How to use HTTP POST request in Java ME

This article demonstrates how to establish a HTTP connection and uses it to send a POST request to web server.

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import java.io.*;


public class HttpPOST extends MIDlet implements CommandListener {

/** the default value for the url string is * * user can change it to some other urls
within the application* */
private static String defaultURL = "http://localhost";

// GUI component for user to enter web url
private Display myDisplay = null;
private Form mainScreen;
private TextField requestField;

// GUI component for displaying header information
private Form resultScreen;
private StringItem resultField;

// the "send" button used on mainScreen
Command sendCommand = new Command("SEND", Command.OK, 1);
// the "back" button used on resultScreen
Command backCommand = new Command("BACK", Command.OK, 1);

public HttpPOST()
{
 // initializing the GUI components for entering web url
 myDisplay = Display.getDisplay(this);
 mainScreen = new Form("Type in a URL:");
 requestField = new TextField(null, defaultURL, 100, TextField.URL);
 mainScreen.append(requestField);
 mainScreen.addCommand(sendCommand);
 mainScreen.setCommandListener(this);
}

public void startApp() {myDisplay.setCurrent(mainScreen);}

public void pauseApp() {}

public void destroyApp(boolean unconditional) {}

public void commandAction(Command c, Displayable s)
{
 // when user clicks on "send" button on mainScreen
 if (c == sendCommand)
 {
  // retrieving the web url that user enteredString
  String urlstring = requestField.getString();
```


(C) Copyright Nokia 2014. All rights reserved.

```
  // sending a POST request to web serverString
  String resultstring = sendPostRequest(urlstring);

  // displaying the page content retrieved from web server
  resultScreen = new Form("POST Result:");
  resultField = new StringItem(null, resultstring);
  resultScreen.append(resultField);
  resultScreen.addCommand(backCommand);
  resultScreen.setCommandListener(this);
  myDisplay.setCurrent(resultScreen);

 }
 else if (c == backCommand)
 {
  // do it all over again
  requestField.setString(defaultURL);
  myDisplay.setCurrent(mainScreen);
 }
}

// send a POST request to web server
public String sendPostRequest(String urlstring)
{
 HttpConnection hc = null;
 DataInputStream dis = null;
 DataOutputStream dos = null;

 String message = "";

 // specifying the query string
 String requeststring = "request=gettimestamp";
 try
 {
  // openning up http connection with the web server
  // for both read and write access
  hc = (HttpConnection) Connector.open(urlstring, Connector.READ_WRITE);

  // setting the request method to POST
  hc.setRequestMethod(HttpConnection.POST);

  // obtaining output stream for sending query string
  dos = hc.openDataOutputStream();
  byte[] request_body = requeststring.getBytes();

  // sending query string to web server
  for (int i = 0; i < request_body.length; i++)
  {
   dos.writeByte(request_body[i]);
  }
  // flush outdos.flush();

  // obtaining input stream for receiving HTTP response
  dis = new DataInputStream(hc.openInputStream());

  // reading the response from web server character by character
  int ch;
  while ((ch = dis.read()) != -1)
```


(C) Copyright Nokia 2014. All rights reserved.

```
   {
    message = message + (char) ch;
   }

  }
  catch (IOException ioe)
  {
   message = "ERROR";
  }
  finally
  {
   // freeing up i/o streams and http connection
   try
   {
    if (hc != null)
     hc.close();
   }
   catch (IOException ignored)
   {
   }
   try
   {
    if (dis != null)
     dis.close();
   }
   catch (IOException ignored)
   {
   }
   try
   {
    if (dos != null)
     dos.close();
   }
   catch (IOException ignored)
   {
   }
  }
  return message;
 }
}
```


(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.