

Исходящий вебхук из CRM Voronka.pro контракт с внешними системами

На примере модуля: Сделки

1. Назначение

Webhook отправляется из Voronka CRM при сохранении записи сделки и предназначен для передачи данных во внешние системы:

- интеграционные шины,
- CRM/ERP/BI,
- колл-трекинг,
- системы аналитики,
- собственные backend-сервисы.

Текущая реализация передаёт:

1. **плоский набор полей сделки;**
 2. **обогащённые объектные данные:**
 - клиент,
 - статус сделки,
 - воронка,
 - стадия воронки,
 - локация;
 3. **custom fields;**
 4. **информацию об изменённых полях.**
-

2. HTTP-параметры запроса

Метод

```
POST
```

Content-Type

```
application/json; charset=UTF-8
```

Тело запроса

JSON-объект.

3. Общая структура payload

Webhook отправляет **один JSON-объект**.

Верхний уровень содержит:

- плоские поля сделки,
- системные поля интеграции,
- вложенные объектные блоки,
- блок изменений,
- блок `meta`.

Пример общей структуры:

```
{
  "subject": "Сделка 07.03.2026 18:27",
  "related_to": 297041,
  "assigned_user_id": 57,
  "ssalesprocesses_no": "C-55732",
  "ssalesprocesses_status": "PLL_SALE_COMPLETED",
  "locations": "32263",
  "listsalesfunnel": "Default",
```

```
"salesfunnelstage": "1::1",

"id": 470906,
"module": "SSalesProcesses",
"label": "Сделка 07.03.2026 18:27",
"event": "ssalesprocesses.saved",
"record_url": "index.php?module=SSalesProcesses&view=Detail&record=470906",

"changed_fields": ["ssalesprocesses_status"],
"changes": {
  "ssalesprocesses_status": {
    "old": "PLL_SALE_OPEN",
    "new": "PLL_SALE_COMPLETED",
    "old_display": "В работе",
    "new_display": "Проведено"
  }
},

"client": {
  "id": 297041,
  "number": "КЛ-20860",
  "name": "ВладимирСРМ2",
  "phone": "79135883488",
  "other_phone": "",
  "email": "",
  "email2": "",
  "type": "",
  "industry": ""
},

"deal_status": {
  "id": 14,
  "value": "PLL_SALE_COMPLETED",
  "label": "Проведено"
},

"sales_funnel": {
  "id": 1,
  "value": "Default",
  "label": "Основная"
```

```
},

"sales_funnel_stage": {
  "raw": "1::1",
  "funnel_id": 1,
  "stage_id": 1,
  "value": "PLL_SALE_NEW",
  "label": "Создано"
},

"location": {
  "id": 32263,
  "name": "НОВОСИБИРСК"
},

"custom_fields": {
  "cf_4040": "",
  "cf_4055": null
},

"meta": {
  "record_id": 470906,
  "module": "SSalesProcesses",
  "label": "Сделка 07.03.2026 18:27",
  "event": "ssalesprocesses.saved",
  "record_url": "index.php?module=SSalesProcesses&view=Detail&record=470906",
  "sent_at": "2026-03-07T14:41:59+03:00",
  "changed_fields": ["ssalesprocesses_status"]
}
}
```

4. Правила интерпретации данных

4.1. Плоские поля сделки

В корне JSON передаются поля сделки в “плоском” виде.

Примеры:

```
{
  "subject": "Сделка 07.03.2026 18:27",
  "related_to": 297041,
  "assigned_user_id": 57,
  "ssalesprocesses_no": "C-55732",
  "ssalesprocesses_status": "PLL_SALE_COMPLETED",
  "leadsource": "vk",
  "listsalesfunnel": "Default",
  "salesfunnelstage": "1::1",
  "locations": "32263"
}
```

Особенности:

- значения справочников и picklist-полей могут передаваться как **внутренние коды CRM**;
- для человекочитаемой интерпретации используйте соответствующие объектные блоки:
 - deal_status
 - deal_type
 - sales_funnel
 - sales_funnel_stage
 - location

4.2. Системные поля интеграции

В payload всегда присутствуют системные поля:

`id`

Внутренний ID записи сделки в CRM.

Тип:

```
number
```

Пример:

```
"id": 470906
```

module

Системное имя модуля CRM.

Тип:

```
string
```

Значение:

```
"SSalesProcesses"
```

label

Человекочитаемое название записи.

Тип:

```
string
```

Пример:

```
"label": "Сделка 07.03.2026 18:27"
```

event

Тип события webhook в текущей реализации.

Тип:

```
string
```

Текущее значение:

```
"ssalesprocesses.saved"
```

Важно

На текущем уровне реализации событие **не различает**:

- создание,
- изменение,
- удаление.

То есть и создание, и обновление записи приходят как:

```
"event": "ssalesprocesses.saved"
```

Для определения факта изменения следует использовать блок `changed_fields` / `changes`.

`record_url`

Относительная ссылка на карточку записи в CRM.

Тип:

```
string
```

Пример:

```
"record_url": "index.php?module=SSalesProcesses&view=Detail&record=470906"
```

Важно

Это **относительный URL**, не абсолютный.

Если внешней системе нужен полный адрес, базовый домен должен быть известен отдельно.

5. Объект клиента

Поле `client`

Содержит данные клиента, связанного со сделкой (`related_to`).

Пример:

```
"client": {
  "id": 297041,
  "number": "КЛ- 20860",
  "name": "ВладимирСРМ2",
  "phone": "79135883488",
  "other_phone": "",
  "email": "",
  "email2": "",
  "type": "",
  "industry": ""
}
```

Поля объекта `client`

`id`

ID клиента в CRM.

Тип:

number

`number`

Внутренний номер клиента.

Тип:

string

name

Название клиента.

Тип:

string

phone

Основной телефон клиента.

Тип:

string

other_phone

Дополнительный телефон клиента.

Тип:

string

email

Основной email клиента.

Тип:

string

email2

Дополнительный email клиента.

Тип:

string

type

Тип клиента.

Тип:

string

industry

Отрасль клиента.

Тип:

string

Дополнительные упрощённые поля клиента в корне

Для удобства внешней обработки дублируются:

```
"related_to_name": "ВладимирСРМ2",  
"related_to_phone": "79135883488",  
"related_to_email": ""
```

Рекомендация

Для интеграций лучше использовать:

- `client.name`
- `client.phone`
- `client.email`

а поля `related_to_name` / `related_to_phone` / `related_to_email` воспринимать как совместимость/shortcut.

6. Статус сделки

Сырые поля

```
"ssalesprocesses_status": "PLL_SALE_COMPLETED"
```

Это внутренний код CRM.

Человекочитаемое представление

```
"ssalesprocesses_status_display": "Проведено"
```

Объект `deal_status`

Пример:

```
"deal_status": {  
  "id": 14,  
  "value": "PLL_SALE_COMPLETED",  
  "label": "Проведено",  
  "picklist_valueid": 899,  
  "sort_order": 3,  
  "color": "#8BC34A",  
  "description": "",  
  "type_status": 2,  
  "icon": null  
}
```

Поля

`id`

Внутренний ID статуса в таблице CRM.

`value`

Системное значение статуса.

`label`

Человекочитаемый перевод статуса.

`picklist_valueid`

ID picklist-значения CRM.

`sort_order`

Порядок сортировки.

`color`

Цвет статуса.

`description`

Описание статуса.

`type_status`

Тип статуса.

`icon`

Иконка статуса, если определена.

Рекомендация

Во внешней системе для отображения использовать:

- `deal_status.label`

Для хранения/синхронизации:

- `deal_status.value`

7. Тип сделки

Если поле типа сделки в записи заполнено, `webhook` может содержать:

- `ssalesprocesses_type`

- `ssalesprocesses_type_display`
- `deal_type`

Пример целевого формата:

```
"deal_type": {  
  "id": 1,  
  "value": "PLL_NEW_SALES",  
  "label": "Новые продажи",  
  "sort_order": 1,  
  "color": ""  
}
```

Важно

В конкретных записях тип сделки может отсутствовать, если поле в самой записи пустое.

8. Воронка

Сырое поле

```
"listsalesfunnel": "Default"
```

Перевод

```
"listsalesfunnel_display": "Основная"
```

Объект `sales_funnel`

Пример:

```
"sales_funnel": {
  "id": 1,
  "value": "Default",
  "label": "Основная",
  "sort_order": 1,
  "color": "#1976D2",
  "type_status": 0,
  "icon": {
    "type": "icon",
    "name": "fas fa-baby-carriage"
  },
  "description": ""
}
```

Поля

`id`

ID воронки.

`value`

Системное значение воронки.

`label`

Человекочитаемое название.

`sort_order`

Порядок сортировки.

`color`

Цвет воронки.

`type_status`

Тип статуса воронки.

`icon`

Иконка воронки.

description

Описание.

Рекомендация

Во внешней системе:

- для UI использовать `sales_funnel.label`
- для логики/сопоставления использовать `sales_funnel.value`

9. Стадия воронки

Сырое поле

```
"salesfunnelstage": "1::1"
```

Формат:

```
{funnel_id}::{stage_id}
```

Пример:

- `1::1`
- `1::5`

Перевод

```
"salesfunnelstage_display": "Создано"
```

Объект `sales_funnel_stage`

Пример:

```
"sales_funnel_stage": {  
  "raw": "1:1",  
  "funnel_id": 1,  
  "stage_id": 1,  
  "value": "PLL_SALE_NEW",  
  "label": "Создано",  
  "picklist_valueid": 1080,  
  "sort_order": 1,  
  "color": "BBDEFB",  
  "type_status": 0  
}
```

Поля

`raw`

Исходное значение CRM.

`funnel_id`

ID воронки.

`stage_id`

ID стадии.

`value`

Системное значение стадии.

`label`

Человекочитаемое название стадии.

`picklist_valueid`

ID picklist-значения.

`sort_order`

Порядок сортировки.

color

Цвет стадии.

type_status

Тип стадии.

Рекомендация

Для интерфейса использовать:

- sales_funnel_stage.label

Для логики:

- sales_funnel_stage.raw
- или пару funnel_id + stage_id

10. Локация

Сырое поле

```
"locations": "32263"
```

Может содержать одну или несколько локаций.

Основной объект location

Если локация одна, дополнительно отдаётся единичный объект:

```
"location": {  
  "id": 32263,  
  "name": "НОВОСИБИРСК"  
}
```

Массив всех локаций

```
"locations_object": [  
  {  
    "id": 32263,  
    "name": "НОВОСИБИРСК"  
  }  
]
```

Массив ID локаций

```
"locations_ids": [32263]
```

Рекомендация

Если внешняя система поддерживает мульти-локации:

- используйте `locations_object`

Если нужна одна основная локация:

- используйте `location`

11. Custom fields

Все custom fields сделки передаются в объекте:

```
"custom_fields": {  
  "cf_4040": "",  
  "cf_4055": null,  
  "cf_4059": null,  
  "cf_4063": null,  
}
```

```
"cf_4065": null,  
"cf_4086": 0,  
"cf_4088": null,  
"cf_4089": null,  
"cf_4092": null  
}
```

Формат имён

Имена передаются в системном виде:

```
cf_XXXX
```

Типы значений

Могут быть:

- string
- number
- null

Особенность текущей реализации

Часть custom fields дополнительно может дублироваться в корне payload:

```
"cf_4055": null,  
"cf_4059": null
```

Рекомендация

Во внешней системе считать основным источником именно:

```
custom_fields
```

12. Информация об изменениях

Если запись была изменена и CRM определила изменённые поля, webhook содержит:

changed_fields

Массив имён изменённых полей:

```
"changed_fields": ["ssalesprocesses_status"]
```

changes

Объект с деталями изменений по каждому полю.

Пример:

```
"changes": {
  "ssalesprocesses_status": {
    "old": "PLL_SALE_OPEN",
    "new": "PLL_SALE_COMPLETED",
    "old_display": "В работе",
    "new_display": "Проведено",
    "old_object": {
      "id": 18,
      "value": "PLL_SALE_OPEN",
      "label": "В работе"
    },
    "new_object": {
      "id": 14,
      "value": "PLL_SALE_COMPLETED",
      "label": "Проведено"
    }
  }
}
```

```
}  
}
```

Формат объекта изменения

`old`

Старое raw-значение.

`new`

Новое raw-значение.

`old_display`

Человекочитаемое старое значение, если для поля доступно.

`new_display`

Человекочитаемое новое значение.

`old_object`

Расширенный объект старого значения для справочников/связей.

`new_object`

Расширенный объект нового значения.

Примечания по изменениям

1. Блок `changes` присутствует **только если CRM определила изменения**.
2. Не все поля обязательно будут иметь `old_display/new_display`.
3. Для обычных строковых полей изменение может выглядеть так:

```
"changes": {
  "subject": {
    "old": "Старая тема",
    "new": "Новая тема"
  }
}
```

4. Для справочных полей (статус, воронка, стадия, локация, клиент) будет расширенная форма с `*_display` и `*_object`.

13. Блок `meta`

Пример:

```
"meta": {
  "record_id": 470906,
  "module": "SSalesProcesses",
  "label": "Сделка 07.03.2026 18:27",
  "event": "ssalesprocesses.saved",
  "record_url": "index.php?module=SSalesProcesses&view=Detail&record=470906",
  "sent_at": "2026-03-07T14:41:59+03:00",
  "changed_fields": ["ssalesprocesses_status"]
}
```

Поля `meta`

`record_id`

Дублирует `id`.

`module`

Дублирует `module`.

`label`

Дублирует `label`.

`event`

Дублирует `event` .

`record_url`

Дублирует `record_url` .

`sent_at`

Время отправки webhook в ISO 8601 с часовым поясом.

Пример:

```
"2026-03-07T14:41:59+03:00"
```

`changed_fields`

Дублирует массив изменённых полей.

Рекомендация

`meta` можно использовать как системный служебный блок для логирования и трассировки.

14. Форматы данных

Даты и время

Дата

Формат:

```
YYYY-MM-DD
```

Пример:

```
"dateevent": "2025-10-24"
```

Дата-время

Формат:

```
YYYY-MM-DD HH:MM:SS
```

Пример:

```
"modifiedtime": "2026-03-07 14:41:59"
```

ISO 8601

Используется в `meta.sent_at`:

```
"sent_at": "2026-03-07T14:41:59+03:00"
```

Числа

Целые числа

Могут передаваться как JSON number:

```
"assigned_user_id": 57
```

Decimal / money

В текущем payload многие денежные значения передаются как **строки**, а не как JSON number:

```
"cost": "0.00000000",  
"amountprepay": "0.00000000"
```

Рекомендация

Во внешней системе денежные поля парсить как decimal/string, не как integer.

Пустые значения

Могут приходиться в одном из вариантов:

- пустая строка ""
- null
- 0
- "0.00000000"

Рекомендация

При проектировании внешней схемы учитывать, что CRM не полностью унифицирует “пустоту”.

15. Поля, которые рекомендуется использовать во внешней системе

Для идентификации сделки

- id
- ssalesprocesses_no
- label

Для отображения статуса

- `deal_status.label`

Для логики статуса

- `deal_status.value`

Для отображения воронки

- `sales_funnel.label`

Для логики воронки

- `sales_funnel.value`

Для отображения стадии

- `sales_funnel_stage.label`

Для логики стадии

- `sales_funnel_stage.raw`
- или `funnel_id + stage_id`

Для клиента

- `client.id`
- `client.name`
- `client.phone`

Для локации

- `location.id`
- `location.name`

Для аудита изменений

- `changed_fields`
 - `changes`
-

16. Ограничения текущей версии контракта

На текущем уровне реализации необходимо учитывать следующие особенности:

16.1. Событие не различает create/update/delete

Сейчас всегда приходит:

```
"event": "ssalesprocesses.saved"
```

16.2. `record_url` ОТНОСИТЕЛЬНЫЙ

Во внешней системе нужно самостоятельно добавлять домен CRM, если нужен полный URL.

16.3. Не все поля БД сделки обязаны присутствовать

Webhook строится на основе активных полей модуля и enrichment-логики. Он **не является полным raw-дампом всей строки БД.**

16.4. Custom fields частично дублируются

`custom_fields` — основной источник custom-полей; часть `cf_*` может повторяться в корне.

17. Рекомендации по интеграции

17.1. Не полагаться только на raw-поля справочников

Например, вместо:

```
"ssalesprocesses_status": "PLL_SALE_COMPLETED"
```

ИСПОЛЬЗОВАТЬ:

```
"deal_status": {  
  "value": "PLL_SALE_COMPLETED",  
  "label": "Проведено"  
}
```

17.2. Для связи сделки с клиентом использовать `client.id`

Поле `related_to` оставлено как raw ID, но основной объект клиента — это `client`.

17.3. Для отслеживания изменений использовать `changes`

Если внешний сервис должен реагировать только на конкретные изменения, ориентироваться на:

- `changed_fields`
- `changes`

Пример:

- если меняется только статус, обрабатывать только `changes.salesprocesses_status`.
-

17.4. Денежные значения парсить как `decimal/string`

Не рассчитывать, что они всегда будут JSON number.

18. Минимальный обязательный набор для

внешней обработки

Если внешней системе не нужен весь payload, минимум, который рекомендуется использовать:

```
{
  "id": 470906,
  "module": "SSalesProcesses",
  "event": "ssalesprocesses.saved",
  "label": "Сделка 07.03.2026 18:27",
  "client": {
    "id": 297041,
    "name": "ВладимирСРМ2",
    "phone": "79135883488"
  },
  "deal_status": {
    "value": "PLL_SALE_COMPLETED",
    "label": "Проведено"
  },
  "sales_funnel": {
    "value": "Default",
    "label": "Основная"
  },
  "sales_funnel_stage": {
    "raw": "1:1",
    "label": "Создано"
  },
  "location": {
    "id": 32263,
    "name": "НОВОСИБИРСК"
  },
  "changed_fields": ["ssalesprocesses_status"],
  "changes": {
    "ssalesprocesses_status": {
      "old": "PLL_SALE_OPEN",
      "new": "PLL_SALE_COMPLETED",
      "old_display": "В работе",
      "new_display": "Проведено"
    }
  }
}
```

```
}  
}  
}
```

19. Рекомендуемая стратегия обработки на стороне внешней системы

1. Принять JSON.
2. Считать `id`, `module`, `event`.
3. Найти или создать запись по `id`.
4. Считать блок `client`.
5. Считать `deal_status`, `sales_funnel`, `sales_funnel_stage`, `location`.
6. Если есть `changed_fields`, использовать их для выборочной логики обновления.
7. Если нужны дополнительные поля — читать их из корня `payload`.
8. Для custom-полей использовать `custom_fields`.

Revision #2

Created 7 March 2026 11:47:15 by Admin

Updated 7 March 2026 11:51:25 by Admin