API TRUY VẤN KẾT QUẢ GIAO DỊCH

1. **Dữ liệu đối tác POST sang.**

String cmd // Mã lệnh. Cố định là : TRANS\_INQUIRY

String merchant\_code // Mã đối tác do viettel cấp cho đối tác

String access\_code // Mã truy cập do Viettel cấp cho đối tác

String hash\_key // Khóa bảo mật để mã hoá dữ liệu do Viettel cấp cho đối tác

String order\_id

String version // Giá trị mặc định là: 2.0. Nếu TMĐT là: TMDT

String check sum // Chuỗi mã hoá SHA1 -> encode base64 -> utf8 từ dữ liệu của:

// access\_code + cmd + merchant\_code + order\_id + version

String data = access\_code + cmd + merchant\_code + order\_id + version

check\_sum = Security.hmac(data,hash\_key); // Hàm lấy check\_sum(hmac) được dựng sẵn(Code ở thư mục 0\_LayCheckSum trong file Security.java)

1. **Chuẩn bị các param để thực hiện request tới viettel**

// Cần khai báo thêm 2 dependencies là HttpClient và Fluent HC vào file pom nếu chưa có. Tham khảo thêm tại http://hc.apache.org/httpcomponents-client-ga/quickstart.html

CloseableHttpClient httpclient = HttpClients.createDefault();

HttpPost httpPost = new HttpPost("http://sandbox.viettel.vn/PaymentAPI/webresources/postData");

// địa chỉ sẽ do viettel cung cấp để thực hiện request tới, đây là địa chỉ của máy chủ test của viettel

httpPost.addHeader("Content-Type", "application/x-www-form-urlencoded");

List<NameValuePair> nvps = new ArrayList<NameValuePair>();

/\*Khai báo NameValuePair(nvps) để truyền đủ các param cần thiết(cmd,merchant\_code... Như trong tài liệu kết nối thanh toán đã mô tả) để thực hiện Post request qua viettel\*/

nvps.add(new BasicNameValuePair("cmd", cmd));

nvps.add(new BasicNameValuePair("merchant\_code", merchant\_code));

nvps.add(new BasicNameValuePair("order\_id", order\_id));

nvps.add(new BasicNameValuePair("version", version));

nvps.add(new BasicNameValuePair("check\_sum", check\_sum));

httpPost.setEntity(new UrlEncodedFormEntity(nvps));

CloseableHttpResponse response2 = httpclient.execute(httpPost); //Thực hiện post request và nhận response chuẩn bị lấy dữ liệu viettel trả về

try {

System.out.println(response2.getStatusLine());

HttpEntity entity2 = response2.getEntity(); //Chuyển response thành entity

1. **Lấy dữ liệu từ response body**

String retSrc = EntityUtils.toString(entity2); //Chuyển entity sang String

// parsing JSON

JSONObject result = new JSONObject(retSrc); // Chuyển String sang JSON Object

//Lấy dữ liệu viettel trả về từ json

String merchant\_code\_res = result.get("merchant\_code");

String order\_id\_res = result.get("order\_id");

String error\_code\_res = result.get("error\_code"); // Thành công sẽ trả về dạng 00. Trường hợp khác kiểm tra trong tài liệu

String vt\_transaction\_id\_res = result.get("vt\_transaction\_id"); // Mã giao dịch bên Viettel

String payment\_status\_res = result.get("payment\_status"); // Trạng thái thanh toán:

// -1: Chưa phát sinh giao dịch

// 0: Giao dịch đang chờ sử lý

// 1: Giao dịch thành công

// 2: Giao dịch thất bại

// 3: Giao dịch chưa rõ kết quả

String version\_res = result.get("version"); // Phiên bản kết nối. Giá trị cố định là: 2.0

String check\_sum = result.get("check\_sum");

// Chuỗi mã hoá SHA1 -> encode base64 -> utf8 từ dữ liệu của:

// access\_code + merchant\_code + order\_id + payment\_status + version

// Check sum này có thể dùng để so sánh với checksum bên hệ thống đối tác

//Tạo check sum bên hệ thống đối tác có thể dùng vào so sánh với checksum viettel để xác thực dữ liệu 2 bên

String data\_doitac = access\_code + merchant\_code + order\_id + payment\_status + version;

String check\_sum\_doitc=Security.hmac(data\_doitac,hash\_key);

EntityUtils.consume(entity2);

} catch (Exception e) {

System.out.println(e);

} finally {

response2.close();

}