message

message/route

Request

HTTP Method – GET

Endpoint / URL - http://base_url/vn-n/ message/route

Content-Type: application/x-www-form-urlencoded

Accept: application/json

Query String Request Parameters

Parameters	Value	Description
from	string	Mobile number of the sender
to	String	Number where to message was sent to
message	String	Sms message received
date	string	Date and time when the message is received.
user-info	String	Any additional information

Example

http://base_url/message/vn-n/route?from=040333333&to=0402222222&message=get appointment 3 &date= yyyy-MM-dd HH:mm:ss

response

Standard HTTP response codes with response body using JSON encoding.

```
{
"name":"route-rule-from" ,
"value":"appointments-by-sms"
"client_id":"myclienid",
"application_id":"myapp_id"
}
```

Response Parameters

Parameters	Value	Description
name	string	Name of the parameter
value	String	Value of the parameter
client_id	String	Sms message received
application_id	string	Date and time when the message is received.

Router Rules

Message routing will be based on all five parameters namely from, to, message, date and user-info.

'From' Rules

Message routing rules related to 'from' field should be defined in a database table called say 'table_route_rules_from'

Proposed Schema of 'table_route_rules_from'

Id (serial , primary key)	From (var char)	Condition (var char)	Response (varchar)
1	Mobile Number	None	appointments-by-sms
2	Mobile number	None	
3			
4			

Condition field may be None or it may provide additional conditions. Additional conditions might require the logic to check 'message'

```
CREATE TABLE table_route_rules_from
(
  id serial NOT NULL,
  client_id character varying,
  application_id character varying,
  from character varying,
  condition character varying,
  response  character varying,
  CONSTRAINT table_route_rules_from_pkey PRIMARY KEY (id )
)
```

'To' Rules

Message routing rules related to 'from' field should be defined in a database table called say 'table_to_route_rules'

Proposed Schema of 'table_route_rules_to'

Id (serial , primary key)	**To (var char)**	**Condition (var char)**	**Response (varchar)**
1	Mobile Number	None	Receive Alerts by SMS
2	Mobile number	None	
3			
4			

Condition field may be None or it may provide additional conditions. Additional conditions might require the logic to check 'message'

```
CREATE TABLE table_route_rules_to
(
  id serial NOT NULL,
  client_id character varying,
  application_id character varying,
  to character varying,
  condition character varying,
  response  character varying,
  CONSTRAINT table_route_rules_to_pkey PRIMARY KEY (id )
)
```

'Message' Rules

Message routing rules should be defined in a database table called say 'table_message_route_rules'

Note: The first word of the Message will be used as the criteria for routing.

Message	Additional Condition	Response
Yes or Ok	From number is in the appointment_table	SMS responses for Appointment reminders
Contains 'get' or 'set'	From number in the verified telephone number table in the database	Get or Set Appointments by SMS
Contains text which matches a field in the 'table_message_route_rules'		Receive Alerts by SMS e.g name of the restaurant
	To number is dedicated to a specific customer and specific application	Routes to the designated application function.

Proposed Schema of 'table_message_route_rules'

Id (serial , primary key)	Message(var char)	Condition (var char)	Response (varchar)
1	Yes	verify-appointment-table	Appointment-response
2	Get	verify-source-verification-table	Appointments-by-sms
3	restaurant name1	none	Receive-alerts
4	restaurant name1	None	Receive-alerts

Note: The sms message received needs to be split using space as the delimiter. Only some basic splitting using java string split or something similar is expected now. Future version of Message Router may use some lexical or syntax analysis library like ANTLR or yacc/lex or perhaps use regular expressions. It is not considered for this version.

http://www.antlr.org/

```
CREATE TABLE table_route_rules_message
(
  id serial NOT NULL,
  client_id character varying,
  application_id character varying,
  message character varying,
  condition character varying,
  response  character varying,
  CONSTRAINT table_route_rules_to_pkey PRIMARY KEY (id )
)
```

Example

See SMS receive operation.