

HTTP API Без SessionID

Обзор API

Предоставляемый API сервис отправки SMS-сообщений позволяет осуществить:

- 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
- Отправку SMS-сообщения на несколько номеров с учётом часового пояса получателя
- Получение статуса отправленного SMS-сообщения
- Получение SMS-сообщений за период
- Получение статистики по SMS-рассылкам
- Отправку Viber-сообщения на один номер без учета часового пояса получателя
- Отправку Viber-сообщения на несколько номеров без учета часового пояса получателя
- Отправка Viber-сообщения на один номер с переотправкой по SMS
- Отправка Viber-сообщения на несколько номеров с переотправкой по SMS
- Получение статуса отправленного Viber-сообщения

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/v2>
- Ресурса, например: /Sms/SendByTimeZone
- Параметров GET или POST-запроса (в кодировке UTF-8)

Получение 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/v2/User/Balance?Login=<Логин>&Password=<Пароль>
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/v2/User/Balance?Login=test_login&Password=test123
```

Табл. 1. Параметры GET-запроса баланса

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

Сервис проверяет валидность Логина/Пароля 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. 16) в виде JSON следующего формата:

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

Например, при ошибке авторизации:

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

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/v2/Sms/Send?Login=<Логин>&Password=<Пароль>&SourceAddress=<Адрес отправителя>&DestinationAddress=<Номер получателя>&Data=<Текст сообщения>&Validity=<Время жизни сообщения>
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/v2/Sms/Send?Login=test_login&Password=test_password&DestinationAddress=79161002030&SourceAddress=DEVINO&Data=test&Validity=0
```

Табл. 2. Параметры запроса на отправку SMS-сообщения

Параметр	Тип данных	Описание
Обязательные параметры		
Login	String	Логин, полученный при регистрации
Password	String	Пароль, соответствующий логину
DestinationAddress	String	Номер получателя сообщения, в международном формате: код страны код сети + номер телефона. Пример: 79031234567; +79031234567; 89031234567

Параметр	Тип данных	Описание
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 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+00:00. Если не требуется отложенная отправка, то передавать данный параметр не нужно.
Validity	Int	Время жизни сообщения (в минутах)

Перед отправкой SMS сервис проверяет запрос на:

- Наличие обязательных параметров;
- Валидность Логина/пароля;
- 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. 16) 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/v2/Sms/SendByTimeZone?Login=<Логин>&Password=<Пароль>&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/v2/Sms/SendByTimeZone?Login=test_login&Password=test123&SourceAddress=TESTS  
MS&DestinationAddress=79001234567&Data=testdata&Validity=10&sendDate=2011-01-28T16:00:00
```

Табл. 3. Параметры POST-запроса на отправку SMS-сообщения с учетом часового пояса

Параметр	Тип данных	Описание
Обязательные параметры		
Login	String	Логин, полученный при регистрации
Password	String	Пароль, соответствующий логину
DestinationAddress	String	Номер получателя сообщения, в международном формате: код страны и код сети плюс номер телефона. Пример: 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 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. Вемя указывается в UTC.
Необязательные параметры		
Validity	Int	Время жизни сообщения (в минутах)

Перед отправкой SMS сервис проверяет запрос на:

- Наличие обязательных параметров;
- Валидность Логина/пароля;
- 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+7916000000-5f3b1972-2-1", "SAR-GW01+7916000000-5f3b1972-2-2"]
```

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
["SAR-GW01+7916000000-5f3b1972-2-1", "SAR-GW01+7916000000-5f3b1972-2-2"]
```

Если какая-нибудь проверка не проходит успешно, то сервис возвращает код ошибки (см. Табл. 16) 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:

```
https://integrationapi.net/rest/v2/Sms/SendBulk?Login=<Логин>&Password=<Пароль>&SourceAddress=<Адрес отправителя>&DestinationAddresses=<Номер(а) получателя(ей)>&Data=<Текст сообщения>&Validity=<Время жизни сообщения>
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/v2/Sms/SendBulk?Login=test_login&Password=test123&SourceAddress=TESTSMS&DestinationAddresses=79001234567&DestinationAddresses=79059999999&Data=testdata&Validity=10
```

Табл. 4. Параметры POST-запроса на отправку SMS-сообщения на несколько номеров

Параметр	Тип данных	Описание
Обязательные параметры		
Login	String	Логин, полученный при регистрации
Password	String	Пароль, соответствующий логину
DestinationAddress	String	Номер получателя сообщения, в международном формате: код страны и код сети плюс номер телефона. Максимальное количество получателей сообщения не должно превышать 2999. Пример: 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		
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. Вемя указывается в UTC.

Перед отправкой SMS Сервис проверяет запрос на:

- Наличие обязательных параметров;
- Валидность Логина/пароля;
- 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 сообщения в SMS-центр и вернет идентификаторы отправленных сообщений 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+7916000000-5f3b1972-2-1","SAR-GW01+7916000000-5f3b1972-2-2",
- ["SAR-GW01+7905350000-5d3b1972-2-1","SAR-GW01+7905350000-5d3b1972-2-2]

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
["SAR-GW01+7916000000-5f3b1972-2-1","SAR-GW01+7916000000-5f3b1972-2-2",
["SAR-GW01+7905350000-5d3d1972-2-1","SAR-GW01+7905350000-5d3d1972-2-2]
```

Если какая-нибудь проверка не проходит успешно, то сервис возвращает код ошибки (см. Табл. 16) в виде JSON следующего формата:

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

Например:

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

Отправка SMS-сообщения на несколько номеров 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/v2/Sms/SendByTimeZoneToAddresses?Login=<Логин>&Password=<Пароль>&SourceAddress=<Адрес отправителя>&DestinationAddresses=<Номер(а) получателя(ей)>&Data=<Текст сообщения>&Validity=<Время жизни сообщения>&SendDate=<Дата отправки сообщения>
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/v2/Sms/SendByTimeZoneToAddresses?Login=test_login&Password=test123&SourceAddress=TESTSMS&&DestinationAddresses=79001234567&DestinationAddresses=79059999999&Data=testdata&Validity=10&sendDate=2011-01-28T16:00:00
```

Табл. 5. Параметры POST-запроса на отправку SMS-сообщения с учетом часового пояса

Параметр	Тип данных	Описание
Обязательные параметры		
Login	String	Логин, полученный при регистрации
Password	String	Пароль, соответствующий логину
DestinationAddresses	String	Номер получателя сообщения, в международном формате: код страны и код сети плюс номер телефона. Пример: 79031234567; +79031234567; 89031234567.
Data	String	Текст сообщения (не более 2000 символов)

Параметр	Тип данных	Описание
SourceAddress	String	Адрес отправителя (не более 11 латинских символов или 15 цифр)
SendDate	DateTime	Дата и время отправки (пример 2010-0601T19:14:00) в UTC. Если в запросе передается этот параметр, то сообщение будет отправлено только при наступлении полученных даты и времени с учетом текущего часового пояса получателя. Время указывается в UTC.
Необязательные параметры		
Validity	Int	Время жизни сообщения (в минутах)

Перед отправкой SMS-сервис выполняет проверку запроса:

- наличие обязательных параметров;
- валидность логина/пароля;
- 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 номеров;
- валидность адреса отправителя;
- длина сообщения.

Если все проверки пройдены успешно, сервис отправляет сообщения в SMS-центр и возвращает идентификаторы отправленных сообщений 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 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+7916000000-5f3b1972-2-1", "SAR-GW01+7916000000-5f3b1972-2-2",
  "SAR-GW01+7905350000-5d3b1972-2-1", "SAR-GW01+7905350000-5d3b1972-2-2" ]
```

Например:

```
HTTP/1.1 200 OK
Cache-Control: private
Connection: Keep-Alive
Content-Type: application/json; charset=utf-8
["SAR-GW01+7916000000-5f3b1972-2-1", "SAR-GW01+7916000000-5f3b1972-2-2",
  "SAR-GW01+7905350000-5f3d1972-2-1", "SAR-GW01+7905350000-5f3d1972-2-2" ]
```

В случае непрохождения других проверок сервис возвращает код ошибки (см. Табл. 16) в виде JSON следующего формата:

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



```
}
```

Например:

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

Получение статуса отправленного SMS-сообщения

Внимание

Внимание! В случае, если сообщение было отправлено более 48 часов назад, то статус сообщения будет "255". (см. Табл. 18. Статусы SMS)

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

```
https://integrationapi.net/rest/v2/Sms/State?  
Login=<Логин>&  
Password=<Пароль>&  
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/v2/Sms/State?Login=test_login&Password=test123&messageId=GW0261BA732
```

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/v2/Sms/State?Login=test_login&Password=test123&messageID=SAR-W+84333
```

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

Параметр	Тип данных	Описание
Login	String	Логин, полученный при регистрации
Password	String	Пароль, соответствующий логину
messageld	String	Идентификатор сообщения (сегмента сообщения). Для одного запроса будет выполнен возврат статуса только одного сообщения (сегмента сообщения).

После получения запроса сервис проверит валидность логина/пароля 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(-
621355968000000)\\/","StateDescription":"Неизвестный","Price":null}

```

Если какая-нибудь проверка не проходит успешно, то сервис возвращает код ошибки (см. Табл. 16) 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. 17)
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/v2/Sms/In?
Login=<Логин>&
Password=<Пароль>&
minDateUTC=<Дата и время начала периода>&
maxDateUTC=<Дата и время окончания периода>

```

Ниже приведен пример запроса:

```

https://integrationapi.net/rest/v2/Sms/In?Login=test_login&Password=test123&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
Login	String	Логин, полученный при регистрации
Password	String	Пароль, соответствующий логину
maxDateUTC	DateTime	Дата и время окончания периода, за который происходит выборка входящих сообщений (например, 2010-06-02T19:14:00).
minDateUTC	DateTime	Дата и время начала периода, за который происходит выборка входящих сообщений (например, 2010-06-01T19:14:00).

Перед получением входящих сообщений, сервис проверяет запрос на:

- Наличие обязательных параметров;
- Валидность Логина/пароля;
- Значение minDateUTC - разница между текущей датой и minDateUTC не может быть больше одного года.
- Значение maxDateUTC и minDateUTC - maxDateUTC должно быть больше чем minDateUTC

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":"7926000001",
"DestinationAddress":"7916000000",
"ID":539187214,
"CreateDateUtc":"\\/Date(1294045911353)\\/"}]
```

Если какая-нибудь проверка не проходит успешно, то сервис возвращает код ошибки (см. Табл. 16) 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/v2/Sms/Statistics?
Login=<Логин>&
Password=<Пароль>&
startDateTime=<Дата и время начала периода>&
endDateTime=<Дата и время конца периода>
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/v2/Sms/Statistics?Login=test_login&Password=test123&startDateTime=2017-07-
18T23:59:00&endDateTime=2017-08-25T23:59:00
```

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

Параметр	Тип данных	Описание
Login	String	Логин, полученный при регистрации
Password	String	Пароль, соответствующий логину
startDateTime	DateTime	Дата и время начала периода, за который необходимо получить статистику, например 2017-07-18T23:59:00.
endDateTime	DateTime	Дата и время конца периода, за который необходимо получить статистику, например 2017-08-25T23:59:00.

После получения запроса сервис проверит валидность логина/пароля 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. 16) в виде JSON следующего формата:

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

Например:

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

Отправка Viber-сообщений

Отправка Viber-сообщения на один номер без учета часового пояса получателя

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

```
https://integrationapi.net/rest/v2/Viber/Send?Login=<Логин>&Password=<Пароль>&SourceAddress=<Адрес отправителя>&DestinationAddress=<Номер получателя>&Data=<Текст сообщения>&Validity=<Время жизни сообщения>&Optional=<Доп. Параметр>
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/v2/Viber/Send?Login=Test&Password=Test&SourceAddress=DTSMS&DestinationAddress=79001234567&Data=testdata&Validity=86400&Optional=123456
```

Табл. 10. Параметры запроса на отправку Viber-сообщения

Параметр	Тип данных	Описание
Обязательные параметры		
Login	String	Логин, полученный при регистрации
Password	String	Пароль, соответствующий логину
DestinationAddress	String	Номер получателя сообщения, в международном формате: код страны + код сети + номер телефона. Пример: 79031234567; +79031234567
Data	String	Текст сообщения, сообщение не должно быть длиннее 1000 символов. Строки разделяются через символ новой строки %0A.
SourceAddress	String	Адрес отправителя сообщения. До 11 латинских или цифровых символов.
Необязательные параметры		
Optional	String	Дополнительный параметр

Параметр	Тип данных	Описание
Validity	Int	Время жизни сообщения (мин, от 1 до 1440)

Перед отправкой Viber-сообщения Сервис проверяет запрос на:

- Наличие обязательных параметров;
- Валидность Логина/пароля;
- Достаточно ли Баланса Пользователя на отправку Viber-сообщения;
- Валидность указанного в запросе номера;
- Валидность адреса отправителя;
- Длину сообщения.

Если все проверки пройдены успешно, то Сервис отправит сообщение 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"]
```

Если какая-нибудь проверка не проходит успешно, то Сервис возвращает Код ошибки (см.Табл. 19) в виде JSON следующего формата:

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

Например:

```
{
  Code: 1
  Desc: "error-address-format"
}
```

Отправка Viber-сообщения на несколько номеров без учета часового пояса получателя

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

```
https://integrationapi.net/rest/v2/Viber/SendBulk?Login=<Логин>&Password=<Пароль>&SourceAddress=<Адрес отправителя>&DestinationAddresses=<Номер(а) получателя>&Data=<Текст сообщения>&Validity=<Время жизни сообщения>&Optional=<Доп. параметр(ы)>
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/v2/Viber/SendBulk?Login=Test&Password=Test&SourceAddress=TESTSMS&DestinationAddresses=79001234567&DestinationAddresses=79059999999&Data=testdata&Validity=86400&Optionals=123456&Optionals=789012
```

Табл. 11. Параметры POST-запроса на отправку Viber-сообщения на несколько номеров

Параметр	Тип данных	Описание
Обязательные параметры		
Login	String	Логин, полученный при регистрации
Password	String	Пароль, соответствующий логину
DestinationAddresses	String	Номер получателя сообщения, в международном формате: код страны + код сети + номер телефона. Пример: 79031234567; +79031234567
Data	String	Текст сообщения, сообщение не должно быть длиннее 1000 символов. Строки разделяются через символ новой строки %0A.
SourceAddress	String	Адрес отправителя сообщения. До 11 латинских или цифровых символов.
Необязательные параметры		
Optionals	String	Дополнительный параметр(или параметры в случае нескольких получателей)
Validity	Int	Время жизни сообщения (мин, от 1 до 1440)

Перед отправкой Viber Сервис проверяет запрос на:

- Наличие обязательных параметров;
- Валидность Логина/Пароля;
- Достаточно ли Баланса Пользователя на отправку Viber;
- 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 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"]
```

Если какая-нибудь проверка не проходит успешно, то Сервис возвращает Код ошибки (см. Табл. 19) в виде JSON следующего формата:

```
{
Code: <Код ошибки>
```

```
Desc: <"Текст ошибки">
}
```

Например:

```
{
Code: 1
Desc: "error-address-format"
}
```

Отправка Viber-сообщения на один номер с переотправкой по SMS

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

```
https://integrationapi.net/rest/v2/Viber/SendWithResend?Login=<Логин>&Password=<Пароль>&SourceAddress=<Адрес отправителя>&DestinationAddress=<Номер получателя>&Data=<Текст сообщения>&Validity=<Время жизни сообщения>&Optional=<Доп. Параметр>
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/v2/Viber/SendWithResend?Login=Test&Password=Test&&SourceAddress=DTSMS&DestinationAddress=79001234567&Data=testdata&Validity=86400&Optional=123456
```

Табл. 12. Параметры запроса на отправку Viber-сообщения

Параметр	Тип данных	Описание
Обязательные параметры		
Login	String	Логин, полученный при регистрации
Password	String	Пароль, соответствующий логину
DestinationAddress	String	Номер получателя сообщения, в международном формате: код страны + код сети + номер телефона. Пример: 79031234567; +79031234567
Data	String	Текст сообщения, сообщение не должно быть длиннее 1000 символов. Строки разделяются через символ новой строки %0A.
SourceAddress	String	Адрес отправителя сообщения. До 11 латинских или цифровых символов. Для корректной работы переотправки адрес отправителя SMS должен быть идентичен используемому адресу отправителя Viber
Необязательные параметры		
Optional	String	Дополнительный параметр
Validity	Int	Время жизни сообщения (мин, от 1 до 1440)

Перед отправкой Viber-сообщения Сервис проверяет запрос на:

- Наличие обязательных параметров;
- Валидность Логина/пароля;
- Достаточно ли Баланса Пользователя на отправку Viber-сообщения;
- Валидность указанного в запросе номера;

- Валидность адреса отправителя;
- Длину сообщения.

Если все проверки пройдены успешно, то Сервис отправит сообщение 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"]
```

Если какая-нибудь проверка не проходит успешно, то Сервис возвращает Код ошибки (см.Табл. 19) в виде JSON следующего формата:

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

Например:

```
{
  Code: 1
  Desc: "error-address-format"
}
```

Отправка Viber-сообщения на несколько номеров с переотправкой по SMS

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

```
https://integrationapi.net/rest/v2/Viber/SendWithResendBulk?Login=<Логин>&Password=<Пароль>&SourceAddress=<Адрес отправителя>&DestinationAddresses=<Номер(а) получателя>&Data=<Текст сообщения>&Validity=<Время жизни сообщения>&Optionals=<Доп. параметр(ы)>
```

Ниже приведен пример запроса:

```
https://integrationapi.net/rest/v2/Viber/SendWithResendBulk?Login=Test&Password=Test&SourceAddress=TESTSMS&DestinationAddresses=79001234567&DestinationAddresses=79059999999&Data=testdata&Validity=86400&Optionals=123456&Optionals=789012
```

Табл. 13. Параметры POST-запроса на отправку Viber-сообщения на несколько номеров

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

Параметр	Тип данных	Описание
DestinationAddresses	String	Номер получателя сообщения, в международном формате: код страны + код сети + номер телефона. Пример: 79031234567; +79031234567
Data	String	Текст сообщения, сообщение не должно быть длиннее 1000 символов. Строки разделяются через символ новой строки %0A.
SourceAddress	String	Адрес отправителя сообщения. До 11 латинских или цифровых символов. Для корректной работы переотправки адрес отправителя SMS должен быть идентичен используемому адресу отправителя Viber
Необязательные параметры		
Optionals	String	Дополнительный параметр(или параметры в случае нескольких получателей)
Validity	Int	Время жизни сообщения (мин, от 1 до 1440)

Перед отправкой Viber Сервис проверяет запрос на:

- Наличие обязательных параметров;
- Валидность Логина/Пароля;
- Достаточно ли Баланса Пользователя на отправку Viber;
- 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 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"]
```

Если какая-нибудь проверка не проходит успешно, то Сервис возвращает Код ошибки (см. Табл. 19) в виде JSON следующего формата:

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

Например:

```
{
  Code: 1
}
```

```
Desc: "error-address-format"
}
```

Получение статуса отправленного Viber-сообщения

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

```
https://integrationapi.net/rest/v2/Viber/State?Login=<Логин>&Password=<Пароль>&messageId=<Идентификатор сообщения>
```

Табл. 14. Параметры GET-запроса статуса отправленного сообщения

Параметр	Тип данных	Описание
Login	String	Логин, полученный при регистрации
Password	String	Пароль, соответствующий логину
messageId	String	Идентификатор сообщения

После получения запроса сервис проверяет валидность связки логина и пароля и наличие отправленного сообщения с присланным идентификатором. Если все проверки пройдены успешно, то сервис вернет статус отправленного viber-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>,
  "ResentSms":[
    {
      "State":<Код статуса переотправленного смс-сообщения>,
      "CreationDateUtc":<Дата создания переотправленного смс-сообщения>,
      "SubmittedDateUtc":<Дата отправки переотправленного смс-сообщения>,
      "ReportedDateUtc":<Дата доставки переотправленного смс-сообщения>,
      "TimeStampUtc":<Дата и время получения отчета по переотправленному смс-сообщению>,
      "StateDescription":<Описание статуса переотправленного смс-сообщения>,
      "Price":<Стоимость переотправленного смс-сообщения>,
      "Id":<Идентификатор переотправленного смс-сообщения>
    }
  ]
}
```

Если какая-нибудь проверка не проходит успешно, то сервис возвращает Код ошибки (см. Табл. 19) 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. 15. Параметры ответа на запрос статуса сообщения

Наименование поля	Описание
State	Статус сообщения
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
ResentSms	Данные о sms-сообщениях, которые были отправлены в рамках переотправки текущего viber-сообщения

Коды ошибок. Статусы SMS и Viber

Табл. 16. Коды ошибок

REST error code	HTTP status code	Описание
•	200	Operation complete
1	400	Argument cannot be null or empty
2	400	Invalid argument
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. 17. Статусы SMS

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

State	Описание
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.

Табл. 18. Статусы viber-сообщений

State	Описание
0	Отправляется
1	Отправлено
2	Доставлено (не прочитано)
3	Доставлено (прочитано)
4	Не доставлено
5	Ошибка
6	Неизвестно
7	Просрочено

State	Описание
8	Переход по ссылке

Табл. 19. Коды возврата обработки сообщения в рамках запроса (Viber-сообщения)

Код	Описание
error-address-format	неправильный формат номера абонента
error-address-not-specified	номер абонента не указан
error-address-unknown	отправка на номерную емкость, к которой относится номер абонента не разрешена клиенту в конфигурации платформы провайдера
error-content-not-specified	содержимое сообщения не указано
error-subject-format	неправильный формат подписи
error-subject-not-specified	подпись не указана
error-subject-unknown	указанная подпись не разрешена клиенту в конфигурации платформы провайдера
error-system	системная ошибка
error-validity-period-seconds-format	неправильно указано значение времени ожидания доставки
ok	исходящее сообщение успешно принято на отправку