

Synchronized Shopping List

Setup and Installation Instructions

© 2018, Jim Schrempp and Bob Glicksman, Team Practical Projects
Version 1.0; date: 4/05/18

NOTICE: Use of this document is subject to the terms of use described in the document "Terms_of_Use_License_and_Disclaimer" that is included in this release package. This document can also be found at:

https://github.com/TeamPracticalProjects/Synchronized_Shopping_List/blob/master/Terms_of_Use_License_and_Disclaimer.docx.pdf



Table of Contents

Table of Contents

1. Introduction.	3
2. Making Your App From The .AIA File.	3
3. Setting Up Your Firebase Real-Time Database.	5
4. Building Your App.	18
5. Deploying Your App.	18

1. Introduction.

This document provides instructions for creating your Firebase database and for building the *Synchronized Shopping List* app that goes with it.

The *Synchronized Shopping List* is an app that you link to a Google *Firebase* Real-time Database. You share this app with people that you share shopping chores with. Everyone that you share this app with has access to the Firebase database that you create, using the instructions in this document. Once you create the Firebase database, you must insert two pieces of information into the app's source code (.aia) file that identifies the database and allows the app to communicate with it. You then build the app, using the instructions in this document, to produce an installable app file (.apk). You then share this installable app file with everyone that you share the database (thus, the shopping chores) with.

All users of the app that is built using these instructions work out of the same Firebase database in the "cloud". This is how that various users stay synchronized. Every time anyone uses the app to change something in the Firebase database, the change is reflected in the app's of all of the users that you share shopping chores with.

The app itself is written using the MIT App Inventor 2 (AI2) integrated development environment (IDE). Both AI2 and Firebase require you to have a free Google account. The AI2 IDE and Firebase are cloud (browser) based and do not require you to install any software on your computer. Both AI2 and Firebase are free to use¹.

Note that AI2 can only create apps for Android mobile devices at this time. The MIT team has promised support for iOS (Apple) mobile devices later this year.

2. Making Your App From The .AIA File.

You begin the process of building the app by importing the *SynchedShoppingList.aia* file into your AI2 development screen (IDE). You start by opening a web browser on your development computer and going to:

https://github.com/TeamPracticalProjects/Synchronized_Shopping_List/blob/master/Software/src/SynchedShoppingList.aia

¹ Firebase is free to use for a limited, basic license. The free basic license is more than sufficient for the purposes of this project.

Click on “Download” and select a location on your computer for the downloaded file (e.g. your Desktop). *Make a note of the location of this file on your computer.*

Next, you will open your AI2 IDE on your web browser by going to:

<http://ai2.appinventor.mit.edu>

If you haven't previously set up an AI2 account, do so now, following the instructions given. You will need a Google account. If you don't have a Google account, you can establish one by getting yourself a free GMAIL account. See the documentation on the AI2 web site for further information.

With the AI2 IDE open on your computer, click on “*Projects*” and click on “*Import project (.aia) from my computer ...*”. Click on “*Choose File*” from the popup dialog box and select the file “*SynchedShoppingList.aia*” from the location where you previously saved it. Now click “*Projects*”, “*Save project as ...*” and give your project a name that you can identify it by in AI2. The project name that you choose is up to you, but you should name the Firebase database that you will create the same name as you choose for this project. In this example, we name the project and the database “*SynchedShoppingList1*”, but you can name it whatever you wish.

Figure 1 is a photo of what the AI2 screen should look like at this point in the instructions:

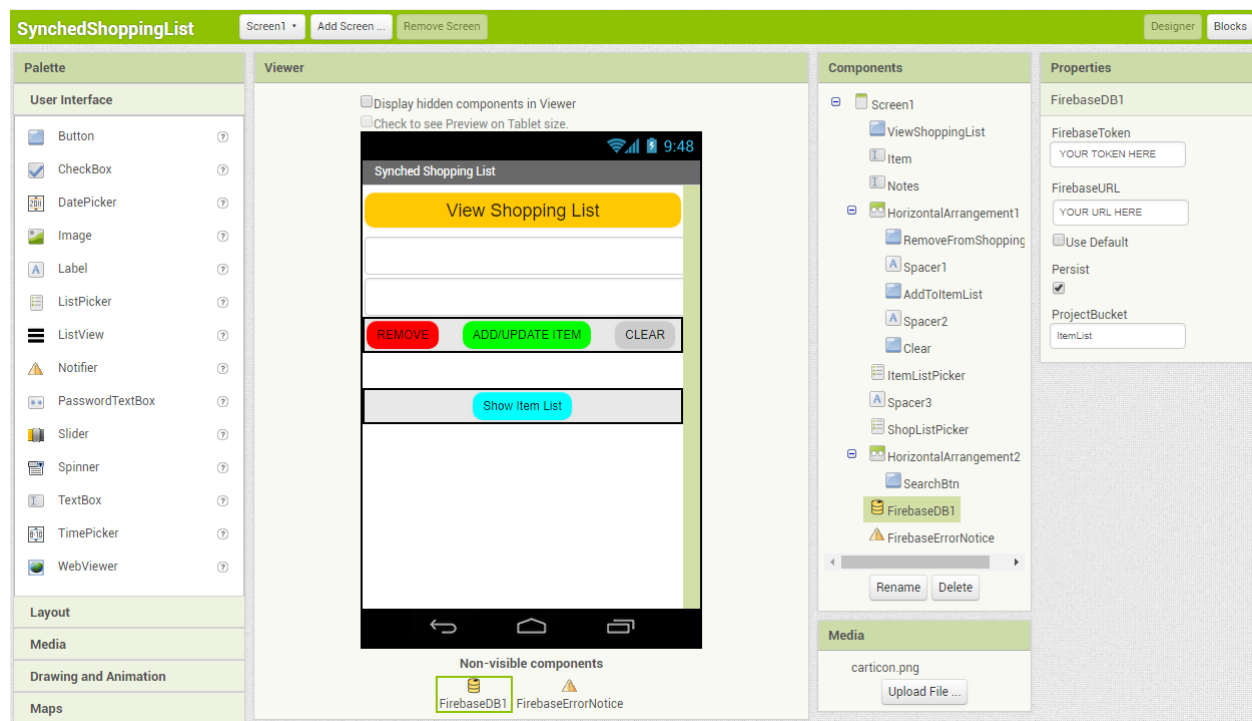


Figure 1. Photo of AI2 IDE screen with app (.aia file) imported into it.

Note the *Properties* window on the far right column of figure 1 (after clicking the *FirebaseDB1* component at the bottom of the Designed Window of the screen). There are two fields that you will need to fill in here:

- *FirebaseToken*
- *FirebaseURL*

You will need to set up a Firebase Realtime Database in order to obtain this information. Leave the AI2 IDE window open on your browser and then proceed to set up a Firebase project using another window on your web browser.

3. Setting Up Your Firebase Real-Time Database.

Open another window on the web browser on your computer and go to:

<https://console.firebase.google.com>

You will be prompted to log in to your Google account if you have not already done so. After logging into Google, your screen will look similar to the photo in figure 2.

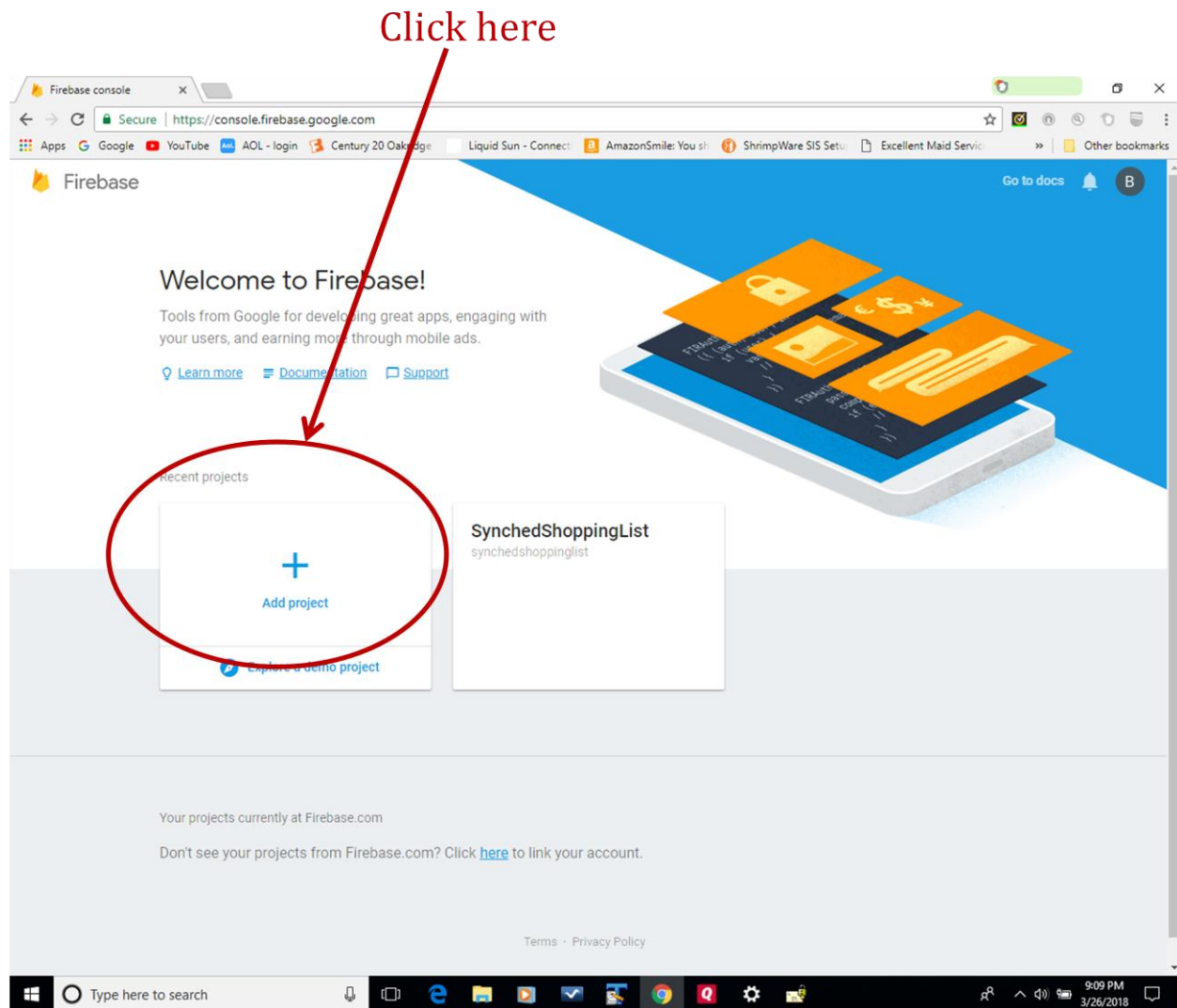


Figure 2. Console Opening Screen.

Click on the "Add project" graphic, as shown in figure 2. You will now see a screen similar to figure 3.

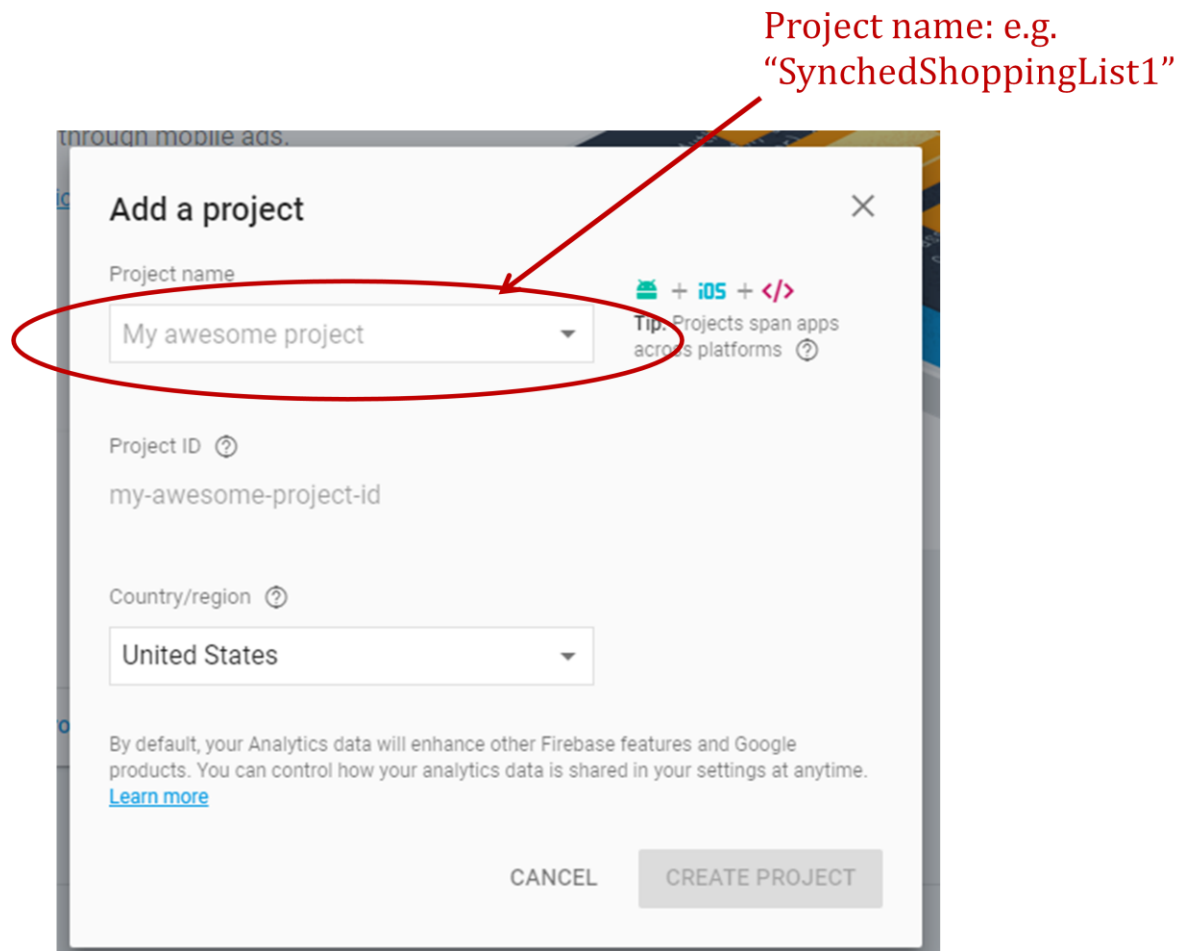


Figure 3. Adding your project.

Type the name of your project in the *Project Name* text field as shown in figure 3. Use the same name as you chose for your AI2 project in the AI2 IDE. In this document we use *SynchedShoppingList1*, but you can choose whatever name you wish. Next, click on *Create Project*, as shown in figure 4.

Add a project ✕

Project name

SynchedShoppingList1 ▼

Project ID ⓘ

synchedshoppinglist1 ✎

Country/region ⓘ

United States ▼

By default, your Analytics data will enhance other Firebase features and Google products. You can control how your analytics data is shared in your settings at anytime. [Learn more](#)

CANCEL **CREATE PROJECT**

Click "Create Project"

Figure 4. Create the project.

Your new Firebase project will be created, as shown in figure 5. Click *Continue* to proceed.

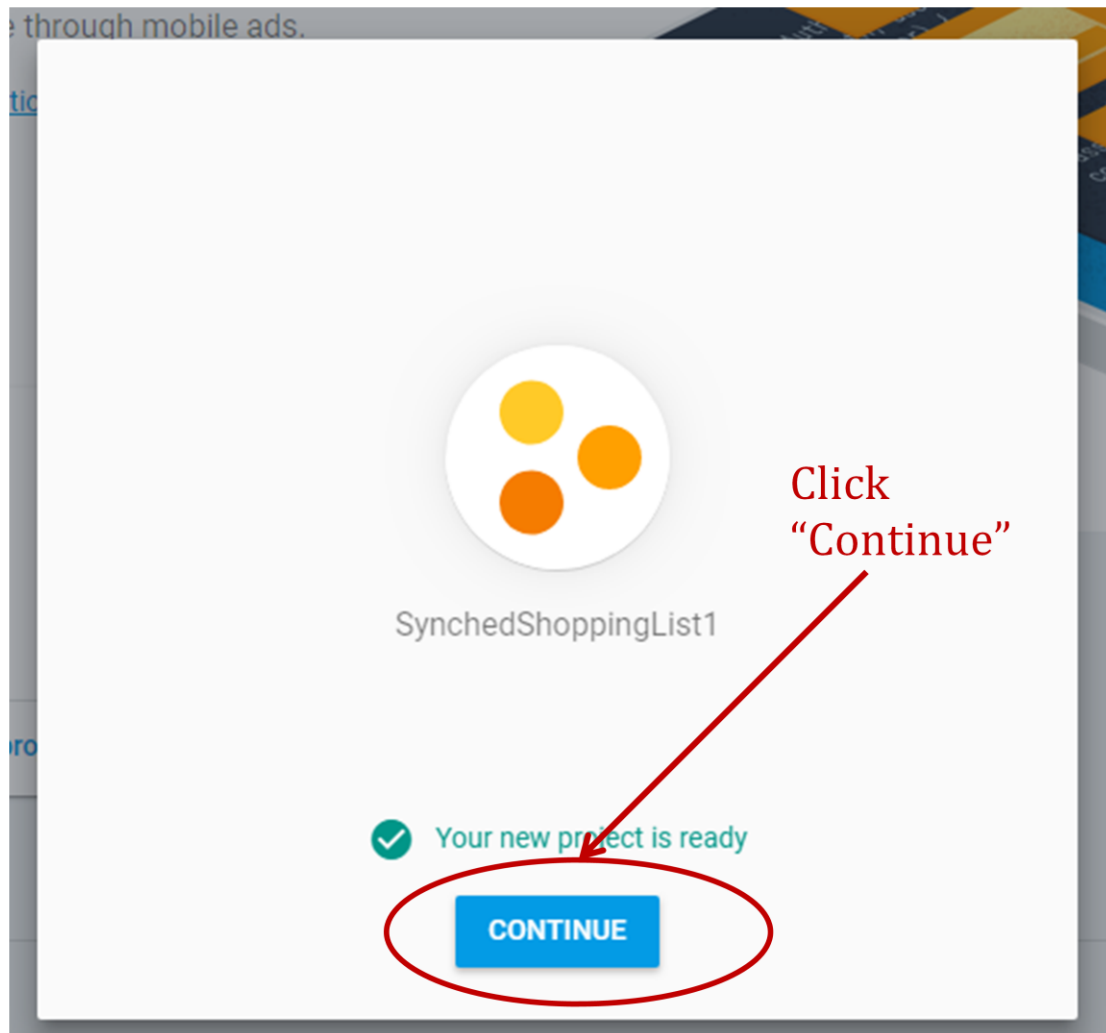


Figure 5. Finish project creation.

At this point in the process, you have created a Firebase project but you don't have any Firebase components in your project. You need to add a Realtime Database component into your Firebase project. Select *Database* on the left side of the screen and click on *Get Started* in the *Realtime Database* panel in the center of the screen, as shown in figure 6.

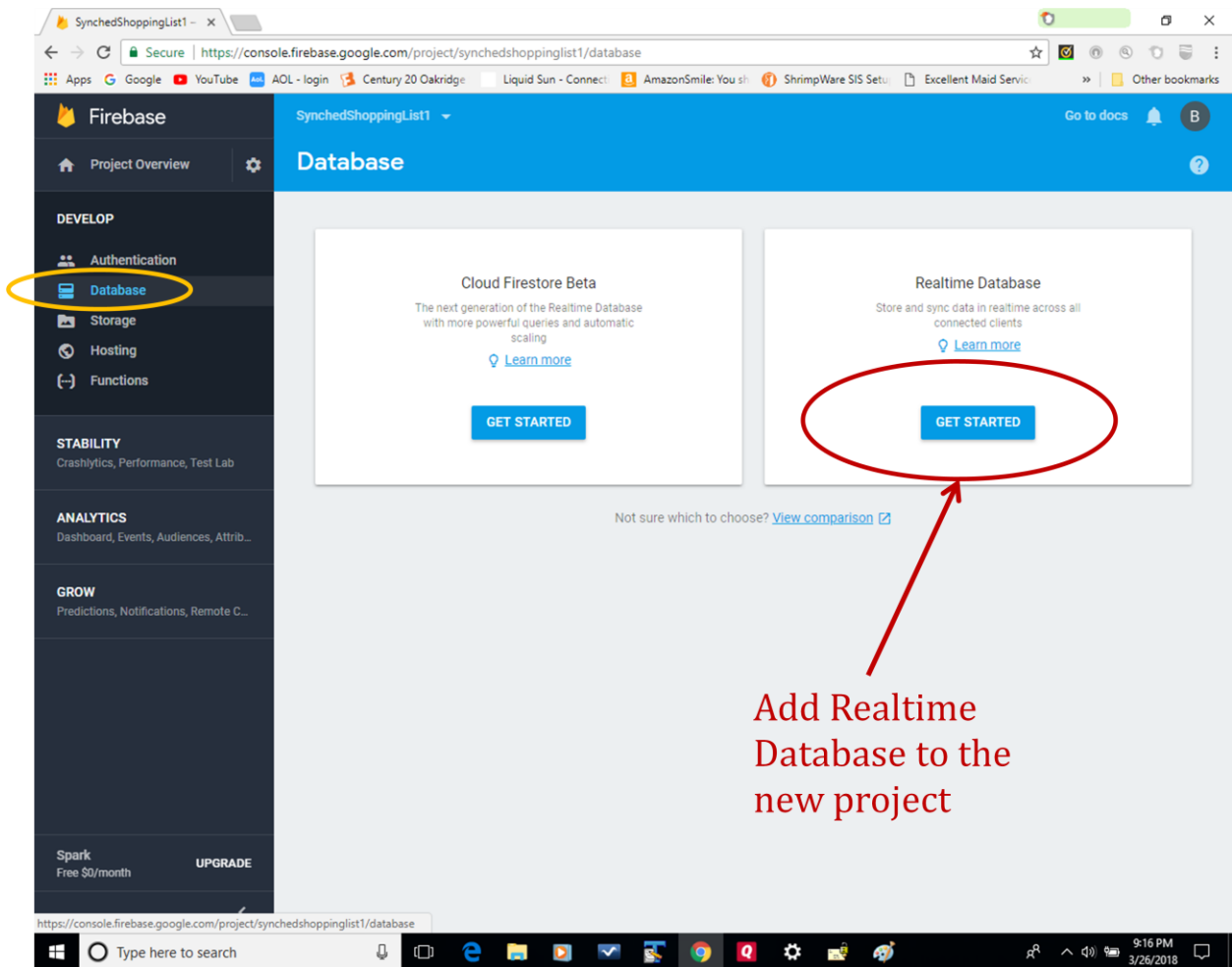


Figure 6. Adding a Realtime Database to the project.

You can now open the new Realtime Database by clicking on *Database* in the panel of the left side of the screen, as shown in figure 7.

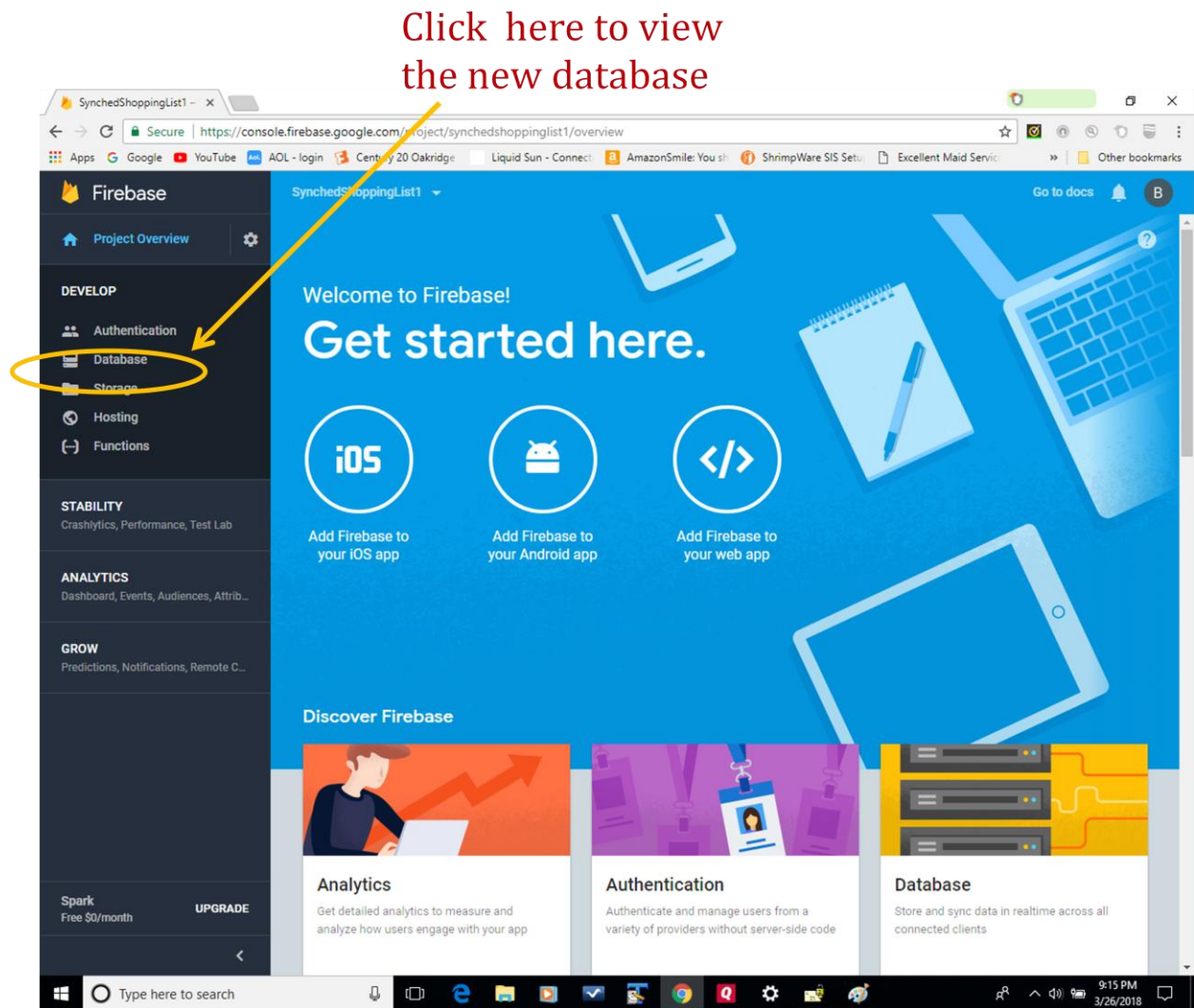


Figure 7. Opening the new Realtime Database.

After opening the Realtime Database, you should see the empty database, similar to the photo in figure 8.

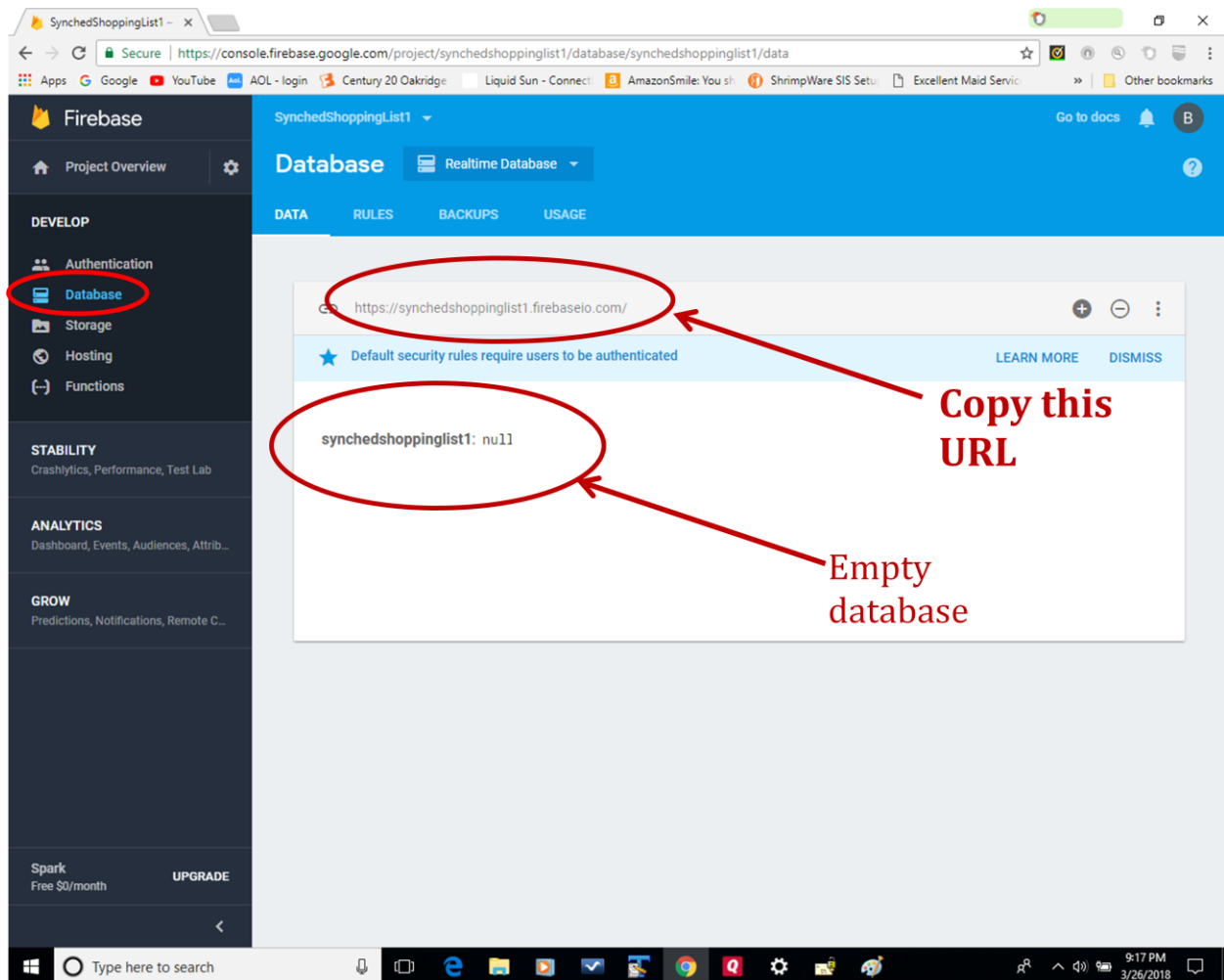


Figure 8. Finding and copying the URL for the app to use.

Copy the URL at the position shown in figure 8. Copy this URL to your computer's clipboard. Be careful to copy the URL where shown in figure 8 and NOT the URL in the browser's URL pane! Now go back to your AI2 IDE window and paste the URL into the *FirebaseURL* field in the *Properties* window of the AI2 IDE, as shown in figure 9.

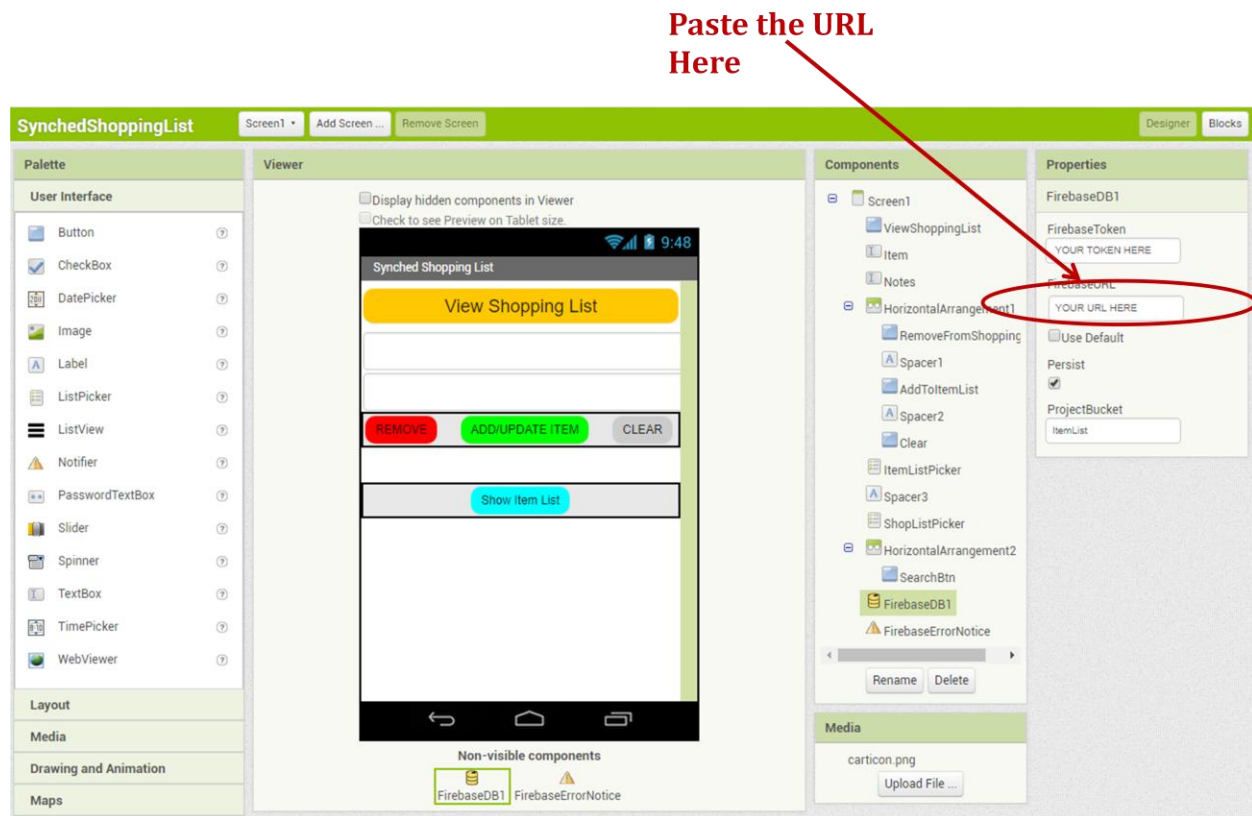


Figure 9. Pasting the URL into the AI2 project.

Next, you will need to change the Firebase access control rules for your Firebase project. Firebase has a number of ways of controlling access to your project. Most of these methods require the user to log in to Firebase, either directly or via another service (e.g. Google, Facebook, etc.).

This project does not use Firebase's internal access control services. Rather, the project will be made "public" as far as Firebase is concerned, but access control will be restricted to only those people who you give your app to, via a project-specific "*Web API key*" that is included in the app.

First, you will need to change Firebase's access control rules from the default to *public*. So click on *Rules* to show the default access control rules, as shown in figure 10.

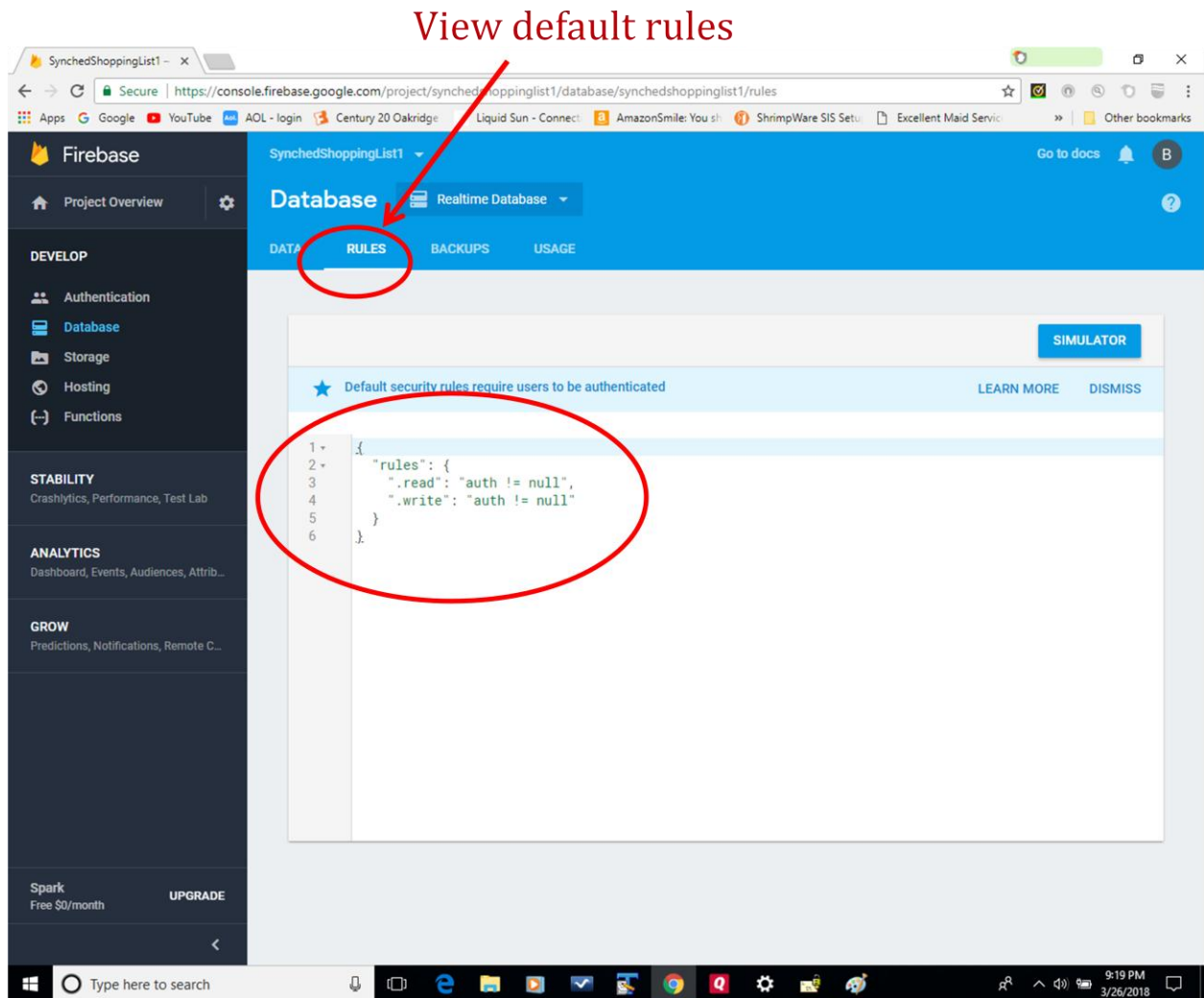


Figure 10. Locating the access control rules.

The Rules window is just editable text. Edit the text to say *true* for both the *.read* and *.write* tags, as shown in figure 11.

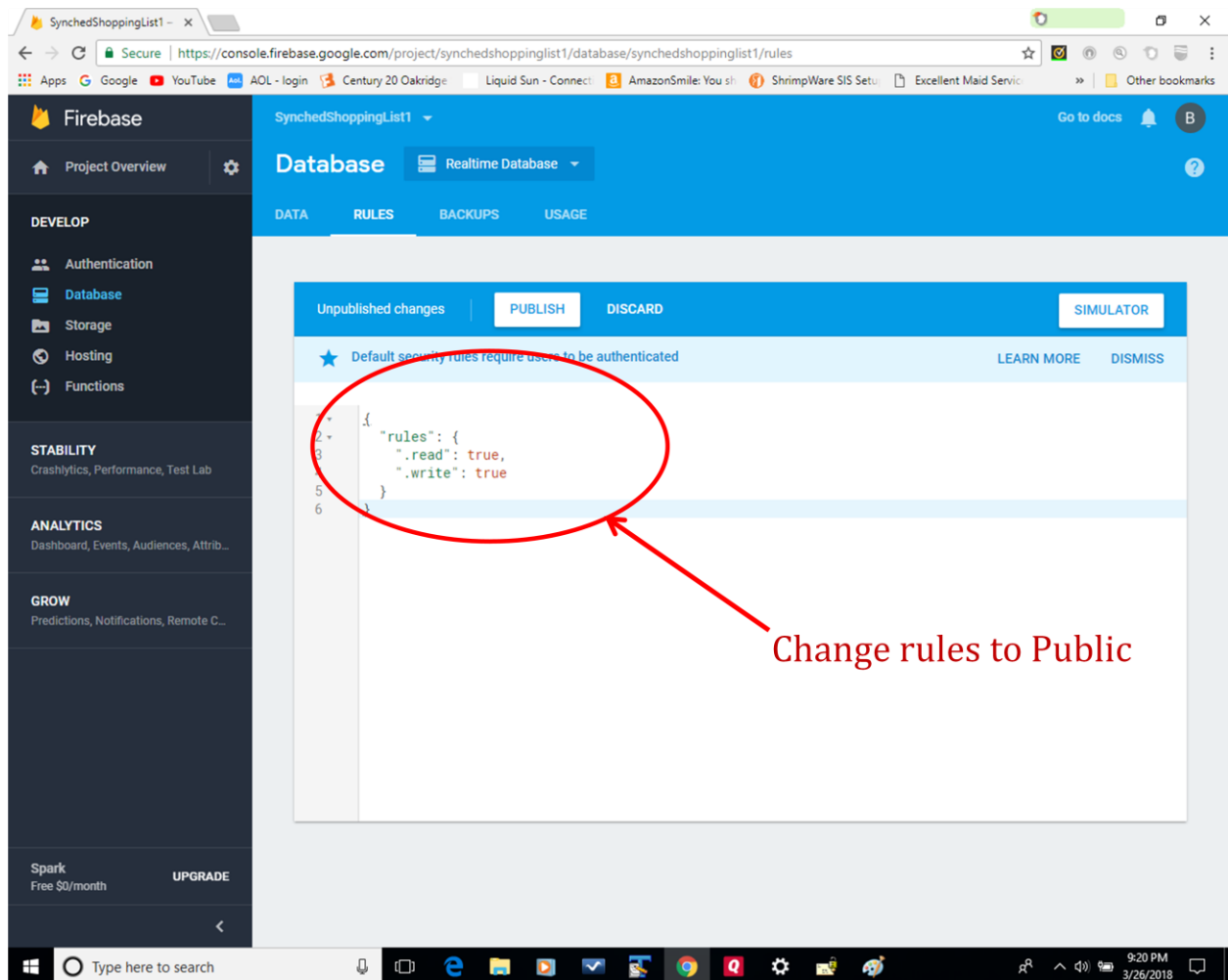


Figure 11. Changing the access control rules to public.

After editing the text as shown in figure 11, click on *Publish*. Then, note the message about the new rules being public, as shown in figure 12, and dismiss this message after verifying that the rules are indeed public.

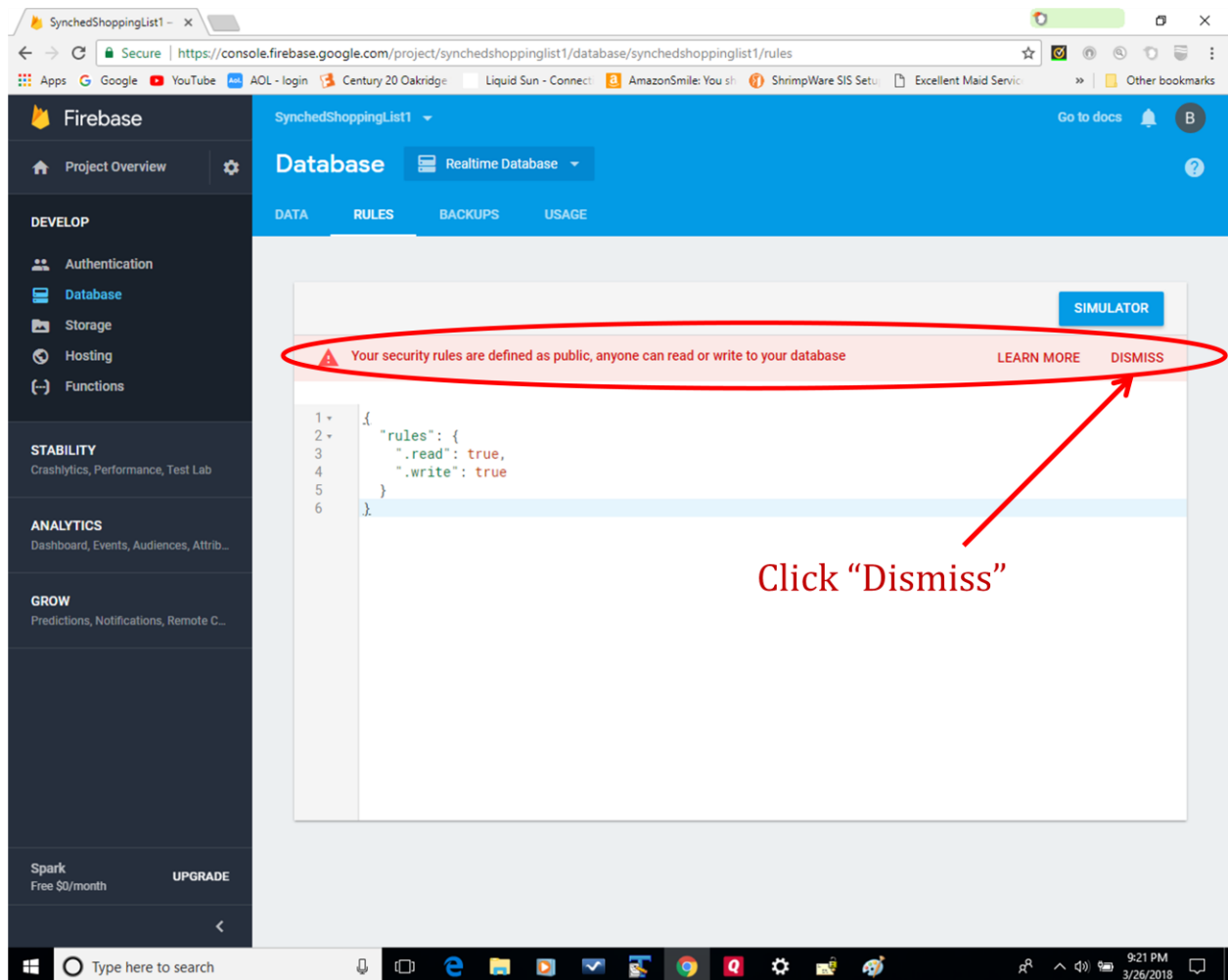


Figure 12. Dismiss the public message.

The final step is to obtain a project-specific access control *Web API Key* for your Firebase project that you will include in the app. Click, the little “gear” symbol as shown in figure 13 to go to the *Settings* window and locate the “Web API Key” where shown in figure 13.

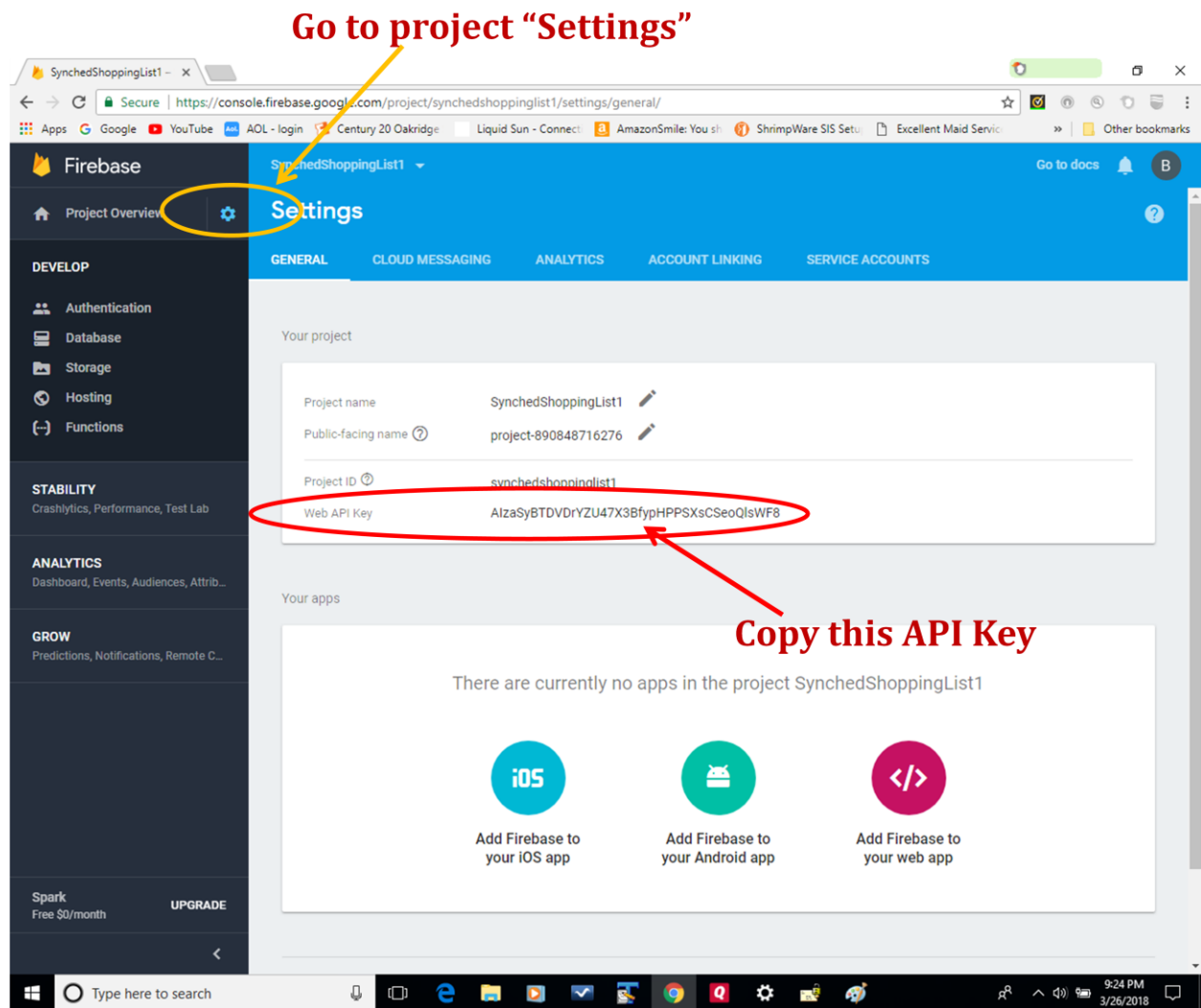


Figure 13. Finding and copying the API Key.

Copy the *Web API Key* to your computer's clipboard, then go to the AI2 IDE window on your computer and paste the *Web API Key* into the *FirebaseToken* field in the AI2 *Firebase Properties* pane, as shown in figure 14.

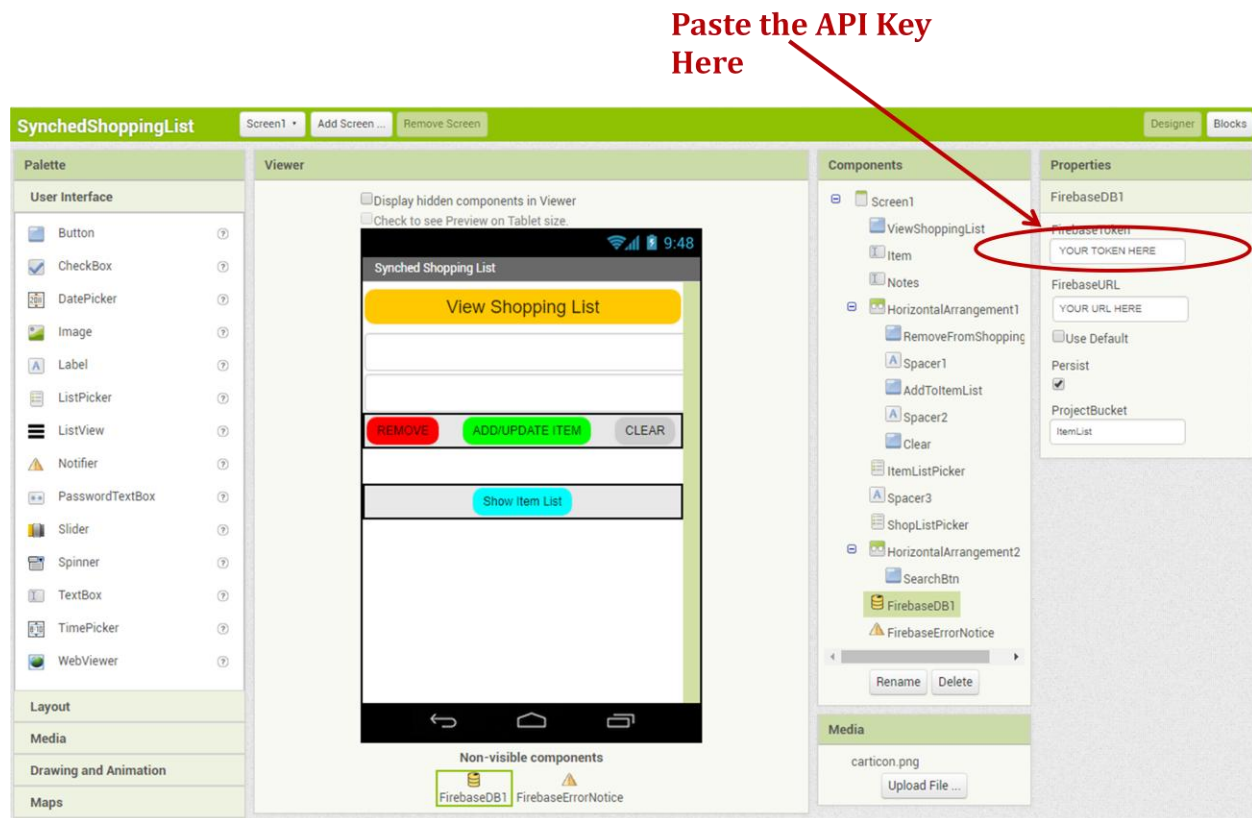


Figure 14. Pasting the API Key into the AI2 IDE screen.

Your app is now complete. You are ready to build your app and deploy it!

4. Building Your App.

You can now close the Firebase window on your computer's web browser, but leave the AI2 IDE window open. At the top of the AI2 IDE window, click on "Build", then "App (save .apk to my computer)". Specify a location on your computer where you can find this file later, such as your desktop. The app will take a while to build – be patient. When the build is complete, a file with the name *YourProjectName.apk* will be in the location that you specified (where *YourProjectName* is the name that you originally selected for this project).

5. Deploying Your App.

Once you have built the .apk file for the app, you will need to get this file onto your Android device and onto the device(s) of all people who will be sharing the shopping list in the Firebase database that you just created. There are several ways to distribute this app. You can upload the .apk file to an Android device via USB cable from your computer, or you can e-mail the .apk

attachment to yourself and the people who will be installing this app on their device(s). Either way that you do this, each person should place this .apk file on their mobile device in a location that they can find using the *MyFiles* app on their device. For people that have an SD card installed on their device, we recommend placing the .apk file in the root directory of the SD card, i.e. /SDcard. However, the actual file location does not matter, as long as the device user can find it after saving it there.

Each device user must then use *MyFiles* to locate the .apk file on their device and tap on this file. The app will then install on the mobile device. NOTE: the Android OS will likely popup a message about the app not being “trusted” and some warning about damage that untrusted apps can do to the device. Follow the instructions on this popup message to install the app anyway (it is safe!). The app should install just fine.

After installing the app on each device, each user may wish to create a shortcut to the app on their home screen. Use Android’s app explorer to find the app, *tap and hold* the app icon until a graphic of the home screen appears and drag the app icon to the place on the home screen where the user wants the short cut. This places a shortcut to the app on the home screen; the app is still available via the Android app explorer.

The app is now installed on an Android device. Opening the app will allow the user to add items and remove items from the Firebase database that you created and the data in the Firebase Realtime Database is then shared and synchronized among all users of the app. Please see the accompanying *User Manual* for instructions about how to use the *Synchronized Shopping App*.
