

ООО «НТК Интерфейс»

Дополнительная программа обработки и рассылки событий ПО сервера

2020

Содержание

| | |
|---|----|
| Принятые обозначения и сокращения | 3 |
| 1. ВВЕДЕНИЕ..... | 4 |
| 2. Установка программы..... | 5 |
| 2.1. Настройка внешней задачи | 6 |
| 2.2. Автономный запуск программы..... | 7 |
| 2.3. Обновление программы | 8 |
| 3. Создание и настройка задачи | 9 |
| 3.1. Опции EmailHandler | 11 |
| 3.2. Опции FileHandler..... | 12 |
| 3.3. Опции HttpHandler..... | 12 |
| 3.4. Опции TelegramHandler | 13 |
| 3.5. Шаблоны сообщений..... | 14 |

Принятые обозначения и сокращения

| Термин | Полная формулировка |
|--------|---|
| АПС | аварийно - предупредительная сигнализация |
| КП | контролируемый пункт |
| ЛКМ | левая кнопка мыши |
| ОИК | оперативно - информационный комплекс |
| ПК | персональный компьютер |
| ПКМ | правая кнопка мыши |
| ПО | программное обеспечение |
| СУРБД | система управления реляционными базами данных |
| ТИИ | телеизмерение интегральное |
| ТИТ | телеизмерение текущее |
| ТС | телесигнал |
| ТУ | телеуправление |
| УСПИ | устройство сбора и передачи информации |
| УТМ | устройство телемеханики |
| ЯРД | язык расчётчика данных |

1. ВВЕДЕНИЕ

Задача предназначена для обработки поступающих событий ПО сервера "ОИК Диспетчер НТ".

Во время работы программа с определенным периодом времени проверяет наличие новых событий на сервере «ОИК Диспетчер НТ».

В случае обнаружения новых событий вызывается один или несколько обработчиков, каждый из которых выполняет заданные действия с набором поступивших событий.

Типы доступных обработчиков событий:

- EmailHandler - e-mail рассылка сообщений журнала событий на заданные адреса;
- FileHandler - запись сообщений журнала событий в текстовый файл;
- HttpHandler - выполняет HTTP-запрос (GET или POST) на заданный сервер;
- TelegramHandler - с помощью бота Telegram рассылает сообщения журнала событий заданным пользователям или группам. Обратите внимание, данный обработчик по умолчанию не работает на территории РФ, т.к. сеть для работы с API Telegram заблокирована Роскомнадзором.;

Программа может работать как внешняя задача ПО сервера «ОИК Диспетчер НТ» или автономно на любом удаленном компьютере. Конфигурирование программы происходит с помощью ручного изменения файлов в формате json.

Исходные коды программы являются открытыми, полная информация о программе доступна по ссылке <https://github.com/alexeid1337/Iface.Oik.EventDispatcher>

Задача состоит из одного установочного каталога в котором располагается:

- исполняемый файл задачи;
- конфигурации обработчиков.

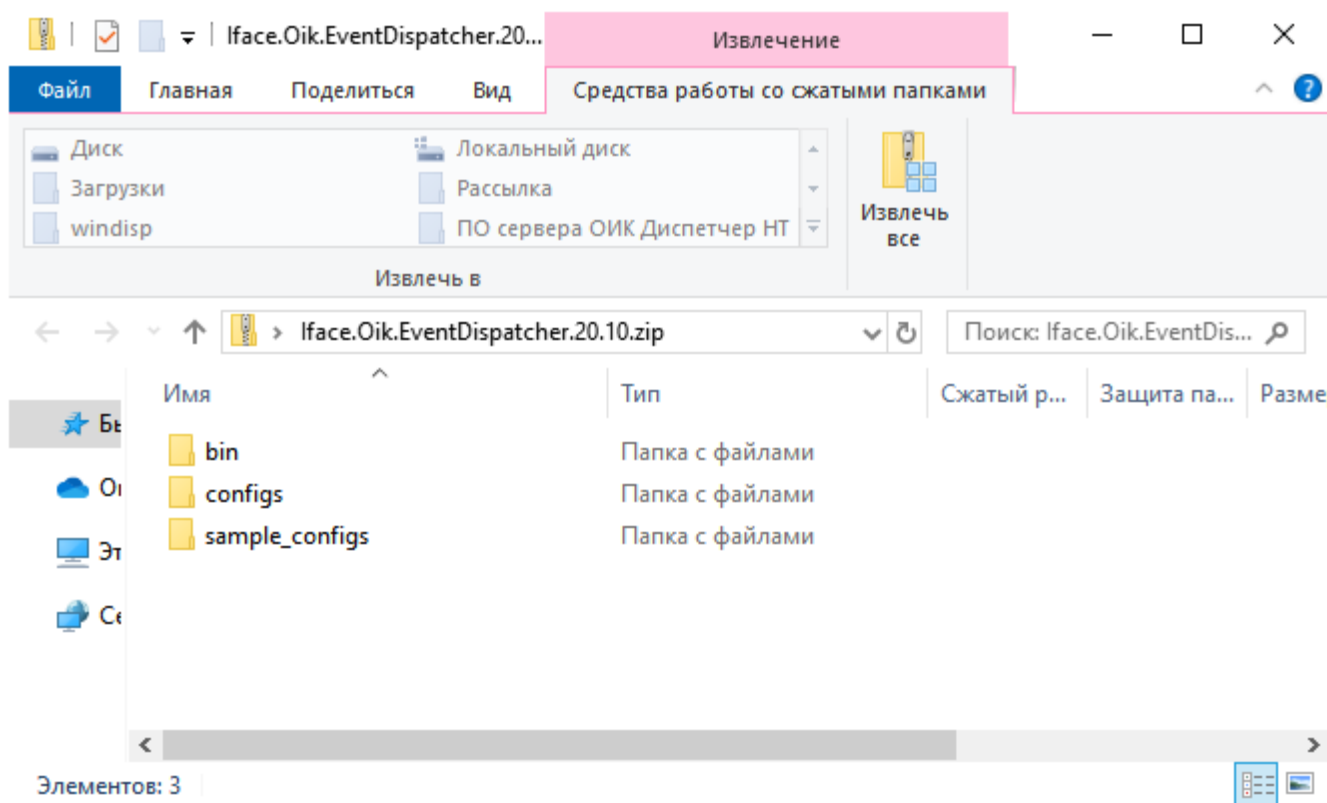
2. Установка программы

Для установки программы необходимо скачать установочный архив

Iface.Oik.EventDispatcher.zip с сайта НТК Интерфейс.

Программа может работать как внешняя задача ПО сервера «ОИК Диспетчер НТ» или автономно на любом удаленном компьютере.

Установочный архив необходимо распаковать и в зависимости от выбранного варианта работы (автономный или внешняя задача) поместить каталог по рабочему пути. При автономной работе каталог может располагаться по произвольному пути, при работе в качестве внешней задачи каталог программы должен располагаться в корневом каталоге установки ПО сервера "ОИК Диспетчер НТ".



Установочный архив состоит из трех каталогов:

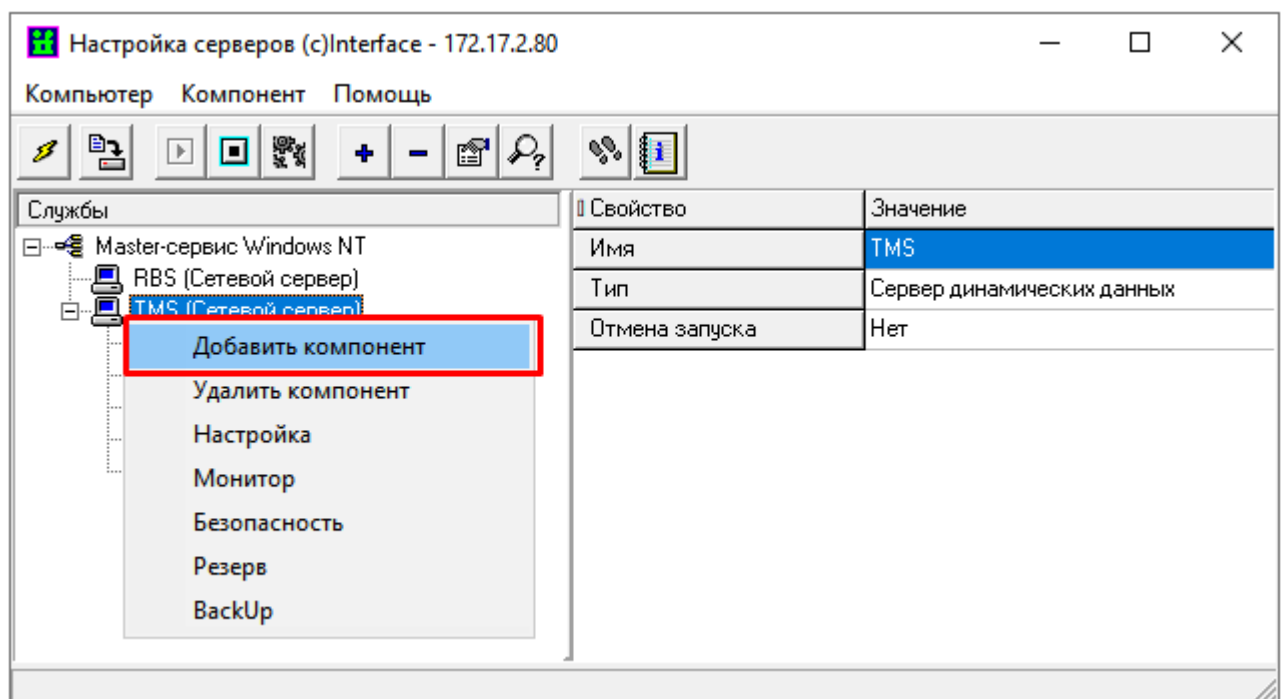
- bin - каталог содержит исполняемый файл программы и необходимые библиотеки для соединения с сервером "ОИК Диспетчер НТ";
- configs - каталог с конфигурациями обработчиков, по умолчанию каталог пустой;
- sample_configs - каталог с примерами конфигураций доступных обработчиков.

2.1. Настройка внешней задачи

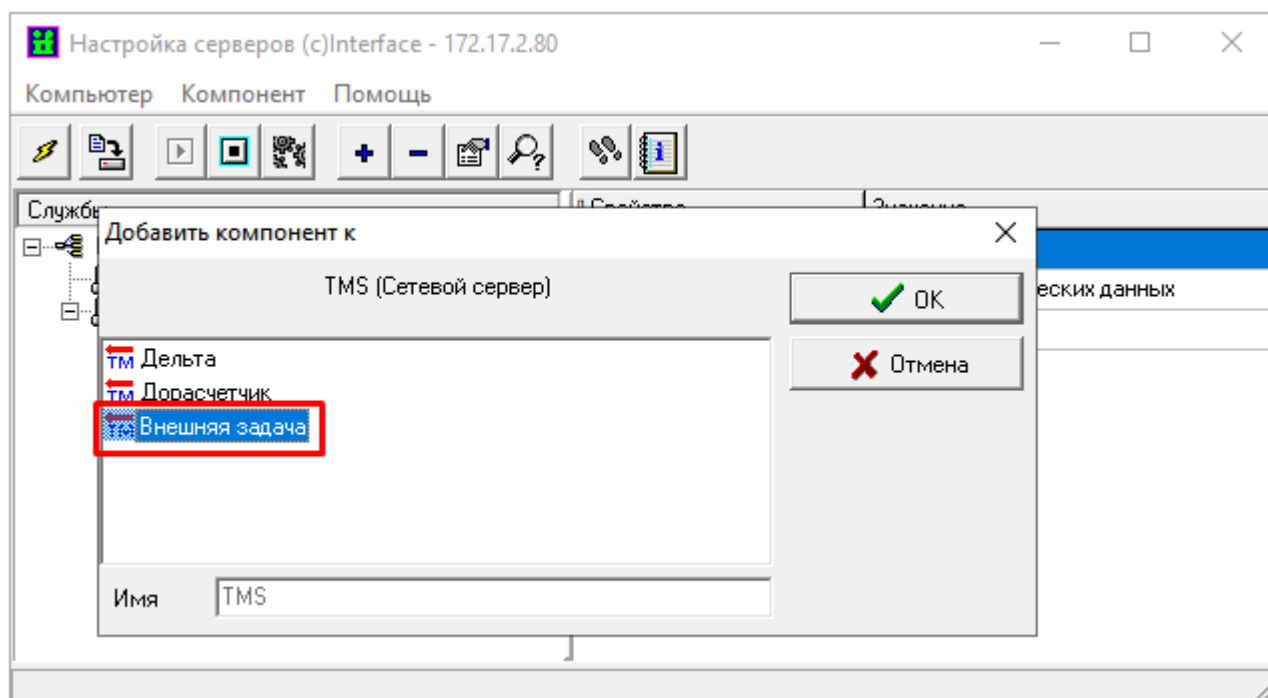
Установочный архив программы **Iface.Oik.EventDispatcher.zip** необходимо распаковать и поместить в корневой каталог установки ПО сервера. По умолчанию данный каталог находится по пути:

- для версий 2.2,2.3 C:\ProgramFiles(x86)\InterfaceSSH\Server;
- для версии 3.0 C:\Program Files\InterfaceSSH\Server64\PortCore.

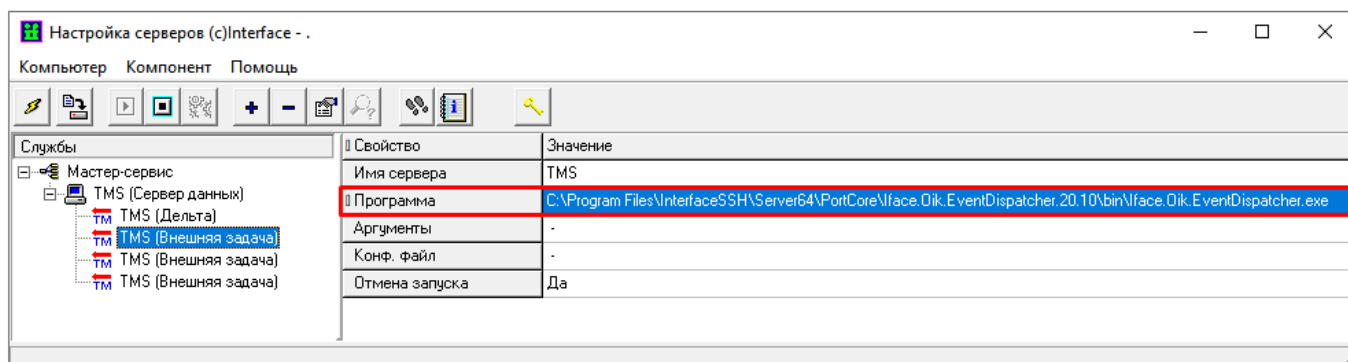
Для настройки необходимо окне настройки серверов встать на уровень TMS сервера, нажать ПКМ, в выпадающем окне выбрать поле "Добавить компонент".




В появившемся окне выбрать пункт "Внешняя задача" и нажать "ОК".



В добавленной строке компонента необходимо указать путь к исполняемому файлу. В версиях 2.2, 2.1 доступен выбор пути средствами обзора папок компьютера, в версиях 2.3, 3.0 путь необходимо прописывать вручную.



После указания пути обязательно нажать кнопку сохранить изменения . Параметр отмены запуска должен быть установлен в положение "Нет", в случае установки данного параметра в "Да" задача не будет запускаться при старте сервера.

При данной настройке программа автоматически запускается при запуске ПО сервера "ОИК Диспетчер НТ".

2.2. Автономный запуск программы

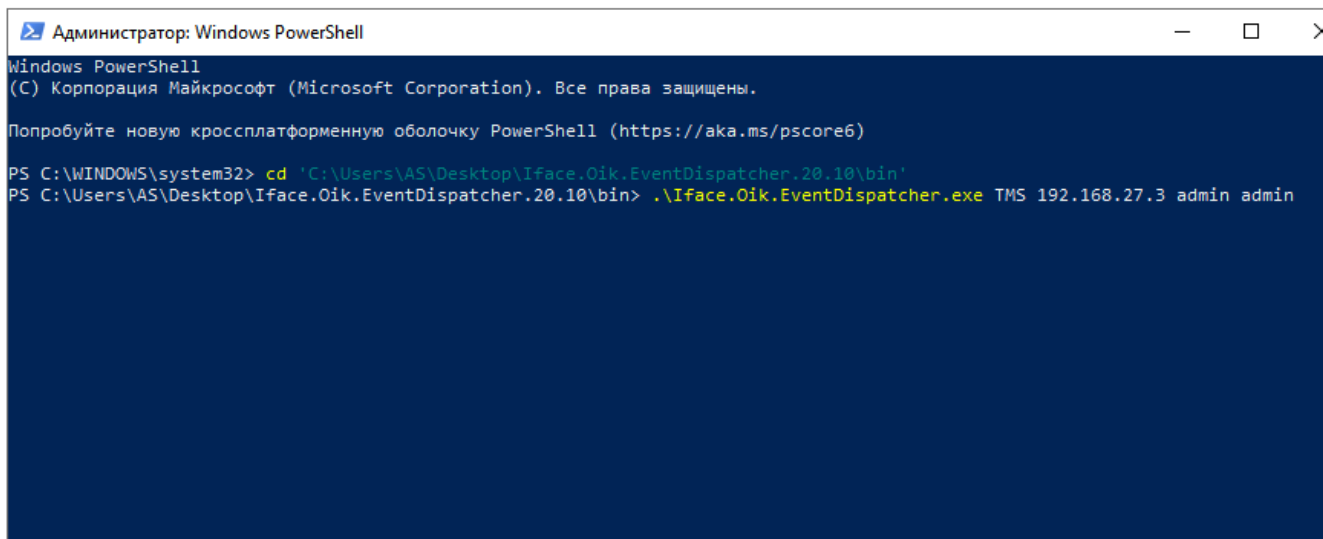
Для автономного запуска установочный архив программы **Iface.Oik.EventDispatcher.zip** необходимо распаковать по произвольному пути.

Запуск задачи производится из командной строки с вводом аргументов для подключения к серверу.

Iface.Oik.EventDispatcher сервер_динамических_данных компьютер имя_пользователя
пароль

Например:

Iface.Oik.EventDispatcher TMS 192.168.27.3 admin admin

A screenshot of a Windows PowerShell terminal window titled "Администратор: Windows PowerShell". The window has a dark blue background. The text inside shows the following commands and output:

```
Windows PowerShell
(С) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\WINDOWS\system32> cd 'C:\Users\AS\Desktop\Iface.Oik.EventDispatcher.20.10\bin'
PS C:\Users\AS\Desktop\Iface.Oik.EventDispatcher.20.10\bin> .\Iface.Oik.EventDispatcher.exe TMS 192.168.27.3 admin admin
```

2.3. Обновление программы

При обновлении программы требуется обновить только каталог bin, не изменяя имеющиеся файлы конфигураций в каталоге configs.

3. Создание и настройка задачи

Конфигурация каждого экземпляра обработчика хранится в папке `samples` в отдельном файле с расширением `.json`, при этом общее количество файлов не ограничено (один обработчик может быть задействован в нескольких конфигурациях, например, с разным фильтром событий). Для конфигурации файлов используется распространенный формат JSON.

Структура конфигурационного файла состоит из трех основных блоков:

```
{
  "Filter": {
    ...
  },
  "Handler" : "имя_обработчика",
  "Options": {
    ...
  }
}
```

– Фильтр

Фильтр является опциональным, может не использоваться, тогда будут обработаны все поступившие события.

Если фильтр задан, можно опционально фильтровать по типам, важности и адресам событий. Пример полного задания фильтра (допускается использование любого количества параметров фильтра):

```
"Filter": {
  "Importances": [2, 3],
  "Types": ["StatusChange", "Alarm"],
  "Statuses": ["20:1:1"],
  "Analog": ["20:1:1..99"]
}
```

Уровни важности событий перечисляются через запятую для ключа `Importances`, где:

0 - оперативного состояния

1 - предупредительные 2

2 - предупредительные 1

3 - аварийные

Типы события перечисляются через запятую для ключа Types, где:

StatusChange - изменение сигналов

Alarm - срабатывание и снятие уставок

Control - команды управления

Acknowledge - квитирование оператором

ManualStatusSet - ручная установка сигналов

ManualAnalogSet - ручная установка измерений

Extended - сообщения

Адреса сигналов перечисляются через запятую для ключа Statuses, при этом можно указать как единичный адрес (например, 0:1:1), так и группу адресов (например, 0:1:1..10, в этом случае будут использованы сигналы 0:1:1, 0:1:2, ..., 0:1:10).

Адреса измерений перечисляются через запятую для ключа Analogs, аналогично сигналам.

Обработчик

На данный момент доступны следующие обработчики:

EmailHandler отправляет сообщения на заданные адреса электронной почты

FileHandler сохраняет записи в текстовый файл

HttpHandler выполняет HTTP-запрос (GET или POST) на заданный сервер

TelegramHandler с помощью бота Telegram отправляет сообщения заданным пользователям или группам. Обратите внимание, данный обработчик по умолчанию не заработает на территории РФ, т.к. сеть для работы с API Telegram заблокирована Роскомнадзором.

Опции

Каждый обработчик имеет свой набор опций для работы, подробное описание опций для каждого обработчика представлены в отдельных разделах.

3.1. Опции EmailHandler

Host - имя почтового сервера SMTP (например, "smtp.gmail.com") - обязательно

Port - порт почтового сервера SMTP (например, 465) - обязательно

UseSsl - использовать безопасное соединение SSL/TLS (true или false) - опционально, по умолчанию false

Login - имя пользователя почтового сервера - опционально

Password - пароль пользователя - опционально

From - имя отправителя письма - обязательно

FromEmail - адрес отправителя письма (для популярных сервисов обычно совпадает с именем пользователя) - обязательно

SendTo - один или несколько адресов получателей писем (например, ["addr1@gmail.com", "addr2@yandex.ru"]) - обязательно

IsHtml - является ли текст сообщения HTML или простым текстом (true или false) - опционально, по умолчанию false

Subject - заголовок письма, разрешены специальные значения шаблонов для подстановки значений события (см. ниже) - опционально, по умолчанию "Новые события ОИК Диспетчер НТ"

Body - текст письма, разрешены специальные значения шаблонов для подстановки значений события (см. ниже) - опционально, по умолчанию обычный текст события, описанный ниже

BatchEvents - группировать события в одном письме или нет (true или false). В случае появления ряда событий, близких по времени, можно сгруппировать события в одном письме, в противном случае отправится много писем одновременно.

Пример:

```
"options": {  
  "host": "smtp.mail.ru",  
  "port": 465,  
  "useSsl": true,
```

```
"login": "<login>",  
"password": "<password>",  
"from": "События «ОИК Диспетчер НТ»",  
"fromEmail": "<login>",  
"sendTo": ["<email-1>", "<email-2>", "<email-3>"],  
"subject": "Событие {importance}: {name}, {state}"  
}
```

Ссылки на настройки популярных почтовых сервисов:

Gmail - <https://support.google.com/mail/answer/7126229?hl=ru>

Яндекс - <https://yandex.ru/support/mail/mail-clients.html>

mail.ru - <https://help.mail.ru/mail/mailer/popsmtpt>

3.2. Опции FileHandler

FilePath - путь к файлу для записи событий - обязательно

Body - текст строки для записи, разрешены специальные значения шаблонов для подстановки значений события (см. ниже) - опционально, по умолчанию обычный текст события, описанный ниже

Пример:

```
"options": {  
  "filePath": "D:/my_folder/events.txt",  
  "body": "Новое событие: {time:HH:mm:ss}, {importance}"  
}
```

3.3. Опции HttpHandler

Method - тип запроса (GET или POST) - обязательно

Url - адрес запроса (например: https://my-api.ru/event_dispatcher/) - обязательно

Body - тело запроса, разрешены специальные значения шаблонов для подстановки значений события (см. ниже) - опционально, по умолчанию обычный текст события, описанный ниже

Пример:

```
"options" : {  
  "method": "get",  
  "url": "http://localhost:3000/?imp={importanceId}",  
  "body" : "{ \"event\": \"{defaultBody}\" }"  
}
```

3.4. Опции TelegramHandler

Обратите внимание, данный обработчик по умолчанию не заработает на территории РФ, т.к. сеть для работы с API Telegram заблокирована Роскомнадзором.

BotToken - токен бота Telegram (например, 987654321:abcdefg123_hijkQWZ) - обязательно

ChatIds - один или несколько ID пользователей или групп для отправки сообщений (например, [823999774, 152687839]) - обязательно

Body - текст сообщения, разрешены специальные значения шаблонов для подстановки значений события (см. ниже) - опционально, по умолчанию обычный текст события, описанный ниже

Пример:

```
"options": {  
  "botToken": "987654321:abcdefg123_hijkQWZ",  
  "chatIds": [823999774, 152687839],  
  "body": "Новое событие: {time:HH:mm:ss}, {importance}"  
}
```

3.5. Шаблоны сообщений

Внутри текста сообщений допускается использовать специальные значения шаблонов для подстановки реальных значений события

{defaultBody} - сообщение события по умолчанию

{importanceId} - номер важности события, где 0 - оперативного состояния; 1 - предупредительные 2; 2 - предупредительные 1; 3 - аварийные

{importance} - краткий текст важности события, где ОС - оперативного состояния; ПС2 - предупредительные 2; ПС1 - предупредительные 1; АС - аварийные

{name} - наименование события, например, наименование сигнала команды ТУ, наименование измерения уставки

{state} - состояние события, например, текст переключения сигнала, взведенность/снятие для уставки

{type} - тип события, например, команда ТУ, наименование уставки

{username} - имя пользователя, выдавшего команду - может быть пустым

{tmAddr} - адрес сигнала или измерения - может быть пустым

{time} - дата и время события

Дополнительно есть возможность вывести конкретные фрагменты даты и времени события, например, только дату, только время, только час, только год и т.д. Для этого используется шаблон {time:<формат_даты_и_времени>}. Используется стандартный формат даты и времени, описанный, например, здесь - <https://docs.microsoft.com/ru-ru/dotnet/standard/base-types/custom-date-and-time-format-strings>

Примеры строк с шаблонами, и во что они могут трансформироваться при поступлении события:

Событие {importance}: {name}, {state}

Событие ПС2: ПС Южная В-220, Включен

Внимание, обнаружено новое событие: {time:HH:mm:ss}, {importance}, {name}, {state}, {type}, {username} Просьба принять меры!

Внимание, обнаружено новое событие: 21:35:12, ОС, ПС Северная ВЛ-35 кВ Северная Главная В-35, Команда ТУ, Выключатель, Петров В. И. Просьба принять меры

