```
import java.util.*;
import java.io.*;
import java.net.*;
import java.security.*;
import java.text.*;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLSession;

class PaymentGateway {

  protected String server;
  protected String port;
  protected String path;
  protected String username;
  protected String password;

  public PaymentGateway(String user, String pass)
  {

    server = "secure.payscout.com";
    port = "443";
    path = "https://secure.payscout.com/api/transact.php";
    username = user;
    password = pass;

  }

  public HashMap doSale( double amount,
                           String ccNumber,
                           String ccExp
                           ) throws Exception
  {
      HashMap result = new HashMap();
      HashMap request = new HashMap();

      DecimalFormat form = new DecimalFormat("#.00");

      request.put("amount", form.format(amount));
      request.put("type", "sale");
      request.put("ccnumber", ccNumber);
      request.put("ccexp", ccExp);

      String data_out = prepareRequest(request);

      String error = "";
      String data_in = "";
      boolean success = true;
      try {
          HashMap retval = postForm(data_out);
          data_in = (String)retval.get("response");
          result.put("transactionid", retval.get("transactionid"));
      } catch (IOException e) {
          success = false;
          error = "Connect error, " + e.getMessage();
      } catch (Exception e) {
        success = false;
        error = e.getMessage();
    }
    if (!success) {
        throw new Exception(error);
    }

    return result;
}

// Utility Functions

public String prepareRequest(HashMap request) {

    if (request.size() == 0) {
       return "";
    }

    request.put("username", username);
    request.put("password", password);

    Set s = request.keySet();
    Iterator i = s.iterator();
    Object key = i.next();
    StringBuffer buffer = new StringBuffer();



    buffer.append(key).append("=")
          .append(URLEncoder.encode((String) request.get(key)));

    while (i.hasNext()) {
        key = i.next();
        buffer.append("&").append(key).append("=")
              .append(URLEncoder.encode((String) request.get(key)));
    }

    return buffer.toString();

}

protected HashMap postForm(String data) throws Exception {

   HashMap result = new HashMap();

   HttpURLConnection postConn;

   HostnameVerifier hv = new HostnameVerifier() {
      public boolean verify(String urlHostName, SSLSession session) {
          return true;
      }
   };

   HttpsURLConnection.setDefaultHostnameVerifier(hv);

   URL post = new URL("https", server, Integer.parseInt(port), path);
   postConn = (HttpURLConnection)post.openConnection();
```

```
     postConn.setRequestMethod("POST");
     postConn.setDoOutput(true);

     PrintWriter out = new PrintWriter(postConn.getOutputStream());
     out.print(data);
     out.close();

     BufferedReader in =
        new BufferedReader(new InputStreamReader(postConn.getInputStream()));

     String inputLine;
     StringBuffer buffer = new StringBuffer();
     while ((inputLine = in.readLine()) != null) {
        buffer.append(inputLine);
     }
     in.close();


     String response = buffer.toString();

     result.put("response", response);

     // Parse Result
     StringTokenizer st = new StringTokenizer(response, "&");
     while (st.hasMoreTokens()) {
        String varString = st.nextToken();
        StringTokenizer varSt = new StringTokenizer(varString, "=");
        if (varSt.countTokens() > 2 || varSt.countTokens()<1) {
            throw new Exception("Bad variable from processor center: " +
varString);
        }
        if (varSt.countTokens()==1) {
            result.put(varSt.nextToken(), "");
        } else {
            result.put(varSt.nextToken(), varSt.nextToken());
        }
     }

     if (result.get("response")=="") {
        throw new Exception("Bad response from processor center" + response);
     }

     if (!result.get("response").toString().equals("1")) {
        throw new Exception(result.get("responsetext").toString());
     }

     return result;
  }

}

public class TestPaymentGateway
{
    public static void main(String arg[])
    {
        HashMap retval = new HashMap();
        PaymentGateway gw = new PaymentGateway("demo", "password");

        try {
            retval = gw.doSale(10.05, "4111111111111111", "0909");
            System.out.println("Success\nTransId: " +
retval.get("transactionid") + "\n");
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }

    }
}
```