

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

«ЦЕНТР ОХРАНЫ 5»

## Описание методов HTTP-API

**C.Nord**

4 июля 2019 г.

# Содержание

<b>1</b>	<b>Общая информация</b>	<b>8</b>
1.1	Ограничение доступа	8
1.2	Настройки службы «C.Nord HTTP-API Service»	8
1.2.1	Url	8
1.2.2	ExternalUrl	8
1.2.3	TransportFileEncoding	8
1.2.4	TransportFileCheckInterval	9
1.2.5	CheckTransportFileOnStartup	9
1.2.6	TransportFilePath	9
1.2.7	TransportFileTempFile	9
1.2.8	ApiKey	9
1.2.9	DefaultPageSize	9
1.2.10	DefaultCheckPanicInterval	9
1.2.11	DefaultCheckStopOnEvent	10
1.3	Общий вид запросов	10
1.4	Ключ безопасности	10
1.5	Статусы ответов	10
1.6	Код 400: описание ошибки	10
1.6.1	Message	11
1.6.2	SpResultCode	11
<b>2</b>	<b>Объект</b>	<b>12</b>
2.1	Поля объекта	12
2.1.1	Значения поля DeviceTypeName	14
2.1.2	Значения поля DebtInformLevel	14
2.1.3	Формат полей с датой и временем	14
2.2	Получить список объектов (GET /api/Sites)	14
2.2.1	Параметры	15
2.2.2	Возможные статусы ответов	15
2.2.3	Возвращаемые данные	15
2.2.4	Пример использования	15
2.3	Получить объект по номеру или идентификатору (GET /api/Sites)	17
2.3.1	Параметры	17
2.3.2	Возможные статусы ответов	17
2.3.3	Возвращаемые данные	17
2.3.4	Пример использования	17

2.4	Создать объект (POST /api/Sites)	18
2.4.1	Параметры	18
2.4.2	Тело запроса	19
2.4.3	Возможные статусы ответов	19
2.4.4	Возвращаемые данные	19
2.4.5	Пример использования	19
2.5	Изменить объект (PUT /api/Sites)	20
2.5.1	Параметры	20
2.5.2	Тело запроса	20
2.5.3	Возможные статусы ответов	21
2.5.4	Возвращаемые данные	21
2.5.5	Пример использования	21
2.6	Удалить объект (DELETE /api/Sites)	21
2.6.1	Параметры	21
2.6.2	Возможные статусы ответов	21
2.6.3	Возвращаемые данные	21
2.6.4	Пример использования	22
<b>3</b>	<b>Раздел</b>	<b>23</b>
3.1	Поля раздела	23
3.2	Получить список разделов объекта (GET /api/Parts)	23
3.2.1	Параметры	23
3.2.2	Возможные статусы ответов	24
3.2.3	Возвращаемые данные	24
3.2.4	Пример использования	24
3.3	Получить раздел по идентификатору (GET /api/Parts)	24
3.3.1	Параметры	24
3.3.2	Возможные статусы ответов	25
3.3.3	Возвращаемые данные	25
3.3.4	Пример использования	25
3.4	Создать раздел (POST /api/Parts)	25
3.4.1	Параметры	25
3.4.2	Тело запроса	26
3.4.3	Возможные статусы ответов	26
3.4.4	Возвращаемые данные	26
3.4.5	Пример использования	26
3.5	Изменить раздел (PUT /api/Parts)	26
3.5.1	Параметры	26
3.5.2	Тело запроса	27

3.5.3	Возможные статусы ответов . . . . .	27
3.5.4	Возвращаемые данные . . . . .	27
3.5.5	Пример использования . . . . .	27
3.6	Удалить раздел (DELETE /api/Parts) . . . . .	27
3.6.1	Параметры . . . . .	27
3.6.2	Возможные статусы ответов . . . . .	28
3.6.3	Возвращаемые данные . . . . .	28
3.6.4	Пример использования . . . . .	28
<b>4</b>	<b>Шлейфы</b>	<b>29</b>
4.1	Поля шлейфа . . . . .	29
4.2	Получить список шлейфов объекта (GET /api/Zones) . . . . .	29
4.2.1	Параметры . . . . .	29
4.2.2	Возможные статусы ответов . . . . .	29
4.2.3	Возвращаемые данные . . . . .	30
4.2.4	Пример использования . . . . .	30
4.3	Получить шлейф по идентификатору (GET /api/Zones) . . . . .	30
4.3.1	Параметры . . . . .	30
4.3.2	Возможные статусы ответов . . . . .	30
4.3.3	Возвращаемые данные . . . . .	31
4.3.4	Пример использования . . . . .	31
4.4	Создать шлейф (POST /api/Zones) . . . . .	31
4.4.1	Параметры . . . . .	31
4.4.2	Тело запроса . . . . .	31
4.4.3	Возможные статусы ответов . . . . .	32
4.4.4	Возвращаемые данные . . . . .	32
4.4.5	Пример использования . . . . .	32
4.5	Изменить шлейф (PUT /api/Zones) . . . . .	32
4.5.1	Параметры . . . . .	32
4.5.2	Тело запроса . . . . .	33
4.5.3	Возможные статусы ответов . . . . .	33
4.5.4	Возвращаемые данные . . . . .	33
4.5.5	Пример использования . . . . .	33
4.6	Удалить шлейф (DELETE /api/Zones) . . . . .	33
4.6.1	Параметры . . . . .	33
4.6.2	Возможные статусы ответов . . . . .	33
4.6.3	Возвращаемые данные . . . . .	34
4.6.4	Пример использования . . . . .	34

<b>5</b>	<b>Ответственные лица</b>	<b>35</b>
5.1	Поля ответственного лица . . . . .	35
5.2	Получить список ответственных лиц объекта (GET /api/Customers) . . . . .	36
5.2.1	Параметры . . . . .	36
5.2.2	Возможные статусы ответов . . . . .	36
5.2.3	Возвращаемые данные . . . . .	36
5.2.4	Пример использования . . . . .	36
5.3	Получить ответственное лицо по идентификатору (GET /api/Customers) . . . . .	37
5.3.1	Параметры . . . . .	37
5.3.2	Возможные статусы ответов . . . . .	38
5.3.3	Возвращаемые данные . . . . .	38
5.3.4	Пример использования . . . . .	38
5.4	Создать ответственное лицо (POST /api/Customers) . . . . .	38
5.4.1	Параметры . . . . .	38
5.4.2	Тело запроса . . . . .	39
5.4.3	Возможные статусы ответов . . . . .	39
5.4.4	Возвращаемые данные . . . . .	39
5.4.5	Пример использования . . . . .	39
5.5	Изменить ответственное лицо (PUT /api/Customers) . . . . .	40
5.5.1	Параметры . . . . .	40
5.5.2	Тело запроса . . . . .	40
5.5.3	Возможные статусы ответов . . . . .	40
5.5.4	Возвращаемые данные . . . . .	40
5.5.5	Пример использования . . . . .	40
5.6	Удалить ответственное лицо (DELETE /api/Customers) . . . . .	41
5.6.1	Параметры . . . . .	41
5.6.2	Возможные статусы ответов . . . . .	41
5.6.3	Возвращаемые данные . . . . .	41
5.6.4	Пример использования . . . . .	41
<b>6</b>	<b>Дополнительные характеристики</b>	<b>42</b>
6.1	Поля дополнительной характеристики . . . . .	42
6.2	Получить список дополнительных характеристик объекта (GET /api/ExtFields) . . . . .	42
6.2.1	Параметры . . . . .	42
6.2.2	Возможные статусы ответов . . . . .	42
6.2.3	Возвращаемые данные . . . . .	43
6.2.4	Пример использования . . . . .	43
6.3	Получить дополнительную характеристику по идентификатору (GET /api/ExtFields) . . . . .	43
6.3.1	Параметры . . . . .	43

6.3.2	Возможные статусы ответов . . . . .	44
6.3.3	Возвращаемые данные . . . . .	44
6.3.4	Пример использования . . . . .	44
6.4	Создать дополнительную характеристику (POST /api/ExtFields) . . . . .	44
6.4.1	Параметры . . . . .	44
6.4.2	Тело запроса . . . . .	45
6.4.3	Возможные статусы ответов . . . . .	45
6.4.4	Возвращаемые данные . . . . .	45
6.4.5	Пример использования . . . . .	45
6.5	Изменить дополнительную характеристику (PUT /api/ExtFields) . . . . .	45
6.5.1	Параметры . . . . .	45
6.5.2	Тело запроса . . . . .	46
6.5.3	Возможные статусы ответов . . . . .	46
6.5.4	Возвращаемые данные . . . . .	46
6.5.5	Пример использования . . . . .	46
6.6	Удалить дополнительную характеристику (DELETE /api/ExtFields) . . . . .	46
6.6.1	Параметры . . . . .	46
6.6.2	Возможные статусы ответов . . . . .	47
6.6.3	Возвращаемые данные . . . . .	47
6.6.4	Пример использования . . . . .	47
<b>7</b>	<b>Проверка КТС</b>	<b>48</b>
7.1	Начать проверку КТС (POST /api/CheckPanic) . . . . .	48
7.1.1	Параметры . . . . .	48
7.1.2	Возможные статусы ответов . . . . .	48
7.1.3	Возвращаемый результат . . . . .	49
7.1.4	Пример использования . . . . .	49
7.2	Получить результат проверки КТС (GET /api/CheckPanic) . . . . .	49
7.2.1	Параметры . . . . .	50
7.2.2	Возможные статусы ответов . . . . .	50
7.2.3	Возвращаемые результаты . . . . .	50
7.2.4	Пример использования . . . . .	51
<b>8</b>	<b>События объекта</b>	<b>52</b>
8.1	Поля события . . . . .	52
8.1.1	Значения поля EventClassType . . . . .	52
8.1.2	Формат полей с датой и временем . . . . .	53
8.2	Получить список событий объекта (GET /api/SiteEvents) . . . . .	53
8.2.1	Параметры . . . . .	53

8.2.2	Тело запроса . . . . .	53
8.2.3	Возможные статусы ответов . . . . .	55
8.2.4	Возвращаемые данные . . . . .	55
8.2.5	Пример использования . . . . .	55
<b>9</b>	<b>Расписание охраны объекта</b>	<b>57</b>
9.1	Поля расписания охраны объекта . . . . .	57
9.1.1	Расписание охраны объекта . . . . .	57
9.1.2	Интервал расписания объекта (Interval) . . . . .	57
9.2	Получить расписание охраны объекта (GET /api/SiteSchedule) . . . . .	58
9.2.1	Параметры . . . . .	58
9.2.2	Возможные статусы ответов . . . . .	58
9.2.3	Возвращаемые данные . . . . .	58
9.2.4	Пример использования . . . . .	58
9.3	Создать/изменить расписание охраны объекта (POST /api/SiteSchedule) . . . . .	59
9.3.1	Параметры . . . . .	59
9.3.2	Тело запроса . . . . .	59
9.3.3	Возможные статусы ответов . . . . .	60
9.3.4	Возвращаемые данные . . . . .	60
9.3.5	Пример использования . . . . .	60
9.4	Удалить расписание охраны объекта (DELETE /api/SiteSchedule) . . . . .	61
9.4.1	Параметры . . . . .	61
9.4.2	Возможные статусы ответов . . . . .	61
9.4.3	Возвращаемые данные . . . . .	62
9.4.4	Пример использования . . . . .	62
<b>10</b>	<b>Управление доступом в MyAlarm</b>	<b>63</b>
10.1	Структуры данных управления доступом в MyAlarm . . . . .	63
10.1.1	Пользователь MyAlarm . . . . .	63
10.1.2	Объект пользователя MyAlarm . . . . .	63
10.2	Структуры данных классов событий, действий и отмен . . . . .	64
10.2.1	Класс события с полями . . . . .	64
10.2.2	Действие или отмена с полями . . . . .	64
10.3	Основные методы http-api . . . . .	64
10.3.1	Получить список пользователей MyAlarm объекта (GET /api/MyAlarm) . . . . .	64
10.3.2	Получить список объектов пользователя MyAlarm (GET /api/MyAlarm/UserObjects) . . . . .	65
10.3.3	Предоставить/забрать доступ пользователю к MyAlarm (PUT /api/MyAlarm) . . . . .	66
10.3.4	Модифицировать право на использование виртуальной КТС (PUT /api/MyAlarm) . . . . .	67
10.4	Управление подписками . . . . .	68

10.4.1	Получить список подписок объекта на классы событий (GET /api/MyAlarm/EventClass) . . . . .	68
10.4.2	Получить список подписок объекта на действия и отмены (GET /api/MyAlarm/UserAction) . . . . .	69
10.4.3	Модификация подписок объекта на классы событий (PUT /api/MyAlarm/EventClass) . . . . .	70
10.4.4	Модификация подписок объекта на действия и отмены (PUT /api/MyAlarm/UserAction) . . . . .	70
10.5	Списки классов событий, действий и отмен . . . . .	71
10.5.1	Получить список классов событий ЦО (GET /api/EventClasses) . . . . .	71
10.5.2	Получить список действий и отмен ЦО (GET /api/UserActions) . . . . .	72
<b>11</b>	<b>Тревоги</b> . . . . .	<b>74</b>
11.1	Поля тревоги . . . . .	74
11.1.1	Формат полей с датой и временем . . . . .	74
11.2	Поля действия или отмены . . . . .	74
11.3	Получить список тревог за период (GET /api/Alarms) . . . . .	75
11.3.1	Параметры . . . . .	75
11.3.2	Возможные статусы ответов . . . . .	76
11.3.3	Возвращаемые данные . . . . .	76
11.3.4	Пример выполнения запроса . . . . .	76
11.4	Получить список действий или отмен по индексу тревоги (GET /api/Alarms) . . . . .	76
11.4.1	Параметры . . . . .	77
11.4.2	Возможные статусы ответов . . . . .	77
11.4.3	Возвращаемые данные . . . . .	77
11.4.4	Пример выполнения запроса . . . . .	77
<b>12</b>	<b>Транспортный файл</b> . . . . .	<b>78</b>
12.1	Что такое транспортный файл . . . . .	78
12.2	Какие поля содержит транспортный файл . . . . .	78
12.3	Что такое уровень информирования . . . . .	79
12.4	Как создать транспортный файл . . . . .	79
12.5	Можно ли изменить только некоторые поля с помощью файла . . . . .	80
12.6	Как изменить значения в нескольких карточках объекта . . . . .	80
12.7	Как загрузить транспортный файл в Центр охраны . . . . .	81
12.8	Как настроить службу CNord.WebApi.Service . . . . .	81
12.8.1	Что произойдет после настройки службы «C.Nord HTTP-API Service» . . . . .	81
<b>13</b>	<b>Техническая поддержка</b> . . . . .	<b>82</b>



# 1 Общая информация

В документе описаны методы API, предоставляемые службой «C.Nord HTTP-API Service», являющейся частью программного обеспечения «Центр охраны». API организован в виде REST-сервисов, описание ресурсов и методов доступа к ним дано в последующих разделах.

## 1.1 Ограничение доступа

Для доступа к методам API в заголовке каждого запроса должен присутствовать параметр «apiKey» со значением, равным ключу безопасности сервиса «C.Nord HTTP-API Service», например: `apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c`. Если ключ доступа не соответствует установленному, то методы будут возвращать код 403 (см. «[Статусы ответов](#)»).

## 1.2 Настройки службы «C.Nord HTTP-API Service»

Параметры службы задаются в конфигурационном файле `CNord.WebApi.Service.exe.config`, расположенном в папке установки «Центра охраны». Файл в формате XML, настройки службы находятся в разделе `CNord.WebApi.ApiSettings`. Для изменения значения параметра необходимо изменить значение элемента «value» соответствующего элемента `setting`.

### 1.2.1 Url

Протокол, локальный адрес и порт, используя которые служба будет принимать запросы.

Для того, чтобы служба принимала запросы из сети на всех имеющихся IP-адресах, необходимо указать значение `http://*:9200`.

**Значение по умолчанию:** `http://localhost:9200`

### 1.2.2 ExternalUrl

Внешний адрес, по которому к службе обращаются клиенты. От этого адреса строятся адреса в предоставляемых пользователям ссылках. Может не совпадать с локальным адресом (Url) если служба расположена за реверс-прокси.

**Значение по умолчанию:** `http://localhost:9200`

### 1.2.3 TransportFileEncoding

Название кодировки транспортного файла. Возможные значения:

- UTF-8;
- Unicode;
- Windows-1251.

**Значение по умолчанию:** UTF-8

### 1.2.4 TransportFileCheckInterval

Интервал проверки наличия нового транспортного файла, в формате hh:mm:ss. Не может быть задано менее 10 секунд.

**Значение по умолчанию:** 00:10:00

### 1.2.5 CheckTransportFileOnStartup

Проверять ли наличие нового транспортного файла сразу при старте сервиса или ожидать истечения интервала (True/False)

**Значение по умолчанию:** True

### 1.2.6 TransportFilePath

Полный путь к транспортному файлу. Если указывать относительный, то адрес будет проверяться от рабочей папки сервиса, которая находится в системной папке Windows.

**Значение по умолчанию:** sc-api-tf.csv

### 1.2.7 TransportFileTempFile

Путь к временному файлу для хранения данных о последнем загруженном транспортном файле (используется для проверки наличия изменений в новой версии файла). При отсутствии файла он создается (если есть права создания по указанному пути). Рекомендуется указывать полный путь. Если указывать относительный путь, то адрес будет проверяться от рабочей папки сервиса, которая находится в системной папке Windows.

**Значение по умолчанию:** sc-api-tf.tmp

### 1.2.8 ApiKey

Ключ безопасности службы, который должен передаваться в заголовке любого запроса. Допустимое значение - произвольная строка. Если указано пустое значение, то обработка запросов будет отключена.

**Значение по умолчанию:** не задано (пустое)

### 1.2.9 DefaultPageSize

Значение для параметра «Размер страницы списка объектов», используемое по умолчанию, если клиент не передал в метод [GET /api/Sites](#) значение для параметра `pageSize`.

**Значение по умолчанию:** 100

### 1.2.10 DefaultCheckPanicInterval

Значение для параметра «Интервал проверки КТС», используемое по умолчанию, если клиент не передал в метод [POST /api/CheckPanic](#) значение для параметра `checkInterval`).

Значение задается в секундах.

**Допустимый диапазон значений:** [30, 180]

**Значение по умолчанию:** 120

### 1.2.11 DefaultCheckStopOnEvent

Значение для параметра «Прекращать проверку КТС при получении события» используемое по умолчанию, если клиент не передал в метод `POST /api/CheckPanic` значение для параметра `stopOnEvent`).

Возможные значения: `True` / `False`.

Значение по умолчанию: `False`

## 1.3 Общий вид запросов

Методы API доступны посредством HTTP-запросов по URL вида:

`http://<хост>:<порт>/api/<methodName>`

Например:

`http://10.7.22.128:9002/api/Sites`

Если метод API требует передать параметры в теле запроса, то параметры должны передаваться в формате JSON.

Если метод API возвращает результат в теле ответа, то результат также возвращается в формате JSON.

## 1.4 Ключ безопасности

В заголовке HTTP-запроса к *любому* методу API должно быть указано значение параметра `apiKey`, которое должно совпадать со значением одноименного параметра в настройках службы.

Если значение для параметра `apiKey` не задано или указано неверно, вызов любого метода API приведет к ошибке со статусом 403.

## 1.5 Статусы ответов

Коды, приведенные в таблице ниже, соответствуют кодам статуса протокола HTTP (HTTP Status Codes).

Код	Описание
200	Запрос успешно выполнен
201	Запрос на создание новой сущности успешно выполнен
400	Ошибка при выполнении запроса. Подробное описание – в результате (см. ниже).
403	Доступ запрещен (отсутствует или указан неправильный ключ безопасности)

## 1.6 Код 400: описание ошибки

Если при выполнении запроса возникла ошибка с кодом 400, то в ответе будет и описание возникшей ошибки. Тип – `application/json`.

```
{
  "Message": string,
  "SpResultCode": int
}
```

### **1.6.1 Message**

Текстовая строка с описанием возникшей ошибки.

### **1.6.2 SpResultCode**

Внутренний код ошибки. В большинстве случаев соответствует описанию ошибки.

## 2 Объект

Методы для управления объектами в базе данных «Центра охраны»: получение списка объектов или отдельного объекта, создание, изменение и удаление объекта.

### 2.1 Поля объекта

Элемент JSON, содержащий поля объекта:

```
{
  "Id": string,
  "AccountNumber": number,
  "Name": string,
  "ObjectPassword": string,
  "Address": string,
  "Phone1": string,
  "Phone2": string,
  "TypeName": string,
  "IsFire": boolean,
  "IsArm": boolean,
  "IsPanic": boolean,
  "DeviceTypeName": string,
  "EventTemplateName": string,
  "ContractNumber": string,
  "ContractPrice": number,
  "MoneyBalance": number,
  "PaymentDate": string,
  "DebtInformLevel": number,
  "Disabled": boolean,
  "DisableReason": number,
  "DisableDate": string,
  "AutoEnable": boolean,
  "AutoEnableDate": string,
  "CustomersComment": string,
  "CommentForOperator": string,
  "CommentForGuard": string,
  "MapFileName": string,
  "WebLink": string,
  "ControlTime": number,
  "CTIgnoreSystemEvent": bool,
  "IsContractPriceForceUpdate": boolean,
  "IsMoneyBalanceForceUpdate": boolean,
  "IsPaymentDateForceUpdate": boolean
}
```

Название поля	Тип	Поле в карточке объекта; примечание
Id	string	Идентификатор объекта
AccountNumber	number	Номер объекта (почти всегда совпадает с номером, запрограммированным в контрольную панель, установленную на объекте)
Name	string	Название объекта

ObjectPassword	string	Пароль
Address	string	Адрес объекта
Phone1	string	Телефон 1
Phone2	string	Телефон 2
TypeName	string	Название типа объекта
IsFire	boolean	Флаг наличия пожарной сигнализации на объекте
IsArm	boolean	Флаг наличия охранной сигнализации на объекте
IsPanic	boolean	Флаг наличия тревожной кнопки на объекте
DeviceTypeName	string	Псевдоним типа оборудования на объекте. Возможные значения: см. ниже
EventTemplateName	string	Название шаблона событий объекта
ContractNumber	string	Номер договора
ContractPrice	number	Сумма ежемесячного платежа по договору. Отображается в приложении MyAlarm
MoneyBalance	number	Баланс лицевого счета. Отображается в приложении MyAlarm
PaymentDate	number	Дата ближайшего списания средств. Отображается в приложении MyAlarm
DebtInformLevel	number	Уровень информирования клиента о состоянии услуг охраны. Отображается в приложении MyAlarm. Возможные значения: см. ниже
Disabled	boolean	Флаг: объект отключен
DisableReason	number	Код: причина отключения объекта (не используется)
DisableDate	string	Дата отключения объекта
AutoEnable	boolean	Флаг: необходимо автоматически включить объект
AutoEnableDate	string	Дата автоматического включения объекта. Имеет значение только в том случае, если поле «AutoEnable» установлено в значение «True»
CustomersComment	string	Комментарий к списку ответственных
CommentForOperator	string	Комментарий для оператора
CommentForGuard	string	Комментарий для ГБР
MapFileName	string	Путь к файлу с картой объекта
WebLink	string	Web-ссылка: ссылка на ресурс с дополнительной информацией об объекте
ControlTime	number	Общее контрольное время (мин.)
CTIgnoreSystemEvent	bool	Игнорировать системные события
IsContractPriceForceUpdate	boolean	Признак принудительной записи поля ContractPrice (необходимо выставить true и пропустить поле ContractPrice при очистке)
IsMoneyBalanceForceUpdate	boolean	Признак принудительной записи поля MoneyBalance (необходимо выставить true и пропустить поле MoneyBalance при очистке)
IsPaymentDateForceUpdate	boolean	Признак принудительной записи поля PaymentDate (необходимо выставить true и пропустить поле PaymentDate при очистке)

### 2.1.1 Значения поля DeviceTypeName

Псевдоним типа оборудования на объекте.

Соответствия значений псевдонима и названий типа оборудования:

Псевдоним	Название типа оборудования
cnord-gsm-cml	«Си-Норд GSM (CML)»
lonta-202	«Lonta-202»
rs200	«RS200»
alarmview	«AlarmView»
puper-type-5	«Puper type 5»
neman	«Neman»
ritm	«Ритм»
other	«Другое»

Тип оборудования, установленного на объекте, указывается в модуле «Менеджер объектов», на вкладке «Оборудование».

### 2.1.2 Значения поля DebtInformLevel

Уровень информирования клиента о состоянии услуг охраны.

Возможные значения:

Значение	Описание
-1	Не отображать информацию в MyAlarm
0	Нет задолженности
1	Информировать о необходимости оплаты
2	Предупреждать о наличии задолженности
3	Запретить взятие под охрану и ограничить функции MyAlarm

### 2.1.3 Формат полей с датой и временем

Для полей PaymentDate, DisableDate, AutoEnableDate допустимые следующие форматы строковых значений:

- “YYYY-MM-DD”;
- “YYYY-MM-DDTHH:mm:ss”;
- “YYYY-MM-DDTHH:mm:ssZ”;
- “YYYY-MM-DDTHH:mm:ss.fffZ”.

## 2.2 Получить список объектов (GET /api/Sites)

Метод предназначен для получения списка объектов. В качестве критерия для поиска объекта может использоваться номер договора.

URL : /api/Sites

Метод : GET

### 2.2.1 Параметры

#### **contractNumber**

Необязательный параметр.

Номер договора, который должен быть указан для возвращаемых объектов. Если значение для параметра не указано, то поиск объектов по номеру договора выполняться не будет.

#### **startFrom**

Необязательный параметр.

С какого по счету элемента возвращать данные. Если значение для параметра не указано, то метод вернет список, начиная с первого элемента.

Параметр **startFrom** совместно с параметром **pageSize** может использоваться, для получения большого списка объектов по частям («**пагинация**»).

#### **pageSize**

Необязательный параметр.

Максимальное количество элементов, которое необходимо вернуть в результате выполнения запроса. Если значение для параметра не указано, то метод вернет количество элементов, не превышающее значение параметра **DefaultPageSize**, указанное в настройках службы «C.Nord HTTP-API Service». Примечание: параметр **DefaultPageSize** не может быть больше 200.

#### **userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 2.2.2 Возможные статусы ответов

200, 403 – см. «**Статусы ответов**».

### 2.2.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с **полями объекта**.

### 2.2.4 Пример использования

Пример выполнения запроса, в котором указано значение для параметра **contractNumber**

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/Sites?contractNumber=2018-12/91&  
        'userName=crm-Ivanova-A-A'
```

Status: 200



```
[
{
  "Id": "94df3af9-36c1-423b-aa88-fb505bda3fa4",
  "AccountNumber": 265,
  "Name": "Вестколл Северо-Запад",
  "ObjectPassword": "1234",
  "Address": "Митрофаньевское шоссе д.2 кор.2 лит.А",
  "Phone1": "нет",
  "Phone2": "",
  "TypeName": "магазин",
  "IsFire": true,
  "IsArm": true,
  "IsPanic": true,
  "DeviceTypeName": "other",
  "EventTemplateName": "Си-Норд GSM",
  "ContractNumber": "2018-12/91",
  "ContractPrice": 0,
  "MoneyBalance": 0,
  "PaymentDate": "1899-12-30T00:00:00",
  "DebtInformLevel": -1,
  "Disabled": false,
  "DisableReason": 0,
  "DisableDate": "1899-12-30T00:00:00",
  "AutoEnable": false,
  "AutoEnableDate": "1899-12-30T00:00:00",
  "CustomersComment": "",
  "CommentForOperator": "",
  "CommentForGuard": "",
  "MapFileName": "",
  "WebLink": "",
  "ControlTime": 0,
  "CTIgnoreSystemEvent": false
},
{
  "Id": "524bf1a5-76ce-43a7-9ed5-56291750933c",
  "AccountNumber": 282,
  "Name": "Инвест-Москва",
  "ObjectPassword": "4321",
  "Address": "Шостаковича ул. д. 3 к. 1",
  "Phone1": "785-03-39",
  "Phone2": "",
  "TypeName": "банк",
  "IsFire": true,
  "IsArm": true,
  "IsPanic": true,
  "DeviceTypeName": "cnord-gsm-cml",
  "EventTemplateName": "Си-Норд GSM",
  "ContractNumber": "2018-12/91",
  "ContractPrice": 0,
  "MoneyBalance": 0,
  "PaymentDate": "1899-12-30T00:00:00",
  "DebtInformLevel": -1,
  "Disabled": false,
  "DisableReason": 0,
  "DisableDate": "1899-12-30T00:00:00",
  "AutoEnable": false,

```

```

        "AutoEnableDate": "1899-12-30T00:00:00",
        "CustomersComment": "",
        "CommentForOperator": "",
        "CommentForGuard": "",
        "MapFileName": "",
        "WebLink": "",
        "ControlTime": 1,
        "CTIgnoreSystemEvent": true
    }
]

```

## 2.3 Получить объект по номеру или идентификатору (GET /api/Sites)

Метод предназначен для получения информации о конкретном объекте. Для поиска объекта может быть использован его идентификатор или номер.

**URL :** /api/Sites

**Метод :** GET

### 2.3.1 Параметры

**id**

Обязательный параметр.

Идентификатор или номер объекта, информацию о котором нужно получить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 2.3.2 Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### 2.3.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с [полями объекта](#).

### 2.3.4 Пример использования

**Пример выполнения запроса, в котором в качестве параметра указан идентификатор объекта**

```

curl --request GET \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url 'http://10.7.22.128:9002/api/Sites?id=524bf1a5-76ce-43a7-9ed5-56291750933c&'
      'userName=crm-Ivanova-A-A'

```

**Status:** 200

```
{
  "Id": "524bf1a5-76ce-43a7-9ed5-56291750933c",
  "AccountNumber": 282,
  "Name": "Инвест-Москва",
  "ObjectPassword": "1234",
  "Address": "Шостаковича ул. д. 3 к. 1",
  "Phone1": "785-03-39",
  "Phone2": "",
  "TypeName": "банк",
  "IsFire": true,
  "IsArm": true,
  "IsPanic": true,
  "DeviceTypeName": "cnord-gsm-cml",
  "EventTemplateName": "Си-Норд GSM",
  "ContractNumber": "",
  "ContractPrice": 0,
  "MoneyBalance": 0,
  "PaymentDate": "1899-12-30T00:00:00",
  "DebtInformLevel": -1,
  "Disabled": false,
  "DisableReason": 0,
  "DisableDate": "1899-12-30T00:00:00",
  "AutoEnable": false,
  "AutoEnableDate": "1899-12-30T00:00:00",
  "CustomersComment": "При тревоге звонить Иванову.",
  "CommentForOperator": "",
  "CommentForGuard": "",
  "MapFileName": "",
  "WebLink": "",
  "ControlTime": 10,
  "CTIgnoreSystemEvent": false
}
```

Пример запроса, в котором в качестве параметра указан номер объекта

```
curl --request GET \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url 'http://10.7.22.128:9002/api/Sites?id=282c&'
  'userName=crm-Ivanova-A-A'
```

## 2.4 Создать объект (POST /api/Sites)

Метод предназначен для создания нового объекта.

**URL** : /api/Sites

**Метод** : POST

### 2.4.1 Параметры

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 2.4.2 Тело запроса

В теле запроса должен быть передан элемент JSON с [полями объекта](#), который нужно создать. При создании объекта обязательно указать только название для нового объекта. Значение для всех остальных полей, включая номер объекта, можно не указывать: будет использовано значение по умолчанию.

Если номер объекта не указан, то новому объекту будет присвоен свободный номер по формуле  $\max(\text{AccountNumber}) + 1$ .

Если номер объекта указан и объект с таким номером уже есть в базе данных «Центра охраны», то метод вернет ошибку.

### 2.4.3 Возможные статусы ответов

201, 400, 403 – см. [«Статусы ответов»](#).

### 2.4.4 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с информацией об идентификаторе и номере, которые получил созданный объект:

```
{
  "Id": string,
  "AccountNumber": number
}
```

Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в теле ответа – см. [«Код 400: описание ошибки»](#)

### 2.4.5 Пример использования

**Пример выполнения запроса, в котором номер для нового объекта не указан**

```
curl --request POST \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url http://192.168.0.2:9002/api/Sites/ \
  --data '{"Name": "Ресторан \У Пальча\","Address": "Виноградная, 26",'
        '"Phone1": "+7 812 327 1633","ObjTypeName": "Ресторан",'
        '"DeviceTypeName": "cnord-gsm-cml",'
        '"EventTemplateName": "Си-Норд GSM"}'
```

Status : 201

```
{
  "Id": "a4835f0e-8909-42fc-a878-217aef0a3aa2",
  "AccountNumber": 9001
}
```

Пример выполнения запроса, в котором указан номер существующего объекта

```
curl --request POST \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url http://192.168.0.2:9002/api/Sites/ \
  --data '{"AccountNumber": 9001, '
    'Name': "Ресторан \"У Михалыча\"", "Address": "Виноградная, 26", '
    'Phone1': "+7 812 327 1633", "ObjTypeName": "Ресторан", '
    'DeviceTypeName': "cnord-gsm-cml", '
    'EventTemplateName': "Си-Норд GSM"}'
```

Status : 400

```
{
  "Message": "Вставка: объект с указанным номером уже существует.",
  "SpResultCode": 205
}
```

## 2.5 Изменить объект (PUT /api/Sites)

Метод предназначен для изменения полей объекта.

URL : /api/Sites

Метод : PUT

### 2.5.1 Параметры

**id**

Обязательный параметр.

Идентификатор объекта, который нужно изменить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 2.5.2 Тело запроса

В теле запроса должен быть передан элемент JSON с **полями объекта**, которые нужно изменить. Если поле необходимо оставить без изменения, то оно не должно быть указано в элементе JSON.

В случае если поле необходимо очистить, то оно не должно быть указано и должен быть выставлен соответствующий признак принудительной записи поля. Список полей которые можно очистить представлен в таблице ниже:

Название поля	Тип	Признак принудительной записи (boolean)
ContractPrice	decimal	IsContractPriceForceUpdate
MoneyBalance	decimal	IsMoneyBalanceForceUpdate
PaymentDate	string	IsPaymentDateForceUpdate

Идентификатор объекта передается параметром в заголовке запроса, поэтому поле `id` в элементе JSON может быть не указано или будет проигнорировано.

### 2.5.3 Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### 2.5.4 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с [полями объекта](#) – с учетом изменений, которые произошли в результате выполнения запроса.

Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в теле ответа – см. «[Код 400: описание ошибки](#)»

### 2.5.5 Пример использования

```
curl --request PUT \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url 'http://192.168.0.2:9002/api/Sites?id=524bf1a5-76ce-43a7-9ed5-56291750933c&'
  'userName=crm-Ivanova-A-A' \
  --data '{"Name": "Инвест-Москва-35", "Address": "Шостаковича ул. д. 3 к. 1", '
  '{"Phone1": "785-03-39", "TypeName": "Банк"}'}
```

## 2.6 Удалить объект (DELETE /api/Sites)

Метод предназначен для удаления объекта из базы данных «Центра охраны».

**URL :** /api/Sites

**Метод :** DELETE

### 2.6.1 Параметры

**id**

Обязательный параметр.

Идентификатор объекта, который нужно удалить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 2.6.2 Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### 2.6.3 Возвращаемые данные

При успешном выполнении метод не возвращает данных. Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в ответе – см. «[Код 400: описание ошибки](#)»

#### 2.6.4 Пример использования

```
curl --request DELETE \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.0.0.2:9002/api/Sites?id=68b3fc63-10b7-4eeb-9308-769c1fb1e6d3&' \  
    'userName=crm-Ivanova-A-A'
```

## 3 Раздел

Методы для управления разделами объекта: получение списка разделов, создание, изменение или удаление раздела объекта.

### 3.1 Поля раздела

Элемент JSON, содержащий все поля раздела:

```
{
  "Id": string,
  "PartNumber": number,
  "ObjectNumber": number,
  "PartDesc": string,
  "PartEquip": string
}
```

Название поля	Тип	Поле в карточке объекта; примечание
Id	string	Идентификатор раздела
PartNumber	number	Номер раздела (обязательный при создании, натуральное число, почти всегда совпадает с номером, запрограммированным в контрольную панель, установленную на объекте)
ObjectNumber	number	Объектовый номер раздела. Используется только для объектовых приборов, поддерживающих индивидуальные объектовые номера для разделов
PartDesc	string	Название (описание) раздела (обязательный при создании, не может быть пустым)
PartEquip	string	Название (описание) оборудования, установленного в разделе

### 3.2 Получить список разделов объекта (GET /api/Parts)

Метод предназначен для получения списка разделов объекта.

URL : /api/Parts

Метод : GET

#### 3.2.1 Параметры

**siteId**

Обязательный параметр.

Идентификатор объекта, список разделов которого нужно получить. Соответствует полю Id элемента JSON с [полями объекта](#).

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.



### 3.2.2 Возможные статусы ответов

200, 403 – см. «Статусы ответов».

### 3.2.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с [полями раздела](#).

### 3.2.4 Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/Parts?siteId=b8144107-31d1-4800-b83d-764f015a54a5&' \  
      'userName=crm-Ivanova-A-A'
```

Status: 200

```
[  
  {  
    "Id": "524bf1a5-76ce-43a7-9ed5-56291750933d",  
    "PartNumber": 1,  
    "ObjectNumber": 0,  
    "PartDesc": "Вход и периметр",  
    "PartEquip": ""  
  },  
  {  
    "Id": "524bf1a5-76ce-43a7-9ed5-56291750933e",  
    "PartNumber": 2,  
    "ObjectNumber": 0,  
    "PartDesc": "Внутренние помещения",  
    "PartEquip": ""  
  },  
]
```

## 3.3 Получить раздел по идентификатору (GET /api/Parts)

Метод предназначен для получения информации о конкретном разделе. Для поиска раздела должен быть использован его идентификатор.

URL : /api/Parts

Метод : GET

### 3.3.1 Параметры

**id**

Обязательный параметр.

Идентификатор раздела, информацию о котором нужно получить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 3.3.2 Возможные статусы ответов

200, 400, 403 – см. «Статусы ответов».

### 3.3.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с полями раздела.

### 3.3.4 Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/Parts?id=524bf1a5-76ce-43a7-9ed5-56291750933e' \  
    'userName=crm-Ivanova-A-A'
```

Status: 200

```
{  
  "Id": "524bf1a5-76ce-43a7-9ed5-56291750933e",  
  "PartNumber": 2,  
  "ObjectNumber": null,  
  "PartDesc": "Внутренние помещения",  
  "PartEquip": null  
}
```

## 3.4 Создать раздел (POST /api/Parts)

Метод предназначен для создания нового раздела для объекта.

URL : /api/Parts

Метод : POST

### 3.4.1 Параметры

**siteId**

Обязательный параметр.

Идентификатор объекта, для которого нужно создать объект. Соответствует полю Id элемента JSON с полями объекта.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 3.4.2 Тело запроса

В теле запроса должен быть передан элемент JSON с [полями раздела](#), который нужно создать. При создании раздела обязательно указать название, номер раздела и описание. Значения для всех остальных полей можно не указывать: будет использовано значение по умолчанию.

Если указан идентификатор раздела, то он будет проигнорирован.

Если номер раздела указан и раздела с таким номером уже есть в списке разделов объекта, то метод вернет ошибку.

### 3.4.3 Возможные статусы ответов

201, 400, 403 – см. [«Статусы ответов»](#).

### 3.4.4 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с информацией об идентификаторе, который получил созданный раздел:

```
{
  "Id": string
}
```

Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в теле ответа – см. [«Код 400: описание ошибки»](#)

### 3.4.5 Пример использования

```
curl --request POST \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url 'http://10.7.22.128:9002/api/Parts?siteId=b8144107-31d1-4800-b83d-764f015a54a5&'
  'userName=crm-Ivanova-A-A' \
  --data '{"PartNumber": 3,"PartDesc": "Подвал"}'
```

Status : 201

```
{
  "Id": "524bf1a5-76ce-43a7-9ed5-56291750933f"
}
```

## 3.5 Изменить раздел (PUT /api/Parts)

Метод предназначен для изменения полей раздела.

URL : /api/Parts

Метод : PUT

### 3.5.1 Параметры

id

Обязательный параметр.

Идентификатор раздела, который нужно изменить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 3.5.2 Тело запроса

В теле запроса должен быть передан элемент JSON с [полями раздела](#), которые нужно изменить. Если поле необходимо оставить без изменения, то оно не должно быть указано в элементе JSON.

Идентификатор объекта передается параметром в заголовке запроса, поэтому поле `id` в элементе JSON может быть не указано или будет проигнорировано.

### 3.5.3 Возможные статусы ответов

200, 400, 403 – см. [«Статусы ответов»](#).

### 3.5.4 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с [полями раздела](#) – с учетом изменений, которые произошли в результате выполнения запроса.

Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в теле ответа – см. [«Код 400: описание ошибки»](#)

### 3.5.5 Пример использования

```
curl --request PUT \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/Parts?id=524bf1a5-76ce-43a7-9ed5-56291750933f&' \  
    'userName=crm-Ivanova-A-A' \  
  --data '{"PartDesc": "Чердак"}'
```

## 3.6 Удалить раздел (DELETE /api/Parts)

Метод предназначен для удаления раздела объекта.

**URL** : /api/Parts

**Метод** : DELETE

### 3.6.1 Параметры

**id**

Обязательный параметр.

Идентификатор раздела, который нужно удалить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 3.6.2 Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### 3.6.3 Возвращаемые данные

При успешном выполнении метод не возвращает данных. Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в ответе – см. «[Код 400: описание ошибки](#)»

### 3.6.4 Пример использования

```
curl --request DELETE \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/Parts?id=524bf1a5-76ce-43a7-9ed5-56291750933f&' \  
    'userName=crm-Ivanova-A-A'
```

## 4 Шлейфы

Методы для управления шлейфами объекта: получение списка шлейфов, создание, изменение или удаление шлейфа объекта.

### 4.1 Поля шлейфа

Элемент JSON, содержащий все поля шлейфа:

```
{
  "Id": string,
  "ZoneNumber": number,
  "ZoneDesc": string,
  "ZoneEquip": string,
}
```

Название поля	Тип	Поле в карточке объекта; примечание
Id	string	Идентификатор шлейфа
ZoneNumber	number	Номер шлейфа (обязательный при создании, натуральное число)
ZoneDesc	string	Описание шлейфа (обязательный при создании, не может быть пустым)
ZoneEquip	string	Оборудование шлейфа

### 4.2 Получить список шлейфов объекта (GET /api/Zones)

Метод предназначен для получения списка шлейфов объекта.

URL : /api/Zones

Метод : GET

#### 4.2.1 Параметры

**siteId**

Обязательный параметр.

Идентификатор объекта, список шлейфов которого нужно получить. Соответствует полю Id элемента JSON с [полями объекта](#).

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

#### 4.2.2 Возможные статусы ответов

200, 403 – см. «[Статусы ответов](#)».

### 4.2.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с [полями шлейфа](#).

### 4.2.4 Пример использования

```
curl --request GET \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url 'http://10.7.22.128:9002/api/Zones?siteId=b8144107-31d1-4800-b83d-764f015a54a5&'
  'userName=crm-Ivanova-A-A'
```

Status: 200

```
[
  {
    "Id": "e6f14ede-65d3-45f7-a79e-80ebaf6f1301",
    "ZoneNumber": 3,
    "ZoneDesc": "Описание шлейфа",
    "ZoneEquip": "Оборудование шлейфа"
  },
  {
    "Id": "e6f14ede-65d3-45f7-a79e-80ebaf6f1302",
    "ZoneNumber": 4,
    "ZoneDesc": "Описание шлейфа 2",
    "ZoneEquip": "Оборудование шлейфа 2"
  }
]
```

## 4.3 Получить шлейф по идентификатору (GET /api/Zones)

Метод предназначен для получения информации о конкретном шлейфе. Для поиска шлейфа должен быть использован его идентификатор.

URL : /api/Zones

Метод : GET

### 4.3.1 Параметры

**id**

Обязательный параметр.

Идентификатор шлейфа, информацию о котором нужно получить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 4.3.2 Возможные статусы ответов

200, 400, 403 — см. [«Статусы ответов»](#).

### 4.3.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с [полями шлейфа](#).

### 4.3.4 Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/Zones?id=524bf1a5-76ce-43a7-9ed5-56291750933e&' \  
      'userName=crm-Ivanova-A-A'
```

Status: 200

```
{  
  "Id": "524bf1a5-76ce-43a7-9ed5-56291750933e",  
  "ZoneNumber": 3,  
  "ZoneDesc": "Описание шлейфа",  
  "ZoneEquip": "Оборудование шлейфа"  
}
```

## 4.4 Создать шлейф (POST /api/Zones)

Метод предназначен для создания нового шлейфа для объекта.

URL : /api/Zones

Метод : POST

### 4.4.1 Параметры

**siteId**

Обязательный параметр.

Идентификатор объекта, для которого нужно создать объект. Соответствует полю Id элемента JSON с [полями объекта](#).

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 4.4.2 Тело запроса

В теле запроса должен быть передан элемент JSON с [полями шлейфа](#), который нужно создать. Значения для всех полей, можно не указывать: будет использовано значение по умолчанию.

Если указан идентификатор шлейфа, то он будет проигнорирован.

Если номер шлейфа указан и шлейф с таким номером уже есть в списке шлейфов объекта, то метод вернет ошибку.



### 4.4.3 Возможные статусы ответов

201, 400, 403 – см. «[Статусы ответов](#)».

### 4.4.4 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с информацией об идентификаторе, который получил созданный шлейф:

```
{
  "Id": string
}
```

Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в теле ответа – см. «[Код 400: описание ошибки](#)»

### 4.4.5 Пример использования

```
curl --request POST \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url 'http://10.7.22.128:9002/api/Zones?siteId=b8144107-31d1-4800-b83d-764f015a54a5&'
  'userName=crm-Ivanova-A-A' \
  --data '{"ZoneNumber": 3,"ZoneDesc": "Описание шлейфа", "ZoneEquip": "Оборудование шлейфа"}'
```

Status : 201

```
{
  "Id": "524bf1a5-76ce-43a7-9ed5-56291750933f"
}
```

## 4.5 Изменить шлейф (PUT /api/Zones)

Метод предназначен для изменения полей шлейфа.

URL : /api/Zones

Метод : PUT

### 4.5.1 Параметры

**id**

Обязательный параметр.

Идентификатор шлейфа, который нужно изменить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 4.5.2 Тело запроса

В теле запроса должен быть передан элемент JSON с [полями шлейфа](#), которые нужно изменить. Если поле необходимо оставить без изменения, то оно не должно быть указано в элементе JSON.

Идентификатор объекта передается параметром в заголовке запроса, поэтому поле `id` в элементе JSON может быть не указано или будет проигнорировано.

### 4.5.3 Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### 4.5.4 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с [полями шлейфа](#) – с учетом изменений, которые произошли в результате выполнения запроса.

Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в теле ответа – см. «[Код 400: описание ошибки](#)»

### 4.5.5 Пример использования

```
curl --request PUT \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/Zones?id=524bf1a5-76ce-43a7-9ed5-56291750933f&' \  
    'userName=crm-Ivanova-A-A' \  
  --data '{"ZoneDesc": "Новое описание шлейфа"}'
```

## 4.6 Удалить шлейф (DELETE /api/Zones)

Метод предназначен для удаления шлейфа объекта.

**URL :** /api/Zones

**Метод :** DELETE

### 4.6.1 Параметры

**id**

Обязательный параметр.

Идентификатор шлейфа, который нужно удалить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 4.6.2 Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### 4.6.3 Возвращаемые данные

При успешном выполнении метод не возвращает данных. Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в ответе – см. [«Код 400: описание ошибки»](#)

### 4.6.4 Пример использования

```
curl --request DELETE \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/Zones?id=524bf1a5-76ce-43a7-9ed5-56291750933f&' \  
    'userName=crm-Ivanova-A-A'
```

## 5 Ответственные лица

Методы для управления ответственными лицами объекта: получение списка ответственных лиц, создание, изменение или удаление ответственных лиц объекта.

### 5.1 Поля ответственного лица

Элемент JSON, содержащий все поля ответственного лица:

```
{
  "Id": string,
  "OrderNumber": number,
  "UserNumber": number,
  "ObjCustName": string,
  "ObjCustTitle": string,
  "ObjCustPhone1": string,
  "ObjCustPhone2": string,
  "ObjCustPhone3": string,
  "ObjCustPhone4": string,
  "ObjCustPhone5": string,
  "ObjCustAddress": string,
  "IsVisibleInCabinet": boolean,
  "ReclosingRequest": boolean,
  "ReclosingFailure": boolean,
  "PINCode": string
}
```

Название поля	Тип	Поле в карточке объекта; примечание
Id	string	Идентификатор ответственного лица
OrderNumber	number	Порядковый номер ответственного в список (уникальный на объекте, может быть не задан)
UserNumber	number	Номер ответственного (номер пользователя на контрольной панели, натуральное число, уникальный на объекте, может быть не задан, нельзя очистить для пользователя MyAlarm)
ObjCustName	string	ФИО
ObjCustTitle	string	Должность
ObjCustPhone1	string	Мобильный телефон (уникальный на объекте, нельзя изменить для пользователя MyAlarm)
ObjCustPhone2	string	Телефон 2
ObjCustPhone3	string	Телефон 3
ObjCustPhone4	string	Телефон 4
ObjCustPhone5	string	Телефон 5
ObjCustAddress	string	Адрес
IsVisibleInCabinet	boolean	Отображать в личном кабинете (нельзя отключить для пользователя MyAlarm)
ReclosingRequest	boolean	Отправлять SMS о необходимости перезакрытия

ReclosingFailure	boolean	Отправлять SMS об отказе от перезакрытия
PINCode	string	PIN для Call-центра

## 5.2 Получить список ответственных лиц объекта (GET /api/Customers)

Метод предназначен для получения списка ответственных лиц объекта.

**URL :** /api/Customers

**Метод :** GET

### 5.2.1 Параметры

#### siteId

Обязательный параметр.

Идентификатор объекта, список ответственных лиц которого нужно получить. Соответствует полю Id элемента JSON с [полями объекта](#).

#### userName

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 5.2.2 Возможные статусы ответов

200, 403 — см. «[Статусы ответов](#)».

### 5.2.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с [полями ответственных лиц](#).

### 5.2.4 Пример использования

```
curl --request GET \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url 'http://10.7.22.128:9002/api/Customers?siteId=b8144107-31d1-4800-b83d-764f015a54a5&'
      'userName=crm-Ivanova-A-A'
```

**Status:** 200

```
[
  {
    "Id": "81d1b742-ea10-4c56-92fb-092c608d41ec",
    "OrderNumber": 1,
    "UserNumber": null,
    "ObjCustName": null,
```

```

        "ObjCustTitle":null,
        "ObjCustPhone1":null,
        "ObjCustPhone2":null,
        "ObjCustPhone3":null,
        "ObjCustPhone4":null,
        "ObjCustPhone5":null,
        "ObjCustAddress":null,
        "IsVisibleInCabinet":null,
        "ReclosingRequest":null,
        "ReclosingFailure":null,
        "PINCode":null
    },
    {
        "Id":"c29632f8-01be-4e6f-83b7-34ac831eff25",
        "OrderNumber":2,
        "UserNumber":null,
        "ObjCustName":"Иванов Иван Иванович",
        "ObjCustTitle":"Главный производственный инженер",
        "ObjCustPhone1":"88005553535",
        "ObjCustPhone2":null,
        "ObjCustPhone3":null,
        "ObjCustPhone4":null,
        "ObjCustPhone5":null,
        "ObjCustAddress":null,
        "IsVisibleInCabinet":null,
        "ReclosingRequest":null,
        "ReclosingFailure":null,
        "PINCode":null
    }
]

```

## 5.3 Получить ответственное лицо по идентификатору (GET /api/Customers)

Метод предназначен для получения информации о конкретном ответственном лице. Для поиска ответственного лица должен быть использован его идентификатор.

**URL** : /api/Customers

**Метод** : GET

### 5.3.1 Параметры

**id**

Обязательный параметр.

Идентификатор ответственного лица, информацию о котором нужно получить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 5.3.2 Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### 5.3.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с [полями ответственного лица](#).

### 5.3.4 Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/Customers?id=c29632f8-01be-4e6f-83b7-34ac831eff25&' \  
      'userName=crm-Ivanova-A-A'
```

Status: 200

```
{  
  "Id": "c29632f8-01be-4e6f-83b7-34ac831eff25",  
  "OrderNumber": 2,  
  "UserNumber": null,  
  "ObjCustName": "Иванов Иван Иванович",  
  "ObjCustTitle": "Главный производственный инженер",  
  "ObjCustPhone1": "88005553535",  
  "ObjCustPhone2": null,  
  "ObjCustPhone3": null,  
  "ObjCustPhone4": null,  
  "ObjCustPhone5": null,  
  "ObjCustAddress": null,  
  "IsVisibleInCabinet": null,  
  "ReclosingRequest": null,  
  "ReclosingFailure": null,  
  "PINCode": null  
}
```

## 5.4 Создать ответственное лицо (POST /api/Customers)

Метод предназначен для создания нового ответственного лица для объекта.

URL : /api/Customers

Метод : POST

### 5.4.1 Параметры

siteId

Обязательный параметр.

Идентификатор объекта, для которого нужно создать объект. Соответствует полю Id элемента JSON с [полями объекта](#).

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 5.4.2 Тело запроса

В теле запроса должен быть передан элемент JSON с [полями ответственного лица](#), который нужно создать. При создании ответственного лица не обязательно указывать все поля, в случае отсутствия будет использовано значение по умолчанию. Если указан идентификатор ответственного лица, то он будет проигнорирован.

Если номер ответственного лица не указан, то новому ответственному лицу будет присвоен свободный номер по формуле  $\max(\text{UserNumber}) + 1$ .

Если номер ответственного лица указан и ответственного лица с таким номером уже есть в списке ответственных лиц объекта, то метод вернет ошибку.

### 5.4.3 Возможные статусы ответов

201, 400, 403 – см. [«Статусы ответов»](#).

### 5.4.4 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с информацией об идентификаторе, который получило созданное ответственное лицо:

```
{
  "Id": string
}
```

Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в теле ответа – см. [«Код 400: описание ошибки»](#)

### 5.4.5 Пример использования

```
curl --request POST \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url 'http://10.7.22.128:9002/api/Customers?siteId=c29632f8-01be-4e6f-83b7-34ac831eff25&'
  'userName=crm-Ivanova-A-A' \
  --data '{ "ObjCustName" : "Иванов Иван Иванович", '
  "ObjCustTitle": "Главный производственный инженер", '
  "ObjCustPhone1": "88005553535" }'
```

Status : 201

```
{
  "Id": "524bf1a5-76ce-43a7-9ed5-56291750933f"
}
```



## 5.5 Изменить ответственное лицо (PUT /api/Customers)

Метод предназначен для изменения полей ответственного лица.

**URL** : /api/Customers

**Метод** : PUT

### 5.5.1 Параметры

**id**

Обязательный параметр.

Идентификатор ответственного лица, параметры которого нужно изменить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 5.5.2 Тело запроса

В теле запроса должен быть передан элемент JSON с [полями ответственного лица](#), которые нужно изменить. Если поле необходимо оставить без изменения, то оно не должно быть указано в элементе JSON.

Идентификатор объекта передается параметром в заголовке запроса, поэтому поле **id** в элементе JSON может быть не указано или будет проигнорировано.

### 5.5.3 Возможные статусы ответов

200, 400, 403 – см. [«Статусы ответов»](#).

### 5.5.4 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с [полями ответственного лица](#) – с учетом изменений, которые произошли в результате выполнения запроса.

Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в теле ответа – см. [«Код 400: описание ошибки»](#)

### 5.5.5 Пример использования

```
curl --request PUT \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.9002/api/Customers?id=c29632f8-01be-4e6f-83b7-34ac831eff25&' \  
    'userName=crm-Ivanova-A-A' \  
  --data '{ "ObjCustName" : "Иванов Иван Иванович",' \  
    ' "ObjCustTitle": "Инженер технолог",' \  
    ' "ObjCustPhone1": "88005553232" }'
```

## 5.6 Удалить ответственное лицо (DELETE /api/Customers)

Метод предназначен для удаления ответственного лица объекта.

Примечание: Нельзя удалить пользователя, которому выдан доступ в MyAlarm.

**URL** : /api/Customers

**Метод** : DELETE

### 5.6.1 Параметры

**id**

Обязательный параметр.

Идентификатор ответственного лица, которое нужно удалить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 5.6.2 Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### 5.6.3 Возвращаемые данные

При успешном выполнении метод не возвращает данных. Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в ответе – см. «[Код 400: описание ошибки](#)»

### 5.6.4 Пример использования

```
curl --request DELETE \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/Customers?id=524bf1a5-76ce-43a7-9ed5-56291750933f&' \  
      'userName=crm-Ivanova-A-A'
```

## 6 Дополнительные характеристики

Методы для управления дополнительными характеристиками объекта: получение списка дополнительных характеристик, создание, изменение или удаление дополнительных характеристик объекта.

### 6.1 Поля дополнительной характеристики

Элемент JSON, содержащий все поля дополнительной характеристики:

```
{
  "Id": string,
  "ExtFieldName": string,
  "ExtFieldValue": string,
  "OrderNumber": number
}
```

Название поля	Тип	Поле в карточке объекта; примечание
Id	string	Идентификатор характеристики
ExtFieldName	string	Название характеристики
ExtFieldValue	string	Значение характеристики
OrderNumber	number	Порядковый номер дополнительной характеристики

### 6.2 Получить список дополнительных характеристик объекта (GET /api/ExtFields)

Метод предназначен для получения списка дополнительных характеристик объекта.

**URL :** /api/ExtFields

**Метод :** GET

#### 6.2.1 Параметры

**siteId**

Обязательный параметр.

Идентификатор объекта, список дополнительных характеристик которого нужно получить. Соответствует полю Id элемента JSON с [полями объекта](#).

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

#### 6.2.2 Возможные статусы ответов

200, 403 – см. [«Статусы ответов»](#).

### 6.2.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с [полями дополнительной характеристики](#).

### 6.2.4 Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/ExtFields?siteId=b8144107-31d1-4800-b83d-764f015a54a5&' \  
    'userName=crm-Ivanova-A-A'
```

Status: 200

```
[  
  {  
    "Id": "c2022370-eaf2-4593-a401-175314c0db96",  
    "ExtFieldName": "ExtraUserPhone",  
    "ExtFieldValue": "88005553232",  
    "OrderNumber": 3  
  },  
  {  
    "Id": "3cdf90f7-9687-44c8-b202-5d3dc9ddf4f3",  
    "ExtFieldName": "ExtraUserCarNumber",  
    "ExtFieldValue": "5dyw382",  
    "OrderNumber": 4  
  }  
]
```

## 6.3 Получить дополнительную характеристику по идентификатору (GET /api/ExtFields)

Метод предназначен для получения информации о конкретной дополнительной характеристике. Для поиска дополнительной характеристики должен быть использован её идентификатор.

URL : /api/ExtFields

Метод : GET

### 6.3.1 Параметры

**id**

Обязательный параметр.

Идентификатор дополнительной характеристики, информацию о которой нужно получить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 6.3.2 Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### 6.3.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с [полями дополнительной характеристики](#).

### 6.3.4 Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/ExtFields?id=3cdf90f7-9687-44c8-b202-5d3dc9ddf4f3&' \  
        'userName=crm-Ivanova-A-A'
```

Status: 200

```
{  
  "Id": "3cdf90f7-9687-44c8-b202-5d3dc9ddf4f3",  
  "ExtFieldName": "ExtraUserCarNumber",  
  "ExtFieldValue": "5dyw382",  
  "OrderNumber": 4  
}
```

## 6.4 Создать дополнительную характеристику (POST /api/ExtFields)

Метод предназначен для создания новой дополнительной характеристики для объекта.

URL : /api/ExtFields

Метод : POST

### 6.4.1 Параметры

**siteId**

Обязательный параметр.

Идентификатор объекта, для которого нужно создать дополнительную характеристику. Соответствует полю Id элемента JSON с [полями объекта](#).

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 6.4.2 Тело запроса

В теле запроса должен быть передан элемент JSON с [полями дополнительной характеристики](#), которую нужно создать. При создании дополнительной характеристики обязательно указать название и значение характеристики. Значения для всех остальных полей, можно не указывать: будет использовано значение по умолчанию.

Если указан идентификатор дополнительной характеристики, то он будет проигнорирован.

### 6.4.3 Возможные статусы ответов

201, 400, 403 – см. «[Статусы ответов](#)».

### 6.4.4 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с информацией об идентификаторе, который получила созданная дополнительная характеристика:

```
{
  "Id": string
}
```

Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в теле ответа – см. «[Код 400: описание ошибки](#)»

### 6.4.5 Пример использования

```
curl --request POST \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url 'http://10.7.22.128:9002/api/ExtFields?siteId=b8144107-31d1-4800-b83d-764f015a54a5&'
  'userName=crm-Ivanova-A-A' \
  --data '{ "ExtFieldName": "ExtraUserPhone", "ExtFieldValue": "+79012345678" }'
```

Status : 201

```
{
  "Id": "524bf1a5-76ce-43a7-9ed5-56291750933f"
}
```

## 6.5 Изменить дополнительную характеристику (PUT /api/ExtFields)

Метод предназначен для изменения полей дополнительной характеристики.

URL : /api/ExtFields

Метод : PUT

### 6.5.1 Параметры

id

Обязательный параметр.

Идентификатор дополнительной характеристики, которую нужно изменить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 6.5.2 Тело запроса

В теле запроса должен быть передан элемент JSON с [полями дополнительной характеристики](#), которые нужно изменить. Если поле необходимо оставить без изменения, то оно не должно быть указано в элементе JSON. Название и значение характеристики обязательные поля.

Идентификатор дополнительной характеристики передается параметром в заголовке запроса, поэтому поле `id` в элементе JSON может быть не указано или будет проигнорировано.

### 6.5.3 Возможные статусы ответов

200, 400, 403 – см. [«Статусы ответов»](#).

### 6.5.4 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с [полями дополнительной характеристики](#) – с учетом изменений, которые произошли в результате выполнения запроса.

Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в теле ответа – см. [«Код 400: описание ошибки»](#)

### 6.5.5 Пример использования

```
curl --request PUT \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/ExtFields?id=524bfa5-76ce-43a7-9ed5-56291750933f&' \  
  --data '{ "ExtFieldName": "ExtraUserPhone", "ExtFieldValue": "88005553232" }'
```

## 6.6 Удалить дополнительную характеристику (DELETE /api/ExtFields)

Метод предназначен для удаления дополнительной характеристики объекта.

**URL :** /api/ExtFields

**Метод :** DELETE

### 6.6.1 Параметры

**id**

Обязательный параметр.

Идентификатор дополнительной характеристики, которую нужно удалить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 6.6.2 Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### 6.6.3 Возвращаемые данные

При успешном выполнении метод не возвращает данных. Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в ответе – см. «[Код 400: описание ошибки](#)»

### 6.6.4 Пример использования

```
curl --request DELETE \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/ExtFields?id=524bf1a5-76ce-43a7-9ed5-56291750933f&'  
      'userName=crm-Ivanova-A-A'
```



## 7 Проверка КТС

### 7.1 Начать проверку КТС (POST /api/CheckPanic)

Метод предназначен для запуска процедуры проверки КТС.

**URL :** /api/CheckPanic

**Метод :** POST

#### 7.1.1 Параметры

##### **siteId**

Обязательный параметр.

Идентификатор объекта, по которому нужно проверить КТС.

##### **checkInterval**

Необязательный параметр.

**Допустимый диапазон значений:** [30, 180]

Интервал в секундах, в течении которого будет продолжаться процедура проверки КТС.

Если для параметра не задано значение, то будет использовано значение параметра [DefaultCheckPanicInterval](#), указанное в настройках службы «C.Nord HTTP-API Service».

Если для параметра задано значение, выходящее за пределы допустимого диапазона значений, то метод вернет ошибку (см. ниже).

##### **stopOnEvent**

Необязательный параметр.

Возможные значения: **True** / **False**.

Если для параметра задано значение **True**, то после получения первого же события о нажатии тревожной кнопки процедура проверки КТС будет прекращена, отсчет интервала, заданного параметром **checkInterval** будет прекращен.

Если же для параметра задано значение **False**, то после получения первого события о нажатии тревожной кнопки, отсчет интервала, заданного параметром **checkInterval** будет продолжен и все события о нажатии тревожной кнопки, которые будут получены в течении этого интервала будут помечаться, как отключенные. Кроме того, повторный запуск процедуры проверки КТС для этого объекта будет возможен только после завершения отсчета этого интервала.

Если значение для параметра не задано, то будет использовано значение параметра [DefaultCheckPanicStopOnEvent](#), указанное в настройках службы «C.Nord HTTP-API Service».

##### **userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

#### 7.1.2 Возможные статусы ответов

200, 403 – см. «[Статусы ответов](#)».

### 7.1.3 Возвращаемый результат

Результат проверки - в теле ответа, тип – `application/json`

```
{
  "Status": 200,
  "Description": string,
  "CheckPanicId": string
}
```

#### Status

Результат выполнения запроса, всегда равен 200.

#### Description

Результат запуска процедуры проверки КТС.

Возможные значения:

- "has alarm" – по объекту есть тревога, проверка КТС запрещена;
- "already run" – по объекту уже выполняется проверка КТС;
- "success" – проверка КТС начата;
- "error" - при выполнении запроса произошла ошибка;
- "invalid checkInterval value" – для параметра `checkInterval` задано значение, выходящее за пределы допустимого диапазона.

#### CheckPanicId

Идентификатор запущенной процедуры проверки КТС. Значение идентификатора необходимо использовать в качестве аргумента при вызове метода [GET /api/CheckPanic](#).

### 7.1.4 Пример использования

```
curl --request POST \
--header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
--url 'http://10.0.0.2:9002/api/CheckPanic?'
      'siteId=ae59ad78-6341-44dc-adb8-4803b7f02412&'
      'userName=automatic_panic_checker'

{
  "Status": 200,
  "Description": "success",
  "CheckPanicId": "5b42d9f1-52f3-4144-9a4f-bd047c60e115"
}
```

## 7.2 Получить результат проверки КТС (GET /api/CheckPanic)

Метод предназначен для получения результата процедуры проверки КТС.

URL : `/api/CheckPanic`

Метод : GET

### 7.2.1 Параметры

#### **checkPanicId**

Обязательный параметр.

Идентификатор процедуры проверки, для которой нужно получить результат. Идентификатор возвращается в результате вызова метода [POST /api/CheckPanic](#).

#### **userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 7.2.2 Возможные статусы ответов

200, 403 – см. [«Статусы ответов»](#).

### 7.2.3 Возвращаемые результаты

Результат проверки - в теле ответа, тип – `application/json`

```
{
  "Status": 200,
  "Description": string
}
```

#### **Status**

Результат выполнения запроса, всегда равен 200.

#### **Description**

Результат выполнения процедуры проверки КТС.

Возможные значения:

- "not found" – проверка с указанным идентификатором не найдена;
- "in progress" – проверка с указанным идентификатором продолжается (не завершена): КТС не получена, тайм-аут не истек;
- "success" – проверка с указанным идентификатором успешно завершена;
- "success, interval continues" – проверка с указанным идентификатором успешно завершена, но продолжается отсчет интервала проверки;
- "time out" – проверка с указанным идентификатором завершена с ошибкой: истек интервал ожидания события КТС;
- "error" - при выполнении запроса произошла ошибка.

Максимальная продолжительность хранения информации о запущенной проверке КТС составляет 4 минуты (240 секунд) с момента начала проверки.

#### 7.2.4 Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.0.0.2:9002/api/CheckPanic?'  
    'checkPanicId=5b42d9f1-52f3-4144-9a4f-bd047c60e115&'  
    'userName=automatic_panic_checker'  
  
{  
  "Status": 200,  
  "Description": "in progress",  
}
```

## 8 События объекта

### 8.1 Поля события

Элемент JSON, содержащий поля события

```
{
  "RChannelName": string,
  "DateTime": string,
  "AccountNumber": number,
  "EventCode": string,
  "EventClassName": string,
  "EventClassType": string,
  "EventDesc": string,
  "PartNumber": number,
  "ZoneUser": number,
  "AlarmIndex": number,
  "SaveDateTime": string
}
```

Название поля	Тип	Описание
RChannelName	string	Мнемоническое название и номер канала связи
DateTime	string	Дата и время формирования события
AccountNumber	number	Номер объекта, по которому сформировано событие
EventCode	string	Код события
EventClassName	string	Название класса события
EventClassType	string	Псевдоним названия типа класса события (см. ниже)
EventDesc	string	Описание события
PartNumber	number	Номер раздела, по которому сформировано событие
ZoneUser	number	Номер зоны или пользователя
AlarmIndex	number	Индекс группы тревог, к которой относится событие
SaveDateTime	string	Дата и время сохранения события в базе данных «Центра охраны»

#### 8.1.1 Значения поля EventClassType

Псевдоним названия типа класса события.

Тип класса события, указывается в модуле «Настройка системы», на вкладке «Классы событий».

Соответствия значений псевдонима и названий типа класса события:

Псевдоним	Название типа класса события
alarm	Тревога
reset	Сброс

fault	Неисправность
restore	Восстановление
arm	Взятие под охрану
disarm	Снятие с охраны
bypass	Исключение
warning	Предупреждение
other	Другое
test	Тест

### 8.1.2 Формат полей с датой и временем

Для полей `DateTime` и `SaveDateTime` строковое значение даты и времени имеет формат “YYYY-MM-DDTHH:mm:ss.fff”. Часовой пояс даты и времени соответствует часовому поясу компьютера, на котором запущен модуль «Менеджер событий».

## 8.2 Получить список событий объекта (GET /api/SiteEvents)

Метод, предназначенный для получения событий объекта. Метод возвращает не только события переданные охранным прибором, установленном на объекте, но и события сформированные программным обеспечением «Центр охраны».

**URL :** /api/SiteEvents

**Метод :** GET

### 8.2.1 Параметры

**id**

Обязательный параметр.

Идентификатор или номер объекта, события которого должен вернуть метод.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 8.2.2 Тело запроса

В теле запроса должен быть передан элемент JSON со значениями параметров, уточняющими интервал и типы классов для событий, которые необходимо вернуть. Если ни для какого параметра значение не указано, то есть в теле передаются пустые скобки {}, то будет использовано значение по умолчанию.

```
{
  "startDate": string,
  "stopDate": string,
  "ectAlarm": boolean,
  "ectReset": boolean,
```

```

    "ectFault": boolean,
    "ectRestore": boolean,
    "ectArm": boolean,
    "ectDisarm": boolean,
    "ectBypass": boolean,
    "ectWarning": boolean,
    "ectOther": boolean,
    "ectTest": boolean,
    "duplicateEvent": boolean
}

```

Параметр	Тип	Значение по умолчанию	Примечание
startDate	string	'Now'	Начало временного интервала, за который нужно вернуть события
stopDate	string	'Now - 32 days'	Конец временного интервала, за который нужно вернуть события
ectAlarm	boolean	true	Вернуть события с типом класса «Тревога»
ectReset	boolean	true	Вернуть события с типом класса «Сброс»
ectFault	boolean	true	Вернуть события с типом класса «Неисправность»
ectRestore	boolean	true	Вернуть события с типом класса «Восстановление»
ectArm	boolean	true	Вернуть события с типом класса «Взятие под охрану»
ectDisarm	boolean	true	Вернуть события с типом класса «Снятие с охраны»
ectBypass	boolean	true	Вернуть события с типом класса «Исключение»
ectWarning	boolean	true	Вернуть события с типом класса «Предупреждение»
ectOther	boolean	true	Вернуть события с типом класса «Другое»
ectTest	boolean	false	Вернуть события с типом класса «Тест»
duplicateEvent	boolean	false	Вернуть дублирующие события

### Значения по умолчанию

Если в теле запроса значение для параметра не передается, то будет использовано значение по умолчанию.

### Формат полей с датой и временем

Возможные строковые значения даты и времени для полей **startDate** и **stopDate**:

- YYYY-MM-DD
- YYYY-MM-DDTHH:mm:ss
- YYYY-MM-DDTHH:mm:ss.fff

Часовой пояс даты и времени соответствует часовому поясу компьютера, на котором запущен модуль «Менеджер событий».

### 8.2.3 Возможные статусы ответов

200, 400, 403 – см. «Статусы ответов».

### 8.2.4 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с полями события.

### 8.2.5 Пример использования

Пример выполнения запроса, в котором указано только значение для идентификатора объекта

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/SiteEvents?id=94df3af9-36c1-423b-aa88-fb505bda3fa4'  
  --data '{  
    }'
```

Status: 200

```
[  
  {  
    "RChannelName": "Gprs6",  
    "DateTime": "2019-02-17T12:08:50.24",  
    "AccountNumber": 265,  
    "EventCode": "E110",  
    "EventClassName": "Пожар",  
    "EventClassType": "alarm",  
    "EventDesc": "Запасной выход",  
    "PartNumber": 1,  
    "ZoneUser": 1,  
    "AlarmIndex": 2019021711525158,  
    "SaveDateTime": "2019-02-17T12:08:50.26"  
  },  
  {  
    "RChannelName": "T3",  
    "DateTime": "2019-02-17T12:22:38.12",  
    "AccountNumber": 265,  
    "EventCode": "R140",  
    "EventClassName": "Сброс",  
    "EventClassType": "reset",  
    "EventDesc": "Запасной выход",  
    "PartNumber": 1,  
    "ZoneUser": 1,  
    "AlarmIndex": 0,  
    "SaveDateTime": "2019-02-17T12:22:38.143"  
  },  
  {  
    "RChannelName": "Sms1",  
    "DateTime": "2019-02-17T12:29:11.75",  
    "AccountNumber": 265,  
    "EventCode": "E624",  
    "EventClassName": "Неисправность",
```



```

        "EventClassType": "fault",
        "EventDesc": "Буфер событий переполнен, есть потерянные события. Главный вход",
        "PartNumber": 1,
        "ZoneUser": 2,
        "AlarmIndex": 0,
        "SaveDateTime": "2019-02-17T12:29:11.767"
    }
]

```

### Пример выполнения запроса, в котором указаны типы классов событий

```

curl --request GET \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url 'http://10.7.22.128:9002/api/SiteEvents?id=94df3af9-36c1-423b-aa88-fb505bda3fa4'
  --data '{
    "startDate": "2019-02-01",
    "stopDate": "2019-02-02",
    "ectReset": false,
    "ectFault": false,
    "ectRestore": false,
    "ectArm": false,
    "ectDisarm": false,
    "ectBypass": false,
    "ectWarning": false,
    "ectOther": false
  },'

```

Status: 200

```

{
  "RChannelName": "Gprs6",
  "DateTime": "2019-02-17T12:08:50.24",
  "AccountNumber": 265,
  "EventCode": "E110",
  "EventClassName": "Пожар",
  "EventClassType": "alarm",
  "EventDesc": "Запасной выход",
  "PartNumber": 1,
  "ZoneUser": 1,
  "AlarmIndex": 2019021711525158,
  "SaveDateTime": "2019-02-17T12:08:50.26"
}

```

## 9 Расписание охраны объекта

Методы для управления расписанием охраны объекта: Получение, создание, изменение или удаление расписания объекта.

### 9.1 Поля расписания охраны объекта

#### 9.1.1 Расписание охраны объекта

```
{
  "ArmSchedule_EarlyArm" : boolean,
  "ArmSchedule_ControlArm" : boolean,
  "ArmSchedule_LaterArm" : boolean,
  "ArmSchedule_EarlyDisarm" : boolean,
  "ArmSchedule_ControlDisarm" : boolean,
  "ArmSchedule_LaterDisarm" : boolean,
  "ArmSchedule_Deviation" : number,
  "Intervals" : Interval[]
}
```

Название поля	Тип	По умолчанию	Поле в карточке объекта; примечание
ArmSchedule_EarlyArm	boolean	false	Контролировать ранне взятие под охрану объекта
ArmSchedule_ControlArm	boolean	false	Контролировать отсутствие взятия под охрану объекта
ArmSchedule_LaterArm	boolean	false	Контролировать позднее взятие под охрану объекта
ArmSchedule_EarlyDisarm	boolean	false	Контролировать раннее снятие с охраны объекта
ArmSchedule_ControlDisarm	boolean	false	Контролировать отсутствие снятия с охраны объекта
ArmSchedule_LaterDisarm	boolean	false	Контролировать позднее снятие с охраны объекта
ArmSchedule_Deviation	number	0	Отклонение расписания объекта
Intervals	Interval[]	Отсутствует	Интервалы расписания объекта

#### 9.1.2 Интервал расписания объекта (Interval)

```
{
  "DayNumber" : number,
  "StartDT" : timespan,
  "StopDT" : timespan
}
```

Название поля	Тип	Поле в карточке объекта; примечание
DayNumber	number	День недели (от 1 до 7)

StartDT	timespan	Начальное время
StopDT	timespan	Конечное время

## 9.2 Получить расписание охраны объекта (GET /api/SiteSchedule)

Метод предназначен для получения расписания охраны объекта.

**URL :** /api/SiteSchedule

**Метод :** GET

### 9.2.1 Параметры

**siteId**

Обязательный параметр.

Идентификатор объекта, расписание охраны объекта, которого нужно получить. Соответствует полю Id элемента JSON с [полями объекта](#).

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 9.2.2 Возможные статусы ответов

200, 403 – см. «[Статусы ответов](#)».

### 9.2.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с [полями расписания охраны объекта](#).

### 9.2.4 Пример использования

```
curl --request GET \
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
  --url 'http://10.7.22.128:9002/api/SiteSchedule?siteId=082232F8-8B06-4F7A-ABCA-AFDCC5593283&'
  'userName=crm-Ivanova-A-A'
```

**Status:** 200

```
{
  "ArmSchedule_EarlyArm" : false,
  "ArmSchedule_ControlArm" : false,
  "ArmSchedule_LaterArm" : false,
  "ArmSchedule_EarlyDisarm" : false,
  "ArmSchedule_ControlDisarm" : false,
  "ArmSchedule_LaterDisarm" : true,
```

```

"ArmSchedule_Deviation" : 15,
"Intervals" :
[
  {
    "DayNumber" : 2,
    "StartDT" : "01:00",
    "StopDT" : "01:30"
  },
  {
    "DayNumber" : 1,
    "StartDT" : "01:00",
    "StopDT" : "01:30"
  },
  {
    "DayNumber" : 1,
    "StartDT" : "02:00",
    "StopDT" : "23:59:59"
  }
]
}

```

## 9.3 Создать/изменить расписание охраны объекта (POST /api/SiteSchedule)

Метод предназначен для создания/изменения расписания охраны объекта.

**URL** : /api/SiteSchedule

**Метод** : POST

### 9.3.1 Параметры

#### siteId

Обязательный параметр.

Идентификатор объекта, расписание охраны объекта, которого нужно получить. Соответствует полю Id элемента JSON с [полями объекта](#).

#### userName

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 9.3.2 Тело запроса

В теле запроса должен быть передан элемент JSON с [полями расписания охраны объекта](#), который нужно создать/изменить.

Если расписание охраны объекта уже создано и отличается то оно будет перезаписано.

Если интервалы расписания охраны объекта пересекаются, то метод вернёт ошибку.

Если в json не передать данных, то расписание примет значение по умолчанию.

Для полей StartDT, StopDT допустим следующий формат строковых значений:

- "HH:mm";.
- "HH:mm:ss";.

где mm - минуты должны быть кратными 30, а ss - секунды всегда 00, за исключением последнего интервала за день, который может закончиться в 23:59:59

Поле ArmSchedule\_Deviation может иметь значения 0, 5, 10, 15, 20, 25, 30.

### 9.3.3 Возможные статусы ответов

201, 400, 403 – см. «[Статусы ответов](#)».

### 9.3.4 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа элемент JSON с [полями расписания охраны объекта](#) изменённого объекта.

Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в теле ответа – см. «[Код 400: описание ошибки](#)»

### 9.3.5 Пример использования

```
curl --request POST \
--header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \
--url 'http://10.7.22.128:9002/api/SiteSchedule?siteId=082232F8-8B06-4F7A-ABCA-AFDCC5593283'
--data '{
  "ArmSchedule_EarlyArm" : false,
  "ArmSchedule_ControlArm" : false,
  "ArmSchedule_LaterArm" : false,
  "ArmSchedule_EarlyDisarm" : false,
  "ArmSchedule_ControlDisarm" : false,
  "ArmSchedule_LaterDisarm" : true,
  "ArmSchedule_Deviation" : 15,
  "Intervals" : [
    {
      "DayNumber" : 2,
      "StartDT" : "01:00",
      "StopDT" : "01:30"
    },
    {
      "DayNumber" : 1,
      "StartDT" : "01:00",
      "StopDT" : "01:30"
    },
    {
      "DayNumber" : 1,
      "StartDT" : "02:00",
      "StopDT" : "23:59:59"
    }
  ]
}
```

Status : 200

```

{
  "ArmSchedule_EarlyArm" : false,
  "ArmSchedule_ControlArm" : false,
  "ArmSchedule_LaterArm" : false,
  "ArmSchedule_EarlyDisarm" : false,
  "ArmSchedule_ControlDisarm" : false,
  "ArmSchedule_LaterDisarm" : true,
  "ArmSchedule_Deviation" : 15,
  "Intervals" :
  [
    {
      "DayNumber" : 2,
      "StartDT" : "01:00",
      "StopDT" : "01:30"
    },
    {
      "DayNumber" : 1,
      "StartDT" : "01:00",
      "StopDT" : "01:30"
    },
    {
      "DayNumber" : 1,
      "StartDT" : "02:00",
      "StopDT" : "23:59:59"
    }
  ]
}

```

## 9.4 Удалить расписание охраны объекта (DELETE /api/SiteSchedule)

Метод предназначен для удаления расписания охраны объекта.

**URL** : /api/SiteSchedule

**Метод** : DELETE

### 9.4.1 Параметры

**siteId**

Обязательный параметр.

Идентификатор объекта, расписание охраны объекта которого нужно удалить.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 9.4.2 Возможные статусы ответов

200, 400, 403 — см. «[Статусы ответов](#)».

### 9.4.3 Возвращаемые данные

При успешном выполнении метод не возвращает данных. Если при выполнении запроса возникла ошибка, то запрос вернет код 400 и описание возникшей ошибки в ответе – см. [«Код 400: описание ошибки»](#)

### 9.4.4 Пример использования

```
curl --request DELETE \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/SiteSchedule?siteId=082232F8-8B06-4F7A-ABCA-AFDCC5593283&' \  
    'userName=crm-Ivanova-A-A'
```

## 10 Управление доступом в MyAlarm

Методы для управления доступом в MyAlarm: Разрешение/запрет доступа или КТС, получение информации по конкретному пользователю и по объекту. Возможность подписаться/отписаться на класс событий или действие.

### 10.1 Структуры данных управления доступом в MyAlarm

#### 10.1.1 Пользователь MyAlarm

```
{  
  "CustomerID" : Guid,  
  "MobilePhone" : string,  
  "MyAlarmPhone" : string,  
  "Role" : string,  
  "IsPanic" : bool  
}
```

Название поля	Тип	Поле в карточке объекта; примечание
CustomerID	Guid	Идентификатор пользователя
MobilePhone	string	Телефон ответственного
MyAlarmPhone	string	Телефон пользователя MyAlarm
Role	string	Роль пользователя
IsPanic	bool	Разрешён или запрещён КТС

#### 10.1.2 Объект пользователя MyAlarm

```
{  
  "ObjectGUID" : Guid,  
  "CustomerID" : Guid,  
  "Role" : string,  
  "IsPanic" : bool  
}
```

Название поля	Тип	Поле в карточке объекта; примечание
ObjectGUID	Guid	Идентификатор объекта
CustomerID	Guid	Идентификатор пользователя
Role	string	Роль пользователя
IsPanic	bool	Разрешён или запрещён КТС



## 10.2 Структуры данных классов событий, действий и отмен

### 10.2.1 Класс события с полями

```
{  
    "EventClassID" : int,  
    "Type" : string,  
    "Name" : string  
}
```

Название поля	Тип	Поле в карточке объекта; примечание
EventClassID	int	Идентификатор класса события
Type	string	Тип класса событий (псевдоним)
Name	string	Название класса событий

### 10.2.2 Действие или отмена с полями

```
{  
    "UserActionID" : int,  
    "Type" : string,  
    "Name" : string  
}
```

Название поля	Тип	Поле в карточке объекта; примечание
UserActionID	int	Идентификатор действия и отмены
Type	string	Тип действия и отмены (псевдоним)
Name	string	Название действия и отмены

## 10.3 Основные методы http-api

### 10.3.1 Получить список пользователей MyAlarm объекта (GET /api/MyAlarm)

Метод предназначен для получения списка пользователей MyAlarm объекта.

**URL :** /api/MyAlarm

**Метод :** GET

#### Параметры

**siteId**

Обязательный параметр.

Идентификатор объекта, список пользователями MyAlarm которого нужно получить. Соответствует полю Id элемента JSON с [полями объекта](#).

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с [полями пользователя MyAlarm](#).

### Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/MyAlarm?siteId=fd2cdaae-585e-44a3-804e-a24537f6de7a'
```

Status: 200

```
[  
  {  
    "CustomerID": "548de89a-8b93-4319-997d-8902d8fb2a8e",  
    "MobilePhone": "89001122345",  
    "MyAlarmPhone": "+79001122345",  
    "Role": "admin",  
    "IsPanic": false  
  }  
]
```

## 10.3.2 Получить список объектов пользователя MyAlarm (GET /api/MyAlarm/UserObject

Метод предназначен для получения списка объектов пользователя MyAlarm.

URL : /api/MyAlarm/UserObjects

Метод : GET

### Параметры

**phone**

Обязательный параметр.

Телефон пользователя MyAlarm.

Примечание знак + необходимо заменить на %2B.

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### Возможные статусы ответов

200, 400, 403 — см. «[Статусы ответов](#)».

### Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с [полями объекта пользователя MyAlarm](#).

### Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/MyAlarm/UserObjects?phone=%2B79001122345'
```

Status: 200

```
[  
  {  
    "ObjectGUID": "fd2cdaae-585e-44a3-804e-a24537f6de7a",  
    "CustomerID": "548de89a-8b93-4319-997d-8902d8fb2a8e",  
    "Role": 1,  
    "IsPanic": false  
  }  
]
```

### 10.3.3 Предоставить/забрать доступ пользователю к MyAlarm (PUT /api/MyAlarm)

Метод предназначен для изменения доступа пользователю в MyAlarm

Примечание: при предоставлении доступа первому пользователю, объект будет подписан на все классы событий, действия и отмены.

Примечание: при снятии доступа у последнего пользователя, объект будет отписан от всех классов событий, действий и отмен.

Для смены роли пользователя ему сперва следует задать роль 'unlink' и только после этого можно заново задать уровень доступа.

Для задания роли пользователю, у него должен быть заполнен мобильный номер в стандарте E.164

URL : /api/MyAlarm

Метод : PUT

#### Параметры

##### custId

Обязательный параметр.

Идентификатор пользователя

##### role

Обязательный параметр.

Роль пользователя, допустимые значения: "unlink", "user", "admin"

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

#### **Возможные статусы ответов**

200, 400, 403 – см. «[Статусы ответов](#)».

#### **Пример использования**

```
curl --request PUT \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/MyAlarm?custId=548DE89A-8B93-4319-997D-8902D8FB2A8E&role=admin'
```

Status: 200

### **10.3.4 Модифицировать право на использование виртуальной КТС (PUT /api/MyAlarm)**

Метод предназначен для изменения права на пользование виртуальной КТС пользователю MyAlarm

URL : /api/MyAlarm

Метод : PUT

#### **Параметры**

**custId**

Обязательный параметр.

Идентификатор пользователя

**isPanic**

Обязательный параметр.

true разрешить использование, false - запретить

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

#### **Возможные статусы ответов**

200, 400, 403 – см. «[Статусы ответов](#)».

#### **Пример использования**

```
curl --request PUT \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/MyAlarm?custId=548DE89A-8B93-4319-997D-8902D8FB2A8E&isPanic=true'
```

Status: 200

## 10.4 Управление подписками

### 10.4.1 Получить список подписок объекта на классы событий (GET /api/MyAlarm/EventClass)

Метод предназначен для получения списка подписок объекта на классы событий

**URL :** /api/MyAlarm/EventClass

**Метод :** GET

#### Параметры

##### siteId

Обязательный параметр.

Идентификатор объекта, список пользователями MyAlarm которого нужно получить. Соответствует полю Id элемента JSON с [полями объекта](#).

##### userName

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

#### Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

#### Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с [полями класса событий](#).

#### Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/MyAlarm/EventClass?siteId=fd2cdaae-585e-44a3-804e-a24537f6de7a'
```

**Status:** 200

```
[  
  {  
    "EventClassID":3,  
    "Type":"alarm",  
    "Name":"Тревога"  
  },{  
    "EventClassID":4,  
    "Type":"alarm",  
    "Name":"Тихая тревога"  
  }  
]
```

## 10.4.2 Получить список подписок объекта на действия и отмены (GET /api/MyAlarm/UserAction)

Метод предназначен для получения списка подписок объекта на действия и отмены

**URL :** /api/MyAlarm/UserAction

**Метод :** GET

### Параметры

#### siteId

Обязательный параметр.

Идентификатор объекта, список пользователями MyAlarm которого нужно получить. Соответствует полю Id элемента JSON с [полями объекта](#).

#### userName

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### Возможные статусы ответов

200, 400, 403 – см. [«Статусы ответов»](#).

### Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с [полями действия или отмены](#).

### Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/MyAlarm/UserAction?siteId=fd2cdaae-585e-44a3-804e-a24537f6de7a'
```

**Status:** 200

```
[  
  {  
    "UserActionID":2,  
    "Type":"SendGuard",  
    "Name":"Вызов группы"  
  },{  
    "UserActionID":3,  
    "Type":"ArriveGuard",  
    "Name":"Прибытие группы"  
  }  
]
```

### 10.4.3 Модификация подписок объекта на классы событий (PUT /api/MyAlarm/EventClass)

Метод предназначен для изменения списка подписок объекта на классы событий

**URL :** /api/MyAlarm/EventClass

**Метод :** PUT

#### Параметры

##### siteId

Обязательный параметр.

Идентификатор объекта, список пользователями MyAlarm которого нужно получить. Соответствует полю Id элемента JSON с [полями объекта](#).

##### userName

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

#### Тело запроса

В теле запроса должен быть передан элемент JSON со списком идентификаторов классов событий, например: [1, 2, 3].

#### Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

#### Пример использования

```
curl --request PUT \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/MyAlarm/EventClass?siteId=fd2cdaae-585e-44a3-804e-a24537f6de7a' \  
  --data '[3, 4]'
```

**Status:** 200

### 10.4.4 Модификация подписок объекта на действия и отмены (PUT /api/MyAlarm/UserAction)

Метод предназначен для изменения списка подписок объекта на действия или отмены

**URL :** /api/MyAlarm/UserAction

**Метод :** PUT

#### Параметры

**siteId**

Обязательный параметр.

Идентификатор объекта, список пользователями MyAlarm которого нужно получить. Соответствует полю Id элемента JSON с [полями объекта](#).

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### Тело запроса

В теле запроса должен быть передан элемент JSON со списком идентификаторов действия или отмены, например: [1, 2, 3].

### Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### Пример использования

```
curl --request PUT \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/MyAlarm/UserAction?siteId=fd2cdaae-585e-44a3-804e-a24537f6de7a' \  
  --data '[3, 4]'
```

Status: 200

## 10.5 Списки классов событий, действий и отмен

### 10.5.1 Получить список классов событий ЦО (GET /api/EventClasses)

Метод предназначен для получения списка классов событий Центра Охраны.

URL : /api/EventClasses

Метод : GET

#### Параметры

**userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с [полями класса событий](#).



### Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/EventClasses'
```

Status: 200

```
[  
  {  
    "EventClassID":3,  
    "Type":"alarm",  
    "Name":"Тревога"  
  },{  
    "EventClassID":4,  
    "Type":"alarm",  
    "Name":"Тихая тревога"  
  }, ...  
]
```

### 10.5.2 Получить список действий и отмен ЦО (GET /api/UserActions)

Метод предназначен для получения списка действий и отмен Центра Охраны.

URL : /api/UserActions

Метод : GET

#### Параметры

userName

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

#### Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

#### Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с [полями действия](#) или [отмены](#).

### Пример использования

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/UserActions'
```

Status: 200

```
[
  {
    "UserActionID":3,
    "Type":"ArriveGuard",
    "Name":"Прибытие группы"
  },{
    "UserActionID":4,
    "Type":"CancelGuard",
    "Name":"Отмена вызова группы"
  }, ...
]
```

# 11 Тревоги

## 11.1 Поля тревоги

Элемент JSON, содержащий поля события тревоги

```
{
  "ObjectGUID": Guid,
  "RChannelName": string,
  "DateTime": DateTime,
  "AlarmIndex": long (Int64),
  "EventCode": string,
  "EventClassName": string,
  "EventClassType": string,
  "EventDesc": string,
  "PartNumber": int,
  "ZoneUser": int,
  "SaveDateTime": DateTime
}
```

Название поля	Тип	Описание
ObjectGUID	Guid	Идентификатор объекта
RChannelName	string	Мнемоническое название и номер канала связи
DateTime	string	Дата и время формирования события
AlarmIndex	long (Int64)	Индекс группы тревог, к которой относится событие
EventCode	string	Код события
EventClassName	string	Класс события
EventClassType	string	Псевдоним названия типа класса события
EventDesc	string	Описание события
PartNumber	number	Номер раздела, по которому сформировано событие
ZoneUser	number	Номер зоны или пользователя
SaveDateTime	string	Дата и время сохранения события в базе данных «Центра охраны»

Значение поля EventClassType всегда будет “alarm”

### 11.1.1 Формат полей с датой и временем

Для полей DateTime и SaveDateTime строковое значение даты и времени имеет формат “YYYY-MM-DDTHH:mm:ss.fff”. Часовой пояс даты и времени соответствует часовому поясу компьютера, на котором запущен модуль «Менеджер событий».

## 11.2 Поля действия или отмены

```
{
  "DateTime": DateTime,
```

```

    "UserName": string,
    "ActionName": string,
    "ActionType": string,
    "GuardName": string,
    "Comment": string
}

```

Название поля	Тип	Описание
DateTime	DateTime	Дата и время
UserName	string	Имя пользователя
ActionName	string	Действие
ActionType	string	Тип действия
GuardName	string	Название охраны
Comment	string	Комментарий

## 11.3 Получить список тревог за период (GET /api/Alarms)

Метод, предназначенный для получения событий тревоги за период. Метод возвращает не только события переданные охранным прибором, установленном на объекте, но и события сформированные программным обеспечением «Центр охраны».

**URL** : /api/Alarms

**Метод** : GET

### 11.3.1 Параметры

#### **dateFrom**

Обязательный параметр

Начальная дата и время периода

#### **dateTo**

Необязательный параметр

Конечная дата и время периода

Примечание: Максимальное количество записей ограничено 1000, для получения всех записей нужно манипулировать начальной и конечной датой

#### **userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

#### **Формат полей с датой и временем**

Возможные строковые значения даты и времени для полей **dateFrom** и **dateTo**:

- yyyyMMddHHmmss

### 11.3.2 Возможные статусы ответов

200, 400, 403 – см. «Статусы ответов».

### 11.3.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с полями события.

### 11.3.4 Пример выполнения запроса

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/Alarms?dateFrom=20160102100013&dateTo=20190319150859'
```

Status: 200

```
[{  
  "ObjectGUID": "fd2cdaae-585e-44a3-804e-a24537f6de7a",  
  "RChannelName": "Sms3",  
  "DateTime": "2019-03-19T15:08:57.41",  
  "AlarmIndex": 2019031915085742,  
  "EventCode": "E151",  
  "EventClassName": "Тревога",  
  "EventClassType": "alarm",  
  "EventDesc": "Утечка газа.  ",  
  "PartNumber": 0,  
  "ZoneUser": 3,  
  "SaveDateTime": "2019-03-19T15:08:57.57"  
},  
{  
  "ObjectGUID": "fd2cdaae-585e-44a3-804e-a24537f6de7a",  
  "RChannelName": "Gprs4",  
  "DateTime": "2019-03-19T15:08:58.49",  
  "AlarmIndex": 2019031915085742,  
  "EventCode": "E136",  
  "EventClassName": "Тревога",  
  "EventClassType": "alarm",  
  "EventDesc": "Снаружи.  ",  
  "PartNumber": 0,  
  "ZoneUser": 2,  
  "SaveDateTime": "2019-03-19T15:08:58.9"  
}, ...]
```

## 11.4 Получить список действий или отмен по индексу тревоги (GET /api/Alarms)

Метод предназначен для получения списков действий или отмен по индексу тревоги

URL : /api/Alarms

Метод : GET

### 11.4.1 Параметры

#### **alarmIndex**

Обязательный параметр.

Индекс тревоги.

#### **userName**

Необязательный параметр.

Имя пользователя, от имени которого выполняется операция.

### 11.4.2 Возможные статусы ответов

200, 400, 403 – см. «[Статусы ответов](#)».

### 11.4.3 Возвращаемые данные

При успешном выполнении метод возвращает в теле ответа список элементов JSON с [полями события](#).

### 11.4.4 Пример выполнения запроса

```
curl --request GET \  
  --header 'apiKey: 41c66fd22dcf4742b65e9f5ea5ebde1c' \  
  --url 'http://10.7.22.128:9002/api/Alarms?alarmIndex=2019031915085742'
```

Status: 200

```
[  
  {  
    "DateTime": "2019-02-17T12:22:38.12",  
    "UserName": "Петров",  
    "ActionName": "Вызов полиции",  
    "ActionType": "Other",  
    "GuardName": "",  
    "Comment": "Запасной выход"  
  }, ...  
]
```

## 12 Транспортный файл

### 12.1 Что такое транспортный файл

Это файл в формате .csv, с помощью которого можно, не заходя в «Центр охраны», изменить данные, которые используются в сервисе **Услуги охраны**. Транспортный файл можно создать вручную или выгрузить его автоматически. Автоматическая выгрузка возможна только с помощью технического специалиста.

Одна строка транспортного файла соответствует одной карточке объекта в «Центре охраны». Но если в строке указан номер договора, который используется во многих объектах, то данные изменятся во всех карточках объекта с этим номером договора.

Все поля в строках имеют разделитель - точка с запятой. В «Центр охраны» передается только первые 6 полей строки.

Каждому полю транспортного файла соответствует поле в карточке объекта «Центра охраны». Поэтому важно соблюдать порядок полей при создании файла.

*Пример* Скриншот транспортного файла, созданного с помощью текстового редактора «Блокнот» в системе Windows.



Рис. 1: Пример транспортного файла

С помощью такого файла можно изменить данные для двух карточек объекта с номерами 38 и 55.

### 12.2 Какие поля содержит транспортный файл

#### 1 - Номер объекта

Только целое число. Например, 38 или 112.

#### 2 - Номер договора

Любое значение, которое не содержит точку с запятой. Потому что этот символ является разделителем в csv-файле. Например, ДДУ-455.

#### 3 - Баланс

Целые и десятичные числа, с разделителем - запятая. Возможно указать отрицательное значение. Например, 123,034 или -1200,45.

#### 4 - Абонентская плата

Целые и десятичные числа с разделителем - запятая. Нельзя указать отрицательное значение. Например, 455,234 или 1200.

#### 5 - Дата списания

Дата в формате гггг-мм-дд. Например, 2018-11-14.

#### 6 - Уровень информирования

Целое число из диапазона от -1 до 3. Например, -1 или 2. Числа вне этого диапазона не принимаются.

Обязательные поля: «Номер объекта» или «Номер договора». Должно быть заполнено хотя бы одно из этих полей.

- Если будет заполнен только «Номер объекта», то строка применится к объекту с указанным номером.
- Если будет заполнен только «Номер договора», то строка применится ко всем объектам, имеющим указанный номер договора.

- Если будут заполнены оба поля, то строка применится к объекту с указанным номером, при условии, что объект имеет указанный номер договора.

При загрузке файла в «Центр охраны» значения запишутся в аналогичные поля карточки объекта в модуле «Менеджер объектов».

- «Номер договора» и «Номер объекта» расположены на вкладке «Объект».
- «Баланс», «Абонентская плата», «Дата списания» и «Уровень информирования» на вкладке «Охрана».

Значение поля «Уровень информирования» из транспортного файла влияет на переключатель в области «Услуги охраны» на вкладке «Охрана».

Описание того, как поля из транспортного файла отображаются в мобильном приложении клиентов, описано в статье [MyAlarm: Услуги охраны](#) на сайте поддержки.

## 12.3 Что такое уровень информирования

Значение уровня информирования (-1, 0, 1, 2 или 3) в транспортном файле соответствует положению переключателя на вкладке «Охрана» в области «Услуги охраны».

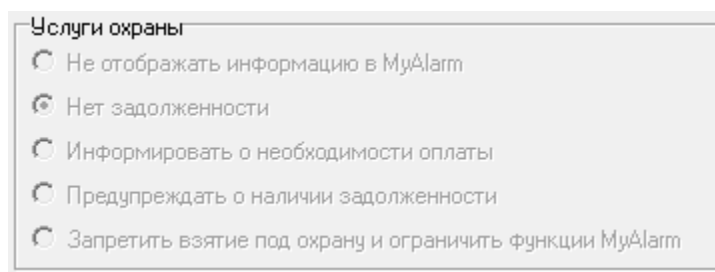


Рис. 2: Менеджер объектов, вкладка Охрана

- -1 Не отображать информацию в MyAlarm
- 0 Нет задолженности
- 1 Информировать о необходимости оплаты
- 2 Предупреждать о наличии задолженности
- 3 Запретить взятие под охраны и ограничить функции MyAlarm

Подробнее о том, как работает уровень информирования в статье [MyAlarm: Услуги охраны](#) на сайте поддержки.

## 12.4 Как создать транспортный файл

Транспортный файл можно создать вручную с помощью текстового редактора, Excel или любой другой похожей программы.

1. Откройте текстовый редактор, например, «Блокнот».
2. Создайте новый файл.
3. В строке перечислите через точку с запятой значения, которые хотите внести в карточку объекта.
4. Сохраните файл в формате \*.csv.



**Важно!** При перечислении значений в строке, нужно точно соблюдать порядок полей. Поля в строке расположены в следующем порядке:

1 номер объекта; 2 номер договора; 3 баланс; 4 абонентская плата; 5 дата списания; 6 уровень информирования

Если вы ошибетесь и внесете значение баланса (номер 3), в поле абонентской платы (номер 4), то клиент увидит в приложении неверную информацию об абонентской плате.

Транспортный файл может создаваться автоматически программой, которая используется в охранном предприятии для учета данных, например, 1С. Настроить выгрузку транспортного файла может технический специалист, обслуживающий эту систему.

## 12.5 Можно ли изменить только некоторые поля с помощью файла

В транспортном файле можно указать значения только для части полей.

Номер объекта и номер договора - это идентификаторы объекта. Если указаны оба поля, то поиск объекта производится по двум полям сразу.

Например, строка в транспортном файле 99;Договор-1;;; не внесет изменения в объект №99, если у номер договора не указан.

Так как номер договора является идентификатором объекта, его невозможно изменить с помощью транспортного файла.

**Важно!** Нельзя просто удалить лишние поля из транспортного файла. Пустые поля должны быть разделены точкой с запятой.

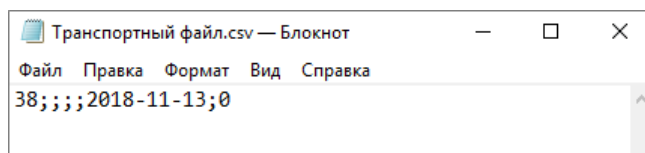


Рис. 3: Пример транспортного файла с частью данных

С помощью такого файла в карточку объекта 38 загрузится только дата списания.

## 12.6 Как изменить значения в нескольких карточках объекта

Если нужно изменить данные по нескольким объектам одновременно, создайте в файле соответствующее количество строк и для каждой из них укажите свои значения.

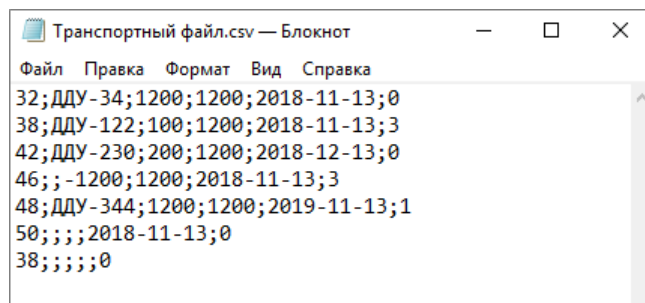


Рис. 4: Пример транспортного файла с частью данных

## 12.7 Как загрузить транспортный файл в Центр охраны

В «Центр охраны» файл загружается автоматически с помощью службы «Центра охраны» C.Nord HTTP-API Service. Чтобы данные из файла начали загружаться в «Центр охраны» автоматически, нужно сделать предварительные настройки, указав путь к файлу и частоту проверки изменений в нем.

## 12.8 Как настроить службу C.Nord.WebApi.Service

Параметры службы задаются в конфигурационном файле C.Nord.WebApi.Service.exe.config, расположенном в папке установки «Центра охраны». В большинстве случаев файл находится по пути: C:\Program Files (x86)\Andromeda\HttpApiService\C.Nord.WebApi.Service.exe.config. Чтобы изменить его содержимое, файл должен быть открыт от имени администратора.

В файле нужно сделать две настройки:

- Частота проверки транспортного файла и загрузки его в «Центр охраны».
- Путь к транспортному файлу.

**Частота проверки транспортного файла и загрузки его в «Центр охраны»** определяет в какие промежутки времени служба будет проверять файл и загружать его в «Центр охраны».

Чтобы изменить параметр: 1. Найдите в файле блок

```
<setting name="TransportFileCheckInterval" serializeAs="String">  
  
    <value>00:30:00</value>
```

2. В тэге укажите желаемое время проверки файла в формате чч:мм:сс.  
Минимальное значение - 10 секунд.

**Путь к транспортному файлу** С помощью этого параметра служба определяет где и какой файл выбрать для загрузки. Чтобы изменить параметр: 1. Найдите в файле блок

```
<setting name="TransportFilePath" serializeAs="String">  
  
    <value>C:\tf\transportfile.csv</value>
```

2. в тэге укажите путь к файлу.  
Например, C:\tf\transportfile.csv

**Важно!** После настройки службы транспортный файл нужно положить в папку, которую вы указали в конфигурационном файле.

### 12.8.1 Что произойдет после настройки службы «C.Nord HTTP-API Service»

Служба проверит наличие файла по адресу, указанному в файле конфигурации. Проверит изменился ли файл с момента последней загрузки. Если изменился, данные из файла загрузятся в «Центр охраны». Если какая-то строка в транспортном файле ошибочна, она игнорируется. Остальные строки будут загружены.

## 13 Техническая поддержка

Если при эксплуатации программного обеспечения «Центр охраны» возникли проблемы или пожелания по его улучшению, мы просим обращаться с ними в службу технической поддержки ООО НТКФ «Си-Норд» по электронной почте [support@cnord.ru](mailto:support@cnord.ru).

При обращении в службу технической поддержки с проблемой, необходимо указать эксплуатируемую версию программного обеспечения «Центр охраны» и описание возникшей неисправности.