

# Интеграция по протоколу WSS

Для отображения карточки клиента в интерфейсе Вашей CRM в режиме real-time, Вам необходимо подключиться к серверу используя Web-сокеты, затем ожидать от сервера сообщения о входящем звонке.

## Описание API Web Socket

### Подключение и авторизация

Только авторизированные пользователи могут подписываться на команды от АТС или отправлять ей такие команды.

`wss://<domain>/live/connector/v3/easy?domain=<domain>&apiKey=<apiKey>&subscriber=<subscriber>` – подключение к Web-socket серверу.

где,

**domain** – API сервер, другими словами домен вашей АТС (например, `qwerty.pbх.vega.ua`);

**apiKey** – API ключ для подключения по API;

**subscriber** – внутренний номер/email сотрудника на события которого Вы хотите получить подписку.

Авторизационные данные (`domain`, `apiKey`, `subscriber`) можно получить открыв страницу «Настройки» и перейдя на вкладку «Интеграция», затем открыв раздел «Другая CRM система».

#### Пример

```
var _socket = new
WebSocket('wss://qwerty.pbх.vega.ua/live/connector/v3/easy?domain=qwerty.pbх.vega.ua&apiKey=S
3Bvc2eNimuQmQWyJt5aEvNjbND1fz0X&subscriber=001');
```

### Получение информации о входящем звонке

В момент звонка, если у менеджера открыта CRM и его браузер подписан на прием web-socket сообщений, информацию о событиях звонка передается в браузер. Затем в CRM можно показать специальное всплывающее окно с карточкой клиента.

#### Параметры

Входящих параметров нет, все что необходимо это подписаться на сообщения от Socket сервера;

```
var _socket = new WebSocket('...');
```

```
t._socket.onmessage = function(e) { ... }
```

#### Результат

Сообщение полученное с сервера, при звонке, имеет формат:

```
{
  "event": "call.dial",
  "uuid": "23-1453990999.41",
  "parentUuid": null,
  "lgDirection": 4,
  "leg": {
    "id": 120,
    "type": 1,
    "displayName": "Support Vega",
    "ext": "911"
  },
  "leg2": null,
  "otherLegs": [
    {
      "id": "29",
      "name": "Тест",
      "num": "+380567777777",
    }
  ]
}
```

```
        "companyName": "Рекламное агентство",
        "url": "https://qwerty.pbx.vega.ua/features/crm/contacts/edit.jsp#/?id=1", "priority": 2
    }
},
    "dialAt": 1473973200000,
    "bridgeAt": null,
    "serverTime": "1473973201000",
    "trunkNum": "567850550",
    "trunkName": "567850550"
}
```

**event** – событие которое произошло с телефонной системой  
Тип: String

Возможные значения:

- call.dial (начало звонка);
- call.bridge (на звонок ответили);
- call.hangup (звонок или разговор был завершен);
- error (ошибка).

**uuid** – уникальный идентификатор звонка  
Тип: String

**parentUuid** – уникальный идентификатор родительского звонка. Если звонок приходит на группу пользователей, затем в группе распределяется любому из сотрудников, тогда звонок сотруднику будет ссылаться на звонок в группу как на родительский звонок.  
Тип: String

**lgDirection** – направление звонка  
Тип: Number;

Возможные значения:

- 1 – внутренний звонок;
- 2 – исходящий звонок;
- 4 – входящий звонок;
- 32 – установка на паузу (нет на месте);
- 64 – снятие с паузы (есть на месте).

**leg** – данные пользователя (с сотрудника компании) выполняющего звонок или принимающего звонок (в зависимости от значения поля lgDirection);  
Тип: Object

**leg.id** – идентификатор сущности на которую пришел звонок или с которой он был совершен (если **lgDirection=2** – звонок был совершен от пользователя, если **lgDirection=4** – звонок был принят сущностью). Это поле может означать идентификатор пользователя (сотрудника компании), группы пользователей (отдела компании) или сценария (IVR меню). Какой тип сущности обозначает данный идентификатор, можно узнать из параметра **leg.type**;  
Тип: Number

**leg.type** – тип сущности на которую пришел звонок или от лица которой он был совершен;  
Тип: Number

Возможные значения:

- 1 – пользователь (сотрудник);
- 2 – группа (отдел сотрудников);
- 4 – сценарий (IVR меню)

**leg.ext** – внутренний номер сущности на которую пришел звонок или с которой он был совершен. Это может быть внутренний номер пользователя или группы пользователей. В случае если leg обозначает сценарий, данный параметр будет равен null (так как у сценария нет внутреннего номера). Какой тип сущности обозначает данный параметр, можно узнать из параметра **leg.type**;  
Тип: String

**leg.displayName** – имя сущности на которую пришел звонок или с которой он был совершен. Это может быть имя пользователя, название группы пользователей или сценария. Какой тип сущности обозначает данный параметр, можно узнать из параметра **leg.type**;  
Тип: String

**leg2** – данные пользователя (сотрудника компании) принимающего внутренний звонок. Данное поле всегда равняется null при исходящих и входящих звонках;  
Тип: Object

**leg2.id** – идентификатор сущности на которую пришел внутренний звонок (если это поле присутствует, тогда **lgDirection=1**). Это поле может означать идентификатор пользователя (сотрудника компании) или группы пользователей (отдела компании). Какой тип сущности обозначает данный идентификатор, можно узнать из параметра **leg2.type**;  
Тип: Number

**leg2.type** – тип сущности на которую пришел внутренний звонок;  
Тип: Number

Возможные значения:

- 1 – пользователь (сотрудник);
- 2 – группа (отдел сотрудников)

**leg2.ext** – внутренний номер сущности на которую пришел внутренний звонок. Это может быть внутренний номер пользователя или группы пользователей. Какой тип сущности обозначает данный параметр, можно узнать из параметра **leg2.type**;  
Тип: String

**leg2.displayName** – имя сущности на которую пришел внутренний звонок. Это может быть имя пользователя, название группы пользователей. Какой тип сущности обозначает данный параметр, можно узнать из параметра **leg2.type**;  
Тип: String

**otherLegs** – детали о клиенте. Данное поле является списком, так как может быть подключено несколько CRM систем и в результате будет получен список деталей со всех подключенных CRM. Поля данного списка, приведены ниже.  
Тип: Array

**otherLegs[0].id** – идентификатор клиента в CRM системе. null – если клиент не существует;  
**otherLegs[0].name** – имя клиента в CRM системе. null – если клиент не существует;  
**otherLegs[0].num** – номер телефона клиента в CRM системе;  
**otherLegs[0].companyName** – название компании клиента в CRM системе. null – если клиент не существует или не указано название компании;  
**otherLegs[0].url** – ссылка на существующий контакт в CRM, или на создание нового клиента;  
**otherLegs[0].priority** – приоритет подключения по которому был получен контакт.

**dialAt** – время когда начал звонить телефон (количество милли секунд прошедших с 1970 года, GMT+0)  
Тип: Number

**bridgeAt** – время в которое ответили на звонок, null если ответа пока нет (количество миллисекунд прошедших с 1970 года, GMT+0)  
Тип: Number

**serverTime** – время когда событие было отправлено с сервера ( количество миллисекунд прошедших с 1970 года, GMT+0)  
Тип: Number

**trunkNum** – внешний номер, который используется для приема текущего входящего звонка или совершения исходящего звонка (в зависимости от значения поля **lgDirection**). Данное поле всегда равняется null при внутренних звонках;  
Тип: String

**trunkName** – подпись для внешнего номера, который используется для приема текущего входящего звонка или совершения исходящего звонка (в зависимости от значения поля **lgDirection**);  
Тип: String

Пример

```
var _socket = new WebSocket('...');

_socket.onmessage = function(e) {
    var data = JSON.parse(e.data);
    if(data["event"] == "call.dial") {
        alert("Call from: " + data.otherLegs[0].num);
    }
};
```

## Набор по клику (click2call)

Звонить с интерфейса CRM вы можете отправив необходимое сообщение на WebSocket в котором Вы

авторизированы.

#### Параметры

```
{  
  "action" : "call",  
  "to" : "380991234567"  
}
```

**action** – значение «call»;

**to** – номер телефона на который нужно позвонить.

#### Результат

При успешном вызове:

```
{  
  "event": "success"  
}
```

При возникновении ошибки:

```
{  
  "event": "error",  
  "message": "..."  
}
```

#### Пример

```
var _socket = new WebSocket('...');  
var data = {  
  "action" : "call",  
  "to" : "380991234567"  
};  
_socket.send(JSON.stringify(data));
```