



MaxPatrol VM

версия 2.0

Руководство разработчика

© Positive Technologies, 2023.

Настоящий документ является собственностью Positive Technologies и защищен законодательством Российской Федерации и международными соглашениями об авторских правах и интеллектуальной собственности.

Копирование документа либо его фрагментов в любой форме, распространение, в том числе в переводе, а также их передача третьим лицам возможны только с письменного разрешения Positive Technologies.

Документ может быть изменен без предварительного уведомления.

Товарные знаки, использованные в тексте, приведены исключительно в информационных целях, исключительные права на них принадлежат соответствующим правообладателям.

Дата редакции документа: 22.09.2023

Содержание

1.	Об этом документе	6
1.1.	Условные обозначения	6
1.2.	Другие источники информации о MaxPatrol VM	7
2.	Виды запросов к API	8
2.1.	Авторизация	8
2.2.	Получение токена доступа	10
2.3.	Работа с группами активов	14
2.3.1.	Получение идентификатора инфраструктуры	14
2.3.2.	Получение идентификатора группы активов	15
2.3.3.	Создание группы активов	18
2.3.4.	Проверка статуса создания группы	20
2.3.5.	Удаление группы активов	21
2.4.	Импорт активов из CSV	22
2.4.1.	Получение пользовательских полей	23
2.4.2.	Запрос формата CSV-файла	24
2.4.3.	Загрузка CSV-файла	25
2.4.4.	Выгрузка файла ошибок	28
2.4.5.	Запуск импорта активов	28
2.4.6.	Проверка статуса импорта	29
2.5.	Интеграция с активами	31
2.5.1.	Получение списка подписчиков	31
2.5.2.	Регистрация подписчика	32
2.5.3.	Получение информации о подписчике	33
2.5.4.	Удаление подписчика и его табличных списков	34
2.5.5.	Получение списка табличных списков подписчика	34
2.5.6.	Создание или обновление табличного списка	35
2.5.7.	Удаление табличных списков подписчика	37
2.5.8.	Получение информации о табличном списке подписчика	37
2.5.9.	Удаление табличного списка подписчика	38
2.5.10.	Получение изменений табличного списка	39
2.6.	Работа с учетными записями сканирования	40
2.6.1.	Добавление учетной записи типа «логин — пароль»	41
2.6.2.	Получение учетной записи типа «логин — пароль»	42
2.6.3.	Обновление учетной записи типа «логин — пароль»	43
2.6.4.	Добавление учетной записи типа «пароль»	44
2.6.5.	Получение учетной записи типа «пароль»	45
2.6.6.	Обновление учетной записи типа «пароль»	45
2.6.7.	Добавление учетной записи типа «сертификат»	46
2.6.8.	Получение учетной записи типа «сертификат»	47
2.6.9.	Обновление учетной записи типа «сертификат»	48
2.6.10.	Получение всех учетных записей	49
2.6.11.	Удаление учетной записи	50
2.6.12.	Получение списка меток учетных записей	51

2.7.	Работа со сканами	51
2.7.1.	Добавление нового скана	51
2.7.2.	Получение метаданных скана по идентификатору	52
2.7.3.	Получение содержания обработанного скана	53
2.7.4.	Получение коллекции данных обработанных сканов	54
2.7.5.	Получение пакета информации об обработанных сканах	55
2.7.6.	Получение метаданных сырого скана по идентификатору	56
2.7.7.	Получение содержания сырого скана в формате XML	57
2.7.8.	Получение коллекции данных сырых сканов	58
2.7.9.	Получение пакета информации о сырых сканах	59
2.8.	Работа с профилями сканирования	60
2.8.1.	Получение списка профилей сканирования	60
2.8.2.	Создание профиля сканирования	61
2.8.3.	Получение расширенной информации о профиле сканирования	63
2.8.4.	Обновление профиля сканирования	65
2.8.5.	Удаление профиля сканирования	66
2.8.6.	Генерация профиля PenTest	67
2.8.7.	Проверка наличия уязвимости при сканировании в режиме пентеста	68
2.8.8.	Получение списка схем профилей сканирования	69
2.8.9.	Получение локализации параметров схемы модуля	70
2.8.10.	Удаление из базы ошибок, возникших при миграции профиля	71
2.9.	Управление задачами сканирования	71
2.9.1.	Запрос отчета о всех задачах	72
2.9.2.	Создание задачи	75
2.9.3.	Запрос количества задач	78
2.9.4.	Запрос отчета о всех точках сохранения всех задач	79
2.9.5.	Запрос количества задач по инфраструктуре	80
2.9.6.	Запрос отчета по задаче по идентификатору	80
2.9.7.	Обновление задачи	83
2.9.8.	Удаление задачи	85
2.9.9.	Валидация задачи	86
2.9.10.	Сброс закладок для задачи сканирования	87
2.9.11.	Запуск задачи сканирования	87
2.9.12.	Остановка запуска задачи	88
2.9.13.	Удаление из базы ошибок, возникших при миграции задачи	88
2.10.	Журналирование действий пользователя	89
2.10.1.	Получение дерева категорий действий	89
2.10.2.	Получение возможного значения дополнительного поля	90
2.10.3.	Получение описания всех дополнительных полей для всех приложений	91
2.10.4.	Получение отчетов о действиях пользователей	92
2.10.5.	Получение отчетов о действиях пользователей по фильтру	93
2.10.6.	Регистрация событий пользователя	96
2.11.	Работа с активами	96
2.11.1.	Запуск задачи на удаление активов	97
2.11.2.	Получение состояния задачи по удалению активов	98

2.11.3.	Запуск задачи на изменение расположения активов в группах	99
2.11.4.	Получение состояния задачи по обновлению расположения активов в группах	100
2.11.5.	Запуск задачи на обновление паспортов активов	101
2.11.6.	Получение состояния задачи по обновлению паспорта активов	102
2.11.7.	Отмена задачи обновления паспорта активов	104
2.11.8.	Сохранение статуса завершенной задачи	104
3.	Отправка уведомлений через POST-запрос	105
3.1.	Поля POST-запроса для уведомления	106
3.2.	Тестирование приема POST-запросов	107
4.	Обращение в службу технической поддержки	110
4.1.	Техническая поддержка на портале	110
4.2.	Время работы службы технической поддержки	110
4.3.	Как служба технической поддержки работает с запросами	111
4.3.1.	Предоставление информации для технической поддержки	111
4.3.2.	Типы запросов	111
4.3.3.	Время реакции и приоритизация запросов	112
4.3.4.	Выполнение работ по запросу	114

1. Об этом документе

Руководство разработчика содержит информацию о доступных в MaxPatrol VM функциях сервиса REST API.

Руководство адресовано разработчикам, выполняющим интеграцию MaxPatrol VM со сторонними приложениями.

Комплект документации MaxPatrol VM включает в себя следующие документы:

- Этот документ.
- Руководство по внедрению — содержит информацию для внедрения продукта в инфраструктуре организации: от типовых схем развертывания до инструкций по установке, первоначальной настройке, обновлению и удалению продукта.
- Руководство администратора — содержит справочную информацию и инструкции по настройке и администрированию продукта.
- Руководство оператора — содержит сценарии использования продукта для управления информационными активами организации.
- Руководство по настройке источников — содержит рекомендации по интеграции элементов IT-инфраструктуры организации с MaxPatrol VM для сбора событий с источников и аудита активов.
- Синтаксис языка запроса PDQL — содержит справочную информацию и примеры синтаксиса, основных функций и операторов языка PDQL, используемых при работе с MaxPatrol VM.
- PDQL-запросы для анализа активов — содержит информацию о стандартных запросах на языке PDQL, предназначенных для проверки конфигураций активов при работе в MaxPatrol VM.

В этом разделе

[Условные обозначения \(см. раздел 1.1\)](#)

[Другие источники информации о MaxPatrol VM \(см. раздел 1.2\)](#)

1.1. Условные обозначения

В документе приняты условные обозначения.

Таблица 1. Условные обозначения

Пример	Описание
Внимание! При выключении модуля снижается уровень защищенности сети	Предупреждения. Содержат информацию о действиях или событиях, которые могут иметь нежелательные последствия

Пример	Описание
Примечание. Вы можете создать дополнительные отчеты	Примечания. Содержат советы, описания важных частных случаев, дополнительную или справочную информацию, которая может быть полезна при работе с продуктом
► Чтобы открыть файл:	Начало инструкции выделено специальным значком
Нажмите кнопку OK	Названия элементов интерфейса (например, кнопок, полей, пунктов меню) выделены полужирным шрифтом
Выполните команду <i>Stop-Service</i>	Текст командной строки, примеры кода, прочие данные, которые нужно ввести с клавиатуры, выделены специальным шрифтом. Также выделены специальным шрифтом имена файлов и пути к файлам и папкам
Ctrl+Alt+Delete	Комбинация клавиш. Чтобы использовать комбинацию, клавиши нужно нажимать одновременно
<Название программы>	Переменные заключены в угловые скобки

1.2. Другие источники информации о MaxPatrol VM

Вы можете найти дополнительную информацию о MaxPatrol VM [на портале технической поддержки](#).

[Портал](#) содержит статьи базы знаний, новости обновлений продуктов Positive Technologies, ответы на часто задаваемые вопросы пользователей. Для доступа к базе знаний и всем новостям нужно создать на портале учетную запись.

Если вы не нашли нужную информацию или решение проблемы самостоятельно, обратитесь в [службу технической поддержки](#) (см. раздел 4).

2. Виды запросов к API

Раздел содержит информацию о протоколе обмена данными между сторонними приложениями и MaxPatrol VM. Обмен данными производится через программный интерфейс продукта (REST API) по протоколу HTTPS.

В этом разделе

[Авторизация \(см. раздел 2.1\)](#)

[Получение токена доступа \(см. раздел 2.2\)](#)

[Работа с группами активов \(см. раздел 2.3\)](#)

[Импорт активов из CSV \(см. раздел 2.4\)](#)

[Интеграция с активами \(см. раздел 2.5\)](#)

[Работа с учетными записями сканирования \(см. раздел 2.6\)](#)

[Работа со сканами \(см. раздел 2.7\)](#)

[Работа с профилями сканирования \(см. раздел 2.8\)](#)

[Управление задачами сканирования \(см. раздел 2.9\)](#)

[Журналирование действий пользователя \(см. раздел 2.10\)](#)

[Работа с активами \(см. раздел 2.11\)](#)

2.1. Авторизация

Запрос на авторизацию и получение кода.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>:3334/connect/authorize

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 2. Параметры в теле запроса connect/authorize

Параметр	Обязательный	Тип данных	Описание
client_id	Да	String	Идентификатор приложения, например mrx для MaxPatrol VM, ptkb для Knowledge Base, idmgr для Management and Configuration
client_secret	Да	String	Ключ доступа к приложению

Параметр	Обязательный	Тип данных	Описание
<code>redirect_uri</code>	Нет	String	URL для редиректа
<code>response_type</code>	Да	Array [String]	Тип ответа: <ul style="list-style-type: none"> — <code>code</code>; — <code>code id_token</code>; — <code>code id_token token</code>; — <code>code token</code>; — <code>id_token</code>; — <code>id_token token</code>; — <code>token</code>
<code>scope</code>	Да	Array [String]	Права доступа по токену. Доступ по API: <ul style="list-style-type: none"> — <code><Идентификатор приложения>.api</code> — доступ к приложению по API (например, <code>mpx.api</code>, <code>ptkb.api</code>); — <code><Привилегия></code> — выдача токена только при наличии у пользователя необходимых привилегий; — <code>authorization</code> — доступ для аутентификации; — <code>offline_access</code> — доступ к приложению от имени пользователя на длительное время (при этом выдается токен для обновления). Доступ к данным: <ul style="list-style-type: none"> — <code>email</code> — адрес электронной почты пользователя; — <code>openid</code> — возможность использования протокола OpenID; — <code>phone</code> — телефон пользователя; — <code>profile</code> — имя и фамилия пользователя

Значение параметра `client_secret` вы можете получить в конфигурации соответствующего приложения.

Для просмотра конфигурации приложений, установленных на Linux, вы можете использовать следующие команды:

- для приложения MaxPatrol VM: `sudo grep ClientSecret /var/lib/deployer/role_instances/core*/params.yaml`
- для Management and Configuration: `sudo grep ClientSecret /var/lib/deployer/role_instances/ma*/params.yaml`

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 302 (Redirect) и код авторизации.

Возможные коды ошибок и их значения:

400 (Bad request) — синтаксическая ошибка в запросе.

2.2. Получение токена доступа

Запрос для получения и обновления токена доступа к приложениям, зарегистрированным в РТ МС. При аутентификации используется протокол OAuth. Для получения токена доступа необходимо указать учетные данные пользователя РТ МС.

Время действия токена доступа ограничено. При наличии у токена прав `offline_access` вы можете обновить его по токену для обновления (выдается вместе с токеном доступа).

Аутентификация для выполнения запроса не требуется.

Метод и URL запроса:

POST <Корневой URL API>:3334/connect/token

Тело запроса может содержать параметры, описанные в таблице ниже.

Внимание! Для передачи данных в теле запроса необходимо использовать формат `application/x-www-form-urlencoded`.

Таблица 3. Параметры в теле запроса /connect/token

Параметр	Обязательный	Тип данных	Описание
<code>client_id</code>	Да	String	Идентификатор приложения, например <code>mpx</code> для MaxPatrol VM, <code>ptkb</code> для Knowledge Base, <code>idmgr</code> для Management and Configuration
<code>client_secret</code>	Да	String	Ключ доступа к приложению
<code>grant_type</code>	Да	Array [String]	Тип разрешений на аутентификацию: <ul style="list-style-type: none"> — <code>authorization_code</code> — код доступа; — <code>client_credentials</code> — учетные данные;

Параметр	Обязательный	Тип данных	Описание
			<ul style="list-style-type: none"> — <code>delegation</code> — делегирование; — <code>implicit</code> — неявное разрешение; — <code>password</code> — пароль; — <code>refresh_token</code> — токен для обновления токена доступа
<code>password</code>	Нет	String	Пароль учетной записи
<code>response_type</code>	Да	Array [String]	Тип ответа: <ul style="list-style-type: none"> — <code>code</code>; — <code>code id_token</code>; — <code>code id_token token</code>; — <code>code token</code>; — <code>id_token</code>; — <code>id_token token</code>; — <code>token</code>
<code>scope</code>	Да	Array [String]	Права доступа по токену. Доступ по API: <ul style="list-style-type: none"> — <code><Идентификатор приложения>.api</code> — доступ к приложению по API (например, <code>mpx.api</code>, <code>ptkb.api</code>); — <code><Привилегия></code> — выдача токена только при наличии у пользователя необходимых привилегий; — <code>offline_access</code> — доступ к приложению от имени пользователя на длительное время (при этом выдается токен для обновления). Доступ к данным: <ul style="list-style-type: none"> — <code>email</code> — адрес электронной почты пользователя; — <code>openid</code> — возможность использования протокола OpenID;

Параметр	Обязательный	Тип данных	Описание
			<ul style="list-style-type: none"> — <code>phone</code> — телефон пользователя; — <code>profile</code> — имя и фамилия пользователя
<code>username</code>	Нет	String	Логин учетной записи пользователя РТ МС
<code>amr</code>	Нет	String	Признак получения токена: <code>ldap</code> . Параметр добавляется в запрос, если токен нужно получить для учетной записи LDAP

Значение параметра `client_secret` вы можете получить в конфигурации соответствующего приложения.

Для просмотра конфигурации приложений, установленных на Linux, вы можете использовать следующие команды:

- для приложения MaxPatrol VM: `sudo grep ClientSecret /var/lib/deployer/role_instances/core*/params.yaml`
- для Management and Configuration: `sudo grep ClientSecret /var/lib/deployer/role_instances/ma*/params.yaml`

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 4. Поля ответа на запрос `/connect/token`

Поле	Тип данных	Описание
<code>access_token</code>	UUID	Токен доступа
<code>expires_in</code>	Number	Время действия токена в секундах
<code>id_token</code>	UUID	Токен текущей сессии
<code>refresh_token</code>	UUID	Токен для обновления токена доступа
<code>token_type</code>	Enum	Тип токена (Bearer)

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе.


```

    "token_type": "Bearer",
    "refresh_token": "139c8b94705bf596cbe14d536268e70e92b2cf600fa5da2dc71349da6d4df2b1"
  }

```

2.3. Работа с группами активов

С помощью запросов к API вы можете получить идентификаторы созданных в MaxPatrol VM пользовательских инфраструктур и групп активов. Также вы можете создавать и удалять группы активов.

В этом разделе

[Получение идентификатора инфраструктуры \(см. раздел 2.3.1\)](#)

[Получение идентификатора группы активов \(см. раздел 2.3.2\)](#)

[Создание группы активов \(см. раздел 2.3.3\)](#)

[Проверка статуса создания группы \(см. раздел 2.3.4\)](#)

[Удаление группы активов \(см. раздел 2.3.5\)](#)

2.3.1. Получение идентификатора инфраструктуры

Запрос для получения идентификаторов созданных инфраструктур.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scopes/v2/scopes
```

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 5. Параметры в теле запроса /api/scopes/v2/scopes

Параметр	Обязательный	Тип данных	Описание
limit	Нет	Number	Максимальное количество элементов инфраструктуры в ответе
offset	Нет	Number	Количество отбрасываемых элементов

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 6. Поля ответа на запрос /api/scopes/v2/scopes

Поле	Тип данных	Описание
id	UUID	Идентификатор инфраструктуры
name	String	Название инфраструктуры
tenantId	UUID	Идентификатор дочерней организации

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/scopes/v2/scopes
```

Ответ:

```
[
  {
    "id": "00000000-0000-0000-0000-000000000005",
    "name": "Инфраструктура по умолчанию",
    "tenantId": "32b17948-5755-4905-b0f3-17db2402256a"
  }
]
```

2.3.2. Получение идентификатора группы активов

Запрос для получения идентификаторов групп активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/assets_temporal_readmodel/v2/groups/hierarchy
```

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 7. Параметры в теле запроса `/api/assets_temporal_readmodel/v2/groups/hierarchy`

Параметр	Обязательный	Тип данных	Описание
<code>type</code>	Нет	Enum	Тип группы: — <code>all</code> — любой; — <code>static</code> — статическая; — <code>dynamic</code> — динамическая
<code>groupId</code>	Нет	UUID	Идентификатор группы

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 8. Поля ответа на запрос `/api/assets_temporal_readmodel/v2/groups/hierarchy`

Поле	Тип данных	Описание
<code>id</code>	UUID	Идентификатор группы
<code>name</code>	String	Имя группы для отображения в интерфейсе
<code>groupType</code>	Enum	Тип группы: — <code>all</code> — любой; — <code>static</code> — статическая; — <code>dynamic</code> — динамическая
<code>isReadOnly</code>	Bool	Возможно ли изменение группы
<code>isRemovable</code>	Bool	Возможно ли удаление группы
<code>isRoot</code>	Bool	Является ли группа корневой
<code>isInvalidPredicate</code>	Bool	—
<code>isSlow</code>	Bool	—
<code>treePath</code>	String	Полный путь к группе в списке
<code>children</code>	Array	Список вложенных групп

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/assets_temporal_readmodel/v2/groups/hierarchy
```

Ответ:

```
[
  {
    "id": "00000000-0000-0000-0000-000000000002",
    "name": "Root",
    "groupType": "static",
    "isReadOnly": true,
    "isRemovable": false,
    "isRoot": true,
    "isInvalidPredicate": false,
    "isSlow": false,
    "treePath": "Root",
    "children": [
      {
        "id": "1685ce7c-e440-0001-0000-000000000005",
        "name": "Import",
        "groupType": "static",
        "isReadOnly": false,
        "isRemovable": true,
        "isRoot": false,
        "isInvalidPredicate": false,
        "isSlow": false,
        "treePath": "Import",
        "children": []
      },
      {
        "id": "1685cf07-70c0-0001-0000-000000000009",
        "name": "Windows",
        "groupType": "dynamic",
        "isReadOnly": false,
        "isRemovable": true,
        "isRoot": false,
        "isInvalidPredicate": false,
        "isSlow": false,
        "treePath": "Windows",
        "children": []
      },
      {
        "id": "00000000-0000-0000-0000-000000000003",
        "name": "Unmanaged hosts",
        "groupType": "dynamic",
```

```

        "isReadOnly": true,
        "isRemovable": false,
        "isRoot": false,
        "isInvalidPredicate": false,
        "isSlow": false,
        "treePath": "Unmanaged hosts",
        "children": []
    }
]
}
]

```

2.3.3. Создание группы активов

Запрос для создания группы активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/assets_processing/v2/groups

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 9. Параметры в теле запроса api/assets_processing/v2/groups

Параметр	Обязательный	Тип данных	Описание
description	Нет	String	Описание группы
groupType	Да	Enum	Тип группы: <ul style="list-style-type: none"> — <code>undefined</code> — не определен; — <code>static</code> — статическая; — <code>dynamic</code> — динамическая
metrics	Да	Array [String]	Контекстные метрики CVSS: <ul style="list-style-type: none"> — <code>ar</code> — требования к доступности; — <code>cdp</code> — вероятность нанесения косвенного ущерба; — <code>cr</code> — требования к конфиденциальности; — <code>ir</code> — требования к целостности; — <code>td</code> — плотность целей
name	Да	String	Название группы

Параметр	Обязательный	Тип данных	Описание
organizationInformation	Да	Array [String]	Информация об организации: <ul style="list-style-type: none"> — <code>contactUserId</code> — идентификатор пользователя, являющегося контактным лицом; — <code>address</code> — адрес
organizationInfrastructure	Да	Array [String]	Информация об инфраструктуре: <ul style="list-style-type: none"> — <code>internetProviders</code> — провайдеры интернета; — <code>numberOfNodes</code> — количество узлов на периметре; — <code>registeredDomains</code> — зарегистрированные домены; — <code>usedNetworkApplications</code> — используемые веб-приложения; — <code>usedNetworks</code> — используемые сети
parentId	Да	UUID	Идентификатор родительской группы
predicate	Нет	String	PDQL-фильтр активов для динамической группы

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created). Ответ может содержать поля, описанные в таблице ниже.

Таблица 10. Поля ответа на запрос `api/assets_processing/v2/groups`

Поле	Тип данных	Описание
operationId	UUID	Идентификатор операции создания группы

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
POST https://localhost/api/assets_processing/v2/groups
{
```

```

"name": "Import",
"description": "Import",
"parentId": "00000000-0000-0000-0000-000000000002",
"groupType": "static",
"metrics": {
  "ar": "ND",
  "cdp": "ND",
  "cr": "ND",
  "ir": "ND",
  "td": "ND"
},
"organizationInformation": {
  "address": "",
  "contactUserId": ""
},
"organizationInfrastructure": {
  "internetProviders": "",
  "numberOfNodes": "",
  "registeredDomains": "",
  "usedNetworkApplications": "",
  "usedNetworks": ""
}
}

```

Ответ:

```

{
  "operationId": "14a5ca6b-f780-a001-0000-00000000003f"
}

```

2.3.4. Проверка статуса создания группы

Запрос для получения статуса операции по созданию группы активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/assets_processing/v2/groups/operations/<Идентификатор операции>
```

URL запроса должен содержать идентификатор операции по созданию группы активов.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 11. Поля ответа на запрос `api/assets_processing/v2/groups/operations/<Идентификатор операции>`

Поле	Тип данных	Описание
<Идентификатор группы>	UUID	Идентификатор группы

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция по созданию группы выполняется.
- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/assets_processing/v2/groups/operations/168600a8-8f00-a001-0000-000000000008
```

Ответ:

```
"168600a9-8dc0-0001-0000-00000000000d"
```

2.3.5. Удаление группы активов

Запрос для удаления группы активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/assets_processing/v2/groups/removeOperation
```

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 12. Параметры в теле запроса `/api/assets_processing/v2/groups/removeOperation`

Параметр	Обязательный	Тип данных	Описание
groupIds	Да	Array [UUID]	Идентификаторы групп

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created). Ответ может содержать поля, описанные в таблице ниже.

Таблица 13. Поля ответа на запрос /api/assets_processing/v2/groups/removeOperation

Поле	Тип данных	Описание
operationId	UUID	Идентификатор операции удаления группы

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
POST https://localhost/api/assets_processing/v2/groups/removeOperation
{
  "groupIds": [
    "14a57a85-8b00-0001-0000-000000000015"
  ]
}
```

Ответ:

```
{
  "operationId": "14cdba4b-c680-a001-0000-000000000384e"
}
```

2.4. Импорт активов из CSV

С помощью запросов к API вы можете импортировать в MaxPatrol VM список активов из CSV-файла. Для этого необходимо:

1. Если в модель активов были добавлены пользовательские поля — получить данные этих полей.
2. Создать CSV-файл в соответствии с требуемым форматом.
3. Загрузить CSV-файл со списком активов.
4. Запустить процедуру импорта активов.

В этом разделе

[Получение пользовательских полей \(см. раздел 2.4.1\)](#)

[Запрос формата CSV-файла \(см. раздел 2.4.2\)](#)

[Загрузка CSV-файла \(см. раздел 2.4.3\)](#)

[Выгрузка файла ошибок \(см. раздел 2.4.4\)](#)

[Запуск импорта активов \(см. раздел 2.4.5\)](#)

[Проверка статуса импорта \(см. раздел 2.4.6\)](#)

2.4.1. Получение пользовательских полей

Запрос для получения данных пользовательских полей в модели активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/assets_processing/v2/csv/metadata
```

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 14. Поля ответа на запрос /api/assets_processing/v2/csv/metadata

Поле	Тип	Описание
modelName	String	Название корневой группы параметров в пользовательской модели активов
propertyDeclarations	Array	Свойства пользовательского поля
propertyName	String	Название пользовательского поля
propertyType	String	Тип данных пользовательского поля

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/assets_processing/v2/csv/metadata
```

Ответ:

```
[
  {
    "propertyName": "uf_assetnumber",
    "propertyDeclarations": [
```

```

    {
      "modelName": "Core.Host",
      "propertyType": "int"
    }
  ],
  {
    "propertyName": "uf_assetowner",
    "propertyDeclarations": [
      {
        "modelName": "Core.Host",
        "propertyType": "string"
      }
    ]
  },
  {
    "propertyName": "uf_assetrevisiondate",
    "propertyDeclarations": [
      {
        "modelName": "Core.Host",
        "propertyType": "datetime"
      }
    ]
  }
]

```

2.4.2. Запрос формата CSV-файла

Запрос для получения формата CSV-файла.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/assets_processing/v2/csv/example

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Тело ответа содержит пример содержимого CSV-файла с данными актива.

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/assets_processing/v2/csv/example
```

Ответ:

```
"typealias";"Fqdn";"Hostname";"Ip";"Mac";"IsVirtual"
"";"dns.somedomain.ru";"w2k3sp2x86s5";"192.168.0.4|182.168.10.1";"00:50:56:A6:0C:36|
00:60:56:A6:0C:36";"false"
```

2.4.3. Загрузка CSV-файла

Запрос для загрузки CSV-файла с данными активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/assets_processing/v2/csv/import_operation
```

Параметры строки запроса описаны в таблице ниже.

Таблица 15. Параметры строки запроса /api/assets_processing/v2/csv/import_operation

Параметр	Обязательный	Тип данных	Описание
scope	Да	UUID	Идентификатор инфраструктуры

В запрос необходимо добавить заголовок `Content-Disposition` с описанием CSV-файла, а в тело запроса — CSV-файл с данными активов.

Формат CSV-файла

Файл должен быть представлен в кодировке UTF-8 с BOM. Первая строка файла должна содержать названия полей актива. Вторая и последующие строки — значения полей импортируемых активов (одна строка соответствует одному активу). Значения полей должны быть разделены точкой с запятой. Значения текстовых полей должны быть заключены в кавычки.

Таблица 16. Колонки CSV-файла

Колонка	Тип данных	Описание
Fqdn	String	Полное доменное имя
Hostname	String	Имя узла
Ip	Array [String]	IP-адрес. Поле может содержать несколько значений, которые должны быть разделены вертикальной чертой

Колонка	Тип данных	Описание
IsVirtual	Bool	Является ли виртуальным
Mac	String [String]	MAC-адрес. Поле может содержать несколько значений, которые должны быть разделены вертикальной чертой
typealias	Enum	<p>Тип актива:</p> <ul style="list-style-type: none"> – Alcatel OmniSwitch (под управлением AOS) — <code>omniswitch</code>; – BSD, FreeBSD и macOS — <code>bsd</code>; – Check Point GAiA OS (межсетевой экран) — <code>checkpoint</code>; – Check Point SPLAT (межсетевой экран) — <code>checkpoint</code>; – Cisco ACS — <code>acs</code>; – Cisco ASA (межсетевой экран) — <code>asa</code>; – Cisco FWSM (межсетевой экран) — <code>fwsn</code>; – Cisco IOS — <code>ios</code>; – Cisco IOS XE — <code>ios</code>; – Cisco ISE — <code>ise</code>; – Cisco Nexus — <code>nexus</code>; – Cisco PIX (межсетевой экран) — <code>pix</code>; – FortiNet FortiGate (межсетевой экран) — <code>fortigate</code>; – HPE HP-UX — <code>hp_ux</code>; – Huawei VRP — <code>vrp</code>; – IBM AIX — <code>aix</code>; – Juniper JunOS — <code>junos</code>; – Linux (все семейство ОС) — <code>linux</code>; – Oracle Solaris — <code>solaris</code>; – Palo Alto Networks PAN-OS (межсетевые экраны) — <code>pan_os</code>; – VMware vSphere Hypervisor (ESXi) — <code>esxi</code>; – Windows — <code>windows</code>

Колонка	Тип данных	Описание
<Пользователь-ские поля>	—	Добавленные пользователем поля актива, не являющиеся стандартными

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 17. Поля ответа на запрос `/api/assets_processing/v2/csv/import_operation`

Поле	Тип данных	Описание
<code>id</code>	UUID	Идентификатор операции импорта
<code>isLogFileCreated</code>	Bool	Создан ли файл с ошибками
<code>rowCountExceeded</code>	Bool	Превышает ли количество строк в импортируемом файле установленное ограничение
<code>totalRowCount</code>	Number	Общее количество строк в импортируемом файле
<code>validRowCount</code>	Number	Количество строк с валидными данными активов в импортируемом файле

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
POST https://localhost/api/assets_processing/v2/csv/import_operation?
scopeId=00000000-0000-0000-0000-000000000005
```

Заголовок запроса:

```
Content-Disposition:form-data; name="upfile"; filename="Assets_list.csv"
```

В тело запроса (в представлении `form-data`) необходимо добавить CSV-файл `Assets_list.csv` с ключом `upfile`.

Ответ:

```
{
  "id": "0e372968-e18b-471e-b3b7-ad0d927c2dd9",
  "isLogFileCreated": true,
  "rowCountExceeded": false,
  "validRowCount": 2,
```

```
"totalCount": 3
}
```

2.4.4. Выгрузка файла ошибок

Запрос для выгрузки списка ошибок, обнаруженных при импорте CSV-файла со списком активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/assets_processing/v2/csv/import_operation/<Идентификатор операции>/logfile
```

URL запроса должен содержать идентификатор операции импорта CSV-файла.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Заголовок ответа **Content-Disposition** содержит информацию о выгруженном файле с ошибками. Тело ответа содержит список ошибок с указанием номера строки в импортированном CSV-файле. Если файл с ошибками не создан, сервис возвращает код 204 (No Content).

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/assets_processing/v2/csv/import_operation/0e372968-e18b-471e-b3b7-ad0d927c2dd9/logfile
```

Ответ:

```
"3";"Значение [version] некорректно для псевдонима"
```

2.4.5. Запуск импорта активов

Запрос для запуска импорта активов из загруженного CSV-файла.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/assets_processing/v2/csv/import_operation/<Идентификатор операции>/start
```

URL запроса должен содержать идентификатор операции импорта CSV-файла.

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 18. Параметры в теле запроса `api/assets_processing/v2/csv/import_operation/<Идентификатор операции>/start`

Параметр	Обязательный	Тип данных	Описание
groupsId	Да	Array [UUID]	Идентификаторы групп, в которые нужно добавить активы

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK).

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
POST https://localhost/api/assets_processing/v2/csv/import_operation/0e372968-e18b-471e-b3b7-ad0d927c2dd9/start
{
  "groupsId": [
    "1499141b-4c00-0001-0000-00000000000d"
  ]
}
```

2.4.6. Проверка статуса импорта

Запрос для получения статуса импорта активов из CSV-файла.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/assets_processing/v2/csv/import_operation/<Идентификатор операции>/state
```

URL запроса должен содержать идентификатор операции импорта CSV-файла.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 19. Поля ответа на запрос `api/assets_processing/v2/csv/import_operation/<Идентификатор файла>/state`

Поле	Тип данных	Описание
<code>errorModel</code>	Array	Данные ошибок
<code>errorModel → code</code>	Number (int32)	Код ошибки
<code>errorModel → errors</code>	Array	Название ошибки
<code>errorModel → message</code>	String	Текст ошибки
<code>state</code>	Enum	Статус импорта: — <code>completed</code> — завершен; — <code>failed</code> — не выполнен; — <code>inprogress</code> — выполняется; — <code>validated</code> — на проверке
<code>succeedCount</code>	Number	Количество импортированных активов
<code>updatedGroups</code>	Array	Данные обновленной группы
<code>updatedGroups → displayName</code>	String	Название группы
<code>updatedGroups → id</code>	UUID	Идентификатор группы

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
POST https://localhost/api/assets_processing/v2/csv/import_operation/0e372968-e18b-471e-b3b7-ad0d927c2dd9/state
```

Ответ:

```
{
  "state": "completed",
  "succeedCount": 1,
  "updatedGroups": [
    {
      "id": "149f37fa-c2c0-0001-0000-000000000011",
      "displayName": "Import"
    }
  ],
  "errorModel": null
}
```

2.5. Интеграция с активами

С помощью запросов к API вы можете получать информацию о подписчике и его табличных списках, регистрировать подписчика, создавать табличные списки, а также удалять подписчика и его табличные списки.

В этом разделе

[Получение списка подписчиков \(см. раздел 2.5.1\)](#)

[Регистрация подписчика \(см. раздел 2.5.2\)](#)

[Получение информации о подписчике \(см. раздел 2.5.3\)](#)

[Удаление подписчика и его табличных списков \(см. раздел 2.5.4\)](#)

[Получение списка табличных списков подписчика \(см. раздел 2.5.5\)](#)

[Создание или обновление табличного списка \(см. раздел 2.5.6\)](#)

[Удаление табличных списков подписчика \(см. раздел 2.5.7\)](#)

[Получение информации о табличном списке подписчика \(см. раздел 2.5.8\)](#)

[Удаление табличного списка подписчика \(см. раздел 2.5.9\)](#)

[Получение изменений табличного списка \(см. раздел 2.5.10\)](#)

2.5.1. Получение списка подписчиков

Запрос для получения списка подписчиков.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/event-tables/v2/subscribers/
```

Параметры строки запроса описаны в таблице ниже.

Таблица 20. Параметры строки запроса `/api/event-tables/v2/subscribers/`

Параметр	Обязательный	Тип данных	Описание
<code>types</code>	Нет	Array [String]	Фильтр по именам типов подписчиков

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 21. Поля ответа на запрос `/api/event-tables/v2/subscribers/`

Поле	Тип данных	Описание
<code>subscribers</code>	Array	<p>Список зарегистрированных подписчиков на изменения табличных списков:</p> <ul style="list-style-type: none"> — <code>id</code> — идентификатор подписчика. Обязательный параметр, тип String; — <code>name</code> — имя подписчика. Обязательный параметр, тип String; — <code>type</code> — имя типа подписчика. Обязательный параметр, тип String

2.5.2. Регистрация подписчика

Запрос для регистрации подписчика.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST `<Корневой URL API>/api/event-tables/v2/subscribers/`

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 22. Параметры в теле запроса `/api/event-tables/v2/subscribers/`

Параметр	Обязательный	Тип данных	Описание
<code>subscriberInfo</code>	Да	Object	Информация о подписчике: <ul style="list-style-type: none"> <code>name</code> — имя подписчика, тип String; <code>type</code> — имя типа подписчика, тип String

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 23. Поля ответа на запрос `/api/event-tables/v2/subscribers/`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор подписчика

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе.

2.5.3. Получение информации о подписчике

Запрос для получения информации о подписчике.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/event-tables/v2/subscribers/{subscriberId}
```

URL запроса содержит path-параметр `subscriberId` — уникальный идентификатор подписчика.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 24. Поля ответа на запрос `/api/event-tables/v2/subscribers/{subscriberId}`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор подписчика

Поле	Тип данных	Описание
name	String	Имя подписчика
type	String	Имя типа подписчика

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе.

2.5.4. Удаление подписчика и его табличных списков

Запрос на идемпотентное удаление подписчика и его табличных списков.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/event-tables/v2/subscribers/{subscriberId}
```

URL запроса содержит path-параметр `subscriberId` — идентификатор подписчика.

Ответ на запрос

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция по созданию группы выполняется.

2.5.5. Получение списка табличных списков подписчика

Запрос для получения списка табличных списков подписчика.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/event-tables/v2/subscribers/{subscriberId}/tables/{tableListId}
```

URL запроса содержит path-параметр `subscriberId` — уникальный идентификатор подписчика.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 25. Поля ответа на запрос `/api/event-tables/v2/subscribers/{subscriberId}`

Поле	Тип данных	Описание
<code>lists</code>	Array	<p>Список табличных списков:</p> <ul style="list-style-type: none"> — <code>id</code> — идентификатор табличного списка, тип String; — <code>name</code> — имя табличного списка. Обязательный параметр, тип String; — <code>isValid</code> — валиден ли табличный список. Обязательный параметр, тип Bool. Значение по умолчанию: <code>true</code>; — <code>notifications</code> — перечень уведомлений о состоянии табличного списка: <ul style="list-style-type: none"> • <code>errorType</code> — тип ошибки. Обязательный параметр, тип String; — <code>ttd</code> — максимальное время жизни табличного списка, указанное при создании. Пример значения: <code>90.00:00:00</code>. Необязательный параметр, тип String (\$time-span)

2.5.6. Создание или обновление табличного списка

Запрос для создания или обновления табличного списка.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/event-tables/v2/subscribers/{subscriberId}/tables/

URL запроса содержит path-параметр `subscriberId` — уникальный идентификатор подписчика.

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 26. Параметры в теле запроса /api/event-tables/v2/subscribers/{subscriberId}/tables/

Параметр	Обязательный	Тип данных	Описание
Параметры команды			
command	Да	Object	<p>Представление команды на создание или обновление табличного списка:</p> <ul style="list-style-type: none"> — <code>name</code> — имя табличного списка подписчика для идентификации списка в рамках подписчика. Не зависит от регистра. Обязательный параметр, тип String; — <code>schema</code> — схема табличного списка. Обязательный параметр, тип Array; <ul style="list-style-type: none"> • <code>fields</code> — набор полей табличного списка (см. ниже). Обязательный параметр, тип Array; • <code>schemaType</code> — поле для передачи дискриминатора типа объекта. Обязательный параметр, тип String; — <code>ttd</code> — время удаления табличного списка с момента создания или последнего получения данных. Пример значения: <code>90.00:00:00</code>. Необязательный параметр, тип String(\$time-span)
Параметры колонок (fields)			
sourceFieldName	Да	String	Имя колонки исходного справочника
isNullable	Да	Bool	Может ли содержать null

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 27. Поля ответа на запрос /api/event-tables/v2/subscribers/{subscriberId}/tables/

Поле	Тип данных	Описание
id	String	Идентификатор табличного списка. Обязательный параметр

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе.

2.5.7. Удаление табличных списков подписчика

Запрос для идемпотентного удаления всех табличных списков подписчика.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/event-tables/v2/subscribers/{subscriberId}/tables/
```

URL запроса содержит path-параметр `subscriberId` — уникальный идентификатор подписчика.

Ответ на запрос

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция выполняется.

2.5.8. Получение информации о табличном списке подписчика

Запрос для получения информации о табличном списке подписчика.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/event-tables/v2/subscribers/{subscriberId}/tables/
```

URL запроса содержит path-параметры `subscriberId` — уникальный идентификатор подписчика и `tableListId` — уникальный идентификатор табличного списка для указанного подписчика.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 28. Поля ответа на запрос `/api/event-tables/v2/subscribers/{subscriberId}/tables/`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор табличного списка. Обязательный параметр
<code>name</code>	String	Имя подписчика. Обязательный параметр
<code>isValid</code>	Bool	Валиден ли табличный список. Обязательный параметр. Значение по умолчанию: <code>true</code>
<code>notifications</code>	Array	Перечень уведомлений о состоянии табличного списка: — <code>errorType</code> — тип ошибки. Обязательный параметр, тип String.
<code>ttl</code>	String (\$time-span)	Время удаления табличного списка с момента создания или последнего получения данных. Пример значения: <code>90.00:00:00</code> . Необязательный параметр

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе.

2.5.9. Удаление табличного списка подписчика

Запрос для идемпотентного удаления табличного списка подписчика.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/event-tables/v2/subscribers/{subscriberId}/tables/{tableListId}
```

URL запроса содержит path-параметры `subscriberId` — уникальный идентификатор подписчика и `tableListId` — уникальный идентификатор табличного списка для указанного подписчика.

Ответ на запрос

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция по созданию группы выполняется.

2.5.10. Получение изменений табличного списка

Запрос для получения изменений табличного списка. Если схема табличного списка стала невалидной, изменения табличного списка возможно получить только за период до инвалидации схемы.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/event-tables/v2/subscribers/{subscriberId}/tables/{tableListId}/changes
```

URL запроса содержит path-параметры `subscriberId` — уникальный идентификатор подписчика и `tableListId` — уникальный идентификатор табличного списка для указанного подписчика.

Параметры строки запроса описаны в таблице ниже.

Таблица 29. Параметры строки запроса `/api/event-tables/v2/subscribers/{subscriberId}/changes`

Параметр	Обязательный	Тип данных	Описание
token	Нет	String	Токен изменений табличного списка. Если параметр отсутствует — выборка из первого элемента
timeout	Нет	String (\$time-span)	Время максимального ожидания ответа сервера. Пример значения: <code>90.00:00:00</code>

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 30. Поля ответа на запрос `/api/event-tables/v2/subscribers/{subscriberId}/tables/changes`

Поле	Тип данных	Описание
Параметры табличного списка		
nextToken	String	Токен для получения набора изменений табличного списка. Необязательный параметр
changes	Array	Набор изменений табличного списка. Необязательный параметр
isSchemaValid	Bool	Валидна ли схема табличного списка. Обязательный параметр

Поле	Тип данных	Описание
Параметры набора изменений табличного списка		
added	Array	Добавленные записи. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1, additionalProp2, additionalProp3 ... additionalPropN</code> . Обязательный параметр
removed	Array	Удаленные записи. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1, additionalProp2, additionalProp3 ... additionalPropN</code> . Обязательный параметр

Возможные коды ошибок и их значения:

— 400 (Bad request) — синтаксическая ошибка в запросе.

2.6. Работа с учетными записями сканирования

С помощью запросов к API вы можете добавлять учетные записи сканирования с типами «логин — пароль», «пароль» и «сертификат», получать и обновлять информацию об учетных записях с такими типами, а также получать список всех учетных записей и их меток и удалять учетные записи.

В этом разделе

[Добавление учетной записи типа «логин — пароль» \(см. раздел 2.6.1\)](#)

[Получение учетной записи типа «логин — пароль» \(см. раздел 2.6.2\)](#)

[Обновление учетной записи типа «логин — пароль» \(см. раздел 2.6.3\)](#)

[Добавление учетной записи типа «пароль» \(см. раздел 2.6.4\)](#)

[Получение учетной записи типа «пароль» \(см. раздел 2.6.5\)](#)

[Обновление учетной записи типа «пароль» \(см. раздел 2.6.6\)](#)

[Добавление учетной записи типа «сертификат» \(см. раздел 2.6.7\)](#)

[Получение учетной записи типа «сертификат» \(см. раздел 2.6.8\)](#)

[Обновление учетной записи типа «сертификат» \(см. раздел 2.6.9\)](#)

[Получение всех учетных записей \(см. раздел 2.6.10\)](#)

[Удаление учетной записи \(см. раздел 2.6.11\)](#)

[Получение списка меток учетных записей \(см. раздел 2.6.12\)](#)

2.6.1. Добавление учетной записи типа «логин — пароль»

Запрос для добавления учетной записи типа «логин — пароль».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/v3/credentials/login_passwords

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 31. Параметры в теле запроса /api/v3/credentials/login_passwords

Параметр	Обязательный	Тип данных	Описание
name	Да	String	Название учетной записи
description	Да	String	Описание
credentialTags	Нет	String	Метки учетной записи — если она создается для определенных методов сбора
domain	Нет	String	Имя домена — если для доступа к источнику используется доменная учетная запись
login	Да	String	Логин учетной записи
password	Нет	String	Пароль учетной записи

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 32. Поля ответа на запрос /api/v3/credentials/login_passwords

Поле	Тип данных	Описание
id	String	Идентификатор учетной записи

Возможные коды ошибок и их значения:

— 400 (Bad request) — синтаксическая ошибка в запросе.

2.6.2. Получение учетной записи типа «логин — пароль»

Запрос для получения учетной записи типа «логин — пароль».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/v3/credentials/login_passwords/{id}

URL запроса содержит path-параметр `id` — идентификатор учетной записи.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 33. Параметры ответа на запрос /api/v3/credentials/login_passwords/{id}

Параметр	Обязательный	Тип данных	Описание
type	Да	String	Тип учетной записи
id	Да	String	Идентификатор учетной записи
name	Да	String	Название учетной записи
description	Да	String	Описание
credentialTags	Нет	String	Метки учетной записи — если она создается для определенных методов сбора
domain	Нет	String	Имя домена — если для доступа к источнику используется доменная учетная запись
login	Да	String	Логин учетной записи

Возможные коды ошибок и их значения:

— 404 (Not Found) — учетная запись с таким идентификатором не найдена.

2.6.3. Обновление учетной записи типа «логин — пароль»

Запрос для обновления учетной записи типа «логин — пароль».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

PUT <Корневой URL API>/api/v3/credentials/login_passwords/{id}

URL запроса содержит path-параметр `id` — идентификатор учетной записи.

Параметры строки запроса описаны в таблице ниже.

Таблица 34. Параметры строки запроса /api/v3/credentials/login_passwords/{id}

Поле	Обязательный	Тип данных	Описание
<code>name</code>	Да	String	Название учетной записи
<code>description</code>	Да	String	Описание
<code>credentialTags</code>	Нет	String	Метки учетной записи — если она создается для определенных методов сбора
<code>domain</code>	Нет	String	Имя домена — если для доступа к источнику используется доменная учетная запись
<code>login</code>	Да	String	Логин учетной записи
<code>shouldResetPassword</code>	Да	Bool	Сбросить ли пароль от учетной записи
<code>password</code>	Да	String	Пароль учетной записи

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content). Ответ не содержит полей.

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 404 (Not Found) — учетная запись с таким идентификатором не найдена.

2.6.4. Добавление учетной записи типа «пароль»

Запрос для добавления учетной записи типа «пароль».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/v3/credentials/passwords_only

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 35. Параметры в теле запроса /api/v3/credentials/passwords_only

Параметр	Обязательный	Тип данных	Описание
name	Да	String	Название учетной записи
description	Да	String	Описание
credentialTags	Нет	String	Метки учетной записи — если она создается для определенных методов сбора
password	Да	String	Пароль учетной записи

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 36. Поля ответа на запрос /api/v3/credentials/passwords_only

Поле	Тип данных	Описание
id	String	Идентификатор созданной учетной записи

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе.

2.6.5. Получение учетной записи типа «пароль»

Запрос для получения учетной записи типа «пароль».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/v3/credentials/passwords_only/{id}

URL запроса содержит path-параметр `id` — идентификатор учетной записи.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 37. Параметры ответа на запрос /api/v3/credentials/passwords_only/{id}

Поле	Тип данных	Описание
type	String	Тип учетной записи
id	String	Идентификатор учетной записи
name	String	Название учетной записи
description	String	Описание
credentialTags	String	Метки учетной записи — если она создается для определенных методов сбора

Возможные коды ошибок и их значения:

— 404 (Not Found) — учетная запись с таким идентификатором не найдена.

2.6.6. Обновление учетной записи типа «пароль»

Запрос для обновления учетной записи типа «пароль».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

PUT <Корневой URL API>/api/v3/credentials/passwords_only/{id}

URL запроса содержит path-параметр `id` — идентификатор учетной записи.

Параметры строки запроса описаны в таблице ниже.

Таблица 38. Параметры строки запроса `/api/v3/credentials/passwords_only/{id}`

Поле	Обязательный	Тип данных	Описание
name	Да	String	Название учетной записи
description	Да	String	Описание
credentialTags	Нет	String	Метки учетной записи — если она создается для определенных методов сбора
shouldResetPassword	Да	Bool	Сбросить ли пароль от учетной записи
password	Да	String	Пароль учетной записи

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content). Ответ не содержит полей.

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 404 (Not Found) — учетная запись с таким идентификатором не найдена.

2.6.7. Добавление учетной записи типа «сертификат»

Запрос для добавления учетной записи типа «сертификат».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/v3/credentials/certificates

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 39. Параметры в теле запроса `/api/v3/credentials/certificates`

Параметр	Обязательный	Тип данных	Описание
<code>name</code>	Да	String	Название учетной записи
<code>description</code>	Да	String	Описание
<code>credentialTags</code>	Нет	String	Метки учетной записи — если она создается для определенных методов сбора
<code>login</code>	Нет	String	Логин учетной записи
<code>password</code>	Нет	String	Пароль учетной записи
<code>certificate</code>	Да	String	Путь к файлу сертификата

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 40. Поля ответа на запрос `/api/v3/credentials/certificates`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор учетной записи

Возможные коды ошибок и их значения:

— 400 (Bad request) — синтаксическая ошибка в запросе.

2.6.8. Получение учетной записи типа «сертификат»

Запрос для получения учетной записи типа «сертификат».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/v3/credentials/certificates/{id}

URL запроса содержит path-параметр `id` — идентификатор учетной записи.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 41. Параметры ответа на запрос `/api/v3/credentials/certificates/{id}`

Поле	Тип данных	Описание
type	String	Тип учетной записи
id	String	Идентификатор учетной записи
name	String	Название учетной записи
description	String	Описание
login	String	Логин учетной записи
credentialTags	String	Метки учетной записи — если она создается для определенных методов сбора

Возможные коды ошибок и их значения:

— 404 (Not Found) — учетная запись с таким идентификатором не найдена.

2.6.9. Обновление учетной записи типа «сертификат»

Запрос для обновления учетной записи типа «сертификат».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

PUT `<Корневой URL API>/api/v3/credentials/certificates/{id}`

URL запроса содержит path-параметр `id` — идентификатор учетной записи.

Параметры строки запроса описаны в таблице ниже.

Таблица 42. Параметры строки запроса `/api/v3/credentials/certificates/{id}`

Поле	Обязательный	Тип данных	Описание
name	Да	String	Название учетной записи
description	Да	String	Описание

Поле	Обязательный	Тип данных	Описание
credentialTags	Нет	String	Метки учетной записи — если она создается для определенных методов сбора
login	Нет	String	Логин учетной записи
shouldResetCertificatePassword	Да	Bool	Сбросить ли пароль от учетной записи
password	Да	String	Пароль учетной записи
certificate	Да	String	Путь к файлу сертификата

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content). Ответ не содержит полей.

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 404 (Not Found) — учетная запись с таким идентификатором не найдена.

2.6.10. Получение всех учетных записей

Запрос для получения всех учетных записей в краткой форме.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/v3/credentials

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 43. Параметры в теле запроса /api/v3/credentials

Поле	Обязательный	Тип данных	Описание
credentialTags	Нет	String	Метки учетных записей — если они созданы для определенных методов сбора

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 44. Параметры ответа на запрос /api/v3/credentials

Поле	Тип данных	Описание
type	String	Тип учетной записи
id	String	Идентификатор учетной записи
name	String	Название учетной записи
description	String	Описание
credentialTags	String	Метки учетной записи — если она создается для определенных методов сбора

2.6.11. Удаление учетной записи

Запрос для удаления учетной записи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

DELETE <Корневой URL API>/api/v3/credentials/{id}

URL запроса содержит path-параметр `id` — идентификатор учетной записи.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content). Ответ не содержит полей.

Возможные коды ошибок и их значения:

— 404 (Not Found) — учетная запись с таким идентификатором не найдена.

2.6.12. Получение списка меток учетных записей

Запрос для получения списка меток учетных записей.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/v3/credential_tags

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ содержит список меток учетных записей.

2.7. Работа со сканами

С помощью запросов к API вы можете получать метаданные сырого и обработанного скана по идентификатору, получать содержание сырого и обработанного скана, получать коллекцию данных сырых и обработанных сканов, получать пакет информации о сырых и обработанных сканах, а также добавлять новые сканы.

В этом разделе

[Добавление нового скана \(см. раздел 2.7.1\)](#)

[Получение метаданных скана по идентификатору \(см. раздел 2.7.2\)](#)

[Получение содержания обработанного скана \(см. раздел 2.7.3\)](#)

[Получение коллекции данных обработанных сканов \(см. раздел 2.7.4\)](#)

[Получение пакета информации об обработанных сканах \(см. раздел 2.7.5\)](#)

[Получение метаданных сырого скана по идентификатору \(см. раздел 2.7.6\)](#)

[Получение содержания сырого скана в формате XML \(см. раздел 2.7.7\)](#)

[Получение коллекции данных сырых сканов \(см. раздел 2.7.8\)](#)

[Получение пакета информации о сырых сканах \(см. раздел 2.7.9\)](#)

2.7.1. Добавление нового скана

Запрос для добавления нового скана.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

PUT <Корневой URL API>/api/v1/scans/{scanId}

URL запроса содержит path-параметр `scanId` — идентификатор скана.

Параметры строки запроса описаны в таблице ниже.

Таблица 45. Параметры строки запроса /api/v1/scans/{scanId}

Параметр	Обязательный	Тип данных	Описание
source	Да	String	Источник сканирования
scopeId	Да	String	Идентификатор инфраструктуры
time	Да	String	Время сканирования
jobId	Нет	String	Идентификатор задачи сканирования
noTtl	Нет	Bool	Отключено ли устаревание актива
replaceEntities	Нет	Bool	Замещаются ли все сущности при мердже. Значение по умолчанию — <code>false</code>
createOnly	Нет	Bool	Помечен ли скан как предназначенный только для создания актива. Значение по умолчанию — <code>false</code>

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 46. Параметры в теле запроса /api/v1/scans/{scanId}

Параметр	Обязательный	Тип данных	Описание
content	Да	String	Тело скана

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (Everything is OK. Scan accepted).

Возможные коды ошибок и их значения:

— 400 (Badly formatted request) — синтаксическая ошибка в запросе.

2.7.2. Получение метаданных скана по идентификатору

Запрос для получения метаданных скана по идентификатору.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans/{scanId}
```

URL запроса содержит path-параметр `scanId` — идентификатор скана.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 47. Поля ответа на запрос `/api/v1/scans/{scanId}`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор скана
<code>timeStamp</code>	String	Время сканирования
<code>source</code>	String	Источник сканирования
<code>scopeId</code>	String	Идентификатор инфраструктуры
<code>jobId</code>	String	Идентификатор задачи сканирования
<code>noTtl</code>	Bool	Отключено ли устаревание актива
<code>replaceEntities</code>	Bool	Замещаются ли все сущности при мердже. Значение по умолчанию — <code>false</code>
<code>createOnly</code>	Bool	Помечен ли скан как предназначенный только для создания актива. Значение по умолчанию — <code>false</code>
<code>type</code>	String	Тип скана
<code>isValid</code>	Bool	Валиден ли скан
<code>orderedId</code>	Integer	Порядковый номер скана

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

2.7.3. Получение содержания обработанного скана

Запрос для получения содержания валидного обработанного скана в формате XML.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans/{scanId}/content
```

URL запроса содержит path-параметр `scanId` — идентификатор скана.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK).

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

2.7.4. Получение коллекции данных обработанных сканов

Запрос для получения коллекции данных обработанных сканов начиная с указанного времени.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans
```

Параметры строки запроса описаны в таблице ниже.

Таблица 48. Параметры строки запроса `/api/v1/scans`

Параметр	Обязательный	Тип данных	Описание
<code>fromDate</code>	Да	String	Дата, начиная с которой нужно получить данные
<code>offset</code>	Да	Integer	Смещение от начала результатов выборки
<code>limit</code>	Да	Integer	Размер страницы выдачи
<code>scopeId</code>	Нет	String	Идентификатор инфраструктуры сканирования

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 49. Поля ответа на запрос `/api/v1/scans`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор скана
<code>timeStamp</code>	String	Время сканирования
<code>source</code>	String	Источник сканирования

Поле	Тип данных	Описание
scopeId	String	Идентификатор инфраструктуры
jobId	String	Идентификатор задачи сканирования
noTtl	Bool	Отключено ли устаревание актива
replaceEntities	Bool	Замещаются ли все сущности при мердже. Значение по умолчанию — <code>false</code>
createOnly	Bool	Помечен ли скан как предназначенный только для создания актива. Значение по умолчанию — <code>false</code>
type	String	Тип скана
isValid	Bool	Валиден ли скан
orderedId	Integer	Порядковый номер скана

Возможные коды ошибок и их значения:

— 400 (Badly formatted request) — синтаксическая ошибка в запросе.

2.7.5. Получение пакета информации об обработанных сканах

Запрос для получения пакета информации об обработанных сканах.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/v1/scans/packages

Параметры строки запроса описаны в таблице ниже.

Таблица 50. Параметры строки запроса /api/v1/scans/packages

Параметр	Обязательный	Тип данных	Описание
fromCheckpoint	Да	Integer	Токен предыдущего пакета

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 51. Поля ответа на запрос /api/v1/scans/packages

Поле	Тип данных	Описание
id	String	Идентификатор скана
timeStamp	String	Время сканирования
source	String	Источник сканирования
scopeId	String	Идентификатор инфраструктуры
jobId	String	Идентификатор задачи сканирования
noTtl	Bool	Отключено ли устаревание актива
replaceEntities	Bool	Замещаются ли все сущности при мердже. Значение по умолчанию — <code>false</code>
createOnly	Bool	Помечен ли скан как предназначенный только для создания актива. Значение по умолчанию — <code>false</code>
type	String	Тип скана
isValid	Bool	Валиден ли скан
orderedId	Integer	Порядковый номер скана

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

2.7.6. Получение метаданных сырого скана по идентификатору

Запрос для получения метаданных сырого скана по идентификатору.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans/raw/{scanId}
```

URL запроса содержит path-параметр `scanId` — идентификатор скана.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 52. Поля ответа на запрос `/api/v1/scans/raw/{scanId}`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор скана
<code>timeStamp</code>	String	Время сканирования
<code>source</code>	String	Источник сканирования
<code>scopeId</code>	String	Идентификатор инфраструктуры
<code>jobId</code>	String	Идентификатор задачи сканирования
<code>noTtl</code>	Bool	Отключено ли устаревание актива
<code>replaceEntities</code>	Bool	Замещаются ли все сущности при мердже. Значение по умолчанию — <code>false</code>
<code>createOnly</code>	Bool	Помечен ли скан как предназначенный только для создания актива. Значение по умолчанию — <code>false</code>
<code>type</code>	String	Тип скана

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

2.7.7. Получение содержания сырого скана в формате XML

Запрос для получения содержания сырого скана в формате XML.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans/raw/{scanId}/content
```

URL запроса содержит path-параметр `scanId` — идентификатор скана.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK).

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

2.7.8. Получение коллекции данных сырых сканов

Запрос для получения коллекции данных сырых сканов начиная с указанного времени.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/v1/scans/raw

Параметры строки запроса описаны в таблице ниже.

Таблица 53. Параметры строки запроса /api/v1/scans/raw

Параметр	Обязательный	Тип данных	Описание
fromDate	Да	String	Дата, начиная с которой нужно получить данные
offset	Да	Integer	Смещение от начала результатов выборки
limit	Да	Integer	Размер страницы выдачи
scopeId	Нет	String	Идентификатор инфраструктуры сканирования

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 54. Поля ответа на запрос /api/v1/scans/raw

Поле	Тип данных	Описание
id	String	Идентификатор скана
timeStamp	String	Время сканирования
source	String	Источник сканирования
scopeId	String	Идентификатор инфраструктуры
jobId	String	Идентификатор задачи сканирования
noTtl	Bool	Отключено ли устаревание актива
replaceEntities	Bool	Замещаются ли все сущности при мердже. Значение по умолчанию — <code>false</code>
createOnly	Bool	Помечен ли скан как предназначенный только для создания актива. Значение по умолчанию — <code>false</code>
type	String	Тип скана

Возможные коды ошибок и их значения:

— 400 (Badly formatted request) — синтаксическая ошибка в запросе.

2.7.9. Получение пакета информации о сырых сканах

Запрос для получения пакета информации о сырых сканах.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/v1/scans/raw/packages

Параметры строки запроса описаны в таблице ниже.

Таблица 55. Параметры строки запроса /api/v1/scans/raw/packages

Параметр	Обязательный	Тип данных	Описание
token	Нет	String	Токен предыдущего пакета

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 56. Поля ответа на запрос /api/v1/scans/raw/packages

Поле	Тип данных	Описание
token	String	Токен пакета
scans	Array	Перечень описаний сканов: <ul style="list-style-type: none"> — <code>id</code> — идентификатор скана; — <code>timeStamp</code> — время сканирования; — <code>source</code> — источник сканирования; — <code>scopeId</code> — идентификатор инфраструктуры; — <code>jobId</code> — идентификатор задачи сканирования; — <code>noTtl</code> — отключено ли устаревание актива; — <code>replaceEntities</code> — замещаются ли все сущности при мердже;

Поле	Тип данных	Описание
		<ul style="list-style-type: none"> — <code>createOnly</code> — помечен ли скан как предназначенный только для создания актива. Значение по умолчанию — <code>false</code>; — <code>type</code> — тип скана

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

2.8. Работа с профилями сканирования

С помощью запросов к API вы можете создавать, обновлять и удалять профили сканирования, получать информацию о профиле сканирования, генерировать профили PenTest, проверять наличие уязвимости при сканировании в режиме пентеста, получать список схем профилей сканирования и локализацию параметров схемы модуля, а также удалять из базы ошибки, возникшие при миграции профиля.

В этом разделе

[Получение списка профилей сканирования \(см. раздел 2.8.1\)](#)

[Создание профиля сканирования \(см. раздел 2.8.2\)](#)

[Получение расширенной информации о профиле сканирования \(см. раздел 2.8.3\)](#)

[Обновление профиля сканирования \(см. раздел 2.8.4\)](#)

[Удаление профиля сканирования \(см. раздел 2.8.5\)](#)

[Генерация профиля PenTest \(см. раздел 2.8.6\)](#)

[Проверка наличия уязвимости при сканировании в режиме пентеста \(см. раздел 2.8.7\)](#)

[Получение списка схем профилей сканирования \(см. раздел 2.8.8\)](#)

[Получение локализации параметров схемы модуля \(см. раздел 2.8.9\)](#)

[Удаление из базы ошибок, возникших при миграции профиля \(см. раздел 2.8.10\)](#)

2.8.1. Получение списка профилей сканирования

Запрос для получения списка профилей сканирования с краткой информацией, с возможностью фильтрации по модулю.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/scanning/v3/scanner_profiles

Параметры строки запроса описаны в таблице ниже.

Таблица 57. Параметры строки запроса /api/scanning/v3/scanner_profiles

Параметр	Обязательный	Тип данных	Описание
moduleId	Нет	String	Идентификатор модуля

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK) и список профилей сканирования с краткой информацией. Ответ может содержать поля, описанные в таблице ниже.

Таблица 58. Поля ответа на запрос /api/scanning/v3/scanner_profiles

Поле	Тип данных	Описание
id	String	Идентификатор профиля сканирования
name	String	Имя профиля сканирования
isSystem	Bool	Является ли системой
baseProfileName	String	Имя базового профиля
moduleName	String	Имя модуля
moduleId	String	Идентификатор модуля
output	String	Путь к профилям сканирования
agentRequired	Bool	Необходим ли коллектор
baseProfileId	String	Идентификатор базового профиля
validationErrors	Array	Ошибки валидации: — type — тип ошибки

2.8.2. Создание профиля сканирования

Запрос для создания профиля сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/scanner_profiles

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 59. Параметры в теле запроса `/api/scanning/v3/scanner_profiles`

Параметр	Обязательный	Тип данных	Описание
<code>profile</code>	Да	Array	Создаваемый профиль: <ul style="list-style-type: none"> — <code>name</code> — имя профиля, тип <code>string</code>; — <code>description</code> — описание профиля; тип <code>string</code>; — <code>baseProfileId</code> — идентификатор базового профиля; тип <code>string</code>; — <code>overrides</code> — перекрытие параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1</code>, <code>additionalProp2</code>, <code>additionalProp3</code>

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created). Ответ может содержать поля, описанные в таблице ниже.

Таблица 60. Поля ответа на запрос `api/scanning/v3/scanner_profiles`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор созданного профиля

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 61. Поля ответа на запрос `api/scanning/v3/scanner_profiles`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

2.8.3. Получение расширенной информации о профиле сканирования

Запрос для получения расширенной информации о профиле сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/scanner_profiles/{profileId}
```

URL запроса содержит path-параметр `profileId` — идентификатор профиля сканирования.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 62. Поля ответа на запрос `/api/scanning/v3/scanner_profiles/{profileId}`

Поле	Тип данных	Описание
<code>output</code>	String	Путь к профилю сканирования
<code>description</code>	String	Описание
<code>isSystem</code>	Bool	Является ли системой

Поле	Тип данных	Описание
moduleName	String	Имя модуля
moduleId	String	Идентификатор модуля
baseProfileId	String	Идентификатор базового профиля
baseProfileName	String	Имя базового профиля
validationErrors	Array	Ошибки валидации: — type — тип ошибки, тип string
profile	Array	Профиль. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: additionalProp1, additionalProp2, additionalProp3
overrides	Array	Перекрытие параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: additionalProp1, additionalProp2, additionalProp3

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 63. Поля ответа на запрос `api/scanning/v3/scanner_profiles/{profileId}`

Поле	Тип данных	Описание
errors	Array	Ошибки: — error — ошибка: • type — тип ошибки; — source — источник ошибки: • displayName — отображаемое имя; • hostName — имя узла; • ipAddresses — IP-адрес; — sensitive — признак значимости ошибки
message	String	Сообщение
code	Integer	Код ошибки

2.8.4. Обновление профиля сканирования

Запрос для обновления профиля сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

PUT <Корневой URL API>/api/scanning/v3/scanner_profiles/{profileId}

URL запроса содержит path-параметр `profileId` — идентификатор профиля сканирования.

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 64. Параметры в теле запроса /api/scanning/v3/scanner_profiles/{profileId}

Параметр	Обязательный	Тип данных	Описание
profile	Да	Array	<p>Создаваемый профиль:</p> <ul style="list-style-type: none"> — <code>name</code> — имя профиля, тип string; — <code>description</code> — описание профиля; тип string; — <code>overrides</code> — перекрытие параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1</code>, <code>additionalProp2</code>, <code>additionalProp3</code>

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK).

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 65. Поля ответа на запрос `api/scanning/v3/scanner_profiles/{profileId}`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

2.8.5. Удаление профиля сканирования

Запрос для удаления профиля сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/scanning/v3/scanner_profiles/{profileId}
```

URL запроса содержит path-параметр `profileId` — идентификатор профиля сканирования.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK).

Возможные коды ошибок и их значения:

- 204 (No Content) — профиль сканирования не найден;
- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 66. Поля ответа на запрос `api/scanning/v3/scanner_profiles/{profileId}`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

2.8.6. Генерация профиля PenTest

Запрос для генерации профиля PenTest.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/scanner_profiles/generate_profile_for_pentest

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 67. Параметры в теле запроса `/api/scanning/v3/scanner_profiles/generate_profile_for_pentest`

Параметр	Обязательный	Тип данных	Описание
<code>settings</code>	Да	Object	Параметры: <ul style="list-style-type: none"> — <code>useStandardPorts</code> — признак использования стандартных портов, тип bool; — <code>identifiers</code> — идентификаторы; тип string

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 68. Поля ответа на запрос `/api/scanning/v3/scanner_profiles/generate_profile_for_pentest`

Поле	Тип данных	Описание
<code>baseProfileId</code>	String	Идентификатор базового профиля
<code>profile</code>	Array	Профиль сканирования. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1</code> , <code>additionalProp2</code> , <code>additionalProp3</code>

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 69. Поля ответа на запрос `api/scanning/v3/scanner_profiles/generate_profile_for_pentest`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

2.8.7. Проверка наличия уязвимости при сканировании в режиме пентеста

Запрос для проверки наличия уязвимости при сканировании в режиме пентеста.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/scanning/v3/scanner_profiles/pentest/validate_vulners_for_pentest
```

Параметры строки запроса описаны в таблице ниже.

Таблица 70. Параметры строки запроса `/api/scanning/v3/scanner_profiles/ pentest/ validate_vulners_for_pentest`

Параметр	Обязательный	Тип данных	Описание
<code>identifier</code>	Да	String	Идентификатор

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 71. Поля ответа на запрос `/api/scanning/v3/scanner_profiles/ pentest/validate_vulners_for_pentest`

Поле	Тип данных	Описание
<code>result</code>	String	Результат поиска уязвимости

2.8.8. Получение списка схем профилей сканирования

Запрос для получения списка схем профилей сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/scanning/v3/scanner_profile_schemas

Параметры строки запроса описаны в таблице ниже.

Таблица 72. Параметры строки запроса `/api/scanning/v3/scanner_profile_schemas`

Параметр	Обязательный	Тип данных	Описание
<code>moduleId</code>	Нет	String	Идентификатор модуля

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 73. Поля ответа на запрос `/api/scanning/v3/scanner_profile_schemas`

Поле	Тип данных	Описание
<code>moduleId</code>	String	Идентификатор модуля
<code>dataSchema</code>	Array	Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1, additionalProp2, additionalProp3</code>
<code>formSchema</code>	Array	Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1, additionalProp2, additionalProp3</code>

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 74. Поля ответа на запрос `api/scanning/v3/scanner_profile_schemas`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

2.8.9. Получение локализации параметров схемы модуля

Запрос для получения локализации параметров схемы модуля.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/scanning/v3/scanner_profile_schemas/localization

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 75. Поля ответа на запрос `/api/scanning/v3/scanner_profile_schemas/localization`

Поле	Тип данных	Описание
<code>additionalProp1</code>	String	—
<code>additionalProp2</code>	String	—
<code>additionalProp3</code>	String	—

2.8.10. Удаление из базы ошибок, возникших при миграции профиля

Запрос для удаления из базы ошибок, возникших при миграции профиля.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

`DELETE <Корневой URL API>/api/scanning/v3/scanner_profiles/{id}/migration_errors`

URL запроса содержит path-параметр `Id` — идентификатор профиля сканирования.

Ответ на запрос

Если файл с ошибками не создан, сервис возвращает 204 (No Content).

2.9. Управление задачами сканирования

С помощью запросов к API вы можете создавать, обновлять, запускать, останавливать и удалять задачи сканирования, запрашивать отчеты по задачам (в том числе по идентификатору), точки сохранения задач, количество задач (в том числе по инфраструктуре), валидировать задачи, а также сбрасывать закладки для задач и удалять из базы ошибки, возникшие при миграции задач.

В этом разделе

[Запрос отчета о всех задачах \(см. раздел 2.9.1\)](#)

[Создание задачи \(см. раздел 2.9.2\)](#)

[Запрос количества задач \(см. раздел 2.9.3\)](#)

[Запрос отчета о всех точках сохранения всех задач \(см. раздел 2.9.4\)](#)

[Запрос количества задач по инфраструктуре \(см. раздел 2.9.5\)](#)

[Запрос отчета по задаче по идентификатору \(см. раздел 2.9.6\)](#)

[Обновление задачи \(см. раздел 2.9.7\)](#)

[Удаление задачи \(см. раздел 2.9.8\)](#)

[Валидация задачи \(см. раздел 2.9.9\)](#)

[Сброс закладок для задачи сканирования \(см. раздел 2.9.10\)](#)

[Запуск задачи сканирования \(см. раздел 2.9.11\)](#)

[Остановка запуска задачи \(см. раздел 2.9.12\)](#)

[Удаление из базы ошибок, возникших при миграции задачи \(см. раздел 2.9.13\)](#)

2.9.1. Запрос отчета о всех задачах

Запрос для получения отчета о всех задачах.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/scanning/v3/scanner_tasks

Параметры строки запроса описаны в таблице ниже.

Таблица 76. Параметры строки запроса /api/scanning/v3/scanner_tasks

Параметр	Обязательный	Тип данных	Описание
mainFilter	Нет	String	Основной фильтр
additionalFilter	Нет	String	Дополнительный фильтр

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 77. Поля ответа на запрос /api/scanning/v3/scanner_tasks

Поле	Тип данных	Описание
name	String	Название задачи, задается пользователем
component	Array	Компонент: — type — тип компонента
agent	Array	Коллектор: — id — идентификатор коллектора; — name — имя коллектора
scope	Array	Задача по инфраструктуре: — id — идентификатор задачи; — name — название задачи
profile	Array	Профиль: — id — идентификатор профиля; — name — название профиля
module	Array	Модуль: — id — идентификатор модуля; — name — название модуля
metatransports	Array	Метатранспорты и их параметры. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: additionalProp1, additionalProp2, additionalProp3
status	String	Статус задачи сканирования
created	String	Дата создания задачи в формате ISO 8601
lastRun	String	Дата последнего запуска в формате ISO 8601
lastRunErrorLevel	String	Уровень ошибки последнего запуска

Поле	Тип данных	Описание
include	Array	<p>Признак включения целей сбора данных:</p> <ul style="list-style-type: none"> — <code>assets</code> — активы: <ul style="list-style-type: none"> • <code>id</code> — идентификатор актива; • <code>name</code> — имя актива; — <code>targets</code> — IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; — <code>assetsGroups</code> — группы активов: <ul style="list-style-type: none"> • <code>id</code> — идентификатор группы активов; • <code>name</code> — название группы активов
exclude	Array	<p>Признак исключения целей сбора данных:</p> <ul style="list-style-type: none"> — <code>assets</code> — активы: <ul style="list-style-type: none"> • <code>id</code> — идентификатор актива; • <code>name</code> — имя актива; — <code>targets</code> — IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; — <code>assetsGroups</code> — группы активов: <ul style="list-style-type: none"> • <code>id</code> — идентификатор группы активов; • <code>name</code> — название группы активов
isFqdnPriority	Bool	Является ли сканирование активов по FQDN более приоритетным, чем сканирование по IP-адресу
validationState	String	Состояние валидности
lastRunError	Array	<p>Ошибка последнего запуска:</p> <ul style="list-style-type: none"> — <code>type</code> — тип ошибки; — <code>category</code> — категория ошибки
hostDiscovery	Array	<p>Признак обнаружения узлов до начала сбора данных:</p> <ul style="list-style-type: none"> — <code>enabled</code> — признак сканирования отвечающих узлов; — <code>profile</code> — профиль: <ul style="list-style-type: none"> • <code>id</code> — идентификатор профиля; • <code>name</code> — название профиля
hasBookmarks	Bool	Есть ли сохраненное состояние источника

Поле	Тип данных	Описание
credentials	Array	Учетные записи: — id — идентификатор учетной записи; — name — название учетной записи
triggerParameters	Array	Базовые параметры срабатывания по таймеру: — type — тип; — fromDate — дата начала срабатывания; — toDate — дата окончания срабатывания; — isEnabled — разрешено ли срабатывание по таймеру

2.9.2. Создание задачи

Запрос для создания задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/scanner_tasks

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 78. Параметры в теле запроса /api/scanning/v3/scanner_tasks

Параметр	Обязательный	Тип данных	Описание
task	Нет	Object	Задача
Параметры задачи			
name	Да	String	Название задачи, задается пользователем
component	Нет	Array	Компонент: — type — тип компонента
agent	Нет	String	Коллектор
scope	Да	Array	Задача по инфраструктуре: — id — идентификатор задачи; — name — название задачи

Параметр	Обязательный	Тип данных	Описание
profile	Да	Array	Профиль сканирования: <ul style="list-style-type: none"> — <code>id</code> — идентификатор профиля; — <code>name</code> — название профиля
include	Да	Array	Признак включения целей сбора данных: <ul style="list-style-type: none"> — <code>assets</code> — активы: <ul style="list-style-type: none"> • <code>id</code> — идентификатор актива; • <code>name</code> — имя актива; — <code>targets</code> — IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; — <code>assetsGroups</code> — группы активов: <ul style="list-style-type: none"> • <code>id</code> — идентификатор группы активов; • <code>name</code> — название группы активов
exclude	Да	Array	Признак исключения целей сбора данных: <ul style="list-style-type: none"> — <code>assets</code> — активы: <ul style="list-style-type: none"> • <code>id</code> — идентификатор актива; • <code>name</code> — имя актива; — <code>targets</code> — IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; — <code>assetsGroups</code> — группы активов: <ul style="list-style-type: none"> • <code>id</code> — идентификатор группы активов; • <code>name</code> — название группы активов
hostDiscovery	Да	Array	Признак обнаружения узлов до начала сбора данных: <ul style="list-style-type: none"> — <code>enabled</code> — признак сканирования отвечающих узлов; — <code>profile</code> — профиль: <ul style="list-style-type: none"> • <code>id</code> — идентификатор профиля; • <code>name</code> — название профиля

Параметр	Обязательный	Тип данных	Описание
triggerParameters	Да	Array	Базовые параметры срабатывания по таймеру: <ul style="list-style-type: none"> — type — тип; — fromDate — дата начала срабатывания; — toDate — дата окончания срабатывания; — isEnabled — разрешено ли срабатывание по таймеру
isFqdnPriority	Нет	Bool	Является ли сканирование сканирование активов по FQDN более приоритетным, чем сканирование по IP-адресу
overrides	Нет	Array	Переопределение параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: additionalProp1, additionalProp2, additionalProp3
savepoints	Нет	Array	Точки сохранения: <ul style="list-style-type: none"> — target — цель задачи сканирования; — savepoint — точка сохранения

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 79. Поля ответа на запрос /api/scanning/v3/scanner_tasks

Поле	Тип данных	Описание
id	String	Идентификатор задачи

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 80. Поля ответа на запрос /api/scanning/v3/scanner_tasks

Поле	Тип данных	Описание
errors	Array	Ошибки: <ul style="list-style-type: none"> — error — ошибка: <ul style="list-style-type: none"> • type — тип ошибки; • category — категория ошибки; — source — источник ошибки: <ul style="list-style-type: none"> • displayName — отображаемое имя; • hostName — имя узла; • ipAddresses — IP-адрес; — sensitive — признак значимости ошибки
message	String	Сообщение
code	Integer	Код ошибки

2.9.3. Запрос количества задач

Запрос для получения количества задач.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/scanner_tasks/count
```

Параметры строки запроса описаны в таблице ниже.

Таблица 81. Параметры строки запроса /api/scanning/v3/scanner_tasks/count

Параметр	Обязательный	Тип данных	Описание
mainFilter	Нет	String	Основной фильтр
additionalFilter	Нет	String	Дополнительный фильтр

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 82. Поля ответа на запрос /api/scanning/v3/scanner_tasks/count

Поле	Тип данных	Описание
total	Integer	Количество задач сканирования

2.9.4. Запрос отчета о всех точках сохранения всех задач

Запрос для получения отчета о всех точках сохранения всех задач.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/scanner_tasks/bookmarks
```

Параметры строки запроса описаны в таблице ниже.

Таблица 83. Параметры строки запроса /api/scanning/v3/scanner_tasks/bookmarks

Параметр	Обязательный	Тип данных	Описание
offset	Нет	Integer	Смещение страницы от начала выборки
limit	Нет	Integer	Количество запрашиваемых элементов выборки

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 84. Поля ответа на запрос /api/scanning/v3/scanner_tasks/bookmarks

Поле	Тип данных	Описание
items	Array	Представление конкретной точки сохранения: <ul style="list-style-type: none"> — taskId — идентификатор задачи; — scopeId — идентификатор скоупа; — target — цель; — savepoint — точка сохранения
totalItems	Integer	Количество точек сохранения

2.9.5. Запрос количества задач по инфраструктуре

Запрос для получения количества задач по инфраструктуре.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/scanning/v3/scanner_tasks/scope/{scopeId}/count

URL запроса содержит path-параметр `scopeId` — идентификатор задачи по инфраструктуре.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 85. Поля ответа на запрос /api/scanning/v3/scanner_tasks/scope/{scopeId}/count

Поле	Тип данных	Описание
total	Integer	Количество задач по инфраструктуре

2.9.6. Запрос отчета по задаче по идентификатору

Запрос для получения отчета по задаче по идентификатору.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 86. Поля ответа на запрос /api/scanning/v3/scanner_tasks/{id}

Поле	Тип данных	Описание
id	String	Идентификатор задачи
name	String	Название задачи

Поле	Тип данных	Описание
<code>include</code>	Array	<p>Признак включения целей сбора данных:</p> <ul style="list-style-type: none"> — <code>assets</code> — активы: <ul style="list-style-type: none"> • <code>id</code> — идентификатор актива; • <code>name</code> — имя актива; — <code>targets</code> — IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; — <code>assetsGroups</code> — группы активов: <ul style="list-style-type: none"> • <code>id</code> — идентификатор группы активов; • <code>name</code> — название группы активов
<code>exclude</code>	Array	<p>Признак исключения целей сбора данных:</p> <ul style="list-style-type: none"> — <code>assets</code> — активы: <ul style="list-style-type: none"> • <code>id</code> — идентификатор актива; • <code>name</code> — имя актива; — <code>targets</code> — IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; — <code>assetsGroups</code> — группы активов: <ul style="list-style-type: none"> • <code>id</code> — идентификатор группы активов; • <code>name</code> — название группы активов
<code>isFqdnPriority</code>	Bool	Является ли сканирование сканирование активов по FQDN более приоритетным, чем сканирование по IP-адресу
<code>component</code>	Array	<p>Компонент:</p> <ul style="list-style-type: none"> — <code>type</code> — тип компонента
<code>agent</code>	Array	<p>Коллектор:</p> <ul style="list-style-type: none"> — <code>id</code> — идентификатор коллектора; — <code>name</code> — имя коллектора
<code>module</code>	Array	<p>Модуль:</p> <ul style="list-style-type: none"> — <code>id</code> — идентификатор модуля; — <code>name</code> — название модуля

Поле	Тип данных	Описание
scoop	Array	Задача по инфраструктуре: — id — идентификатор задачи; — name — название задачи
profile	Array	Профиль сканирования: — id — идентификатор профиля; — name — название профиля
completed	String	Дата завершения задачи
status	String	Статус задачи сканирования
created	String	Дата создания задачи
lastRun	String	Дата последнего запуска
hasBookmarks	Bool	Сохранено ли состояние источника
lastRunErrorLevel	String	Уровень ошибки последнего запуска
validationState	String	Состояние валидности
hostDiscovery	Array	Признак обнаружения узлов до начала сбора данных: — enabled — признак сканирования отвечающих узлов; — profile — профиль: • id — идентификатор профиля; • name — название профиля
credentials	Array	Учетные записи: — id — идентификатор учетной записи — name — название учетной записи
triggerParameters	Array	Базовые параметры срабатывания по таймеру: — type — тип; — fromDate — дата начала срабатывания; — toDate — дата окончания срабатывания; — isEnabled — разрешено ли срабатывание по таймеру
overrides	Array	Переопределение параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: additionalProp1, additionalProp2, additionalProp3

Возможные коды ошибок и их значения:

404 (Task not found) — задача не найдена.

2.9.7. Обновление задачи

Запрос для обновления задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

PUT <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}

URL запроса содержит path-параметр `id` — идентификатор задачи.

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 87. Параметры в теле запроса /api/scanning/v3/scanner_tasks/{id}

Параметр	Обязательный	Тип данных	Описание
name	Да	String	Название задачи, задается пользователем
component	Нет	Array	Компонент: — <code>type</code> — тип компонента
agent	Нет	Array	Идентификатор коллектора
scope	Да	Array	Идентификатор задачи по инфраструктуре
profile	Да	Array	Идентификатор основного профиля сканирования
include	Да	Array	Признак включения целей сбора данных: — <code>assets</code> — активы: • <code>id</code> — идентификатор актива; • <code>name</code> — имя актива; — <code>targets</code> — IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; — <code>assetsGroups</code> — группы активов: • <code>id</code> — идентификатор группы активов; • <code>name</code> — название группы активов

Параметр	Обязательный	Тип данных	Описание
exclude	Да	Array	Признак исключения целей сбора данных: <ul style="list-style-type: none"> — <code>assets</code> — активы: <ul style="list-style-type: none"> • <code>id</code> — идентификатор актива; • <code>name</code> — имя актива; — <code>targets</code> — IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; — <code>assetsGroups</code> — группы активов: <ul style="list-style-type: none"> • <code>id</code> — идентификатор группы активов; • <code>name</code> — название группы активов
hostDiscovery	Да	Array	Признак обнаружения узлов до начала сбора данных: <ul style="list-style-type: none"> — <code>enabled</code> — признак сканирования отвечающих узлов; — <code>profile</code> — профиль
triggerParameters	Да	Array	Базовые параметры срабатывания по таймеру: <ul style="list-style-type: none"> — <code>type</code> — тип; — <code>fromDate</code> — дата начала срабатывания; — <code>toDate</code> — дата окончания срабатывания; — <code>isEnabled</code> — разрешено ли срабатывание по таймеру
isFqdnPriority	Нет	Bool	Является ли сканирование активов по FQDN более приоритетным, чем сканирование по IP-адресу
overrides	Нет	Array	Перекрытие настроек профиля
savepoints	Нет	Array	Точки сохранения: <ul style="list-style-type: none"> — <code>target</code> — цель задачи сканирования; — <code>savepoint</code> — точка сохранения

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 88. Поля ответа на запрос `/api/scanning/v3/scanner_tasks/{id}`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор задачи

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 89. Поля ответа на запрос `/api/scanning/v3/scanner_tasks/{id}`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; • <code>category</code> — категория ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

2.9.8. Удаление задачи

Запрос для удаления задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (Успешное удаление).

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе.

2.9.9. Валидация задачи

Запрос для валидации задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}/validation
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK).

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 90. Поля ответа на запрос `/api/scanning/v3/scanner_tasks/{id}/validation`

Поле	Тип данных	Описание
<code>errors</code>	Array	<p>Ошибки:</p> <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; • <code>category</code> — категория ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

2.9.10. Сброс закладок для задачи сканирования

Запрос для сброса закладок для задачи сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}/cleanBookmarks

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (Успешный сброс).

2.9.11. Запуск задачи сканирования

Запрос для запуска задачи сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}/start

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 91. Поля ответа на запрос /api/scanning/v3/scanner_tasks/{id}/start

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор задачи

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 92. Поля ответа на запрос `/api/scanning/v3/scanner_tasks/{id}/start`

Поле	Тип данных	Описание
<code>errors</code>	Array	<p>Ошибки:</p> <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; • <code>category</code> — категория ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

2.9.12. Остановка запуска задачи

Запрос для остановки запуска задачи сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}/stop
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (Успешная остановка).

Возможные коды ошибок и их значения:

— 400 (Bad request) — синтаксическая ошибка в запросе.

2.9.13. Удаление из базы ошибок, возникших при миграции задачи

Запрос для удаления из базы ошибок, возникших при миграции задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}/migration_errors
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content).

2.10. Журналирование действий пользователя

С помощью запросов к API вы можете получать отчеты о действиях пользователей (в том числе по фильтру), дерево категорий действий, возможные значения дополнительного поля, описания всех дополнительных полей для всех приложений, а также регистрировать события пользователя.

В этом разделе

[Получение дерева категорий действий \(см. раздел 2.10.1\)](#)

[Получение возможного значения дополнительного поля \(см. раздел 2.10.2\)](#)

[Получение описания всех дополнительных полей для всех приложений \(см. раздел 2.10.3\)](#)

[Получение отчетов о действиях пользователей \(см. раздел 2.10.4\)](#)

[Получение отчетов о действиях пользователей по фильтру \(см. раздел 2.10.5\)](#)

[Регистрация событий пользователя \(см. раздел 2.10.6\)](#)

2.10.1. Получение дерева категорий действий

Запрос для получения дерева категорий действий.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>:3334/ptms/api/ual/v2/action_categories
```

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (Success). Ответ может содержать поля, описанные в таблице ниже.

Таблица 93. Поля ответа на запрос /api/ual/v2/action_categories

Поле	Тип данных	Описание
domains	Array	Домены объектов
Параметры доменов объектов		
id	String	Идентификатор домена объекта
name	String	Название домена объекта
applicationId	UUID	Идентификатор приложения
applicationName	String	Название приложения
types	Array	Тип объекта
Параметры типа объекта		
id	String	Идентификатор типа объекта
name	String	Название типа объекта

2.10.2. Получение возможного значения дополнительного поля

Запрос для получения возможного значения дополнительного поля.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>:3334/ptms/api/ual/v2/user_actions/applications/
{applicationId}/field_values/{key}
```

URL запроса содержит path-параметры: `applicationId` — идентификатор приложения и `key` — ключ поля.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (Success). Ответ может содержать поля, описанные в таблице ниже.

Таблица 94. Поля ответа на запрос /api/ual/v2/user_actions/applications/{applicationId}/field_values/{key}

Поле	Тип данных	Описание
type	Array	Тип поля

Поле	Тип данных	Описание
Параметры типа поля		
value	String	Значение
relatedKeyValue	String	Значение поля, к которому относится данное поле

2.10.3. Получение описания всех дополнительных полей для всех приложений

Запрос для получения описания всех дополнительных полей для всех приложений.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>:3334/ptms/api/ual/v2/user_actions/additional_fields

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (Success). Ответ может содержать поля, описанные в таблице ниже.

Таблица 95. Поля ответа на запрос /api/ual/v2/user_actions/applications/{applicationId}/field_values/{key}

Поле	Тип данных	Описание
type	Array	Тип поля
Параметры типа поля		
applicationId	String	Идентификатор приложения
fieldDefinitions	Array	Список описаний дополнительных полей
Параметры списка описаний дополнительных полей		
key	String	Ключ
localizedName	String	Локализованное название
isHidden	String	Признак сокрытия поля для отображения в списке дополнительных полей. Значение по умолчанию — false
useForFullTextSearch	Bool	Используются ли значения в полнотекстовом поиске
filter	Array	Описание доступного фильтра по данному полю

Поле	Тип данных	Описание
Параметры фильтра		
type	String	Дискриминатор типа фильтра
selectAllLocalizedName	String	Локализованное название строки «Выбрать все» при открытии фильтра

2.10.4. Получение отчетов о действиях пользователей

Запрос для получения отчетов о действиях пользователей.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>:3334/ptms/api/ual/v2/user_actions
```

Параметры строки запроса описаны в таблице ниже.

Таблица 96. Параметры строки запроса /api/ual/v2/user_actions

Параметр	Обязательный	Тип данных	Описание
userId	Да	String	Идентификатор пользователя для фильтрации
category	Да	String	Категория в формате domainId.typeId
orderby	Да	String	Сортировка, поддерживается только по времени. Пример — orderby=\"time desc\"
query	Да	String	Строка для поиска, введенная пользователем
limit	Да	Integer (int32)	Количество запрашиваемых действий
offset	Да	Integer (int32)	Смещение относительно начала списка действий
timeFrom	Нет	Long (int64)	Начало периода фильтрации (Unix time). Значение по умолчанию — 0
timeTo	Нет	Long (int64)	Конец периода фильтрации (Unix time). Значение по умолчанию — текущее время

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (Success). Ответ может содержать поля, описанные в таблице ниже.

Таблица 97. Поля ответа на запрос /api/ual/v2/user_actions

Поле	Тип данных	Описание
id	UUID	Идентификатор действия
beginDateTime	Date (date-time)	Время действия в формате ISO 8601
beginTime	Long (int64)	Время действия (Unix time)
beginUtcOffset	String	Сдвиг временной зоны действия относительно UTC
duration	String	Длительность
userId	String	Идентификатор пользователя
userLogin	String	Отображаемое имя пользователя
userDomain	String	Домен пользователя
objectDomainId	String	Идентификатор домена объекта
objectDomain	String	Домен объекта
objectTypeId	String	Идентификатор типа объекта
objectType	String	Тип объекта
objectId	String	Идентификатор объекта
objectDisplayName	String	Отображаемое имя объекта
applicationId	String	Идентификатор приложения
applicationName	String	Название приложения
code	String	Код действия
result	String	Результат действия
failReason	String	Причина ошибки
details	String	Подробное описание изменений
additionalFields	Object	Значение дополнительных полей событий

2.10.5. Получение отчетов о действиях пользователей по фильтру

Запрос для получения отчетов о действиях пользователей по фильтру.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>:3334/ptms/api/ual/v2/user_actions

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 98. Параметры в теле запроса /api/ual/v2/user_actions/

Параметр	Обязательный	Тип данных	Описание
filter	Да	UserActionFilter	Фильтр сбора событий
Параметры фильтра			
userIds	Нет	Array [String]	Массив идентификаторов пользователей
categories	Нет	Array [String]	Массив категорий в формате domainId.typeId
orderby	Нет	String	Сортировка, поддерживается только по времени. Пример – orderby=\"time desc\"
query	Нет	String	Строка для поиска, введенная пользователем
timeFrom	Нет	Date (date-time)	Дата начала периода фильтрации
timeTo	Нет	Date (date-time)	Дата конца периода фильтрации
additionalFields	Нет	Array	Фильтры по дополнительным полям
Параметры фильтра по дополнительному полю			
key	Да	String	Ключ дополнительного поля
value	Да	String	Значение для поиска

Параметры строки запроса описаны в таблице ниже.

Таблица 99. Параметры URL запроса /api/ual/v2/user_actions/

Параметр	Обязательный	Тип данных	Описание
limit	Да	Integer	Количество запрашиваемых действий
offset	Нет	Integer	Смещение относительно начала списка действий

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (Success). Ответ может содержать поля, описанные в таблице ниже.

Таблица 100. Поля ответа на запрос `/api/ual/v2/user_actions/`

Поле	Тип данных	Описание
<code>totalItems</code>	Integer	Общее количество событий, удовлетворяющих условиям запроса
<code>items</code>	Array	Список действий пользователя
Параметры списка действий пользователя		
<code>id</code>	UUID	Идентификатор действия
<code>beginDateTime</code>	Date (date-time)	Время действия в формате ISO 8601
<code>beginTime</code>	Long (int64)	Время действия (Unix time)
<code>beginUtcOffset</code>	String	Сдвиг временной зоны действия от UTC
<code>duration</code>	String	Длительность
<code>userId</code>	String	Идентификатор пользователя
<code>userLogin</code>	String	Отображаемое имя пользователя
<code>userDomain</code>	String	Домен пользователя
<code>objectDomainId</code>	String	Идентификатор домена объекта
<code>objectDomain</code>	String	Домен объекта
<code>objectTypeId</code>	String	Идентификатор типа объекта
<code>objectType</code>	String	Тип объекта
<code>objectId</code>	String	Идентификатор объекта
<code>objectDisplayName</code>	String	Отображаемое имя объекта
<code>applicationId</code>	String	Идентификатор приложения
<code>applicationName</code>	String	Название приложения
<code>code</code>	String	Код действия
<code>result</code>	String	Результат действия
<code>failReason</code>	String	Причина ошибки
<code>details</code>	String	Подробное описание изменений

Поле	Тип данных	Описание
<code>additionalFields</code>	Object	Значение дополнительных полей событий

2.10.6. Регистрация событий пользователя

Запрос для регистрации событий пользователя.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>:3334/ptms/api/ual/v2/user_events

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 101. Параметры в теле запроса /api/ual/v2/user_events

Параметр	Обязательный	Тип данных	Описание
<code>actions</code>	Да	Array	Список событий пользователя
Параметры списка событий пользователя			
<code>id</code>	Да	UUID	Идентификатор действия
<code>time</code>	Да	Date (date-time)	Время действия
<code>type</code>	Да	String	Тип события
<code>utcOffset</code>	Да	String	Сдвиг временной зоны действия от UTC

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created).

2.11. Работа с активами

С помощью запросов к API вы можете сохранить статус завершенной задачи, отменить задачу обновления паспорта активов, запускать задачи на удаление активов, изменение расположения активов в группах, обновление паспортов активов, а также получать состояние этих задач.

В этом разделе

[Запуск задачи на удаление активов \(см. раздел 2.11.1\)](#)

[Получение состояния задачи по удалению активов \(см. раздел 2.11.2\)](#)

[Запуск задачи на изменение расположения активов в группах \(см. раздел 2.11.3\)](#)

[Получение состояния задачи по обновлению расположения активов в группах \(см. раздел 2.11.4\)](#)

[Запуск задачи на обновление паспортов активов \(см. раздел 2.11.5\)](#)

[Получение состояния задачи по обновлению паспорта активов \(см. раздел 2.11.6\)](#)

[Отмена задачи обновления паспорта активов \(см. раздел 2.11.7\)](#)

[Сохранение статуса завершенной задачи \(см. раздел 2.11.8\)](#)

2.11.1. Запуск задачи на удаление активов

Запрос для запуска задачи на удаление активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/assets_processing/v1/asset_operations/removeAssets

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 102. Параметры в теле запроса /api/assets_processing/v1/asset_operations/removeAssets

Параметр	Обязательный	Тип данных	Описание
assetsIds	Да	String	Идентификаторы удаляемых активов

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created. Возвращаем token для получения состояния задачи). Ответ может содержать поля, описанные в таблице ниже.

Таблица 103. Поля ответа на запрос /api/assets_processing/v1/asset_operations/removeAssets

Поле	Тип данных	Описание
operationId	String	Идентификатор операции

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации;
- 503 (ServiceUnavailable) — ошибка доступа.

2.11.2. Получение состояния задачи по удалению активов

Запрос для получения состояния задачи по удалению активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/assets_processing/v1/asset_operations/removeAssets

Параметры строки запроса описаны в таблице ниже.

Таблица 104. Параметры строки запроса /api/assets_processing/v1/asset_operations/removeAssets

Параметр	Обязательный	Тип данных	Описание
operationId	Да	String	Идентификатор задачи

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 105. Поля ответа на запрос /api/assets_processing/v1/asset_operations/removeAssets

Поле	Тип данных	Описание
type	String	Тип
totalCount	Integer	Общее количество активов
succeedCount	Integer	Количество успешных удалений активов
failedCount	Integer	Количество неуспешных удалений активов

Поле	Тип данных	Описание
errorModel	Integer	Модель ошибок: — errors — ошибки: <ul style="list-style-type: none"> error — ошибка: type — тип ошибки; source — источник; sensitive — признак значимости ошибки; — message — сообщение; — code — код ошибки
Параметры источника (source)		
displayName	String	Отображаемое имя
hostName	String	Имя узла
ipAddresses	String	IP-адрес

Возможные коды ошибок и их значения:

- 202 (Accepted) — задача еще выполняется;
- 400 (Bad request) — синтаксическая ошибка в запросе.

2.11.3. Запуск задачи на изменение расположения активов в группах

Запрос для запуска задачи на изменение расположения активов в группах.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/assets_processing/v1/asset_operations/updateGroupEntries

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 106. Параметры в теле запроса /api/assets_processing/v1/asset_operations/updateGroupEntries

Параметр	Обязательный	Тип данных	Описание
assetsIds	Да	String	Идентификаторы активов
includeInGroups	Да	String	Включенные в группы активы

Параметр	Обязательный	Тип данных	Описание
excludeFromGroups	Да	String	Исключенные из групп активы

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created. Возвращаем token для получения состояния задачи). Ответ может содержать поля, описанные в таблице ниже.

Таблица 107. Поля ответа на запрос `/api/assets_processing/v1/asset_operations/updateGroupEntries`

Поле	Тип данных	Описание
operationId	String	Идентификатор операции

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации;
- 503 (ServiceUnavailable) — ошибка доступа.

2.11.4. Получение состояния задачи по обновлению расположения активов в группах

Запрос для получения состояния задачи по обновлению расположения активов в группах.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/assets_processing/v1/asset_operations/updateGroupEntries

Параметры строки запроса описаны в таблице ниже.

Таблица 108. Параметры строки запроса `/api/assets_processing/v1/asset_operations/updateGroupEntries`

Параметр	Обязательный	Тип данных	Описание
operationId	Да	String	Идентификатор задачи

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 109. Поля ответа на запрос /api/assets_processing/v1/asset_operations/updateGroupEntries

Поле	Тип данных	Описание
type	String	Тип
totalCount	String	Общее количество
succeedCount	String	Количество успешных задач
failedCount	String	Количество неуспешных задач
errorModel	Array	Данные ошибок
errorModel → errors	Array	Название ошибки: — error — ошибка: <ul style="list-style-type: none"> • type — тип ошибки; — source — источник ошибки: <ul style="list-style-type: none"> • displayName — отображаемое имя; • hostName — имя узла; • ipAddresses — IP-адрес; — sensitive — признак значимости ошибки
errorModel → message	String	Текст ошибки
errorModel → code	Integer	Код ошибки
updatedGroups	Array	Данные обновленной группы
updatedGroups → id	String	Идентификатор группы
updatedGroups → displayName	String	Название группы

Возможные коды ошибок и их значения:

- 202 (Accepted) — задача еще выполняется;
- 400 (Bad request) — синтаксическая ошибка в запросе.

2.11.5. Запуск задачи на обновление паспортов активов

Запрос для запуска задачи на обновление паспортов активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/assets_processing/v1/asset_operations/updatePassports

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 110. Параметры в теле запроса /api/assets_processing/v1/asset_operations/updatePassports

Параметр	Обязательный	Тип данных	Описание
assetsIds	Да	Array	Идентификаторы активов
changes	Да	Array	Изменение: <ul style="list-style-type: none"> ChangeCommand: <ul style="list-style-type: none"> changeType — тип изменения
operationId	Нет	UUID	Идентификатор операции

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created. Возвращаем token для получения состояния задачи). Ответ может содержать поля, описанные в таблице ниже.

Таблица 111. Поля ответа на запрос /api/assets_processing/v1/asset_operations/updatePassports

Поле	Тип данных	Описание
operationId	String	Идентификатор операции

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации;
- 503 (ServiceUnavailable) — ошибка доступа.

2.11.6. Получение состояния задачи по обновлению паспорта активов

Запрос для получения состояния задачи по обновлению паспорта активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/assets_processing/v1/asset_operations/updatePassports

Параметры строки запроса описаны в таблице ниже.

Таблица 112. Параметры строки запроса `/api/assets_processing/v1/asset_operations/updatePassports`

Параметр	Обязательный	Тип данных	Описание
<code>operationId</code>	Да	String	Идентификатор задачи

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 113. Поля ответа на запрос `/api/assets_processing/v1/asset_operations/updatePassports`

Поле	Тип данных	Описание
<code>type</code>	String	Тип
<code>totalCount</code>	Integer	Общее количество задач
<code>succeedCount</code>	Integer	Количество успешных задач
<code>failedCount</code>	Integer	Количество неуспешных задач
<code>errorModel</code>	Array	Данные ошибок
<code>errorModel → errors</code>	Array	Название ошибки: — <code>error</code> — ошибка: <ul style="list-style-type: none"> <code>type</code> — тип ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> <code>displayName</code> — отображаемое имя; <code>hostName</code> — имя узла; <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>errorModel → message</code>	String	Текст ошибки
<code>errorModel → code</code>	Integer	Код ошибки

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция еще выполняется;
- 400 (Bad request) — синтаксическая ошибка в запросе.

2.11.7. Отмена задачи обновления паспорта активов

Запрос для отмены задачи по обновлению паспорта активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/assets_processing/v1/asset_operations/updatePassports/{operationId}
```

URL запроса содержит path-параметр `operationId` — идентификатор задачи обновления активов.

Параметры в теле запроса отсутствуют.

Ответ на запрос

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция еще выполняется;
- 400 (Bad request) — синтаксическая ошибка в запросе.

2.11.8. Сохранение статуса завершенной задачи

Запрос для сохранения статуса завершенной задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/assets_processing/v1/asset_operations/completed/{operationType}/{operationId}
```

URL запроса содержит path-параметры `operationType` — тип задачи и `operationId` — идентификатор задачи.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content).

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе.

3. Отправка уведомлений через POST-запрос

В MaxPatrol VM предусмотрена возможность отправки уведомлений на внешний сервер через POST-запросы. Вы можете настроить отправку таких уведомлений для следующих случаев:

- изменение состава активов в системе или в выбранной группе;
- получение события, удовлетворяющего выбранному фильтру;
- запуск или остановка задачи сбора данных;
- выход параметров потока событий от источника из выбранной группы за пределы допустимых значений;
- появление уведомления о состоянии системы.

Для приема запросов на сервере должно быть установлено специализированное ПО.

Для отправки уведомлений через POST-запросы в веб-интерфейсе MaxPatrol VM в разделе **Система** на странице **Уведомления** нужно создать задачу на отправку уведомлений. В параметрах задачи нужно включить отправку уведомления через POST-запрос и указать URL внешнего сервера. Вы можете настроить отправку мгновенных уведомлений или уведомлений за период времени (см. Руководство администратора).

После создания задачи в случае выполнения условия отправки уведомления MaxPatrol VM автоматически собирает данные для уведомления и отправляет POST-запрос на внешний сервер. Специализированное ПО на внешнем сервере принимает POST-запрос и по ссылке в запросе получает из MaxPatrol VM данные уведомления.

Для приема POST-запросов и получения данных уведомления к ПО выдвигаются следующие требования:

- поддержка обмена данными по протоколам HTTP или HTTPS;
- прием POST-запросов без аутентификации;
- отправка подтверждения с кодом ответа 2xx при приеме POST-запроса;

Примечание. При отсутствии подтверждения получения запроса от внешнего сервера (с кодом ответа 2xx) POST-запрос будет отправляться повторно до 10 раз с интервалом в минуту.

- получение данных уведомления GET-запросом по ссылке в POST-запросе;
- при получении данных по протоколу HTTPS поддержка проверки подлинности сервера с помощью сертификата SSL.

В этом разделе

[Поля POST-запроса для уведомления \(см. раздел 3.1\)](#)

[Тестирование приема POST-запросов \(см. раздел 3.2\)](#)

3.1. Поля POST-запроса для уведомления

MaxPatrol VM отправляет POST-запрос в формате JSON. В зависимости от типа запрос может содержать следующие поля:

- `message_id` — идентификатор POST-запроса.
 - `notification_type` — тип POST-запроса: `test` — тестовый запрос, `event` — для мгновенного уведомления, `schedule` — для уведомления за период времени.
 - `notification_source` — условие создания уведомления: `AssetStateMetaTrigger` — изменение состава активов в системе; `EventsMetaTrigger` — получение события, удовлетворяющего выбранному фильтру; `EventsMonitoringControlsMetaTrigger` — выход параметров потока событий от источника из выбранной группы за пределы допустимых значений; `GroupContentMetaTrigger` — изменение состава активов в выбранной группе; `HealthMonitoringIssuesMetaTrigger` — появление уведомления о состоянии системы; `ScannerTaskNotificationsMetaTrigger` — запуск или остановка задачи на сбор данных.
 - `notification_uid` — идентификатор уведомления.
 - `notification_name` — название задачи MaxPatrol VM на отправку уведомления.
 - `uri` — ссылка на данные уведомления.
- Внимание!** Данные уведомления доступны по ссылке в течение 24 часов с момента создания уведомления.
- `schema_uri` — ссылка на схему данных уведомления в формате JSON.
 - `event_time_stamp` — дата и время создания мгновенного уведомления (UTC+0).
 - `time_interval` — для уведомления за период содержит поля `event_time_start` и `event_time_end` с датами и временем начала и конца периода (UTC+0).
 - `time_stamp` — дата и время создания POST-запроса (UTC+0).

Пример тестового POST-запроса:

```
{
  "notification_type": "test",
  "notification_source": "AssetStateMetaTrigger",
  "schema_uri": "https://siem-server.ru:8733/api/assets_triggers/v1/triggers_data/meta_triggers/AssetStateMetaTrigger/reactions/webhook_notification/schema",
  "time_stamp": "2019-04-15T11:20:49.7031727Z"
}
```

Пример POST-запроса для уведомления за период времени:

```
{
  "message_id": 1194,
  "notification_type": "schedule",
  "notification_source": "AssetStateMetaTrigger",
  "notification_uid": "110a1602-d980-0001-0000-000000000006",
}
```

```

    "notification_name": "1",
    "uri": "https://siem-server.ru:8733/api/assets_triggers/v1/triggers_data/
meta_triggers/AssetStateMetaTrigger/reactions/webhook_notification/
110a718e7300000100000000000000261",
    "schema_uri": "https://siem-server.ru:8733/api/assets_triggers/v1/triggers_data/
meta_triggers/AssetStateMetaTrigger/reactions/webhook_notification/schema",
    "event_time_stamp": "2019-04-12T09:50:21.4309566Z",
    "time_interval": {
        "event_time_start": "2019-04-12T09:50:21.4309566Z",
        "event_time_end": "2019-04-12T09:55:21.4309566Z"
    },
    "time_stamp": "2019-04-12T09:55:33.0096243Z"
}

```

3.2. Тестирование приема POST-запросов

В качестве примера ПО для приема POST-запросов и получения данных уведомления приводится сценарий на языке Python. В сценарии реализованы следующие возможности:

- При получении POST-запроса в интерфейс командной строки выводится его текст.
- При получении POST-запроса в MaxPatrol VM отправляется подтверждение с кодом ответа 2xx.
- По ссылке в POST-запросе из MaxPatrol VM скачиваются не более 10 страниц данных уведомления и выводятся в интерфейс командной строки.
- При необходимости по ссылке в POST-запросе из MaxPatrol VM скачивается схема данных уведомления и выводится в интерфейс командной строки.
- При получении данных из MaxPatrol VM по протоколу HTTPS для проверки подлинности сервера может быть предоставлен сертификат SSL.

Для выполнения сценария на сервере должны быть установлены операционные системы Windows 2012 R2 или Debian 9 и интерпретатор языка Python версии 3.6 или 3.7 с библиотеками Flask и Requests.

Примечание. Для отправки уведомлений на сервер в веб-интерфейсе MaxPatrol VM нужно создать задачу на отправку уведомлений через POST-запросы. В параметрах задачи нужно указать URL сервера в виде `http://<IP-адрес или FQDN>/handle`.

► Чтобы настроить прием POST-запросов и получение данных уведомления:

1. Создайте файл с расширением .py и скопируйте в него код сценария.
2. Если требуется, с помощью параметра `HOST` укажите сетевой интерфейс сервера.
3. Если требуется выводить в интерфейс командной строки схему данных уведомления, для параметра `SHOW_SCHEMA` укажите значение `True`.
4. Если требуется, с помощью параметров `HTTP_PORT` и `HTTPS_PORT` измените порты для приема запросов по протоколам HTTP и HTTPS.

5. Если требуется использовать протокол HTTPS, для параметра USE_HTTPS укажите значение True.

Внимание! Для проверки подлинности при получении сценарием данных из MaxPatrol VM по протоколу HTTPS нужно выпустить сертификат SSL. Файлы сертификата и ключа к нему нужно поместить в одну папку с файлом сценария. С помощью параметров сценария `certfile` и `keyfile` нужно указать имена файлов сертификата и ключа.

6. Сохраните файл.
7. Откройте интерфейс командной строки и запустите сценарий:
`python <Имя файла сценария>.py`

Прием POST-запросов настроен. Данные уведомлений будут выводиться в интерфейс командной строки.

Код сценария для приема POST-запросов

```
"""pip install flask requests"""
import requests
from pathlib import Path
from flask import Flask, request
HOST = '0.0.0.0'
HTTP_PORT = 10080
HTTPS_PORT = 10443
SHOW_SCHEMA = False
USE_HTTPS = False
cert_dir = Path(__file__).parent
certfile = cert_dir / '<Имя файла сертификата>.crt'
keyfile = cert_dir / '<Имя файла ключа>.pem'
app = Flask('ExampleTriggerService')
@app.route('/handle', methods=['POST'])
def handle():
    data = request.get_json()
    print('INCOMING DATA', data)
    if data['notification_type'] == 'test':
        return print('TEST') or 'test'
    if SHOW_SCHEMA:
        print('GET SCHEMA', data['schema_uri'])
        schema_data = requests.get(data['schema_uri'], verify=False).text
        print('SCHEMA DATA', schema_data)
    page_uri = data['uri']
    for page_num in range(10):
        if not page_uri:
            break
        print('GET PAGE', page_num, page_uri)
        page_data = requests.get(page_uri, verify=False).json()
```

```
        print('PAGE DATA', page_data)
        page_uri = page_data['dataSetInfo']['nextDataSetUri']
    return print('OK') or 'ok'
if __name__ == '__main__':
    if USE_HTTPS:
        import ssl
        context = ssl.SSLContext(ssl.PROTOCOL_TLS)
        # context = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2) # для Python версии 3.6
        context.load_cert_chain(str(certfile), str(keyfile))
        port = HTTPS_PORT
    else:
        context = None
        port = HTTP_PORT
    app.run(ssl_context=context, host=HOST, port=port, debug=True)
```

4. Обращение в службу технической поддержки

Техническая поддержка продукта включает в себя следующие услуги:

- решение вопросов эксплуатации продукта, помощь в использовании его функциональных возможностей;
- диагностику сбоев продукта, включая поиск причины сбоя и информирование клиента о найденных проблемах;
- разрешение проблем с продуктом, предоставление решений или возможностей обойти проблему с сохранением всей необходимой производительности;
- устранение ошибок в продукте (в рамках выпуска обновлений к продукту).

Вы можете получать техническую поддержку [на портале](#).

Этот раздел содержит информацию о способах и условиях получения технической поддержки.

В этом разделе

[Техническая поддержка на портале \(см. раздел 4.1\)](#)

[Время работы службы технической поддержки \(см. раздел 4.2\)](#)

[Как служба технической поддержки работает с запросами \(см. раздел 4.3\)](#)

4.1. Техническая поддержка на портале

[Портал](#) предоставляет вам возможность создавать запросы на техническую поддержку.

Вы можете создать учетную запись на портале, используя адреса электронной почты, расположенные на официальном домене вашей организации. Вы также можете указывать другие адреса электронной почты для учетной записи в качестве дополнительных. Для оперативной связи укажите в профиле учетной записи название вашей организации и контактный телефон.

[Портал](#) содержит статьи базы знаний, новости обновлений продуктов Positive Technologies, ответы на часто задаваемые вопросы пользователей. Для доступа к базе знаний и всем новостям нужно создать на портале учетную запись.

Техническая поддержка на портале предоставляется на русском и английском языках.

4.2. Время работы службы технической поддержки

На портале технической поддержки вы можете круглосуточно создавать и обновлять запросы, читать новости продуктов и пользоваться базой знаний.

4.3. Как служба технической поддержки работает с запросами

При получении вашего запроса специалист службы технической поддержки классифицирует его (присваивает запросу тип и уровень значимости) и выполняет дальнейшие шаги по выполнению запроса.

В этом разделе

[Предоставление информации для технической поддержки \(см. раздел 4.3.1\)](#)

[Типы запросов \(см. раздел 4.3.2\)](#)

[Время реакции и приоритизация запросов \(см. раздел 4.3.3\)](#)

[Выполнение работ по запросу \(см. раздел 4.3.4\)](#)

4.3.1. Предоставление информации для технической поддержки

При обращении за технической поддержкой по первому требованию специалиста Positive Technologies нужно предоставить:

- номер лицензии на использование продукта;
- файлы журналов и другие наборы диагностических данных, хранящихся в продукте;
- снимки экрана;
- результаты выполнения рекомендаций специалиста технической поддержки;
- каналы для удаленного доступа к продукту (по взаимному согласованию оптимального канала диагностики).

Positive Technologies не несет обязательств по оказанию технической поддержки в случае отказа предоставить указанную выше информацию.

Если информация по обращению не предоставлена в течение значительного времени (от двух недель с момента последней активности), специалист технической поддержки имеет право считать ваше обращение неактуальным и, уведомив вас, закрыть запрос.

4.3.2. Типы запросов

Специалист технической поддержки относит ваш запрос к одному из следующих типов.

Вопросы по установке, повторной установке и предстартовой настройке продукта

Подразумевается помощь в подготовке продукта к работе, ответы на вопросы на данном этапе эксплуатации продукта. Техническая поддержка по этим вопросам доступна в течение 30 дней с момента активации продукта.

Вопросы по администрированию и настройке продукта

Включают в себя вопросы, возникающие в процессе эксплуатации продукта, рекомендации по оптимизации и настройке параметров продукта.

Восстановление работоспособности продукта

В случае критического сбоя и потери доступа к основной функциональности продукта специалист Positive Technologies оказывает помощь в восстановлении работоспособности продукта. Восстановление заключается либо в помощи по установке продукта заново с потенциальной потерей накопленных до сбоя данных, либо в откате продукта на доступную резервную копию (резервное копирование должно быть настроено заблаговременно). Positive Technologies не несет ответственность за потерю данных в случае неверно настроенного резервного копирования.

Обновление продукта

Positive Technologies предоставляет пакеты обновления в течение срока обновления, указанного в лицензии на продукт.

Positive Technologies не несет ответственности за проблемы, возникшие при нарушении регламентированного процесса обновления.

Устранение дефектов продукта

Если по результатам диагностики обнаружен дефект продукта, Positive Technologies обязуется предпринять разумные усилия по предоставлению обходного решения (если возможно), а также включить исправление дефекта в ближайшие возможные обновления продукта.

4.3.3. Время реакции и приоритизация запросов

Время реакции на запрос рассчитывается с момента получения запроса до первичного ответа специалиста технической поддержки с уведомлением о взятии запроса в работу.

Время обработки запроса рассчитывается с момента отправки уведомления о взятии вашего запроса в работу до предоставления описания дальнейших шагов по устранению проблемы либо классификации вопроса, указанного в запросе, как дефекта ПО и передачи запроса ответственным лицам для исправления дефекта.

Время реакции и время обработки зависят от указанного вами уровня значимости запроса (см. таблицу 114).

Специалист службы технической поддержки оставляет за собой право переопределять уровень значимости запроса по приведенным ниже критериям. Указанные сроки являются целевыми и подразумевают стремление и разумные усилия исполнителя для их соблюдения, но возможны отклонения от данных сроков по объективным причинам.

Таблица 114. Время реакции на запрос и время его обработки

Уровень значимости запроса	Критерии значимости запроса	Время реакции на запрос	Время обработки запроса
Критический	Аварийные сбои, полностью препятствующие штатной работе продукта (исключая первоначальную установку) либо оказывающие критическое влияние на бизнес	До 4 часов	Не ограничено
Высокий	Сбои, затрагивающие часть функциональности продукта и проявляющиеся в любых условиях эксплуатации либо оказывающие значительное влияние на бизнес	До 24 часов	Не ограничено
Обычный	Сбои, проявляющиеся в специфических условиях эксплуатации продукта либо не оказывающие значительного влияния на бизнес	До 24 часов	Не ограничено
Низкий	Вопросы информационного характера либо сбои, не влияющие на эксплуатацию продукта	До 24 часов	Не ограничено

Указанные часы относятся только к рабочему времени специалистов технической поддержки (времени обработки запроса).

4.3.4. Выполнение работ по запросу

По мере выполнения работ по вашему запросу специалист технической поддержки сообщает вам:

- о диагностике проблемы и ее результатах;
- о поиске решения или возможности обойти причины возникновения проблемы;
- о планировании и выпуске обновления продукта (если требуется для устранения проблемы).

Если по итогам обработки запроса необходимо внести изменения в продукт, Positive Technologies включает работы по исправлению в ближайшее возможное плановое обновление продукта (в зависимости от сложности изменений).

Работы по запросу считаются выполненными, если:

- предоставлено решение или возможность обойти проблему, не влияющая на производительность и критически важную функцию продукта;
- диагностирован дефект продукта, собрана техническая информация о дефекте и условиях его воспроизведения; исправление дефекта запланировано к выходу в рамках планового обновления продукта;
- проблема вызвана программными продуктами или оборудованием сторонних производителей, не подпадающих под гарантийные обязательства по продукту;
- проблема классифицирована как неподдерживаемая.



Positive Technologies — лидер рынка результативной кибербезопасности. Компания является ведущим разработчиком продуктов, решений и сервисов, позволяющих выявлять и предотвращать кибератаки до того, как они причинят неприемлемый ущерб бизнесу и целым отраслям экономики. Наши технологии используют более 3300 организаций по всему миру, в том числе 80% компаний из рейтинга «Эксперт-400». Positive Technologies — первая и единственная компания из сферы кибербезопасности на Московской бирже (MOEX: POSI), у нее более 170 тысяч акционеров.