Table of Contents

# Документация по методу calculatePayments

## Оглавление

1. [Введение](#введение)
2. [Входные данные](#входные-данные)
3. [Алгоритм расчета](#алгоритм-расчета)
4. [Результат расчета](#результат-расчета)
5. [Примеры](#примеры)
6. [Обработка ошибок](#обработка-ошибок)

## Введение

Метод calculatePayments предназначен для автоматического расчета графика платежей по сделкам в системе Bitrix24. Система поддерживает расчет платежей по двум типам договоров: основному договору (ДДУ, земля, доп. лот) и договору подряда.

### Назначение системы

* Автоматический расчет графика платежей
* Поддержка различных частот платежей
* Разделение платежей по типам договоров
* Гибкая настройка первоначальных взносов
* Возможность последующего редактирования

## Входные данные

### 2.1 Основные параметры

|  |  |  |
| --- | --- | --- |
| Параметр | Описание | Обязательность |
| deal\_id | ID сделки в Bitrix24 | Да |
| auth | Токен авторизации приложения | Да |
| number\_graph | Номер графика (по умолчанию 0) | Нет |

### 2.2 Финансовые параметры

|  |  |  |
| --- | --- | --- |
| Поле Bitrix24 | Описание | Тип данных |
| OPPORTUNITY | Общая сумма сделки | Число |
| UF\_CRM\_1602665856 | Первоначальный взнос (ПВ) по основному договору | Число |
| UF\_CRM\_1610624104 | Сумма по земле | Число |
| UF\_CRM\_1610624123 | Сумма по подряду | Число |
| UF\_CRM\_1612333754254 | ПВ за подряд | Число |

### 2.3 Даты

|  |  |  |
| --- | --- | --- |
| Поле Bitrix24 | Описание | Обязательность |
| UF\_CRM\_1612354364 | Дата окончания строительства | Да |
| UF\_CRM\_1602665724 | Дата создания договора | Нет |
| UF\_CRM\_1611646586442 | Дата заключения подряда | Нет |
| UF\_CRM\_AMO\_560317 | Дата создания основного договора | Да |

### 2.4 Временные параметры

|  |  |  |
| --- | --- | --- |
| Поле Bitrix24 | Описание | По умолчанию |
| UF\_CRM\_1610624156 | Дней на внесение ПВ (ДДУ/земля/доп лот) | 0 |
| UF\_CRM\_1617787747 | Дней на внесение ПВ (подряд) | 0 |
| UF\_CRM\_1602666205 | Рассрочка, мес. (количество платежей) | 1 |

### 2.5 Частота платежей

|  |  |  |
| --- | --- | --- |
| Код | Описание | Период |
| 2302 | Месяц | 1 месяц |
| 2304 | Квартал | 3 месяца |
| 2306 | Полгода | 6 месяцев |
| 2308 | Год | 12 месяцев |

### 2.6 Номера договоров

|  |  |  |
| --- | --- | --- |
| Поле Bitrix24 | Описание | Обязательность |
| UF\_CRM\_1611646713185 | Номер основного договора | Да |
| UF\_CRM\_1611646728314 | Номер договора подряда | Нет |

## Алгоритм расчета

### 3.1 Проверки перед расчетом

1. **Проверка авторизации**
   * Валидация токена приложения
   * Проверка прав доступа
2. **Проверка существования графика**
   * Если график уже составлен, расчет не выполняется
   * Предотвращение дублирования
3. **Проверка суммы сделки**
   * Сумма должна быть больше 0
   * Обработка ошибки “Сумма сделки равна 0”
4. **Проверка обязательных дат**
   * Дата окончания строительства
   * Дата заключения основного договора
5. **Проверка номеров договоров**
   * Номер основного договора обязателен
   * Номер договора подряда (если есть подряд)

### 3.2 Расчет количества платежей

$count = floor(Carbon::create($finishDate)->diffInMonths(  
 Carbon::create($dateStartMain)->addDays($daysFirstPayment)  
) / $quantityFromFrequency);

**Логика расчета:** 1. Вычисляется период от даты ПВ до даты окончания строительства 2. Период делится на частоту платежей 3. Используется floor() для округления вниз

### 3.3 Расчет по основному договору

#### Шаг 1: Первоначальный взнос (ПВ)

mainDownPaymentCalculate($dealId, $numberLand, $firstPayment, $dateStartMain, $daysFirstPayment, $numberGraph)

**Параметры платежа:** - Тип: DOWN\_PAYMENT - Дата: дата начала + дни на ПВ - Статус: BLOCK (заблокирован) - Сумма: значение из поля ПВ

#### Шаг 2: Регулярные платежи

mainRegularPaymentCalculate($dealId, $numberLand, $landAmount, $firstPayment, $dateStartMain, $daysFirstPayment, $count, $paymentFrequency, $finishDate, $numberGraph)

**Алгоритм расчета суммы:**

$amount = ceil(($allAmount - $firstAmount) / $count);

**Логика распределения:** 1. Если платеж один: вся оставшаяся сумма 2. Если платежей несколько: - Первые (n-1) платежей: равные суммы - Последний платеж: остаток суммы

### 3.4 Расчет по подряду

#### Шаг 1: ПВ по подряду

contractDownPaymentCalculate($dealId, $numberContract, $firstPaymentContract, $dateStart, $daysFirstContractPayment, $numberGraph)

**Особенности:** - Дата ПВ подряда = дата ПВ земли - Статус: BLOCK (заблокирован)

#### Шаг 2: Регулярные платежи по подряду

contractRegularPaymentCalculate($dealId, $numberContract, $contractAmount, $firstPaymentContract, $dateStart, $daysFirstContractPayment, $count, $paymentFrequency, $numberGraph)

**Алгоритм аналогичен основному договору**

### 3.5 Расчет дат платежей

switch ($type) {  
 case '2302': $carbDate = $carbDate->addMonth(); break; // Месяц  
 case '2304': $carbDate = $carbDate->addMonths(3); break; // Квартал  
 case '2306': $carbDate = $carbDate->addMonths(6); break; // Полгода  
 case '2308': $carbDate = $carbDate->addYear(); break; // Год  
}

## Результат расчета

### 4.1 Структура данных платежа

Каждый платеж содержит следующие поля:

|  |  |  |
| --- | --- | --- |
| Поле | Описание | Тип данных |
| deal\_id | ID сделки | Integer |
| type | Тип (MAIN/CONTRACT) | String |
| pay\_type | Тип платежа (DOWN\_PAYMENT/REGULAR\_PAYMENT) | String |
| doc\_number | Номер договора | String |
| amount | Сумма платежа | Decimal |
| date | Дата платежа | Date |
| blocked | Статус блокировки | Integer |
| note | Комментарий | String |
| order | Порядок | Integer |
| number\_graph | Номер графика | Integer |

### 4.2 Типы платежей

#### DOWN\_PAYMENT (Первоначальный взнос)

* **Статус:** BLOCK (заблокирован)
* **Изменяемость:** Не подлежит изменению
* **Создание:** Автоматически при расчете

#### REGULAR\_PAYMENT (Регулярные платежи)

* **Статус:** ACTIVE (активен)
* **Изменяемость:** Может быть изменен
* **Распределение:** Равномерно

### 4.3 Результат расчета

1. **График платежей** - полный список плановых платежей
2. **Два типа договоров:**
   * Основной договор (ДДУ, земля, доп. лот)
   * Договор подряда (если есть)
3. **Структурированные данные** для отображения в интерфейсе
4. **Возможность дальнейшего редактирования** платежей

## Примеры

### 5.1 Пример расчета

**Входные данные:** - Сумма сделки: 1,000,000 руб. - ПВ: 200,000 руб. - Сумма по земле: 800,000 руб. - Сумма по подряду: 200,000 руб. - ПВ по подряду: 50,000 руб. - Частота: ежемесячно (2302) - Период: 12 месяцев

**Результат:**

#### Основной договор:

1. ПВ: 200,000 руб. (заблокирован)
2. Регулярные платежи: 11 платежей по 54,545 руб. + 1 платеж 54,545 руб.

#### Договор подряда:

1. ПВ: 50,000 руб. (заблокирован)
2. Регулярные платежи: 11 платежей по 13,636 руб. + 1 платеж 13,637 руб.

**Итого:** 24 платежа с точным соответствием общей сумме.

### 5.2 Особенности алгоритма

#### Распределение сумм

**Принцип “последний платеж получает остаток”:**

$lastAmount = $allAmount - $firstAmount - ($amount \* ($count - 1));

Это гарантирует точное соответствие общей сумме.

#### Обработка резервов

При пересчете учитываются заблокированные платежи:

$planPaymentMainAmountReserv = $this->mPlanPayment  
 ->where('deal\_id', $dealId)  
 ->where('type', $this->mPlanPayment::MAIN)  
 ->where('blocked', $this->mPlanPayment::BLOCK)  
 ->sum('amount');

## Обработка ошибок

### 6.1 Типы ошибок

|  |  |  |
| --- | --- | --- |
| Ошибка | Описание | Условие возникновения |
| “Приложение не авторизовано” | Неверный токен | Неправильный auth |
| “График уже составлен” | Дублирование | График уже существует |
| “Сумма сделки равна 0” | Нулевая сумма | OPORTUNITY = 0 |
| “Не указана дата окончания строительства” | Отсутствует дата | UF\_CRM\_1612354364 пустое |
| “Не указана дата заключения основного договора” | Отсутствует дата | UF\_CRM\_AMO\_560317 пустое |
| “Не указан номер основного договора” | Отсутствует номер | UF\_CRM\_1611646713185 пустое |
| “Не указан номер договора подряда” | Отсутствует номер | UF\_CRM\_1611646728314 пустое (если есть подряд) |

### 6.2 Логирование ошибок

Все ошибки логируются в систему:

Log::error('Perspektivakrym: ' . $e->getMessage());

### 6.3 Возврат результата

При успешном выполнении: - Возвращается обновленная страница с графиком платежей

При ошибке: - Возвращается страница с сообщением об ошибке - Сохраняется контекст сделки для повторной попытки

## Заключение

Метод calculatePayments представляет собой комплексную систему автоматического расчета графиков платежей с учетом различных типов договоров, частот платежей и особенностей бизнес-логики. Система обеспечивает точность расчетов, гибкость настройки и надежную обработку ошибок.