SIGUR

Руководство по настройке системы для использования функционала REST API

Редакция от 13.03.2025.

«Промавтоматика», г. Н. Новгород, 2025 г.

Оглавление

1. Введение

Данный документ содержит общее описание интеграционных возможностей системы с использованием REST API и описание процесса первичной настройки программного обеспечения Sigur.

Представленная в данном документе информация соответствует функциональности ПО Sigur версии 1.6.4.53.

Руководство по установке и настройке системы Sigur можно найти в отдельных документах – «Руководстве администратора ПО Sigur» и «Руководстве пользователя ПО Sigur».

Подробное описание запросов REST API представлено в «Руководстве разработчика по REST API Sigur».

2. Версии документа

Данный документ имеет следующую историю ревизий.

Ревизия	Дата публикации	Что изменилось
0001	11 декабря 2023 г.	Соответствует версии ПО 1.6.0.1. Частично основан на документе для ПО Sigur версии 1.2.0.8. Изменения: обновлен порядок настройки системы, обновлена информация о процессе авторизации и эндпоинтах группы Bindings в «Руководстве разработчика по REST API Sigur».
0002	27 декабря 2023 г.	Добавлена информация о предыдущем эндпоинте авторизации api/v1/application-keys/auth в «Руководство разработчика по REST API Sigur».
0003	13 марта 2024 г.	Изменения в «Руководстве разработчика по REST API Sigur»: • Добавлено описание новых эндпоинтов для работы с операторами системы. • Актуализировано содержимое запросов и ответов системы при работе с зонами. • Добавлен корректный пример использования фильтра tabId[operation] в запросе к api/v1/employees. • Актуализировано описание полей, передаваемых при использовании предыдущего эндпоинта авторизации api/v1/application-keys/auth.

Ревизия	Дата публикации	Что изменилось
0004	4 сентября 2024 г.	Добавлена информация об актуальном способе управления автотранспортом с помощью REST API Sigur. Изменения в «Руководстве разработчика по REST API Sigur»: • Добавлено описание новых эндпоинтов для управления допуском сотрудников и служебных автомобилей на точки доступа. • Эндпоинты для управления личными автомобилями теперь отмечены как устаревшие, пользоваться ими на текущий момент не рекомендуется. • Актуализирована информация об использовании параметров limit и offset для GET-запросов. • Добавлено пояснение о работе с полем "name" в запросе к /api/v1/cards/update.
0005	10 января 2025 г.	Добавлена информация о работе с фильтрами startTime и endTime для эндпоинтов api/v1/events и api/v1/events/parsed в «Руководство разработчика по REST API Sigur».
0006	13 марта 2025 г.	Актуализирована информация о присвоении карт сотрудникам средствами REST API в разделе «Bindings» – «Employees-Cards» – «Assign Cards to Employees» «Руководства разработчика по REST API Sigur».

3. Используемые определения, обозначения и сокращения

СКУД	Система контроля и управления доступом. Программно–аппаратный комплекс, предназначенный для осуществления функций контроля и управления доступом.
Точка доступа	Место, где осуществляется контроль доступа. Например: дверь, турникет, ворота, шлагбаум, оборудованные считывателем, электромеханическим замком и другими необходимыми средствами.
Объект доступа	Сотрудник, посетитель, автомобиль или другое транспортное средство, действия которого регламентируются правилами разграничения доступа.
ПО	Программное обеспечение.
БД	База данных.

4. Системные требования

Рекомендуется руководствоваться конфигурацией сервера, описанной в разделе «Системные требования СКУД Sigur» «Руководства администратора ПО Sigur».

5. Общее описание порядка взаимодействия

Сервер Sigur обеспечивает возможность интеграционного взаимодействия посредством RESTful интерфейса. Данный интерфейс обеспечивается веб-сервером Sigur, с которым сторонние системы могут настроить взаимодействие посредством HTTP(S)-запросов.

REST интерфейс позволяет читать данные базы Sigur, изменять их, создавать новые объекты данных и удалять существующие. На текущий момент доступны следующие возможности:

- Получение списка отделов и информации об их количестве, создание новых, редактирование и удаление существующих.
- Получение списка должностей сотрудников, создание новых, редактирование и удаление существующих.
- Получение списка карт доступа, создание новых, редактирование и удаление существующих.
- Получение списка сотрудников и информации об их количестве, создание новых, редактирование и удаление существующих, блокировка и разблокировка доступа.
- Получение списка служебного автотранспорта, создание нового, редактирование и удаление существующего.
- Получение списка операторов системы, назначение прав оператора существующим сотрудникам, редактирование и удаление назначенных ранее прав.
- Получение списка дополнительных параметров, создание новых, редактирование и удаление существующих.
- Получение списка точек доступа и информации об их количестве.
- Получение списка режимов доступа и информации об их количестве, создание новых, редактирование и удаление существующих.
- Получение списка зон доступа.
- Просмотр расширенной информации по объектам доступа (выданные карты, предоставленный доступ к точкам доступа, назначенные режимы), назначение таких связей и удаление существующих.
- Получение списка кодов событий, существующих в системе, и получение самих событий.

С подробным описанием запросов разработчик интеграционного решения может ознакомиться, перейдя на ресурс *http://<server>:9500/swagger*, где *<server>* - это сетевой адрес компьютера, на котором установлен сервер Sigur.

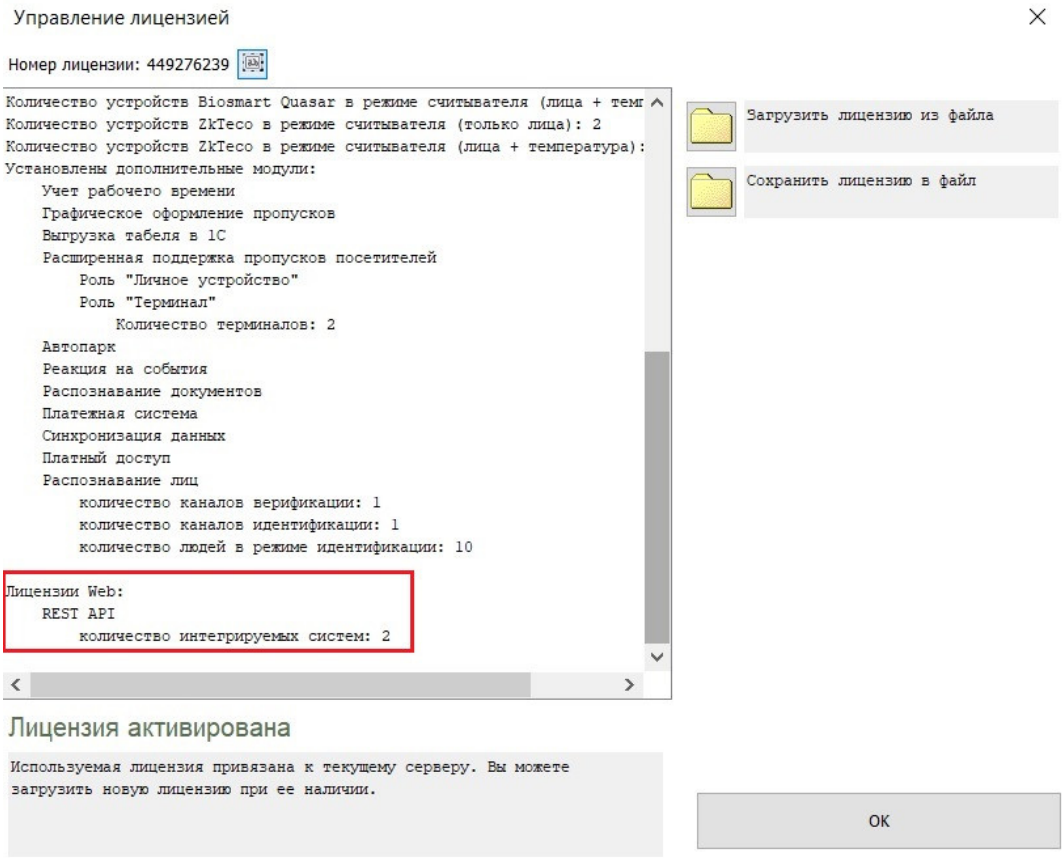
Описание запросов также доступно в «Руководстве разработчика по REST API Sigur».

В ответ на GET-запросы по умолчанию возвращается 50 записей. Для конфигурирования количества возвращаемых записей предназначен параметр запроса limit. Получение большого количества записей рекомендуется осуществлять итерационно, сочетая параметры limit и offset. Параметр offset определяет количество записей с начала, которые должны быть пропущены в ответе.

6. Настройка на стороне Sigur

6.1. Основные настройки

Для организации успешного общения с сервером Sigur по REST API необходимо предварительно загрузить на сервер дополнительный лицензионный модуль «Лицензии Web: REST API». Процесс загрузки лицензии описан в разделе «Лицензирование функционала ПО» «Руководства пользователя ПО Sigur». Функционал лицензируется по количеству интегрируемых систем.

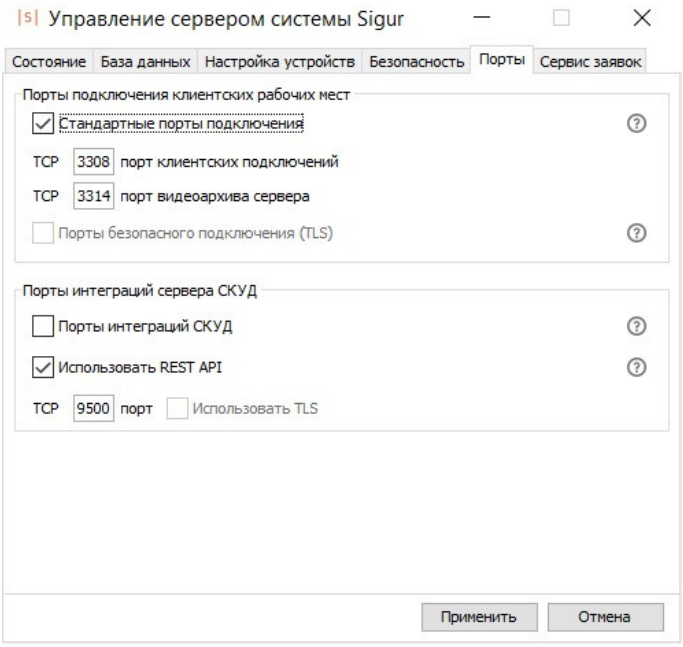


Окно «Управление лицензией».

В случае использования программной лицензии, после загрузки файла необходимо перезапустить ПО «Клиент». Также после изменения состава лицензии автоматически будет перезагружен серверный модуль Sigur.

Для активации порта веб-сервера необходимо перейти на вкладку «Порты» ПО «Управление сервером» и включить чекбокс «Использовать REST API» в блоке «Порты интеграций сервера СКУД». По умолчанию для подключения в защищенном и незащищенном режиме к веб-серверу СКУД используется порт TCP 9500. Вы можете использовать порт по умолчанию или изменить это значение.

Для сохранения настроек нажмите кнопку «Применить» и перезагрузите серверный модуль.



Вкладка «Порты» ПО «Управление сервером».

Далее требуется создать реквизиты для авторизации на веб-сервере Sigur. Для этого необходимо:

1. Предоставить какому-либо сотруднику на вкладке «Персонал» права оператора (процесс создания оператора описан в разделе «Операторы системы» «Руководства пользователя ПО Sigur»).
2. Активировать чекбокс «Доступ по REST API (Интеграции)» в списке прав оператора.
3. Задать логин и пароль оператора, от имени которого внешний сервис будет производить авторизацию. Логин оператора может содержать цифры и буквы латинской или русской раскладки.
4. Сохранить изменения, нажав кнопку «Применить».

Система, авторизовавшаяся на веб-сервере с данными реквизитами, имеет доступ к использованию всех запросов, перечисленных в «Руководстве разработчика по REST API Sigur».

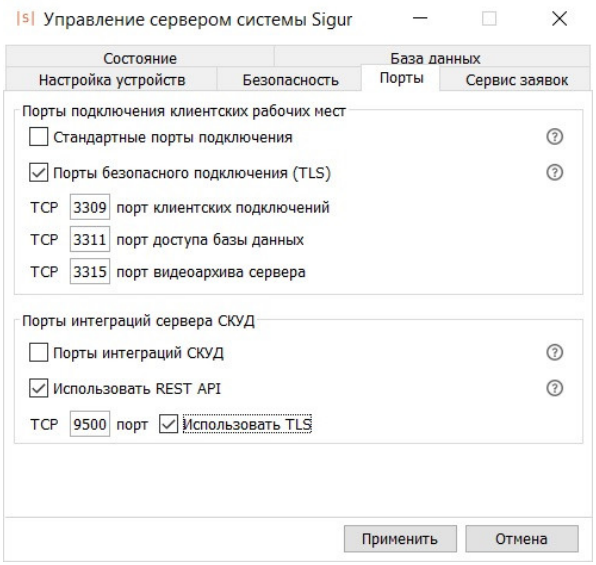
6.2. Шифрование трафика по TLS

Вы можете активировать шифрование трафика по протоколу TLS между веб-сервером Sigur и сторонними интеграционными сервисами.

По умолчанию шифрование отключено, взаимодействие с веб-сервером осуществляется по протоколу HTTP.

Перед активацией шифрования необходимо настроить хранилище сертификатов формата PKCS#12 на сервере СКУД (подробнее – в разделе «Установка зашифрованного соединения между клиентом и сервером» «Руководства администратора ПО Sigur»).

На вкладке «Порты» ПО «Управление сервером» необходимо включить чекбокс «Использовать TLS» в разделе «Порты интеграций сервера СКУД», нажать кнопку «Применить» и перезапустить серверный модуль. После этого внешнее подключение на порт веб-сервера Sigur будет возможно только по протоколу HTTPS.



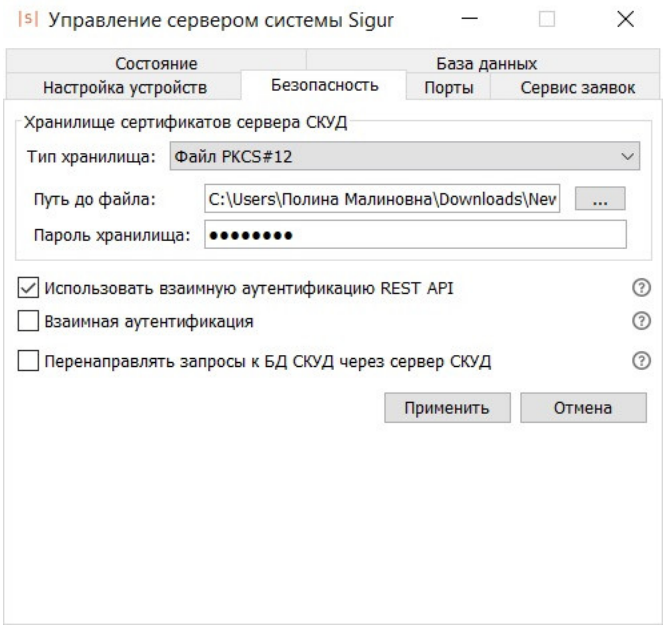
Вкладка «Порты» ПО «Управление сервером».

На стороне внешней системы, подключающейся на порт веб-сервера с использованием TLS, должно быть сконфигурировано хранилище доверенных сертификатов для валидации сертификата сервера Sigur. В противном случае внешняя система не сможет подключиться к веб-серверу Sigur.

Настройка взаимной аутентификации и списка отозванных сертификатов

Опционально вы можете активировать взаимную аутентификацию при подключении на порт веб-сервера. В этом случае внешняя система, пытающаяся установить защищенное соединение, также должна предоставить свой сертификат безопасности.

Для этого необходимо произвести предварительную настройку веб-сервера согласно описанию выше и затем активировать чекбокс «Использовать взаимную аутентификацию REST API» на вкладке «Безопасность» ПО «Управление сервером». После сохранения настроек необходимо перезапустить серверный модуль. Сертификат внешней системы должен быть подписан доверенным корневым центром сертификации или промежуточным центром в цепочке доверия сервера Sigur.



Вкладка «Безопасность» ПО «Управление сервером».

Если ранее в системе уже был задан список отозванных сертификатов, он также будет использоваться для проверки статуса сертификатов внешних систем, подключающихся на порт веб-сервера Sigur. Подробнее о настройке списка отозванных сертификатов – в разделе «Проверка статуса отзыва сертификата» «Руководства администратора ПО Sigur».

Хранилище сертификатов сервера и список отозванных сертификатов являются общими для всей системы Sigur.

7. Управление автотранспортом в Sigur

Система Sigur позволяет управлять как личным, так и служебным автотранспортом с помощью ПО «Клиент» и REST API.

7.1. Особенности текущей реализации

В настоящее время существуют два рекомендуемых способа администрирования данных, которые описаны ниже. Выбор способа зависит от конкретной задачи.

1. Автомобиль выступает в качестве дополнительного параметра сотрудника.

Такой способ актуален для тех задач, когда для предоставления доступа персонал может использовать не только идентификаторы типа карт/смартфонов/браслетов и пр., но также и идентификаторы автотранспорта (радиометка/брелок/гос. номер). При этом предполагается, что идентификатор автомобиля постоянно закреплён за конкретным человеком, то есть автомобиль является личным.

Для этого в учётную карточку сотрудника добавляется идентификатор автомобиля. В зависимости от типа идентификации это будет либо дополнительный номер пропуска (актуально для радиобрелоков, радиометок), либо специальный дополнительный параметр, в котором указывается гос. номер. При проезде автотранспорта система будет регистрировать событие прохода сотрудника.

При взаимодействии через REST API можно управлять идентификаторами автомобилей с помощью эндпоинтов групп Cards (радиобрелоки, радиометки) и Custom fields (гос. номера). Добавить идентификаторы автомобилей в учётные карточки сотрудников и назначить персоналу права доступа можно с использованием эндпоинтов групп Bindings и Employees.

Система также позволяет создавать отдельные объекты доступа типа «Личный автомобиль» (Employee vehicle). Однако в настоящее время не рекомендуется управлять личными автомобилями таким методом. Вместо этого рекомендуется использовать указанные выше способы, такие как добавление кода пропуска или доп. параметра в карточку сотрудника.

2. Автомобиль является отдельной сущностью в рамках системы, в конкретный момент времени он может быть связан с тем или иным сотрудником.

Чаще всего такая схема применима для организации работы со служебным автотранспортом и путевыми листами. Для использования в Sigur подобной функциональности необходим программный модуль «Автопарк».

Принцип работы состоит в том, что отдельные сущности типа «Служебный автомобиль» временно ассоциируются с ранее созданными сотрудниками с помощью функции администрирования путевых листов, предусмотренной в модуле «Автопарк». В системе создаётся «Служебный автомобиль» и ему присваивается либо Wiegand-идентификатор, либо гос. номер (в зависимости от способа идентификации автотранспорта). При попытке идентификации система учитывает права доступа автомобиля, а не сотрудника, который связан с ним на этот момент. Проезд транспорта фиксируется как два события: проход сотрудника и проход, совершенный объектом «Служебный автомобиль».

Через REST API можно создавать, редактировать и удалять служебные автомобили с помощью эндпоинтов группы Official Vehicles, а управлять их правами доступа - с помощью эндпоинтов группы Bindings.

Для администрирования через REST API доступен следующий набор атрибутов объектов «Служебный автомобиль» (Official vehicle):

- ID.
- Гос. номер (помещается в штатное поле «Гос. номер» в карточке автомобиля на вкладке «Персонал»).
- Отдел, к которому относится объект.
- Произвольные дополнительные параметры. При создании дополнительных параметров для автотранспорта в ПО «Клиент» нужно установить опцию «Применимо к автомобилям», а при создании через REST API - убедиться, что значение параметра "entity" содержит "VEHICLES".

Управление другими информационными атрибутами автомобилей (модель, примечание и пр.) производится в интерфейсе ПО «Клиент».

В настоящее время невозможно связывать служебные автомобили с персоналом через REST API для открытия и закрытия путевых листов. Если в системе есть открытый путевой лист, то управление этим сотрудником и служебным автомобилем через REST API также будет ограничено (например, удаление таких сущностей невозможно). Для управления путевыми листами и другими функциями модуля «Автопарк» следует использовать интерфейс ПО «Клиент».

7.2. Практические примеры настройки

Рассмотрим следующие ситуации:

1. Необходимо разрешить доступ на предприятие сотруднику по распознаванию RFID-метки или гос. номера его личного автомобиля.

Для решения этой задачи можно выполнить следующие действия:

- При необходимости распознавания RFID-метки:
  - Создать новый идентификатор с помощью запроса POST api/v1/cards.
  - Добавить Wiegand-код метки в учётную карточку сотрудника с помощью запроса POST api/v1/bindings/employees-cards.
- При необходимости распознавания гос. номера:
  - Создать дополнительный параметр «Гос. номер» для сотрудников (в англоязычной локализации - LP number). Если планируется добавлять некоторым сотрудникам несколько гос. номеров, то нужно создать параметр «Гос. номера» (LP numbers). В него можно внести как один, так и несколько номеров, перечисленных через запятую.

    Параметр можно создать с помощью запроса POST api/v1/custom/fields. При этом название параметра нужно помещать в атрибут "description", а значение параметра "entity" должно содержать "EMPLOYEES".
  - Создать или отредактировать данные сотрудника, указав в доп. параметре гос. номер автомобиля. Для выполнения этой операции через REST API используются эндпоинты группы Employees. В Sigur необходимо добавлять гос. номера в том же формате, в котором их отправляет внешняя видеосистема (например, Trassir отправляет гос. номера в SIgur на латинице).

    Важно, чтобы в системе отсутствовали самостоятельные объекты доступа типа «Личный/Служебный автомобиль» с такими же гос. номерами. В таком случае система расценит факт распознавания как попытку доступа самостоятельного объекта-автомобиля, а не сотрудника с таким же гос. номером в доп. параметре.
- Далее нужно создать режим доступа с помощью запроса POST api/v1/accessrules. По умолчанию новый режим разрешает доступ по распознаванию гос. номеров на вход и на выход. При необходимости правила режима можно отредактировать через интерфейс ПО «Клиент».

- Привязать режим к сотруднику и нужной точке доступа с помощью запросов POST api/v1/bindings/employees-accessrules и POST api/v1/bindings/accessrules-accesspoints.

- Предоставить сотруднику доступ на точку доступа с помощью запроса POST api/v1/bindings/employees-accesspoints.

2. Необходимо добавить в систему служебную машину, которой поочерёдно пользуются несколько сотрудников.

В этом случае можно выполнить следующие настройки:

- Создать объект типа Official vehicle (соответствует типу «Служебный автомобиль» в ПО «Клиент») с помощью запроса POST api/v1/vehicles/official.

- При необходимости распознавания RFID-метки - создать новый идентификатор с помощью запроса POST api/v1/cards и присвоить его служебному автомобилю с помощью запроса POST api/v1/bindings/vehicles-cards.

- Создать режим доступа с помощью запроса POST api/v1/accessrules. По умолчанию новый режим разрешает доступ по распознаванию гос. номеров на вход и на выход. При необходимости правила режима можно отредактировать через интерфейс ПО «Клиент».

- Присвоить созданный режим служебному автомобилю и точкам доступа запросами POST api/v1/bindings/vehicles-accessrules и POST api/v1/bindings/accessrules-accesspoints.

- Предоставить служебному автомобилю доступ на нужные точки доступа с помощью запроса POST api/v1/bindings/vehicles-accesspoints.

Открытие/закрытие путевых листов и иные действия, связанные с модулем «Автопарк», выполняются в ПО «Клиент».

8. Контакты

ООО «Промышленная автоматика – контроль доступа»
Адрес: 603001, Нижний Новгород, ул. Керченская, д. 13, 4 этаж.

Система контроля и управления доступом «Sigur»
Сайт: www.sigur.com
По общим вопросам: info@sigur.com
Техническая поддержка: support@sigur.com
Телефон: +7 (800) 700 31 83, +7 (495) 665 30 48, +7 (831) 260 12 93