

# Документация по API «Нью-Тел» для услуги CallPassword – «Авторизация по звонку»

Данный документ описывает возможности взаимодействия с API «Нью-Тел» для использования услуги CallPassword – «Авторизация по звонку», доступной всем зарегистрированным абонентам, которым она подключена.

Для осуществления взаимодействия с API CallPassword должны быть сформированы ключи для авторизации и подписи запросов к API-серверу. Операции с ключами и настройка услуги производятся в личном кабинете «Нью-Тел» в разделе «Авторизация по звонку».

Документацию по всем методам API «Нью-Тел» можно найти по адресу:

<https://web.new-tel.net/public/New-Tel API reference.pdf>

## Описание протокола

1. Запросы к API «Нью-Тел» осуществляются посредством протокола HTTPS на адрес:  
<https://api.new-tel.net/>
2. API-сервер обрабатывает только запросы с методом POST, следуя спецификации HTTP/1.1
3. Параметры запроса передаются в теле сообщения в формате:  
[application/json](#) или [application/x-www-form-urlencoded](#)
4. Строка параметров запроса должна быть сформирована в кодировке UTF-8 без дополнительных пробелов и переносов строк. Например, в PHP это можно сделать методами:  
[json\\_encode\(\\$params\)](#) или [http\\_build\\_query\(\\$params\)](#),  
где: [\\$params](#) – ассоциативный массив параметров или объект;
5. Формат данных запроса должен быть указан в заголовке: [Content-Type](#)
6. Корректная длина сообщения должна быть указана в заголовке: [Content-Length](#)
7. Ответные данные возвращаются всегда в кодировке UTF-8. Формат данных может быть указан в заголовке [Accept](#), где допускаются следующие значения:  
[application/json](#) – для ответа в формате JSON;  
[application/xml](#) – для ответа в формате XML;  
В случае отсутствия в запросе заголовка [Accept](#) данные возвращаются в формате JSON.

Примечание: в данной документации все примеры ответов API-сервера отформатированы с добавлением пробелов и переводов строк для лучшей читаемости.

## Аутентификация и авторизация

Для прохождения аутентификации и авторизации каждый запрос должен содержать заголовок:

**Authorization: Bearer** [ключ\\_запроса](#)

где: [ключ\\_запроса](#) – это строка длиной 122 символа, полученная соединением ключа API для авторизации запросов с меткой времени запроса и вычисленной подписью запроса.

Ключ API для авторизации запросов представляет собой строку из 48-и шестнадцатеричных цифр в нижнем регистре (десятичных цифр и латинских букв «a», «b», «c», «d», «e» и «f»).

Метка времени запроса передаётся в виде Unix timestamp и представляет собой строку, состоящую из 10-и десятичных цифр. Для прохождения аутентификации и авторизации все запросы должны иметь метку времени в пределах  $\pm 10$  минут от текущего времени.

Подпись запроса состоит из 64-х шестнадцатеричных цифр в нижнем регистре, которые являются текстовым представлением результата хеширования по алгоритму SHA-256 от объединения в указанном порядке следующих данных: строки с наименованием метода, метки времени запроса, ключа API для авторизации запросов, строки параметров запроса и ключа API для подписи запросов, между которыми вставлен символ новой строки (0x0A или "\n").

Например, для сформированного 1-го июля 2018 года в 15:00:00 (МСК) запроса к API по адресу <https://api.new-tel.net/group/method> с параметрами "param1" и "param2" со значениями 10 (число) и "some string" (строка) соответственно, передаваемых в формате JSON, данные для вычисления подписи запроса будут выглядеть следующим образом:

```
group/method
1530446400
ключ_API_для_авторизации_запросов
{"param1":10,"param2":"some string"}
ключ_API_для_подписи_запросов
```

Ключи для авторизации и подписи запросов к API-серверу для использования услуги CallPassword необходимо взять в личном кабинете «Нью-Тел» в разделе «Авторизация по звонку».

## Параметры ответа

При формировании ответа API-сервер устанавливает соответствующие HTTP-статусы и заголовки. В теле ответа возвращается объект, обязательно содержащий свойство `status`, отображающее результат обработки запроса API-сервером – строка с одним из значений: "success" – обработка запроса прошла успешно (аутентификация и авторизация произведены, запрашиваемый метод существует, все необходимые параметры присутствуют и валидны); "error" – при обработке запроса произошла ошибка.

Если при обработке запроса произошла ошибка, объект в теле ответа будет также содержать свойство `message`, в котором будет находиться текстовое описание ошибки. При этом будет установлен соответствующий HTTP-статус. Например, при корректном запросе, отправленном на адрес несуществующего метода будет установлен HTTP-статус "404 Not Found" и сформирован ответ:

```
{
  "status": "error",
  "message": "Requested method not found"
}
```

При успешной обработке запроса будет установлен HTTP-статус "200 OK" и объект в теле ответа будет содержать объект **data**, обязательно содержащий свойство **result**, отображающее результат работы запрашиваемого метода, строка с одним из значений:

"**success**" – запрашиваемый метод успешно выполнен;

"**error**" – при выполнении метода возникла ошибка.

Если запрашиваемый метод подразумевает возврат данных, то при успешном выполнении метода объект **data** будет также содержать указанное в описании метода свойство, массив или объект, где будут находиться ответные данные.

Например, при корректном запросе, отправленном на адрес: <https://api.new-tel.net/> формируется ответ:

```
{
  "status": "success",
  "data": {
    "result": "success",
    "value": "New-Tel API Server"
  }
}
```

Если при выполнении метода возникла ошибка, объект **data** будет также содержать свойство **message**, в котором будет находиться текстовое описание ошибки. Возможные ситуации возникновения ошибки и соответствующие сообщения перечислены в описании методов.

Пример ответа, при котором возникла ошибка при выполнении метода:

```
{
  "status": "success",
  "data": {
    "result": "error",
    "message": "Error occured"
  }
}
```

## Подпись ответа

Клиентская сторона может проверять достоверность получаемых данных, сверяя подпись ответа. Подпись ответа вычисляется по алгоритму, аналогичному для вычисления подписи запроса, где вместо строки параметров запроса используется текстовый контент ответного HTTP-сообщения. Результат вычисления отправляется в текстовом виде с HTTP-заголовком **Signature**.

**ВАЖНО:** со стороны API-сервера подписываются только ответы на авторизованные запросы с корректной подписью!

## Пример кода на PHP

Пример кода на PHP формирования запроса для асинхронного вызова на номер +7 904 111-22-33 с передачей кода "01234" и тайм-аутом 30 секунд:

```
// функция для вычисления ключа запроса
function getRequestKey($methodName, $time, $accessKey, $params, $signatureKey)
{
    return $accessKey . $time . hash('sha256',
        $methodName . "\n" .
        $time . "\n" .
        $accessKey . "\n" .
        $params . "\n" .
        $signatureKey
    );
}

$data = json_encode([
    'async' => 1,
    'dstNumber' => '79041112233',
    'pin' => '01234',
    'timeout' => 30,
]);

$time = time();
$resId = curl_init();
$requestKey = getRequestKey(
    'call-password/start-password-call', // наименование метода
    $time, // метка времени запроса
    '1234567890abcdef1234567890abcdef1234567890abcdef', // ключ авторизации
    $data, // строка параметров запроса
    'abcdef1234567890abcdef1234567890abcdef1234567890' // ключ подписи
);

curl_setopt_array($resId, [
    CURLOPT_HEADER_OUT => true,
    CURLOPT_HEADER => 0,
    CURLOPT_HTTPHEADER => [
        'Authorization: Bearer ' . $requestKey,
        'Content-Type: application/json',
    ],
    CURLOPT_POST => true,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_SSL_VERIFYPEER => false,
    CURLOPT_URL => 'https://api.new-tel.net/call-password/start-password-call',
    CURLOPT_POSTFIELDS => $data,
]);

$response = curl_exec($resId);
$curlInfo = curl_getinfo($resId);
```

## Дополнительные требования

1. Клиентская сторона должна проверять корректность SSL-сертификата сервера, и если SSL-сертификат не прошёл проверку, следует немедленно прекратить сессию во избежание утечки данных.
2. Ключи для взаимодействия с API не должны храниться в открытом виде, например, в cookie.

## Группа методов системы CallPassword для авторизации по звонку (call-password/\*)

Генерация вызовов через API, получение данных по идентификатору сгенерированного вызова.

Доступные методы:

[Инициирование вызова "CallPassword"](#)

[Инициирование вызова "VoicePassword"](#)

[Завершение вызова "CallPassword" или "VoicePassword"](#)

[Получение данных о вызове "CallPassword" или "VoicePassword"](#)

## Инициирование вызова "CallPassword"

Метод осуществляет вызов "CallPassword" на номер назначения, отображая в качестве источника номер из пула системы "CallPassword" для использования авторизации по звонку.

### Наименование метода

`call-password/start-password-call`

### Параметры запроса

- `async` – флаг асинхронности запроса, число 0 или 1
- `dstNumber` – номер телефона назначения, строка, от 9 до 15 десятичных цифр, не начиная с 0
- `pin` – код, который будет использован как последние 5 цифр номера источника, строка, 5 десятичных цифр, необязательный параметр, в случае отсутствия будет сгенерирован API-сервером
- `timeout` – время ожидания ответа в секундах, число от 20 до 99, необязательный параметр, в случае отсутствия, время ожидания ответа будет установлено 20 секунд
- `callbackLink` – ссылка вызова callback при окончании звонка, строка, адрес http/https запроса, необязательный параметр

Примечание: при синхронном запросе (`async=0`) API-сервер сформирует ответ только после получения статуса вызова, определение которого может продолжаться не более времени ожидания ответа, указанного в параметре `timeout`; при асинхронном (`async=1`) – ответ будет сформирован сразу после обработки запроса, при этом статус вызова будет неопределён.

### Ограничение запросов

стандартное

### Данные ответа

объект `callDetails`, содержащий свойства:

- `callId` – идентификатор сгенерированного вызова, строка
- `pin` – код, который использован как последние 5 цифр номера источника, строка
- `status` – статус вызова, строка, возможные значения при синхронном запросе:  
"answered" – отвечено, "busy" – занято, "no answer" – нет ответа,  
"no such number" – номер не существует, "not available" – сеть недоступна  
или `null` при асинхронном запросе
- `operator` – принадлежность `dstNumber` к оператору связи.

## Пример данных ответа

```
{
  "status": "success",
  "data": {
    "result": "success",
    "callDetails": {
      "callId": "1234567890123456",
      "pin": "12345",
      "status": "answered",
      "operator": "Вымпелком"
    }
  }
}
```



## Инициирование вызова "VoicePassword"

Метод осуществляет вызов "VoicePassword" на номер назначения. При ответе абонента, ему проговаривается код, указанный при вызове метода для авторизации по звонку.

### Наименование метода

`call-password/start-voice-password-call`

### Параметры запроса

- `async` – флаг асинхронности запроса, число 0 или 1
- `dstNumber` – номер телефона назначения, строка, от 9 до 15 десятичных цифр, не начиная с 0
- `text` – код, который будет произнесён абоненту, строка, 5 десятичных цифр, необязательный параметр, в случае отсутствия будет сгенерирован API-сервером
- `timeout` – время ожидания ответа в секундах, число от 20 до 99, необязательный параметр, в случае отсутствия время ожидания ответа будет установлено 20 секунд
- `callbackLink` – ссылка вызова callback при окончании звонка, строка, адрес http/https запроса, необязательный параметр

Примечание: при синхронном запросе (`async=0`) API-сервер сформирует ответ только после получения статуса вызова, определение которого может продолжаться не более времени ожидания ответа, указанного в параметре `timeout`; при асинхронном (`async=1`) – ответ будет сформирован сразу после обработки запроса, при этом статус вызова будет неопределён.

### Ограничение запросов

стандартное

### Данные ответа

объект `callDetails`, содержащий свойства:

- `callId` – идентификатор сгенерированного вызова, строка
- `text` – код, который использован как последние 5 цифр номера источника, строка
- `status` – статус вызова, строка, возможные значения при синхронном запросе:  
"answered" – отвечено, "busy" – занято, "no answer" – нет ответа,  
"no such number" – номер не существует, "not available" – сеть недоступна  
или `null` при асинхронном запросе
- `operator` – принадлежность `dstNumber` к оператору связи.

Пример данных ответа:

```
{
  "status": "success",
  "data": {
    "result": "success",
    "callDetails": {
      "callId": "7257174731586440148",
      "status": null,
      "operator": "Теле2-Санкт-Петербург",
      "text": "01234"
    }
  }
}
```

## Завершение вызова “CallPassword” или “VoicePassword”

Метод осуществляет завершение звонка, сгенерированного методом инициирования вызова “CallPassword” или “VoicePassword”.

### Наименование метода

`call-password/hangup-password-call`

### Параметры запроса

`callId` – идентификатор звонка, полученный при вызове метода генерации звонка

### Ограничение запросов

стандартное

### Данные ответа

нет (только статус работы запрашиваемого метода)

### Пример данных ответа

```
{
  "status": "success",
  "data": {
    "result": "success"
  }
}
```

## Получение данных о вызове “CallPassword” или “VoicePassword”

Метод для получения данных о звонке, сгенерированным методом инициирования вызова “CallPassword” или “VoicePassword”.

### Наименование метода

`call-password/get-password-call-status`

### Параметры запроса

`callId` – идентификатор звонка, полученный при вызове метода генерации звонка

### Ограничение запросов

стандартное

### Данные ответа

объект `callDetails`, содержащий свойства:

- `callId` – идентификатор сгенерированного вызова, строка
- `status` – статус вызова, строка, возможные значения:
  - "answered" – отвечено, "busy" – занято, "no answer" – нет ответа,
  - "no such number" – номер не существует,
  - "not available" – сеть недоступна
- `reasonCode` – цифровой код статуса вызова, возможные значения:
  - 0 – "no such number", 1 – "no answer", 3 – "busy", 4 – "answered",
  - 5 – "busy", 8 – "not available"

Если информации по указанному идентификатору `callId` нет, будет сформировано сообщение об ошибке "Call not found".

### Пример данных ответа

```
{
  "status": "success",
  "data": {
    "result": "success",
    "callDetails": {
      "callId": "5004076351586439594",
      "reasonCode": "8",
      "status": "not available"
    }
  }
}
```