

ИНФОРМАЦИОННАЯ СИСТЕМА «ЕДИНАЯ ГОСУДАРСТВЕННАЯ ПЛАТФОРМА
СБОРА ДАННЫХ, ПРОМЫШЛЕННОГО ИНТЕРНЕТА ВЕЩЕЙ И ИНСТРУМЕНТОВ
АНАЛИЗА ОБЪЕКТИВНЫХ ДАННЫХ О НАБЛЮДАЕМЫХ ОБЪЕКТАХ В СОСТАВЕ
ПЛАТФОРМЫ ИСПОЛНЕНИЯ ГОСУДАРСТВЕННЫХ ФУНКЦИЙ»

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ПО РАЗРАБОТКЕ АГЕНТА

ИНФОРМАЦИОННОЙ СИСТЕМЫ ПЛАТФОРМЫ СБОРА ДАННЫХ

(Основание: Государственный контракт №0173100007519000136_144316 от 07.11.2019 г.)

2020

Ине. № подл.	Подпись и дата	Взам. инв. №	Ине. № дубл.	Подпись и дата

ОГЛАВЛЕНИЕ

1.	ОБЩИЕ СВЕДЕНИЯ.....	3
1.1.	Термины и сокращения.....	3
2.	Разработка Агента.....	4
2.1.	Вводная часть.....	4
2.2.	Общая архитектура.....	5
2.3.	Порядок информационного обмена HTTP Агента с Платформой.....	6
2.4.	Настройки и данные в платформе.....	15
2.5.	Варианты реализации HTTP Агентов.....	15
2.5.1.	Пример Агента на базе Node-RED.....	15
2.5.2.	Пример HTTP Агента LoRaWAN устройств на базе Java приложения.....	16
2.5.2.1.	Архитектура Агента LoRaWAN устройств на базе Java приложения.....	16
2.5.2.2.	Реализация приложения Агента.....	18
2.5.2.3.	Структура проекта, сборка и зависимости.....	18
2.5.2.4.	Получение данных с датчиков.....	18
2.5.2.5.	Преобразование данных.....	19
2.5.2.6.	Получение метаинформации.....	19
2.5.2.7.	Отправка данных в платформу.....	19
2.5.2.8.	Настройки агента.....	20
3.	Описание интерфейса взаимодействия с Платформой для работы Агентов.....	21
3.1.	Описание HTTP REST API по стандарту OpenAPI в формате JSON.....	21
3.2.	Описание HTTP REST API по стандарту OpenAPI в формате YAML.....	26
4.	Листинг Агента Node-RED.....	31

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата	
Изм.	Лист	№ документа	Подпись	Дата	Лист
					2

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Термины и сокращения

Сокращение/Термин	Наименование/Определение
Агент	Программный объект, проводящий сбор данных по заданным алгоритмам и их передачу в центр сбора данных, настраиваемый локально или удаленно из центра сбора данных и коммуницирующий с другими объектами или центром сбора данных
Браузер, веб-браузер	Программное обеспечение на компьютере или мобильном устройстве пользователя, предназначенное для просмотра веб-страниц, содержания веб-документов, управления веб-приложениями, размещенными в Интернете
Веб-приложение	Клиент-серверное приложение, в котором клиентом выступает веб-браузер, а сервером — веб-сервер
Веб-сервер	Сервер (программное обеспечение), принимающий запросы от клиентов, обычно веб-браузеров , и выдающий им ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными
Клиент	В контексте сетевой архитектуры «клиент-сервер»: программное обеспечение, являющееся заказчиком услуг сервера (поставщика услуг)
Клиент-сервер	Вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами
ИС ПСД, Система	информационная система «Единая государственная платформа сбора данных, промышленного интернета вещей и инструментов анализа объективных данных о наблюдаемых объектах»
Платформа	Набор базовых сервисов, включающий в себя: подсистему работы с источниками данных, подсистему работы с данными и подсистему хранения данных
Сервер	В контексте сетевой архитектуры «клиент-сервер»: программное обеспечение, являющееся поставщиком услуг заказчикам (клиентам)
СУБД	Система управления базами данных
API	Application Programming Interface (интерфейс программирования приложений) — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах
JSON	(англ. JavaScript Object Notation) — текстовый формат обмена данными, основанный на JavaScript
IoT	(англ. Internet of Things) — Интернет вещей концепция — вычислительной сети физических предметов («вещей»), оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой
LoRaWAN	Открытый стандарт беспроводной энергоэффективной технологии передачи небольших по объёму данных на дальние расстояния на базе радиопрототокола LoRa, разработанная для распределённых сетей телеметрии, межмашинного взаимодействия и Интернета вещей

Име. № подл.	Подпись и дата
Взам. ине. №	Име. № дубл.
Подпись и дата	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					3

2.2. Общая архитектура

Архитектура прикладных сервисов приведена ниже (Рисунок 1).

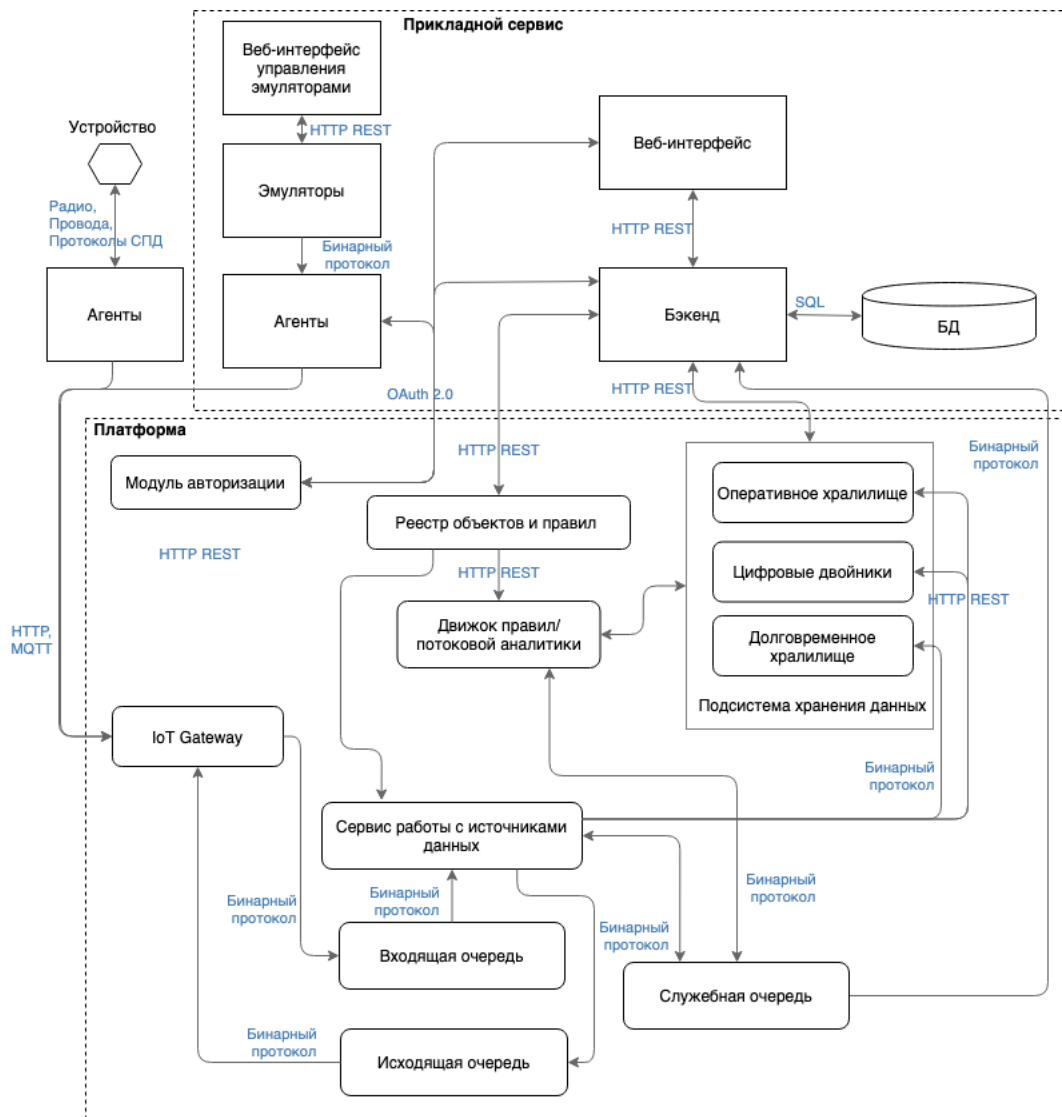


Рисунок 1 – Архитектура прикладных сервисов

Агенты платформы создаются на базе специализированного SDK, предоставляемого платформой. По сути, Агент представляет собой специализированное ПО, которое с одной стороны создается для получения данных с устройств (эта часть разрабатывается под конкретное устройство) и с другой стороны для передачи данных в Платформу (эта часть входит в состав SDK).

Агент предназначен для подключения внешнего устройства к Платформе и обеспечивает функции сбора данных и управления устройством, а также передачу собранных данных в Платформу и управляющих команд от Платформы к устройству.

Агент может быть расположен как непосредственно на устройстве, так и на сервере в периметре Системы или вне его.

Для авторизации Агента в Платформе используется токен доступа, который генерируется Платформой при создании экземпляра Агента в ней.

Име. № подл.	Подпись и дата
Име. № дубл.	Подпись и дата
Взам. име. №	Подпись и дата
Име. № подл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					5

Для агентов, расположенных за пределами ИС ПСД, рекомендуется написание логики Агента для буферизации принятых от оборудования данных на период отсутствия связи с Платформой с привязкой ко времени регистрации этих данных. Также для таких Агентов рекомендуется создавать встроенный или внешний модуль управления подключенностью, осуществляющий повторное подключение к сети в случае разрыва соединения.

Кроме того, что Агент передает данные с датчиков устройств, он передает статус датчиков и устройства.

2.3. Порядок информационного обмена HTTP Агента с Платформой

Порядок информационного обмена Агента с Платформой включает следующие шаги:

2.3.1. Авторизация Агента в Подсистеме обеспечения информационной безопасности (Модуле авторизации)

На этом шаге выполняется HTTP POST запрос на авторизацию и получение токена сессии под пользователем типа Агент. Пример такого запроса:

```
curl --location --request POST
'http://10.20.22.30/public/auth/authenticator/api/internalauth/auth' \
--header 'Authorization: Basic MTUzM0Bpb3QuNDoxMjM0NTY3OA=='
```

где 10.20.22.30 – адрес Модуля ИБ, в качестве авторизационного параметра при Basic авторизации передается строка в формате Логин:Пароль в кодировке base-64.

В ответ возвращается HTTP заголовок с полем Set-Cookie, содержащий HTTP Cookie, который необходимо использовать в HTTP заголовке Cookie всех последующих запросов.

2.3.2. Запрос на получение конфигурации Агента

Конфигурация агента – это json объект, который описывает как сам агент (включая все поддерево тэгов агента), так и все подключенные к агенту устройства (также включая все поддерева тэгов указанных устройств) и содержит настройки того, как агент должен подключаться и собирать данные с устройств.

Конфигурация агента имеет уникальную версию, которая монотонно возрастает при изменении метаданных агента, либо одного из подключенных к нему устройств.

При первом запуске агент автоматически получает последнюю версию конфигурации из модуля Gateway платформы.

В дальнейшем обновление конфигурации агента активируется с помощью специальной команды.

После получения и применения конфигурации агент рапортует в платформу текущую версию конфигурации и время обновления в специальные системные подтэги типа state \$state/\$config/\$version \$state/\$config/\$updated_at.

На основе разницы между версией конфигурации агента в метаданных и версией, которую рапортует агент (хранится в цифровых двойниках) платформа и пользователь могут понимать находится ли на агенте последняя конфигурация или требуется отправка команды на обновление.

Пример HTTP GET запроса получение конфигурации агента:

```
curl --location --request GET 'http://10.20.22.30/public/api/http-gateway/api/v1/agents/config' \
--header 'Cookie:
SESSION_TOKEN="eyJzZXNzaW9uSWQiOiJlM2U5Y2JjNC01MjE1LTRiZDUtYjUxZWZlMzQ1NWEzYWYzYzQiLCJjdXJyZW50VXNlcklkIjoimjI4YjM3MmEtNTEzYS00YTQ1LWE2Y2MtY2Y2MTM4YmQ4YjRlIn0=";Version=1;Domain=;Path=/;HttpOnly' \
--header 'Accept: application/json' \
--header 'X-Agent-Token: b3b4424c-e6d2-4e85-9604-99f8552c6ad7' \
--header 'Authorization: Basic MTUzM0Bpb3QuNDoxMjM0NTY3OA=='
```

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата					Лист
									6
Изм.	Лист	№ документа	Подпись	Дата					

В этом запросе необходимо использовать в HTTP заголовке Cookie значение, полученное в авторизационном запросе. В HTTP заголовке X-Agent-Token необходимо передавать значение токена Агента, присвоенного Агенту Платформой. В ответе возвращается JSON объект с описанием устройств и датчиков внутри каждого устройства в виде дерева тэгов, например:

```

{
  "agent": { //Конфигурация агента
    "devices": [ //Перечень устройств агента
      {
        "driver": {"id": 2132, "name": "gen_embedded_1"}, // Драйвер
        "driver_config": {}, // Конфигурация драйвера
        "id": 3477, //Идентификатор устройства
        "name": "post_01", //Название устройства
        "tag": { //Тэг устройства
          "attrs": { //Атрибуты устройства - задаются пользователем
            //можно использовать для идентификации устройств в Агенте (в этом примере для
            //этого используется ftp_prefix)
            "address": "ул.Мамина 19Б",
            "device_config": {
              "data_transmit_frequency": "600",
              "sensor_request_frequency": "600"
            },
            "ftp_prefix": "01",
            "lat": "55.176095",
            "lng": "61.482532"
          },
          "children": [ //датчики устройства и статусы
            {
              "attrs": {},
              "children": [
                {
                  "attrs": {},
                  "children": [
                    { // дочерний элемент для статуса определяется полем name =
                    "$status" - задается системой
                      "attrs": {},
                      "children": null,
                      "id": 114258,
                      "name": "$history",
                      "properties": {
                        "autosave": {
                          "digital_twins": false,
                          "operational_storage": false
                        },
                        "units": "",
                        "value_type": "string"
                      },
                      "type": {
                        "id": 1,
                        "name": "event"
                      }
                    }
                  ]
                },
                {
                  "id": 114232,
                  "name": "$status",
                  "properties": {
                    "autosave": {
                      "digital_twins": true
                    },
                    "readonly": true,
                    "units": "",
                    "value_type": "string"
                  },
                  "type": {
                    "id": 2,
                    "name": "state"
                  }
                }
              ]
            }
          ]
        }
      }
    ]
  }
}

```

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата
--------------	----------------	--------------	--------------	----------------

Изм.	Лист	№ документа	Подпись	Дата	Лист
					7

```

    },
    // ... аналогичные по структуре системные теги $vendor,
    $version, $firmware, $updated_at, $last_seen_at, $is_working не приведены в
    примере
  },
  { //тег датчика
    "attrs": {},
    "children": [
      {
        "attrs": {},
        "children": [
          {
            "attrs": {},
            "children": null,
            "id": 114266,
            "name": "$history",
            "properties": {
              "autosave": {
                "digital_twins": false,
                "operational_storage": false
              },
              "units": "",
              "value_type": "boolean"
            },
            "type": {
              "id": 1,
              "name": "event"
            }
          }
        ],
        "id": 114251, // id тэга состояния датчика
        "name": "$is_working", // название тэга состояния датчика
        "properties": {
          "autosave": {
            "digital_twins": true
          },
          "readonly": true,
          "units": "",
          "value_type": "boolean"
        },
        "type": {
          "id": 2,
          "name": "state"
        }
      }
    ],
    "id": 114242, //id датчика
    "name": "CH2O", //название датчика
    "properties": {
      "autosave": {
        "long_term_storage": true,
        "operational_storage": true
      },
      "units": "мг/м³", //единицы измерения датчика
      "value_type": "float" //тип значения датчика
    },
    "type": {
      "id": 1,
      "name": "event" //тип тэга event = датчик
    }
  }
],
"id": 114005, //id тэга статуса всего устройства
"name": "$state",
"properties": {},
"type": {
  "id": 3,
  "name": "node"
}
}

```

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					8


```

    }
  ],
  "id": 114004, // id тэга Агента
  "name": "FTPAgent", //название агента
  "properties": {
    "agent_id": 2141 //ID агента
  },
  "type": {
    "id": 5,
    "name": "agent"
  }
}
}
}
}

```

По полям приведенной выше конфигурации Агент определяет тэги, в которые нужно опривить значения. Агент может отправлять значения только в тэги подчиненных этому Агенту устройств. В данном примере Агент определяет датчик по его названию и отправляет значение по идентификатору тэга этого датчика.

2.3.3. Запрос на отправку данных в Платформу

Агент отправляет полученные от устройств данные в платформу на основе полученной конфигурации (маппинг показаний устройств на соответствующие тэги типа event и state).

Также агент отправляет в платформу данные о своём статусе, а также о статусе всех подключенных устройств в тэги \$state/\$status.

В случае обрыва связи с платформой агент буферизирует получаемые от устройств данные для дальнейшей отправки в платформу после восстановления связи.

Пример запроса на отправку данных в Платформу, HTTP POST запрос:

```

curl --location --request POST 'http://10.20.22.30/public/api/http-gateway/api/v1/events?X-Agent-Token=b3b4424c-e6d2-4e85-9604-99f8552c6ad7&Content-Type=application/json' \
--header 'Content-Type: text/plain' \
--header 'Cookie: SESSION_TOKEN="eyJzZXNzaW9uSWQiOiJmMDFjNjE2YS0zNWNlLTQ2ZDYtYjIyMC1jMjg0Y2RlMjk5NWEiLCJjdxJyZW50VXNlcklkIjoIn2VmZjY5YzQtMwVlOC00ODYyLWI3ZGI3MTNkZDNmYTBiOGQwIn0="' \
--data-raw '{
  "tags": [
    {
      "id": 374046,
      "timestamp": 1590998326436000,
      "value": 0.015
    }
  ]
}'

```

Здесь в поле timestamp передается время замера в микросекундах. Передавать можно и нужно как сами значения измерений датчиков, так и состояния датчиков и устройства. Запрос можно выполнять в пакетном режиме, передавая не более 20 значений тэгов в одном запросе.

2.3.4. Выполнение команд

Агент, реализующий двунаправленную связь между устройством и платформой, должен поддерживать получение команд из платформы.

Команда — это JSON объект с набором новых значений тэгов типа state и уникальным идентификатором, например, команда на обновление конфигурации агента будет выглядеть как agent_tag/\$state/\$config/\$version = new_version.

Име. № подл.	Подпись и дата
Взам. име. №	Име. № дубл.
Подпись и дата	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					9

Команда может быть адресована как агенту, так и любому подключенному к агенту устройству, адресат команды вычисляется по переданным в команде тэгам (в чьем поддереве тэгов находятся переданные тэги).

Одна команда может быть адресована только одному агенту/устройству.

Команда может содержать произвольный набор новых значений для любых тэгов типа state с полем properties.readonly = false из поддерева тэгов целевого агента/устройства.

Команда может иметь следующий набор статусов:

– платформенные статусы:

- new - команда была только что создана в Digital Twins;
- sending - команда была взята в работу в Peons;
- sent - команда была отправлена на передачу агенту в HTTP Gateway или MQTT Gateway;
- failed - отправка команды по каким-то причинам не удалась (подробно описывается в поле reason).

– статусы, которые обязан рапортовать агент:

- received - команда была получена агентом;
- done - команда успешно завершена;
- skipped - команда была по каким-то причинам пропущена агентом (например, при получении сразу нескольких команд. При обновлении конфигурации агент может выполнить только последнюю команду, а остальные перевести в skipped заполнив, поле reason);
- failed - агенту не удалось выполнить команду, подробные причины должны быть указаны в поле reason.

По результатам выполнения команды агент должен отчитаться о статусе команды в платформу через специальное API, послав статус команды done или failed с описанием возникшей ошибки в поле reason.

Данный протокол описывает лишь транспорт для передачи команд от платформы к агенту, реализация выполнения этих команд остаётся на усмотрение агента.

2.3.5. Отправка логов

Протокол поддерживает отправку агентом произвольных логов в виде произвольной строки.

Платформа может ограничивать количество принимаемых логов с агента в единицу времени.

Логи могут содержать как информацию о функционировании самого агента, так и информацию о функционировании подключенных устройств.

Полученные платформой от агента логи доступны на просмотр в компоненте платформы User Logs.

Име. № подл.	
Подпись и дата	
Взам. име. №	
Име. № дубл.	
Подпись и дата	

Изм.	Лист	№ документа	Подпись	Дата		<i>Лист</i>
						10

2.4. Использование API Агента платформы

2.4.1. Для HTTP(S) агентов

При подключении к платформе только по протоколу HTTP(S) платформа не может напрямую пересылать команды агенту, поэтому агент обязан периодически опрашивать API платформу для получения команд на выполнение.

Точкой подключение агента к платформе является компонент платформы HTTP Gateway.

При обращении к HTTP API агент должен в каждом запросе отправлять HTTP заголовок X-Agent-Token с уникальным токеном агента из поля agent.token, которое можно посмотреть в Registry.

- GET /v1/agents/config - эндпоинт для получения конфигурации, вызывается с параметром version=latest при запуске агента для получения начальной конфигурации, а также с указанной в команде версией для обновления конфигурации по пришедшей команде;
- POST /v1/events - эндпоинт для отправки данных в платформу, принимает массив значений вида tag_id=value;
- POST /v1/logs - эндпоинт для отправки логов в платформу, принимает массив значений вида timestamp=timestamp_with_microseconds, msg=string;
- GET /v1/commands - эндпоинт для получения команд на выполнение. Данный эндпоинт необходимо периодически вызывать для получения новых команд на выполнение. Данный эндпоинт будет всегда возвращать список всех активных команд;
- PATCH /v1/commands/{command_id}/status - эндпоинт для перевода взятой команды в конечный статус, принимает объект вида status=received/done/failed/skipped, reason=optional_string_message_for_failed_or_skipped_status;
- GET /v1/devices/{device_id}/config/{version} - эндпоинт для получения конфигурации, которую необходимо физически применить на подключенное устройство в рамках выполнения команды (подробнее смотри раздел с командами).

2.4.2. Для HTTP(S) + MQTT агентов

При подключении агента к платформе через связку протоколов HTTP(S) + MQTT платформа может напрямую взаимодействовать с агентом и отправлять ему команды без необходимости постоянного опроса HTTP API со стороны агента.

Точкой подключение агента к платформе является компонент платформы HTTP Gateway для HTTP(S) и компонент платформы MQTT Gateway для MQTT.

2.4.2.1. HTTP API

В протоколе на базе MQTT отсутствует поддержка следующих операций:

- GET /v1/agents/config
- GET /v1/devices/{device_id}/config/{version}

Име. № подл.	Подпись и дата	Име. № дубл.	Подпись и дата
	Взам. име. №		

Изм.	Лист	№ документа	Подпись	Дата	Лист
					11

2.4.2.2. MQTT API

2.4.2.2.1. Авторизация

При подключении к MQTT брокеру клиент должен предоставить уникальный токен в поле username.

Уникальный токен для каждого клиента генерируется отдельным компонентом платформы.

На основе предоставленного токена брокер получает настройки клиента из Registry.

2.4.2.2.2. Отправка событий

Агент должен отправлять данные в топик `iot/event/fmt/json` с QOS=1 в следующем формате:

```
{
  "tags": [
    {
      "id": ..., // id конечного тэга, который представляет собой датчик
      "value": ..., // значение
      "timestamp": ... // timestamp in microseconds
    },
    ...
  ]
}
```

Например, чтобы послать сообщение для тэга с `id=10` со значением '100', необходимо опубликовать сообщение:

```
{
  "tags": [
    {
      "id": 10,
      "value": 100,
      "timestamp": 1
    }
  ]
}
```

2.4.2.2.3. Команды

Для получения команд от платформы агент должен быть подписан на топик, вида `iot/cmd/agent/+ /fmt/json`, вместо одноуровневого wildcard + должен стоять уникальный id клиента.

Сообщение с командой публикуется с флагом `retain=1`, что позволяет клиенту гарантированно получить последнюю команду.

Например, чтобы подписаться на команды для агента с `id=1`, необходимо подписаться на топик `iot/cmd/agent/1 /fmt/json`

Команды имеют следующий формат:

```
{
  "id": "...", // command id
  "tags": [
    {
      "id": ..., // tag to change
      "value": ... // new tag value
    },
    ...
  ],
  "timestamp": ... // timestamp in microseconds
}
```

Подпись и дата
Инв. № дубл.
Взам. инв. №
Подпись и дата
Инв. № подл.

Изм.	Лист	№ документа	Подпись	Дата	Лист
					12

Агент должен сообщать статусы команд в отдельный топик `iot/cmd/status/fmt/json` с QOS=1 сообщениями вида:

```
{
  "id": "...", // command id
  "status": "...", // received | failed | done
  "reason": "...", // optional field for failed status
  "timestamp": ... // timestamp in microseconds
}
```

Поддерживаемые статусы команд:

- `received` - команда получена агентом/устройством;
- `failed` - агент/устройство не смогли применить команду, дополнительно в поле `reason` должна быть указан причина;
- `done` - агент/устройство успешно применили команду.

2.4.2.2.4. Логирование

Для отправки произвольных логов в платформу агент должен использовать топик `iot/log/fmt/json` с QOS=1 в формате:

```
[
  {
    "msg": "...",
    "timestamp": ... // timestamp in microseconds
  }
]
```

2.4.2.3. Общие команды

2.4.2.3.1. Обновление конфигурации агента

Данная команда обязательна к реализации всеми агентами.

Данная команда всегда содержит набор значений для следующих тэгов агента:

- `$state/$config/$version`.

По получении данной команды агент обязан получить конфигурацию указанной в команде версии из соответствующего эндпоинта компонента HTTP Gateway.

По успешному выполнению команды агент обязан отправить в платформу новые значения для следующих тэгов агента:

- `$state/$config/$version`;
- `$state/$config/$updated_at`.

2.4.2.3.2. Обновление конфигурации на устройстве

Данная команда необязательна к реализации агентом и может быть применена только к устройству, которое поддерживает конфигурирование агентом.

Признаком того, что устройство может быть сконфигурировано агентом является флаг `properties.readonly` выставленный в `false` у тэга устройства `$state/$config/$version`.

Данная команда всегда содержит набор значений для следующих тэгов устройства:

- `$state/$config/$version`.

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					13

По получении данной команды агент обязан получить конфигурацию для устройства указанной в команде версии из соответствующего эндпоинта компонента HTTP Gateway.

Реализация обновления конфигурации устройства полностью находится в зоне ответственности агента.

По успешному выполнению команды агент обязан отправить в платформу новые значения для следующих тэгов устройства:

- \$state/\$config/\$version;
- \$state/\$config/\$updated_at.

2.4.2.3.3. Обновление прошивки на агенте

Данная команда необязательна к реализации агентом и может быть применена только к агенту, который поддерживает автоматическое обновление прошивки.

Признаком того, что агент может обновлять свою прошивку является флаг `properties.readonly` выставленный в `false` у тэга агента `$state/$firmware/$source`.

Данная команда всегда содержит набор значений для следующих тэгов агента:

- `$state/$firmware/$source` - источник, например, `http url` для скачивания новой прошивки.

А также команда может содержать следующие необязательные тэги:

- `$state/$firmware/$vendor`;
- `$state/$firmware/$version`.

Реализация обновления прошивки полностью находится в зоне ответственности агента.

По успешному выполнению команды агент обязан отправить в платформу новые значения для следующих тэгов агента:

- `$state/$firmware/$source`;
- `$state/$firmware/$updated_at`.

А также агент может заполнить следующие тэги в зависимости от содержания команды / особенностей реализации:

- `$state/$firmware/$vendor`;
- `$state/$firmware/$version`.

2.4.2.3.4. Обновление прошивки на устройстве

Данная команда необязательна к реализации агентом и может быть применена только к устройству, которое поддерживает автоматическое обновление прошивки.

Признаком того, что агент может обновлять прошивку у устройства является флаг `properties.readonly` выставленный в `false` у тэга устройства `$state/$firmware/$source`.

Данная команда всегда содержит набор значений для следующих тэгов устройства:

- `$state/$firmware/$source` - источник, например, `http url` для скачивания новой прошивки.

А также команда может содержать следующие необязательные тэги:

- `$state/$firmware/$vendor`;
- `$state/$firmware/$version`.

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата					Лист
									14
					Изм.	Лист	№ документа	Подпись	Дата

Реализация обновления прошивки на устройстве полностью находится в зоне ответственности агента.

По успешному выполнению команды агент обязан отправить в платформу новые значения для следующих тэгов устройства:

- \$state/\$firmware/\$source;
- \$state/\$firmware/\$updated_at.

А также агент может заполнить следующие тэги устройства в зависимости от содержания команды / особенностей реализации:

- \$state/\$firmware/\$vendor;
- \$state/\$firmware/\$version.

2.5. Настройки и данные в платформе

Для работы агента требуется завести на стороне платформы соответствующий экземпляр Агента на базе предварительно созданного шаблона Агента. В экземпляре Агента будет храниться поле token, значение которого и нужно подставлять в HTTP заголовок X-Agent-Token в HTTP запросах. Для работы устройств необходимо создать устройства указав для них созданного Агента.

2.6. Варианты реализации HTTP Агентов

Приведенные ниже варианты реализации демонстрируют общий подход и вариативность средств и могут быть использованы для создания собственного Агента по образцу приведенных.

2.6.1. Пример Агента на базе Node-RED

Для создания Агента можно использовать различные инструменты, например среду конфигурирования потоков данных Node-RED (Пример приведен на рисунке 2).

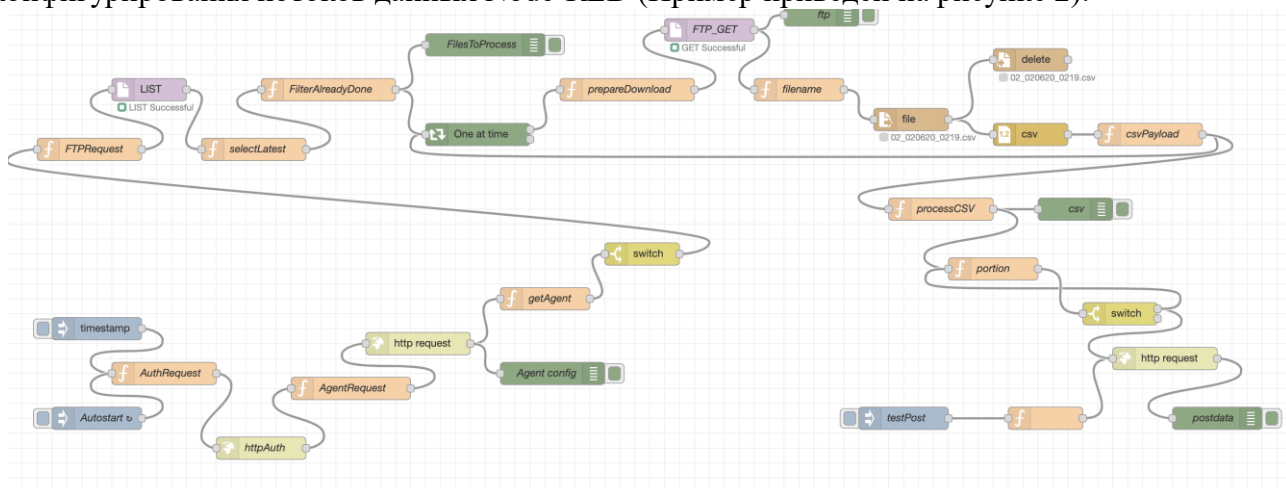


Рисунок 2 – Агент в среде Node-RED

Среда Node-RED предоставляет возможности работы с основными сетевыми протоколами, хорошо развита сообществом и имеет большое количество дополнительных плагинов для реализации разных задач. Последовательное выполнение запросов к Платформе легко реализуется на стандартных компонентах среды. Дополнительная обработка данных

Име. № дубл.
Име. № дубл.
Взам. име. №
Подпись и дата
Име. № подл.

Изм.	Лист	№ документа	Подпись	Дата	Лист
					15

может осуществляться на серверном JavaScript. Основной объект, которым оперирует поток в Node-RED – это JavaScript объект сообщения msg, который передается от одного действия (узла) потока к другому. Приведенный выше пример Агента реализует следующую логику:

- поток стартует автоматически по таймеру «Autostart»;
- заполняет параметры авторизационного запроса в msg (узел AuthRequest);
- выполняет авторизационный запрос к Модулю ИБ (узел httpAuth);
- заполняет параметры запроса на получение конфигурации Агента (узел AgentRequest);
- выполняет запрос конфигурации Агента (узел http request);
- сохраняет ответ в поле msg.agent (узел getAgent);
- получает список файлов csv с FTP сервера, куда эти файлы, содержащие значения датчиков, периодически доставляются внешним сервисом (узлы FTPRequest, LIST);
- выбирают последние по дате и еще не обработанные ранее файлы (узлы selectLatest, FilterAlreadyDone);
- осуществляют последовательный обход по каждому файлу из списка (узел One at time);
- для каждого файла получает его содержимое (узел prepareDownload, FTP_GET, filename, file);
- парсит CSV файл (узел csv) и удаляет скаченный файл (узел delete);
- готовит пакет данных, осуществляет поиск датчиков по их именам в конфигурации Агента и подставляет данные в пакет данных из соответствующей колонки csv файла (узлы csvPayload, processCSV);
- делит пакет на порции запросов (узел portion);
- отправляет каждую порцию HTTP запросом (узел http request).

Полный листинг Агента приведен в п.2 и может быть использован в среде Node-RED. Сама среда может быть развернута как микросервис на мощностях ИС ПСД или на любом другом сервере во вне ИС ПСД.

2.6.2. Пример HTTP Агента LoRaWAN устройств на базе Java приложения

2.6.2.1. Архитектура Агента LoRaWAN устройств на базе Java приложения

Агенты для подключения устройств из LoRaWAN сети реализованы в виде Агента, размещаемого в контуре Системы.

Архитектура Агента LoRaWAN сети приведена ниже (Рисунок 3).

Име. № подл.	Подпись и дата	Взам. ине. №	Име. № дубл.	Подпись и дата						Лист
										16
					Изм.	Лист	№ документа	Подпись	Дата	

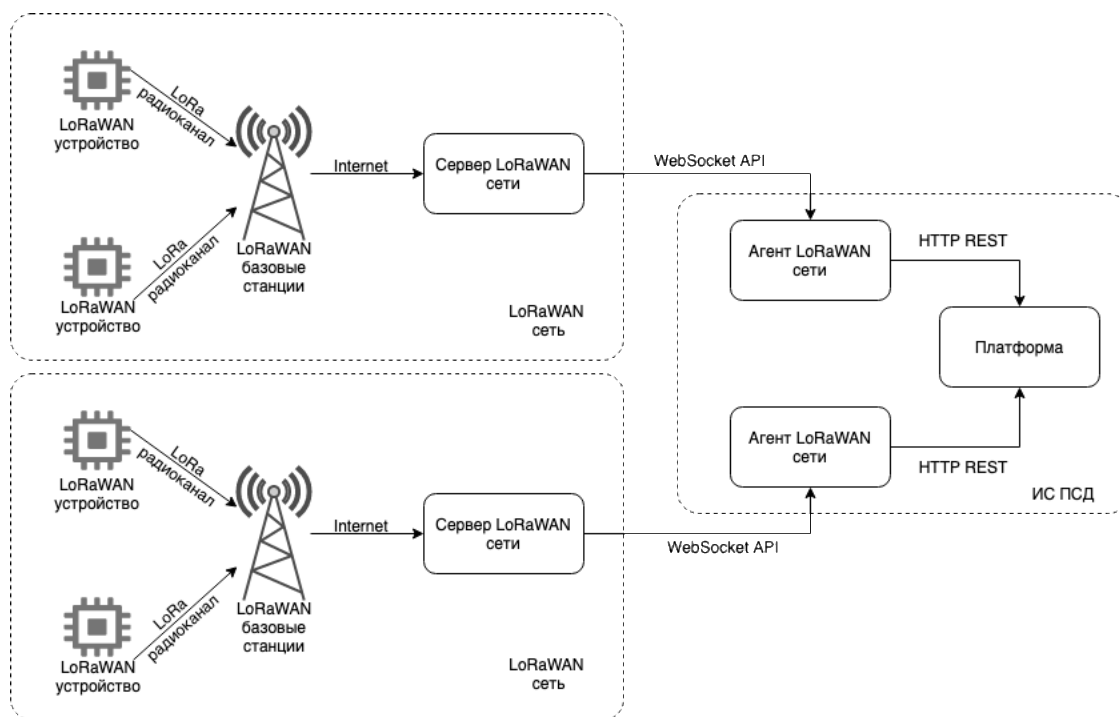


Рисунок 3 – Архитектура подключения устройств LoRaWAN сети

Для каждой LoRaWAN сети используется свой сервер LoRaWAN сети и соответствующий ему Агент LoRaWAN сети. Агент соединяется с сервером LoRaWAN сети по WebSocket API и принимает данные от всех устройств этой сети. Пакет от устройства содержит данные о замерах и идентифицируется уникальным идентификатором устройства devEui. Получив пакет от устройства, Агент получает из платформы тэг устройства, у которого в атрибутах имеется тэг devEui с равным значением. В этом же тэге в атрибуте model имеется информация о модели устройства, по которому Агент понимает, как необходимо распаковывать пакет данных. Если же устройство является модемом, то Агент анализирует набор датчиков под тэгом устройства. В этом случае, Агент вместе в пакете получает так же и номер входа, к которому подключен датчик, приславший пакет. Агент анализирует идентификаторы (name) каждого датчика устройства, по его значению формата (input<цифра>) и определяет нужный датчик. В тэге самого датчика в атрибуте model указана модель датчика. По этой информации Агент понимает, как дешифруется пакет. Расшифрованный пакет отправляется по HTTP API Агента отправляет данные по каждому датчику (в пакете могут быть переданы данные сразу от нескольких датчиков одного устройства) в Платформу в соответствующие тэги событий (event).

Кроме того, Агент LoRaWAN сети предоставляет HTTP REST API для удаленной конфигурации Агента.

Агент отслеживает состояние соединения с сервером LoRaWAN сети и восстанавливает его в случае разрыва.

Агент LoRaWAN сети поддерживает следующие модели устройств:

- Vega Smart-HS0101 - датчик влажности/температуры/открытия/ускорения;
- Vega Smart-SS0101 - датчик дыма;
- Vega СИ-11 - счётчик импульсов с 4 охранными входами и поддержкой подключаемых проводным способом датчиков:
 - Vega ДП-1 - датчик протечки;
 - извещатель пожарный дымовой автономный ИП 212-03К "ДОКА-а";

Име. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					17

- датчик разбития стекла.

Для облегчения организации процесса закупки требуемых устройств разработан специальный шаблон требований (Приложение З). Перечень подключенных на текущий момент типов LoRaWAN устройств приведен в Приложении Н. Порядок подключения LoRaWAN устройств приведен в Приложении О.

2.6.2.2. Реализация приложения Агента

Агент является Spring Boot приложением и компилируется в jar файл, в котором содержатся все зависимости и контейнер сервлетов Tomcat. Приложение запускается как сервис, и подключается к LoRaWAN серверу по протоколу WebSocket и остается постоянно подключенным, в случае проблем со связью автоматическое переподключается. При получении значений от LoRaWAN сервера, Агент выполняет преобразования и отправляет значения в платформу.

Для Агента можно посмотреть настройки через API:

- GET /agent-token;
- GET /connection-properties.

И изменить их через API (но изменения не будут сохранены после перезапуска, для этого существует файл конфигурации):

- POST /agent-token;
- POST /connection-properties.

2.6.2.3. Структура проекта, сборка и зависимости

Приложение агента является стандартным Spring Boot приложением. Структура определяется системой сборки Maven.

Для сборки агента сперва нужно собрать общую часть командой – `iot-solution-backend` (находясь в директории общей части):

```
mvn clean install
```

, после этого можно собрать агента командой (находясь в директории агента):

```
mvn clean package
```

Агент собирается в виде исполняемого jar файла со встроенным контейнером сервлетов Tomcat и всеми зависимостями и может быть запущен командой:

```
java -jar <название jar файла>
```

2.6.2.4. Получение данных с датчиков

При запуске Агента, класс `ru.waveaccess.adapterservice.service.WebSocketClientHolder` устанавливает соединение с LoRaWAN сервером, к которому подключены датчики, параметры доступа к серверу лежат в файле настроек `application.yml`

```
# Параметры LORAWAN сервера
websocket:
  url: "ws://89.208.221.67:8002"
  login: "root"
  password: "*****"
```

При получении события от LoRaWAN сервера, Агент передает их для обработки в `ru.waveaccess.adapterservice.service.DataParseService`, который после обработки передает их для отправки в `ru.waveaccess.adapterservice.service.DataSendingService`, который для отправки использует клиент платформы `io.swagger.service.HttpGatewayService`.

Описание полей пакета данных, приходящих от LoRaWAN сервера приведено в таблице 1.

Име. № дубл.	Взаим. име. №	Подпись и дата	Подпись и дата			Лист
			Име. № подл.	Лист	№ документа	
						18

Таблица 1. Описание полей пакета данных от LoRaWAN сервера.

Название поля	Описание
devEui	Уникальный идентификатор, который присваивается устройству в процессе производства
appEui	Идентификатор приложения, который записывается в устройство
data	Данные в бинарном виде – распознаются в DataParseService
port	Порт, используемы для соединения устройства с сервером

2.6.2.5. Преобразование данных

Для того чтобы преобразовать данные из бинарного формата датчика в структурированный формат, понимаемый Платформой, нужно понимать, от какого датчика пришли значения. В первую очередь определяется заполнено ли поле appEui. Если оно заполнено, то сразу известно, каким образом нужно преобразовывать данные из бинарного вида в структурированный. Если же данное значение не прописано для датчика, то нужно получить информацию об устройстве из метаданных (описывается в следующем разделе) по devEui и уже в описании устройства будет указана модель устройства (атрибут model в теге устройства), по которой мы сможем определить, какой алгоритм использовать. Далее по этому алгоритму из бинарного массива извлекаются нужные группы байт, которые затем преобразуются в различные необходимые значения. На данный момент реализована поддержка следующих моделей устройств:

- Вега Абсолют HS0101;
- Вега Абсолют SS0101;
- Вега Абсолют СИ-11.

2.6.2.6. Получение метаданных

Так как существует необходимость знать в какой тэг устройства должны быть записаны данные, а также модель устройства (если по какой-то причине не указан appEui), то нужно получать метаданные из Платформы. Это реализовано в классе ru.waveaccess.adapterservice.service.StorageService, который с заданной периодичностью обновляет информацию об устройствах. Для этого используется кэш в виде компонента Spring ru.waveaccess.iot.cache.service.TagsCache, который является компонентом – iot-solution-backend – общей частью проектов вертикального решения. Кэш с заданной периодичностью получает всю структуру метаданных от Платформы.

2.6.2.7. Отправка данных в платформу

Для того, чтобы отправлять данные в платформу, в платформе должна быть создана запись агента (описывается в отдельном документе). При создании такой записи там будет указан токен агента, который нужно прописать в настройках:

```
mail:
  http-gateway:
    agent-token : "4db1c5a6-8401-4f39-a6ff-e85fd165b97c"
```

Отправку данных осуществляет сервис ru.waveaccess.adapterservice.service.DataSendingService.

Для отправки данных по протоколу HTTP нужно знать идентификаторы тэгов устройств в структуре Платформы, чтобы значение измерения отправить для данного тэга,

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					19

поэтому мы получаем из StorageService значение тэга (от которого нам нужен только идентификатор) и затем, зная данный идентификатор, отправляем в платформу значение через метод сервиса io.swagger.service.HttpGatewayService (также является компонентом общей части iot-solution-backend) – addEvent.

2.6.2.8. Настройки агента

Настройки агента находятся в файле: src/main/resources/application.yml

Часть настроек представлена ниже:

```
# Порт для развертывания эндпоинтов конфигурации
server:
  port: 8070

# Адреса сервисов платформы IOT
mail:
  registry:
    base-url: "http://iiot.gov.ru/registry/v1"
  http-gateway:
    base-url: "http://api.iiot.gov.ru/http-gateway/v1"
    agent-token : "4db1c5a6-8401-4f39-a6ff-e85fd165b97c"
    eventsBatchSize: 20
  tarantool-storage:
    base-url: "http://iiot.gov.ru/operational-storage/api/v1"
  digital-twins:
    base-url: "http://iiot.gov.ru/digital-twins/api/v1"
  clickhouse-storage:
    base-url: "http://iiot.gov.ru/long-term-storage/api/v1"

login:
  userInfoUrl: https://iiot.gov.ru/keycloak/auth/realms/iiot/protocol/openid-connect/userinfo
  clientInfoUrl: https://iiot.gov.ru/identity-manager/api/v1/user_info

# Параметры LORAWAN сервера
websocket:
  url: "ws://89.208.221.67:8002"
  login: "root"
  password: "*****"

# Настройки отправки статуса устройств в платформу
sensor-status:
  cron: 0 0/2 * * * *
  last-values-count: 3
  increased-period-by-times: 1.5

<namespace>
```

Име. № подл.	Подпись и дата	Взам. ине. №	Име. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					20

1. Описание интерфейса взаимодействия с Платформой для работы Агентов

1.1. Описание HTTP REST API по стандарту OpenAPI в формате JSON

```
{
  "schemes": [
    "http"
  ],
  "swagger": "2.0",
  "info": {
    "description": "COIIOT Gateway HTTP REST API",
    "title": "COIIOT Gateway HTTP REST API",
    "version": "1.0.0"
  },
  "host": "localhost",
  "basePath": "/v1",
  "paths": {
    "/agents/config": {
      "get": {
        "produces": [
          "application/json"
        ],
        "tags": [
          "agents"
        ],
        "summary": "Get agent config",
        "operationId": "getAgentConfig",
        "parameters": [
          {
            "type": "string",
            "x-nullable": true,
            "name": "version",
            "in": "query"
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "$ref": "#/definitions/ConfigObject"
            }
          },
          "400": {
            "description": "Bad params supplied",
            "schema": {
              "$ref": "#/definitions/ErrorResponse"
            }
          },
          "401": {
            "description": "Unauthorized",
            "schema": {
              "$ref": "#/definitions/ErrorResponse"
            }
          },
          "404": {
            "description": "Tag not found",
            "schema": {
              "$ref": "#/definitions/ErrorResponse"
            }
          },
          "500": {
            "description": "Internal server error",
            "schema": {
              "$ref": "#/definitions/ErrorResponse"
            }
          }
        }
      }
    }
  }
}
```

Име. № подл.	Подпись и дата	Взам. ине. №	Инв. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	

```

"/events": {
  "post": {
    "consumes": [
      "application/json"
    ],
    "produces": [
      "application/json"
    ],
    "tags": [
      "events"
    ],
    "summary": "Add event",
    "operationId": "addEvent",
    "parameters": [
      {
        "description": "Event object.",
        "name": "body",
        "in": "body",
        "required": true,
        "schema": {
          "$ref": "#/definitions/AddEvent"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "OK"
      },
      "400": {
        "description": "Bad params supplied",
        "schema": {
          "$ref": "#/definitions/ErrorResponse"
        }
      },
      "401": {
        "description": "Unauthorized",
        "schema": {
          "$ref": "#/definitions/ErrorResponse"
        }
      },
      "404": {
        "description": "Tag not found",
        "schema": {
          "$ref": "#/definitions/ErrorResponse"
        }
      },
      "500": {
        "description": "Internal server error",
        "schema": {
          "$ref": "#/definitions/ErrorResponse"
        }
      }
    }
  }
},
"/swagger.json": {
  "get": {
    "produces": [
      "application/json"
    ],
    "tags": [
      "swagger"
    ],
    "summary": "Get swagger json",
    "operationId": "getSwaggerJSON",
    "responses": {
      "200": {
        "description": "OK",
        "schema": {

```

Име. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					22

```

        "type": "object"
      }
    },
    "401": {
      "description": "Unauthorized",
      "schema": {
        "$ref": "#/definitions/ErrorResponse"
      }
    },
    "500": {
      "description": "Internal server error",
      "schema": {
        "$ref": "#/definitions/ErrorResponse"
      }
    }
  }
},
"definitions": {
  "AddEvent": {
    "type": "object",
    "required": [
      "tags"
    ],
    "properties": {
      "tags": {
        "type": "array",
        "maxItems": 20,
        "items": {
          "$ref": "#/definitions/TagValueObject"
        }
      }
    }
  },
  "AgentConfigObject": {
    "type": "object",
    "properties": {
      "devices": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/DeviceConfigObject"
        }
      },
      "x-nullable": true
    },
    "id": {
      "type": "integer",
      "x-nullable": true,
      "example": 1
    },
    "name": {
      "type": "string",
      "x-nullable": true,
      "example": "some_agent"
    },
    "tag": {
      "$ref": "#/definitions/TagConfigObject"
    }
  },
  "x-nullable": true
},
"ConfigObject": {
  "type": "object",
  "properties": {
    "agent": {
      "$ref": "#/definitions/AgentConfigObject"
    },
    "version": {
      "type": "string",

```

Име. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					23

```

        "x-nullable": true,
        "example": "v1"
    }
},
"DeviceConfigObject": {
    "type": "object",
    "properties": {
        "driver": {
            "$ref": "#/definitions/DriverConfigObject"
        },
        "driver_config": {
            "description": "Non typed dictionary for setting template.",
            "type": "object",
            "example": "{\"key\": \"value\"}"
        },
        "id": {
            "type": "integer",
            "x-nullable": true,
            "example": 1
        },
        "name": {
            "type": "string",
            "x-nullable": true,
            "example": "some_agent"
        },
        "tag": {
            "$ref": "#/definitions/TagConfigObject"
        }
    },
    "x-nullable": true
},
"DriverConfigObject": {
    "type": "object",
    "properties": {
        "id": {
            "type": "integer",
            "x-nullable": true,
            "example": 1
        },
        "name": {
            "type": "string",
            "x-nullable": true,
            "example": "modbus driver"
        },
        "protocol": {
            "type": "string",
            "x-nullable": true,
            "example": "modbus"
        }
    }
},
"ErrorResponse": {
    "type": "object",
    "properties": {
        "code": {
            "type": "integer",
            "example": 300
        },
        "message": {
            "type": "string",
            "example": "Something bad happens."
        }
    }
},
"TagConfigObject": {
    "type": "object",
    "properties": {
        "attrs": {

```

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	


```

    "description": "Non typed dictionary for user defined data.",
    "type": "object",
    "x-nullable": true,
    "x-omitempty": true,
    "example": "{\"key\": \"value\"}"
  },
  "children": {
    "type": "array",
    "items": {
      "$ref": "#/definitions/TagConfigObject"
    },
    "x-nullable": true
  },
  "id": {
    "type": "integer",
    "x-nullable": true,
    "example": 1
  },
  "name": {
    "type": "string",
    "x-nullable": true,
    "example": "some_tag"
  },
  "properties": {
    "type": "object",
    "x-nullable": true,
    "example": {
      "a": "b"
    }
  },
  "type": {
    "$ref": "#/definitions/TagTypeObject"
  }
}
},
"TagTypeObject": {
  "type": "object",
  "properties": {
    "id": {
      "type": "integer",
      "x-nullable": true,
      "example": 1
    },
    "name": {
      "type": "string",
      "x-nullable": true,
      "example": "undefined"
    }
  }
},
"TagValueObject": {
  "type": "object",
  "required": [
    "id",
    "value",
    "timestamp"
  ],
  "properties": {
    "id": {
      "type": "integer",
      "example": 1
    },
    "timestamp": {
      "description": "Timestamp with microseconds precision",
      "type": "integer",
      "format": "int64",
      "example": "1257894000000000"
    },
    "value": {

```

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					25


```

    $ref: '#/definitions/ErrorResponse'
  '401':
    description: Unauthorized
    schema:
      $ref: '#/definitions/ErrorResponse'
  '404':
    description: Tag not found
    schema:
      $ref: '#/definitions/ErrorResponse'
  '500':
    description: Internal server error
    schema:
      $ref: '#/definitions/ErrorResponse'
/events:
  post:
    consumes:
      - application/json
    produces:
      - application/json
    tags:
      - events
    summary: Add event
    operationId: addEvent
    parameters:
      - description: Event object.
        name: body
        in: body
        required: true
        schema:
          $ref: '#/definitions/AddEvent'
    responses:
      '200':
        description: OK
      '400':
        description: Bad params supplied
        schema:
          $ref: '#/definitions/ErrorResponse'
      '401':
        description: Unauthorized
        schema:
          $ref: '#/definitions/ErrorResponse'
      '404':
        description: Tag not found
        schema:
          $ref: '#/definitions/ErrorResponse'
      '500':
        description: Internal server error
        schema:
          $ref: '#/definitions/ErrorResponse'
/swagger.json:
  get:
    produces:
      - application/json
    tags:
      - swagger
    summary: Get swagger json
    operationId: getSwaggerJSON
    responses:
      '200':
        description: OK
        schema:
          type: object
      '401':
        description: Unauthorized
        schema:
          $ref: '#/definitions/ErrorResponse'
      '500':
        description: Internal server error
        schema:

```

Име. № подл.	
Подпись и дата	
Взам. ине. №	
Име. № дубл.	
Подпись и дата	

Изм.	Лист	№ документа	Подпись	Дата	
------	------	-------------	---------	------	--

```

    $ref: '#/definitions/ErrorResponse'
definitions:
  AddEvent:
    type: object
    required:
      - tags
    properties:
      tags:
        type: array
        maxItems: 20
        items:
          $ref: '#/definitions/TagValueObject'
  AgentConfigObject:
    type: object
    properties:
      devices:
        type: array
        items:
          $ref: '#/definitions/DeviceConfigObject'
      x-nullable: true
      id:
        type: integer
        x-nullable: true
        example: 1
      name:
        type: string
        x-nullable: true
        example: some_agent
      tag:
        $ref: '#/definitions/TagConfigObject'
      x-nullable: true
  ConfigObject:
    type: object
    properties:
      agent:
        $ref: '#/definitions/AgentConfigObject'
      version:
        type: string
        x-nullable: true
        example: v1
  DeviceConfigObject:
    type: object
    properties:
      driver:
        $ref: '#/definitions/DriverConfigObject'
      driver_config:
        description: Non typed dictionary for setting template.
        type: object
        example: '{"key": "value"}'
      id:
        type: integer
        x-nullable: true
        example: 1
      name:
        type: string
        x-nullable: true
        example: some_agent
      tag:
        $ref: '#/definitions/TagConfigObject'
      x-nullable: true
  DriverConfigObject:
    type: object
    properties:
      id:
        type: integer
        x-nullable: true
        example: 1
      name:
        type: string

```

Ине. № подл.	
Подпись и дата	
Взам. ине. №	
Ине. № дубл.	
Подпись и дата	

Изм.	Лист	№ документа	Подпись	Дата	
------	------	-------------	---------	------	--

```

    x-nullable: true
    example: modbus driver
  protocol:
    type: string
    x-nullable: true
    example: modbus
ErrorResponse:
  type: object
  properties:
    code:
      type: integer
      example: 300
    message:
      type: string
      example: Something bad happens.
TagConfigObject:
  type: object
  properties:
    attrs:
      description: Non typed dictionary for user defined data.
      type: object
      x-nullable: true
      x-omitempty: true
      example: '{"key": "value"}'
    children:
      type: array
      items:
        $ref: '#/definitions/TagConfigObject'
      x-nullable: true
    id:
      type: integer
      x-nullable: true
      example: 1
    name:
      type: string
      x-nullable: true
      example: some_tag
    properties:
      type: object
      x-nullable: true
      example:
        a: b
  type:
    $ref: '#/definitions/TagTypeObject'
TagTypeObject:
  type: object
  properties:
    id:
      type: integer
      x-nullable: true
      example: 1
    name:
      type: string
      x-nullable: true
      example: undefined
TagValueObject:
  type: object
  required:
    - id
    - value
    - timestamp
  properties:
    id:
      type: integer
      example: 1
    timestamp:
      description: Timestamp with microseconds precision
      type: integer
      format: int64

```

Име. № подл.	
Подпись и дата	
Взам. име. №	
Име. № дубл.	
Подпись и дата	

Изм.	Лист	№ документа	Подпись	Дата	
------	------	-------------	---------	------	--

```

    example: '1257894000000000'
  value:
    description: Value of the tag
    example: 1
  securityDefinitions:
    AgentTokenHeader:
      type: apiKey
      name: X-Agent-Token
      in: header
  security:
    - AgentTokenHeader: []
  tags:
    - name: swagger
    - name: agents
    - name: events
  externalDocs:
    description: Find out more about Swagger
    url: 'http://swagger.io'

```

Име. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата					Лист
								30	
Изм.	Лист	№ документа	Подпись	Дата					


```

        "noerr": 0,
        "x": 330,
        "y": 260,
        "wires": [
            [
                "c3af6bef.2d61d8"
            ]
        ]
    },
    {
        "id": "6c338b6a.818004",
        "type": "function",
        "z": "45305b0.cbb40a4",
        "name": "prepareDownload",
        "func":      "//msg.filename      =      'test.csv';\nmsg.localFilename      =
'/home/saper/'+msg.payload.name;//'/home/saper/test1.csv';\nmsg.filename      =
'/#recycle/Chelyabinsk/receive/'+msg.payload.name;\nmsg.localFilename      =
msg.filename.replace(/^.*[\\\\\\\\\\\\\\\\]/,
'');//'/home/saper/test1.csv';\nmsg.password=''\nmsg.operation      =      \"get\";\nreturn
msg;\",
        "outputs": 1,
        "noerr": 0,
        "x": 830,
        "y": 180,
        "wires": [
            [
                "e66ec11.4cc1b4"
            ]
        ]
    },
    {
        "id": "e66ec11.4cc1b4",
        "type": "advanced-ftp",
        "z": "45305b0.cbb40a4",
        "ftp": "d306a2ec.1a3fe",
        "operation": "status",
        "dataType": "binary",
        "filename": "",
        "localFilename": "",
        "workingDir": "",
        "oldPath": "",
        "newPath": "",
        "command": "",
        "throwError": false,
        "showError": true,
        "name": "FTP_GET",
    }
}

```

Име. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					32


```

"x": 940,
"y": 100,
"wires": [
  [
    "7f7c659f.5cd90c",
    "c4817ac1.0ef6b8"
  ]
]
},
{
  "id": "5515926f.2c72dc",
  "type": "csv",
  "z": "45305b0.cbb40a4",
  "name": "",
  "sep": ";",
  "hdrin": "",
  "hdrout": "",
  "multi": "one",
  "ret": "\\n",
  "temp": "",
  "skip": "0",
  "strings": false,
  "x": 1370,
  "y": 240,
  "wires": [
    [
      "8bddf702.c09ae8"
    ]
  ]
},
{
  "id": "e507857d.04a5c8",
  "type": "file in",
  "z": "45305b0.cbb40a4",
  "name": "",
  "filename": "",
  "format": "utf8",
  "chunk": false,
  "sendError": false,
  "encoding": "none",
  "x": 1210,
  "y": 220,
  "wires": [
    [
      "5515926f.2c72dc",
      "8684c859.888ec8"
    ]
  ]
}

```

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					33

```

    ]
  ]
},
{
  "id": "7f7c659f.5cd90c",
  "type": "function",
  "z": "45305b0.cbb40a4",
  "name": "filename",
  "func": "msg.filename = msg.localFilename\nreturn msg;",
  "outputs": 1,
  "noerr": 0,
  "x": 1060,
  "y": 180,
  "wires": [
    [
      "e507857d.04a5c8"
    ]
  ]
},
{
  "id": "2d0ead6f.bcf2a2",
  "type": "inject",
  "z": "45305b0.cbb40a4",
  "name": "",
  "topic": "",
  "payload": "",
  "payloadType": "date",
  "repeat": "",
  "crontab": "",
  "once": false,
  "onceDelay": 0.1,
  "x": 120,
  "y": 500,
  "wires": [
    [
      "a6f0be2c.d827a"
    ]
  ]
},
{
  "id": "2c3f2efd.6f5f42",
  "type": "Serial Iterator",
  "z": "45305b0.cbb40a4",
  "name": "",
  "property": "payload",
  "inputFlow": "input",

```

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					34

```

    "saveOutput": 0,
    "recursive": 0,
    "storeId": 1,
    "x": 630,
    "y": 240,
    "wires": [
      [
        "6c338b6a.818004"
      ],
      []
    ]
  },
  {
    "id": "91f92921.ff5b18",
    "type": "http request",
    "z": "45305b0.cbb40a4",
    "name": "",
    "method": "use",
    "ret": "txt",
    "paytoqs": false,
    "url": "http://10.20.22.30/public/api/http-gateway/api/v1/events",
    "tls": "",
    "persist": false,
    "proxy": "",
    "authType": "",
    "x": 1550,
    "y": 540,
    "wires": [
      [
        "dd4b070d.4dc288"
      ]
    ]
  },
  {
    "id": "66aadb7.72627",
    "type": "function",
    "z": "45305b0.cbb40a4",
    "name": "",
    "func":
      "msg.method=\"POST\";\nmsg.headers={};\n//msg.headers['X-Agent-
Token']=\"3131fab2-5b1c-479d-a30a-d65c64d8792b\";\nmsg.headers['X-Agent-
Token']=\"aa9c5088-95b4-4b7e-ade5-8fc3a59c72dc\";\nmsg.headers['Content-
Type']=\"application/json\";\nlet
now
=
(new
Date()).getTime();\nmsg.payload={\n\t'tags' : [\n\t\t'id' : 226509,\n\t\t'timestamp' :
1588887143222,\n\t\t'value'
:
11.99\n\t]}\n};\n\n\n//msg.payload.tags=[{\n'id\":226508,\n'timestamp\":1588904340485,\n
'value\":0.033},{\n'id\":226509,\n'timestamp\":1588904340485,\n'value\":0.062},{\n'id\":2

```

Ине. № подл.	Подпись и дата	Взам. ине. №	Инв. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					35

```

26510, \"timestamp\":1588904340485, \"value\":1.3}, {\"id\":226511, \"timestamp\":1588904
340485, \"value\":0.004}, {\"id\":226512, \"timestamp\":1588904340485, \"value\":0.0008},
{\"id\":226513, \"timestamp\":1588904340485, \"value\":0.581}, {\"id\":226514, \"timestam
p\":1588904340485, \"value\":0.017}, {\"id\":226516, \"timestamp\":1588904340485, \"value
\":0.03}}\n//msg.payload={\"tags\":[{\"id\":226508, \"timestamp\":1588904340485, \"valu
e\":0.033}, {\"id\":226509, \"timestamp\":1588904340485, \"value\":0.062}, {\"id\":226510
, \"timestamp\":1588904340485, \"value\":1.3}, {\"id\":226511, \"timestamp\":158890434048
5, \"value\":0.004}, {\"id\":226512, \"timestamp\":1588904340485, \"value\":0.0008}, {\"id
\":226513, \"timestamp\":1588904340485, \"value\":0.581}, {\"id\":226514, \"timestamp\":1
588904340485, \"value\":0.017}, {\"id\":226516, \"timestamp\":1588904340485, \"value\":0.
03}}];\nmsg = {\"payload\":\n
                                {\"tags\":[\n
                                {\"id\":226508, \"timestamp\":1588929540000000, \"value\":0.002}, \n
                                {\"id\":226509, \"timestamp\":1588929540000000, \"value\":0.004}, \n
                                {\"id\":226511, \"timestamp\":1588929540000000, \"value\":0.002}, \n
                                {\"id\":226513, \"timestamp\":1588929540000000, \"value\":0.05}, \n
                                {\"id\":226514, \"timestamp\":1588929540000000, \"value\":0.009}, \n
                                {\"id\":226516, \"timestamp\":1588929540000000, \"value\":0.01}}\n
                                }, \n
                                \"method\": \"POST\", \n
                                \"headers\": {\n
                                \"X-Agent-Token\": \"3131fab2-5b1c-479d-a30a-d65c64d8792b\", \n
                                \"Content-Type\": \"application/json\"}, \n
                                \"_msgid\": \"bae2c015.db3ef\"};\nreturn
msg;\",

```

```

    "outputs": 1,
    "noerr": 0,
    "x": 1390,
    "y": 620,
    "wires": [
      [
        "91f92921.ff5b18"
      ]
    ]
  },
  {
    "id": "3d747395.1288ec",
    "type": "inject",
    "z": "45305b0.cbb40a4",
    "name": "testPost",
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "repeat": "",
    "crontab": "",
    "once": false,
    "onceDelay": 0.1,
    "x": 1200,
    "y": 620,
    "wires": [

```

Ине. № подл.	Подпись и дата	Ине. № дубл.	Подпись и дата
	Взам. ине. №	Ине. № дубл.	Ине. № дубл.
	Ине. № дубл.	Ине. № дубл.	Ине. № дубл.

Изм.	Лист	№ документа	Подпись	Дата	Лист
					36

```

        [
            "66aadfb7.72627"
        ]
    ],
    {
        "id": "dd4b070d.4dc288",
        "type": "debug",
        "z": "45305b0.cbb40a4",
        "name": "postdata",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "true",
        "targetType": "full",
        "x": 1620,
        "y": 620,
        "wires": []
    },
    {
        "id": "c0df7042.000a7",
        "type": "http request",
        "z": "45305b0.cbb40a4",
        "name": "",
        "method": "use",
        "ret": "obj",
        "paytoqs": false,
        "url": "http://10.20.22.30/public/api/http-gateway/api/v1/agents/config",
        "tls": "",
        "persist": false,
        "proxy": "",
        "authType": "",
        "x": 550,
        "y": 520,
        "wires": [
            [
                "7999ba97.0bbf44",
                "b2533155.5218"
            ]
        ]
    },
    {
        "id": "b906f85b.47d038",
        "type": "inject",
        "z": "45305b0.cbb40a4",

```

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					37

```

        "name": "Autostart",
        "topic": "",
        "payload": "",
        "payloadType": "date",
        "repeat": "120",
        "crontab": "",
        "once": true,
        "onceDelay": 0.1,
        "x": 120,
        "y": 620,
        "wires": [
            [
                "a6f0be2c.d827a"
            ]
        ]
    },
    {
        "id": "8bddf702.c09ae8",
        "type": "function",
        "z": "45305b0.cbb40a4",
        "name": "csvPayload",
        "func": "let csv=msg.payload;\n//msg.payload={};\nmsg.payload.csv =
csv;\nmsg.csv=csv\n//msg.paths={};\n//msg.paths[\"csv\"]=csv;\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 1530,
        "y": 240,
        "wires": [
            [
                "2c3f2efd.6f5f42",
                "c7832dbb.1ac3a"
            ]
        ]
    },
    {
        "id": "7999ba97.0bbf44",
        "type": "function",
        "z": "45305b0.cbb40a4",
        "name": "getAgent",
        "func": "//msg.paths={};\nmsg.agent = null;\nif (msg.payload){\n    if
(msg.payload.agent){\n        msg.agent = msg.payload.agent;\n
}\n}\n//msg.paths[\"agent\"]=msg.agent;\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 720,
        "y": 460,
    }
}

```

Ине. № подл.	Подпись и дата
Взам. ине. №	Подпись и дата
Ине. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					38

```

    "wires": [
      [
        "c246608.b9a70a"
      ]
    ],
    {
      "id": "636bb922.fba58",
      "type": "debug",
      "z": "45305b0.cbb40a4",
      "name": "csv",
      "active": true,
      "tosidebar": true,
      "console": false,
      "tostatus": false,
      "complete": "true",
      "targetType": "full",
      "x": 1430,
      "y": 340,
      "wires": []
    },
    {
      "id": "c7832dbb.1ac3a",
      "type": "function",
      "z": "45305b0.cbb40a4",
      "name": "processCSV",
      "func": "let filePrefix = '';\nif (msg.filename){\n    let match =
/^((\\d+)_)\\.exec(msg.filename);\n    if (match.length == 2)
filePrefix=match[1];\n}\n\nlet tz = 0;\nif (msg.timezone) tz = msg.timezone;\nif
(msg.csv){\n    if (msg.csv.col1){\n        let m =
msg.csv.col1.match(/^(\\d{2})\\. (\\d{2})\\. (\\d{2}) (\\d{2}): (\\d{2})$/);\n    if
(m.length == 6){\n        let d = new Date();\n    d.setUTCFullYear(\"20\"+m[3]);\n    d.setUTCMonth(m[2]-1);\n    d.setUTCDate(m[1]);\n    d.setUTCMinutes(m[5]);\n    d.setUTCHours(m[4]-
tz);\n    d.setUTCSeconds(0);\n    msg.csv.tm = d.toUTCString();\n    msg.csv.ts= d.getTime()*1000;\n    }\n    }\n    //CO, мг/м3 NO, мг/м3\тNO2,
мг/м3\тNH3, мг/м3\тSO2, мг/м3\тH2S, мг/м3\тCH2O, мг/м3\тPM10, мг/м3\тPM2.5, мг/м3\тT,
°C\тH, %\тP, мм.рт.ст.\тV, м/с\тN, град\тT зонд 1<10,\т°C T зонд 1>35,\т°C T зонд
2<10,\т°C T зонд 2>35,\т°C T зонд 3<10,\т°C T зонд 3>35,\т°C T зонд 4<10,\т°C T зонд
4>35,\т°C T пав, 'C\тU, B\тВскрытие Пожар\n    //CO, NO, NO2, NH3, SO2, H2S,
CH2O,PM10,PM2.5, T, H, P, V, N, T 1<10,\т°C T зонд 1>35,\т°C T зонд 2<10,\т°C T зонд
2>35,\т°C T зонд 3<10,\т°C T зонд 3>35,\т°C T зонд 4<10,\т°C T зонд 4>35,\т°C T пав,
'C\тU, B\тВскрытие Пожар\n    let values=[];\n    let v = null;\n    if (msg.csv.col2)
{\n        v = parseFloat(msg.csv.col2.replace(\"\\\", \"\\\", \".\")); \n    if (!isNaN(v))
values[\"CO\"]=v;\n    }\n    if (msg.csv.col3) {\n        v =
parseFloat(msg.csv.col3.replace(\"\\\", \"\\\", \".\")); \n    if (!isNaN(v))

```

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата

```

values[\"NO\\\"]=v;\n          }\n          if (msg.csv.col4) {\n          v =
parseFloat(msg.csv.col4.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"NO2\\\"]=v;\n          }\n          if (msg.csv.col5){\n          v =
parseFloat(msg.csv.col5.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"NH3\\\"]=v;\n          } \n          if (msg.csv.col6){\n          v =
parseFloat(msg.csv.col6.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"SO2\\\"]=v;\n          } \n          if (msg.csv.col7){\n          v =
parseFloat(msg.csv.col7.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"H2S\\\"]=v;\n          } \n          if (msg.csv.col8){\n          v =
parseFloat(msg.csv.col8.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"CH2O\\\"]=v;\n          } \n          if (msg.csv.col9) {\n          v =
parseFloat(msg.csv.col9.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"PM10\\\"]=v;\n          }\n          if (msg.csv.col10) {\n          v =
parseFloat(msg.csv.col10.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"PM2.5\\\"]=v;\n          }\n          if (msg.csv.col11) {\n          v =
parseFloat(msg.csv.col11.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"temperature\\\"]=v;\n          }\n          if (msg.csv.col12) {\n          v =
parseFloat(msg.csv.col12.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"humidity\\\"]=v;\n          }\n          if (msg.csv.col13) {\n          v =
parseFloat(msg.csv.col13.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"pressure\\\"]=v;\n          }\n          if (msg.csv.col14) {\n          v =
parseFloat(msg.csv.col14.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"velocity\\\"]=v;\n          }\n          if (msg.csv.col15) {\n          v =
parseFloat(msg.csv.col15.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"direction\\\"]=v;\n          }\n          if (msg.csv.col24) {\n          v =
parseFloat(msg.csv.col24.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"pavilion_temperature\\\"]=v;\n          }\n          if (msg.csv.col25) {\n          v =
parseFloat(msg.csv.col25.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v))
values[\"voltage\\\"]=v;\n          }\n          if (msg.csv.col26) {\n          v =
parseFloat(msg.csv.col26.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v)) {\n
values[\"open_alert\\\"]= (v > 0);\n          values[\"open_alarm\\\"]= (v > 0);\n
}\n          }\n          if (msg.csv.col27) {\n          v =
parseFloat(msg.csv.col27.replace(\"\\\",\\\",\\\".\")); \n          if (!isNaN(v)) {\n
values[\"fire_alert\\\"]= (v > 0);\n          values[\"fire_alarm\\\"]= (v > 0);\n
}\n          }\n          \n          msg.csv.values = values;\n          \n          /*\n          if (msg.csv.col2) msg.csv.CO
= parseFloat(msg.csv.col2.replace(\"\\\",\\\",\\\".\")); \n          if (msg.csv.col3) msg.csv.NO =
parseFloat(msg.csv.col3.replace(\"\\\",\\\",\\\".\")); \n          if (msg.csv.col4) msg.csv.NO2 =
parseFloat(msg.csv.col4.replace(\"\\\",\\\",\\\".\")); \n          if (msg.csv.col5) msg.csv.NH3 =
parseFloat(msg.csv.col5.replace(\"\\\",\\\",\\\".\")); \n          if (msg.csv.col6) msg.csv.SO2 =
parseFloat(msg.csv.col6.replace(\"\\\",\\\",\\\".\")); \n          if (msg.csv.col7) msg.csv.H2S =
parseFloat(msg.csv.col7.replace(\"\\\",\\\",\\\".\")); \n          if (msg.csv.col8) msg.csv.CH2O =
parseFloat(msg.csv.col8.replace(\"\\\",\\\",\\\".\")); \n          if (msg.csv.col9) msg.csv.PM10 =
parseFloat(msg.csv.col9.replace(\"\\\",\\\",\\\".\")); \n          if (msg.csv.col10) msg.csv.PM25 =
parseFloat(msg.csv.col10.replace(\"\\\",\\\",\\\".\")); \n          if (msg.csv.col11)
msg.csv.temperature = parseFloat(msg.csv.col11.replace(\"\\\",\\\",\\\".\")); \n          if
(msg.csv.col12) msg.csv.humidity = parseFloat(msg.csv.col12.replace(\"\\\",\\\",\\\".\")); \n

```

Име. № подл.	
Подпись и дата	
Взам. ине. №	
Име. № дубл.	
Подпись и дата	

Изм.	Лист	№ документа	Подпись	Дата	Лист
					40


```

//if (msg.csv.col13) msg.csv.pressure =
parseFloat(msg.csv.col13.replace(",",".\\"));\n if (msg.csv.col14)
msg.csv.velocity = parseFloat(msg.csv.col14.replace(",",".\\"));\n if
(msg.csv.col15) msg.csv.grad = parseFloat(msg.csv.col15.replace(",",".\\"));\n
//if (msg.csv.col24) msg.csv.post_temperature =
parseFloat(msg.csv.col24.replace(",",".\\"));\n //if (msg.csv.col25)
msg.csv.voltage = parseFloat(msg.csv.col25.replace(",",".\\"));\n if
(msg.csv.col26) msg.csv.open_alert = (parseFloat(msg.csv.col26.replace(",",".\\"))
=== 0) ? false:true;\n if (msg.csv.col27) msg.csv.fire_alert =
(parseFloat(msg.csv.col27.replace(",",".\\")) === 0) ? false:true;\n */\n
\n\nmsg.payload={'tags':[]};\n\nif (msg.agent){\n let devices =
msg.agent.devices;\n if (devices && devices.length > 0){\n for (let i = 0;
i < devices.length; i++){
let device = devices[i];\n if
(device.tag){\n if (device.tag.attrs && device.tag.attrs.ftp_prefix ==
filePrefix){\n msg.method="POST";\n msg.headers={};\n
msg.headers['X-Agent-Token']="3131fab2-5b1c-479d-a30a-d65c64d8792b";\n
msg.headers['Content-Type']="application/json";\n
msg.payload={'tags':[]};\n \t//'tags' : [{\n
\t//\t'id' : 226509,\n \t//\t'timestamp': 1588887143222,\n
\t//\t'value' : 11.99\n \t//}];\n if
(device.tag.children){\n let ch = device.tag.children;\n\n
//reset status first\n /*\n for (let
j=0; j < ch.length; j++){
if (ch[j]){
for (let k=0; k < ch[j].children.length; k++){
if
(ch[j].children[k].name===" $is_working"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
'true'});\n }\n
}\n } \n */\n //
set values \n for (let j=0; j < ch.length;
j++){
if (ch[j] && ch[j].name === "state"){
for (let k=0; k < ch[j].children.length; k++){
if
(ch[j].children[k].name===" $status"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
'online'});\n }\n
} else {\n if (ch[j].name in msg.csv.values){
msg.payload.tags.push({'id': ch[j].id, 'timestamp':msg.csv.ts, 'value':
msg.csv.values[ch[j].name]});\n for (let k=0; k <
ch[j].children.length; k++){
if
(ch[j].children[k].name===" $is_working"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n }\n
}\n \n */\n
if (ch[j] && ch[j].name === "NO2" && ("NO2" in msg.csv)){
msg.payload.tags.push({'id': ch[j].id, 'timestamp':msg.csv.ts, 'value':
msg.csv.NO2});\n for (let k=0; k < ch[j].children.length;
k++){
if (ch[j].children[k].name===" $is_working"){\n

```

Ине. № подл.	
Подпись и дата	
Взам. ине. №	
Ине. № дубл.	
Подпись и дата	

Изм.	Лист	№ документа	Подпись	Дата	Лист
					41

```

msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n
}\n
}\n
if (ch[j] && ch[j].name === \"CO\" && (\"CO\" in msg.csv)
){\n
msg.payload.tags.push({'id': ch[j].id,
'timestamp':msg.csv.ts, 'value': msg.csv.CO});\n
for
(let k=0; k < ch[j].children.length; k++){
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n
}\n
}\n
if (ch[j] && ch[j].name === \"NO\" && (\"NO\" in
msg.csv)){\n
msg.payload.tags.push({'id': ch[j].id,
'timestamp':msg.csv.ts, 'value': msg.csv.NO});\n
for
(let k=0; k < ch[j].children.length; k++){
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n
}\n
}\n
if (ch[j] && ch[j].name === \"NH3\" && (\"NH3\" in
msg.csv)){\n
msg.payload.tags.push({'id': ch[j].id,
'timestamp':msg.csv.ts, 'value': msg.csv.NH3});\n
for
(let k=0; k < ch[j].children.length; k++){
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n
}\n
}\n
if (ch[j] && ch[j].name === \"SO2\" && (\"SO2\" in
msg.csv)){\n
msg.payload.tags.push({'id': ch[j].id,
'timestamp':msg.csv.ts, 'value': msg.csv.SO2});\n
for
(let k=0; k < ch[j].children.length; k++){
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n
}\n
}\n
if (ch[j] && ch[j].name === \"H2S\" && (\"H2S\" in
msg.csv)){\n
msg.payload.tags.push({'id': ch[j].id,
'timestamp':msg.csv.ts, 'value': msg.csv.H2S});\n
for
(let k=0; k < ch[j].children.length; k++){
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n
}\n
}\n
if (ch[j] && ch[j].name === \"CH2O\" && (\"CH2O\" in
msg.csv)){\n
msg.payload.tags.push({'id': ch[j].id,
'timestamp':msg.csv.ts, 'value': msg.csv.CH2O});\n
for
(let k=0; k < ch[j].children.length; k++){
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n
}\n
}\n
}\n
if (ch[j] && ch[j].name === \"PM10\" && (\"PM10\" in
msg.csv)){\n
msg.payload.tags.push({'id': ch[j].id,

```

Име. № подл.	
Подпись и дата	
Взам. име. №	
Име. № дубл.	
Подпись и дата	

Изм.	Лист	№ документа	Подпись	Дата		Лист
						42

```

'timestamp':msg.csv.ts, 'value': msg.csv.PM10});\n                                     for
(let k=0; k < ch[j].children.length; k++){ \n                                     if
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n                                     }\n                                     }\n
} \n                                     if (ch[j] && ch[j].name === \"PM2.5\" && (\"PM25\" in
msg.csv)){ \n                                     msg.payload.tags.push({'id': ch[j].id,
'timestamp':msg.csv.ts, 'value': msg.csv.PM25});\n                                     for
(let k=0; k < ch[j].children.length; k++){ \n                                     if
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n                                     }\n                                     }\n
} \n                                     if (ch[j] && ch[j].name === \"temperature\" &&
( \"temperature\" in msg.csv)){ \n                                     msg.payload.tags.push({'id':
ch[j].id, 'timestamp':msg.csv.ts, 'value': msg.csv.PM25});\n                                     for
(let k=0; k < ch[j].children.length; k++){ \n                                     if
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n                                     }\n                                     }\n
} \n                                     if (ch[j] && ch[j].name === \"humidity\" && (\"humidity\"
in msg.csv)){ \n                                     msg.payload.tags.push({'id': ch[j].id,
'timestamp':msg.csv.ts, 'value': msg.csv.PM25});\n                                     for
(let k=0; k < ch[j].children.length; k++){ \n                                     if
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n                                     }\n                                     }\n
} \n                                     if (ch[j] && ch[j].name === \"pressure\" && (\"pressure\"
in msg.csv)){ \n                                     msg.payload.tags.push({'id': ch[j].id,
'timestamp':msg.csv.ts, 'value': msg.csv.PM25});\n                                     for
(let k=0; k < ch[j].children.length; k++){ \n                                     if
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n                                     }\n                                     }\n
} \n                                     if (ch[j] && ch[j].name === \"velocity\" && (\"velocity\"
in msg.csv)){ \n                                     msg.payload.tags.push({'id': ch[j].id,
'timestamp':msg.csv.ts, 'value': msg.csv.PM25});\n                                     for
(let k=0; k < ch[j].children.length; k++){ \n                                     if
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':
true});\n                                     }\n                                     }\n
} \n                                     if (ch[j] && ch[j].name === \"grad\" && (\"grad\" in
msg.csv)){ \n                                     msg.payload.tags.push({'id': ch[j].id,
'timestamp':msg.csv.ts, 'value': msg.csv.PM25});\n                                     for
(let k=0; k < ch[j].children.length; k++){ \n                                     if
(ch[j].children[k].name===\"$is_working\"){\n
msg.payload.tags.push({'id': ch[j].children[k].id, 'timestamp':msg.csv.ts, 'value':

```

Подпись и дата

Инв. № дубл.

Взам. инв. №

Подпись и дата

Инв. № подл.


```

    ]
  ]
},
{
  "id": "c4817ac1.0ef6b8",
  "type": "debug",
  "z": "45305b0.cbb40a4",
  "name": "ftp",
  "active": true,
  "tosidebar": true,
  "console": false,
  "tostatus": false,
  "complete": "true",
  "targetType": "full",
  "x": 1090,
  "y": 80,
  "wires": []
},
{
  "id": "3456d46f.74ae6c",
  "type": "switch",
  "z": "45305b0.cbb40a4",
  "name": "",
  "property": "payload.tags.length",
  "propertyType": "msg",
  "rules": [
    {
      "t": "gt",
      "v": "0",
      "vt": "str"
    },
    {
      "t": "else"
    }
  ],
  "checkall": "false",
  "repair": false,
  "outputs": 2,
  "x": 1490,
  "y": 480,
  "wires": [
    "91f92921.ff5b18",
    "ceae6f7e.d56ba"
  ],
  []

```

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					45

```

    ]
  },
  {
    "id": "9e55c0dd.65d66",
    "type": "function",
    "z": "45305b0.cbb40a4",
    "name": "FTPRequest",
    "func":      "//msg.filename      =      'test.csv';\n//msg.localFilename      =
'/home/saper/'+msg.payload.name;//'/home/saper/test1.csv';\nmsg.filename      =
'/#recycle/Chelyabinsk/receive/';\n//msg.localFilename      =
msg.filename.replace(/^.*[\\|\\|\\|\\|\\|\\|/],
');//'/home/saper/test1.csv';\nmsg.password='' \nmsg.operation      =      \"list\";\nreturn
msg;\",
    "outputs": 1,
    "noerr": 0,
    "x": 110,
    "y": 260,
    "wires": [
      [
        "b2bdcecc.1d465"
      ]
    ]
  },
  {
    "id": "8684c859.888ec8",
    "type": "file",
    "z": "45305b0.cbb40a4",
    "name": "",
    "filename": "",
    "appendNewline": true,
    "createDir": false,
    "overwriteFile": "delete",
    "encoding": "none",
    "x": 1370,
    "y": 140,
    "wires": [
      []
    ]
  },
  {
    "id": "b2533155.5218",
    "type": "debug",
    "z": "45305b0.cbb40a4",
    "name": "Agent config",
    "active": true,
    "tosidebar": true,

```

Име. № подл.	
Подпись и дата	
Взам. име. №	
Име. № дубл.	
Подпись и дата	

Изм.	Лист	№ документа	Подпись	Дата	
------	------	-------------	---------	------	--

```

"console": false,
"toaststatus": false,
"complete": "payload",
"targetType": "msg",
"x": 730,
"y": 560,
"wires": []
},
{
  "id": "c3af6bef.2d61d8",
  "type": "function",
  "z": "45305b0.cbb40a4",
  "name": "FilterAlreadyDone",
  "func": "let filesToProcess = msg.payload;\nlet maxFiles=20;\n\nmsg.debug =
[];\nlet processedFiles = flow.get(\"processedFiles\");\nmsg.payload=[];\nlet now =
(new Date()).getTime();\n//node.debug(\"filesToProcess.length = \" +
filesToProcess.length);\n//msg.debug.push({\"t\": \"filesToProcess.length = \" +
filesToProcess.length});\nfor (let i = 0; i < filesToProcess.length; i++){
\n//msg.debug.push({\"t\": \"001\"});\n  let found = false;\n  let minIndex = -1;\n  if (processedFiles){\n    let n = maxFiles;\n    if (processedFiles.length <
maxFiles) n = processedFiles.length;\n    //msg.debug.push({\"t\": \"002 n=\\\"+n});\n    for (let j = 0; j < n; j++){
\n      if (processedFiles[j].name ===
filesToProcess[i].name){\n        //msg.debug.push({\"t\": \"003 found i=\\\"+i});\n        found = true;\n        //break;\n      }\n      if (minIndex < 0
|| processedFiles[j].time < processedFiles[minIndex].time){\n        minIndex
= j;\n      }\n    }\n    if (!found){\n      msg.payload.push(filesToProcess[i]);\n      filesToProcess[i].time = now;\n      if
(!processedFiles) processedFiles = [];\n      if (processedFiles.length >=
maxFiles){\n        \n        if (minIndex < maxFiles && minIndex >= 0){\n          //msg.debug.push({\"t\": \"004 found=\\\"+found+\\\" i=\\\"+i + \\\" minIndex=\\\"+minIndex + \\\"
val=\\\"+filesToProcess[i].name});\n          processedFiles[minIndex] =
filesToProcess[i];\n        }\n      }\n    }\n    processedFiles.push(filesToProcess[i]);\n  }\n}\n\nflow.set(\"processedFiles\", processedFiles);\n\n\nreturn msg;\",
  "outputs": 1,
  "noerr": 0,
  "x": 430,
  "y": 180,
  "wires": [
    [
      "2c3f2efd.6f5f42",
      "53aec2be.0bd26c"
    ]
  ]
},
{

```

Ине. № подл.	
Подпись и дата	
Взам. ине. №	
Ине. № дубл.	
Подпись и дата	

Изм.	Лист	№ документа	Подпись	Дата
------	------	-------------	---------	------

```

        "id": "ceae6f7e.d56ba",
        "type": "function",
        "z": "45305b0.cbb40a4",
        "name": "portion",
        "func": "if (msg.tagsToSend){\n            let n = 10;\n            if (n >
msg.tagsToSend.length) n = msg.tagsToSend.length;\n            msg.payload.tags =
msg.tagsToSend.slice(0, n); //copy to tags\n            msg.tagsToSend.splice(0, n); //remove
copied \n)\n\nmsg.method = \"POST\";\nmsg.headers['X-Agent-Token']='e0138e64-7729-
4045-95f2-7d6d5fb245cc\";\nmsg.headers['Content-
Type']='application/json\";\nmsg.headers['Cookie']=flow.get('Cookie');\nreturn
msg;\",
        "outputs": 1,
        "noerr": 0,
        "x": 1320,
        "y": 420,
        "wires": [
            [
                "3456d46f.74ae6c"
            ]
        ]
    },
    {
        "id": "53aec2be.0bd26c",
        "type": "debug",
        "z": "45305b0.cbb40a4",
        "name": "FilesToProcess",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "true",
        "targetType": "full",
        "x": 640,
        "y": 120,
        "wires": []
    },
    {
        "id": "a6f0be2c.d827a",
        "type": "function",
        "z": "45305b0.cbb40a4",
        "name": "AuthRequest",
        "func":
        /*\nPOST
'http://89.208.229.18/public/auth/authenticator/api/internalauth/auth'  \n\n--header
'Accept: application/json, application/*+json'  \n\n--header 'Authorization: Basic
dXNlckB3YTM6eWZNN3Q4dWN2QlJKVkZPdA=='  \n\n--header 'Cookie:
SESSION_TOKEN='eyJzZXNzaW9uSWQioiIOMmY2MTU3Ni1lM2Y1LTQzOWMtOGZiMS0wNzFjZWJlYWE1NDIiIl

```

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата


```

CJjdXJyZW50VXNlcklkIjoiOGUwMDQ0YTMtNmU5Zi00MWMzLWE5MTctYz4ZTBhODA4MjlmIn0=\";
JSESSIONID=FBD51668EAC3532F10DC91944ADBC907'\n\n*/\n\n\nmsg.method=\"POST\";\nmsg.hea
ders={};\nmsg.headers['Accept'] = \"application/json\";\nmsg.headers['Authorization']
= \"Basic \"+ msg.payload;\n//msg.headers['X-Agent-Token']=\"aa9c5088-95b4-4b7e-ade5-
8fc3a59c72dc\";\nmsg.headers['Content-Type']=\"application/json\";\nlet now = (new
Date()).getTime();\nmsg.payload={\n};\nreturn msg;\",
    \"outputs\": 1,
    \"noerr\": 0,
    \"x\": 210,
    \"y\": 560,
    \"wires\": [
      [
        \"7503ed67.b9de84\"
      ]
    ]
  },
  {
    \"id\": \"7503ed67.b9de84\",
    \"type\": \"http request\",
    \"z\": \"45305b0.cbb40a4\",
    \"name\": \"httpAuth\",
    \"method\": \"POST\",
    \"ret\": \"obj\",
    \"paytoqs\": false,
    \"url\":
      \"http://10.20.22.30/public/auth/authenticator/api/internalauth/auth\",
    \"tls\": \"\",
    \"persist\": true,
    \"proxy\": \"\",
    \"authType\": \"basic\",
    \"x\": 340,
    \"y\": 660,
    \"wires\": [
      [
        \"26e691eb.046fee\"
      ]
    ]
  },
  {
    \"id\": \"26e691eb.046fee\",
    \"type\": \"function\",
    \"z\": \"45305b0.cbb40a4\",
    \"name\": \"AgentRequest\",
    \"func\": \"msg.method=\"GET\";\nlet cookie = msg.headers['set-
cookie'];\nmsg.headers={};\nmsg.headers['X-Agent-Token']=\"e0138e64-7729-4045-95f2-
7d6d5fb245cc\";\nmsg.headers['Content-Type']=\"application/json\";\nlet now = (new

```

Ине. № дубл.	Подпись и дата
Взам. ине. №	Подпись и дата
Ине. № подл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					49

```
Date()).getTime());\nmsg.timezone=5;\nif (cookie && cookie.length > 0) {\n
flow.set(\"Cookie\",cookie[0]);\n msg.headers['Cookie']=cookie[0];\n}\nreturn
msg;\",
```

```
    \"outputs\": 1,
    \"noerr\": 0,
    \"x\": 460,
    \"y\": 580,
    \"wires\": [
      [
        \"c0df7042.000a7\"
      ]
    ]
  },
  {
    \"id\": \"c246608.b9a70a\",
    \"type\": \"switch\",
    \"z\": \"45305b0.cbb40a4\",
    \"name\": \"\",
    \"property\": \"payload\",
    \"propertyType\": \"msg\",
    \"rules\": [
      {
        \"t\": \"hask\",
        \"v\": \"agent\",
        \"vt\": \"str\"
      }
    ],
    \"checkall\": \"true\",
    \"repair\": false,
    \"outputs\": 1,
    \"x\": 850,
    \"y\": 400,
    \"wires\": [
      [
        \"9e55c0dd.65d66\"
      ]
    ]
  },
  {
    \"id\": \"d306a2ec.1a3fe\",
    \"type\": \"advanced-ftp-config\",
    \"z\": \"\",
    \"host\": \"92.255.247.230\",
    \"port\": \"\",
    \"secure\": false,
    \"secureOptions\": \"\",
```

Ине. № подл.	Подпись и дата	Взам. ине. №	Ине. № дубл.	Подпись и дата

Изм.	Лист	№ документа	Подпись	Дата	Лист
					50

```

        "user": "mincifra",
        "connTimeout": "",
        "pasvTimeout": "",
        "keepalive": "",
        "name": ""
    }
]

```

Ине. № подл.	Подпись и дата				Ине. № дубл.	Подпись и дата
	Взам. ине. №					
	Ине. № дубл.					
Изм.	Лист	№ документа	Подпись	Дата		
					Лист	
					51	