



[Experience software development with ROPARDO S.R.L.](#)



- [Home](#)
- [About ROPARDO S.R.L](#)
- [Our websites](#)

GWT/GXT – File Upload

If you are looking for a straightforward **file upload** solution in your **Google Web Toolkit/Ext GWT** based project, this article might help you. The problem with uploading files in such an application is that we can't use the GWT RPC infrastructure for this. On the server side, the RemoteServiceServlet expects the request to be a RPC request, and therefore won't work if we try to upload a file. The remaining possibility would be to use plain Servlet technology, including the apache Commons FileUpload utility. Also, after a short web search, I found the [GWTUpload](#) library, which, based on the Commons FileUpload package, makes it easy to add file upload capability to a GWT application. It also provides great features like showing a progress bar, server response with real information about the process, etc. Here is a complete example of how to use it in a GWT/GXT project:

File Upload using GWTUpload library, GWT and GXT

Download latest version of the library: gwtupload-0.6.3-compat.jar and include it in your classpath. Also, add these libraries to your application: commons-fileupload-1.2.2.jar, commons-io-2.0.1.jar and log4j-1.2.16.jar

Adjust the module file:

```
<module>
<inherits name="com.google.gwt.user.User"/>
<inherits name="com.google.gwt.user.theme.standard.Standard"/>
<entry-point class="com.mycompany.project.client.EP"/>
<inherits name="com.extjs.gxt.ui.GXT"/>

<!-- Include GWTUpload library -->
<inherits name="gwtupload.GWTUpload"/>
<!-- Load dinamically predefined styles in the library when the app starts -->
<stylesheet src="Upload.css"/>
<inherits name="com.mycompany.project.EP"/>
</module>
```

Client Side

Entry point class:

```
package com.mycompany.project.client;

import com.extjs.gxt.ui.client.Style.LayoutRegion;
import com.extjs.gxt.ui.client.widget.Viewport;
import com.extjs.gxt.ui.client.widget.layout.BorderLayout;
import com.extjs.gxt.ui.client.widget.layout.BorderLayoutData;
import com.google.gwt.core.client.EntryPoint;
import com.google.gwt.user.client.ui.RootPanel;
```

```

/**
 * Entry point
 */
public class EP implements EntryPoint {

    private static Viewport viewport;

    private static MyUploader uploader;

    public void onModuleLoad() {

        viewport = new Viewport();

        viewport.setLayout(new BorderLayout());

        showUploader();

        RootPanel.get().add(viewport);
    }

    public static void showUploader() {
        if (uploader == null) {
            uploader = new MyUploader();
            viewport.add(uploader, new BorderLayoutData(LayoutRegion.NORTH,
                MyUploader.SIZE));
            viewport.layout();
        }
    }

    public static void closeUploader() {
        if (uploader != null) {
            viewport.remove(uploader);
            viewport.layout();
            uploader = null;
        }
    }
}

```

File Uploader widget:

```

package com.mycompany.project.client;

import gwtupload.client.IFileInput.FileInputType;
import gwtupload.client.IUploadStatus.Status;
import gwtupload.client.IUploader;
import gwtupload.client.IUploader.OnFinishUploaderHandler;
import gwtupload.client.IUploader.Utills;
import gwtupload.client.MultiUploader;

import com.extjs.gxt.ui.client.Style.HorizontalAlignment;
import com.extjs.gxt.ui.client.Style.Scroll;
import com.extjs.gxt.ui.client.event.ButtonEvent;
import com.extjs.gxt.ui.client.event.SelectionListener;
import com.extjs.gxt.ui.client.widget.LayoutContainer;
import com.extjs.gxt.ui.client.widget.VerticalPanel;
import com.extjs.gxt.ui.client.widget.button.Button;
import com.extjs.gxt.ui.client.widget.layout.FitLayout;
import com.google.gwt.user.client.Element;
import com.google.gwt.user.client.Window;
import com.google.gwt.xml.client.Document;
import com.google.gwt.xml.client.XMLParser;

public class MyUploader extends LayoutContainer {

    public static final float SIZE = 130;

    @Override
    protected void onRender(Element parent, int index) {
        super.onRender(parent, index);

        setLayout(new FitLayout());
    }
}

```

```

final VerticalPanel uploaderPanel = new VerticalPanel();

uploaderPanel.setScrollMode(Scroll.AUTO);

uploaderPanel.setSpacing(10);

uploaderPanel.setHorizontalAlign(HorizontalAlignment.RIGHT);

MultiUploader uploader = new MultiUploader(FileInputType.LABEL);
// we can change the internationalization by creating custom Constants
// file

uploader.setAvoidRepeatFiles(false);

uploader.setServletPath("uploader.fileUpload");
uploader.addOnFinishUploadHandler(new OnFinishUploaderHandler() {
    public void onFinish(IUploader uploader) {

        if (uploader.getStatus() == Status.SUCCESS) {

            String response = uploader.getServerResponse();

            if (response != null) {
                Document doc = XMLParser.parse(response);
                String message = Utils.getXmlNodeValue(doc, "message");
                String finished = Utils
                    .getXmlNodeValue(doc, "finished");

                Window.alert("Server response: \n" + message + "\n"
                    + "finished: " + finished);
            } else {
                Window.alert("Unaccessible server response");
            }

            // uploader.reset();
        } else {
            Window.alert("Uploader Status: \n" + uploader.getStatus());
        }
    }
});

uploaderPanel.add(uploader);

Button closeButton = new Button("Close",
    new SelectionListener<ButtonEvent>() {

        public void componentSelected(ButtonEvent ce) {
            EP.closeUploader();
        }
    });

uploaderPanel.add(closeButton);

add(uploaderPanel);
}
}

```

Server Side

Customized servlet (extending ActionUpload and overriding executeAction):

```

package com.mycompany.project.server;

import gwtupload.server.UploadAction;
import gwtupload.server.exceptions.UploadActionException;

import java.util.List;

```

```

import javax.servlet.http.HttpServletRequest;
import org.apache.commons.fileupload.FileItem;

public class FileUploadServlet extends UploadAction {

    private static final long serialVersionUID = 1L;

    public String executeAction(HttpServletRequest request,
                               List<FileItem> sessionFiles) throws UploadActionException {
        String response = "Received file:";

        for (FileItem item : sessionFiles) {
            if (!item.isFormField()) {
                try {

                    // we can save the received file
                    // File file = File.createTempFile("receivedFile", ".tmp",
                    // new File("C:\\"));
                    // item.write(file);

                    // response += " " + file.getPath();

                    response += " " + item.getName() + ", size= "
                        + item.getSize() + ", ContentType= "
                        + item.getContentType();

                } catch (Exception e) {
                    throw new UploadActionException(e.getMessage());
                }
            }
        }

        try {
            // Remove files from session
            removeSessionFileItems(request);
        } catch (Exception e) {
            throw new UploadActionException(e.getMessage());
        }

        // Send information of the received files to the client.
        return response;
    }
}

```

log4j.properties

```

# Set root category priority to INFO and its only appender to CONSOLE.
log4j.rootCategory=INFO, CONSOLE

# CONSOLE is set to be a ConsoleAppender using a PatternLayout.
log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.CONSOLE.Threshold=DEBUG
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=- %m%n

```

Finally, edit your web.xml and include the customized servlet:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>

    <!-- Default page to serve -->
    <welcome-file-list>
    <welcome-file>EP.html</welcome-file>
    </welcome-file-list>

```

```

<!-- customized file upload servlet -->
<servlet>
<servlet-name>fileUploadServlet</servlet-name>
<servlet-class>com.mycompany.project.server.FileUploadServlet</servlet-class>
</servlet>

<servlet-mapping>
<servlet-name>fileUploadServlet</servlet-name>
<url-pattern>*.fileUpload</url-pattern>
</servlet-mapping>

<!-- max size of the upload request -->
<context-param>
<param-name>maxSize</param-name>
<param-value>262144000</param-value>
</context-param>

<!-- Useful in development mode to slow down the uploads in fast networks.
Put the number of milliseconds to sleep in each block received in the server.
false or 0, means don't use slow uploads -->
<context-param>
<param-name>slowUploads</param-name>
<param-value>0</param-value>
</context-param>

</web-app>

```

Tags: [File Upload GWT](#) [GWTUpload](#) [GXT](#)

Posted in: [Java](#)

[March 3, 2011](#) | [Alexandru Gyulai](#) | [4 Comments](#)

4 Responses



edw

March 19, 2011 1

exactly what i need, thanks for sharing.
Greetings from Indonesia.
:)



Gavin

April 8, 2011 2

Hi,

I tried your code but got an exception launched when trying to upload a file. I added gwtupload-0.6.3-compatible.jar to my buildpath and copied commons-fileupload-1.2.2.jar, commons-io-2.0.1.jar and log4j-1.2.16.jar to the libraries (war/WEB-INF/lib).

On clicking the button send, I got this error :

[WARN] Error for /suImage.fileUpload
java.lang.NoClassDefFoundError: java.rmi.server.UID is a restricted class. Please see the Google App Engine developer's guide


for more details.

```
at com.google.appengine.tools.development.agent.runtime.Runtime.reject(Runtime.java:51)
at org.apache.commons.fileupload.disk.DiskFileItem.(DiskFileItem.java:109)
at org.apache.commons.fileupload.disk.DiskFileItemFactory.createItem(DiskFileItemFactory.java:199)
at org.apache.commons.fileupload.FileUploadBase.parseRequest(FileUploadBase.java:361)
at org.apache.commons.fileupload.servlet.ServletFileUpload.parseRequest(ServletFileUpload.java:126)
at gwtupload.server.UploadServlet.parsePostRequest(UploadServlet.java:796)
at gwtupload.server.UploadAction.doPost(UploadAction.java:159)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:637)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
at org.mortbay.jetty.servlet.ServletHolder.handle(ServletHolder.java:511)
at org.mortbay.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1166)
at com.google.appengine.api.blobstore.dev.ServeBlobFilter.doFilter(ServeBlobFilter.java:58)
at org.mortbay.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1157)
at com.google.apphosting.utils.servlet.TransactionCleanupFilter.doFilter(TransactionCleanupFilter.java:43)
at org.mortbay.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1157)
at com.google.appengine.tools.development.StaticFileFilter.doFilter(StaticFileFilter.java:122)
at org.mortbay.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1157)
at org.mortbay.jetty.servlet.ServletHandler.handle(ServletHandler.java:388)
at org.mortbay.jetty.security.SecurityHandler.handle(SecurityHandler.java:216)
at org.mortbay.jetty.servlet.SessionHandler.handle(SessionHandler.java:182)
at org.mortbay.jetty.handler.ContextHandler.handle(ContextHandler.java:765)
at org.mortbay.jetty.webapp.WebAppContext.handle(WebAppContext.java:418)
at com.google.apphosting.utils.jetty.DevAppEngineWebAppContext.handle(DevAppEngineWebAppContext.java:70)
at org.mortbay.jetty.handler.HandlerWrapper.handle(HandlerWrapper.java:152)
at com.google.appengine.tools.development.JettyContainerService$ApiProxyHandler.handle(JettyContainerService.java:351)
at org.mortbay.jetty.handler.HandlerWrapper.handle(HandlerWrapper.java:152)
at org.mortbay.jetty.Server.handle(Server.java:326)
at org.mortbay.jetty.HttpConnection.handleRequest(HttpConnection.java:542)
at org.mortbay.jetty.HttpConnection$RequestHandler.content(HttpConnection.java:938)
at org.mortbay.jetty.HttpParser.parseNext(HttpParser.java:755)
at org.mortbay.jetty.HttpParser.parseAvailable(HttpParser.java:218)
at org.mortbay.jetty.HttpConnection.handle(HttpConnection.java:404)
at org.mortbay.io.nio.SelectChannelEndPoint.run(SelectChannelEndPoint.java:409)
at org.mortbay.thread.QueuedThreadPool$PoolThread.run(QueuedThreadPool.java:582)
```

I couldn't find any answer to this problem, can you ?

Thanks!

Gavin

-  **Nebojsa**
May 7, 2011 3

Thanks a lot. Works like a charm. Greetings from Macedonia ;)

-  **masfworld**
June 21, 2011 4

The problem is that gwtupload don't work very fine with app engine. I work with gwtupload well in localhost but crash in appengine.

Any idea???

Leave a Reply

Name, required

Email (will not be published), required

Website (if present)

reckoned

fussf

Type the two words:



Submit

☒ Notify me of follow-up comments via e-mail

- [« Previous post](#)
- [Next post »](#)

• Recent Posts

- [Checkboxes unchecked value](#)
- [Meet Ropardo at CeBit 2012](#)
- [Installing PyGraphviz on Windows](#)
- [Convert python object to XML representation](#)
- [Liferay Portlet Development](#)

ROPARDO
IS HIRING!

Click here to find what exciting
positions are available at
ROPARDO

• Subscribe

-    BY FEEDBURNER

• Recent Comments

- fei on [Installing PyGraphviz on Windows](#)
- Stelian Morariu on [GWT 2.1 – Uploading a file using the RPC mechanism](#)
- Thomas Broyer on [GWT 2.1 – Uploading a file using the RPC mechanism](#)
- Mariah on [GWT 2.1 – Uploading a file using the RPC mechanism](#)
- Ioan Seicean on [Installing PyGraphviz on Windows](#)

• Archives

- [March 2012](#) (1)
- [February 2012](#) (1)
- [November 2011](#) (1)
- [September 2011](#) (4)
- [July 2011](#) (3)
- [June 2011](#) (2)
- [May 2011](#) (4)
- [April 2011](#) (4)
- [March 2011](#) (3)
- [February 2011](#) (2)
- [January 2011](#) (2)
- [December 2010](#) (1)
- [November 2010](#) (4)
- [October 2010](#) (4)
- [August 2010](#) (3)
- [July 2010](#) (3)
- [June 2010](#) (6)
- [May 2010](#) (8)
- [April 2010](#) (7)
- [March 2010](#) (9)
- [February 2010](#) (6)
- [January 2010](#) (5)
- [December 2009](#) (7)
- [November 2009](#) (9)
- [October 2009](#) (10)
- [September 2009](#) (14)
- [August 2009](#) (10)
- [July 2009](#) (1)
- [June 2009](#) (1)
- [May 2009](#) (1)
- [April 2009](#) (1)
- [March 2009](#) (1)
- [October 2008](#) (3)
- [October 2007](#) (3)
- [July 2007](#) (4)
- [June 2007](#) (1)
- [May 2007](#) (3)

• Meta

- [Log in](#)
- [Entries RSS](#)
- [Comments RSS](#)
- [WordPress.org](#)

• Categories

- [News](#) (16)
- [Ropardo Team](#) (8)
- [Ropardo Products](#) (6)
 - [File Tracking Client](#) (4)
 - [iManagement](#) (2)
- [Software Development](#) (83)
 - [Microsoft.NET](#) (22)
 - [Java](#) (41)
 - [Oracle](#) (8)
 - [Power Builder](#) (3)
 - [Liferay](#) (5)
 - [Lotus Notes](#) (9)
 - [xWiki](#) (4)
- [System Administration](#) (13)
 - [Linux](#) (10)
 - [Windows](#) (3)
- [Programming](#) (1)
- [Uncategorized](#) (3)
- [Databases](#) (10)
 - [MSSQL](#) (5)
 - [PostgreSQL](#) (3)
- [Microsoft.NET](#) (1)
- [Web Development](#) (29)
 - [ASP/ASPX](#) (3)
 - [Content Management Systems](#) (1)
 - [HTML/CSS](#) (6)
 - [Javascript/AJAX](#) (8)
 - [PHP](#) (7)
- [Oracle E Business Suite](#) (6)

• Tags

[.NET](#)
[ajax](#)
[blog](#)
[C#](#)
[certification](#)
[client](#)
[CMS](#)
[control](#)
[css](#)
[database](#)
[Debugging](#)
[django](#)
[Domino](#)
[Eclipse](#)
[extension](#)
[file tracking](#)
[filter](#)
[fun](#)
[gentoo](#)
[google](#)
[Hibernate](#)
[how to](#)
[html](#)
[image](#)
[iManagement](#)
[import](#)
[Java](#)
[javascript](#)
[jQuery](#)
[liferay](#)
[Linux](#)
[Lotus Notes](#)
[lotus script](#)
[Oracle](#)
[Oracle BI Publisher 11g](#)
[PHP](#)
[portal](#)
[PostgreSQL](#)
[powerbuilder](#)
[Python](#)
[SQL](#)
[Telerik](#)
[velocity](#)
[xml](#)
[Xwiki](#)

☺

© 2012 [ROPARDO s.r.l.](#)

Powered by [WordPress](#). Styled by [Ropardo](#)