



Протоколы работы с СМС-сообщениями

Техническое описание

Версия 2.3, 2025-12-18

Содержание

Формат номера телефона	1
Адрес отправителя	2
Надёжность и отказоустойчивость	3
HTTP API v2 (JSON + API Key)	4
Отправка СМС сообщений	5
Запрос статусов отправленных сообщений	9
Автоматическое получение статусов в JSON	12
Установка статуса "Проверено"	14
Проверка баланса	16
Выгрузка сообщений	17
Создание задания на выгрузку отчёта	17
Скачивание отчета	22
HTTP API v1 (XML)	24
Отправка СМС сообщений	25
Запрос статусов отправленных сообщений	36
Автоматическое получение статусов	39
Установка статуса "Проверено"	40
Проверка баланса	42
Управление списками для рассылок	43
SMPP протокол	46
Кодировки	48
Типичные ошибки	49
UDH-склейка	50
SOAP протокол	52
Метод SendBulkSms	55
Метод SendIndividualSms	56
Метод GetSmsStatus	57
Метод GetBalance	58
SMTP протокол (вариант 1)	59
SMTP протокол (вариант 2)	61
Список кодов ошибок в поле <i>err</i> отчёта о доставке	63
Входящие СМС-сообщения	64
Пересылка входящих сообщений на email	64
Пересылка входящих сообщений на HTTP(S)-скрипт	64
Пересылка входящих сообщений по SMPP	64
Запрос списка входящих сообщений	65

Короткий номер	67
Коды ошибок API	68
Справка по статусам СМС сообщений	71
История версий	74

Формат номера телефона

Номер телефона абонента всегда должен передаваться в международном формате в виде: *код_страны код_оператора номер_телефона* (без ведущего знака «+»).

Код страны	Код оператора	Номер телефона	Страна, оператор
7	916	1112233	Россия, МТС
375	29	2134567	Беларусь, МТС
7	701	1234567	Казахстан, Кселл
380	67	1234567	Украина, Киевстар
34	6400	12345	Испания, Jazz Telecom
371	2231	1234	Латвия, Tele2

Если у абонента прямой городской номер (это касается только российских абонентов), у него должен быть также и второй номер в федеральном формате. В частности, московские абоненты Билайн имеют федеральный аналог своего номера с кодом 903, московские абоненты МТС — с кодом 985, московские абоненты МегаФон — с кодом 925. Прямые номера необходимо указывать с федеральным кодом, например, для абонентов московского МТС не **7495**1234567, а **7985**1234567. Если нет информации, какому оператору принадлежит прямой московский номер, можно указывать его с кодом 495. Нам известно большинство диапазонов прямых московских номеров и, как правило, мы можем сконвертировать номер в федеральный формат. В этом случае в журнале сообщений номер будет отображаться в федеральном формате. В отчёте о доставке, переданном через любой из предоставляемых нами протоколов, номер телефона также будет в федеральном формате.

Адрес отправителя

При отправке СМС в качестве адреса отправителя можно подставить число или альфанумерическую (буквенно-цифровую) последовательность. При этом максимальная длина числового отправителя ограничена 15 цифрами, а альфанумерического - 11 символами. Это ограничение стандарта GSM. В сетях других стандартов могут быть другие ограничения. Например, в CDMA альфанумерические отправители недоступны.

Помимо базовых ограничений стандарта, операторы могут накладывать дополнительные ограничения на длину и содержание имени отправителя. Например: запрет буквенных имен, требование предварительной активации имени отправителя у оператора, запрет использования несуществующих номеров, запрет использования номеров из определённого диапазона и т.п.

На российские номера телефонов запрещена отправка сообщений с цифровым отправителем. Принимаются только альфанумерические.

Большинство операторов создают и используют чёрные списки имён отправителей. В них заносятся имена, замеченные в спам-рассылках, мошенничестве и иных нарушениях, предусмотренных законодательством страны. У SMS Traffic тоже есть подобный список. Как и чёрные списки операторов, наш список не публикуется и не предоставляется по запросу, так как является коммерческой тайной организации.

Если в сеть оператора поступает сообщение с именем, не подходящим под налагаемые ограничения, наиболее типичная реакция — отказ в передаче (хотя все зависит от конкретного оператора, а в некоторых случаях - даже от настроек каждого отдельного СМС-центра в пределах одного оператора). У зарубежных операторов встречается практика передачи ложно-положительных статусов доставки.

Формально, в адресе отправителя можно использовать любой символ из основной таблицы кодировки «default GSM alphabet» (см. [GSM 03.38](#)). При этом, если в тексте встречаются символы из расширенной таблицы, они будут заменены на нашей стороне на похожие по начертанию символы из основной таблицы.

Некоторые СМС-центры и мобильные телефоны поддерживают не весь диапазон символов GSM кодировки в имени отправителя. Поэтому мы рекомендуем пользоваться символами из набора:

A-Z	a-z	0-9	-	_	.
-----	-----	-----	---	---	---

Надёжность и отказоустойчивость

Для обеспечения отказоустойчивости и надёжности сервисов, мы предоставляем два физически распределённых узла.

В случае проблем с соединением с основным узлом, рекомендуется направлять сообщения на дублирующий.

Протокол	Основной	Дублирующий
HTTP API	https://api.smstraffic.ru	https://api2.smstraffic.ru
SMPP	server1.smstraffic.ru:4442	server2.smstraffic.ru:4442
SMPP (TLS)	server1.smstraffic.ru:4441	server2.smstraffic.ru:4441

Для SOAP протокола адрес дублирующего сервера прописан в WSDL.

Для SMTP протоколов все запасные сервера прописаны в MX записях соответствующих протоколам доменов.

Согласно спецификации SMTP, распределение трафика между MX серверами происходит автоматически и, как правило, производить дополнительные настройки для обеспечения отказоустойчивости не нужно.

HTTP API v2 (JSON + API Key)

В версии HTTP API v2 используется современный формат обмена данными JSON и авторизация по ключу API Key, который передаётся в HTTP-заголовке **Authorization: Bearer <API_KEY>**.

Тело запроса (при наличии) и ответы сервера передаются в формате *JSON* с кодировкой *UTF-8*. При отправке запросов методом **POST** обязательно указание заголовка **Content-Type: application/json**. Имена параметров в теле запроса чувствительны к регистру.



Токен состоит из двух частей, разделенных прямым слешем (символ «/»). Первая часть – **несекретная** и ее можно сообщить службе технической поддержки в случае возникновения вопросов по работе протокола, вторая часть – **секретная** и ее необходимо *хранить в тайне*.

Пример токена - **1837/ptf9Bs9Apk4EgyP4gUCy2ajwffVchwaR**. Несекретная часть - **1837**. Секретная - **ptf...waR**.

Каждый токен имеет ограниченную область действия. Если у токена отсутствует разрешение на выполнение действия, сервер вернёт ответ с HTTP кодом **403**.

Также у токена есть ограничения по количеству запросов в единицу времени. При превышении заданного количества сервер возвращает ответ с HTTP кодом **429**. В ответе может быть указан HTTP заголовок *Retry-After*, в котором указывается рекомендуемое время для совершения повторного запроса.

В таблицах с описанием параметров:

- обязательные параметры помечены символом *
- колонка **Тип** содержит тип данных:

string — строка

string (RFC 3339) - строка в формате RFC 3339 (раздел 5.6), например *2025-01-01T23:59:59+03:00*

number — число

boolean — логическое значение (true/false)

object — вложенный JSON-объект

array[x] — массив элементов типа *x*

enum - перечисление разрешенных значений

Вложенность параметров может описываться через точечную нотацию.

Запись **name.nested.subnested** эквивалентна структуре:

```
{ "name": { "nested": { "subnested": ... } } }
```

В настоящее время получить API Key и отредактировать список разрешений для конкретного ключа можно по запросу через *техническую поддержку*.

Отправка СМС сообщений

POST `api.smstraffic.ru/v2/send`

Описание параметров запроса

Поле	Тип	Описание
channels.sms	object	Настройка отправки по смс каналу. См. таблицу параметры объекта SendRequest.Channels.Sms .
destinations *	array[object]	Список абонентов и их индивидуальные параметры. См. таблицу параметры объекта SendRequest.Destinations .
originator	string	Имя отправителя. Если не задано, берется по умолчанию, установленное в личном кабинете.
trackingData	string	Глобальная пользовательская метка trackingData для всех сообщений без индивидуального параметра trackingData. Не более 40 символов.
scheduling	object	Планирование отправки сообщений. См. таблицу параметры объекта SendRequest.Scheduling .
batchName	string	Идентификатор рассылки.
restrictiveId	string	Уникальный идентификатор, предотвращающий повторную отправку в течение часа. Не более 36 символов.

Параметры объекта SendRequest.Channels.Sms

Поле	Тип	По умолчанию	Описание
message	string	-	Текст сообщения общий для всех получателей, если у них не указан индивидуальный текст.
transliterate	boolean	false	Транслитерировать текст сообщения.
maxParts	number	255	Максимальное количество частей сообщения для абонента.

Параметры объекта SendRequest.Destinations

Поле	Тип	Описание
to	number	Номер телефона абонента.
params	object	Параметры подстановки в тексте сообщения.

Поле	Тип	Описание
trackingData	string	Пользовательская метка для отслеживания. Не более 40 символов.
content.sms.message	string	Индивидуальный текст СМС сообщения.

Параметры объекта *SendRequest.Scheduling*

Поле	Тип	По умолчанию	Описание
startDateTime	string (RFC 3339)	-	Дата-время отправки сообщений.
localtime	boolean	true	Учитывать локальное время абонента.

Пример запроса

```

POST /v2/send HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/json
Authorization: Bearer 1837/ptf9Bs9Apk4EgyP4gUCy2ajwffVchwaR

{
  "channels": {
    "sms": {
      "message": "Hello, [[name]]!",
      "timeout": 86400,
      "transliterate": false,
      "maxParts": 255
    }
  },
  "destinations": [
    {
      "to": 78000000001,
      "params": {"name": "Ivan"},
      "trackingData": "2de9efa4-3ee7-4691-bec0-81d93c5a54cf"
    },
    {
      "to": 78000000002,
      "content": {
        "sms": {
          "message": "Individual message for 78000000002"
        }
      }
    }
  ],
  "originator": "Originator",
  "trackingData": "common-tracking-0001",
  "scheduling": {

```

```

"startTime": "2025-01-01T10:30:00+03:00",
"localtime": false
},
"batchName": "Batch via API 0001",
"restrictiveId": "ba52-5ad86d408887"
}

```

Описание параметров ответа

Поле	Тип	Описание
success *	boolean	Признак успешности всего запроса.
error	object	Ошибка, из-за которой весь запрос не был принят к обработке. Присутствует при <code>success = false</code> . См. таблицу параметры объекта Error .
destinations *	array[object]	Список абонентов и статусы отправки по каждому. См. таблицу параметры объекта SendResponse.Destinations .

Описание параметров объекта SendResponse.Destinations

Поле	Тип	Описание
phone *	number	Номер телефона абонента.
id *	string	Идентификатор отправленного сообщения.
trackingData	string	Пользовательская метка, переданная из запроса.
error	object	Ошибка, из-за которой сообщение конкретному абоненту не было принято к обработке или отправке. См. таблицу параметры объекта Error .

Описание параметров объекта Error

Поле	Тип	Описание
code *	number	Код ошибки. Возможные коды ошибок указаны в таблице коды ошибок API .
message *	string	Текстовое описание ошибки.

Пример ответа

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "success": true,
  "destinations": [
    {

```

```
"phone": 78000000001,  
"id": "1046379029789147161",  
"trackingData": "2de9efa4-3ee7-4691-bec0-81d93c5a54cf"  
},  
{  
  "phone": 78000000002,  
  "id": "1046379029789147194",  
  "trackingData": "common-tracking-0001",  
  "error": {  
    "code": 432,  
    "message": "blocked phone: 78000000002. No messages has been sent"  
  }  
}  
]  
}
```

Пример ответа (с ошибкой)

```
HTTP/1.1 200 OK  
Content-Type: application/json  
  
{  
  "success": false,  
  "error": {  
    "code": 105,  
    "message": "Duplicated request"  
  }  
}
```

Запрос статусов отправленных сообщений

POST `api.smstraffic.ru/v2/statuses/list`

Описание параметров запроса

Поле	Тип	По умолчанию	Описание
ids	array[string]	-	Список идентификаторов смс, полученных при отправке через <code>/send</code> .
trackingData	array[string]	-	Список пользовательских <code>trackingData</code> , присвоенных сообщениям при отправке через <code>/send</code> .
extendedInfo	boolean	false	Нужно ли возвращать дополнительное поле <code>statusExtended</code> с деталями ошибки от оператора.

Пример запроса

```
POST /v2/statuses/list HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/json
Authorization: Bearer 1837/ptf9Bs9Apk4EgyP4gUCy2ajwffVchwaR
```

```
{
  "ids": [
    "1046379029789147161",
    "1046379029789147162"
  ],
  "trackingData": [
    "108c3492-98ef-44b4-ab17-ee20c864584a",
    "108c3492-98ef-44b4-ab17-ee20c864584b"
  ],
  "extendedInfo": false
}
```

Описание параметров ответа

Поле	Тип	Описание
success *	boolean	Признак успешности всего запроса.

Поле	Тип	Описание
error	object	Ошибка, из-за которой весь запрос не был принят к обработке. Присутствует при <code>success = false</code> . См. таблицу параметры объекта Error .
data *	array[object]	Список сообщений и их текущих статусов. См. таблицу таблицу StatusesListResponse.Data .

Описание параметров объекта *StatusesListResponse.Data*

Поле	Тип	Описание
smsId *	string	Идентификатор отправленного смс сообщения.
trackingData	string	Пользовательская <code>trackingData</code> метка.
submissionDate *	string (RFC 3339)	Дата и время принятия сообщения платформой.
sendDate	string (RFC 3339)	Дата и время отправки сообщения оператору.
deliveryDate	string (RFC 3339)	Дата и время получения финального статуса.
status *	string	Текущий статус сообщения. Возможные значения см. в справке по статусам СМС сообщений .
statusExtended	object	Дополнительная информация об ошибке от оператора (возвращается при <code>extendedInfo = true</code>). См. таблицу описание параметров StatusesListResponse.StatusExtended .
operatorId	number	Идентификатор оператора.
mcc	number	Mobile Country Code.
mnc	number	Mobile Network Code.

Описание параметров объекта *StatusesListResponse.StatusExtended*

Поле	Тип	Описание
code *	number	Код ошибки от оператора.
description *	string	Описание ошибки от оператора.

Пример ответа

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "success": true,
  "data": [
    {
      "smsId": "1046379029789147161",
      "trackingData": "108c3492-98ef-44b4-ab17-ee20c864584a",
    }
  ]
}

```

```
"submissionDate": "2025-11-27T13:56:41.562+03:00",
"sendDate": "2025-11-27T13:56:41.593+03:00",
"deliveryDate": "2025-11-27T13:56:42.341+03:00",
"status": "Delivered",
"operatorId": 1971,
"mcc": 250,
"mnc": 99
},
{
  "smsId": "1046379029789147162",
  "submissionDate": "2025-11-27T13:58:12.021+03:00",
  "sendDate": "2025-11-27T13:58:12.593+03:00",
  "deliveryDate": "2025-11-27T13:59:42.341+03:00",
  "status": "Non Delivered",
  "statusExtended": {
    "code": 3,
    "description": "Unknown subscriber"
  }
}
]
}
```

Пример ответа (с ошибкой)

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "success": false,
  "error": {
    "code": 105,
    "message": "Invalid request format"
  }
}
```

Автоматическое получение статусов в JSON

Можно автоматически получать от нас статусы сообщений после каждого изменения статуса (*рекомендуется*) в формате JSON. Вы будете получать JSON запросы, в теле которых будет массив из объектов, в одном запросе по умолчанию не более 50 статусов (ограничение можно отрегулировать через обращение в техническую поддержку).

Вашему серверу необходимо отдавать в ответ на запросы HTTP статус код ответа 200 или 204 (заголовки и тело ответа не имеют значения), в противном случае наша система отправки статусов пометит ваш скрипт как "временно недоступный", частота отправки статусов будет снижена до момента, пока скрипт снова не начнет отдавать корректные HTTP статус коды.



Для включения автоматического получения статусов в JSON формате необходимо обратиться в техническую поддержку.

Описание параметров объектов-элементов массива

Поле	Тип	Описание
login *	string	Логин учетной записи.
sms_id *	string	Идентификатор отправленного сообщения.
phone_number *	string	Номер телефона абонента.
status *	string	Текущий статус сообщения. Возможные значения см. в справке по статусам СМС сообщений .
delivery_date *	string	Дата получения текущего статуса в формате «ГГГГ-ММ-ДД ЧЧ:ММ:СС», часовой пояс UTC+3.
channel *	string	Канал сообщения, по которому пришел статус. Всегда значение "sms".
parts_count	number	Количество частей сообщения. Возвращается только при отправке сообщений через /v2/send.
error_code *	number	Код ошибки. Если указан 0, то запрос был обработан без ошибок. Возможные коды ошибок указаны в таблице коды ошибок API .
tracking_data	string	Пользовательская метка для отслеживания, указанная при отправке через /v2/send.
operator_id	number	Идентификатор оператора.
mcc	number	Mobile Country Code.
mnc	number	Mobile Network Code.

Пример запроса (от SMS Traffic)

```
POST /status_callback_url HTTP/1.1
```

```
User-Agent: StatusSender/2.0 Java/21
```

```
Host: example.com
```

```
Content-type: application/json
```

```
[
  {
    "login": "login:subacc",
    "sms_id": "8049537157380826112",
    "phone_number": "78000000001",
    "status": "Delivered",
    "delivery_date": "2025-12-06 12:31:00",
    "channel": "sms",
    "parts_count": 2,
    "err_code": 0,
    "tracking_data": "2de9efa4-3ee7-4691-bec0-81d93c5a54cf",
    "operatorId": 1971,
    "mcc": 250,
    "mnc": 99
  },
  ...
]
```

Пример ответа

```
HTTP/1.1 200 OK
```

```
Content-Type: text/plain
```

Установка статуса "Проверено"

POST `api.smstraffic.ru/v2/statuses/checked`

Описание параметров запроса

Поле	Тип	Описание
(body)	array[string]	Список идентификаторов сообщений, которым необходимо установить статус Checked .

Пример запроса

```
POST /v2/statuses/checked HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/json
Authorization: Bearer 1837/ptf9Bs9Apk4EgyP4gUCy2ajwffVchwaR
```

```
[
  "1046379029789147161",
  "1046379029789147162"
]
```

Описание параметров ответа

Поле	Тип	Описание
success *	boolean	Признак успешности всего запроса.
error	object	Ошибка, из-за которой весь запрос не был принят к обработке. Присутствует при <code>success = false</code> . См. таблицу параметры объекта Error .
data *	object	Результат обработки запроса. См. таблицу параметры объекта StatusesCheckedResponse.Data .

Описание параметров объекта StatusesCheckedResponse.Data

Поле	Тип	Описание
updated *	array[string]	Список id сообщений, для которых статус Checked был успешно установлен.
failed *	array[object]	Список сообщений, для которых не удалось установить статус Checked . См. таблицу описание параметров StatusesCheckedResponse.Data.Failed .

Описание параметров объекта StatusesCheckedResponse.Data.Failed

Поле	Тип	Описание
id *	string	Id сообщения, которому не был установлен статус.
error *	string	Текстовое описание причины ошибки.

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "success": true,
  "data": {
    "updated": [
      "1046379029789147161"
    ],
    "failed": [
      {
        "id": "1046379029789147162",
        "error": "no such message or this message does not belong to you
1046379029789147162"
      }
    ]
  }
}
```

Пример ответа (с ошибкой)

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "success": false,
  "error": {
    "code": 403,
    "message": "Access denied"
  }
}
```

Проверка баланса

```
GET api.smstraffic.ru/v2/account
```

Описание параметров запроса

Без параметров

Пример запроса

```
GET /v2/account HTTP/1.1  
Host: api.smstraffic.ru  
Authorization: Bearer 1837/ptf9Bs9Apk4EgyP4gUCy2ajwfFVchwaR
```

Описание параметров ответа

Поле	Тип	Описание
success *	boolean	Признак успешности всего запроса.
data.account *	string	Текущий баланс. Может быть отрицательным числом.

Пример ответа

```
HTTP/1.1 200 OK  
Content-Type: application/json  
  
{  
  "success": true,  
  "data": {  
    "account": "12520.35"  
  }  
}
```

Выгрузка сообщений

Выгрузка списка сообщений осуществляется двумя отдельными вызовами REST API.

1. Создание задания на выгрузку отчёта
2. Скачивание отчёта

Создание задания на выгрузку отчёта и скачивание этого отчета должны выполняться **на одном и том же узле** – оба запроса должны идти либо на `api.smstraffic.ru`, либо на `api2.smstraffic.ru`, поскольку сгенерированные отчеты не синхронизируются между узлами.

Создание задания на выгрузку отчёта

POST `api.smstraffic.ru/v2/report`

Описание параметров запроса

Поле	Тип	Описание
from	string (RFC 3339)	Дата и время начала отчёта. В отчёт попадут сообщения, отправленные с указанного времени (включительно).
to	string (RFC 3339)	Дата и время окончания отчёта. В отчёт попадут сообщения, отправленные до указанного времени (включительно).
originator	string	Имя отправителя. Будут выгружены сообщения с указанным именем отправителя (регистр учитывается).
subscriber	string	Получатель сообщения, в зависимости от канала: номер телефона в sms, viber, whatsapp, vk, ok и telegram, произвольный идентификатор в push. Будут выгружены сообщения, отправленные указанному получателю.
type	string (enum: mt, mo)	Тип сообщения – mt (исходящее), mo (входящее). Если параметр не задан, выгружаются сообщения всех типов.

Поле	Тип	Описание
status	string (enum: delivered, not_delivered, read)	Статус доставки сообщений. <i>delivered</i> – сообщение в статусах “Delivered” или “Read”. <i>not_delivered</i> – сообщения в любых статусах, кроме промежуточных, «Delivered» и «Read». <i>read</i> – только в статусе «Read». Если параметр не задан, выгружаются сообщения в любом статусе.
channels	array[string] (enum: sms, push, viber, whatsapp, vk, ok, telegram)	Список каналов отправки сообщений в формате JSON массива. Если параметр не задан, выгружаются сообщения по всем каналам.
department	string	Логин департамента, отправившего сообщение. Специальное значение «» (пустая строка) обозначает главную учётную запись. Если параметр не задан, выгружаются сообщения по главной учётке и всем департаментам. Если запрос совершается от имени департамента, данный параметр игнорируется и автоматически устанавливается в логин того департамента, от имени которого совершается запрос.
fields	array[string] (enum: id, subscriber, message, submission_date, send_date, delivery_date, read_date, channel, status, cost, originator, ip, error, parts, operator_name, country_name, batch_name, department_name)	Список полей для выгрузки в формате JSON массива. Поля в выгрузке будут идти в том порядке, в котором указаны в запросе. Если параметр не указан, значение по умолчанию – [id, subscriber, message, submission_date, send_date, delivery_date, status, originator]. Описание полей см. в таблице описание полей fields .

Поле	Тип	Описание
format	string (enum: zip, csv, json)	Формат выгрузки. Значение по умолчанию – json. JSON генерирует отчёт в форме JSON объекта вида {"count":1,"messages":[["1","2"]]}, где count - общее количество отфильтрованных данных, messages – массив сообщений, представленных в виде массива полей, указанных в параметре fields. Поля идут в том же порядке, в котором задавались в параметре fields. Формат zip - это csv, упакованный в zip архив.
limit	number	Максимальное количество сообщений, возвращаемое в ответе. Может принимать значение в диапазоне от 1 до 1000. Значение по умолчанию – 50. Обычно используется совместно с параметром offset для организации постраничной навигации. Применяется только при format=json. Для остальных форматов выгрузки данный параметр игнорируется и значение по умолчанию не применяется.
offset	number	Номер сообщения, начиная с которого будет производиться выгрузка. Может принимать значение от 0 до 1000. Обычно используется совместно с параметром limit для организации постраничной навигации. Значение по умолчанию – 0. Применяется только при format=json. Для остальных форматов выгрузки данный параметр игнорируется и значение по умолчанию не применяется.

Описание полей fields

Поле	Описание
id	Уникальный идентификатор сообщения на платформе SMS Traffic (строка, содержащая до 40 цифр).
subscriber	Получатель сообщения, в зависимости от канала: номер телефона в sms, viber, whatsapp, vk и ok, произвольный идентификатор в push.
message	Текст сообщения.
submission_date	Дата и время поступления сообщения от клиента в формате RFC 3339.
send_date	Дата и время отправки сообщения в канал в формате RFC 3339. Поле может быть пустым, если сообщение ещё не отправлялось в канал.
delivery_date	Дата и время доставки или последнего изменения статуса в формате RFC 3339. Поле может быть пустым, если сообщение ещё не получило финальный статус.

Поле	Описание
read_date	Дата и время прочтения сообщения получателем в формате RFC 3339. Поле может быть пустым, если сообщение ещё не прочитано или канал не поддерживает статус Read.
channel	Канал отправки сообщения.
status	Статус доставки.
originator	Имя отправителя, с которым сообщение отправлялось в канал.
ip	IP-адрес, с которого было отправлено сообщение.
error	В случае недоставки здесь указывается код причины недоставки.
parts	Количество сегментов СМС. Заполняется только для сообщений из канала sms, по остальным каналам в данном поле будет пустая строка.
operator_name	Имя оператора, за которым закреплён номер телефона получателя. Заполняется только для каналов, у которых получателем является номер телефона (sms, viber, whatsapp, vk, ok, push). Пример: «Мобильные ТелеСистемы», «Йота».
country_name	Название страны, за которой закреплён номер телефона получателя. Заполняется только для каналов, у которых получателем является номер телефона (sms, viber, whatsapp, vk, ok). Пример: Russian Federation, Kazakhstan.
batch_name	Имя рассылки, в которой участвует данное сообщение.
department_name	Имя департамента, отправившего сообщение. Или пустая строка, если сообщение отправлено с основной учётной записи.

Пример запроса

```

POST /v2/report HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/json
Authorization: Bearer 1837/ptf9Bs9Apk4EgyP4gUCy2ajwfFVchwaR

{
  "from": "2025-01-01T00:00:00+03:00",
  "to": "202-01-01T23:59:59+03:00",
  "channels": [
    "sms"
  ],
  "fields": [
    "id",
    "submission_date",
    "message",
    "status"
  ]
}

```

```
}
```

Описание параметров ответа

Поле	Тип	Описание
code *	number	Числовой код ошибки, возникшей при обработке запроса. Принимает значение 0 в случае успеха.
description *	string	Текстовая расшифровка кода ошибки. Рекомендуется записать в лог. Если code = 0, в данном параметре будет пустая строка.
report_id *	string	Идентификатор отчёта. Данный параметр отсутствует если code > 0.

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "code": 0,
  "description": "",
  "reportId": "16e15fec1d27803e18511793b5bf699e"
}
```

Скачивание отчета

```
GET api.smstraffic.ru/v2/report/{report_id}
```

Вместо `{report_id}` необходимо подставить идентификатор отчёта, выданный на предыдущем этапе. Если отчёт уже готов, в ответе будет HTTP статус код 200, а в теле ответа содержимое самого отчёта.

Пример запроса

```
GET /v2/report/16e15fec1d27803e18511793b5bf699e HTTP/1.1
Host: api.smstraffic.ru
Authorization: Bearer 1837/ptf9Bs9Apk4EgyP4gUCy2ajwfFVchwaR
```

Пример ответа (с отчётом в формате CSV):

```
HTTP/1.1 200 OK
Content-Disposition: attachment; filename="16e15fec1d27803e18511793b5bf699e.csv"
Accept-Ranges: bytes
Content-Type: text/csv
Content-Length: 53

12345;2021-01-01T00:00:00+03:00;hello world;Delivered
```

Пример ответа (с отчётом в формате JSON):

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 78

{"count":1,"messages":[[{"12345"}, {"2021-01-01T00:00:00+03:00"}, {"hello world"}, {"Delivered"}]]}
```

Архитектурно возможно появление в json ответе новых свойств, поэтому необходимо при проектировании архитектуры HTTP-клиента учитывать эту возможность.

В случае если отчёт ещё не готов, запрос будет заблокирован до момента готовности отчёта (техника long polling). Максимальное время нахождения запроса в блокировке — 5 минут, по истечении которых сервер генерирует ответ с HTTP статус кодом 204. Данный ответ обозначает, что отчёт ещё не готов.

Пример ответа (если отчет еще не готов):

```
HTTP/1.1 204 No Content
```

В случае получения 204 статус кода рекомендуется повторять запросы до получения 200, 404 или иного ответа. Рекомендуется выдерживать интервал между запросами не менее 1 секунды. Готовые отчёты хранятся 7 дней, по истечению которых запрос на скачивание будет возвращать статус код 404.

Пример ответа (если отчет более недоступен):

```
HTTP/1.1 404 Not Found
```

```
Content-Type: application/json
```

```
Content-Length: 76
```

```
{"code":1, "description":"Report 16e15fec1d27803e18511793b5bf699e not found"}
```

HTTP API v1 (XML)

Отправка сообщений на телефон абонента осуществляется через вызов веб-скрипта ***api.smstraffic.ru/multi.php***, которому передаются необходимые параметры методом GET или POST. При отсутствии связи или других проблемах на *api.smstraffic.ru*, можно использовать наш запасной сервер: *api2.smstraffic.ru/multi.php*. Обращения принимаются как по протоколу HTTP, так и по HTTPS (с использованием TLS-шифрования). Номера портов стандартные: HTTP — 80, HTTPS — 443. HTTP-запросы должны соответствовать официальным спецификациям протокола HTTP ([RFC 2616](#)). Параметры должны быть корректно кодированы ([RFC 2396](#), раздел 2.4). Поддерживается HTTP протокол версий 1.0 и 1.1. При работе по протоколу HTTP 1.1 ответ может передаваться в [chunked transfer encoding](#). При отправке запросов методом POST обязательно указание заголовка «*Content-Type: application/x-www-form-urlencoded*».

Отправка СМС сообщений

POST / GET `api.smstraffic.ru/multi.php`

Описание параметров запроса

Имя параметра	Возможные значения	Описание
login *	Текст	Ваш логин в системе SMS Traffic.
password *	Текст	Ваш пароль в системе SMS Traffic.
phones	Список номеров через запятую	Список телефонов в международном формате, разделённых запятыми (формат номера телефона описан в разделе " Формат номера телефона "). Можно ограничиться только одним номером. Этот параметр обязателен (кроме случаев с использованием параметра <i>group</i>).
message	Текст длиной до 17085 (при <i>rus=1</i> или <i>rus=5</i>) или 39015 (при <i>rus=0</i>) символов	Текст сообщения, не более 160 символов латинскими буквами или 70 символов кириллицы в кодировке Windows-1251. При превышении этих границ сообщение автоматически разделится на несколько частей, каждая из которых не превышает 153 символа в латинице и 67 символов кириллицы. Максимальное количество частей регулируется параметром <i>max_parts</i> . Параметр обязателен (кроме случаев с использованием параметра <i>individual_messages</i>).

Имя параметра	Возможные значения	Описание
rus	0	Сообщение передано в кодировке Windows-1251. Символы кириллицы будут транслитерироваться в латиницу. Максимальное количество символов в одном СМС сообщении — 160 (153 для склеенного сообщения). Значение по умолчанию — 0.
	1	Сообщение передано в кодировке Windows-1251. Максимальное количество символов в одном СМС сообщении — 70 (67 для склеенного сообщения). Если сообщение состоит исключительно из латинских символов, максимальное количество символов не изменяется.
	5	Сообщение передано в кодировке UTF-8. Максимальное количество символов в одном СМС-сообщении — 70 (67 для склеенного сообщения). Если сообщение состоит исключительно из латинских символов, максимальное количество символов не изменяется.
originator	Альфанумерический (максимальная длина 11 символов)	Параметр задаёт отправителя сообщения, как он будет выглядеть на телефоне получателя. Может состоять из латинских букв, цифр и знаков пунктуации. Буквы кириллицы в отправителе принудительно транслитерируются в латинские. Так же допустимо использовать схожие с кириллицей по начертанию латинские буквы. На альфанумерического отправителя нельзя позвонить или отправить ответное сообщение. Примеры: «MyCompany», «787-35-95», «SMS.Traffic», «СМС.ТРАФИК».
	Только цифры (максимальная длина 15 цифр)	Если указан цифровой отправитель, абонент может отправить ответное сообщение или перезвонить по указанному в отправителе номеру. Примеры: «74957873595», «88001000258», «3299».

Имя параметра	Возможные значения	Описание
flash	1	Сообщение отправляется как flash SMS. Flash SMS появляется сразу на экране телефона и не сохраняется в памяти телефона автоматически. Сообщения данного типа имеют максимальную длину: в латинице - 160 символов (rus=0), в кириллице — 70 символов (rus 1 или 5).
	0	СМС отправляется как обычное сообщение. Значение по умолчанию — 0.
start_date	Дата в формате «ГГГГ-ММ-ДД ЧЧ:ММ:СС»	Дата и время отправки СМС. Поле можно оставить пустым, в таком случае сообщения уйдут немедленно. Время указывается в часовом поясе UTC+3. Start_date может быть не более чем на 3 суток в будущем. Если start_date находится в будущем более чем на 5 минут, то применение параметра want_sms_ids невозможно. Пример: «2020-05-09 09:00:00».
max_parts	Число от 1 до 255	Максимальное количество частей, на которые будет при необходимости разбит текст сообщения. Если текст сообщения не укладывается в одну часть, то длина одной части сообщения ограничивается 153 символами для латиницы и 67 для кириллицы. Если сообщение после разбивки превышает установленное значение max_parts, то отправлены будут первые max_parts частей, а остальные отброшены. По умолчанию установлено максимальное значение — 255.
group	Текст	Имя группы (списка для рассылок), определенное в личном кабинете клиента, на которую идет рассылка. Этот параметр заменяет параметр phones. Сообщение будет отправлено на все телефоны в группе.

Имя параметра	Возможные значения	Описание
timeout	Положительное целое число	<p>С помощью этого параметра можно установить время жизни СМС. Сообщение перестанет автоматически доставляться через <i>timeout</i> секунд с момента получения сообщения платформой. Данный параметр поддерживается не всеми операторами. Минимально возможное значение так же зависит от оператора. Большинство операторов ограничивают минимальное значение 10 минутами. При указании меньшего значения, параметр может быть проигнорирован (сброшен до максимального значения), или же в ответ на запрос оператор может вернуть статус "отклонено". Со стороны нашего СМС-центра минимальное значение не ограничивается. Максимальное значение — 86400 (24 часа).</p>
individual_messages	0 или 1	<p>Если необходимо отправить индивидуальное сообщение каждому абоненту, можно либо несколько раз запрашивать скрипт, передавая в качестве параметра <i>phones</i> только один телефон, либо (что более предпочтительно) передать дополнительный параметр <i>individual_messages=1</i>, оставить поле <i>message</i> пустым, а в поле <i>phones</i> передать список телефонов и сообщений в формате:</p> <p><i>телефон1 сообщение1</i> <i>телефон2 сообщение2</i> <i>телефон3 сообщение3</i></p> <p>Телефон и сообщение разделяются одним пробелом, а пары телефон-сообщение разделяются знаком перевода строки (символ с ASCII-кодом 0xA или 0xD), при этом текст сообщения не может содержать символа перевода строки. Значение по умолчанию — 0.</p>

Имя параметра	Возможные значения	Описание
delimiter	Текст	<p>Этот параметр работает совместно с параметром <i>individual_messages</i> при необходимости передачи символа перевода строки внутри сообщения. Например, при указании <i>delimiter=ABC</i> параметр <i>phones</i> может принимать вид:</p> <p><i>76161234567 текст сообщения ABC</i> <i>7903123456 текст</i> <i>сообщения на</i> <i>нескольких строках</i></p> <p>В итоге каждый абонент получит предназначенное ему сообщение, даже если в нём есть перевод строки. При этом текст сообщения не должен содержать в себе ABC, иначе произойдёт некорректный разбор параметра. По умолчанию разделителем является символ переноса строки (ASCII-код 0xA или 0xD).</p>
want_sms_ids	0 или 1	<p>Если необходимо получить информацию об идентификаторах, присвоенных каждому сообщению (они понадобятся при проверке статуса доставки сообщения), нужно передать параметр <i>want_sms_ids=1</i>. Тогда ответный XML будет содержать информацию о каждом телефоне и идентификаторе, присвоенном соответствующему сообщению. Параметр <i>want_sms_ids=1</i> нельзя использовать в отсроченной рассылке, то есть одновременно с параметром <i>start_date</i>. Этот параметр можно применять только при условии, что сообщение отправится не позднее 5 минут с момента поступления запроса. Идентификатор представлен целым беззнаковым числом размером 8 байт. Максимальное число сообщений в одном запросе с использованием данного параметра не должно превышать 6000. Значение по умолчанию — 0.</p>

Имя параметра	Возможные значения	Описание
with_push_id	0 или 1	<p>Используется только совместно с параметрами <i>want_sms_ids=1</i> и <i>individual_messages=1</i>. Параметр <i>with_push_id</i> необходим, когда требуется передать каждое сообщение со своим уникальным идентификатором и в ответ получить привязку переданных идентификаторов к выданным нашей системе идентификаторам. Типичный случай использования - передача длинных сообщений. В обычных условиях, при отправке длинного сообщения, оно разбивается на несколько частей и каждой части присваивается свой идентификатор. Чтобы связать несколько идентификаторов одного сообщения, используется <i>with_push_id</i>. При этом, в параметре <i>phones</i> перед каждым номером должен быть указан произвольный идентификатор (это может быть идентификатор сообщения в вашей базе), отделённый от номера двоеточием. Сам идентификатор не должен содержать двоеточия.</p> <p>Пример:</p> <p><i>push_id1:телефон1 сообщение1 из двух частей</i> <i>12345678:телефон2 сообщение2 из двух частей</i> <i>one-more:телефон3 сообщение3 из двух частей</i></p> <p>Значение по умолчанию — 0.</p>

Имя параметра	Возможные значения	Описание
ignore_phone_format	0 или 1	<p>Данный параметр используется при единовременной рассылке на несколько номеров телефонов. Если при отправке сообщения хотя бы один номер в запросе будет некорректен, то при <i>ignore_phone_format=0</i>, возвращается ошибка 418 и ни одно из сообщений не отправляется. Если же установить параметр <i>ignore_phone_format=1</i>, то проверка номеров отключается и все сообщения, независимо от корректности номеров, становятся в очередь на отправку и биллинруются соответственно вашему тарифу.</p> <p>Значение по умолчанию — 0.</p>
two_byte_concat	0 или 1	<p>Параметр позволяет указать способ UDH-склейки. Если указано 1 — используется склейка с reference number размером 2 байта. В противном случае используется склейка с reference number размером 1 байт. 2-х байтовый reference number позволяет значительно снизить вероятность некорректной склейки сообщений в телефоне, однако уменьшает максимальный размер одной части на 1 символ. То есть, не более 152 символов латиницей и 66 символов - кириллицей.</p> <p>Значение по умолчанию — 0.</p>

Примеры ответных сообщений

В качестве ответа скрипт отдает XML с результатом постановки сообщений в очередь. В примерах ниже по тексту, XML представлен в отформатированном виде, поэтому значение заголовка *Content-Length* может не совпадать с актуальной длиной отформатированного XML в примере. Так же, для удобочитаемости, параметры запроса могут разбиваться на несколько строк. На самом деле, при использовании протокола HTTP API всё тело запроса должно идти в одной строке.

Пример отправки одного сообщения:

Пример запроса

```
POST /multi.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded

login=mylogin&password=mypassword&phones=78001234567&message=test+%F2%E5%F1%F2
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" ?>
<reply>
  <result>OK</result>
  <code>0</code>
  <description>queued 1 messages</description>
</reply>
```

В случае успеха:

- поле *result* - содержит строку OK;
- *code* = 0;
- *description* - содержит число сообщений, успешно поставленных в очередь.

В случае возникновения ошибки:

- *result* = ERROR;
- *code* > 0, возможные коды ошибок указаны в таблице [коды ошибок API](#);
- *description* - содержит описание ошибки (например, 'Authentication failed'). Все сообщения в *description* передаются на английском языке.

Ошибка 1000 означает временные проблемы на сервере. При ее получении можно попробовать повторить запрос через некоторое время или отправить запрос на

дублирующий сервер. Остальные ошибки следует обрабатывать соответственно их описанию.

Пример ответа скрипта в случае ошибки:

Пример запроса

```
POST /multi.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded

LOGIN=mylogin&password=mypassword&phones=78001234567&message=test+%F2%E5%F1%F2
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" ?>
<reply>
  <result>ERROR</result>
  <code>401</code>
  <description>login param is missing</description>
</reply>
```

При передаче параметра `want_sms_ids=1` в ответном XML появляется дополнительный элемент со списком идентификаторов. Идентификатор представлен целым беззнаковым числом размером 8 байт.

Пример запроса

```
POST /multi.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded

login=mylogin&password=mypassword&want_sms_ids=1&phones=79051112233,79261112233&message=test+%F2%E5%F1%F2
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" ?>
<reply>
  <result>OK</result>
  <code>0</code>
```

```
<description>queued 2 messages</description>
<message_infos>
  <message_info>
    <phone>79051112233</phone>
    <sms_id>1000472891</sms_id>
  </message_info>
  <message_info>
    <phone>79261112233</phone>
    <sms_id>1000472892</sms_id>
  </message_info>
</message_infos>
</reply>
```

При передаче параметра `with_push_id=1` в ответном XML появляются дополнительные данные:

Пример запроса

```
POST /multi.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
```

```
login=mylogin&password=mypassword&want_sms_ids=1&with_push_id=1&individual_messages=1&
delimiter=ABC&phones=a:79051112233+<long message>ABCb:79261112233+hello+test
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: text/xml

<reply>
  <result>OK</result>
  <code>0</code>
  <description>queued 3 messages</description>
  <message_infos>
    <message_info>
      <phone>79051112233</phone>
      <sms_id>8287366071</sms_id>
      <push_id>a</push_id>
    </message_info>
    <message_info>
      <phone>79051112233</phone>
      <sms_id>8287366073</sms_id>
      <push_id>a</push_id>
    </message_info>
    <message_info>
      <phone>79261112233</phone>
```

```
<sms_id>8287366075</sms_id>  
<push_id>b</push_id>  
</message_info>  
</message_infos>  
</reply>
```

Запрос статусов отправленных сообщений

POST / GET `api.smstraffic.ru/multi.php`

Описание параметров запроса

Имя параметра	Возможные значения	Описание
login *	Текст	Ваш логин в системе SMS Traffic.
password *	Текст	Ваш пароль в системе SMS Traffic.
operation *	Текст	Всегда значение <i>status</i> .
sms_id *	Текст	Список идентификаторов смс, разделённых запятой.

В ответ сервер возвращает XML с результатом запроса, где:

- *submission_date* - содержит дату приёма сообщения платформой SMS Traffic;
- *send_date* - дата отправки сообщения оператору;
- *last_status_change_date* - дата доставки или время последнего изменения статуса сообщения.

Мы работаем с операторами по протоколу SMPP, в котором, согласно описанию формата статуса, используется время с точностью до минуты. Поэтому дата доставки часто имеет нулевые секунды. Таким образом, если сообщение было отправлено, например, в 18:43:38, а в *last_status_change_date* указано время 18:43:00, это означает что сообщение было доставлено в 43 минуту 18 часа. В данном случае на секунды можно не обращать внимания.

Все даты передаются в формате «ГГГГ-ММ-ДД ЧЧ:ММ:СС», часовой пояс *UTC+3*.

Статусы хранятся в течение двух дней, после чего они перемещаются в архив. На попытки запроса статусов сообщений, отправленных более двух дней назад, будет возвращаться ошибка *"No such message or this message does not belong to you"*.

Пример запроса статуса одного сообщения:

Пример запроса

```
POST /multi.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded

login=mylogin&password=mypassword&operation=status&sms_id=8287713301
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: text/xml

<reply>
  <submission_date>2020-05-09 14:11:39</submission_date>
  <send_date>2020-05-09 14:11:39</send_date>
  <last_status_change_date>2020-05-09 14:12:00</last_status_change_date>
  <status>Delivered</status>
  <error></error>
  <sms_id>8287713301</sms_id>
</reply>
```

Пример запроса статуса нескольких сообщений:

Пример запроса

```
POST /multi.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded

login=mylogin&password=mypassword&operation=status&sms_id=8287713301,8287713303,82877133031
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" ?>
<reply>
  <sms>
    <error></error>
    <submission_date>2020-05-09 14:11:39</submission_date>
    <send_date>2020-05-09 14:11:39</send_date>
    <last_status_change_date>2020-05-09 14:12:00</last_status_change_date>
    <sms_id>8287713301</sms_id>
    <status>Expired</status>
  </sms>
  <sms>
    <error></error>
    <submission_date>2020-05-09 14:11:39</submission_date>
    <send_date>2020-05-09 14:11:40</send_date>
    <last_status_change_date>2020-05-09 14:12:00</last_status_change_date>
    <sms_id>8287713303</sms_id>
    <status>Delivered</status>
  </sms>
</reply>
```

```
</sms>  
<sms>  
  <error>  
    no such message or this message does not belong to you  
  </error>  
  <sms_id>82877133031</sms_id>  
</sms>  
</reply>
```

Автоматическое получение статусов

Можно автоматически получать от нас статусы сообщений в POST запросах на ваш скрипт после каждого изменения статуса (рекомендуется).

Вашему серверу необходимо отдавать в ответ на запросы HTTP статус код ответа 200 или 204 (заголовки и тело ответа не имеют значения), в противном случае наша система отправки статусов пометит ваш скрипт как "временно недоступный", частота отправки статусов будет снижена до момента, пока скрипт снова не начнет отдавать корректные HTTP статус коды.



Для включения автоматического получения статусов необходимо обратиться в техническую поддержку.

Описание параметров в теле POST запроса

Поле	Описание
sms_id *	Идентификатор сообщения, представлен целым беззнаковым числом размером 8 байт.
status *	Статус доставки. Возможные значения статуса см. в справке по статусам СМС сообщений .
delivery_date *	Дата доставки или последнего изменения статуса. Указывается в формате «ГГГГ-ММ-ДД ЧЧ:ММ:СС», часовой пояс UTC+3.
error_code	Код ошибки в случае, если СМС не доставлено. <i>error_code</i> = 0, если СМС доставлено или оператор не передал информацию о коде ошибки. Остальные значения кодов ошибок описаны в разделе Список кодов ошибок в поле err отчёта о доставке .

Пример запроса (from SMS Traffic)

```
POST /status_callback_url HTTP/1.1
User-Agent: StatusSender/2.0 Java/21
Host: example.com
Content-type: application/x-www-form-urlencoded

sms_id=15166254108&status=Delivered&delivery_date=2020-05-09 17:04:00
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: text/plain
```

Установка статуса "Проверено"

POST / GET `api.smstraffic.ru/multi.php`

Описание параметров запроса

Имя параметра	Возможные значения	Описание
login *	Текст	Ваш логин в системе SMS Traffic.
password *	Текст	Ваш пароль в системе SMS Traffic.
operation *	Текст	Всегда значение <code>set_checked</code> .
sms_id *	Текст	Список идентификаторов смс, разделённых запятой.

Запросы с `operation=set_checked` являются идемпотентными.



Запрос разрешен для выполнения основной учетной записью и учетным записям с ролью «отправка СМС».

Ограничения запроса:

- Не более 5000 sms_id в списке за запрос
- Сообщения с датой/временем отправки, не ранее 3-х дней от текущей даты.

Пример запроса

```
POST /multi.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
```

```
login=mylogin&password=mypassword&operation=set_checked&sms_id=709031382353069057,709031527266247680,709031527266247681
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: text/xml
```

```
<?xml version="1.0" encoding="windows-1251" ?>
<reply>
  <result>OK</result>
  <code>0</code>
  <sms_ids>
```

```
<sms_id>709031382353069057</sms_id>
<sms_id>709031527266247680</sms_id>
<sms_id>709031527266247681</sms_id>
</sms_ids>
</reply>
```

Пример запроса (with error)

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" encoding="windows-1251" ?>
<reply>
  <result>ERROR</result>
  <code>434</code>
  <description>No such sms_id or this sms_id does not belong to you :
709031382353069057,709031527266247680,709031527266247681</description>
</reply>
```

Проверка баланса

POST / GET `api.smstraffic.ru/multi.php`

Описание параметров запроса

Имя параметра	Возможные значения	Описание
login *	Текст	Ваш логин в системе SMS Traffic.
password *	Текст	Ваш пароль в системе SMS Traffic.
operation *	Текст	Всегда значение <i>account</i> .

Пример запроса

```
POST /multi.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded

login=mylogin&password=mypassword&operation=account
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" ?>
<reply>
  <account>10025</account>
</reply>
```

Управление списками для рассылок

POST / GET `api.smstraffic.ru/list.php`

Описание параметров запроса

Имя параметра	Возможные значения	Описание
login *	Текст	Ваш логин в системе SMS Traffic.
password *	Текст	Ваш пароль в системе SMS Traffic.
operation	status_all	Возвращается информация обо всех списках в аккаунте. Считается значением по умолчанию при отсутствии в запросе параметра <i>operation</i> .
	status	Возвращается информация о конкретном списке.
	add_member	Добавление номеров телефонов в список для рассылки.
	remove_member	Удаление номеров телефонов из списка для рассылки.
member	Список номеров через запятую	В данном параметре указывается список номеров телефонов при совершении запросов с указанием <i>operation=add_member</i> или <i>remove_member</i> . Рекомендуется указывать не более 5000 номеров за один раз.
group_id	Положительное целое число (4 байта)	Уникальный идентификатор списка для рассылок. Используется при совершении манипуляций с ним. Узнать идентификатор списка можно запросом с <i>operation=status_all</i> .

Примеры

Получение всех списков для рассылки:

Пример запроса

```
POST /list.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded

login=mylogin&password=mypassword
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" ?>
<reply>
  <result>OK</result>
  <code>0</code>
  <description>total groups: 4</description>
  <groups>
    <group>
      <id>59353</id> ①
      <name>hello world</name> ②
      <created>2020-05-09 11:46:06</created> ③
      <congratulate>0</congratulate> ④
    </group>
    <group>
      <id>59355</id>
      <name>клиенты</name>
      <created>2020-05-09 11:48:15</created>
      <congratulate>0</congratulate>
    </group>
    <group>
      <id>59357</id>
      <name>test</name>
      <created>2020-05-09 11:48:49</created>
      <congratulate>0</congratulate>
    </group>
  </groups>
</reply>
```

Поля результата:

- ① уникальный идентификатор списка для рассылки;
- ② текстовое имя списка для рассылки. Отображается в личном кабинете;
- ③ дата и время создания списка (в формате «ГГГГ-ММ-ДД ЧЧ:ММ:СС», часовой пояс UTC+3);
- ④ статус поздравления с днём рождения.

Добавление номеров в список:

Пример запроса

```
POST /list.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded
```

```
Login=mylogin&password=mypassword&group_id=59353&operation=add_member&member=79012223344,79082223344
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" ?>
<reply>
  <result>OK</result>
  <code>0</code>
  <description>added or updated members: 2</description>
</reply>
```

Удаление номеров из списка:

Пример запроса

```
POST /list.php HTTP/1.1
Host: api.smstraffic.ru
Content-Type: application/x-www-form-urlencoded

Login=mylogin&password=mypassword&group_id=59353&operation=remove_member&member=7901223344,79082223344
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0" ?>
<reply>
  <result>OK</result>
  <code>0</code>
  <description>added or updated members: 2</description>
</reply>
```

SMPP протокол

Отправку сообщений, получение информации о статусе доставки, а также получение входящих сообщений от абонентов можно осуществлять через SMPP (Short Message Peer to Peer) — стандартный протокол передачи коротких сообщений. Мы используем версию 3.4 протокола. Скачать спецификацию вы можете с [нашего сайта](#).

На нашем СМС-центре можно ограничивать доступ к SMSC по IP-адресу. По умолчанию никакие ограничения по IP не накладываются. Если вы хотите установить ограничения, то необходимо написать соответствующее письмо в адрес технической поддержки, и мы пропишем ваши подключения только на указанных адресах. Помимо конкретных IP адресов мы можем прописать подсети классов А, В и С (/8, /16 и /24 по CIDR классификации). Поддерживается любой вид соединения со следующим ограничением: на один наш хост+system_id не более одного receiver/transceiver.

Если вам необходимо больше соединений и есть возможность увеличить количество receiver'ов, то рекомендуем создать дополнительные департаменты в рамках вашей основной учетной записи. При создании департамента вы автоматически получаете новый system_id.

Рекомендуемая интенсивность передачи — до 20 смс/сек. Для каждого клиента пропускная способность выставляется индивидуально, при превышении пропускной способности отправляется ошибка ESME_RTHROTTLED.

Enquire_links обязательны. Если клиент подключается к нам transmitter'ом, мы ожидаем от него enquire_links и, при их отсутствии в течение двух минут, связь считается нарушенной и принудительно рвется. Если клиент подключается к нам receiver'ом, мы отправляем ему enquire_links каждые 30 секунд и ожидаем от него enquire_link_resps. При их отсутствии в течение двух минут, связь считается нарушенной и принудительно рвется. При подключении transceiver'ом обмен enquire_links ведется с обеих сторон. Рекомендуемый интервал отправки enquire_links в нашу сторону — 30 секунд.

Особенностей PDU нет, они определяются протоколом. Не поддерживаются: query_sm, data_sm, submit_multi_sm, replace_sm, sar, tlv (для SUBMIT_SM).

Таблица 1. Параметры для подключения к нашему СМС-центру (BIND)

Параметр	Описание
Хост	server1.smstraffic.ru (основной сервер) или server2.smstraffic.ru (дублирующий сервер)
Порт	4441
system_id	Логин и пароль для подключения выдаются вашим менеджером. Они совпадают с логином и паролем от личного кабинета.
password	
interface_version	0x34

Параметр	Описание
system_type	Игнорируется нашим SMPP-сервером. Можно оставить пустым.
addr_ton	Игнорируется нашим SMPP-сервером. Можно оставить пустым.
addr_npi	Игнорируется нашим SMPP-сервером. Можно оставить пустым.
address_range	Игнорируется нашим SMPP-сервером. Можно оставить пустым.

Предпочтительно присоединяться к **server1.smstraffic.ru**, так как он является основным, а **server2.smstraffic.ru** — дублирующим.

Таблица 2. Параметры при отправке сообщения (SUBMIT_SM)

Параметр	Описание
source_addr_ton	1 для цифрового отправителя или 5 - для буквенного
source_addr_npi	1 для цифрового отправителя или 0 - для буквенного
source_addr	Альфанумерический отправитель (длина до 11 символов) или цифровой (длина до 15 цифр). См. раздел «Адрес отправителя».
dest_addr_ton	1
dest_addr_npi	1
destination_addr	Номер телефона в международном формате. См. раздел "Формат номера телефона".
esm_class	Согласно спецификации SMPP
protocol_id	Согласно спецификации SMPP
validity_period	Согласно спецификации SMPP
registered_delivery	Согласно спецификации SMPP
data_coding	Согласно спецификации SMPP
sm_length	Согласно спецификации SMPP
short_message	Согласно спецификации SMPP
service_type	Не поддерживается
priority_flag	Не поддерживается
schedule_delivery_time	Не поддерживается
replace_if_present_flag	Не поддерживается
sm_default_msg_id	Не поддерживается

Таблица 3. Формат поля short_message отчёта о доставке (DELIVER_SM)

Field	Size (octets)	Type
Id	10	C-Octet String (Decimal)

Field	Size (octets)	Type
sub	3	C-Octet String Fixed Length (Decimal)
dlvrd	3	C-Octet String Fixed Length (Decimal)
submit date	10	C-Octet String Fixed Length (Decimal)
done date	10	C-Octet String Fixed Length (Decimal)
stat	7	C-Octet String Fixed Length
err	var. max. 3	C-Octet String (Decimal)
text	var. max. 65	C-Octet String (Decimal)

В отчёт о доставке добавляются два TLV:

- *message_state* (0x0427) - в нем передаётся статус доставки сообщения;
- *received_message_id* (0x001e) - указывается идентификатор сообщения, назначенный нашей платформой.

Если отчёт о доставке негативный, добавляется TLV *network_error_code* (0x0423). Байт "Network Type" устанавливается в 0x08. В разделе [«Список кодов ошибок в поле err отчёта о доставке»](#) вы найдете подробное описание возможных ошибок.

Кодировки

Default SMSC alphabet (*data_coding=0*) на наших серверах воспринимается как default GSM alphabet ([GSM 03.38](#)). При отправке сообщений в кодировке GSM 03.38 или latin1 (*data_coding=0* или 3, соответственно) поддерживаются только следующие символы (hex коды символов указаны для кодировки GSM):

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
0.	@	£	\$	¥	è	é	ù	ì	ò	ç		Ø	ø		Å	å
1.		–											Æ	æ	ß	É
2.	SP	!	"	#	¤	%	&	'	()	*	+	,	-	.	/
3.	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4.	i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5.	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ñ	Ü	Ş
6.	ı	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7.	p	q	r	s	t	u	v	w	x	y	z	ä	ö	ñ	ü	à

При использовании *data_coding=0* или 3 поддерживается корректная передача только символов из таблицы выше. По всем остальным значениям *data_coding* не выполняется анализ или модификация исходного текста сообщения (текст передается оператору как

есть).

При отправке сообщений в кодировке GSM 03.38 упаковывать текст сообщения в 7 бит не нужно.

Типичные ошибки

При взаимодействии по протоколу SMPP могут возникать различные ошибки. В данном разделе перечислены наиболее часто встречающиеся из них, а также рассматриваются способы их устранения.

ESME Already in Bound State (0x00000005)

Ошибка ESME_RALYBND возникает при попытке инициировать новое подключение в режимах transceiver или receiver при уже имеющемся активном соединении с таким же system_id на одном нашем сервере в режиме transceiver или receiver. Если необходимо держать несколько соединений в режиме transceiver или receiver (на режим transmitter ограничение не распространяется), можно создать дополнительные учётные записи. Дополнительные учётные записи привязаны к основному аккаунту и используют единый биллинг. Подробное описание этой возможности вы найдете в ["Руководстве по работе с личным кабинетом"](#) (требуется предварительная авторизация в ЛК). Для включения функционала обратитесь к вашему менеджеру.

Throttling error (0x00000058)

Ошибка ESME_RTHROTTLED возникает при превышении максимально допустимой скорости отправки сообщений по SMPP соединению. По умолчанию ограничение составляет 20 СМС/сек. Необходимо настроить своё ПО таким образом, чтобы возвращенные с этой ошибкой сообщения перепосылались через небольшой таймаут, либо следует уменьшить интенсивность передачи сообщений.

Invalid Password (0x0000000E)

Ошибка ESME_RINVPASWD возникает на этапе выполнения операции BIND при указании неверного system-id и(или) пароля. Проверьте корректность логина и пароля. Убедитесь, что логин и пароль совпадают с теми, которые выдал ваш менеджер.

Bind Failed (0x0000000D)

Ошибка ESME_RBINDFAIL возникает на этапе выполнения операции BIND при отсутствии денежных средств на аккаунте. Проверьте состояние баланса на аккаунте.

Invalid Source Address (0x0000000A)

Ошибка ESME_RINVSRCADR возникает при отправке сообщения с некорректным именем отправителя. Возможно, вам не разрешено отправлять сообщения с данным именем, либо по требованию операторов имя отправителя заблокировано на нашей платформе. Убедитесь, что имя отправителя соответствует требованиям и рекомендациям, описанным в разделе [«Адрес отправителя»](#). Проверьте, что оно не содержит лишних пробелов в начале и в конце. Если имя корректно, то свяжитесь со своим менеджером

для прояснения ситуации. Контакты менеджера, а также удобная форма обратной связи находятся в вашем личном кабинете.

Invalid Dest Address (0x0000000B)

Ошибка ESME_RINVDSTADR возникает при отправке сообщения на некорректный номер телефона. Возможно, на вашем аккаунте запрещена отправка сообщений на данное направление, либо по требованию операторов номер телефона заблокирован на нашей платформе. Проверьте, что номер указан в международном формате так, как описано в разделе "[Формат номера телефона](#)". Вместо прямых городских номеров должны быть указаны их федеральные аналоги. Убедитесь, что номер телефона принадлежит оператору, отправка на который прописана в вашем договоре. Принадлежность к оператору можно определить в личном кабинете на [странице определения оператора](#).

UDH-склейка

Для передачи склеенных сообщений (сообщений из нескольких частей) следует использовать UDH-склейку (User Data Header). Это единственно возможный способ передачи длинных сообщений в GSM сетях. UDH является частью тела сообщения (поле *short_message* из обязательных параметров). Существует два вида склейки, которые различаются сигнатурой и размером reference number.

Наиболее часто используемый вариант — reference number размером 1 байт (в примере тело сообщения в hex форме):

05 00 03 D4 03 02 04 3F 04 40 04 38 04 32 04 35 04 42 00 20 00 3A 00 29 ...

длина UDH	сигнатура (IEI)	длина IE	reference number	общее кол-во частей	текущая часть (отсчет идёт от 1)	текст части сообщения
05	00	03	D4	03	02	04 3F 04 40 04 38 04 32 04 35 04 42 00 20 00 3A 00 29 ...

Другой вариант — reference number размером 2 байта (в примере тело сообщения в hex форме):

06 08 04 A7 D4 03 02 04 3F 04 40 04 38 04 32 04 35 04 42 00 20 00 3A 00 29 ...

длина UDH	сигнатура (IEI)	длина IE	reference number	общее кол-во частей	текущая часть (отсчет идёт от 1)	текст части сообщения
06	08	04	A7 D4	03	02	04 3F 04 40 04 38 04 32 04 35 04 42 00 20 00 3A 00 29 ...

При применении 2-х байтовой склейки значительно снижается вероятность неправильной склейки сообщений на телефоне абонента. Под неправильной склейкой подразумевается ситуация, когда, например, первая половина текста передана верно, а вторая - взята из старого, ранее отправленного СМС-сообщения. Такое может происходить из-за малого количества вариантов reference number.

Для того чтобы указать наличие UDH в теле сообщения, необходимо взвести бит UDHI Indicator (седьмой бит) в esm_class. Например, типичное значение esm_class для склеенных сообщений — 0x40.

Пример СМС из двух частей:

05 00 03 AF 02 01 ... тело сообщения ... — первая часть
05 00 03 AF 02 02 ... тело сообщения ... — вторая часть

Из-за того, что невозможно заранее предугадать маршрут доставки каждого отдельно взятого сообщения, используется параметр reference number (в примере это AF). Он позволяет принимающему аппарату определить, что поступившие части относятся к одному и тому же склеенному сообщению. Последние два октета позволяют аппарату склеить сообщения в правильном порядке и определить общее количество частей сообщения для корректной индикации о неполном сообщении.

Правила индикации зависят от конкретной модели телефона. Наиболее часто встречаются два алгоритма индикации:

1. Телефон копит во внутреннем буфере все части сообщения и отображает его на своем экране только после получения всех частей.
2. Телефон сразу начинает отображать поступившие части, а в списке сообщений неполное сообщение отмечается особым значком.

Из-за того, что UDH является частью сообщения, максимальный размер текста уменьшается на длину UDH. Например, для сообщений в семибитной кодировке, максимальная длина текста — 153 символа, а для UCS-2 — 67.

При использовании склейки с 2-х байтовым reference number максимальная длина текста уменьшается до 152 символов в семибитной кодировке и до 66 символов в UCS-2.

SOAP протокол

Отправка сообщений на телефон абонента осуществляется путем обращения к веб-сервису и вызовом необходимых методов. Описание типов и поддерживаемых методов представлено в WSDL.

WSDL для сервиса находится по адресу: <https://soap.smstraffic.ru/soap.wsdl>.

Адрес сервера: <https://soap.smstraffic.ru/soap.php>.

Поведение параметров, описанных в WSDL, соответствует поведению одноимённых параметров [HTTP API v1 \(XML\) протокола](#).

В WSDL представлены следующие методы:

- **SendBulkSms** — отправка одного сообщения на несколько номеров одним запросом;
- **SendIndividualSms** — отправка индивидуального сообщения нескольким получателям;
- **GetSmsStatus** — запрос статуса доставки;
- **GetBalance** — запрос баланса учётной записи.

Для отправки сообщений предназначены методы **SendBulkSms** и **SendIndividualSms**.

В запросах должен присутствовать элемент *Options* - в нем указываются логин и пароль к аккаунту, а также набор необязательных опций:

Элемент	Описание
Login *	Ваш логин в системе SMS Traffic.
Password *	Ваш пароль в системе SMS Traffic.
Originator	Имя отправителя. Если не указано, берётся имя отправителя по умолчанию из настроек аккаунта. Более подробно описано в параметре originator HTTP API v1 (XML) протокола.
WantSmsIds	Если установлен в true, то в ответ вернутся идентификаторы, присвоенные каждому сообщению. Значение по умолчанию — false. Более подробно описано в параметре want_sms_ids HTTP API v1 (XML) протокола.
IgnorePhoneFormat	Отключает проверку корректности номера телефона. Значение по умолчанию — false. Более подробно описано в параметре ignore_phone_format HTTP API v1 (XML) протокола.

Элемент	Описание
LatinEncoding	Если установлено в true, то предполагается отправка СМС в латинице или транслите. Символы кириллицы будут транслитерироваться в латиницу. Максимальное количество символов в одном СМС-сообщении — 160 (153 для склеенного сообщения). Если установлено в false, то сообщение будет отправлено в кодировке UCS-2, при этом максимальное количество символов в одном СМС-сообщении — 70 (67 для склеенного сообщения). Значение по умолчанию — false.
Timeout	Время жизни СМС (в секундах). Значение по умолчанию — 86400 (24 часа). Более подробно описано в параметре timeout HTTP API v1 (XML) протокола.
Flash	Отправлять сообщение как flash-SMS. Значение по умолчанию — false. Более подробно описано в параметре flash HTTP API v1 (XML) протокола.
StartDate	Дата и время отправки СМС. По умолчанию сообщения отправляются немедленно. Более подробно описано в параметре start_date HTTP API v1 (XML) протокола.
MaxParts	Максимальное количество частей, на которые при необходимости будет разбит текст сообщения. Значение по умолчанию — 255. Более подробно описано в параметре max_parts HTTP API v1 (XML) протокола.
TwoByteConcat	Параметр позволяет указать способ UDH-склейки. Значение по умолчанию — false. Более подробно описано в параметре two_byte_concat HTTP API v1 (XML) протокола.

В ответе каждого метода присутствует элемент *Status*. Он содержит статус обработки всего запроса.

Элемент	Описание
IsSuccess	true, если запрос обработан успешно и false - в противном случае.
Code	Код ошибки. Если указан 0, то запрос был обработан без ошибок. Возможные коды ошибок указаны в таблице коды ошибок API .
Description	Текстовое описание ошибки. Значение этого элемента полезно записать в логи.

Если элемент *WantsSmsIds* был установлен в true, то для каждого абонента в ответе также будет присутствовать элемент *MessageInfo* со структурой, приведенной ниже:

Элемент	Описание
Phone	Номер телефона абонента
ProcessResult	Текстовый статус обработки сообщения
SmsId	Идентификатор СМС-сообщения (в случае, если <i>ProcessResult=Accepted</i>)

В свою очередь, элемент *ProcessResult* может содержать следующие значения:

Элемент	Описание
Accepted	Сообщение было принято и обработано.
Unroutable	Сообщение было принято, но не обработано (например, по причине отсутствия роутинга на оператора абонента).

Метод SendBulkSms

Метод SendBulkSms позволяет отправить одинаковый текст сообщения на список номеров телефонов.

Пример запроса

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <BulkSms>
      <Options>
        <Login>example</Login>
        <Password>p@ssw0rd!</Password>
      </Options>
      <Message>hello world</Message>
      <Phone>78003336655</Phone>
      <Phone>78002224477</Phone>
      <Phone>78001114444</Phone>
    </BulkSms>
  </env:Body>
</env:Envelope>
```

Пример ответа

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <SmsResponse>
      <Status>
        <IsSuccess>>true</IsSuccess>
        <Code>0</Code>
        <Description>queued 3 messages</Description>
      </Status>
    </SmsResponse>
  </env:Body>
</env:Envelope>
```

Метод SendIndividualSms

Метод SendIndividualSms позволяет отправить каждому получателю своё индивидуальное сообщение.

Пример запроса

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <IndividualSms>
      <Options>
        <Login>example</Login>
        <Password>p@ssw0rd!</Password>
      </Options>
      <Pairs>
        <Phone>78003336655</Phone>
        <Message>individual</Message>
      </Pairs>
      <Pairs>
        <Phone>78002224477</Phone>
        <Message>message</Message>
      </Pairs>
      <Pairs>
        <Phone>78001114444</Phone>
        <Message>by phone</Message>
      </Pairs>
    </IndividualSms>
  </env:Body>
</env:Envelope>
```

Пример ответа

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <SmsResponse>
      <Status>
        <IsSuccess>>true</IsSuccess>
        <Code>0</Code>
        <Description>queued 3 messages</Description>
      </Status>
    </SmsResponse>
  </env:Body>
</env:Envelope>
```

Метод GetSmsStatus

Метод GetSmsStatus возвращает статус доставки сообщения по *sms_id*.

Пример запроса

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <StatusRequest>
      <Login>example</Login>
      <Password>p@ssw0rd!</Password>
      <SmsId>27371658818</SmsId>
      <SmsId>1</SmsId>
    </StatusRequest>
  </env:Body>
</env:Envelope>
```

Пример ответа

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <StatusResponse>
      <Status>
        <IsSuccess>>true</IsSuccess>
        <Code>0</Code>
        <Description></Description>
      </Status>
      <SmsInfo>
        <SmsId>27374567618</SmsId>
        <Error></Error>
        <SubmissionDate>2020-05-09T15:19:00</SubmissionDate>
        <SendDate>2020-05-09T15:19:00</SendDate>
        <LastStatusChangeDate>2020-05-09T15:19:00</LastStatusChangeDate>
        <Status>Expired</Status>
      </SmsInfo>
      <SmsInfo>
        <SmsId>1</SmsId>
        <Error>no such message or this message does not belong to you</Error>
      </SmsInfo>
    </StatusResponse>
  </env:Body>
</env:Envelope>
```

Метод GetBalance

Метод GetBalance возвращает текущий баланс аккаунта.

Пример запроса

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <BalanceRequest>
      <Login>texample</Login>
      <Password>p@ssw0rd!</Password>
    </BalanceRequest>
  </env:Body>
</env:Envelope>
```

Пример ответа

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns="http://soap.smstraffic.ru/soap.wsdl">
  <env:Body>
    <BalanceResponse>
      <Status>
        <IsSuccess>true</IsSuccess>
        <Code>0</Code>
        <Description></Description>
      </Status>
      <Amount>52782.35</Amount>
    </BalanceResponse>
  </env:Body>
</env:Envelope>
```

SMTP протокол (вариант 1)

Для того чтобы отправить СМС на телефон абонента посредством электронной почты, необходимо написать (или ваша программа должна сгенерировать) письмо на адрес: login@corp.smsmail.ru, где login – имя, выданное вам после регистрации. Номер телефона следует указать в теме письма (subject), а само сообщение – в теле письма.

Сообщение можно написать кириллицей - в этом случае, в зависимости от настроек вашего аккаунта, система либо сама перекодирует его в транслит (эта настройка действует по умолчанию), либо отправит в кодировке UCS-2.

Для изменения этой настройки воспользуйтесь страницей [настроек](#) в личном кабинете.

Пример:

```
From: yourname@yourdomain.ru  
To: yourlogin@corp.smsmail.ru  
Subject: 78000000001
```

You need to pay an invoice of \$17.65. For inquiries, call 1234567.

Телефон абонента можно указать одним из двух способов:

- В теме письма - в этом случае почта отправляется на адрес типа login@corp.smsmail.ru, как описано выше.
- В адресе - разделяя номер телефона и логин точкой. Например, 78000000001.login@corp.smsmail.ru.

Пример:

```
From: yourname@yourdomain.ru  
To: 78000000001.yourlogin@corp.smsmail.ru  
Subject: ne imeyet znacheniya
```

You need to pay an invoice of \$17.65. For inquiries, call 1234567.



Обратите внимание - SMTP v1 не предназначен для массовых рассылок. Для массовых рассылок используйте протоколы SMPP, HTTP API или личный кабинет. Рекомендуемое ограничение — не более 100 писем в минуту.

По желанию клиента можно ограничить отсылку таких писем с конкретного SMTP сервера (рекомендуется). Аутентификация осуществляется по заголовкам Received. Для того чтобы установить защиту, необходимо отправить тестовое электронное сообщение на tech.support@smstraffic.ru через тот SMTP сервер, который будет использоваться для

отправки рабочих сообщений. В теле письма укажите, для какого аккаунта вы хотите установить защиту. Также, по желанию клиента можно ограничить список телефонов, на которые возможна отправка. Список можно задать на странице [настроек](#) в личном кабинете. По умолчанию принимаются сообщения на любые телефоны.

Мы настоятельно рекомендуем использовать хотя бы один тип защиты — либо по заголовкам SMTP, либо с помощью ограничения списка получателей. Возможно также полностью запретить прием сообщений через email и пользоваться только протоколами HTTP API или SMPP. Установить такой запрет можно на странице [настроек](#) в личном кабинете. Для всех новых аккаунтов отправка по email описанным выше способом по умолчанию отключена. При отправке по email невозможно динамически задавать имя или телефон отправителя (поле *originator* при передаче по HTTP), поэтому в качестве отправителя всегда будет подставлено одно и то же значение, по умолчанию. Это значение можно исправить на странице настроек вашего личного кабинета. Если необходимо динамически изменять отправителя, рекомендуем использовать протоколы HTTP API, SMPP или SMTP вариант 2.

SMTP протокол (вариант 2)

Существует другой способ отправлять СМС-сообщения через email и при этом пользоваться практически всеми возможностями, которые предоставляет протокол HTTP API. Отправьте email на адрес multi@smtp2.smsmail.ru. В теле сообщения перечислите поля и их значения (согласно описанию протокола HTTP), разделенные двоеточием. Каждая пара "поле: значение" должна быть на отдельной строке и отделена от соседних пар по крайней мере одной пустой строкой. Тема сообщения не имеет значения.

Пример:

```
From: yourname@yourdomain.ru
To: multi@smtp2.smsmail.ru
Subject: ne imeyet znacheniya

login: yourlogin

password: yourpass

phones: 78000000001

message: You need to pay an invoice of $17.65. For inquiries, call 1234567.

originator: MyCompany
```

Ограничение: в поле *message* нельзя использовать два символа перевода строки подряд (такое сочетание символов будет воспринято как разделитель полей). Дополнительно можно использовать специальный параметр *reply_to_email* со значением 1, как в примере ниже:

```
From: yourname@yourdomain.ru
To: multi@smtp2.smsmail.ru
Subject: ne imeyet znacheniya

login: yourlogin

password: yourpass

phones: 78000000001

message: You need to pay an invoice of $17.65. For inquiries, call 1234567.

originator: MyCompany

reply_to_email: 1
```

В этом случае на email-адрес отправителя будет выслан ответ, такой же как при использовании протокола HTTP API v1 (XML).

Пример:

```
<?xml version="1.0"?>
<reply>
  <result>OK</result>
  <code>0</code>
  <description>queued 1 messages</description>
  <message_infos>
    <message_info>
      <phone>78000000001</phone>
      <sms_id>1014190631</sms_id>
    </message_info>
  </message_infos>
</reply>
```



Обратите внимание - данный метод не предназначен для массовых рассылок. Для массовых рассылок используйте протоколы SMPP, HTTP API или личный кабинет. Рекомендуемое ограничение — не более 100 писем в минуту и не более 10000 СМС в одном письме.

Список кодов ошибок в поле *err* отчёта о доставке

Код	Описание на английском	Описание на русском
1	The subscriber is absent or out of a coverage	Абонент недоступен или отключен.
2	Call barred service activated	У абонента включен запрет на прием сообщений или абонента заблокировал оператор (возможно, в связи с отрицательным балансом).
3	Unknown subscriber	Номер телефона не существует или не обслуживается.
4	Memory capacity exceeded	Память телефона абонента переполнена.
5	Equipment protocol error	Аппаратная ошибка телефона абонента.
6	Teleservice not provisioned	Сервис коротких сообщений не предоставляется.
7	Facility not supported	Аппарат абонента не поддерживает прием коротких сообщений.
8	Subscriber is busy	Аппарат абонента занят операцией, препятствующей получению короткого сообщения.
9	Roaming restrictions	Абонент находится в роуминге.
10	Timeout	Время ожидания ответа от SMSC абонента истекло.
11	SS7 routing error	Внутренняя ошибка маршрутизации.
12	Internal system failure	Внутренняя ошибка системы.
13	SMSC failure	Ошибка коммутатора (внутренняя ошибка передачи данных).
14	Illegal subscriber	Блокировка оператором или незарегистрированный пользователь.
15	Message queue full	Очередь сообщений для абонента со стороны оператора переполнена.
16	Invalid source address	Некорректное имя отправителя
17	Unroutable direction	Сообщение не маршрутизируемо.

Входящие СМС-сообщения

Если у вас подключена услуга "Входящий номер", то вы можете принимать СМС-сообщения от ваших сотрудников и клиентов. Суть услуги в том, что абонент отправляет СМС на специальный номер, настроенный в СМС-центре, после чего это сообщение пересылается вам либо по email, либо на ваше HTTP-приложение, либо по SMPP-протоколу.

Пересылка входящих сообщений на email

Для получения входящих сообщений по email, пришлите нам адрес, на который вы хотите получать входящие, и мы пропишем этот адрес в настройках. В дальнейшем вы сможете изменить этот адрес на странице [настроек](#) в личном кабинете.

Пересылка входящих сообщений на HTTP(S)-скрипт

Для пересылки входящих сообщений на ваше HTTP-приложение и их автоматической обработки, вам необходимо установить на своем веб-сервере скрипт, который будет принимать методом GET или POST от нашего сервера следующие данные:

- *phone* – номер телефона абонента;
- *message* – текст сообщения (по умолчанию кодировка Windows-1251);
- *sms_id* – уникальный идентификатор сообщения. Передается для исключения дублей: если *sms_id* повторяется, то необходимо дать тот же ответ, что и при первом запросе.

На успешный запрос нашего сервера ваш скрипт должен вернуть HTTP статус код 200.

Вы можете самостоятельно редактировать параметры скрипта на странице [настроек](#) в личном кабинете.

В случае передачи данных *POST*-запросом, достаточно указать путь к вашему скрипту.

В случае передачи данных *GET*-запросом, необходимо придерживаться шаблона следующего вида:

`Путь_к_вашему_скрипту?message={{message}}&phone={{phone}}&sms_id={{sms_id}}`

Также вы можете прислать нам URL вашего скрипта, чтобы мы настроили его вызов при поступлении любого входящего сообщения.

Пересылка входящих сообщений по SMPP

Для пересылки входящих (MO) сообщений по протоколу SMPP необходимо в настройках входящего номера в личном кабинете выбрать опцию "SMPP" (установлена по умолчанию). Входящие сообщения будут поступать только в режимах подключения Receiver и

Transceiver.

Запрос списка входящих сообщений

Для получения списка входящих сообщений вызывается скрипт api.smstraffic.ru/multi.php методом GET или POST.

Описание параметров запроса

Имя параметра	Описание
login *	Ваш логин в системе SMS Traffic.
password *	Ваш пароль в системе SMS Traffic.
operation *	Всегда значение <i>incoming</i> .
from_date	Дата, начиная с которой необходимо получить список. Формат: «ГГГГ-ММ-ДД» или «ГГГГ-ММ-ДД ЧЧ:ММ:СС», часовой пояс UTC+3.
from_phone	Телефон, с которого приходили сообщения.
count	Максимальное количество сообщений, выдаваемое за один запрос.
want_sms_ids	Если want_sms_ids=1, то ответный XML включает идентификатор СМС. По умолчанию - 0.

В качестве ответа скрипт отдает XML с информацией о сообщении. Например:

```
<?xml version="1.0" ?>
<reply>
  <count>125</count>
  <messages>
    <message>
      <from_phone>79031234567</from_phone>
      <text>sms message text</text>
      <send_date>2020-05-09 01:12:59</send_date>
      <delivery_date>2020-05-09 01:13:05</delivery_date>
      <sms_id>5269156759</sms_id>
    </message>
    ...
    <message>
      <from_phone>79037654321</from_phone>
      <text>sms message</text>
      <send_date>2020-05-09 06:02:13</send_date>
      <delivery_date>2020-05-09 06:02:18</delivery_date>
      <sms_id>5269156775</sms_id>
    </message>
  </messages>
</reply>
```

Срок хранения и выдачи сообщений по данному запросу не должен превышать двух дней.

Если для вас важна оперативность получения сообщений, используйте возможность автоматического уведомления по HTTP протоколу, см. раздел [Пересылка входящих сообщений на HTTP\(S\)](#).

Короткий номер

Для получения сообщений, поступивших на ваш короткий номер, необходимо установить на своем веб-сервере скрипт, который будет принимать методом GET или POST следующие параметры:

- *phone* – телефонный номер абонента;
- *message* – текст его запроса (по умолчанию, кодировка Windows-1251);
- *shortcode* – короткий номер;
- *password* – пароль для защиты от несанкционированного использования вашего скрипта;
- *sms_id* – уникальный идентификатор сообщения. В случае повторного поступления запроса с одним и тем же *sms_id* необходимо дать абоненту точно такой же ответ. Пришлите нам URL вашего скрипта, и мы настроим его вызов после каждого поступления запроса на ваш короткий номер. Скрипт должен обработать запрос и дать ответ, который мы без изменений перешлем абоненту по СМС. Ответ может содержать, например, запрошенную текстовую информацию, подтверждение принятия запроса или ссылку на war-сайт, по которой абонент может скачать запрошенный контент. Ответ не должен содержать никакого форматирования. Вы можете самостоятельно редактировать параметры вызова скрипта на странице [настроек](#) в личном кабинете. Необходимо придерживаться шаблона следующего вида:

Путь_к_вашему_скрипту?message={{message}}&phone={{phone}}&sms_id={{sms_id}}&shortcode={{shortcode}}.

Коды ошибок API

Ошибки с кодами 1XX относятся только к HTTP API v2. Остальные коды используются протоколами HTTP API, SMTP, SOAP.

Код	Описание
103	Пустое поле destinations
104	Значение trackingData слишком длинное
105	Запрос не был принят по ограничению restrictiveId
106	Значение restrictiveId слишком длинное
401	Не указан логин
402	Не указан пароль
403	Не указаны номера телефонов
404	Несовместимые параметры запроса
405	Не указан текст сообщения
407	Не указан ни один телефон
408	Неподдерживаемый тип сообщения: <i>"тип_сообщения"</i>
409	Не указан udh
410	Нельзя использовать параметр <i>max_parts</i> : автоматическая разбивка бинарных сообщений не поддерживается.
411	Неверный логин или пароль
412	Неверный IP
413	Такой группы не существует: <i>"имя_группы"</i>
414	В группе нет ни одного телефона
415	Недостаточно средств
416	Неверный формат даты начала рассылки: <i>"дата_старта_рассылки"</i> .
417	Дата начала рассылки <i>"дата_старта_рассылки"</i> находится в прошлом.
418	Идентификаторы не предоставляются для отложенных сообщений.
419	Вам не разрешено использовать данный маршрут.
420	Сообщение <i>"текст_сообщения"</i> слишком длинное.
421	Имя отправителя слишком длинное
422	Не указан телефон в строке <i>"номер_строки"</i> : <i>"строка"</i> .
423	Пустое сообщение для телефона <i>"номер_телефона"</i> .

Код	Описание
424	Сообщение " <i>текст_сообщения</i> " для телефона " <i>номер_телефона</i> " слишком длинное.
425	Номер телефона " <i>номер_телефона</i> " слишком короткий. Ни одно сообщение не было отправлено.
426	Номер телефона " <i>номер_телефона</i> " слишком длинный. Ни одно сообщение не было отправлено.
427	" <i>номер_телефона</i> ": неверная длина номера телефона. Ни одно сообщение не было отправлено.
428	" <i>номер_телефона</i> ": неверный формат номера телефона. Ни одно сообщение не было отправлено.
429	" <i>номер_телефона</i> ": неподдерживаемый оператор. Ни одно сообщение не было отправлено.
430	" <i>номер_телефона</i> ": неверный номер телефона. Ни одно сообщение не было отправлено.
431	Телефон " <i>номер_телефона</i> " не подписан на рассылку. Ни одно сообщение не было отправлено.
432	Заблокированный номер телефона: " <i>номер_телефона</i> ". Ни одно сообщение не было отправлено.
433	Не указан параметр <i>sms_id</i> .
434	Такого сообщения нет или оно вам не принадлежит.
435	Невозможно отменить сообщение " <i>sms_id</i> ".
436	Отправитель " <i>отправитель</i> " запрещен.
437	Сообщение превышает 160 символов после транслитерации " <i>текст_сообщения</i> ".
438	В сообщении найден шаблон, но не задана ни одна группа.
439	Вы не можете отправлять SMS-сообщения через HTTP.
440	Параметр " <i>phones</i> " не задан или задан некорректно.
441	Неверный формат файла параметров.
442	Неверное число параметров.
501	Время окончания рассылки в прошлом.
502	Время начала рассылки больше времени окончания рассылки.
503	Время старта продолжения рассылки должно быть раньше времени приостановки рассылки.
504	Невозможно создание рассылки с отложенным более чем на 30 дней стартом.

Код	Описание
1000	Временные проблемы на сервере.

Справка по статусам СМС сообщений

Статус по API	Статус в личном кабинете	Описание
Нет статуса (пустая строка)	Нет данных	Ожидает отправки. Этот статус временный и через некоторое время обязательно изменится. Если у сообщения есть дата отправки, это значит, что еще не получено никаких данных от SMS-центра о статусе сообщения. Если у сообщения нет даты отправки, это значит, что оно ещё доставляется. Обычно это происходит при отложенной отправке, когда время отправки данного сообщения еще не подошло.
Buffered SMSC	Доставляется	<p>Делаются попытки доставить сообщение. Этот статус временный и через некоторое время обязательно изменится.</p> <p>Возможные причины:</p> <ol style="list-style-type: none"> 1. Телефон выключен или находится вне зоны действия сети. 2. Память для сообщений в телефоне переполнена. 3. Еще не получен актуальный статус сообщения от SMS-центра.
Delivered	Доставлено	Сообщение доставлено на телефон абонента. Только данный статус означает 100% доставку сообщения на телефон абонента, но не означает, что сообщение было прочитано. Статус может не являться финальным, если канал отправки поддерживает статус " Клик " или " Прочитано ".
Non Delivered	Не доставлено	<p>Не удалось доставить сообщение на телефон абонента, дальнейшие попытки доставки предприниматься не будут.</p> <p>Возможные причины недоставки:</p> <ol style="list-style-type: none"> 1. Несуществующий номер. 2. Абонент заблокирован за неуплату. 3. У абонента установлен "запрет вызова".

Статус по API	Статус в личном кабинете	Описание
Rejected	Отказ в передаче	<p>Сообщение не было доставлено. Сообщение было отвергнуто SMS-центром.</p> <p>Возможные причины:</p> <ol style="list-style-type: none"> 1.Неправильный номер абонента. 2.Неправильный формат имени отправителя. 3.Неправильный формат сообщения. 4.Отвергнуто спам-фильтром из-за того, что зафиксировано слишком много попыток отправить одно и то же сообщение на один и тот же номер (более 4 за час или более 8 за 12 часов).
Expired	Просрочено	<p>Предпринималось несколько попыток доставить сообщение на телефон абонента, но они не удались в течение срока жизни сообщения.</p> <p>Возможные причины:</p> <ol style="list-style-type: none"> 1.Телефон был выключен или находился вне зоны действия сети в течении срока жизни сообщения. 2.Память для сообщений в телефоне была переполнена, и абонент не освободил память в течение всего срока жизни сообщения. 3.Многочисленные сбои в оборудовании сотовой сети при попытках передать сообщение абоненту.
Deleted	Удалено	<p>Сообщение было удалено из SMS-центра.</p> <p>Возможные причины:</p> <ol style="list-style-type: none"> 1.Сообщение не может быть маршрутизируемо ввиду наличия ограничений на стороне оператора. 2.В процессе отправки сообщения произошла ошибка, препятствующая дальнейшим попыткам доставки сообщения до целевого абонента. 3.В случае подключенной услуги по сокращению ссылок, статус может быть присвоен тем сегментам, что фактически не были отправлены в сеть оператора ввиду уменьшения фактической длины SMS сообщения.

Статус по API	Статус в личном кабинете	Описание
Unknown status	Неизвестный статус	От SMS-центра не был получен статус доставки сообщения. Возможные причины: 1.Сбой на оборудовании сотовой сети. 2.На данном направлении не предоставляется информация о статусах.
Clicked	Клик	Может быть присвоен сообщению в том случае, если на аккаунте подключена услуга и канал отправки поддерживает данный статус. Статус присваивается СМС сообщению после перехода абонентом по модернизированной платформой SMS Traffic ссылке, имеющейся в направленном абоненту сообщении.
Checked	Проверено	Абонент передал информацию из текста полученного им сообщения компании, отправившей сообщение
Read	Прочитано	Означает, что абонент передал тем или иным образом информацию из текста полученного им сообщения компании, отправившей сообщение. Например, написал в поле для ввода кода на сайте компании присланные в тексте сообщения цифры.

Любое сообщение достигает окончательного статуса как правило не позднее чем через сутки после отправки.

Всё статусы, кроме "**Нет данных**" и "**Доставляется**", являются окончательными.

История версий

Дата изменения	Версия документа	Описание
2025-12-18	2.3	Добавлен блок HTTP API v2. Удален параметр <code>gar</code> в HTTP API, SOAP.
2025-02-20	2.2.3	Минорное изменение.
2024-04-23	2.2.2	Минорное изменение.
2023-08-21	2.2.1	Минорное изменение. В связи с обновлением интерфейса личного кабинета.
2023-07-06	2.2	Новая операция для HTTP протокола: установка статуса "Проверено".
2021-09-29	2.1.4	Минорное изменение. Исправлена/добавлена формулировка типичных ошибок (<i>Invalid Password, Bind Failed</i>).
2021-06-17	2.1.3	Минорное изменение. Дополнение кода ошибки канала СМС (код 17).
2021-05-24	2.1.2	Минорные изменения. Убрана неактуальная информация (о длине пароля).
2020-05-09	2.1.1	Минорные изменения. Убрана неактуальная информация.
2019-09-18	2.1	Изменен формат документации. Добавлены перекрестные ссылки.
2019-03-28	1.90	Изменены названия статусов для русской версии ЛК.
2019-02-19	1.89	Дополнена документация по SOAP протоколу.
2017-03-21	1.88	Добавлен параметр <code>error_code</code> для автоматической передачи статусов СМС.
2016-12-07	1.87	Добавлены коды ошибок 503 и 504 для протокола HTTP(S).
2016-07-19	1.86	Добавлен код ошибки 16 для протокола SMPP.
2015-12-10	1.85	Переработана структура документа. Добавлены новые разделы, расширены старые. Новый протокол SOAP. Изменены адреса для доступа к HTTP(S) API.
2012-09-25	1.84	Добавлено подробное описание функционала автоматического получения статусов.
2012-06-08	1.81	Добавлен параметр <code>want_sms_ids</code> для <code>operation=incoming</code> при работе по протоколу HTTP(S).

Дата изменения	Версия документа	Описание
2012-03-23	1.80	Добавлено подробное описание особенностей работы по SMPP протоколу.
2012-01-22	1.77	Сервер server2.smstraffic.ru указан как основной при работе по протоколу SMPP.
2011-12-29	1.76	Добавлены несколько новых кодов ошибок при работе по протоколу HTTP(S).
2011-10-04	1.74	Добавлен параметр timeout в протокол HTTP(S).
2011-09-27	1.72	Описание ограничений при работе с протоколами SMTP, добавлен код ошибки 440 для работы по протоколу HTTP(S).
2011-08-12	1.69	Коррекция описания работы с параметром want_sms_ids для протокола HTTP(S).
2010-12-14	1.63	IP адреса заменены на адреса хостов для усиления отказоустойчивости.
2010-01-11	1.55	Добавлен новый список кодов ошибок для протокола HTTP(S).
2009-07-06	1.50	Дополнено описание протокола SMPP.
2009-05-14	1.48	Добавлен протокол SOAP.
2009-03-31	1.47	Изменены IP адреса для работы по SMPP протоколу.
2008-12-02	1.45	Добавлено значение rus=5 в протокол HTTP(S).
2007-11-23	1.39	Добавлен параметр flash в протокол HTTP(S).
2007-10-28	1.37	Добавлены примеры использования протоколов.
2007-07-18	1.35	В протоколе SMTP v.2. изменен адрес назначения на multi@smtp2.smsmail.ru .
2007-06-8	1.31	Добавлен параметр start_date в протокол HTTP(S).
2007-05-29	1.28	Добавлено описание дублирующего сервера.
2006-10-24	1.19	Добавлена возможность запрашивать список входящих сообщений по протоколу HTTP(S).
2006-08-15	1.18	Добавлена возможность запрашивать состояние счета по протоколу HTTP(S).
2006-08-08	1.17	Добавлен параметр wap_push_url в протокол HTTP(S).
2006-06-30	1.13	Уточнено описание параметра originator протокола HTTP(S). Добавлен новый протокол — SMTP v2.
2006-06-28	1.9	Добавлен параметр gap в протокол HTTP(S).

Дата изменения	Версия документа	Описание
2006-03-23	1.6	Добавлены параметры <i>max_parts</i> и <i>UDH</i> в протокол HTTP(S).
2005-04-18	1.5	Добавлена возможность передачи статуса сообщения методом вызова стороннего скрипта.
2005-02-21	1.3	Добавлены дополнительные переменные для протокола HTTP(S).
2004-08-02	1.1	Документ создан.