**Assignment No-1**

**Aim:** Install Google App Engine. Create hello world app and other simple web applications using python/java.

# Objective:

* Install and use the Google App Engine..
* Understand the development process for building web applications in cloud.

# Theory:

Google App Engine (GAE) is a platform-as-a-service product that provides web app developers and enterprises with access to Google's scalable hosting and tier 1 internet service.GAE requires that applications be written in Java or Python store data in Google Bigtable and use the Google query language. Noncompliant applications require modification to use GAE.

Google App Engine primarily supports Go, PHP, Java, Python, Node.js, .NET and Ruby applications, although it can also support other languages via "custom runtimes". The service is free up to a certain level of consumed resources and only in standard environment but not in flexible environment. ython web frameworks that run on Google App Engine include Django, CherryPy, Pyramid, Flask, web2py and webapp2, as well as a custom Google-written webapp framework and several others designed specifically for the platform that emerged since the release

**Task1: To install and run with Google App Engine**

You need to first install the Google App Engine Software Development Kit (SDK)  on Windows. 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.

# Pre-­Requisites: Python 2.5.4

If you don't already have Python 2.5.4 installed in your computer, download and Install Python 2.5.4 from:

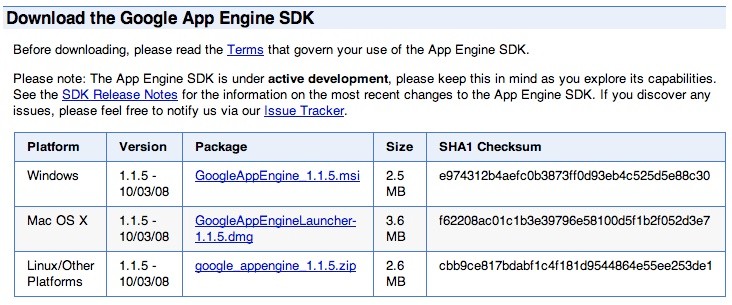
<http://www.python.org/download/releases/2.5.4/>

# Download and Install

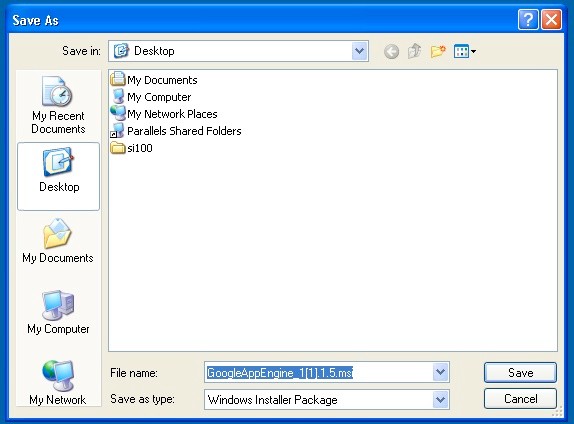
You can download the Google App Engine SDK by going to:

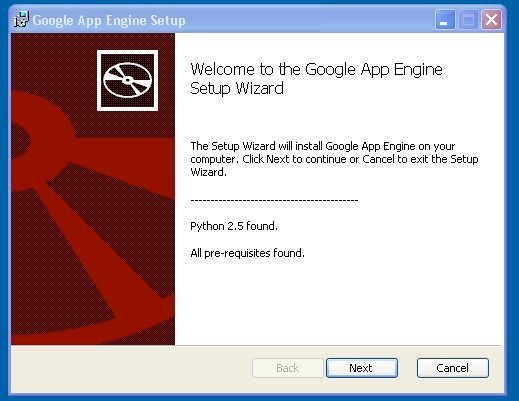
<http://code.google.com/appengine/downloads.html>

and download the appropriate install package.



Download the Windows installer – the simplest thing is to download it to your Desktop or another folder that you remember.



Double Click on the **GoogleApplicationEngine** installer.

Click through the installation wizard, and it should install the App Engine. If you do not have Python 2.5, it will install Python 2.5 as well.

Once the install is complete you can discard the downloaded installer



# Task 2: 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 Desktop called “**apps**” – the path to this folder is:

# C:\Documents and Settings\csev\Desktop\apps

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

# C:\ Documents and Settings \csev\Desktop\apps\ae-­01-­trivial

Using a text editor such as JEdit (www.jedit.org), 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

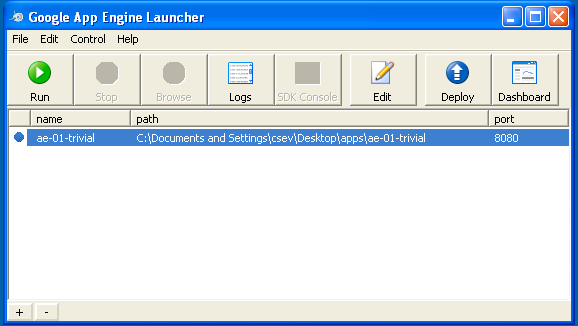
**Note:** Please do not copy and paste these lines into your text editor – you might end up with strange characters – simply type them into your editor.

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

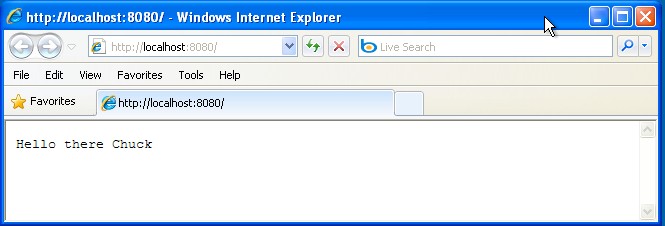
print 'Content-Type: text/plain' print ' '

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.



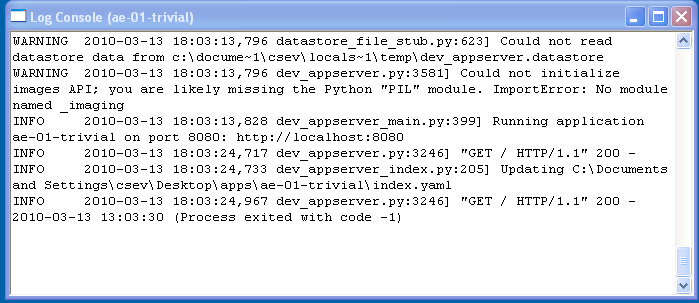
Once you have selected your application and press **Run**. After a few moments your application will start and the launcher will show a little green icon next to your application. Then press **Browse** to open a browser pointing at your application which is running at **http://localhost:8080/**

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

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.

# Task 3: Watching the Log

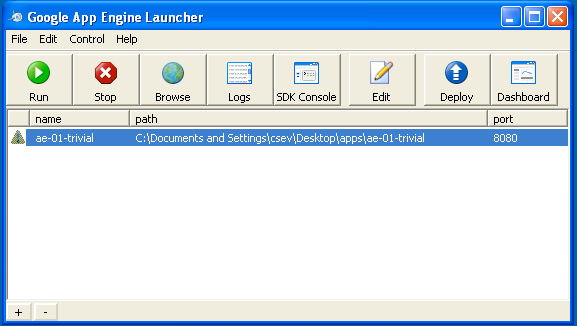
You can watch the internal log of the actions that the web server is performing when you are interacting with your application in the browser. Select your application in the Launcher and press the **Logs** button to bring up a log window:



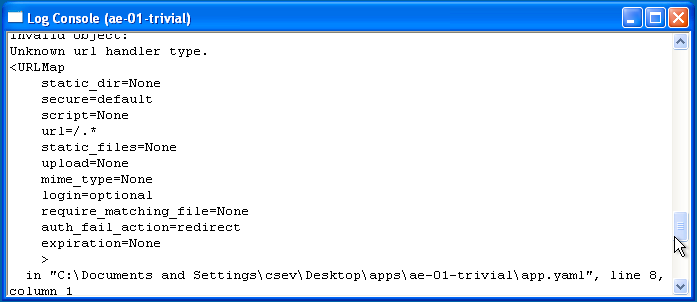
Each time you press **Refresh** in your browser – you can see it retrieving the output with a **GET** request.

# Task 4: 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:

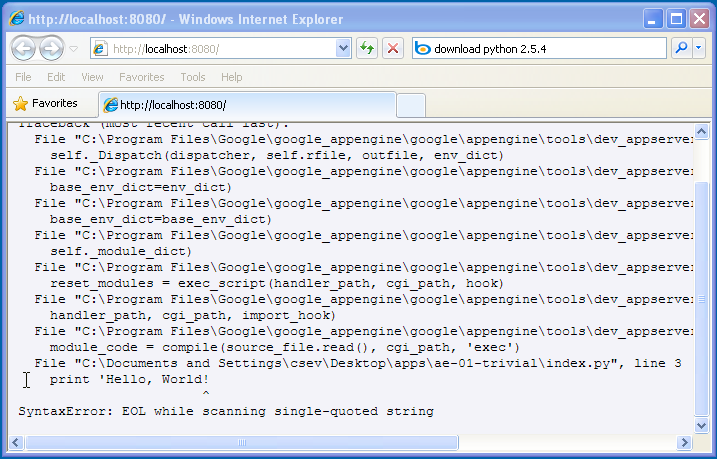


To get more detail on what is going wrong, take a look at the log for the application:



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.



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

When you make a mistake in the **app.yaml** file – you must then 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.