Click on ‘***Add collaborator***’
2. Add the email address of the Heroku account you want to give access to and click ‘*Save Changes*’
3. This will send an email to the person you want to give access to, telling them that they have been given access to the application on Heroku. When they log into Heroku with their credentials, they should see the ‘***ttcinfotracker***’ app.

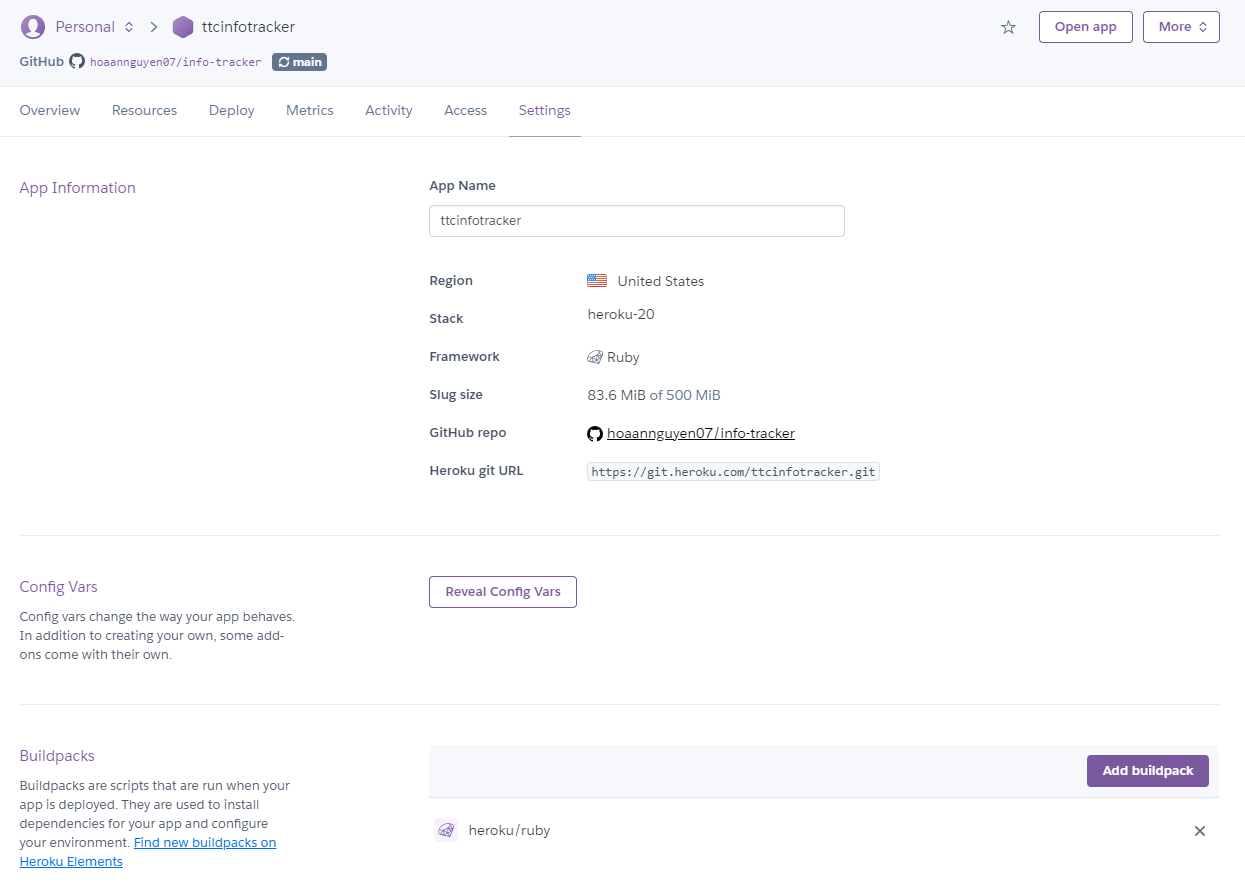
### Environment Variables

There are 4 environment variables you should be aware of, and they are critical to the running of this application. These are the secret keys to connect to the Google API and Azure Storage Blob, which will be expanded upon later in this document. The Google API and Azure Storage Blob environment variables come in pairs: Client ID and Client Secret, which is why there are 4 in total. These keys have been made and no changes will be required whatsoever unless you decide to do away with Google Authentication or change Azure Storage Blobs.

**Note**: These keys are confidential and will cause a major security breach if known publicly. It is important that you do not share these keys unless necessary. During development, it is best practice to make new keys to locally test and make sure the production app (app currently used by users) has a separate set of keys that only it uses. Make sure you know what you are doing if you are going to change a key.

To view these keys, follow the step bellow:

1. Go to the ‘***Settings*’** tab
2. Click on ‘***Reveal Config Vars***’ in the **“Config Vars”** section



### Transfer Application

In case you want to transfer the ownership of this application to someone else’s Heroku account, you use the follow steps:

Note: make sure that you have added that person as a collaborator (see the [Add Access](#_Add_Access) section for more information)

1. Go to the ‘***Settings*’** tab
2. Scroll to the **“Transfer Ownership”** section
3. Choose the Heroku account you want to transfer the ownership of the application to
4. Click ‘***Transfer App…***’

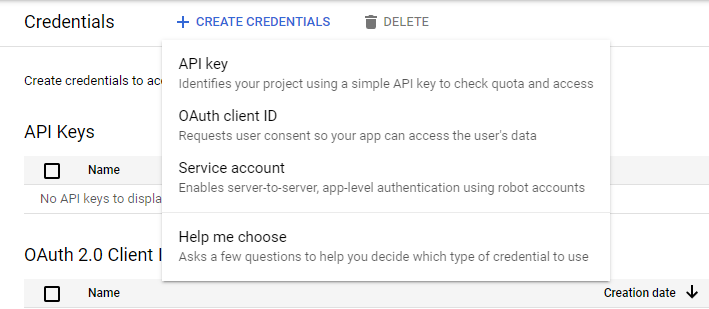
# Google API

## What is Google API and why it significant?

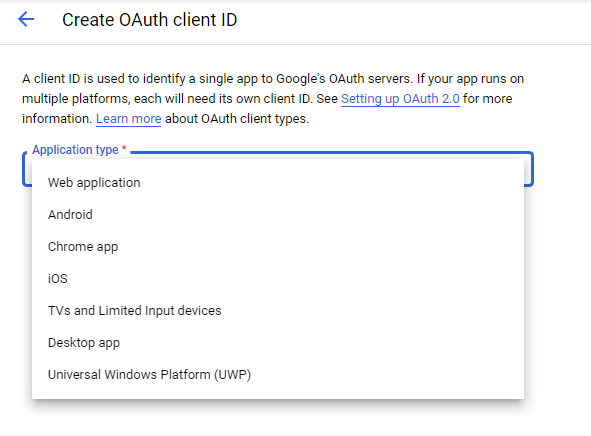
Google API is a service the Table Tennis Information Tracker application uses to allow members to sign into the application (authentication system). The authentication of the application has been delegated to a third-party authenticator to ensure the highest possible level of security. This is the reason for the Google Client ID and Secret keys included in the set of environment variables. This application has been registered with Google to allow it to use Google’s authentication system.

In order to view keys or create new keys (possibly for development purposes):

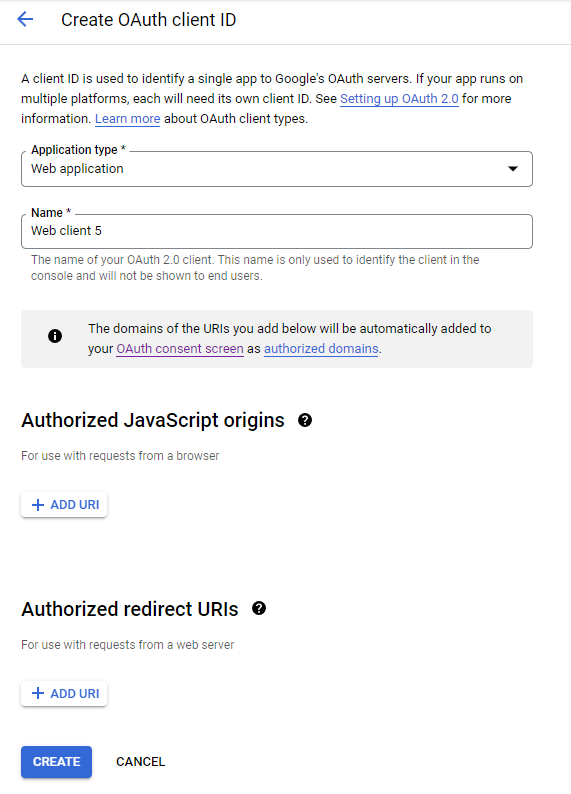
1. Visit  <https://console.developers.google.com/cloud-resource-manager>
2. Log into your desired account
3. Navigate to the **“Credentials”** section (in dropdown menu) under the **“APIs & Services”** tab on the sidebar
   1. If you are logged in with credentials made for this application, then you will see the Table Tennis Information Tracker Application under **“OAuth 2.0 Client IDs”**
      1. Upon clicking on it, you can view the client id & secret
4. To create keys to test locally, you need not be logged into the Google account used for this application. Any account will work.
5. To create a new a new set of keys, follow these steps:
   1. On the **“Credentials”** tab, click on ‘***Create Credentials***’ at the top
   2. Select ‘***OAuth client ID***’



* 1. Under **“Application Type”**, choose Web Application



* 1. Click on ‘***Add URI***’ under **“Authorized redirect URIs”**



* 1. Add the in the URL that will be called back to
     1. Should be: URL of application (ending with a ‘/’) + ‘admins/auth/google\_oauth2/callback’
     2. Example: https://ttcinfotracker.herokuapp.com/admins/auth/google\_oauth2/callback
        1. This is the call back URI for the current Table Tennis Information Tracker production application
        2. ‘https://ttcinfotracker.herokuapp.com/’ is the URL of the app and the callback is ‘admins/auth/google\_oauth2/callback’
  2. Click ‘***Save***’
  3. Use the Client ID and Secret (have to click on the app you just created again) as environment variables
     1. The environment variables are: GOOGLE\_OAUTH\_CLIENT\_ID and GOOGLE\_OAUTH\_CLIENT\_SECRET for the client id and secret, respectively

# Azure

## What is Azure and why it significant?

Azure is Microsoft’s Cloud platform service that the application uses to store images in the photo gallery. The use of secret keys for Azure is the same as Google API. However, there is no need to use Azure for development so there is no need to create new keys like in Google API. There is no need to worry about Azure unless the amount of photos in gallery is too large or you want to change the Azure Bucket housing all the application’s images. In the case of have too many photos, causing the bucket to run out of space, the easiest way to remedy this is to delete photos from the gallery. As for changing Azure Buckets, that requires setting up a new Azure Bucket and getting the keys for that. However, something to note is that if you switch buckets, by default, the images stored in the current own will not transfer over unless you transfer it over yourself, which will require research on your end.