

# Google App Engine for Java 快速使用指南

Version : 1.0.0 2009-10-13



# DOCUMENT HISTORY

Ed.	Version	Author	Change
1	0.1.0	孟宁(mengning@ustc.edu.cn)	Initial(2009-08-25)
2	1.0.0	朱想(xiangcumt@163.com)	Finished Google App Engine for Java 快速 使用指南(2009-10-13)
3	1.0.1	张晓东(jakiejia@163.com)	对注册与部署做了一些修改(2009-10-15)

办公地点:明德楼 A302 电话:0512-68839302 QQ 群: 40931981 E-mail:mengning@ustc.edu.cn





目录	2
1 App Engine 简介	3
1.1 什么是 Google App Engine?	3
1.2 应用程序环境	3
1.3 Java 运行时环境	4
2 注册与部署	4
2.1 申请 Google 账户	4
2.2 申请 Application 域名	5
2.3 搭建 Google App Engine for Java的本地模拟运行环境	6
2.4 Appengine-java-sdk本地运行	6
2.5 上传应用程序到 App Engine 上	7
3 参考资料	9

## 1 App Engine 简介

## 1.1 什么是 Google App Engine?

Google App Engine 可让您在 Google 的基础架构上运行您的网络应用程序。App Engine 应用程序易于构建和维护,并可根据您的访问量和数据存储需要的增长轻松扩展。使用 Google App Engine,将不再需要维护服务器:您只需上传您的应用程序,它便可立即为您的用户提供服务。您可以使用 Google 企业应用套件通过自己的域名(例如 http://www.example.com/)提供应用程序。或者,您可以使用 appspot.com 域上的免费域名来为您的应用程序提供服务。您可以与全世界的人共享您的应用程序,也可以限制为只有贵组织的成员可以访问。

Google App Engine 支持以几种编程语言编写的应用程序。通过 App Engine 的 Java 运行时环境,您可以使用标准 Java 技术(包括 JVM、Java servlet 和 Java 编程语言,或使用基于 JVM 的解释器或解译器的任何其他语言,例如 JavaScript 或 Ruby)构建应用程序。App Engine 还提供一个专用的 Python 运行时环境,该环境包括一个快速 Python 解释器和 Python 标准 库。Java 和 Python 运行时环境构建为确保应用程序快速、安全运行,并不受系统上的其他应用程序的干扰。

在 App Engine 中,您只需为您使用的资源付费。没有设置成本,也没有 重复的费用。您的应用程序使用的资源,如存储空间和带宽以千兆字节衡量, 并以有竞争力的费率收费。您可以控制您的应用程序可以消费的最大资源量, 使其一直保持在预算范围内。

可以免费开始使用 App Engine。所有应用程序都可以使用多达 500 MB 的存储空间,以及可支持每月约 500 万页面浏览量的足够的 CPU 和带宽,完全免费。为您的应用程序启用付费后,您的免费配额将提高,您只需为使用的超过免费水平的资源付费。

## 1.2 应用程序环境

通过 Google App Engine,即使在负载很重和数据量极大的情况下,也可以 轻松构建能安全运行的应用程序。App Engine 包括以下功能:

- 动态网络服务,提供对常用网络技术的完全支持
- 持久存储空间,支持查询、分类和事务
- 自动扩展和负载平衡
- 用于对用户进行身份验证和使用 Google 帐户发送电子邮件的 API
- 一种功能完整的本地开发环境,可以在您的计算机上模拟 Google App Engine

#### 中国科学技术大学软件学院 SCHOOL OF SOFTWARE ENGINEERING OF USTC

• 用于在指定时间和定期触发事件的计划任务

## 1.3 Java 运行时环境

您可以使用常用 Java 网络开发工具和 API 标准开发在 Java 运行时环境 中运行的应用程序。您的应用程序使用 Java Servlet 标准和环境交互,并可 以使用常用网络应用程序技术,例如 JavaServer 页面 (JSP)。

Java 运行时环境使用 Java 6。App Engine Java SDK 支持使用 Java 5 或 6 开发应用程序。

该环境包括 Java SE 运行时环境 (JRE) 6 平台和库。沙盒环境的限制在 JVM 中实现。只要应用程序不超过沙盒的限制,就可以使用任何 JVM 字节码或 库功能。例如,尝试打开套接字或向文件写入的字节码将引发运行时异常。

您的应用程序使用 Java 标准 API 来访问大多数 App Engine 服务。对于 App Engine 数据存储区, Java SDK 包括 Java 数据对象(JDO)和 Java 持久 性 API (JPA) 接口的实现。应用程序可使用 JavaMail API 通过 App Engine 邮件服务发送电子邮件。java.net HTTP API 访问 App Engine 网址抓取服 务。App Engine 还包括低级 API 以使用其服务实现额外适配器或直接从应用 程序使用。

通常, Java 开发人员使用 Java 编程语言和 API 来实现 JVM 的网络应用 程序。通过使用 JVM 兼容的解译器或解释器,您还可以使用其他语言来开发网 络应用程序,例如 JavaScript、Ruby 或 Scala。

## 2 注册与部署

## 2.1 申请Google 账户

首先,如果要想在Google 的基础架构上运行自己的网络应用程序,享受 Google 为我们提供的优异环境,必须注册一个Google 账户;其注册步骤如下:

注册网址为: <u>http://appengine.google.com/</u>





## 2.2 申请Application 域名

在 <u>http://appengine.google.com</u> 链接处登录刚刚创建的账户,进入如下 界面(如下图):

Google app engine

#### Welcome to Google App Engine

Before getting started, you want to learn more about developing and depl Learn more about Google App Engine by reading the <u>Getting Started Gui</u>

Create an Application

点击 "Create an Application" 按钮, 会出现一个手机验证界面, 当输入一个正确的手机号码时, 会向手机发送一个验证码(如下图):

#### Verify Your Account by SMS

To create applications with Google App Engine, you need a verification code. Select the country and carrier for your mobile phone number. The verification code will be sent to it via SMS. Note you will only need to verify your account once.

Country and Carrier:
Other (Not Listed) 
If your country and carrier are not on the list, select Other (Not Listed). What carriers are supported?

Mobile Number:
Include your country code and full phone number. eg. +1 650 555 1212

Send

点击Country and Carrier 选择所在的国家,默认只有三个国家,如果所在的国家不在上面所列出的名称,可以选择默认Other(Not Listed)选项(如下图):



当填入Mobile Number 时要注意,并不是简单的将手机号码填入,而是要加入 county code,可以点击Mobile Number下面的超链接county code去寻找所在国 家的county code,中国的county code 为+86,输入格式为"+86 手机号码", 等待片刻,将手机上的Account Code 输入如下界面即可(如下图)

Within a few minutes, you should receive a text message on your phone that includes a verification code. When you receive it, enter it below. If you don't receive the text message, try sending it again, or see the App Engine FAQ.

Send

然后点击"Send"按钮。会出现如下界面:



## Create an Application

Application Identifier:
.appspot.com Check Availability
You can map this application to your own domain later. Learn more
Application Title:
Displayed when users access your application.
Authentication Options (Advanced):       Learn more         Google App Engine provides an API for authenticating your users. If you choose not to use this, anyone in the world will access your application. However, if you choose to use this, you'll need to specify now who can sign in to your application.         Open to all Google Accounts users (default)       If your application uses authentication, anyone with a valid Google Account may sign in. (This includes all Gmail Accounts *not* include accounts on any Google Apps domains.)         Edit
Save Cancel

输入一个您的应用程序标识,点"Save"保存。 记住你的 Application-ID,十分有用,您的域名即为 Application-ID. appspot.com

## 2.3 搭建 Google App Engine for Java 的本地模拟运行环境

首先下载并安装 JDK,可以选择最新版本进行安装,网址如下: <u>http://java.com/en/download/faq/develop.xml</u>

其次,下载Google App Engine SDK for Java如下网址下载最新的SDK: <u>http://code.google.com/intl/zh-CN/appengine/downloads.Html</u>,选择 windows 版本,根据安装提示安装即可。

## 2.4 Appengine-java-sdk本地运行

点击"开始" -> "运行",输入 cmd 进入命令行模式在 C:\GAE\appengine-javasdk\appengine-java-sdk\bin>目录下执行 dev\_appserver.cmd ..\demos\guestbook\war 启动服务器,(如下图):



打开浏览器,在地址栏中输入 <u>http://localhost:8080</u>可以看到示例程序 的运行结果。(如下图):



## 2.5 上传应用程序到 App Engine 上

在 Web 创建,完成以后,可以将其打包成 war 文件发布,过程如 jsp 开发差不多,现在有几种操作方法如下,

## 方法一: cmd 中实现

在上传我们的示例程序之前必须还要做两件事情:

(1) 进入 Google SDK 的 bin 目录,用编辑器打开 appcfg.cmd,在文件中填入如下命令(如下图):

appcfg.cmd



否则会出现 Google 引擎找不到 javac. exe 的错误,这是因为默认搜索的是 jre 中的 javac. exe 文件,而此文件在 jdk 的安装目录中。

(2)进入示例程序中 war\WEB-INF 目录打开 appengine-web. xml 文件将我们 申请到的域名不要加后缀填入<application>中(见下图):

appengine-web.xml appcfg.cmd

* 🗋 🗳 🖆 🗐	1 A 🖹 🗮 🛍 🖹 🥝 😕 📬 🏝 💌 💌 🖬 🗐 🖬 🗛 🐜 🖧 😓
▲ 打开 资源管理器	Q
	<pre>3 <application>china-it-jackie</application> 4 <version>1</version> 5</pre>

OK,在命令行窗口下执行命令进入 SDK 的 bin 目录:输入如下命令将示例程 序上传到 Google App Engine 上(见下图):



C:\GAE\appengine-java-sdk>cd appengine-java-sdk

C:\GAE\appengine-java-sdk\appengine-java-sdk>cd bin

C:\GAE\appengine-java-sdk\appengine-java-sdk\bin>appcfg.cmd update ..\demos\gues tbook\war

稍后,屏幕上会出现输入 Email 的操作,应该将 Google 账户输入,并且输入 Google 账户的密码,而不是邮箱的密码。等待片刻即可看到 success (如下图):

```
Reading application configuration data...
2009-10-12 22:50:44.253::INFO: Logging to STDERR via org.mortbay.log.StdErrLog
Beginning server interaction for ...
Success.
Cleaning up temporary files...
C:\GAE\appengine-java-sdk\appengine-java-sdk\bin>
```

登录到我们的 Google 账户,并且进入我们的 Application 中,可以看到上传的 示例程序的版本信息,在浏览器的地址栏中输入我们的 Application 域名即可看 到我们的示例程序的效果(如下图):

🚱 🕞 👻 🖉 http://china-it-jackie. appspot. com/
文件 (E) 编辑 (E) 查看 (Y) 收藏夹 (A) 工具 (E) 帮助 (H)
NOTION ◇ 内贝切目的サピナカ日 ▼ ● 身切安王 ▼ ・・・・ 豆求信息 ▼
😭 🏟 🔡 🗸 🤧 网易电子邮箱 - 极速3.0 🏈 http://staff.ustc.edu 🏈 http://china-it-jac 🗙
Hello! <u>Sign in</u> to include your name with greetings you post.
An anonymous person wrote:
good
An anonymous person wrote:
Post Greeting

## 方法二:利用 IDE 工具打包,如 Eclipse

右键点击你想打包的文件或者项目,选择"导出...",在弹出的对话框中选择"WAR文件",上面有许多选项,还可以选"EAR","JAR",个人觉得这个很方便的!

### 方法三:利用 ANT 工具打包

首先配置好 build. xml 文件, 然后 dos 下输入 ant ... war, 即可。

打包程序之前,您需要在 WEB-INF 文件夹下添加一个 appengine-web.xml 文件,把下面内容中 *application-id* 换成您实际 id 就可以了: <?xml version="1.0" encoding="utf-8"?>

<appengine-web-app

xmlns="http://appengine.google.com/ns/1.0">
<application>application-id</application>
<version>1</version>
</appengine-web-app> **建议**: 建议你把 UTF-8 作为所有文件的编码,否则可能会出现意外效果。
建立工程并打包以后,你会得到一个 app.war,由它解压出来的目录比如为
E:\app,现在下载并解压 Google 提供的 SDK,用记事本打开,把@java 换成@你
的 JDK 路径,形如:
@set PATH=C:\Program Files\Java\jdk1.6.0\_10\bin **其中:** 引号在路径中有空格的时候不能去掉。

保存后,用windows下cmd命令行进入到它的解压文件夹下的bin文件夹, 附带参数为"空格update E:\app",例如:

C:\Program Files\Googleengine\bin>appcfg.cmd update demos\guestbook 接着会提示你输入邮箱名和密码,看见"Success."就说明成功了。

之后访问 <u>http://appengine.google.com/</u> 就可以看到自己的应用程序已经挂在 App Engine 服务器上了。

## 3 参考资料

- [1] <u>http://code.google.com/intl/en/appengine/docs/java/gettingstarted/</u>
- [2] <u>http://blog.csdn.net/truewell/archive/2009/07/03/4318843.aspx</u>
- [3] <u>http://www.sun.com/</u>