

## Документация:

- Техническая документация. (Основания разработки, Требования, Архитектура, API, Исходники)
- Пользовательская документация. (Как этим пользоваться)

## Пользовательская документация:

- Вводное руководство
  1. Термины и сокращения
  2. Введение
  3. Объекты системы
    - 3.1. Тип счета
    - 3.2. Схема обработки операции
    - 3.3. Операция
    - 3.4. Аналитический счет
    - 3.5. Проводка
    - 3.6. Синтетический счет
    - 3.7. План счетов
    - 3.8. Закрытый период
- Руководство администратора
  4. Установка ПО
    - 4.1. Системные требования
    - 4.2. Установка программы
    - 4.3. Запуск сервисов
    - 4.4. Остановка сервисов
    - 4.5. Настройки сервисов
    - 4.6. Удаление программы
    - 4.7. Техническая поддержка
    - 4.8. Контакты разработчика
- Руководство пользователя
  5. Настройка
    - 5.1. Синтаксис "Правил".
    - 5.2. Настройка Типов Счетов
    - 5.3. Настройка Схем Операций
    - 5.4. Настройка Плана счетов
    - 5.5. Настройка Счетов синтетического учета
    - 5.6. Настройка аналитических счетов
  6. Эксплуатация
    - 6.1. Регистрация Операций
    - 6.2. Просмотр проводок
    - 6.3. Просмотр аналитических счетов
    - 6.4. Остатки аналитических счетов
    - 6.5. Обороты аналитических счетов
    - 6.6. Оборотно-сальдовая ведомость по синтетическим счетам.
    - 6.7. Закрытие периода
    - 6.8. Расчет корректировок

# Вводное руководство

## 1. Термины и сокращения

**ДС** – денежные средства

**Система учета (СУ)** - информационная система для сбора, регистрации, обработки и представления данных об активах, хозяйственных и других операциях организации.

**Учетный модуль (далее ПО)** - программное обеспечение, служащее для организации и обеспечения работы "Система учета" организации или группы компаний.

**ИС** – информационная система

**Первичные документы** — это оформленные надлежащим образом документы, при помощи которых компания оформляет произошедшие на предприятии хозяйственные события (п. 1 ст. 9 закона «О бухучете» от 06.12.2011 № 402-ФЗ) и которые служат основанием для отражения записей в учетных регистрах.

**Операция** - действие или их совокупность для достижения какой-либо цели. В рамках системы учета можно трактовать как совокупность проводок, созданных согласно схеме на основании переданных параметров, для учета хозяйственной операции.

**Проводка** - это запись об изменении состояния учитываемых объектов.

**Двойная запись** - способ ведения бухгалтерского учета, при котором каждое изменение состояния средств организации отражается по крайней мере на двух бухгалтерских счетах, обеспечивая общий баланс.

**Счет** - это особый регистр, представляющий собой способ экономической группировки и учета однородных по экономическому содержанию хозяйственных средств, их источников, хозяйственных процессов и финансовых результатов.

**Аналитический счет** - это счет бухгалтерского учета, которые предназначены для детализации и конкретизации информации о наличии, состоянии и движении средств и их источников

**Объект учета** - конкретный договор, клиент, касса, сотрудник и т.п. в отношении которого ведется учет

**Тип объекта учета** - договор, клиент, касса, сотрудник и т.п.

**Тип счета** - экономическое содержание хозяйственных средств

**Баланс (Остаток)** - денежная оценка стоимости имущества или имущественных прав, учитываемых на счёте

**Оборот** - суммарная денежная оценка всех хозяйственных операций за промежуток времени, приведших к изменению имущества/имущественных прав, которые учитываются на рассматриваемом счёте.

**Синтетический учёт** - обобщённое отражение в денежном измерении экономически однородных хозяйственных средств.

**Синтетический счет** - счет для синтетического учета

**План счетов** - Иерархически организованная система счетов

**Учет движения ДС** - система упорядоченного сбора, регистрации и обобщения информации в денежном выражении о состоянии имущества, обязательствах и капитале организации, и их изменениях путём сплошного, непрерывного и документального отражения всех хозяйственных операций.

**Субсчет** - способ сводной группировки данных учёта. Иерархически субсчета являются бухгалтерскими счетами второго порядка и используются для получения более детальных показателей в дополнение к данным синтетических счетов

**Сторнировать** - исправлять, отменять. В бухгалтерии термин используется в случае необходимости внести корректирующие данные или запись.

**График платежей** - это документ, наделенный юридической силой с таблицей, в которой прописаны все предстоящие платежи.

**Договор займа** - это соглашение, в котором одна сторона (Займодавец) передает в собственность другой стороне (Заемщику) деньги или другие вещи, определенные родовыми признаками на определённый срок.

**Касса** - участок бухгалтерского учёта, предназначенный для отражения информации о движении наличных денежных средств и денежных документов

**ЕПС** - единый план счетов бухгалтерского учета.

**ОД** - основной долг.

**Сущность** - смысл данной вещи, то, что она есть сама по себе, в отличие от всех других вещей и в отличие от изменчивых (под влиянием тех или иных обстоятельств) состояний вещи.

**Атрибут** - отличительная принадлежность, присвоенный кому-либо или чему-либо знак, предмет для отличия.

**Схема обработки операции** - Настройка обработки операции, с целью корректного учета хозяйственных операций на счетах системы учета. Включает в себя описание параметров операции, правила генерации проводок. Обязательно включает название операции и дату совершения.

**Действующая схема** - схема операции с максимальной датой начала действия из схем, у которых дата начала действия меньше текущей даты.

**Планируемая схема** - Схема с датой начала действия больше текущей даты.

**Архивные схемы** - Схемы операции с датой начала действия меньше максимальной даты начала действия из схем операции, у которых дата начала действия меньше текущей даты.

**Параметры схемы обработки операций (Параметры)** - Названия параметров, передаваемых системой учета при регистрации операции и их тип. В дальнейшем это название используется при создании Правил.

**Правила формирования проводок (Правила)** - Перечень выражений, вычисляемых при регистрации операции и определяющие параметры проводок, которые необходимо зарегистрировать в системе.

**REST** (сокращение от англ. Representational State Transfer - «передача состояния представления») - архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

**Микросервисная архитектура** - вариант сервис-ориентированной архитектуры программного обеспечения, ориентированный на взаимодействие насколько это возможно небольших, слабо связанных и легко изменяемых модулей — микросервисов

**JSON** (англ. JavaScript Object Notation) - текстовый формат обмена данными, основанный на JavaScript.

**.Net Core 3.0** - обеспечивает проверку подлинности в одностраничных приложениях с помощью поддержки авторизации веб-API.

**ASP.NET Core** - фреймворк для создания веб-приложений на платформе .NET с открытым исходным кодом.

**ADO.NET Entity Framework** - объектно-ориентированная технология доступа к данным, является object-relational mapping (ORM) решением для .NET Framework от Microsoft. Предоставляет возможность взаимодействия с объектами как посредством LINQ в виде LINQ to Entities, так и с использованием Entity SQL.

**Одностраничное приложение (SPA)** - это веб-приложение или веб-сайт, использующий единственный HTML-документ как оболочку для всех веб-страниц и организующий взаимодействие с пользователем через динамически подгружаемые HTML, CSS, JavaScript, обычно посредством AJAX.

**Идентификатор (ИД)** - уникальный признак объекта, позволяющий отличать его от других объектов, то есть идентифицировать.

## 2. Введение

ПО "Учетный модуль" предназначено для встраивания в ИС компании или группы компаний посредством Rest API.

ПО реализует функции:

- Ведение учета на счетах путем двойной записи;
- Соблюдения принципов бухгалтерского учета;
- Ведения информации о счетах:
  - Настройки "Типов счетов" учета и правил формирования реквизитов и атрибутов счетов в соответствии с нормативными документами организации;
  - Хранение и доступ к реквизитам и атрибутам счетов;
  - Получение информации о балансе счета;
  - Получение оборотов по счету;
  - Получение проводок по счету.
- Настройки типовых операций:
  - Настройка "Схем учета операций" в соответствии с нормативными документами организации;
  - Настройка "Параметров" "Схем учета операций";
  - Настройка "Правила формирования проводок" для "Схем учета операций";
- Обработку операций:
  - Автоматическое создание необходимых аналитических счетов;
  - Формирование и регистрация "Проводок" в соответствии с "Правилами";
- Блокировка несанкционированных изменений.
  - Обеспечение неизменности информации отчетов;
  - Автоматический расчет и регистрация корректирующих проводок.
- Ведения информации об операциях:
  - Хранение и доступ к параметрам "Операций";
  - Получение списка "Операций";
  - Получение списка "Проводок", зарегистрированных при обработке "Операции";
- Регистрация и настройка планов счетов:
  - Настройка иерархии счетов синтетического учета;
  - Настройка отображения оборотно-сальдовой ведомости.
- Получение оборотно-сальдовой ведомости с детализацией по атрибутам "счетов".

ПО предоставляет Web интерфейс для настройки и управления расчетным модулем.

### 3. Объекты системы

Учетный модуль состоит из объектов:

- обеспечивающих настройку функций;
- обеспечивающих учет.

Настройку функций обеспечивают:

- Тип счета;
- Схема обработки операции;
- План счетов;
- Закрытый период.

Учет обеспечивают:

- Аналитические счета;
- Проводка;
- Операция.

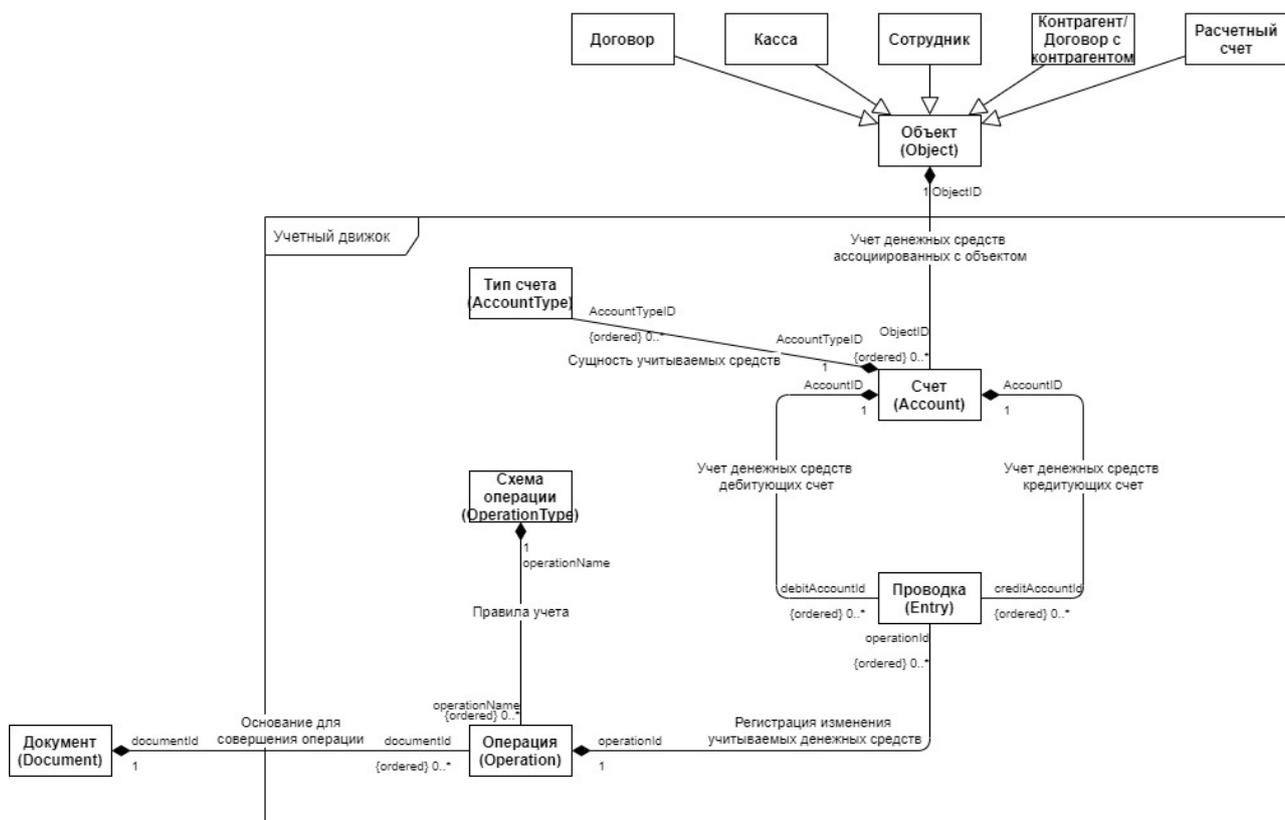


Рисунок 1: Схема объектов учетного модуля

#### 3.1. Тип счета

Объект "Тип счета" определяет экономическое содержание хозяйственных средств и позволяет гибко настраивать атрибуты счетов необходимые для настройки учета в соответствии с принятой учетной политикой организации.

Обязательными параметрами "Типа счета" являются:

- Название;

- Признак активности (активный, пассивный, активно-пассивный).

Дополнительно настраиваются параметры необходимые для идентификации счета, его атрибуты и параметры группировки счетов.

Дополнительный параметр состоит из:

- Название;
- Человеко читаемое описание;
- Дату начала действия "Правила";
- "Правило" заполнения значения параметра;

**Название** - однозначно, в пределах объекта, определяет параметр

**Человеко-читаемое описание параметра** - описание сущности параметра.

**Дата начала действия правила** - определяет с какого момента применяется параметр и его правило.

При автоматическом создании аналитического счета, создаются параметры "действующие" на момент создания со значениями заполненными согласно действующего "Правила". (см. 5.1. Синтаксис "Правил".)

Пример дополнительных параметров счета:

- Название офиса
- Код офиса.
- Процентная ставка по договору.
- Вид займа.
- И т.п.

Пример объекта «Тип счёта»:

```
{
  "accountTypeId": "cf613df5-c7ee-47a0-b2bf-1d75eacaed43",
  "accountTypeName": "К а с с а",
  "apType": " а к т и в н ы й",
  "extParameterRules": [
    {
      "name": "EPS",
      "dateFrom": "2021-12-31T21:00:00.000000Z",
      "description": "Номер счета согласно ЕПС",
      "rule":
"concat(¥"202028100¥", Format(toNumber(GetParameter(concat(¥"http://ip:5000/checkouts/¥", objectId), ¥"number¥")), ¥"00000000000¥"))"
    },
    {
      "name": "officeName",
      "dateFrom": "2021-12-31T21:00:00.000000Z",
      "description": "Офис к которому относится касса",
      "rule": "GetParameter(concat(¥"http://ip:5000/checkouts/¥", objectId), ¥"officeName¥")"
    },
    {
      "name": "region",
      "dateFrom": "2021-12-31T21:00:00.000000Z",
      "description": "Регион к которому относится касса",
      "rule": "GetParameter(concat(¥"http://ip:5000/checkouts/¥", objectId), ¥"region¥")"
    }
  ]
}
```



#### Поля объекта «Тип счёта»:

П а р а м е т р	О п и с а н и е
"accountTypeId"	И д. Т и п а с ч е т а
"accountTypeName"	Н а з в а н и е "Т и п а с ч е т а"
"apType"	П р и з н а к а к т и в н о с т и (а к т и в н ы й, п а с с и в н ы й, а к т и в н о - п а с с и в н ы й)
"extParameterRules"	С е к ц и я д о п о л н и т е л ь н ы х п а р а м е т р о в
"name"	Н а з в а н и е п а р а м е т р а
"dateFrom"	Д а т а н а ч а л а д е й с т в и я "П р а в и л а"
"description"	Ч е л о в е к о ч и т а е м о е о п и с а н и е
"rule"	"П р а в и л о" з а п о л н е н и я з н а ч е н и я п а р а м е т р а п р и с о з д а н и и а н а л и т и ч е с к о г о с ч е т а

#### Для манипулирования Типами Счетов используется следующее API

М е т о д	Endpoint	О п и с а н и е
GET	/api/accounting-system/v1/account-types	П о л у ч е н и е с п и с к а "т и п о в с ч е т о в"
POST	/api/accounting-system/v1/account-types	Р е г и с т р а ц и я "т и п а с ч е т а"
GET	/api/accounting-system/v1/account-types/{accountTypeId}	П о л у ч е н и е "т и п а с ч е т а"
PUT	/api/accounting-system/v1/account-types/{accountTypeId}	О б н о в л е н и е "т и п а с ч е т а"
DELETE	/api/accounting-system/v1/account-types/{accountTypeId}	У д а л е н и е "т и п а с ч е т а"

**Внимание!** Система не позволит удалять тип счета если уже существуют аналитические счета данного типа.

## 3.2. Схема обработки операции

Объект "Схема обработки операции" определяет экономическую суть операции и особенности ее учета, позволяет гибко настраивать обработку операций подлежащих учету.

"Схема обработки операции" состоит из:

- Названия;
- Даты начала действия схемы;
- Параметры схемы обработки операций;
- Правил формирования проводок.

**Название** - однозначно, в пределах системы, определяет экономическую суть операции. (Зачисление, списание, начисление и т.п.)

**Даты начала действия схемы** - определяет с какого момента для указанной операции применяется данная схема.

**Параметры схемы обработки операций** - определяют обязательные параметры операции и их типы. Указанные параметры могут быть использованы в "Правилах формирования проводок".

***Правила формирования проводок*** - определяют расчет параметров "Проводок" (Дата учета, Дата влияния на расчёты, Счет дебета, Счет кредита, Сумма, Описание). (см. 3.5. Проводка и 5.1. Синтаксис "Правил".)

### Пример объекта "Схема обработки операции":

```
{
  "dateFrom": "2010-01-01T00:00:00+03",
  "operationName": "Выдача займа",
  "parameters": [
    {"name": "organization", "type": "String"},
    {"name": "checkout", "type": "Number"},
    {"name": "contract", "type": "Number"},
    {"name": "amount", "type": "Number"},
    {"name": "description", "type": "String"}
  ],
  "rules": [
    {
      "number": 0,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Основной долг¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/checkouts/?number=¥", checkout), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111112,
¥"Касса¥")",
      "amount": "amount",
      "description": "concat(¥"Выдача займа. ¥", description)"
    }
  ]
}
```

### Поля объекта "Схема обработки операции":

П а р а м е т р	О п и с а н и е
"dateFrom"	Дата начала действия схемы
"operationName"	Название "Схемы обработки операции"
"parameters"	Секция описания Параметров схемы операции
"name"	Название параметра
"type"	Тип данных параметра
"rules"	Секция описания Правил формирования проводок
"number"	Порядковый номер правила
"entryDate"	Правило определения даты проводки
"debitAccount"	Правило определения счета дебита
"creditAccount"	Правило определения счета кредита
"amount"	Правило определения суммы проводки
"description"	Правило определения назначения платежа проводки

### Для манипулирования Схемами обработки операций используется Rest API

М е т о д	Endpoint	О п и с а н и е
GET	/api/accounting-system/v1/operations/types	Получение списка "Схем обработки операций"
POST	/api/accounting-system/v1/operations/types	Регистрация "Схемы обработки операции"
GET	/api/accounting-system/v1/operations/types/{operationTypeId}	Получение "Схемы обработки операции"
PUT	/api/accounting-system/v1/operations/types/{operationTypeId}	Обновление "Схемы обработки операций"
DELETE	/api/accounting-system/v1/operations/types/{operationTypeId}	Удаление "Схемы обработки операций"

### 3.3. Операция

Объект "Операция" состоит из названия операции, даты операции, ссылки на документ основание и "списка параметров операции".

Название операции и Дата операции определяют схему, по которой будут рассчитываться параметры проводок.

Ссылка на документ основание — это UUID документа во внешней системе.

Список параметров операции — это набор пар "название" - "значение" определяющие суть операции подлежащей учету.

Если операция была отменена, то заполняется "Ссылка на отменяющую операцию".

Пример объекта "Операция":

```
{
  "documentId": null,
  "operationName": "Выдача займа",
  "creationDate": "2022-11-24T13:17:38.7308410Z",
  "operationDate": "2021-11-22T09:13:01.0000000Z",
  "operationId": "e74eb6ca-367c-4258-9a1a-587f0ea7ea43",
  "stornoOperationId": null,
  "parameters": [
    {"name": "organization", "value": "Займы Всем"},
    {"name": "checkout", "value": "1"},
    {"name": "contract", "value": "12345"},
    {"name": "amount", "value": "1000"},
    {"name": "description", "value": "Test"}
  ]
  "entries": [
    {
      "entryId": "cad8b56e-09a7-49de-a46a-b414cec9734a",
      "operationId": "e74eb6ca-367c-4258-9a1a-587f0ea7ea43",
      "creationDate": "2022-11-24T13:17:39.4606220Z",
      "accountingDate": "2021-11-22T09:13:01.0000000Z",
      "affectingDate": "2021-11-22T09:13:01.0000000Z",
      "debitAccountId": "97b405f7-2294-42f4-acd6-459197825ad2",
      "creditAccountId": "f93603db-cc86-4648-b2c4-c404f826209a",
      "amount": 1000,
      "description": "Выдача займа. Test",
      "stornoEntryId": null
    }
  ]
}
```

Поля объекта "Операция":

П а р а м е т р	О п и с а н и е
"documentId"	Ид. «документа основания»
"operationName"	Название "Схемы обработки операции"
"creationDate"	Дата создания операции
"operationDate"	Дата операции
"operationId"	Ид. Операции
"stornoOperationId"	Ид. сторнирующей операции
"parameters"	Секция описания Параметров операции
"name"	Название параметра
"value"	Значение параметра
"entries"	Секция проводок операции (см. 3.5. Проводка)

Для манипулирования Операциями используется Rest API

М е т о d	Endpoint	О п и с а н и е
-----------	----------	-----------------

GET	/api/accounting-system/v1/operations/	Получение списка "Операций"
POST	/api/accounting-system/v1/operations/	Регистрация "Операции"
GET	/api/accounting-system/v1/operations/{operationId}	Получение "Операции"
PUT	/api/accounting-system/v1/operations/{operationId}	Повторная генерация проводок "Операции"
PUT	/api/accounting-system/v1/operations/storno	Добавление ссылки на сторнирующую операцию
DELETE	/api/accounting-system/v1/operations/{operationId}	Удаление "Операции"

---

**Внимание!** При удалении операции проводки операции так же будут удалены

### 3.4. Аналитический счет

Объект "Аналитический счет" — это идентификатор учетного регистра и его "параметры" (реквизиты и атрибуты)

Обычно создается при регистрации первой проводки, влияющей на средства данного вида.

Каждый аналитический счет связан с "Объектом" (Филиал, Касса, Договор, Склад... и т.п.), "Типом счета" (экономическое содержание хозяйственных средств) и Организацией.

### Пример объекта "Аналитический счет":

```
{
  "accountId": "f93603db-cc86-4648-b2c4-c404f826209a",
  "creationDate": "2022-11-22T14:17:38.5498270Z",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "objectId": "cd42b975-aa3e-4042-82f6-97edd97e1bc4",
  "objectType": "11111111-0000-4a41-b0e9-111111111112",
  "apType": "активный",
  "accountTypeId": "cf613df5-c7ee-47a0-b2bf-1d75eacaed43",
  "accountTypeName": "Касса",
  "extParameters": [
    {
      "name": "EPS",
      "value": "20202810000000000001"
    },
    {
      "name": "officeName",
      "value": "Офис 1"
    },
    {
      "name": "region",
      "value": "Москва"
    }
  ]
}
```

### Поля объекта "Аналитический счет":

Параметр	Описание
"accountId"	Ид. Счета
"creationDate"	Дата создания счета
"organizationId"	Ид. Организации
"objectId"	Ид. Объекта учета
"objectType"	Тип Объекта учета
"apType"	Признак активности (активный, пассивный, активно-пассивный)
"accountTypeId"	Тип счета
"accountTypeName"	Название «типа счета»
"extParameters"	Секция описания Параметров счета
"name"	Название параметра
"value"	Значение параметра

### Для манипулирования Аналитическими счетами используется Rest API

Метод	Endpoint	Описание
GET	/api/accounting-system/v1/accounts/	Получение списка "Аналитических счетов"
POST	/api/accounting-system/v1/accounts/	Регистрация "Аналитических счетов"
GET	/api/accounting-system/v1/accounts/{accountId}	Получение "Аналитического счета"
PUT	/api/accounting-system/v1/accounts/{accountId}	Обновление "Аналитического счета"
DELETE	/api/accounting-system/v1/accounts/{accountId}	Удаление "Аналитического счета"

**Внимание!** Система не позволит удалять счета по которым существуют проводки

### **3.5. Проводка**

Объект "Проводка" фиксирует движение хозяйственных средств между "Аналитическими счетами" и реализует принцип двойной бухгалтерской записи.

Содержит ссылки на счет дебета, счет кредита, операцию, а также даты учета, влияния на расчеты и описание экономической сути (назначение платежа).

### Пример объекта "Проводка":

```
{
  "entryId": "cad8b56e-09a7-49de-a46a-b414cec9734a",
  "operationId": "e74eb6ca-367c-4258-9a1a-587f0ea7ea43",
  "creationDate": "2022-11-24T13:17:39.4606220Z",
  "accountingDate": "2021-11-22T09:13:01.0000000Z",
  "affectingDate": "2021-11-22T09:13:01.0000000Z",
  "debitAccountId": "97b405f7-2294-42f4-acd6-459197825ad2",
  "creditAccountId": "f93603db-cc86-4648-b2c4-c404f826209a",
  "amount": 1000,
  "description": "Выдача займа. Test",
  "stornoEntryId": null
}
```

### Поля объекта "Проводка":

П а р а м е т р	О п и с а н и е
"entryId"	И д. П р о в о д к и
"operationId"	И д. О п е р а ц и и
"creationDate"	Д а т а с о з д а н и я
"accountingDate"	Д а т а п р и н я т и я к у ч е т у
"affectingDate"	Д а т а в л и я н и я н а р а с ч е т ы
"debitAccountId"	И д. с ч е т а д е б е т а
"creditAccountId"	И д. с ч е т а к р е д и т а
"amount"	С у м м а
"description"	Н а з н а ч е н и е п л а т е ж а
"stornoEntryId"	И д. О т м е н я ю щ е й п р о в о д к и

### Для манипулирования Проводками счетами используется Rest API

М е т о д	Endpoint	О п и с а н и е
GET	/api/accounting-system/v1/entries/	П о л у ч е н и е с п и с к а "П р о в о д о к"
POST	/api/accounting-system/v1/entries/	Р е г и с т р а ц и я "П р о в о д к и"
GET	/api/accounting-system/v1/entries/{entryId}	П о л у ч е н и е "П р о в о д к и"
PUT	/api/accounting-system/v1/entries/storno	Д о б а в л е н и е с с ы л к и н а с т о р н и р у ю щ ю "П р о в о д к у"
DELETE	/api/accounting-system/v1/entries/{entryId}	У д а л е н и е "П р о в о д к и"

## 3.6. План счетов

Объект «План счетов» - регистрирует иерархически организованную систему счетов.

### Пример объекта «План счетов»:

```
{
  "chartOfAccountsId": "28a03813-4676-466b-986a-cd261ec8f4ea",
  "name": "Основной план счетов",
  "description": "Принят УП от 01.10.2022"
}
```

### Поля объекта "План счетов":

П а р а м е т р	О п и с а н и е
"chartOfAccountsId"	И д. п л а н а с ч е т о в
"name"	Н а з в а н и е
"description"	Ч е л о в е к о ч и т а е м о е о п и с а н и е

Для манипулирования Планами счетов используется Rest API

Метод	Endpoint	Описание
GET	/api/accounting-system/v1/charts-of-accounts	Получение списка "Планов счетов"
POST	/api/accounting-system/v1/charts-of-accounts	Регистрация "Плана счетов"
PUT	/api/accounting-system/v1/charts-of-accounts/{chartOfAccountsId}	Обновление "Плана счетов"
GET	/api/accounting-system/v1/charts-of-accounts/{chartOfAccountsId}	Получение "Плана счетов"
DELETE	/api/accounting-system/v1/charts-of-accounts/{chartOfAccountsId}	Удаление "Плана счетов"

### 3.7. Синтетический счет

Объект "Синтетический счет" реализует абстракцию синтетического учета и позволяет группировать однородные по экономической сущности счета.

Счет, у которого указан параметр "parentAccountId" является субсчётом указанного счета.

Все «аналитические счета» того же типа, что и «синтетический счет» и его субсчёта образуют его остаток и оборот

Пример объекта "Синтетический счет":

```
{
  "accountId": "71259d43-b04c-4d73-a4bb-652399f901fa",
  "accountTypeId": "cf613df5-c7ee-47a0-b2bf-1d75eacaed43",
  "apType": "активный",
  "chartOfAccountsId": "28a03813-4676-466b-986a-cd261ec8f4ea",
  "parentAccountId": null,
  "description": "К а с с а"
}
```

Поля объекта "Синтетический счет":

П а р а м е т р	О п и с а н и е
"accountId"	И д. С ч е т а
"accountTypeId"	И д. т и п а с ч е т а
"apType"	П р и з н а к а к т и в н о с т и
"chartOfAccountsId"	И д. П л а н а с ч е т о в
"parentAccountId"	И д. Р о д и т е л ь с к о г о с ч е т а
"description"	Ч е л о в е к о ч и т а е м о е о п и с а н и е

Для манипулирования Синтетическими счетами используется Rest API

М е т о д	Endpoint	О п и с а н и е
GET	/api/accounting-system/v1/charts-of-accounts/{chartOfAccountsId}/accounts/	П о л у ч е н и е с п и с к а "С и н т е т и ч е с к и х с ч е т о в"
POST	/api/accounting-system/v1/charts-of-accounts/{chartOfAccountsId}/accounts/	Р е г и с т р а ц и я "С и н т е т и ч е с к о г о с ч е т а"
GET	/api/accounting-system/v1/charts-of-accounts/{chartOfAccountsId}/accounts/{accountId}	П о л у ч е н и е "С и н т е т и ч е с к о г о с ч е т а"
PUT	/api/accounting-system/v1/charts-of-accounts/{chartOfAccountsId}/accounts/{accountId}	О б н о в л е н и е "С и н т е т и ч е с к о г о с ч е т а"
DELETE	/api/accounting-system/v1/charts-of-accounts/{chartOfAccountsId}/accounts/{accountId}	У д а л е н и е "С и н т е т и ч е с к о г о с ч е т а"

### 3.8. Закрытый период

Объект "Закрытый период" позволяет блокировать изменения до "Закрытой даты". Служит для обеспечения неизменности отчетности.

Пример объекта "Закрытый период":

```
{
  "closedPeriodId": "f13bf295-23f1-4dcf-ba93-66775a5ece48",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "date": "2022-12-07T21:00:00.000000Z",
  "description": "З а к р ы т и е  о п е р а ц и о н н о г о  д н я"
}
```

Поля объекта "Закрытый период":

П а р а м е т р	О п и с а н и е
"closedPeriodId"	И д. З а к р ы т о г о  п е р и о д а
"organizationId"	И д. О р г а н и з а ц и и
"date"	Д а т а  п о  к о т о р у ю  з а п р е щ е н а  р е г и с т р а ц и я  п р о в о д о к
"description"	Ч е л о в е к о ч и т а е м о е  о п и с а н и е

Для манипулирования Закрытыми периодами используется Rest API

М е т о д	Endpoint	О п и с а н и е
GET	/api/accounting-system/v1/{organizationId}/closed-periods/	П о л у ч е н и е  с п и с к а  "З а к р ы т ы х  п е р и о д о в"
POST	/api/accounting-system/v1/{organizationId}/closed-periods/	Р е г и с т р а ц и я  "З а к р ы т о г о  п е р и о д а"
GET	/api/accounting-system/v1/{organizationId}/closed-periods/{closedPeriodId}	П о л у ч е н и е  "З а к р ы т о г о  п е р и о д а"
PUT	/api/accounting-system/v1/{organizationId}/closed-periods/{closedPeriodId}	О б н о в л е н и е  о п и с а н и я  "З а к р ы т о г о  п е р и о д а"
DELETE	/api/accounting-system/v1/{organizationId}/closed-periods/{closedPeriodId}	У д а л е н и е  "З а к р ы т о г о  п е р и о д а"

# Руководство администратора

## 4. Установка ПО

### 4.1. Требования

Для установки и работы необходим компьютер с операционной системой семейства Linux с предустановленными Docker Engine и плагином Docker Compose. Рекомендованные версии пакетов.

Docker Engine 2.10.0 и выше

Docker Compose 2.10.0 и выше

Также в случае, если планируется развертывание с профилем "full", включающим в себя контейнеры БД Postgres и RabbitMQ, требуется доступ в сеть Интернет.

Пользователь, от имени которого выполняются операции по развертыванию, должен состоять в группе docker или обладать правами sudo.

**ВАЖНО.** Пакет docker.io входящий в состав некоторых дистрибутивов ОС может вызывать неполадки при развертывании и работе модуля расчетов. Работа модуля расчетов в нем не гарантируется.

Рекомендуется удалить стандартный пакет и переустановить Docker в соответствии с руководством с сайта разработчика системы Docker.

<https://docs.docker.com/compose/install/>

### 4.2. Установка программы

#### Подготовка и запуск

Распакуйте полученный архив с дистрибутивом, например, в папку ~/version.

Настройте значения переменных в файле setup.sh. Описание переменных можно посмотреть в конце руководства в разделе «Описание переменных в файле setup.sh», либо в комментариях в файле setup.sh.

После завершения настройки переменных присвойте файлу setup.sh права на выполнение и запустите его.

```
chmod +x setup.sh
sudo ./setup.sh
```

После запуска будут сгенерированы конфигурационные файлы с параметрами, указанными в переменных файла setup.sh и запущены контейнеры сервисов.

### 4.3. Мониторинг и проверка работоспособности

После завершения работы файлы setup.sh, выполните команду

```
docker compose ls
```

```
root@DBB:/home/shota/version# docker compose ls
NAME                STATUS              CONFIG FILES
version            running(8)          /home/shota/version/docker-compose.yml
```

Если в выводе в столбце STATUS отображаются 8 запущенных сервисов (6, если при настройке использовался профиль установки service), то развертывание и запуск прошли успешно.

**ВАЖНО.** В некоторых конфигурациях не все сервисы запускаются при первом развертывании и значение running равно 3 или 1. Для исправления достаточно повторно запустить файл setup.sh

Работоспособность сервисов можно проверить при помощи веб-браузера.

*http://<IP-сервера>:8080* – UI модуля расчетов

*http://<IP-сервера>:15672* – RabbitMQ

## 4.4. Остановка и удаление

Командой

```
docker compose --project-name <имя проекта> down
```

можно полностью остановить проект и удалить все созданные в нём контейнеры и сети.

Имя проекта можно выяснить в столбце NAME при выполнении команды

```
docker compose ls
```

Для полного удаления файлов, сгенерированных проектом, необходимо после остановки проекта выполнить команду

```
docker system prune -a
```

## 4.5. Изменение конфигурации

Для перенастройки необходимо полностью остановить проект и удалить созданные им образы. Затем внести необходимые изменения в переменные и снова запустить файл setup.sh.

### Описание переменных в файле setup.sh

- **INSTALLTYPE** - вид установки

Возможные значения full или service. По умолчанию значение full.

**full** дополнительно устанавливает в контейнере сервисы Postgres и RabbitMQ. Требуется подключение к сети Интернет для скачивания соответствующих образов.

**service** устанавливает только сервисы модуля расчетов и используется в случае, если планируется использовать уже развернутые сервисы Postgres и RabbitMQ.

- **[POSTGRES\_SERVER\_VALUE]=postgre-db**

Адрес сервиса БД Postgres. По умолчанию контейнер postgre-db

- **[POSTGRES\_PORT\_VALUE]=5432**

Порт сервиса БД Postgres. По умолчанию 5432

- **[POSTGRES\_USER\_VALUE]=postgres\_user**

Имя пользователя для подключения к БД Postgres. По умолчанию postgres\_user

- **[POSTGRES\_PASS\_VALUE]=postgres\_password**

Пароль для подключения к БД Postgres. По умолчанию postgres\_password

- **[RABBITMQ\_DEFAULT\_SERVER\_VALUE]=rabbitmq-manager**

Сервер RabbitMQ. По умолчанию контейнер rabbitmq-manager

- **[RABBITMQ\_DEFAULT\_USER\_VALUE]=guest**

Пользователь RabbitMQ. По умолчанию guest

- **[RABBITMQ\_DEFAULT\_PASS\_VALUE]=guest**

Пароль RabbitMQ. По умолчанию guest

# Руководство пользователя

## 5. Настройка

ПО универсально и позволяет в зависимости от конфигурации вести учет любой деятельности организации с соблюдением принципов бухгалтерского учета.

Настройки могут производиться через:

- Web-интерфейс
- Swagger
- REST API

Web-интерфейс доступен по адресу

*<http://{ip сервера}:8080/>*

Swagger доступен по адресу

*<http://{ip сервера}:55002/swagger/index.html>*

Базовый адрес REST API

*<http://{ip сервера}:55002/>*

### 5.1. Синтаксис "Правил".

"Правило" — это выражение, вычисляемое при необходимости и определяющие алгоритм расчета значения параметров счетов и проводок.

Фактически это вычисляемая формула. Приоритет определяется по правилам математики.

Функции, переменные и константы однородные по типам могут объединяться операторами.

Названия функций регистронезависимые.

#### 5.1.1. Типы данных

В "Правилах" можно использовать следующие типы:

- Логические (TRUE|FALSE)
- Числовые (10, 0.1 -11.1)
- Даты (2022-10-01T01:11:23.234+0300 ISO 8601)
- GUID (123e4567-e89b-12d3-a456-426655440000 ISO 15745)
- Строковые ("Булочка")

Число - десятичное

. - разделитель целой и дробной части.

## 5.1.2. Операторы

### Числовые операторы:

- + Сложение
- Вычитание
- \* Умножение
- / Деление
- ^ Возведение в степень

### Логические операторы:

- || Логическое ИЛИ
- && Логическое И
- ~ Логическое ОТРИЦАНИЕ

### Операторы сравнения:

- > Больше
- >= Больше или равно
- < Меньше
- <= Меньше или равно
- = Равно
- <> Не равно

## 5.1.3. Числовые функции:

**Min(numeric, ...)** - Минимальное значение из переданных

Пример:

```
Min(11, 2, 3, 14.1, 1) => 1
```

**Max(numeric, ...)** - Максимальное значение из переданных

Пример:

```
Max(11, 2, 3, 14.1, 1) => 14.1
```

**Abs(numeric)** - Абсолютное значение

Пример:

```
Abs(-11.13) => 11.13
```

**IeeeRemainder(numeric, numeric)** - Остаток от деления

Пример:

```
IeeeRemainder(11, 3) => 3
```

**Round(numeric(numeric)? )** - Округление числа к ближайшему с точность указываемой вторым параметром. Если второй параметр не указан, он принимается равным 0.

Пример:

```
Round(123.123, 2) => 123.12
```

**Ceil(numeric,numeric)? )** - Округление числа в верх с заданной точностью. Если второй параметр не указан, он принимается равным 0.

Пример:

```
Ceil(123.123, 2) => 123.13
```

**Floor(numeric,numeric)? )** - Округление числа вниз

Пример:

```
Floor(123.123, 2) => 123.12
```

**ToNumber(numeric), ToNumber(string)** - Явное указание что переменная типа Numeric или Преобразование строки в число

**IIF(logical, any, any)** - Выбор первого или второго выражения в зависимости от результата логического выражения

Пример:

```
IIF(0=1, 12, 24) => 24
```

#### 5.1.4. Функции обработки дат:

**GetDate()** - Текущая дата

Пример:

```
GetDate() => 2022-10-04T13:34:19.8478580Z
```

**ToDate(date), ToDate(string)** - Преобразование строки в дату

Пример:

```
toDate(Concat("2020-12-05", "T", "00:00:00", ".000", "Z")) => 2020-12-05T00:00:00.0000000Z  
toDate("2020-12-05") => 2020-12-05T00:00:00.0000000Z
```

**Trunc(string, date)** - Округление даты

Пример:

```
trunc("Year", "2020-12-05T00:00:00.000Z") => "2020-01-01T00:00:00.000Z"
```

**DateAdd(date, string, numeric)** - Увеличение даты

Пример:

```
dateAdd(toDate("2020-12-05T00:00:00.000Z"), "YEAR", 1) => "2021-12-05T00:00:00.000Z"
```

**GetDatePart(string, date)** - Получение значения части даты

Пример:

```
GetDatePart("Year", toDate("2020-12-05T00:00:00.000Z")) => 2020
```

**GetDateInterval(date, date)** - Получение интервала между датами в днях

Пример:

```
GetDateInterval(toDate("2020-12-05T00:00:00.000Z"), toDate("2020-12-07T00:00:00.000Z")) => -2
```

### 5.1.5. UUID функции:

**ToGUID(guid), ToGUID(string)** - Явное указание, что переменная типа guid или Конвертация строки в guid

### 5.1.6. Строковые функции:

**Concat(string, ...)** - Объединение строк

Пример:

```
Concat("A", "B", "A") => "ABA"
```

**ToLower(string)** - Преобразование всех символов в строчные

Пример:

```
ToLower("AbbA") => "abba"
```

**ToUpper(string)** - Преобразование всех символов в прописные

Пример:

```
ToUpper("AbbA") => "ABBA"
```

**Trim(string), Trim(string, string)** - Удаление символов в конце и начале строки (Текст, Удаляемый символ), если не указан второй параметр, то удаляем пробельные символы

Пример:

```
Trim("AbbA", "A") => "bb"
```

**Translate(string, string, string)** - Замена в символов в строке (Текст, Заменяемые символы, Замещающие символы)

Пример:

```
Translate("AbbA", "Ab", "Ba") => "BaaB"
```

**SubString(string, numeric), SubString(string, numeric, numeric)** - Получить часть строки (Текст, Начальный индекс, Кол. символов), если не указан третий параметр, то до конца строки.

Пример:

```
SubString("AbbA", 2, 2) => "bb"
```

**Remove(string, numeric, numeric)** - Удаление части строки (Текст, Начальный индекс, Количество удаляемых символов)

Пример:

```
Remove("AbbA", 2, 2) => "AA"
```

**Replace(string, string, string)** - Замена в тексте (Текст, Искомая подстрока, Замещающая строка)

Пример:

```
Replace("AbbA", "b", "CC") => "ACCCCA"
```

**RegexMatch(string, string)** - Поиск в тексте согласно регулярному выражению

Пример:

```
RegexMatch("AbbA", "[a-z]+") => "bb"
```

**RegexReplace(string, string, string)** - Замена в тексте согласно регулярному выражению

Пример:

```
RegexReplace("AbbA", "[a-z]+", "d") => "AdA"
```

**Format(numeric, string)** - Форматирование числа

Пример:

```
Format(121.452, "0.00") => 121.45
```

**Format(date, string)** - Форматирование даты

**h, hh** — Часы в 12 часовом формате

**H, HH** — Часы в 24 часовом формате

**mm** - Минуты

**ss** - Секунды

**f, ff, ffff** - Доли секунды

**zzz** – Таймзона

**t** — до P|A до полудня/после полудня

**d, dd, D, DD** – День месяца

**MM** – Месяц

**yy, yyy, YY, YYYY** - Год

Пример:

```
Format("2022-10-01T12:23:43.234Z", "yyyy-MM-dd") => "2022-10-01"
```

## 5.1.7. Функции работы со счетами:

**GetBalance(guid, date)** - Получение остатка счета на дату

Пример:

```
GetBalance(toGuid("3fa85f64-5717-4562-b3fc-2c963f66afa6"), toDate("2020-12-04T11:48:17.111Z")) => number
```

**GetBalance(guid)** - Получение текущего остатка счета

Пример:

```
GetBalance(toGuid("3fa85f64-5717-4562-b3fc-2c963f66afa6")) => number
```

**GetBalanceByAffectingDate(guid, date)** - Получение остатка счета на дату (по дате влияния на расчет)

Пример:

```
GetBalance(toGuid("3fa85f64-5717-4562-b3fc-2c963f66afa6"), toDate("2020-12-04T11:48:17.111Z")) => number
```

**GetBalanceByAffectingDate(guid)** - Получение текущего остатка счета (по дате влияния на расчет)

Пример:

```
GetBalance(toGuid("3fa85f64-5717-4562-b3fc-2c963f66afa6")) => number
```

**GetDebit(guid, date, date)** - Получение оборота по дебету счета

Пример:

```
GetDebit(toGuid("3fa85f64-5717-4562-b3fc-2c963f66afa6"), toDate("2020-12-04T00:00:00.000Z"), toDate("2020-12-05T00:00:00.000Z")) => number
```

**GetDebit(guid, guid, date, date)** - Получение оборота по дебету счета в корреспонденции со счетом2

Пример:

```
GetDebit(toGuid("3fa85f64-5717-4562-b3fc-2c963f66afa6"), toGuid("527b757c-aaf3-4d51-ada1-3992686e2af1"), toDate("2020-12-04T00:00:00.000Z"), toDate("2020-12-05T00:00:00.000Z")) => number
```

**GetCredit(guid, date, date)** - Получение оборота по кредиту счета

Пример:

```
GetCredit(toGuid("3fa85f64-5717-4562-b3fc-2c963f66afa6"), toDate("2020-12-04T00:00:00.000Z"), toDate("2020-12-05T00:00:00.000Z")) => number
```

**GetCredit(guid, guid, date, date)** - Получение оборота по кредиту счета в корреспонденции со счетом2

Пример:

```
GetCredit(toGuid("3fa85f64-5717-4562-b3fc-2c963f66afa6"), toGuid("527b757c-aaf3-4d51-ada1-3992686e2af1"), toDate("2020-12-04T00:00:00.000Z"), toDate("2020-12-05T00:00:00.000Z")) => number
```

**GetAccount ( guid , guid , guid , string )** - Получения guid счета. Если счет с заданными параметрами в системе отсутствует, произойдет его регистрация.

Пример:

```
getAccount(organizationId, objectId, objectType, accountTypeName) => guid
```

## 5.1.8. Функции для интеграции с другими системами:

**GetParameter(string, string)** - Обращение к внешнему URL для получения значения параметра

Допустим, что Http запрос GET `http://a.com/api?d=12` возвращает `{"test": 123, "test2" 321}` тогда

```
GetParameter("http://a.com/api?d=12", "test") => 123
```

## 5.2. Настройка Типов Счетов

Для настройки "Типов Счетов" через Web-интерфейс зайдите в разделе "Типы Счетов"

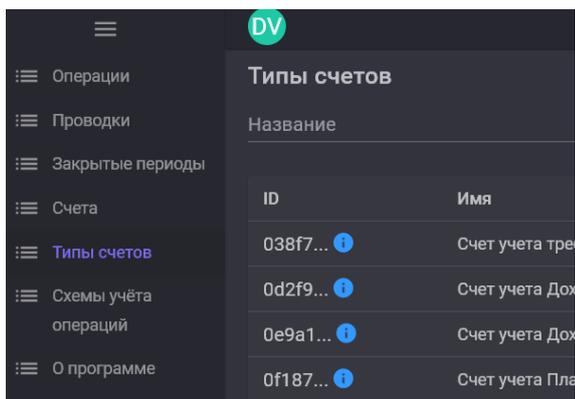


Рисунок 2: Раздел "Типы Счетов"

При переходе в раздел вы видите все типы счетов, настроенных в системе.

Можно применить фильтр по названию или части названия "Типа счета". Для этого введите в поле "Название" и нажмите кнопку "Применить фильтр"

Записи таблицы можно сортировать нажатием на заголовок поля.

### 5.2.1 Создание "Типа Счета"

Для добавления нового "Типа Счета" нажмите на кнопку  - "Добавить тип счета" в заголовке таблицы.

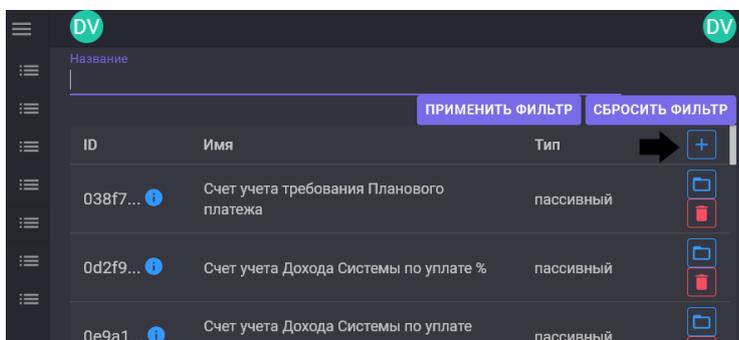


Рисунок 3: кнопка "Добавить тип

счета"

В открывшемся диалоге "Добавление типа счета" заполните название и тип "активности"

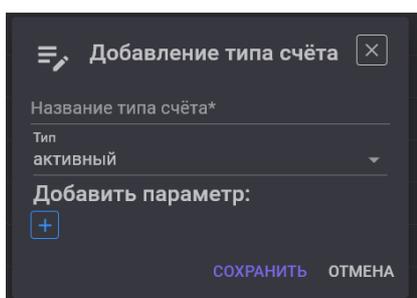


Рисунок 4: диалог "Добавление типа счета"

При необходимости добавляем описание дополнительных параметров.

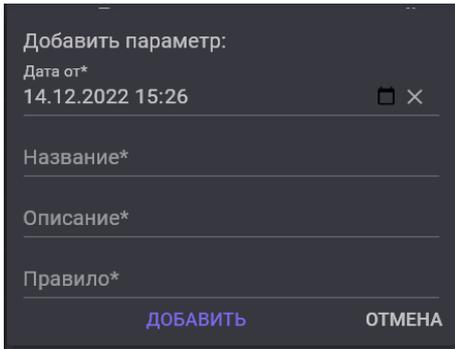


Рисунок 5: диалог "Добавление параметра на типа счета"

В поле правило добавляем "выражение" для расчета значения параметра счета. Это может быть запрос во внешнюю систему для получения характеристик объекта, по которому ведется учет. См. 5.1. Синтаксис "Правил".

По завершению заполнения, сохраняем результат нажатием на кнопку "Сохранить".

### Пример API запроса:

POST /api/accounting-system/v1/account-types

```
{
  "accountTypeName": "К а с с а",
  "apType": "а к т и в н ы й",
  "extParameterRules": [
    {
      "name": "EPS",
      "dateFrom": "2021-12-31T21:00:00.000000Z",
      "description": "Номер счета согласно ЕПС",
      "rule": "concat(¥"202028100¥", Format(toNumber(GetParameter(concat(¥"http://ip:5000/checkouts/¥", objectId), ¥"number¥")), ¥"00000000000¥"))"
    },
    {
      "name": "officeName",
      "dateFrom": "2021-12-31T21:00:00.000000Z",
      "description": "О ф и с к которому относится касса",
      "rule": "GetParameter(concat(¥"http://ip:5000/checkouts/¥", objectId), ¥"officeName¥")"
    },
    {
      "name": "region",
      "dateFrom": "2021-12-31T21:00:00.000000Z",
      "description": "Р е г и о н к которому относится касса",
      "rule": "GetParameter(concat(¥"http://ip:5000/checkouts/¥", objectId), ¥"region¥")"
    }
  ]
}
```

## 5.2.2 Изменение "Типа Счета"

Для изменения "типа счета" найдите интересующий тип счета в списке и нажмите кнопку

 - "просмотра/редактирования счета"

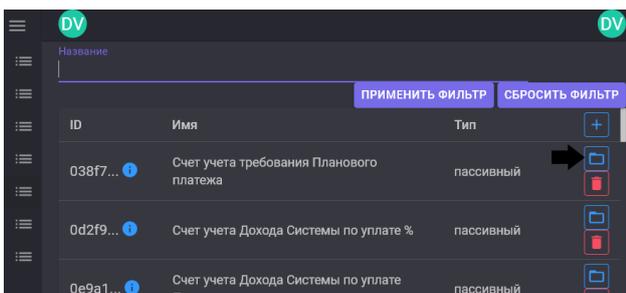


Рисунок 6: кнопка "просмотра/редактирования счета"

В открывшемся диалоге "Редактирование типа счета" вы можете изменять атрибуты и описания дополнительных параметров.

Рисунок 7: диалог "Редактирование типа счета"

Вы можете изменить описание дополнительных параметров входя в соответствующие поля.

Вы можете добавить дополнительные параметры нажатием на кнопку  - "Добавление параметра типа счета"

Вы можете удалить дополнительные параметры нажатием на кнопку  - "Удаление параметра типа счета"

По завершению редактирования сохраняем результат нажатием на кнопку "Сохранить"

### 5.2.3 Удаление "Типа Счета"

Для удаления "типа счета" в списке типов счетов, в строчке с интересующим типом счета нажмите кнопку  - "Удалить тип счета"

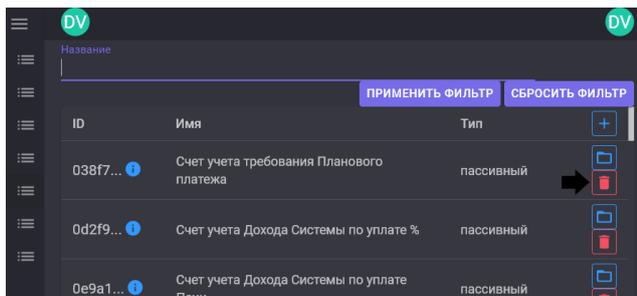


Рисунок 8: кнопка "Удалить типа счета"

**Внимание!!!** Нельзя удалять тип счета пока в системе зарегистрированы счета этого типа.

Пример API запроса:

```
DELETE /api/accounting-system/v1/account-types/{account-types-id}
```

### 5.3. Настройка Схем обработки операций

Для настройки "Схем обработки операций" через Web-интерфейс зайдите в разделе "Схемы обработки операций"

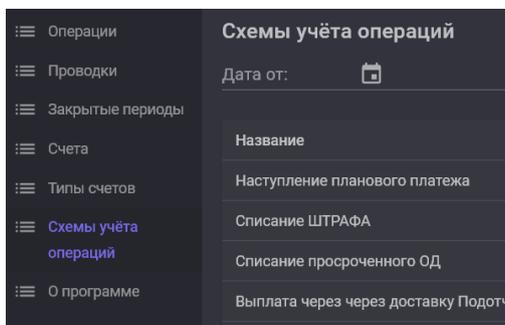


Рисунок 9: Раздел "Схемы обработки операций"

При переходе в раздел вы видите все схемы обработки операций, настроенные в системе.

Можно применить фильтр по названию "Схемы операции", дате начала действия схемы. Для этого заполните поля "Название", "Дата от", "Дата до" введите в поле и нажать кнопку "Применить фильтр"

Если заполнено поле "На дату», то выведутся только схемы, действующие на указанную дату.

Записи таблицы можно сортировать нажатием на заголовок поля.

### 5.3.1 Создание "Схемы обработки операции"

Создавать схему можно при помощи кнопки  - "Добавления схемы операции" и  - "Копирования схемы операции".

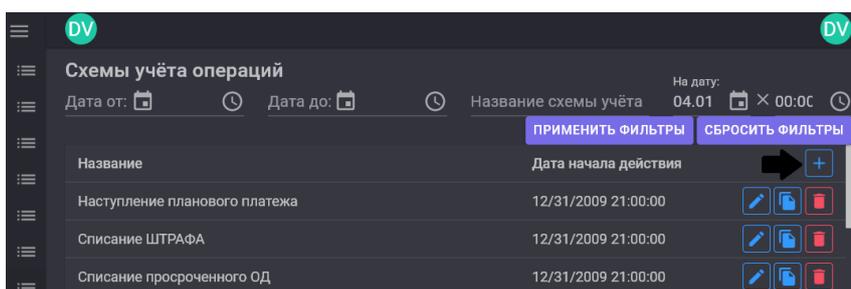


Рисунок 10: кнопка

"Добавления схемы операции"

При нажатии на кнопку "Добавления схемы операции" откроется диалог "Создать схему учёта операций"

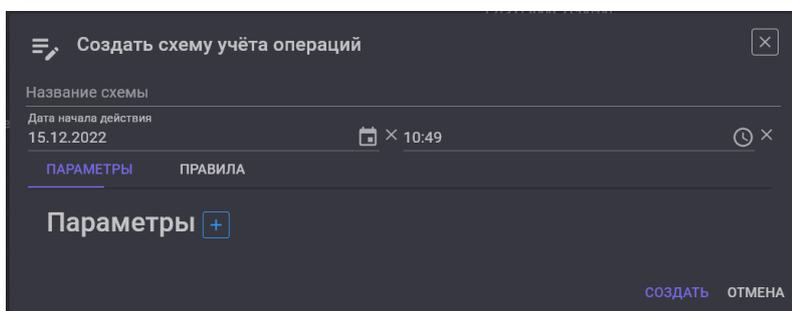


Рисунок 11: диалог "Создать схему

учёта операций"

В открывшемся диалоге "Создать схему учёта операций" заполняем название и дату начала действия.

На закладке "Параметры" добавляем обязательные параметры операции кнопкой  - "Добавить параметр"

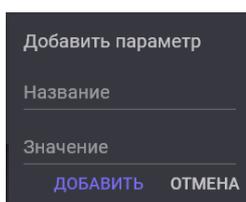


Рисунок 12: диалог "Добавить параметр"

На закладке "Правила" добавляем правила формирования параметров операции кнопкой

 - "Добавить правило"

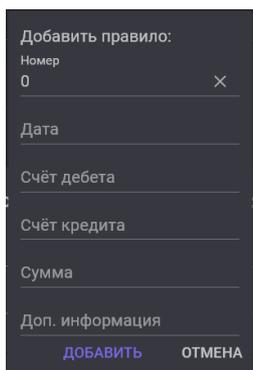


Рисунок 13: диалог "Добавить параметр"

В поля "Дата", "Счёт дебета", "Счёт кредита", "Сумма" и "Доп. Информация", добавляем "выражение" для расчета значения атрибута проводки. Это могут быть в т.ч. функции получения ид. "аналитических счетов", их балансов или оборотов, а также запрос во внешнюю систему для получения характеристик или состояний объектов См. 5.1. Синтаксис "Правил".

Поле "номер" влияет на порядок расчета правил.

В выражении можно использовать названия параметров, указанных в этой схеме операции и специальные переменные.

**operationName** - название операции

**operationDate** - дата операции

**RuleAmount{n}** - значение суммы для вычисления правил с номером {n} (Для правила №3 будут доступны *RuleAmount0*, *RuleAmount1*, *RuleAmount2* если в схеме были определены правила №№ 0, 1, 2, 3, 4... т.е. Уже вычисленные к моменту расчета правила.)

Для правил, у которых рассчитанное значение "Сумма"  $\neq 0$  будет сформирована проводка.

По завершению заполнения сохраняем результат нажатием на кнопку "Сохранить".

При нажатии на кнопку "Копирования схемы операции", откроется диалог "Копировать схему учёта операций"

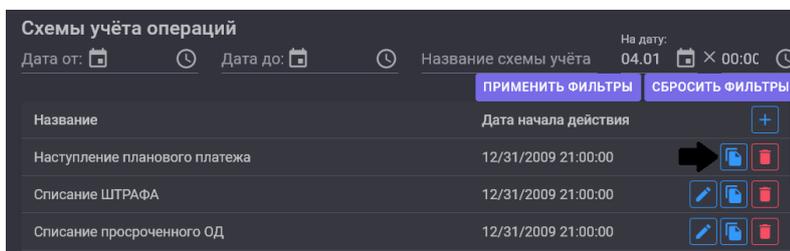


Рисунок 14: кнопка "Копирования

схемы операции"

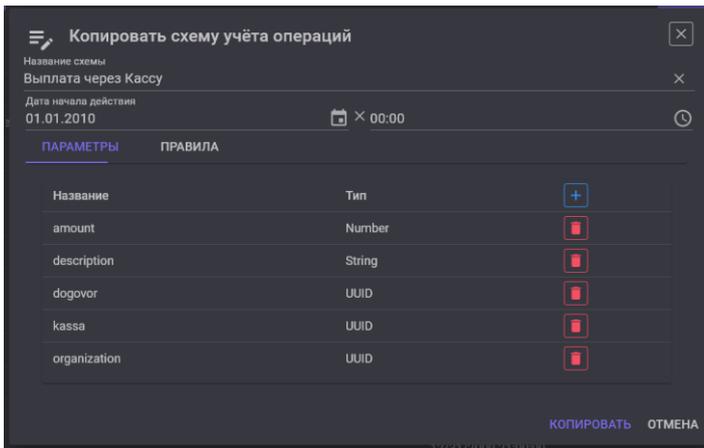


Рисунок 15: диалог "Копировать схему

учёта операций"

Атрибуты, параметры и правила "Схемы обработки операции" будут заполнены на основе выбранной схемы. Внесите необходимые изменения и нажмите кнопку "Копировать"

### Пример API запроса:

POST /api/accounting-system/v1/operations/types/

```
{
  "dateFrom": "2010-01-01T00:00:00+03",
  "operationName": "Выдача займа",
  "parameters": [
    {"name": "organization", "type": "String"},
    {"name": "checkout", "type": "Number"},
    {"name": "contract", "type": "Number"},
    {"name": "amount", "type": "Number"},
    {"name": "description", "type": "String"}
  ],
  "rules": [
    {
      "number": 0,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥http://d-00224:5000/organizations/?name=¥, organization), ¥id¥)),
toGUID(GetParameter(concat(¥http://d-00224:5000/contracts/?number=¥, contract), ¥id¥)), 1111111-0000-4a41-b0e9-111111111111,
¥Основной долг¥)",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥http://d-00224:5000/organizations/?name=¥, organization), ¥id¥)),
toGUID(GetParameter(concat(¥http://d-00224:5000/checkouts/?number=¥, checkout), ¥id¥)), 1111111-0000-4a41-b0e9-111111111112,
¥Касса¥)",
      "amount": "amount",
      "description": "concat(¥Выдача займа. ¥, description)"
    }
  ]
}
```

## 5.3.4 Изменение "Схемы обработки операции"

Для изменения "Схемы обработки операции" найдите интересующую схему в списке и нажмите кнопку  - "Редактирование схемы"

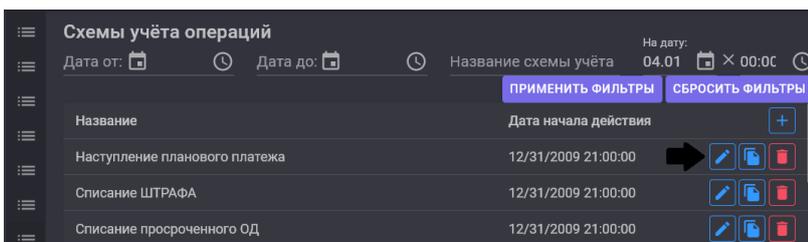


Рисунок 16: кнопка

"Редактирование схемы"

В открывшемся диалоге "Схемы обработки операции" вы можете изменять атрибуты, параметры и правила.

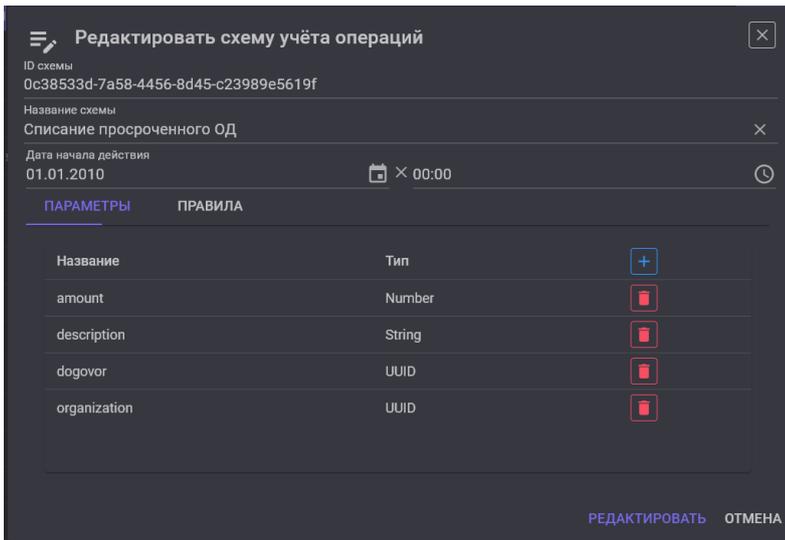


Рисунок 17: диалог

### "Редактирование Схемы обработки операции"

Вы можете изменить описание параметров входя в соответствующие поля.

Вы можете добавить параметры нажатием на кнопку  - "Добавление параметра операции"

Вы можете удалить параметры нажатием на кнопку  - "Удаление параметра операции"

На закладке Правила вы можете изменять, удалять и добавлять правила. (См 5.1. Синтаксис "Правил". и 5.3.1 Создание "Схемы обработки операции")

По завершению редактирования сохраняем результат нажатием на кнопку "Сохранить"

### 5.3.7 Удаление "Схемы обработки операции"

Для удаления "Схемы обработки операции" в списке типов счетов в строчке с интересующим типом счета нажмите кнопку  - "Удалить схему"

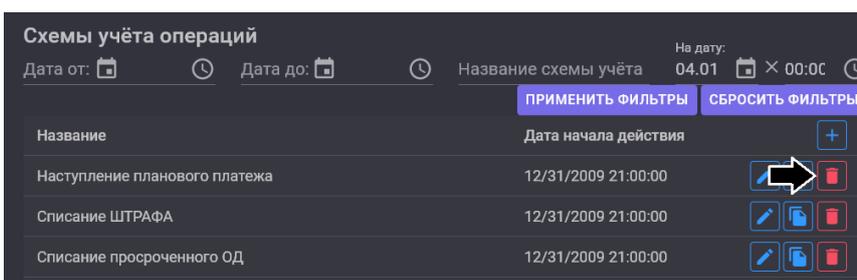


Рисунок 18: кнопка "Удалить

схему"

#### Пример API запроса:

```
DELETE /api/accounting-system/v1/operation/types/{operation-types-id}
```

## 6. Эксплуатация

### 6.1. Операции

#### 6.1.1. Список Операций и их поиск

Для просмотра списка операций перейдите в раздел "Операции"

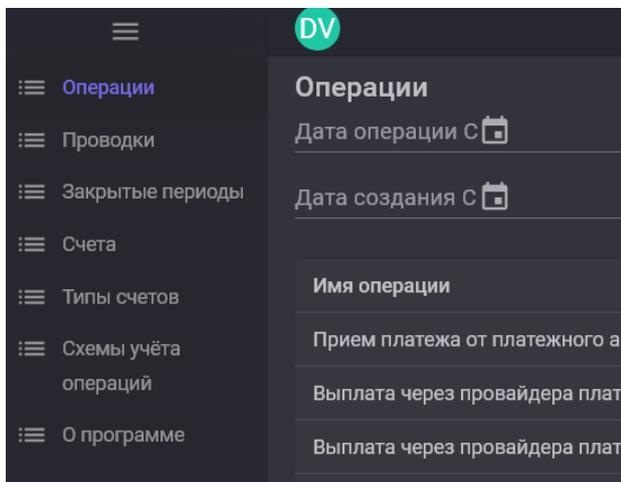


Рисунок 19: Раздел "Операции"

При переходе в раздел вы видите первую страницу списка операций.

Имя операции	Дата операции	Дата создания	ID документа
Прием платежа от платежного агента	11/30/2022 11:50:25	11/30/2022 11:49:07	85971...
Выплата через провайдера платежей	11/17/2022 06:38:18	11/18/2022 15:22:04	32a83...
Выплата через провайдера платежей	11/17/2022 05:45:04	11/18/2022 15:22:04	d1bd0...
Выплата через провайдера платежей	11/16/2022 21:00:00	11/18/2022 15:22:04	2e5d9...
Выплата через провайдера платежей	11/17/2022 12:48:09	11/18/2022 15:22:04	873d2...
Выплата через провайдера платежей	11/17/2022 12:10:13	11/18/2022 15:22:04	c896b...

Рисунок 20: Раздел "Операции"

В разделе Операции вы можете осуществлять поиск операций по "Дате операции" заполняя поля "Дата операции с" и "Дата операции по".

Для поиска операций по "Дате создания операции" заполните поля "Дата создания с" и "Дата создания по"

Так же можно искать операции по "Названию операции", "ID Документа" и "ID Операции"

Метод: GET /api/accounting-system/v1/operations

### Описание параметров метода

П а р а м е т р	О п и с а н и е
DocumentId	Идентификатор документа основания
StornoOperationId	Идентификатор сторнирующей операции
OperationName	Название типа операции
OperationDateStart	Выбрать операции с датой операции больше...
OperationDateEnd	Выбрать операции с датой операции меньше...
OperationId	Выбрать операции Ид Операции
CreateDateStart	Выбрать операции с датой создания больше...
CreateDateEnd	Выбрать операции с датой создания больше...
ParameterName	Название параметра
ParameterValue	Значение параметра
page-size	Количество операций в выборке
page-token	Токен следующего пакета выборки

### Пример запроса:

```
GET /api/accounting-system/v1/operations?operationDateStart=2021-08-11%2009%3A56%3A52%2B03&operationDateEnd=2022-08-11%2009%3A56%3A52%2B03
```

### Пример ответа:

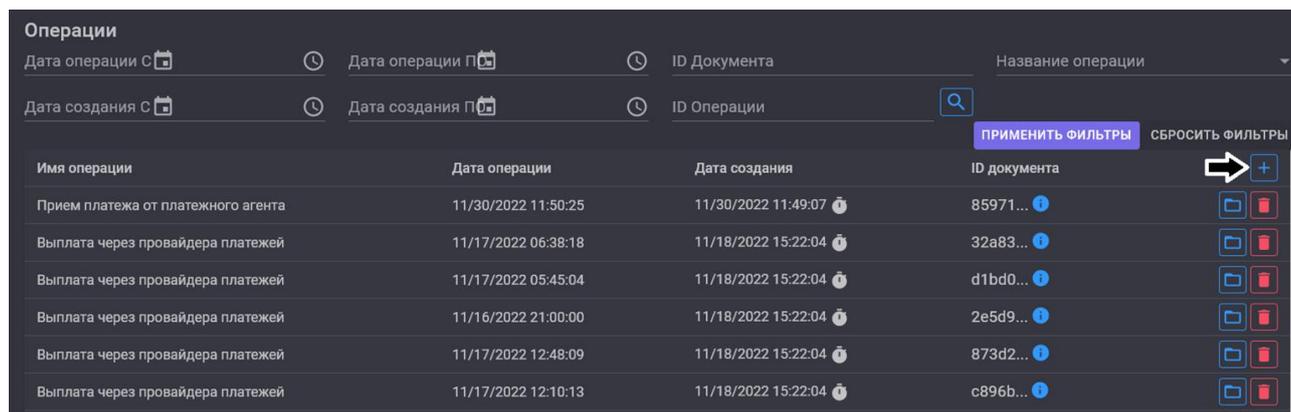
```
{
  "token": null,
  "operations": [
    {
      "documentId": null,
      "operationName": "Требование платежа",
      "creationDate": "2022-11-24T13:24:37.4439170Z",
      "operationDate": "2021-12-19T20:59:59.000000Z",
      "operationId": "d028c3f0-8281-40df-b331-96b32dd7a982",
      "stornoOperationId": null,
      "parameters": [
        {
          "name": "contract",
          "value": "12345"
        },
        {
          "name": "description",
          "value": "Платеж по графику от 2021-12-28"
        },
        {
          "name": "organization",
          "value": "Займы Всем"
        }
      ]
    },
    {
      "entries": [
        {
          "entryId": "f6c2c2c0-576c-42b7-a85f-c3dde0a1df92",
          "operationId": "d028c3f0-8281-40df-b331-96b32dd7a982",
          "creationDate": "2022-11-24T13:24:38.2439460Z",
          "accountingDate": "2021-12-19T20:59:59.000000Z",
          "affectingDate": "2021-12-19T20:59:59.000000Z",
          "debitAccountId": "6221e58a-270b-456a-a5e7-0fda94fd441e",
          "creditAccountId": "fe5c1743-a87d-4143-8a92-ec1df0801903",
          "amount": 118.82,
          "description": "Требование (Уплата процентов) (3). Платеж по графику от 2021-12-28",
          "stornoEntryId": null
        },
        ...
      ]
    },
    ...
  ]
}
```

Г д е :

П а р а м е т р	О п и с а н и е
"token"	Токен для получения следующей страницы запроса
"operations"	Блок списка операций см. 3.3. Операция

## 6.1.2. Регистрация Операции

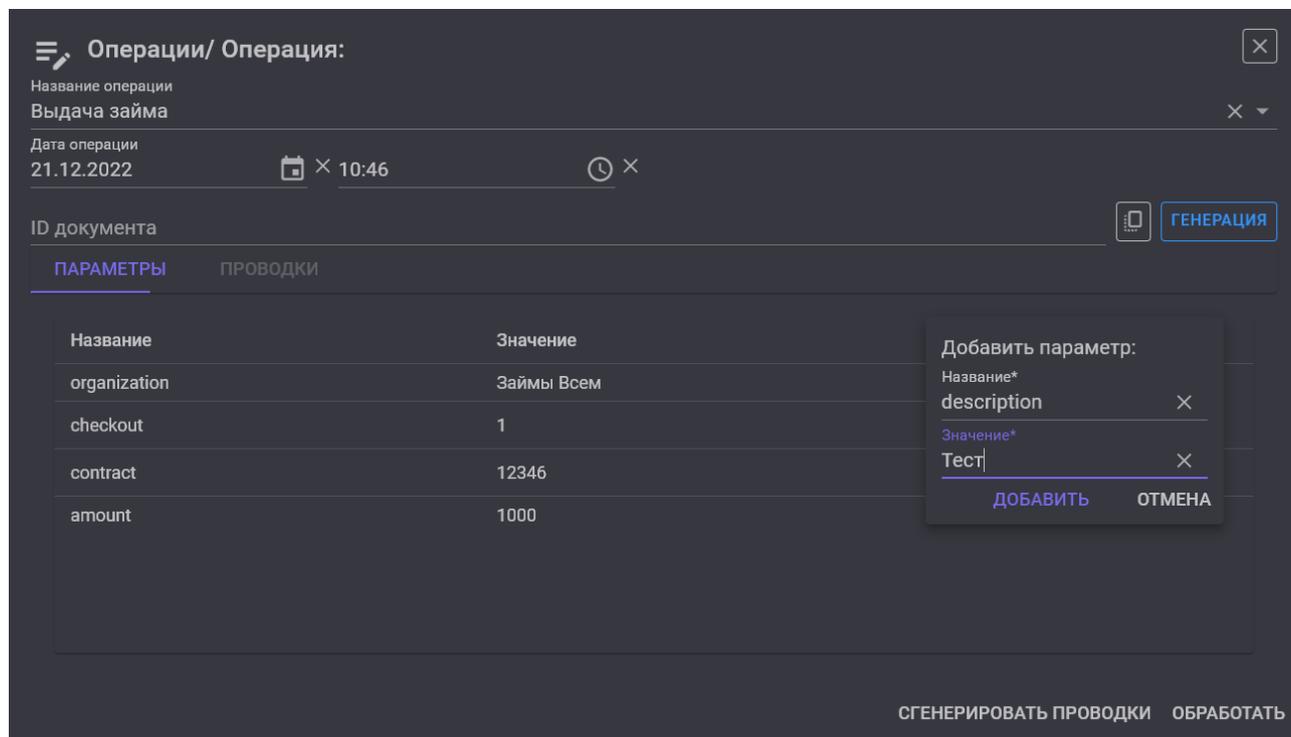
Для регистрации новой операции в разделе "Операции" нажмите кнопку  - "Добавить Операцию" в заголовке таблицы.



Имя операции	Дата операции	Дата создания	ID документа
Прием платежа от платежного агента	11/30/2022 11:50:25	11/30/2022 11:49:07	85971...
Выплата через провайдера платежей	11/17/2022 06:38:18	11/18/2022 15:22:04	32a83...
Выплата через провайдера платежей	11/17/2022 05:45:04	11/18/2022 15:22:04	d1bd0...
Выплата через провайдера платежей	11/16/2022 21:00:00	11/18/2022 15:22:04	2e5d9...
Выплата через провайдера платежей	11/17/2022 12:48:09	11/18/2022 15:22:04	873d2...
Выплата через провайдера платежей	11/17/2022 12:10:13	11/18/2022 15:22:04	c896b...

Рисунок 21: кнопка "Добавить операцию"

В открывшемся диалоге "Операция" заполните "название", "Дату операции" и добавьте необходимые параметры.



Название	Значение
organization	Займы Всем
checkout	1
contract	12346
amount	1000

Рисунок 22: диалог "Добавление операции"

Нажатием на кнопку "Сгенерировать проводки" можно получить список проводок, которые будут сформированы при обработке операции без фактической регистрации операции (Предварительно просмотреть результат)

По завершению заполнения отправляем операцию на обработку нажатием на кнопку "Обработать".

Метод: POST /api/accounting-system/v1/operations

Пример запроса:

POST /api/accounting-system/v1/operations

```
{
  "documentId": null,
  "operationName": "Выдача займа",
  "operationDate": "2021-11-22T09:13:01.000000Z",
  "parameters": [
    {"name": "organization", "value": "Займы Всем"},
    {"name": "checkout", "value": "1"},
    {"name": "contract", "value": "12346"},
    {"name": "amount", "value": "1000"},
    {"name": "description", "value": "Test"}
  ]
}
```

Пример ответа:

```
{
  "documentId": null,
  "operationName": "Выдача займа",
  "creationDate": "2022-12-21T07:38:20.9654640Z",
  "operationDate": "2021-11-22T09:13:01.000000Z",
  "operationId": "59a64052-b73f-401f-9e1f-bed58f95db11",
  "stornoOperationId": null,
  "parameters": [
    {"name": "amount", "value": "1000"},
    {"name": "checkout", "value": "1"},
    {"name": "contract", "value": "12346"},
    {"name": "description", "value": "Test"},
    {"name": "organization", "value": "Займы Всем"}
  ],
  "entries": [
    {
      "entryId": "26a68120-e877-4f89-af7f-3faeb9d23d41",
      "operationId": "59a64052-b73f-401f-9e1f-bed58f95db11",
      "creationDate": "2022-12-21T07:38:21.6594930Z",
      "accountingDate": "2021-11-22T09:13:01.000000Z",
      "affectingDate": "2021-11-22T09:13:01.000000Z",
      "debitAccountId": "598b2acb-fab8-4ae4-985d-9f976831dee9",
      "creditAccountId": "f93603db-cc86-4648-b2c4-c404f826209a",
      "amount": 1000,
      "description": "Выдача займа. Test",
      "stornoEntryId": null
    }
  ]
}
```

См. 3.3. Операция

### 6.1.3. Массовая регистрация Операций

Для регистрации большого количества операций (например начисление процентов по списку договоров) предусмотрен отдельный асинхронный метод регистрации пакета операций.

**Внимание!!!** порядок обработки пакета операций не гарантирован.

Метод: POST /api/accounting-system/v1/async/operations/batch

#### Пример запроса:

POST /api/accounting-system/v1/async/operations/batch

```
[
  {
    "documentId": "ee72d129-1e7a-42f4-a77e-e0fce274a4e1",
    "operationName": "Р а с ч ё т  п р о ц е н т о в  и  п е н и",
    "operationDate": "2021-12-19T21:00:00.000000Z",
    "parameters": [
      {"name": "organization", "value": "З а й м ы  В с е м"},
      {"name": "contract", "value": "12345"},
      {"name": "begin", "value": "2021-12-20"},
      {"name": "end", "value": "2021-12-21"},
      {"name": "interest_rate", "value": "1"},
      {"name": "penalty_rate", "value": "0.5"},
      {"name": "description", "value": "Р а с ч ё т  п р о ц е н т о в  и  п е н и  п о  2021-12-29"}
    ]
  },
  {
    "documentId": "ee72d129-1e7a-42f4-a77e-e0fce274a4e1",
    "operationName": "Р а с ч ё т  п р о ц е н т о в  и  п е н и",
    "operationDate": "2021-12-19T21:00:00.000000Z",
    "parameters": [
      {"name": "organization", "value": "З а й м ы  В с е м"},
      {"name": "contract", "value": "12346"},
      {"name": "begin", "value": "2021-12-20"},
      {"name": "end", "value": "2021-12-21"},
      {"name": "interest_rate", "value": "1"},
      {"name": "penalty_rate", "value": "0.5"},
      {"name": "description", "value": "Р а с ч ё т  п р о ц е н т о в  и  п е н и  п о  2021-12-29"}
    ]
  }
]
```

#### Пример ответа:

b7a8d374-08bf-4a7f-a674-ad485fa03dae

Для просмотра результата обработки можно воспользоваться методом GET /api/accounting-system/v1/async/operations/status/create/batch/{correlationId}

{correlationId} получается в ответ на регистрацию пакета.

#### Пример запроса:

GET /api/accounting-system/v1/async/operations/status/create/batch/b7a8d374-08bf-4a7f-a674-ad485fa03dae

#### Пример ответа:

```
{
  "batchId": "b7a8d374-08bf-4a7f-a674-ad485fa03dae",
  "totalOperations": 2,
  "completed": 2,
  "operations": [
    {
      "correlationId": "66b4bfe3-020b-4efe-9331-dd83ef13d8ca",
      "currentState": "Completed"
    },
    {
      "correlationId": "78550940-108e-46ad-8e47-32746fe6a80c",
      "currentState": "Completed"
    }
  ]
}
```

Где:

Параметр	Описание
"batchId"	Идентификатор пакета
"totalOperations"	Общее количество операций в пакете
"completed"	Количество успешно обработанных операций
"operations"	Раздел список операций
"correlationId"	Идентификатор зарегистрированной Операции
"currentState"	Результат обработки операции "Completed" – успешно обработана. "Faulted" – обработана с ошибкой

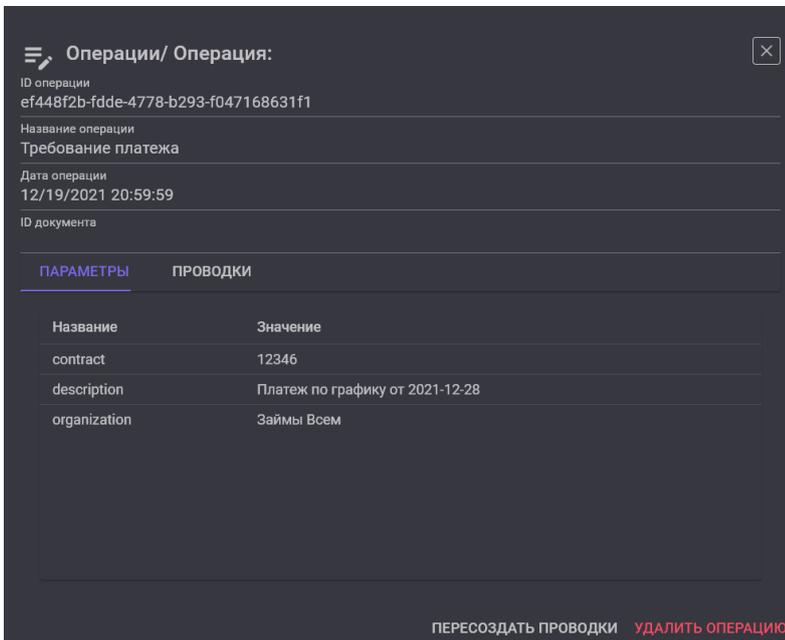
#### 6.1.4. Просмотр Операции

Для просмотра "Операции" найдите интересующую операцию в списке и нажмите кнопку

 - "Просмотр операции"

*Рисунок 23: кнопка "Просмотр операции"*

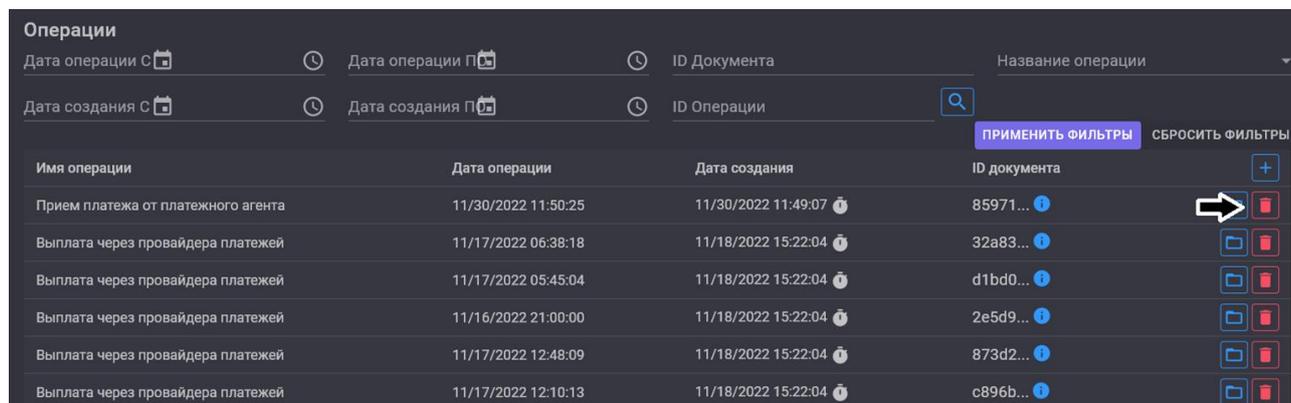
В открывшемся диалоге "Операция", вы можете просмотреть параметры операции и проводки, а также запустить процесс пересоздания проводок или удалить операцию.



*Рисунок 24: диалог "Операция"*

## 6.1.5. Удаление Операции

Для удаления "Операции" в списке операций в строчке с интересующей Операции, нажмите кнопку  - "Удалить Операцию"



Имя операции	Дата операции	Дата создания	ID документа	
Прием платежа от платежного агента	11/30/2022 11:50:25	11/30/2022 11:49:07	85971...	
Выплата через провайдера платежей	11/17/2022 06:38:18	11/18/2022 15:22:04	32a83...	
Выплата через провайдера платежей	11/17/2022 05:45:04	11/18/2022 15:22:04	d1bd0...	
Выплата через провайдера платежей	11/16/2022 21:00:00	11/18/2022 15:22:04	2e5d9...	
Выплата через провайдера платежей	11/17/2022 12:48:09	11/18/2022 15:22:04	873d2...	
Выплата через провайдера платежей	11/17/2022 12:10:13	11/18/2022 15:22:04	c896b...	

Рисунок 25: кнопка "Удалить Операцию"

**Внимание!!!** При удалении Операции проводки созданные при обработке операции будут автоматически удалены.

Пример API запроса:

```
DELETE /api/accounting-system/v1/operations/66b4bfe3-020b-4efe-9331-dd83ef13d8ca
```

## 6.2. Проводки

### 6.2.1. Список проводок и их поиск

Для просмотра списка операций перейдите в раздел "Проводки"

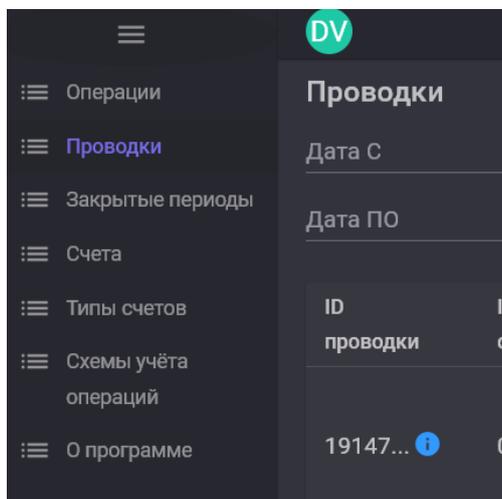


Рисунок 26: раздел "Проводки"

При переходе в раздел вы видите первую страницу списка проводок.

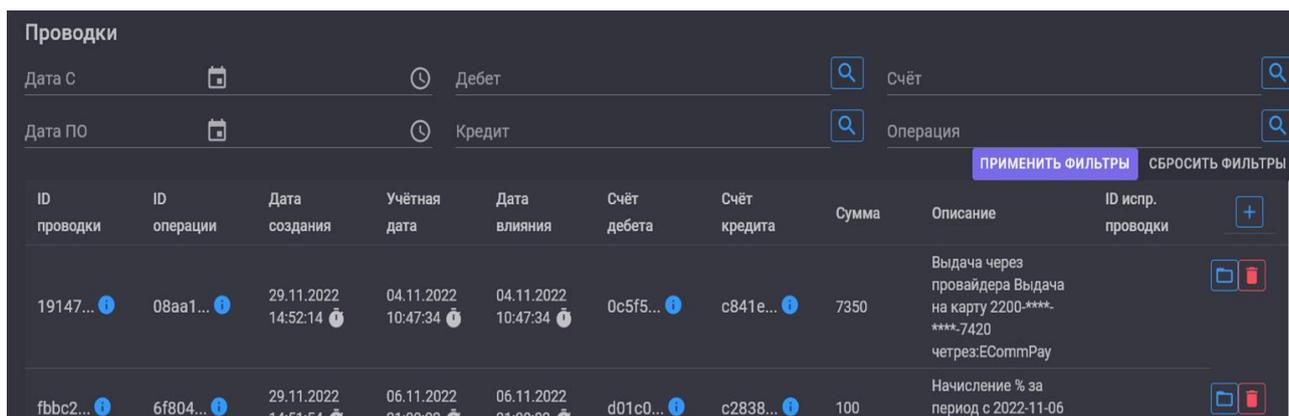


Рисунок 27: Раздел "Проводки"

В разделе Проводки вы можете осуществлять поиск операций по "Дате учета" заполняя поля "Дата с" и "Дата по".

Так же можно искать операции по Ид. "Счета Дебета", Ид. "Счета Кредита", Ид. "Операции"

Заполнив поле «Счет» Поиск будет осуществляться и в счете "Счета Дебета" и "Счета Кредита"

Метод: GET /api/accounting-system/v1/entries

П а р а м е т р	О п и с а н и е
begin-date	Выбрать проводки с датой учета после begin-date
end-date	Выбрать проводки с датой учета до end-date
account-id	Выбрать проводки изменяющие остаток счета account-id
debit-account-id	Выбрать проводки с указанным Ид. "Счета Дебита"

credit-account-id	Выбрать проводки с указанным Ид. "Счета Кредита"
operation-id	Выбрать проводки с указанной Ид. "Операции"
page-size	Количество операций в выборке
page-token	Токен следующего пакета выборки

### Пример запроса:

```
GET /api/accounting-system/v1/entries?begin-date=2011-08-11%2009%3A56%3A52%2B03&end-date=2021-08-11%2009%3A56%3A52%2B03&account-id=8d396728-a761-4c3c-9a51-0c3666ae3215
```

### Пример ответа:

```
{
  "token": null,
  "entries": [
    {
      "entryId": "c0e6a0b1-343d-4f84-a944-d3102ff05eff",
      "operationId": "75ef6ef6-19df-4290-b29e-6393bdf171e7",
      "createDate": "2021-08-05T10:11:30.9352590Z",
      "accountingDate": "2021-04-05T21:00:00.0000000Z",
      "affectingDate": "2021-04-05T21:00:00.0000000Z",
      "debitAccountId": "8d396728-a761-4c3c-9a51-0c3666ae3215",
      "creditAccountId": "5a615731-db39-41ed-a25b-ad53c8d53c8c",
      "amount": 51500,
      "description": "Подготовка суммы для уплаты процентов Принял
сотрудник: Васькина Дана Владимировна От: КАССИН ДМИТРИЙ ВАЛЕРЬЕВИЧ / 12/29/1988
ГОР. КИРОВ / ",
      "stornoOperationId": null
    },
    {
      "entryId": "ae8192f8-a1d3-429b-b407-85d4ed3c6b3e",
      "operationId": "61b95b41-a1b5-4a32-84b3-8bdaa42f127e",
      "createDate": "2021-08-05T10:07:19.5370390Z",
      "accountingDate": "2020-01-02T21:00:00.0000000Z",
      "affectingDate": "2020-01-02T21:00:00.0000000Z",
      "debitAccountId": "dfdbb4b2-0b9a-4a1f-be67-3eeca91df89d",
      "creditAccountId": "8d396728-a761-4c3c-9a51-0c3666ae3215",
      "amount": 20600,
      "description": "Выдача через кассу Самовывоз сотрудник: Васькин Дмитрий
Павлович",
      "stornoOperationId": null
    }
  ]
}
```

Параме	Описание
--------	----------

"token"	Токен для получения следующей страницы запроса
"entries"	Блок списка проводок см. 3.5. Проводка 3.5. Проводка

## 6.2.2. Регистрация Проводки

Для регистрации новой проводки в разделе "Проводки" нажмите кнопку  - "Добавить проводку" в заголовке таблицы.

Рисунок 28: кнопка "Добавить проводку"

В открывшемся диалоге "Проводка" заполните "ID операции", "Учетная дата" и «Дата влияния на расчет», «Счет дебета», «Счет кредита», «Назначение проводки» и при необходимости «ID испр. Проводки».

"ID операции", «Счет дебета», «Счет кредита» можно заполнить при помощи диалога поиска. Вызывается он по кнопке  - поиск

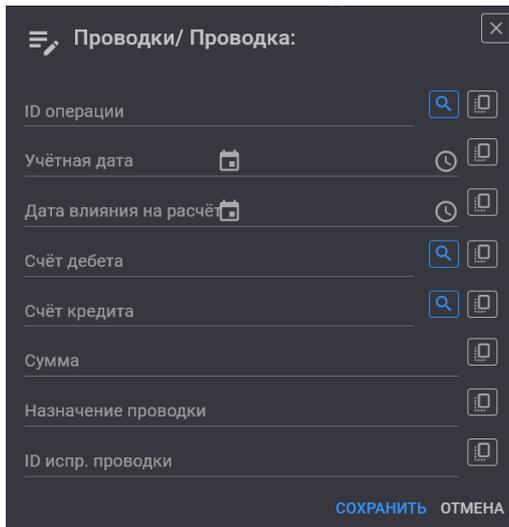


Рисунок 29: диалог "Добавление проводки"

По завершению заполнения регистрируем проводку нажатием на кнопку "Сохранить".

Метод: POST /api/accounting-system/v1/entries

#### Пример запроса:

POST /api/accounting-system/v1/entries

```
{
  "operationId": "98403a4f-f5fb-4dce-ab77-21a2de43f162",
  "accountingDate": "2021-12-18T21:00:01.000000Z",
  "affectingDate": "2021-12-18T21:00:00.000000Z",
  "debitAccountId": "4d874db7-ccf9-4d4e-b9b5-e028ea5a9370",
  "creditAccountId": "f2254a9e-a7ca-41f3-a845-b8c12ab7820a",
  "amount": 8.4871,
  "description": "Начисление процентов. Расчёт процентов и пени за 2021-12-19"
}
```

#### Пример ответа:

```
{
  "entryId": "662b2ea0-d2d1-48f9-9517-bf22c1c2d3fd",
  "operationId": "98403a4f-f5fb-4dce-ab77-21a2de43f162",
  "creationDate": "2022-12-27T08:03:00.5153228Z",
  "accountingDate": "2021-12-18T21:00:01.000000Z",
  "affectingDate": "2021-12-18T21:00:00.000000Z",
  "debitAccountId": "4d874db7-ccf9-4d4e-b9b5-e028ea5a9370",
  "creditAccountId": "f2254a9e-a7ca-41f3-a845-b8c12ab7820a",
  "amount": 8.4871,
  "description": "Начисление процентов. Расчёт процентов и пени за 2021-12-19",
  "stornoEntryId": null
}
```

С м. 3.5. Проводка

Рисунок 30: кнопка "Просмотр проводки"

В открывшемся диалоге "Проводка" вы можете просмотреть/скопировать атрибуты проводки, а также удалить проводку.

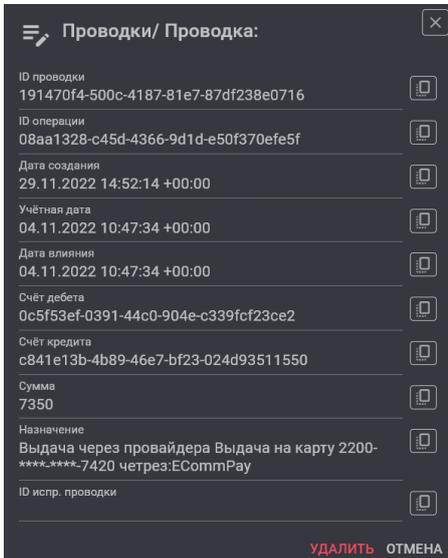


Рисунок 31: диалог "Проводка"

## 6.2.4. Удаление Проводки

Для удаления "Проводки" в списке операций в строчке с интересующей проводкой нажмите кнопку  - "Удалить проводку"

Рисунок 32: кнопка "Удалить Проводку"

Пример API запроса:

```
DELETE /api/accounting-system/v1/entries/662b2ea0-d2d1-48f9-9517-bf22c1c2d3fd
```

## 6.3. Аналитические счета

### 6.3.1. Получение списка счетов и поиск аналитических счетов

Для просмотра списка операций перейдите в раздел "Счета"

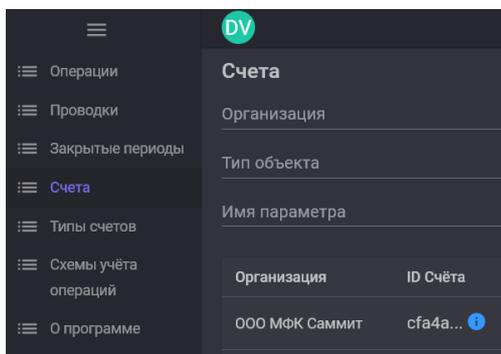


Рисунок 33: раздел "Счета"

При переходе в раздел вы видите первую страницу списка счетов.

Рисунок 34: Раздел "Счета"

В разделе Счета вы можете осуществлять поиск аналитических счетов по "Организации", "ID объекта", "Тип счета", "Тип объекта" или по паре "Имя параметра", "Значение параметра".

Метод: GET /api/accounting-system/v1/accounts

П а р а м е т р	О п и с а н и е
object-id	ID объекта во внешней системе
object-type	Тип объекта
organization-id	ID организации
account-type-id	ID типа счета
ext-parameter-name	Имя параметра
ext-parameter-value	Значение параметра
page-size	Количество объектов в возвращаемом ответе
page-token	Токен для запроса следующей порции данных

Пример запроса:

```
GET /api/accounting-system/v1/accounts?object-id=a42a9998-53ac-44ed-aa73-9d9f26dd2ca1
```

Пример ответа:

```
{
  "token": null,
  "analyticalAccounts": [
    {
      "accountId": "ac44dca1-dd9c-448f-964a-b7db4447e578",
      "creationDate": "2022-12-22T14:42:11.6206140Z",
      "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
      "objectId": "a42a9998-53ac-44ed-aa73-9d9f26dd2ca1",
      "objectType": "11111111-0000-4a41-b0e9-111111111111",
      "apType": "активный",
      "accountTypeId": "682f6412-4f15-4b7e-a1f1-58605cca0a71",
      "accountTypeName": "Пени уплаченные",
      "extParameters": [
        {"name": "EPS", "value": "-"},
        {"name": "officeName", "value": "Офис 2"},
        {"name": "product", "value": "Просто Займ"},
      ]
    }
  ]
}
```

```

    {"name": "region", "value": "К а з а н ь"}
  ]
},
...
{
  "accountId": "598b2acb-fab8-4ae4-985d-9f976831dee9",
  "creationDate": "2022-12-16T14:29:52.0916990Z",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "objectId": "a42a9998-53ac-44ed-aa73-9d9f26dd2ca1",
  "objectType": "11111111-0000-4a41-b0e9-111111111111",
  "apType": "а к т и в н ы й",
  "accountTypeId": "af6b16b3-eca7-4c28-8c18-374e7978c40f",
  "accountTypeName": "О с н о в н о й д о л г",
  "extParameters": [
    {"name": "EPS", "value": "-"},
    {"name": "EPS2", "value": "3121654564"},
    {"name": "officeName", "value": "О ф и с 2"},
    {"name": "product", "value": "П р о с т о З а й м"},
    {"name": "region", "value": "К а з а н ь"}
  ]
}
]
}
}

```

П а р а м е т р	О п и с а н и е
"token"	Токен для получения следующей страницы запроса
"analyticalAccounts"	Блок списка аналитических счетов см. 3.4. Аналитический счет

### 6.3.2. Регистрация аналитического счета

Регистрация аналитического счета происходит автоматически при вызове метода GetAccount в правилах создания проводок. См. 5.1.7. Функции работы со счетами:

Для "ручной" регистрации новой нового счета в разделе "Счета" нажмите кнопку  - "Добавить счет" в заголовке таблицы.

*Рисунок 35: кнопка "Добавить счет"*

В открывшемся диалоге "Счет" заполните "ID Организации", "ID Объекта", «Тип объекта», «Типа счета».

*Рисунок 36: диалог "Добавление счета"*

По завершению заполнения регистрируем счет нажатием на кнопку "Сохранить".

Метод: POST /api/accounting-system/v1/accounts

Пример запроса:

```

POST /api/accounting-system/v1/accounts
{
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "objectId": "3cc39b29-a492-49cb-aae5-101605bfed7b",
  "objectType": "11111111-0000-4a41-b0e9-111111111111",
  "accountTypeId": "682f6412-4f15-4b7e-a1f1-58605cca0a71"
}

```

Пример ответа:

```

{
  "accountId": "ec10714e-95d4-4303-858d-794e0005fe2c",
  "creationDate": "2022-12-27T09:35:38.3305807Z",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",

```

```

"objectId": "3cc39b29-a492-49cb-aae5-101605bfd7b",
"objectType": "11111111-0000-4a41-b0e9-111111111111",
"apType": "активный",
"accountTypeId": "682f6412-4f15-4b7e-a1f1-58605cca0a71",
"accountTypeName": "Пени уплаченные",
"extParameters": [
  {
    "name": "EPS",
    "value": "-"
  },
  {
    "name": "officeName",
    "value": "Офис 1"
  },
  {
    "name": "product",
    "value": "Льготный Займ"
  },
  {
    "name": "region",
    "value": "Москва"
  }
]
}

```

См. 3.4. Аналитический счет

### 6.3.3. Просмотр/Редактирование аналитического счета

Для просмотра или редактирования "Аналитического счета" найдите интересующий счет в списке и нажмите кнопку  - "Просмотр/редактирование счета"

Рисунок 37: кнопка "Просмотр/редактирование счета"

В открывшемся диалоге "Редактирование счёта" вы можете просмотреть/изменить атрибуты и параметры счета, а также удалить аналитический счет.

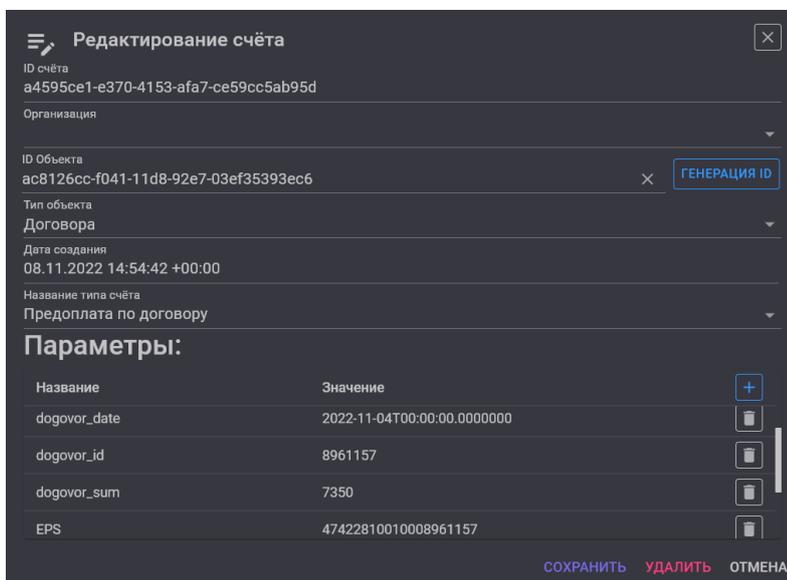


Рисунок 38: диалог "Редактирование счета"

Метод: GET /api/accounting-system/v1/accounts/{accountId}

П а р а м е т р	О п и с а н и е
{accountId}	ID аналитического счета

Пример запроса:

GET /api/accounting-system/v1/accounts/ec10714e-95d4-4303-858d-794e0005fe2c

## Пример ответа:

```
{
  "accountId": "ec10714e-95d4-4303-858d-794e0005fe2c",
  "creationDate": "2022-12-27T09:35:38.3305800Z",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "objectId": "3cc39b29-a492-49cb-aae5-101605bfd7b",
  "objectType": "11111111-0000-4a41-b0e9-111111111111",
  "apType": "активный",
  "accountTypeId": "682f6412-4f15-4b7e-a1f1-58605cca0a71",
  "accountTypeName": "Пени уплаченные",
  "extParameters": [
    {
      "name": "EPS",
      "value": "-"
    },
    {
      "name": "officeName",
      "value": "Офис 1"
    },
    {
      "name": "product",
      "value": "Льготный Займ"
    },
    {
      "name": "region",
      "value": "Москва"
    }
  ]
}
```

Г д е :

П а р а м е т р	О п и с а н и е
"accountId"	ID счета
"creationDate"	Дата создания
"organizationId"	ID организации (НСИ)
"objectId"	ID объекта во внешней системе (ИД Договора, Кассы и т.п.)
"objectType"	ID типа объекта
"apType"	"пассивный", "активный", "активно-пассивный"
"accountTypeId"	ID типа счета (см. Тип Счета (AccountType))
"extParameters"	Блок доп. Параметров (Настраиваемые атрибуты)
"name"	Название доп. параметра
"value"	Значение доп. параметра

### 6.3.4. Удаление аналитического счета

Для удаления "Аналитического счета" в списке счетов в строчке с интересующим счетом нажмите кнопку  - "Удалить счет"

*Рисунок 39: кнопка "Удалить счет"*

Пример API запроса:

```
DELETE /api/accounting-system/v1/accounts/ec10714e-95d4-4303-858d-794e0005fe2c
```

### 6.3.5. Получение баланса счета

Метод: GET /api/accounting-system/v1/accounts/{accountId}/balance

П а р а м е т р	О п и с а н и е
{accountId}	ID счета
date	Дата баланса

Пример запроса:

```
GET /api/accounting-system/v1/accounts/598b2acb-fab8-4ae4-985d-9f976831dee9/balance?date=2021-11-22T09%3A13%3A02.000000Z
```

Пример ответа:

```
1000
```

### 6.3.6. Получение оборота по дебету счета

Метод: GET /api/accounting-system/v1/entries/debit

П а р а м е т р	О п и с а н и е
debit-account-id	ID счета дебета
begin-date	Дата начала периода
end-date	Дата конца периода

Пример запроса:

```
GET /api/accounting-system/v1/entries/debit?debit-account-id=598b2acb-fab8-4ae4-985d-9f976831dee9&begin-date=2021-11-22T09%3A13%3A02.000000Z
```

Пример ответа:

0

### 6.3.7. Получение оборота по кредиту счета

Метод: GET /api/accounting-system/v1/entries/credit

П а р а м е т р	О п и с а н и е
credit-account-id	ID счета дебета
begin-date	Дата начала периода
end-date	Дата конца периода

Пример запроса:

```
GET /api/accounting-system/v1/entries/credit?credit-account-id=598b2acb-fab8-4ae4-985d-9f976831dee9&begin-date=2021-11-22T09%3A13%3A02.000000Z
```

Пример ответа:

323.76

### 6.3.8. Получение отчета «обороты счета»

Метод: GET /api/accounting-system/v1/accounts/{accountId}/turnover-balance

П а р а м е т р	О п и с а н и е
{accountId}	ID счета
date-from	Дата начала периода
date-to	Дата конца периода

Пример запроса:

```
GET /api/accounting-system/v1/accounts/598b2acb-fab8-4ae4-985d-9f976831dee9/turnover-balance?date-from=2021-11-22T09%3A13%3A02.000000Z
```

Пример ответа:

```
{
  "dateFrom": "2021-11-22T09:13:02.000000Z",
  "dateTo": "2022-12-27T09:50:05.3803120Z",
  "accountId": "598b2acb-fab8-4ae4-985d-9f976831dee9",
  "apType": "активный",
  "objectId": "a42a9998-53ac-44ed-aa73-9d9f26dd2ca1",
  "objectType": "11111111-0000-4a41-b0e9-111111111111",
  "debitInBalance": 1000,
  "creditInBalance": 0,
  "debitAmount": 0,
  "creditAmount": 323.76,
  "debitOutBalance": 676.24,
  "creditOutBalance": 0,
  "extParameter": []
}
```

Г д е :

П а р а м е т р	О п и с а н и е
"dateFrom"	Дата начала периода
"dateTo"	Дата конца периода
"accountId"	ID счета
"apType"	Признак активности/пассивности счета
"objectId"	ID объекта во внешней системе (ИД Договора, Кассы и т.п.)
"objectType"	ID типа объекта
"debitInBalance"	Пассивный входящий баланс
"creditInBalance"	Активный входящий баланс
"debitAmount"	Оборот по дебету счета
"creditAmount"	Оборот по кредиту счета

"debitOutBalance"	П а с с и в н ы й и с х о д я щ и й б а л а н с
"creditOutBalance"	А к т и в н ы й и с х о д я щ и й б а л а н с
"extParameter"	Д о п . п а р а м е т р ы с ч е т а

### 6.3.6. Получение срока образования "остатка" счета

Метод предназначен для определения срока образования остатка на счете. Это может использоваться для определения срока образования просрочки по договору.

Метод: /api/accounting-system/v1/accounts/{accountId}/time\_of\_debt

П а р а м е т р	О п и с а н и е
{accountId}	И Д с ч е т а
on-date	Д а т а н а к о т о р у ю о п р е д е л я е т с я с р о к о б р а з о в а н и я з а д о л ж е н н о с т и

Пример запроса:

```
GET /api/accounting-system/v1/accounts/598b2acb-fab8-4ae4-985d-9f976831dee9/time_of_debt?on-date=2021-11-25T09%3A13%3A02.000000Z
```

Пример ответа:

```
{
  "value": 4
}
```

### 6.6. Оборотно-сальдовая ведомость по синтетическим счетам.

Отчет по синтетическому счету. Выдает иерархически организованный (в соответствии со структурой, указанной в плане счетов) отчет, который содержит описание счета оборот по аналитическим счетам того же типа, плюс обороты по субсчету за указанный период.

Параметр detail-by, позволяет рассчитать суммы с разделением по значениям доп. параметра аналитического счета.

Метод: GET /api/accounting-system/v1/charts-of-accounts/{chartOfAccountsId}/accounts/{accountId}/turnover-balance

П а р а м е т р	О п и с а н и е
{chartOfAccountsId}	И д . П л а н а с ч е т о в
{accountId}	И д . С и н т е т и ч е с к о г о С ч е т а д л я к о т о р о г о ф о р м и р у е т с я о т ч е т
date-from	Д а т а н а ч а л а п е р и о д а
date-to	Д а т а к о н ц а п е р и о д а
organization-id	И д . О р г а н и з а ц и и
detail-by	Р а с ш и ф р о в к а п о п а р а м е т р у

## Пример запроса:

```
GET /api/accounting-system/v1/charts-of-accounts/28a03813-4676-466b-986a-cd261ec8f4ea/accounts/a706bf01-fac7-4591-b84c-686c45adb25b/turnover-balance?date-from=2021-12-01&date-to=2022-01-01
```

## Пример ответа:

```
{
  "dateFrom": "2021-12-01T00:00:00.0000000Z",
  "dateTo": "2022-01-01T00:00:00.0000000Z",
  "accountId": "a706bf01-fac7-4591-b84c-686c45adb25b",
  "name": "Пассивы по договорам",
  "apType": "пассивный",
  "debitInBalance": 0,
  "creditInBalance": 200,
  "debitAmount": 291.29,
  "creditAmount": 735.06815,
  "debitOutBalance": 0,
  "creditOutBalance": 643.77815,
  "children": [
    {
      "dateFrom": "2021-12-01T00:00:00.0000000Z",
      "dateTo": "2022-01-01T00:00:00.0000000Z",
      "accountId": "22b81092-c14f-4c74-9038-2bb27483ffa0",
      "name": "Доход по пени",
      "apType": "пассивный",
      "debitInBalance": 0,
      "creditInBalance": 0,
      "debitAmount": 0,
      "creditAmount": 0.75645,
      "debitOutBalance": 0,
      "creditOutBalance": 0.75645,
      "children": [],
      "details": [
        {
          "accounts": [],
          "extParameter": "Прочее",
          "debitInBalance": 0,
          "creditInBalance": 0,
          "debitAmount": 0,
          "creditAmount": 0.75645,
          "debitOutBalance": 0,
          "creditOutBalance": 0.75645
        }
      ]
    }
  ],
  "details": [
    {
      "accounts": [],
      "extParameter": "Прочее",
      "debitInBalance": 0,
      "creditInBalance": 200,
      "debitAmount": 291.29,
      "creditAmount": 735.06815,
      "debitOutBalance": 0,
      "creditOutBalance": 643.77815
    }
  ]
}
```

## Пример2 запроса (С детализацией):

```
GET /api/accounting-system/v1/charts-of-accounts/28a03813-4676-466b-986a-cd261ec8f4ea/accounts/a706bf01-fac7-4591-b84c-686c45adb25b/turnover-balance?date-from=2021-12-01&date-to=2022-01-01&detail-by=officeName
```

## Пример2 ответа:

```
{
  "dateFrom": "2021-12-01T00:00:00.000000Z",
  "dateTo": "2022-01-01T00:00:00.000000Z",
  "accountId": "a706bf01-fac7-4591-b84c-686c45adb25b",
  "name": "Пассивы по договорам",
  "apType": "пассивный",
  "debitInBalance": 0,
  "creditInBalance": 200,
  "debitAmount": 291.29,
  "creditAmount": 735.06815,
  "debitOutBalance": 0,
  "creditOutBalance": 643.77815,
  "children": [
    ...
    {
      "dateFrom": "2021-12-01T00:00:00.000000Z",
      "dateTo": "2022-01-01T00:00:00.000000Z",
      "accountId": "5dd4a32c-00d9-434a-936d-7acff3153bb0",
      "name": "Доход по процентам",
      "apType": "пассивный",
      "debitInBalance": 0,
      "creditInBalance": 200,
      "debitAmount": 0,
      "creditAmount": 319.1517,
      "debitOutBalance": 0,
      "creditOutBalance": 519.1517,
      "children": [],
      "details": [
        {
          "accounts": [],
          "extParameter": "officeName: Офис 1",
          "debitInBalance": 0,
          "creditInBalance": 100,
          "debitAmount": 0,
          "creditAmount": 158.8194,
          "debitOutBalance": 0,
          "creditOutBalance": 258.8194
        },
        {
          "accounts": [],
          "extParameter": "officeName: Офис 2",
          "debitInBalance": 0,
          "creditInBalance": 100,
          "debitAmount": 0,
          "creditAmount": 160.3323,
          "debitOutBalance": 0,
          "creditOutBalance": 260.3323
        }
      ]
    },
    ...
  ],
  "details": [
    {
      "accounts": [],
      "extParameter": "officeName: Офис 2",
      "debitInBalance": 0,
      "creditInBalance": 100,
      "debitAmount": 291.29,
      "creditAmount": 567.53875,
      "debitOutBalance": 0,
      "creditOutBalance": 376.24875
    },
    {
      "accounts": [],
      "extParameter": "officeName: Офис 1",
      "debitInBalance": 0,
      "creditInBalance": 100,
      "debitAmount": 0,
      "creditAmount": 167.5294,
      "debitOutBalance": 0,

```

```
    "creditOutBalance": 267.5294  
  }  
]  
}
```

Где:

Параметр	Описание
"dateFrom"	Отчет сформирован за период начиная с этой даты
"dateTo"	Отчет сформирован за период начиная до этой даты
"accountId"	Ид. синтетического счета
"name"	Описание синтетического счета
"apType"	Признак активный/пассивный счета
"debitInBalance"	Пассивный входящий баланс
"creditInBalance"	Активный входящий баланс
"debitAmount"	Оборот по дебету счета
"creditAmount"	Оборот по кредиту счета
"debitOutBalance"	Пассивный исходящий баланс
"creditOutBalance"	Активный исходящий баланс
"children"	Информация по субсчетам (Имеет ту же структуры и может содержать собственные субсчета)
"details"	Блок детализации.
"extParameter"	Значение доп. Параметра счетов по которому рассчитана детализация.
"accounts"	Список синтетических счетов (Заполняется в случае запроса с детализацией по account-id) см. 3.4. Аналитический счет

## 6.7. Заккрытие периода

### 6.7.1. Просмотр списка закрытых периодов системы

Для просмотра списка операций перейдите в раздел "Закрытые периоды"

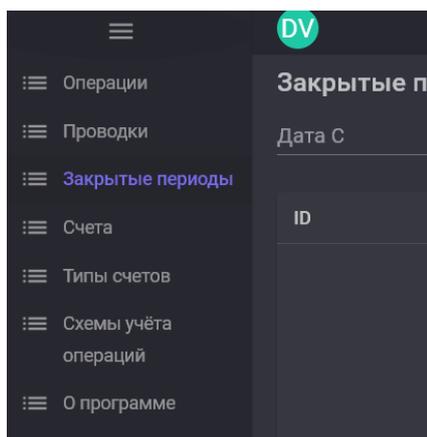


Рисунок 40: раздел "Закрытые периоды"

При переходе в раздел вы видите первую страницу списка счетов.

ID	Дата	Описание	
dd61cc50-28cd-437e-ae95-4dad4b6e5ac4	12/01/2022 00:00:00		ОТКРЫТЬ ПЕРИОД
fa873956-5776-4982-ab8f-ac96e450f2d3	12/04/2022 00:00:00		ОТКРЫТЬ ПЕРИОД

Рисунок 41: Раздел "Закрытые периоды"

В разделе Счета вы можете осуществлять поиск закрытых периодов по дате и организации

Регистрировать новый «Закрытый период» нажатием на кнопку «Заккрыть период».

Удалить существующий «Закрытый период» нажатием на кнопку «Открыть период».

Метод: GET /api/accounting-system/v1/{organizationId}/closed-periods

П а р а м е т р	О п и с а н и е
organization-id	ID организации
"dateFrom"	З ак р ы т ы е п е р и о д ы с date > dateFrom
"dateTo"	З ак р ы т ы е п е р и о д ы с date < dateTo

Пример запроса:

GET /api/accounting-system/v1/eef268f7-33aa-4772-b37d-f86a8626603f/closed-periods

Пример ответа:

```
[
  {
    "closedPeriodId": "82d66c6f-0436-4290-bc1b-f84453659d18",
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "date": "2022-01-01T00:00:00.000000Z",
    "description": "З ак р ы т и е 2022 г о д а"
  },
  ...
]
```

см. 3.8. Закрытый период

## 6.7.2. Закрытие периода (Регистрация)

Для регистрации нового закрытого периода перейдите в раздел «Закрытые периоды» и нажмите кнопку «Заккрыть период»

ID	Дата	Описание	
dd61cc50-28cd-437e-ae95-4dad4b6e5ac4	12/01/2022 00:00:00		ОТКРЫТЬ ПЕРИОД
fa873956-5776-4982-ab8f-ac96e450f2d3	12/04/2022 00:00:00		ОТКРЫТЬ ПЕРИОД

Рисунок 42: кнопка "Заккрыть период"

В диалоге. Заккрыть период заполните «дату», «описание» и «организацию» и нажмите кнопку «ок».

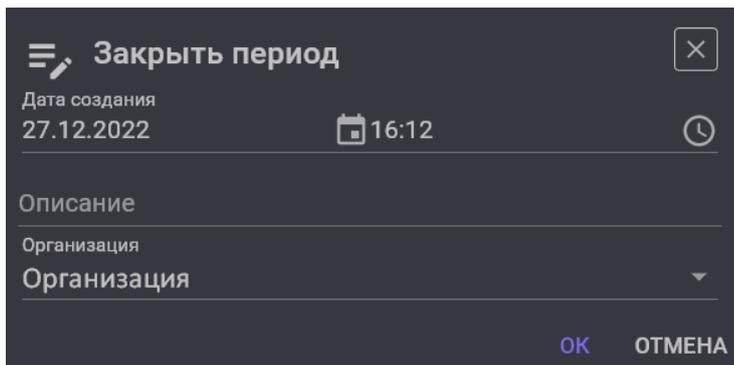


Рисунок 43: диалог "Закреть период"

Метод: `POST /api/accounting-system/v1/{organizationId}/closed-periods`

П а р а м е т р	О п и с а н и е
organization-id	ID организации
"date"	Дата и время по которую блокируется создание проводок
"description"	Человеко читаемое описание

Пример запроса:

```
POST /api/accounting-system/v1/eef268f7-33aa-4772-b37d-f86a8626603f/closed-periods
{
  "date": "2022-01-01",
  "description": "З а к р ы т и е 2022 г о д а"
}
```

Пример ответа:

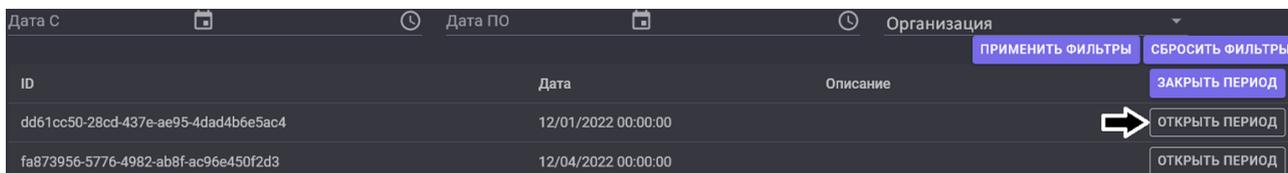
```
{
  "closedPeriodId": "82d66c6f-0436-4290-bc1b-f84453659d18",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "date": "2022-01-01T00:00:00.000000Z",
  "description": "З а к р ы т и е 2022 г о д а"
}
```

см. 3.8. Закрытый период

### 6.7.3. Отмена закрытия периода (Удаление)

Для удаления закрытого периода перейдите в раздел «Закрытые периоды» и нажмите кнопку «Открыть период».

Внимание!!! Если удален период в середине списка, то проводки все равно не смогут регистрироваться. т.к. существует другой закрытый период с большей датой.



Метод: `DELETE /api/accounting-system/v1/{organizationId}/closed-periods/{closedPeriodId}`

П а р а м е т р	О п и с а н и е
organizationId	ID организации
closedPeriodId	ID закрытого периода

Пример запроса:

```
DELETE /api/accounting-system/v1/eef268f7-33aa-4772-b37d-f86a8626603f/closed-periods/82d66c6f-0436-4290-bc1b-f84453659d18
```

## Приложение 1. Пример Учета займа

Вам потребуется установленный интерпретатор python для работы с примером.

Модуль УД предназначен интеграции в информационную систему организации.

Для создания демонстрационного окружения (далее «сервис объектов») сервиса запустите сервис objects.py из дистрибутива. Для работы сервиса требуются файлы contracts.json, organizations.json, checkouts.json

Пример объекта договор:

```
{
  "id": "03e04e76-f1c7-419c-8c45-25af5fe8eb40",
  "number": "12350",
  "product": "Льготный Займ",
  "officeName": "Офис 3",
  "region": "Санкт-Петербург"
}
```

Обязательные параметры

**id** – guid договора в системах организации

**number** — «номер» договора. Текстовый идентификатор или юридический номер

Остальные параметры введены для демонстрации формирования параметров счетов.

Методы сервиса для объекта договор

Для манипулирования объектами договор используется Rest API

Метод	Endpoint	Описание
GET	/contracts?id=<id>&number=<number>	Поиск объекта договор по id или number
GET	/contracts/<id>	Поиск объекта договор по id
POST	/contracts?id=<id>	Создание/Изменение объекта договор по id
PUT	/contracts?id=<id>	Создание/Изменение объекта договор по id
PUT	/contracts/<id>	Создание/Изменение объекта договор по id
POST	/contracts/<id>	Создание/Изменение объекта договор по id
DELETE	/contracts?id=<id>&number=<number>	Удаление объекта договор по id или number
DELETE	/contracts/<id>	Удаление объекта договор по id или number

Пример объекта организация

```
{
  "id": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "name": "Займы Всем",
  "inn": "25623784563487568"
}
```

Обязательные параметры

**id** - guid организации в системах организации/группы компаний

**name** - Название организации.

Остальные параметры введены для демонстрации формирования параметров счетов.

Методы сервиса для объекта организация

Метод	Endpoint	Описание
GET	/organizations?id=<id>&number=<name>	Поиск объекта организация по id или name
GET	/organizations/<id>	Поиск объекта организация по id
POST	/organizations?id=<id>	Создание/Изменение объекта организация по id
PUT	/organizations?id=<id>	Создание/Изменение объекта организация по id
PUT	/organizations/<id>	Создание/Изменение объекта организация по id
POST	/organizations/<id>	Создание/Изменение объекта организация по id
DELETE	/organizations?id=<id>&number=<name>	Удаление объекта организация по id или number
DELETE	/organizations/<id>	Удаление объекта организация по id или name

### Пример объекта касса

```
{
  "id": "cd42b975-aa3e-4042-82f6-97edd97e1bc4",
  "number": "1",
  "officeName": "О ф и с 1",
  "region": "М о с к в а"
}
```

### Обязательные параметры

**id** – guid кассы в системах организации

**number** — «номер» офиса/кассы. Текстовый идентификатор или юридический номер

Остальные параметры введены для демонстрации формирования параметров счетов.

### Методы сервиса для объекта касса

Метод	Endpoint	Описание
GET	/checkouts?id=<id>&number=<name>	Поиск объекта касса по id или name
GET	/checkouts/<id>	Поиск объекта касса по id
POST	/checkouts?id=<id>	Создание/Изменение объекта касса по id
PUT	/checkouts?id=<id>	Создание/Изменение объекта касса по id
PUT	/checkouts/<id>	Создание/Изменение объекта касса по id
POST	/checkouts/<id>	Создание/Изменение объекта касса по id
DELETE	/checkouts?id=<id>&number=<name>	Удаление объекта касса по id или number
DELETE	/checkouts/<id>	Удаление объекта касса по id или name

С целью демонстрации возможностей настройки модуля использована упрощенная конфигурация учета договоров займа.

Скрипт config.py приложенный в дистрибутиве, производит настройку модуля упрощенной конфигурацией учета договоров займа

В качестве параметра скрипту передается базовый адрес REST API сервиса УД и сервиса, симулирующего окружение.

```
python config.py http://{ip сервера УД}:55002 http://{ip сервис объектов}:5000
```

Будут настроены "Схемы учета операций"

- Выдача займа
- Расчёт процентов и пени
- Платеж по графику
- Требование платежа
- Платеж

Будут настроены "Типы счетов"

- Касса
- Доход по процентам
- Доход по пени
- Основной долг
- Процент начисленный
- Пеня начисленные
- Платеж ОД
- Платеж Процентов
- Просроченный платеж ОД
- Просроченный платеж Процентов
- ОД Уплаченный
- Предоплата(Текущий счет)
- ОД Уплаченный
- Процент уплаченный
- Пеня уплаченные

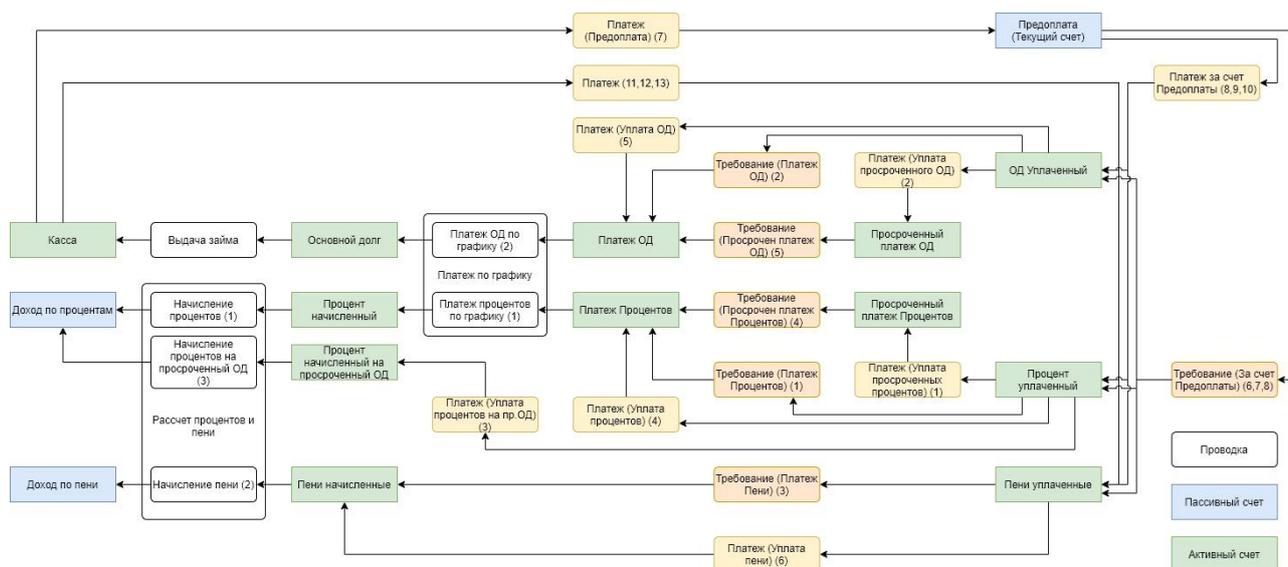


Рисунок 44: Схема движения денежных средств демонстрационной конфигурации

## Описания потока операций договора

В нашей «Упрощенной» схеме учет договора займа предполагает следующий порядок операций:

1. Выдача займа
2. Ежедневно в начале дня (00:00:00) регистрируется операция «Расчёт процентов и пени»
3. В дату платежа по графику, после начисления процентов (00:00:01) регистрируется операция «Платеж по графику»
4. В дату платежа по графику, в конце дня (23:59:59) регистрируется операция «Требование платежа»

В любой момент клиент может внести платеж, при этом регистрируется операция «Платеж».

В дистрибутиве вы найдете скрипт создания демонстрационных операций договора `operations.py`

В качестве параметра скрипту передается базовый адрес REST API сервиса.

```
python operations.py http://[ip сервера]:55002
```

## Описание Схем учета операций

### Выдача займа

При выдаче займа денежные средства перемещаются со счета «Основной долг» договора на счет «Касса» офиса выдачи займа.

Схема Операции «Выдача займа»:

```
{
  "dateFrom": "2010-01-01T00:00:00+03",
  "operationName": "Выдача займа",
  "parameters": [
    {"name": "organization", "type": "String"},
    {"name": "checkout", "type": "Number"},
    {"name": "contract", "type": "Number"},
    {"name": "amount", "type": "Number"},
    {"name": "description", "type": "String"}
  ],
  "rules": [
    {
      "number": 0,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat("http://ip:5000/organizations/?name=", organization), "id")),
toGUID(GetParameter(concat("http://ip:5000/contracts/?number=", contract), "id")), 1111111-0000-4a41-b0e9-111111111111,
"Основной долг"),
      "creditAccount": "getAccount(toGUID(GetParameter(concat("http://ip:5000/organizations/?name=", organization), "id")),
toGUID(GetParameter(concat("http://ip:5000/checkouts/?number=", checkout), "id")), 1111111-0000-4a41-b0e9-111111111112,
"Касса"),
      "amount": "amount",
      "description": "concat("Выдача займа. ", description)"
    }
  ]
}
```

Схема предусматривает создание одной проводки.

Регистрируем операцию со следующими параметрами:

**Дата операции:** 2021-11-22T12:13:01.0000000+0300

**Организация:** "Займы Всем"

**Касса:** №1

**Договор:** №12346

**Сумма:** 1000р.

**Комментарий:** Test

Запрос на регистрацию операции "Выдача займа":

```
POST /api/accounting-system/v1/operations
{
  "operationDate": "2021-11-22T12:13:01.0000000+0300",
  "operationName": "Выдача займа",
  "parameters": [
    {"name": "organization", "value": "Займы Всем"},
    {"name": "checkout", "value": "1"},
    {"name": "contract", "value": "12346"},
    {"name": "amount", "value": "1000"},
    {"name": "description", "value": "Test"}
  ]
}
```

### Обработка правил определения debitAccountId (Счета дебета)

```
getAccount(toGUID(GetParameter(concat("http://ip:5000/organizations/?name=", organization), "id")),
toGUID(GetParameter(concat("http://ip:5000/contracts/?number=", contract), "id")), 1111111-0000-4a41-b0e9-111111111111, "Основной
долг")
```

Порядок обработки правил следующий:

1. Формируется URL запроса во внешнюю систему содержащую информацию об организациях

```
concat("http://ip:5000/organizations/?name=", organization)
=> http://ip:5000/organizations/?name=З а й м ы %20 В с е м
```

2. Запрашиваем во внешней системе id(GUID) организации.

```
GET http://ip:5000/organizations/?name=З а й м ы %20 В с е м
```

Возвращает

```
{
  "id": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "name": "З а й м ы В с е м"
  ...
}
```

Тогда

```
GetParameter("http://ip:5000/organizations/?name=З а й м ы %20 В с е м", "id") => "eef268f7-33aa-4772-b37d-f86a8626603f"
```

3. toGUID конвертирует значение параметра в тип GUID

4. Формируем URL запроса во внешнюю систему содержащую информацию о договоре

```
concat("http://ip:5000/contracts/?number=", contract) => http://ip:5000/contracts/?number=12345
```

5. Запрашиваем во внешней системе id(GUID) договора.

6. toGUID конвертирует значение параметра в тип GUID

7. Запрашиваем Ид. Счета с полученными Ид.Организации, Ид.Договора, Типом объекта «11111111-0000-4a41-b0e9-111111111111» - «договор займа» и типом "Основной долг"

```
getAccount(eef268f7-33aa-4772-b37d-f86a8626603f, e068f6f5-3e46-4c21-9b91-0e48bedf60ee, 11111111-0000-4a41-b0e9-111111111111,
"Основной долг") => 97b405f7-2294-42f4-acd6-459197825ad2
```

Аналогично обрабатываются остальные поля

Ответ на регистрацию операции "Выдача займа":

```
{
  "documentId": null,
  "operationName": "В ы д а ч а з а й м а",
  "operationName": "В ы д а ч а з а й м а",
  "creationDate": "2022-12-22T11:52:35.7147830Z",
  "operationDate": "2021-11-22T09:13:01.0000000Z",
  "operationId": "54ba6223-baeb-461a-9bc6-947bb05fa42c",
  "stornoOperationId": null,
  "parameters": [
    {"name": "amount", "value": "1000"},
    {"name": "checkout", "value": "1"},
    {"name": "contract", "value": "12346"},
    {"name": "description", "value": "Test"},
    {"name": "organization", "value": "З а й м ы В с е м"}
  ],
  "entries": [
    {
      "entryId": "2f4ab95d-a47e-47a8-8af2-bf9b58aafe6b",
      "operationId": "54ba6223-baeb-461a-9bc6-947bb05fa42c",
      "creationDate": "2022-12-22T11:52:38.0866510Z",
      "accountingDate": "2021-11-22T09:13:01.0000000Z",
      "affectingDate": "2021-11-22T09:13:01.0000000Z",
      "debitAccountId": "598b2acb-fab8-4ae4-985d-9f976831dee9",
      "creditAccountId": "f93603db-cc86-4648-b2c4-c404f826209a",
      "amount": 1000,
      "description": "В ы д а ч а з а й м а. Test",
      "stornoEntryId": null
    }
  ]
}
```

В результате обработки у нас в системе зарегистрированы:

1. счет accountId:"598b2acb-fab8-4ae4-985d-9f976831dee9" типа «Основной долг» договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1"

```
GET /api/accounting-system/v1/accounts/97b405f7-2294-42f4-acd6-459197825ad2
```

```
{
  "accountId": "598b2acb-fab8-4ae4-985d-9f976831dee9",
  "creationDate": "2022-12-16T14:29:52.0916990Z",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "objectId": "a42a9998-53ac-44ed-aa73-9d9f26dd2ca1",
  "objectType": "11111111-0000-4a41-b0e9-111111111111",
  "apType": "активный",
  "accountTypeId": "af6b16b3-eca7-4c28-8c18-374e7978c40f",
  "accountTypeName": "Основной долг",
  "extParameters": [
    {"name": "EPS", "value": "-"},
    {"name": "officeName", "value": "Офис 1"},
    {"name": "product", "value": "Просто Займ"},
    {"name": "region", "value": "Москва"}
  ]
}
```

2. счет accountId:"f93603db-cc86-4648-b2c4-c404f826209a" типа «Касса» кассы №"1" id:"cd42b975-aa3e-4042-82f6-97edd97e1bc4"

```
GET /api/accounting-system/v1/accounts/f93603db-cc86-4648-b2c4-c404f826209a
```

```
{
  "accountId": "f93603db-cc86-4648-b2c4-c404f826209a",
  "creationDate": "2022-11-22T14:17:38.5498270Z",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "objectId": "cd42b975-aa3e-4042-82f6-97edd97e1bc4",
  "objectType": "11111111-0000-4a41-b0e9-111111111112",
  "apType": "активный",
  "accountTypeId": "cf613df5-c7ee-47a0-b2bf-1d75eacaed43",
  "accountTypeName": "Касса",
  "extParameters": [
    {"name": "EPS", "value": "20202810000000000001"},
    {"name": "officeName", "value": "Офис 1"},
    {"name": "region", "value": "Москва"}
  ]
}
```

3. проводка entryId:"2f4ab95d-a47e-47a8-8af2-bf9b58aaf6b" на сумму 1000 р. с датой учета "2021-11-22T09:13:01.0000000Z"

Остаток счета "598b2acb-fab8-4ae4-985d-9f976831dee9" (Основной долг) после операции:

Запрос остатка:

```
GET /api/accounting-system/v1/accounts/598b2acb-fab8-4ae4-985d-9f976831dee9/balance?date=2021-11-22T09%3A13%3A02.0000000Z
```

Ответ:

```
1000
```

# Расчёт процентов и пени

Ежедневно с момента выдачи займа в системе регистрируется операция «Расчёт процентов и пени»

Схема Операции «Расчёт процентов и пени»:

```
{
  "dateFrom": "2010-01-01T00:00:00+03",
  "operationName": "Расчёт процентов и пени",
  "parameters": [
    {"name": "organization", "type": "String"},
    {"name": "contract", "type": "Number"},
    {"name": "interest_rate", "type": "Number"},
    {"name": "penalty_rate", "type": "Number"},
    {"name": "description", "type": "String"}
  ],
  "rules": [
    {
      "number": 1,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Процент начисленный¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111, ¥"Доход
по процентам¥")",
      "amount":
"(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Основной долг¥"), toDate(operationDate)) +
getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111, ¥"Платеж
ОД¥"), toDate(operationDate))) * interest_rate / 100",
      "description": "concat(¥"Начисление процентов. ¥", description)"
    },
    {
      "number": 2,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111, ¥"Пени
начисленные¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111, ¥"Доход
по пени¥")",
      "amount":
"(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Просроченный платеж ОД¥"), toDate(operationDate))) * penalty_rate / 100",
      "description": "concat(¥"Начисление пени. ¥", description)"
    },
    {
      "number": 3,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Процент начисленный на просроченный ОД¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111, ¥"Доход
по процентам¥")",
      "amount":
"(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Просроченный платеж ОД¥"), toDate(operationDate))) * interest_rate / 100",
      "description": "concat(¥"Начисление процентов на просроченный ОД. ¥", description)"
    }
  ]
}
```

Схема предусматривает возможность создания трех проводок.

## 1. Начисление процентов

**Счет дебета:** «Процент начисленный» договора

**Счет кредита:** «Доход по процентам» договора

**Сумма проводки:** (Остаток счета «Основной долг» договора + Остаток счета «Платеж ОД» договора) \* «дневная % ставка»/100

## 2. Начисление пени

**Счет дебета:** «Пени начисленные» договора

**Счет кредита:** «Доход по пени» договора

**Сумма проводки:** (Остаток счета «Просроченный платеж ОД» договора) \* «дневная % ставка пени»/100

## 3. Начисление процентов на просроченный ОД

**Счет дебета:** «Процент начисленный на просроченный ОД» договора

**Счет кредита:** «Доход по процентам» договора

**Сумма проводки:** (Остаток счета «Просроченный платеж ОД» договора) \* «дневная % ставка»/100

Регистрируем операцию со следующими параметрами:

**Дата операции:** 2021-11- 22T12:13:02.0000000+0300

**Организация:** "Займы Всем"

**Договор:** №12346

**% ставка:** 1

**% ставка пени:** 0.5

**Комментарий:** Расчёт процентов и пени за 2021-11-22

Запрос на регистрацию операции "Расчёт процентов и пени":

```
POST /api/accounting-system/v1/operations
{
  "operationDate": "2021-11-22T12:13:02.0000000+0300",
  "operationName": "Расчёт процентов и пени",
  "parameters": [
    {"name": "organization", "value": "Займы Всем"},
    {"name": "contract", "value": "12346"},
    {"name": "interest_rate", "value": "1"},
    {"name": "penalty_rate", "value": "0.5"},
    {"name": "description", "value": "Расчёт процентов и пени за 2021-11-22"}
  ]
}
```

Ответ на регистрацию операции "Расчёт процентов и пени":

```
{
  "documentId": null,
  "operationName": "Расчёт процентов и пени",
  "creationDate": "2022-12-22T14:41:54.2901920Z",
  "operationDate": "2021-11-22T09:13:02.0000000Z",
  "operationId": "6cf95c22-4180-40c3-acd4-661647074195",
  "stornoOperationId": null,
  "parameters": [
    {"name": "contract", "value": "12346"},
    {"name": "description", "value": "Расчёт процентов и пени за 2021-11-22"},
    {"name": "interest_rate", "value": "1"},
    {"name": "organization", "value": "Займы Всем"},
    {"name": "penalty_rate", "value": "0.5"}
  ],
  "entries": [
    {
      "entryId": "52669e8b-a0bc-41a7-a838-0572830ff1e6",
      "operationId": "6cf95c22-4180-40c3-acd4-661647074195",
      "creationDate": "2022-12-22T14:41:54.6538770Z",
      "accountingDate": "2021-11-22T09:13:02.0000000Z",
      "affectingDate": "2021-11-22T09:13:02.0000000Z",
      "debitAccountId": "4d874db7-ccf9-4d4e-b9b5-e028ea5a9370",
      "creditAccountId": "f2254a9e-a7ca-41f3-a845-b8c12ab7820a",
      "amount": 10,
      "description": "Начисление процентов. Расчёт процентов и пени за 2021-11-22",
    }
  ]
}
```

```
    "stornoEntryId": null
  }
}
}
```

Т.к. при расчете суммы проводки, для проводок №2 и №3, получилось значение 0 проводки не были созданы.

В результате обработки у нас в системе зарегистрированы:

1. счет accountId:"4d874db7-ccf9-4d4e-b9b5-e028ea5a9370" типа «Процент начисленный» договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1"

```
GET /api/accounting-system/v1/accounts/4d874db7-ccf9-4d4e-b9b5-e028ea5a9370
{
  "accountId": "4d874db7-ccf9-4d4e-b9b5-e028ea5a9370",
  "creationDate": "2022-12-21T11:15:44.3174930Z",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "objectId": "a42a9998-53ac-44ed-aa73-9d9f26dd2ca1",
  "objectType": "11111111-0000-4a41-b0e9-111111111111",
  "apType": "активный",
  "accountTypeId": "679324ab-fc75-4178-80ff-5009f5e802f3",
  "accountTypeName": "Процент начисленный",
  "extParameters": [
    {"name": "EPS", "value": "-"},
    {"name": "officeName", "value": "Офис 2"},
    {"name": "product", "value": "Просто Займ"},
    {"name": "region", "value": "Казань"}
  ]
}
```

2. счет accountId:"f2254a9e-a7ca-41f3-a845-b8c12ab7820a" типа «Доход по процентам» договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1"

```
GET /api/accounting-system/v1/accounts/f2254a9e-a7ca-41f3-a845-b8c12ab7820a
{
  "accountId": "f2254a9e-a7ca-41f3-a845-b8c12ab7820a",
  "creationDate": "2022-12-21T11:15:44.5792410Z",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "objectId": "a42a9998-53ac-44ed-aa73-9d9f26dd2ca1",
  "objectType": "11111111-0000-4a41-b0e9-111111111111",
  "apType": "пассивный",
  "accountTypeId": "d7e9df22-80fd-4145-b7fa-e593dc507256",
  "accountTypeName": "Доход по процентам",
  "extParameters": [
    {"name": "EPS", "value": "7060181000000012346"},
    {"name": "officeName", "value": "Офис 2"},
    {"name": "product", "value": "Просто Займ"},
    {"name": "region", "value": "Казань"}
  ]
}
```

3. проводка entryId:"31dcd578-c1b4-47d7-b3c7-ff77747fe2f6" на сумму 10 р. с датой учета "2021-11-22T09:13:01.0000000Z"

Остаток счета "4d874db7-ccf9-4d4e-b9b5-e028ea5a9370" («Процент начисленный») после операции: 10

Остаток счета "f2254a9e-a7ca-41f3-a845-b8c12ab7820a" («Доход по процентам») после операции: 10

## Платеж по графику

В день платежа по графику после начисления процентов в системе регистрируется операция «Платеж по графику»

Схема Операции «Платеж по графику»:

```
{
  "dateFrom": "2010-01-01T00:00:00+03",
  "operationName": "П л а т е ж  п о  г р а ф и к у",
  "parameters": [
    {"name": "organization", "type": "String"},
    {"name": "contract", "type": "Number"},
    {"name": "prc_amount", "type": "Number"},
    {"name": "body_amount", "type": "Number"},
    {"name": "description", "type": "String"}
  ],
  "rules": [
    {
      "number": 1,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111, ¥"П л а т е ж
П р о ц е н т о в ¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"П р о ц е н т  н а ч и с л е н н ы й ¥")",
      "amount": "max(prc_amount, 0)",
      "description": "concat(¥"П л а т е ж  п р о ц е н т о в  п о  г р а ф и к у.  ¥", description)"
    },
    {
      "number": 2,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111, ¥"П л а т е ж
О Д ¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"О с н о в н о й  д о л г ¥")",
      "amount": "max(body_amount, 0)",
      "description": "concat(¥"П л а т е ж  О Д  п о  г р а ф и к у.  ¥", description)"
    }
  ]
}
```

Схема предусматривает возможность создания двух проводок.

### 1. Платеж процентов по графику

**Счет дебета:** «Платеж Процентов» договора

**Счет кредита:** «Процент начисленный» договора

**Сумма проводки:** Сумма платежа % по графику

### 2. Платеж ОД по графику

**Счет дебета:** «Платеж ОД» договора

**Счет кредита:** «Основной долг» договора

**Сумма проводки:** Сумма платежа ОД по графику

Регистрируем операцию со следующими параметрами:

**Дата операции:** 2021-12-05T00:00:01.0000000+0300

**Организация:** "Займы Всем"

**Договор:** №12346

**Сумма платежа % по графику:** 140

**Сумма платежа ОД по графику: 151.29**

**Комментарий: Платеж по графику от 2021-12-05**

Запрос на регистрацию операции "Платеж по графику":

```
POST /api/accounting-system/v1/operations
{
  "operationDate": "2021-12-05T00:00:01.0000000+0300",
  "operationName": "Платеж по графику",
  "parameters": [
    {"name": "organization", "value": "Займы Всем"},
    {"name": "contract", "value": "12346"},
    {"name": "prc_amount", "value": "140"},
    {"name": "body_amount", "value": "151.29"},
    {"name": "description", "value": "Платеж по графику от 2021-12-05"}
  ]
}
```

Ответ на регистрацию операции "Платеж по графику":

```
{
  "documentId": null,
  "operationName": "Платеж по графику",
  "creationDate": "2022-12-22T14:42:04.3446330Z",
  "operationDate": "2021-12-04T21:00:01.0000000Z",
  "operationId": "2fa98a0c-0cf0-4ee2-9cf3-fa54ed5af060",
  "stornoOperationId": null,
  "parameters": [
    {"name": "body_amount", "value": "151.29"},
    {"name": "contract", "value": "12346"},
    {"name": "description", "value": "Платеж по графику от 2021-12-05"},
    {"name": "organization", "value": "Займы Всем"},
    {"name": "prc_amount", "value": "140"}
  ],
  "entries": [
    {
      "entryId": "a0ee8ea0-b653-4a1d-8863-a20ae219864e",
      "operationId": "2fa98a0c-0cf0-4ee2-9cf3-fa54ed5af060",
      "creationDate": "2022-12-22T14:42:05.5752170Z",
      "accountingDate": "2021-12-04T21:00:01.0000000Z",
      "affectingDate": "2021-12-04T21:00:01.0000000Z",
      "debitAccountId": "f8061d55-e6bc-43bd-b6e6-c34690b28a4e",
      "creditAccountId": "4d874db7-ccf9-4d4e-b9b5-e028ea5a9370",
      "amount": 140,
      "description": "Платеж процентов по графику. Платеж по графику от 2021-12-05",
      "stornoEntryId": null
    },
    {
      "entryId": "27216042-4c88-4c47-9257-1b44ca9925eb",
      "operationId": "2fa98a0c-0cf0-4ee2-9cf3-fa54ed5af060",
      "creationDate": "2022-12-22T14:42:05.5754550Z",
      "accountingDate": "2021-12-04T21:00:01.0000000Z",
      "affectingDate": "2021-12-04T21:00:01.0000000Z",
      "debitAccountId": "473729da-0a87-4cc2-a2ce-1ce5196b01f8",
      "creditAccountId": "598b2acb-fab8-4ae4-985d-9f976831dee9",
      "amount": 151.29,
      "description": "Платеж ОД по графику. Платеж по графику от 2021-12-05",
      "stornoEntryId": null
    }
  ]
}
```

В результате обработки у нас в системе зарегистрированы:

1. счет accountId:"f8061d55-e6bc-43bd-b6e6-c34690b28a4e" типа «Платеж Процентом» договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1"

```
GET /api/accounting-system/v1/accounts/4d874db7-ccf9-4d4e-b9b5-e028ea5a9370
"accountId": "f8061d55-e6bc-43bd-b6e6-c34690b28a4e",
"creationDate": "2022-12-22T14:42:04.9423550Z",
"organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
"objectId": "a42a9998-53ac-44ed-aa73-9d9f26dd2ca1",
"objectType": "11111111-0000-4a41-b0e9-111111111111",
"apType": "активный",
"accountTypeId": "78e52175-816b-46a8-a63a-33594659113f",
```

```
"accountTypeName": "Платеж Процентов",  
"extParameters": [  
  {"name": "EPS", "value": "-"},  
  {"name": "officeName", "value": "Офис 2"},  
  {"name": "product", "value": "Просто Займ"},  
  {"name": "region", "value": "Казань"}  
]  
}
```

2. счет accountId:"473729da-0a87-4cc2-a2ce-1ce5196b01f8" типа «Платеж ОД» договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1"

```
GET /api/accounting-system/v1/accounts/f2254a9e-a7ca-41f3-a845-b8c12ab7820a
```

```
{
  "accountId": "473729da-0a87-4cc2-a2ce-1ce5196b01f8",
  "creationDate": "2022-12-21T11:15:43.6134900Z",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "objectId": "a42a9998-53ac-44ed-aa73-9d9f26dd2ca1",
  "objectType": "11111111-0000-4a41-b0e9-111111111111",
  "apType": "активный",
  "accountTypeId": "7870b699-3f31-46de-8857-ad9e6a6b4aeb",
  "accountTypeName": "Платеж ОД",
  "extParameters": [
    {"name": "EPS", "value": "-"},
    {"name": "officeName", "value": "Офис 2"},
    {"name": "product", "value": "Просто Займ"},
    {"name": "region", "value": "Казань"}
  ]
}
```

3. проводка entryId:"31dcd578-c1b4-47d7-b3c7-ff77747fe2f6" на сумму 10 р. с датой учета "2021-11-22T09:13:01.0000000Z"

Остатки по счетам договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1" до операции были:

```
GET /api/accounting-system/v1/objects/a42a9998-53ac-44ed-aa73-9d9f26dd2ca1/balances?date=2021-12-05T00%3A00%3A01.0000000%2B0300
```

Ответ:

```
[
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Процент начисленный",
    "balance": 140
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Основной долг",
    "balance": 1000
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Доход по процентам",
    "balance": 140
  }
]
```

Остатки по счетам договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1" после операции стали:

GET /api/accounting-system/v1/objects/a42a9998-53ac-44ed-aa73-9d9f26dd2ca1/balances?date=2021-12-05T00%3A00%3A02.0000000%2B0300

Ответ:

```
[
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "П л а т е ж О Д",
    "balance": 151.29
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "П р о ц е н т н а ч и с л е н н ы й",
    "balance": 0
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "О с н о в н о й д о л г",
    "balance": 848.71
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Д о х о д п о п р о ц е н т а м",
    "balance": 140
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "П л а т е ж П р о ц е н т о в",
    "balance": 140
  }
]
```

## Требование платежа

В день платежа по графику в конце дня в системе регистрируется операция «Требование платежа»

Схема Операции «Требование платежа»:

```
{
  "dateFrom": "2010-01-01T00:00:00+03",
  "operationName": "Требование платежа",
  "parameters": [
    {"name": "organization", "type": "String"},
    {"name": "contract", "type": "Number"},
    {"name": "description", "type": "String"}
  ],
  "rules": [
    {
      "number": 1,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Процент уплаченный¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111, ¥"Платеж
Процентом¥")",
      "amount":
      "min(max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Предоплата¥"), toDate(operationDate)), 0),
max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111, ¥"Платеж
Процентом¥"), toDate(operationDate)), 0))",
      "description": "concat(¥"Требование (Уплата процентов) (3). ¥", description)"
    },
    {
      "number": 2,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111, ¥"ОД
Уплаченный¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111, ¥"Платеж
ОД¥")",
      "amount":
      "min(max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Предоплата¥"), toDate(operationDate)), 0) - RuleAmount1,
max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111, ¥"Платеж
ОД¥"), toDate(operationDate)), 0))",
      "description": "concat(¥"Требование (Уплата ОД) (4). ¥", description)"
    },
    {
      "number": 3,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111, ¥"Пени
уплаченные¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111, ¥"Пени
начисленные¥")",
      "amount":
      "min(max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Предоплата¥"), toDate(operationDate)), 0) - RuleAmount1 - RuleAmount2,
max(floor(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111, ¥"Пени
начисленные¥"), toDate(operationDate)), 2), 0))",
      "description": "concat(¥"Требование (Уплата пени) (5). ¥", description)"
    },
    {
      "number": 4,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Просроченный платеж Процентом¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111, ¥"Платеж
```

```

    "amount":
    "max (getBalanceByAffectingDate (getAccount (toGUID (GetParameter (concat ("http://ip:5000/organizations/?name=%", organization), %id%))),
    toGUID (GetParameter (concat ("http://ip:5000/contracts/?number=%", contract), %id%))), 11111111-0000-4a41-b0e9-111111111111, %Платеж
    Процентом%), toDate (operationDate)) - RuleAmount1, 0)",
    "description": "concat (%Требование Платежа Процентом. %, description)"
  },
  {
    "number": 5,
    "entryDate": "toDate (operationDate)",
    "debitAccount": "getAccount (toGUID (GetParameter (concat ("http://ip:5000/organizations/?ame=%", organization), %id%))),
    toGUID (GetParameter (concat ("http://ip:5000/contracts/?number=%", contract), %id%))), 11111111-0000-4a41-b0e9-111111111111,
    %Просроченный платеж ОД%),
    "creditAccount": "getAccount (toGUID (GetParameter (concat ("http://ip:5000/organizations/?name=%", organization), %id%))),
    toGUID (GetParameter (concat ("http://ip:5000/contracts/?number=%", contract), %id%))), 11111111-0000-4a41-b0e9-111111111111, %Платеж
    ОД%)",
    "amount":
    "max (getBalanceByAffectingDate (getAccount (toGUID (GetParameter (concat ("http://ip:5000/organizations/?name=%", organization), %id%))),
    toGUID (GetParameter (concat ("http://ip:5000/contracts/?number=%", contract), %id%))), 11111111-0000-4a41-b0e9-111111111111, %Платеж
    ОД%), toDate (operationDate)) - RuleAmount2, 0)",
    "description": "concat (%Требование Платежа ОД. %, description)"
  },
  {
    "number": 6,
    "entryDate": "toDate (operationDate)",
    "debitAccount": "getAccount (toGUID (GetParameter (concat ("http://ip:5000/organizations/?name=%", organization), %id%))),
    toGUID (GetParameter (concat ("http://ip:5000/contracts/?number=%", contract), %id%))), 11111111-0000-4a41-b0e9-111111111111,
    %Предоплата%),
    "creditAccount": "getAccount (toGUID (GetParameter (concat ("http://ip:5000/organizations/?name=%", organization), %id%))),
    toGUID (GetParameter (concat ("http://ip:5000/contracts/?number=%", contract), %id%))), 11111111-0000-4a41-b0e9-111111111111,
    %Процент уплаченный)",
    "amount": "RuleAmount1",
    "description": "concat (%Требование (Уплата процентов) (3). %, description)"
  },
  {
    "number": 7,
    "entryDate": "toDate (operationDate)",
    "debitAccount": "getAccount (toGUID (GetParameter (concat ("http://ip:5000/organizations/?name=%", organization), %id%))),
    toGUID (GetParameter (concat ("http://ip:5000/contracts/?number=%", contract), %id%))), 11111111-0000-4a41-b0e9-111111111111,
    %Предоплата%),
    "creditAccount": "getAccount (toGUID (GetParameter (concat ("http://ip:5000/organizations/?name=%", organization), %id%))),
    toGUID (GetParameter (concat ("http://ip:5000/contracts/?number=%", contract), %id%))), 11111111-0000-4a41-b0e9-111111111111, %ОД
    Уплаченный)",
    "amount": "RuleAmount2",
    "description": "concat (%Требование (Уплата ОД) (4). %, description)"
  },
  {
    "number": 8,
    "entryDate": "toDate (operationDate)",
    "debitAccount": "getAccount (toGUID (GetParameter (concat ("http://ip:5000/organizations/?name=%", organization), %id%))),
    toGUID (GetParameter (concat ("http://ip:5000/contracts/?number=%", contract), %id%))), 11111111-0000-4a41-b0e9-111111111111,
    %Предоплата%),
    "creditAccount": "getAccount (toGUID (GetParameter (concat ("http://ip:5000/organizations/?name=%", organization), %id%))),
    toGUID (GetParameter (concat ("http://ip:5000/contracts/?number=%", contract), %id%))), 11111111-0000-4a41-b0e9-111111111111, %Пени
    уплаченные)",
    "amount": "RuleAmount3",
    "description": "concat (%Требование (Уплата пени) (5). %, description)"
  }
]
}

```

Схема предусматривает возможность создания 8 проводок.

1. Уплата процентов с транзитного счета

**Счет дебета:** «Процент уплаченный» договора

**Счет кредита:** «Платеж Процентом» договора

**Сумма проводки:** Остаток счета «Платеж Процентом» в пределах остатка счета

«Предоплата»

2. Уплата ОД с транзитного счета

**Счет дебета:** «ОД Уплаченный» договора

**Счет кредита:** «Платеж ОД» договора

**Сумма проводки:** Остаток счета «Платеж ОД» в пределах остатка счета «Предоплата», уменьшенного на Сумму проводки №1

3. Уплата пени с транзитного счета

**Счет дебета:** «Пени уплаченные» договора

**Счет кредита:** «Пени начисленные» договора

**Сумма проводки:** Остаток счета «Пени начисленные» в пределах остатка счета «Предоплата», уменьшенного на Сумму проводки №1 и 2

4. Просрочен Платежа Процентов

**Счет дебета:** «Просроченный платеж Процентов» договора

**Счет кредита:** «Платеж Процентов» договора

**Сумма проводки:** Остаток счета «Платеж Процентов», уменьшенный на сумму проводки №1

5. Просрочен Платеж ОД

**Счет дебета:** «Просроченный платеж ОД» договора

**Счет кредита:** «Платеж ОД» договора

**Сумма проводки:** Остаток счета «Платеж ОД», уменьшенный на сумму проводки №2

6. Уплата процентов на транзитный счет

**Счет дебета:** «Предоплата» договора

**Счет кредита:** «Процент уплаченный» договора

**Сумма проводки:** Сумма проводки №1

7. Уплата ОД на транзитный счет

**Счет дебета:** «Предоплата» договора

**Счет кредита:** «ОД Уплаченный» договора

**Сумма проводки:** Сумма проводки №2

8. Уплата пени на транзитный счет

**Счет дебета:** «Предоплата» договора

**Счет кредита:** «Пени уплаченные» договора

**Сумма проводки:** Сумма проводки №3

Регистрируем операцию со следующими параметрами:

Дата операции: 2021-12-05T23:59:59.0000000+0300

Организация: "Займы Всем"

Договор: №12346

Комментарий: Платеж по графику от 2021-12-05

Запрос на регистрацию операции "Требование платежа":

```
POST /api/accounting-system/v1/operations
```

```
{
```

```
  "operationDate": "2021-12-05T23:59:59.0000000+0300",
```

```
  "operationName": "Платеж по графику",
```

```
"parameters": [  
  {"name": "organization", "value": "Займы Всем"},  
  {"name": "contract", "value": "12346"},  
  {"name": "description", "value": "Платеж по графику от 2021-12-05"}  
]
```

## Ответ на регистрацию операции "Требование платежа":

```
{
  "documentId": null,
  "operationName": "Требование платежа",
  "creationDate": "2022-12-22T14:42:06.0853670Z",
  "operationDate": "2021-12-05T20:59:59.000000Z",
  "operationId": "30fba379-6a31-4eee-a39d-0d30b8cdb320",
  "stornoOperationId": null,
  "parameters": [
    {"name": "contract", "value": "12346"},
    {"name": "description", "value": "Платеж по графику от 2021-12-05"},
    {"name": "organization", "value": "Займы Всем"}
  ],
  "entries": [
    {
      "entryId": "77dd112c-2bad-4a99-940d-e5a0b134a1c4",
      "operationId": "30fba379-6a31-4eee-a39d-0d30b8cdb320",
      "creationDate": "2022-12-22T14:42:08.4307320Z",
      "accountingDate": "2021-12-05T20:59:59.000000Z",
      "affectingDate": "2021-12-05T20:59:59.000000Z",
      "debitAccountId": "a29eea24-71f3-49a0-9a35-ab0f34da19dd",
      "creditAccountId": "f8061d55-e6bc-43bd-b6e6-c34690b28a4e",
      "amount": 140,
      "description": "Требование Платежа Процентов. Платеж по графику от 2021-12-05",
      "stornoEntryId": null
    },
    {
      "entryId": "435375f4-abea-4e5e-a5c2-1252acb2c822",
      "operationId": "30fba379-6a31-4eee-a39d-0d30b8cdb320",
      "creationDate": "2022-12-22T14:42:08.4309810Z",
      "accountingDate": "2021-12-05T20:59:59.000000Z",
      "affectingDate": "2021-12-05T20:59:59.000000Z",
      "debitAccountId": "7f4f2568-71c3-439d-aea0-7e89da75b9f2",
      "creditAccountId": "473729da-0a87-4cc2-a2ce-1ce5196b01f8",
      "amount": 151.29,
      "description": "Требование Платежа Од. Платеж по графику от 2021-12-05",
      "stornoEntryId": null
    }
  ]
}
```

В результате обработки у нас в системе зарегистрированы:

1. счет accountId:"a29eea24-71f3-49a0-9a35-ab0f34da19dd" типа «Просроченный платеж Процентов» договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1"

```
GET /api/accounting-system/v1/accounts/a29eea24-71f3-49a0-9a35-ab0f34da19dd
{
  "accountId": "a29eea24-71f3-49a0-9a35-ab0f34da19dd",
  "creationDate": "2022-12-22T14:42:07.7704480Z",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "objectId": "a42a9998-53ac-44ed-aa73-9d9f26dd2ca1",
  "objectType": "11111111-0000-4a41-b0e9-111111111111",
  "apType": "активный",
  "accountTypeId": "4018c7fb-4faa-426d-8413-c4b6499e768a",
  "accountTypeName": "Просроченный платеж Процентов",
  "extParameters": [
    {"name": "EPS", "value": "-"},
    {"name": "officeName", "value": "Офис 2"},
    {"name": "product", "value": "Просто Займы"},
    {"name": "region", "value": "Казань"}
  ]
}
```

2. счет accountId:"7f4f2568-71c3-439d-aea0-7e89da75b9f2" типа «Просроченный платеж Од» договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1"

```
GET /api/accounting-system/v1/accounts/7f4f2568-71c3-439d-aea0-7e89da75b9f2
{
  "accountId": "7f4f2568-71c3-439d-aea0-7e89da75b9f2",
  "creationDate": "2022-12-21T11:15:43.7506830Z",
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "objectId": "a42a9998-53ac-44ed-aa73-9d9f26dd2ca1",
  "objectType": "11111111-0000-4a41-b0e9-111111111111",
  "apType": "активный",

```

```
"accountTypeId": "ea24e610-3caa-48cc-aeb2-0f2c0464eef7",
"accountTypeName": "Просроченный платеж ОД",
"extParameters": [
  {"name": "EPS", "value": "-"},
  {"name": "officeName", "value": "О ф и с 2"},
  {"name": "product", "value": "П р о с т о З а й м"},
  {"name": "region", "value": "К а з а н ь"}
]
}
```

3. проводка entryId:"77dd112c-2bad-4a99-940d-e5a0b134a1c4" на сумму 140 р.

4. проводка entryId:"435375f4-abea-4e5e-a5c2-1252acb2c822" на сумму 151.29 р.

Остатки по счетам договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1" до операции были:

```
GET /api/accounting-system/v1/objects/a42a9998-53ac-44ed-aa73-9d9f26dd2ca1/balances?date=2021-12-05T23%3A59%3A59.0000000%2B0300
```

Ответ:

```
[
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "П л а т е ж О Д",
    "balance": 151.29
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "П р о ц е н т н а ч и с л е н н ы й",
    "balance": 0
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "О с н о в н о й д о л г",
    "balance": 848.71
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Д о х о д п о п р о ц е н т а м",
    "balance": 140
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "П л а т е ж П р о ц е н т о в",
    "balance": 140
  }
]
```

Остатки по счетам договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1" после операции стали:

```
GET /api/accounting-system/v1/objects/a42a9998-53ac-44ed-aa73-9d9f26dd2ca1/balances?date=2021-12-06T00%3A00%3A00.0000000%2B0300
```

Ответ:

```
[
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "П л а т е ж О Д",
    "balance": 0
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "П р о ц е н т н а ч и с л е н н ы й",
    "balance": 0
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "О с н о в н о й д о л г",
    "balance": 848.71
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "П р о с р о ч е н н ы й п л а т е ж О Д",
    "balance": 151.29
  },
]
```

```
{
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "accountTypeName": "Просроченный платеж Процентом",
  "balance": 140
},
{
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "accountTypeName": "Доход по процентам",
  "balance": 140
},
{
  "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
  "accountTypeName": "Платеж Процентом",
  "balance": 0
}
]
```

## Платеж

При получении оплаты в системе регистрируется операция «Платеж». Сумма платежа распределяется согласно остаткам по счетам «Просроченный платеж Процентов», «Просроченный платеж ОД», «Платеж Процентов», «Платеж ОД»

Если после распределения в платеже осталась сумма, она относится на счет «Предоплата»

Схема Операции «Платеж»:

```
{
  "dateFrom": "2010-01-01T00:00:00+03",
  "operationName": "П л а т е ж",
  "parameters": [
    {"name": "organization", "type": "String"},
    {"name": "checkout", "type": "Number"},
    {"name": "contract", "type": "Number"},
    {"name": "amount", "type": "Number"},
    {"name": "description", "type": "String" }
  ],
  "rules": [
    {
      "number": 1,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Процент уплаченный¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Просроченный платеж Процентов¥")",
      "amount": "min(amount +
max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Предоплата¥"), toDate(operationDate)), 0),
max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Просроченный платеж Процентов¥"), toDate(operationDate)), 0))",
      "description": "concat(¥"Платеж (Уплата просроченных процентов) (1). ¥", description)"
    },
    {
      "number": 2,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"ОД
Уплаченный¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Просроченный платеж ОД¥")",
      "amount": "min(amount +
max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Предоплата¥"), toDate(operationDate)), 0) - RuleAmount1,
max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Просроченный платеж ОД¥"), toDate(operationDate)), 0))",
      "description": "concat(¥"Платеж (Уплата просроченного ОД) (2). ¥", description)"
    },
    {
      "number": 3,
      "entryDate": "toDate(operationDate)",
      "debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Процент
уплаченный¥")",
      "creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Процент
начисленный на просроченный ОД¥")",
      "amount": "min(amount +
max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Предоплата¥"), toDate(operationDate)), 0) - RuleAmount1 - RuleAmount2,
max(floor(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 11111111-0000-4a41-b0e9-111111111111,
¥"Процент
начисленный на просроченный ОД¥"), toDate(operationDate)), 2), 0))",
      "description": "concat(¥"Платеж (Уплата процентов на просроченный ОД) (3). ¥", description)"
    }
  ],
}
```



```

max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Предоплата¥"), toDate(operationDate)) - RuleAmount7, 0)),
"description": "concat(¥"Платеж (Уплата процентов) (8). ¥", description)"
},
{
"number": 9,
"entryDate": "toDate(operationDate)",
"debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Предоплата¥"),
"creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111, ¥"ОД
Уплаченный¥)",
"amount": "min(RuleAmount2 + RuleAmount5,
max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Предоплата¥"), toDate(operationDate)) - RuleAmount7 - RuleAmount8, 0))",
"description": "concat(¥"Платеж (Уплата ОД) (9). ¥", description)"
},
{
"number": 10,
"entryDate": "toDate(operationDate)",
"debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Предоплата¥"),
"creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111, ¥"Пени
уплаченный¥)",
"amount": "min(RuleAmount6,
max(getBalanceByAffectingDate(getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Предоплата¥"), toDate(operationDate)) - RuleAmount7 - RuleAmount8 - RuleAmount9, 0))",
"description": "concat(¥"Платеж (Уплата пени) (10). ¥", description)"
},
{
"number": 11,
"entryDate": "toDate(operationDate)",
"debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/checkouts/?number=¥", checkout), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Касса¥"),
"creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Процентуплаченный¥)",
"amount": "RuleAmount1 + RuleAmount3 + RuleAmount4 - RuleAmount8",
"description": "concat(¥"Платеж (Уплата процентов) (11). ¥", description)"
},
{
"number": 12,
"entryDate": "toDate(operationDate)",
"debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/checkouts/?number=¥", checkout), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Касса¥"),
"creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111, ¥"ОД
Уплаченный¥)",
"amount": "RuleAmount2 + RuleAmount5 - RuleAmount9",
"description": "concat(¥"Платеж (Уплата ОД) (12). ¥", description)"
},
{
"number": 13,
"entryDate": "toDate(operationDate)",
"debitAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/checkouts/?number=¥", checkout), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111,
¥"Касса¥"),
"creditAccount": "getAccount(toGUID(GetParameter(concat(¥"http://ip:5000/organizations/?name=¥", organization), ¥"id¥")),
toGUID(GetParameter(concat(¥"http://ip:5000/contracts/?number=¥", contract), ¥"id¥")), 1111111-0000-4a41-b0e9-111111111111, ¥"Пени
уплаченный¥)",
"amount": "RuleAmount6 - RuleAmount10",
"description": "concat(¥"Платеж (Уплата пени) (13). ¥", description)"
}
]
}

```

Схема предусматривает возможность создания 13 проводок.

## 1. Уплата просроченных процентов с транзитного счета

**Счет дебета:** «Процент уплаченный» договора

**Счет кредита:** «Просроченный платеж Процентов» договора

**Сумма проводки:** Остаток счета «Просроченный платеж Процентов» в пределах Суммы платежа и Остатка счета «Предоплата»

2. Уплата просроченного ОД с транзитного счета

**Счет дебета:** «ОД Уплаченный» договора

**Счет кредита:** «Просроченный платеж ОД» договора

**Сумма проводки:** Остаток счета «Просроченный платеж ОД» в пределах Суммы платежа и Остатка счета «Предоплата», уменьшенная на Сумму проводки №1

3. Уплата процентов на просроченный ОД с транзитного счета

**Счет дебета:** «Процент уплаченный» договора

**Счет кредита:** «Процент начисленный на просроченный ОД» договора

**Сумма проводки:** Остаток счета «Процент начисленный на просроченный ОД» в пределах Суммы платежа и Остатка счета «Предоплата» уменьшенная на Сумму проводок №1, №2

4. Уплата процентов с транзитного счета

**Счет дебета:** «Процент уплаченный» договора

**Счет кредита:** «Платеж Процентов» договора

**Сумма проводки:** Остаток счета «Платеж Процентов» в пределах Суммы платежа и Остатка счета «Предоплата», уменьшенная на Сумму проводок №1, №2, №3

5. Уплата ОД с транзитного счета

**Счет дебета:** «ОД Уплаченный» договора

**Счет кредита:** «Платеж ОД» договора

**Сумма проводки:** Остаток счета «Платеж ОД» в пределах Суммы платежа и Остатка счета «Предоплата», уменьшенная на Сумму проводок №1, №2, №3, №4

6. Уплата пени с транзитного счета

**Счет дебета:** «Пени уплаченные» договора

**Счет кредита:** «Пени начисленные» договора

**Сумма проводки:** Остаток счета «Пени начисленные» в пределах Суммы платежа и Остатка счета «Предоплата» уменьшенная на Сумму проводок №1, №2, №3, №4, №5

7. Образование Предоплаты

**Счет дебета:** «Касса» договора

**Счет кредита:** «Предоплата» договора

**Сумма проводки:** Сумма платежа и Остатка счета «Предоплата» уменьшенная на Сумму проводок №1, №2, №3, №4, №5, №6

8. Уплата процентов на транзитный счет за счет Предоплаты

**Счет дебета:** «Предоплата» договора

**Счет кредита:** «Процент уплаченный» договора

**Сумма проводки:** Остаток счета «Предоплата» уменьшенный на Сумма проводки №7 в пределах Суммы проводок №1, №3 и №4

9. Уплата ОД на транзитный счет за счет Предоплаты

**Счет дебета:** «Предоплата» договора

**Счет кредита:** «ОД Уплаченный» договора

**Сумма проводки:** Остаток счета «Предоплата» уменьшенный на Сумма проводки №7 и №8 в пределах Суммы проводок №2 и №5

10. Уплата пени на транзитный счет за счет Предоплаты

**Счет дебета:** «Предоплата» договора

**Счет кредита:** «Пени уплаченные» договора

**Сумма проводки:** Остаток счета «Предоплата» уменьшенный на Сумма проводки №7, №8 и №9 в пределах Суммы проводки №6

11. Уплата процентов на транзитный счет за счет Платежа

**Счет дебета:** «Касса»

**Счет кредита:** «Процент уплаченный» договора

**Сумма проводки:** Сумма проводок №1, №3 и 4 уменьшенная на Сумму проводки №8

12. Уплата ОД на транзитный счет за счет Платежа

**Счет дебета:** «Касса»

**Счет кредита:** «ОД Уплаченный» договора

**Сумма проводки:** Сумма проводок №2 и №5 уменьшенная на Сумму проводки №9

13. Уплата пени на транзитный счет за счет Платежа

**Счет дебета:** «Касса»

**Счет кредита:** «Пени уплаченные» договора

**Сумма проводки:** Сумма проводки №6, уменьшенная на Сумму проводки №10

Регистрируем операцию со следующими параметрами:

**Дата операции:** 2021-12-06T14:54:00.0000000+0300

**Организация:** "Займы Всем"

**Касса:** №1

**Договор:** №12346

**Сумма:** 700р

**Комментарий:** Прием платежа 2021-12-07

Запрос на регистрацию операции "Платеж":

```
POST /api/accounting-system/v1/operations
{
  "operationDate": "2021-12-06T14:54:00.0000000+0300",
  "operationName": "П л а т е ж",
  "parameters": [
    {"name": "organization", "value": "З а й м ы  В с е м"},
    {"name": "checkout", "value": "1"},
    {"name": "contract", "value": "12346"},
    {"name": "amount", "value": "700"},
    {"name": "description", "value": "П р и е м  п л а т е ж а  2021-12-07" }
  ]
}
```

## Ответ на регистрацию операции "Платеж":

```
{
  "documentId": null,
  "operationName": "П л а т е ж",
  "creationDate": "2022-12-22T14:42:10.7151460Z",
  "operationDate": "2021-12-06T11:54:00.0000000Z",
  "operationId": "f4d804af-a54b-4a0e-88c4-05f471eab474",
  "stornoOperationId": null,
  "parameters": [
    {"name": "amount", "value": "700"},
    {"name": "checkout", "value": "1"},
    {"name": "contract", "value": "12346"},
    {"name": "description", "value": "П р и е м п л а т е ж а 2021-12-07"},
    {"name": "organization", "value": "З а й м ы В с е м"}
  ],
  "entries": [
    {
      "entryId": "8293d132-be89-471a-9378-6ec54dbef6a0",
      "operationId": "f4d804af-a54b-4a0e-88c4-05f471eab474",
      "creationDate": "2022-12-22T14:42:11.9758480Z",
      "accountingDate": "2021-12-06T11:54:00.0000000Z",
      "affectingDate": "2021-12-06T11:54:00.0000000Z",
      "debitAccountId": "c1c52eef-1356-4a51-a7f0-7b08c7bc6ff6",
      "creditAccountId": "a29eea24-71f3-49a0-9a35-ab0f34da19dd",
      "amount": 140,
      "description": "П л а т е ж (У п л а т а п р о с р о ч е н н ы х п р о ц е н т о в) (1). П р и е м п л а т е ж а 2021-12-07",
      "stornoEntryId": null
    },
    {
      "entryId": "7f202d6c-df8b-4ba8-bbcc-9b8e859c8445",
      "operationId": "f4d804af-a54b-4a0e-88c4-05f471eab474",
      "creationDate": "2022-12-22T14:42:11.9760860Z",
      "accountingDate": "2021-12-06T11:54:00.0000000Z",
      "affectingDate": "2021-12-06T11:54:00.0000000Z",
      "debitAccountId": "0bef5f11-836c-45a7-b3c8-d8068741311f",
      "creditAccountId": "7f4f2568-71c3-439d-aea0-7e89da75b9f2",
      "amount": 151.29,
      "description": "П л а т е ж (У п л а т а п р о с р о ч е н н о г о О Д) (2). П р и е м п л а т е ж а 2021-12-07",
      "stornoEntryId": null
    },
    {
      "entryId": "ae44acea-a265-4996-b49c-9e377984ccd4",
      "operationId": "f4d804af-a54b-4a0e-88c4-05f471eab474",
      "creationDate": "2022-12-22T14:42:11.9761030Z",
      "accountingDate": "2021-12-06T11:54:00.0000000Z",
      "affectingDate": "2021-12-06T11:54:00.0000000Z",
      "debitAccountId": "c1c52eef-1356-4a51-a7f0-7b08c7bc6ff6",
      "creditAccountId": "5b3c088a-95e1-4a4f-9bad-f76bd985937e",
      "amount": 1.51,
      "description": "П л а т е ж (У п л а т а п р о ц е н т о в н а п р о с р о ч е н н ы й О Д) (3). П р и е м п л а т е ж а 2021-12-07",
      "stornoEntryId": null
    },
    {
      "entryId": "817e75eb-de1b-439c-9b45-90571015440f",
      "operationId": "f4d804af-a54b-4a0e-88c4-05f471eab474",
      "creationDate": "2022-12-22T14:42:11.9761160Z",
      "accountingDate": "2021-12-06T11:54:00.0000000Z",
      "affectingDate": "2021-12-06T11:54:00.0000000Z",
      "debitAccountId": "ac44dca1-dd9c-448f-964a-b7db4447e578",
      "creditAccountId": "542f8c70-cb1c-4bb7-9aff-809a333ddc1a",
      "amount": 0.75,
      "description": "П л а т е ж (У п л а т а п е н и) (6). П р и е м п л а т е ж а 2021-12-07",
      "stornoEntryId": null
    },
    {
      "entryId": "53ccea1a-75bd-40b6-a9bc-c83a28818b16",
      "operationId": "f4d804af-a54b-4a0e-88c4-05f471eab474",
      "creationDate": "2022-12-22T14:42:11.9761270Z",
      "accountingDate": "2021-12-06T11:54:00.0000000Z",
      "affectingDate": "2021-12-06T11:54:00.0000000Z",
      "debitAccountId": "f93603db-cc86-4648-b2c4-c404f826209a",
      "creditAccountId": "0d48dfb0-9531-4163-8cdd-99fb64e1c2ba",
      "amount": 406.45,
      "description": "П л а т е ж (П р е д о п л а т а) (7). П р и е м п л а т е ж а 2021-12-07",
      "stornoEntryId": null
    }
  ],
}
```

```

{
  "entryId": "39b77de8-3dfd-4083-8089-289aac53a617",
  "operationId": "f4d804af-a54b-4a0e-88c4-05f471eab474",
  "creationDate": "2022-12-22T14:42:11.9761390Z",
  "accountingDate": "2021-12-06T11:54:00.0000000Z",
  "affectingDate": "2021-12-06T11:54:00.0000000Z",
  "debitAccountId": "f93603db-cc86-4648-b2c4-c404f826209a",
  "creditAccountId": "c1c52eef-1356-4a51-a7f0-7b08c7bc6ff6",
  "amount": 141.51,
  "description": "Платеж (Уплата процентов) (11). Прием платежа 2021-12-07",
  "stornoEntryId": null
},
{
  "entryId": "19d2b24e-da1a-4ca0-93b7-ff6096b4fd36",
  "operationId": "f4d804af-a54b-4a0e-88c4-05f471eab474",
  "creationDate": "2022-12-22T14:42:11.9761500Z",
  "accountingDate": "2021-12-06T11:54:00.0000000Z",
  "affectingDate": "2021-12-06T11:54:00.0000000Z",
  "debitAccountId": "f93603db-cc86-4648-b2c4-c404f826209a",
  "creditAccountId": "0bef5f11-836c-45a7-b3c8-d8068741311f",
  "amount": 151.29,
  "description": "Платеж (Уплата ОД) (12). Прием платежа 2021-12-07",
  "stornoEntryId": null
},
{
  "entryId": "365675bc-735b-4b57-83ae-9068b226c1a9",
  "operationId": "f4d804af-a54b-4a0e-88c4-05f471eab474",
  "creationDate": "2022-12-22T14:42:11.9761620Z",
  "accountingDate": "2021-12-06T11:54:00.0000000Z",
  "affectingDate": "2021-12-06T11:54:00.0000000Z",
  "debitAccountId": "f93603db-cc86-4648-b2c4-c404f826209a",
  "creditAccountId": "ac44dca1-dd9c-448f-964a-b7db4447e578",
  "amount": 0.75,
  "description": "Платеж (Уплата пени) (13). Прием платежа 2021-12-07",
  "stornoEntryId": null
}
]
}

```

Остатки по счетам договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1" до операции были:

GET /api/accounting-system/v1/objects/a42a9998-53ac-44ed-aa73-9d9f26dd2ca1/balances?date=2021-12-06T14%3A54%3A00.0000000%2B0300

Ответ:

```

[
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Платеж ОД",
    "balance": 0
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Процент начисленный",
    "balance": 8.4871
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Доход по пени",
    "balance": 0.75645
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Пени начисленные",
    "balance": 0.75645
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Основной долг",
    "balance": 848.71
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Процент начисленный на просроченный ОД",
    "balance": 1.5129
  }
]

```

```
[
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Просроченный платеж ОД",
    "balance": 151.29
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Просроченный платеж Процент",
    "balance": 140
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Доход по процентам",
    "balance": 150
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Платеж Процент",
    "balance": 0
  }
]
```

Остатки по счетам договора №"12346" id:"a42a9998-53ac-44ed-aa73-9d9f26dd2ca1" после операции стали:

GET /api/accounting-system/v1/objects/a42a9998-53ac-44ed-aa73-9d9f26dd2ca1/balances?date=2021-12-06T14:3A54:3A01.0000000%2B0300

Ответ:

```
[
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "ОД Уплаченный",
    "balance": 0
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Предоплата",
    "balance": 406.45
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Платеж ОД",
    "balance": 0
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Процент начисленный",
    "balance": 8.4871
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Доход по пени",
    "balance": 0.75645
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Пени начисленные",
    "balance": 0.00645
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Основной долг",
    "balance": 848.71
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Процент начисленный на просроченный ОД",
    "balance": 0.0029
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Просроченный платеж ОД",
    "balance": 0
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Просроченный платеж Процент",
    "balance": 0
  }
]
```

```
    "balance": 0
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Пени уплаченные",
    "balance": 0
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Процент уплаченный",
    "balance": 0
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Доход по процентам",
    "balance": 150
  },
  {
    "organizationId": "eef268f7-33aa-4772-b37d-f86a8626603f",
    "accountTypeName": "Платеж Процентов",
    "balance": 0
  }
]
```