

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

Сборка 3.0.31.10976, 2023-02-21

## 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
q9Ctj5o472wIC0R7iboPh8cFyiUS+6DoOTU8HFkpWpAchPy6SncxaFeCypad+1vq
eLk2v/jhE1M8WdotwK61MVQZ4DzxMyZ7JwNUz4ZbObOUCJaD3FwU/nca/p9rCeYj
NER1rYujEgdJ9G7trvdu8r2OzsANsUBmAA65vNLKkONDSMtTr+XaN9L4tp6AfZFu
mRvVPJMEV2rJ2dExs8yiLq6a08Vx8HvIOBmQYRtBBjxycgbFHB74JVZNjqonKSSE
7Zznii1eETsgtEMCagpC5/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/0ukaFAohMWhZRt3zQUtdqBnP9LDEcksWet1S9mc2qj3I9Y4xYsFV7m3udp65CXaqH8Ck1j/Rdt3Y2z3RCL yaYuopYpKyfxZv492R4lOrt6NdRs6RdCUVES3fX166SdWHP+HnudpcPiwSc+hTntfOgB4jIBQjSAaYWarT9Ko mYXTO/musMbQwMUBJyA2pE8u8Wc1VWkk+I1IE9B8SdKmqOSFYc8BQ/JBQyQSsvbO/jXv8DFUYZW4NS2 p25+7HNPas8fTNqgIJd09PIuJHuzXsLBDz+EMm4DdS+0xY11Ha3tlpz04c03infDp9UzuCRL+cfkaphWTFsIL Ucusc1sTA==

## 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 конкретной 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": "111111111111",
  "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 C/H
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
ProductVersionName	Наименование версии, если установлено в Центре Лицензирования
BlankKeyNo	C/H 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
Message	Информационное сообщение, установленное для данного

	ключа в Центре Лицензирования
MessageType	Тип сообщения: 0 – информационное 1 – предупреждение 2 – ошибка
Recovered	Признак, что ключ был восстановлен командой восстановления
SupportDate	Дата начала действия договора ИТС для данного рег. номера
SupportEndDate	Дата окончания действия договора ИТС
SupportPeriod	Срок действия договора ИТС в днях
SupportType	Тип договора ИТС: 0 – неизвестный / неопределенный 1 – 1С:ИТС Проф 2 – 1С:ИТС Техно
IndustrySupportDate	Дата начала действия сервиса ИТС Отраслевой для данного рег. номера
IndustrySupportEndDate	Дата окончания действия сервиса ИТС Отраслевой
IndustrySupportPeriod	Срок действия сервиса ИТС Отраслевой в днях
IndustrySupportType	Тип сервиса: 0 – неизвестный / неопределенный 1 – 1С:ИТС Отраслевой Базовый 2 – 1С:ИТС Отраслевой 1-й Категории 3 – 1С:ИТС Отраслевой 2-й Категории 4 – 1С:ИТС Отраслевой 3-й Категории 5 – 1С:ИТС Отраслевой 4-й Категории 6 – 1С:ИТС Отраслевой 5-й Категории
UpdateDate	Дата и время последнего обновления

## Чтение пользовательской памяти

### GET /api/licences/{KeyNo}/memory[?Password={Password}]

Где:

KeyNo	Уникальный С/Н ключа
Password	Пароль на чтение

Пароль может быть передан или в параметре запроса (см. выше), или в его теле:

```
{
  "Password": "123456"
}
```

Если пароль не указан, то считается равным пустой строке.

В случае успеха возвращается код **200** и в теле ответа структура, содержащая в поле **Value** прочитанное значение, например:

```
GET /api/licences/7121173/memory
200 OK
{
  "KeyNo": 7121173,
  "Value": "Это обычная модифицированная память"
}
```

## Запись пользовательской памяти

**POST /api/licences/{KeyNo}/memory[?Password={Password} [&Value={Value}]]**

Где:

KeyNo	Уникальный С/Н ключа
Password	Пароль на запись
Value	Значение

Пароль и данные для могут быть переданы или в параметрах запроса (см. выше), или в его теле:

```
{
  "Password": "123456",
  "Value": "Это модифицированное значение"
}
```

Если какое-то из значений не указано, то оно считается равным пустой строке.

В случае успеха возвращается код **202**.

## Установка паролей на доступ к пользовательской памяти

**POST**

**/api/licences/{KeyNo}/memory[?ReadPassword={ReadPassword}&WritePassword={WritePassword}&NewReadPassword={NewReadPassword}&NewWritePassword={NewWritePassword}]]**

Где:

KeyNo	Уникальный С/Н ключа
ReadPassword	Текущий пароль на чтение
WritePassword	Текущий пароль на запись
NewReadPassword	Новый пароль на чтение
NewWritePassword	Новый пароль на запись

Значения паролей могут быть переданы или в параметрах запроса (см. выше), или в его теле:

```
{
```



```

"ReadPassword": "123456",
"WritePassword": "123456",
"NewReadPassword": "",
"NewWritePassword": "111111",
}

```

Если какое-то из значений не указано, то оно считается равным пустой строке.

В случае успеха возвращается код [202](#).

## Сброс пользовательской памяти

### **DELETE /api/licences/{KeyNo}/memory**

Где:

KeyNo                      Уникальный С/Н ключа

Команда выполняет установку памяти и паролей в пустые значения, в случае успеха возвращается код [202](#).

## 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](#)).

## Обновление / восстановление лицензий

### POST /api/licences/update

В случае успеха возвращается код 200 и в теле ответа информация об обновленных или восстановленных лицензиях аналогично команде **получения лицензий**.

## 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 платформы
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)
InfoBaseName **	Имя информационной базы
AccessCode **	Код доступа
KeyNo **	С/Н конкретного ключа, лицензию которого необходимо получить
ITN **	ИНН
IEC **	КПП

\* Необязательные информационные параметры

\*\* Необязательные параметры отбора лицензий; при их указании лицензия сможет получена только при наличии ключей, параметры которых соответствуют указанным

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](#)).