

Клиент модуля интеграции Орион Икс

Инструкция



Данный документ описывает архитектуру и технологии, используемые для взаимодействия между клиентом и модулем интеграции Орион Икс. Протоколом для обмена сообщениями является JSON-RPC 2.0, который может передаваться через различные протоколы, включая HTTP(S) и WebSocket(S).

Понимание этих технологий поможет разработчикам эффективно использовать и интегрироваться с предоставляемым API.

Использованные технологии:

ReactJs – библиотека для создания пользовательских интерфейсов, особенно SPA (одностраничных приложений).

TypeScript – надстройка над JavaScript, добавляющая статическую типизацию для надёжности кода.

TanstackQuery (или **React Query**) – библиотека для управления состоянием серверных данных, кэширования и синхронизации.

ShadcnUi – коллекция готовых UI-компонентов для быстрой сборки интерфейсов.

Tailwindcss – CSS-фреймворк с готовыми CSS-классами, позволяющий быстро стилизовать компоненты.

Vite – быстрый инструмент для сборки frontend-приложений.

Vitest – инструмент для тестирования frontend-приложений.

Zustand – легковесная, современная библиотека для управления состоянием (state management) в приложениях на React.

Zod – библиотека для валидации данных.

Содержание

Глава 1. Начало работы.....	5
1.1 Тестирование	5
Глава 2. Логика работы.....	6
2.1 Авторизация в системе	7
2.2 Получение элементов	8
2.3 Изменение элементов	10
2.4 Связи между элементами	10
2.5 Поиск.....	12
2.6 События.....	13
2.7 Работа с WebSocket.....	13
2.8 Запросы общей информации	14
Глава 3. Работа с SSL-сертификатами	15
Глава 4. Класс jsonrpc-websockets для взаимодействия с сервером.....	17
Глава 5. Запросы данных	18
get_computers (получение списка идентификаторов компьютеров).....	18
get_elements (получение информации об элементах указанного типа)	19
get_computer_elements (получение информации об элементах указанного типа для идентификатора компьютера)	20
get_element (получение информации об элементе)	21
get_elements_by_link (получение информации об элементах, связанных с передаваемым)..	22
get_incident_types (получение списка типов инцидентов)	24
get_message_groups (получение списка групп сообщений)	25
get_message_desc (получение описания сообщений)	26
find_elements_by_param_i (поиск элементов по целочисленному значению)	27
find_elements_by_param_s (поиск элементов по строковому значению).....	29
find_elements_by_any_param_s (поиск элементов по строковому значению любого параметра).....	31
find_elements_by_name (поиск элементов по имени)	33
find_elements_by_additional_param_s (поиск пользователя по ключу)	34
types_by_tag (получение списка типов для прототипа).....	36
get_events (получение истории событий)	37
СОЗДАНИЕ, УДАЛЕНИЕ, РЕДАКТИРОВАНИЕ	39
create_element (создание элемента).....	39
create_element_by_link (создание дочернего элемента)	40
delete_element (удаление элемента)	41
update_element (изменение элемента)	42
create_link (создание связи между элементами)	43
delete_link (удаление связи между элементами).....	44

УПРАВЛЕНИЕ	45
get_possible_commands_by_element (получение списка возможных команд).....	45
send_command_query_for_element (отправка запроса на выполнение команды)	46

Глава 1. Начало работы

Перед установкой и запуском проекта необходимо установить программу NodeJS version 22+. Далее по шагам выполните алгоритм.

Шаг 1. Установка зависимостей. Для установки зависимостей откройте консоль в директории проекта и выполните команду:

```
npm ci
```

Шаг 2. Запуск проекта. Выполните команду:

```
npm start
```

Шаг 3. Запуск Development-сервера. Чтобы запустить сервер для разработки, перейдите в директорию «**client**» выполните команду:

```
npm run dev
```

Для просмотра перейдите по ссылке в консоли.

Шаг 4. Сборка проекта. Чтобы выполнить production-сборку, выполните команду:

```
npm run build
```

1.1 Тестирование

Проект содержит юнит-тесты и интеграционные тесты. Для того чтобы их запустить в терминале выполните команду:

```
npm run test
```

Глава 2. Логика работы

Процедура запуска и входа предназначена для организации защищённой сеансовой связи клиента с сервером. Здесь работает механизм `handshake` – процесс идентификации сервера и клиента при установлении безопасного соединения по HTTPS или по WSS. В результате генерируется общий секретный ключ, который используется для шифрования данных. Для дальнейшего взаимодействия клиент получает обязательные идентификаторы: одноразовую метку **nonce**, уникальный идентификатор пользователя **user_id**, токен авторизации **token** и идентификатор канала событий **events_channel**.

При старте происходит обмен RSA-ключами, используется парольная фраза для создания HMAC-подписи.

Процесс регистрации можно разбить на три этапа:

1. Инициализация:

- Клиент генерирует RSA-ключи, из публичного ключа и парольной фразы создаёт подпись.
- Отправляет запрос с методом **start**, передавая:
 - **user**: имя пользователя.
 - **public_key**: публичный ключ в формате `pem`.
 - **signature**: подпись в формате `Base64`.
- Сервер проводит проверку имени пользователя и подписи следующим образом: используя полученный открытый ключ и заранее известную серверу парольную фразу, вычисляется контрольная подпись, которая сравнивается с переданной клиентом подписью. Если проверка успешна, сервер отправляет клиенту объект результата, включающий в себя:
 - **nonce**: уникальный идентификатор для авторизации (действителен в течение 30 с).
 - **server_public_key_b64**: публичный ключ сервера, зашифрованный публичным ключом клиента (в формате `Base64`).

2. Авторизация:

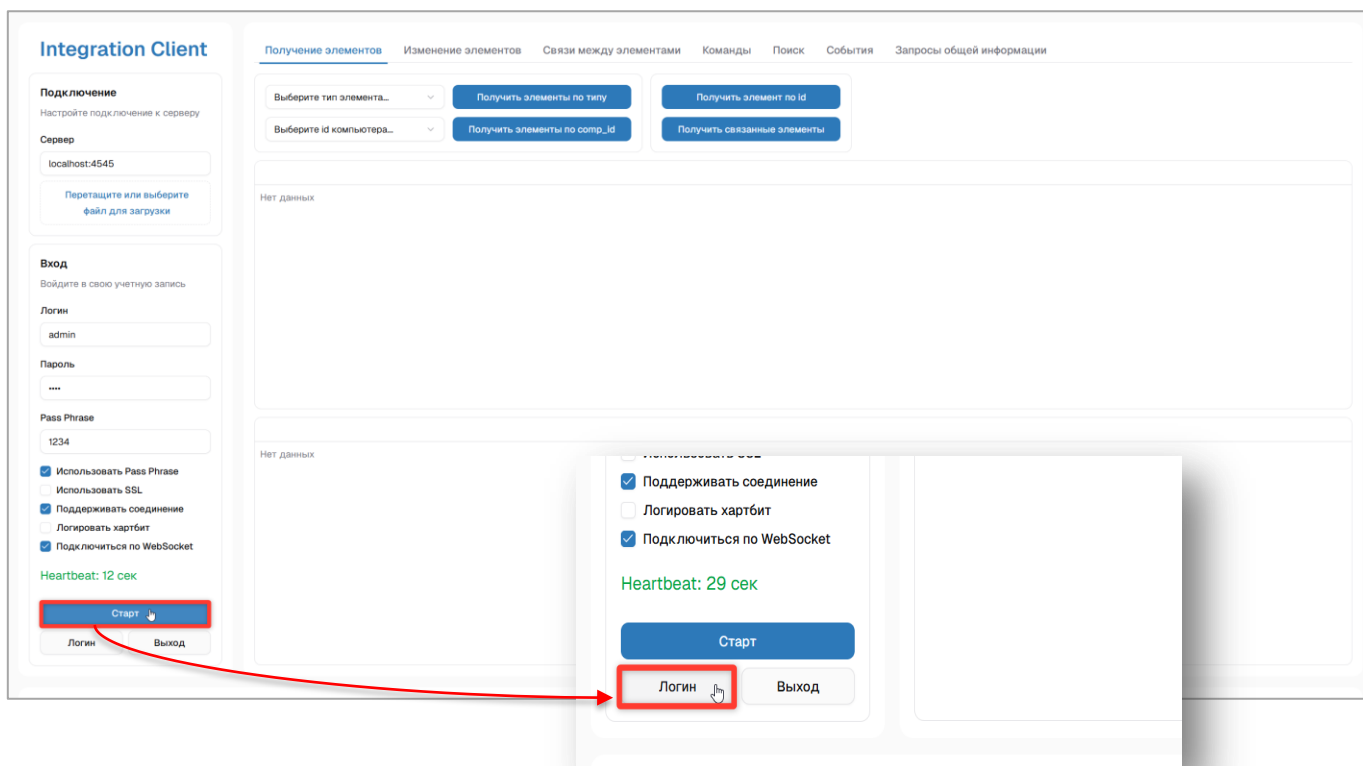
- Клиент отправляет запрос с методом **login**, передавая:
 - **nonce**: полученный ранее идентификатор.
 - **encrypted_password_b64**: пароль пользователя, зашифрованный публичным ключом сервера (в формате `Base64`).
- Сервер проверяет данные и возвращает:
 - **token**: токен сессии для аутентификации последующих запросов (действителен в течение 20 мин).
 - **events_channel**: канал для подключения к WebSocket.

3. Поддержание сессии (heart-beat):

- Для продления времени действия **token** клиент отправляет запрос с методом **renew_connection**, используя текущий **token**.
- Сервер подтверждает активность сессии, сбрасывая таймер (30 с), после чего клиент должен повторить отправку запроса.
- Если запрос не был отправлен в указанный промежуток времени, требуется повторное выполнение процедуры авторизации для получения нового действительного **token**.

2.1 Авторизация в системе

Чтобы между сервером и клиентом произошёл handshake (обмен ключами) нажмите кнопку «**Старт**». Затем сразу после него нажимаем кнопку «**Логин**», чтобы авторизоваться в системе.

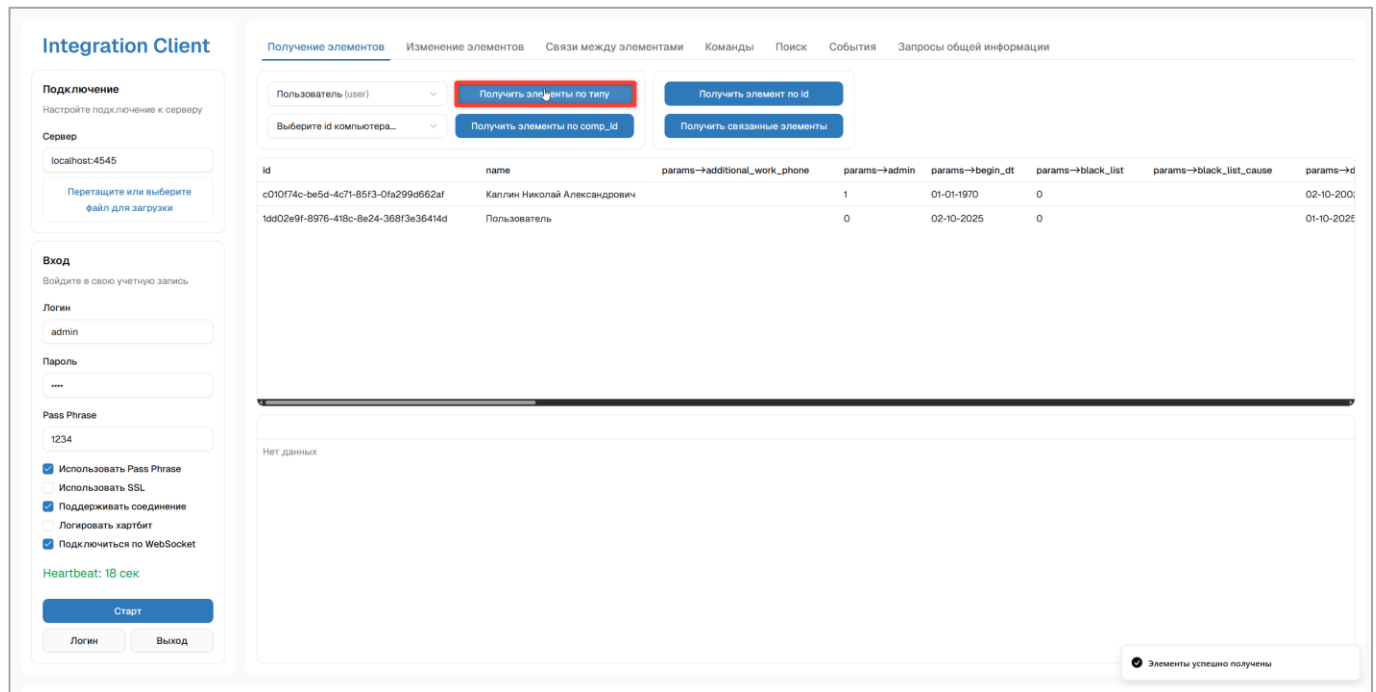


Пока вы видите надпись «**Heartbeat**» над кнопкой «**Старт**» это означает, что вы авторизованы и можете использовать программу.

Все методы обращения к серверу находятся в *client/src/services/api.service.ts*

На странице «**Получение элементов**» вы можете получить подробную информацию об элементах.

2.2 Получение элементов



Integration Client

Подключение

Настройте подключение к серверу

Сервер

localhost:4545

Перетащите или выберите файл для загрузки

Вход

Войдите в свою учетную запись

Логин

admin

Пароль

....

Pass Phrase

1234

☒ Использовать Pass Phrase

☐ Использовать SSL

☒ Поддерживать соединение

☐ Логировать хэштег

☒ Подключиться по WebSocket

Heartbeat: 18 сек

Старт

Логин

Выход

Получение элементов

Изменение элементов

Связи между элементами

Команды

Поиск

События

Запросы общей информации

Пользователь (user)

Получить элементы по типу

Получить элемент по id

Выберите id компьютера...

Получить элементы по comp_id

Получить связанные элементы

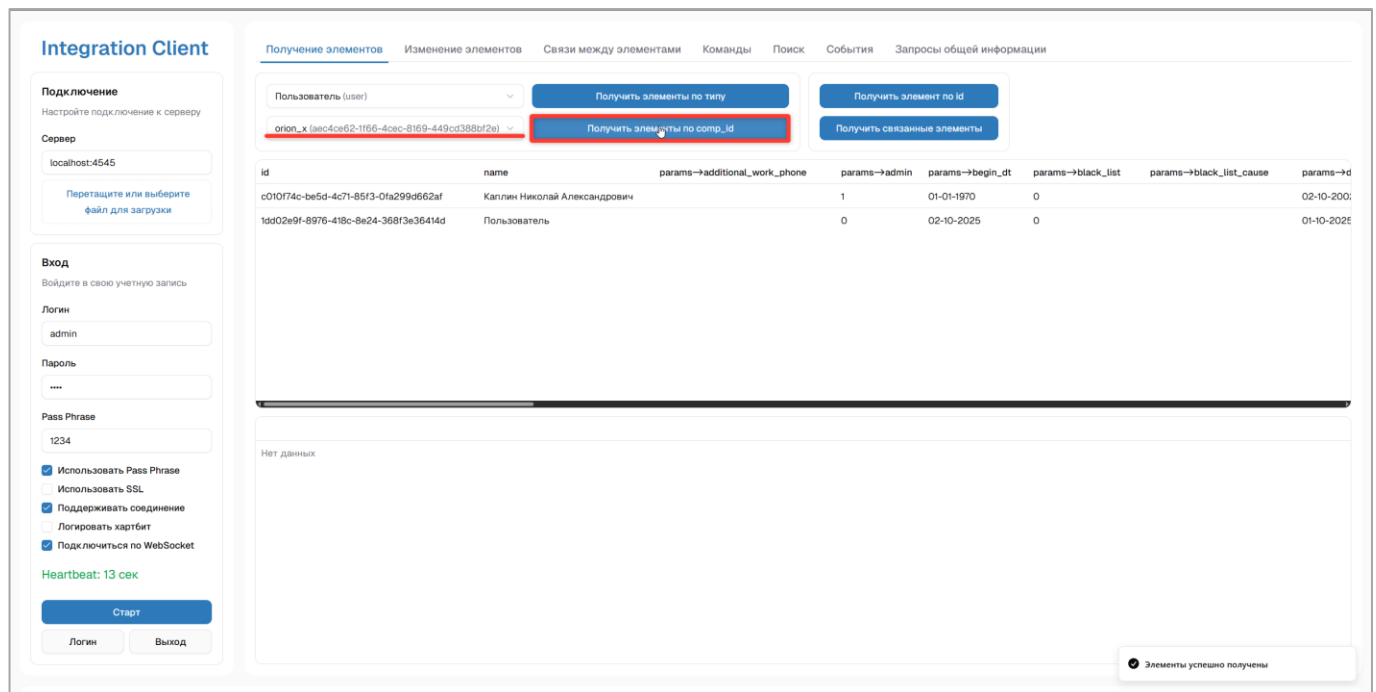
id	name	params→additional_work_phone	params→admin	params→begin_dt	params→black_list	params→black_list_cause	params→d
c010f74c-be5d-4c71-85f3-Ofa299d662af	Каплин Николай Александрович		1	01-01-1970	0		02-10-2001
1dd02e9f-8976-418c-8e24-368f3e36414d	Пользователь		0	02-10-2025	0		01-10-2025

Нет данных

Элементы успешно получены

При клике клиент вызывает метод **getElements** из API-сервиса и помещает в параметры выбранный тип и токен авторизации, после чего сервер присылает данные об элементах.

Если же выбрать id компьютера и элемент, то можно получить элемент в соответствии с id компьютера.



Integration Client

Подключение

Настройте подключение к серверу

Сервер

localhost:4545

Перетащите или выберите файл для загрузки

Вход

Войдите в свою учетную запись

Логин

admin

Пароль

....

Pass Phrase

1234

☒ Использовать Pass Phrase

☐ Использовать SSL

☒ Поддерживать соединение

☐ Логировать хэштег

☒ Подключиться по WebSocket

Heartbeat: 13 сек

Старт

Логин

Выход

Получение элементов

Изменение элементов

Связи между элементами

Команды

Поиск

События

Запросы общей информации

Пользователь (user)

Получить элементы по типу

Получить элемент по id

орон_x (aee4ce62-1f56-4cec-8169-449cd388bf2e)

Получить элементы по comp_id

Получить связанные элементы

id	name	params→additional_work_phone	params→admin	params→begin_dt	params→black_list	params→black_list_cause	params→d
c010f74c-be5d-4c71-85f3-Ofa299d662af	Каплин Николай Александрович		1	01-01-1970	0		02-10-2001
1dd02e9f-8976-418c-8e24-368f3e36414d	Пользователь		0	02-10-2025	0		01-10-2025

Нет данных

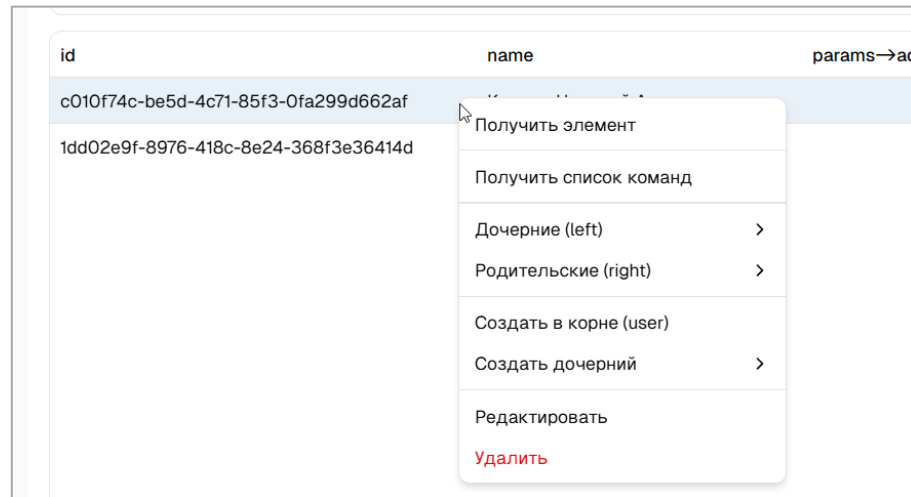
Элементы успешно получены

Клиент вызывает метод **getComputerElements**, передавая туда:

- **type** – тип элемента
- **comp_id** – id выбранного компьютера

и сервер возвращает элементы.

Если кликнуть по элементу ПКМ, то появится меню для действий с элементом.



«Получить элемент» – элемент отобразится в отдельной таблице, это удобно когда данных много.

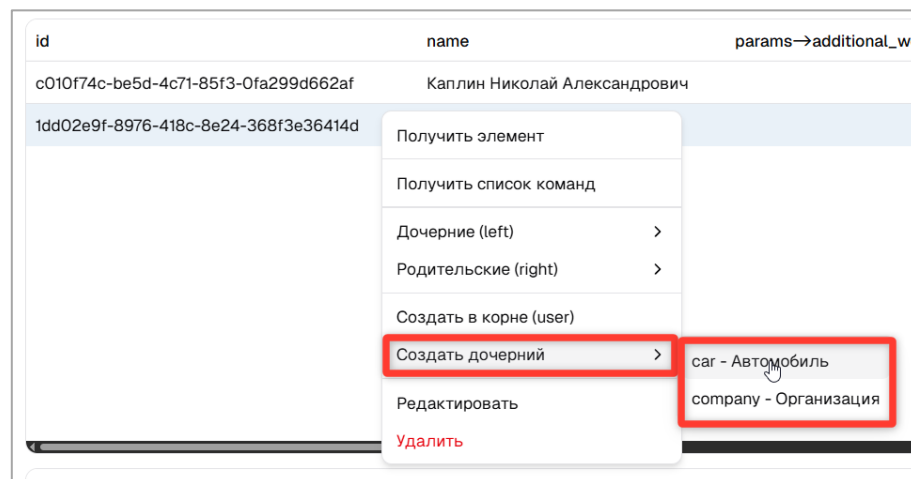
«Дочерние (left)» – получить дочерние элементы, они отобразятся во второй таблице, клиент вызывает метод **getElementsByLink**, передает параметры:

- **type** – тип элемента;
- **link_type** – тип получаемых дочерних элементов;
- **left** – true.

«Родительские (right)» – получить родительские элементы, они так же будут в отдельной таблице. Вызывается тот же метод что и в дочерних, только left – false.

«Создать в корне (user)» – создать в корне типа выбранного элемента новый элемент, клиент вызывает метод **createElementByLink**, передаёт туда указанные вами параметры, тип элемента, тип ссылочного элемента, id ссылки. В ответ сервер создает новый элемент и возвращает его.

«Создать дочерний» – происходит все тоже самое, что и в корне.



2.3 Изменение элементов

The screenshot shows the 'Integration Client' interface. On the left is a sidebar with sections for 'Подключение' (Connection) and 'Вход' (Login). The main area has a top navigation bar with tabs: 'Получение элементов', 'Изменение элементов' (selected), 'Связи между элементами', 'Команды', 'Поиск', 'События', and 'Запросы общей информации'. Below the tabs are four buttons: 'Создать элемент в корне', 'Создать дочерний элемент', 'Редактировать элемент', and 'Удалить элемент'. A table displays a list of elements with columns: id, name, params→additional_work_phone, params→admin, params→begin_dt, params→black_list, params→black_list_cause, and params→d. The table contains two rows of data. Below the table is a section labeled 'Нет данных' (No data).

id	name	params→additional_work_phone	params→admin	params→begin_dt	params→black_list	params→black_list_cause	params→d
c01074c-be5d-4c71-85f3-0fa299d662af	Каплин Николай Александрович		1	01-01-1970	0		02-10-2001
1dd02e9f-8976-418c-8e24-368f3e36414d	Пользователь		0	02-10-2025	0		01-10-2025

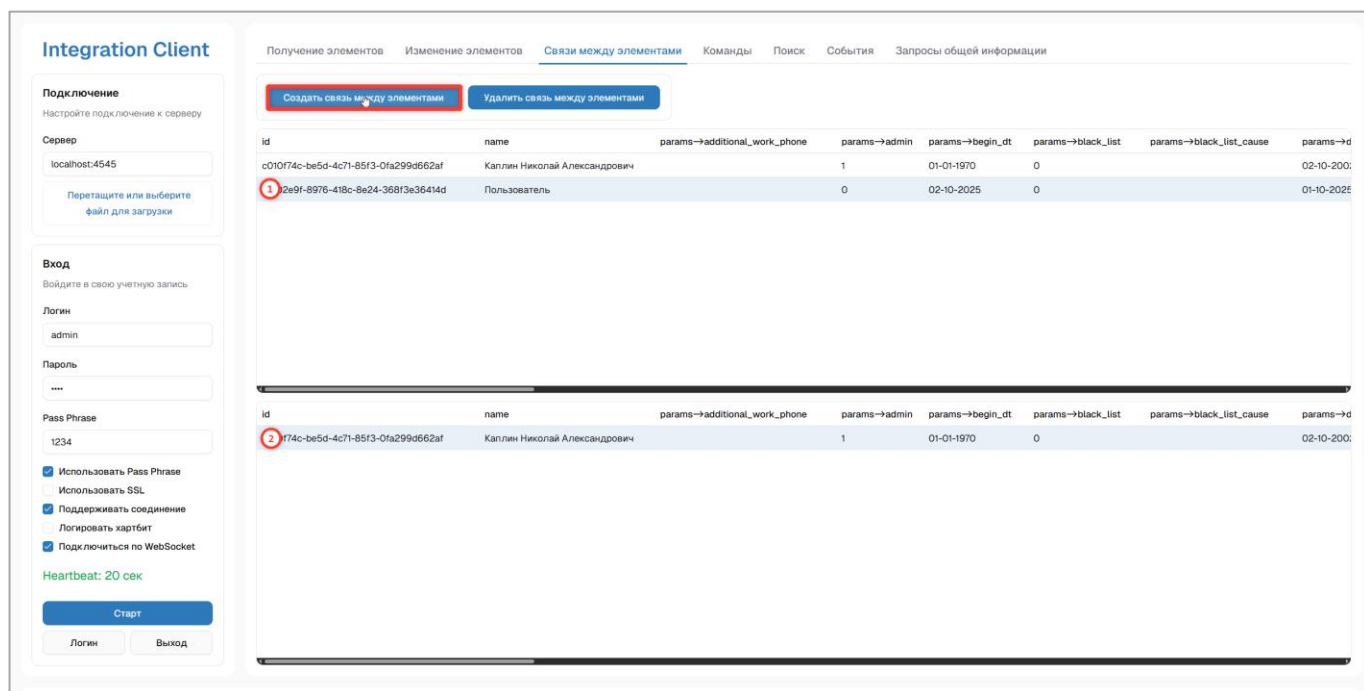
На странице изменения элементов повторяется функционал меню, для действия нужно лишь выбрать строку в таблице. Клиент использует те же методы, что и функции меню.

«**Редактировать элемент**» – позволяет редактировать элемент. Клиент вызывает метод **updateElement**, в него передаётся обновленный элемент и при успешном обновлении сервер возвращает его.

«**Удалить**» – удаляет элемент, клиент вызывает метод **deleteElement**, при успешном удалении сервер возвращает его id и тип.

2.4 Связи между элементами

На данной странице можно создавать и удалять связи между элементами.



Для создания или удаления связей между элементами нужно выбрать по элементу в каждой таблице.

«Создать связь между элементами» – клиент вызывает метод **createLink**, в который передаёт **left_type** – тип дочернего элемента, **left_id** – id дочернего элемента, **right_type** – тип дочернего элемента, **right_id** – id родительского элемента.

«Удалить связь между элементами» – клиент вызывает метод **deleteLink**, в который передаёт **left_type** – тип дочернего элемента, **left_id** – id дочернего элемента, **right_type** – тип дочернего элемента, **right_id** – id родительского элемента.

2.5 Поиск

Integration Client

Получение элементов | Изменение элементов | Связи между элементами | Команды | **Поиск** | События | Запросы общей информации

Подключение
Настройте подключение к серверу

Сервер: localhost:4545

Перетащите или выберите файл для загрузки

Вход
Войдите в свою учетную запись

Логин: admin

Пароль: ****

Pass Phrase: 1234

☒ Использовать Pass Phrase
☐ Использовать SSL
☒ Поддерживать соединение
☐ Логировать хэштег
☒ Подключиться по WebSocket

Heartbeat: 17 сек

Старт | Логин | Выход

Общие настройки поиска

Тип элемента: Пользователь (user)

☐ Игнорировать регистр
☐ Искать точное совпадение

Поиск по имени

Значение: Николай

Поиск

Поиск по параметру

☐ Поиск в доп.таблицах

Параметр: Имя поля

Значение:

Поиск по целочисленному значению
Поиск по строковому значению
Поиск по любому параметру

Поиск по параметру в доп.таблице

Имя доп.таблицы:

Параметр в доп.таблице:

Значение:

Поиск

Поиск по коду ключа

Значение:

Поиск

id	name	params→additional_work_phone	params→admin	params→begin_dt	params→black_list	params→black_list_cause	params→d
c01074c-be5d-4c71-85f3-0fa299d662af	Каплин Николай Александрович		1	01-01-1970	0		02-10-200:

Поиск выполнен

На странице поиска можно производить конкретный поиск с настройками.

«Поиск по имени» – клиент вызывает метод **findElementsByName**, передавая туда имя элемента.

«Поиск по целочисленному значению» – клиент вызывает метод **findElementsByParamI**, передавая туда параметр и значение (число).

«Поиск по строковому значению» – клиент вызывает метод **findElementsByParam**, передавая туда параметр и строковое значение.

«Поиск по любому параметру» – клиент вызывает метод **findElementsByAnyParamS**, передаёт туда параметр и строковое или целочисленное значение.

«Поиск по параметру в доп.таблице» – клиент вызывает метод **findElementsByAdditionalParamS**.

«Поиск по коду ключа» – клиент вызывает метод **findElementsByAdditionalParamS**.

2.6 События

Integration Client

Подключение

Настройте подключение к серверу

Сервер

localhost:4545

Перетащите или выберите файл для загрузки

Вход

Войдите в свою учетную запись

Логин

admin

Пароль

Pass Phrase

1234

☒ Использовать Pass Phrase

☐ Использовать SSL

☒ Поддерживать соединение

☐ Логировать хэштег

☒ Подключиться по WebSocket

Heartbeat: 19 сек

Старт

Логин

Выход

Получение элементов

Изменение элементов

Связи между элементами

Команды

Поиск

События

Запросы общей информации

Получение доп. информации

Типы инцидентов

Группы событий

Описание событий для

Пользователь (user)

Получение событий

Искать после указ. времени

14.11.2025

0:00

Число записей

1000

Получить события

Получить события с описанием

category	msg_id	date_time	comp_id	type	id	action	desc_text	desc_color	desc_group_nu
regular	a5b4a0e4-06a6-4a21-83d7-015b5d05354e	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	shell	00000000-0000-0000-0000-000000000000	1043			0
regular	fc1a99bd-084a-429e-9c34-6bb1f7393609	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	core	00000000-0000-0000-0000-000000000000	450			0
regular	27c91bca-898f-4e93-87dc-21a812189b68	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	core	00000000-0000-0000-0000-000000000000	4101			0
regular	25e243f1-1755-4f86-82af-574693ff8012	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	shell	00000000-0000-0000-0000-000000000000	1043			0
regular	85e673fd-3818-453d-ab67-500580f64294	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	core	00000000-0000-0000-0000-000000000000	450			0
regular	a04cd815-8447-4a86-8cf1-f0869d97ec3f	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	core	00000000-0000-0000-0000-000000000000	4101			0
regular	3193b0e0-a8e1-4217-8f5c-35cd286abee4	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	core	00000000-0000-0000-0000-000000000000	451			0
regular	8782290a-105b-4414-951f-674765bcb5b5	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	shell	00000000-0000-0000-0000-000000000000	1100			0
regular	8adc28b1-1899-416d-b5e3-6ea6088963f3	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	shell	00000000-0000-0000-0000-000000000000	1043			0
regular	4fd182e9-1012-4112-933c-844e262241ac	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	core	00000000-0000-0000-0000-000000000000	450			0
regular	3e4cb9e4-06ce-454f-ba3b-2a73b0af1672	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	core	00000000-0000-0000-0000-000000000000	4101			0
regular	65aa5555-a025-4bfc-967f-83838ca55ef8	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	core	00000000-0000-0000-0000-000000000000	451			0
regular	ba1552dd-8750-4f3d-9e12-68fc3c4c035a	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	shell	00000000-0000-0000-0000-000000000000	1100			0
regular	f29de5bb-3ca4-4b62-b2eb-68569b68aa97	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	shell	00000000-0000-0000-0000-000000000000	1043			0
regular	7d230031-3510-4f82-bde3-0cbcdf03926c	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	core	00000000-0000-0000-0000-000000000000	450			0
regular	5dca0dc5-ac17-488a-b3c1-12040c925ad3	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	core	00000000-0000-0000-0000-000000000000	4101			0
regular	f1a1976-8b4b-4b9b-a986-1935fdd89892	13-11-2025	aec4ce62-1166-4cec-8169-449cd388bf2e	shell	00000000-0000-0000-0000-000000000000	1100			0

События с описанием получены

На странице событий можно получить информацию о событиях.

«Типы инцидентов» – клиент вызывает метод **getIncidentTypes** и сервер возвращает список инцидентов.

«Группы событий» – клиент вызывает метод **getMessageGroups** и сервер возвращает список групп событий.

«Описание событий для (выбранная категория)» – клиент вызывает метод **getMessagesDesc** и сервер возвращает список события для категории.

«Получить событий» – позволяет получить определенные события по указанному времени, клиент вызывает метод **getEvents**.

«Получить события с описанием» – клиент так же вызывает метод **getEvents**, затем передаёт данные методу **getMessagesDesc** и выводит все вместе в таблицу.

2.7 Работа с WebSocket

WebSocket-соединение используется для получения уведомлений от сервера, таких как обновления данных или команды на выполнение запросов. Это позволяет клиенту динамически реагировать на события сервера.

2.8 Запросы общей информации

The screenshot shows the 'Integration Client' interface. On the left is a sidebar with sections for 'Подключение' (Connection) and 'Вход' (Login). The main area has a top navigation bar with tabs: 'Получение элементов', 'Изменение элементов', 'Связи между элементами', 'Команды', 'Поиск', 'События', and 'Запросы общей информации'. Below the tabs are four buttons: 'Получить типы элементов' (highlighted with a red box), 'Получить список типов по прототипу orion_device', 'Получить список компьютеров', and 'Получить список приборов'. Below these buttons is a table with three columns: 'key', 'description', and 'left'. The table lists various system components like 'user', 'car', 'cars', 'companies', 'company', 'department', 'departments', 'logic_group_zones', 'logic_zone', 'orion_group_zone', 'orion_zone', 'orion_input', 'orion_output', 'orion_reader', 'orion_destination', 'microphone_console_20', 'mip_12', 'mip_12_11', 'mip_24', 'mip_24_11', 'mip_24_s', and 'mip_24_11'. At the bottom right, a status message says 'Типы элементов успешно получены' (Element types successfully received).

key	description	left
user	Пользователь	car, company
car	Автомобиль	
cars	Автомобили	car
companies	Организации	company
company	Организация	
department	Отдел	department, user
departments	Отделы	department
logic_group_zones	Системная группа зон	logic_zone, orion_zone
logic_zone	Системная зона	s2000m, signal_20, signal_20p, s2000_sp1, s2000_4, s2000_4_v300, s2000_k, s2000_1t, s2000_kdl, s2000_bi, s2000_bki, v220, s2000_bi_v223, signal_20_02, s2000...
orion_group_zone	Аппаратная группа зон	orion_zone
orion_zone	Аппаратная зона	s2000m, signal_20, signal_20p, s2000_sp1, s2000_4, s2000_4_v300, s2000_k, s2000_1t, s2000_kdl, s2000_bi, s2000_bki, v220, s2000_bi_v223, signal_20_02, s2000...
orion_input	Вход	
orion_output	Выход	orion_zone, orion_group_zone
orion_reader	Считыватель	orion_zone, orion_group_zone
orion_destination	Адресат	
microphone_console_20	Микрофонная консоль - 20	orion_input
mip_12	МИП-12	orion_input
mip_12_11	МИП-12 исп. 11	orion_input
mip_24	МИП-24	orion_input
mip_24_11	МИП-24 исп. 11	orion_input
mip_24_s	МИП-24-С	orion_input

«Получить типы элементов» — позволяет получить все типы элементов, клиент вызывает метод **getTypesElements** и сервер возвращает типы элементов.

«Получить список типов по прототипу orion_device» — клиент вызывает метод **getTypesByTag**.

«Получить список компьютеров» — клиент вызывает метод **getComputers** и сервер возвращает список компьютеров.

«Получить список приборов» — клиент вызывает метод **getTypesByTag**, сервер возвращает все типы приборов, затем клиент выполняет batch request со всеми полученными типами для получения списка всех приборов.

Глава 3. Работа с SSL-сертификатами

Модуль интеграции Орион Икс поддерживает использование SSL (сертификаты шифрования) для безопасной передачи данных.

Для работы с https необходим сервер, который будет раздавать статичные файлы, и управлять сертификатами. Клиенту нужно знать куда обращаться к серверу, для этого есть переменные окружения .env.

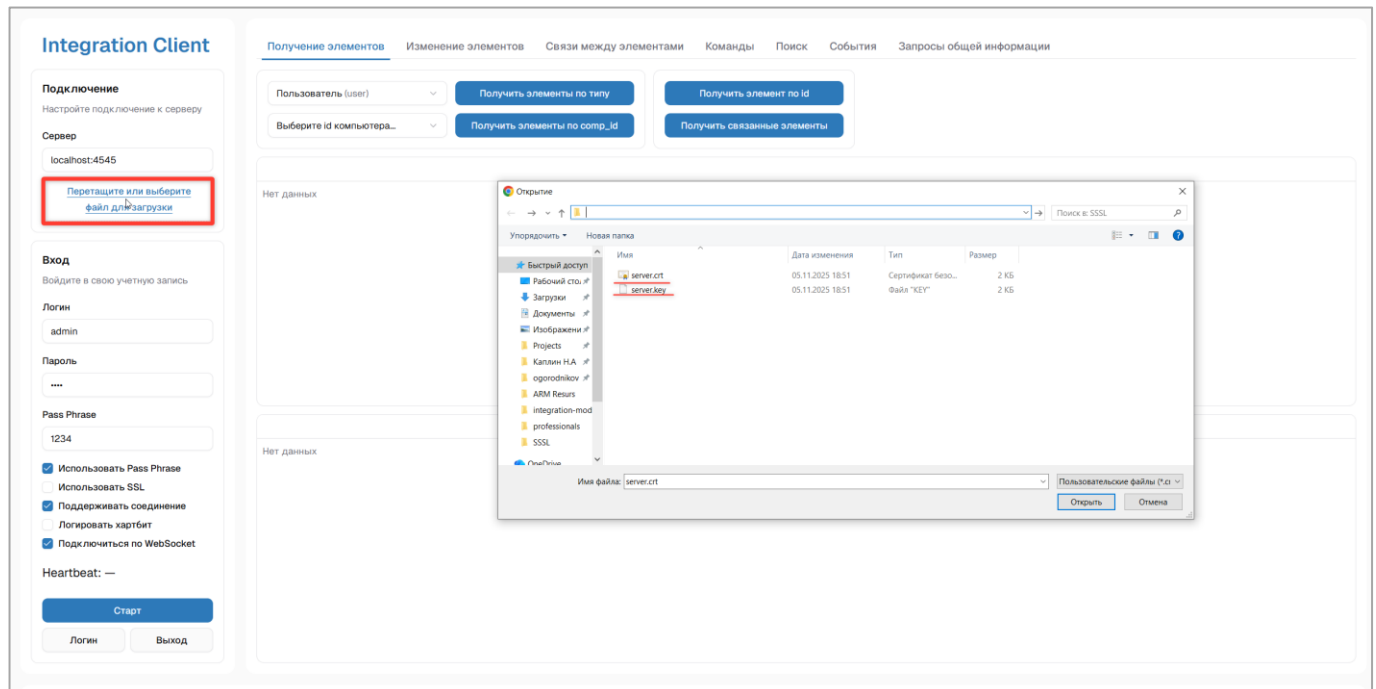
Переменная **VITE_APP_DOMAIN** нужна для того, чтобы клиент понимал, где находится сервер.

Переменная **VITE_APP_PORT** нужна для того, чтобы клиент понимал на каком порту сервер.

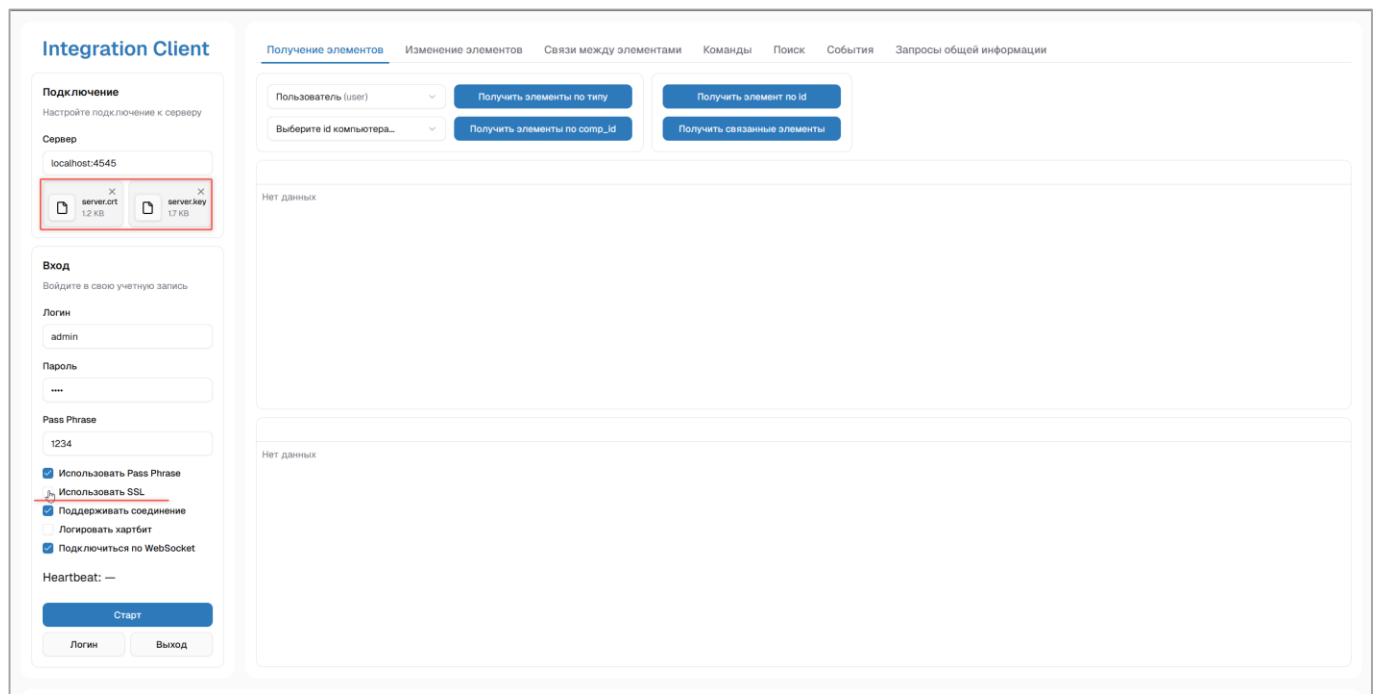
При изменении этих переменных, для того чтобы их применить необходимо выполнить:

`npm run build`

Для того чтобы пересобрать клиент с новыми переменными окружения. Чтобы активировать SSL в клиенте, необходимо загрузить ключ и сертификат.



После загрузки сертификата и ключа выберите использовать SSL и клиент перейдёт в SSL-режим.



Чтобы вернуться обратно, необходимо просто выключить SSL.

Глава 4. Класс `jsonrpc-websockets` для взаимодействия с сервером

Класс `jsonrpc-websockets` – это универсальный клиент для выполнения RPC-запросов к серверу по протоколам HTTP или WebSocket. Он реализует все необходимые механизмы для отправки отдельных и пакетных (batch) RPC-запросов, подписки на события, управления очередью запросов, автоматического переподключения, а также обработки уведомлений и ошибок.

Рекомендуется использовать этот класс как базовый при построении собственного клиента взаимодействия с JSON-RPC сервером. Находится в `src/lib/jsonrpc-websockets/src/client`.

Ключевые возможности:

- Поддержка JSON-RPC 2.0 через HTTP и WebSocket.
- Автоматическое управление соединением, включая опциональный автоконнект и автоматическое переподключение при разрыве.
- Ведение очереди запросов и батчей, установка индивидуальных таймаутов на каждый запрос.
- Обработка стандартных событий транспорта: открытие, закрытие, ошибка, получение сообщения.
- Методы для вызова или пакетного вызова методов RPC (`call`, `callBatch`), отправки уведомлений (`notify`), (от)подписки на события (`subscribe`, `unsubscribe`), и динамической смены транспорта (`switchTransport`).
- Опциональное управление таймаутами, количеством переподключений и интервалом между ними.

Глава 5. Запросы данных

Каждый запрос клиента соответствует стандарту JSON-RPC 2.0 и состоит из следующих полей:

Структура запроса:

- jsonrpc: Версия протокола (всегда «2.0»).
- method: Название метода, выполняемого на сервере.
- params: Параметры метода, включающие:

get_computers

(получение списка идентификаторов компьютеров)

Возвращает список comp_id.

- method: get_computers
- jsonrpc: 2.0
- id:
- params:
- security:
- token:

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "get_computers",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      }
8:    },
9:    "id": 4
10: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 4,
4:    "result": [
5:      {
6:        "comp_id": "c849386f-4812-425e-be6d-b853b6d30f1c",
```

```
7:      "name": "orion_x"
8:    }
9:  ]
10: }
```

get_elements

(получение информации об элементах указанного типа)

Возвращает список объектов указанного типа со всеми свойствами и дополнительными таблицами.

- method: get_elements
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type:

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "get_elements",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "car"
9:    },
10:   "id": 5
11: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 5,
4:    "result": [
5:      {
6:        "id": "3ae19ed4-51e6-40f2-8bd8-468498fd8f0e",
7:        "name": "Автомобиль",
8:        "params": {
```

```

9:         "color": "red",
10:        "comment": "",
11:        "model": "lada vesta",
12:        "registration_number": "777",
13:        "vin": ""
14:    },
15:    "additional_params": []
16: }
17: ]
18: }

```

get_computer_elements

(получение информации об элементах указанного типа для идентификатора компьютера)

Возвращает список объектов указанного типа со всеми свойствами и доп.таблицами, связанными с передаваемым comp_id.

- method: get_computer_elements
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type:
- comp_id:

Пример запроса:

```

1:  {
2:    "jsonrpc": "2.0",
3:    "method": "get_computer_elements",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "car",
9:      "comp_id": "c849386f-4812-425e-be6d-b853b6d30f1c"
10:    },
11:    "id": 6
12:  }

```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 6,
4:    "result": [
5:      {
6:        "id": "3ae19ed4-51e6-40f2-8bd8-468498fd8f0e",
7:        "name": "Автомобиль",
8:        "params": {
9:          "color": "red",
10:         "comment": "",
11:         "model": "lada vesta",
12:         "registration_number": "777",
13:         "vin": ""
14:       },
15:       "additional_params": []
16:     ]
17:   }
18: }
```

get_element

(получение информации об элементе)

Возвращает запрашиваемый (по типу и id) объект со всеми свойствами и доп.таблицами.

- method: get_element
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type:
- id:

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "get_element",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
```

```
7:      },
8:      "type": "car",
9:      "id": "3ae19ed4-51e6-40f2-8bd8-468498fd8f0e"
10:    },
11:    "id": 7
12:  }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 7,
4:    "result": {
5:      "id": "3ae19ed4-51e6-40f2-8bd8-468498fd8f0e",
6:      "name": "Автомобиль",
7:      "params": {
8:        "color": "red",
9:        "comment": "",
10:       "model": "lada vesta",
11:       "registration_number": "777",
12:       "vin": ""
13:     },
14:     "additional_params": []
15:   }
16: }
```

get_elements_by_link

(получение информации об элементах, связанных с передаваемым)

Возвращает список объектов указанного типа со всеми свойствами и дополнительными таблицами.

- method: get_elements_by_link
- jsonrpc: 2.0
- id:
- params:
 - security:
 - token:
- type: <тип элемента, связь с которым запрашивается>
- id: <элемент, связь с которым запрашивается>
- link_type:

- left: true/false (тип связи между элементами)

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "get_elements_by_link",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "logic_zone",
9:      "id": "alf9ed32-bef5-4721-841d-ed5c4bd25a94",
10:     "link_type": "orion_reader",
11:     "left": true
12:   },
13:   "id": 8
14: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 8,
4:    "result": [
5:      {
6:        "id": "73db82c3-bfa1-4ccc-b371-6e7b9546b5fe",
7:        "name": "Считыватель",
8:        "params": {
9:          "contact_id": 0,
10:         "number": 1,
11:         "type_id": 1
12:       },
13:       "additional_params": []
14:     }
15:   ]
16: }
```

get_incident_types (получение списка типов инцидентов)

Возвращает список объектов.

- method: get_incident_types
- jsonrpc: 2.0
- id:
- params:
- security:
- token:

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "get_incident_types",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      }
8:    },
9:    "id": 9
10: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 9,
4:    "result": [
5:      {
6:        "number": 0,
7:        "id": "start",
8:        "name": "Пуски",
9:        "text": "Инцидент в категории \"Пуски\""
10:      },
11:      ...
12:    ]
13: }
```

get_message_groups (получение списка групп сообщений)

Возвращает список объектов.

- method: get_message_groups
- jsonrpc: 2.0
- id:
- params:
- security:
- token:

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "get_message_groups",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      }
8:    },
9:    "id": 10
10: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 10,
4:    "result": [
5:      {
6:        "number": 1,
7:        "id": "blink\_mask",
8:        "text": "Маски мигания реле",
9:        "color": "#fff8a15a",
10:       "text\_color": "#ffffffff"
11:      },
12:      ...
13:    ]
14: }
```

get_message_desc
(получение описания сообщений)

Возвращает список объектов с описанием сообщений для указанного типа.

- method: get_message_desc
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type: <тип элементов, для которых запрашивается описание событий>

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "get_messages_desc",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "user"
9:    },
10:   "id": 11
11: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 11,
4:    "result": [
5:      {
6:        "action": 449,
7:        "text": "Изменение комментария по инциденту",
8:        "hide": true,
9:        "log": true,
10:       "color": "#00ffffff",
11:       "text\_color": "#ff000000",
12:       "group\_number": 51,
13:       "incident\_type\_number": 4294967295
14:     },

```

```
15: ...
16: ]
17: }
```

find_elements_by_param_i (поиск элементов по целочисленному значению)

Поиск элементов по целочисленному значению указанного параметра среди указанного типа элементов. Возвращает список найденных объектов со всеми свойствами.

- method: find_elements_by_param_i
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type: <тип элементов, среди которых выполняется поиск>
- param_name: (допускаются только параметры, хранящие целочисленный тип)
- param_value: (int)
- additional: true/false (поиск в доп.таблицах)
- case_insensitive: true/false (игнорировать регистр)
- exact: true/false (искать точное совпадение)

Пример запроса:

```
1: {
2:   "jsonrpc": "2.0",
3:   "method": "find_elements_by_param_i",
4:   "params": {
5:     "security": {
6:       "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:     },
8:     "type": "user",
9:     "param_name": "fired",
10:    "param_value": 1,
11:    "additional": false,
12:    "case_insensitive": true,
13:    "exact": false
14:  },
15:  "id": 12
16: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 12,
4:    "result": [
5:      {
6:        "id": "bb26eef8-7b60-4cf2-9cd2-d6edeb736193",
7:        "name": "Поль Зователь ",
8:        "params": {
9:          "additional\_work\_phone": "300",
10:        ...
11:        "black\_list": 1,
12:        "black\_list\_cause": "",
13:        ...
14:        "user\_name": "Зователь",
15:        "work\_phone": "+74957757155"
16:      },
17:      "additional\_params": [
18:        {
19:          "name": "keys",
20:          "params": [
21:            {
22:              "key\_code": "1234",
23:              "locked": 0,
24:              "type\_key": 0
25:            }
26:          ]
27:        }
28:      ]
29:    },
30:    ...
31:  ]
32: }
```

find_elements_by_param_s (поиск элементов по строковому значению)

Поиск элементов по строковому значению указанного параметра среди указанного типа элементов. Возвращает список найденных объектов со всеми свойствами.

- method: find_elements_by_param_s
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type: <тип элементов, среди которых выполняется поиск>
- param_name: (допускаются только параметры, хранящие строки)
- param_value: (str)
- additional: true/false (поиск в доп.таблицах)
- case_insensitive: true/false (игнорировать регистр)
- exact: true/false (искать точное совпадение)

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "find_elements_by_param_s",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "user",
9:      "param_name": "email",
10:     "param_value": "pochta",
11:     "additional": false,
12:     "case_insensitive": true,
13:     "exact": false
14:   },
15:   "id": 13
16: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 13,
```

```

4:     "result": [
5:         {
6:             "id": "a65eaeb1-3241-41f3-86d2-13035e07c9c6",
7:             "name": "Иванов Иван Иванович",
8:             "params": {
9:                 "additional\_work\_phone": "437",
10:                 ...
11:                 "email": "pochta@pochta.ru",
12:                 ...
13:                 "work\_phone": "+74957757155"
14:             },
15:             "additional\_params": [
16:                 {
17:                     "name": "keys",
18:                     "params": [
19:                         {
20:                             "key\_code": "2345",
21:                             "locked": 0,
22:                             "type\_key": 0
23:                         },
24:                         {
25:                             "key\_code": "3456",
26:                             "locked": 0,
27:                             "type\_key": 0
28:                         }
29:                     ]
30:                 }
31:             ]
32:         },
33:         ...
34:     ]
35: }

```

find_elements_by_any_param_s

(поиск элементов по строковому значению любого параметра)

Поиск элементов по строковому значению любого параметра среди указанного типа элементов.

Возвращает список найденных объектов со всеми свойствами.

- method: find_elements_by_any_param_s
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type: <тип элементов, среди которых выполняется поиск>
- param_value: (str)
- additional: true/false (поиск в доп.таблицах)
- case_insensitive: true/false (игнорировать регистр)
- exact: true/false (искать точное совпадение)

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "find_elements_by_any_param_s",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "user",
9:      "param_value": "pochta",
10:     "additional": false,
11:     "case_insensitive": true,
12:     "exact": false
13:   },
14:   "id": 14
15: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 14,
4:    "result": [
5:      {
```

```
6:  "id": "a65eaeb1-3241-41f3-86d2-13035e07c9c6",
7:  "name": "Иванов Иван Иванович",
8:  "params": {
9:    "additional\_work\_phone": "437",
10:   ...
11:   "email": "pochta@pochta.ru",
12:   ...
13:   "work\_phone": "+74957757155"
14: },
15: "additional\_params": [
16: {
17:  "name": "keys",
18:  "params": [
19:  {
20:   "key\_code": "2345",
21:   "locked": 0,
22:   "type\_key": 0
23:  },
24:  {
25:   "key\_code": "3456",
26:   "locked": 0,
27:   "type\_key": 0
28:  }
29: ]
30: }
31: ]
32: },
33: ...
34: ]
35: }
```

find_elements_by_name (поиск элементов по имени)

Поиск элементов по имени среди указанного типа элементов. Возвращает список найденных объектов со всеми свойствами.

- method: find_elements_by_name
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type: <тип элементов, среди которых выполняется поиск>
- name: (str)
- case_insensitive: true/false (игнорировать регистр)
- exact: true/false (искать точное совпадение)

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "find_elements_by_name",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "user",
9:      "name": "Иван",
10:     "case_insensitive": false,
11:     "exact": false
12:   },
13:   "id": 15
14: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 15,
4:    "result": [
5:      {
6:        "id": "a65eaeb1-3241-41f3-86d2-13035e07c9c6",
7:        "name": "Иванов Иван Иванович",
```

```

8:  "params": {
9:    "additional\_work\_phone": "437",
10:   ...
11:   "work\_phone": "+74957757155"
12: },
13: "additional\_params": [
14: {
15:   "name": "keys",
16:   "params": [
17:   {
18:     "key\_code": "2345",
19:     "locked": 0,
20:     "type\_key": 0
21:   },
22:   {
23:     "key\_code": "3456",
24:     "locked": 0,
25:     "type\_key": 0
26:   }
27: ]
28: }
29: ]
30: },
31: ...
32: ]
33: }

```

find_elements_by_additional_param_s

(поиск пользователя по ключу)

Поиск элементов по строковому значению (ключа) в дополнительной таблице. Возвращает список найденных объектов со всеми свойствами.

- method: find_elements_by_additional_param_s
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type: <тип элементов, среди которых выполняется поиск>

- additional_params_name: "keys" (название доп.таблицы)
- additional_param_name: "key_code" (название поля)
- additional_param_value: (str)
- case_insensitive: true/false (игнорировать регистр)
- exact: true/false (искать точное совпадение)

Пример запроса:

```

1:  {
2:    "jsonrpc": "2.0",
3:    "method": "find_elements_by_additional_param_s",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "user",
9:      "additional_params_name": "keys",
10:     "additional_param_name": "key_code",
11:     "additional_param_value": "9",
12:     "case_insensitive": false,
13:     "exact": false
14:   },
15:   "id": 16
16: }

```

Ответ сервера:

```

1:  {
2:    "jsonrpc": "2.0",
3:    "id": 16,
4:    "result": [
5:      {
6:        "id": "a65eaeb1-3241-41f3-86d2-13035e07c9c6",
7:        "name": "Иванов Иван Иванович",
8:        "params": {
9:          "additional\_work\_phone": "437",
10:         ...
11:         "work\_phone": "+74957757155"
12:       },
13:       "additional\_params": [
14:         {
15:           "name": "keys",

```

```

16: "params": [
17: {
18: "key\_code": "5678",
19: "locked": 0,
20: "type\_key": 0
21: },
22: {
23: "key\_code": "9999",
24: "locked": 0,
25: "type\_key": 0
26: }
27: ]
28: }
29: ]
30: },
31: ...
32: ]
33: }

```

types_by_tag

(получение списка типов для прототипа)

Возвращает список типов, соответствующих указанному тэгу (прототипу).

- method: types_by_tag
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- tag: (orion_device)

Пример запроса:

```

1: {
2:   "jsonrpc": "2.0",
3:   "method": "types_by_tag",
4:   "params": {
5:     "security": {
6:       "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:     },
8:     "tag": "orion_device"

```

```
9:     },
10:     "id": 17
11: }
```

Ответ сервера:

```
1: {
2:   "jsonrpc": "2.0",
3:   "id": 17,
4:   "result": [
5:     "user\_device",
6:     "unknown\_device",
7:     "rupor\_75",
8:     ...
9:     "signal\_20",
10:    "s2000m"
11:  ]
12: }
```

get_events

(получение истории событий)

Возвращает список объектов событий.

- method: get_events
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- date_time: (в формате “2025-05-19T00:00:00.0000000Z”)
- count: (int, максимальное количество в одном запросе - 1000)
- direction: “+/-” (направление выборки относительно date_time: если “+”, то выбираются события после указанного времени, если “-”, то выборка идет в обратном направлении)

Пример запроса:

```
1: {
2:   "jsonrpc": "2.0",
3:   "method": "get_events",
4:   "params": {
5:     "security": {
6:       "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
```

```
7:      },
8:      "date_time": "2025-08-13T04:00:00.0000000Z",
9:      "count": 2,
10:     "direction": "+"
11:  },
12:  "id": 18
13: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 18,
4:    "result": [
5:      {
6:        "category": "regular",
7:        "msg_id": "31d8c965-a33b-4adc-ac92-72ffdb65857a",
8:        "date_time": "2025-08-13T05:19:25.7734710Z",
9:        "comp_id": "c849386f-4812-425e-be6d-b853b6d30f1c",
10:       "type": "shell",
11:       "id": "00000000-0000-0000-0000-000000000000",
12:       "action": 1043
13:     },
14:     {
15:       "category": "regular",
16:       "msg_id": "12f47fe6-cf00-478d-856d-813c0b845922",
17:       "date_time": "2025-08-13T05:19:36.9078690Z",
18:       "comp_id": "c849386f-4812-425e-be6d-b853b6d30f1c",
19:       "type": "core",
20:       "id": "00000000-0000-0000-0000-000000000000",
21:       "action": 450
22:     }
23:   ]
24: }
```

СОЗДАНИЕ, УДАЛЕНИЕ, РЕДАКТИРОВАНИЕ

create_element

(создание элемента)

Создаёт элемент указанного типа с указанным именем (создаётся в корне). Возвращает созданный объект.

- method: create_element
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type:
- name:
- params: { } (может быть пустой)
- additional params: [<объекты с доп.параметрами>] (может быть пустой)

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "create_element",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "department",
9:      "name": "Новый отдел",
10:     "params": {},
11:     "additional_params": []
12:   },
13:   "id": 19
14: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 19,
4:    "result": {
5:      "id": "43f9eecc-da5c-4529-b7f3-b385289643ab",
6:      "name": "Новый отдел",
```

```
7:      "params": {},
8:      "additional_params": []
9:    }
10: }
```

create_element_by_link (создание дочернего элемента)

Создаёт элемент указанного типа с указанным именем под выбранным элементом. Возвращает созданный объект.

- method: create_element_by_link
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type:
- name:
- params: { } (может быть пустой)
- additional params: [<объекты с доп.параметрами>] (может быть пустой)
- link_type:
- link_id:

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "create_element_by_link",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "department",
9:      "name": "Отдел под Новым отделом",
10:     "link_type": "department",
11:     "link_id": "43f9eccc-da5c-4529-b7f3-b385289643ab",
12:     "params": {},
13:     "additional_params": []
14:   },
15:   "id": 20
16: }
```

Ответ сервера:

```
17: {
18:   "jsonrpc": "2.0",
19:   "id": 20,
20:   "result": {
21:     "id": "09aae008-3b52-42b5-8e08-7411aaead563",
22:     "name": "Отдел под Новым отделом",
23:     "params": {},
24:     "additional_params": []
25:   }
26: }
```

delete_element

(удаление элемента)

Удаляет указанный элемент. Возвращает true при успехе.

- method: delete_element
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type:
- id:

Пример запроса:

```
1: {
2:   "jsonrpc": "2.0",
3:   "method": "delete_element",
4:   "params": {
5:     "security": {
6:       "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:     },
8:     "type": "department",
9:     "id": "aec4e4b3-4b95-47f0-a146-355719a52899"
10:   },
11:   "id": 21
12: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 21,
4:    "result": true
5:  }
```

update_element (изменение элемента)

Изменяет указанные параметры выбранного элемента. Возвращает изменённый объект.

- method: update_element
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type:
- id:
- name: <имя (или новое имя) элемента>
- params: { } (может быть пустой)
- additional params: [<изменяемые объекты с доп.параметрами>] (может быть пустой)

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "update_element",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "department",
9:      "id": "43f9eecd-da5c-4529-b7f3-b385289643ab",
10:     "name": "Первый сектор",
11:     "params": {},
12:     "additional_params": []
13:   },
14:   "id": 22
15: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 22,
4:    "result": {
5:      "id": "43f9eecd-da5c-4529-b7f3-b385289643ab",
6:      "name": "Первый сектор",
7:      "params": {},
8:      "additional_params": []
9:    }
10: }
```

create_link

(создание связи между элементами)

Создаёт связь между указанными элементами. Возвращает true при успехе.

- method: create_link
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- left_type: (например, user)
- left_id: <id левого связываемого элемента>
- right_type: (например, car)
- right_id: <id правого связываемого элемента>

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "create_link",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "left_type": "user",
9:      "left_id": "bb26eef8-7b60-4cf2-9cd2-d6edeb736193",
10:     "right_type": "company",
11:     "right_id": "23245ffa-50f0-4655-83ae-fed45cb09401"
12:   },
```

```
13:   "id": 23
14: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 23,
4:    "result": true
5:  }
```

delete_link

(удаление связи между элементами)

Удаляет связь между указанными элементами. Возвращает true при успехе.

- method: delete_link
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- left_type: (например, user)
- left_id: <id левого связанного элемента>
- right_type: (например, car)
- right_id: <id правого связанного элемента>

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "delete_link",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "left_type": "user",
9:      "left_id": "bb26eef8-7b60-4cf2-9cd2-d6edeb736193",
10:     "right_type": "company",
11:     "right_id": "23245ffa-50f0-4655-83ae-fed45cb09401"
12:   },
13:   "id": 24
14: }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "id": 24,
4:    "result": true
5:  }
```

УПРАВЛЕНИЕ

get_possible_commands_by_element

(получение списка возможных команд)

Возвращает список возможных команд (название, номер, возможные параметры) для выбранного элемента, учитывая его текущее состояние.

- method: get_possible_commands_by_element
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type:
- id:

Пример запроса:

```
1:  {
2:    "jsonrpc": "2.0",
3:    "method": "get_possible_commands_by_element",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "orion_output",
9:      "id": "1238743b-c588-46df-9e5e-0cd47c93c0cd"
10:    },
11:    "id": 25
12:  }
```

Ответ сервера:

```
1:  {
2:    "jsonrpc": "2.0",
```

```

3:   "id": 25,
4:   "result": {
5:     "commands": [
6:       {
7:         "name": "Выключение контроля",
8:         "command": 1104,
9:         "params": []
10:      }
11:    ]
12:  }
13: }

```

send_command_query_for_element

(отправка запроса на выполнение команды)

Запрос на выполнение команды для указанного элемента. Возвращает **success** при успехе.

- method: send_command_query_for_element
- jsonrpc: 2.0
- id:
- params:
- security:
- token:
- type:
- id:
- command:
- params: { <данные параметра*> } (может быть пустой)

Если у параметра есть поле «optional» и оно равно «true», то этот параметр в запросе является необязательным.

Пример запроса:

```

1:  {
2:    "jsonrpc": "2.0",
3:    "method": "send_command_query_for_element",
4:    "params": {
5:      "security": {
6:        "token": "ae067f28-bb67-46aa-847f-066ff634bb78"
7:      },
8:      "type": "orion_output",
9:      "id": "1238743b-c588-46df-9e5e-0cd47c93c0cd",

```

```
10:     "command": 1104,  
11:     "params": {}  
12: },  
13:     "id": 26  
14: }
```

Ответ сервера:

```
1:  {  
2:    "jsonrpc": "2.0",  
3:    "id": 26,  
4:    "error": {  
5:      "code": 1001,  
6:      "message": "success"  
7:    }  
8:  }
```