BÁO CÁO TÌM HIỂU SMPP – SMS GETWAYS

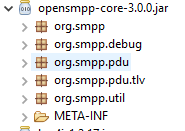
1, Tìm hiểu SMPP:

Các Short Message Peer-to-Peer (SMPP) giao thức là một ngành công nghiệp viễn thông giao thức để trao đổi tin nhắn SMS tin nhắn SMS đồng đẳng giữa các thực thể như các trung tâm dịch vụ nhắn tin ngắn và / hoặc ngoài ngắn Messaging Entities. It is often used to allow third parties (eg value-added service providers like news organizations) to submit messages, often in bulk. Nó thường được dùng để cho phép các bên thứ ba (ví dụ như giá trị gia tăng các nhà cung cấp dịch vụ như các tổ chức tin tức) để gửi tin nhắn, thường ở số lượng lớn.

2, Triển khai smpp getways:

- Hầu hết các thư viện liên quan đến smpp ở C# đều có tính phí.

- Triển khai bằng bộ thư viện mã nguồn mở: openSMPP ở ngôn ngữ java (bao gồm các phiên bản như 3.0, 3.4,… và mới nhất là 5.0)



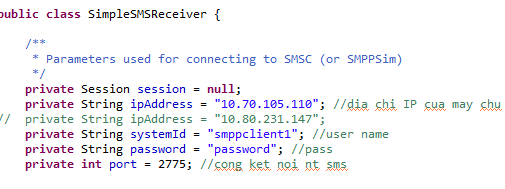
- Cần có một nơi receiver SMS (theo phương thức pdu).

**3, CODE tự làm đơn luồng**:

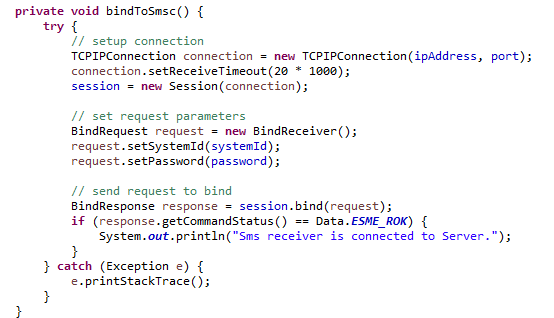
- Tạo 1 server để tiếp nhận, sử dụng thư viện opensmpp để chạy server (server có địa chỉ do mình tạo, ở đây là để địa chỉ localhost)

- Tạo 1 project truy cập đến server để lắng nghe:

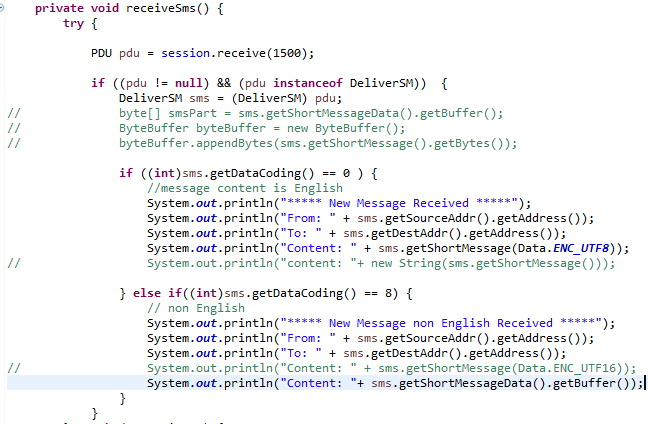
+ khởi tạo các thành phần liên quan:



+ Hàm liên kết với SMSC:



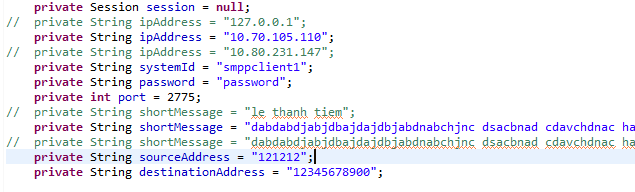
+ Khi đã kết nối được với server thì chỉ cần lắng nghe và tiếp nhận tin nhắn chuyển đến:



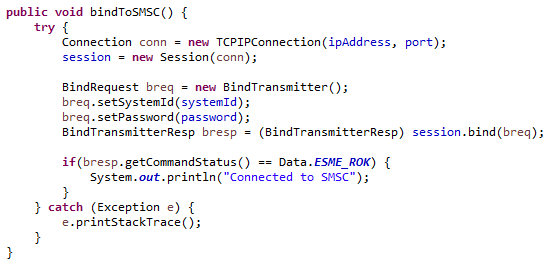
-> Tuy nhiên ở đây mới chỉ tiếp nhận được sortSMS (từ 256 ký tự trở lui), Play\_load sms chưa show lên được (chỉ báo null) ***-> chưa fix***

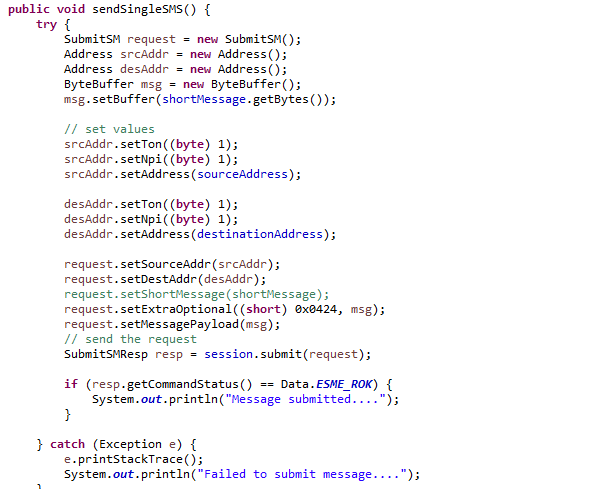
- Tạo 1 project để Test gửi tin SMS đi (SMSTransmiter):

+ khởi tạo các thành phần biến liên quan:



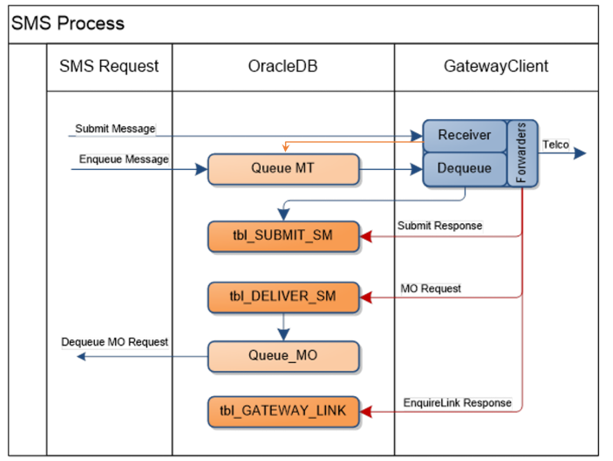
+ Kết nối đến server (bindToSMSC):



 + Gửi tin đến server: có thể gửi được cả sortSMS và play\_loadSMS

4, Đánh giá SOURCE THAM KHẢO – Đa luồng (VNPAY):

\* mô hình của VNPay như hình sau:



Mô hình được viết bằng ngôn ngữ java, sử dụng thư viện mã nguồn mở Opensmpp, lưu trữ thông tin SMS như nội dung, địa chỉ gửi, nhận, ngày tháng gửi,… bằng hệ quản trị cơ sở dữ liệu ( ở đây sử dụng Oracle database để quản lý, nếu cần ta có thể chuyển qua hệ quản trị csdl khác) hoặc qua file log

Chương trình tích hợp hai phương án:

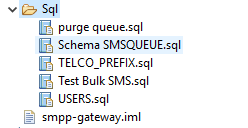
+ Kết nối qua TCP/IP:

* Sử dụng giao thức SMPP V3.4
* Hỗ trợ PDU message: SUBMIT\_SM, SUBMIT\_SM\_RESP
* Các Command Status trả về:
  + 0: Submit thành công, tin nhắn được forwarder tới telco
  + 1: Gateway chưa kết nối telco, tin nhắn bị bỏ qua, kết nối bị ngắt
  + 11: Số điện thoại nhận không đúng format, tin nhắn bị bỏ qua

+ Kết nối qua database ***(chưa tìm hiểu kỹ):***

A, Chuẩn bị cơ sở dữ liệu: (nếu muốn kết nối qua database oracle)

- có các file như sau:



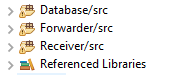
-> tạo các trường các bảng cho csdl,

-> Tạo các đầu số liên quan đến các nhà mạng lớn (Viettel, vinaphone, mobiphone,…)

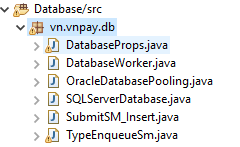
-> Lưu thông tin user

B, Chương trình liên quan:

Chương trình gồm có nơi tiếp nhận(receiver), nơi xử lí database, nơi chuyển tiếp tin nếu cần(forwarder)



A, với database:



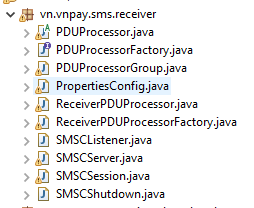
-> Kết nối với csdl với server(nếu kết nối qua database oracle)

-> Tạo các hàm set get để submit và get dữ liệu.(TCP/IP)

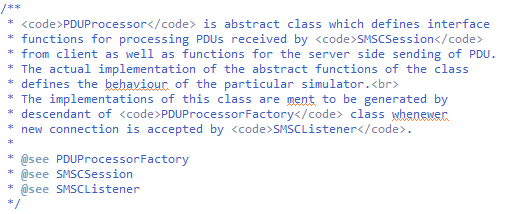
\*=> ***Thực hiện chuyển csdl sang hệ quản trị csdl khác (nhưu postgre chẳng hạn)***

B, Với phía receiver:

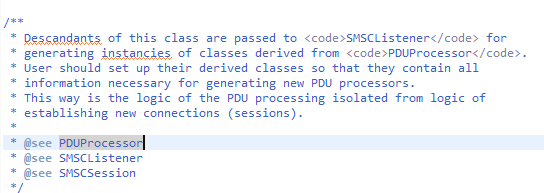
Khởi tạo server:



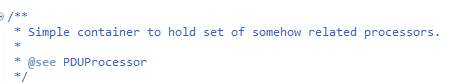
Ở trong code, mỗi class đều có comment cụ thể chức năng của mỗi hàm. Tóm tắt ở đây như sau:

- Lớp PDU\_Processor: 

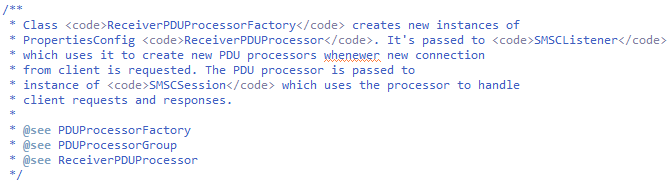
- Lớp PDU\_Processor\_factory:



- Lớp PDUProcessorGroup:

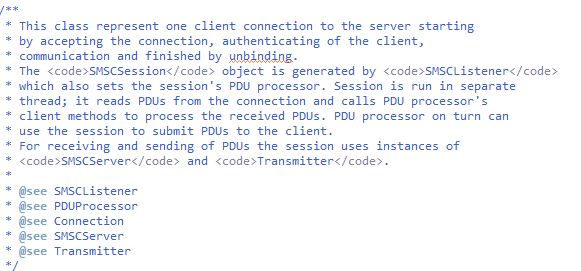


-Lớp receiverPDUProcessorFactory:



-Lớp SMSCServer: Chạy server cho chương trình.

- Lớp SMSCSession:



C, Với forwarder: