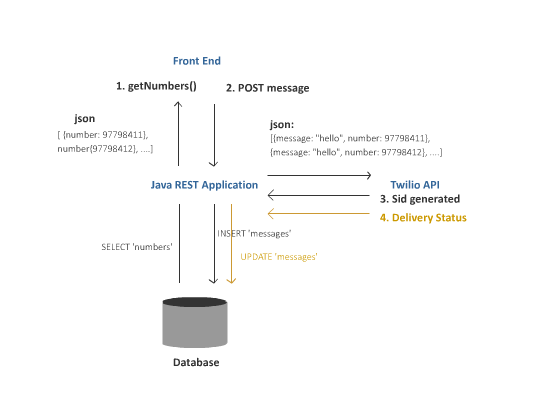
Bulk SMS app

Two tables are created in database named numbers and messages. Mobile numbers are stored in ‘numbers’ while correspoding messages with ssid are stored in ‘messages’ table. A one to many relationship is established between those tables so one number can have many messages identified by unique sid.

Numbers table:

|  |  |
| --- | --- |
| Id | Number |
| 1 | 9779841759703 |
| 2 | 97798410913221 |
| 3 | 977984100000 |
| 4 | 9779841111111 |

Messages table: (foreign key number\_id references Id from Numbers table)

|  |  |  |  |
| --- | --- | --- | --- |
| sid | number\_id | body | status |
| SM30d0ea16d0b744f892b2505b3b74188c | 1 | Hello | null |
| SM6481a37961cb45c78e0b8053e951ac23 | 1 | Hello World | delivered |

A Spring REST API is developed to handle various http requests and connecting to database and API service provided by Twilio.

First when webpage is loaded by user, all numbers are loaded into the client. (Messages and sid are also loaded in the page but this will not be displayed first time as no sid would have been created yet.) Then the user inputs a message through a input field provided. After the user click submit, a JOSN array is created with message and numbers which are posted to Spring REST API. The Spring REST API then makes each request to Twilio API to send sms while promply receiving the corresponding Sid for each number. This information is then stored in the table ‘messages’.

While making request to twilio:

Message message = Message.creator(new PhoneNumber("+" + numbers.getNumber()),

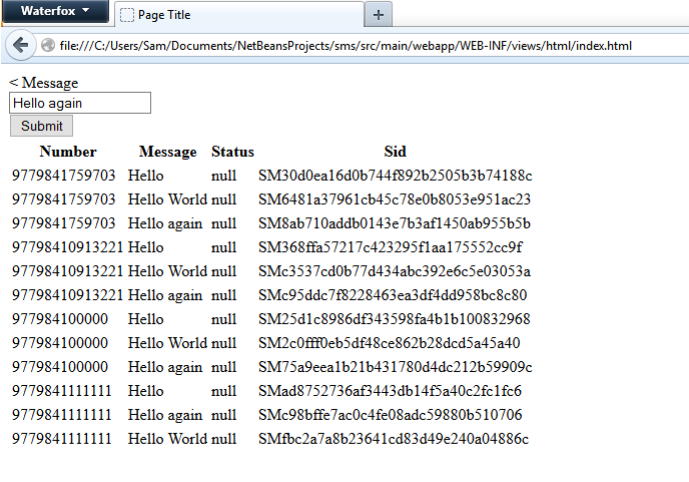
new PhoneNumber("+15005550006"),

smsmessage).setStatusCallback("..our\_server/api/delivery").create();

A parameterized servlet is created to capture webhook created by twilio when delivery happens. The address prepared is ("..our\_server/api/delivery").

The status callback sent by twilio will have parameters MessageSid and MessageStatus. And whenever they are send, we will update the delivery field of ‘messages’ table with corresponding sid.

Sample of application:

After submitting 3 different messages.

Simulating webhook:



After webook simulation:

