

Assignment 1: GWT Tutorial

Due

See course schedule for due date.

Objectives

The purpose of this assignment is to guide you through setting up your IDE (integrated development environment), introduce you to the Google Web Toolkit, have you build a sample GWT application and use the debugger to step through your code to fix a bug.

Procedure and Deliverables

For this assignment you will have to work through the GWT tutorial (step 3). Step 1 and 2 are to help you set up your environment properly.

By the end of this tutorial you should have created the StockWatcher application, used the debugger to step through your code to fix a bug and compiled and tested the application in production mode.

The TAs will check whether you completed the tutorial during the next lab.

Note

If you just copy and paste the code from the tutorial into the editor, you might be able to get everything running, but you will miss out on understanding what the code actually does. Since you will be using GWT in the project you should spend more time to understand the code.

[Step 1 - Eclipse Installation](#) (skip steps 1-3 if software is already installed on your machine)

Download Eclipse 4.4 (Luna) from this site:

<https://eclipse.org/downloads/packages/release/Luna/R>

Be sure to select the appropriate version for your operating system. Even if you have an old version of Eclipse, you must install Eclipse 4.4 for this course. I also recommend creating a new directory to use as your workspace for CPSC 310; do not reuse a workspace that you've used previously.

There were some painful versioning issues in the past with the 310 project, so it's very important that you use the posted version.

Step 2 - Java Software Development Kit (Java SDK) Installation

Eclipse usually comes pre-packaged with a Java SDK. However, you should make sure you have the newest Java SDK. You can download it following the link

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Make sure to choose the correct operating system and bit version (32 or 64).

After downloading the JDK, just install it. It should create the corresponding environment variables automatically.

Step 3 - Installing the Google Web Toolkit

The lab computers already have GWT installed. If you are working on your own computer, you need to install the Eclipse 4.3 version of Google Web Toolkit (GWT) SDK (software development kit, which is a self-contained package with everything you should need to build an application). We will use the Google Plugin for Eclipse 4.3. Follow the instructions at:

<https://developers.google.com/eclipse/docs/install-eclipse-4.3>

You need to install the Google Plugin for Eclipse, the GWT Designer for GPE and the SDKs. You do not need to install the Developer Tools, Google App Engine Tools for Android or NDK Plugins.

In case you encounter any problem on your installation, there is an FAQ page:

<http://code.google.com/eclipse/docs/faq.html>

If you have trouble installing, you may need to try using http rather than https for the installation URL.

Step 4 - Setting up your Eclipse preferences

Update your preferences to point at the Google App Engine SDK and the GWT SDK that you just installed.

On Windows, open Windows > Preferences > Google.

On a Mac, open Eclipse > Preferences > Google.

Under App Engine, add your SDK. It should be in

`/eclipse/plugins/com.google.appengine.eclipse.sdkbundle_1.7.4`

or somewhere similarly named. You can find the eclipse directory wherever you installed eclipse.

If you do not see it there, you will need to download it from

https://developers.google.com/appengine/downloads#Google_App_Engine_SDK_for_Java.

Now, add the SDK under Web Toolkit. It will be in

```
/eclipse/plugins/com.google.gwt.eclipse.sdkbundle_2.5.0.v201208080121-rel-r42
```

(or something similar).

[Step 5 - Building a Sample GWT Application \(StockWatcher\)](#)

Follow the step-by-step tutorial to building a sample GWT application on Google <https://developers.google.com/web-toolkit/doc/latest/tutorial/gettingstarted> Make sure to follow the steps thoroughly and try to understand the code (not just copy and paste).

While working on the tutorial you may encounter some problems:

- In Step 2.4 of Creating the Stockwatcher Application (using Eclipse) you **MUST** select Use Google App Engine and Use default SDK (App Engine). If you don't, you'll get this error "Caused by: javax.jdo.JDOUserException: You have either specified for this PMF to use a "persistence-unit" of "transactions-optional" (yet this doesnt exist!) or you called JDOHelper.getPersistenceManagerFactory with "transactions-optional" as the name of a properties file (and this doesn't exist in the CLASSPATH) "
- You have to add the proper JRE (1.7) to your environment and your project (Windows > Preferences > Java > Installed JREs > Add, and then add the appropriate JRE); then in the project properties under "Java Compiler" check "Enable project specific settings" and choose 1.7 (the proper JRE needs to be installed as well). You should also set 1.7 for the Java Project Facet (right-click the project directory > Properties > Project Facets)
- If you create StockWatcher, run the application, open the specified web site and it then tells you that it may need to be (re)compiled and the text field does not appear, please right click on the project and select Google > GWT Compile (this should help)
- In Step 4 of the "Build a sample GWT App" under point 2. > Handling Keyboard Events: You can use event.getCharCode() to get the character that was pressed, but in order to get the key code (int) that was pressed you need to use the following method: event.getNativeEvent().getKeyCode()
- In Step 5, on the 3rd point, Eclipse may warn you about the deprecated function call for getMediumDateFormat, when you add the following line:

```
// Display timestamp showing last refresh.  
lastUpdatedLabel.setText("Last update : "  
    + DateFormat.getMediumDateFormat().format(new Date()));
```

A deprecated warning just tells you that in future versions of the SDK, this function may not be available. Usually this means we should look for another way to accomplish the task. However, we will ignore the warning for now.

- In step 6, it doesn't explicitly state that you should use "Debug as" instead of "Run as" in order to debug the code.
- Some web browsers don't work that well (or at all) with GWT. You may find external plugins that allow you to use your favourite browser. Internet Explorer 8 is working for me on my Windows machine.

Step 6 - Deploying to the App Engine

Please read this whole section before starting to work on the tutorial.

This tutorial

<https://developers.google.com/web-toolkit/doc/latest/tutorial/appengine>

walks you through the steps of deploying your application using Google's authentication mechanism and communicating with the data storage using JDO. Pay particular attention to the last part (persisting data), but you must to follow all steps.

Note that you must sign up for a Google App Engine account. You need to verify your account by entering a code that Google sends to you via text message. In the past, some students had trouble receiving the text message. It worked for me with Telus, and Google says that they support Rogers. If you're having trouble verifying your account, you can use a friend's number, but each phone number can only be used to verify one Google App Engine account. Past students have also said they needed to turn on email to text by sending a text message with only Subscribe in the body to 0000000000 (yes, 10 zeros).

Use your Stockwatcher project from the first tutorial when you work on this one.

DO NOT DOWNLOAD THE VERSION THAT IS LINKED FROM THE DEPLOYING TO THE APP ENGINE TUTORIAL.

This sample project is based on GWT 2.1 and is therefore incompatible with our setup.

I needed to add two imports that were not specified in the tutorial. When you are working on the "Update the UI" step after adding the LoginService stuff, you'll need to add these two imports

```
import com.google.gwt.user.client.rpc.AsyncCallback;  
import com.google.gwt.core.client.GWT;
```

Be prepared to show your working, deployed web application to the TA at the end of Lab #1 on May 14th.

MORE PRACTICE

You may also want to look through the RPC tutorial (<https://developers.google.com/web-toolkit/doc/latest/tutorial/RPC>). This tutorial will also be helpful for the class project.

You may also want to get some practice retrieving JSON objects from an external server. Start by reading over the overview section, i.e. the first four to five paragraphs, on <https://developers.google.com/web-toolkit/doc/latest/tutorial/clientserver> to get an idea about how client-server communication works. Again, these tutorials are to prepare you for your first sprint, so they are well worth the time it takes to do them.

Now, work on the "Making cross-site requests" tutorial here:

<https://developers.google.com/web-toolkit/doc/latest/tutorial/Xsite>

You will also want to look at the JSON tutorial (<https://developers.google.com/web-toolkit/doc/latest/tutorial/JSON>) as they "Making cross-sites requests" tutorial assumes that you've completed the JSON tutorial.