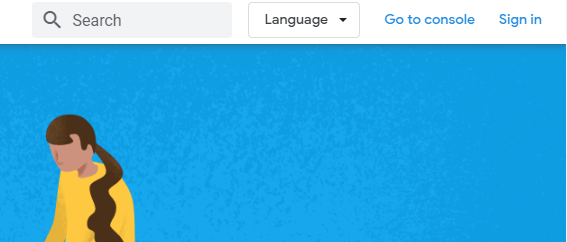
**Using Google Firebase Cloud Database**

# Introductory Notes

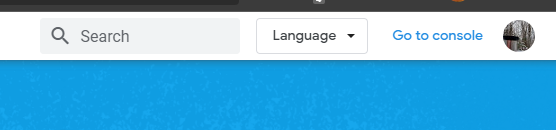
Google Firebase is a free[[1]](#footnote-1) database service offered by Google that can be used for data storage in the cloud. The interface is very user-friendly and can be used for many projects. The following document will describe *a* way to get set up with a Google Firebase database (not necessarily the best way).

# Setting Up a Google Firebase (Google Cloud) Account

The first step in using Google Firebase is first getting an account set up. For this, you will need a general-purpose Google Account. To get started, go to <https://firebase.google.com>, and get signed in. Please note: you will be required to enter a credit card to use the Google Cloud services, however it will not be charged unless you invoke a service that has an associated charge. The process described in this document was charge free on 12/15/2019.



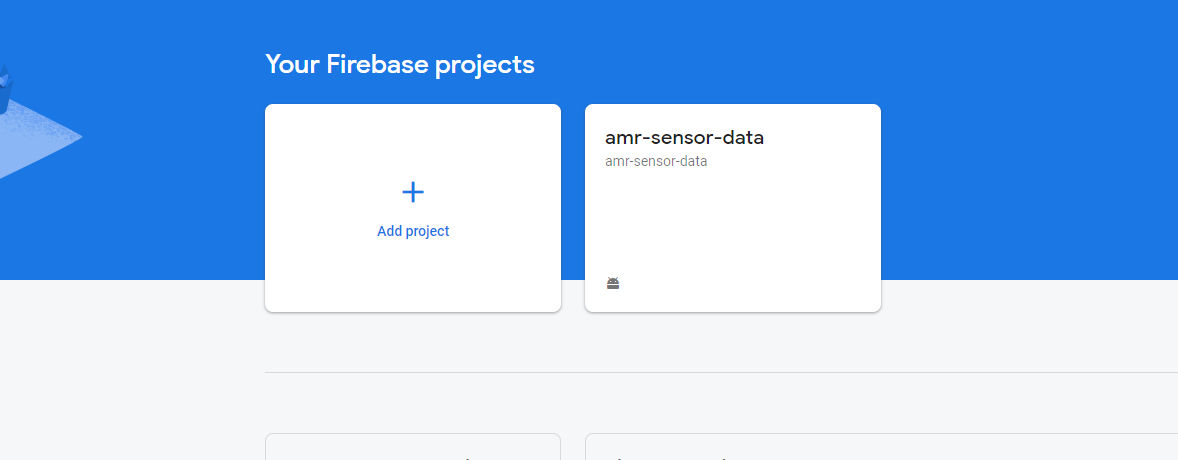
Upon entering in the required information and agreeing to the terms of service, you should be returned to the homepage again, but this time logged in. At this point, you can go ahead and click “Go to console”.

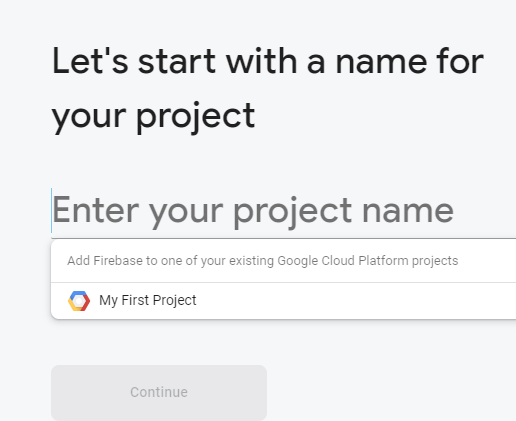


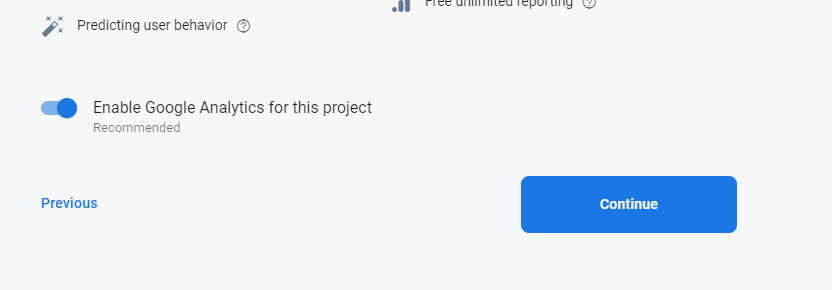
You will then be greeted with the homepage for the firebase console. From here, you can begin making your first database, as described in the next section.

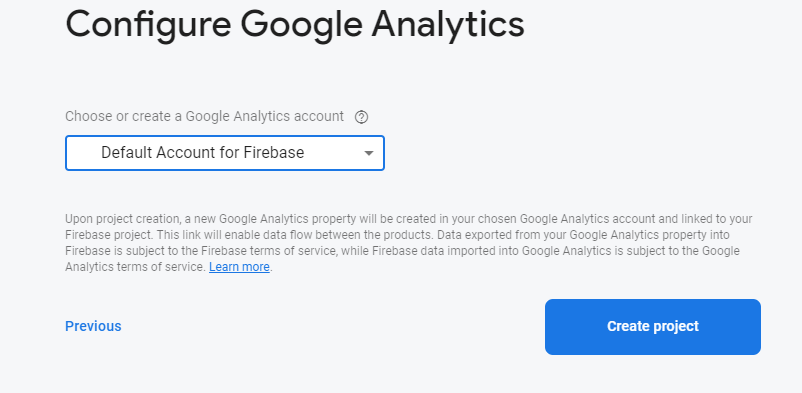
# Making a Database

To make your first project, click on the “Add project” button on the console home page.

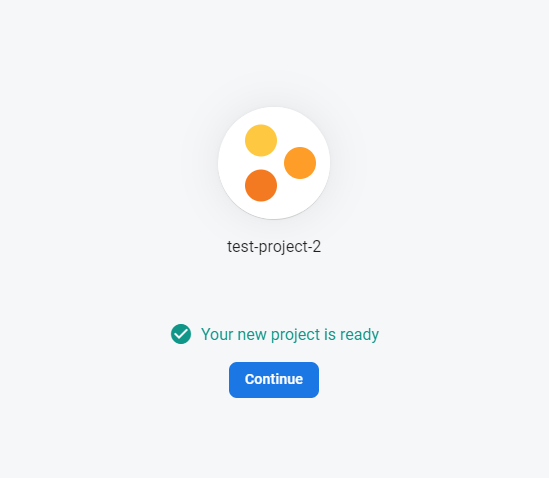


Make a name for your project, and press continue.

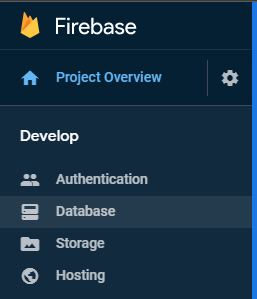
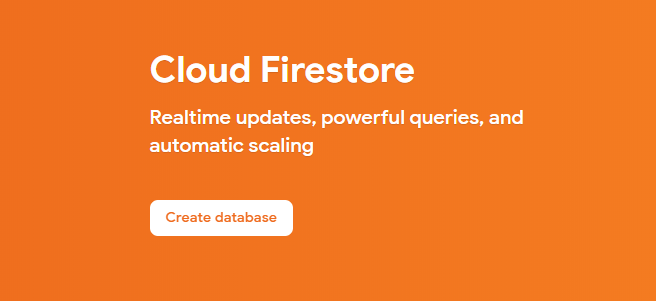
Go ahead and enable Google Analytics for the project:

Allow Google Analytics to use the Default Account for Firebase, or the account you just signed in with.

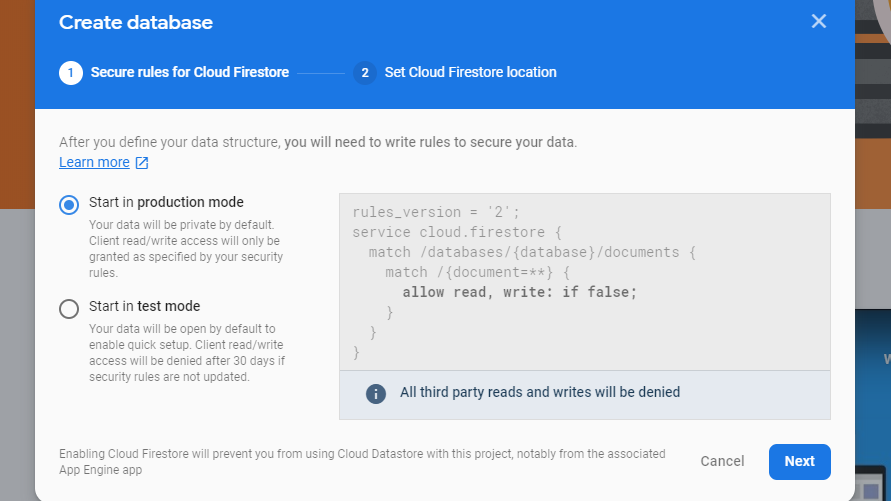
At this point, Google Firebase will take a few moments to set up your new project. When it’s done, click the “Continue” button to continue to the project dashboard.



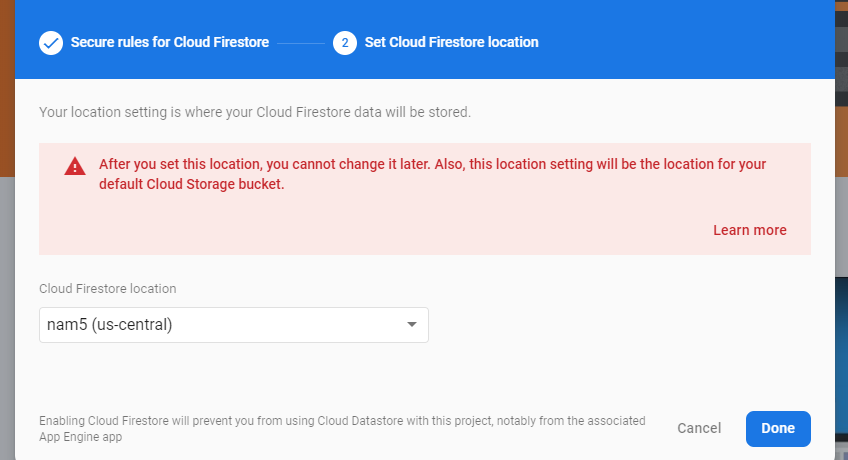
To make a database, click the “Database” tab on the column on the left-hand side of the screen. On the following page, click the “Create Database” button.

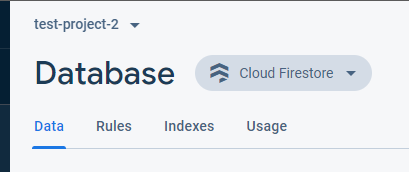


For safety, choose “Start in **production mode**” and click next. We will address this in a few steps.

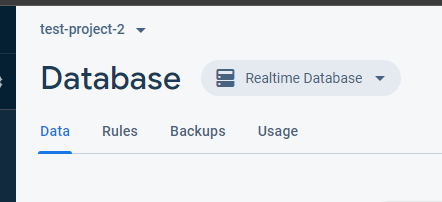


Choose a geographical location you’d like to host your database in (I believe the default will usually be the closest to you) and click done. After a few moments, your database will be created.

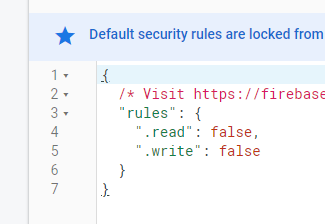


Now we need to configure the database. For the purposes of this guide, we will make a Real-Time Database for sending and receiving data. To do this, click on “Cloud Firestore” next to “Database”, and change it to “Realtime Database”.

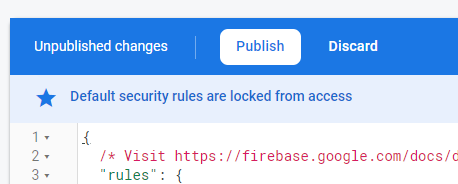
On the page that appears, change to the “Rules” tab.



The rules here are a result of our choice earlier to do a “production mode”. As you can see, our .read and .write rules for the database are false, meaning no one can read or write to the database currently with simple API calls. As the comment suggests, there are ways to set up proper database authentication at <https://firebase.google.com/docs/database/security>. However, for the purposes of this guide, we will simply set .read and .write to true to quickly continue[[2]](#footnote-2).

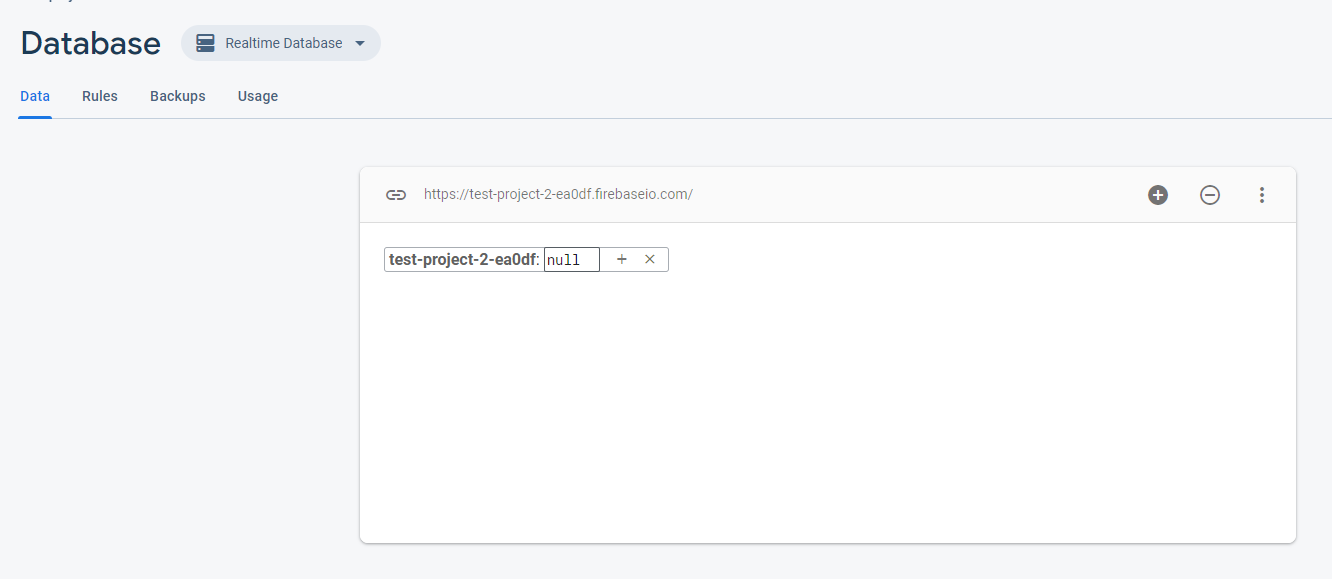


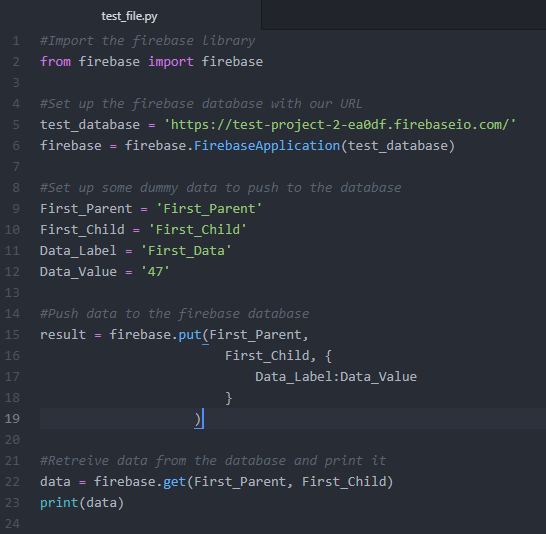
Finally, click “Publish” to make your rules live. Congratulations, you have a firebase database up and running! To view data as it comes in, simply navigate back to the “Data” tab.

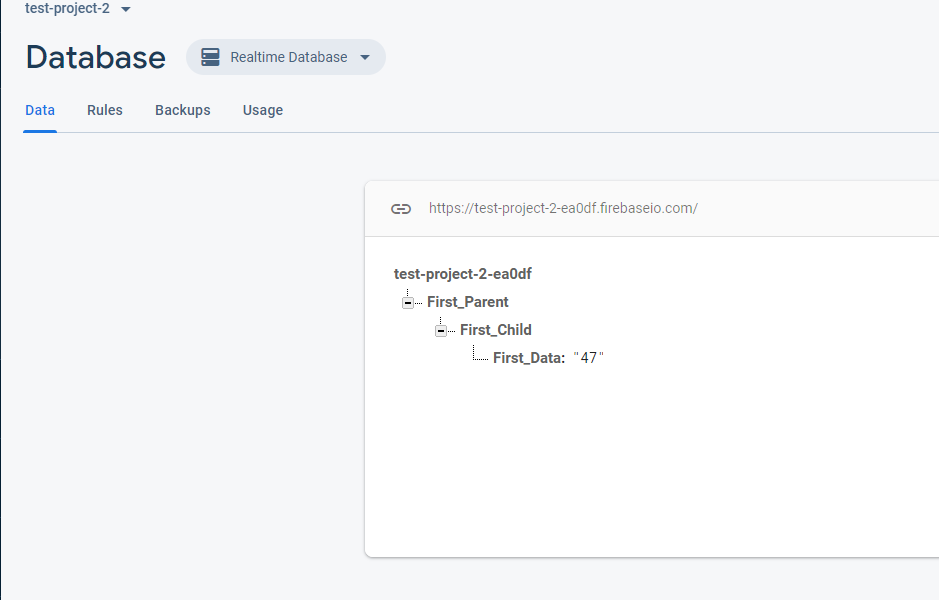


# Using Python to Interact with the Database

Python can be very straightforward to interact with a Firebase instance. If you can run Python 2, simply go to the machine you’d like to use it on and run the command “sudo pip install python-firebase”. Then, go to your firebase database and find the URL for it.



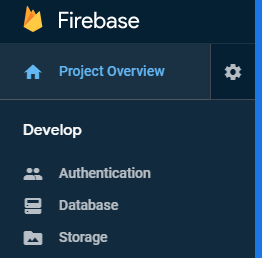
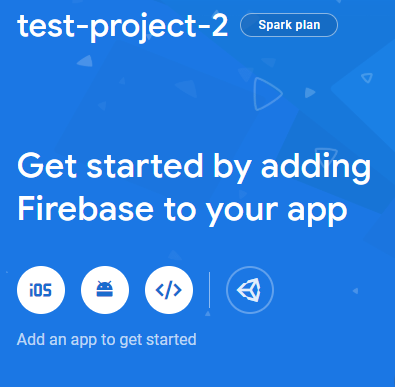
Next, write up a quick python script. The following script goes through most actions you’d want to take when interacting with the database.

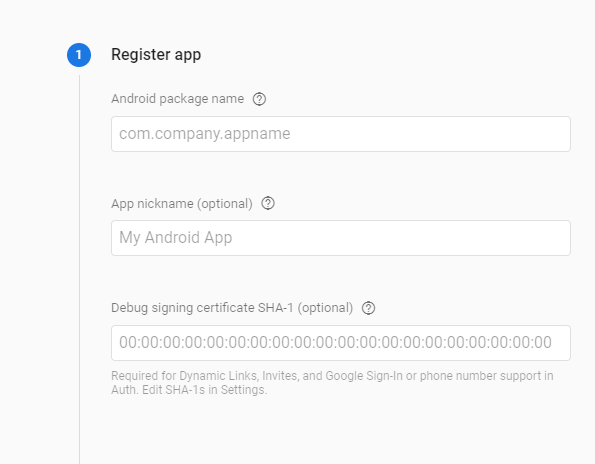
When run, the script successfully pushed to the database. You can now see it in the database when viewing it in your console.

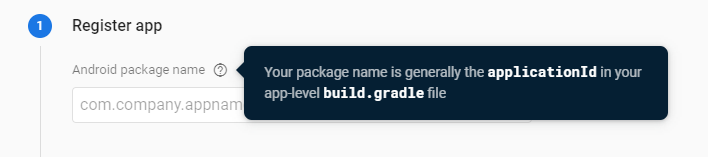
Additionally, the retrieval worked when the script was run.

# Adding Database Support to an Android Application

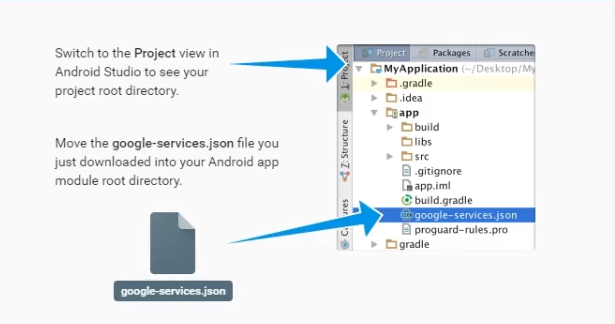
Though the process for integrating Firebase into an Android Application is slightly different from Python, it is nearly as easy. In fact, Google itself has great documentation for doing this. To start, go to “Project Overview” in your firebase console, and click the Android icon under “Get started by adding Firebase to your app”.

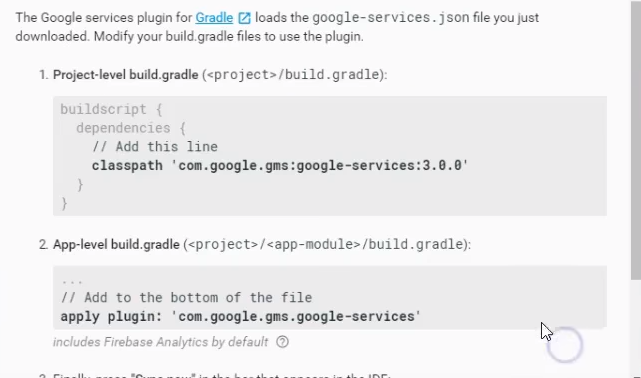


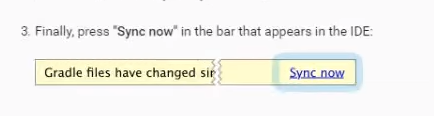
First, register your app with its Android package name. In your project, this can be found as the applicationId in the app-level build.gradle file. Don’t worry about the App nickname or signing certificate for the purposes of this guide.



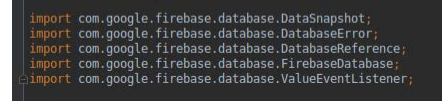
You’ll then need to download the google-services.json file and insert it into your project where Google instructs you to.

You’ll then need to follow Google’s instructions for ensuring that you import the proper plugins to use the google-services.json in your project. These seem to change over time but would look something like this.



Finally, you can press the “Sync now” button that appears and run your app to test it.

Now, to interact with your database in your app, all you have to do is:

* Import the necessary libraries
* Set up the database object
* Make get requests or push requests



# Helpful Links

* <https://www.youtube.com/watch?v=rJnm3MFVO7U&list=PL73qvSDlAVVh5MO1Bfujfb_SDPABjJ2BY&index=37>
* <https://www.youtube.com/watch?v=aPLh31MWewc>
* <https://www.hackster.io/varuldcube100/send-sensor-data-to-firebase-real-time-database-4d6b83>

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Editor | Changes Made | Contact |
| 12/15/19 | Jacob Hillebrand | Created Document | hillebrandj3779@my.uwstout.edu |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Many of the tiers are free to use, however some are not. All uses discussed in this document were free as of 12/15/2019 [↑](#footnote-ref-1)
2. DO NOT DO THIS if your project will have any longevity at all. As you can imagine, having a database that is accessible by literally anyone is a huge security hole, and is NOT acceptable in a production environment. [↑](#footnote-ref-2)