



O'REILLY®

Web Services > Excerpts >

Installing and Running Google App Engine on a Linux System: Appendix D - Using Google App Engine


by [Charles Severance](#)

Marc Alier

Jordi Piguillem Poch

PROD: The names above are the chapter authors (not the epigraph authors); please place them under the title.

This appendix describes the installation of the Google App Engine Software Development Kit (SDK) on Linux, plus how to run a simple "Hello, world" application.



This excerpt is from [Using Google App Engine](#). With this book, you can build exciting, scalable web applications quickly and confidently, using Google App Engine -- even if you have little or no experience in programming or web development. Using Google App Engine provides an overview of the tools necessary to use Google App Engine, including Python, HTML, Cascading Style Sheets (CSS), and HTTP. You'll also learn what's required to deploy your applications to Google servers.

Buy it now

The App Engine SDK allows you to run Google App Engine Applications on your local computer. It simulates the runtime environment of the Google App Engine infrastructure.

Checking for Python

Google App Engine needs the Python interpreter to work on your system. Python comes usually preinstalled on most Linux distributions. To check whether Python is installed, open a Terminal window and type `python` to execute the Python interpreter:

```
python
Python 2.5.2 (r252:60911, Feb 22 2008, 07:57:53)
...
```

The first line will show you the version of Python running on your system. Type `exit()` to return to the command line.

If Python is not installed or you have a version other than 2.5.x, you will need to download this version (2.5) of Python from <http://python.org>. Check to see whether there is a package containing Python 2.5 for your Linux distribution.

Download and Install

Download the Google App Engine SDK from <http://code.google.com/appengine/downloads.html>; download the appropriate install package, as shown in [Figure D.1, "Downloading Google Application Engine"](#).

Figure D.1. Downloading Google Application Engine

Download the Google App Engine SDK

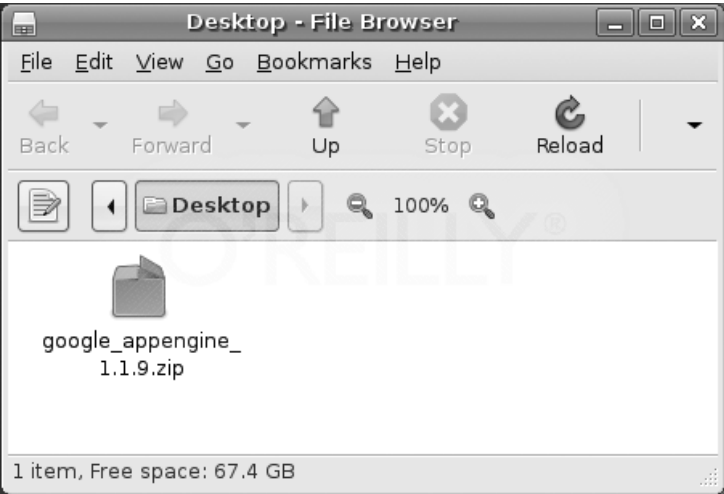
Before downloading, please read the [Terms](#) that govern your use of the App Engine SDK.

Please note: The App Engine SDK is under **active development**, please keep this in mind as you explore its capabilities. See the [SDK Release Notes](#) for the information on the most recent changes to the App Engine SDK. If you discover any issues, please feel free to notify us via our [Issue Tracker](#).

Platform	Version	Package	Size	SHA1 Checksum
Windows	1.1.9 - 02/09/09	GoogleAppEngine 1.1.9.msi	2.7 MB	c0a09f23950de606a4c4933687335582c18bd3cd
Mac OS X	1.1.9 - 02/09/09	GoogleAppEngineLauncher-1.1.9.dmg	3.8 MB	d8ca6afa091fa1e48e2132454a71bf7ecb1d627
Linux/Other Platforms	1.1.9 - 02/09/09	google_appengine 1.1.9.zip	2.8 MB	aa11e2e6940071cf61c3092b2365605a01d739e7

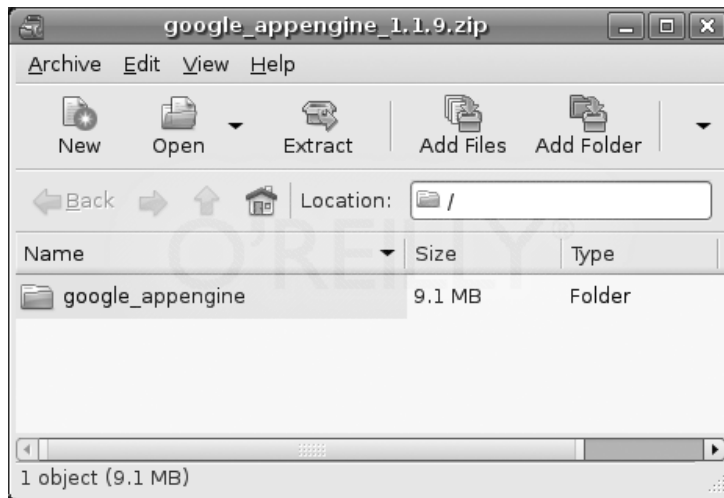
In this case, you must download the Linux package onto your computer's desktop, as shown in [Figure D.2, "The App Engine package"](#).

Figure D.2. The App Engine package



Decompress the downloaded file. In [Figure D.3, "Unzipping the file using File Roller"](#), we are using File Roller to unzip the file.

Figure D.3. Unzipping the file using File Roller



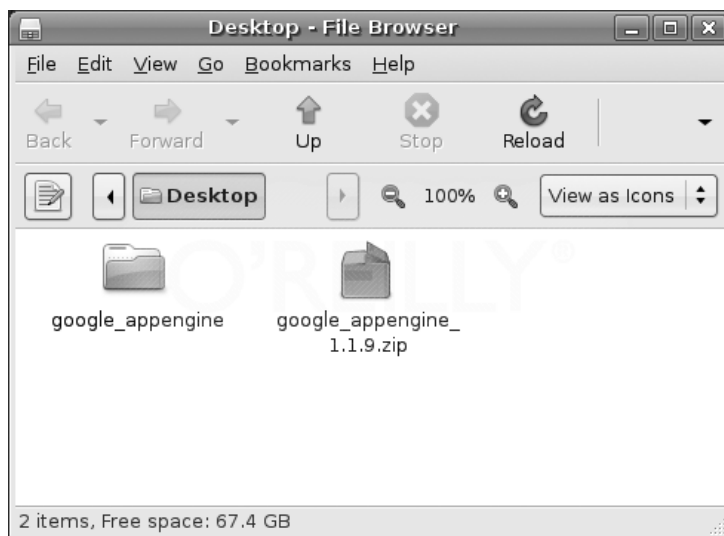
[Figure D.4, "Unzipping the file using Terminal"](#) shows how you can also use the built-in `unzip` command-line tool in the Terminal application. Here we are uncompressing `google_appengine_1.1.9.zip` (the SDK) to our desktop folder.

Figure D.4. Unzipping the file using Terminal



Once you have completed the unzipping process, you will get a folder named `google_appengine` that contains all the files necessary to develop your applications, as shown in [Figure D.5, "Unzipped content"](#).

Figure D.5. Unzipped content



The last step is to include the path to Google App Engine scripts into the system's `PATH` variable to launch its applications from a terminal using the command:

```
export PATH=$PATH:/home/computer_name/Desktop/google_appengine/
```

Substitute your username instead of `computer_name`.

The `export` command will work only during your current session. Check your Linux distribution manual to see which file (such as `.bashrc_login`) you can edit to add this command so that the `PATH` variable is set properly in each new terminal window that is opened.

We will use this window for the remainder of the steps in this process. If you start a new window, type the previous `PATH` command once in each new window until you edit your login script to make the path change automatic.

Making Your First Application

Make a folder for your Google App Engine applications. I am going to put the folder on my desktop and call it `apps`. The path to this

folder is `/home/pigui/Desktop/apps`. Then make a subfolder in within `apps` called `ae-01-trivial`. The path to this folder would be `/home/pigui/Desktop/apps/ae-01-trivial`.

Create a file called `app.yaml` in the `ae-01-trivial` folder with the following contents:

```
application: ae-01-trivial
version: 1
runtime: python
api_version: 1

handlers:
- url: /.
  script: index.py
```

Note

If you are looking at a PDF copy of this book, please do not copy and paste these lines into your text editor. You might end up with strange characters. Just type them into your editor.

Then create a file in the `ae-01-trivial` folder called `index.py`, with three lines of Python:

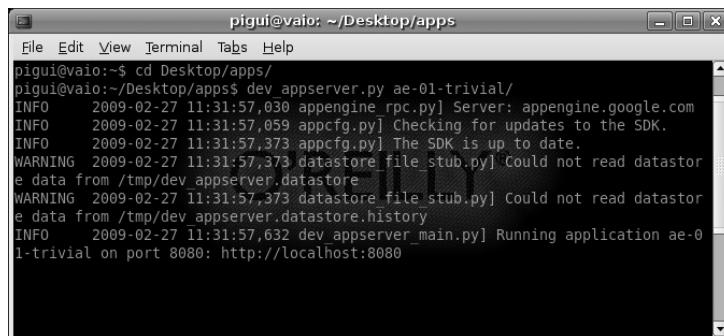
```
print 'Content-Type: text/plain'
print ''
print 'Hello there Chuck'
```

Using the Terminal, navigate to the `apps` directory, start the Google App Engine Web Server, and run your application using the following command:

```
dev_appserver.py ae-01-trivial
```

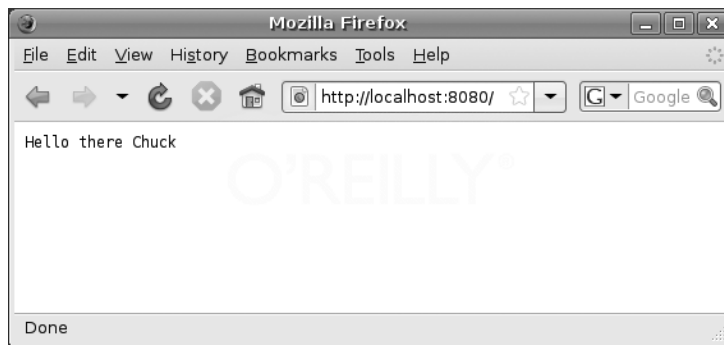
You will be asked if you want App Engine to check for updates (type `y`). After a few messages, the server will start up, as shown in Figure D.6, "Starting the application server".

Figure D.6. Starting the application server



The last line tells you which port your application is running on and what URL you should use to access your application. In this case, our application is at `http://localhost:8080`. Paste `http://localhost:8080` into your browser, and you should see your application, as shown in Figure D.7, "Your Google application".

Figure D.7. Your Google application



Just for fun, edit the `index.py` to change the name "Chuck" to your own name, and refresh the page in your browser to verify your updates.

Dealing with Errors

With two files to edit, there are two general categories of errors that you may encounter. The first common error is making a mistake in your `app.yaml` file. If you make a mistake on the `app.yaml` file, App Engine will not start, and you will see an error as shown in Figure D.8, "Error in app.yaml".

Figure D.8. Error in app.yaml

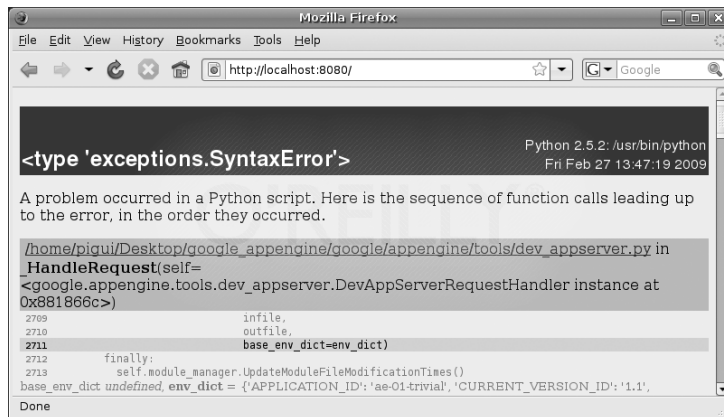


In this instance, the mistake is an incorrect indentation of the final line in `app.yaml` (line 8).

When you make a mistake in the `app.yaml` file, you must fix the mistake and attempt to start the application again.

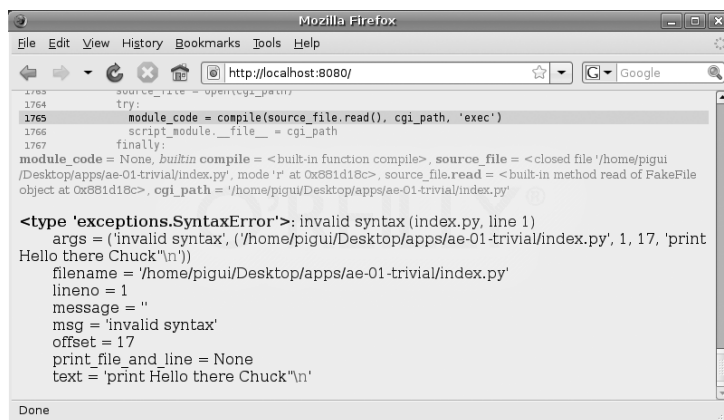
If you make a syntax error in the `index.py` file, the error will appear in your browser. The error looks terrible, like everything went wrong, as shown in [Figure D.9, "Syntax error"](#).

Figure D.9. Syntax error



Do not be alarmed! Ignore most of the output and scroll to the very bottom of the error output. The error that you need to see is likely to be the very last line of the output. In this case, I made a Python syntax error on the first line of our one-line application, as shown in [Figure D.10, "Finding the syntax error"](#).

Figure D.10. Finding the syntax error



See also http://en.wikipedia.org/wiki/Stack_trace for more information.

If you make a mistake in a file like `index.py`, you can fix the file and refresh the page in your browser. There is no need to restart the server.

Shutting Down the Server

To shut down the server, go into the window where you started the server and press Ctrl-C to abort the server. You should see a message that says something like "Server interrupted by user, terminating," and the server will shut down. You can start it back up by using the `dev_appserver.py` command again. When the server is shut down, you will notice that navigating to `http://localhost:8080` will fail because there is no software running on and listening to port 8080.

If you enjoyed this excerpt, buy a copy of [Using Google App Engine](#).

