Система лицензирования и защиты конфигураций платформы 1С:Предприятие 8, версия 3.0

Руководство администратора

Поддерживаемые операционные системы

СЛК поддерживает следующие, 32-разрядные и 64-разрядные, операционные системы:

- ОС Microsoft Windows XP и выше: XP, Server 2003, Vista, 7, Server 2008 (R2), 8, 8.1, Server 2012 (R2), 10, Server 2016
- ОС Linux на базе Debian: Debian 7.5 и выше, Ubuntu 12.04 и выше
- ОС Linux на базе RedHat: CentOS 6.7 и выше, Fedora 22 и выше

Состав системы

СЛК состоит из двух основных частей – сервера и клиента.

Сервер СЛК обслуживает ключи защиты (программные и аппаратные), раздает их лицензии и обрабатывает запросы клиентов.

Клиент, выполненный в виде внешней компоненты 1С:Предприятие, или Компонента СЛК, встраивается в конфигурацию ее разработчиком и используется для создания защищенных объектов, получения лицензий и данных с сервера СЛК.

> *Версии сервера и компоненты должны соответствовать друг другу. Компонента от предыдущих версий СЛК 2.* не сможет работать с сервером СЛК 3.0 и наоборот, компонента от СЛК 3.0 не сможет работать с сервером СЛК 2.*.*

Сервер СЛК

Сервер является основным элементом системы и предназначен для контроля количества лицензий и доступа к ключам защиты, защищенным объектам и лицензионным параметрам.

Сервер СЛК поддерживает одновременную работу с ключами от нескольких продуктов, поэтому рекомендуется

Установочные пакеты

Windows	32 / 64 бит	licenceserver-{version}.win.exe upkey-{version}.exe
Debian / Ubuntu	32 бит	licenceserver-{version}.i386.deb
	64 бит	licenceserver-{version}.amd64.deb
CentOS / Fedora	32 бит	licenceserver-{version}.i386.rpm
	64 бит	licenceserver-{version}.x86_64.rpm

Установка в ОС Windows

При запуске установочного пакета выполняется установка в интерактивном режиме. Также возможен **запуск установочного пакета с параметрами командной строки**.

Если в папке установочного пакета сервера присутствует установочный пакет драйвера ключа защиты upkey-{version}.exe, то пользователю предлагается установить драйвер.

> *При установке / обновлении драйвера рекомендуется отключить подключенные к USB портам ключи защиты.*

Установка в ОС Linux

Установка выполняется при помощи стандартного менеджера пакетов ОС. Например, для установки в 64-разрядных Debian / Ubuntu необходимо в терминале с правами суперпользователя выполнить следующую команду:

```
dpkg –i licenceserver-{version}.amd64.deb
```

Аналогично, для установки в CentOS / Fedora:

```
yum localinstall licenceserver-{version}.x86_64.rpm
```

При завершении установки пакета выполняется настройка и запуск системной службы (демона) сервера. Проверить, что сервер успешно установлен и запущен можно при помощи команды service:

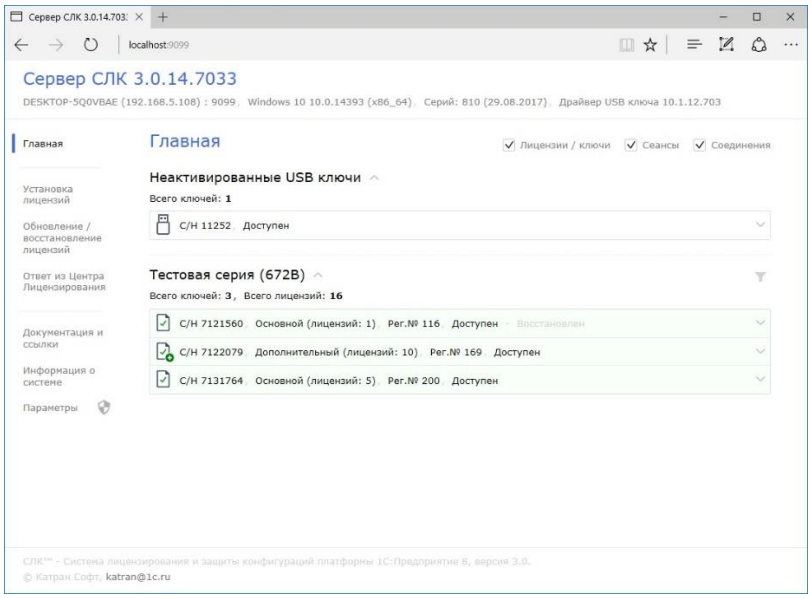
```
service licenceserver status
```

В CentOS / Fedora при обновлении установленной версии сервера ниже 3.0.11 в возможно удаление службы сервера. В этом случае необходимо принудительно установить службу, выполнив с правами суперпользователя скрипт **installdaemon.sh**:

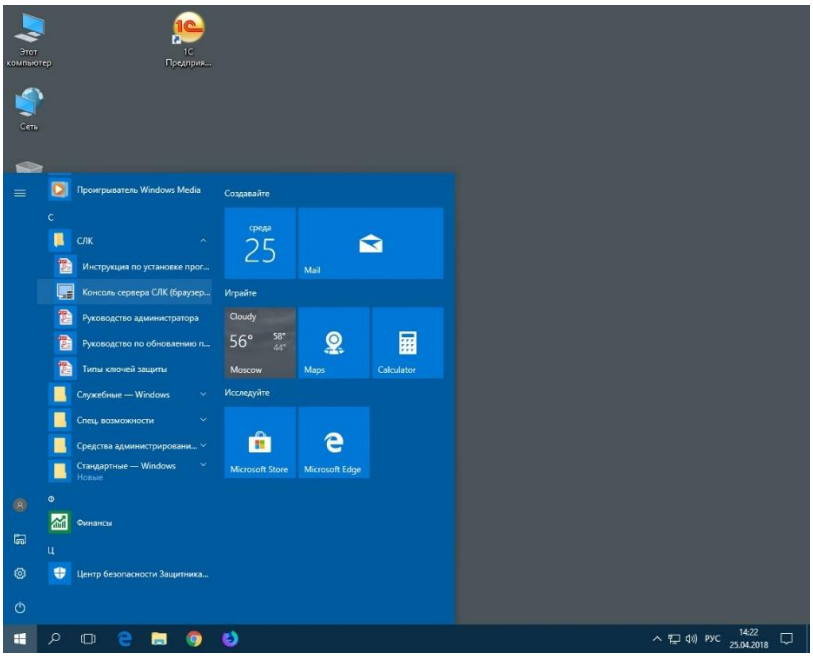
```
cd /opt/1C/licence/3.0/
./installdaemon.sh
```

Консоль сервера

Администрирование сервера, установка лицензий и настройка параметров выполняется при помощи консоли, которая выполнена в виде веб-приложения и доступна при помощи веб браузера по адресу и порту сервера СЛК. Например, на локальном компьютере при использовании стандартного порта адрес консоли будет **http://localhost:9099**

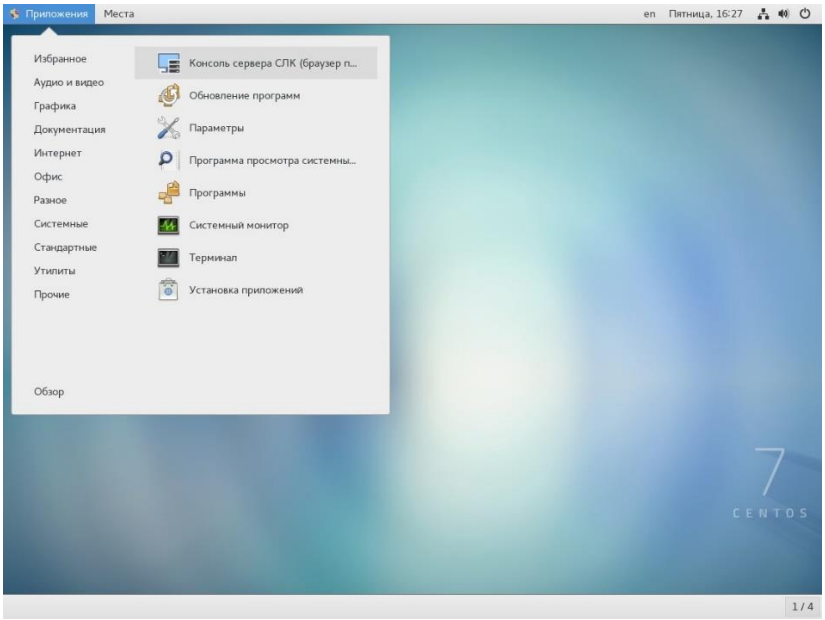


> *Для ОС Windows ссылка на консоль добавляется в меню «Пуск» при установке сервера СЛК:*
>
> ***Пуск – Программы – СЛК – 3.0 – Консоль сервера (браузер по умолчанию)***



Windows Server 2016

> *Для ОС Linux с графическим окружением ссылка на консоль добавляется в меню приложений при установке сервера СЛК:*
>
> ***Приложения – Системные / Офис – Консоль сервера СЛК (браузер по умолчанию)***



CentOS 7

Ubuntu 17.10

Пароль по умолчанию

> *По соображениям безопасности некоторые операции (изменение настроек, получение резервных кодов) по умолчанию требует авторизации, поэтому при обращении к ним будут запрошены имя пользователя и пароль.*
>
> *Рекомендуется не отключать авторизацию и изменить имя пользователя и пароль по умолчанию.*

По умолчанию используются имя пользователя **admin** и пароль **admin**.

См. также **Установка программного ключа**.

Конфигурационный файл

В общем случае необходимости вручную редактировать конфигурационный файл сервера нет – основные параметры настраиваются в консоли сервера в соответствующем разделе.

Имя файла

licenceserver.conf

Расположение в ОС Windows

%ProgramData%\1C\licence\3.0\licenceserver.conf

Где %ProgramData% - системная папка общих настроек. Например, для ОС Windows Vista и выше это может быть:

C:\Program Data\

Расположение в ОС Linux

/var/1C/licence/3.0/licenceserver.conf

В установочные пакеты сервера включен пример конфигурационного файла с комментариями (на английском).

Сброс пароля на доступ к параметрам и служебным операциям

При необходимости логин и пароль можно сбросить в конфигурационном файле в разделе *Console*:

```
[Console]
; Режим авторизации консоли (используется базовая HTTP авторизация)
; 0 – только параметры и служебные операции (по-умолчанию)
; 1 – всегда
; 2 – отключена (не рекомендуется)
Authorization=0

; Параметры авторизации
UserName=admin
Password=admin
```

Внешнее API

Для автоматизации администрирования и управления сервером СЛК реализовано внешнее REST API, доступное по адресу и порту сервера:

http://localhost:9099/api

Доступность API и параметры аутентификации настраивается в консоли сервера в разделе Параметры. По умолчанию используются имя пользователя **admin** и пароль **admin**.

Подробное описание доступных команд см. в **приложении**.

Ключи защиты

Ключи защиты – компонент системы, предназначенный для доступа к защищенным объектам и контроля количества лицензий на основе указанного в ключе значения.

Для **каждой защищаемой конфигурации** создается своя, **уникальная, серия ключей**, что делает ключи от одной конфигурации несовместимыми с ключами от других конфигураций.

В рамках серии ключи могут быть **основными** (обычный вариант использования) и **дополнительными** (для расширения функционала и количества лицензий) и **демонстрационными** (временными, для демонстрации возможностей продукта и организации промо-акций).

> ***Дополнительные*** *ключи не работают без основного ключа и должны быть установлены на том же компьютере, что и основной ключ.*

> ***Демонстрационные*** *ключи не могут быть использованы для увеличения числа лицензий и не могут быть установлены вместе с другими ключами такой же серии.*

Каждый ключ содержит количество лицензий, которое определяет максимально возможное число одновременно занимаемых лицензий. Кроме этого, каждый ключ имеет уникальный серийный номер, присваиваемый при создании ключа.

По виду исполнения ключи могут быть **аппаратными** (USB) или **программными**, привязанными к конфигурации компьютера или **неактивированному** аппаратному ключу.

Для работы **аппаратных** ключей в ОС Windows необходима установка специального драйвера, поддерживающего все версии и типы (32-разрядные и 64-разрядые) ОС Windows, начиная с Windows XP.

Для работы аппаратных ключей в ОС Linux необходимо наличие стандартной библиотеки LibUSB версии 1.0 или выше.

Аппаратные ключи поставляются **активированными** (уже подготовленными для работы конкретной конфигурации и содержащими лицензии определенного типа) и **неактивированными** (также называемыми **аппаратными носителями лицензий** или **болванками**), которые необходимо активировать при помощи кода активации программного ключа.

Подробнее см. **Ключи защиты**, **Установка программного ключа**.

Внешняя компонента

Внешняя компонента СЛК – это клиентская часть системы, при помощи которой конфигурация обращается к серверу СЛК.

Компонента СЛК должна размещаться разработчиком в общем макете конфигурации, поэтому отдельной установки компоненты в общем случае не требуется.

Отдельная установка компоненты позволяет обновить компоненту без внесения изменений в конфигурацию / конфигурации используется в случаях:

- Когда необходимо обеспечить одновременную работу нескольких конфигураций, содержащих различные версии компонент СЛК. В этой ситуации отдельная установка выполняет унификацию всех версий компоненты до единой версии – той, которая устанавливается.
- Когда необходимо выполнить общее обновление СЛК, но изменить макет компоненты в самой конфигурации невозможно.

Установка компоненты должна выполнятся на тех компьютерах, где выполняется рабочий процесс системы 1С:Предприятие:

- На компьютере сервера приложений для клиент-серверных баз.
- На компьютерах, где запускается клиентское приложение (тонкий / толстый клиент).
- На компьютере, где работает веб-сервер, при публикации на нем файловых баз

См. также примеры установки компоненты в **инструкции по обновлению предыдущих версий**.

Установочные пакеты

Windows	32 бит / 64 бит	licenceaddin-{version}.win.exe
Debian / Ubuntu	32 бит	licenceaddin-{version}.i386.deb
	64 бит	licenceaddin-{version}.amd64.deb
CentOS / Fedora	32 бит	licenceaddin-{version}.i386.rpm
	64 бит	licenceaddin-{version}.x86_64.rpm

Установка в ОС Windows

При запуске установочного пакета выполняется установка в интерактивном режиме. Также возможен **запуск установочного пакета с параметрами командной строки**.

> *При обнаружении работающих процессов 1С:Предприятия (сервер приложений или тонкий/толстый клиенты) программа установки автоматически выполняет их завершение или перезапуск.*

Установка в ОС Linux

Установка выполняется при помощи стандартного менеджера пакетов ОС. Например, для установки в 64-разрядных Debian / Ubuntu необходимо в терминале с правами суперпользователя выполнить следующую команду:

```
dpkg –i licenceaddin-{version}.amd64.deb
```

Аналогично, для установки в CentOS / Fedora:

```
yum localinstall licenceaddin-{version}.x86_64.rpm
```

> *Перед установкой пакета рекомендуется остановить рабочие процессы 1С:Предприятия сервера приложений и тонкие/толстые клиенты).*

Конфигурационный файл

В общем случае параметры связи компоненты с сервером СЛК должны указываться разработчиком в коде конфигурации и настраиваться средствами 1С. Однако, в некоторых случаях (например, при большом количестве баз и / или необходимости запретить изменение настроек в конфигурации) возможно указание параметров в конфигурационном файле.

> *Для клиентских приложений 1С (тонкий / толстый клиенты) сначала проверяется наличие конфигурационного файла в папке текущего пользователя ОС. Если для пользователя ОС файл отсутствует, проверяется наличие файла в общей папке. Для сервера приложений или веб сервера проверяется наличие файла в общей папке.*

Имя файла

licenceaddin.conf

Расположение в ОС Windows

%ProgramData%\1C\licence\3.0\licenceaddin.conf

Где %ProgramData% - системная папка общих настроек. Например, для ОС Windows Vista и выше это может быть:

C:\Program Data\

Расположение в ОС Windows для текущего пользователя ОС

%LocalAppData%\1C\licence\3.0\licenceaddin.conf

Где %LocalAppData% - папка настроек пользователя, от имени которого запущен процесс 1С, например:

C:\Users\%User%\AppData\Local\1C\Licence\3.0\

Расположение в ОС Linux

/var/1C/licence/3.0/licenceaddin.conf

Расположение в ОС Linux для текущего пользователя ОС

/home/{user}/.1C/licence/3.0/licenceaddin.conf

Где {user} - пользователь, от имени которого запущен процесс 1С.

Пример

Конфигурационный файл представляет себя обычный текстовый ini-файл к кодировке UTF-8:

```
[Common]
; Установить в 1, чтобы игнорировать настройки, сделанные в конфигурации
ForceUseConfig=0
; IP адрес или имя компьютера, где установлен сервер СЛК
ServerAddr=localhost
; Порт, на котором работает сервер СЛК
ServerPort=9099

; Список резервных серверов СЛК. Если указаны, компонента будет
; последовательно перебирать эти адреса до успешного соединения.
; При использовании стандартного порта 9099, его можно не указывать.
; Addr1[:Port1],Addr2[:Port2],...AddrN[:PortN]
ReservedServers=

; Для указания отдельных параметров для конкретной серии ключей необходимо
; для этой серии создать отдельную секцию.
; Например, для серии 672B:

[672B]
; Установить в 1, чтобы игнорировать настройки, сделанные в конфигурации
ForceUseConfig=1
; IP адрес или имя компьютера, где установлен сервер СЛК
ServerAddr=SERVER2
; Порт, на котором работает сервер СЛК
ServerPort=9099
; Список резервных серверов СЛК
ReservedServers=
```

Принцип учета лицензий

Система реализует контроль лицензий на стороне сервера, особенностью подхода является то, что при работе в клиент-серверном режиме (сервер приложений 1С, веб сервер) каждый сеанс информационной базы будет занимать отдельную лицензию. При

работе в файловом режиме одну лицензию будут занимать все подключения с одного компьютера (при работе в терминальном режиме одну лицензию будут занимать все подключения с одного терминального клиента).

Обновление предыдущих версий

Для обновления предыдущих версий необходимо выполнить обновление сервера и компоненты при помощи соответствующих установочных пакетов.

При этом установка компоненты должна выполнятся на тех компьютерах, где выполняется рабочий процесс системы 1С:Предприятие:

- На компьютере сервера приложений для клиент-серверных баз.
- На компьютерах, где запускается клиентское приложение (тонкий / толстый клиент) для файловых.

Например, для ОС Windows необходимо:

1. Обновить сервер при помощи пакета licenceserver-{version}.win.exe
2. Обновить компоненту при помощи пакета licenceaddin-{version}.win.exe

> *Обновление сервера СЛК возможно без завершения работы пользователей 1С*

Подробнее см. **обновление**, а также **сервер СЛК** и **внешняя компонента**.

Приложение: Перенос лицензии с компьютера на аппаратный носитель

В случае необходимости переноса уже установленного на компьютере программного ключа на аппаратный носитель (неактивированный аппаратный ключ) необходимо:

1. Получить в консоли сервера СЛК резервный код для программного ключа, который нужно перенести на аппаратный носитель
2. Подключить аппаратный носитель к компьютеру, где работает сервер СЛК
3. Выполнить установку полученного резервного кода на подключенном носителе

Подробнее см. **установка программного ключа**, разделы установка лицензий и получение резервных кодов.

Приложение: Запуск установочного пакета с параметрами командной строки (Windows)

Установочные пакеты для ОС Windows поддерживают запуск со следующими параметрами:

/Silent

Выполнить установку без отображения окон программы установки, но с отображением прогресса операции

/VerySilent

Выполнить установку полностью без отображения окон программы установки

/SuppressMsgBoxes

Не показывать сообщений об ошибках

/CloseRunning

Завершить / перезапустить процессы и службы, мешающие выполнению операции (установочный пакет компоненты)

Коды возврата

0 - установка выполнена успешно

100 - установка прервана, т.к. такой же или более новый пакет уже установлен

1 - ошибка инициализации программы установки

2 - пользователь отменил операцию

3-6 - внутренние ошибки программы установки

7 - обнаружены процессы, мешающие установке (1с-клиенты, сервер 1с)

8 - необходимо выполнить перезагрузку системы для продолжения установки

Пример

licenceserver-{version}.win.exe /VERYSILENT /SUPPRESSMSGBOXES

Эта команда выполнит установку сервера СЛК в «бесшумном» режиме – без отображения окон установщика и сообщений об ошибках.

Приложение: Расположение конфигурационных файлов

Для ОС Windows

%ProgramData%\1C\licence\3.0\

Где %ProgramData% - системная папка общих настроек. Например, для ОС Windows Vista и выше это может быть:

C:\Program Data\

Для ОС Linux

/var/1C/licence/3.0/

Конфигурационные файлы представляет себя обычные текстовые ini-файлы в кодировке UTF-8. Примеры конфигурационных файлов с комментариями (на английском) включены в установочные пакеты как сервера, так и компоненты.

Приложение: Расположение системных логов

Для сервера СЛК

Имя файла

licenceserver.{%timestamp%}.{%pid%}.log

Где:

{%timestamp%} – время создания файла в виде YYYYMMDD-HHNNSS

{%pid%} – идентификатор процесса, создавшего файл

Расположение в ОС Windows

%ProgramData%\1C\licence\3.0\logs\licenceserver*.log

Где %ProgramData% - системная папка общих настроек. Например, для ОС Windows Vista и выше это может быть:

C:\ProgramData\

Расположение в ОС Linux

/var/1C/licence/3.0/logs/licenceserver*.log

Для компоненты

Имя файла

licenceaddin.{%timestamp%}.{%pid%}.{%modulename%}.log

Где:

{%timestamp%} – время создания файла в виде YYYYMMDD-HHNNSS

{%pid%} – идентификатор процесса, создавшего файл

{%modulename%} – имя файла компоненты без расширения

Расположение в ОС Windows

%LocalAppData%\1C\licence\3.0\logs\licenceaddin*.log

Где %LocalAppData% - папка настроек пользователя, от имени которого запущен процесс 1С. Например, для сервера приложений это может быть:

C:\Users\USR1CV8\AppData\Local\1C\Licence\3.0\logs\licenceaddin*.log

Логи компоненты, подключаемой веб-сервером (IIS, Apache) при публикации файловых баз, сохраняются в общей папке, аналогично логам сервера СЛК:

%ProgramData%\1C\licence\3.0\logs\licenceaddin*.log

Расположение в ОС Linux

/home/{user}/.1C/licence/3.0/logs/licenceaddin*.log

Где {user} - пользователь, от имени которого запущен процесс 1С. Например, для сервера приложений это может быть:
/home/usr1cv8/.1C/licence/3.0/logs/licenceaddin*.log

Режим сохранения / перезаписи

По умолчанию логи сохраняются только для текущего сеанса работы и при следующем запуске приложения (сервера СЛК или защищенной конфигурации) эти файлы удаляются / перезаписываются. Однако в целях отладки логи предыдущих сеансов работы можно хранить, для этого в конфигурационных файлах необходимо установить флаг сохранения KeepPreviousLogs:

```
[Common]

; Хранить логи предыдущих сеансов работы
KeepPreviousLogs=1
```

Приложение: Внешнее API сервера СЛК

Доступ и авторизация

API доступно по адресу и порту сервера:

http://localhost:9099/api

Используется базовая HTTP аутентификация, по умолчанию имя пользователя **admin** и пароль **admin**.

Выполнение команд

Для выполнения команд используются стандартные HTTP-методы:

GET – получение данных
POST – выполнение команд / изменение параметров
PUT – выполнение команд / изменение параметров

Параметры запросов (в случае их необходимости) передаются в теле HTTP запроса в виде JSON строки в кодировке UTF-8 (строка должна соответствовать спецификации JSON: **http://www.json.org/json-ru.html**).

Аналогично, в теле HTTP ответа возвращаются данные ответа и расшифровка ошибок.

> *Если при описании запросов указаны значения входных параметров по умолчанию, то эти параметры могут отсутствовать в запросе.*

Дополнительные параметры фильтрации/форматирования/представления данных могут быть переданы внутри обращения к базовым адресам, например:

```
Базовый адрес[?Параметр=Значение[&Параметр=Значение]]
```

> *Значения даты / времени содержат универсальное время (UTC) в формате ISO8601*

Коды ответа

Используются следующие HTTP коды ответа:

200 OK
201 Created (Запись создана)
202 Updated (Запись изменена)

400 Bad Request (Некорректный запрос)
401 Unauthorized (Неавторизованный доступ)
403 Forbidden (Доступ запрещен)
404 Not Found (Данные не найдены)
500 Internal server error (Внутренняя ошибка сервера)
501 Not Implemented (Не реализовано)
503 Service Unavailable (Доступ к API отключен в настройках)

В случае ошибки в теле HTTP ответа содержится JSON строка с расшифровкой, например:

```
{
	"Code": 404,
	"Text": "Not found",
	"Message": "(EHTTPNotFound) The requested URL \"/api/someurl\" was not found.",
	"Exception": "EHTTPNotFound"
}
```

Где:

Code	HTTP код возврата
Text	HTTP текст
Message	Сообщение об ошибке
Exception	Класс исключения

Идентификация запроса и цифровая подпись

Ответы на запросы внешнего API подписаны цифровой подписью (RSA, 2048, SHA256), которая передается в виде Base64 в поле Licence-Signature HTTP-заголовка ответа.

Если при отправке запроса в HTTP-заголовке установлено поле Licence-QueryID, то это значение также возвращается в HTTP-заголовке ответа и используется при расчете цифровой подписи:

> *Licence-Signature = Цифровая подпись RSA (Licence-QueryID + Тело ответа)*

В поле Licence-QueryID рекомендуется передавать уникальный идентификатор запроса, формируемый, например, на основе текущего времени.

> *Проверка цифровой подписи с использованием идентификатора запроса позволит исключить / затруднить злоумышленнику подмену / модификацию ответов от сервера СЛК.*

Для проверки электронной подписи необходимо использовать следующий открытый ключ:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAt1STAPAz99fXcOhzGfuf
q9Ctj5o472wlC0R7iboPh8cFyiUS+6DoOTU8HFkpWpAChPy6SncxaFeCypad+1vq
eLk2v/jhE1M8WdotwK61MVQZ4DzxMyZ7JwNUz4ZbObOUCJaD3FwU/nca/p9rCeYj
NER1rYujEgdJ9G7trvdu8r2OzsANsUBmAA65vNLKkONDSMtTr+XaN9L4tp6AfZFu
mRvVPJMEV2rJ2dExs8yiLq6a08Vx8HvlOBmQYRtBBjxycgbFHB74JVZNjqonKSSE
7Zzniil1eETsgtEMCagpC5/JQIcl2lh+NJUB31FCwsATVc+srIuD0kEawpcPvLQ8
OQIDAQAB
-----END PUBLIC KEY-----
```

Пример HTTP-заголовка ответа:

Status Code	400 Bad Request
Content-Length	146
Content-Type	application/json;charset=utf-8
Licence-ServerVersion	3.0.20.8508,Win32,i386
Licence-QueryID	20200127230600
Licence-QuerySignature	D/0ukaFAohMWhZRt3zQUtdqBnP9LDEcksWet1S9mc2qj3l9Y4xYsFV7m3udp65CXaqH8Ck1j/Rdt3Y2z3RCLyaYuopYpKyfxZv492R4lOrt6NdRs6RdCUVES3fX166SdWHp+HnudpcPiwSc+hTNtfOgB4jlBQjSAaYWarT9KomYXTO/musMbQwMUBJyA2pE8u8Wc1VWkk+I1lE9B8SdKmqOSFYc8BQ/JBQyQSsvbO/jXv8DFUYZW4NS2p25+7HNPaS8fTNqgIJd09PIuJHUzXsLBDz+EMm4DdS+0xY11Ha3tlpz04c03infDp9UzuCRL+cfkaphWTFSlLUcusc1sTA==

Получение данных о API

GET /api

В случае успеха возвращается код 200 и в теле ответа версия сервера и список возможных адресов, например:

```
{
	"Version": "3.0.15.7528",
	"Urns": ["/api/serverinfo", "/api/licences", "/api/connections", "/api/sessions"]
}
```

Где:

Version	Версия сервера
Urns	Массив возможных адресов

Получение информации о сервере

GET /api/serverinfo

В случае успеха возвращается код 200 и в теле ответа информация о сервере, например:

```
{
	"ComputerName": "MORA-HP",
	"LocalIP": "192.168.56.1,192.168.1.38",
	"OSType": "Windows",
	"OSVersion": "Windows 10 10.0.16299 (x86_64)",
	"ModuleName": "C:\\Program Files (x86)\\1C\\Licence\\v3.0\\licenceserver.exe",
	"PID": 5736,
	"CPU": "i386",
	"Version": "3.0.15.7528",
	"Started": "2018-04-02T20:41:13.272Z",
	"ComputerID": "D365ACC38180015CD3A30E62BF1072392D4DD5C1"
}
```

Где:

ComputerName	Имя компьютера, где выполняется сервер СЛК
LocalIP	Список локальных IPv4 адресов
OSType	Тип ОС (Windows / Linux)
OSVersion	Версия ОС
ModuleName	Исполняемый модуль сервера
PID	Идентификатор процесса
CPU	Разрядность процесса (i386 / x86_64)
Version	Версия сервера
Started	Время запуска
ComputerID	Идентификатор оборудования

Получение установленных лицензий

GET /api/licences

В случае успеха возвращается код 200 и в теле ответа массив лицензий, например:


```
{
	"Licences": [{
			"Type": "Virtual",
			"FileName": "C:\\ProgramData\\1C\\Licence\\Data\\672B.7121172.licence",
			. . .
		},
		{
			"Type": "Virtual",
			"FileName": "C:\\ProgramData\\1C\\Licence\\Data\\672B.7131767.licence",
			. . .
		},
		{
			"Type": "Virtual",
			"FileName": "C:\\ProgramData\\1C\\Licence\\Data\\672B.7181155.licence",
			. . .
		}
	]
}
```


Описание полей см. ниже в описании команды получения информации об отдельном ключе.

Получение информации о конкретном ключе / лицензии

GET /api/licences/{KeyNo}

Где:

KeyNo	Уникальный С/Н ключа

В случае успеха возвращается код 200 и в теле ответа информация о указанном ключе, например:


```
{
	"Type": "Virtual",
	"FileName": "C:\\ProgramData\\1C\\Licence\\Data\\672B.7131766.licence",
	"Enabled": true,
	"IsBlank": false,
	"Workable": true,
	"OpenError": null,
	"Version": "6",
	"KeyNo": 7131766,
	"KeyID": "672B",
	"RegNo": 202,
	"KeyType": 3,
	"ServiceType": 0,
	"LicenceCount": 5,
	"ProductVersion": 0,
	"BlankKeyNo": null,
	"ActivationCode": "4345-9320-2035-7421-5361",
	"ActivationDate": "2018-03-30T21:00:00.414Z",
	"StartDate": null,
	"Period": 0,
```

```
            "ExpireDate": null,
            "DaysLeft": null,
            "Expired": false,
            "Flags": "00000000000000000000000000000000",
            "ITN": 1234567890,
            "IEC": null,
            "ProductName": "Тестовая серия",
            "Article": "1111111111111",
            "ArticleName": "Тестовая номенклатура для проверки работоспособности системы (максимально 100 символов, выгруженных из ТС)"
}
```


Где:

Type	Тип ключа (Virtual – программный, USB – аппаратный)
FileName	Имя файла
OpenError	Ошибка открытия, если есть
Enabled	Общий признак доступности
Workable	Признак работоспособности, возвращает ложь если устройство неисправно или изменился идентификатор компьютера
Version	Версия внутреннего ПО или формата файла
KeyNo	Уникальный С/Н
KeyID	Серия
RegNo	Рег. номер
KeyType	Тип ключа: 3 – Основной, 4 – Дополнительный 5 – Демонстрационный
ServiceType	Тип сервиса при использовании ключа в специальных сервисах 1С: 0 – Обычный вариант использования 1 – Демонстрационный сервер 2 – ГРМ 3 – Fresh 4 – …
LicenceCount	Количество лицензий
ProductVersion	Версия продукта
BlankKeyNo	С/Н аппаратного носителя, если программный ключ установлен на нем
ActivationCode	Код активации
ActivationDate	Дата активации
StartDate	Дата начала действия
Period	Срок действия в днях

ExpireDate	Дата окончания действия
DaysLeft	Оставшийся срок действия в днях
Expired	Признак, что срок действия истек
Flags	Флаги функциональности в двоичном виде
ITN	ИНН
IEC	КПП
ProductName	Название продукта
Article	Артикул номенклатуры
ArticleName	Название артикула

Установка лицензии

POST /api/licences

В теле запроса передаются параметры установки лицензии:

ActivationCodeString	Код активации
ITN	ИНН
IEC	КПП
BlankKeyNo	С/Н аппаратного носителя, где необходимо установить программный ключ (устройство должно быть подключено в момент выполнения)

Например:

```
{
        "ActivationCodeString": "4345-9320-2035-7421-5361",
        "ITN": 1234567890,
        "IEC": null,
        "BlankKeyNo": null
}
```

В случае успеха возвращается код 201 и в теле ответа информация об установленной лицензии (описание полей см. в **получении информации о конкретном ключе / лицензии**).

Получение текущих соединений

GET /api/connections

В случае успеха возвращается код 200 и в теле ответа массив соединений, например:

```
{
        "Connections": [{
                        "ID": "AF72CC4124DD4B588AA9039D31035A5C",
                        . . .
```

```
                },
                {
                        "ID": "2F72DD41489D4B558AA9039D36035BF01",
                        . . .
        }]
}
```

Описание полей см. ниже в описании команды получения информации об отдельном соединении.

Получение информации о конкретном соединении

GET /api/connections/{ID}

Где:

ID	Идентификатор соединения

В случае успеха возвращается код 200 и в теле ответа информация об указанном соединении, например:

```
{
        "ID": "AF72CC4124DD4B588AA9039D31035A5C",
        "PeerIP": "127.0.0.1",
        "Created": "2018-04-03T14:06:59.317Z",
        "LastUsed": "2018-04-03T14:07:09.320Z",
        "OSVersion": "Windows 10 10.0.16299 (x86_64)",
        "ComputerName": "MORA-HP",
        "LocalIP": "192.168.56.1,192.168.1.38",
        "Process": "C:\\Program Files (x86)\\1cv8\\8.3.11.2954\\bin\\1cv8c.exe",
        "CPU": "i386",
        "PID": "8236",
        "IsServer": false,
        "PlatformVersion": "8.3.11.2954",
        "ModuleName": "C:\\Users\\MORA\\AppData\\Local\\Temp\\v8_3F2F_d.tmp",
        "Version": "3.0.15.7522",
        "KeyID": "672B",
        "ExpireTime": "2018-04-03T14:07:29.320Z",
        "Expired": "False"
}
```

Где:

ID	Идентификатор соединения
PeerIP	IP адрес
Created	Время подключения
LastUsed	Время последнего обращения
OSVersion	Версия ОС
ComputerName	Имя компьютера
LocalIP	Локальные IP адреса

Process	Имя файла процесса
CPU	Разрядность (i386 / x86_64)
PID	Идентификатор процесса
IsServer	Признак серверного процесса платформы 1С:Предприятия
PlatformVersion	Версия платфрормы
ModuleName	Имя модуля компоненты
Version	Версия компоненты
KeyID	Серия ключей
ExpireTime	Время, когда время жизни соединения истечет
Expired	Признак, что время жизни соединения истекло

Получение текущих сеансов

GET /api/sessions

В случае успеха возвращается код 200 и в теле ответа массив сеансов, например:

```
{
	"Sessions": [{
			"ID": "DB4AC14978FA4B5815A813550A031424B",
			. . .
		},
		{
			"ID": "AA3465414B672B33CEA19039537F31BF04",
			. . .
	}]
}
```

Описание полей см. ниже в описании команды получения информации об отдельном сеансе.

Получение информации о конкретном сеансе

GET /api/sessions/{ID}

Где:

ID	Идентификатор сеанса

В случае успеха возвращается код 200 и в теле ответа информация об указанном сеансе, например:

```
{
	"ID": "DB4AC14978FA4B5815A813550A031424B",
	"SessionNumber": "2",
	"IsServer": false,
	"InfoBaseConnectionStr": "File=\"D:\\Projects\\Licence\\v3.0\\DB\\DemoDb\";",
```

```
        "AppName": "1CV8C",
        "ComputerName": "MORA-HP",
        "UserName": null,
        "SessionStarted": "2018-04-03T14:06:58.000Z",
        "SessionConnection": "AF72CC4124DD4B588AA9039D31035A5C",
        "AccessCode": null,
        "ExpireTime": "2018-04-03T14:23:43.787Z",
        "Expired": "False",
        "KeyID": "672B",
        "UsedLicence": 7131766
}
```


Где:

ID	Идентификатор сеанса
SessionNumber	Номер сеанса
IsServer	Признак серверного сеанса
InfoBaseConnectionStr	Строка соединения ИБ
AppName	Приложение ИБ
ComputerName	Имя компьютера ИБ
UserName	Имя пользователь ИБ
SessionStarted	Время начала работы сеанса
SessionConnection	Идентификатор соединения, создавшего сеанс
AccessCode	Код доступа
KeyID	Серия ключей
ExpireTime	Время, когда время жизни сеанса истечет
Expired	Признак, что время жизни сеанса истекло
UsedLicence	С/Н использованной лицензии

Создание внешнего сеанса и получение им лицензии

POST /api/sessions

В теле запроса указываются параметры внешнего сеанса, который должен получить лицензию:

KeyID	Серия ключей
ClientSessionID	Внешний идентификатор сеанса
AppName	Название приложения
UserName	Имя пользователя
SessionStarted	Время начала сеанса в виде строки
ComputerName	Имя компьютера (или устройства)

Например:

```
{
        "KeyID": "672B",
        "ClientSessionID": "0123456789ABCDEF",
        "AppName": "MobileClient",
        "UserName": "MobileUser",
        "ComputerName": "Androud Table Sevice",
        "SessionStarted": "2020-01-01T12:52:20"
}
```

В случае успеха возвращается код 201 (если сеанс был создан) или 200 (если сеанс, с указанным ClientSessionID уже существует) и в теле ответа информация об занимаемой сеансом лицензии (описание полей см. в **получении информации о конкретном ключе / лицензии**).