Installing and Running the Google App Engine
On a Macintosh System

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

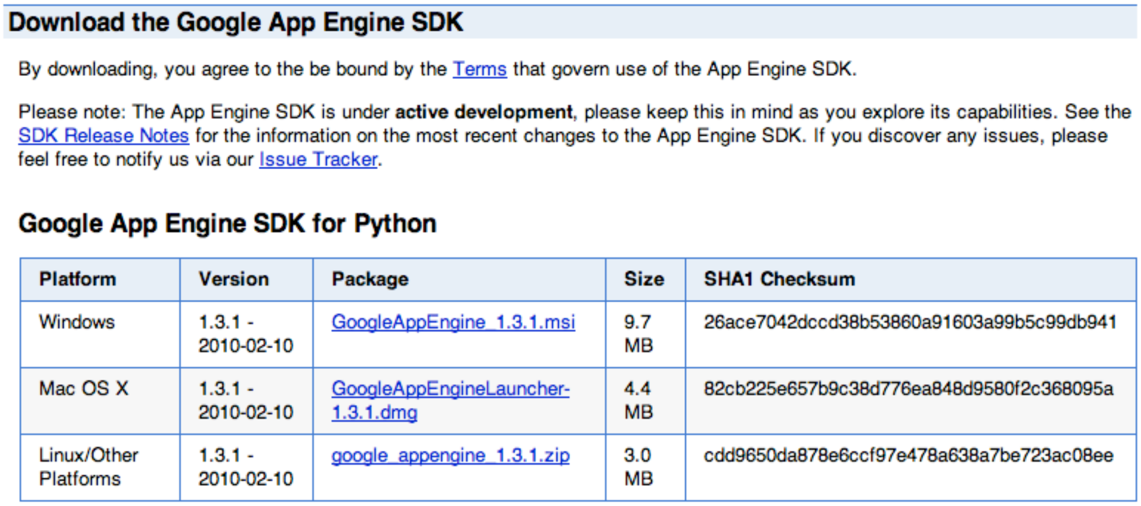
The App Engine SDK allows you to run Google App Engine Applications on your local computer. It simulates the run-time environment of the Google App Engine infrastructure.

Download and Install

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

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

and downloading the appropriate install package.



Download the Google App Engine SDK

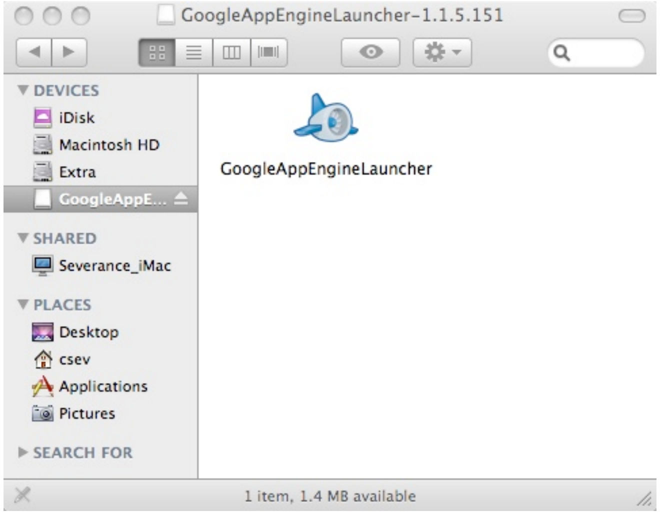
By downloading, you agree to the be bound by the Terms that govern use of the App Engine SDK.

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.

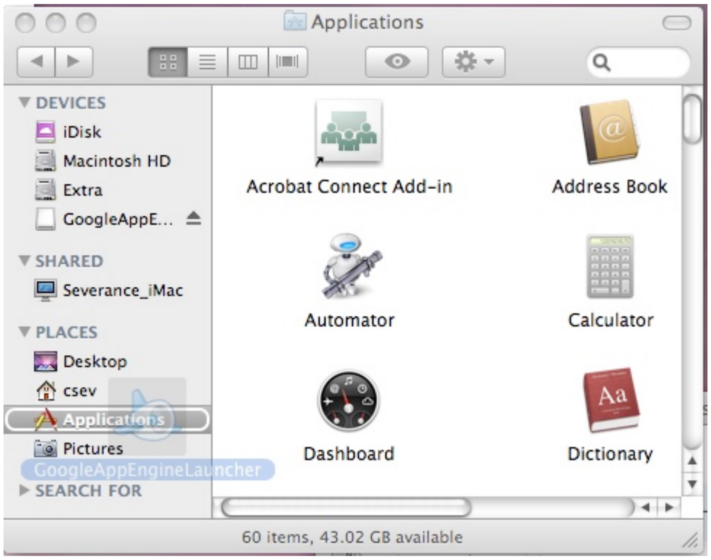
Google App Engine SDK for Python

Platform	Version	Package	Size	SHA1 Checksum
Windows	1.3.1 - 2010-02-10	GoogleAppEngine_1.3.1.msi	9.7 MB	26ace7042dccd38b53860a91603a99b5c99db941
Mac OS X	1.3.1 - 2010-02-10	GoogleAppEngineLauncher-1.3.1.dmg	4.4 MB	82cb225e657b9c38d776ea848d9580f2c368095a
Linux/Other Platforms	1.3.1 - 2010-02-10	google_appengine_1.3.1.zip	3.0 MB	cdd9650da878e6ccf97e478a638a7be723ac08ee

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



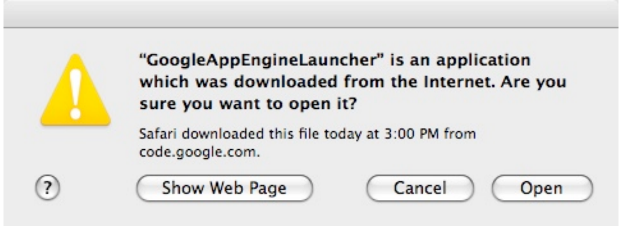
Drag the **GoogleAppEngineLauncher** to the **Applications** folder on your hard drive. This will copy the Google App Engine and install it as an application on your system.



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 App Engine icon.

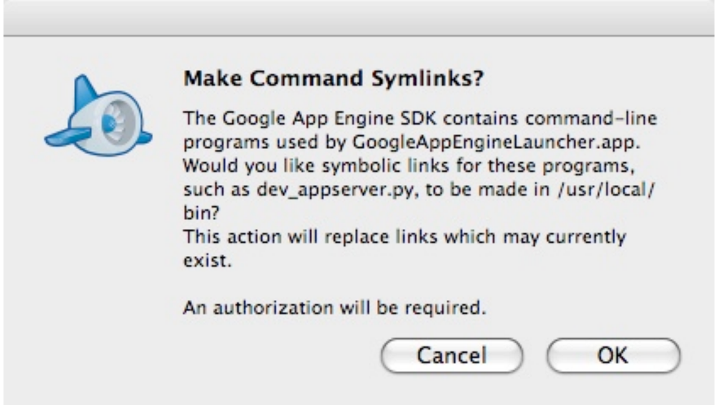
Accept any dialog box that asks if it is "OK to launch".



"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 App 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 App 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: 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 – simply type them into your editor.

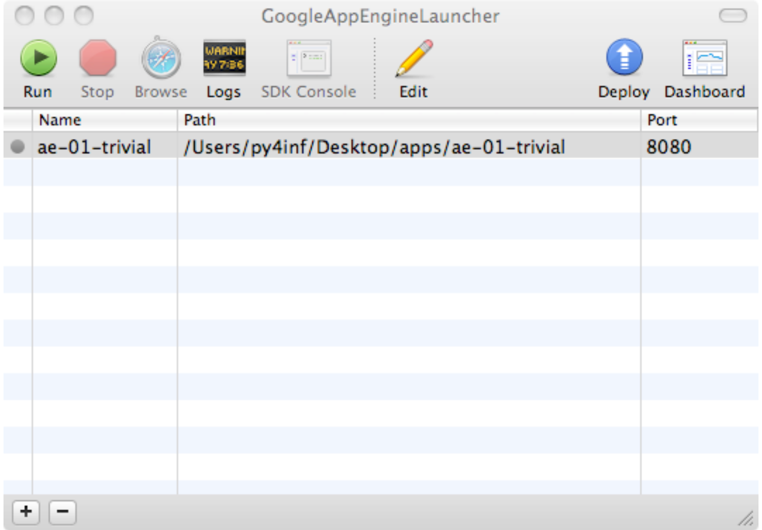
Then create a file in the **ae-01-trivial** folder called **index.py** with three lines in it:

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

Then create a file in the **ae-01-trivial** folder called **index.py** with three lines in it:

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

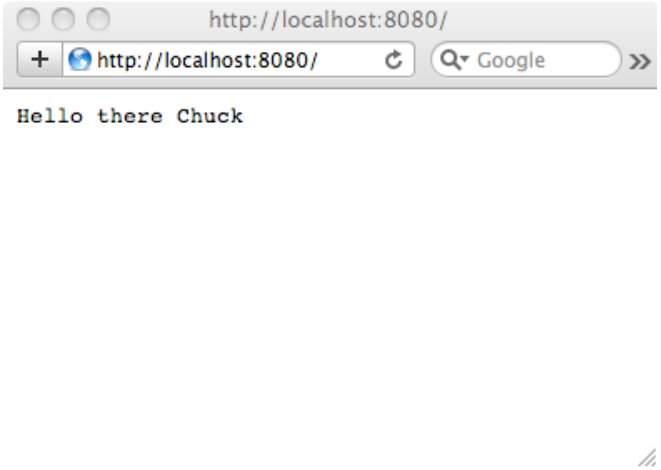
Then start the **GoogleAppEngineLauncher** program that can be found under **Applications**. Use the **File -> Add Existing Application** command and navigate into the **apps** directory and select the **ae-01-trivial** folder. Once you have added the application, select it so that you can control the application using the launcher.



Name	Path	Port
ae-01-trivial	/Users/py4inf/Desktop/apps/ae-01-trivial	8080

Once you have selected your application and press **Run**. After a few moments your application will start and the launcher will show a little green icon next to your application. Then press **Browse** to open a browser pointing at your application which is running at **http://localhost:8080/**

Paste **http://localhost:8080** into your browser and you should see your application as follows:

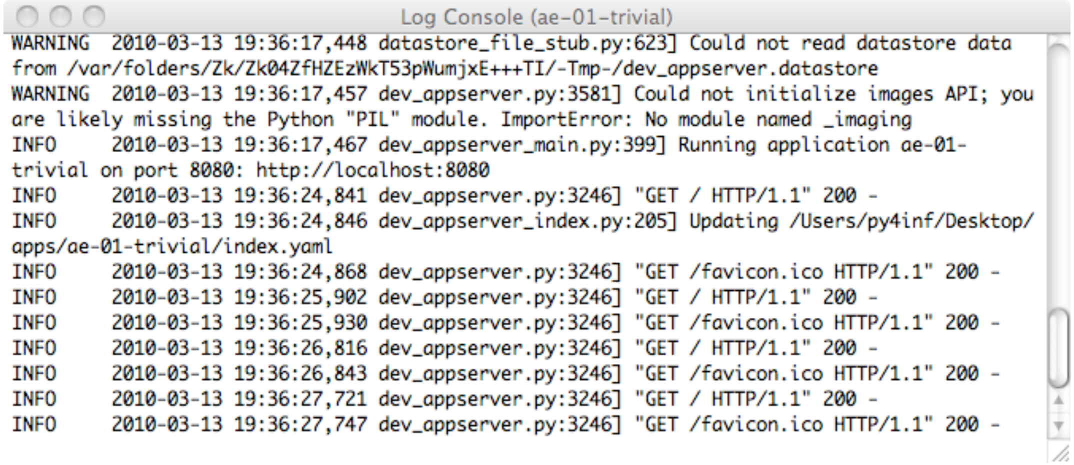


```
Hello there Chuck
```

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 when you are interacting with your application in the browser. Select your application in the Launcher and press the **Logs** button to bring up a log window:

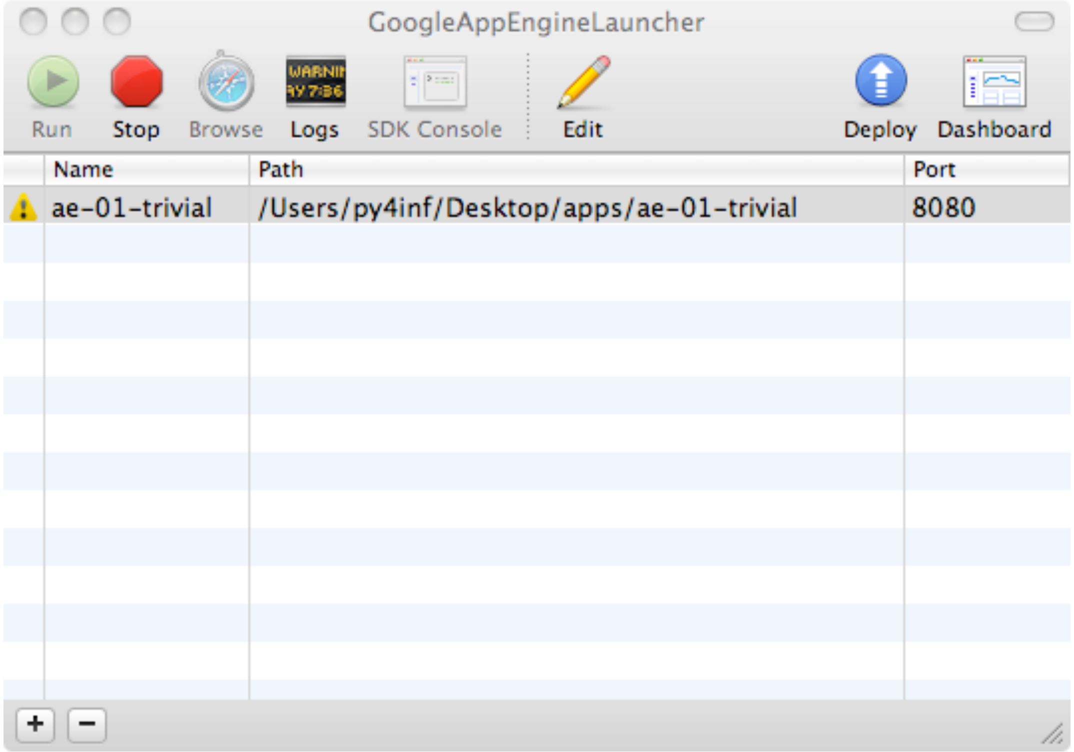


```
WARNING  2010-03-13 19:36:17,448 datastore_file_stub.py:623] Could not read datastore data
from /var/folders/Zk/Zk04ZfHZEzWkT53pWumjxE+++TI/-Tmp-/dev_appserver.datastore
WARNING  2010-03-13 19:36:17,457 dev_appserver.py:3581] Could not initialize images API; you
are likely missing the Python "PIL" module. ImportError: No module named _imaging
INFO     2010-03-13 19:36:17,467 dev_appserver_main.py:399] Running application ae-01-
trivial on port 8080: http://localhost:8080
INFO     2010-03-13 19:36:24,841 dev_appserver.py:3246] "GET / HTTP/1.1" 200 -
INFO     2010-03-13 19:36:24,846 dev_appserver_index.py:205] Updating /Users/py4inf/Desktop/
apps/ae-01-trivial/index.yaml
INFO     2010-03-13 19:36:24,868 dev_appserver.py:3246] "GET /favicon.ico HTTP/1.1" 200 -
INFO     2010-03-13 19:36:25,902 dev_appserver.py:3246] "GET / HTTP/1.1" 200 -
INFO     2010-03-13 19:36:25,930 dev_appserver.py:3246] "GET /favicon.ico HTTP/1.1" 200 -
INFO     2010-03-13 19:36:26,816 dev_appserver.py:3246] "GET / HTTP/1.1" 200 -
INFO     2010-03-13 19:36:26,843 dev_appserver.py:3246] "GET /favicon.ico HTTP/1.1" 200 -
INFO     2010-03-13 19:36:27,721 dev_appserver.py:3246] "GET / HTTP/1.1" 200 -
INFO     2010-03-13 19:36:27,747 dev_appserver.py:3246] "GET /favicon.ico HTTP/1.1" 200 -
```

Each time you press **Refresh** in your browser – you can see it retrieving the output with a **GET** request.

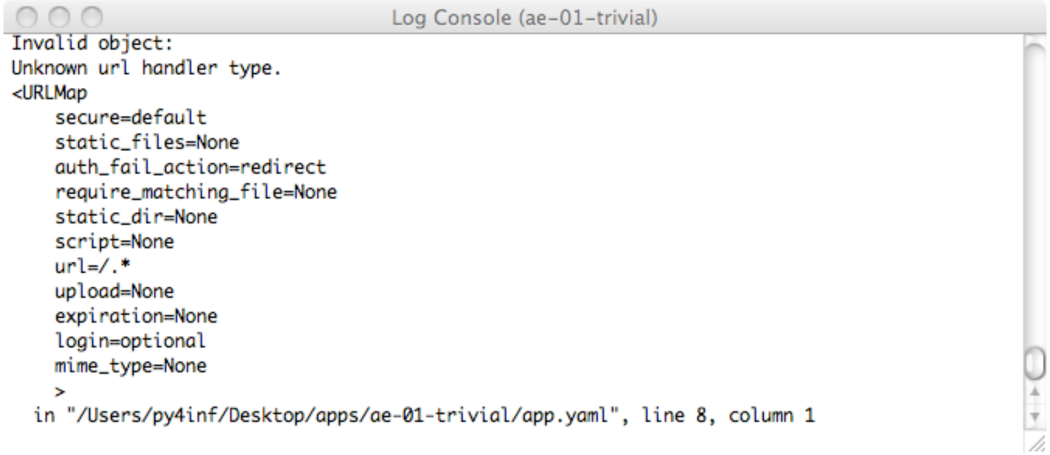
Dealing With Errors

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 App Engine will not start and your launcher will show a yellow icon near your application:



	Name	Path	Port
⚠	ae-01-trivial	/Users/py4inf/Desktop/apps/ae-01-trivial	8080

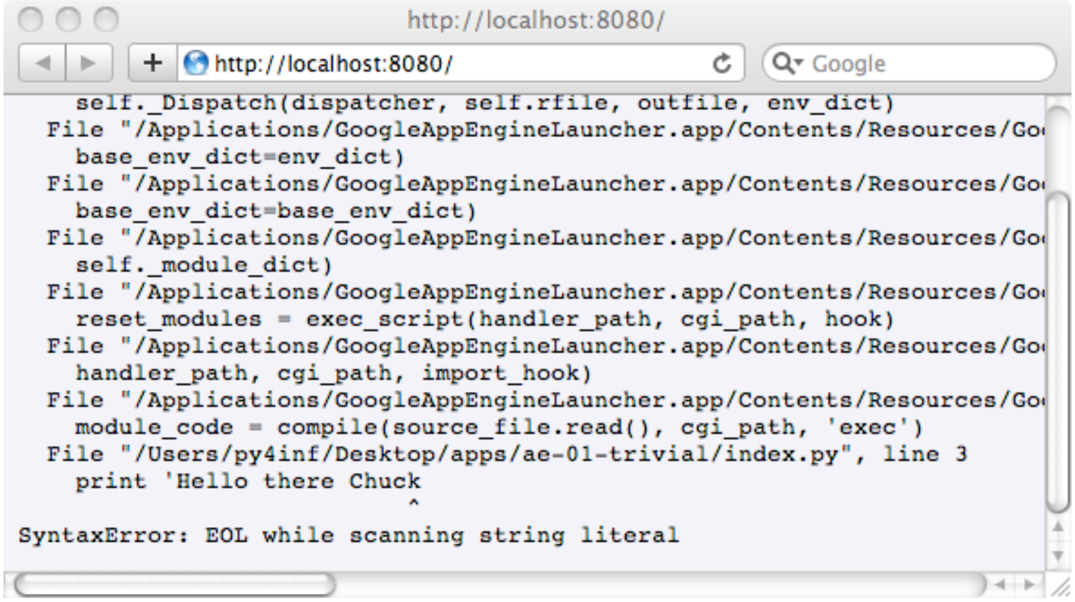
To get more detail on what is going wrong, take a look at the log for the application:



```
Invalid object:
Unknown url handler type.
<URLMap
    secure=default
    static_files=None
    auth_fail_action=redirect
    require_matching_file=None
    static_dir=None
    script=None
    url=/.*
    upload=None
    expiration=None
    login=optional
    mime_type=None
    >
  in "/Users/py4inf/Desktop/apps/ae-01-trivial/app.yaml", line 8, column 1
```

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, a Python trace back error will appear in your browser.



```
    self._Dispatch(dispatcher, self.rfile, outfile, env_dict)
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/Go
    base_env_dict=env_dict)
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/Go
    base_env_dict=base_env_dict)
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/Go
    self._module_dict)
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/Go
    reset_modules = exec_script(handler_path, cgi_path, hook)
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/Go
    handler_path, cgi_path, import_hook)
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/Go
    module_code = compile(source_file.read(), cgi_path, 'exec')
  File "/Users/py4inf/Desktop/apps/ae-01-trivial/index.py", line 3
    print 'Hello there Chuck
                           ^
SyntaxError: EOL while scanning string literal
```

The error you need to see is likely to be the last few lines of the output – in this case I made a Python syntax error on line one of our one-line application.

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, use the Launcher, select your application and press the **Stop** button.

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