



RUSTACK CLOUD PLATFORM

Интеграция с ERP-системой для автоматизации процесса
оплаты услуг физических лиц

Релиз 1.0.0

2024

Оглавление

1. Термины.....	3
2. Общая информация.....	4
3. Подготовка к интеграции.....	5
4. Интеграция с сервисом ЮKassa.....	7
5. Описание процессов обмена данными.....	8
5.1. Получение контрагентов.....	8
5.1.1. Получение списка клиентов.....	8
5.2. Получение платежей клиентов.....	10
6. Описание структур данных.....	14
6.1. Структура данных «Клиенты».....	14
6.2. Структура данных «Платежи».....	15

1. Термины

Платформа (служба облачных вычислений) — набор облачных услуг («инфраструктуры как сервис», «платформы как сервис» и др.), предоставляемых потребителям через определённый интерфейс. Фактически, это одна или несколько инсталляций платформ виртуализации, в том числе территориально распределённых, под управлением RCP.

Поставщик услуг (провайдер) — организация, продвигающая облачные услуги на рынке и заключающая договора с потребителями услуг. Поставщиком услуг может быть как сам Оператор, так и его бизнес партнёры. Для осуществления своей деятельности Поставщик услуг распоряжается частью ресурсов Службы облачных вычислений, выделенных ему Оператором.

Домен — Интернет-домен, с которого потребителям предоставляются услуги, включая функции регистрации, заказа, оплаты и администрирования услуг. По сути, домен является каналом предоставления услуг, которому назначается некоторая часть ресурсов Платформы. Для домена настраиваются собственные правила работы (лимиты, настройки по умолчанию), интеграция со сторонними службами (DNS, SMS, LDAP, ERP), визуальные настройки white-label. В Оркестраторе можно одновременно управлять несколькими доменами. За каждым доменом должен быть закреплён администратор либо со стороны организации-провайдера, либо партнёрской организации-поставщика услуг. В случае партнёрской схемы можно дополнительно настроить биллинг для взаиморасчётов партнёра с провайдером. В публичных облаках в зависимости от выбранной модели оплаты для домена могут быть сконфигурированы индивидуальные тарифные планы для каждого из потребителей услуг или один общий тарифный план-оферта. Для предоставления услуг в RCP должен быть сконфигурирован хотя бы один партнёр и связанный с ним домен.

Потребителем услуг (клиентом) может выступать организация, группа лиц, физическое лицо. В частном облаке это некоторый коллектив или подразделение, центр затрат для учёта потребления, центр финансовой ответственности. В публичном облаке это хозяйствующий субъект (юридическое или физическое лицо), потребляющий услуги облака по договору с поставщиком услуг (провайдером или его партнёром).

Баланс — отображает текущую сумму на виртуальном счёте потребителя в RCP. Баланс может быть положительным или отрицательным в зависимости от выбранной клиентом системы оплаты. Баланс уменьшается в соответствии с потреблением услуг и увеличивается в соответствии с пополнением виртуального счёта клиента.

Контракт (Договор) — сущность, моделирующая договор между контрагентами (сроки, цены, лимиты), а также лицевой счёт клиента (баланс). Контракты делятся на коммерческие, демо и внутренние. Внутренний контракт предназначен для учёта потребления в организации.

Тарифный план — отношение наименований предоставляемых услуг и цен за единицу услуги. Тарифные планы формируются провайдером услуг. Базовый тарифный план (тарифный план по умолчанию) используется в договоре-оферте, доступен для ознакомления пользователям, используется в калькуляторе для оценки стоимости. При заключении контракта выбирается или создаётся новый тарифный план.

ЮKassa (Яндекс касса) — платёжный провайдер, обеспечивает сервис по приёму платежей через Интернет в пользу юридических лиц, индивидуальных предпринимателей и некоммерческих организаций.

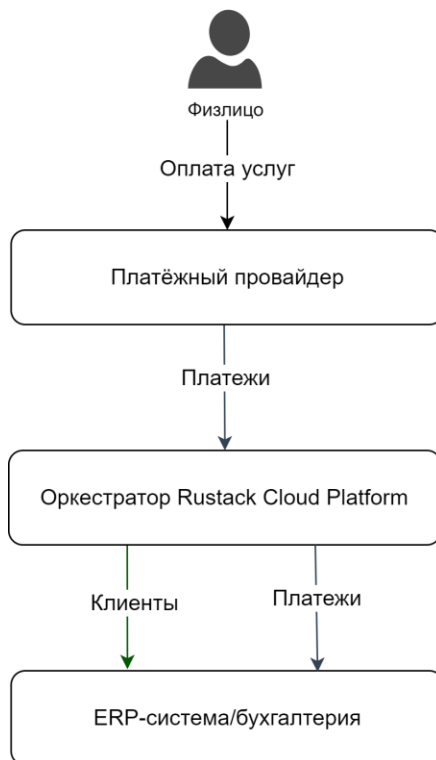
2. Общая информация

Расчёты провайдера с клиентами (физическими лицами) осуществляются по предоплатной системе. Клиент сначала пополняет баланс, а затем использует ресурсы публичного облака. Деньги списываются со счёта клиента в портале самообслуживания RCP в соответствии с тарифным планом.

Пополнение баланса выполняется через интегрированную систему оплаты банковскими картами ЮKassa. Интернет-эквайринг ЮKassa позволяет принимать онлайн-платежи различными способами. При помощи ЮKassa пользователи облачных сервисов могут вносить оплату с банковских карт, электронными деньгами, со счёта мобильного телефона, через интернет-банк. Для интеграции с ЮKassa необходимо заключить соглашение с ЮKassa и задать соответствующие настройки в портале самообслуживания.

Оркестратор RCP получает информацию о платеже из ЮKassa, фиксирует его и далее информация о платежах загружается в ERP-систему. Интеграция Оркестратора с ERP-системой упрощает получение информации о платежах клиентов сотрудниками бухгалтерии.

На схеме представлены потоки передачи данных и их направление:



Подробное описание потоков данных приведено в разделе [Описание структур данных](#).

Процессы обмена данными имеют различные условия инициации и периодичность.

№	Процесс	Условия инициации
1	Обновление справочников контрагентов в ERP-системе на основе данных из Оркестратора	Ежедневно в автоматическом режиме регламентным заданием из ERP-системы, либо вручную при необходимости
2	Сопоставление контрагентов в ERP-системе с клиентами из Оркестратора для автоматического учёта платежей	Пользователь ERP-системы вручную после загрузки данных о новых клиентах
3	Передача информации о платежах физических лиц из Оркестратора в ERP-систему	Пользователь ERP-системы вручную после окончания отчётного периода (месяца)

3. Подготовка к интеграции

Используйте учётную запись пользователя с ролью бухгалтера в портале самообслуживания RCP. Дальнейшее подключение ERP-системы к Оркестратору выполняется под данной учётной записью через программный интерфейс с помощью токена доступа. Программный интерфейс Оркестратора RCP доступен по ссылкам:

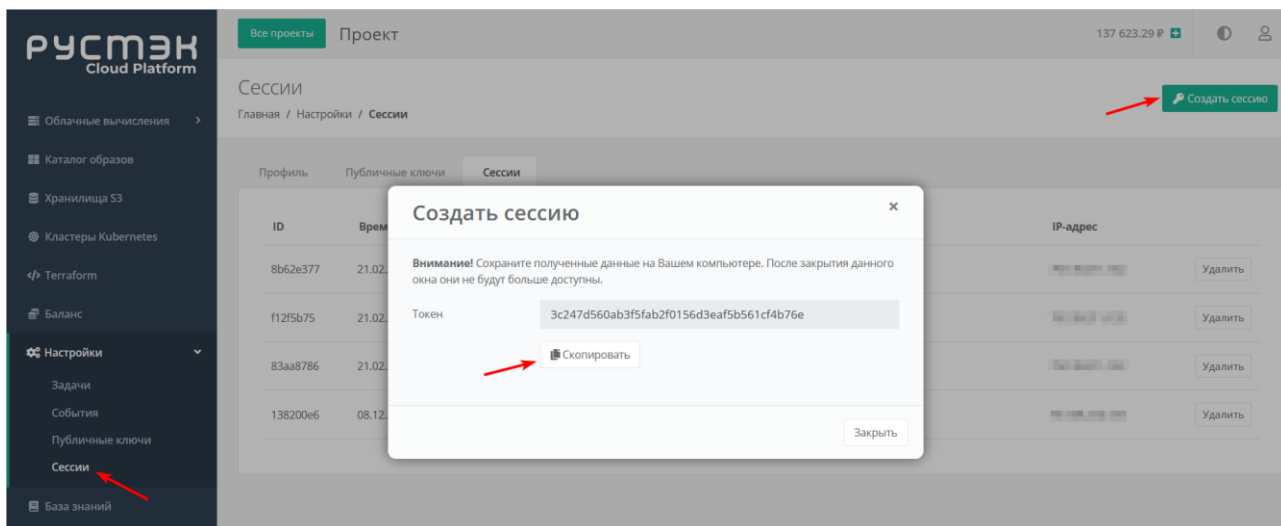
REST API: `https://{{control_panel_address}}/v1`

Swagger: `https://{{control_panel_address}}/swagger/`

Здесь `{{control_panel_address}}` — адрес портала самообслуживания, например, `newcrp.cloud.online`. Совпадает с доменом личного кабинета.

Чтобы получить токен доступа в портале самообслуживания:

1. Авторизуйтесь под учётной записью бухгалтера.
2. Перейдите в раздел меню **Настройки** → **Сессии** и нажмите кнопку **Создать сессию**.
3. В открывшемся окне введите пароль и нажмите кнопку **Применить**.
4. В окне отобразится токен, скопируйте его.



Сохраните полученный токен доступа к Оркестратору в настройках ERP-системы для дальнейшего обмена данными. Токен отправляется в заголовках последующих запросов к API: `-H 'Authorization: Bearer $TOKEN'`.

Токен может быть получен и программным способом при наличии учётной записи (логина и пароля) с помощью метода **POST** по эндпоинту `/v1/auth/token`:

```
curl -X 'POST' 'https://{{control_panel_address}}/v1/auth/token' \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{"code": "string","domain": "default","login": "user_login","password": "user_password","ttl": 1314000}'
```

Заполните поля запроса:

- `domain` — домен, в котором зарегистрирован пользователь (`default`),
- `login` — логин пользователя (`user_login`),
- `password` — пароль пользователя (`user_password`).

Ответ со статусом 200 будет содержать JSON-объект с токеном:

```
{  
  "id": "16f3b266-d622-4b83-b541-90a50d7f2107",  
  "key": "da3730bfe8f2e9fff1450c67a577b46dcb1b9df6",  
  "ctime": "2023-04-05T17:09:15.799058",  
  "issued_device_info": "string",  
  "issued_ip_address": "string",  
  "expires": "2023-04-20T22:09:15.795989"  
}
```

Значение поля "key" будет содержать токен пользователя — `$TOKEN`. В данном примере это `da3730bfe8f2e9fff1450c67a577b46dcb1b9df6`.

Помимо настроек систем, в компании должны быть выстроены соответствующие бизнес-процессы по ведению справочников.

На этапе регистрации нового клиента в Оркестраторе в справочнике ERP-системы должна быть добавлена информация о новом контрагенте, так как в ERP и в Оркестраторе ведётся параллельный учет контрагентов (клиентов). Создание нового контрагента в ERP-системе выполняет ответственный сотрудник в соответствии с принятыми в компании регламентами по ведению справочников.

Сопоставление клиентов в Оркестраторе с контрагентами в ERP выполняется вручную в ERP, подробнее в подразделе [Получение контрагентов](#).

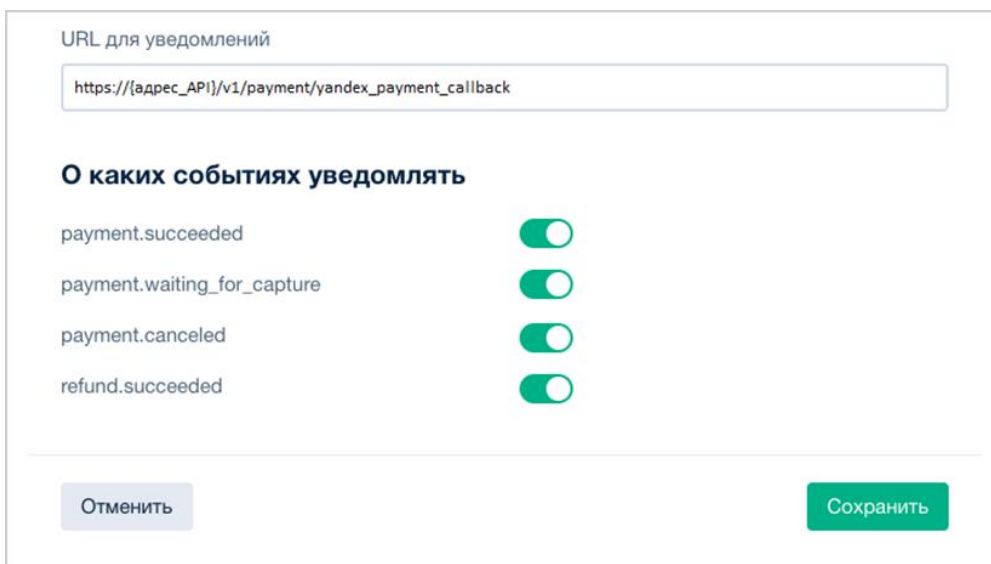
4. Интеграция с сервисом ЮKassa

Для приёма платежей в портале самообслуживания через банковские карты подключите сервис ЮKassa <https://yookassa.ru/>.

Интеграция с сервисом ЮKassa выполняется администратором платформы для домена, с которого клиенты будут оплачивать облачные ресурсы посредством ЮKassa.

Чтобы подключить сервис ЮKassa:

1. Зарегистрируйтесь на сайте <https://yookassa.ru> и получите идентификатор (ID) магазина и секретный ключ, подробнее см. в [официальной документации сервиса](#).
2. Перейдите в раздел меню **Администрирование** → **Партнёры**.
3. Нажмите на имя выбранного партнёра или на кнопку **Действия** → **Изменить**.
4. На вкладке **Основные настройки** введите полученный идентификатор магазина и секретный ключ в соответствующие поля. При необходимости установите флаг **Разрешить автоплатежи** — в этом случае для администратора клиента с атрибутом «Владелец» будет доступно автопополнение баланса клиента.
5. Нажмите кнопку **Изменить** для сохранения настроек.
6. Выполните настройку HTTP-уведомлений в личном кабинете ЮKassa для отправки уведомлений о пополнении в портал самообслуживания (см. рисунок ниже). URL для уведомлений: `https://{адрес_API}/v1/payment/yandex_payment_callback`.



Требования к URL для уведомлений — протокол HTTPS и TCP-порт 443 или 8443. TLS/SSL-сертификат подойдет любой: самоподписанный или выданный центром сертификации. Версия TLS/SSL — 1.2 или выше.

Подробнее на <https://yookassa.ru/developers/using-api/webhooks#configuration>.

Для проверки работы интеграции рекомендуется использовать тестовый магазин, подробнее на <https://yookassa.ru/developers/payment-acceptance/testing-and-going-live/testing>.

В портале самообслуживания метод оплаты ЮKassa для клиента задаётся администратором партнёра при создании или изменении клиента в меню **Администрирование** → **Клиенты**. При создании или изменении клиента выберите **ЮKassa** в поле **Методы оплаты**.

Администратор платформы также может задать метод оплаты в настройках клиентов по умолчанию при создании и редактировании партнёров.

5. Описание процессов обмена данными

5.1. Получение контрагентов

В ERP-системе платежи за предоставленные услуги связаны с контрагентами, а в Оркестраторе с клиентами.

Загрузку контрагентов рекомендуется выполнять ежедневно в автоматическом режиме регламентным заданием ERP-системы.

В случаях нарушения работы регламентного задания или необходимости оперативной загрузки данных уполномоченный пользователь ERP-системы может запустить загрузку контрагентов.

После загрузки в ERP-систему информации о новых контрагентах пользователь ERP должен сопоставить их с соответствующими сущностями в ERP (процесс не автоматизирован).

5.1.1. Получение списка клиентов

Чтобы получить список клиентов в Оркестраторе, обратитесь методом **GET** к эндпоинту `/v1/client`:

```
curl 'https://{{control_panel_address}}/v1/client' -H 'Accept: application/json' -H 'Authorization: Bearer $TOKEN'
```

Ответ со статусом 200 будет содержать массив объектов-клиентов в секции `items`, а также поля `total` и `limit`:

```
{
  "items": [
    {
      "id": "e2cb4854-53c7-456c-84f8-901e0700c876", // идентификатор клиента
      "ctime": "2023-03-31T06:38:43.051283", // дата создания клиента
      "projects": [
        {...},
        {...}
      ],
      "contact_emails": [ // список администраторов клиента
        {
          "id": "02b665e8-42b7-4167-8e16-0b7eeab982fe",
          "username": "Николай Красовский",
          "email": null,
          "phone": ""
        },
        {...}
      ],
      "partner": {
        "id": "2e3be96a-e389-4cf2-98f8-34096a49ecdc", // идентификатор партнёра
        "name": "ООО Облачное облако" // название партнёра
      },
      "internet": 1000,
      "allowed_hypervisors": [
        {...},
        {...}
      ],
      "lan": 1000,
      "payment_model": "prepay", // модель оплаты клиента
      "payment_methods": [ // массив доступных методов оплаты
        {
          "id": "yandex", // идентификатор метода оплаты

```



```

        "name": "ЮKassa", // название метода оплаты
        "min_amount": 0
    }
},
"resource_planning": false,
"contract": {
    "id": "dda4a3f0-3aa4-4fa6-be7a-d6774d9506fd",
    "balance": -3917.92, // баланс клиента
    "counterparty_left": {
        "id": "2e3be96a-e389-4cf2-98f8-34096a49ecdc",
        "name": "ООО \"Облачное облако\"",
        "entity": "partner"
    },
    "counterparty_right": {
        "id": "e2cb4854-53c7-456c-84f8-901e0700c876",
        "name": "ООО \"СмартМониторинг\"",
        "entity": "client"
    },
    "inactive": false,
    "is_new": false,
    "name": "Контракт для клиента ООО \"СмартМониторинг\"",
    "billing_plan": { // объект тарифного плана клиента
        "id": "a8507ae8-5968-4979-b655-86adf46a2fa6", // идентификатор тарифного
        "name": "Выгодный план для клиентов", // название тарифного плана
        "contracts": 4
    }
},
"name": "ООО СмартМониторинг", // название клиента
"billing_enabled": true,
"billing_visible": true
},
...
],
"total": 73, // всего элементов
"limit": 10 // элементов на страницу
}

```

Из ответа зафиксируйте поля:

- идентификатор и название клиента,
- идентификатор и название партнёра,
- модель оплаты клиента:
 - `prepay` — предоплата,
 - `postpay` — постоплата,
- информацию о контракте,
- идентификаторы и названия доступных методов оплаты,
- идентификатор и название тарифного плана,
- баланс клиента.

В случае, если значение поля `total` больше значения поля `limit`, следует рассчитать количество страниц, как результат деления `total` на `limit` и округленное до единиц в большую сторону, и запросить последовательно все страницы, добавив к запросу параметр `page` со значением равным номеру страницы. Следует обратить внимание, что если не передать параметр `page`, то в ответе будет содержаться первые {значение `limit`} элементов списка.

```
curl 'https://{{control_panel_address}}/v1/client?page=1' -H 'Accept: application/json' -H 'Authorization: Bearer $TOKEN'
```

5.2. Получение платежей клиентов

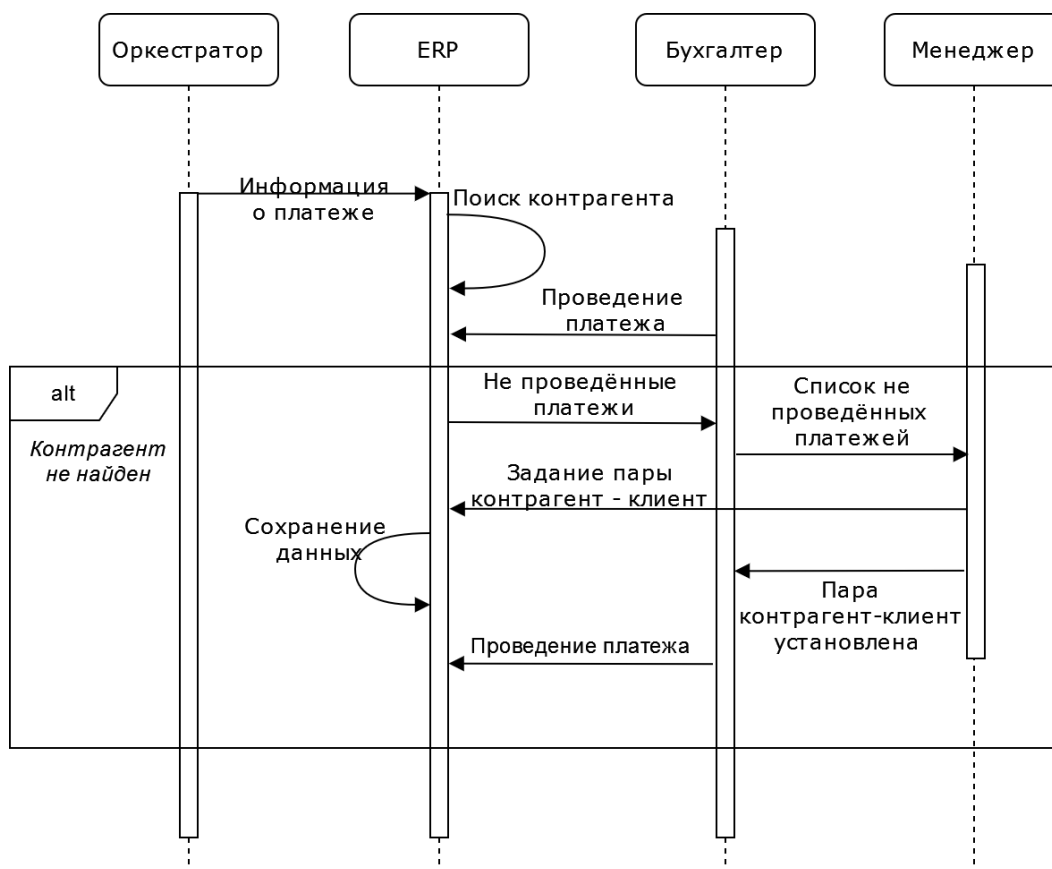
В Оркестраторе регистрируются платежи клиентов. Для ведения бухгалтерского и управленческого учёта платежи загружаются в ERP-систему.

Получение информации о платежах физических лиц запускается вручную по завершению отчётного периода (месяца).

После загрузки платежа ERP-система производит поиск контрагента. В случае определения контрагента ERP-система сохраняет платёж. Если такого сопоставления не найдено, загрузка платежа не будет выполнена до появления связи контрагента ERP-системы с клиентом портала самообслуживания. В этом случае сотрудник бухгалтерии привлекает к работе по настройке связи уполномоченного пользователя ERP-системы (менеджера).

Проведение платежей выполняется под контролем сотрудников бухгалтерии. Поэтому проведение платежей происходит не автоматически, а при создании бухгалтером соответствующего платёжного документа. При создании платёжного документа его заполнение должно происходить автоматически на основе загруженных данных. Таким образом бухгалтер может проверить документ и внести корректировки. Затем документ проводится и производится движение по регистрам. Контроль платежей бухгалтер может выполнять по информации в личном кабинете ЮKassa или настроить информирование о платежах на электронную почту.

Загрузка и проведение платежей:



Чтобы получить все платежи, зафиксированные в Оркестраторе, обратитесь методом **GET** к эндпоинту `/v1/payment`:

```
curl 'https://{control_panel_address}/v1/payment' -H 'Accept: application/json' -
H 'Authorization: Bearer $TOKEN'
```

Ответ со статусом 200 будет содержать массив объектов платежей `items`:

```
{
  "items": [
    {
      "id": "b62c65bc-0dc6-45aa-9ac7-b9d0a4612c3f", // идентификатор платежа
      "contract": {
        "id": "abf6911c-7c3b-4566-9adf-4f8ee9308d7c", // идентификатор контракта
        (договора)
        "name": "Контракт для клиента Клиент 310823-23", // название контракта
        "balance": 201.0 // текущий баланс
      },
      "client": {
        "id": "1ad433c2-7b79-4f85-bc1c-976e1484dbdd", // идентификатор клиента
        "name": "Клиент 310823-23" // имя клиента
      },
      "payment_method": {
        "id": "yandex", // идентификатор метода оплаты
        "name": "ЮKassa" // название метода оплаты
      },
      "transaction_id": "2c824d74-000f-5000-a000-1cc72978ef03", // идентификатор
      транзакции
      "status": "succeeded", // статус платежа
      "error_reason": "", // причина отмены платежа
      "amount": 100.0, // сумма платежа
      "ctime": "2023-08-31T09:58:31.223712", // дата и время выполнения платежа
      "user": { // информация о пользователе, который выполнил платёж или коррекцию
        баланса клиента
        "id": "7455358b-4c9c-4ccc-8441-6a79f8cbd361", // идентификатор пользователя
        "domain": {
          "id": "00000000-0000-0000-0000-000000000000",
          "name": "default",
          "aliases": [
            {
              "id": "2a597deb-27bc-4054-b2e6-be1961d9806c",
              "alias": "default"
            }
          ]
        }
      },
      "login": "anton_frolov@yandex.ru", // логин пользователя
      "username": "Антон Фролов", // имя пользователя
      "is_ghost": false
    }
  ],
  {
    "id": "dfd71aff-3f60-4dd4-9dce-bb135a165b2c",
    "contract": {
      "id": "dda4a3f0-3aa4-4fa6-be7a-d6774d9506fd",
      "name": "Контракт для клиента ООО \"СмартМониторинг\"",
      "balance": -24245.68
    },
    "client": {
      "id": "e2cb4854-53c7-456c-84f8-901e0700c876",
      "name": "ООО \"СмартМониторинг\""
    },
    "payment_method": {
      "id": "internal",

```

```

        "name": "Коррекция"
    },
    "transaction_id": "Коррекция",
    "status": "succeeded",
    "error_reason": "",
    "amount": 150000,
    "ctime": "2023-06-01T12:03:41.566512",
    "user": {
        "id": "f58f009e-1645-4d43-b3f4-076caf3ac6af",
        "domain": {
            "id": "00000000-0000-0000-0000-000000000000",
            "name": "default",
            "aliases": [
                {
                    "id": "2a597deb-27bc-4054-b2e6-be1961d9806c",
                    "alias": "default"
                }
            ]
        },
        "login": "petrol123123@mail.com",
        "username": "Администратор партнёра",
        "is_ghost": false
    }
    },
    ...
],
"total": 374,
"limit": 10
}

```

В случае, если значение поля `total` больше значения поля `limit`, следует рассчитать количество страниц, как результат деления `total` на `limit` и округленное до единиц в большую сторону, и запросить последовательно все страницы, добавив к запросу параметр `page` со значением равным номеру страницы. Следует обратить внимание, что если не передать параметр `page`, то в ответе будет содержаться первые `{значение limit}` элементов списка.

```

curl 'https://{control_panel_address}/v1/payment?page=1' -H 'Accept: application/json' -H 'Authorization: Bearer $TOKEN'

```

Статус платежа определяется значением поля `status`:

- `pending` — платёж оплачен, деньги авторизованы и ожидают списания; из этого статуса платёж может перейти в `succeeded`, если оплата списана, или `canceled`, если платёж отменён;
- `succeeded` — платёж успешно завершён, деньги будут перечислены на расчётный счёт в соответствии с договором; это финальный и неизменяемый статус;
- `canceled` — платёж отменён; статус устанавливается, если истекло время на принятие платежа или платёж был отклонен платёжным провайдером; это финальный и неизменяемый статус.

Метод оплаты определяется значением поля `payment_method.id`:

- `yandex` — оплата с помощью сервиса ЮKassa,
- `bank` — безналичная оплата,
- `internal` — коррекция баланса со стороны вышестоящего администратора или системы.

Значение поля `user` содержит объект с информацией о пользователе, который выполнил платёж: идентификатор, домен, логин и имя пользователя. Если была выполнена коррекция баланса, объект `user` будет содержать информацию об администраторе, который её выполнил.

Список платежей может быть отфильтрован и отсортирован. Фильтры и сортировки могут быть применены по отдельности или вместе.

Доступные параметры запроса (`query params`) для фильтрации:

Параметр	Значение
<code>contract</code>	идентификатор контракта (договора)
<code>ctime_before</code>	конец периода в формате гггг-мм-дд
<code>ctime_after</code>	начало периода в формате гггг-мм-дд
<code>client</code>	идентификатор клиента
<code>status</code>	статус платежа: <code>pending</code> , <code>succeeded</code> , <code>canceled</code>
<code>payment_method</code>	метод оплаты: <code>yandex</code> , <code>bank</code> , <code>internal</code>
<code>transaction_id</code>	идентификатор транзакции

При фильтрации списка по датам:

1. Чтобы получить все платежи **до** определённой даты, отправьте запрос с параметром `ctime_before` со значением в формате гггг-мм-дд.
2. Чтобы получить все платежи **после** определённой даты, отправьте запрос с параметром `ctime_after` со значением в формате гггг-мм-дд.
3. Чтобы получить все платежи за определённый **период**, отправьте запрос с параметрами `ctime_after` (начало периода) и `ctime_before` (конец периода) в формате гггг-мм-дд.

Пример запроса для получения списка платежей за определённый период:

```
curl 'https://{{control_panel_address}}/v1/payment?page=1&ctime_after=2022-10-05&ctime_before=2022-10-20' -H 'Accept: application/json' -H 'Authorization: Bearer $TOKEN'
```

Пример запроса с фильтрацией по одному клиенту:

```
curl 'https://{{control_panel_address}}/v1/payment?page=1&client=e2cb4854-53c7-456c-84f8-901e0700c876' -H 'Accept: application/json' -H 'Authorization: Bearer $TOKEN'
```

Список может быть отсортирован по параметрам:

- `amount` — сумма платежа,
- `contract` — название контракта,
- `ctime` — дата платежа,
- `id` — идентификатор платежа,
- `payment_method` — метод оплаты,
- `transaction_id` — идентификатор транзакции.

По умолчанию порядок сортировки по возрастанию, для выбора порядка по убыванию добавьте символ "-" перед параметром, например: `-amount`.

Пример запроса с сортировкой по сумме платежа:

```
curl 'https://{{control_panel_address}}/v1/payment?page=1&sort=amount' -H 'Accept: application/json' -H 'Authorization: Bearer $TOKEN'
```

6. Описание структур данных

6.1. Структура данных «Клиенты»

№	Реквизит	Описание	Тип поля	Обязательное	Ограничение допустимых значений
1	id	Идентификатор клиента	string (uuid)	Да	
2	name	Наименование клиента	string		
3	ctime	Дата создания	datetime		Формат даты ISO 8601: YYYY-MM-DDT00:00:00+03:00, Где YYYY-MM-DD – год-месяц-число
4	partner.id	Идентификатор партнёра	string (uuid)		
5	partner.name	Наименование партнёра	string		
6	payment_model	Модель оплаты клиента	string		Возможные значения: <code>prepay</code> , <code>postpay</code>
7	payment_methods.id	Идентификатор метода оплаты	string (uuid)		
8	payment_methods.name	Наименование метода оплаты	string		Возможные значения: <code>yandex</code> , <code>bank</code> , <code>internal</code>
9	payment_methods.min_amount	Минимальная сумма	integer		
10	contract.balance	Баланс клиента	float		
11	contract.billing_plan.id	Идентификатор тарифного плана клиента	string (uuid)		
12	contract.billing_plan.name	Наименование тарифного плана клиента	string		

6.2. Структура данных «Платежи»

№	Реквизит	Описание	Тип поля	Ограничение допустимых значений
1	id	Идентификатор платежа	string (uuid)	
2	contract.id	Идентификатор контракта	string (uuid)	
3	contract.name	Название контракта	string	
4	contract.balance	Текущий баланс	float	
5	client.id	Идентификатор клиента	string (uuid)	
6	client.name	Имя клиента	string	
7	payment_method.id	Идентификатор метода оплаты	string	Возможные значения: yandex, bank, internal
8	payment_method.name	Название метода оплаты	string	
9	transaction_id	Идентификатор транзакции	string (uuid)	
10	status	Статус платежа	string	Возможные значения: pending, succeeded, canceled
11	error_reason	Причина отмены платежа	string	
12	amount	Сумма платежа	float	
13	ctime	Дата и время выполнения платежа	datetime	Формат даты ISO 8601: YYYY-MM-DDT00:00:00+03:00, где YYYY-MM-DD — год-месяц-число
14	user.id	Идентификатор пользователя	string (uuid)	
15	user.domain	Информация о домене	объект	
16	user.login	Логин пользователя	string	
17	user.username	Имя пользователя	string	