

# HTTP API

## Обзор API

Предоставляемый API сервис отправки SMS-сообщений позволяет осуществить:

- Аутентификацию
- Отправку SMS-сообщения на один номер без учета часового пояса получателя
- Отправку SMS-сообщения на один номер с учетом часового пояса получателя
- Отправку SMS-сообщения на несколько номеров без учета часового пояса получателя
- Получение статуса отправленного SMS-сообщения
- Получение входящих SMS-сообщений за период
- Получение статистики по SMS-рассылкам
- Получение баланса авторизованного Пользователя

API Сервиса отправки SMS-сообщений организован в соответствии с принципами REST, что позволяет обмениваться HTTPS-запросами с URL-encoded кодировкой. HTTPS - это обычный HTTP, работающий через шифрованные транспортные механизмы SSL и TLS. Это позволяет обеспечить защиту от атак, основанных на прослушивании сетевого соединения (снифферских атак и атак типа man-in-the-middle), при условии использования шифрующих средств и подтвержденной надежности сертификата сервера.

Запрос к API состоит из следующих элементов:

- основной URL запроса: <https://integrationapi.net/rest/>
- ресурс (например: /Sms/SendByTimeZone)
- параметры GET или POST-запроса (в кодировке UTF-8)

[Скачать пример решений](#)

## Аутентификация

Сервис создает идентификатор сессии в Системе после прохождения аутентификации данных, передаваемых сервису в GET-запросе следующего формата:

```
https://integrationapi.net/rest/user/sessionid?login=<Login>&password=<Password>
```

Параметры запроса представляют собой последовательность пар вида **{имя параметра}={значение параметра}**, разделенных символом амперсанда (&). Content-Type для параметров запроса: application/x-www-form-urlencoded Это формат для кодирования пар «ключ-значение» с возможностью дублирования ключей. Каждая пара ключ-значение отделяется символом «&», ключ отделен от значения символом «=». При этом пробелы должны заменяться на знак «+», а затем, используя URL-кодирование, могут быть заменены на буквенно-цифровые символы. Например:

```
login: Jonathan password: a+b==13!
```

Должно быть закодировано как:

```
login=Jonathan&password=a%2Bb%3D%3D13%25!
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/user/sessionid?login=test&password=11111
```

В Табл. 1 приводится полный список параметров GET-запроса.

Табл. 1. Параметры GET-запроса для аутентификации

Параметр	Тип данных	Описание
Обязательные параметры		
Login	String	Логин, полученный при регистрации
Password	String	Пароль, соответствующий логину

В случае успешной аутентификации предоставленных данных от сервиса отправки SMS-сообщений будет получен ответ со следующими параметрами:

```
HTTP status code: 200 OK (статус «OperationComplete»);
Cache-Control: private (указание на то, что ответ разрешается сохранять только в закрытом кэше, т.е. только для текущего Пользователя);
Connection: Keep-Alive (наименование заголовка соединения, которое не надо обновлять в кэше);
Content-Type: application/json; charset=utf-8 (фактически значение вернется в виде строки в кавычках (не в виде JSON) и кодировке utf-8);
"Идентификатор сессии (GUID)"
```

Ниже приведен пример ответа:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
"Z5CYSZEKDL1DPICU37WENQVOYKP0T1GSLHX1"
```

В случае возникновения исключительной ситуации в ходе обработки запроса или ошибки аутентификации Сервис возвращает Код ошибки (см. Табл. 10) в виде JSON следующего формата:

```
{
  Code: <Код ошибки>,
  Desc: <"Текст ошибки">
}
```

Например, при ошибке авторизации:

```
{
  Code: 4,
  Desc: "Invalid user login or password"
}
```

**Полученный идентификатор сессии действителен в течение 120 минут.**

## Получение баланса авторизованного пользователя

Протокол HTTP не имеет состояний. Это означает, что веб-сервер обрабатывает каждый HTTP-запрос со стороны внешнего приложения или сайта независимо и не сохраняет значения переменных, использованных в предшествующих запросах. Поэтому при выполнении запроса на получение баланса пользователя также необходимо передавать данные, полученные при авторизации этого пользователя. Сервис возвращает значение баланса авторизованного пользователя в соответствии со значениями параметров, передаваемых сервису в GET-запросе следующего формата:

```
https://integrationapi.net/rest/User/Balance?SessionID=<Идентификатор сессии>
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/User/Balance?SessionID=Z5CYSZEKDL1DPICU37WENQV0YKP0T1GSLHX1
```

В Табл. 2 приводится полный список параметров запроса.

Табл. 2. Параметры GET-запроса баланса

Параметр	Тип данных	Описание
Обязательные параметры		
SessionID	String	Идентификатор сессии, полученный при аутентификации

Сервис проверяет валидность полученного SessionID (проверяет актуальность и наличие в Системе). В случае успеха сервис авторизует пользователя и в ответе передает баланс пользователя со следующими параметрами:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
<Баланс пользователя>
```

Ниже приведен пример ответа:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
```

В случае возникновения исключительной ситуации при обработке запроса или ошибки аутентификации сервис возвращает код ошибки (см. Табл. 10) в виде JSON следующего формата:

```
{
  Code: <Код ошибки>,
  Desc: <"Текст ошибки">
}
```

Например, при ошибке аутентификации идентификатора сессии:

```
{
  Code: 4,
  Desc: "SessionID expired"
}
```

## Отправка SMS-сообщений

### Отправка SMS-сообщения на один номер без учета часового пояса получателя

Сервис инициирует отправку SMS-сообщения в соответствии со значениями параметров, передаваемых сервису в POST-запросе следующего формата:

```
https://integrationapi.net/rest/Sms/Send?SessionID=<Идентификатор сессии>&SourceAddress=<Адрес отправителя>& DestinationAddress=<Номер получателя>&Data=<Текст сообщения>&Validity=<Время жизни сообщения>
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/Sms/Send?SessionId=C619DF83829F4C3094CB54F4D62878786B5B&DestinationAddress=79161002030&SourceAddress=DEVIN0&Data=test&Validity=0
```

В Табл. 3 приводится полный список параметров запроса.

Табл. 3. Параметры запроса на отправку SMS-сообщения

Параметр	Тип данных	Описание
Обязательные параметры		
SessionID	String	Идентификатор сессии, полученный при аутентификации (36 символов)
DestinationAddress	String	Номер получателя сообщения, в международном формате: код страны и код сети плюс номер телефона. Пример:  79031234567; +79031234567;
Data	String	Текст сообщения, не более 2000 символов
SourceAddress	String	Адрес отправителя, не более 11 латинских символов или 15 цифр
Необязательные параметры		
SendDate	DateTime	Дата и время отправки (пример 2011-01-28T16:00:00). Если в запросе передается этот параметр, то сообщение будет отправлено только при наступлении полученных даты и времени без учета текущего часового пояса получателя. Сообщение отправится при наступлении переданного времени в часовом поясе: GMT+03:00. Если не требуется отложенная отправка, то передавать данный параметр не нужно.
Validity	Int	Время жизни сообщения (в минутах)

Перед отправкой SMS-сервис выполняет проверку запроса:

- наличие обязательных параметров;
- валидность сессии Пользователя (аутентификацию и определение, не истекло ли его время жизни SessionID);
- баланс пользователя на отправку SMS (достаточность средств на балансе определяется тарифом текущего пользователя на отправку SMS для мобильного оператора указанного в запросе номера);
- валидность указанного в запросе номера;
- валидность адреса отправителя;
- длина сообщения.

Если все проверки пройдены успешно, сервис отправляет сообщение в SMS-центр и возвращает идентификатор отправленного сообщения с параметрами:

Формат ответа:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
<Идентификатор сообщения>
```

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
["579700854169272358"]
```

В случае превышения длины отправляемого сообщения (70 символов на кириллице или 160 символов на латинице) сервис возвращает ответ в виде последовательности идентификаторов сообщений. Например:

```
["579700854169272358", "579700854169272359"]
```

В случае непрохождения других проверок сервис возвращает код ошибки (см. Табл. 10) в виде JSON следующего формата:

```
{
  Code: <Код ошибки>,
  Desc: <"Текст ошибки">
}
```

Например:

```
{
  Code: 6,
  Desc: "Invalid source address"
}
```

## Отправка SMS-сообщения на один номер с учетом часового пояса получателя:

Сервис инициирует отправку SMS-сообщения в соответствии со значениями параметров, передаваемых сервису в POST-запросе следующего формата:

```
https://integrationapi.net/rest/Sms/SendByTimeZone?SessionID=<Идентификатор сессии>&SourceAddress=<Адрес отправителя>&DestinationAddress=<Номер получателя>&Data=<Текст сообщения>&Validity=<Время жизни сообщения>&SendDate=<Дата отправки сообщения>
```

Ниже приведен пример запроса:

https://integrationapi.net/rest/Sms/SendByTimeZone?SessionId=Z5CYSZEKDL1DPICU37WEHQV0YKP0T1GSLHX1&SourceAddress=TESTSMS&DestinationAddress=79001234567&Data=testdata&Validity=10&destinationAddress=79160000000&data=testdata&sendDate=2011-01-28T16:00:00&validity=10

В Табл. 4 приводится полный список параметров запроса.

Табл. 4. Параметры POST-запроса на отправку SMS-сообщения с учетом часового пояса

Параметр	Тип данных	Описание
<b>Обязательные параметры</b>		
SessionID	String	Идентификатор сессии, полученный при аутентификации (36 символов)
DestinationAddress	String	Номер получателя сообщения, в международном формате: код страны и код сети плюс номер телефона. Пример:  79031234567; +79031234567; +79031234567.
Data	String	Текст сообщения (не более 2000 символов)
SourceAddress	String	Адрес отправителя (не более 11 латинских символов или 15 цифр)
SendDate	DateTime	Дата и время отправки (пример 2011-01-28T16:00:00). Если в запросе передается этот параметр, то сообщение будет отправлено только при наступлении полученных даты и времени с учетом текущего часового пояса получателя. Если не требуется отложенная отправка, то передавать данный параметр не нужно.
<b>Необязательные параметры</b>		
Validity	Int	Время жизни сообщения (в минутах)

Перед отправкой SMS-сервис выполняет проверку запроса:

- наличие обязательных параметров;
- валидность сессии Пользователя (аутентификацию и определение, не истекло ли его время жизни SessionID);

- баланс пользователя на отправку SMS (достаточность средств на балансе определяется тарифом текущего пользователя на отправку SMS для мобильного оператора указанного в запросе номера);
- валидность указанного в запросе номера;
- валидность адреса отправителя;
- длина сообщения.

Если все проверки пройдены успешно, сервис отправляет сообщение в SMS-центр и возвращает идентификатор отправленного сообщения с параметрами:  
Формат ответа:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
<Идентификатор сообщения>
```

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
["579700854169272359"]
```

В случае превышения длины отправляемого сообщения (70 символов на кириллице или 160 символов на латинице) сервис возвращает ответ в виде последовательности идентификаторов сообщений. Например:

```
["579700854169272358", "579700854169272359"]
```

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
["579700854169272358", "579700854169272359"]
```

В случае непрохождения других проверок сервис возвращает код ошибки (см. Табл. 10) в виде JSON следующего формата:

```
{
  Code: <Код ошибки>,
  Desc: <"Текст ошибки">
}
```

Например:

```
{
  Code: 6,
  Desc: "Invalid source address"
```

}

## Отправка SMS-сообщения на несколько номеров без учета часового пояса получателя:

Сервис инициирует отправку SMS-сообщения на несколько номеров в соответствии со значениями параметров, передаваемых сервису в POST-запросе следующего формата:

Отправка SMS-сообщения на несколько номеров без учета часового пояса получателя: Сервис инициирует отправку SMS-сообщения на несколько номеров в соответствии со значениями параметров, передаваемых сервису в POST-запросе следующего формата:

```
https://integrationapi.net/rest/Sms/SendBulk?SessionID=<Идентификатор сессии>&SourceAddress=<Адрес отправителя>&DestinationAddresses=<Номер(а) получателя>&Data=<Текст сообщения>&Validity=<Время жизни сообщения>
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/Sms/SendBulk?SessionID=Z5CYSZEKDL1DPICU37WENQV0YKP0T1GSLHX1&SourceAddress=TESTSMS&DestinationAddresses=79001234567&Data=testdata&Validity=10&DestinationAddresses=79160000000&data=testdata&sendDate=2011-01-28T16:00:00&validity=10
```

В Табл. 5 приводится полный список параметров запроса.

Табл. 5. Параметры POST-запроса на отправку SMS-сообщения на несколько номеров

Параметр	Тип данных	Описание
Обязательные параметры		
SessionID	String	Идентификатор сессии, полученный при аутентификации (36 символов)
DestinationAddresses	String	Номер получателя сообщения, в международном формате: код страны и код сети плюс номер телефона. Пример:  +79031234567; +79031234567; +79031234567.
Data	String	Текст сообщения (не более 2000 символов)
SourceAddress	String	Адрес отправителя (не более 11 латинских символов или 15 цифр)
Необязательные параметры		

Параметр	Тип данных	Описание
Validity	Int	Время жизни сообщения (в минутах)
SendDate	DateTime	Дата и время отправки (пример 2010-0601T19:14:00). Если не требуется отложенная отправка, то передавать данный параметр не нужно.

Перед отправкой SMS-сервис выполняет проверку запроса:

- наличие обязательных параметров;
- валидность сессии пользователя (аутентификацию и определение, не истекло ли его время жизни SessionID);
- баланс пользователя на отправку SMS (достаточность средств на балансе определяется тарифом текущего пользователя на отправку SMS для мобильного оператора указанного в запросе номера);
- валидность указанного в запросе номера;
- валидность адреса отправителя;
- длина сообщения.

Если все проверки пройдены успешно, сервис отправляет сообщение в SMS-центр и возвращает идентификатор отправленного сообщения с параметрами:

Формат ответа:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
<Идентификатор сообщения>
```

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
["579700854169272358"]
```

В случае превышения длины отправляемого сообщения (70 символов на кириллице или 160 символов на латинице) сервис возвращает ответ в виде последовательности идентификаторов сообщений. Для нескольких сообщений идентификаторы сегментов будут расположены последовательно – сначала последовательно все сегменты одного сообщения, затем – все сегменты другого. Например:

```
["579700854169272358", "579700854169272359", "579700854169272360", "579700854169272361"]
```

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
["579700854169272358", "579700854169272359", "579700854169272360", "579700854169272361"]
```

В случае непрохождения других проверок сервис возвращает код ошибки (см. Табл. 10) в виде JSON следующего формата:

```
{
  Code: <Код ошибки>,
  Desc: <"Текст ошибки">
}
```

Например:

```
{
  Code: 6,
  Desc: "Invalid source address"
}
```

**Внимание!** Возможность отправки sms на несколько номеров с учетом часового пояса получателя пока недоступна.

## Получение статуса отправленного SMS-сообщения

Сервис возвращает статус отправленного sms-сообщения в соответствии со значениями параметров, передаваемых сервису в GET-запросе следующего формата:

```
https://integrationapi.net/rest/Sms/State?sessionId=<Идентификатор сессии>&messageId=<Идентификатор сообщения>
```

Ниже приведен пример запроса для односегментного сообщения (длина которого не превышает 70 символов на кириллице или 160 символов на латинице):

```
https://integrationapi.net/rest/Sms/State?sessionId=Z5CYSZEKDL1DPICU37WEHQVOYKP0T1GSLHX1&messageId=579700854169272358
```

Для многосегментных сообщений (длина сообщения превышает 70 символов на кириллице и 160 на латинице) запрос должен формироваться для каждого сегмента сообщения, например:

```
https://integrationapi.net/rest/Sms/State?sessionId=1AED345F65DD4C27BD37A17970C427FAE991&messageId=579700854169272358
```

Табл. 6. Параметры GET-запроса статуса отправленного сообщения (сегмента сообщения)

Параметр	Тип данных	Описание
SessionID	String	Идентификатор сессии, полученный при аутентификации (36 символов)
DestinationAddress	String	Идентификатор сообщения (сегмента сообщения). Для одного запроса будет выполнен возврат статуса только одного сообщения (сегмента сообщения).

После получения запроса сервис проверяет валидность идентификатора сессии и наличие отправленного сообщения (сегмента сообщения) с присланным идентификатором. Если все проверки пройдены успешно, то сервис вернет статус отправленного sms-сообщения в json-формате со следующими параметрами:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
{"State":<Код статуса сообщения>,
"CreationDateUtc":<Дата создания>,
"SubmittedDateUtc":<Дата отправки сообщения>,
"ReportedDateUtc":<Дата доставки сообщения>,
"TimeStampUtc": "<Дата и время получения отчета>",
"StateDescription": "<Описание статуса>",
"Price":<Стоимость>}
```

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
{"State":255,"CreationDateUtc":null,"SubmittedDateUtc":null,"ReportedDateUtc":null,"TimeStampUtc":"\\/Date(-621359680000)\\/","StateDescription":"Неизвестный","Price":null}
```

Если какая-либо проверка неуспешна, сервис возвращает код ошибки (см. Табл. 10) в виде JSON следующего формата:

```
{
  Code: <Код ошибки>,
  Desc: <"Текст ошибки">
}
```

Например:

```
{
  Code: 1,
  Desc: "MessageID can not be null or empty Parameter name: messageId"
}
```

Табл. 7. Параметры ответа на запрос статуса сообщения

Наименование поля	Описание
State	Статус сообщения (см. Табл. 11)
TimeStampUtc	Дата и время получения отчета (Гринвич GMT00:00)
StateDescription	Описание статуса
CreationDateUtc	Дата создания
SubmittedDateUtc	Дата отправки
ReportedDateUtc	Дата доставки
Price	Цена за сообщение

## Получение SMS-сообщений за период

Сервис возвращает входящие sms-сообщения за период в соответствии со значениями параметров, передаваемых сервису в GET-запросе следующего формата:

<https://integrationapi.net/rest/Sms/In?sessionId=<Идентификатор сессии>&minDateUTC=<Дата и время начала периода>&maxDateUTC=<Дата и время окончания периода>>

Ниже приведен пример запроса:

<https://integrationapi.net/rest/Sms/In?sessionId=Z5CYSZEKDL1DPICU37WENQVOYKP0T1GSLHX1&minDateUTC=2011-01-01T00:00:00&maxDateUTC=2011-01-11T00:00:00>

Табл. 8. Параметры GET-запроса на получение сообщений за период

Параметр	Тип данных	Описание
Session ID	String	Идентификатор сессии, полученный при аутентификации (36 символов)
maxDateUTC	DateTime	Дата и время окончания периода, за который происходит выборка входящих сообщений (например, 2010-06-02T19:14:00).
Необязательные параметры		
minDateUTC	DateTime	Дата и время начала периода, за который происходит выборка входящих сообщений (например, 2010-06-01T19:14:00).

После получения запроса сервис проверит валидность идентификатора сессии и даты-времени начала и окончания периода присланным идентификатором. Если

все проверки пройдены успешно, то сервис вернет перечень сообщений и их параметров за период в json-файла следующего формата:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
[{"Data":<Текст сообщения>,
"SourceAddress":<Адрес отправителя>,
"DestinationAddress":<Номер получателя>,
"ID":<Идентификатор сообщения>,
"CreateDateUtc":<Дата создания>}]
```

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
[{"Data":"test1",
"SourceAddress":"7926000000",
"DestinationAddress":"7916000000",
"ID":539187174,
"CreateDateUtc":"\\/Date(1294045911213)\\/"},
{"Data":"test2",
"SourceAddress":"79260000001",
"DestinationAddress":"7916000000",
"ID":539187214,
"CreateDateUtc":"\\/Date(1294045911353)\\/"}]
```

Если какая-либо проверка неуспешна, сервис возвращает код ошибки (см. Табл. 10) в виде JSON следующего формата:

```
{
  Code: <Код ошибки>,
  Desc: <"Текст ошибки">
}
```

Например:

```
{
Code: 9,
Desc: "The parameters dictionary contains a null entry for parameter
'maxDateUtc' of non-nullable type 'DateTime' for method
'System.Web.Mvc.ActionResult In(System.String, DateTime, DateTime)' in
'RestService.Controllers.SmsController'. An optional parameter must be a reference type, a
nullable type, or be declared as an optional parameter. Parameter name: parameters"
}
```

## Получение статистики по SMS-рассылкам

Сервис возвращает статистику по SMS-рассылкам за период в соответствии со значениями параметров, передаваемых сервису в GET-запросе следующего формата:

```
https://integrationapi.net/rest/Sms/Statistics?sessionId=FBHKZT9TBBTUWYUR1PYUTYRAGRLUUG0R8A8Z&
startDateTime=2012-01-18%2000:00:00&endDateTime=2012-01-18%2023:59:00
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/Sms/Statistics?sessionId=FBHKZT9TBBTUWYUR1PYUTYRAGRLUUG0R8A8Z&startDateTime=2012-01-18%2000:00:00&endDateTime=2012-0118%2023:59:00
```

Табл. 9. Параметры GET-запроса на формирование статистики за период

Параметр	Тип данных	Описание
Обязательные параметры		
SessionID	String	Идентификатор сессии (36 символов)
startDateTime	DateTime	Дата и время начала периода, за который необходимо получить статистику, например 2012-01-18%2000:00:00.
endDateTime	DateTime	Дата и время конца периода, за который необходимо получить статистику, например 2012-01-18%2023:59:00.
Необязательные параметры		

После получения запроса сервис проверяет валидность присланного идентификатора сессии, корректность дат начала/окончания формирования статистики, диапазон дат (не более 3 месяцев). Если все проверки пройдены успешно, сервис возвращает статистику по sms-сообщениям в json-формате со следующими параметрами:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
{"Sent":<Отправлено>,
"Delivered":<Доставлено>,
"Errors":<С ошибками>,
"InProcess":<В процессе>,
"Expired":<С истекшим сроком доставки>,
"Rejected":<Отмененные>,
"Total":<Всего>,
"TotalWithErrors":<Всего с ошибками>,
"DeliveryRatio":<Успешно доставлено>}
```

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
{"Sent":9,
"Delivered":0,
"Errors":0,
"InProcess":7780,
"Expired":0,
"Rejected":56876,
"Total":64665,
"TotalWithErrors":64665,
```

```
"DeliveryRatio":0}
```

Если какая-нибудь проверка не проходит успешно, то сервис возвращает код ошибки (см. Табл. 10) в виде JSON следующего формата:

```
{  
  Code: <Код ошибки>,  
  Desc: <"Текст ошибки">  
}
```

Например:

```
{  
Code: 2,  
Desc: "Нельзя указывать диапазон дат более 90 дней."  
}
```

## Коды ошибок и статусы сообщений

Табл. 10. Коды ошибок

REST error code	HTTP status code	Описание
	200	Operation complete
1	400	Argument cannot be null or empty
2	400	Invalid argument
3	400	Invalid session id
4	401	Unauthorized access
5	403	Not enough credits
6	400	Invalid operation
7	403	Forbidden
8	500	Gateway error
9	500	Internal server error

Табл. 11. Статусы сообщений

State	Описание
-1	Отправлено (передано в мобильную сеть)
-2	В очереди
47	Удалено

<b>State</b>	<b>Описание</b>
-98	Остановлено
0	Доставлено абоненту
10	Неверно введен адрес отправителя
11	Неверно введен адрес получателя
41	Недопустимый адрес получателя
42	Отклонено смс-центром
46	Просрочено (истек срок жизни сообщения)
48	Отклонено Платформой
69	Отклонено
99	Неизвестный
255	статус: *сообщение еще не успело попасть в БД, *сообщение старше 48 часов.