

МЕТОДЫ ЗАПРОСОВ СОБЫТИЙ
ЖУРНАЛОВ ПО REST API

Версия 1.0

Листов 27

СОДЕРЖАНИЕ

1. Общие сведения	3
2. Настройка компьютеров и модуля для запроса событий	4
2.1. Настройка сервера (компьютера с VideoNet).....	4
2.2. Настройка клиента (компьютера с внешней системой)	8
2.3. Настройка модуля для запроса событий в VideoNet	11
3. Методы запросов событий.....	13
3.1. Общие сведения.....	13
3.2. Метод запроса событий журнала «Журнал. СКУД»	14
Приложение Тестовое приложение.....	26
Исходный код на Python	26
Исходный код на C#.....	26

1. ОБЩИЕ СВЕДЕНИЯ

1.1. VideoNet имеет возможность передавать сторонним приложениям (далее – Внешняя система) с помощью REST API сообщения о событиях системы, протоколируемых в журнале СКУД.

1.1.1. Для запроса сообщений, Внешняя система должна представлять собой REST API клиент (описание настройки см. в разделе 2.2), для которого VideoNet будет выступать в качестве сервера (описание настройки см. в разделе 2.1).

1.1.2. VideoNet отправляет в ответ на запрос данные о сообщениях, которые затем могут быть обработаны Внешней системой.

1.2. Передача сообщений о событиях системы VideoNet реализована с помощью модуля для запроса событий по REST API (далее – модуль).

1.2.1. Компьютер, к которому подключён модуль, отправляет данные о запрошенных событиях журнала в ответ на метод GET.

1.2.2. К одному компьютеру может быть подключён только один модуль.

1.2.3. Для работы с модулем на компьютере VideoNet должна быть лицензия «Модуль запроса событий по REST API» (SM-LOG-API).

1.2.4. Существует возможность заблокировать модуль, в результате чего отправка ответов на запросы Внешней системы производиться не будет.

1.3. В случае большого количества запросов к VideoNet может возникнуть ситуация, при которой обработка запросов не будет возможна некоторое время. В таком случае в ответ на запрос VideoNet вернёт код ошибки сервера 503. Запрос к VideoNet можно будет направить повторно через некоторое время.

2. НАСТРОЙКА КОМПЬЮТЕРОВ И МОДУЛЯ ДЛЯ ЗАПРОСА СОБЫТИЙ

Внимание! Выполнение настроек, описанных в разделах 2.1 и 2.2, необходимо только для работы с использованием HTTPS. Если Вы хотите использовать HTTP – отключите проверку SSL-сертификата во внешней системе и выполните настройки, описанные в разделе 2.3.

Корректность выполнения настроек можно проверить при помощи тестового приложения (см. Приложение к документу).

2.1. Настройка сервера (компьютера с VideoNet)

2.1.1. На компьютере, на котором установлен VideoNet, запустите Windows PowerShell от имени администратора.

2.1.2. Введите следующую команду в Windows PowerShell для выпуска на сервере SSL-сертификата (далее – Сертификат):

```
$cert = New-SelfSignedCertificate -certstorelocation cert:\localmachine\my -  
dnsname Желаемое DNS имя
```

Например, если DNS должен быть test.videonet.ru, то команда должна выглядеть следующим образом: \$cert = New-SelfSignedCertificate -certstorelocation cert:\localmachine\my -dnsname test.videonet.ru

2.1.3. Введите следующую команду для создания пароля:

```
$pwd = ConvertTo-SecureString -String "Желаемый пароль" -Force -  
AsPlainText
```

Пароль должен указываться в кавычках.

*Пример команды: \$pwd = ConvertTo-SecureString -String "112233" -Force -
AsPlainText*

2.1.4. Введите следующую команду для получения информации о сертификате:

```
$certpath = "Cert:\localMachine\my\$(($cert.Thumbprint))"
```

2.1.5. Введите следующую команду для экспорта созданного сертификата в желаемую папку:

`Export-PfxCertificate -Cert $certpath -FilePath Полный путь папки и имя файла.pfx -Password $pwd`

Например, для сохранения сертификата с именем «cert» в папку `F:\certificate` команда должна выглядеть следующим образом: `Export-PfxCertificate -Cert $certpath -FilePath F:\certificate\cert.pfx -Password $pwd`

В результате выполнения команды появится информация об экспорте (см. Рисунок 1). В указанную папку будет сохранён файл сертификата.

```
PS C:\Windows\system32> $cert = New-SelfSignedCertificate -certstorelocation cert:\localmachine\my -dnsname test.videonet.ru
PS C:\Windows\system32> $pwd = ConvertTo-SecureString -String "112233" -Force -AsPlainText
PS C:\Windows\system32> $certpath = "Cert:\localMachine\my\${$cert.Thumbprint}"
PS C:\Windows\system32> Export-PfxCertificate -Cert $certpath -FilePath D:\Work\cert.pfx -Password $pwd

Каталог: D:\Work

Mode                LastWriteTime         Length Name
----                -
-a-----         16.05.2024   11:15           2661 cert.pfx
```

Рисунок 1

2.1.6. После сохранения файла сертификата необходимо получить его отпечаток для добавления в конфигурационный файл VideoNet. Для этого введите следующую команду в Windows PowerShell:

`$cert.Thumbprint`

2.1.6.1. Скопируйте отобразившийся отпечаток сертификата (см. Рисунок 2).

```
PS C:\Windows\system32> $cert.Thumbprint
6E1E62D39F2FC3C553967276F47A6EB61EC7C72B
```

Рисунок 2

2.1.7. Откройте файл «VideoNet.config», находящийся в папке, в которую установлен VideoNet (по умолчанию: `C:\Program Files (x86)\SKYROS\VideoNet 9`).

2.1.8. Добавьте в файл в секцию «Behaviour» следующую строку (см. Рисунок 3) и сохраните его:

`<ETCRestServiceThumbprint>Отпечаток сертификата</ETCRestServiceThumbprint>`

Например:

```
<ETCRestServiceThumbprint>42a800962e93c245d45500da08ab3f26e701ff8d</E
TCRestServiceThumbprint>
```

```
<ETCRestServiceThumbprint>42a800962e93c245d45500da08ab3f26e701ff8d</ETCRestServiceThumbprint>
</Behaviour>
<!-- Отладочные параметры -->
</VideoNetConfig>
```

Рисунок 3

2.1.9. Откройте файл «hosts», находящийся по адресу «C:\Windows\System32\drivers\etc\hosts» и добавьте в него IP-адрес компьютера и DNS имя, которое было указано при выполнении команды, описанной в п. 2.1.2 (см. Рисунок 4).

```
# localhost name resolution is handled within DNS itself.
# 127.0.0.1      localhost
# ::1          localhost
192.168.5.99 test.videonet.ru
```

Рисунок 4

2.1.10. После этого запустите сохранённый файл сертификата (например, «cert.pfx»). Откроется «Мастер импорта сертификатов» (см. Рисунок 5).

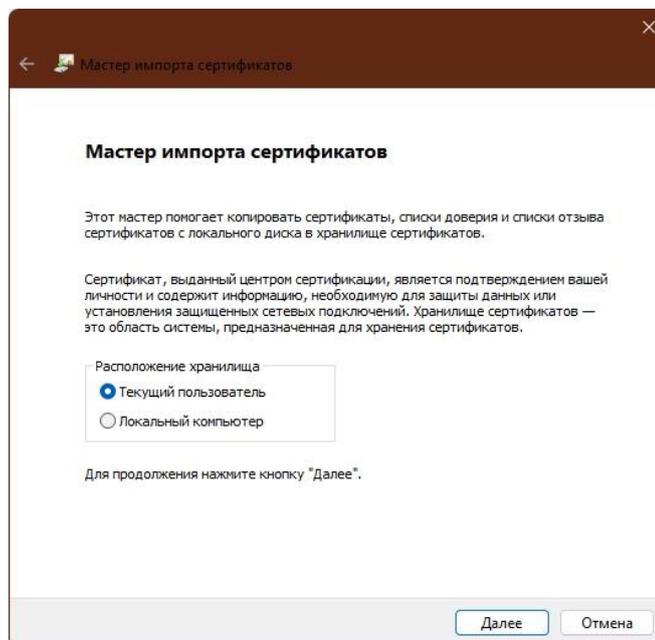


Рисунок 5

2.1.11. Выберите пункт «Локальный компьютер» и нажмите «Далее».

2.1.12. Проверьте имя файла в строке «Имя файла» и нажмите «Далее».

2.1.13. Введите пароль, указанный при выполнении команды, описанной в п. 2.1.3, в поле «Пароль».

2.1.14. Включите параметр «Пометить этот ключ как экспортируемый...» (см. Рисунок 6) и нажмите «Далее».

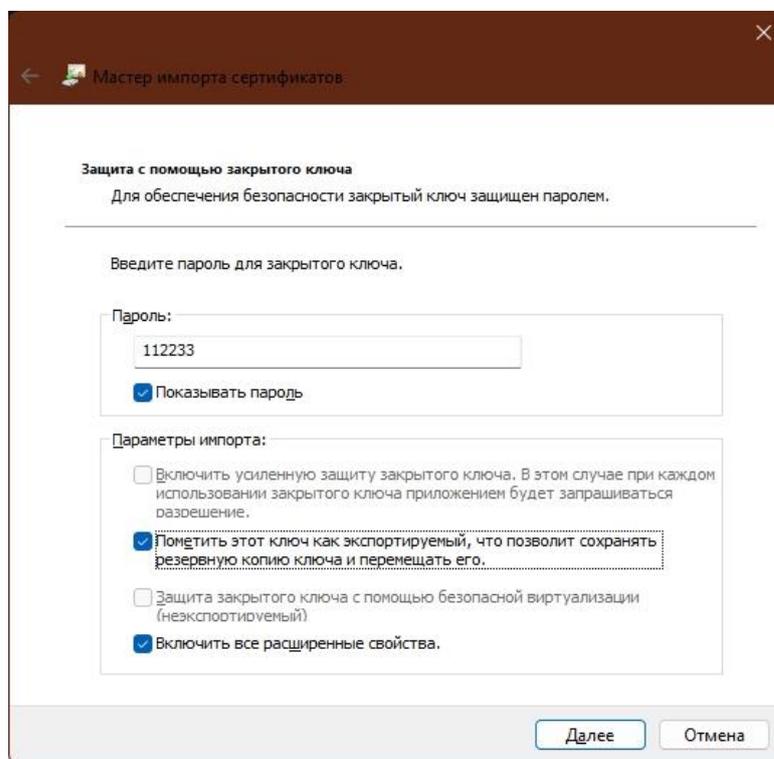


Рисунок 6

2.1.15. Выберите пункт «Поместить все сертификаты в следующее хранилище», а затем нажмите кнопку «Обзор». Откроется окно «Выбор хранилища сертификата».

2.1.16. В окне «Выбор хранилища сертификата» выберите пункт «Доверенные корневые центры сертификации» и нажмите «ОК» (см. Рисунок 7). Нажмите «Далее» в окне мастера импорта сертификатов.

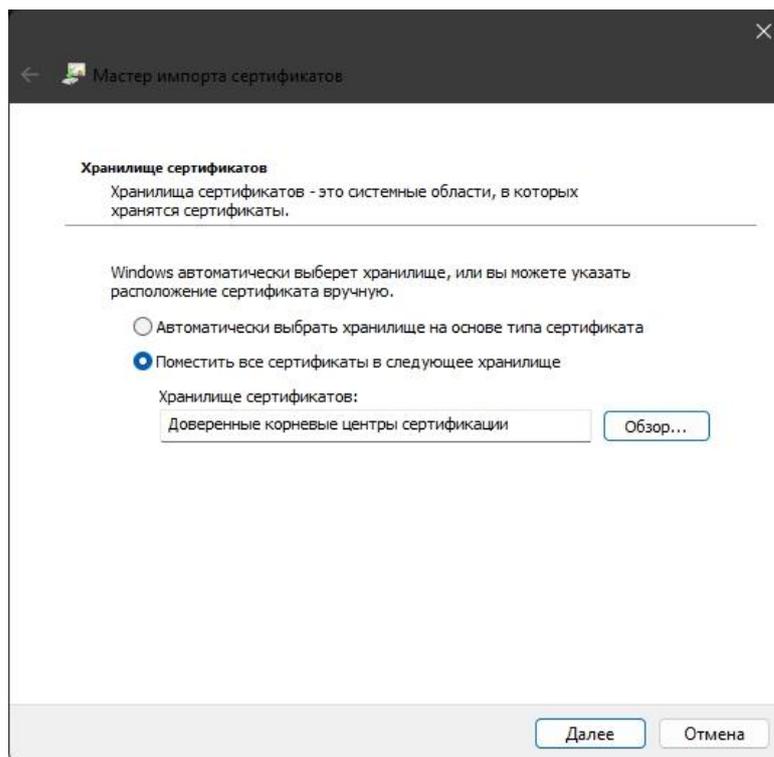


Рисунок 7

2.1.17. На шаге «Завершение мастера импорта сертификатов» нажмите «Готово». Отобразится сообщение «Импорт успешно выполнен».

На этом настройка сервера будет завершена, и можно будет запускать VideoNet.

2.2. Настройка клиента (компьютера с внешней системой)

2.2.1. Перенесите файл сертификата (с расширением «.pfx»), созданный на сервере (см. п. 2.1.5) на компьютер-клиент.

2.2.2. После этого запустите файл сертификата (например, «cert.pfx»). Откроется «Мастер импорта сертификатов» (см. Рисунок 5).

2.2.3. Выберите пункт «Локальный компьютер» и нажмите «Далее».

2.2.4. Проверьте имя файла в строке «Имя файла» и нажмите «Далее».

2.2.5. Введите пароль, указанный при выполнении команды, описанной в п. 2.1.3, в поле «Пароль».

2.2.6. Включите параметр «Пометить этот ключ как экспортируемый...» (см. Рисунок 6) и нажмите «Далее».

2.2.7. Выберите пункт «Поместить все сертификаты в следующее хранилище», а затем нажмите кнопку «Обзор». Откроется окно «Выбор хранилища сертификата».

2.2.8. В окне «Выбор хранилища сертификата» выберите пункт «Доверенные корневые центры сертификации» и нажмите «ОК» (см. Рисунок 7). Нажмите «Далее» в окне мастера импорта сертификатов.

2.2.9. На шаге «Завершение мастера импорта сертификатов» нажмите «Готово». Отобразится сообщение «Импорт успешно выполнен».

2.2.10. Повторите действия по п. 2.2.2-2.2.9, но в окне «Выбор хранилища сертификата» (см. п. 2.2.8) выберите пункт «Личное».

2.2.11. Откройте файл «hosts», находящийся по адресу «C:\Windows\System32\drivers\etc\hosts» и добавьте в него IP-адрес и DNS имя сервера, которое было указано при выполнении команды, описанной в п. 2.1.2 (см. Рисунок 4).

2.2.12. Запустите Windows PowerShell от имени администратора.

2.2.13. Выполните следующую команду, чтобы установить расширение Chocolatey:

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object  
System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```

2.2.14. Выполните следующую команду, чтобы установить OpenSSL.

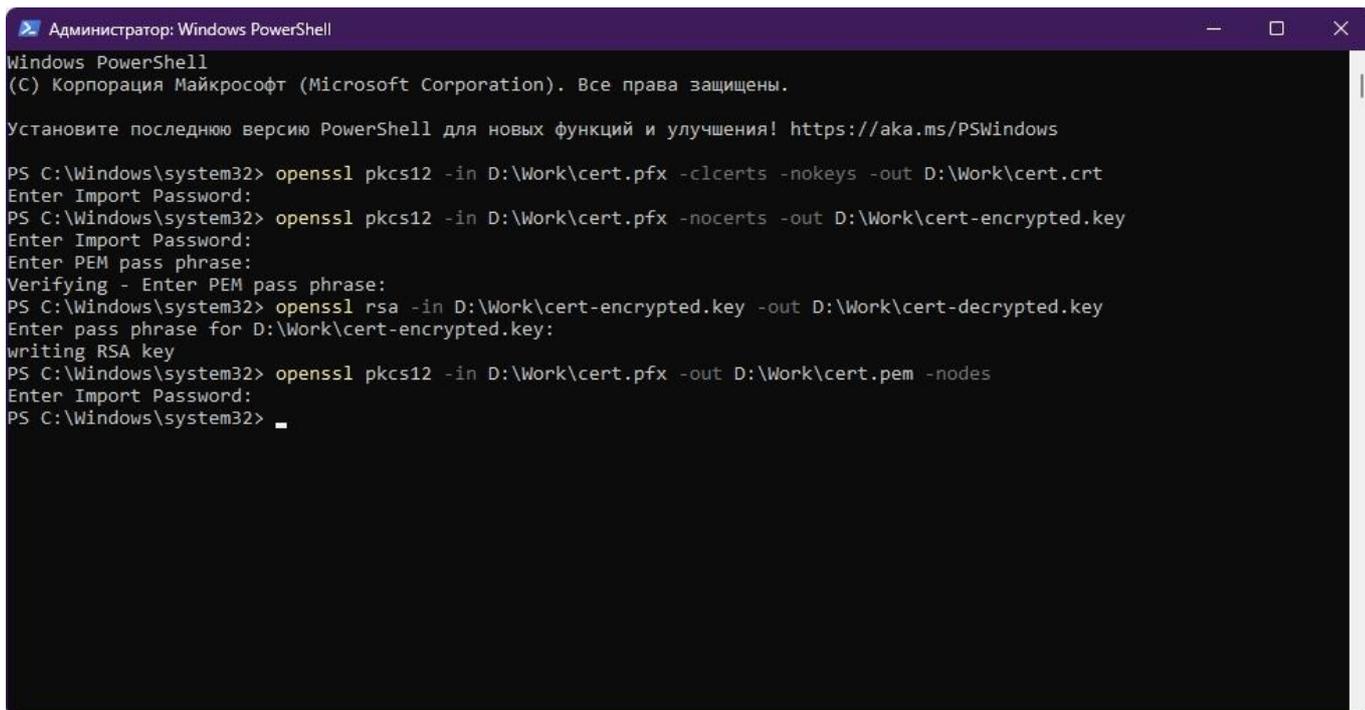
```
choco install openssl
```

2.2.15. Выполните следующие команды для генерации файлов из сертификата (см. Рисунок 8):

```
1. openssl pkcs12 -in Полный путь и имя сертификата.pfx -clcerts -nokeys -out  
Полный путь и имя сертификата.crt
```

```
Например, openssl pkcs12 -in D:\Work\cert.pfx -clcerts -nokeys -out  
D:\Work\cert.crt.
```

2. `openssl pkcs12 -in Полный путь и имя сертификата.pfx -nocerts -out Полный путь и имя сертификата-encrypted.key`
3. `openssl rsa -in Полный путь и имя сертификата-encrypted.key -out Полный путь и имя сертификата-decrypted.key`
4. `openssl pkcs12 -in Полный путь и имя сертификата.pfx -out Полный путь и имя сертификата.pem -nodes`



```
Администратор: Windows PowerShell
Windows PowerShell
(С) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.
Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows
PS C:\Windows\system32> openssl pkcs12 -in D:\Work\cert.pfx -clcerts -nokeys -out D:\Work\cert.crt
Enter Import Password:
PS C:\Windows\system32> openssl pkcs12 -in D:\Work\cert.pfx -nocerts -out D:\Work\cert-encrypted.key
Enter Import Password:
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
PS C:\Windows\system32> openssl rsa -in D:\Work\cert-encrypted.key -out D:\Work\cert-decrypted.key
Enter pass phrase for D:\Work\cert-encrypted.key:
writing RSA key
PS C:\Windows\system32> openssl pkcs12 -in D:\Work\cert.pfx -out D:\Work\cert.pem -nodes
Enter Import Password:
PS C:\Windows\system32>
```

Рисунок 8

2.2.15.1. Когда Windows PowerShell запросит пароль, введите пароль, указанный при выполнении команды, описанной в п. 2.1.3.

2.2.15.2. В результате выполнения команд, описанных в п. 2.2.15, в папке появятся файлы с расширениями «.crt», «decrypted.key» и «.pem» (см. Рисунок 9).

cert.pem	17.05.2024 12:18	Файл "PEM"	4 КБ
cert-decrypted.key	17.05.2024 12:17	Файл "KEY"	2 КБ
cert-encrypted.key	17.05.2024 12:15	Файл "KEY"	3 КБ
cert.crt	17.05.2024 12:13	Сертификат безо...	2 КБ

Рисунок 9

2.3. Настройка модуля для запроса событий в VideoNet

2.3.1. Параметры модуля настраиваются в VideoNet на странице «Интеграция».

Страница «Интеграция» входит в группу страниц «Сеть VideoNet» среды «Конфигурирование».

2.3.1.1. Подробная информация о настройке модуля и работе со страницей «Интеграция» указана в Руководстве пользователя VideoNet.

2.3.2. Чтобы добавить модуль и обеспечить возможность его работы необходимо:

1) Выбрать в дереве элементов компьютер, от которого планируется получать сообщения;

2) Установить флажок «Принимать запросы по REST API» в области свойств выбранного компьютера (см. Рисунок 10);

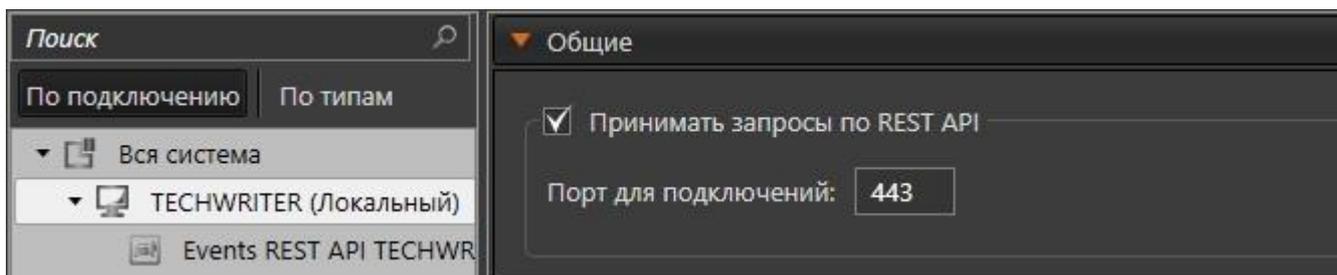


Рисунок 10

3) Установить необходимое значение параметра «Порт для подключений» в области свойств выбранного компьютера (см. Рисунок 10);

4) Нажать кнопку «Добавить» на панели инструментов страницы «Интеграция»;

5) Выбрать в выпадающем списке элемент «Events REST API».

2.3.2.1. При добавлении модуль отобразится в дереве в виде элемента с определённым пользователем именем (по умолчанию «Events REST API <Имя компьютера>»).

2.3.3. При выделении модуля в дереве элементов, в области свойств элементов (в правой части экрана) отобразятся его свойства.

2.3.3.1. Скриншот области свойств модуля представлен на Рисунок 11.



Рисунок 11

2.3.4. Чтобы заблокировать модуль и временно прекратить приём запросов необходимо выделить модуль в дереве элементов и нажать кнопку «Блокировать элемент» на панели инструментов.

3. МЕТОДЫ ЗАПРОСОВ СОБЫТИЙ

3.1. Общие сведения

3.1.1. При необходимости Внешней системы получить перечень событий, произошедших за определённый период на определённом компьютере, Внешней системой должен быть выполнен запрос методом GET.

3.1.2. В заголовке Authorization должны быть переданы параметры username и password (Basic авторизация).

3.1.2.1. Требуемые для авторизации параметры username и password передаются пользователям Внешней системы в административном порядке.

3.1.2.2. Пользователь VideoNet, username и password которого будут переданы для авторизации при запросе данных, должен иметь права на доступ к журналам, к которым планируется отправлять запросы.

3.1.2.3. В случае если username и password не будут соответствовать друг другу или будут неизвестны, Внешней системе будет передана ошибка с кодом 401.

3.1.2.4. В случае если пользователь, чей username и password использованы в запросе, не имеет прав для работы с данными журнала, к которому передаётся запрос, Внешней системе будет передана ошибка с кодом 403.

3.1.2.5. В случае если пользователь, чей username и password использованы в запросе, не имеет прав для работы с определёнными устройствами, в ответе на запрос будут отсутствовать данные (события) от этих устройств.

3.1.3. Ответ на запрос может содержать более одной страницы.

3.1.3.1. Полное количество страниц ответов указывается в параметре ответа TotalPages.

3.1.3.2. Для обращения к необходимой странице ответа необходимо повторно обратиться с запросом, указав в значении параметра Page требуемый номер. Нумерация страниц начинается с 0.

Пример: TotalPages = 5. Для получения всех страниц ответа необходимо последовательно отправить запросы с Page = 0,1,2,3,4.

3.2. Метод запроса событий журнала «Журнал. СКУД»

3.2.1. Для получения событий журнала «Журнал. СКУД» необходимо отправить запрос методом:

GET `https://{IP:port}/log/acs`

где {IP:port} – IP-адрес и порт компьютера VideoNet, события которого необходимо получить.

3.2.2. Определенные параметры строки запроса должны быть обязательно. Описание параметров строки запроса представлено в Таблица 1.

Таблица 1

№	Наименование атрибута	Описание	Тип данных	Обяз.	Примечание
1.	Page	Страница данных, которую необходимо вернуть.	int	-	От 0 и далее При отсутствии параметра будет возвращена страница 0 (первая страница)
2.	DateTimeFrom	Время, с которого надо запросить события	datetime	+	время в формате RFC3339 в UTC
3.	DateTimeTo	Время, по которое надо получить события	datetime	+	время в формате RFC3339 в UTC
4.	Type	Типы событий, которые необходимо получить	Type	-	Может принимать значение: – 3 (Предупреждение); – 5 (Информация). Если параметр отсутствует, то поиск будет осуществляться без учёта типов событий
5.	DeviceType	Тип устройств, по которым надо получить события	DeviceType	-	Возможные значения см. в п. 3.2.3. Если параметр отсутствует, то поиск будет осуществляться без учёта конкретного типа

№	Наименование атрибута	Описание	Тип данных	Обяз.	Примечание
					устройства.
6.	EventName	Наименование сообщений, которые необходимо получить	EventName	-	Возможные значения см. в п. 3.2.4. Если параметр отсутствует, то поиск будет осуществляться без учёта конкретного события
7.	DeviceId	Уникальный идентификатор устройств, по которым надо получить события	GUID	-	Если параметр отсутствует, то поиск будет осуществляться без учёта конкретного устройства
8.	PersonId	Уникальный идентификатор человека, события по которому необходимо получить	GUID	-	Если параметр отсутствует, то поиск будет осуществляться без учёта конкретного человека
9.	CarId	Уникальный идентификатор автомобиля, по которым надо получить события	GUID	-	Если параметр отсутствует, то поиск будет осуществляться без учёта конкретного автомобиля
10.	Full	Признак необходимости полного описания объектов допуска в ответ	boolean	-	Если «true», то в событиях будут выводиться полные описания объектов доступа, при наличии данных в БД СКУД. Иначе только GUID.

3.2.3. Перечень возможных значений типов устройств (DeviceType) представлен в Таблица 2.

Таблица 2

№	Значение	Наименование типа устройства
1.	8	Точка доступа
2.	9	Точка проезда
3.	10	Точка прохода

3.2.4. Перечень возможных значений параметра EventName (тип int) представлен в Таблица 3.

Таблица 3

№ п/п	EventName	Имя события
1.	1	Вход
2.	2	Выход
3.	3	Въезд
4.	4	Выезд
5.	5	Проход
6.	6	Въезд по аварийному открытию
7.	7	Выезд по аварийному открытию
8.	8	Проход по аварийному открытию
9.	9	Проезд по аварийному открытию
10.	10	Доступ разрешён
11.	11	Доступ запрещён
12.	12	Неизвестный ключ
13.	13	Неизвестный номер
14.	14	Неизвестное лицо
15.	15	Возможно неправомерное использование ключа
16.	16	Ошибка идентификации
17.	17	Проход не совершен по таймауту
18.	18	Успешное подтверждение доступа
19.	19	Проезд
20.	20	Неизвестный код
21.	21	Открытие
22.	22	Закрытие

3.2.5. Пример URL с заполненными параметрами для выполнения метода:

<https://192.168.0.1:443/log/acs?Page=0&DateTimeFrom=2023-06-01T19:00:00.000+03:00&DateTimeTo=2023-06-01T21:00:00.000+03:00>

3.2.6. Описание параметров тела ответа представлено в Таблица 4.

Таблица 4

№	Наименование атрибута	Описание	Тип данных	Обяз.	Примечание
1.	TotalNumber	Полное количество событий по запросу (в ответе)	int	+	
2.	TotalNumberOnPage	Количество событий по запросу на текущей странице	int	+	
3.	TotalPages	Всего страниц с ответами по запросу	int	+	
4.	Page	Возвращенная страница данных	int	+	Соответствует номеру запрошенной страницы
5.	DateTimeFrom	Время из запроса	datetime	+	время в формате RFC3339 в UTC
6.	DateTimeTo	Время из запроса	datetime	+	время в формате RFC3339 в UTC
7.	Type	Типы событий из запроса	Type	-	
8.	DeviceType	Тип устройства из запроса	DeviceType	-	
9.	EventName	Наименование сообщения из запроса	EventName	-	
10.	DeviceId	Уникальный идентификатор устройства из запроса	GUID	-	

№	Наименование атрибута	Описание	Тип данных	Обяз.	Примечание
11.	PersonId	Уникальный идентификатор человека из запроса	GUID	-	
12.	CarId	Уникальный идентификатор автомобиля из запроса	GUID	-	
13.	Full	Признак полного описания объектов допуска	boolean	-	
14.	EventList	Массив с описанием событий	Event	-	Множество записей о событиях, которое равно TotalNumberOnPage.

3.2.7. Описание типа данных Event представлено в Таблица 5.

Таблица 5

№	Наименование атрибута	Наименование поля	Тип	Обяз.	Примечание
1.	EventId	Уникальный идентификатор события в VideoNet	GUID	+	
2.	DateTime	Время события	datetime	+	Время в формате RFC3339 в UTC
3.	Type	Тип события	int	+	Может принимать значение: – 3 (Предупреждение); – 5 (Информация).
4.	EventName	Наименование сообщения	int	+	Возможные значения см. в п. 3.2.4
5.	KeyData	Код/номер ключа	string	-	Протоколируется, если в событии в качестве

№	Наименование атрибута	Наименование поля	Тип	Обяз.	Примечание
					идентификатора указан ключ
6.	CarNumber	Номер автомобиля	string	-	Протоколируется, если в событии в качестве идентификатора указан номер автомобиля
7.	DeviceType	Тип устройства, по которому сформировано сообщения	int	+	Возможные значения см. в п. 3.2.3.
8.	DeviceId	Уникальный идентификатор устройства	GUID	+	
9.	DeviceName	Имя устройства	string	+	
10.	OriginalEventText	Полный текст сообщения	string	+	
11.	HostId	Уникальный идентификатор компьютера	GUID	+	
12.	HostName	Имя компьютера, на котором произошло событие	string	+	
13.	PersonDescription	Параметры человека, к которому относится событие доступа	PersonResult	-	Описание структуры PersonResult см. в п. 3.2.8. Данные передаются, если событие доступа относится к человеку, зарегистрированному в БД СКУД, или к

№	Наименование атрибута	Наименование поля	Тип	Обяз.	Примечание
					связанному с этим человеком автомобилю.
14.	CarDescription	Параметры автомобиля, к которому относится событие доступа	CarResult	-	Описание структуры CarResult см. в п. 3.2.9. Данные передаются, если событие доступа относится к автомобилю, зарегистрированному в БД СКУД
15.	ACSDBInfoDescriptionResult	Параметры БД СКУД, с которой проходило сравнение при возникновении события	ACSDBResult	+	Описание структуры ACSDBResult см. в п. 3.2.11

3.2.8. Описание структуры «PersonResult», передающей в теле запроса параметры человека, представлено в Таблица 6.

Таблица 6

№	Имя атрибута	Описание поля	Тип	Обяз.	Примечание
1.	PersonId	Уникальный идентификатор человека	GUID	+	
2.	Name	Имя человека	string	-	
3.	Patronymic	Отчество человека	string	-	
4.	Surname	Фамилия человека	string	-	
5.	Number	Табельный номер	string	-	

№	Имя атрибута	Описание поля	Тип	Обяз.	Примечание
6.	PositionId	Уникальный идентификатор должности	GUID	-	
7.	PositionName	Название должности	string	-	
8.	GroupId	Уникальный идентификатор группы (подразделения), в которую определён человек	GUID	-	
9.	GroupName	Название группы (подразделения), в которую определён человек	string	-	
10.	Comment	Комментарий к данным человека	string	-	
11.	PersonalPhoneNumber	Номер личного телефона	string	-	
12.	WorkPhoneNumber	Номер рабочего телефона	string	-	
13.	Email	Адрес электронной почты	string	-	Если почтовых адресов несколько, то они будут отделены друг от друга точкой с запятой

3.2.9. Описание структуры «CarResult», передающей в теле запроса параметры автомобиля, представлено в Таблица 7.

Таблица 7

№	Наименование атрибута	Наименование поля	Тип данных	Обяз.	Примечание
1.	CarId	Уникальный идентификатор	GUID	+	

№	Наименование атрибута	Наименование поля	Тип данных	Обяз.	Примечание
		автомобиля			
2.	Number	Номер	string	-	
3.	Color	Цвет	int	-	Возможные значения см. в п. 3.2.10
4.	Model	Модель	string	-	
5.	Comment	Комментарий к данным автомобиля	string	-	

3.2.10. Перечень возможных значений параметра Color (тип int) представлен в Таблица 8.

Таблица 8

№ п/п	Color	Название цвета
1.	0	Отсутствует значение цвета в параметрах автомобиля
2.	1	Черный
3.	2	Серебристый
4.	3	Белый
5.	4	Серый
6.	5	Синий
7.	6	Красный
8.	7	Зеленый
9.	8	Коричневый
10.	9	Бежевый
11.	10	Голубой
12.	11	Золотой
13.	12	Пурпурный
14.	13	Фиолетовый
15.	14	Желтый
16.	15	Оранжевый
17.	16	Розовый

3.2.11. Описание структуры ASCDBResult, передающей в теле запроса параметры БД СКУД, представлено в Таблица 9.

Таблица 9

№	Наименование атрибута	Наименование поля	Тип данных	Обяз.	Примечание
1.	Id	Уникальный идентификатор БД СКУД	GUID	+	
2.	Name	Имя БД СКУД	string	+	

3.2.12. Пример тела ответа:

```
{
  "TotalNumber":2,
  "TotalNumberOnPage": 2,
  "TotalPages": 1,
  "Page":0,
  "DateTimeFrom": "2023-06-01T19:00:00.000+03:00",
  "DateTimeTo": "2023-06-01T21:00:00.000+03:00",
  "Type":5,
  "DeviceType": 8,
  "EventName": 1,
  "DeviceId": "BF94A6DA-6366-4F9C-958A-E02C00B50C4D",
  "PersonId": "5615E6F9-79C3-4623-9B18-21C194FEC2CC",
  "Full": false,
  "EventList":[
    {"EventId": "DDC46864-C6B8-4514-989E-9BD333B66374",
      "DateTime": "2023-06-01T19:10:00Z",
      "Type": 5,
      "EventName": 1,
      "KeyData": "5/34",
      "DeviceType": 8,
```

```
"DeviceId": "BF94A6DA-6366-4F9C-958A-E02C00B50C4D",
"DeviceName": "Главный вход",
"OriginalEventText": "Вход «Петров С.И.» по ключу 5/34. Точка доступа
«Главный вход»",
"HostId": "804EE028-FF27-47A4-9AA7-7CE5FD3F4ABE",
"HostName": "Server 1",
"PersonDescription": {"PersonId": "5615E6F9-79C3-4623-9B18-
21C194FEC2CC"},
"ACSDBInfoDescriptionResult": {"Id": "2376B4EC-7661-4EEF-89DC-
E2F720425BE8",
                                "Name": "База СКУД KONDRATIEVA"}
},
{"EventId": "D36B28B5-AE1C-4D7A-AD1A-FE51236E2064",
"DateTime": "2023-06-01T20:40:00Z",
"Type": 5,
"EventName": 1,
"KeyData": "5/34",
"DeviceType": 8,
"DeviceId": "BF94A6DA-6366-4F9C-958A-E02C00B50C4D",
"DeviceName": "Главный вход",
"OriginalEventText": "Вход «Петров С.И.» по ключу 5/34. Точка доступа
«Главный вход»",
"HostId": "804EE028-FF27-47A4-9AA7-7CE5FD3F4ABE",
"HostName": "Server 1",
"PersonDescription": {"PersonId": "5615E6F9-79C3-4623-9B18-
21C194FEC2CC"},
"ACSDBInfoDescriptionResult": {"Id": "2376B4EC-7661-4EEF-89DC-
E2F720425BE8",
                                "Name": "База СКУД KONDRATIEVA"}
}
```

}]

ПРИЛОЖЕНИЕ

ТЕСТОВОЕ ПРИЛОЖЕНИЕ

Тестовое приложение выполняет запрос на сервер VideoNet и позволяет оценить корректность выполнения настроек, указанных в разделе 2.

Исходный код на Python

Для корректной отправки запросов на сервер VideoNet на языке Python в запросе необходимо указать файлы «Имя сертификата.crt», «Имя сертификата-decrypted.key» и «Имя сертификата.pem» (см. 2.2.15.2).

Пример скрипта:

```
import requests

def main():
    url = 'https://test.videonet.ru/log/acs?DateTimeTo=2025-12-27T19:00:00.000+03:00&DateTimeFrom=2020-07-10T11:00:00.000+03:00'
    r = requests.get(url,
                     auth=('admin', '111111'),
                     cert=('ssl/cert.crt', 'ssl/cert-decrypted.key'),
                     verify='ssl/cert.pem')

    print(r.status_code)

if __name__ == "__main__":
    main()
```

В результате выполнения скрипта с запросом при правильной настройке, выполненной согласно разделу 2, должен прийти код 200.

Исходный код на C#

Код написан на .NET Core 7.0 и должен размещаться в файле Program.cs.

Пример кода:

```
using System.Runtime.InteropServices;
using System.Security.Authentication;
using System.Security.Cryptography.X509Certificates;
using System.Text;

var username = "admin";
var password = "111111";
```

```
var baseAddress = "https://test.videonet.ru";
var crtFileName = "ssl/cert.crt";
var decryptedKeyFileName = "ssl/cert-decrypted.key";
var getQuery = "https://test.videonet.ru/log/acs?DateTimeTo=2025-12-27T19:00:00.000+03:00&DateTimeFrom=2020-07-10T11:00:00.000+03:00";

X509Certificate2 CreateCertFromPemFile(string certPath, string keyPath)
{
    if (!RuntimeInformation.IsOSPlatform(OSPlatform.Windows))
        return X509Certificate2.CreateFromPemFile(certPath, keyPath);

    using var cert = X509Certificate2.CreateFromPemFile(certPath, keyPath);
    return new X509Certificate2(cert.Export(X509ContentType.Pkcs12));
}

var handler = new HttpClientHandler();
handler.ClientCertificateOptions = ClientCertificateOption.Manual;
handler.SslProtocols = SslProtocols.Tls12;
handler.ClientCertificates.Add(CreateCertFromPemFile(crtFileName, decryptedKeyFileName));

var client = new HttpClient(handler)
{
    BaseAddress = new Uri(baseAddress),
};

var encoded = Convert.ToBase64String(Encoding.GetEncoding("ISO-8859-1").GetBytes(username + ":" + password));

client.DefaultRequestHeaders.Add("Authorization", $"Basic {encoded}");

var result = await client.GetAsync(getQuery);

Console.WriteLine($"Result code: {(int)result.StatusCode} - {result.StatusCode}");
Console.WriteLine($"Result body: {result.Content.ReadAsStringAsync().Result}");

Console.WriteLine($"Press Enter for exit..");

Console.ReadLine();
```

При запуске скомпилированного тестового примера при правильной настройке, выполненной согласно разделу 2, должен прийти ответ на запрос с кодом 200.