



FortNotification

**Team Members:**

s3588952 – Isaac Underhill  
s3602940 – Alex Rosenzweig

**Contributions:**

50/50

**Links:**

Github: <https://github.com/isybub/fortnotification>

FortNotification: <http://www.fortnotification.appspot.com>

**Project**

Option 1: Development of a cloud application using your own idea and strengths

## Summary

The purpose of our project is to use the Fortnite API, and quite a few of Google’s APIs to create a meaningful website that fills a current gap in the market. There is no current easy alternative to FortNotification, meaning it has the potential for marketing, advertising and real-world usage.

## Introduction

Fortnite is a currently popular video game, that can be played on many different machines and computers. It has a player base of approximately 125 million people.

This game has an internal “Store”, where players can spend virtual currency to purchase “Skins” and “Items”. These skins and items are rotated through the store on a 24-hour basis, and no two days will have the same items. There are currently nearly 400 items in the game that will be sold on the store, and as such it can take quite some time for certain items to arrive on the daily rotation.

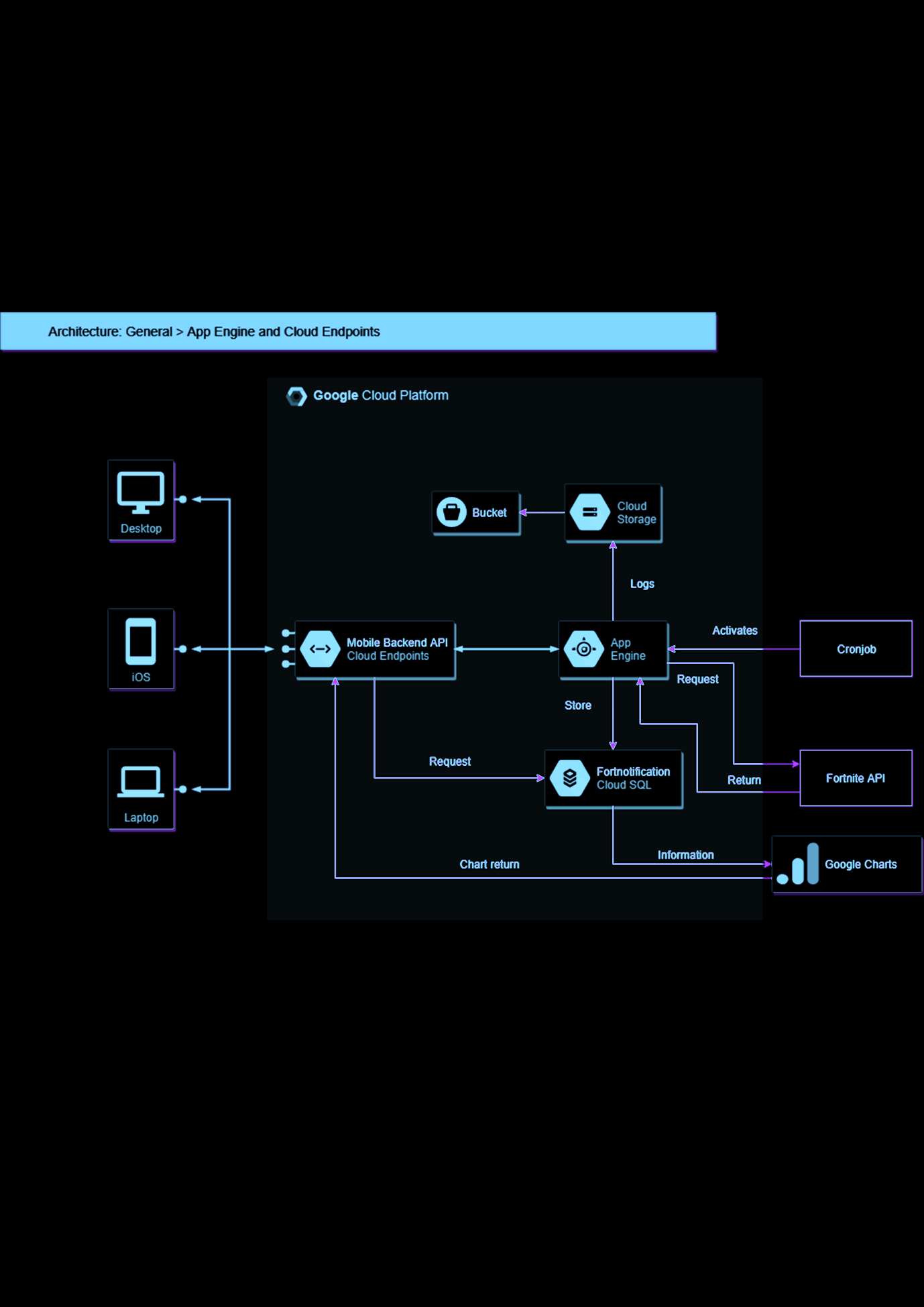
This leads to an issue. Players want a specific item. Players will have to log into the game every single day for an undetermined amount of time to purchase that specific item.

Here’s where FortNotification comes in. A user will log in with Google Authentication, list the few items they wish to purchase, and each day as the store rotates, if it contains one of the items that the user wants, they will receive an email ASAP describing the items in the store.

This means users can rest easy knowing they will be able to purchase the item they want, and not have to log into the game every single day to check.



## Architecture

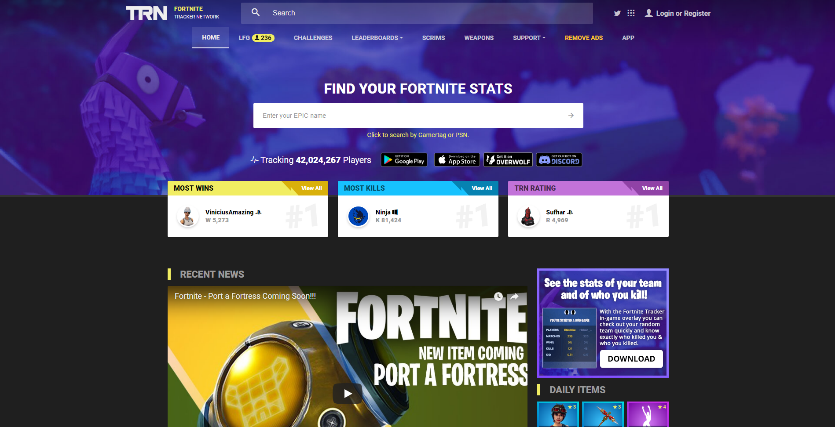
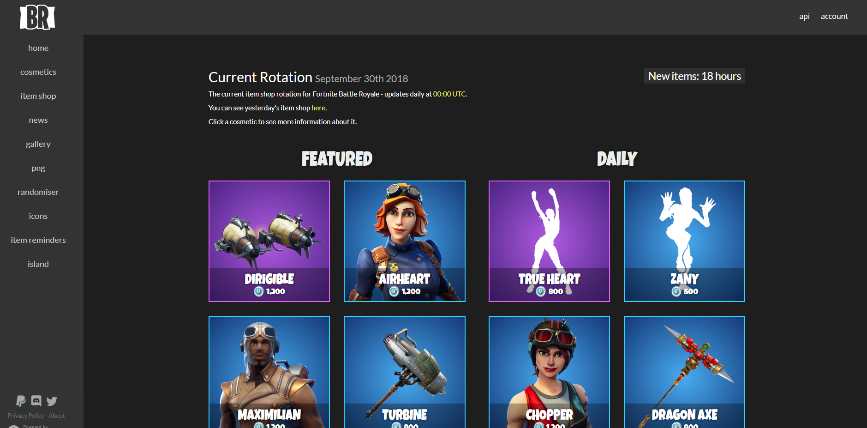


The website is hosted on Google Apps Engine, using the SQL API, Google Storage API, Charts API, User Service API, and the Fortnite API. Most of these services are backend, and the user will not notice, aside from google charts.

The user will access the website, which runs a call to the Google Charts API, to display the top 5 most popular “requested” items. The user may then login to the website, using the Google User Service API. Having logged in, the user may continue to their requested items.  
On the item selection page, the user is greeted by a text area where they can specify the items they wish to “request” from the store. Entering this data begins the backend process as follows:  
Process the input and itemize each point as separate elements in an array, store the information using google SQL API, and save the user’s information using Google Storage API.   
Each morning at 10:05AM a cronjob will be run to get the information from the Fortnite API about the items in the store on that day. These items will be compared against users and then emails are sent out to the users that have matching items in the store they “requested”.

# 

# Related work

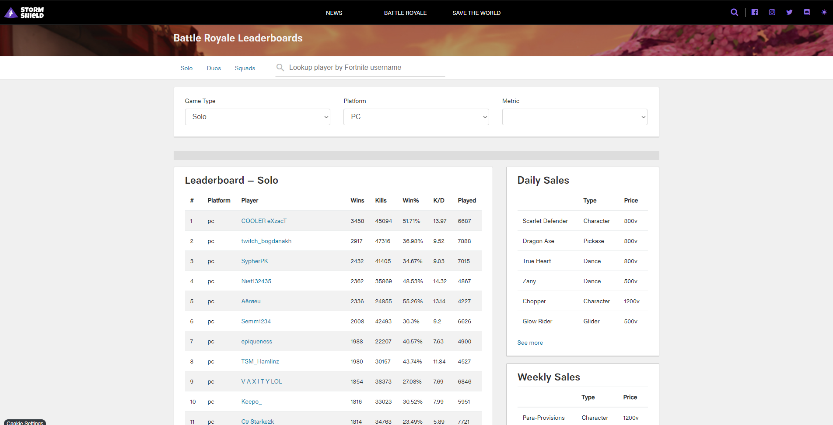
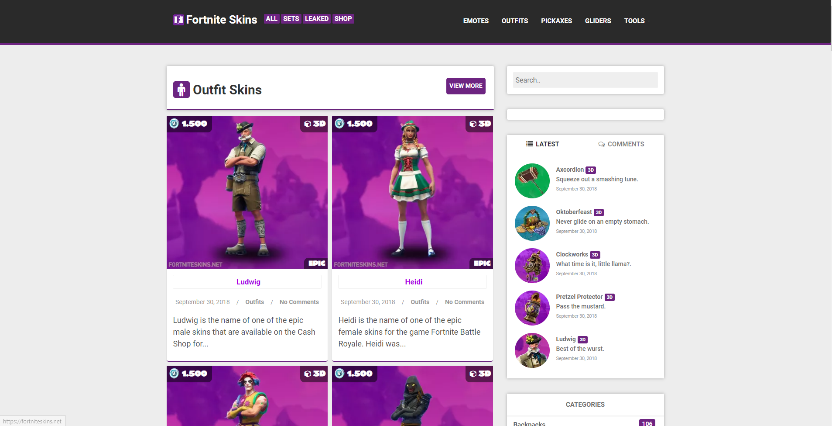


<https://fnbr.co/shop/>

A hub to view the store  
items online rather than logging into the game.   
This still takes time, and a good memory to check each day. Lacking a notification feature.

[https://fortnitetracker.com/](https://fortnitetracker.com)

Using a very similar Fortnite API, players can get their stats of the last season and compare with other players. Similarly lacking a notification feature.



<https://www.stormshield.one/pvp>

Fortnite Tracker clone – slightly less favourable website design, however simpler to look at with less on the screen at once.

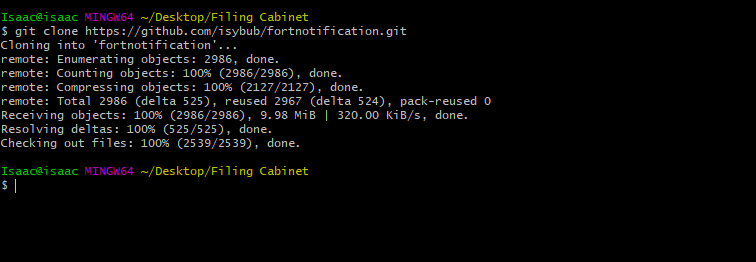
<https://fortniteskins.net>

Also using Fortnite API, listing all of the items that have ever been available in the store ,and the ones in the store right now.



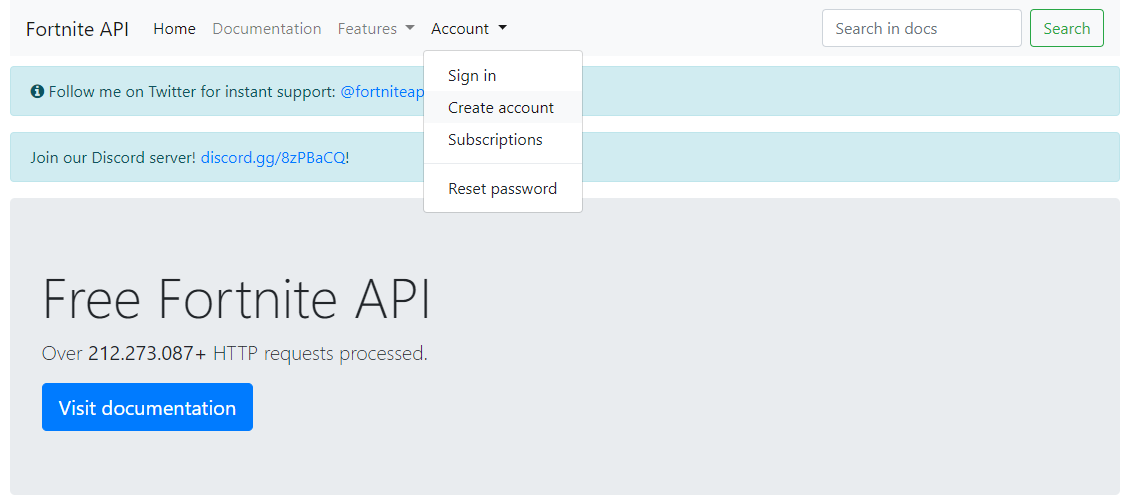
## Step 1.

Clone the repository



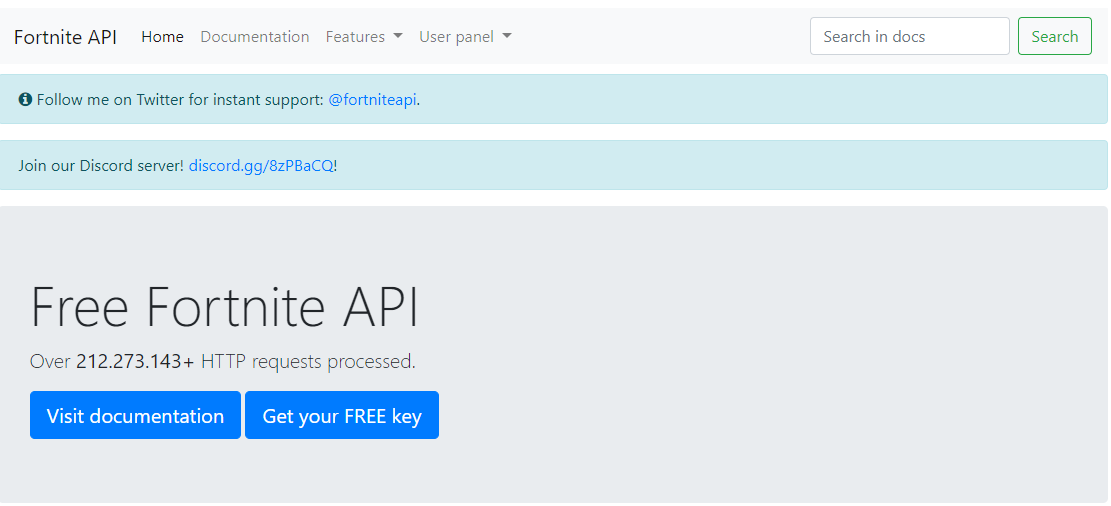
# Step 2.

Create an account on <https://fortniteapi.com>



## Step 3.

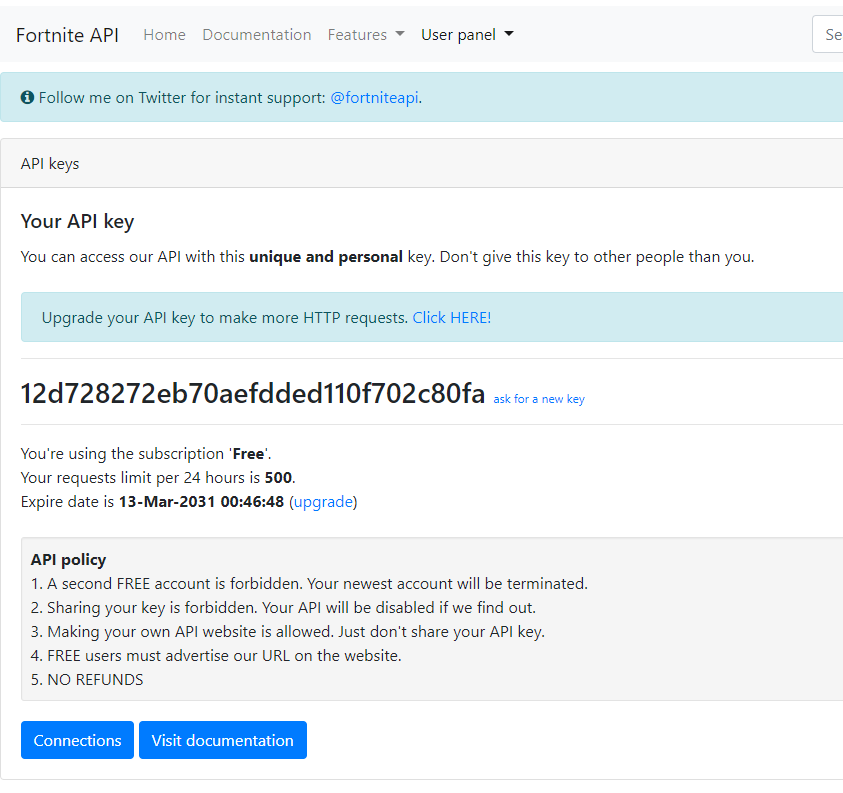
Once logged in, “Get your FREE key”



# 

# Step 4.

Copy your new key to clipboard



## Step 5.

Paste into the setKey function in your local files



# Step 6. Last step.

Follow the Cloud App Deployment PDF tutorial from RMIT’S Cloud Computing class, tutorial #3

<https://rmit.instructure.com/courses/17760/files/4498035/download?wrap=1>



## User Manual

Alright! Once all of this has been completed, you, and your users are ready to sign up. Simply open your web browser and navigate to <https://www.fortnotification.appspot.com>

1. Click on the “Sign in or register” Button.
2. Click “Go to my subscribed items!”.
3. Enter a list of items you wish to be notified of, for example “Fresh, Work It Out”.
4. Click submit.

And you’re all done!

FortNotification will now send you an email when the item store has the items you asked to be notified of. It’s that simple.

If you wish to change the items at any point, follow these exact same instructions, and the items you have already requested will be shown, and you can add, remove or keep the same items.

## ­­

## References

1. <https://cloud.google.com/sql/docs/>
2. <https://cloud.google.com/appengine/docs/php/>
3. <https://cloud.google.com/appengine/docs/standard/php/googlestorage/>
4. <https://cloud.google.com/appengine/docs/flexible/php/using-cloud-storage>
5. <https://cloud.google.com/php/getting-started/using-cloud-sql-with-mysql>
6. <https://cloud.google.com/sql/docs/mysql/quickstart>
7. <https://cloud.google.com/appengine/docs/standard/php/mail/>
8. <https://cloud.google.com/appengine/docs/standard/python/config/cronref>
9. <https://cloud.google.com/storage/docs/>
10. <https://cloud.google.com/appengine/docs/standard/php/googlestorage/advanced>
11. <https://cloud.google.com/php/getting-started/using-cloud-storage>
12. <https://cloud.google.com/pubsub/docs/?hl=en_GB>
13. <https://cloud.google.com/storage/docs/access-public-data>
14. <https://cloud.google.com/storage/docs/access-control/lists>
15. <https://googleapis.github.io/google-cloud-php/#/docs/google-cloud/v0.36.0/storage/bucket>
16. <https://fortniteapi.com>