

УДК 004.4

**ПРИМЕНЕНИЕ БИБЛИОТЕКИ NODE-CRON ДЛЯ АВТОМАТИЗАЦИИ  
РАСЧЁТА ЗАДОЛЖЕННОСТЕЙ И ОТПРАВКИ УВЕДОМЛЕНИЙ**

Хивинцева П.С., студент гр. ИТб-222, IV курс

Научный руководитель: Ванеев О.Н., к.т.н., доцент

Кузбасский государственный технический университет имени Т.Ф. Горбачёва  
г. Кемерово

В настоящее время одним из ключевых факторов повышения эффективности предприятий становится автоматизация рутинных операций. Но несмотря на широкое распространения информационных систем, на предприятиях, особенно небольших, по-прежнему остаётся значительное количество ручной работы. Это повышает вероятность ошибок и увеличивает временные затраты.

ООО «АРКА» – организация, занимающаяся сдачей недвижимости в аренду – ведёт учёт договоров, поступающих платежей и расчёт задолженностей арендаторов преимущественно вручную с использованием электронных таблиц Excel. Данный подход трудоёмкий, зависит от человеческого фактора, несёт риск ошибок при расчётах и несвоевременного выявления задолженностей, что сказывается на финансовых результатах компании.

Особой проблемой является отсутствие автоматического контроля за долгами. Бухгалтеры вынуждены самостоятельно отслеживать задолженности и вручную уведомлять арендаторов через мессенджеры или телефонные звонки.

В данной работе разрабатывается информационная система автоматизации учёта договоров аренды и платежей для предприятия ООО «АРКА».

**Описание системы**

Разрабатываемая система учёта договоров аренды и платежей предназначена для учёта данных об арендаторах, договорах, платежах и объектах недвижимости, расчёта задолженностей и информировании арендаторов о долгах.

Основные функции системы:

- авторизация;
- автоматическое уведомление о задолженностях;
- формирование уведомления арендатору (текст сообщения);
- автоматический расчёт задолженности;
- добавление/изменение/удаление платежей;
- добавление/изменение/удаление договоров;
- добавление/изменение/удаление арендаторов;
- добавление/изменение/удаление объектов недвижимости;
- генерация отчёта Excel.

```

graph TD
    Actor[<<worker>> Бухгалтер]
    
    subgraph "Подсистема авторизации"
        UC1([Авторизация])
    end
    
    subgraph "Подсистема уведомлений"
        UC2([Автоматическое уведомление о задолженностях])
        UC3([Формирование уведомления арендатору])
        UC2 -.->|<<Include>>| UC3
    end
    
    subgraph "Подсистема управления данными"
        UC4([Автоматический расчёт суммы задолженности])
        UC5([Удаление платежа])
        UC6([Добавление платежа])
        UC7([Редактирование платежа])
        UC8([Удаление договора])
        UC9([Добавление договора])
        UC10([Редактирование договора])
        UC4 -.->|<<Include>>| UC5
        UC4 -.->|<<Include>>| UC6
        UC4 -.->|<<Include>>| UC7
    end
    
    subgraph "Подсистема отчётности"
        UC11([Удаление арендатора])
        UC12([Добавление арендатора])
        UC13([Редактирование арендатора])
        UC14([Удаление объекта])
        UC15([Добавление объекта])
        UC16([Редактирование объекта])
        UC17([Генерация отчёта Excel])
    end
    
    Actor --- UC1
    Actor --- UC5
    Actor --- UC8
    Actor --- UC11
    Actor --- UC14
    Actor --- UC17
    Actor --- UC6
    Actor --- UC9
    Actor --- UC12
    Actor --- UC15
    Actor --- UC7
    Actor --- UC10
    Actor --- UC13
    Actor --- UC16
    Actor --- UC3

```

Бухгалтер взаимодействует с системой вручную: авторизуется, добавляет и редактирует данные об арендаторах, объектах, договорах, платежах, формирует отчёт в Excel. А расчёт задолженности и отправка уведомлений арендаторам выполняется автоматически – без участия пользователя.

В создании системы используются следующие технологии:

- Система разрабатывается по архитектуре «клиент-сервер».

Серверная часть на платформе Node.js принимает запросы от браузера, выполняет бизнес-логику и возвращает результат.

Все данные хранятся в реляционной базе данных PostgreSQL.

## Node.js и npm

Node.js – платформа для разработки серверных приложений на языке JavaScript [1]. Вместе с Node.js устанавливается npm [2] (Node Package Manager) – это каталог готовых библиотек, благодаря которому любая нужная функциональность подключается одной командой в терминале – не нужно писать всё с нуля!

Ниже представлены основные библиотеки, планируемые к использованию в серверной части проекта.

- Express.js – это фреймворк, который упрощает построение серверной части, она позволяет коротко и понятно описать маршруты: какой запрос куда идёт и что с ним делать;
- pg – драйвер для работы с PostgreSQL для взаимодействия с базой данных: от подключения до выполнения SQL-запросов из кода на JavaScript.
- nodemailer – инструмент для отправки писем. Он подключается к почтовому серверу и отправляет сформированное сообщение на адрес арендатора;
- exceljs – генерирует файлы в формате .xlsx прямо на сервере.
- node-cron – библиотека, позволяющая запускать любую функцию автоматически в заданное время. Именно благодаря ей планируется расчёт задолженностей и рассылка уведомлений.

### Библиотека node-cron

node-cron – библиотека для Node.js, позволяющая запускать функции автоматически по расписанию [3].

Принцип работы прост: разработчик указывает, когда и какую функцию нужно выполнить, и библиотека запускает её в нужный момент без участия пользователя.

Расписание задаётся с помощью cron-выражения – это строка из шести полей, соответственно отвечающих за секунды, минуты, часы, день месяца, месяц и день недели. Например, выражение (0 16 \* \* 1-5) означает: запуск каждый рабочий день в 16:00. Звёздочка в выражении – это любое значение, что позволяет гибко настраивать расписание под любую задачу.

В разрабатываемой системе с помощью node-cron планируется реализация расчёта задолженностей и отправки уведомлений арендаторам-должникам. Пример задания расписания для двух процессов системы представлен на листинге 1.

#### Листинг 1. Пример использования node-cron

```
// Расчёт задолженностей – каждый день в 00:00
cron.schedule('0 0 * * *', () => {
  calculateDebts();
});

// Отправка уведомлений – каждый рабочий день в 16:00
cron.schedule('0 16 * * 1-5', () => {
  sendNotifications();
});
```

Первый процесс – расчёт задолженностей. Он запускается автоматически при каждом добавлении, изменении или удалении платежа, а также ежедневно в 00:00, как итоговая сверка за прошедший день. Это позволяет выявить договоры, по которым обязательный платёж за день не поступил, и рассчитать по ним задолженность.

Логика работы: сервер обходит все активные договоры и для каждого из них вычисляет разницу между тем, сколько арендатор должен был заплатить за прошедшие месяцы, и тем, сколько он на самом деле заплатил. Результат сохраняется в базу данных, и бухгалтер каждое утро уже видит актуальные данные о долгах, не тратя на это время.

Функция запуска расчёта задолженностей планировщиком `node-cron` будет находиться внутри класса Задолженности. Диаграмма последовательности процесса представлена на рисунке 2.

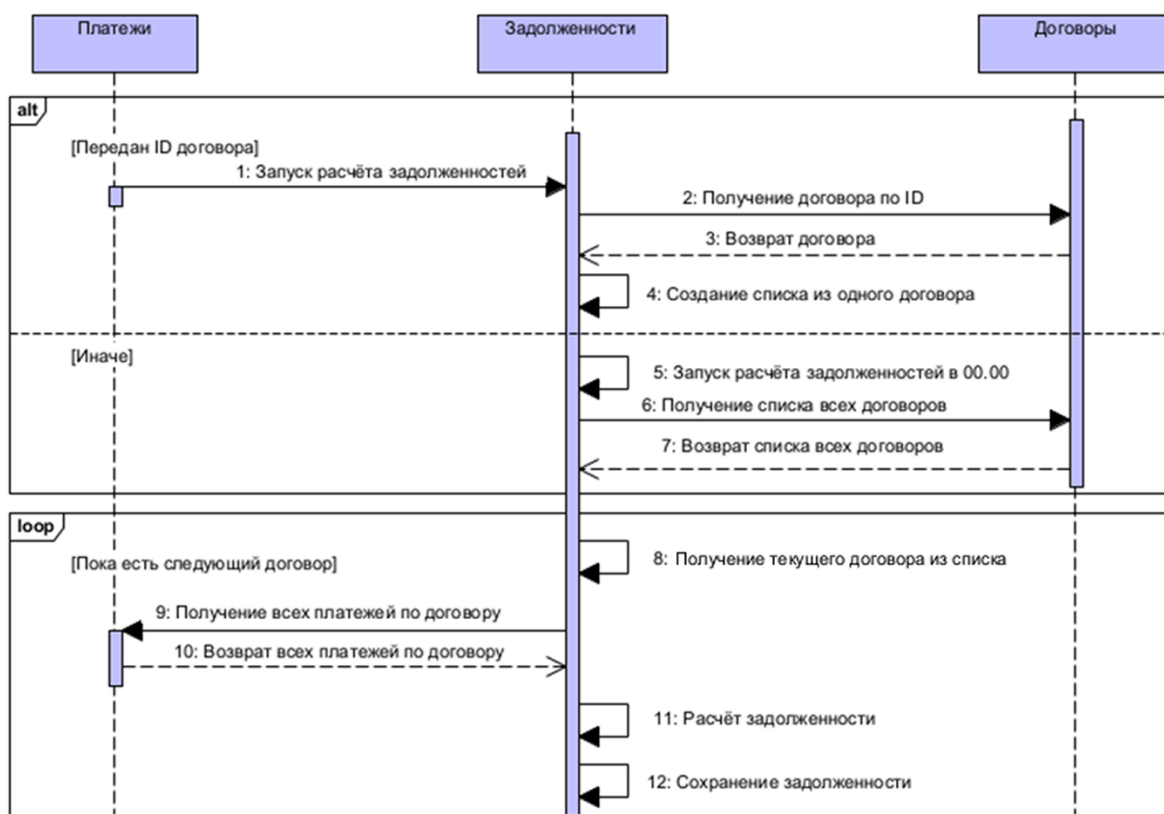


Рисунок 2 – Диаграмма последовательности процесса «Автоматический расчёт задолженностей»

Второй процесс – отправка уведомлений – запускается каждый рабочий день в 16:00 (конец рабочего дня бухгалтеров). К этому моменту данные о задолженностях уже актуальны – бухгалтер успел внести все поступившие на день платежи, а значит все долги пересчитаны.

Логика работы: сервер находит всех арендаторов с задолженностью больше нуля, формирует для каждого письмо с напоминанием и суммой долга, и отправляет его по электронной почте. Арендатор получает уведомление без участия бухгалтера.

Процесс отправки уведомлений аналогичен по структуре с первым процессом: node-cron запускает функцию по расписанию, которая обращается к базе данных и выполняет нужную логику.

Таким образом, применение данной библиотеки позволяет полностью исключить из работы бухгалтеров два трудоёмких процесса – расчёт задолженностей и уведомление арендаторов. Оба процесса выполняются автоматически в заданное время, независимо от действий пользователя.

### **Заключение**

В данной статье рассмотрено применение библиотеки node-cron в разрабатываемой системе автоматизации учёта договоров аренды и платежей для предприятия ООО «АРКА».

Благодаря node-cron автоматизируемые процессы выполняются по заданному расписанию без участия бухгалтеров, что снижает трудозатраты и исключает возникновение ошибок.

Разрабатываемая система позволит повысить эффективность работы предприятия и снизить финансовые потери, связанные с неверными расчётами и несвоевременным обнаружением задолженностей.

### **Список литературы:**

1. Node.js – Introduction to Node.js. [Электронный ресурс] – URL: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs> (дата обращения: 22.03.2026).
2. npm Docs. [Электронный ресурс] – URL: <https://docs.npmjs.com/> (дата обращения: 22.03.2026).
3. node-cron / npm. [Электронный ресурс] – URL: <https://www.npmjs.com/package/node-cron> (дата обращения: 22.03.2026).