Introduction

This quick reference card will help you get started with the Vaadin framework on Google App Engine (GAE). If you have the prerequisites installed, you'll literary be creating highly interactive web applications in 5 minutes.

Prerequisites

At a minimum you'll need:

- The Java 1.6 JDK. If you don't have this already, this tutorial is probably going to be a bit too advanced for you.
- Maven. Download and install it from http://maven.apache.org/. It'll be very handy if you add it to your path.
- The Google App Engine SDK for Java. From http://code.google.com/appengine/downloads.html. This tutorial was written for version 1.4.0, but it should work flawlessly with any later versions.

Optionally, you might find the Eclipse IDE with the following plugins useful:

- m2eclipse. http://m2eclipse.sonatype.org/installing-m2eclipse.html
- Google Plugin for Eclipse. http://code.google.com/eclipse/docs/download.html

Those are not necessary and not used in this document, but development can be a lot more enjoyable with an IDE.

Google App Engine Account

You can run App Engine applications locally, but if you want to deploy them to Google's infrastructure you will need an App Engine account.

After you have created an account, you'll also need to create an App Engine application. Don't worry too much about the name, you can always change it later or create additional applications.

https://appengine.google.com/

Configuration

The first step is configuring your environment. We'll put everything in your Maven settings.xml file. Although not everyone will agree this is the right place to put this kind of information, it'll save you a lot of typing later on.

You can read all about the settings.xml file on the Maven site. For Windows 7 users, it's in C:\Users\<user id>\.m2\settings.xml

Create a file containing the following:

```
<settings>
 <servers>
  <server>
   <id>appengine.google.com</id>
   <username>your email</username>
   <password>your pass</password>
  </server>
 </servers>
 <profiles>
   <profile>
    <id>gae-config</id>
   <properties>
    <gae.home>GAE directory</gae.home>
   </properties>
  </profile>
 </profiles>
 <activeProfiles>
  <activeProfile>
   gae-config
  </activeProfile>
 </activeProfiles>
</settings>
```

(NOTE: don't forget to replace the *italic* placeholders with the correct information)

The Project

The quick start project is a Maven project. It will obtain all its dependencies from the parent "Powered by Reindeer" project.

For now we'll use the Sonatype Snapshot repository until a final release is available.

To get started, create a subdirectory where the project will live.

Inside the project directory create the following pom.xml file:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                             http://maven.apache.org/maven-v4_0_0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <groupId>com.streamhead</groupId>
 <artifactId>pow-tryout</artifactId>
 <packaging>war</packaging>
 <version>0.0.1-SNAPSHOT</version>
 <name>Powered By Reindeer project</name>

 <properties>
  <pow.version>0.0.1-SNAPSHOT</pow.version>
 </properties>

 <repositories>
  <repository>
   <id>sonatype-snapshots</id>
   <name>Sonatype Open Source snapshot repository</name>
   <url>https://oss.sonatype.org/content/groups/public/</url>
  </repository>
 </repositories>
 <dependencies>
  <dependency>
   <groupId>com.pow</groupId>
   <artifactId>pow-war</artifactId>
   <version>${pow.version}</version>
   <type>war</type>
  </dependency>
  <dependency>
   <groupId>com.pow</groupId>
   <artifactId>pow-core</artifactId>
   <version>${pow.version}</version>
  </dependency>
 </dependencies>

 <build>
  <plugins>
   <plugin>
    <groupId>net.kindleit</groupId>
    <artifactId>maven-gae-plugin</artifactId>
    <version>0.8.1</version>
    <configuration>
     <serverId>appengine.google.com</serverId>
     <sdkDir>${gae.home}</sdkDir>
    </configuration>
   </plugin>
   <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <configuration>
     <source>1.6</source>
     <target>1.6</target>
     <encoding>ISO-8859-1</encoding>
    </configuration>
   </plugin>
  </plugins>
 </build>
</project>
```

There's a whole lot in there and we won't need it all right away. But this project will help you a very long way.

Let's go over some of the details:

- We first define some project details. Feel free to change them to your liking.
- This project uses the 0.0.1 snapshot of Powered by Reindeer
- The dependencies are loaded from the Sonatype Snapshot repository
- We have a WAR dependency for all the configuration files and a JAR dependency to start creating our own application (see further)
- The maven-gae-plugin makes working with Google App Engine very easy.
- Last is the definition of the JDK version for the project

Running the Web Application

Just enter:

```
> gae:run
```

Maven will now download all dependencies and launch the GAE local container. Open a browser and surf to http://localhost:8080/

You're now running your very first Vaadin web application!

Deploying

Before deploying your application to the Google App Engine infrastructure, we first need to define where to deploy to.

In your project directory create the following directory structure:

```
./src/main/webapp/WEB-INF
```

Inside the WEB-INF directory create a file called appengine-web.xml. It should contain this code:

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns=
 "http://appengine.google.com/ns/1.0">

 <application>your_app</application>
 <version>test</version>

 <system-properties>
  <property name=
       "java.util.logging.config.file"
            value=
       "WEB-INF/logging.properties"
  />
  <property name="com.google.gdata.
                DisableCookieHandler"
            value="true"
  />
 </system-properties>

 <static-files>
  <include path="/VAADIN/**" />
 </static-files>

 <sessions-enabled>
  true
 </sessions-enabled>

 <precompilation-enabled>
  true
 </precompilation-enabled>

</appengine-web-app>
```

(NOTE: make sure "com.google.gdata.DisableCookieHandler" is on one line)

Replace "*your_app*" with the id of the application you created. Now you can deploy the application by simply giving the command:

```
> gae:deploy
```

You can now surf to your application at the url: http://test.*your_app*.appspot.com/

Vaadin Hello World

As a final step, it is now time to create your own implementation of the MainWindow that gets displayed.

You do this by overriding the existing MainWindowImpl class. When the Powered by Reindeer framework is expanded in the future, other options will be available.

Again in your project dir, create the directories:

```
./src/main/java/com/pow/ui
```

Inside the "ui" directory create the file MainWindowImpl.java:

```
package com.pow.ui;

import com.vaadin.ui.Label;
import com.vaadin.ui.Window;
import com.pow.MainApplication;

public class MainWindowImpl extends
Window implements MainWindow {

 private static final long
                serialVersionUID = 1L;

 private MainApplication app;

 public MainWindowImpl(
  final MainApplication app) {

  super(app.getMessage(
                 "MainWindow.title"));
  this.app = app;

  addComponent(
           new Label("Hello world!"));
 }

 @Override
 public Window getMainWindow() {
  return this;
 }
}
```

After this is saved, switch to the root directory of your project and once again run "mvn gae:run". When everything is compiled, you can open your browser and go to http://localhost:8080/

It should now display the message "Hello world!"

Congratulations, you've just created your first Vaadin web application.

Studying Vaadin

Next, you'll want to learn Vaadin. Luckily, this is really simple: head over to http://vaadin.com/book for a very comprehensive, easy to follow and free book.

Troubleshooting

If you receive this error when running Maven:

```
[INFO] Compilation failure
Failure executing javac, but could not
parse the error:
javac: invalid target release: 1.6
```

This means you are using an old version of Java. Make sure you have an up-to-date 1.6 JDK on your system and have it in your path and have set the JAVA_HOME variable.

You change something in the project (like a .java file or an xml configuration), but don't see any changes when you surf to the web application.

Don't forget to deploy the application. If this problem is on the local server, check that previous servers were correctly terminated.

It happens that escaping from "mvn gae:run" doesn't always close the server (especially when running from a Windows version of Eclipse), so you are still seeing another server. Check your task manager or process monitor and kill the appropriate servers.

If you're still stuck, you can always get help at support@streamhead.com

About

This card was created by Peter Backx of http://www.streamhead.com. Peter is the creator of Powered by Reindeer, a framework to easy development of Vaadin applications on Google App Engine. This is a work in progress, so check back often for updates.