

Руководство разработчика

СЕРВИС ОТПРАВКИ SMS СООБЩЕНИЙ

История документа

| Версия | Дата | Комментарии |
|--------|------------|------------------------------------|
| 0.1 | 01.10.2013 | Первая версия документа. |
| 0.2 | 02.10.2013 | Обновленная версия документа |
| 1.0 | 03.10.2013 | Первая полная версия документа |
| 1.1 | 04.10.2013 | Актуализированная версия документа |

Оглавление

| | |
|--|----|
| 1. ОБЩИЕ СВЕДЕНИЯ..... | 4 |
| 2. НАЗНАЧЕНИЕ ДОКУМЕНТА | 4 |
| 3. ОБЗОР API..... | 4 |
| 4. АУТЕНТИФИКАЦИЯ | 5 |
| 5. ПОЛУЧЕНИЕ БАЛАНСА АВТОРИЗОВАННОГО ПОЛЬЗОВАТЕЛЯ | 6 |
| 6. ОТПРАВКА SMS-СООБЩЕНИЯ..... | 7 |
| 6.1. Отправку SMS-сообщения на один номер без учета часового пояса получателя | 7 |
| 6.2. Отправку SMS-сообщения на один номер с учетом часового пояса получателя | 9 |
| 6.3. Отправку SMS-сообщения на несколько номеров без учета часового пояса получателя | 12 |
| 7. ПОЛУЧЕНИЕ СТАТУСА ОТПРАВЛЕННОГО SMS-СООБЩЕНИЯ..... | 14 |
| 8. ПОЛУЧЕНИЕ SMS-СООБЩЕНИЙ ЗА ПЕРИОД | 16 |
| 9. ПОЛУЧЕНИЕ СТАТИСТИКИ ПО SMS-РАССЫЛКАМ..... | 17 |
| ПРИЛОЖЕНИЕ. КОДЫ ОШИБОК И СТАТУСЫ СООБЩЕНИЙ..... | 19 |
| 9.1. Коды ошибок..... | 19 |
| 9.2. Статусы сообщений..... | 19 |

1. ОБЩИЕ СВЕДЕНИЯ

Сервис отправки SMS сообщений – это сервис, предоставляемый Клиентам, для реализации массовых рассылок сообщений на мобильные телефоны по технологии SMS.

Для начала использования API:

1. Зарегистрируйтесь в Личном кабинете;
2. Заключите договор;
3. Ознакомьтесь с текущей документацией.

2. НАЗНАЧЕНИЕ ДОКУМЕНТА

Руководство содержит описание API Сервиса отправки SMS сообщений.

Оно включает описание схемы работы с API, краткий обзор всех параметров URL, а также подробные описания по каждому из параметров, снабженные реальными примерами.

Документ предназначен для разработчиков, которые хотят добавить возможность взаимодействия с Сервисом отправки SMS сообщений на страницы своих сайтов или в свои приложения.

3. ОБЗОР API

Предоставляемый API Сервиса отправки SMS сообщений позволяет осуществить:

- [Аутентификацию;](#)
- [Получение баланса авторизованного Пользователя;](#)
- [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 SMS-сообщений за период;](#)
- [Получение статистики по SMS-рассылкам.](#)

API Сервиса отправки SMS сообщений организовано в соответствии с принципами REST, что позволяет обмениваться HTTPS URL–encoded запросами.

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, что будут использоваться шифрующие средства и сертификат сервера проверен и ему доверяют.

Запрос к API состоит из следующих элементов:

- Основного URL запроса:

```
https://integrationapi.net/rest/
```

- Ресурса, например:

```
/Sms/SendByTimeZone
```

- Параметров GET или POST-запроса (в кодировке UTF-8).

4. 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=<Логин>
&password=<Пароль>
```

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 все не буквенно-цифровые символы.

Например,

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

Должно быть закодировано как

```
login=Jonathan+Do&password=a+%2B+b+%3D%3D+13%25%21
```

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 успешного прохождения аутентификации присланных данных Сервис отправки SMS сообщений пришлет 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 этого 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, а 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
"Z5CYSZEKDL1DPICU37WEHQVOYKP0T1GSLHX1"
```

В случае возникновения исключительной ситуации во время 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.

5. 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, а сервер не сохраняет данные о значениях переменных, использованных в предшествующих запросах. Поэтому данные, полученные при авторизации Пользователя, должны быть переданы и при осуществлении запроса получения баланса авторизованного 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=Z5CYSZEKDL1DPICU37W
EHQVOYKP0T1GSLHX1
```

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) и в случае успеха авторизует Пользователя и в ответе присылает 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
20015.3
```

В случае возникновения исключительной ситуации во время 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 аутентификации идентификатора сессии:

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

6. ОТПРАВКА SMS-СООБЩЕНИЯ

6.1. Отправку 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=Z5CYSZEKDL1DPICU37WEHQV
OYKP0T1GSLHX1&SourceAddress=TESTSMS&DestinationAddress=79001234567&Data=te
stdata&Validity=10
```

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 | <p>Номер получателя сообщения, в международном формате: код страны + код сети + номер телефона.</p> <p>Пример:</p> <ul style="list-style-type: none"> • 79031234567; • +79031234567; • 89031234567. |
| Data | String | Текст сообщения, сообщение не должно быть длиннее 2000 символов |
| SourceAddress | String | Адрес отправителя сообщения. До 11 латинских символов или до 15 цифровых. |
| Необязательные параметры | | |
| SendDate | DateTime | <p>Дата и время отправки (пример 2010-06-01T19:14:00).</p> <p>Если в запросе передается этот параметр, то сообщение будет отправлено только при наступлении полученных даты и времени без учета текущего часового пояса получателя.</p> <p>Сообщение отправится при наступлении переданного времени в часовом поясе: GMT+04:00.</p> <p>Если не требуется отложенная отправка, то передавать данный параметр не нужно.</p> |
| Validity | Int | Время жизни сообщения (в минутах) |

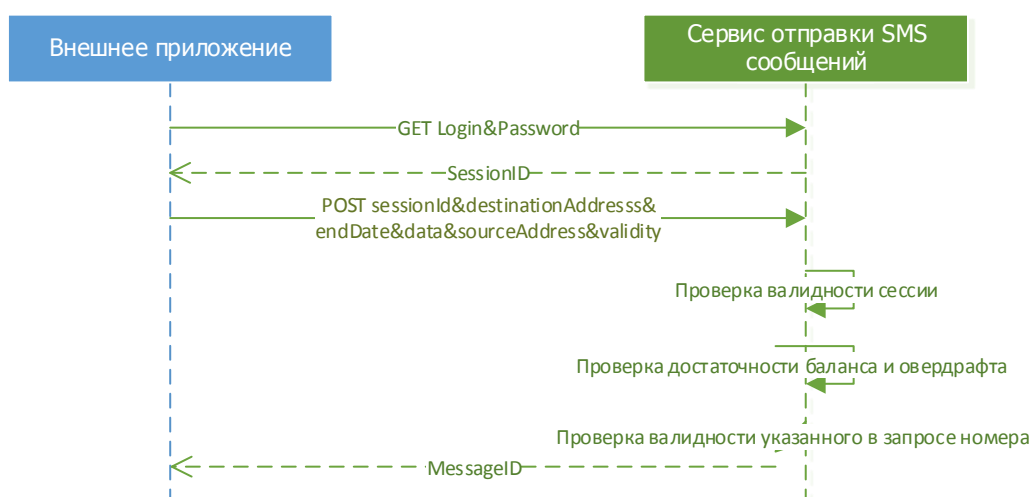


Рис. 1. Отправка SMS через Сервис отправки SMS

Перед отправкой SMS Сервис проверяет запрос на:

- Наличие обязательных параметров;
- Валидность сессии Пользователя (аутентификацию и определение, не истекло ли его время жизни SessionID);
- Достаточно ли Баланса Пользователя на отправку SMS. (Достаточность определяется на основании тарифа Пользователя на отправку SMS для мобильного оператора указанного в запросе номера);
- Валидность указанного в запросе номера;
- Валидность адреса отправителя;
- Длину сообщения.

Если все проверки пройдены успешно, то Сервис отправит сообщение в SMS-центр и вернет идентификатор отправленного сообщения со следующими 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
["GW0261BBD6B3"]
```

В случаях, когда длина отправляемого сообщения превышает 70 символов на кириллице или 160 символов на латинице, ответ от сервиса будет в виде последовательности идентификаторов сообщений, например:

```
["SAR-GW01+79160000000-5f3b1972-2-1", "SAR-GW01+79160000000-5f3b1972-2-2"]
```

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: 6
  Desc: "Invalid source address"
}
```

6.2. Отправку 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/SendByTymeZone?
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/SendByTymeZone?
sessionId=Z5CYSZEKDL1DPICU37WEHQVOYKP0T1GSLHX1&
sourceAddress=TESTSMS
&destinationAddress= 791600000000&
data=testdata&
sendDate=2011-01-28T16:00:00&
validity=10

```

В Табл. 3 приводится полный список параметров запроса.

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

| 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 + код сети + номер телефона. Пример: <ul style="list-style-type: none"> • 79031234567; • +79031234567; • 89031234567. |
| Data | String | Текст сообщения, сообщение не должно быть длиннее 2000 символов |
| SourceAddress | String | Адрес отправителя сообщения. До 11 латинских символов или до 15 цифровых. |
| SendDate | DateTime | Дата и время отправки (пример 2010-06-01T19:14:00). 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. |
| Необязательные параметры | | |
| Validity | Int | Время жизни сообщения (в минутах) |

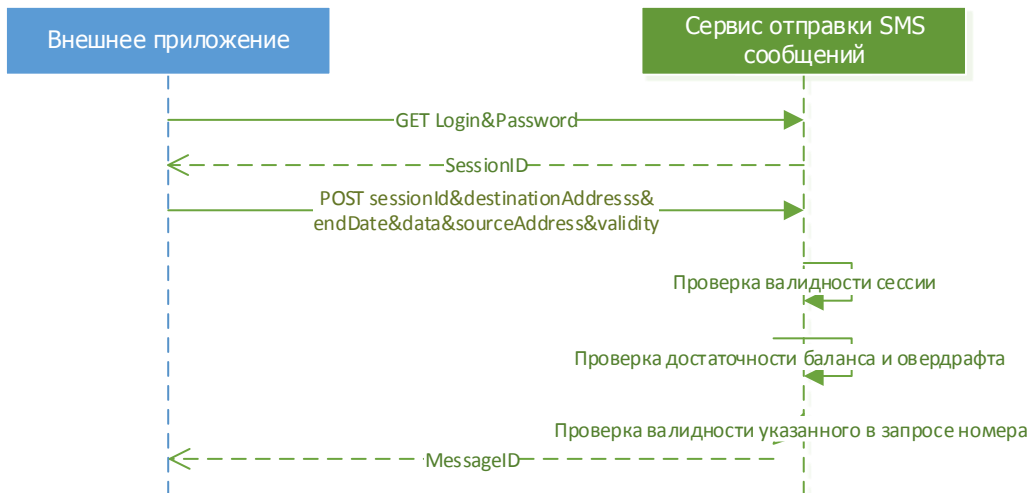


Рис. 2. Отправка SMS через Сервис отправки SMS с учетом часового пояса

Перед отправкой SMS Сервис проверяет запрос на:

- Наличие обязательных параметров;
- Валидность сессии Пользователя (аутентификацию и определение, не истекло ли его время жизни SessionID);
- Достаточно ли Баланса Пользователя на отправку SMS. (Достаточность определяется на основании тарифа Пользователя на отправку SMS для мобильного оператора указанного в запросе номера);
- Валидность указанного в запросе номера;
- Валидность адреса отправителя;
- Длину сообщения.

Если все проверки пройдены успешно, то Сервис отправит сообщение в SMS-центр и вернет идентификатор отправленного сообщения со следующими 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
["GW0261BBD6B3"]
  
```

В случаях, когда длина отправляемого сообщения превышает 70 символов на кириллице или 160 символов на латинице, ответ от сервиса будет в виде последовательности идентификаторов сообщений:

```
["SAR-GW01+79160000000-5f3b1972-2-1", "SAR-GW01+79160000000-5f3b1972-2-2"]
```

Например:

```

HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
  
```

```
Content-Type: application/json; charset=utf-8
["SAR-GW01+79160000000-5f3b1972-2-1", "SAR-GW01+79160000000-5f3b1972-2-2"]
```

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: 6
  Desc: "Invalid source address"
}
```

6.3. Отправку 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=Z5CYSZEKDL1DPICU37W
EHQVOYKP0T1GSLHX1&SourceAddress=TESTSMS&DestinationAddresses=79001234567&
estinationAddresses=79059999999&Data=testdata&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 | Номера получателей сообщения, в международном формате: код страны + код сети + номер телефона. Пример: <ul style="list-style-type: none">• 79031234567;• +79031234567;• 89031234567. |
| Data | String | Текст сообщения, сообщение не должно быть длиннее 2000 символов |

| | | |
|--------------------------|----------|--|
| SourceAddress | String | Адрес отправителя сообщения. До 11 латинских символов или до 15 цифровых. |
| Необязательные параметры | | |
| SendDate | DateTime | Дата и время отправки (пример 2010-06-01T19:14:00). Если не требуется отложенная отправка, то передавать данный параметр не нужно. |
| Validity | Int | Время жизни сообщения (в минутах) |

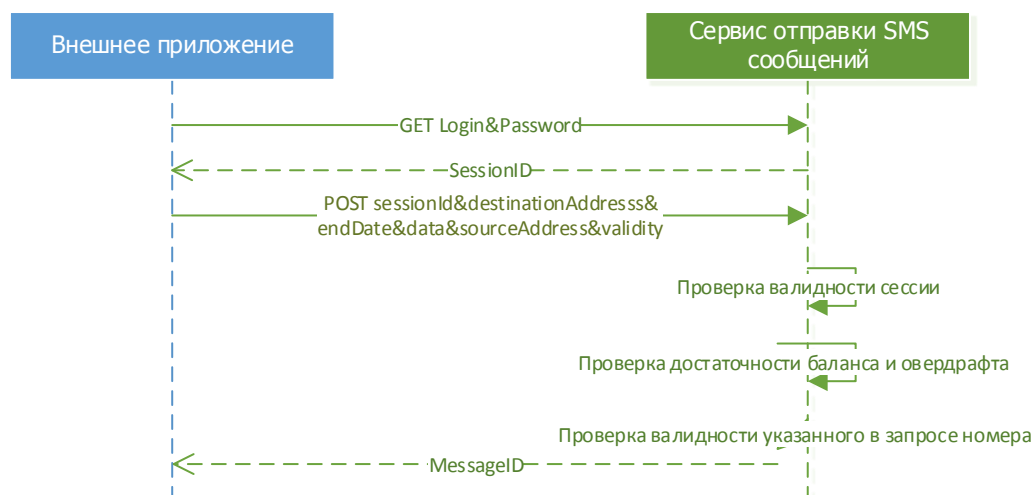


Рис. 3. Отправка SMS на несколько номеров через Сервис отправки SMS

Перед отправкой SMS Сервис проверяет запрос на:

- Наличие обязательных параметров;
- Валидность сессии Пользователя (аутентификацию и определение, не истекло ли его время жизни SessionID);
- Достаточно ли Баланса Пользователя на отправку SMS. (Достаточность определяется на основании тарифа Пользователя на отправку SMS для мобильного оператора указанного в запросе номера);
- Валидность указанных в запросе номеров (если хоть один номер не проходит валидацию, то сообщения не отправляются);
- Валидность адреса отправителя;
- Длину сообщения.

Если все проверки пройдены успешно, то Сервис отправит сообщение в SMS-центр и вернет идентификатор отправленного сообщения со следующими 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
["GW0261BBD6B3"]
```

В случаях, когда длина отправляемого сообщения превышает 70 символов на кириллице или 160 символов на латинице, ответ от сервиса будет в виде последовательно расположенных идентификаторов сегментов сообщения. Для нескольких сообщений идентификаторы сегментов будут расположены последовательно – сначала последовательно все сегменты одного сообщения, затем – все сегменты другого, например:

```
["SAR-GW01+79160000000-5f3b1972-2-1", "SAR-GW01+79160000000-5f3b1972-2-2",
["SAR-GW01+79053500000-5d3b1972-2-1", "SAR-GW01+79053500000-5d3b1972-2-2"]
```

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
["SAR-GW01+79160000000-5f3b1972-2-1", "SAR-GW01+79160000000-5f3b1972-2-2",
["SAR-GW01+79053500000-5f3d1972-2-1", "SAR-GW01+79053500000-5f3d1972-2-2"]
```

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: 6
  Desc: "Invalid source address"
}
```

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

7. 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=Z5CYSZEKDL1DPICU37WENQVOYKP0T1GSLHX1
&messageId=GW0261BA732A
```

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

```
https://integrationapi.net/rest/Sms/State?
sessionId= 1AED345F65DD4C27BD37A17970C427FAE991
&messageID=SAR-W+84333377-f71d25b0-2-1
```

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

| Параметры | Тип данных | Описание |
|--------------|------------|---|
| Обязательные | | |
| sessionId | String | Идентификатор сессии (36 символов). |
| messageId | String | Идентификатор сообщения (сегмента сообщения). Для одного запроса будет выполнен возврат статуса только одного сообщения (сегмента сообщения). |

После получения запроса сервис проверит 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,"ReportedDateU
tc":null,"TimeStampUtc":"\\/Date(-
62135596800000)\\/","StateDescription":"Неизвестный","Price":null}
```

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: 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 |

8. 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

| Параметры | Тип данных | Описание |
|----------------|------------|---|
| Обязательные | | |
| sessionId | String | Идентификатор сессии (36 символов). Обязательный. |
| maxDateUTC | DateTime | Дата и время окончания периода, за который происходит выборка входящих сообщений (например, 2010-06-02T19:14:00). |
| Необязательные | | |
| 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>,
"CreatedDateUtc":<Дата создания>}]
```

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
[{"Data":"test1",
"SourceAddress":"79260000000",
"DestinationAddress":"79160000000",
"ID":539187174,
"CreatedDateUtc":"\\/Date(1294045911213)\\/"},
{"Data":"test2",
"SourceAddress":"79260000001",
"DestinationAddress":"79160000000",
"ID":539187214,
"CreatedDateUtc":"\\/Date(1294045911353)\\/"}]
```

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: 9
  Desc: "The parameters dictionary contains a null entry for parameter
'endDateUtc' 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"
}
```

9. ПОЛУЧЕНИЕ СТАТИСТИКИ ПО SMS-РАССЫЛКАМ

Сервис возвращает входящие 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=FBHKZT9TBBTUWYUR1
PYUTYRAGRLUUG0R8A8Z&
startDateTime=2012-01-18%2000:00:00&
endDateTime=2012-01-18%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 проверит валидность присланного идентификатора сессии и дат начала/окончания формирования статистики (включая ограничение на то, что охватываемый диапазон должен не превышать 3 месяцев).

Если все проверки пройдены успешно, то сервис вернет 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."
}
```

ПРИЛОЖЕНИЕ. КОДЫ ОШИБОК И СТАТУСЫ СООБЩЕНИЙ

9.1. Коды ошибок

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 |

9.2. Статусы сообщений

Табл. 11. Статусы сообщений

| State | Описание |
|-------|--|
| -1 | Отправлено (передано в мобильную сеть) |

| | |
|-----|--|
| -2 | В очереди |
| 47 | Удалено |
| -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 | По запросу возвращается этот статус, если сообщения еще не успело попасть в БД, либо сообщение старше 48 часов |