Search [Find]

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

GWT 2.1 – Uploading a file using the RPC mechanism

Uploading a file in a project using GWT/EXT-GWT can be done in 2 ways : creating a separate servlet or using a library like the one presented [here](#). Normally, because of browser security restrictions you can't upload a file using RPC. But normally doesn't mean never. This article will show you how to upload files using RPC in GWT projects.

Every now and then you might encounter a situation where you need to upload a file, but it would be very convenient to be able to use a RPC(Remote Procedure Call) to do that because you would like to perform some additional operations with that file.Well not to worry, this article will show you a way to do that.

We are going to start with the RPC implementation. First of all, we need to declare the RPC interfaces, this is no different from implementing an usual RPC. We are going to name our RPC "FileService", and bellow you have the declaration of both interfaces

1. The *FileService* interface

```java
@RemoteServiceRelativePath("GWTWebCbr3")
public interface FileService extends RemoteService {

    /**
     * Method that stores a file on the server. It computes the save path
     * depending on the case type and the case id.
     */
    public String uploadAttachement(String caseId, String fieldName,boolean isNewCase);

    /**
     * Method that deletes a file from the server. It computes the file path
     * depending on the case type and the case id.
     */
    public boolean deleteAttachement(String filePath, int caseID,
                    String fieldName);

    /**
```

```
     * Update the name of the attachment
     */
    public String updateFileName(final String name);
}
```

2. The *FileServiceAsync* interface

```
public interface FileServiceAsync {

    /**
     * Method that stores a file on the server. It computes the save path
     * depending on the case type and the case id.
     */
    void uploadAttachement(String caseId, String id, boolean isNewCase,
                    AsyncCallback callback);

    /**
     * Method that deletes a file from the server. It computes the file path
     * depending on the case type and the case id.
     */
    void deleteAttachement(String filePath, int id, String id2,
                    AsyncCallback callback);

    /**
     * Update the name of the attachment
     */
    void updateFileName(String name, AsyncCallback callback);

}
```

So far there is nothing different, but the service implementation will reveal what makes uploading a file with RPC possible. Bassically you need to overwrite the **service(final HttpServletRequest request,HttpServletResponse response)** method and that's it.

Here's the code for *FileServiceImpl*

```
public class FileServiceImpl extends RemoteServiceServlet implements
                FileService {

//code omitted

    private FileItem uploadedFileItem;

    @Override
    protected void service(final HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {

        boolean isMultiPart = ServletFileUpload
                        .isMultipartContent(new ServletRequestContext(request));

        if(isMultiPart) {
                FileItemFactory factory = new DiskFileItemFactory();
                ServletFileUpload upload = new ServletFileUpload(factory);

                try {
                        @SuppressWarnings("unchecked")
                        List items = upload.parseRequest(request);
                        uploadedFileItem = items.get(0); // we only upload one file

                        if(uploadedFileItem == null) {
                                super.service(request, response);
                                return;
```

```java
                } else if(uploadedFileItem.getFieldName().equalsIgnoreCase(
                            "uploadFormElement")) {
                        fileName = uploadedFileItem.getName();
                        response.setStatus(HttpServletResponse.SC_CREATED);
                        response.getWriter().print("OK");
                        response.flushBuffer();
                }

        } catch(FileUploadException e) {
                LOG.error(e);
        }
    }

    else {
        super.service(request, response);
        return;
    }
}

//code omitted
}
```

After you implement your service, you need to register it and you do that by adding the following lines to the *web.xml* file.

```xml
<servlet>
    <servlet-name>FileService</servlet-name>
    <servlet-class>ro.ropardo.cbr.server.Services.FileServiceImpl</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>FileService</servlet-name>
    <url-pattern>/gwtwebcbr/GWTWebCbr3</url-pattern>
</servlet-mapping>
```

On the client side you need to use a **FormPanel** to upload your file. To tie the components together you need to set the action of the FormPanel to the RPC servlet. Here is an example :

```java
private final FormPanel formPanel;
formPanel = new FormPanel();
formPanel.setAction(GWT.getModuleBaseURL()+"GWTWebCbr3");
formPanel.setEncoding(FormPanel.ENCODING_MULTIPART);
formPanel.setMethod(FormPanel.METHOD_POST);
```

Now all you have to do is submit the form. For example you can submit the form when a user clicks a button :

```java
Button uploadButton = new Button(PARAM_BUTTON_UPLOAD);
uploadButton.addClickHandler(new ClickHandler() {

    @Override
    public void onClick(final ClickEvent event) {
        if(!uploader.getFilename().equalsIgnoreCase(PARAM_EMPTY_STRING)) {
            formPanel.submit();
        }

    }
});
```

You can also add a handler to the form and do some action when the submit completed. For example, I trigger an event if the submit completed and the file was succesfully uploaded. To check if the upload was successful you need to write the status of the operation in the result object from the overriden service() method and then, from the client side, check

that status. Here is an example :

```
formPanel
.addSubmitCompleteHandler(new FormPanel.SubmitCompleteHandler() {

    @Override
    public void onSubmitComplete(final SubmitCompleteEvent event) {
        // call upload method from service with the file's name
        String result = event.getResults();
        if(result.contains(PARAM_OK_STRING)) {
            String name = uploader.getFilename();

            // trigger some event that custom file editor will
             FileReadyForUploadEvent ev = new FileReadyForUploadEvent(name);
            fireEvent(ev);

        }
    }
});
```

And that is how you can upload a file using the RPC mechanism in GWT based projects. I hope you can find this article usefull.

Tags: file GWT 2.1 RPC upload

Posted in: Java, Software Development, Web Development
April 19, 2011 | Stelian Morariu | 11 Comments

11 Responses

- **Jeff MAURY**
  August 18, 2011 1

  Is the code available ?

- **Jorge**
  October 5, 2011 2

  Excellent article, came in very useful, cheers!

- **Fredrik**
  October 26, 2011 3

  Great article! Is there any special reason why we shouldn't do this for all uploading?

- **rkd80**
November 16, 2011 4

What is uploader?

In the case of: uploader.getFilename();

Is that the GWT fileuploader or something else??

- **Sergio**
December 13, 2011 5

It does not work, I get the following error:

Exception while dispatching incoming RPC call
javax.servlet.ServletException: Content-Type was 'multipart / form-data; boundary =————————
28945529321871'. Expected 'text / x-gwt-rpc'.
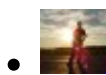
Is there source code?

- **Stelian Morariu**
December 20, 2011 6

Sorry for the late reply :
@ rkd80 yes, that is the GWT fileuploader;
@ Sergio as far as I can tell from your message, your error is related to the content-type you send, have you tried different settings for

```
formPanel.setEncoding(FormPanel.ENCODING_MULTIPART);
```

?

- **[Artiom](#)**
January 28, 2012 7

Hello. Thx for useful post.
Do you have an example how to use gwtupload with progressbar?

Best regards/

- **ze_felipe**
February 9, 2012 8

hi, i´m brazilian and my english is so so.

Then, woks fine for me, but the problem is when i call this service, it

redirect to http:// … / 'myapplication' / services and i cant back to

my apllication. can you help me?

tks

- **Mariah**
  May 25, 2012 9

  Hello thanks 4 da example… but i have a doubt…. the class FileServiceImpl send me an error and say that the method service cannot be override… what im doing wrong? thanks a lot

- [**Thomas Broyer**](#)
  May 27, 2012 10

  This is NOT "uploading using GWT-RPC", it's just using the same servlet for GWT-RPC and "something else" (in this case, file upload, but ti could really be anything).

- **Stelian Morariu**
  May 28, 2012 11

  @Mariah – some details about that error ?

  @Thomas Broyer – what else do you understand by "uploading using GWT-RPC" other than using GWT's RPC servlet for uploading files ?

  Thx for reading and posting.

Leave a Reply

Name, required

Email (will not be published), required

Website (if present)

herexh

wa8

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](#)

  RO PARDO
  IS HIRING!
  Click here to find what exciting
  positions are available at
  ROPARDO

- ## Subscribe

  - 51 TOPOFBLOGS   8 readers 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 Adminstration (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)
    - Javascrip/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

☺