

CCS Lab FAQ: Using Google App Engine to host websites

Lauren Kennedy
School of Psychology
University of Adelaide

Abstract

This document is intended to be used a step-by-step guide to using Google App Engine to host experiments. It assumes that you are working using the Experiment Base available here <https://github.com/drewhendrickson/gae-experiment-base>. This document will cover how to start use the google app engine web interface, how to get the sample experiment running, and how to download data. Information describing the finer grade detail of website design will be contained elsewhere.

Creating an Application

1. Go to <https://appengine.google.com/>. You will need a Google email account(if you don't have one, you can sign up for free).
2. Sign in with you google account. You will be shown a list of your current applications.
3. To create your first application, click on **Create Application**.
4. You should be directed to a form like in figure 1
5. You only really need to focus on the first two boxes. The first is the web address that participants will use to access your website. So the screenshot above, we would be creating an address `adelaide-clcl-example.appspot.com`. This needs to be unique (you will not be able to proceed unless it is). The next box contains the text your participants will see in the tab of your website. In this case it will say **Adelaide GAE Example**. This does not need to be unique.
6. Once you click create your application will be created. If you go back to your google app engine account, it should now be on your list of applications. If you click on the name of your app, it will take you to a dashboard. The dashboard contains a lot of useful information about what resources the app is using(more below). At the moment, the app's status says **None Deployed** . That's normal, it just means that you haven't deployed any code to describe your website yet.

The Three Languages Groups of Website Development

This section is basically to say that this guide will be most useful if you're using our Experiment Base.

Google app engine lauren.kennedy729@gmail.com | [My Account](#) | [Help](#) | [Sign out](#)

Create an Application

You have 18 applications remaining.

Application Identifier:
 .appspot.com **Yes, "adelaiade-clc-example" is available!**
All Google account names and certain offensive or trademarked names may not be used as Application Identifiers. You can map this application to your own domain later. [Learn more](#)

Application Title:

Displayed when users access your application.

Authentication Options (Advanced): [Learn more](#)
Google App Engine provides an API for authenticating your users, including Google Accounts, Google Apps, and OpenID. If you choose to use this feature for some parts of your site, you'll need to specify now what type of users can sign in to your application.

Open to all Google Accounts users (default)
If your application uses authentication, anyone with a valid Google Account may sign in.

Restricted to the following Google Apps domain:

e.g. foo.com
 If your application uses authentication, only members of this Google Apps domain may sign in. If your organization uses Google Apps, use this option to create an application (e.g. an HR tracking tool) that is only accessible to accounts on your Google Apps domain. This option cannot be changed once it has been set.

(Experimental) Open to all users with an OpenID Provider
If your application uses authentication, anyone who has an account with an OpenID Provider may sign in.

Figure 1. Form to create an application with Google App Engine

Your online experiment works by your participant (the client side) sending a call to the the server (which is Google App Engine for us), which returns some code to display your experimental stimuli. At some point in the experiment, the clien will probably will need to send some data (survey answers, response times etc.) back to the server, which you (the researcher) will later download.

The server side language that we use in the Experiment Base is Python. You shouldn't need to touch the python script for simple experiments. Other server side languages supported by Google App Engine are PHP, JAVA and GO.

The client side languages that we use are HTML(displays static text) and JavaScript(can interact and change dynamically without calling the server).

We also use CSS to style to experiments we make. This is similar to creating a template that says how titles, main texts and other html objects should look.

Creating an Experiment Locally.

There is more than one way of doing this. In this guide we will use the Google App Engine SDK for Python. This allows us to run the experiment locally (i.e. just on your computer) for testing before deploying it (so others can see it). As deploying can take a few minutes(depending on file sizes and the number of images used), this makes development quicker.

1. The first thing that you will need to do is install Python on your system. See <https://www.python.org/downloads/> for details.

2. Download and install the Google App Engine SDK for Python. <https://cloud.google.com/appengine/downloads>

GAE HOW TO

3. Go to Drew Hendrickson's Github page <https://github.com/drewhendrickson/gae-minimal-exp> and either clone the gae-minimal-experiment (if you're a Github user) or just download the zip file and unzip onto your computer.
4. Go to File then Add Existing Application. The only thing you need to change is the Application Path. Click Browse... and find where you saved the unzipped gae-minimal-experiment. Within this file find the exp file and set that as your Application Path.
5. This App should now be in the list of Apps in your SDK module. If you highlight the app, you should find some of the greyed out icons become active.
6. Click Run and wait until the bullet next to your app turns green (may take a few seconds)
7. Once it has, click the world icon, Browse, to open your online experiment locally.

Putting your Experiment up Online

1. Open your preferred text editor (our lab members use a mix of EMACS, Sublime and brackets).
2. Open the `app.yaml` file (this should be in the `exp` folder of the experiemnt base you downloaded).
3. You only need to change one thing in order to deploy your experiment. In the first line change the application name to the unique extension you created when you created your app. In our case it should be changed to `adelaide-clcl-example`.
4. Now if you go back to your App Engine Launcher and click the blue arrow pointing up `deploy`, you can deploy your code to your application. You'll need to sign in using the Google Account you created the application in.
5. You should now be able to go to your location. If you're not sure that it looks the way it ought to, our example is currently running at `adelaide-clcl-example.appspot.com`

Downloading Data

1. Log back into your online Google App Engine Account and click you App name to get to the Dashboard for your app. The dashboard contains a lot of information about the resources that your app is using.
2. Complete the short survey in the sample experiment a few times. Now you should have stored some data on the app engine. If you click on `Datastore Viewer`, you can see this data. Each participant is a row in the Datastore.
3. If you run a Mac or Linux OS, enter the code below into the command line to download the data.

```
appcfg.py download_data --config_file=bulkloader.yaml --filename=data.csv
--kind=DataObject --url=http://<app_name>.appspot.com/_ah/remote_api
```

4. If you run a Windows OS, you will first need to install a Unix emulator such as Cygwin www.cygwin.com. Make sure you elect to install the python app during the install (even if Python is already installed on your computer). In the cygwin terminal navigate to your experiment folder, and then enter the above line to download your data.

Billing

1. If you are running your experiment on a crowd source website such as Amazon Mechanical Turk, we advice that add a credit card(can be Autralian) to your account. This is like insurance in case you exceed the free allowances (which can happen because so many people visit the site in a short period of time).

2. To do this, go to the Dashboard for your experiment. It should look something like in figure 2

3. Click **settings** next to billing status

4. You should have a banner telling you that you **Billing for this application is managed from the Google Cloud Console**. Click [here](#). You **DO NOT** need to create a premier account.

5. Here you can select to **Enable Billing**, and link up a credit card.

6. You will be given an option to select a cap of how much you want to spend per day. You will be charged for usage up to this amount, after which your data will no longer save. It is generally a good idea to allocate more than you think you'll use (depending on the resources your experiment needs, \$20 should be more than sufficient), but something you would be willing to pay if a bug in your code causes you to hit this cap.

7. You will need to do this for every app you run seperately.

GAE HOW TO

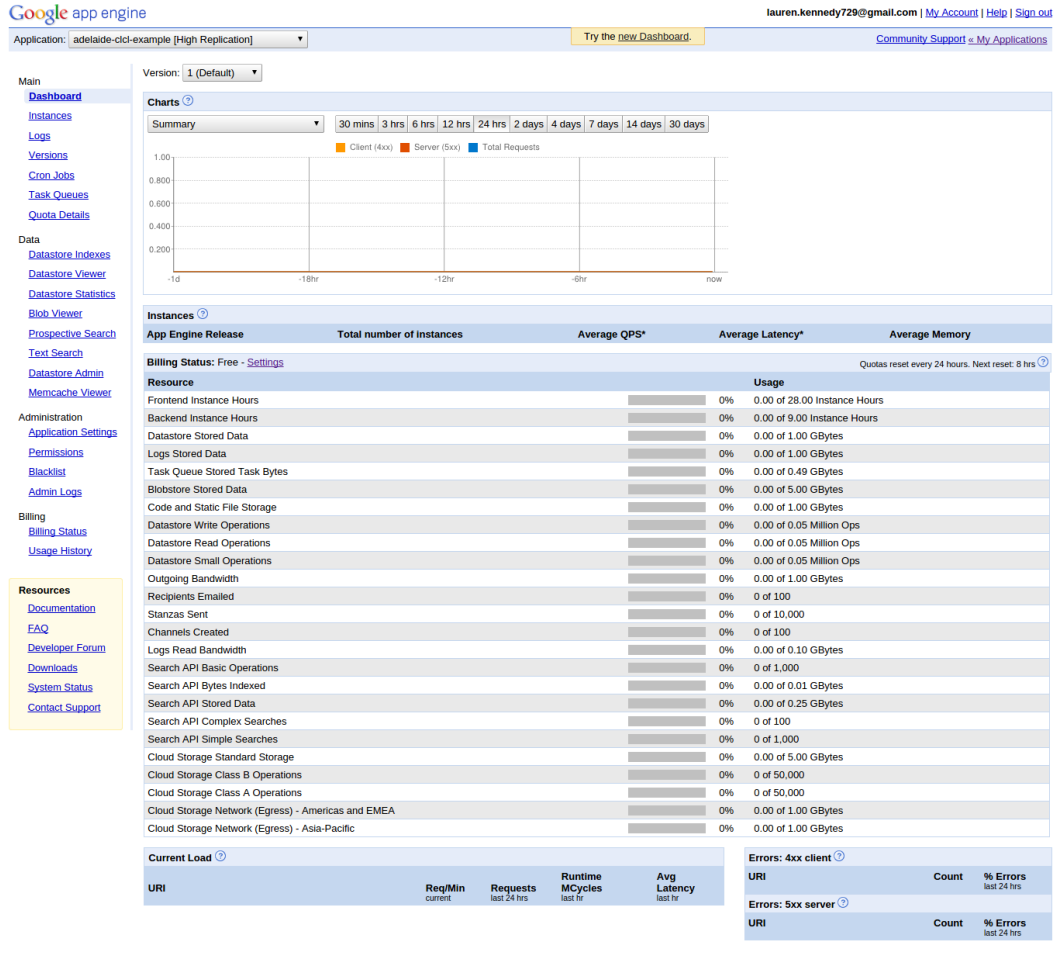


Figure 2. Google App Engine Application Dashboard