

# Using GoogleWebToolkit in a Plugin

---



Coolcat 793 posts since

Mar 19, 2007

In short, for all who don't know [GoogleWebToolkit](#) (GWT):

GWT is a compiler that compiles Java code into JavaScript code. You can write your [AJAX](#) application, under some restrictions, in Java. That enables you to share code between client and server side, and remote calls can easily done by an method call. The webpage is build using AWT like widgets, which make your webpage look identically, regardless which browser you are using.

Openfire provides an servlet container, so it should basically be possible to run an GWT servlet with openfire. Anyone did this before?

I wrote a minimal plugin, that uses GWT. But for some reason it produces the following error, each time I'am doing an remote call:

```
2008.05.05 12:53:58 An IncompatibleRemoteServiceException was thrown while processing this call.
com.google.gwt.user.client.rpc.IncompatibleRemoteServiceException: Could not locate requested interface 'org.jivesoftware.openfire
at com.google.gwt.user.server.rpc.RPC.decodeRequest(RPC.java:261)
at com.google.gwt.user.server.rpc.RemoteServiceServlet.processCall(RemoteServiceServlet.java:264)
at com.google.gwt.user.server.rpc.RemoteServiceServlet.doPost(RemoteServiceServlet.java:187)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:727)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:820)
at org.jivesoftware.openfire.container.PluginServlet.handleServlet(PluginServlet.java:251)
at org.jivesoftware.openfire.container.PluginServlet.service(PluginServlet.java:91)
...
Caused by: java.lang.ClassNotFoundException: org.jivesoftware.openfire.plugin.gwt.client.DateService
at java.net.URLClassLoader$1.run(URLClassLoader.java:200)
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(URLClassLoader.java:188)
at java.lang.ClassLoader.loadClass(ClassLoader.java:306)
at java.lang.ClassLoader.loadClass(ClassLoader.java:251)
at org.mortbay.jetty.webapp.WebAppClassLoader.loadClass(WebAppClassLoader.java:375)
at org.mortbay.jetty.webapp.WebAppClassLoader.loadClass(WebAppClassLoader.java:337)
at java.lang.ClassLoader.loadClassInternal(ClassLoader.java:319)
at java.lang.Class.forName0(Native Method)
at java.lang.Class.forName(Class.java:247)
at com.google.gwt.user.server.rpc.RPC.getClassFromSerializedName(RPC.java:681)
at com.google.gwt.user.server.rpc.RPC.decodeRequest(RPC.java:252)
```

I attached the minimal plugin to this post. I'm trying to find the bug for some days now. Would be nice some of you could take a look on it. It's not much code, only many files.

Thanks in advance 😊

*gwt-servlet.jar* is not included, because its "big" and part of GWT, so you have it already.

Directory structure of the plugin:

```
gwt
|- lib
|  |- gwt-servlet.jar <- NOT SUPPLIED, copy it from your GWT installation
|- scripts <- (ant script does ignore this directory)
|  |- gwt-client <- compile Java client code to JavaScript and place it in 'src/web/gwt'
|  |- gwt-server <- compile Plugin to JAR file (call 'ant plugins')
|- src
|  |- gwt/org/jivesoftware/openfire/plugin <- (ant script does ignore this directory)
|  |  |- gwt
|  |  |  |- Test.gwt.xml <- client side module config
|  |  |  |- public <- content of this directory is copied to 'src/web/gwt'
|  |  |  |  |- Test.html
|  |  |  |  |- client <- directory for client side code
|  |  |  |  |  |- Test.java <- entry point for client
|  |  |  |  |  |- DateServiceAsync.java
|  |  |- java/org/jivesoftware/openfire/plugin
|  |  |  |- gwt
|  |  |  |  |- server <- directory for server side code
|  |  |  |  |  |- DateServiceImpl.java
|  |  |  |  |  |- client <- directory for shared code
|  |  |  |  |  |  |- DateService.java
|  |  |  |  |  |- GWT.java <- basic plugin
|  |- web
|  |  |- WEB-INF
|  |  |  |- web-custom.xml <- servlet config
|  |  |  |- gwt <- compiled JavaScript code for clients is placed here
|  |  |  |- gwt.jsp <- usual JSP page
|- plugin.xml
|- LICENSE.html
```

**Attachments:**

- [gwt-test.tar.gz](#) (78.0 K)

Tags: plugin, gwt



[Coolcat](#) 793 posts since

Mar 19, 2007 1. **Re: Using GoogleWebToolkit in a Plugin** May 6, 2008 1:30 PM

I found out that the plugins classes are not in classpath of the servlet Thread.

I have added the following code to the constructor of the servlet (class *DateServiceImpl*)

```
ClassLoader loader = Thread.currentThread().getContextClassLoader();
if (loader instanceof URLClassLoader) {
    URLClassLoader urlLoader = (URLClassLoader)loader;
    URL[] urls = urlLoader.getURLs();
    for (URL url : urls) {
        Log.info("GWT: loader url: "+url.toString()+"");
    }
}
else {
    Log.info("GWT: loader is a "+loader.getClass().getSimpleName()+"");
}
```

Output of this code is:

```
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/jtds.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/servlet.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/activation.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/jasper-runtime.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/startup.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/hsqldb.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/mail.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/commons-el.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/postgres.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/jasper-compiler.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/openfire.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/jdic.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/mysql.jar'
2008.05.06 21:25:19 GWT: loader url: 'file:/opt/openfire/bin/../lib/bouncycastle.jar'
```

Any ideas?

## Using GoogleWebToolkit in a Plugin

thx 😊



Coolcat 793 posts since

Mar 19, 2007 2. **Re: Using GoogleWebToolkit in a Plugin** May 8, 2008 4:39 AM

It does not work now:

I copied *gwt-servlet.jar* and *plugin-gwt.jar* from the extracted JAR file in *openfire/plugins/gwt/* directly into *openfire/lib/*. This works, but is in my opinion a **very bad** solution, especially because it needs an restart of openfire.

Are there any suggestions for a better solution of this problem?



Coolcat 793 posts since

Mar 19, 2007 3. **Re: Using GoogleWebToolkit in a Plugin** May 8, 2008 6:14 AM

The following works:

```
public class GWT implements Plugin {  
  
    /* ... */  
  
    public synchronized void initializePlugin(PluginManager pluginManager, File pluginDirectory) {  
        /* ... */  
  
        pluginClassLoader = pluginManager.getPluginClassLoader(this);  
        DateServiceImpl.pluginClassLoader = pluginClassLoader;  
    }  
  
    /* ... */  
}
```

```
public class DateServiceImpl extends RemoteServiceServlet implements DateService {  
    public static PluginClassLoader pluginClassLoader = null;  
  
    /* ... */  
  
    public String processCall(String payload) throws SerializationException {  
        Log.info("GWT: DateServiceImpl.processCall");  
        if (pluginClassLoader != null) {
```

## Using GoogleWebToolkit in a Plugin

```
        Thread.currentThread().setContextClassLoader(pluginClassLoader);
    }
    return super.processCall(payload);
}

/* ... */
}
```



[Coolcat](#) 793 posts since

Mar 19, 2007 4. **Re: Using GoogleWebToolkit in a Plugin** May 8, 2008 10:52 AM

I have optimized my implementation a bit, added a few comments and published it here:

<http://www.igniterealtime.org/community/docs/DOC-1519>

Have fun 😊



[Gaston Dombiak](#) 3,858 posts since

Sep 26, 2001 5. **Re: Using GoogleWebToolkit in a Plugin** May 8, 2008 1:18 PM

Hey Martin,

Thanks for working on this and creating the document. It certainly looks very interesting. 😊

Regards,

-- Gato



[Larry Sherrill](#) 14 posts since

Sep 30, 2005 6. **Re: Using GoogleWebToolkit in a Plugin** May 11, 2008 9:02 AM

Hi Coolcat,

Thanks for a great article and sample code. I did a gwt/opefire project a few weeks ago and did not have your patience in getting the gwt remoting to work. I ended up using a different approach that does not require gwt-servlet.jar. I created a normal servlet (registered via servlet-mapping in web-custom.xml) to provide the ajax service. I used the gwt RequestBuilder to talk to the servlet. The servlet uses the incoming parameters to decide how to respond. It builds a response in json format. The client uses the gwt json api to parse the response. It's a poor-man's serialization but was sufficient for the project.

I had to put a line in the servlet init method to remove the need to login to hit the ajax service (it seemed necessary at the time):

```
AuthCheckFilter.addExclude("exampleserver/service");
```

Other than that my approach was similar to yours. I used a separate gwt directory and a build.sh to precompile the gwt stuff into the normal plugin directory and then invoked the usual plugin build.xml to build and deploy the plugin. I'll try to bundle up an example showing the json/servlet approach in the next few days.

Larry



[Maxime](#) 61 posts since

Jul 25, 2007 7. **Re: Using GoogleWebToolkit in a Plugin** May 11, 2008 12:31 PM

Hi,

Sorry for my question, I don't know gwt (i.e never dev with it), so I was wondering if it's possible to write a jabber client which would be full ajax client side and with a classic jabber connexion server side (using smack for example) ?

My question is not limited for a server using openfire but it would be great of course to have an ajax client without bosh in a plugin.

Thanks;



**Coolcat** 793 posts since

Mar 19, 2007 8. **Re: Using GoogleWebToolkit in a Plugin** May 12, 2008 1:10 AM

GWT compiles Java source that is compatible with J2SE 1.4.2 or earlier. It does support this [classes from standard library](#). You will find a full list of limitations [here](#).

On Server-side, GWT has no limitations. Server-side code even doesn't need to be translatable.

=> I think thats possible.