

Contents

1	Introduction	2
1.1	GAE Application Environment	2
1.2	Services Provided	2
1.2.1	The Data Store	2
1.2.2	Google Accounts	3
1.2.3	URL Fetch	3
1.2.4	Mail	3
1.2.5	Memcache	4
1.2.6	Image Manipulation	4
1.2.7	Scheduled Tasks and Task Queues	4
2	Environment	4
2.1	Java Runtime Environment	4
2.2	Python Runtime Environment	5
3	Advantages	5
4	Disadvantages	5
5	Working of Google App Engine	6
5.1	Steps	6
5.2	Deploying The Application	7
6	GAE Account Review	7
6.1	Dashboard	7
6.2	Quotas	8
6.3	Cron Jobs	8
6.4	Task Queue	8
6.5	Billing and Budgeting Resources	8
6.6	Datastore Index	9
6.7	Datastore Statistics	9
7	GAE Program Policies	9
7.1	Prohibited Content	9
7.2	Prohibited Actions	10
8	Why App Engine Only	11
9	Conclusion	11

List of Figures

1	Services Provided by GAE	3
2	GAE Administration Console	8

1 Introduction

Google App Engine lets you run your web applications on Google's infrastructure. App Engine applications are easy to build, easy to maintain, and easy to scale as your traffic and data storage needs grow. With App Engine, there are no servers to maintain: You just upload your application, and it's ready to serve your users.

You can serve your app from your own domain name (such as `http://www.example.com/`) using Google Apps or you can serve your app using a free name on the `appspot.com` domain. You can share your application with the world, or limit access to members of your organization.

Google App Engine supports apps written in several programming languages. With App Engine's Java runtime environment, you can build your app using standard Java technologies, including the JVM, Java servlets, and the Java programming language—or any other language using a JVM-based interpreter or compiler, such as JavaScript or Ruby. App Engine also features a dedicated Python runtime environment, which includes a fast Python interpreter and the Python standard library. The Java and Python runtime environments are built to ensure that your application runs quickly, securely, and without interference from other apps on the system.

With App Engine, you only pay for what you use. There are no set-up costs and no recurring fees. The resources your application uses, such as storage and bandwidth, are measured by the gigabyte, and billed at competitive rates. You control the maximum amounts of resources your app can consume, so it always stays within your budget.

App Engine costs nothing to get started. All applications can use up to 500 MB of storage and enough CPU and bandwidth to support an efficient app serving around 5 million page views a month, absolutely free. When you enable billing for your application, your free limits are raised, and you only pay for the resources you use above the free levels.

1.1 GAE Application Environment

Google App Engine makes it easy to build an application that runs reliably, even under heavy load and with large amounts of data. App Engine includes the following features:

- persistent storage with queries, sorting and transactions
- automatic scaling and load balancing
- APIs for authenticating users and sending email using Google Accounts
- task queues for performing work outside of the scope of a web request
- scheduled tasks for triggering events at specified times and regular intervals
- dynamic web serving, with full support for common web technologies

1.2 Services Provided

1.2.1 The Data Store

App Engine provides a powerful distributed data storage service that features a query engine and transactions. Just as the distributed web server grows with your traffic, the distributed data store grows with your data.

The App Engine data store is not like a traditional relational database. Data objects, or "entities," have a kind and a set of properties. Queries can retrieve entities of a given kind filtered and sorted by the values of the properties. Property values can be of any of the supported property value types.

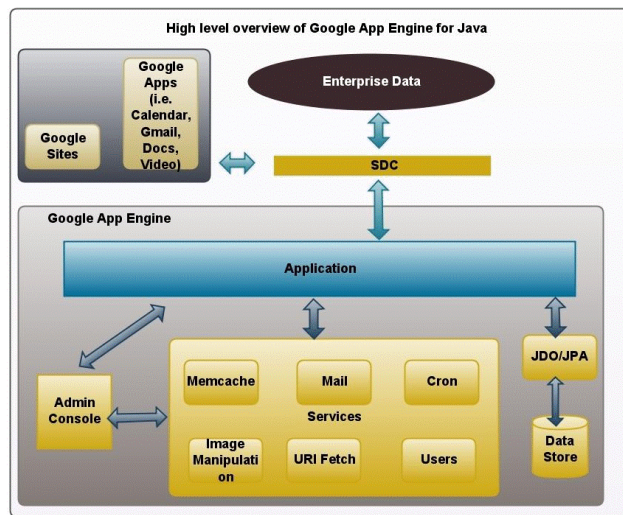


Figure 1: Services Provided by GAE

Datastore entities are schemaless. The structure of data entities is provided by and enforced by your application code. The Java JDO/JPA interfaces and the Python data store interface includes features for applying and enforcing structure within your app. Your app can also access the data store directly to apply as much or as little structure as it needs.

The data store is strongly consistent and uses optimistic concurrency control. An update of an entity occurs in a transaction that is retried a fixed number of times if other processes are trying to update the same entity simultaneously. Your application can execute multiple data store operations in a single transaction which either all succeed or all fail, ensuring the integrity of your data.

The data store implements transactions across its distributed network using "entity groups." A transaction manipulates entities within a single group. Entities of the same group are stored together for efficient execution of transactions. Your application can assign entities to groups when the entities are created.

1.2.2 Google Accounts

App Engine supports integrating an app with Google Accounts for user authentication. Your application can allow a user to sign in with a Google account, and access the email address and displayable name associated with the account. Using Google Accounts lets the user start using your application faster, because the user may not need to create a new account. It also saves you the effort of implementing a user account system just for your application.

If your application is running under Google Apps, it can use the same features with members of your organization and Google Apps accounts.

The Users API can also tell the application whether the current user is a registered administrator for the application. This makes it easy to implement admin-only areas of your site.

1.2.3 URL Fetch

Applications can access resources on the Internet, such as web services or other data, using App Engine's URL fetch service. The URL fetch service retrieves web resources using the same high-speed Google infrastructure that retrieves web pages for many other Google products.

1.2.4 Mail

Applications can send email messages using App Engine's mail service. The mail service uses Google infrastructure to send email messages.

1.2.5 Memcache

The Memcache service provides your application with a high performance in-memory key-value cache that is accessible by multiple instances of your application. Memcache is useful for data that does not need the persistence and transactional features of the data store, such as temporary data or data copied from the data store to the cache for high speed access.

1.2.6 Image Manipulation

The Image service lets your application manipulate images. With this API, you can resize, crop, rotate and flip images in JPEG and PNG formats.

1.2.7 Scheduled Tasks and Task Queues

An application can perform tasks outside of responding to web requests. Your application can perform these tasks on a schedule that you configure, such as on a daily or hourly basis. Or, the application can perform tasks added to a queue by the application itself, such as a background task created while handling a request. Scheduled tasks are also known as "cron jobs," handled by the Cron service.

Task queues are currently released as an experimental feature. At this time, only the Python runtime environment can use task queues. A task queue interface for Java applications will be released in the near future.

2 Environment

Your application can run in one of two runtime environments: the Java environment, and the Python environment. Each environment provides standard protocols and common technologies for web application development.

2.1 Java Runtime Environment

You can develop your application for the Java runtime environment using common Java web development tools and API standards. Your app interacts with the environment using the Java Servlets standard, and can use common web application technologies such as Java Server Pages.

The Java runtime environment uses Java 6. The App Engine Java SDK supports developing apps using either Java 5 or 6.

The environment includes the Java SE Runtime Environment (JRE) 6 platform and libraries. The restrictions of the sandbox environment are implemented in the JVM. An app can use any JVM byte code or library feature, as long as it does not exceed the sandbox restrictions. For instance, byte code that attempts to open a socket or write to a file will throw a runtime exception.

Your app accesses most App Engine services using Java standard APIs. For the App Engine data store, the Java SDK includes implementations of the Java Data Objects (JDO) and Java Persistence API (JPA) interfaces. Your app can use the JavaMail API to send email messages with the App Engine Mail service. The `java.net` HTTP APIs accesses the App Engine URL fetch service. App Engine also includes low-level APIs for its services to implement additional adapters, or to use directly from the application. See the documentation for the data store, memcache, URL fetch, mail, images and Google Accounts APIs.

Typically, Java developers use the Java programming language and APIs to implement web applications for the JVM. With the use of JVM-compatible compilers or interpreters, you can also use other languages to develop web applications, such as JavaScript, Ruby.

2.2 Python Runtime Environment

With App Engine's Python runtime environment, you can implement your app using the Python programming language, and run it on an optimized Python interpreter. App Engine includes rich APIs and tools for Python web application development, including a feature rich data modeling API, an easy-to-use web application framework, and tools for managing and accessing your app's data. You can also take advantage of a wide variety of mature libraries and frameworks for Python web application development, such as Django.

The Python runtime environment uses Python version 2.5.2. Additional support for Python 3 is being considered for a future release.

The Python environment includes the Python standard library. Of course, not all of the library's features can run in the sandbox environment. For instance, a call to a method that attempts to open a socket or write to a file will raise an exception. For convenience, several modules in the standard library whose core features are not supported by the runtime environment have been disabled, and code that imports them will raise an error.

Application code written for the Python environment must be written exclusively in Python. Extensions written in the C language are not supported.

The Python environment provides rich Python APIs for the data store, Google Accounts, URL fetch and email services. App Engine also provides a simple Python web application framework called webapp to make it easy to start building applications.

You can upload other third-party libraries with your application, as long as they are implemented in pure Python and do not require any unsupported standard library modules.

3 Advantages

Google App Engine enables you to build web applications on the same scalable systems that power Google applications. App Engine applications are easy to build, easy to maintain, and easy to scale as your traffic and data storage needs grow. With App Engine, there are no servers to maintain: You just upload your application, and it's ready to serve to your users.

Find out why App Engine may be right for your business.

- Easy to get Started
- Automatic scalability
- The reliability, performance and security of Google's infrastructure
- Cost efficient hosting and risk free trial period

4 Disadvantages

Does Google give any service guarantees of any kind? Google is prone to occasionally change its algorithms and mechanisms in a very opaque and downright secretive manner. Everyone in the field of search engine optimization will know the story. Google tweaks their page ranking algorithms and suddenly your site appears below the fold or (gasp!) not even on the first page any more. Your business literally might evaporate in an instance. What if something similar happens with App Engine? What if they change the service levels your site receives? What if they suddenly decide that your site actually doesn't need that great request latency you have been enjoying and they change their scaling methods in a way that suddenly impacts your user's experience negatively?

Also, do you know what Google does with your data? Do you want Google to know? What if you have a good idea that competes with some of Google's many activities? Do you want to be

dependent on their infrastructure - and worse - their APIs? Changing hosts is always painful. But with something like Amazon's VM hosting service (EC2), you at least know that you can deploy your app as it is on another VM hosting environment (as long as you didn't start to rely on Amazon's S3, at least). But once your APIs are bound to Google's, your cost of switching becomes much higher, since you will need to make change to your source code.

5 Working of Google App Engine

Creating an App Engine application is easy, and only takes a few minutes. And it's free to start: upload your app and share it with users right away, at no charge and with no commitment required.

Google App Engine applications can be written in either the Java or Python programming languages.

The Steps for how to create an application and deploy on app engine is shown below:

5.1 Steps

1. Eclipse is the tool for creating the applications.
 - (a) Eclipse comes with 3 versions, Latest is the Eclipse 3.5 Galileo.
 - (b) We can use even net beans and RAD for developing an application.
2. Google App Engine supports Java 5 and Java 6. When your Java application is running on App Engine, it runs using the Java 6 virtual machine (JVM) and standard libraries.
3. With the Google Plugins for Eclipse, it's easy to develop your Java App Engine application, just as you can to develop any other servlet-based web application. The plug-in lets you create, test and upload App Engine applications from within Eclipse.
4. Google Plugins are available in Google sites just download the Plugins and start creating project.
 - (a) Google Plug-in comes with the following bundle:
 - Google Plug-in for eclipse
 - Google SDK
 - GWT-Google Web Tool Kit
5. Now the next step is to develop an application, now create project using Google Web Application project. Application contains compiled classes, JARs, static files and configuration files are arranged in a directory structure using the WAR standard layout for Java web applications. You can use any development process you like to develop web servlets and produce a WAR directory.
6. We can create JSP, servlet class and even html static files.
7. When the web server receives a request, it determines which servlet class to call using a configuration file known as the "web application deployment descriptor." This file is named web.xml. When servlet is created alter the web.xml.
8. App Engine needs one additional configuration file to figure out how to deploy and run the application. . It includes the registered ID of your application (Eclipse creates this with an empty ID for you to fill in later), the version number of your application, and lists of files that ought to be treated as static files (such as images and CSS) and resource files (such as JSPs and other application data).

9. The App Engine SDK includes a web server application you can use to test your application. You can test your applications in two ways; one is by running the application locally and another by deploying the application into Google server.
10. In case if you found any errors, debug console will help you to identify the errors.

5.2 Deploying The Application

Our developed application can be hosted on Google's infrastructure to access all over the internet. The following steps are to be followed:

STEPS:

1. For Google App engine you should have Google account. To setup your app engine account visit <http://appengine.google.com>, this is the developer site of Google. Google makes a basic level of App engine service available to anyone at no charge.
2. To create your first account, you will be required to verify your account by mobile phone. You will receive sms message with verification code. Enter the code during the application process. After verifying your account, you will be allowed to create application to be hosted on Google infrastructure.
3. After Login home page is displayed, each application is given a hostname on appspot.com which corresponds to the application field in web.xml file. This is called Application identifier, now currently accounts are limited to 10 applications.
4. To create a new application click on Create an Application button.
5. In this, you need to specify Application Identifier and title.
 - Application Identifier: Which must be unique and not already in use.
 - Application Title: This will be displayed in places like the login screen.
 - (Optional) Authentication Mode: if your application uses the user API, it will decide whether your application is available to the public or restricted to specific Google Apps domain.
6. After this step, now it is time to deploy your application on to the Google server which you have created in eclipse environment.
7. In eclipse, click on Deploy App Engine project, provide email and password.
8. In Deployment window, click on App Engine project setting, provide the Application ID and click on Deploy button. Now your application is ready.
9. You can access your application using **ApplicationID.appspot.com**.

6 GAE Account Review

6.1 Dashboard

The System Status Dashboard makes it easier for developers to evaluate and monitor the entire App Engine system by enabling them to measure historical uptime, error rates and latency for each of the major App Engine components.

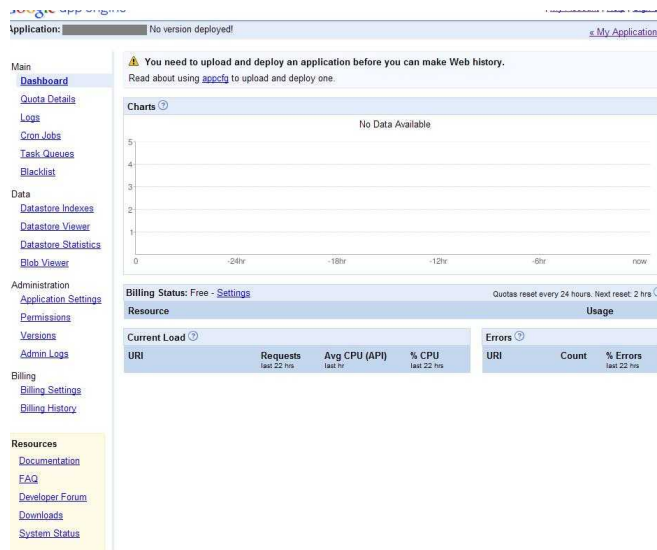


Figure 2: GAE Administration Console

6.2 Quotas

An App Engine application can consume resources up to certain maximums, or quotas. With quotas, App Engine ensures that your application won't exceed your budget, and that other applications running on App Engine won't impact the performance of your app.

6.3 Cron Jobs

The App Engine Cron Service allows you to configure regularly scheduled tasks that operate at defined times or regular intervals. These tasks are commonly known as cron jobs. These cron jobs are automatically triggered by the App Engine Cron Service. For instance, you might use this to send out a report email on a daily basis, to update some cached data every 10 minutes, or to update some summary information once an hour. A cron job will invoke a URL at a given time of day. A URL invoked by cron is subject to the same limits and quotas as a normal HTTP request, including the request time limit.

6.4 Task Queue

App Engine applications can perform background processing by inserting tasks (modeled as web hooks) into a queue. App Engine will detect the presence of new, ready-to-execute tasks and automatically dispatch them for execution, subject to scheduling criteria.

6.5 Billing and Budgeting Resources

Each App Engine application can consume a certain level of computing resources for free, controlled by a set of quotas. Developers who want to grow their applications beyond these free quotas can do so by enabling billing for their application and using Google Checkout to set a daily resource budget, which will allow for the purchasing of additional resources if and when they are needed. App Engine will always be free to get started, and after you've enabled billing for your app all usage up to the free quotas will remain free.

6.6 Datastore Index

The App Engine datastore uses indexes for every query your application makes. These indexes are updated whenever an entity changes, so the results can be returned quickly when the app makes a query. To do this, the datastore needs to know in advance which queries the application will make. You specify which indexes your app needs in a configuration file. The development server can generate the datastore index configuration automatically as you test your app.

6.7 Datastore Statistics

The datastore maintains statistics about the data stored for an application, such as how many entities there are of a given kind, or how much space is used by property values of a given type.

7 GAE Program Policies

To uphold the quality and reputation of Google products and services, Google App Engine (the "Service") is subject to program policies. If you are found to be in violation of Google policies at any time, as determined by Google in its sole discretion, Google may warn you or suspend or terminate your account.

7.1 Prohibited Content

The Content displayed and/or processed through your Application or other web site utilizing the Service shall not contain any of the following types of content:

1. Content that infringes a third party's rights (e.g., copyright) according to applicable law;
2. Pornographic, obscene or excessively profane content;
3. Hate-related or violent content;
4. Content advocating racial or ethnic intolerance;
5. Content intended to advocate or advance computer hacking or cracking;
6. Gambling;
7. Other illegal activity, including without limitation illegal export of controlled substances or illegal software;
8. Drug paraphernalia;
9. Phishing;
10. Malicious content;
11. Other material, products or services that violate or encourage conduct that would violate any criminal laws, any other applicable laws, or any third -party rights.

7.2 Prohibited Actions

In addition to (and/or as some examples of) the violations described in the Terms, you may not and may not allow any third party, including your end users, to:

1. Generate or facilitate unsolicited commercial email ("spam"). Such activity includes, but is not limited to:
 - (a) sending email in violation of the CAN-SPAM Act or any other applicable anti-spam law;
 - (b) imitating or impersonating another person or his, her or its email address, or creating false accounts for the purpose of sending spam;
 - (c) data mining any web property (including Google) to find email addresses or other user account information;
 - (d) sending unauthorized mail via open, third-party servers;
 - (e) sending emails to users who have requested to be removed from a mailing list;
 - (f) selling, exchanging or distributing to a third party the email addresses of any person without such person's knowing and continued consent to such disclosure; and
 - (g) Sending unsolicited emails to significant numbers of email addresses belonging to individuals and/or entities with whom you have no preexisting relationship.
2. Send, upload, distribute or disseminate or offer to do the same with respect to any unlawful, defamatory, harassing, abusive, fraudulent, infringing, obscene, or otherwise objectionable content;
3. Intentionally distribute viruses, worms, defects, Trojan horses, corrupted files, hoaxes, or any other items of a destructive or deceptive nature;
4. Conduct or forward pyramid schemes and the like;
5. Transmit content that may be harmful to minors;
6. Impersonate another person (via the use of an email address or otherwise) or otherwise misrepresent yourself or the source of any email;
7. Illegally transmit another's intellectual property or other proprietary information without such owner's or licensor's permission;
8. Use the Service to violate the legal rights (such as rights of privacy and publicity) of others;
9. Promote or encourage illegal activity;
10. Interfere with other users' enjoyment of the Service;
11. Sell, trade, resell or otherwise exploit the Service for any unauthorized commercial purpose;
12. Modify, adapt, translate, or reverse engineer any portion of the Service;
13. Remove any copyright, trademark or other proprietary rights notices contained in or on the Service;
14. Reformat or frame any portion of the web pages that are part of the Service's Administration Console;
15. Use the Service in connection with illegal peer-to-peer file sharing;

16. Display any content on the Service (including but not limited to the customizable login page) that contains any pornographic, hate-related or violent content or contain any other material, products or services that violate or encourage conduct that would violate any criminal laws, any other applicable laws, or any third party rights; or
17. Modify the Google logo or any other Google Brand Features.
18. Use the Service, or any interfaces provided with the Service, to access any Google product or service in a manner that violates the terms of service or other terms and conditions for use of such Google product or service.

8 Why App Engine Only

Google App Engine enables you to build web applications on the same scalable systems that power Google applications. App Engine applications are easy to build, easy to maintain, and easy to scale as your traffic and data storage needs grow. With App Engine, there are no servers to maintain: You just upload your application, and it's ready to serve to your users.

- Easy to get started
- Automatic scalability
- The reliability, performance and security of Google's infrastructure
- Cost efficient hosting
- Risk free trial period

9 Conclusion

App Engine is a great "engine" for building highly scalable web applications backed by a world-class infrastructure, but it's our responsibility to use the tools provided as effectively and efficiently as possible. A large part of this is designing your data model to leverage the core strengths of App Engine's underlying datastore.

References

- [1] Charles Severance, Using Google App Engine, 2nd Edition, O'Reilly Media.
- [2] Google App Engine Explained, <https://www.youtube.com/watch?v=QX95po8nxiQ>
- [3] <https://appengine.google.com>
- [4] <http://code.google.com/p/googleappengine/issues/list>