

# SMSimple USSD

---

## Описание и интерфейс API

### Введение

---

Применительно к сообщениям USSD используется понятие сессии (USSD-сессии). Сессия состоит из последовательности связанных друг с другом сообщений по принципу вопрос-ответ-вопрос-ответ... и.т.д. В дальнейшем вместо термина «вопрос» будет использоваться термин «меню».

Меню — это текст содержащий полезную информацию и/или инструкции по продолжению сессии. Примером меню может быть такой текст:

```
На Вашем счете 100000р.  
1 — перевод  
2 — операции  
3 — счета
```

Примером работы USSD-сессий может служить SIM-меню, имеющееся в любом мобильном телефоне. Однако, в отличие от SIM-меню, где инициатором начала сессии является пользователь мобильного телефона, USSD-сессии SMSimple иницируются не получателем, а отправителем, то есть Вами.

Возможны две кодировки меню:

GSM-7 — «латиница», 150 символов на меню  
UCS2 — «юникод» UTF-16, 60 символов на меню

Ответы пользователей возможны только латиницей и цифрами. Цифр обычно бывает вполне достаточно.

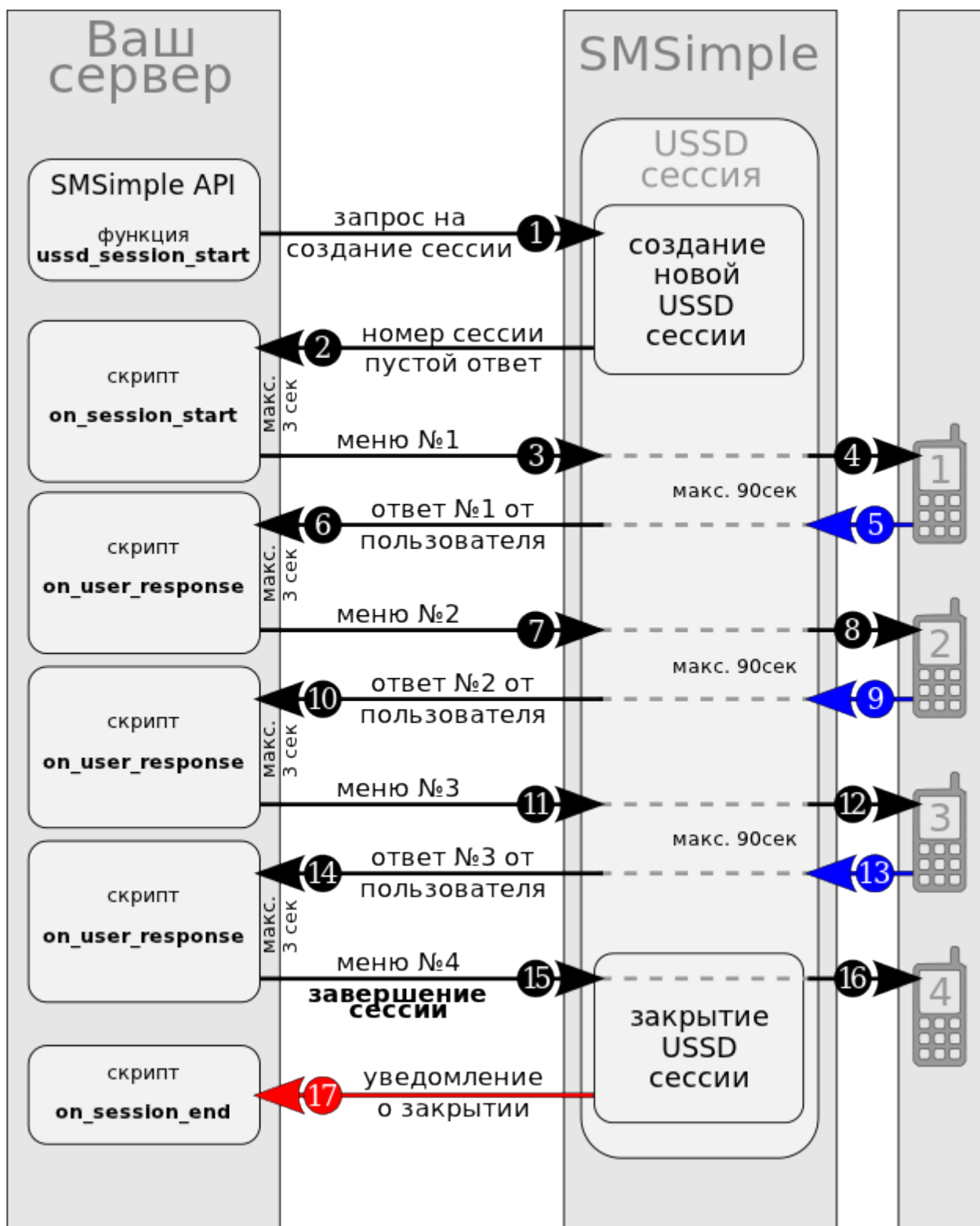
Для старта новой сессии отправляется запрос на старт сессии с указанием номера мобильного телефона и кодировки (GSM-7 / UCS2). Далее в открытую сессию отправляются первичное меню-приглашение. Последующие меню отправляются в сессию в качестве «ответов на ответы» с мобильного телефона получателя. Таким образом меню может создаваться каждый раз динамически исходя из предыдущих ответов. Каждая новая сессия получает уникальный номер, к которому идёт привязка всех передаваемых в сессию меню и приходящих от получателей ответов на них. Для строгой привязки к внутренним данным приложения кроме номера телефона получателя имеется возможность передать опциональный параметр (например какой-то внутренний номер), который будет сопровождать каждый ответ пользователя.

### API

---

Для ведения диалога в рамках USSD-сессии используется механизм вызова пользовательского скрипта, который должен достаточно быстро (не более 2-3 сек) возвращать ответ в формате XML с новым меню и флагом на завершение сессии.

На схеме показано течение USSD-сессии.



Комментарии к схеме.

- Стрелки пронумерованы в хронологическом порядке.
- Скрипты `on_session_start` и `on_user_response` имеют одинаковый набор входных параметров и возвращают одинаковый по виду результат. Это может быть один и тот же скрипт. Отличие состоит лишь в том, что `on_server_start` всегда вызывается

первым в сессии, только один раз и в него передаётся пустой текст ответа. Создавать два скрипта или один — зависит от удобства реализации.

- Скрипт `on_session_end` не возвращает ничего. Он вызывается для информирования о фактическом закрытии сессии с указанием кода завершения.
- Количество передаваемых меню на рисунке взято равным 4 исключительно для примера, количество шагов может быть любым. Ограничение количества меню зависит от конкретного оператора.
- Меню №4 — это последнее, что будет показано на экране телефона в течение этой сессии. В этом меню телефон уже не предложит ввести ответ, так как сессия закрыта.
- На схеме показано обычное течение USSD-сессии. Однако, если пользователь не отвечает на очередное меню (переход по синей стрелке не происходит в течение 90 секунд), то осуществляется переход сразу к красной стрелке и сессия завершается по таймауту.

Для начала сессии используется вызов функции

```
SMSimple::ussd_session_start($phone,$optional=null,$encoding='GSM-7')
```

из SMSimple PHP API (`smsimple.class.php`). Телефон `$phone` является обязательным параметром, в формате 79991234567. Необязательный параметр `$optional` используется для привязки вызова начала сессии к самой сессии когда номера телефона для этих целей недостаточно. При наличии этого параметра, он будет передаваться во все вызываемые скрипты наравне с номером сессии и прочими параметрами. Параметр `$encoding` может принимать два значения: `GSM-7` (значение по умолчанию) и `UCS2`. При `UCS2` сессия будет использовать кодировку `UTF16`, дающую возможность использовать русские буквы, однако уменьшающую объём текста меню до 60 символов (примерно, точные ограничения зависят от конкретного оператора сотовой связи; например, для Билайна, это 65 символов).

После формирования сессии на первой итерации диалога с пользователем будет вызван скрипт `on_session_start`, на второй и последующих - `on_user_response`, методом `GET`<sup>\*)</sup> с параметрами сессии и очередного ответа, например:

```
on_session_start?session_id=<номер_сессии>&phone=<номер_телефона>
&optional=<optional>&encoding=<encoding>&text=<введённый_текст>
```

```
on_user_response?session_id=<номер_сессии>&phone=<номер_телефона>
&optional=<optional>&encoding=<encoding>&text=<введённый_текст>
```

где

`<номер_сессии>` — номер вновь созданной USSD-сессии,  
`<номер_телефона>` — номер телефона, с которым установлена сессия,  
`<введённый_текст>` — текст, введенный на телефоне в ответ на предыдущее меню,  
`<optional>` — второй аргумент, переданный при вызове `ussd_session_start`,  
`<encoding>` — `GSM-7` или `UCS2`.

Ответом обоих скриптов должен быть документ в формате XML вида

---

\*) подробнее см. в разделе «Подключение» в конце документа

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <menu>Текст меню, который будет показан пользователю</menu>
  <close>true/false</close>
</response>
```

где в `<close>` стоит одно из двух значение `true` или `false`, отвечающих за то, должна ли сессия быть закрыта, (то есть является ли это сообщение последним в сессии).

В конце сессии будет вызов `on_session_end` с параметрами сессии и результатом завершения:

```
on_session_end?session_id=<номер_сессии>&phone=<номер_телефона>
&optional=<optional>&encoding=<encoding>&exit_code=<код_завершения>
&reason=<расшифровка_кода_завершения>
```

### Возможные коды завершения сессии

200 — завершилась нормально, путём закрытия сессии Вашим сервисом,  
 500 — ошибка, среди которых: прервана по инициативе оператора связи, прервана по инициативе приложения (SMSimple::ussd\_session\_abort), прервана пользователем (отказ ответа на очередное меню), прервана вследствие устаревания (по истечении 90 секунд на ответ), прочие причины (ошибки). Описание ошибки передается в `<расшифровка_кода_завершения>`.

В любой момент сессии приложение может прервать сессию. Для этого должна быть вызвана функция

```
SMSimple::ussd_session_abort($ussd_session_id)
```

Сразу за этим произойдет закрытие USSD-сессии и вызов `on_session_end` с параметром `exit_code=500`.

### Рабочий пример

Для примера создадим игру «Угадай число!». В качестве приглашения (первого меню) будет выводиться текст «Привет! Угадай число от 0 до 9.» Далее каждый ответ будет сравниваться с загаданным числом, например, 7. Если число не равно 7, например, 5, то выводится меню «Ты ввёл число 5 и не угадал. Попробуй ещё.». Если же число угадано, выводим «Правильно! Ты угадал число 7. До свидания!» и завершаем сессию.

Посмотреть в эту игру в действии можно, перейдя по ссылке <http://smsimple.ru/api-samples/example9-start-ussd-session.php> и указав там свой номер телефона.

Итак, нам нужны два скрипта [on\\_session\\_start.php](#) и [on\\_user\\_response.php](#). Для более серьёзных задач может понадобиться скрипт [on\\_session\\_end.php](#). В нашем случае для самой игры он не нужен, но для демонстрации мы его все же напишем. Кроме того, нужен скрипт, инициирующий запуск сессии с помощью SMSimple PHP API, назовём его `start.php`.

#### **start.php**

```
<?php

// Подключаем интерфейс для доступа к API
```

```

require_once('./smsimple.class.php');

// Создаем класс для доступа к API
$sms = new SMSimple(array(
    'url'      => 'http://api.smsimple.ru',
    'username' => 'xxxxxx', // имя учетной записи
    'password' => 'xxxxxx', // и пароль
));

try {
    // Подключаемся к сервису
    $sms->connect();

    // Телефон получателя
    $phone = '79991234567';
    $optional = 'asdf1234'; // например, внутренний номер сессии
    $encoding = 'UCS2';

    // Производим старт USSD сессии
    $sms->ussd_session_start($phone,$optional,$encoding);

    echo "Запрос на открытие USSD-сессии для $phone отправлен.";
}
catch (SMSimpleException $e) {
    echo $e->getMessage();
}
?>

```

#### on\_session\_start.php

```

<?
$session_id = $_GET['session_id']; // что-то типа 1239487029375123
$phone = $_GET['phone']; // 7999123456
$optional = $_GET['optional']; // взятое для примера значение asdf1234
$encoding = $_GET['encoding']; // UCS2 или GSM-7
$text = $_GET['text']; // пустой при запросе первого меню

header('Content-type: text/xml; encoding=utf-8');
echo '<?xml version="1.0" encoding="utf-8"?>';

?>
<response>
<menu><?=$_GET["encoding"]=="UCS2"?'Привет. Угадай число от 0 до 9!':'privet.
ugaday chislo ot 0 do 9!' ?></menu>
<close>false</close>
</response>

```

#### on\_user\_response.php

```

<?php
$session_id = $_GET['session_id']; // что-то типа 1239487029375123
$phone = $_GET['phone']; // 7999123456
$optional = $_GET['optional']; // взятое для примера значение asdf1234
$encoding = $_GET['encoding']; // UCS2 или GSM-7
$text = $_GET['text']; // пустой при запросе первого меню

header('Content-type: text/xml; encoding=utf-8');
echo '<?xml version="1.0" encoding="utf-8"?>';

?>
<response>
<menu><? if($_GET["text"]=="7"): ?>

```

```
<?= $_GET["encoding"]=="UCS2"? 'Правильно! Бывай здоров!': 'pravilno! byvay
zdorov!' ?>
<? else: ?>
<?= $_GET["encoding"]=="UCS2"? ('Ты ввёл число ' .$_GET['text']. ' и не угадал.
Давай ещё!'): ('ty vvel chislo ' .$_GET['text']. ' i ne ugadal. davay eshche!') ?>
<? endif ?></menu>
<close><?= $_GET["text"]=="7"? "true": "false" ?></close>
</response>
```

#### on\_session\_end.php

```
<?php

$session_id = $_GET['session_id']; // что-то типа 1239487029375123
$phone = $_GET['phone']; // 7999123456
$optional = $_GET['optional']; // взятое для примера значение asdf1234
$encoding = $_GET['encoding']; // UCS2 или GSM-7
$exit_code = $_GET['exit_code']; // код завершения
$reason = $_GET['reason']; // описание кода завершения

?>
```

Скрипты on\_session\_start.php и on\_user\_response.php вполне можно объединить в один, но в данном случае это не имеет смысла.

Итак, на Вашей стороне запускается скрипт start.php, после чего SMSimple сам будет вызывать остальные Ваши скрипты по мере надобности.

## Подключение

Для использования возможностей USSD обратитесь в [SMSimple.ru](https://smsimple.ru). Для подключения Вам следует подготовить скрипты для использования в качестве on\_session\_start, on\_session\_end и on\_user\_response. Адреса скриптов должны быть предоставлены в виде URL с обозначенными позициями для подстановки передаваемых параметров. Например, приведённые выше скрипты игры «Угадай число» должны быть представлены в виде:

```
https://smsimple.ru/ussd/on_session_start.php?session_id=:session_id&
phone=:phone&optional=:optional&encoding=:encoding&text=:text
```

```
https://smsimple.ru/ussd/on_user_response.php?session_id=:session_id&
phone=:phone&optional=:optional&encoding=:encoding&text=:text
```

```
https://smsimple.ru/ussd/on_session_start.php?session_id=:session_id&
phone=:phone&optional=:optional&encoding=:encoding&
exit_code=:exit_code&reason=:reason
```

Для URL-style адресов это могло бы выглядеть, например, так:

```
https://smsimple.ru/ussd/on_session_start/session_id/:session_id/phone/
:phone/optional/:optional/encoding/:encoding/text/:text
```

```
https://smsimple.ru/ussd/on_user_response/session_id/:session_id/phone/
:phone/optional/:optional/encoding/:encoding/text/:text
```

```
https://smsimple.ru/ussd/on_session_end/session_id/:session_id/phone/
:phone/optional/:optional/exit_code/:exit_code/reason/:reason
```