



HOSPITAL INFORMATION SYSTEM

Short SMS module specification

1. About

HIS needs to send and receive SMS messages to / from its clients. So called SMS server is used for that purpose.

HIS sends HTTP requests to SMS server's REST API.

Content type – application/json.

There are several types of requests:

- a) Send SMS to client
- b) Get incoming SMS from clients
- c) Get hospital phone numbers

2. Authorization

HIS uses MS Azure Active Directory access tokens for authorization. Tokens are sent as an authorization key in HTTP request's header.

Key	Value
Authorization	Bearer <i>SomeAccessToken</i>

Authorization : Bearer SomeAccessToken



3. Send SMS

HTTP POST <https://SMSServer/api/messages>.

Request's body:

```
{
  "Id": null,
  "From": "37000000001",
  "To": "37000000002",
  "Text": "Reminding, that N. SURNAME is registered on 2019-11-12 (Tu) 07:00, in 321 cab. Will You come?",
  "ExternalKey": 27342
},
```

where

- Id – SMS server unique SMS message identifier
- From – hospital phone number (must be assigned by SMS provider for response messages)
- To – person phone number
- Text – SMS message text
- ExternalKey – HIS unique SMS message identifier

Response:

```
{
  "Data": {
    "Id": 623178,
    "CreatedAt": "2019-11-11T08:08:21.8072824+02:00"
  },
  "Status": {
    "StatusCode": "OK",
    "ErrorCode": "OK",
    "ErrorMessage": null
  }
}
```

where

- Data:
 - Id – SMS server unique SMS message identifier
 - CreatedAt – SMS message received in SMS server
- Status:
 - StatusCode – response HttpStatusCode (like OK, Bad Request, Internal Server Error, etc.)
 - ErrorCode – error code from enum (see below)
 - ErrorMessage – error text

ErrorCode:

OK = 1,
Internal = 500,
InvalidMessageOwner = 600,
InvalidClientNumber = 700

4. Get SMS

HIS can receive SMS messages from SMS server. Now this method is used, when HIS needs to get all incoming messages from the clients (used only type Incoming).

HTTP GET <https://SMSServer/api/messages?type=Incoming&dateFrom=2019-11-12 11:01:26.8912252>

Parameters:

- type (Outgoing = 1, Incoming = 2)
- dateFrom – SMS message change date not less than stated

Response:

```
{
  "Data": {
    "Messages": [
      {
        "Id": 531977,
        "From": "37000000002",
        "To": "37000000001",
        "Type": "Incoming",
        "CreatedAt": "2019-11-12T11:01:51.2143795",
        "UpdatedAt": "2019-11-12T11:01:51.2143907",
        "Text": "Yes"
      },
      {
        "Id": 531978,
        "From": "37000000003",
        "To": "37000000001",
        "Type": "Incoming",
        "CreatedAt": "2019-11-12T11:01:55.0139225",
        "UpdatedAt": "2019-11-12T11:01:55.0139335",
        "Text": "No"
      }
    ]
  },
  "Status": {
    "StatusCode": "OK",
    "ErrorCode": "OK",
    "ErrorMessage": null
  }
}
```

where

- Data.Messages[i]:
 - Id – SMS server unique SMS message identifier
 - From – person phone number
 - To – hospital phone number
 - Type (Outgoing = 1, Incoming = 2)
 - CreatedAt – SMS message created in SMS server
 - Updated – SMS message updated in SMS server
 - Text – SMS message text
- Status:
 - StatusCode – response HttpStatusCode (like OK, Bad Request, Internal Server Error, etc.)
 - ErrorCode – error code from enum (see below)
 - ErrorMessage – error text



5. Get hospital phone numbers

HIS needs to know which phone numbers or sender names can be used for hospital. Phone number and valid sender names must be assigned by SMS service provider.

HTTP GET <https://SMSServer/api/clients/operatorNumbers>

Response:

```
{
  "Data": {
    "ClientOperatorNumbers": [
      {
        "OperatorNumberId": 4,
        "Number": "37000000000"
      },
      {
        "OperatorNumberId": 7,
        "Number": "RSL_HOSP"
      }
    ]
  },
  "Status": {
    "StatusCode": "OK",
    "ErrorCode": "OK",
    "ErrorMessage": null
  }
}
```

where

- Data. ClientOperatorNumbers [i]:
 - OperatorNumberId – SMS server unique SMS number identifier
 - Number – hospital phone number or valid sender name
- Status:
 - StatusCode – response HttpStatusCode (like OK, Bad Request, Internal Server Error, etc.)
 - ErrorCode – error code from enum (see below)
 - ErrorMessage – error text