

# Google App Engine

Scary\_\_

November 2019

## 1 Abstract

## 2 Introduction

Google App Engine is one of the most used Cloud Computing Providers out there. This Paper looks a bit into what makes it attractive and features and offers a look into the development history of the Engine. I will also show an example on how to use the api using java as axample for the supported languages.

## 3 What exactly is the Google App Engine

The Google App Engine - in the following reffered to as App Engine or (G) AE - is a 'Platform as a Service'-service part of the Google Cloud, hosted by Google Inc. It is used to host web applications uploaded by the user. It provides automatic scaling of resources and splits and app across multiple servers. It was released in 2008 in Beta and fully in 2011.

## 4 What does it offer ?

AE's second generation runtime api prominently supports for Python 3 and Java 11 (and other Languages based on the Java Virtual Machine ,such as Kotlin ). Popular scripting languages, that are supported, are Node.js, PHP, aswell as Ruby and Go. Those supports are generally up-to-date with the version of the language itself. Furthermore it includes support for Microsoft's DotNet Controllor View Model applications and support for the c# language. Support for relational databanks using sql and mysql is embedded since 2011. Limited support for older types within the first generation runtime are Python 2(.7), Java 8, GO 1.9 and PHP 5.5. . As mentioned in the section above, Google provides automatic resource scaling. Google App Engine requires a Google account to get started, and an account may allow the developer to register up to 25 free applications and an unlimited number of paid applications. It also includes APIs using Google Accounts for automatically sending emails and authentication.

–Google App Engine defines usage quotas for free applications. Extensions to these quotas can be requested, and application authors can pay for additional resources.[47]

**Dedicated Memcache** As the name suggests, the memcache is basically a memory cache, but with some helpful features, dedicated to the users specific web application. What it does is save the returns for requests that occur especially often. Google offers up to 100 GB of dedicated memcache (only on the us-central server). As long as not fully used the memcache will save any requests. The free version does not offer dedicated in this regard, as it excludes control over the size of the cache, which can make the app, depending on what it runs , exponentially slower. (A Chess engine like Stockfish for an example..) You can not hope for anything, which is disappointing. Other mankos may be the limitation of key length: 250 Byte, though it is rather big and should not cause problems in most cases. Every other given key will be hashed. For efficient use we recommend the keys rather short and in relation to the dedicated cache not that big though.

**BigQuery** "Big Query is a scalable, interactive ad hoc query system for analysis of read-only nested data." -Wikipedia Big Query is the service offered to analalyze used Data. According to Hackernoon.com <sup>1</sup> "BigQuery is append-only and does not have support for primary keys", which leads to duplicate, lthough this can be addressed with workarounds in the apps. Google also has a record of bad communication and reporting including the status page on <https://status.cloud.google.com/> not updating and showing wrong information. <sup>2</sup>

## 5 The Pay Model

The Pay Model is a simple Pay-as-you-go model, which makes it really attractive for dynamic environments. There is also the limit of 1 Gigabyte memory and Traffic. You can use AE for free until you have reached it. Some functions have a different limit though; "to guarantee to stability of the system", according to Google. The model also occupies a choosable resource contingent for day and minute. You can combine this with an optional cost contingent. Whenever one these is reached, the applications will be shut down until the reached contingent is renewed. The guarantees both User safety and a fair system, and saves unnecessary counting. Application instance pricing: For both runtime apis there are nine instance classes of resources available to you. The charge is accordingly to the instance class chosen. They determine (1st) how much memory and processor speed are available for your programm and (2nd) how to handle the scaling of the programm: either automatically or manually. The Handling has

---

<sup>1</sup><https://hackernoon.com/going-gae-our-experience-with-google-app-engine-deaf2b7171c1>

<sup>2</sup><http://latex.hpfsc.de/>

no influence on the pricing though. The B8 class is only comes with manual handling. Storage: App Engine offers one Gigabyte for free a month. Writing, deleting and changing entries differ in pricing: Everything extending 1 Gygabyte will be charged at these prices \*includes pic\* [1] Else \*includes pic\*

## 6 Restrictions

Developers have read-only access to the filesystem on App Engine. Applications can use only virtual filesystems, like gae-filestore.[29] App Engine can only execute code called from an HTTP request (scheduled background tasks allow for self calling HTTP requests). Users may upload arbitrary Python modules, but only if they are pure-Python; C and Pyrex modules are not supported. Java applications may only use a subset (The JRE Class White List) of the classes from the JRE standard edition.[30] This restriction does not exist with the App Engine Standard Java8 runtime. A process started on the server to answer a request can't last more than 60 seconds (with the 1.4.0 release, this restriction does not apply to background jobs anymore). Does not support sticky sessions (a.k.a. session affinity), only replicated sessions are supported including limitation of the amount of data being serialized and time for session serialization.

## 7 In Hindsight/Conclusion

pros: + ez to use +sexy python libraries +much options +for free options +good options for you budget +hard to calculate yourself if you've got the time to calculate it -complex - maybe a bit too much options portability concerns + fear being locked to google (open source projects to fix that ) -limited to google's api no c or c++ support overall: very good good

### 7.1 popular web services using gae

## 8 sources

<https://www.icsr.agh.edu.pl/malawski/google-appengine-ieee-2011.pdf>

<https://cloud.google.com/appengine/docs/standard/?hl=de> <http://googlecode.blogspot.com/2011/10/google-cloud-sql-your-database-in-cloud.html> <https://hackernoon.com/going-gae-our-experience-with-google-app-engine-deaf2b7171c1> 16:62 23.10.19

INSERT G(APP) ZU GAP JOKE WEIL BEHIND