

2. Интеграция с окружением.

1. По-умолчанию, система готова к работе сразу после старта.

В большинстве случаев использования, администратору и пользователям системы только нужен доступ к web-интерфейсу и API.

Конфигурации баз данных оборудования, различных CRM-систем, почтовых шлюзов, ПО автоматической рассылки настраиваются администратором системы через web-интерфейс.

Администрирование и настройка сторонних сервисов и систем не входит в состав sms-router. Для более детальной конфигурации воспользуйтесь документацией и техподдержкой соответствующего ПО.

2. Если предполагается использование через файловую систему сервера:

Смонтируйте локальный каталог или диск в виртуальное окружение docker. Для этого, добавьте в конфигурационный файл docker-compose.yml в секции routines-file строку, помеченную красным:

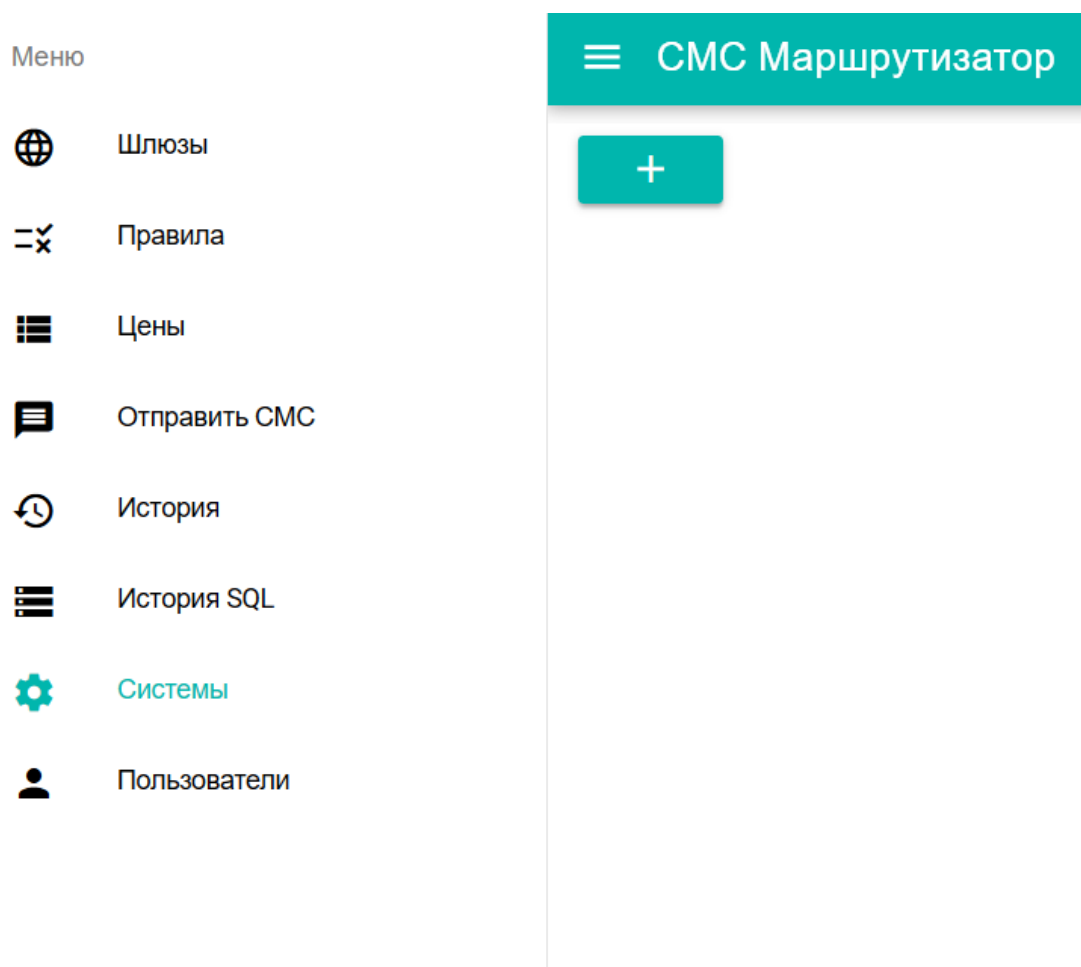
```
routines-file:
  image: <cr-wifi-click>/routines-file:${routines-file_version}
  expose:
    - 4512
  volumes:
    - ./config/production.json:/usr/app/config/production.json
    - ./<Путь до вашего каталога>:/usr/app/routines/file/messages
  ...
```

Пересоздайте контейнер с новыми параметрами:

```
docker compose up -d routines-file
```

3. Управление через веб-интерфейс.

Веб-интерфейс состоит из бокового меню и основной страницы приложения.



4.1. Системы

«Системой» в смс-маршрутизаторе называется точка входа в шлюз. Она реализована только для получения запросов от клиентов на отправку сообщений.

На данный момент «система» реализована в 5-ти видах:

1. Встроенный web-server, с различными API-ключами для разных клиентов
2. PDU-receiver для получения клиентских сообщений по SMPP-протоколу
3. Интеграция со сторонней SQL-базой для приема и обработки сообщений
4. SMTP-сервер для приема e-mail через почтовые приложения и клиенты
5. Оправка сообщений через файловую систему сервера

Файловые и sql-системы подхватываются «на лету», без перезагрузки и рестарта сервиса.

4.1.1. Web-сервер с REST-API.

Новая система

Имя системы *
Новая система

API Key
472cdf49-b662-49ee-a659-ec9361a46ed0

+ SMPP

+ БАЗА ДАННЫХ

+ EMAIL

+ ФАЙЛ

[ОТМЕНА](#) [СОХРАНИТЬ](#)

Для создания новой системы необходим API-ключ. По-умолчанию, он генерируется сервисом по стандарту UUID v4, но при желании может быть заменен любой строкой.

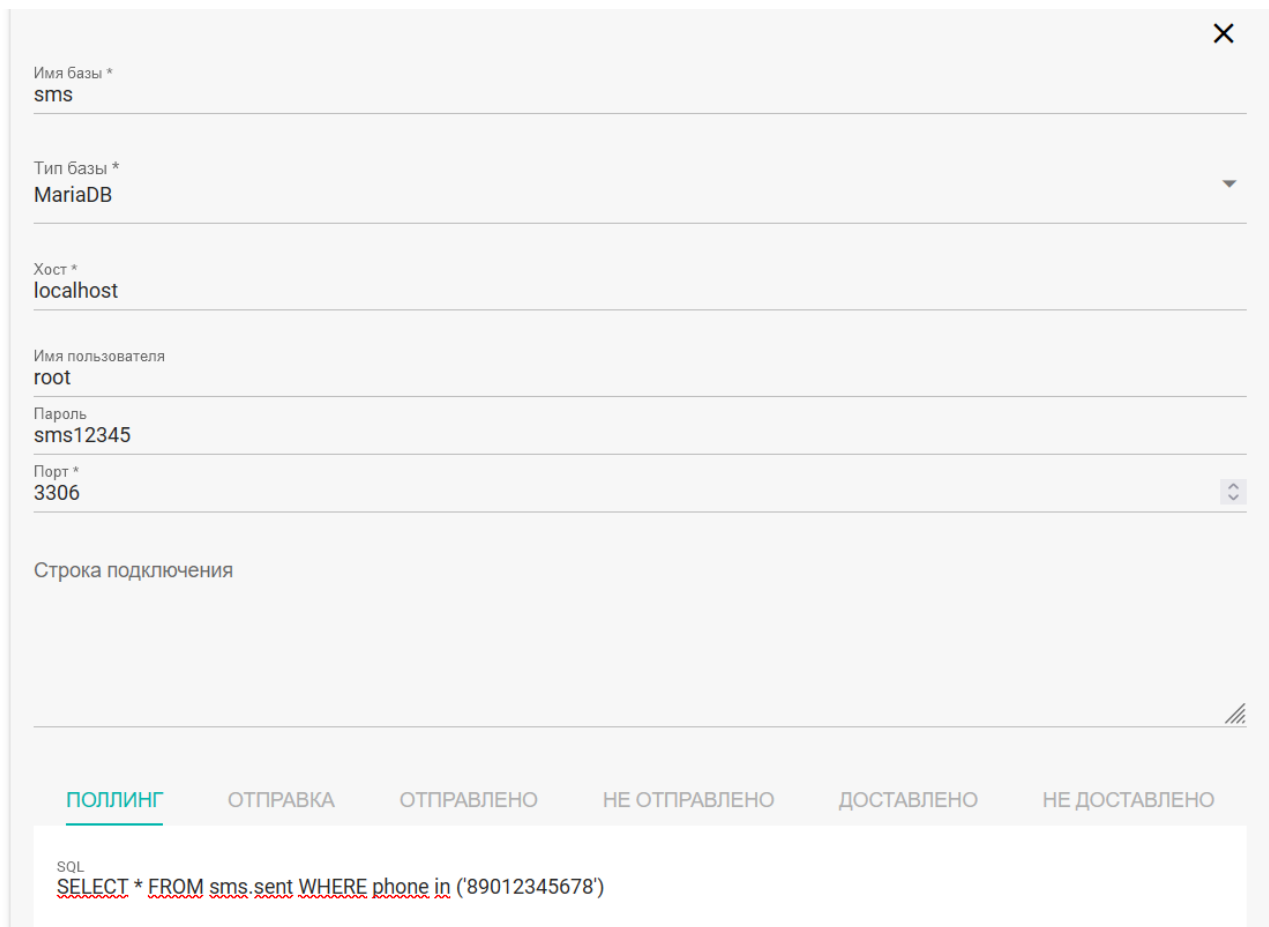
4.1.2. SMPP

Необходим логин/пароль от SMPP-шлюза.

4.1.3. Email

Необходим email-адрес клиента вашего почтового сервера, для приема сообщений через почтовые клиенты и приложения.

4.1.4. База Данных.



Имя базы *
sms

Тип базы *
MariaDB

Хост *
localhost

Имя пользователя
root

Пароль
sms12345

Порт *
3306

Строка подключения

ПОЛЛИНГ ОТПРАВКА ОТПРАВЛЕНО НЕ ОТПРАВЛЕНО ДОСТАВЛЕНО НЕ ДОСТАВЛЕНО

SQL
`SELECT * FROM sms.sent WHERE phone in ('89012345678')`

(Пример настройки SQL-системы)

«Поллинг, Отправка, Отправлено, Не Отправлено, Доставлено, Не Доставлено» - являются SQL-запросами к БД вашего приложения или оборудования, которые выполняет сервис при перечисленных событиях соответственно.

(На примере выше показано, что при поллинге БД, система выбирает новые сообщения из коллекции sent от номера 89012345678. Опрос бд на наличие новых записей выполняется каждые 5 секунд по-умолчанию)

Настройка SQL-запросов производится системным администратором в соответствии с устройством и разметкой данных вашей БД.

! При составлении SQL-запросов избегайте циклического выполнения и SQL-инъекций. Используйте одинарные кавычки (').

4.1.5. Файловая система

Система, подключенная для чтения с дисковых устройств, должна быть разбита на каталоги. Файлы помещаемые в каталог './out', будут обработаны смс-шлюзом и помещены в соответствующие каталоги на сервере по результатам их обработки.

✕

Исходящие

../имя_папки%/out

Отправленные

../имя_папки%/sent

Не отправленные

../имя_папки%/notsent

Доставленные

../имя_папки%/delivered

- [+ SMPP](#)
- [+ БАЗА ДАННЫХ](#)
- [+ EMAIL](#)
- [+ ФАЙЛ](#)

[ОТМЕНА](#) [СОХРАНИТЬ](#)

4.2. Шлюзы

«Шлюзом» в sms-router называется точка выхода из маршрутизатора после получения реквеста из системы на отправку сообщения, пост-обработки и разрешения на отправку. Шлюзы управляют конфигурацией API смс-провайдеров.

На данный момент шлюзы реализованы в 2-ух типах:

1. HTTP (поддерживает методы GET и POST)
2. SMPP (поддерживает кастомные параметры при подключении к шлюзу)

Возможна также обратная маршрутизация, которая позволяет записывать в базу и на диск входящие сообщения. Функционал будет доступен в следующей версии ПО.

4.3. Правила

Правила представляют собой список фильтрации (номерные пулы, текст и время отправки) и конечных действий (отклонить/отправить) формируемый для запросов из конкретной системы.

! Чтобы связать систему со шлюзом напрямую, необходимо создать хотя бы одно правило.

Более подробная информация описана в интерфейсе системы.

4.4. Отправить СМС.

☰ СМС Маршрутизатор

Тестовый режим

Система: wi-fi.click | Шлюз: tele2 / HTTP

Текст сообщения: Ваш код активации: 1234 | Список номеров: +79050003300

79050003300

ОТПРАВИТЬ

Страница отладки и тестирования работоспособности систем, правил и шлюзов. Чек-бокс «Тестовый режим» позволяет создать и проверить аri-реквест без отправки конечному получателю.

4.5. История/История SQL.

Таблица с пагинацией запросов к вашим системам в хронологическом порядке. Содержит статус запроса, время/дату, номер телефона, сообщение, примененное правило и имя шлюза, через который было отправлено сообщение.

Статусы запросов:

- ok – сообщение успешно направлено в соответствующий шлюз
- error – ошибка отправки сообщения
- rejected – сообщение отклонено sms-router
- await – запрос находится в процессе обработки маршрутизатором

Поиск по запросам, сообщениям, номерам телефонов, фильтрация по шлюзам и системам, находятся в стадии разработки.

4.6. Пользователи

Интерфейс для управления пользователями СМС-шлюза.

Можно добавлять/удалять менять пароль и отключать учетные записи.

Пользователь по-умолчанию: admin@admin. Пароль: 12345678.

У всех пользователей одинаковые права на управление системой.

4. Описание API. Примеры использования.

5.1. HTTP.

Web-server обрабатывает GET и POST запросы по адресу:

`http(s)://<ваш ip-адрес | домен>(:3000)/api/sendsms`

Подробный пример для GET запроса можно посмотреть в веб-интерфейсе.
POST запрос поддерживает JSON и x-www-form-urlencoded.

Входные данные (обязательные поля **выделены красным**):

```
{
  "phone": <+79xxxxxxxx>,
  "api_key": <uuid> (Строка, генерируется пользователем в веб-приложении, см. 4.1.1.),
  "message": <String> (Строка в UTF-8, 70 символов максимум),
  "test": <Boolean> (default: false. Если true, то запрос обрабатывается в тестовом режиме),
  "gateway": <ObjectId> (default: null. Если отправить id шлюза, то запрос направится на него,
                                                                минуя блок правил)
}
```

Ответ (в формате json):

```
{
  "status": <ok | error | rejected>, (статусы текущего реквеста)
  "req_id": <ObjectId>, (генерируемый id)
  "subsystem": "New caller", (имя подсистемы)
  "ms": 124,
  "time": "2023-12-05T15:12:13.525Z", (текущее время)
  "gatewayName": "gateWay 1" (имя шлюза)
}
```

Примеры запросов:

```
curl -X POST https://<domain>/api/sendsms
  -H 'Content-Type: application/json'
  -d '{"phone": "79123456789", "api_key": "123e4567-e89b...", "message": "Код: 1234"}'
```

Либо:

```
curl
  https://localhost:3000/api/sendsms?
phone=79123456789&api\_key=1234567&test=true
```


5.2. SQL

Для работы с SQL, потребуются несколько таблиц (out, sent, notsent, и т.д.) со следующими обязательными колонками:

status: **STRING(20)**
receiver: **STRING(30)**
msg: **STRING(160)**
msgtype: **STRING(45)**

Пример запроса:

```
INSERT INTO sms_out (status, receiver, msg, msgtype)
VALUES ('start', '89001112233', 'код: 1234', 'test');
```

Сообщение попадёт в поллинг системы БД, и запустится следующий SQL- сценарий в цепочке. (Подробнее: см п. 4.1.4 и веб-интерфейс «Системы»)

5.3. Файловая система.

Поместить в каталог, помеченную на отправку (/out) XML-файл такого типа:

```
<messages>
<message>
<text>Ваш код: 1234</text>
<recipient>
+79800001122
</recipient>
</message>
<message>
<text>test</text>
<recipient>
+79800001122
</recipient>
<messageType>test</messageType>
</message>
</messages>
```

Система прочтет файл, попытается его отправить, если успешно, то поместит в директорию (/sent), если возникла ошибка, то в директорию (/notsent)

5.4. Email (smtp)

Для отправки e-mail сообщений на шлюз, потребуется почтовый клиент (Outlook, Mozilla Thunderbird, Gmail, Mutt).

В параметрах, нужно добавить новую учетную запись.

Имя учетной записи должно совпадать с именем указанном в системах (пункт 4.1.3)

Сервер исходящей почты: SMTP

Адрес: ваш ip/домен

Порт: 8200

При отправке сообщения в поле «Кому» указать <79XXXXXXXXXX>@<ваш домен>

В тексте письма написать сообщение для отправки.