Installing and Running the Google Application Engine
On a Macintosh System
University of Michigan – Informatics

This document describes the installation of the Google Application Engine Software Development Kit (SDK) on a Macintosh and running a simple "hello world" application.

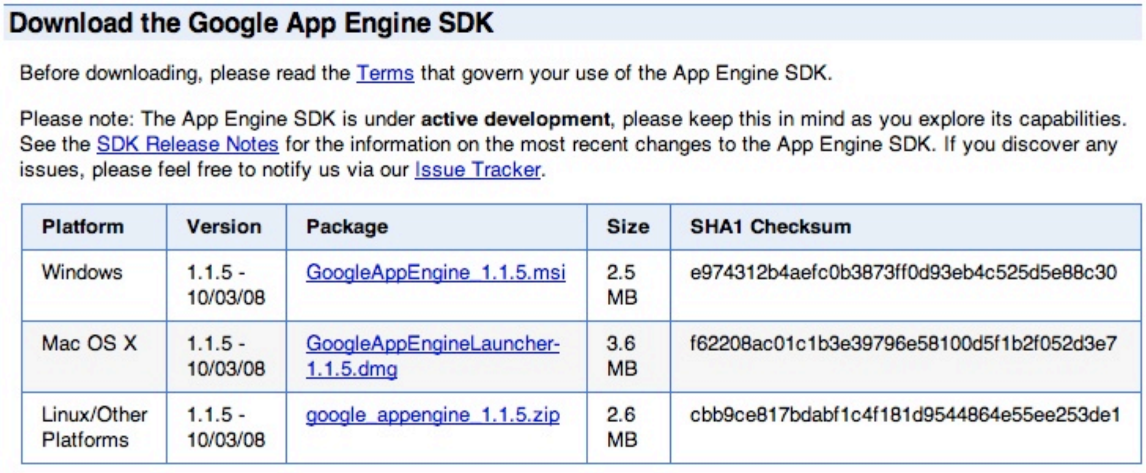
The AppEngine SDK allows you to run Google Application Engine Applications on your local computer. It simulates the run-time environment of the Google App Engine infrastructure.

Download and Install

You can download the Google Application Engine SDK by going to:

```
http://code.google.com/appengine/downloads.html
```

And downloading the appropriate install package.



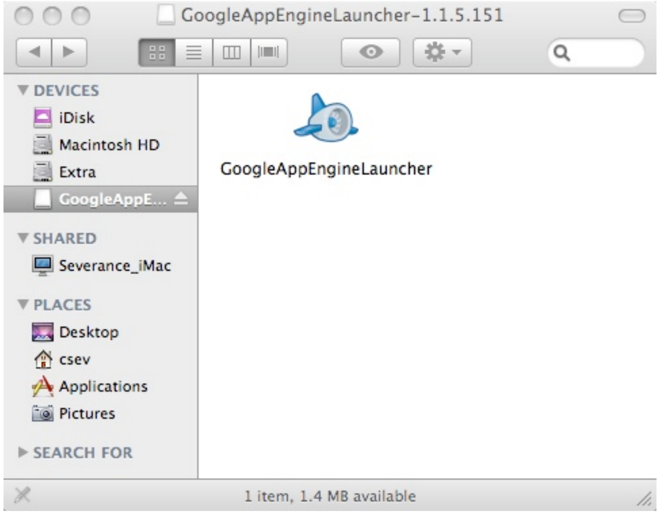
Download the Google App Engine SDK

Before downloading, please read the Terms that govern your use of the App Engine SDK.

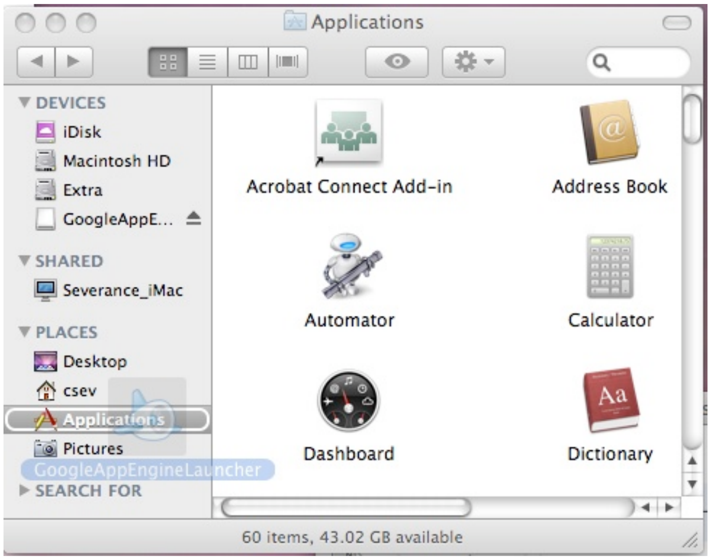
Please note: The App Engine SDK is under **active development**, please keep this in mind as you explore its capabilities. See the SDK Release Notes for the information on the most recent changes to the App Engine SDK. If you discover any issues, please feel free to notify us via our Issue Tracker.

Platform	Version	Package	Size	SHA1 Checksum
Windows	1.1.5 - 10/03/08	GoogleAppEngine_1.1.5.msi	2.5 MB	e974312b4aefc0b3873ff0d93eb4c525d5e88c30
Mac OS X	1.1.5 - 10/03/08	GoogleAppEngineLauncher-1.1.5.dmg	3.6 MB	f62208ac01c1b3e39796e58100d5f1b2f052d3e7
Linux/Other Platforms	1.1.5 - 10/03/08	google_appengine_1.1.5.zip	2.6 MB	cbb9ce817bdabf1c4f181d9544864e55ee253de1

Download the Mac OS X installer – it should automatically mount as a virtual drive.



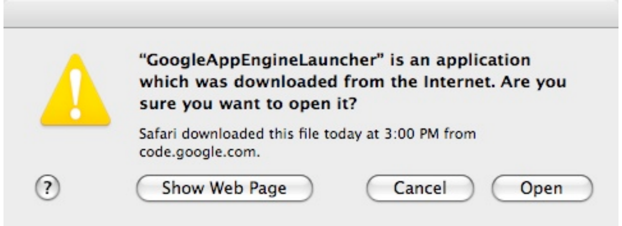
Drag the **GoogleAppEngineLauncher** to the **Applications** folder on your hard drive. This will copy the Google Application Engine and install it as an application on your system.



Once this is done – you can eject the virtual drive by pressing on the Eject button.

Navigate to the **/Applications** folder on your main disk, find the AppEngineLauncher icon and launch it.   You may need to scroll to the bottom of your screen to see the Application Engine icon.

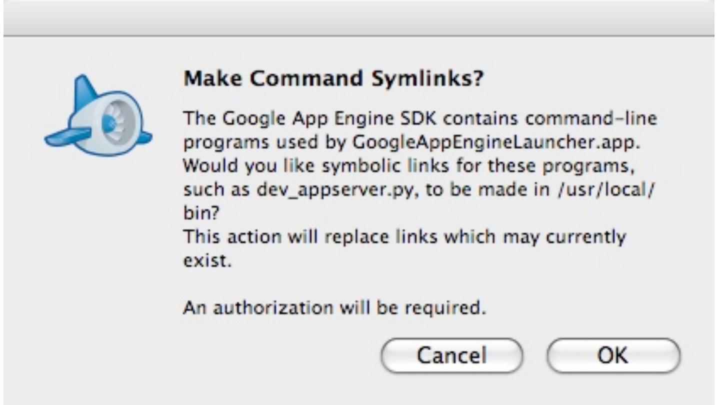
Accept any dialog box that asks if it is “OK to launch”.



"GoogleAppEngineLauncher" is an application which was downloaded from the Internet. Are you sure you want to open it?

Safari downloaded this file today at 3:00 PM from code.google.com.

When the Engine launches for the first time, it asks if you want to make "Command Links":



Make Command Symlinks?

The Google App Engine SDK contains command-line programs used by GoogleAppEngineLauncher.app. Would you like symbolic links for these programs, such as dev_appserver.py, to be made in /usr/local/bin?
This action will replace links which may currently exist.

An authorization will be required.

Press "OK" – this will allow us to run the App Engine from the command line later. You will have to type an administrator password to make the links.

At this point, you can actually close the Application Engine Launcher – we will run the application from the Command Line Interface (Terminal) instead of using the Launcher user interface.

Making your First Application

Now you need to create a simple application. We could use the "+" option to have the launcher make us an application – but instead we will do it by hand to get a better sense of what is going on.

Make a folder for your Google Application Engine applications. I am going to make the Folder on my Macintosh Desktop called "apps" – the path to this folder is:

> **/Users/csev/Desktop/apps**

And then make a sub-folder in within **apps** called "**ae-01-trivial**" – the path to this folder would be:

> **/Users/csev/Desktop/apps/ae-01-trivial**

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

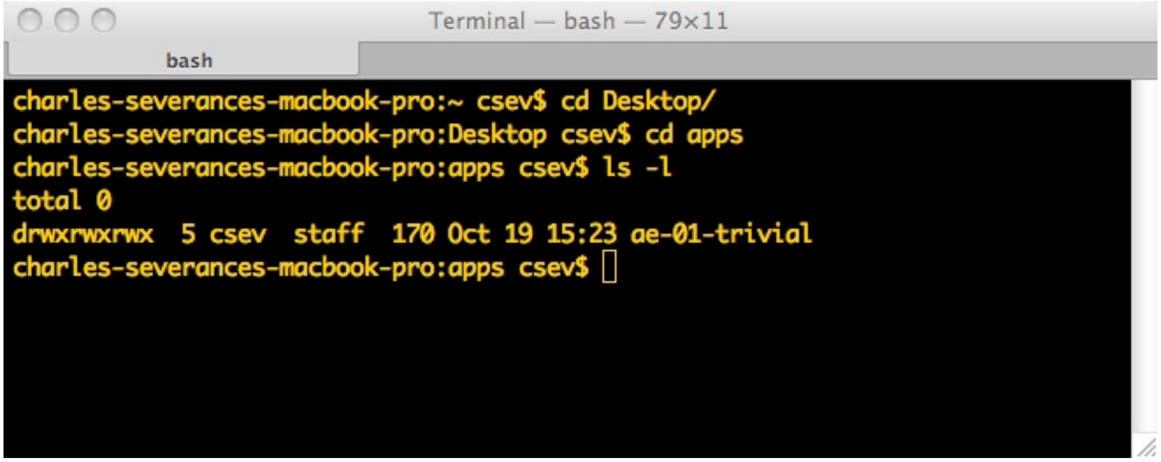
```
application: ae-01-trivial
version: 1
runtime: python
api_version: 1

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

Note: Please do not copy and paste these lines into your text editor – you might end up with strange characters – simply type them into your editor.
Then create a file in the **ae-01-trivial** folder called **index.py** with a single line in it:

```
print "Hello there Chuck"
```

Then start the **Terminal** program that can be found under **Applications -> Utilities -> Terminal**. Use the **cd** command to navigate into the **apps** directory.

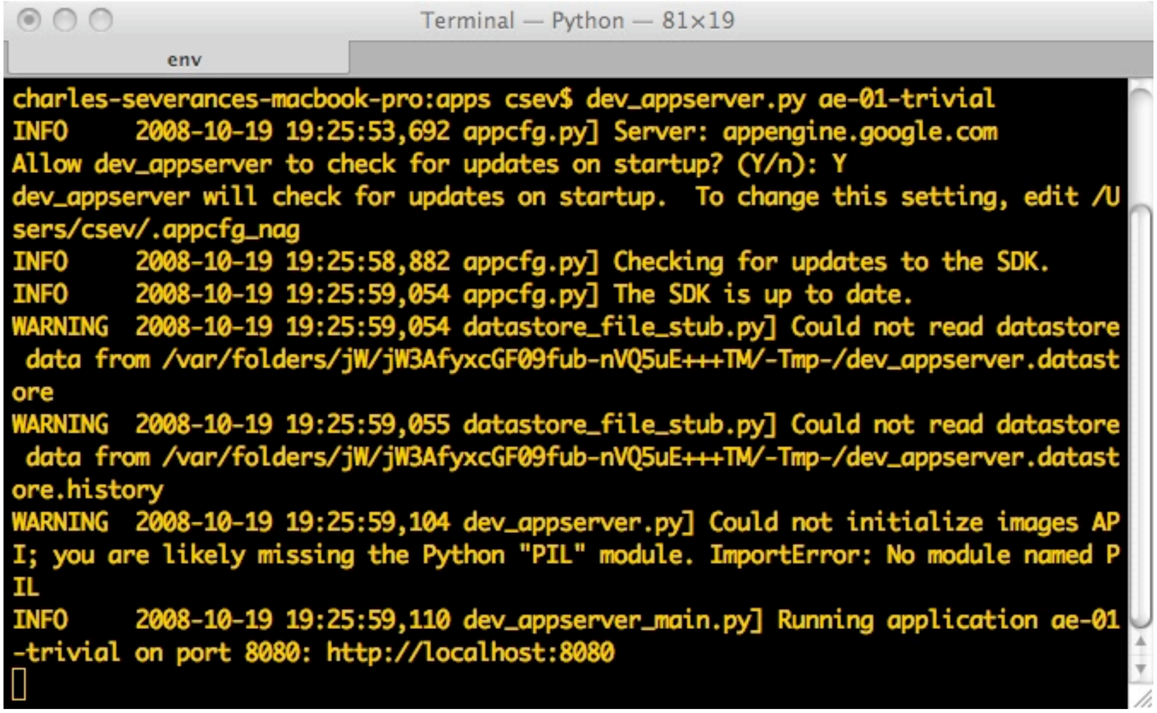


```
charles-severances-macbook-pro:~ csev$ cd Desktop/
charles-severances-macbook-pro:Desktop csev$ cd apps
charles-severances-macbook-pro:apps csev$ ls -l
total 0
drwxrwxrwx  5 csev  staff  170 Oct 19 15:23 ae-01-trivial
charles-severances-macbook-pro:apps csev$ 
```

Then start the Google Application Engine Web Server and run your application using the command:

```
/usr/local/bin/dev_appserver.py ae-01-trivial
```

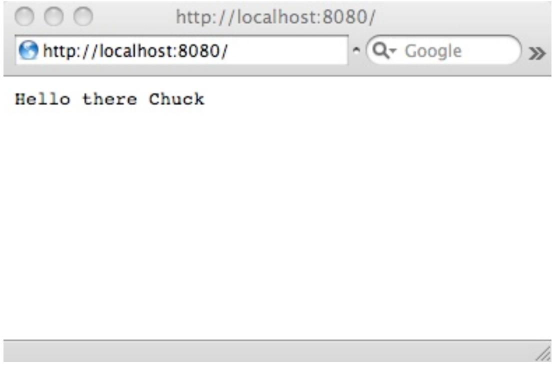
You will be asked if you want the AppEngine to check for updates (Y is OK here) and then after a few messages – the server will start up as shown:



```
charles-severances-macbook-pro:apps csev$ dev_appserver.py ae-01-trivial
INFO     2008-10-19 19:25:53,692 appcfg.py] Server: appengine.google.com
Allow dev_appserver to check for updates on startup? (Y/n): Y
dev_appserver will check for updates on startup.  To change this setting, edit /U
sers/csev/.appcfg_nag
INFO     2008-10-19 19:25:58,882 appcfg.py] Checking for updates to the SDK.
INFO     2008-10-19 19:25:59,054 appcfg.py] The SDK is up to date.
WARNING  2008-10-19 19:25:59,054 datastore_file_stub.py] Could not read datastore
 data from /var/folders/jW/jW3AfyxcGF09fub-nVQ5uE+++TM/-Tmp-/dev_appserver.datast
ore
WARNING  2008-10-19 19:25:59,055 datastore_file_stub.py] Could not read datastore
 data from /var/folders/jW/jW3AfyxcGF09fub-nVQ5uE+++TM/-Tmp-/dev_appserver.datast
ore.history
WARNING  2008-10-19 19:25:59,104 dev_appserver.py] Could not initialize images AP
I; you are likely missing the Python "PIL" module. ImportError: No module named P
IL
INFO     2008-10-19 19:25:59,110 dev_appserver_main.py] Running application ae-01
-trivial on port 8080: http://localhost:8080
```

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 follows:

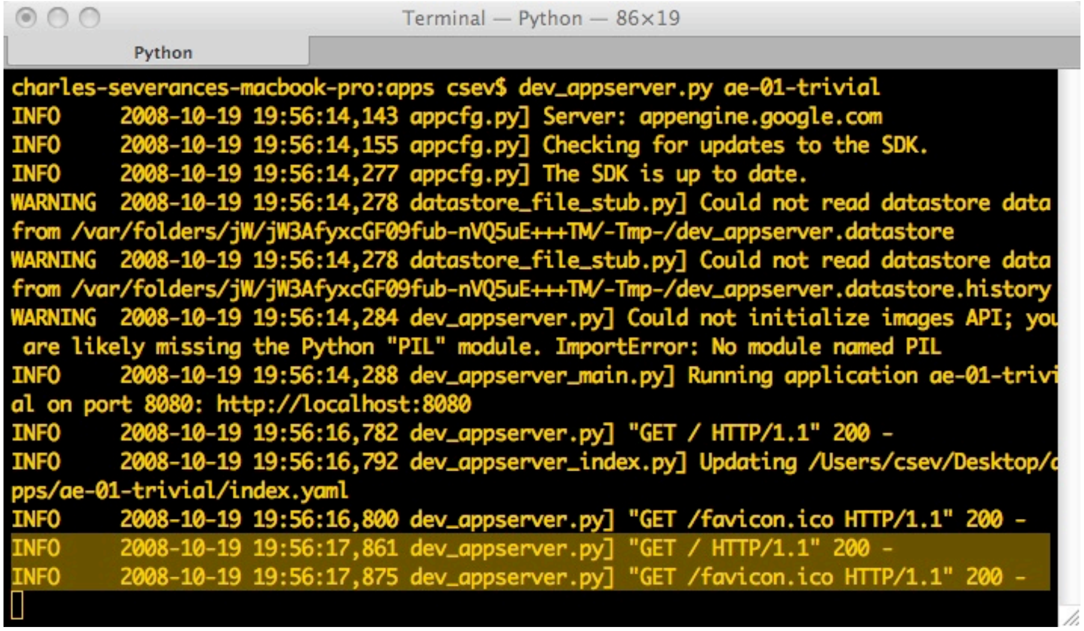


```
Hello there Chuck
```

Just for fun, edit the **index.py** to change the name "Chuck" to your own name and press Refresh in the browser to verify your updates.

Watching the Log

You can watch the internal log of the actions that the web server is performing in the same window as you started the application server:

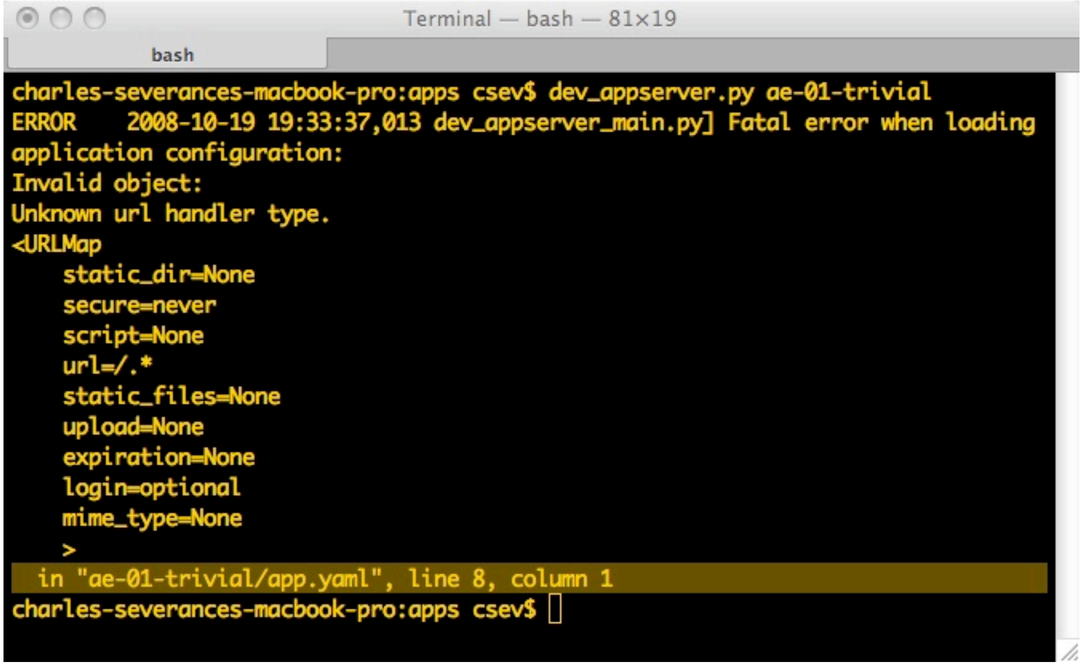


```
charles-severances-macbook-pro:apps csev$ dev_appserver.py ae-01-trivial
INFO     2008-10-19 19:56:14,143 appcfg.py] Server: appengine.google.com
INFO     2008-10-19 19:56:14,155 appcfg.py] Checking for updates to the SDK.
INFO     2008-10-19 19:56:14,277 appcfg.py] The SDK is up to date.
WARNING  2008-10-19 19:56:14,278 datastore_file_stub.py] Could not read datastore data
from /var/folders/jW/jW3AfyxcGF09fub-nVQ5uE+++TM/-Tmp-/dev_appserver.datastore
WARNING  2008-10-19 19:56:14,278 datastore_file_stub.py] Could not read datastore data
from /var/folders/jW/jW3AfyxcGF09fub-nVQ5uE+++TM/-Tmp-/dev_appserver.datastore.history
WARNING  2008-10-19 19:56:14,284 dev_appserver.py] Could not initialize images API; you
 are likely missing the Python "PIL" module. ImportError: No module named PIL
INFO     2008-10-19 19:56:14,288 dev_appserver_main.py] Running application ae-01-trivi
al on port 8080: http://localhost:8080
INFO     2008-10-19 19:56:16,782 dev_appserver.py] "GET / HTTP/1.1" 200 -
INFO     2008-10-19 19:56:16,792 dev_appserver_index.py] Updating /Users/csev/Desktop/a
pps/ae-01-trivial/index.yaml
INFO     2008-10-19 19:56:16,800 dev_appserver.py] "GET /favicon.ico HTTP/1.1" 200 -
INFO     2008-10-19 19:56:17,861 dev_appserver.py] "GET / HTTP/1.1" 200 -
INFO     2008-10-19 19:56:17,875 dev_appserver.py] "GET /favicon.ico HTTP/1.1" 200 -
```

Each time you press **Refresh** – you can see it retrieving the output with a **GET** request. The attempt to retrieve the **favicon.ico** is your browser checking to see if there is an icon that is to be displayed next to the address bar.

Dealing With Errors

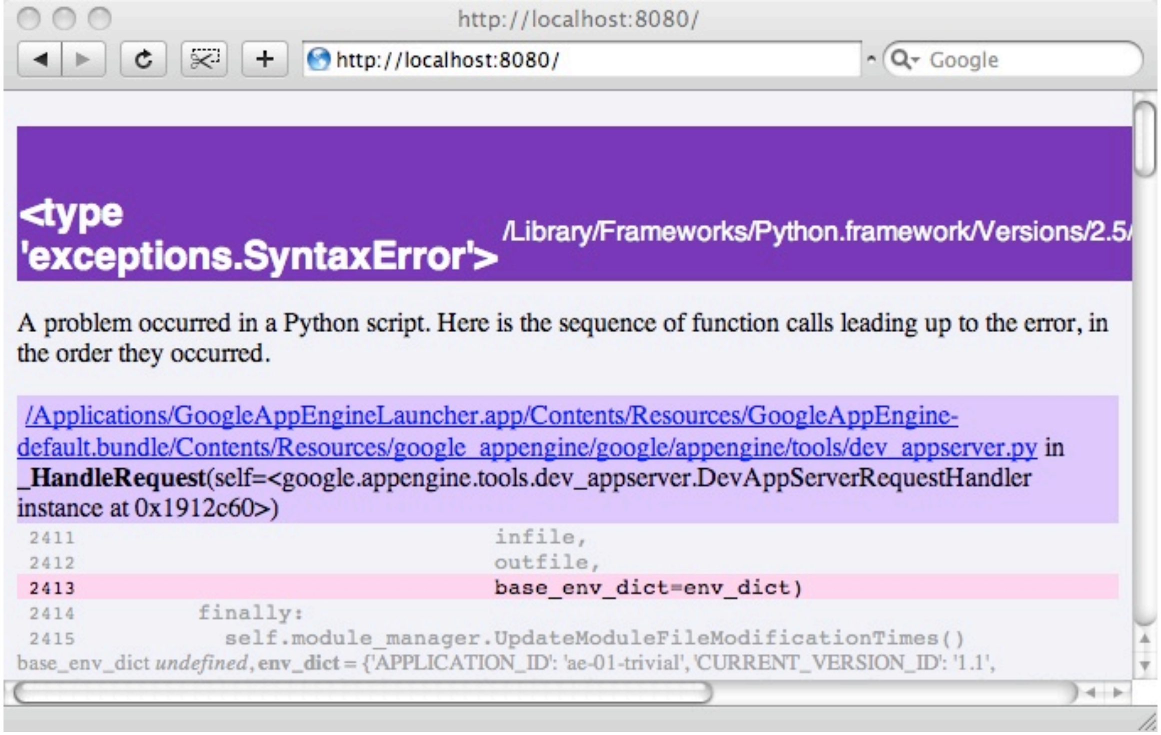
With two files to edit, there are two general categories of errors that you may encounter. If you make a mistake on the **app.yaml** file, the AppEngine will not start and you will see an error like the following in the command line:



```
charles-severances-macbook-pro:apps csev$ dev_appserver.py ae-01-trivial
ERROR    2008-10-19 19:33:37,013 dev_appserver_main.py] Fatal error when loading
application configuration:
Invalid object:
Unknown url handler type.
<URLMap
    static_dir=None
    secure=never
    script=None
    url=/.*
    static_files=None
    upload=None
    expiration=None
    login=optional
    mime_type=None
    >
  in "ae-01-trivial/app.yaml", line 8, column 1
charles-severances-macbook-pro:apps csev$
```

In this instance – the mistake is mis-indenting the last line in the **app.yaml** (line 8).

If you make a syntax error in the **index.py** file, the error will appear in your browser. The error looks terrible and looks like everything went wrong:



<type 'exceptions.SyntaxError'> /Library/Frameworks/Python.framework/Versions/2.5/

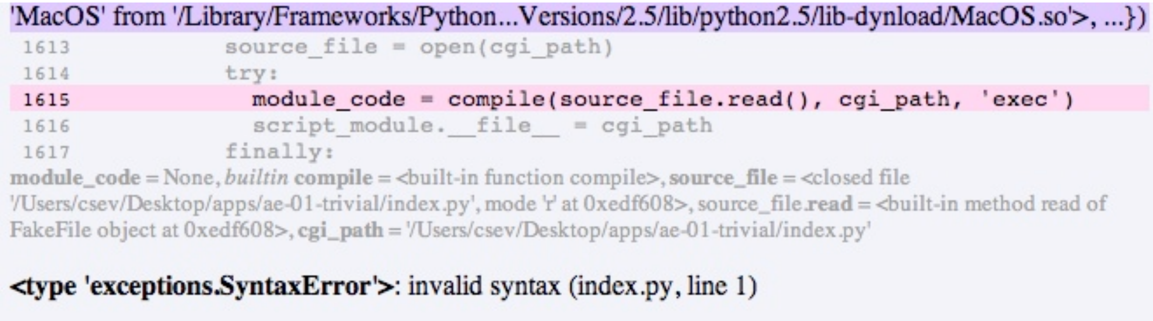
A problem occurred in a Python script. Here is the sequence of function calls leading up to the error, in the order they occurred.

/Applications/GoogleAppEngineLauncher.app/Contents/Resources/GoogleAppEngine-default.bundle/Contents/Resources/google_appengine/google/appengine/tools/dev_appserver.py in **_HandleRequest**(self=<google.appengine.tools.dev_appserver.DevAppServerRequestHandler instance at 0x1912c60>)

```
 2411                                   infile,
 2412                                   outfile,
 2413                                   base_env_dict=env_dict)
 2414               finally:
 2415                 self.module_manager.UpdateModuleFileModificationTimes()
```

base_env_dict *undefined*, **env_dict** = {'APPLICATION_ID': 'ae-01-trivial', 'CURRENT_VERSION_ID': '1.1',

Do not be alarmed – ignore most of the output and scroll to the very bottom of the error output. The error 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 line one of our one-line application.



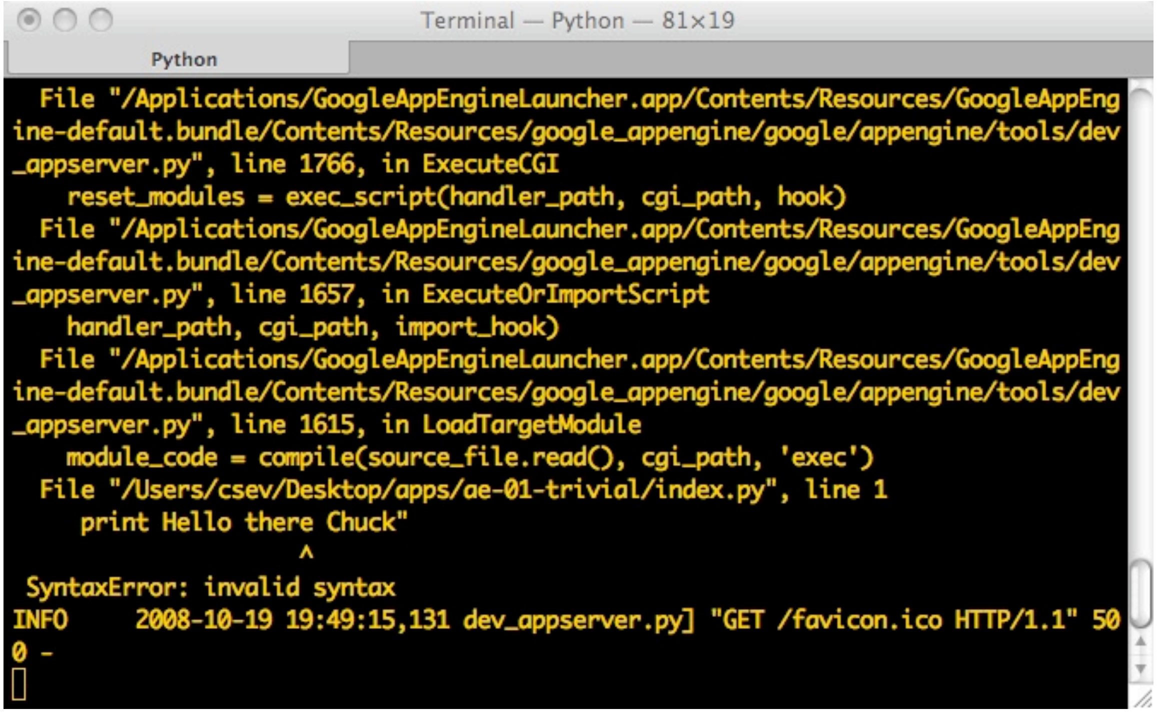
'MacOS' from '/Library/Frameworks/Python...Versions/2.5/lib/python2.5/lib-dynload/MacOS.so'>, ...})

```
 1613               source_file = open(cgi_path)
 1614               try:
 1615                 module_code = compile(source_file.read(), cgi_path, 'exec')
 1616                 script_module.__file__ = cgi_path
 1617               finally:
```

module_code = None, *builtin* **compile** = <built-in function compile>, **source_file** = <closed file '/Users/csev/Desktop/apps/ae-01-trivial/index.py', mode 'r' at 0xedf608>, **source_file.read** = <built-in method read of FakeFile object at 0xedf608>, **cgi_path** = '/Users/csev/Desktop/apps/ae-01-trivial/index.py'

<type 'exceptions.SyntaxError'>: invalid syntax (index.py, line 1)

If you are watching the window where you started the application server, the log will also show the error as follows:



```
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/GoogleAppEng
ine-default.bundle/Contents/Resources/google_appengine/google/appengine/tools/dev
_appserver.py", line 1766, in ExecuteCGI
    reset_modules = exec_script(handler_path, cgi_path, hook)
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/GoogleAppEng
ine-default.bundle/Contents/Resources/google_appengine/google/appengine/tools/dev
_appserver.py", line 1657, in ExecuteOrImportScript
    handler_path, cgi_path, import_hook)
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/GoogleAppEng
ine-default.bundle/Contents/Resources/google_appengine/google/appengine/tools/dev
_appserver.py", line 1615, in LoadTargetModule
    module_code = compile(source_file.read(), cgi_path, 'exec')
  File "/Users/csev/Desktop/apps/ae-01-trivial/index.py", line 1
     print Hello there Chuck"
                     ^
 SyntaxError: invalid syntax
INFO     2008-10-19 19:49:15,131 dev_appserver.py] "GET /favicon.ico HTTP/1.1" 50
0 -
```

Again, there is a lot of seemingly important output. Most of it can be ignored – it is "trace back" telling exactly where within the Google App Engine the error was detected. For you can ignore most of is and find the mistake you made at the very end.

Reference: http://en.wikipedia.org/wiki/Stack_trace

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

If you make a mistake in a file like i**ndex.py**, you can simply fix the file and press refresh in your browser – there is no need to restart the server.

Shutting Down the Server

To shut down the server, simply 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 shutdown, you will notice that navigating to **http://localhost:8080** will fail because there is no software running on and listening to port 8080.

Comments and questions to csev@umich.edu www.dr-chuck.com