SerwerSMS.pl®
System Obsługi Marketingu Mobilnego

SQL API Documentation

Introduction

The SQL API service consists in configuring a connection to the database on the SerwerSMS or Client side and creating appropriate tables for sending and receiving messages. The message is sent on the principle of adding a record to the appropriate table in the database. In the case of receiving a message, SerwerSMS will add a new record to the relevant table when receiving the message assigned to the selected Customer.
Connection configuration can take place in two ways:
1. On the SerwerSMS side, i.e. the system creates a separate access to a part of the database and the client operates on the database and tables located on the SerwerSMS platform.
2. On the client's side, ie the client creates a table structure in his own database on his server, configures the settings in the SerwerSams client panel and the SerwerSMS system connects cyclically to the customer's database and checks whether, for example, there are any new records to process. If so, it will download and forward it for processing.
Available options:
- dispatch of messages ECO SMS, SMS FULL VOICE
- update delivery reports
- receiving messages SMS ECO+, ND, SC, NDI, SCI

To send SMS ECO+, the sender field should be left empty. For news VOICE, the field should be set to "VOICE".
If you have more than one active SQL API configuration within your main account or user account, incoming messages will be included in the table for each configuration.

Error codes

After the message has been delivered, the delivery report should be updated for selected messages (it is also possible that the message will be sent and delivered, however, for various reasons we will not receive a delivery report back then the report in the SQL API will not be updated). In the event of an error related to shipping (unsent) or delivery (undelivered), the system may return, in addition to the delivery report "-1", an additional code o and a description of the problem.

Code	Description
2000	validity of a text message has expired
2001	an incorrect recipient number
2002	unsupported number
2003	message rejected
2004	operator error
2005	received calls
2200	Invalid recipient's number
2201	Limit messages exhausted
2202	Dispatch of messages is locked
2203	Invalid message
2204	Error on the side of the Operator
2205	Invalid sender name
2206	Number is located on the black list
2207	Dispatch to foreign networks is blocked
2208	No permitted to send

Creating a configuration

In order for the service to work properly, first you need to create a configuration in the Customer Panel in the tab "Interface settings -> SQL API". The form for adding a new configuration contains the following fields:

Field	Description
nazwa	connection name (descriptive)
aktywne	connection status, or tables to be checked
typ	local / remote (or local database and tables on the side SerwerSMS, or remote database and tables on the Client)
silnik bazy	eg. MySQL (or other, depending on availability)
host	host to connect
numer portu	port under which the service is available
login	login to the database
hasło	password to the database
baza	database name
struktura tabel	STANDARD / INDIVIDUAL (STANDARD - structure proposed by SerwerSMS, INDIVIDUAL - the structure of self-imposed by the client)

For individual table structure sending messages (MT)

table name	table name for dispatch of messages
field id	field with the identifier record
field numer	field with phone number
field nazwa nadawcy	field with sender name
field wiadomość	field with the content of the message
field data	field with possibly. the date of shipment. At the time of transfer messages for execution, set the current date and time)
field smsid	field with the identifier message (SerwerSMS)
field raport doręczenia	field with the status messages ("-1" - unposted, 0 - the default value 1 - delivered, 2 - undelivered)
field data doręczenia	field of the date of report update
field kod raportu doręczenia	field of the numerical code description delivery report (for more information, see Error Codes)
field opis raportu doręczenia	field with a text description of a delivery report (for more information, see Error Codes)
field flagi	field with optional flags, eg. UTF FLASH
field części	field with the length of the message (how many parts it consists SMS)

For individual table structure, message reception (MO)

table name	table name to receive messages
field numer	number from which the SMS
field nadawca	number to which the SMS was received (eg. NDI)
field wiadomość	the content of the received message
field data	date received messages
field typ	the type of the received message (ECO, ND, NDI)

From the level of the Client Panel, it is possible to test the correctness of the connection by clicking on the test icon. The system will check the connection and display information whether access to the database is possible, whether there are relevant fields in the table and if there are any records in the table for sending messages.
It is also possible to download a configuration containing full information about a given connection (access to the database, names of tables and fields, examples of connection and sending messages).

Configuring the Client

The customer can create a database or share a part of the existing database. In the case of tables, the Customer may use the structure proposed by SerwerSMS (the code for creating the tables below) or, if necessary, configure access to the database and table by filling in the Customer Panel fields related to the connection to the database, table name and structure. The IP address from which the connection will be made in order to download messages for sending, updating reports and adding answers is: 94.152.131.145.

For a table with messages to send, it is necessary to specify fields:

- table name
- record id
- number
- sender
- message
- smsid
- delivery report
- delivery report time
- delivery report code
- description of the delivery report
- flags
- parts

Default table structure for sending messages:

```
CREATE TABLE IF NOT EXISTS `SerwerSMS_MT` (
  `id` int(10) unsigned NOT NULL,
  `number` varchar(20) COLLATE utf8_polish_ci NOT NULL,
  `sender` varchar(20) COLLATE utf8_polish_ci NOT NULL,
  `message` text COLLATE utf8_polish_ci NOT NULL,
  `date` datetime NOT NULL,
  `flags` varchar(50) COLLATE utf8_polish_ci NOT NULL,
  `smsid` varchar(20) COLLATE utf8_polish_ci NOT NULL,
  `dlr` enum('-1','0','1','2') COLLATE utf8_polish_ci NOT NULL DEFAULT '0',
  `dlr_date` datetime NOT NULL,
  `dlr_code` int(10) unsigned NOT NULL,
  `dlr_description` varchar(100) COLLATE utf8_polish_ci NOT NULL,
  `parts` tinyint(1)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8 COLLATE=utf8_polish_ci;
ALTER TABLE `SerwerSMS_MT`  ADD PRIMARY KEY (`id`),  ADD KEY `smsid` (`smsid`),  ADD KEY `new` (`smsid`,`date`);
ALTER TABLE `SerwerSMS_MT`  MODIFY `id` int(10) unsigned NOT NULL AUTO_INCREMENT;
```

Default table structure for message reception (optional):

```
CREATE TABLE IF NOT EXISTS `SerwerSMS_MO` (
  `id` int(10) unsigned NOT NULL,
  `number` varchar(20) COLLATE utf8_polish_ci NOT NULL,
  `sender` varchar(20) COLLATE utf8_polish_ci NOT NULL,
  `message` text COLLATE utf8_polish_ci NOT NULL,
  `date` datetime NOT NULL,
  `type` enum('ECO','ND','NDI') COLLATE utf8_polish_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_polish_ci;
ALTER TABLE `SerwerSMS_MO` ADD PRIMARY KEY (`id`);
ALTER TABLE `SerwerSMS_MO` MODIFY `id` int(10) unsigned NOT NULL AUTO_INCREMENT;
```

Table structure for MSSQL for sending messages:

```
CREATE TABLE SerwerSMS_MT(
  id int PRIMARY KEY NOT NULL IDENTITY(1,1),
  number varchar(20) NOT NULL,
  sender varchar(20) NOT NULL,
  message text NOT NULL,
  date datetime NOT NULL,
  flags varchar(50) NOT NULL,
  smsid varchar(20) NOT NULL,
  dlr VARCHAR(2) NOT NULL CHECK (dlr IN('-1', '0', '1', '2')) DEFAULT '0',
  dlr_date datetime NOT NULL,
  dlr_code int NOT NULL,
  dlr_description varchar(100) NOT NULL,
  parts tinyint
)
```

Struktura tabeli dla MSSQL do odbioru wiadomości (optional):

```
CREATE TABLE SerwerSMS_MO (
  id int PRIMARY KEY NOT NULL IDENTITY(1,1),
  number varchar(20) NOT NULL,
  sender varchar(20) NOT NULL,
  message text NOT NULL,
  date datetime NOT NULL,
  type VARCHAR(5) NOT NULL CHECK (type IN('ECO+','ND','NDI')) DEFAULT '0',
)
```

Configuration side SerwerSMS

To configure access to the SQL API side SerwerSMS, you must create a new configuration by selecting the field called "Type" to "LOCAL". Saving these settings will create two tables in the database SerwerSMS and the user who will have access to these tables (INSERT and SELECT). The following describes the structure of the table for the dispatch of messages and received messages (SMS replies and incoming).

The structure of the table for dispatch of messages:

Filed	Description
id	record ID
number	the recipient's number (preferably a full international format 48500600700)
sender	the name of the sender previously credited to the customer's account. If a message ECO, the name should remain empty. For voice messages should be set to "VOICE".
message	message encoding ASCII or UTF-8
date	dispatch date in ISO format "YYYY-MM-DD HH: ii: ss". For immediate dispatch blank or "0000-00-00 00:00:00"
flags	additional attributes for example. FLASH - message class zero, the SPEED - a high-priority message sent fastest channels, UTF - SMS message sent to FULL coding UCS2 allows, among others, to use Polish diacritics.
smsid	message ID (assigned by SerwerSMS). When is writing a new record value should be empty or NULL
dlr	delivery report (update SerwerSMS. 0 - waiting for the update, 1 - delivered, 2 - undelivered, 3 - is not sent)
dlr_date	updated delivery report (update SerwerSMS, is writing the new record value should be empty or NULL)
dlr_code	numeric code delivery report (update SerwerSMS. The codes are described in the section Error codes)
dlr_description	descriptions on delivery report (update SerwerSMS. The codes are described in the section Error codes)
parts	The number of parts that make up the message (update SerwerSMS, is writing the new record value should be empty or NULL)

The structure of the table to receive messages:

Field	Description
id	record ID
number	the recipient's number (the number on which the message was sent)
sender	number (number from which the message was sent)
message	message encoding ASCII or UTF-8
date	date received in ISO format "YYYY-MM-DD HH: ii: ss".
type	type of message (reply ECO, ND / SC, NDI / SCI)

Data archivization

For configuration maintained on the side SerwerSMS, data (messages sent and received) will be archived after six months. In the case of the configuration of the Client, depending on the amount of data stored in tables, and server performance also we recommend archiving from time to time in order to maintain high productivity by mail records and update them.