

Here is a J2ME class to handle file uploads via HTTP POST Multipart Requests.

Source Code: HttpMultipartRequest class

```
import java.io.ByteArrayOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Enumeration;
import java.util.Hashtable;

import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;

public class HttpMultipartRequest
{
    static final String BOUNDARY = "-----V2ymHFg03ehbqgZCaK06jy";

    byte[] postBytes = null;
    String url = null;

    public HttpMultipartRequest(String url, Hashtable params, String fileField, String fileName, String fileType)
    {
        this.url = url;

        String boundary = getBoundaryString();

        String boundaryMessage = getBoundaryMessage(boundary, params, fileField, fileName, fileType);

        String endBoundary = "\r\n--" + boundary + "--\r\n";

        ByteArrayOutputStream bos = new ByteArrayOutputStream();

        write(boundaryMessage.getBytes());

        write(fileBytes);

        write(endBoundary.getBytes());

        this.postBytes = bos.toByteArray();

        close();    bos.
    }

    String getBoundaryString()
    {
        return BOUNDARY;
    }

    String getBoundaryMessage(String boundary, Hashtable params, String fileField, String fileName, String fileType)
    {
        StringBuffer res = new StringBuffer("--").append(boundary).append("\r\n");

        Enumeration keys = params.keys();

        while(keys.hasMoreElements())
        {
            String key = (String)keys.nextElement();
            String value = (String)params.get(key);
```

HTTP_Post_multipart_file_upload_in_Java_ME

```
        append("Content-Disposition: form-data; name=\""").append(key).append("\"\\r\\n")
        append("\\r\\n").append(value).append("\\r\\n")
        append("--").append(boundary).append("\\r\\n");
    }
    append("Content-Disposition: form-data; name=\""").append(fileField).append("\"; filename=\""")
    append("Content-Type: ").append(fileType).append("\\r\\n\\r\\n");

return res.toString();
}

public byte[] send() throws Exception
{
    HttpURLConnection hc

    InputStream is = null;

    ByteArrayOutputStream bos = new ByteArrayOutputStream();

    byte[] res = null;

    try
    {
        = (HttpURLConnection) Connector.open(url);

        setRequestProperty("Content-Type", "multipart/form-data; boundary=" + getBoundaryString());

        setRequestMethod(HttpURLConnection.POST);

        OutputStream dout = hc.openOutputStream();

        write(postBytes);    dout.

        close();            dout.

    int ch;

        = hc.openInputStream(is

    while ((ch = is.read()) != -1)
    {
        write(ch);            bos.
    }
        = bos.toByteArray();res
    }
    catch(Exception e)
    {
        printStackTrace();    e.
    }
    finally
    {
    try
    {
    if(bos != null)
        close();            bos.

    if(is != null)
        close();            is.

    if(hc != null)
        close();            hc.
    }
    catch(Exception e2)
```

HTTP_Post_multipart_file_upload_in_Java_ME

```
{
    printStackTrace();          e2.
}
}
return res;
}
}
```

Sample usage

Here's a code snippet to upload a file via `HttpMultipartRequest` class:

```
byte[] fileBytes = getFileBytes(); //retrieve file bytes with your own code

Hashtable params = new Hashtable();
params.put("custom_param", "param_value");
params.put("custom_param2", "param_value2");

HttpMultipartRequest req = new HttpMultipartRequest(
    "http://www.server.com/uploadScript.php",
    params,
    "upload_field", "original_filename.png", "image/png", fileBytes
);

byte[] response = req.send();
```

Sample server code (PHP)

This is a sample PHP script that handles the upload. It doesn't actually save the uploaded file, but only displays some infos about the upload size and parameters.

```
<?php

$filesize = filesize($_FILES['upload_field']['tmp_name']);

echo "The uploaded file size is " . $filesize . " bytes\n";

foreach($_POST as $key => $value)
{
    echo "Parameter name: " . $key . ", value: " . $value . "\n";
}

?>
```