

Crabler

- [Functions and Installation](#)
- [TechStack](#)
- [User Manual Crabler](#)
- [Prices](#)
- [Support](#)
- [Бизнес-процессы](#)

Functions and Installation

О продукте CRABLER

Программный комплекс CRABLER — базовое SaaS-программное обеспечение для разработки мобильных приложений для разных сфер малого, среднего бизнеса, а также управления объектами коммерческой недвижимости. ПО распространяется в виде интернет-сервиса, специальные действия по установке ПО на стороне пользователя не требуются.

В настоящий момент на его базе работают следующие проекты (сборки):

- **Crabler** бизнес-центр — одноименное приложение для автоматизации работы бизнес-центров, содержит все возможности производных проектов;
- **inDoctor** — медицинский агрегатор;
- **Услуги района** — агрегатор бытовых услуг;
- **Ферма на районе** - доставка фермерских продуктов;
- **Вертолет** — доставка из местных магазинов;
- **CargoMan** — агрегатор грузоперевозчиков;
- **ProfCosmetic** — доставка профессиональной косметики.

Все приложения ООО “Краблер АйТи” - производные (whitelabel) приложения CRABLER адаптированные под конкретный рынок и проект:

— Все приложения IOS <https://apps.apple.com/ru/developer/crabler-llc/id1438484747?see-all=i-phoneipad-apps>

— Все приложения Android <https://play.google.com/store/apps/developer?id=Crabler+IT,+LLC>

Основные области применения продукта:

- Автоматизация работы объектов коммерческой недвижимости
- Маркетинг новых и существующих клиентов
- Автоматизация обработки заказов
- Автоматизация оплат
- Контроль качества поставляемых услуг
- Информирование клиентов

Функциональные характеристики

- Нативные мобильные приложения — IOS / Android
- Личный кабинет — веб-версия для ПК и Смартфона
- Основные функции продукта и его производных сборок:
 - Определение гео-локации пользователя в приложении
 - Профиль организации по локации, тегам, формату услуг: выезды, онлайн и очные приемы
 - Фильтрация организаций по установленным администратором тегам в приложении
 - Каталог услуг организации — любое количество услуг, настраиваемые формы заказа
 - Каталог товаров — поддержка импорта из файла, мобильная витрина, корзина, доп.опции товаров.
 - Чаты — организаций и пользователей
 - Прием платежей: карты МИР, VISA, MasterCard, ApplePay, Google Pay
 - Интеграция с онлайн-кассами — cloudpayments.ru
 - Настраиваемые бизнес-процессы обработки заказов, по этапам и ролям доступа
 - Лента постов организации и пользователя
 - Подписка на организации и пользователей
 - Рассылка push-уведомлений подписчикам
 - Публикация веб-страниц в приложении на экране профиля организации
 - Управление сотрудниками, подписчиками, контактами, чатами, товарами, процедурами, услугами, заказами, ценами, страницами, постами, договорами

Установка и авторизация

Установка на мобильный телефон IOS/Android

1. Для IOS-устройств - откройте приложение AppStore, введите в поиске "Crabler", найдите приложение в списке результатов и нажмите установить. Для Android-устройств - откройте приложение Play Market, введите в поиске "Crabler", найдите приложение в списке результатов и нажмите установить. Или перейдите по прямым ссылкам:

— Android <https://play.google.com/store/apps/details?id=com.crabler.android>

— IOS <https://apps.apple.com/ru/app/crabler/id1438484748>

2. После установки откройте приложение, следуйте инструкциям на экране смартфона. Разрешите доступ к гео-локации и отправке push-уведомлений.

3. Перейдите в раздел Мой профиль и нажмите Войти. Введите свой номер мобильного телефона для получения кода. Введите полученный код (при поступлении звонка - введите последние 4 цифры номера, в ином случае запросите код через смс). При первой авторизации приложение запросит дополнительные данные для регистрации - укажите их и нажмите Подтвердить.

Для авторизации под демо-доступом используйте:

— Номер телефона **+79995000009**

— Код доступа **5566**

Авторизация в Личном кабинете администратора

1. Перейдите по ссылке <https://lk.crabler.com> для входа в личный кабинет
2. Используйте для авторизации свой логин/пароль из **договора**
3. Для авторизации под демо-доступом используйте:
 - Номер телефона **+79995000009**
 - Постоянный код доступа **5566**
4. Выберите нужный профиль организации, нажмите "Войти" для входе в профиль

Руководство пользователя - ПО Crabler

Личный кабинет — веб-сайт с закрытым доступом для управления мобильным приложением клиента.


Войти в личный кабинет по ссылке <https://lk.crabler.com>

Для авторизации используйте свой персональный логин и пароль из договора.

Демо-доступ в личный кабинет: логин +79995000009, пароль 5566.


Личный кабинет организации

После авторизации появится перечень доступных профилей для работы:




Демо Администратор CRABLER
Выйти


Выберите сообщество



БЦ Lighthouse
Сообщество места
Войти




Клиника АО Медицина
Сообщество провайдера
Войти



Ресторан Montis
Сообщество провайдера
Войти

Выберите ваш профиль или в демо-примере Ресторан Montis, нажмите кнопку Войти. Откроется личный кабинет выбранного профиля.

Разделы Личного кабинета организации




Демо Администратор
CRABLER
Администратор

Сменить сообщество

Ресторан Montis
Сообщество провайдера

О сообществе

- Чаты
- Акции
- Страницы
- Контакты
- Роли участников
- Участники
- Сотрудники
- Стена
- Новости
- Заказы
- Услуги
- Товары




Ресторан Montis
Москва, Валовая, 26

Montis' friends food & bar – это демократичный уютный бар в самом центре Москвы с разнообразной кухней, авторскими коктейлями и шумными вечеринками по выходным.

Обновить

Зона доставки (Местная доставка БЦ LightHouse)



Режим работы	Режим доставки
Понедельник	09:00 — 23:59
Вторник	09:00 — 23:59
Среда	09:00 — 23:59
Четверг	09:00 — 23:59
Пятница	09:00 — 05:00
Суббота	12:00 — 05:00
Воскресенье	12:00 — 23:59

Дополнительные параметры

Обновить

Основные контакты

Сайт	Сайт http://montisfriends.ru
Телефон	Бронирование столов 89808904916
Email	Почта montismanager@gmail.com

Настройки

<input type="checkbox"/>	Запрос ручной модерации
<input type="checkbox"/>	Запрос дополнительных реквизитов
<input checked="" type="checkbox"/>	Скрывать контакты участников
<input checked="" type="checkbox"/>	Автоматическое формирование топа услуг
<input checked="" type="checkbox"/>	Автоматическое формирование топа товаров
<input type="checkbox"/>	Отправлять sms участнику при добавлении
<input type="checkbox"/>	Отображать кнопку "Все услуги БЦ"

Обновить

Основные разделы Личного кабинета представлены в боковом меню:

- О сообществе - основная информации об организации, географии работы и графике;
- Чаты — управление чатами и участниками;
- Акции — активация специальных цен для пользователей;
- Страницы — управление публикацией контента в приложении;
- Контакты — управление способами связи в приложении;
- Роли участников — права доступа на отдельные услуги;
- Участники — пользователи в приложении подписанные на организацию;
- Сотрудники — сотрудники организации с определенными доступами;
- Стена / Новости — управление постами и рассылками пуш-уведомлений участникам;
- Заказы — анализ и экспорт заказов организации;
- Услуги — каталог с категориями и параметрами оказания, управление формами заказа услуг;
- Товары — физические и электронные товары для продажи;
- Топ услуг — выделение определенных услуг на главном экране организации;
- Топ товаров — выделение определенных товаров на главном экране организации;
- Места обслуживания — привязка организации к местам (зданиям, бизнес-центрам, районам);
- Справочники — база специальных справочных данных организации;
- Договор с сервисом — условия работы организации в части платежей;
- Договор с клиентом — договор-оферта между организацией и пользователем приложения;
- Профиль — настройка профиля авторизованного пользователя;

Подробная видео-инструкция по личному кабинету ПО Crabler

Содержание личного кабинета объекта недвижимости (на примере бизнес-центра)

- О сообществе - основная информации о месте (бизнес-центре), адресе и графике работы;
- Чаты — управление чатами и участниками;
- Страницы — управление публикацией контента в приложении;
- Контакты — управление способами связи в приложении с администрацией места;
- Роли участников — права доступа на отдельные услуги;
- Участники — пользователи в приложении подписанные на организацию;

- Стена / Новости — управление постами и рассылками пуш-уведомлений участниками;
- Сотрудники — сотрудники места с определенными доступами;
- Локальные места — каталог управления местами внутри основного места;
- Арендаторы — база организация-арендаторов бизнес-центра;
- Справочники — база специальных справочных данных организации;
- Профиль — настройка профиля авторизованного пользователя;

Мобильное приложение Crabler - интерфейс

Видео-уроки производных модулей и проектов

Видео-инструкция по работе приложения - CRABLER
(сборка inDoctor)

Видео-инструкция по работе Личного кабинета - Crabler
(сборка inDoctor)

Видео-инструкция по обработке заказов в приложении
- Crabler (сборка inDoctor)

Полная база видео-уроков в группе VK

https://vk.com/video/@crabler_it

Смотрите все инструкции по ПО

Тарифы на ПО

TechStack

Используемые технологии в платформе Crabler и производных сборках проектов

Backend

* PHP <http://www.php.net/>

* Yii2 <https://www.yiiframework.com/>

* PostgreSQL <https://www.postgresql.org/>

* API SMSЦентр <https://smc.ru/>

IOS библиотеки

название	url
Swinject	https://github.com/Swinject/Swinject
SwinjectStoryboard	https://github.com/Swinject/SwinjectStoryboard
LightRoute	https://github.com/SpectralDragon/LightRoute
Alamofire	https://github.com/Alamofire/Alamofire
AlamofireImage	https://github.com/Alamofire/AlamofireImage
Eureka	https://github.com/xmartlabs/Eureka
InputMask	https://github.com/RedMadRobot/input-mask-ios/blob/master/Source/Sample/Podfile

YandexMapKit	https://cocoapods.org/pods/YandexMapKit
SVProgressHUD	https://github.com/SVProgressHUD/SVProgressHUD
CropViewController	https://github.com/TimOliver/TOCropViewController
Firebase/Core	https://cocoapods.org/pods/Firebase
Fabric	https://cocoapods.org/pods/Fabric
Crashlytics	https://cocoapods.org/pods/Crashlytics
HockeySDK	https://cocoapods.org/pods/HockeySDK
FreshchatSDK	https://cocoapods.org/pods/FreshchatSDK
ESPullToRefresh	https://github.com/eggswift/pull-to-refresh
PopupDialog	https://cocoapods.org/pods/PopupDialog
Cosmos	https://cocoapods.org/pods/Cosmos
PagingMenuController	https://github.com/kitasuke/PagingMenuController

Android библиотеки

Название	Урл
cat.ereza.customactivityoncrash	https://github.com/Ereza/CustomActivityOnCrash
com.a65apps.clustering	https://github.com/65apps/android-clustering-for-yandex-mapkit
com.afollestad.materialdialogs	https://github.com/afollestad/material-dialogs
com.afollestad.materialdialogs.input	https://github.com/afollestad/material-dialogs
com.airbnb.lottie	https://github.com/airbnb/lottie-android
com.airbnb.lottiecom.facebook.login	https://github.com/airbnb/lottie-android
com.alimuzaffar.lib.pin	https://github.com/alphamu/PinEntryEditText
com.bumptech.glide	https://github.com/bumptech/glide
com.crashlytics.android	https://firebase.google.com/products/crashlytics
com.facebook	https://github.com/facebook/facebook-android-sdk
com.facebookcat.ereza.customactivityoncrash	https://github.com/Ereza/CustomActivityOnCrash

com.flurry.android.analytics.sdk	http://flurry.github.io/flurry-android-sdk/analytics/index-all.html
com.freshchat.consumer.sdk	https://github.com/freshworks/freshchat-android
com.getkeepsafe.relinker	https://github.com/KeepSafe/ReLinker
com.github.chrisbanes.photoview	https://github.com/chrisbanes/photoview
com.github.florent37.singledateandtimepicker	https://github.com/florent37/SingleDateAndTimePicker
com.github.omadahealth.typefaceview	https://github.com/omadahealth/TypefaceView
com.mancj.materialsearchbar	https://github.com/mancj/MaterialSearchBar
com.nowfal.kdroidext	https://github.com/nowfalsalahudeen/KdroidExt
com.pierfrancescosoffritti.androidyoutubeplayer	https://github.com/PierfrancescoSoffritti/android-youtube-player
com.rd.pageindicatorview	https://github.com/romandanylyk/PageIndicatorView
com.redmadrobot.inputmask	https://github.com/RedMadRobot/input-mask-android
com.rilixtech.widget.countrycodepicker	https://github.com/joielechong/CountryCodePicker
com.stfalcon.chatkit	https://github.com/stfalcon-studio/ChatKit
com.stfalcon.imageviewer	https://github.com/stfalcon-studio/StfalconImageViewer
com.tsongkha.spinnerdatepicker	https://github.com/drawers/SpinnerDatePicker/blob/master/SpinnerDatePickerLib/src/main/java/com/tsongkha/spinnerdatepicker/DatePicker.java
com.vbytsyuk.ahbottomnavigation	https://github.com/aurelhubert/ahbottomnavigation
com.yandex.mapkit	https://yandex.ru/dev/maps/mapkit/?from=mapsapi
com.yandex.metrika	https://github.com/yandexmobile/metrika-sdk-android
com.yandex.metrika.push	https://github.com/yandexmobile/metrika-sdk-android
com.yandex.metrika.push.core	https://github.com/yandexmobile/metrika-sdk-android
com.yandex.runtime	https://yandex.ru/dev/maps/mapkit/doc/android-ref/lite/com/yandex/runtime/package-summary.html
de.hdodenhof.circleimageview	https://github.com/hdodenhof/CircleImageView
id.zelory.compressor	https://github.com/zetbaitu/Compressor

io.fabric.sdk.android	https://firebase.google.com/docs/reference/android/io/fabric/sdk/android/fabric/package-summary
io.michaelrocks.libphonenumber.android	https://github.com/MichaelRocks/libphonenumber-android
me.toptas.fancyshowcase	https://github.com/faruktoptas/FancyShowCaseView/blob/master/fancyshowcaseview/src/main/java/me/toptas/fancyshowcase/FancyShowCaseView.kt
org.jetbrains.anko	https://github.com/Kotlin/anko
org.matrix.androidsdk	https://github.com/matrix-org/matrix-android-sdk
org.matrix.olm	https://github.com/matrix-org/olm
pub.devrel.easypPermissions	https://firebaseopensource.com/projects/googlesamples/easypPermissions/

User Manual Crabler

Руководство пользователя - ПО Crabler

Личный кабинет — веб-сайт с закрытым доступом для управления мобильным приложением клиента.

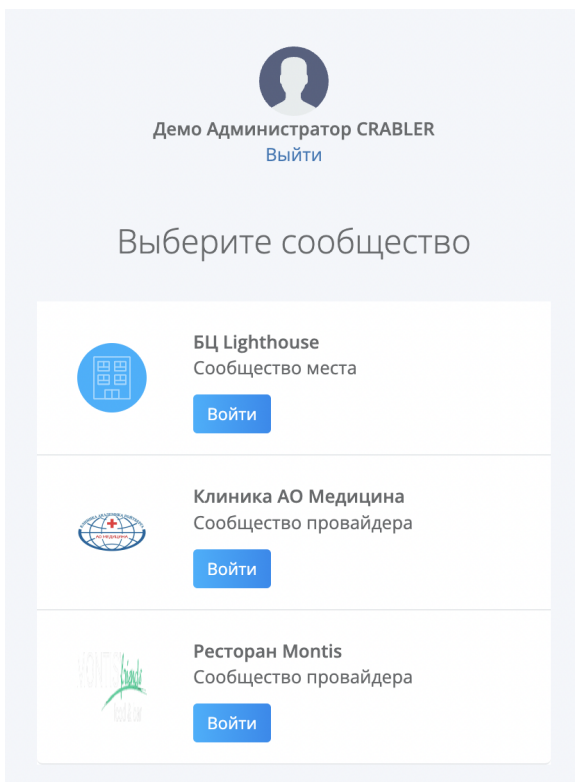
Войти в личный кабинет по ссылке <https://lk.crabler.com>

Для авторизации используйте свой персональный логин и пароль из договора.

Демо-доступ в личный кабинет: логин +79995000009, пароль 5566.

Личный кабинет организации

После авторизации появится перечень доступных профилей для работы:



Выберите ваш профиль или в демо-примере Ресторан Montis, нажмите кнопку Войти. Откроется личный кабинет выбранного профиля.

Разделы Личного кабинета организации

The screenshot shows the personal cabinet for 'Ресторан Montis'. On the left is a dark sidebar menu with options: 'О сообществе', 'Чаты', 'Акции', 'Страницы', 'Контакты', 'Роли участников', 'Участники', 'Сотрудники', 'Стена', 'Новости', 'Заказы', 'Услуги', and 'Товары'. The main content area includes: a profile card with a photo of the restaurant, name 'Ресторан Montis', address 'Москва, Валаовая, 26', and a description; a 'Режим работы' table; 'Дополнительные параметры' with an 'Обновить' button; 'Основные контакты' with website, phone, and email; and a 'Настройки' panel with various toggle switches and an 'Обновить' button.

Режим работы	Режим доставки
Понедельник	09:00 — 23:59
Вторник	09:00 — 23:59
Среда	09:00 — 23:59
Четверг	09:00 — 23:59
Пятница	09:00 — 05:00
Суббота	12:00 — 05:00
Воскресенье	12:00 — 23:59

Настройки	
<input type="checkbox"/> Нет	Запрос ручной модерации
<input type="checkbox"/> Нет	Запрос дополнительных реквизитов
<input checked="" type="checkbox"/> Да	Скрывать контакты участников
<input checked="" type="checkbox"/> Да	Автоматическое формирование топа услуг
<input checked="" type="checkbox"/> Да	Автоматическое формирование топа товаров
<input type="checkbox"/> Нет	Отправлять смс участнику при добавлении
<input type="checkbox"/> Нет	Отображать кнопку "Все услуги БЦ"

Основные разделы Личного кабинета представлены в боковом меню:

- О сообществе - основная информации об организации, географии работы и графике;
- Чаты — управление чатами и участниками;
- Акции — активация специальных цен для пользователей;
- Страницы — управление публикацией контента в приложении;
- Контакты — управление способами связи в приложении;
- Роли участников — права доступа на отдельные услуги;
- Участники — пользователи в приложении подписанные на организацию;
- Сотрудники — сотрудники организации с определенными доступами;
- Стена / Новости — управление постами и рассылками пуш-уведомлений участникам;
- Заказы — анализ и экспорт заказов организации;
- Услуги — каталог с категориями и параметрами оказания, управление формами заказа услуг;
- Товары — физические и электронные товары для продажи;
- Топ услуг — выделение определенных услуг на главном экране организации;
- Топ товаров — выделение определенных товаров на главном экране организации;
- Места обслуживания — привязка организации к местам (зданиям, бизнес-центрам, районам);
- Справочники — база специальных справочных данных организации;
- Договор с сервисом — условия работы организации в части платежей;
- Договор с клиентом — договор-оферта между организацией и пользователем приложения;

- Профиль — настройка профиля авторизованного пользователя;

Подробная видео-инструкция по личному кабинету ПО Crabler

Содержание личного кабинета объекта недвижимости (на примере бизнес-центра)

- О сообществе - основная информации о месте (бизнес-центре), адресе и графике работы;
- Чаты — управление чатами и участниками;
- Страницы — управление публикацией контента в приложении;
- Контакты — управление способами связи в приложении с администрацией места;
- Роли участников — права доступа на отдельные услуги;
- Участники — пользователи в приложении подписанные на организацию;
- Стена / Новости — управление постами и рассылками пуш-уведомлений участниками;
- Сотрудники — сотрудники места с определенными доступами;
- Локальные места — каталог управления местами внутри основного места;
- Арендаторы — база организация-арендаторов бизнес-центра;
- Справочники — база специальных справочных данных организации;
- Профиль — настройка профиля авторизованного пользователя;

Мобильное приложение Crabler - интерфейс

Видео-уроки производных модулей и проектов

Видео-инструкция по работе приложения - проект
inDoctor

Видео-инструкция по работе Личного кабинета - проект
inDoctor

Видео-инструкция по обработке заказов в приложении
- проект inDoctor

Полная база видео-уроков в группе VK

https://vk.com/video/@crabler_it

Prices

Тарифы на SaaS-платформу Crabler для бизнес-центров (договор услуг)

- 1-3 бизнес-центра — 19.500 рублей / месяц / бизнес-центр
- 4+ бизнес-центра — 17.500 рублей / месяц / бизнес-центр

Выпуск whitelabel-версии приложения IOS/Android — 199.500 рублей единовременно.

Поддержка whitelabel-версии приложения — 7.500 рублей / месяц / платформа.

Комиссия за прием платежей в приложении — 7% от транзакции, включая аренду онлайн-кассы.

Дилерские проекты (франчайзинг ПО)

Проект / Размер города работы	Стоимость пакета в зависимости от численности города			
	до 300 тыс.	до 500 тыс.	до 1 млн.	более 1 млн.
inDoctor	250 000	300 000	400 000	500 000
CargoMan	250 000	300 000	400 000	500 000
Услуги района	250 000	300 000	400 000	500 000
Вертолет	240 000	300 000	400 000	500 000

Ферма на районе	250 000	300 000	400 000	500 000
Crabler - бизнес-центры	200 000	300 000	350 000	450 000

Все цены указаны в рублях и без НДС. Дилерский договор дает эксклюзивное право работы дилера в выбранном городе.

Видео-презентации проектов

inDoctor, медсестра на дом

https://www.youtube.com/embed/8_6VL92IQrl

Услуги района

https://www.youtube.com/embed/_wZWDoaBLWE?t=640s

CargoMan

<https://www.youtube.com/embed/2YUiKIFYPcM?t=725s>

Ферма на районе

<https://www.youtube.com/embed/KaKH0vEAA2g?t=190s>

Support

Информация по поддержке,
устранению неисправностей в ходе
эксплуатации, совершенствовании ПО и
персонале

Обращения в службу поддержки ПО и
совершенствования

- <https://crabler-it.com/support>
- по e-mail: support@crabler-it.com
- по телефону: +7 (499) 286-96-40 (рабочее время с 10 до 18, по времени Москвы)
- в чате телеграм из договора бизнес-пользователя
- Адрес офиса поддержки и развития: Россия, Москва, переулок Гамсоновский 2с1

1. БП "Разработка ПО"

Разработка ПО идет спринтами (рабочими циклами по 2 недели каждый).

1. В начале спринта (первый рабочий день недели) менеджером продукта проводится формирование беклога исходя из приоритетов развития продукта на текущий квартал, а также заявок от службы поддержки (ошибки, запросы улучшений)
2. Задачи группируются на ошибки и фичи в соотношении (20/80)
3. Формируется команда текущего спринта из внутренних и внешних участников
4. На основании данных анализа беклога команда спринта формирует план спринта и выполняет его оценку в условных единицах (сторипоинты)
5. Ежедневно происходят встречи команды спринта (хадлы) для сверки промежуточных результатов спринта
6. По результатам завершения спринта

1. Выполняется релиз и/или багфикс
2. Создается документация для группы поддержки и разработки
3. Публикуется описание реализованных задач в открытых клиентских каналах
4. Производится публичная презентация нового функционала среди представителей клиентов
5. Результат видео-демонстрации публикуется в закрытой клиентской группе

2. БП "Поддержка ПО"

1. Запрос принимается поддержкой в рабочее время по всем каналам указанным на странице <https://crabler-it.com/support> с темой "Ошибка"
2. Все заявки обрабатываются в течение 2-х рабочих дней
3. Ответ по заявке отправляется на e-mail пользователя
4. При обнаружении ошибки - выполняется классификация ошибки
 1. Критическая — блокирует работу основных сценариев использование (заказы, оплаты, работа приложений) — отправляется с пометкой "блокер" в отдел разработки и устраняется за 1 рабочий день
 2. Типовая — отправляется с пометкой "normal" в отдел разработки и устраняется за 10 рабочих день
 3. После исправления ошибки специалист службы поддержки сообщает о результате клиенту

3. БП "Улучшение и развитие ПО"

1. Запрос принимается поддержкой в рабочее время по всем каналам указанным на странице <https://crabler-it.com/support> с темой "Улучшение"
2. Все заявки обрабатываются в течение 2-х рабочих дней
3. Ответ по заявке отправляется на e-mail пользователя
4. Все запросы на улучшения передаются в тикет-систему для рассмотрение менеджером продукта
5. Раз в две недели менеджер продукта формирует из новых запросов беклог для будущих спринтов
6. По каждому запросу указывается: вес стоимости разработки, вес влияния на цели клиента, статус (в спринт, в беклог, на будущее, отказ)
7. В случае взятие заявки в спринт — автору заявки отправляется уведомление.
8. Ежемесячно командой продукта производится пересмотр ключевых обращений на улучшение ПО и обновление их статусов
9. Менеджер продукта ежемесячно на основании общения с клиентами и пользователям пополняет список заявок на улучшение.

Персонал службы поддержки и совершенствования

- Аккаунт-менеджер (3 единицы) — прием и обработка заявок, составление клиентской документации
- Тестировщик (2 единицы) — проверка ошибки и классификация критичности
- Менеджер продукта (2 единицы) — рассмотрение новых обращений на улучшение функциональности и совершенствования ПО, формирование беклога, приемка спринта
- Ведущий разработчик (2 единицы) - развитие и поддержка архитектуры
- Разработчик мобильных приложений (2 единица) - обновление и выкладка приложений
- Разработчик fullstack (2 единицы) — кросс-разработки веб/мобильных систем

Местонахождение сервера — Россия, г. Санкт-Петербург, ул. Цветочная, д. 21, лит. А

Бизнес-процессы организации по работе над ПО

1. Аналитика
 1. По запросам от поддержки клиентов
 2. По результатам анализа рынка
 3. По инициативе команды
 1. Анализ базы транзакций
 2. Анализ базы организаций
 3. Анализ базы заказов
 4. Анализ базы гео-объектов
 5. Иная аналитика БД
 6. Построение нового VI-отчета
 1. Анализ трафика
 2. Анализ вовлеченности пользователей
 3. Анализ соцдема пользователей
 4. Анализ технических параметров пользователей
 1. Аудит рекламной кампании
 2. Анализ SEO-трафика
 3. Анализ партнерского трафика
 4. A/B тестирование дизайна
 5. A/B тестирование функциональности
 6. A/B тестирование предложения
 1. Анализ и синтез новой функциональности
 2. Анализ базы данных
 3. Анализ работы приложений
 4. Анализ маркетинга
2. Разработка ПО
 1. Новая функциональность
 2. Поддержка существующей функциональности
 3. Ошибки и уязвимости
 1. Анализ требований

2. Изучение существующих решений
 3. Формирование оптимальных требований
 4. Разработка моделей системы
 5. Визуализация моделей в фреймворке
 1. Анализ моделей и требований
 2. Определение стека
 3. Написание кода
 4. Квалификация кода (ревью)
 5. Оптимизация и доработка кода
 1. Выкладка в тестовую среду
 2. Сборка проекта
 3. Публикация для внутреннего тестирования
 4. Публикация для внешнего тестирования
 5. Публикация релиза
 6. Проверка заявленной обратной совместимости
 1. Проверка процессов
 2. Проверка ролей
 3. Проверка требований
 4. Проверка сборки приложения
 5. Проверка серверных компонент
 6. Проверка данных
 7. Написание автотестов
 8. Формирование отчета тестирования
 9. Повторная проверка
 1. Подключение нового сервера
 2. Резервирование данных
 3. Восстановление данных
 4. Работа с подрядчиками
 1. Определение целей и требований к подрядчику
 2. Поиск подрядчика
 3. Квалификация подрядчика
 4. Заключение договора
 5. Подключение к работе над проектом
 6. Контроль задач и бюджета
 7. Отчетность
-
1. Формирование беклога
 2. Приоритезация беклога
 3. Планирование спринта
 4. Ретро спринта
 5. Ретро релиза
 6. Демодей
 7. Проектирование
 8. Конструирование
 9. Сборка

10. Тестирование
11. Администрирование

3. Поддержка

1. Учет критических (blocker)
2. Учет иных дефектов (normal)
3. Оформление задачи с дефектом для беклога
4. Оформление задачи с дефектом для спринта (blocker)
 1. Учет дефектов (багов)
 2. Обработка обращений
 3. Создание задачи для беклога
 4. Обновление документации
 5. Публикация релиза
 6. Управление конфигурацией

4. Маркетинг

1. Написание статьи в блог или SMM
2. Разработка дизайна кампании
3. Разработка видео-ролика продукта
4. Проведение e-mail рассылки
5. Запуск кампании
6. Завершение кампании

5. Продажа

1. Создание лендинг-страницы продукта
2. Создание видео-ролика продукта
3. Разработка ценообразования
4. Публикация в открытых источниках
5. Консультация клиентов
6. Выставление счетов
7. Подписание договоров
8. Бухгалтерская отчетность
9. Обмен документами

Бизнес-процессы

1. Предназначены для управления заказами.
2. Привязаны к услугам. Один процесс может быть привязан к нескольким услугам.
3. При создании заказа создается копия бизнес-процесса выбранной услуги, после чего заказ управляется этой копией. Необходимо для обратной совместимости изменяемых процессов и уже запущенных по заказам.
4. В созданном заказе процесс всегда находится в определенном состоянии.
5. Доступны 4 роли пользователей для управления процессами (роли назначаются администратором провайдера в личном кабинете):
 - `customer` - заказчик,
 - `executor` - исполнитель,
 - `courier` - курьер,
 - `moderator` - модератор
6. Могут запускать в тестовом режиме - проходят тестовые транзакции оплаты без фактического списания средств. См.
<https://developers.cloudpayments.ru/#testirovanie>.
7. Синтаксис представлен в `json`-формате.

Пример настроенного бизнес-процесса

Показать переходы	При входе	Заказчик	Исполнитель	Модератор
Создание заказа (state0)	Процесс			
Оплата заказа (state1)		<div>Оплатить</div> <div>Отменить</div>		
На модерации (state2)		<div>Написать сообщение</div> <div>Отменить</div>		<div>Модерация пройдена</div> <div>Назначить исполнителя</div> <div>Написать сообщение</div> <div>Отклонить</div>
Поиск исполнителя (state3)	Процесс	<div>Написать сообщение</div> <div>Отменить</div>	<div>Взять в работу</div>	<div>Написать сообщение</div> <div>Отклонить</div>
Исполнитель назначен (state4)	Процесс	<div>Написать сообщение</div>	<div>Подтвердить выезд</div> <div>Отменить выезд</div> <div>Написать сообщение</div>	<div>Вернуть в поиск</div> <div>Написать сообщение</div> <div>Отклонить</div>
В работе (state5)	Процесс	<div>Написать сообщение</div>	<div>Завершить заказ</div> <div>Написать сообщение</div>	<div>Заменить исполнителя</div> <div>Отклонить</div>
Выполнен (stateDone)	Процесс	<div>Оценить выезд</div>	<div>Оценить выезд</div>	
Отменен (stateCancel)	Процесс			
Отклонен (stateReject)	Процесс			

Действие обычное.

Действие отображается при выполнении определенных условий.

Действие отмены / отклонения заказа.

JSON-файл в котором производится настройка

```
{
  "state0": {
    "label": "Создание заказа",
    "onStart": {
      "bp": {
        "step0": {
          "next": "step1",
          "type": "setData",
          "fields": {
            "field*": []
          }
        }
      }
    }
  },
}
```

```

"step1": {
  "next": "step2",
  "text": "Заказ #{{ order.code }}<br/><br/><strong>Услуга</strong>: {{
order.service.title }}<br/><strong>Доставка</strong>: {{ order.delivery
}}<br/><strong>0плачен</strong>: {{ order.payed }}<br/><strong>Сумма к оплате</strong>: {{
order.sum }}<br/><strong>Выплаченная сумма</strong>: {{ order.sum_payed }}<br/><strong>Общая
сумма</strong>: {{ order.sum_total }}<br/><strong>Организация заказчика</strong>: {{
order.customer.title }}<br/><strong>Заказчик</strong>: {{ order.customerUser.title }} {{
order.customerUser.phone_number }}<br/><strong>Провайдер</strong>: {{ order.executor.title
}}<br/><strong>Исполнитель</strong>: {{ order.executorUser.title
}}<br/><strong>Статус</strong>: {{ order.status.title }}<br/><strong>Дата запланированного
завершения</strong>: {{ order.deadline }}<br/><br/>{% for item in order.fieldsData
%}<strong>{{ item.field.title }}</strong>: {{ item.valueFormatted }}<br/>{% endfor
%}<br/>Детали заказа доступны в <a href=\"https://lk.crabler.com/site/{{
order.executor.communityProvider.id }}/jump\">личном кабинете</a>.",
  "type": "email",
  "subject": "Новый заказ",
  "recipients": [
    "zakaz@crabler.com",
    "moderator"
  ]
},
"step2": {
  "type": "setState",
  "state": "state1"
}
}
},
"state1": {
  "label": "Оплата заказа",
  "actions": [
    {
      "bp": {
        "step0": {
          "next": "step1",
          "type": "pay"
        },
        "step1": {

```

```
    "type": "if",
    "conditions": [
      [
        {
          "service.need_moderation": true
        },
        "step2"
      ],
      [
        true,
        "step3"
      ]
    ]
  },
  "step2": {
    "type": "setState",
    "state": "state2"
  },
  "step3": {
    "type": "setState",
    "state": "state3"
  }
},
"code": "pay",
"allow": [
  "customer"
],
"label": "Оплатить"
},
{
  "bp": {
    "step0": {
      "type": "setState",
      "state": "stateCancel"
    }
  },
  "code": "cancel",
  "allow": [
    "customer"
  ]
}
```

```
    ],
    "label": "Отменить"
  }
]
},
"state2": {
  "label": "На модерации",
  "actions": [
    {
      "bp": {
        "step0": {
          "type": "setState",
          "state": "state3"
        }
      },
      "code": "action1",
      "allow": [
        "moderator"
      ],
      "label": "Модерация пройдена"
    },
    {
      "bp": {
        "step0": {
          "next": "step1",
          "type": "setData",
          "fields": {
            "executor_user_id": {
              "required": true
            }
          }
        },
        "step1": {
          "body": "Вас назначили исполнителем. Свяжитесь с клиентом и подтвердите выезд.",
          "next": "step2",
          "type": "push",
          "title": "Заказ #{{ order.code }}",
          "recipients": [
            "executor"
          ]
        }
      }
    }
  ]
}
```

```
    ]
  },
  "step2": {
    "type": "setState",
    "state": "state4"
  }
},
"code": "action2",
"allow": [
  "moderator"
],
"label": "Назначить исполнителя"
},
{
  "bp": {
    "step0": {
      "next": "step1",
      "type": "setData",
      "fields": {
        "field1416182395": {
          "required": true
        }
      }
    },
    "step1": {
      "body": "{{ clientData['field1416182395'] }}",
      "type": "push",
      "title": "Сообщение от {{ order.customerUser.title }} к заказу #{{ order.code }}",
      "recipients": [
        "moderator"
      ]
    }
  },
  "code": "action1_chat",
  "allow": [
    "customer"
  ],
  "label": "Написать сообщение"
},
```

```
{
  "bp": {
    "step0": {
      "next": "step1",
      "type": "setData",
      "fields": {
        "field1416182395": {
          "required": true
        }
      }
    },
    "step1": {
      "body": "{{ clientData['field1416182395'] }}",
      "type": "push",
      "title": "Сообщение от {{ order.moderatorUser.title }} к заказу #{{ order.code
    }}",
      "recipients": [
        "customer"
      ]
    },
    "code": "action2_chat",
    "allow": [
      "moderator"
    ],
    "label": "Написать сообщение"
  },
  {
    "bp": {
      "step0": {
        "type": "setState",
        "state": "stateCancel"
      }
    },
    "code": "cancel",
    "allow": [
      "customer"
    ],
    "label": "Отменить"
```

```
    },
    {
      "bp": {
        "step0": {
          "type": "setState",
          "state": "stateReject"
        }
      },
      "code": "cancel",
      "allow": [
        "moderator"
      ],
      "label": "Отклонить"
    }
  ]
},
"state3": {
  "label": "Поиск исполнителя",
  "actions": [
    {
      "bp": {
        "step0": {
          "next": "step1",
          "type": "setData",
          "fields": {
            "executor_user_id": {
              "value": "_CURRENT_USER_"
            },
            "hidden_customer_contact": {
              "value": false
            }
          }
        },
        "step1": {
          "type": "setState",
          "state": "state4"
        }
      },
      "code": "action1",
```

```
"allow": [
  "executor"
],
"label": "Взять в работу"
},
{
  "bp": {
    "step0": {
      "next": "step1",
      "type": "setData",
      "fields": {
        "field1416182395": {
          "required": true
        }
      }
    },
    "step1": {
      "body": "{{ clientData['field1416182395'] }}",
      "type": "push",
      "title": "Сообщение от {{ order.customerUser.title }} к заказу #{{ order.code }}",
      "recipients": [
        "moderator"
      ]
    }
  },
  "code": "action1_chat",
  "allow": [
    "customer"
  ],
  "label": "Написать сообщение"
},
{
  "bp": {
    "step0": {
      "next": "step1",
      "type": "setData",
      "fields": {
        "field1416182395": {
          "required": true
        }
      }
    }
  }
}
```



```
    }
  }
},
"step1": {
  "body": "{{ clientData['field1416182395'] }}",
  "type": "push",
  "title": "Сообщение от {{ order.moderatorUser.title }} к заказу #{{ order.code
}}",
  "recipients": [
    "customer"
  ]
}
},
"code": "action2_chat",
"allow": [
  "moderator"
],
"label": "Написать сообщение"
},
{
  "bp": {
    "step0": {
      "type": "setState",
      "state": "stateCancel"
    }
  },
  "code": "cancel",
  "allow": [
    "customer"
  ],
  "label": "Отменить"
},
{
  "bp": {
    "step0": {
      "type": "setState",
      "state": "stateReject"
    }
  },
},
```

```
"code": "cancel",
"allow": [
  "moderator"
],
"label": "Отклонить"
}
],
"onStart": {
  "bp": {
    "step0": {
      "body": "{{ order.service.title}}, {{ order.fieldDeliveryAddress }} {{
order.fieldDateTimeDelivery }} - {{ order.service.organization.currency.sum }} {{
order.service.organization.currency.short_title }}",
      "type": "push",
      "title": "Новый заказ #{{ order.code }}",
      "recipients": [
        "executor"
      ]
    }
  }
}
},
"state4": {
  "label": "Исполнитель назначен",
  "actions": [
    {
      "bp": {
        "step0": {
          "body": "Исполнитель {{ order.executor.title }} подтвердил выезд",
          "next": "step1",
          "type": "push",
          "title": "Заказ #{{ order.code }}",
          "recipients": [
            "customer"
          ]
        },
        "step1": {
          "type": "setState",
          "state": "state5"
```

```
    }
  },
  "code": "action1",
  "allow": [
    "executor"
  ],
  "label": "Подтвердить выезд"
},
{
  "bp": {
    "step0": {
      "next": "step1",
      "type": "setData",
      "fields": {
        "executor_user_id": {
          "value": "_NULL_"
        }
      }
    },
    "step1": {
      "body": "На заказ будет назначен другой исполнитель, ожидайте",
      "next": "step2",
      "type": "push",
      "title": "Заказ #{{ order.code }}",
      "recipients": [
        "customer"
      ]
    },
    "step2": {
      "type": "setState",
      "state": "state3"
    }
  },
  "code": "action2",
  "allow": [
    "executor"
  ],
  "label": "Отменить выезд"
},
```

```
{
  "bp": {
    "step0": {
      "next": "step1",
      "type": "setData",
      "fields": {
        "executor_user_id": {
          "value": "_NULL_"
        }
      }
    },
    "step1": {
      "body": "На заказ будет назначен другой исполнитель, ожидайте",
      "next": "step2",
      "type": "push",
      "title": "Заказ #{{ order.code }}",
      "recipients": [
        "customer"
      ]
    },
    "step2": {
      "type": "setState",
      "state": "state3"
    }
  },
  "code": "action2",
  "allow": [
    "moderator"
  ],
  "label": "Вернуть в поиск"
},
{
  "bp": {
    "step0": {
      "next": "step1",
      "type": "setData",
      "fields": {
        "field1416182395": {
          "required": true
        }
      }
    }
  }
}
```

```
    }
  }
},
"step1": {
  "body": "{{ clientData['field1416182395'] }}",
  "type": "push",
  "title": "Сообщение от {{ order.customerUser.title }} к заказу #{{ order.code }}",
  "recipients": [
    "executor",
    "moderator"
  ]
}
},
"code": "action1_chat",
"allow": [
  "customer"
],
"label": "Написать сообщение"
},
{
  "bp": {
    "step0": {
      "next": "step1",
      "type": "setData",
      "fields": {
        "field1416182395": {
          "required": true
        }
      }
    }
  },
  "step1": {
    "body": "{{ clientData['field1416182395'] }}",
    "type": "push",
    "title": "Сообщение от {{ order.executorUser.title }} к заказу #{{ order.code }}",
    "recipients": [
      "customer"
    ]
  }
},
},
```

```
"code": "action2_chat",
"allow": [
  "executor"
],
"label": "Написать сообщение"
},
{
  "bp": {
    "step0": {
      "next": "step1",
      "type": "setData",
      "fields": {
        "field1416182395": {
          "required": true
        }
      }
    },
    "step1": {
      "body": "{{ clientData['field1416182395'] }}",
      "type": "push",
      "title": "Сообщение от {{ order.moderatorUser.title }} к заказу #{{ order.code
    }}",
      "recipients": [
        "customer"
      ]
    }
  },
  "code": "action3_chat",
  "allow": [
    "moderator"
  ],
  "label": "Написать сообщение"
},
{
  "bp": {
    "step0": {
      "type": "setState",
      "state": "stateReject"
    }
  }
}
```

```
    },
    "code": "cancel",
    "allow": [
      "moderator"
    ],
    "label": "Отклонить"
  }
],
"onStart": {
  "bp": {
    "step0": {
      "body": "Исполнитель найден, ожидайте звонка",
      "type": "push",
      "title": "Заказ #{{ order.code }}",
      "recipients": [
        "customer"
      ]
    }
  }
},
"state5": {
  "label": "В работе",
  "actions": [
    {
      "bp": {
        "step0": {
          "type": "setState",
          "state": "stateDone"
        }
      },
      "code": "action2",
      "allow": [
        "executor"
      ],
      "label": "Завершить заказ"
    },
    {
      "bp": {
```

```
"step0": {
  "next": "step1",
  "type": "setData",
  "fields": {
    "executor_user_id": {
      "required": true
    }
  }
},
"step1": {
  "body": "Вас назначили исполнителем. Свяжитесь с клиентом и подтвердите выезд.",
  "next": "step2",
  "type": "push",
  "title": "Заказ #{{ order.code }}",
  "recipients": [
    "executor"
  ]
},
"step2": {
  "type": "setState",
  "state": "state4"
},
"code": "action2",
"allow": [
  "moderator"
],
"label": "Заменить исполнителя"
},
{
  "bp": {
    "step0": {
      "next": "step1",
      "type": "setData",
      "fields": {
        "field1416182395": {
          "required": true
        }
      }
    }
  }
}
```



```
    },
    "step1": {
      "body": "{{ clientData['field1416182395'] }}",
      "type": "push",
      "title": "Сообщение от {{ order.customerUser.title }} к заказу #{{ order.code }}",
      "recipients": [
        "executor",
        "moderator"
      ]
    }
  },
  "code": "action1_chat",
  "allow": [
    "customer"
  ],
  "label": "Написать сообщение"
},
{
  "bp": {
    "step0": {
      "next": "step1",
      "type": "setData",
      "fields": {
        "field1416182395": {
          "required": true
        }
      }
    },
    "step1": {
      "body": "{{ clientData['field1416182395'] }}",
      "type": "push",
      "title": "Сообщение от {{ order.executorUser.title }} к заказу #{{ order.code }}",
      "recipients": [
        "customer"
      ]
    }
  },
  "code": "action2_chat",
  "allow": [
```

```
    "executor"
  ],
  "label": "Написать сообщение"
},
{
  "bp": {
    "step0": {
      "type": "setState",
      "state": "stateReject"
    }
  },
  "code": "cancel",
  "allow": [
    "moderator"
  ],
  "label": "Отклонить"
}
],
"onStart": {
  "bp": {
    "step0": {
      "type": "setData",
      "fields": {
        "status_id": {
          "value": "ACCEPTED"
        }
      }
    }
  }
}
},
"stateDone": {
  "label": "Выполнен",
  "actions": [
    {
      "bp": [],
      "code": "rate",
      "allow": [
        "customer",
```

```
        "executor"
    ],
    "label": "Оценить выезд"
}
],
"onStart": {
    "bp": {
        "step0": {
            "next": "step1",
            "type": "setData",
            "fields": {
                "status_id": {
                    "value": "DONE"
                }
            }
        },
        "step1": {
            "body": "Заказ #{{ order.code }} выполнен, пожалуйста, оцените",
            "type": "push",
            "title": "{{ order.executor.title }}",
            "recipients": [
                "customer"
            ]
        }
    }
}
},
"stateCancel": {
    "label": "Отменен",
    "onStart": {
        "bp": {
            "step0": {
                "next": "step1",
                "type": "setData",
                "fields": {
                    "status_id": {
                        "value": "REJECTED"
                    }
                }
            }
        }
    }
}
```

```
    },
    "step1": {
      "body": "Заказ #{{ order.code }} отменен",
      "type": "push",
      "title": "{{ order.executor.title }}",
      "recipients": [
        "executor"
      ]
    }
  }
},
"stateReject": {
  "label": "Отклонен",
  "onStart": {
    "bp": {
      "step0": {
        "next": "step1",
        "type": "setData",
        "fields": {
          "status_id": {
            "value": "REJECTED"
          }
        }
      },
    },
  },
  "step1": {
    "body": "Заказ #{{ order.code }} отклонен",
    "type": "push",
    "title": "{{ order.executor.title }}",
    "recipients": [
      "customer"
    ]
  }
}
}
```

Структура

```
{
  "state0": { // Состояние (индексация всегда должна начинаться с 0).
    "label": "Заголовок состояния",
    "onState": { // Действия, выполняемые при переходе в состояние (каждый раз)
      "bp": {
        "step0": { // Индексация шагов всегда должна начинаться с 0
          "type": "Тип действия",
          "next": "step1", // Переход к следующему шагу (необязательный параметр)
          // Дополнительные параметры шага
        },
        "step1": {},
        ...
      }
    },
    "actions": [ // Действия пользователей
      {
        "label": "Заголовок кнопки",
        "code": "action1", // Код действия в формате `[\w]+`
        "allow": [ // Доступ к действию для роли / ролей
          "customer",
          "executor",
          "courier",
          "moderator"
        ],
        "bp": [], // Действия, выполняемые после нажатия кнопки пользователем.
        "visible": { // Условия отображения кнопки (необязательный параметр).
          "conditions": []
        }
      },
      ...
    ]
  },
  ...
}
```

Типы действий

Действие	Описание
<code>if</code>	Проверка условия.
<code>setData</code>	Установка значений заказа (включая значения доп. полей).
<code>setState</code>	Переход к состоянию.

Действие	Описание
<code>pay</code>	Оплата заказчиком: <ul style="list-style-type: none"> • При одностадийной оплате средства списываются сразу. • При двустадийной оплате происходит заморозка средств, после чего их можно списать с помощью метода <code>confirmPay</code> либо вернуть средства с помощью <code>voidPay</code>.
<code>confirmPay</code>	Подтверждение оплаты заказчика.
<code>confirmPayWithoutPayment</code>	Подтверждение оплаты заказчика без фактической оплаты.
<code>confirmPayExecutorWithoutPayment</code>	Подтверждение оплаты исполнителя без фактической оплаты.
<code>voidPay</code>	Отмена оплаты.
<code>registerExecutor</code>	Регистрация исполнителя.
<code>createEntity</code>	Создание сущности.
<code>push</code>	Отправка <code>push</code> -уведомления.
<code>sms</code>	Отправка <code>sms</code> -сообщения.
<code>email</code>	Отправка <code>email</code> -сообщения.

Проверка условия (`if`)

```
{
  "type": "if",
  "conditions": [ // Список условий, выполняемых по порядку, до первого истинного
    [
      [], // Условие в json-формате с поддержкой twig-переменных. См. https://www.yiiframework.com/doc-2.0/yii-commerce-commerce.widget.payment.condition.html
      "step1" // Переход к шагу, если условие истинно
    ],
    // Другие условия
    [
      true, // Выполняется всегда, когда неистинны все предыдущие условия
      "step2"
    ]
  ]
}
```

Установка значений заказа (включая значения доп. полей) (`setData`)

```
{
  "type": "setData",
  "fields": {
    "field*": [], // Вывести все поля заказа и его доп. поля для заполнения со своими настройками
    // Управление конкретными доп. полями
    "field<Code>": { // Управление конкретным полем (необязательный параметр)
      "required": false, // Переписывает значение обязательности доп. поля (необязательный параметр)
      "value": "Значение" // Установка конкретного значения (необязательный параметр)
    },
    // Управление полями заказа
    "<code>": [] // См. структуру заказа в https://admin.crabler.com/constructor/object/view
  }
}
```

Переход к состоянию (`setState`)

```
{
  "type": "setState",
  "state": "stateN" // Указывается состояние, к которому осуществляется переход.
}
```

Оплата заказчиком (`pay`)

```
{
  "type": "pay"
}
```

Подтверждение оплаты заказчика (`confirmPay`)

```
{
  "type": "confirmPay"
}
```

Подтверждение оплаты заказчика без фактической оплаты (`confirmPayWithoutPayment`)

```
{  
  "type": "confirmPayWithoutPayment"  
}
```

Подтверждение оплаты исполнителя без фактической оплаты (`confirmPayExecutorWithoutPayment`)

```
{  
  "type": "confirmPayExecutorWithoutPayment"  
}
```

Отмена оплаты (`voidPay`)

```
{  
  "type": "voidPay"  
}
```

Регистрация исполнителя (`registerExecutor`)

```
{  
  "type": "registerExecutor"  
}
```

Создание сущности (`createEntity`)

```
{  
  "type": "createEntity"  
}
```

Отправка `push`-уведомления (`push`)

```
{  
  "type": "push",
```



```

"title": "Заголовок", // Поддерживает twig-переменные
"body": "Текст", // Поддерживает twig-переменные
"recipients": [ // Получатели (один либо несколько ролей / пользователей)
  "executor", // Название роли. Если у заказа задан исполнитель, то push-уведомление получит
  "<uuid>", // В качестве получателя может быть передан UUID конкретного пользователя.
  "+70001234567" // В качестве получателя может быть передан номер телефона в свободном формате
]
}

```

Отправка sms-сообщения (sms)

```

{
  "type": "sms",
  "message": "Текст сообщения", // Поддерживает twig-переменные
  "recipients": [ // Получатели (один либо несколько ролей / пользователей)
    "executor", // Название роли. Если у заказа задан исполнитель, то push-уведомление получит
    "<uuid>", // В качестве получателя может быть передан UUID конкретного пользователя.
    "+70001234567" // В качестве получателя может быть передан номер телефона в свободном формате
  ]
}

```

Отправка email-сообщения (email)

```

{
  "type": "email",
  "subject": "Тема письма", // Поддерживает twig-переменные
  "text": "Текст письма", // Поддерживает twig-переменные
  "recipients": [ // Получатели (один либо несколько ролей / пользователей)
    "executor", // Название роли. Если у заказа задан исполнитель, то push-уведомление получит
    "<uuid>", // В качестве получателя может быть передан UUID конкретного пользователя.
    "example@crabler.com" // В качестве получателя может быть передан конкретный email.
  ]
}

```

Константы для значений полей

Константа	Значение
<code>_NULL_</code>	<code>null</code>
<code>_CURRENT_USER_</code>	ID текущего пользователя

Twig-переменные

Переменная	Значение
<code>order</code>	Текущий заказ. Можно обращаться к вложенным сущностям.
<code>clientData</code>	Данные, полученные с клиента через <code>POST</code> .