HTTP Post multipart file upload in Java ME

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 = "----------V2ymHFg03ehbqgZCaKO6jy";

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

 public HttpMultipartRequest(String url, Hashtable params, String fileField, String fileName, String fileType, byte[] fileBytes) thr
 {
  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();

  bos.write(boundaryMessage.getBytes());

  bos.write(fileBytes);

  bos.write(endBoundary.getBytes());

  this.postBytes = bos.toByteArray();

  bos.close();
 }

 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);

   res.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");
  }
  res.append("Content-Disposition: form-data; name=\"").append(fileField).append("\"; filename=\"").append(fileName).append("\"\r\n"
   .append("Content-Type: ").append(fileType).append("\r\n\r\n");

  return res.toString();
 }

 public byte[] send() throws Exception
 {
  HttpConnection hc = null;

  InputStream is = null;

  ByteArrayOutputStream bos = new ByteArrayOutputStream();

  byte[] res = null;

  try
  {
   hc = (HttpConnection) Connector.open(url);

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

   hc.setRequestMethod(HttpConnection.POST);

   OutputStream dout = hc.openOutputStream();

   dout.write(postBytes);

   dout.close();

   int ch;

   is = hc.openInputStream();

   while ((ch = is.read()) != -1)
   {
    bos.write(ch);
   }
   res = bos.toByteArray();
  }
  catch(Exception e)
```


(C) Copyright Nokia 2013. All rights reserved.

```
  {
   e.printStackTrace();
  }
  finally
  {
   try
   {
    if(bos != null)
     bos.close();

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

    if(hc != null)
     hc.close();
   }
   catch(Exception e2)
   {
    e2.printStackTrace();
   }
  }
  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";
}

?>
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.