



РУТОЛЛ

МОТИВАЦИЯ ОПЛАТЫ. МОДУЛЬ СВП-2+

РУКОВОДСТВО АДМИНИСТРАТОРА

Версия 1.0

ОГЛАВЛЕНИЕ

Журнал изменений.....	4
Область применения	5
Термины.....	6
Принятые сокращения.....	9
1 Общие сведения о программе.....	10
1.1 Обозначение и наименование программы	10
1.2 Назначение программы.....	10
1.3 Требования к ПК пользователя.....	10
1.4 Требования к серверу для установки	10
1.5 Краткое описание возможностей ПО «Мотивация оплаты. Модуль СВП-2+»	11
2 Установка и настройка ПО «МОтивация оплаты. модуль свп-2+»	12
2.1 Состав программного приложения	12
2.2 Установка БД PostgreSQL	12
2.3 Установка Apache NiFi 2.5.0.....	13
2.4 Установка RabbitMQ	15
2.5 Установка KeyDB 6.3.3	16
2.6 Установка MiniO	17
2.7 Установка сервиса rutoll-debtweb	18
2.8 Сервис rutoll-debt-listener	19
2.9 Установка интернет-браузера	20
3 Конфигурирование ПО «Мотивация оплаты. модуль свп-2+»	21
3.1 Конфигурирование сервиса rutoll-debtweb	21
3.1.1 Общая конфигурация сервиса.....	21
3.1.2 Настройка документов.....	21
3.1.3 Настройка генерации QR-кодов	22
3.1.4 Настройка REST API	22

3.1.5	Настройка БД.....	22
3.1.6	Настройка работы с хранилищем MiniO	23
3.1.7	Настройка очереди сообщений RabbitMQ.....	24
3.1.8	Настройка кэширования данных	24
3.1.9	Настройка интеграции с внешним шлюзом.....	25
3.1.10	Swagger-документация.....	25
3.1.11	Метрики Prometheus	25
3.1.12	Настройка WebSocket-соединения	26
3.1.13	Настройка аутентификации и токенов	26
3.2	Конфигурирование сервиса rutoll-debt-listener	26
3.2.1	Настройка работы с брокером сообщений	27
3.2.2	Настройка работы с БД	27
3.2.3	Настройка логгирования	27
4	Основные операции при эксплуатации	28
4.1	Обновление ПО «Мотивация оплаты. Модуль СВП-2+»	28
4.2	Логирование ПО «Мотивация оплаты. Модуль СВП-2+».....	28

ЖУРНАЛ ИЗМЕНЕНИЙ

Дата/Версия	Описание	Автор
27.10.2025/v 1.0	Начальная версия документа	Вишнякова Е.С.

ОБЛАСТЬ ПРИМЕНЕНИЯ

Настоящее руководство регламентирует работы по установке и обновлению компонентов программного обеспечения «Мотивация оплаты. Модуль СВП-2+» (далее - ПО "Мотивация оплаты») на объектах заказчика.

В основной части документа приведены общие сведения о структуре программного обеспечения и об архитектуре ПО «Мотивация оплаты», в частности, приведены последовательности выполнения операций по установке, настройке и обновлению на рабочем месте специалиста.

В документе отсутствуют сведения об установке операционной системы на ПК.

СВП РУТОЛЛ находится в непрерывном развитии, поэтому этот документ имеет ограниченный срок действия. Для получения актуальной версии документа необходимо обращаться к разработчикам системы для получения релевантной информации и необходимых рекомендаций.

Все права защищены. Настоящее руководство не может быть полностью или частично воспроизведено, тиражировано и распространено в любой форме, будь то печатной или электронной, в качестве официального издания без разрешения ООО «РУТОЛЛ».

Несмотря на гарантию того, что содержащаяся в документе информация тщательно проверена, ООО «РУТОЛЛ» не несет ответственности за то, как документ используется.

Информация о найденных ошибках или предложения о том, как можно улучшить документ, приветствуются.

Поскольку продукты постоянно тестируются и улучшаются, ООО «РУТОЛЛ» оставляет за собой право вносить изменения в содержащуюся в данном руководстве информацию без предварительного уведомления.

ТЕРМИНЫ

Автоматизированное рабочее место кассира-оператора — специальное оборудование и программное обеспечение, с помощью которого кассир-оператор выполняет свои обязанности по взиманию платы на полосе проезда. Основным устройством АРМ, посредством которого осуществляется ввод команд, является компьютер с сенсорным монитором.

Автоматизированный режим работы полосы (Stop&Go) — режим работы полосы, при котором оплата проезда производится только электронными средствами оплаты (транспондерами, БСК и т.д.) или наличными с помощью АПП. Кассир-оператор на полосе отсутствует.

Автоматический режим работы полосы (Non-Stop) — режим работы полосы, при котором оплата проезда производится только транспондерами, без необходимости остановки для оплаты. Кассир-оператор на полосе отсутствует.

Бесконтактная смарт-карта (БСК) — персонифицированное или анонимное устройство (пластиковая карта) на основе электронной карты, содержащее идентификационные данные и/или денежные ресурсы своего владельца (держателя), являющееся техническим средством осуществления платежа. Принимает и передает данные на считыватель с помощью беспроводной технологии. На каждой БСК указан ее уникальный PAN-номер.

БСК служебная — идентификационная карта сотрудника компании-концессионера, предназначенная для осуществления доступа в программные продукты системы взимания платы RUTOLL.

БСК персонифицированная — электронное платежное средство, содержащее идентификационные данные и денежные ресурсы держателя карты (пользователя платного участка).

БСК анонимная — электронное платежное средство, содержащее денежные ресурсы держателя карты (пользователя платного участка).

БСК-считыватель — устройство для считывания информации с БСК и записи на нее данных.

Владелец дороги — организация, владеющая платными участками автомобильной дороги, устанавливающая размеры тарифов и скидок за проезд по ним, а также правила подключения к услуге интероперабельности.

Въездной ПВП — это ПВП, находящийся на въезде на платный участок автодороги.

Выездной ПВП — это ПВП, находящийся на выезде с платного участка автодороги.

Диспетчер — представитель Оператора дороги, ответственный за осуществление корректного сбора платы на подотчетных ПВП и являющийся пользователем АРМ диспетчера.

Интероперабельность — возможность для пользователя с помощью одного транспондера безостановочно проезжать по платным дорогам, эксплуатируемым различными

операторами — участниками системы межоператорского взаимодействия. Для того чтобы проезд мог быть осуществлен, достаточно заключить договор и приобрести транспондер только у одного из операторов.

Интероперабельный транспондер — транспондер, подключенный к услуге интероперабельности.

Кассир-оператор — сотрудник компании-концессионера, осуществляющий сбор платы за проезд на полосе проезда.

Класс (категория) транспортного средства — совокупность характеристик транспортных средств, отвечающая определенным условиям, в отношении которой установлен одинаковый базовый размер оплаты.

Клиент — физическое, юридическое лицо или организация, заключившие Договор с Оператором дороги на оказание Услуги.

Компания-концессионер — владелец (арендатор) платного участка (автодороги), на котором используется СВП RUTOLL.

Контроллер полосы — промышленный компьютер, предназначенный для управления оборудованием полосы проезда, записи информации о транзакциях и событиях, происходящих на полосе проезда, а также обмена данными с уровнем ПВП.

Нарушитель (ТС «Нарушитель») — транспортное средство, водитель которого совершил проезд без оплаты, хотя должен был по условиям пользования платным участком дороги.

Оператор — Общество, осуществляющее эксплуатацию платной автомагистрали.

Оператор дороги — организация, эксплуатирующая платный участок дороги и осуществляющая на этом участке сбор платы за проезд.

Подключение к услуге интероперабельности особенно актуально для водителей — физических лиц, а также для компаний-перевозчиков грузов, совершающих регулярные дальние поездки по платным дорогам, эксплуатируемым разными операторами.

Полоса проезда — сооружение, являющееся частью ПВП, обслуживающее одну полосу движения, предназначенное для оплаты Пользователем проезда и препятствующее несанкционированному бесплатному проезду ТС через этот участок. Обслуживается сотрудниками платного участка (операторами-кассирами).

Пользователь — представитель Оператора дороги, осуществляющий выполнение своих должностных обязанностей с использованием системы коммерческого управления; Клиент, использующий оборудование ПВП, например в режиме самообслуживания.

Пункт взимания платы — это часть СВП, включающая в себя полосы проезда, сооружения для размещения различных служб платного участка, кабины взимания платы и другие технические устройства.

Ручной режим работы полосы (stop&go) — режим работы полосы, при котором оплата проезда производится как электронными средствами оплаты (транспондерами, БСК и т.д.), так и наличными. Кассир-оператор на полосе присутствует.

Система взимания платы — это совокупность оборудования и программных средств, предназначенных для сбора платы на автодорогах.

Считыватель штрихкодов — устройство для считывания штрихкода, нанесённого на БСК, транспондер или чек, и занесения данной информации в АРМ кассира-оператора.

Транзакция — общепринятый в финансовом мире термин, которым обозначается совершение операции.

Транспондер — автономное бортовое устройство, предназначенное для бесконтактной регистрации фактов проезда ТС, содержащее идентификационные данные своего держателя (пользователя платного участка).

Услуга — платная услуга по организации проезда по платной автомагистрали.

Фискальный принтер (он же **фискальный регистратор, контрольно-кассовая машина**) — устройство, предназначенное для регистрации проданной Услуги и печати кассового чека.

Цветные списки ЭСРП — реестр электронных средств регистрации проезда, который содержит актуальную информацию о возможности их использовать для оплаты проезда. Стандартный набор списков перечислен ниже.

Белый — список устройств, оплата по которым разрешена.

Оранжевый — список устройств, оплата по которым разрешена, но баланс средств на счете низкий, и его стоит пополнить.

Серый — список устройств, оплата по которым запрещена, так как не хватает средств на счете.

Черный — список устройств, оплата по которым запрещена из-за блокировки. Блокировка является необратимой и может производиться как по запросу владельца (например, при утере), так и без такого запроса (в результате нарушения владельцем правил оплаты или проезда).

ЭСО — электронное средство оплаты проезда по платной автомагистрали. Например, БСК, транспондер, банковская карта.

ЭСРП — электронное средство регистрации проезда, частный случай ЭСО. Например, транспондер.

POS-терминал (банковский терминал) — электронное программно-техническое устройство для приема к оплате банковских карт.

ПРИНЯТЫЕ СОКРАЩЕНИЯ

Сокращение	Расшифровка
АРМ	Автоматизированное рабочее место
БД	База данных
ГРЗ	Государственный регистрационный знак транспортного средства
КП	Контроллер полосы
ПВП	Пункт взимания платы
СВП	Система взимания платы
ТС	Транспортное средство

1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

1.1 Обозначение и наименование программы

Наименование программного обеспечения — «Мотивация оплаты. Модуль СВП-2+».

1.2 Назначение программы

«Мотивация оплаты. Модуль СВП-2+» обеспечивает информационное взаимодействие с ИС ГК «Автодор» и предоставляет пользовательский интерфейс для автоматизации документооборота в части взаимодействия с пользователями по сценариям мотивации.

Работу с ПО «Мотивации оплаты» могут осуществлять следующие специалисты:

- руководитель пункта взимания платы (далее — ПВП),
- начальник смены ПВП,
- диспетчер,
- и другие уполномоченные специалисты.

1.3 Требования к ПК пользователя

АРМ «Мотивации оплаты» в составе ПО «Мотивация оплаты. Модуль СВП-2+» является веб-приложением, поэтому для работы пользователя необходим интернет-браузер. Установленная ОС не имеет значения.

Рекомендуемые системные требования к рабочей станции пользователя:

- CPU: Intel Core i5 или более производительный;
- RAM 16 Гб;
- HDD 512 Гб;
- видеокарта GeForce GTX 1650;
- монитор с поддержкой разрешения не менее 1920 x 1080 px;
- установленный браузер Google Chrome.

1.4 Требования к серверу для установки

Серверная часть ПО «Мотивация оплаты. Модуль СВП-2+» управляется ОС Linux.

Рекомендуемые системные требования к серверу:

- CPU: Intel Core i5 или более производительный;
- RAM 16 Гб;
- HDD 512 Гб.

1.5 Краткое описание возможностей ПО «Мотивация оплаты. Модуль СВП-2+»

Программное обеспечение «Мотивация оплаты. Модуль СВП-2+» :

- реализует интеграцию с ИС ГК «Автодор» в части получения информации о неоплаченных проездах по безбарьерным платным дорогам;
- обеспечивает передачу информации о неоплаченных проездах на уровень ПВП с учётом заданного интервала (по умолчанию 30 дней);
- агрегирует информацию о проездах должников (первичный с уведомлением, вторичные с выставлением счёта);
- предоставляет интерфейс для печати доказательной базы и фиксации результатов взаимодействия с должниками, а также построения отчётности и мониторинга работы системы.

2 УСТАНОВКА И НАСТРОЙКА ПО «МОТИВАЦИЯ ОПЛАТЫ. МОДУЛЬ СВП-2+»

2.1 Состав программного приложения

ПО «Мотивация оплаты. Модуль СВП-2+» включает в себя следующие компоненты:

- сервис веб-приложения `rutoll-debtweb`,
- сервис интеграционного шлюза с ИС ГК «Автодор» `rutoll-debt-listener`,
- БД PostgreSQL,
- Apache NiFi (ПО для автоматизации потоков данных между системами и сервисами),
- RabbitMQ (программный брокер сообщений, который обеспечивает асинхронный обмен данными между компонентами системы),
- Minio (высокопроизводительную объектную систему хранения данных),
- БД KeyDB.

Перед установкой и настройкой модулей и компонентов необходимо обновить систему. Чтобы обновить список пакетов и установить последние версии, следует воспользоваться командой:

```
```bash
sudo apt update && sudo apt upgrade -y
```

Перед установкой сервисов `rutoll-debtweb` и `rutoll-debt-listener`, необходимо добавить записи в файл `/etc/hosts`:

1. Отредактировать файл `/etc/hosts`:

```
sudo nano /etc/hosts
```

2. Добавить строку:

```
<ip-адрес шлюза> доменное_имя_шлюза
```

### 2.2 Установка БД PostgreSQL

Чтобы установить БД PostgreSQL на сервер, требуется:

1. Установить пакет и проверить установленную версию:

```
sudo apt install postgresql
psql -version
```

Пример вывода:

```
psql (PostgreSQL) 14.19 (Ubuntu 14.19-0ubuntu0.22.04.1)
```

2. Настроить конфигурацию, открыв основной конфигурационный файл. Изменить основные параметры в соответствии с проектом:

```
sudo nano /etc/postgresql/14/main/postgresql.conf
```

Python

```
listen_addresses = '0.0.0.0'
port = 5432
max_connections = 100
```

Эти настройки позволяют принимать подключения со всех адресов и задают максимальное количество соединений.

3. Создать пользователей и роли с необходимыми правами, авторизовавшись как системный пользователь PostgreSQL:

```
sudo -i -u postgres
```

SQL

```
CREATE USER debt WITH PASSWORD '<password>';
```

```
CREATE ROLE debt_admin WITH
 LOGIN
 PASSWORD '<password>'
 SUPERUSER
 CREATEDB
 CREATEROLE
 INHERIT
 REPLICATION;
```

4. Настроить доступ, открыв файл управления доступом и добавив строки в конец файла:

```
sudo nano /etc/postgresql/14/main/pg_hba.conf
host all debt 0.0.0.0/0 md5
host all debt_admin 0.0.0.0/0 md5
```

Это разрешает подключение к базе данных пользователям `debt` и `debt_admin` с любых IP-адресов по паролю.

5. Активировать и запустить PostgreSQL:

```
sudo systemctl enable postgresql.service
sudo systemctl start postgresql.service
sudo systemctl status postgresql.service
```

Пример успешного вывода:

```
● postgresql.service - PostgreSQL RDBMS
 Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
```

## 2.3 Установка Apache NiFi 2.5.0

Для установки Apache NiFi необходимо:

1. Настроить зависимости и установить необходимые пакеты:

```
sudo apt install unzip openjdk-21-jdk -y
```

## 2. Проверить установленную версию Java:

```
java -version
```

Пример вывода:

```
openjdk version "21.0.8" 2025-07-15
OpenJDK Runtime Environment (build 21.0.8+9-Ubuntu-0ubuntu122.04.1)
OpenJDK 64-Bit Server VM (build 21.0.8+9-Ubuntu-0ubuntu122.04.1, mixed mode,
sharing)
```

## 3. Создать каталог и скачать архив:

```
sudo mkdir -p /opt/nifi \
&& sudo wget -P /opt/nifi https://archive.apache.org/dist/nifi/2.5.0/nifi-2.5.0-
bin.zip \
&& sudo unzip /opt/nifi/nifi-2.5.0-bin.zip -d /opt/nifi \
&& sudo rm /opt/nifi/nifi-2.5.0-bin.zip \
&& cd /opt/nifi/nifi-2.5.0
```

## 4. Создать unit-файл для автозапуска NiFi, настроить systemd-сервиса:

```
sudo bash -c 'cat > /etc/systemd/system/nifi.service <<EOF
[Unit]
Description=Apache NiFi 2.5.0
After=network.target

[Service]
Type=forking
User=support
Group=support
WorkingDirectory=/home/support/apache_nifi/nifi-2.5.0

Java 21 обязателен для NiFi 2.5.0
Environment="JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64"
Environment="NIFI_HOME=/home/support/apache_nifi/nifi-2.5.0"

ExecStart=/home/support/apache_nifi/nifi-2.5.0/bin/nifi.sh start
ExecStop=/home/support/apache_nifi/nifi-2.5.0/bin/nifi.sh stop
ExecReload=/home/support/apache_nifi/nifi-2.5.0/bin/nifi.sh restart

Restart=on-failure
LimitNOFILE=65536
TimeoutStartSec=300
SuccessExitStatus=143
KillMode=process

[Install]
WantedBy=multi-user.target
EOF'
```

## 5. Применить изменения, запустить и проверить службы:

```
sudo systemctl daemon-reload
sudo systemctl enable nifi
sudo systemctl start nifi
sudo systemctl status nifi
```

Пример успешного статуса:

```
● nifi.service - Apache NiFi 2.5.0
 Loaded: loaded (/etc/systemd/system/nifi.service; disabled; vendor preset:
 enabled)
 Active: active (running) since Thu 2025-10-23 14:41:34 UTC; 37s ago
 Process: 3352 ExecStart=/home/support/apache_nifi/nifi-2.5.0/bin/nifi.sh
 start (code=exited, status=0/SUCCESS)
 Main PID: 929 (nifi.sh)
 Tasks: 138 (limit: 9159)
 Memory: 2.4G
```

## 2.4 Установка RabbitMQ

Для установки и настройки брокера сообщений RabbitMQ необходимо:

1. Установить пакет:

```
sudo apt-get install -y rabbitmq-server
```

2. Проверить версию:

```
sudo rabbitmqctl version
```

Пример вывода:

```
3.9.27
```

3. Включить веб-интерфейс, активировав плагин управления:

```
sudo rabbitmq-plugins enable rabbitmq_management
```

Веб-интерфейс доступен по адресу:

```
http://<IP Address>:15672
```

4. Создать виртуальный хост для приложения:

```
sudo rabbitmqctl add_vhost vh_main
```

5. Создать пользователя и назначить ему права администратора:

```
sudo rabbitmqctl add_user admin <password>
sudo rabbitmqctl set_user_tags admin administrator
sudo rabbitmqctl set_permissions -p vh_main admin ".*" ".*" ".*"
```

6. Создать обменник debt в виртуальном хосте vh\_main:

```
curl -u admin:<password> -H "content-type: application/json" -X PUT \
-d '{"type": "topic", "durable": true}' \
http://localhost:15672/api/exchanges/vh_main/debt
```

7. Добавить RabbitMQ в автозагрузку и запустить:

```
sudo systemctl enable rabbitmq-server
sudo systemctl start rabbitmq-server
sudo systemctl status rabbitmq-server
```

Пример успешного статуса:

```
● rabbitmq-server.service - RabbitMQ Messaging Server
 Loaded: loaded (/lib/systemd/system/rabbitmq-server.service; enabled; vendor
 preset: enabled)
 Active: active (running) since Thu 2025-09-25 18:25:22 UTC; 3 weeks 5 days
 ago
 Main PID: 829 (beam.smp)
 Tasks: 36 (limit: 9159)
 Memory: 62.6M
 CPU: 3h 39min 37.430s
```

## 2.5 Установка KeyDB 6.3.3

Для установки KeyDB версии 6.3.3 необходимо:

1. Добавить официальный PPA-репозиторий KeyDB:

```
sudo add-apt-repository ppa:equalpha/keydb -y
```

2. Обновить список пакетов:

```
sudo apt update
```

3. Установить версию 6.3.3:

```
sudo apt install keydb=6:6.3.3-1~jammy1 -y
```

4. Зафиксировать версию, чтобы избежать автоматического обновления:

```
sudo apt-mark hold keydb
```

5. Проверить установленную версию:

```
keydb-server -version
```

Пример вывода:

```
KeyDB server v=6.3.3 sha=00000000:0 malloc=jemalloc-5.2.1 bits=64
build=e067ff01e34480a9
```

6. Настроить конфигурацию, отредактировав файл настроек:

```
sudo nano /etc/keydb/keydb.conf
```

Изменить или добавить параметры:

```
bind 0.0.0.0
protected-mode yes
port 6379
daemonize no
```

7. Запустить и проверить службы, активировать автозапуск:

```
sudo systemctl enable keydb-server
sudo systemctl start keydb-server
sudo systemctl status keydb-server
```

Пример успешного статуса:

```
● keydb-server.service - Advanced key-value store
 Loaded: loaded (/lib/systemd/system/keydb-server.service; enabled; vendor
 preset: enabled)
 Active: active (running) since Thu 2025-10-23 14:33:20 UTC; 19min ago
 Docs: https://docs.keydb.dev,
 man:keydb-server(1)
 Main PID: 993 (keydb-server)
 Tasks: 11 (limit: 9159)
 Memory: 10.9M
```

## 2.6 Установка MinIO

Чтобы установить и настроить MinIO, требуется:

1. Загрузить и установить пакет:

```
wget https://dl.min.io/aistor/minio/release/linux-
amd64/minio_20250829212749.0.0_amd64.deb \
&& sudo dpkg -i minio_20250829212749.0.0_amd64.deb \
&& rm minio_20250829212749.0.0_amd64.deb
```

2. Обновить бинарный файл, загрузить актуальную версию в системный путь:

```
sudo wget https://dl.min.io/server/minio/release/linux-amd64/minio -O /usr/lo-
cal/bin/minio
```

3. Проверить версию:

```
/usr/local/bin/minio --version
```

Пример вывода:

```
minio version RELEASE.2025-07-23T15-54-02Z (commit-
id=7ced9663e6a791fef9dc6be798ff24cda9c730ac)
Runtime: go1.24.5 linux/amd64
License: GNU AGPLv3 - https://www.gnu.org/licenses/agpl-3.0.html
Copyright: 2015-2025 MinIO, Inc.
```

4. Подготовить директории и права:

```
sudo mkdir -p /var/minio/data
sudo chown -R minio-user:minio-user /var/minio
```

5. Создать файл конфигурации /etc/default/minio:

```
sudo tee /etc/default/minio >/dev/null <<'EOF'
MINIO_VOLUMES="/var/minio/data"
MINIO_OPTS="--console-address :9001"
MINIO_ROOT_USER=minio
MINIO_ROOT_PASSWORD=password
EOF
```

6. Настроить systemd-службы:

```
sudo systemctl edit minio
```

## 7. Добавить конфигурацию:

```
[Service]
EnvironmentFile=-/etc/default/minio
ExecStart=
ExecStart=/usr/local/bin/minio server $MINIO_OPTS $MINIO_VOLUMES
User=minio-user
Group=minio-user
```

## 8. Запустить и проверить службы:

```
sudo systemctl enable minio.service
sudo systemctl start minio.service
sudo systemctl status minio.service
```

Пример успешного статуса:

```
● minio.service - MinIO
 Loaded: loaded (/lib/systemd/system/minio.service; enabled; vendor preset: en-
 abled)
 Drop-In: /etc/systemd/system/minio.service.d
 └─override.conf
 Active: active (running) since Fri 2025-09-05 07:49:00 UTC; 1 month 17 days
 ago
 Docs: https://docs.min.io
 Main PID: 12042 (minio)
 Tasks: 15
 CPU: 4h 3min 2.398s
 CGroup: /system.slice/minio.service
 └─12042 /usr/local/bin/minio server --console-address :9001
 /var/minio/data
```

Веб-интерфейс MinIO доступен по адресу: `http://<server-ip>:9001`

## 9. Установить Midnight Commander:

```
sudo apt install mc -y
mc --version
```

Пример вывода:

```
GNU Midnight Commander 4.8.27
Built with GLib 2.68.4
Built with S-Lang 2.3.2 with terminfo database
```

## 2.7 Установка сервиса rutoll-debtweb

Файл дистрибутива: `rutoll-debtweb-vX.Y.Z.deb`.

Чтобы установить сервис веб-приложения rutoll-debtweb на сервер, требуется:

## 1. Установить дистрибутив и удалить его после установки:

```
sudo dpkg -i rutoll-debtweb-vX.Y.Z.deb
rm rutoll-debtweb-vX.Y.Z.deb
```

## 2. Скопировать пример конфигурации:

```
cp /etc/rutoll-debtweb/example/config.yml /etc/rutoll-debtweb/config.yml
```

3. Отредактировать файл, исходя из требуемых настроек (см. п.3.1 с описанием конфигурационных параметров).

4. Запустить и проверить службы:

```
sudo systemctl enable rutoll-debtweb
sudo systemctl start rutoll-debtweb
sudo systemctl status rutoll-debtweb
```

Пример успешного статуса:

```
● rutoll-debtweb.service - rutoll-debtweb service
 Loaded: loaded (/lib/systemd/system/rutoll-debtweb.service; enabled; vendor
 preset: enabled)
 Active: active (running) since Thu 2025-10-23 15:39:19 UTC; 5min ago
 Main PID: 10513 (rutoll-debtweb)
 Tasks: 13 (limit: 9159)
 Memory: 139.4M (max: 512.0M)
 CPU: 3.214s
```

## 2.8 Сервис rutoll-debt-listener

Для установки и настройки сервиса rutoll-debt-listener необходимо:

1. Установить пакет:

```
sudo dpkg -i rutoll-debt-listener-vX.Y.Z.deb
rm rutoll-debt-listener-vX.Y.Z.deb
```

2. Настроить конфигурацию, скопировав пример настроек:

```
cp /etc/rutoll-debt-listener/example/config.yml /etc/rutoll-debt-listener/config.yml
```

3. Отредактировать файл. Пример файла и описание параметров см. в п.3.2.

4. Запустить и проверить службы:

```
sudo systemctl enable rutoll-debt-listener
sudo systemctl start rutoll-debt-listener
sudo systemctl status rutoll-debt-listener
```

5. Пример успешного статуса:

```
● rutoll-debt-listener.service - rutoll-debt-listener service
 Loaded: loaded (/lib/systemd/system/rutoll-debt-listener.service; enabled;
 vendor preset: enabled)
 Active: active (running) since Thu 2025-10-23 15:27:10 UTC; 6min ago
 Main PID: 9620 (rutoll-debt-lis)
 Tasks: 13 (limit: 9159)
 Memory: 7.2M (max: 512.0M)
 CPU: 10.251s
```

## 2.9 Установка интернет-браузера

Чтобы установить на сервер интернет-браузер Google Chrome, следует:

1. Загрузить пакет, скачав актуальный установочный файл браузера:

```
wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
```

2. Установить пакет:

```
sudo dpkg -i google-chrome-stable_current_amd64.deb
```

3. Если во время установки появились зависимости — исправить их:

```
sudo apt -f install
```

4. Удалить установочный файл:

```
rm google-chrome-stable_current_amd64.deb
```

## 3 КОНФИГУРИРОВАНИЕ ПО «МОТИВАЦИЯ ОПЛАТЫ. МОДУЛЬ СВП-2+»

### 3.1 Конфигурирование сервиса rutoll-debtweb

Конфигурационный файл по умолчанию расположен в директории `/etc/rutoll-debtweb/example/config.yml`

#### 3.1.1 Общая конфигурация сервиса

В секции `service` описываются общие параметры.

Таблица 1. Параметры секции `service`

Параметр	Описание
<code>endpoint</code>	Адрес основного REST API сервиса (по умолчанию: нет)
<code>logging</code>	Настройки логгирования
<code>level</code>	Уровень логгирования: TRACE, DEBUG, INFO, WARN, ERROR (по умолчанию: INFO)
<code>color</code>	Цветной вывод логов в консоль (по умолчанию: false)

#### 3.1.2 Настройка документов

В секции `documents` указываются настройки внешнего вида документов.

Таблица 2. Параметры секции `documents`

Параметр	Описание
<code>debt_trip_report</code>	Отчёт по поездкам
<code>max_trip_count</code>	Максимальное количество детализаций с фото в одном отчёте (по умолчанию: 50)
<code>assets</code>	Логотип для отчёта (по умолчанию: нет)
<code>logo_url</code>	Адрес, где расположен логотип (например, <code>http://assets.local/static/logo.png</code> )
<code>template_dir</code>	Папка с HTML-шаблонами отчётов (по умолчанию: <code>document/report/trips</code> )
<code>layouts</code>	Шаблоны отчета (с QR-кодом и без)
<code>no_qr</code>	Версия отчёта без QR-кода
<code>with_qr</code>	Версия отчёта с QR-кодом
<code>id</code>	Идентификатор отчёта в шаблоне. По умолчанию:      <code>page_report_noqr</code> для версии без QR-кода,     <code>page_report_qr</code> для отчёта с QR-кодом.
<code>template</code>	Путь к HTML-шаблону внутри <code>~document/report/trips/pages~</code> . По умолчанию:      <code>report_noqr.html</code> для версии без QR-кода,     <code>page_report_qr</code> для отчёта с QR-кодом.
<code>render_timeout</code>	Время ожидания рендеринга отчёта (по умолчанию: 1h)

Параметр	Описание
table_rows	Настройка строк в таблице
first_page	Количество строк таблицы на первой странице (по умолчанию: нет)
other_pages	Количество строк таблицы на остальных страницах (по умолчанию: нет)
debt_event_summary	Сводный отчёт по событиям
template_path	Путь к XLSX-шаблону (по умолчанию: document/event/summary/summary.xlsx)

### 3.1.3 Настройка генерации QR-кодов

В секции `qr_code` указываются настройки генерации QR-кодов.

Таблица 3. Параметры секции `qr_code`

Параметр	Описание
payment	Настройки платёжного шлюза (по умолчанию: нет)
system	Платёжная система (по умолчанию: нет)
base_url	Базовый адрес платёжного шлюза (по умолчанию: нет)
target_lane	Путь для направления платежа (по умолчанию: нет)
https	Использовать HTTPS (по умолчанию: нет)
secret	Секрет для подписи запросов (по умолчанию: нет)

### 3.1.4 Настройка REST API

В секции `rest_api` настраивается взаимодействие клиента с сервером.

Таблица 4. Параметры секции `rest_api`

Параметр	Описание
host	IP, на котором слушает сервер (по умолчанию: 0.0.0.0)
port	Порт API (по умолчанию: 80)
timeouts	Настройки тайм-аутов
read	Тайм-аут чтения запросов (по умолчанию: 10m)
write	Тайм-аут записи ответов (по умолчанию: 10m)

### 3.1.5 Настройка БД

Взаимодействие с БД PostgreSQL настраивается в секции `database`.

Таблица 5. Параметры секции database

Параметр	Описание
host	Хост БД (по умолчанию: localhost)
port	Порт БД (по умолчанию: 5432)
user	Пользователь БД (по умолчанию: нет)
password	Пароль БД (по умолчанию: нет)
name	Имя БД (по умолчанию: нет)
ssl_mode	Режим SSL (по умолчанию: disable)
connect_timeout	Тайм-аут подключения (по умолчанию: 10m)
logging	Настройки логгирования
enable	Включить логирование БД (по умолчанию: false)
level	Уровень логирования БД: DEBUG, INFO, WARN, ERROR (по умолчанию: DEBUG)
pool	Настройка соединения
max_open	Максимум открытых соединений (по умолчанию: 50)
max_idle	Максимум неактивных соединений (по умолчанию: 10)
max_lifetime	Максимальное время жизни соединения (по умолчанию: 30m)
max_idle_time	Максимальное время простоя соединения (по умолчанию: 10m)

### 3.1.6 Настройка работы с хранилищем MiniO

Параметры для MiniO, где хранятся изображения, фотографии, шаблоны для отчетов, а также сами отчеты, указываются в секции `minio`.

Таблица 6. Параметры секции minio

Параметр	Описание
host	Хост MinIO (по умолчанию: localhost)
port	Порт MinIO (по умолчанию: 9000)
bucket	Бакет (по умолчанию: debt)
access_key	Ключ доступа (по умолчанию: нет)
secret_key	Секретный ключ (по умолчанию: нет)
ssl	Использовать SSL (по умолчанию: false)
rules	Правила автоудаления файлов
photo	Настройки для фотографий
prefix	Папка с фотографиями (по умолчанию: Reports/Photos/)
days	Срок хранения в днях (по умолчанию: 14)
status	Статус правила (по умолчанию: Enabled)
notification	Настройки для уведомлений
prefix	Папка с уведомлениями (по умолчанию: Notifications/)
days	Срок хранения в днях (по умолчанию: 1)
status	Статус правила (по умолчанию: Enabled)

### 3.1.7 Настройка очереди сообщений RabbitMQ

Взаимодействие с RabbitMQ настраивается в секции `rabbitmq`.

Таблица 7. Параметры секции `rabbitmq`

Параметр	Описание
host	Хост RabbitMQ (по умолчанию: localhost)
port	Порт RabbitMQ (по умолчанию: 5672)
vhost	VHost (по умолчанию: vh_main)
username	Имя пользователя (по умолчанию: нет)
password	Пароль (по умолчанию: нет)
queues	Настройка очереди
notify	Настройка сообщений
name	Имя очереди (по умолчанию: debt)
exchange	Имя обменника (по умолчанию: debt.notify)
routing_key	Ключ маршрутизации (по умолчанию: "#")

### 3.1.8 Настройка кэширования данных

В секции `cache` настраивается временное хранение часто запрашиваемых данных.

Таблица 8. Параметры секции `cache`

Параметр	Описание
type	Тип кэша: go-cache или redis (по умолчанию: redis)
redis	Настройки для redis
host	Хост Redis (по умолчанию: localhost)
port	Порт Redis (по умолчанию: 6379)
db	Индекс БД Redis (по умолчанию: 0)
password	Пароль Redis (по умолчанию: "")
timeouts	Настройка тайм-аутов
dial	Тайм-аут Dial (по умолчанию: 5s)
read	Тайм-аут Read (по умолчанию: 3s)
write	Тайм-аут Write (по умолчанию: 3s)
pool	Настройки пула
size	Размер пула (по умолчанию: 10)
min_idle	Минимум неактивных соединений (по умолчанию: 2)
idle_timeout	Тайм-аут простоя (по умолчанию: 5m)
gocache	Настройки для gocache
expiration	TTL (Time To Live, время жизни) записей в памяти (по умолчанию: 1h)
cleanup_interval	Интервал очистки устаревших записей (по умолчанию: 12h)

### 3.1.9 Настройка интеграции с внешним шлюзом

Взаимодействие с внешним шлюзом настраивается в секции `gateway`.

Таблица 9. Параметры секции `gateway`

Параметр	Описание
<code>base_url</code>	Базовый адрес шлюза (по умолчанию: нет)
<code>ssl</code>	Использовать SSL для шлюза (по умолчанию: false)
<code>token</code>	Настройки токена
<code>access_before</code>	За сколько времени обновлять токен (по умолчанию: 2m)
<code>credentials</code>	Учетные данные (по умолчанию: нет)
<code>username</code>	Имя пользователя
<code>password</code>	Пароль
<code>client_id</code>	Идентификатор клиента
<code>client_secret</code>	Секретное поле клиента
<code>routes</code>	Эндпоинты
<code>token</code>	Эндпоинт получения токена (по умолчанию: /auth/integration/oauth/token)
<code>trips_details</code>	Эндпоинт получения детализации поездок (по умолчанию: api/trips/details)
<code>photo_trips</code>	Эндпоинт получения файлов/фото (по умолчанию: api/files)

### 3.1.10 Swagger-документация

Секция `swagger` отвечает за настройки пути, авторизации и доступа к документации в Swagger.

Таблица 10. Параметры секции `swagger`

Параметр	Описание
<code>path</code>	Путь (по умолчанию: /swagger)
<code>enabled</code>	Включено (по умолчанию: true)
<code>auth_required</code>	Требуется авторизация (по умолчанию: true)

### 3.1.11 Метрики Prometheus

Секция `metrics` задает настройки пути, авторизации и доступа для метрик веб-приложения.

Таблица 11. Параметры секции metrics

Параметр	Описание
path	Путь (по умолчанию: /metrics)
enabled	Включено (по умолчанию: true)
auth_required	Требуется авторизация (по умолчанию: true)

### 3.1.12 Настройка WebSocket-соединения

Настройки WebSocket-соединения указываются в секции `websocket`.

Таблица 12. Параметры секции websocket

Параметр	Описание
ping_interval	Интервал пинга соединения (по умолчанию: 1s)
ping_deadline	Тайм-аут ожидания ответа (по умолчанию: 5s)

### 3.1.13 Настройка аутентификации и токенов

За настройки аутентификации и токенов отвечает секция `auth`.

Таблица 13. Параметры секции auth

Параметр	Описание
basic	Основные настройки
username	Имя пользователя basic-auth (по умолчанию: нет)
password	Пароль basic-auth (по умолчанию: нет)
password_salt	Соль для хеширования паролей (по умолчанию: нет)
tokens	Настройки токенов
access	Настройки доступа
ttl	Время жизни access-токена (по умолчанию: 12m)
secret	Секрет для подписи JWT (по умолчанию: нет)
refresh	Настройки обновления
ttl	Время жизни refresh-токена (по умолчанию: 24h)
secret	Секрет для подписи JWT (по умолчанию: нет)

## 3.2 Конфигурирование сервиса rutoll-debt-listener

Конфигурационный файл по умолчанию расположен в директории `/etc/rutoll-debt-listener/config.yml`.

### 3.2.1 Настройка работы с брокером сообщений

Взаимодействие с брокером сообщений Apache Kafka настраивается в секции `kafka-broker`.

Таблица 14. Параметры секции `kafka-broker`

Параметр	Описание
<code>url</code>	"доменное_имя_шлюза"
<code>consumer</code>	Параметры запросов
<code>limit</code>	Максимальное количество сообщений в одной партиции
<code>request-timeout</code>	Интервал между запросами
<code>auth</code>	Параметры авторизации
<code>username</code>	Имя пользователя
<code>password</code>	Пароль
<code>client-id</code>	Идентификатор клиента
<code>secret</code>	Секретное зашифрованное поле
<code>request-timeout</code>	Интервал между запросами

### 3.2.2 Настройка работы с БД

Взаимодействие с БД настраивается в секции `database`.

Таблица 15. Параметры секции `database`

Параметр	Описание
<code>username</code>	Имя пользователя
<code>password</code>	Пароль
<code>connection-string</code>	Строка подключения к базе данных. Формат: <code>protocol://address:port/database</code>
<code>maximum-open-connections</code>	Максимальное количество одновременных соединений с базой данных (например, 5)
<code>maximum-idle-connections</code>	Максимальное количество соединений, находящихся в состоянии ожидания (например, 10)
<code>maximum-connection-life-time</code>	Время жизни соединения (например, 3600s)
<code>maximum-connection-idle-time</code>	Время жизни соединения, находящегося в ожидании (например, 1800s)
<code>query-execute-timeout</code>	Тайм-аут запроса к базе данных. По истечении запрос будет отменен, соединение — закрыто (например, 3600s)

### 3.2.3 Настройка логгирования

В секции `log` указывается уровень логгирования для параметра `level`. Например:

```
level: DEBUG
```

## 4 ОСНОВНЫЕ ОПЕРАЦИИ ПРИ ЭКСПЛУАТАЦИИ

### 4.1 Обновление ПО «Мотивация оплаты. Модуль СВП-2+»

Для обновления АРМ необходимо обновить сервисы `rutoll-debtweb` и `rutoll-debt-listener`. Процесс обновления аналогичен.

Перед обновлением сервисов вначале следует останавливать службы, командой:

```
sudo systemctl stop rutoll-название-сервиса
```

Далее для обновления сервисов необходимо:

1. Установить новую версию пакета:

```
sudo dpkg -i rutoll-название-сервиса-vX.Y.Z.deb
rm rutoll-название-сервиса-vX.Y.Z.deb
```

2. Проверить корректность конфигурации:

```
sudo nano /etc/rutoll-название-сервиса/config.yml
```

3. Запустить службу и убедиться, что она работает корректно:

```
sudo systemctl start rutoll-название-сервиса
sudo systemctl status rutoll-название-сервиса
```

Пример успешного статуса:

```
• rutoll-debt-listener.service - rutoll-debt-listener service
 Loaded: loaded (/lib/systemd/system/rutoll-debt-listener.service; enabled;
 vendor preset: enabled)
 Active: active (running) since Thu 2025-10-23 15:27:10 UTC; 6min ago
 Main PID: 9620 (rutoll-debt-lis)
 Tasks: 13 (limit: 9159)
 Memory: 7.2M (max: 512.0M)
 CPU: 10.251s
```

### 4.2 Логирование ПО «Мотивация оплаты. Модуль СВП-2+»

Логи сервисов `rutoll-debtweb` и `rutoll-debt-listener` записываются в каталог `/var/log/rutoll-название-сервиса/`.

Каждый день в 00:00 выполняется ротация логов — текущий файл `rutoll-название-сервиса.log` архивируется в формате: `rutoll-название-сервиса.log-[YYYYMMdd].gz`.

Например:

```
rutoll-debtweb.log-20251002.gz
```

Для просмотра актуальных логов в реальном времени следует использовать команду:

```
sudo tail -f /var/log/rutoll-название-сервиса/rutoll-название-сервиса.log
```

Для открытия архивированных логов за конкретную дату следует:

1. Распаковать архив во временную директорию:

```
sudo gzip -dkc /var/log/rutoll-название-сервиса/rutoll-название-сервиса.log-
20251002.gz > /tmp/rutoll-название-сервиса.log-20251002
```

2. Просмотреть содержимое:

```
less /tmp/rutoll-название-сервиса.log-20251002
```