Hands-on: App Engine Fundamentals

Updated: March 20, 2015

This exercise leads you through how to create a simple “Hello World” application and deploy to Google App Engine.

> **Important:** The instructions in this tutorial assume that you are using a terminal on your local machine and running a browser on that same machine to test against the local development server.

Contents

What You Will Learn

You will learn:

- How to create a “Hello World” application using Java or Python
- How to configure, deploy and view your hosted application on App Engine

What You Will Do

You will create your first App Engine application. You will then run it locally and deploy to App Engine.

Get Set Up

Before we continue, you will need to download the Google Cloud SDK for your preferred language (Java or Python), which includes a web server application that simulates the App Engine environment and tools to deploy your application to the App Engine production environment.

1. Please refer to the installation guides for Java and Python to setup your development machine with the necessary software. This includes the Google Cloud SDK for Python or Java developers, a text editor (such as TextMate) or an IDE such as Eclipse or PyCharm. (Note: as of March 20, 2015 we strongly recommend using a text editor).

2. You'll need an application ID (project ID) in order to deploy your app to App Engine in production. If you already have a project to test, you can skip this section. Otherwise, create a new project as follows:

a. Visit Google Developers Console in your web browser, and click **Create Project**.
b. Supply the project name *HelloWorld* and accept the project ID that is auto-generated for you. **Note down the Project ID since we will need that later while deploying our application**.
c. Click **Create**.
d. Enable billing.

Creating the Application

Follow the next steps to create a “Hello World” App Engine application.

1. You can use the Try Google App Engine Now guide to download starter projects for a language of your choice.
2. Select your prefered language i.e. Java or Python. This will show the respective project code and you can study the various files in brief. The page also shows you the next steps in terms of downloading the project, running it locally and deploying the project, which is what we will do next.
3. To download the Project code, simply click on the blue **Download this code** button.

Now follow the steps for Java or Python language as given below:

Java - Local Testing and Deployment

1. Once the download is complete, you will have a copy of the **appengine-try-java.zip** file on your machine. Copy it to an appropriate folder and unzip it. This will create a Maven project directory named **appengine-try-java** with a src/ directory and a **pom.xml** file.
2. Open a terminal or command prompt to go to the **appengine-try-java** folder.
3. **Build** the project locally via the following command :

```
mvn package
```

4. Ensure you have the necessary Cloud SDK component for developing Java projects:

```
gcloud components update pkg-java
```

5. **Run** the project locally via the following command:

   Mac OS or Linux:

   ```
   dev_appserver.sh target/appengine-try-java-1.0
   ```

   Windows:

   ```
   dev_appserver.cmd target/appengine-try-java-1.0
   ```

6. Visit the locally running app in your browser at 'http://localhost:8080' to try it out.
7. You can stop the server by pressing Ctrl-C in the command window.
8. To Deploy your application, you will need an existing **application id** that you have or the one that you created in *Get Set up* above.
9. Execute the following command from the terminal / command prompt. This assumes that you are in the root directory i.e. **appengine-try-java .** Note that you will need to replace the <APP_ID> below with your application id.

   Mac OS or Linux:

   ```
   appcfg.sh -A <APP_ID> update target/appengine-try-java-1.0
   ```

   Windows:

   ```
   appcfg.cmd -A <APP_ID> update target/appengine-try-java-1.0
   ```

   On successful deployment, you can visit **http://<APP_ID>.appspot.com** url.

Python - Local Testing and Deployment

1. Once the download is complete, you will have a copy of the **appengine-try-python-X.zip** file on your machine, where X is either **Django**, **Flask** or **Bottle** depending on the Python framework that you choose. Copy it to an appropriate folder and unzip it. We will assume that you have downloaded the **appengine-try-python-django.zip** file.
2. Ensure you have the necessary Cloud SDK component for developing Python projects:

   ```
   gcloud components update pkg-python
   ```

3. Go to **appengine-try-python-django** directory and **Run** the application locally via the following command:

   ```
   dev_appserver.py appengine-try-python-django
   ```

4. Visit the locally running app in your browser at 'http://localhost:8080' to try it out.

5. You can stop the server by pressing Ctrl-C in the command window.
6. To Deploy your application, you will need an existing **application id** that you have or the one that you created in *Get Set up* above.
7. Execute the following command from the terminal / command prompt. This assumes that you are in the root directory i.e. **appengine-try-django.** Note that you will need to replace the <APP_ID> below with your application id.

   ```
   appcfg.py -A <APP_ID> update appengine-try-python-django
   ```

   On successful deployment, you can visit **http://<APP_ID>.appspot.com** url.

Project Dashboard

Once you have deployed the application, you can visit the project dashboard, which gives you information on your instances, quota, log files and much more.

To visit the project dashboard:

1. Open the Admin Console for the application.
2. Click on the application Id that matches the one that you deployed. This will lead you to the project dashboard page.

Introducing Conference Central

The instructor will now introduce Conference Central, an existing App Engine application that incorporates App Engine features that you will learn through this course. These features include Users API, Datastore API, Memcache API, Task Queue API, Cron Jobs and more.

The Conference Central application will come up during your Code lab exercises for some modules where you will be guided in implementing a particular feature in the application. As an example in Module 4, you will be provided with the project ZIP file (Java or Python) that includes a stub of Conference Central and you will implement Security via the Users API.

Additional Resources

- Google App Engine Documentation
- Admin Console
- Developer Console

Summary

In this exercise, you learned how to build and deploy apps with App Engine.

Although they are not covered in this exercise, Google App Engine has an extensive feature list to explore.