

Программное обеспечение «Litoria Crypto Service 2»
Руководство системного программиста

Аннотация

В документе приводится руководство системного программиста программного обеспечения «Litoria Crypto Service 2» версия 2.2.1.1 (далее – ПО «Litoria Crypto Service 2» или приложение).

В разделе «Назначение, функции и состав приложения» приводятся сведения о назначении приложения, перечислены основные функции и приведено краткое описание функций.

В разделе «Условия применения» указаны условия, необходимые для использования приложения, требования к аппаратным средствам автоматизированного рабочего места (АРМ).

В разделе «Настройка и активация ПО «Litoria Crypto Service 2» описаны действия, которые необходимо выполнить перед использованием приложения.

В разделе «Описание функций REST интерфейса ПО «Litoria Crypto Service 2» описаны действия, необходимые для обращения к основным функциям приложения.

В конце документа приведен список использованных сокращений.

Содержание

1	Назначение, функции и состав приложения	5
1.1	Назначение приложения.....	5
1.2	Описание функций ПО «Litoria Crypto Service 2».....	5
1.2.1	Управление ключевой информацией и сертификатами.....	5
1.2.2	Управление списком криптопровайдеров.....	6
1.2.3	Подписание и шифрование файла	6
1.2.4	Пролонгация подписи	6
1.2.5	Проверка подписи	6
1.2.6	Извлечение данных.....	7
1.2.7	Присоединение подписи.....	7
1.2.8	Использование службы ДТС для проверки ЭП.....	7
1.2.9	Получение информации о МЧД.....	7
2	Условия применения.....	9
2.1	Требования к техническим средствам.....	9
2.2	Требования к программному обеспечению.....	9
3	Запуск и активация ПО «Litoria Crypto Service 2»	10
3.1	Запуск ПО «Litoria Crypto Service 2»	10
3.1.1	Запуск ПО «Litoria Crypto Service 2» в ОС семейства Windows	10
3.1.2	Запуск ПО «Litoria Crypto Service 2» в ОС семейства Linux	11
3.2	Активация приложения	12
3.2.1	Активация ознакомительной лицензии онлайн.....	12
3.2.2	Получение ключа активации	14
3.2.3	Активация онлайн.....	17
3.2.4	Активация офлайн.....	19
3.2.5	Проверка статуса лицензии.....	21
3.2.6	Web-активация	22
4	Описание функций REST интерфейса ПО «Litoria Crypto Service 2»	23
4.1	Общие настройки для работы ПО «Litoria Crypto Service 2».....	23
4.1.1	Настройка пути для логирования операций	23
4.1.2	Настройка работы с прокси-сервером	24
4.1.3	Настройка подключения к базе данных.....	24

4.1.4	Настройка выбора криптопровайдера для ОС Linux	25
4.1.5	Настройка квалифицированного режима	26
4.1.5.1	Скачивание/обновление списка аккредитованных УЦ	27
4.1.6	Получение заданных настроек	29
4.2	Настройка автоматической пролонгации подписи	34
4.2.1	Регистрация и запуск службы	34
4.2.2	Настройка службы	35
4.3	Обращение к хранилищу сертификатов	36
4.3.1	Получение всех сертификатов из хранилища	36
4.3.2	Установка сертификата в хранилище	38
4.3.3	Удаление сертификата из хранилища	39
4.3.4	Получение информации о сертификате	39
4.4	Создание запроса на сертификат или самоподписанного сертификата	48
4.5	Получение списка криптопровайдеров	51
4.6	Подпись и шифрование	54
4.6.1	Подпись отдельных секций XAdES	58
4.7	Пролонгация подписи	60
4.8	Проверка подписи	63
4.9	Извлечение данных	69
4.10	Хеширование	71
4.11	Подпись хеша и создание отделенной подписи	72
4.12	Присоединение подписи	74
4.13	DVCS-запрос и разбор квитанции	76
4.14	Обращение к базе данных	87
4.14.1	Добавление информации в базу данных	87
4.14.2	Поиск информации в базе данных	89
4.15	Проверка соответствия ДТС квитанции данным	91
4.16	Аутентификация запросов по токену безопасности	93
4.17	Получение отчета о проверке подписи в ДТС	93
4.18	Получение информации о машиночитаемой доверенности	95
5	Возвращаемые значения и коды ответов	101
	Перечень сокращений	106

1 Назначение, функции и состав приложения

1.1 Назначение приложения

Основным назначением ПО «Litoria Crypto Service 2» является предоставление пользователю доступа к базовым функциям криптографической платформы «Litoria Crypto Platform» через REST интерфейс.

Базовые функции криптографической платформы «Litoria Crypto Platform» включают в себя операции создания электронной подписи (ЭП) и шифрования, извлечения и проверки ЭП (в том числе локальная проверка ЭП с машиночитаемой доверенностью (МЧД)), функции просмотра хранилища сертификатов; создания запроса на выпуск сертификата; установки сертификата из устройства в хранилище; просмотра установленных криптопровайдеров и их параметров; пролонгации подписи; использования службы доверенной третьей стороны (ДТС) для проверки ЭП (в том числе проверка ЭП с МЧД с использованием службы ДТС), просмотр информации о МЧД.

1.2 Описание функций ПО «Litoria Crypto Service 2»

При использовании ПО «Litoria Crypto Service 2» пользователю доступен весь комплект необходимых функций для работы с ЭП:

- создание запроса на сертификат, в т.ч. самоподписанного;
- просмотр сертификатов в контейнерах;
- установка сертификата в хранилище сертификатов;
- управление сертификатами:
- просмотр списка криптопровайдеров;
- создание ЭП и шифрование для файлов произвольного типа;
- добавление ЭП для подписанных файлов;
- извлечение и проверка ЭП для подписанных файлов с возможностью получения исходного документа;
- пролонгация подписи, в случае истечения срока действия;
- использование службы ДТС для проверки ЭП.

1.2.1 Управление ключевой информацией и сертификатами

С помощью приложения можно выполнять следующие функции с сертификатами:

- получение сертификата(-ов) из хранилища;
- установка сертификата в заданное хранилище;
- удаление сертификата из хранилища;
- получение информации о сертификате;
- создание запроса на сертификат или самоподписанного сертификата.

1.2.2 Управление списком криптопровайдеров

С помощью приложения можно выполнить просмотр установленных на компьютере криптопровайдеров и получить информацию о них.

1.2.3 Подписание и шифрование файла

Приложение предоставляет возможность одновременного создания ЭП и шифрования файла.

Для создания ЭП должен быть осуществлен выбор сертификата ключа проверки подписи и параметров создания ЭП (отделенная ЭП или совмещенная, с вложенным штампом времени, усовершенствованная ЭП).

Шифрование производится с использованием ключа проверки ЭП, содержащегося в сертификате получателя. Ключ ЭП есть только у владельца использованного сертификата ключа проверки ЭП. Таким образом, при шифровании файла никто, кроме владельца ключа ЭП, не сможет расшифровать файл.

1.2.4 Пролонгация подписи

В ПО «Litoria Crypto Service 2» реализована поддержка электронных документов длительного архивного хранения включая доказательство момента подписи документа и действительность сертификата ключа подписи при создании ЭП и проверке ее корректности и возможность доказательства корректности подписи и целостности файла после истечения срока действия сертификата подписи.

При истечении срока действия ключа проверки подписи службы штампа времени, для продления срока доверенного архивного хранения электронных документов, предоставляется возможность добавления к подписи архивного штампа времени.

1.2.5 Проверка подписи

Проверка ЭП подразумевает подтверждение подлинности электронной подписи в электронном документе, то есть:

- принадлежности электронной подписи в электронном документе владельцу сертификата ключа проверки подписи;

- отсутствия искажений в подписанном данной электронной подписью электронном документе;
- подтверждение момента подписи;
- подтверждение действительности сертификата ключа проверки подписи на момент проверки либо на момент создания ЭП при наличии в подписи доказательств, определяющих этот момент.

В том числе возможна проверка ЭП с машиночитаемой доверенностью.

Проверка усовершенствованной ЭП файла с отделенной подписью – проверка корректности самого файла подписи.

1.2.6 Извлечение данных

При получении зашифрованного документа извлечение пройдет успешно при условии наличия ключа ЭП, связанного с одним из ключей проверки ЭП, на которых производилось шифрование файла. Если существует несколько ключей ЭП, которым соответствуют несколько ключей проверки ЭП, участвующих при шифровании, то расшифровывание произойдет на первом из ключей ЭП. После извлечения можно получить информацию о том, на каком сертификате была произведена операция расшифровывания.

1.2.7 Присоединение подписи

В случае, когда в подписании документа участвует несколько лиц, и каждый должен поставить в нем свою подпись (например, в листе согласования), используется операция добавления ЭП. В отличие от операции создания ЭП, добавление ЭП производится в уже подписанный ранее документ.

1.2.8 Использование службы ДТС для проверки ЭП

ПО «Litoria Crypto Service 2» позволяет получать подтверждение корректности ЭП электронного документа (Validation of Digitally Signed Document – VSD) от службы ДТС.

При наличии машиночитаемой доверенности (МЧД) также осуществляется проверка соответствия полномочий, определенных МЧД, в соответствии с требованиями Федерального закона № 63-ФЗ и соответствующих подзаконных актов.

Для обращения к службе необходимо наличие личного сертификата ключа проверки ЭП, зарегистрированного ранее на сервере ДТС. Подписанный указанным личным сертификатом DVCS-запрос отправляется для проверки на сервер службы ДТС. В ответ сервер присылает информацию о действительности ЭП документа.

1.2.9 Получение информации о МЧД

МЧД – доверенность, созданная в электронной форме и подписанная электронной

подписью лица, уполномоченного на действия от имени участника информационного обмена без доверенности. В МЧД указываются именно те полномочия, которые руководитель доверяет конкретному представителю.

ПО «Litoria Crypto Service 2» позволяет получить информацию о машиночитаемой доверенности при ее наличии в ЭП.

2 Условия применения

2.1 Требования к техническим средствам

Минимальные требования к рабочей станции, на которую устанавливается ПО «Litoria Crypto Service 2», обусловлены применением используемых ОС.

2.2 Требования к программному обеспечению

ПО «Litoria Crypto Service 2» функционирует под управлением следующих ОС:

- Windows 7 (32 бит/64 бит) ¹;
- Windows 8/8.1 (32 бит/64 бит);
- Windows 10 (32 бит/64 бит).
- Windows Server 2008 R2;
- Windows Server 2012/2012R2/2016R2/2019R2/2022 (64 бит);

и ОС семейств Linux, поддерживающих системную библиотеку GNU C Library (Glibc) версии не ниже 2.22.

Гарантирована работа ПО «Litoria Crypto Service 2» на следующих ОС семейства Linux:

- AltLinux версии 8.2 или выше (64 бит);
- Ubuntu версии 16 или выше (64 бит);
- RedOS версии 7.2, 7.3 (64 бит);
- Astra Linux Special Edition версии 1.6, 1.7 (64 бит).

Дополнительно должно быть установлено следующее программное обеспечение:

- Средство криптографической защиты информации, реализованное в соответствии с технологией Microsoft CSP (при работе с отечественной криптографией необходимо использовать соответствующие СКЗИ)².
- СКЗИ «КриптоПро CSP» с поддержкой графического представления или ViPNet CSP.³
- СУБД Postgre SQL.

Рекомендуется установка Advanced REST Client или любого другого REST клиента.

Для выполнения web-активации сервиса лицензирования необходим Chrome совместимый браузер.

¹ Срок эксплуатации ОС определяется сроками выпуска обновлений критических уязвимостей.

² При функционировании ПО на ОС семейства Windows.

³ При функционировании ПО на ОС семейства Linux.

3 Запуск и активация ПО «Litoria Crypto Service 2»

3.1 Запуск ПО «Litoria Crypto Service 2»

Приложение «Litoria Crypto Service 2» обращается к базовым функциям криптографической платформы «Litoria Crypto Platform» через REST интерфейс. После запуска приложения запускается локальная веб-служба, принимающая REST запросы в соответствии с представленным ниже интерфейсом.

Для выполнения функций приложения можно использовать любой REST клиент, например, «Advanced REST client».

При запуске ПО «Litoria Crypto Service 2» с помощью ярлыка показателем успешного запуска приложения, является значок  на панели задач.

3.1.1 Запуск ПО «Litoria Crypto Service 2» в ОС семейства Windows

ПО «Litoria Crypto Service 2» в ОС семейства Windows, начиная с версии 2.2.1.1, может быть запущен от имени обычного пользователя, от имени пользователя с правами администратора, от встроенного администратора и в качестве службы.

При первом запуске от имени пользователя (независимо от наличия у него прав администратора) конфигурационный файл для ПО «Litoria Crypto Service 2» создается по пути «C:\Users*имя пользователя*\AppData\Roaming\GIS\litoriacs» с адресом <http://127.0.0.1>, номер порта определяется из числа свободных в диапазоне 8001-8100:

```
main:
{
  url = "http://127.0.0.1:8001";
};
```

Для работы ПО «Litoria Crypto Service 2» из-под учетной записи пользователя необходимо, чтобы был указан адрес приложения <http://127.0.0.1>. Если в более ранних версиях ПО «Litoria Crypto Service 2» для пользователя был настроен адрес с указанием IP-адреса машины, необходимо изменить адрес на <http://127.0.0.1>, иначе приложение не будет работать.

При первом запуске от встроенного администратора ПО «Litoria Crypto Service 2» определяет возможный для работы IP-адрес и порт с учетом всех ограничений ОС, и записывает этот адрес в конфигурационный файл, создаваемый по пути «C:\Users*имя встроенного администратора*\AppData\Roaming\GIS\litoriacs».

Запуск ПО «Litoria Crypto Service 2» в качестве службы необходимо выполнять для исполняемого файла «*litoriacs.exe*», расположенного в каталоге «*service*» по пути установки ПО (по умолчанию, *C:\Program Files (x86)\GIS\Litoria Crypto Service 2*).

Если служба ПО «Litoria Crypto Service 2» будет работать от системной учетной записи, конфигурационный файл будет расположен по пути «*C:\Windows\SysWOW64\config\systemprofile\AppData\Roaming\GIS\litoriacs*» или «*C:\Windows\System32\config\systemprofile\AppData\Roaming\GIS\litoriacs*». При запуске ПО «Litoria Crypto Service 2» как службы, определение его адреса осуществляется путем выбора возможного для работы IP-адреса и порта с учетом всех ограничений ОС.

Активация ПО «Litoria Crypto Service 2» для пользователя должна выполняться при запуске приложения от встроенного администратора, потому что только при запуске от встроенного администратора или в качестве службы в ветке реестра создается параметр адреса сервера лицензирования.

После установки и запуска ПО «Litoria Crypto Service 2» из-под учетной записи администратора адрес сервера лицензирования создается в ветке реестра:

- *\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\GIS\Litoria Crypto Service 2* – для 32-битной версии ПО;
- *\HKEY_LOCAL_MACHINE\SOFTWARE\GIS\Litoria Crypto Service 2* – для 64-битной версии ПО.

Значение лицензии и закрытого ключа после активации ПО «Litoria Crypto Service 2» также создаются в этих ветках реестра.

3.1.2 Запуск ПО «Litoria Crypto Service 2» в ОС семейства Linux

Для начала работы с ПО «Litoria Crypto Service 2» необходимо запустить исполняемый файл «*litoriacs*» по пути установки приложения на APM с правами администратора.

Для просмотра IP-адреса и порта, на котором запущен сервис, нужно зайти в конфигурационный файл или выполнить команду:

```
sudo systemctl status litoriacs.service
```

После первого запуска приложения конфигурационный файл по умолчанию содержит информацию об адресе сервера лицензирования:

```
activation:
{
  server = "https://license.gaz-is.ru";
};
```

<code>server</code>	Адрес сервиса лицензирования
---------------------	------------------------------

Выполнять запуск и работать с ПО «Litoria Crypto Service 2» должен только пользователь, обладающий правами администратора.

3.2 Активация приложения

Для проведения активации ПО «Litoria Crypto Service 2» должно быть запущено от имени администратора или в качестве службы.

3.2.1 Активация ознакомительной лицензии онлайн

Для активации ознакомительной лицензии приложения «Litoria Crypto Service 2» при имеющемся подключении к сети Интернет необходимо использовать функцию «*activate*», имеющую следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/activate>

JSON:

```
{  
  "trialactivation" : bool  
}
```

<code>"trialactivation"</code>	Активация ознакомительной лицензии. Принимает значение <i>true/false</i> .
--------------------------------	--

Ознакомительная лицензия доступна пользователю в течении 30 дней. При использовании ознакомительной лицензии, в ответах приложения «Litoria Crypto Service 2» будет присутствовать информация о том, что используется ознакомительная лицензия:

```
"licenseinfo" : "Триальная лицензия \"Litoria Crypto Service\" истекает через  
N д."
```

где *N* – количество дней, оставшееся до истечения лицензии.

В результате успешного выполнения операции возвращается ответ в формате JSON:

```
{  
  activation:
```

```
{
  "status" : "",
  "errorcode": "",
  "message": ""
}
```

<p>"status"</p>	<p>Статус активации ПО «Litoria Crypto Service 2». Может принимать следующие значения:</p> <p><i>Undefined</i> = 0,</p> <p><i>Success</i> = 1 – в случае успешного выполнения операции,</p> <p><i>ActivationRegistrationError</i> = 8 – невозможно выполнить активацию до регистрации ключа,</p> <p><i>ActivationError</i> = 9 – ошибка активации</p>
<p>"errorcode"</p>	<p>Код ошибки. Может принимать следующие значения:</p> <p><i>UNDEFINED</i> = 101 – статус не определен,</p> <p><i>BAD_REQUEST</i> = 102 – неверный запрос,</p> <p><i>NO_LICENSE_IN_REQUEST</i> = 103 – лицензия не содержится в запросе,</p> <p><i>LICENSE_KEY_NOT_EXIST</i> = 104 – данный серийный номер не существует,</p> <p><i>LICENSE_KEY_NOT_ACTIVE</i> = 105 – данный серийный номер не активен,</p> <p><i>EMAIL_NOT_EXIST</i> = 106 – данный адрес электронной почты не существует,</p> <p><i>LICENSE_KEY_CANT_BE_ACTIVATED</i> = 107 – данный серийный номер больше не может быть активирован,</p> <p><i>ACTIVATION_NOT_FOUND</i> = 108 – не найдена ни одна активация продукта,</p> <p><i>BAD_ACTIVATION_KEY</i> = 109 – неверный ключ активации,</p> <p><i>ACTIVATION_KEY_EXPIRED</i> = 110 – время на выполнение активации истекло,</p> <p><i>CA_ERROR</i> = 111 – ошибка УЦ,</p> <p><i>LICENSE_KEY_CANT_REACTIVATED_FOR_MACHINE</i> = 112 – реактивация на данном устройстве невозможна,</p> <p><i>MISMATCH_LICENSE_KEY_PRODUCT</i> = 113 – не найден лицензионный ключ и продукт,</p> <p><i>NO_LICENSE_REQUEST_IN_REQUEST</i> = 114 – запрос на регистрацию не содержит запрос,</p>

	<p><i>NO_EMAIL_IN_REQUEST</i> = 115 – адрес электронной почты не содержится в запросе, <i>NO_PRODUCT_IN_REQUEST</i> = 116 – продукт не найден в запросе, <i>GET_COUNT_CURRENT_ACTIVATION</i> = 117 – не удалось получить количество активированных копий, <i>FAILED_GET_LICENSE</i> = 118 – ошибка получения файла лицензии, <i>SEND_MAIL_ERROR</i> = 119 – ошибка отправки ключа подтверждения по почте, <i>REG_REQUEST_NOT_EXIST</i> = 120 – ошибка создания запроса на регистрацию, <i>HARDWAREDIFFERS</i> = 121 – несоответствие оборудования, <i>UPDATELICENSENOTFOUND</i> = 122 – ошибка обновления лицензии, <i>ACTIVATIONNOTACTIVE</i> = 123 – действие лицензии приостановлено, <i>LICENSEKEYEXPIRED</i> = 124 – срок действия лицензии истек</p>
"message"	Информационное сообщение

3.2.2 Получение ключа активации

Для получения ключа активации приложения «Litoria Crypto Service 2» необходимо использовать функцию «register», имеющую следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/register>

JSON:

```
{
  "licensekey" : "",
  "email" : "",
  "forcereactivation" : bool,
  "license" : ""
}
```

"licensekey"	Ключ лицензии для активации
"email"	Адрес электронной почты для активации лицензии. На этот адрес также поступит файл лицензии
"forcereactivation"	Флаг принудительной реактивации. Принимает значение <i>true/false</i> . В случае <i>true</i> – происходит реактивация лицензии.

	Используется опционально, указывается только при реактивации
<code>"license"</code>	Старая лицензия для реактивации. Используется в случае, когда на один адрес электронной почты зарегистрировано больше одной лицензии (администраторская лицензия). Используется опционально, указывается только при реактивации

Возможные варианты ответа системы:

```
{
  registration:
  {
    "status" : "",
    "errorcode": ""
    "message" : "",
    "timevalid" : "",
    "email": "",
    "offlinerequest" : ""
  }
}
```

<code>"status"</code>	Тип ответа системы. Может принимать следующие значения: <i>Undefined</i> = 0, <i>Success</i> = 1 – в случае успешного выполнения запроса, <i>LicenseKeyError</i> = 2 – ошибка ключа лицензии, <i>ReactivationAvailable</i> = 3 – необходима реактивация, <i>LicenseReactivationError</i> = 4 – отсутствует лицензия для реактивации, <i>EmailError</i> = 5 – отсутствие электронного адреса для регистрации, <i>RegistrationError</i> = 6 – ошибка регистрации, <i>RegistrationOffline</i> = 7 – регистрация офлайн
<code>"errorcode"</code>	Код ошибки. Может принимать следующие значения: <i>UNDEFINED</i> = 101 – статус не определен, <i>BAD_REQUEST</i> = 102 – неверный запрос, <i>NO_LICENSE_IN_REQUEST</i> = 103 – лицензия не содержится в запросе, <i>LICENSE_KEY_NOT_EXIST</i> = 104 – данный серийный номер не существует, <i>LICENSE_KEY_NOT_ACTIVE</i> = 105 – данный серийный номер не

	<p>активен,</p> <p><i>EMAIL_NOT_EXIST</i> = 106 – данный адрес электронной почты не существует,</p> <p><i>LICENSE_KEY_CANT_BE_ACTIVATED</i> = 107 – данный серийный номер больше не может быть активирован,</p> <p><i>ACTIVATION_NOT_FOUND</i> = 108 – не найдена ни одна активация продукта,</p> <p><i>BAD_ACTIVATION_KEY</i> = 109 – неверный ключ активации,</p> <p><i>ACTIVATION_KEY_EXPIRED</i> = 110 – время на выполнение активации истекло,</p> <p><i>CA_ERROR</i> = 111 – ошибка УЦ,</p> <p><i>LICENSE_KEY_CANT_REACTIVATED_FOR_MACHINE</i> = 112 – реактивация на данном устройстве невозможна,</p> <p><i>MISMATH_LICENSE_KEY_PRODUCT</i> = 113 – не найден лицензионный ключ и продукт,</p> <p><i>NO_LICENSE_REQUEST_IN_REQUEST</i> = 114 – запрос на регистрацию не содержит запрос,</p> <p><i>NO_EMAIL_IN_REQUEST</i> = 115 – адрес электронной почты не содержится в запросе,</p> <p><i>NO_PRODUCT_IN_REQUEST</i> = 116 – продукт не найден в запросе,</p> <p><i>GET_COUNT_CURRENT_ACTIVATION</i> = 117 – не удалось получить количество активированных копий,</p> <p><i>FAILED_GET_LICENSE</i> = 118 – ошибка получения файла лицензии,</p> <p><i>SEND_MAIL_ERROR</i> = 119 – ошибка отправки ключа подтверждения по почте,</p> <p><i>REG_REQUEST_NOT_EXIST</i> = 120 – ошибка создания запроса на регистрацию,</p> <p><i>HARDWAREDIFFERS</i> = 121 – несоответствие оборудования,</p> <p><i>UPDATELICENSENOTFOUND</i> = 122 – ошибка обновления лицензии,</p> <p><i>ACTIVATIONNOTACTIVE</i> = 123 – действие лицензии приостановлено,</p> <p><i>LICENSEKEYEXPIRED</i> = 124 – срок действия лицензии истек</p>
"message"	Информационное сообщение

"timevalid"	Время до окончания действия ключа активации
"email"	Адрес электронной почты, на который отправляется ключ активации и файл лицензии
"offlinerequest"	Запрос в случае офлайн активации

3.2.3 Активация онлайн

Для активации приложения «Litoria Crypto Service 2» необходимо использовать функцию «*activate*», имеющую следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/activate>

JSON:

```
{
  "activekey" : ""
}
```

"activekey"	Код активации для лицензии
-------------	----------------------------

В результате успешного выполнения операции возвращается ответ в формате JSON:

```
{
  activation:
  {
    "status" : "",
    "errorcode": "",
    "message": ""
  }
}
```

"status"	Статус активации ПО «Litoria Crypto Service 2». Может принимать следующие значения: <i>Undefined = 0</i> , <i>Success = 1</i> – в случае успешного выполнения операции, <i>ActivationRegistrationError = 8</i> – невозможно выполнить активацию до регистрации ключа, <i>ActivationError = 9</i> – ошибка активации
----------	---

Код ошибки. Может принимать следующие значения:

UNDEFINED = 101 – статус не определен,

BAD_REQUEST = 102 – неверный запрос,

NO_LICENSE_IN_REQUEST = 103 – лицензия не содержится в запросе,

LICENSE_KEY_NOT_EXIST = 104 – данный серийный номер не существует,

LICENSE_KEY_NOT_ACTIVE = 105 – данный серийный номер не активен,

EMAIL_NOT_EXIST = 106 – данный адрес электронной почты не существует,

LICENSE_KEY_CANT_BE_ACTIVATED = 107 – данный серийный номер больше не может быть активирован,

ACTIVATION_NOT_FOUND = 108 – не найдена ни одна активация продукта,

BAD_ACTIVATION_KEY = 109 – неверный ключ активации,

ACTIVATION_KEY_EXPIRED = 110 – время на выполнение активации истекло,

CA_ERROR = 111 – ошибка УЦ,

"errorcode"

LICENSE_KEY_CANT_REACTIVATED_FOR_MACHINE = 112 – реактивация на данном устройстве невозможна,

MISMATCH_LICENSE_KEY_PRODUCT = 113 – не найден лицензионный ключ и продукт,

NO_LICENSE_REQUEST_IN_REQUEST = 114 – запрос на регистрацию не содержит запрос,

NO_EMAIL_IN_REQUEST = 115 – адрес электронной почты не содержится в запросе,

NO_PRODUCT_IN_REQUEST = 116 – продукт не найден в запросе,

GET_COUNT_CURRENT_ACTIVATION = 117 – не удалось получить количество активированных копий,

FAILED_GET_LICENSE = 118 – ошибка получения файла лицензии,

SEND_MAIL_ERROR = 119 – ошибка отправки ключа подтверждения по почте,

REG_REQUEST_NOT_EXIST = 120 – ошибка создания запроса на регистрацию,

HARDWAREDIFFERS = 121 – несоответствие оборудования,

UPDATELICENSENOTFOUND = 122 – ошибка обновления лицензии,

ACTIVATIONNOTACTIVE = 123 – действие лицензии приостановлено,

	<i>LICENSEKEYEXPIRED = 124</i> – срок действия лицензии истек
"message"	Информационное сообщение

3.2.4 Активация офлайн

Для активации приложения «Litoria Crypto Service 2» офлайн (при отсутствии сети Интернет) необходимо использовать функцию «*license*», имеющую следующий синтаксис:

Content-Type: *application/json*

Метод: *PUT, POST*

URL: <http://localhost:8080/license>

JSON:

```
{  
  "license" : ""  
}
```

"license"	Лицензия (данные), полученная из центра офлайн активации продуктов ООО «Газинформсервис», для офлайн активации приложения
-----------	---

Результатом успешного выполнения операции является ответ в формате JSON:

```
{  
  "errorcode": <Код_ошибки>,  
  "message": "",  
  "status": <Статус_активации>  
}
```

"errorcode"	Код ошибки. Принимает следующие значения: <i>UNDEFINED = 101</i> – статус не определен, <i>BAD_REQUEST = 102</i> – неверный запрос, <i>NO_LICENSE_IN_REQUEST = 103</i> – лицензия не содержится в запросе, <i>LICENSE_KEY_NOT_EXIST = 104</i> – данный серийный номер не существует, <i>LICENSE_KEY_NOT_ACTIVE = 105</i> – данный серийный номер не активен, <i>EMAIL_NOT_EXIST = 106</i> – данный адрес электронной почты не существует, <i>LICENSE_KEY_CANT_BE_ACTIVATED = 107</i> – данный серийный номер
-------------	---

	<p>больше не может быть активирован, <i>ACTIVATION_NOT_FOUND</i> = 108 – не найдена ни одна активация продукта, <i>BAD_ACTIVATION_KEY</i> = 109 – неверный ключ активации, <i>ACTIVATION_KEY_EXPIRED</i> = 110 – время на выполнение активации истекло, <i>CA_ERROR</i> = 111 – ошибка УЦ, <i>LICENSE_KEY_CANT_REACTIVATED_FOR_MACHINE</i> = 112 – реактивация на данном устройстве невозможна, <i>MISMATH_LICENSE_KEY_PRODUCT</i> = 113 – не найден лицензионный ключ и продукт, <i>NO_LICENSE_REQUEST_IN_REQUEST</i> = 114 – запрос на регистрацию не содержит запрос, <i>NO_EMAIL_IN_REQUEST</i> = 115 – адрес электронной почты не содержится в запросе, <i>NO_PRODUCT_IN_REQUEST</i> = 116 – продукт не найден в запросе, <i>GET_COUNT_CURRENT_ACTIVATION</i> = 117 – не удалось получить количество активированных копий, <i>FAILED_GET_LICENSE</i> = 118 – ошибка получения файла лицензии, <i>SEND_MAIL_ERROR</i> = 119 – ошибка отправки ключа подтверждения по почте, <i>REG_REQUEST_NOT_EXIST</i> = 120 – ошибка создания запроса на регистрацию, <i>HARDWAREDIFFERS</i> = 121 – несоответствие оборудования, <i>UPDATELICENSENOTFOUND</i> = 122 – ошибка обновления лицензии, <i>ACTIVATIONNOTACTIVE</i> = 123 – действие лицензии приостановлено, <i>LICENSEKEYEXPIRED</i> = 124 – срок действия лицензии истек</p>
<p>"message"</p>	<p>Информационное сообщение</p>
<p>"status"</p>	<p>Статус активации ПО «Litoria Crypto Service 2». Принимает значения: <i>Success</i> = 1 – успешно, <i>LicenseKeyError</i> = 2 – ошибка ключа лицензии, <i>ReactivationAvailable</i> = 3 – необходима реактивация, <i>LicenseReactivationError</i> = 4 – отсутствие лицензии для реактивации, <i>EmailError</i> = 5 – отсутствие электронного адреса для регистрации, <i>RegistrationError</i> = 6 – ошибка регистрации, <i>RegistrationOffline</i> = 7 – регистрация офлайн,</p>

ActivationRegistrationError = 8 – невозможность выполнения активации до регистрации ключа,
ActivationError = 9 – ошибка активации,
LicenseError = 10 – ошибка лицензии

3.2.5 Проверка статуса лицензии

Для проверки статуса лицензии приложения «Litoria Crypto Service 2» необходимо использовать функцию «*license*», имеющую следующий синтаксис:

Content-Type: *application/json*

Метод: *GET*

URL: <http://localhost:8080/license>

Результатом успешного выполнения операции является ответ в формате JSON:

```
{
  "message": "",
  "policy": "",
  "properties": "",
  "status": <Статус активации>
}
```

<code>"message"</code>	Сообщение о состоянии лицензии
<code>"policy"</code>	Любые свойства лицензии на сервере лицензирования
<code>"properties":</code>	Свойства файла лицензии
<code>"status"</code>	Статус активации ПО «Litoria Crypto Service 2». Принимает значения: <i>Success</i> = 1 – успешно, <i>LicenseKeyError</i> = 2 – ошибка ключа лицензии, <i>ReactivationAvailable</i> = 3 – необходима реактивация, <i>LicenseReactivationError</i> = 4 – отсутствие лицензии для реактивации, <i>EmailError</i> = 5 – отсутствие электронного адреса для регистрации, <i>RegistrationError</i> = 6 – ошибка регистрации, <i>RegistrationOffline</i> = 7 – регистрация офлайн, <i>ActivationRegistrationError</i> = 8 – невозможность выполнения активации до регистрации ключа,

<i>ActivationError</i> = 9 – ошибка активации, <i>LicenseError</i> = 10 – ошибка лицензии
--

3.2.6 Web-активация

Активация приложения «Litoria Crypto Service 2» возможна также с использованием web-активации. Описание приведено в разделах 2.5 и 3.7 документа «ПО «Litoria Crypto Service 2». Руководство по инсталляции»).

4 Описание функций REST интерфейса ПО «Litoria Crypto Service 2»

Приложение позволяет выполнить операции создания ЭП и шифрования, извлечения и проверки ЭП, просмотра хранилища сертификатов; создания запроса на выпуск сертификата; установки сертификата из устройства в хранилище; просмотра установленных криптопровайдеров и их параметров; пролонгации подписи; использования службы ДТС для проверки ЭП.

4.1 Общие настройки для работы ПО «Litoria Crypto Service 2»

Для корректной работы ПО «Litoria Crypto Service 2» необходимо установить настройки, описанные ниже, при помощи функции «*settings*».

Параметры, перечисленные в пунктах 4.1.1-4.1.5 могут задаваться отдельно или совместно, в зависимости от поставленных задач.

После установки одной или нескольких из перечисленных настроек, возможно дополнение другими настройками.

Для удаления настроек, необходимо указать при выполнении запроса пустые значения не используемых параметров.

4.1.1 Настройка пути для логирования операций

В конфигурационном файле необходимо задать путь для логирования операций приложения при помощи функции «*settings*», имеющей следующий синтаксис:

Content-Type: *application/json*

Метод: *PUT*

URL: <http://localhost:8080/settings>

JSON:

```
{
  "main" :
  {
    "url" : "",
    "iscreatelog" : ,
    "logpath" : ""
  }
}
```

"main"	Группа настроек для подключения к ПО «Litoria Crypto Service 2»
"url"	Адрес, используемый для подключения к базовым функциям ПО «Litoria Crypto Service 2»
"iscreateolog"	Необходимость логирования операций сервиса (true/false). Используется опционально
"logpath"	Путь к файлу для логирования. Используется опционально

4.1.2 Настройка работы с прокси-сервером

Настройка работы с прокси-сервером осуществляется при помощи функции «*settings*», имеющей следующий синтаксис:

Content-Type: *application/json*

Метод: *PUT*

URL: <http://localhost:8080/settings>

JSON:

```
{
  "proxy":
  {
    "url": "",
    "username": "",
    "password": ""
  }
}
```

"proxy"	Набор параметров для настройки работы с прокси: Содержит параметры: <i>url</i> – адрес прокси-сервера и порт в формате <адрес:порт>; <i>username, password</i> – имя пользователя и пароль аутентификации на прокси-сервере
---------	--

4.1.3 Настройка подключения к базе данных

Перед настройкой подключения к базе данных, необходимо убедиться в наличии установленной БД (СУБД PostgreSQL).

Для создания таблицы, используемой в процессе работы с БД, выполнить sql скрипт, соответствующий установленной БД, из директории установки продукта («C:\Program

Files (x86)\GIS\Litoria Crypto Service 2\scripts») воспользовавшись приложением DBeaver(<https://dbeaver.io/>) или SQL Shell.

Настройка подключения к базе данных осуществляется при помощи функции «*settings*», имеющей следующий синтаксис:

Content-Type: *application/json*

Метод: *PUT*

URL: <http://localhost:8080/settings>

JSON:

```
{
  "db":
  {
    "dbname": "",
    "host": "",
    "password": "",
    "port": ,
    "username": ""
  }
}
```

"db"	Набор настроек для подключения к БД. Содержит параметры: <i>dbname</i> – имя БД, <i>host</i> – хост для подключения, <i>password</i> – пароль для подключения к БД, <i>port</i> – используемый для подключения БД порт, <i>username</i> – имя пользователя для подключения к БД
------	---

4.1.4 Настройка выбора криптопровайдера для ОС Linux

Выбор криптопровайдера для ОС Linux осуществляется в блоке общих настроек при помощи функции «*settings*», имеющей следующий синтаксис:

Content-Type: *application/json*

Метод: *PUT*

URL: <http://localhost:8080/settings>

JSON:

```
{
```

```
"linux_cryptoapi":  
  {  
    "apiCore": ""  
  }  
}
```

"linux_cryptoapi"	Блок настроек криптопровайдера для ОС Linux. Содержит параметр: <i>apiCore</i> – используемый криптопровайдер, может принимать значения “cryptopro” или “vipnet”
-------------------	---

Если при старте приложения в поле *apiCore* криптопровайдер не задан – запускается метод библиотеки для поиска криптопровайдеров и устанавливается в приоритете, начиная с *cryptopro*. Библиотека инициализируется найденным провайдером.

Если в поле корректное имя провайдера, но его нет в системе, проинициализировать криптопровайдер не удастся. Необходимо задать настройки нового криптопровайдера и выполнить перезапуск сервиса.

4.1.5 Настройка квалифицированного режима

Включение/выключение проверки сертификата на квалифицированность осуществляется в блоке общих настроек при помощи функции «*settings*», имеющей следующий синтаксис:

Content-Type: *application/json*

Метод: *PUT*

URL: <http://localhost:8080/settings>

JSON:

```
{  
  "qualify":  
  {  
    "usequalifiedmode": bool,  
    "url": "",  
    "tsltrustedcert": ""  
  }  
}
```

"qualify"	Блок настроек квалифицированного режима. Содержит параметры: <i>usequalifiedmode</i> – использование квалифицированного режима. При установке значения <i>true</i> – будет осуществляться проверка подписи и сертификата на квалифицированность, <i>false</i> – проверка на
-----------	---

квалифицированность не будет осуществляться;
url – адрес для скачивания списка аккредитованных удостоверяющих центров;
tsltrustedcert – издатель списка аккредитованных УЦ в BASE64 кодировке

4.1.5.1 Скачивание/обновление списка аккредитованных УЦ

Перед первой проверкой ЭП на квалифицированность необходимо скачать список аккредитованных УЦ.

При включенном квалифицированном режиме проверки электронной подписи скачивание/обновление списка аккредитованных УЦ осуществляется при помощи функции «*tslupdate*», имеющей следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/tslupdate>

JSON:

```
{
  "downloadnewtsl": bool,
  "tslfile" : ""
}
```

"downloadnewtsl"	При отправке значения <i>true</i> – осуществляется скачивание нового списка аккредитованных УЦ из сети Интернет по адресу, указанному в поле <i>url</i> настройки <i>qualify</i> .
"tslfile"	Загрузка списка аккредитованных УЦ из файла

В результате выполнения операции скачивания/обновления списка аккредитованных УЦ может возвращаться ошибочный ответ в формате JSON:

```
{
  "status" : "";
  "tsltrustedcert" : ""
}
```

"status"	Статус ответа. Принимаемые значения: 0 undefined Не определено 1list_corrupted Некорректный формат списка 2 list_modified Список изменен 3 cert_error Действительность сертификата ключа подписи не
----------	---

	<p>подтверждена</p> <p>4 good_status Действителен</p> <p>5 third_party_cert Подписано сторонним сертификатом (не имеющим отношение к Минкомсвязи)</p>
" <code>tsltrustedcert</code> "	Сертификат, которым подписан tsl список, при статусе 5

Скачивание/обновление списка аккредитованных УЦ можно выполнить либо из сети Интернет по указанному адресу (параметр "`downloadnewtsl`"), либо из файла (параметр "`tslfile`"). В функции «`tslupdate`» задается только один из указанных параметров.

Перед загрузкой файла списка аккредитованных УЦ необходимо предварительно установить в хранилище локального компьютера «Доверенные корневые центры сертификации» сертификат "Russian Trusted Root CA" ssl-сертификата сайта, с которого скачивается tsl-список (рисунок 3).

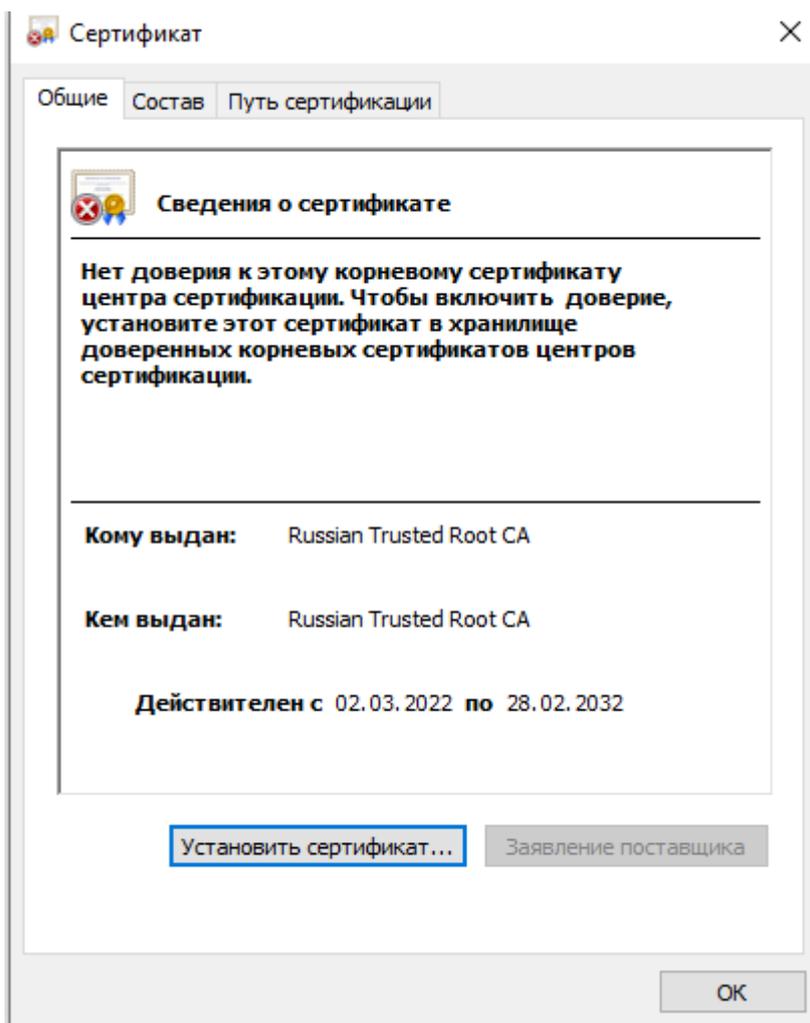


Рисунок 3 – Установка корневого сертификата

4.1.6 Получение заданных настроек

Получение информации об установленных настройках возможно при помощи функции «*settings*», имеющей следующий синтаксис:

Content-Type: *application/json*

Метод: *GET*

URL: <http://localhost:8080/settings>

В случае успешного выполнения запроса для ПО «Litoria Crypto Service 2» под управлением ОС семейства Linux возвращается ответ в формате JSON с параметрами настроек:

```
{
  "linux_cryptoapi":
  {
    "apiCore": ""
  },
  "proxy":
  {
    "password": "",
    "url": "",
    "username": ""
  },
  "activation":
  {
    "license": "",
    "server": "",
    "sshkey": ""
  },
  "db":
  {
    "dbname": "",
    "host": "",
    "password": "",
    "port": <номер порта>,
    "username": ""
  },
  "main":
  {
    "logpath": "",
    "iscreatelog": ,
    "pid": "",
    "url": "",
    "securitytokenenable": bool,
    "securitytoken": ""
  }
}
```

```
},  
"qualify":  
{  
  "url": "",  
  "usequalifiedmode": bool,  
  "tsltrustedcert": ""  
},  
"syslog":  
{  
  "hostname": ""  
}  
}
```

В случае успешного выполнения запроса для ПО «Litoria Crypto Service 2» под управлением ОС семейства Windows возвращается ответ в формате JSON с параметрами настроек:

```
{  
  "proxy":  
  {  
    "password": "",  
    "url": "",  
    "username": ""  
  },  
  "db":  
  {  
    "dbname": "",  
    "host": "",  
    "password": "",  
    "port": <номер порта>,  
    "username": ""  
  },  
  "main":  
  {  
    "logpath": "",  
    "iscreatelog": ,  
    "pid": "",  
    "url": "",  
    "securitytokenenable": bool,  
    "securitytoken": ""  
  },  
  "qualify":  
  {  
    "url": "",  
    "usequalifiedmode": bool,  
    "tsltrustedcert": ""  
  },  
  "syslog":
```

```
{
  "hostname": ""
}
```

"activation"	Блок параметров для настроек, относящихся к активации системы
"license"	Лицензия в формате BASE64 для активации системы
"server"	Адрес url сервера активации
"sshkey"	Ssh-ключ в формате BASE64 (для возможности обновления)
"db"	Блок параметров для настроек базы данных
"dbname"	Имя базы данных
"host"	Адрес для подключения к БД
"password"	Пароль для подключения к БД
"port"	Порт для подключения к БД
"username"	Имя пользователя для подключения к БД
"main"	Блок параметров для общих настроек
"logpath"	Путь к файлу для логирования. Используется опционально
"iscreatelog"	Необходимость логирования операций сервиса (true/false). Используется опционально
"pid"	Номер запущенного процесса криптосервиса в ОС Linux
"url"	Адрес для работы ПО «Litoria Crypto Service 2»
"securitytokenenable"	Флаг включения аутентификации по токену
"securitytoken"	Токен аутентификации
"qualify"	Блок параметров настроек квалифицированного режима
"url"	Адрес скачивания списка аккредитованных УЦ
"usequalifiedmode"	Флаг включения квалифицированного режима проверки ЭП
"tsltrustedcert"	Издатель списка аккредитованных УЦ
"syslog"	Блок параметров настроек для подключения к syslog-серверу
"hostname"	Адрес для подключения к syslog-серверу
"linux_cryptoapi"	Блок настроек криптопровайдера для ОС Linux. Содержит параметр: <i>apiCore</i> – используемый криптопровайдер, может принимать значения "cryptopro" или

	"vipnet"
"proxy"	Набор параметров для настройки работы с прокси: Содержит параметры: <i>url</i> – адрес прокси-сервера и порт в формате <адрес:порт>; <i>username</i> , <i>password</i> – имя пользователя и пароль аутентификации на прокси-сервере

Пример ответа в ОС Linux:

```
{
  "linux_cryptoapi": {
    "apiCore": "cryptopro"
  },
  "proxy": {
    "password": "*****",
    "url": "http://10.116.111.105:8080",
    "username": "pruser"
  },
  "archive": {
    "period": "* * * * *",
    "stampoid": "1.2.643.7.1.1.2.2",
    "tspaddress": "http://testca.gaz-is.ru/tsp/tsp2012.srf",
    "limitdays": 300
  },
  "main": {
    "securitytoken": "*****",
    "pid": 43405,
    "url": "http://192.168.72.130:8001",
    "securitytokenenable": false
  },
  "syslog": {
    "hostname": "http://localhost:514"
  },
  "activation": {
    "license": "-----BEGIN CERTIFICATE-----
\r\nMIIGmjCCBgOgAwIBAgIPVdviBQSTPVVLxAAD7KU6MA0GCSqGSIb3DQEBBQUAMIHCMR0wGwYJKo
ZI\.....\wQCMAAwDQYJKoZIhvcNAQEFBQADgYEAS\IJqqFSkNsT5AcyHl57yvruledTJ1G
S1UvLth\k\r\nKirQepqABzv7ik00RY8yTFq5cPQmCtBKMrNQUhMC8tzXYPFohkAGp+GFwipfkCO\
/clVw+zKDYNtY\r\njIIdHm2vvhFZBxVzHzQ13P0C0TcE6tKHF+wS4o3nGGqjnSmS2SovTXI=\r\n-
-----END CERTIFICATE-----\r\n",
    "server": "http://10.80.2.11",
    "sshkey": "-----BEGIN RSA PRIVATE KEY-----
\r\nMIIEogIBAAKCAQEA7Wh8vFYdIR3MWHfL7wvbZGIbveiGYO8WwyAlKDmNJ8DPp8Q+\n75di7rWiY5
gIHsolSVEZ9ptUOpBE6WHeM4JW7Ieh6wi0ZSMsiTaEr\k2gEyuZiHw\nFutFwEozW\.....\nboN
GZMgmdMPjSjF2IPaFiLZ3wMXd\haxw\kp194zrBY2EbPQ5+gZXIfyWuIcCbIl\nLWYpHxdy9Sisu
V6A9hVYFzRodFq5dG249aGPOd+FZTCr4gdFQtc=\n-----END RSA PRIVATE KEY-----\n"
  },
}
```

```
"qualify": {
  "tsltrustedcert":
"MIIGgTCCBi6gAwIBAgIPVdlG1V3zkeXTq3tSiGanMAoGCCqFAwcbAQMCMIIIBPTEdMBsGCSqGSIb3D
QEJARYOcmVzcEBnYXotaXMucnUxCzAJBgNVBAYTALJVMR4wHAYDVQQHDBXQoS3Qn9C10YLQtdGA0LH
Rg9GA0LMxJzAlBgNVBAoMHtCT0JDQl9CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoTELMAkGA1UECwwCS
VQxZTBjBgNVBAMMXNC\.....\i0LXRgdGC0L7QstGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0Y
\.....+eaJd\IW21bbFc0dFecaTaCbY0HyejJIfkKjD6V0wsIUZzFvPrhm9Y+UugJmvj8Y5Wt0agaF
PN4Vdg==",
  "usequalifiedmode": true
},
"db": {
  "password": "*****",
  "dbname": "postgres",
  "port": 5432,
  "host": "localhost",
  "username": "postgres"
}
}
```

Пример ответа в ОС Windows:

```
{
  "proxy": {
    "password": "*****",
    "url": "http://10.116.111.105:8080",
    "username": "pruser"
  },
  "archive": {
    "period": "* * * * *",
    "stampoid": "1.2.643.7.1.1.2.2",
    "tspaddress": "http://testca.gaz-is.ru/tsp/tsp2012.srf",
    "limitdays": 300
  },
  "main": {
    "logpath": "K:\\log.txt",
    "securitytoken": "*****",
    "iscreateelog": true,
    "url": "http://127.0.0.1:8002",
    "securitytokenenable": true
  },
  "syslog": {
    "hostname": "http://localhost:514"
  },
  "qualify": {
    "tsltrustedcert":
"MIIGgTCCBi6gAwIBAgIPVdlG1V3zkeXTq3tSiGanMAoGCCqFAwcbAQMCMIIIBPTEdMBsGCSqGSIb3D
QEJARYOcmVzcEpNCtNC+0YHRgtC+0LLQtdGA0Y\RjtGJ0LjQuSDRhtC10L3RgtGAINCT0JDQl9CY0
J3QpNCe0KDQnNCh0JXQoNCS0JjQoSAYMDEyMQ8wDQYDVQQFEWYxOTgwOTYxQTA\.....\0L3
RiNGC0LDQtNGC0YHQtCw0Y8sINC0LjEwLCDQu9C40YLQtdGA0LAg0JCCD3vnAAvMTgMJPxU3AAAA6
```

```
DAMBgNVHRMBAf8EAjAAMAoGCCqFAwCBAQMCA0EAI9f4QyHrAeaJd\ /IW21bbFc0dFecaTaCbY0Hyej  
JIIfkKjD6V0wsIUZzFvPrhm9Y+UugJmvj8Y5Wt0agaFPN4Vdg==" ,  
  "usequalifiedmode": true  
},  
"db": {  
  "password": "*****",  
  "dbname": "postgres",  
  "port": 5432,  
  "host": "localhost",  
  "username": "postgres"  
}  
}
```

4.2 Настройка автоматической пролонгации подписи

4.2.1 Регистрация и запуск службы

Перед использованием службы автоматической пролонгации подписи необходимо настроить автоматический запуск от имени учетной записи пользователя с правами администратора.

В оснастке «**Services**» найдите службу «**AuthomaticSignatureProlongation**» и по щелчку правой кнопки мыши на ней перейдите в свойства созданной службы (рисунок 4).

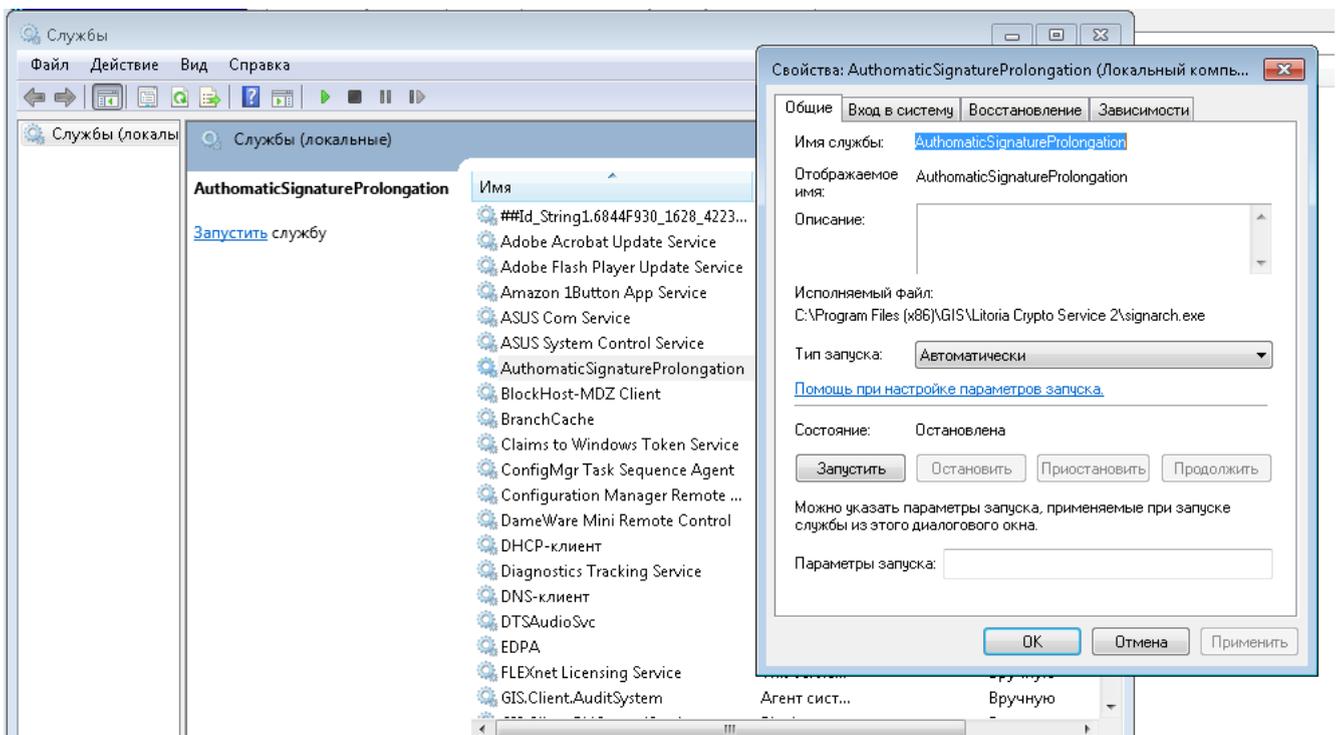


Рисунок 4 – Регистрация и запуск службы

Выберите **Тип запуска** → **Автоматически**. Во вкладке **Вход в систему** установите

переключатель в позицию «С учетной записью:» и по кнопке «Обзор» добавьте учетную запись пользователя рабочей станции, на которой осуществляется запуск службы.

После сохранения настроек, выполните запуск созданной службы по кнопке «Запустить службу».

4.2.2 Настройка службы

Для настройки автоматической пролонгации подписи используется функция «*settings*», имеющей следующий синтаксис:

Content-Type: *application/json*

Метод: *PUT*

URL: <http://localhost:8080/settings>

JSON:

```
{
  "archive":
  {
    "limitdays": ,
    "period": "",
    "tspaddress": "",
    "stampoid": ""
  },
  "syslog" :
  {
    "hostname": ""
  }
}
```

"archive"	Параметры для настройки работы службы signarch.exe , отвечающей за периодическую проверку истекающих подписей и их продление. Используемые параметры: <i>limitdays</i> – количество дней до истечения срока действия подписи; <i>period</i> – периодичность запуска службы signarch.exe (задается в формате, подходящем для работы <i>cron</i> (https://ru.wikipedia.org/wiki/Cron). Например, запуск каждую минуту задаётся выражением * * * * *); <i>tspaddress</i> – адрес tsp-службы, используемой для продления срока действия подписи;
------------------	--

Перед началом использования приложения «Litoria Crypto Service 2» необходимо убедиться в наличии корневого сертификата для цепочки сертификатов службы штампов времени в хранилище сертификатов

	«Доверенные корневые центры сертификации». <i>stampoid</i> – алгоритм хеширования, с которым работает tsp-сервер
<i>"syslog"</i>	Параметры для работы с syslog-сервером. <i>hostname</i> – url-адрес, где будет происходить регистрация сообщений от службы signarch.exe по протоколу <i>syslog</i> . Используется опционально

Служба **signarch.exe** предназначена для отслеживания хранящихся в базе данных подписей с истекающим сроком действия, и автоматического продления таких подписей с использованием заданного tsp-сервера.

Для настройки периодичности отслеживания таких подписей, используется планировщик «*Cron*», который запускается службой **signarch.exe** в указанный момент времени (параметр «*period*»).

Планировщик отслеживает в базе данных подписи с истекающим сроком действия. Истекающей считается подпись, окончание срока действия которой меньше количества дней, указанных в параметре «*limitdays*».

В случае обнаружения такой подписи происходит автоматическая пролонгация подписи с использованием заданного tsp-сервера (параметр «*tspaddress*») и добавление в базу данных обновленной информации о продленной подписи.

При работе со службой **signarch.exe** возможно отслеживание событий на *syslog*-сервере. Для этого необходимо указать его адрес в параметре «*hostname*».

Если за время работы службы подписи не успеют обработаться и обновиться, новая проверка не начнется до окончания оставшихся задач.

Если в базе нет подписей для обновлений, служба **signarch.exe** отправит на *syslog*-сервер сообщение об отсутствии подписей, нуждающихся в продлении.

4.3 Обращение к хранилищу сертификатов

4.3.1 Получение всех сертификатов из хранилища

Для получения всех сертификатов из заданного хранилища используется функция «*certs*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *GET*

URL: <http://localhost:8080/certs?storename=My>

storename	Имя хранилища сертификатов. По умолчанию в системе используются хранилища: <i>My</i> – хранилище личных сертификатов, <i>Root</i> – хранилище корневых сертификатов удостоверяющих центров (УЦ), <i>CA</i> – хранилище сертификатов промежуточных УЦ и списков отозванных сертификатов, <i>AddressBook</i> – хранилище сертификатов других пользователей
-----------	---

В результате успешного выполнения операции будет получен массив сертификатов, находящихся в указанном хранилище, в формате JSON:

```
{
  "certinfos":
  [
    "<Сертификат_1>",
    "<Сертификат_2>",
    ...,
    "<Сертификат_n"
  ]
}
```

Пример ответа:

```
{
  "certinfos": [
    "MIIGgzCCBe+gAwIBAgIPVdhTPrXVSRrVsJ1tr31WMAoGCCqFAwcBAQMDMIIBRDEdMBsGCSqGSIB3DQEJARYOcmVzceBnYXotaXM\.....\ucnUxCzAJBgNVBAYTAlJVMR4wHAYDVQQHDBXQoS3Qn9C10YLQtdGA0LHRg9GA0LMxJzAlBgNVBAoMHtCT0JDQ19CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoTELMakGA1UECwwCSVQxbDBqBgNVBAMMY9Ci0LXRgdGC0L7QstGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0Y\/+1rSghirZzE6KQeVDeY9BRuosH18N\QkFv9vDlw5ufiQ+gdpXl3FBIhbFE7jmFLVE3hdMYUrC",
    "MIIDwzCCA3CgAwIBAgIPE+cAC8x0Awk\FTcaAADoMAoGCCqFAwcBAQMCMIIIBPTEdMBsGCSqGSIB3DQEJARYOcmVzceBnYXotaXMucn\.....\UxCzAJBgNVBAYTAlJVMR4wHAYDVQQHDBXQoS3Qn9C10YLQtdGA0LHRg9GA0LMxJzAlBgNVBAoMHtCT0JDQ19CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoTELMakGA1UECwwCSVQxZTBjBgNVBAMMXNCi0LXRgdGC0L7QstGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0Y\ /RjtGJ0LjQuSDRhtC10L3RgtGAINCT0JDQ19CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoSAYMDEyMQ8wDQYDVQQFEwYxOTgwOTYxQTA\ /eJHigSxCgf\ /00YE7A==",
    "MIIFzDCCBLsgAwIBAgIKGsyeSQAAAAAijANBqkqhkiG9w0BAQUFADBWMRMwEQYKCZImiZPyLQGByDRGV2MRMwEQYKCZImiZPyLQBGryDQXBwMRIwEAYKCZImiZPyLQBGryCQkgxFjAUBgNVBAMTduJILVNlcnZpY2U0EwHhcNMjEwNDE\.....\ /2MTAyNjI0WhcNMjIwNDE2MTAyNjI0WjB\ \ /MTauNKSjjggf0j1hq\ /43tJOGRPBzHuHryb57aPZg4xSXQ\ /uK7S5xOd+R96Ua+6v2sgQNx+5kIrUQd+vvQeNjw4agOjKiRm+\ /Dk+41VgiDSnt1D95CNgeoeBuulahIXCwuQv001EFHg4c7I90vKi06mGlYaxAd\ /n6frznc09YVZjTGKtrg0ALRBzahy2m0xPlHTwAE6HIgk2+6H93q00"
  ]
}
```

4.3.2 Установка сертификата в хранилище

Для установки сертификата в хранилище используется функция «*certdata*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *PUT*

URL: <http://localhost:8080/certdata>

JSON:

```
{
  "storename": "",
  "certificate": ""
}
```

"storename"	Имя хранилища, в которое устанавливается сертификат. По умолчанию в системе используются хранилища: <i>My</i> – хранилище личных сертификатов, <i>Root</i> – хранилище корневых сертификатов УЦ, <i>CA</i> – хранилище сертификатов промежуточных УЦ и списков отозванных сертификатов, <i>AddressBook</i> – хранилище сертификатов других пользователей
"certificate"	Сертификат, устанавливаемый в хранилище в формате BASE64

Пример запроса:

```
{
  "storename": "CA",
  "certificate":
  "MIIGLDCCBdmgAwIBAgIPVdf8mI/rjtOwcWUKzjGDMAoGCCqFAwCBAQMCMIIIBPTEdMBsGCSqGSIb3D
  QEJARYOcmVzcEBnYXotaXMucnUxCzAJBgNVBAYTAlJVMR4wHAYDVQQHDBXQoS3Qn9C10YLQtdGA0LH
  Rg9GA0LMxJzAlBgNVBAoMhtCT0JDQ19CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoTELMAkGA1UECwwCS
  VQxZTBjBgNVBAMMXNCi0LXRgdGC0L7Qs\.....\tGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0Y
  /RjtGJ0LjQuSDRhtC10L3RgtGAINCT0JDQ19CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoSAYMDEyMQ8w
  DQYDVQQFEWYxOTgwOTYxQTA/wQCMAAwCgYIKoUDbWEBAwIDQQAayWdsZPMypAYhahZYU0iVMzPqKxS
  Svq7gQ2A3JJ85ag7dVVOJ1j3mZTz2dYCRrAGfDe800ZHVV5s1C7/kCyKU"
}
```

В результате успешного выполнения операции возвращается код состояния запроса «200», в указанное хранилище добавляется новый сертификат.

4.3.3 Удаление сертификата из хранилища

Для удаления сертификата из хранилища используется функция «*certdata*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *DELETE*

URL: <http://localhost:8080/certdata>

JSON:

```
{
  "storename": "",
  "certificate": ""
}
```

"storename"	Имя хранилища, из которого удаляется сертификат. По умолчанию в системе используются хранилища: <i>My</i> – хранилище личных сертификатов, <i>Root</i> – хранилище корневых сертификатов УЦ, <i>CA</i> – хранилище сертификатов промежуточных УЦ и списков отозванных сертификатов, <i>AddressBook</i> – хранилище сертификатов других пользователей
"certificate"	Сертификат, удаляемый из хранилища в формате BASE64

Пример запроса:

```
{
  "storename": "CA",
  "certificate":
  "MIIGLDCCBdmgAwIBAgIPVdf8mI/rjTOWcWUKzjGDMAoGCCqFAwCBAQMCMIIIBPTEdMBsGCSqGSIb3D
  QEJARYOcmVzcEBnYXotaXMucnUxCzAJBgNVBAYTAlJVMR4wHAYDVQQHDBXQoS3Qn9C10YLQtdGA0LH
  Rg9GA0LMxJzAlBgNVBAoMhtCT0JDQ19CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoTELMAkGA1UECwwCS
  VQxZTBjBgNVBAMMXNCi0LXRgdGC0L7\.....\QstGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0Y
  /+cAC8xOAwk/FTcAAADoMAwGA1UdEwEB/wQCMAAwCgYIKoUDBwEBAwIDQQAayWdsZPMypAYhahZYU0
  iVMzPqKxSSvq7gQ2A3JJ85ag7dVVOJl3mZTz2dYCRrAGfDe800ZHvY5s1C7/kCyKU"
}
```

В результате успешного выполнения операции возвращается код состояния запроса «200», указанный сертификат удаляется из хранилища.

4.3.4 Получение информации о сертификате

Для получения информации о сертификате используется функция «*certinfo*», имеющая

следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/certinfo>

JSON:

```
{
    "certificate": "",
    "findkey": bool,
    "pincode": "",
    "buildchain": bool,
    "checkqualify": bool
}
```

"certificate"	Представление сертификата в формате BASE64
"findkey"	Логический флаг, включающий отображение информации о закрытом ключе для сертификата. Принимает значения <i>true/false</i> . Используется опционально
"pincode"	PIN-код к контейнеру с сертификатом. Используется опционально, необходим если флаг «findkey»=true
"buildchain"	Построение и вывод цепочки сертификации. Принимаемое значение: <i>true/false</i> . Используется опционально
"checkqualify"	Проверка квалифицированности сертификата. Принимаемое значение: <i>true/false</i> Предварительно необходимо убедиться, что настроен квалифицированный режим работы (п. 4.1.4) и скачан список аккредитованных УЦ (п. 4.1.4.1). Используется опционально

Пример запроса:

```
{
    "certificate": "MIIGSTCCBfagAwIBAgIQcEL1i7sFtU2EmWAaX176EDAKBggqhQMHAQED  
AjCCAQsxGDAWBgUqhQNkARINMTAzNzcwMDAxMzAyMDEVMBMGBSsqFA2QEEgo3NzAyMjM1MTMzMQswCQ  
YDVQQGEwJSVTEpMCCcGA1UECAwgNzgg0KHQsNC90LrRgi3Qn9C10YLQtdGA0LHRg9GA0LMxJjAkBgNV  
BAcMHdCh0LDQvdC60YIt0J/QtdGC0LXRg\.....\NCx0YPRgNCzMS4wLAYDVQQKDCXQotC10  
YHRgtC+/QtdGC0LXRgNCx0YPRgNCzMS4wLAYDVQQKDCXQotC10YHRgtC+0LLRi9C5INCa0LLQsNC70  
LjRhNC40YbQuNGAMS4wLAYDVQQDDDCXQotC10YHRgtC+0LLRi9C5INCa0LLQsNC70LjRhNC40YbQuNG  
AMRgwFgYDVQQJDA/QodGC0LDRh9C10LogNDeCEAnWWr/ONqhKiZlPwwR2LtUwDAYDVROTAQH/BAIwA  
DAKBggqhQMHAQEDAgNBAHygwOIfWd3W/jvLJaZ1a4WKbCSilKoFNGLTQRNorFJrVuq2fjbmHA2DD1E  
S6VoHy+bGbXL4FdEIFEAPSNM4lfc=",
    "pincode": "12345678",
}
```

```
"findkey": false,  
"buildchain": true,  
"checkqualify": true  
}
```

В результате выполнения данной операции при значениях *false* для параметров «*findkey*» и «*buildchain*» возвращается код состояния запроса «200», ответ в формате JSON следующего вида:

```
{  
  "keyalg": "",  
  "serialnumber": "",  
  "publickey": "",  
  "issuerinfo":  
  {  
    "country": "",  
    "commonname": "",  
    "orgunit": "",  
    "organization": "",  
    "deviceserialnumber": "",  
    "locality": "",  
    "streetaddress": "",  
    "email": ""  
  },  
  "timenotafter": "",  
  "timenotbefore": "",  
  "certsignalg": "",  
  "subjectinfo":  
  {  
    "commonname": ""  
  }  
}
```

"keyalg"	OID алгоритма подписи
"serialnumber"	Серийный номер сертификата
"publickey"	Открытый ключ сертификата
"issuerinfo"	Группа данных, отражающих информацию о сертификате издателя
"country"	Страна
"commonname"	Общее имя сертификата
"orgunit"	Подразделение
"organization"	Организация
"deviceserialnumber"	Серийный номер устройства

"locality"	Город
"streetaddress"	Адрес
"email"	Адрес электронной почты
"timenotafter"	Время окончания действия сертификата
"timenotbefore"	Время начала действия сертификата
"certsignalg"	OID алгоритма хэш-значения подписи
"subjectinfo"	Группа данных, отражающих информацию о сертификате пользователя

В результате выполнения данной операции при значениях *true* для параметров «*findkey*» и «*buildchain*» возвращается код состояния запроса «200», ответ в формате JSON следующего вида:

```
{
  "keyalg": "",
  "certsignparam": {
    "provsignparam": {
      "containername": "",
      "signalg": "",
      "corename": "",
      "hashalg": ""
    }
  },
  "serialnumber": "",
  "publickey": "",
  "issuerinfo": {
    "country": "",
    "commonname": "",
    "orgunit": "",
    "organization": "",
    "deviceserialnumber": "",
    "locality": "",
    "streetaddress": "",
    "email": ""
  },
  "timenotafter": "",
  "timenotbefore": "",
  "certificatechain": {
    "ischaincorrect": bool,
    "chainelements": [
      {
        "revokeinfo": {
          "isselfsignerroot": bool,
          "revstatus": "unrevoked",

```

```

        "ocspinfo": {
            "ocsptesptime": "",
            "revocationstatus": -1,
            "serialnumber": "",
            "ocspchain": {
                "ischaincorrect": bool,
                "chainelements": [
                    {
                        "revokeinfo": {
                            "isselfsignerroot": bool,
                            "revstatus": "unrevoked",
                            "crlinfos": {
                                "revocationstatus": "unrevoked",
                                "crlinfo": [
                                    {
                                        "expirydate": "",
                                        "issuerchain": {
                                            "ischaincorrect":
true,
                                                "chainelements": [
                                                    {
"revokeinfo": {
"isselfsignerroot": bool,
"revstatus": ""
"certificate": ""
                                                    },
                                                ]
                                            },
                                        ],
                                        "crlrawdata": "",
                                        "effectivedate": ""
                                    }
                                ]
                            },
                            "certificate": ""
                        },
                        {
                            "revokeinfo": {
                                "isselfsignerroot": bool,
                                "revstatus": "undertermined"
                            },
                            "certificate": ""
                        }
                    ]
                }
            },
            "certificate": ""
        },
        {
            "revokeinfo": {
                "isselfsignerroot": bool,
                "revstatus": "undertermined"
            },
            "certificate": ""
        }
    ],
    }
},

```

```

        "certificate": ""
    },
    {
        "revokeinfo": {
            "isselfsignerroot": bool,
            "revstatus": "undertermined"
        },
        "certificate": ""
    }
]
},
    "qualifyinfo": {
        "caqualifiedstatus": "",
        "certfielderror": "",
        "accreditstatus": "",
        "qualifiedstatus": bool
        "nonqualstrings": []
    },
    "certsignalg": "",
    "subjectinfo": {
        "commonname": "<Имя_сертификата>"
    }
}

```

"keyalg"	OID алгоритма подписи
"certsignparam"	Параметры ключа подписи проверяемого сертификата
"containername"	Имя контейнера
"provsignparam"	Информация о криптопровайдере ключа подписи проверяемого сертификата
"signalg"	Алгоритм подписи
"corename"	Наименование криптопровайдера
"hashalg"	OID хэш-алгоритма подписи
"serialnumber"	Серийный номер сертификата
"publickey"	Открытый ключ сертификата
"issuerinfo"	Группа данных, отражающих информацию о сертификате издателя
"country"	Страна
"commonname"	Общее имя сертификата
"orgunit"	Подразделение

"organization"	Организация
"deviceserialnumber"	Серийный номер устройства
"locality"	Город
"streetaddress"	Адрес
"email"	Адрес электронной почты
"timenotafter"	Время окончания действия сертификата
"timenotbefore"	Время начала действия сертификата
"certificatechain"	Группа параметров, содержащих информацию о цепочке сертификатов
"ischaincorrect"	Параметр, показывающий корректность построения цепочки сертификатов. Принимает значения <i>true</i> – в случае корректности цепочки, <i>false</i> – если цепочка нарушена
"revokeinfo"	Информация об отзыве сертификатов
"isselfsignerroot"	Параметр, показывающий, является ли сертификат самовыпущенным
"revstatus"	Статус отзыва сертификата
"ocspinfo"	Информация о проверке отзыва по OCSP
"ocspresptime"	Время создания списка отзыва
"revocationstatus"	Состояние списка отзыва
"crlinfos"	Информация о проверке отзыва по CRL
"expirydate"	Дата истечения срока действия списка отзыва
"issuerchain"	Информация о цепочке сертификата-издателя
"crlrawdata"	Список отзыва в формате BASE64
"effectivedate"	Дата формирования списка отзыва
"ocspchain"	Цепочка сертификатов для списка отзывов
"chainelements"	Описание всех сертификатов в цепочке
"certsignalg"	OID алгоритм подписи
"subjectinfo"	Группа данных, отражающих информацию о сертификате пользователя
"qualifyinfo"	Набор параметров проверки подписи на квалифицированность: <i>saqalifiedstatus</i> – статус квалифицированности УЦ

	<p>(qualified/unknown/nonqualified);</p> <p><i>certfielderror</i> – код ошибки квалифицированности, если нет, то равен 0;</p> <p><i>accreditstatus</i> – статус аккредитации УЦ;</p> <p><i>qualifiedstatus</i> – статус квалифицированности подписи (true/false);</p> <p><i>nonqualstrings</i> – описание ошибки квалифицированности</p>
--	--

Пример ответа:

```
{
  "keyalg": "1.2.643.7.1.1.1.1",
  "serialnumber": "7042F58BBB05B54D8499601A5F5EFA10",
  "publickey":
"4CF781D6F791F1393014271DC5A4EBB920F2BE0729B6EF89EEC4478C8336FD47913802D0D88C7
E453484729E6C55A03FC5774B490BD09CDA631442A5FADA69E5",
  "issuerinfo": {
    "ogrn": "1037700013020",
    "country": "RU",
    "commonname": "Тестовый Квалифицицир",
    "organization": "Тестовый Квалифицицир",
    "locality": "Санкт-Петербург",
    "streetaddress": "Стачек 47",
    "state": "78 Санкт-Петербург"
  },
  "timenotafter": "2032-03-25T12:13:55Z",
  "timenotbefore": "2022-03-28T12:13:55Z",
  "certificatechain": {
    "ischaincorrect": true,
    "chainelements": [
      {
        "certificate":
"MIIGSTCCBfagAwIBAgIQcEL1i7sFtU2EmWAAx176EDAKBggqhqMHAQEDAJCCAQsxDAAWBgUqhQNkA
RINMTAzNzcwMDAxMzAyMDEVMBMGBSsqFA2QEEgo3NzAyMjM1MTMzMzQswCQYDVQQGEwJSVTEpMCcGA1U
ECAwGNgzgg0KHQsNC90LrRgi3Qn9C1.....0YLQtdGA0LHRg9GA0LMxJjAkBgNVBACMHdCh0LD
QvdC60YIt0J\+/0LLRi9C5INCa0LLQsNC70LjRhNC40YbQuNGAMRgwFgYDVQQJDA\ /QodGC0LDRh9C
10LogNDeCEAnWWr\ /ONqhKiZlPwwR2LtUwDAYDVR0TAQH\ /BAIwADAKBggqhqMHAQEDAgNBAHygwOI
fWd3W\ /jvLJaZla4WKbCSilKofNglTQrNorFJrVuq2fjbmHA2DD1ES6VoHy+bGbXL4FdEIFEAPSNM4
lfc=",
        "revokeinfo": {
          "isselfsignerroot": false,
          "revstatus": "unrevoked",
          "crlinfos": {
            "revocationstatus": "unrevoked",
            "crlinfo": [
              {
                "expirydate": "2024-02-22T09:33:48Z",

```



```
    ]
  },
  "qualifyinfo": {
    "caqualifiedstatus": "qualified",
    "certfielderror": 0,
    "accreditstatus": "valid",
    "qualifiedstatus": true
  },
  "certsignalg": "1.2.643.7.1.1.3.2",
  "subjectinfo": {
    "country": "RU",
    "commonname": "Физлицо Нового Квалиф",
    "givenname": "Нового Квалиф",
    "surname": "Физлицо",
    "inn": "110805887539",
    "snils": "01228113390"
  }
}
```

4.4 Создание запроса на сертификат или самоподписанного сертификата

Для создания запроса на сертификат или самоподписанного сертификата используется функция «*certrequest*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/certrequest>

JSON:

```
{
  "csp":
  {
    "corename": "",
    "alname": "",
    "containername": "",
    "pincode": "",
    "exportkey": bool
    "keysize":
  },
  "requestparams":
  {
    "oldcertificate": "",
    "requestfields":
    {
```

```

        "OID": "",
        "OID": "",
        "OID": "",
        "OID": ""
    },
    "keyusage": [],
    "keyusageiscritical": bool,
    "extkeyusage": [],
    "exkeyusageiscritical": bool
},
"isselfcert": bool
}

```

"csp"	Набор параметров криптопровайдера
"corename"	Наименование криптопровайдера для генерации запроса на сертификат
"algnamе"	Наименование алгоритма для генерации закрытого ключа
"containername"	Генерируемое имя контейнера
"pincode"	PIN-код для доступа к контейнеру
"exportkey"	Создание экспортируемого закрытого ключа. Принимаемое значение: <i>true/false</i> . Используется опционально
"keysize"	Длина закрытого ключа
"oldcertificate"	Старый сертификат для создания запроса на основе сертификата. Записывается в формате BASE64. Используется опционально
"requestparams"	Набор параметров запроса
"requestfields"	Поля сертификата и присваиваемые значения. Содержит параметры: <i>OID</i> – объектный идентификатор поля сертификата (принимает строковое значение).
"keyusage"	Массив параметров использования ключа
"keyusageiscritical"	Возможность выбранных параметров ключа принимать критическое значение.

	Принимаемое значение: <i>true/false</i>
<code>"extkeyusage"</code>	Массив параметров использования ключа
<code>"exkeyusageiscritical"</code>	Возможность выбранных параметров ключа принимать критическое значение. Принимаемое значение: <i>true/false</i>
<code>"isselfcert"</code>	Параметр, показывающий, является ли выпускаемый сертификат самовыпущенным. Принимаемое значение: <i>true/false</i> . Используется опционально

Пример запроса:

```
{
  "csp": {
    "corename": "Microsoft Strong Cryptographic Provider",
    "alname": "RSA Key Exchange",
    "containername": "6913454Test",
    "pincode": "12345678",
    "exportkey": false,
    "keysize": 512
  },
  "requestparams": {
    "requestfields": {
      "1.2.840.113549.1.9.1": "Test_cert6665375",
      "2.5.4.3": "RSA_req",
      "2.5.4.4": "RSA_req",
      "2.5.4.6": "RU"
    },
    "keyusage": ["digitalsign", "encryptdata"],
    "keyusageiscritical": false,
    "extkeyusage": ["1.3.6.1.5.5.7.3.3",
      "1.3.6.1.5.5.7.3.8"],
    "exkeyusageiscritical": false
  },
  "isselfcert": true
}
```

В результате успешного выполнения операции запроса на самоподписанный сертификат возвращается ответ в формате JSON:

```
{
  "selfsignedcert": ""
}
```

<code>"selfsignedcert"</code>	Самоподписанный сертификат в формате BASE64
-------------------------------	---

Пример ответа:

```
{
  "selfsignedcert": "-----BEGIN CERTIFICATE-----
\nMIIB9jCCAaCgAwIBAgIBATANBgkqhkiG9w0BAQUFADBSMR8wHQYJKoZIhvcN\nAQkBFhBUZXN0X2
NlcnQ2NjY1Mzc1MQswCQYDVQGEwJSVTEQMA4GA1UEAwwH\nU1NBX3JlcTEQMA4GA1UEBAwHU1NBX3
JlcTAeFw0yMjA2MDkxMTQ5MTRaFw0y\nMzA2MDkxMTQ5MTRaMFExHzAdBgkqhkiG9w0BCQEWFRlc3
RfY2VydyDY2NjUz\nnNzUxCzAJBgNVBAYTA.....1JVMRAwDgYDVQQDDAdSU0FfcMvMRawDgYD
VQQEDAdS\nU0FfcMvMFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAKRou9YHSbPU+1oAv+Bj\nnylcWl\
/KaYrZ\ /rxD9KJvnu6KUhy8apzuaXZiwUSqa5oo1JcKQzclGxev2WlV\nn7Zoou7kCAwEAAANhMF8w
DgYDVR0PAQH\ /BAQDAgKUMB0GA1UdJQQWMBQGCCsG\nAQUFBwMDBggrBgEFBQcDCDAPBgNVHRMBAf8
EBTADAQH\ /MB0GA1UdDgQWBQBm\nnwo+0Cya6zVYm6eI94pBEYUxIMzANBgkqhkiG9w0BAQUFAANBAA
3pJMu+r4Sr\nnmrcgPsWZQ2RpkWcgEHZ3jj9T0nUGDGWB8PtNBCvnPDOXi93esgaGRR55lbGQ\n\nUQt4
DoG\ /dWRqpaA=\n-----END CERTIFICATE-----\n"
}
```

В результате успешного выполнения операции запроса на сертификат возвращается ответ в формате JSON:

```
{
  "certrequest": " "
}
```

"certrequest"

Запрос на сертификат в формате BASE64

Пример ответа:

```
{
  "certrequest": "-----BEGIN NEW CERTIFICATE REQUEST-----
\nMIIBsjCB9QIBADBMSMR8wHQYJKoZIhvcNAQkBFhBUZXN0X2NlcnQ2NDk1ODI0\nnMQswCQYDVQGEwJSVTEQMA4GA1UEAwwHU1NBX3JlcTEQMA4GA1UEBAwHU1NB\n\nX3JlcTBcMA0GCSqGSIb3DQEBAQUAA0
sAMEgCQQDGL0CJzMsK1RMnz7WT+k21\nnr5FT.....102LsuTyOCD8ZAWkOej17urEtTIxRW2c
H8ecjYx9hwB+zPd069yPsgPn\nnykv1AgMBAAGgPjA8BgorBgEEAYI3AgEOMS4wLDALBgNVHQ8EBAMC
BJAwHQYD\nnVR01BBYwFAYIKwYBBQUHAWMGCCsGAQUFBwMIMA0GCSqGSIb3DQEBBQUAA0EA\n\nXJQ8Ye
1XYFq30N8LB4Yjr2l9uxI+4jVxXWNO+gEO16LqktCcU+pYHJRKbBvO\n\nnPis5boX6elnOQ0n2RSIwJ0
4iWQ==\n-----END NEW CERTIFICATE REQUEST-----\n"
}
```

4.5 Получение списка криптопровайдеров

Для получения списка криптопровайдеров используется функция «*coreinfo*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *GET*

URL: <http://localhost:8080/coreinfo>

В результате успешного выполнения операции возвращается код состояния запроса «200», ответ в формате JSON следующего вида:

```
{
  "coreinfos":
  [
    {
      "algsinfo":
      [
        {
          "alname": "",
          "keysize": <Длина ключа>
        }
      ],
      "corename": ""
    },
    {
      "algsinfo":
      [
        {
          "alname": "",
          "keysize": <Длина ключа>
        }
      ],
      "corename": ""
    }
  ]
}
```

"coreinfos"	Массив криптопровайдеров
"algsinfo"	Описание алгоритмов генерации ключевой пары
"alname"	Имя алгоритма
"keysize"	Длина ключа
"corename"	Имя криптопровайдера

Пример ответа:

```
{
  "coreinfos": [
    {
      "algsinfo": [
        {
          "alname": "RSA Key Exchange",
          "keysize": 1024
        }
      ],
    }
  ],
}
```

```
    "corename": "Aktiv ruToken CSP v1.0"
  },
  {
    "algsinfo": [
      {
        "alname": "GOST R 34.10-2001 256 DH",
        "keysize": 512
      }
    ],
    "corename": "Crypto-Pro GOST R 34.10-2001 Cryptographic Service
Provider"
  },
  {
    "algsinfo": [
      {
        "alname": "GOST R 34.10-2012 256 DH",
        "keysize": 512
      }
    ],
    "corename": "Crypto-Pro GOST R 34.10-2012 Cryptographic Service
Provider"
  },
  {
    "algsinfo": [
      {
        "alname": "GOST R 34.10-2012 512 DH",
        "keysize": 1024
      }
    ],
    "corename": "Crypto-Pro GOST R 34.10-2012 Strong Cryptographic
Service Provider"
  },
  {
    "algsinfo": [
      {
        "alname": "RSA Key Exchange",
        "keysize": 1024
      }
    ],
    "corename": "eToken Base Cryptographic Provider"
  },
  {
    "algsinfo": [
      {
        "alname": "RSA Key Exchange",
        "keysize": 1024
      }
    ],
  },
```

```
        "corename": "Microsoft Strong Cryptographic Provider"  
    }  
]  
}
```

4.6 Подпись и шифрование

Для подписи и шифрования файла используется функция «*cryptcreator*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/cryptcreator>

JSON:

```
{  
  "source":  
  {  
    "rawdata":  
    {  
      "data": ""  
    },  
    "file":  
    {  
      "input": "",  
      "originfile": "",  
      "output": ""  
    }  
  },  
  "sign":  
  {  
    "certificateinfo":  
    {  
      "certificate": "",  
      "pincode": ""  
    },  
    "parameters":  
    {  
      "comment": "",  
      "detached": bool,  
      "includetimestamp":  
      {  
        "advanced": bool,  
        "tspaddress": ""  
      }  
    }  
  },  
}
```

```
"forcecms": bool
},
"encrypt":
{
  "certificates": ["" ]
}
}
```

"source"	Набор параметров для ввода исходных данных подписи. Содержит параметры <i>rawdata</i> или <i>file</i> .
"rawdata"	Набор параметров для работы с данными. Содержит параметры: <i>data</i> – обрабатываемая информация, закодированная в BASE64. <i>detacheddata</i> – исходные данные для подписи в формате BASE64 (используется при работе с отделенной подписью).
"file"	Набор параметров для работы с файлами. Содержит параметры: <i>input</i> – путь к входному файлу (подпись), <i>originfile</i> – путь к файлу с исходными данными в случае добавления/обновления отделенной подписи, <i>output</i> – путь к выходному файлу. Путь задается в формате C:/Folder/File
"sign"	Набор параметров для создания подписи
"certificateinfo"	Информация о сертификате. Содержит параметры: <i>certificate</i> – сертификат подписанта в формате BASE64, <i>pincode</i> – PIN-код для доступа к контейнеру
"parameters"	Дополнительные (опциональные) параметры для создания подписи. Содержит параметры: <i>comment</i> – комментарий к подписи, <i>detached</i> – отделенная ли подпись (true/false), <i>includetimestamp</i> – параметры для создания усовершенствованной ЭП, <i>advanced</i> – включение/отключение создания усовершенствованной подписи (true/false), <i>tspaddress</i> – адрес службы штампов времени. Используется опционально

<p><i>"forcecms"</i></p>	<p>Формат создания подписи. Значение <i>true</i> – создание подписи в формате CMS, <i>false</i> – выполняется автоопределение формата - CMS или PAdES Для создания подписи в формате XAdES вместо этого флага нужно указать следующий флаг: "forcexades" = true. Используется опционально</p>
<p><i>"encrypt"</i></p>	<p>Набор параметров для шифрования данных (может использоваться отдельно или одновременно с функцией создания подписи). Содержит параметры: <i>certificates</i> – массив сертификатов получателей в формате BASE64.</p>

Пример запроса для подписи данных:

```
{
"source":
{
"rawdata":
{
"data": "eyJAKI1NvdXJjZSI6IHsgCiAgIkZpbGUiOiB7IAoiSW5wdXQiOiJD0lxVc2Vyc1xkdWtyZW
V2YS1rXERlc2t0b3Bc0L/RgNC+0LLQtDGA0LrQsDEudHh0IiwgCiJEZXRhY2hlZElucHV0IjoiPNCf
0YPRgtGMINC00L4g0YTQsNC50LvQsCDRgSDQuNGB0YXQvtC00L3Ri9C80Lgg0LTQsNC90L3Ri9C80L
gg0LIg0YHQ09GD0YfQsNC1INC00L7QsdCw0LLQu9C10L3QuNGPL9C+0LHQvdC+0LLQu9C10L3QuNGP
INC+0YLQtNC10LvQtD90L3QvtC5INC...../0L7QtNC/0LjRgdC4PiIsIAoiT3V0cHV0Ijoi
QzpcVXNlcnNcZHVrcmVldmEta1xEZXRrdG9wXNC/0YDQvtCy0LXRgNC60LAXLnR4dC5wN3MiIAp9IA
p9LCAKI1NpZ24iOiB7IAoiQ2VydGlmawNhdGVJbmZvIjogeyAKIkNlcnRpZmljYXRlIjoiTUlJSVl6
Q0NCOctnQX=="
}
},
"sign":
{
"certificateinfo":
{
"certificate": "MIIGgzCCBe+gAwIBAgIPVdhTPrXVSRrVsJ1tr31WMAoGCCqFAwCBAQMDMIIBRDE
dMBsGCSqGSIB3DQEJARYOcmVzcEbnYXotaXMucnUxXzAjbG9NVBAYTAlJVMR4wHAYDVQQHDBXQoS3Qn
9C10YLQtDGA0LHRg9GA0LMxJzAlBgNVBAo.....MHtCT0JDQl9CY0J3QpNCe0KDQnNCh0JXQo
NCS0JjQoTELMaKGA1UECwwCSVQxbDBqBgNVBAMMY9Ci0LXRgdGC0L7QstGL0Lkg0YPQtNC+0YHRgtC
+",
"pincode": "12345678"
}
},
"parameters":
```

```

    {
      "comment": "check sing",
      "detached": false
    }
  }
}

```

Пример запроса для шифрования данных:

```

{
"source": {
  "rawdata":
    {
      "data": "aGVsbG8="
    }
},
"encrypt":
  {
    "certificates":
["MIIGgzCCBe+gAwIBAgIPVdhTPrXVSRrVsJ1tr31WMAoGCCqFAwcBAQMDMIIBRDEdMBsGCSqGSIB3
DQEJARYOcmVzcEBnYXotaXMucnUxCzAJBgNVBAYTA1JVMR4wHAYDVQQHDBXQoS3Qn9C10YLQtdGA0L
HRg9GA0LMxJzAlBgNVBAoMHtCT0JDQl9C.....Y0J3QpNce0KDQnNCh0JXQoNCS0JjQoTELMA
kGA1UECwwCSVQxbDBqBgNVBAMMY9Ci0LXRgdGC0L7QstGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0Y/E
7jmFLVE3hdMYUrC"]
  }
}

```

На выходе в результате успешного выполнения операции подписи или шифрования, в случае представления данных в виде *rawdata*, возвращается блок подписанных или зашифрованных данных в формате JSON:

```

{
  "resultdata": ""
}

```

"resultdata"	Созданная подпись / зашифрованное сообщение в формате BASE64
---------------------	--

В случае представления данных в виде *file* результат выполнения операции запишется в файл *output*, код ответа «200».

Пример ответа:

```

{
  "resultdata":
"MIAGCSqGSIB3DQEHAgCAMIACAQExDDAKBggqhQMHAQECAzCABGkqhkiG9w0BBwGggCSABIINGXsgC
iJTb3VyY2UiOiB7IAogICJGaWxlIjogeyAKIklucHV0IjoiQzpcVXNlcnNcZHVrcmVldmEta1xEZXRN
rdG9wXNC\0YDQvtCy0LXRgNC60LAXL.....nR4dCIsIAoiRGV0YWN0ZWRJbnBldCI6IjzQn9

```

```
GD0YLRjCDQtNC+INGE0LDQudC70LAg0YEG0LjRgdGF0L7QtNC90YvQvNC4INC00LDQvdC90YvQvNC4
INCyINGB0LvRg9GH0LDQtSDQtNC+0LHQsNCy0LvQtdC90LjRjy\QvtCx0L3QvtCy0LvQtdC90LjRj
yDQvtGC0LTQtC70LXQvdC90L7QuSDQv9C+/GUfqPE8dPz\+/AAAAAAAAA"
}
```

4.6.1 Подпись отдельных секций XAdES

Для создания подписи в формате XAdES для различных секций xml-документа используется также функция «cryptcreator», при этом операция осуществляется с обязательным наличием флага «*detached*» со значением «*true*».

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/cryptcreator>

JSON:

```
{
  "source":{
    "file":{
      "input":"","
      "output":""
    },
    "rawdata":{
      "data": ""
    }
  },
  "sign":{
    "parameters":{
      "detached": bool,
      "signxadesids": [ "", "", "" ]
    },
    "certificateinfo":{
      "pincode" : "",
      "certificate": ""
    }
  }
}
```

"source"	Набор параметров для ввода исходных данных подписи. Содержит параметры <i>rawdata</i> или <i>file</i> .
"file"	Набор параметров для работы с файлами. Содержит параметры: <i>input</i> – путь к входному файлу (подпись), <i>originfile</i> – путь к файлу с исходными данными в случае добавления/обновления отделенной подписи, <i>output</i> – путь к выходному файлу.

	Путь задается в формате C:/Folder/File
"rawdata"	Набор параметров для работы с данными. Содержит параметры: <i>data</i> – обрабатываемая информация, закодированная в BASE64. <i>detacheddata</i> – исходные данные для подписи в формате BASE64 (используется при работе с отделенной подписью).
"sign"	Набор параметров для создания подписи
"certificateinfo"	Информация о сертификате. Содержит параметры: <i>certificate</i> – сертификат подписанта в формате BASE64, <i>pincode</i> – PIN-код для доступа к контейнеру
"parameters"	Дополнительные (опциональные) параметры для создания подписи. Содержит параметры: <i>comment</i> – комментарий к подписи, <i>detached</i> – отделенная ли подпись (true/false), <i>includetimestamp</i> – параметры для создания усовершенствованной ЭП, <i>advanced</i> – включение/отключение создания усовершенствованной подписи (true/false), <i>tspaddress</i> – адрес службы штампов времени. Используется опционально
"detached"	Отделенная ли подпись - для подписи XAdES по блокам этот флаг должен всегда иметь значение <i>true</i> .
"signxadesids"	Список идентификаторов секций xml-документа, которые необходимо подписать

Пример запроса:

```
{
  "source":{
    "file":{
      "input":"C:/work/projects/test_for_sign.XML",
      "output":"C:/work/projects/sign_test_for_sign.xml"
    }
  },
  "sign":{
    "parameters":{
      "detached": true,

```

```
    "signxadesids": ["car-5c32beba-3401-4827-96fb-564b98a30dd8", "car-
a6e4a37a-7adf-484e-a06e-5d501dd600da", "car-2ec59244-dc8b-495a-8778-
b065bda7ec96", "car-2d993a29-b6de-468b-ba76-d39b7ee278ff", "car-c586ea8f-f017-
47aa-8954-c50bd2918e06", "car-bcf1572a-7efc-4b8b-87b5-15095e71e8f2"]
  },
  "certificateinfo":{
    "pincode" : "123456",
    "certificate":
"MIIG1DCCBoGgAwIBAgIPVdopRm+fB6ouwIGYNBHDMaOGCCqFAwcBAQMCMIIIBPTEdMBSGCSqGSIb3D
QEJARYOcmVzcEBnYXotaXMucnUxCzAJBgNVBAYTA1JVMR4wHAYDVQQHDBXQoS3Qn9C10YLQtdGA0LH
Rg9GA0LMxJzAlBgNVBAoMhtCT0JDQl9CY0J3Q.....pNce0KDQnNCh0JXQoNCS0JjQoTELMak
GA1UECwwCSVQxZTBjBgNVBAMMXNCi0LXRgdGC0L7QstGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0Y/
+cAC8xOAwk/FTcAAADoMAwGA1UdEwEB/wQCMAAwCgYIKoUDBwEBAwIDQQC0YDE2oEtIoD94uLiGJGK
mbzraLHwKESzEiZ/AEby5kYwC438UfLL5hnr4tgCH4IQMgN/voe2RN6pnjdDukvbQ"
  }
}
```

На выходе в результате успешного выполнения операции, в случае представления данных в виде *rawdata*, возвращается блок подписанных данных в формате JSON:

```
{
  "resultdata": ""
}
```

"resultdata"	Созданная подпись в формате BASE64
--------------	------------------------------------

На выходе в результате успешного выполнения операции будет создан файл, указанный в поле «*output*», содержащий отделенную ЭП для указанных в поле «*signxadesids*» секций, код ответа «200».

Для проверки созданной отделенной ЭП в формате XAdES необходимо добавить подпись в определенную секцию для подписей в исходном файле, указанном в поле «*input*», и выполнить функцию «*cryptverifier*» (подробное описание приведено в разделе 4.8 «Проверка подписи») для этого файла.

4.7 Пролонгация подписи

В случае, когда срок действия подписи истекает, ее можно продлить, используя функцию «*cryptcreator*». Данная функция имеет синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/cryptcreator>

JSON:

```
{
  "source":
  {
    "rawdata":
    {
      "data": "",
      "detacheddata": ""
    },
    "file":
    {
      "input": "",
      "originfile": "",
      "output": ""
    }
  },
  "upgrade":
  {
    "tspaddress": "",
    "stampoid": ""
  }
}
```

"source"	Набор параметров для ввода исходных данных для подписи. Содержит параметры <i>rawdata</i> или <i>file</i>
"rawdata"	Набор параметров для работы с данными. Содержит параметры: <i>data</i> – обрабатываемая информация, закодированная в BASE64. <i>detacheddata</i> – исходные данные для подписи в формате BASE64 (используется при работе с отделенной подписью)
"file"	Набор параметров для работы с файлами. Содержит параметры: <i>input</i> – путь к входному файлу (подпись), <i>originfile</i> – путь к файлу с исходными данными в случае добавления/обновления отделенной подписи, <i>output</i> – путь к выходному файлу. Путь задается в формате C:/Folder/File
"upgrade"	Параметр для продления срока действия подписи. Содержит параметры: <i>tspaddress</i> – адрес службы штампов времени;

[Перед началом использования приложения «Litoria Crypto Service 2»](#)

необходимо убедиться в наличии корневого сертификата для цепочки сертификатов службы штампов времени в хранилище сертификатов «Доверенные корневые центры сертификации».

stampoid – алгоритм хэширования для штампа (поддерживаемый службой tsp)

Пример запроса:

```
{
  "source":
    {
      "rawdata":
        {
          "data":
            "MIAGCSqGSIb3DQEHAqCAMIACAQEExDDAKBggqhQMHAQECAzCABgkqhkiG9w0BBwGggCSABAtIZWxsbyx3b3JkIQAAAAAAAAKCCBogwggaEMINwQGdheilpcy5ydTELMaKGA1UEBhMCU1UxHjAcBgNVBACMFdChLdCf0LXRgtC10YDQsdGD0YDQszEnMCUGA1UECgwe0JPQkNCX0JjQndCk0J7QoNCc0KHQldCg0JLQmNChMQswCQYDVQQLEDAJVVdfSfMGoGA.....1UEAwXj0KLQtdGB0YLQvtCy0YvQuSDRg9C00L7RgdGC0L7QstC10YDRj9GO0YnQuNC5INGG0LXQvdGC0YAg0QTARgtC+0J/RgNC++BRMGMTk4MDk2MUEwPwYDVQQJDDjRg9C7LiDQmtGA0L7QvdGI0YLQsNC00YLRgdC60LDRjywg0LQuMTAsINC70LjRgtC10YDQsCDQkAIPVdhpUM6H+/yN38Xvg7D2TzhWgAoh7hb5eq237CTtLW9Wnt4I3MAAAAAAAAA="
        }
      },
    "upgrade":
      {
        "tspaddress": "http://testca.gaz-is.ru/tsp2012.srf ",
        "stampoid": "1.2.643.7.1.1.2.2"
      }
    }
}
```

На выходе в результате успешного выполнения операции пролонгации, в случае представления данных в виде *rawdata*, возвращается блок подписанных данных в формате JSON:

```
{
  "resultdata": ""
}
```

<code>"resultdata"</code>	Продленная подпись в формате BASE64
---------------------------	-------------------------------------

В случае представления данных в виде *file* результат выполнения запишется в файл *output*, код ответа «200».

Пример ответа:

```
{
  "resultdata":
```

```
"MIAGCSqGSIB3DQEHAqCAMIACAQEExKTAKBggqhqMHAQECAzAbBgsqhkig9w0BCRACMDEMMAoGCCqFA  
wcBAQICMIAGCSqGSIB3DQEHAkBgqhqMHAQEDAzCCAUQxHTAbBgkqhkiG9w0BCQEWdnJlc3BAZ2F6L  
WlZLnJlMQswCQYDVQQGEwJSVTEeMBwGA1UEBwwV0KEt0J\+0L3RiNGC0LDQtNGC0YHQuTcw0Y8sINC  
0LjEwLCDQu9C40YLQtdGA0LAg0J.....CCD3vnAAvMTgMJPxU3AAABbzAPBgNVHRMBAf8EBTAA  
DAQH\MAoGCCqFAwcBAQMDA4GBACKaGrPByXStm2yU7viqXV002ZLqBlGthTfKXiWflDeSeJLv5Vj5  
oGGEEv9rw\daS5av+6+DrRZPcRpyRzmiEjfiRca4TsNVtW1oFmOZntdwpr23z2eWbWApPnuFWPR2q  
J1XZBf7MFXyiltDnf9vhs3oJUsCxYiIc+wRIiHkrMZQAAAAAAAA"
```

4.8 Проверка подписи

Для проверки подписи используется функция «*cryptverifier*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/cryptverifier>

JSON:

```
{  
  "source":  
  {  
    "rawdata":  
    {  
      "data": "",  
      "detacheddata": "",  
      "m4ddata": "",  
      "m4ddetachedcms": ""  
    },  
    "file":  
    {  
      "input": "",  
      "originfile": "",  
      "output": "",  
      "m4dfile": "",  
      "m4ddetachedcmsfile": ""  
    }  
  },  
  "verifyparam":  
  {  
    "getdata": bool  
  }  
}
```

"source"

Набор параметров для ввода исходных данных для подписи.

	Содержит параметры <i>rawdata</i> и <i>file</i>
"rawdata"	Набор параметров для работы с данными. Содержит параметры: <i>data</i> – обрабатываемая информация, закодированная в BASE64; <i>detacheddata</i> – исходные данные для подписи в формате BASE64 (используется при работе с отделенной подписью); <i>m4ddata</i> – данные МЧД (используется опционально); <i>m4ddetachedcms</i> – отделённый cms (используется опционально).
"file"	Набор параметров для работы с файлами. Содержит параметры: <i>input</i> – путь к входному файлу (подпись); <i>originfile</i> – путь к файлу с исходными данными в случае проверки отделенной подписи; <i>output</i> – путь к выходному файлу; (путь задается в формате C:/Folder/File) <i>m4dfile</i> – файл МЧД (используется опционально); <i>m4ddetachedcmsfile</i> – отделённый cms файл (используется опционально).
"verifyparam"	Набор параметров для проверки подписи. Содержит параметры: <i>getdata</i> – флаг получения данных после проверки подписи (<i>true/false</i>) Используется опционально

Пример запроса:

```
{
  "source":
  {
    "rawdata":
    {
      "data":
      "MIAGCSqGSIb3DQEHAqCAMIACAQExDDAKBggqhQMHAQECAjCABgkqhkiG9w0BBwGggCSABA5I ZWxsbywgd29yZCEhIQAAAAAAAAAKChvcNAQkBFg5yZXNwQGdhei1pcy5ydTElMAkGA1UEBhMCU1UxHjAcBgNVBACMFdChLdCf0LXRgtC10YDQsdGD0YDQs.....zEnMCUGA1UECgwe0JPQkNCX0JjQndCk0J7QoNCC0KHQldCg0JLQmNChMQswCQYDVQQLEDAJJDVDF1MGMGGA1UEAwxc0KLQtdGB0YLQvtCy0YvQuSDRg9C00L7RgdGC0L7QstC10YANBgNVBAUTBjE5ODA5NjFBMD8GA1UECQw40YPQuy4H/9DmBOWAAAAAAAAA="
    }
  }
}
```

Положительным результатом выполнения операции проверки подписи является код

состояния запроса «200» и ответ, содержащий результаты проверки каждой подписи в сообщении и информацию о сертификатах, участвующих в подписи документа, в формате JSON:

```
{
  "signinfosex":
  [
    {
      "signerchain":
      {
        "ischaincorrect": bool,
        "chainelements":
        [
          {
            "revokeinfo":
            {
              "isselfsignerroot": bool,
              "revstatus": "",
              "ocspinfo":
              {
                "ocsPRESPTIME": "",
                "revocationstatus":,
                "serialnumber": "",
                "ocspchain":
                {
                  "ischaincorrect": bool,
                  "chainelements":
                  [
                    {
                      "revokeinfo":
                      {
                        "isselfsignerroot": bool,
                        "revstatus": ""
                      },
                      "certificate": ""
                    },
                    {
                      "revokeinfo":
                      {
                        "isselfsignerroot": bool,
                        "revstatus": ""
                      },
                      "certificate": ""
                    }
                  ]
                }
              }
            }
          ]
        }
      },
    ]
  ]
},
```

```

    "certificate": ""
  },
  {
    "revokeinfo":
    {
      "isselfsignerroot": bool,
      "revstatus": ""
    },
    "certificate": ""
  }
]
},
"deathtime": "",
"signatureindex":,
"verifycertificateresult": bool,
"signalg": "",
"comment": "",
"signtype": [
"qualifyinfo": {
  "caqualifiedstatus": "",
  "certfielderror": "",
  "accreditstatus": "",
  "qualifiedstatus": bool
  "nonqualstrings": []
},
],
"verifyresult": bool,
"signaturetime": ""
}
]
}

```

"signinfosex"	Совокупность подписей в сообщении
"signerchain"	Информация о цепочке сертификации ключа подписи
"ischaincorrect"	Параметр, показывающий корректность построения цепочки. Принимает значение <i>true</i> – в случае, если цепочка корректна, <i>false</i> – если цепочка нарушена
"chainelements"	Описание всех сертификатов в цепочке
"revokeinfo"	Информация о статусе отзыва сертификата
"isselfsignerroot"	Параметр, показывающий, является ли сертификат самовыпущенным
"revstatus":	Статус отзыва сертификата
"ocspinfo":	Информация о проверке отзыва по OCSP

"certificate"	Сертификат ключа подписи в BASE64
"deathtime"	Срок окончания действия подписи
"signatureindex"	Индекс подписи в сообщении
"verifycertificateresult"	Флаг корректной проверки действительности сертификата ключа подписи
"signalg"	Алгоритм подписи
"comment"	Комментарий к подписи
"signtype"	Тип подписи. <i>Basic</i> - усиленная, <i>Advanced</i> - усовершенствованная, <i>Timestamp included</i> – содержит штамп времени, <i>Archived</i> - архивная
"qualifyinfo"	Набор параметров проверки подписи на квалифицированность: <i>caqualifiedstatus</i> – статус квалифицированности УЦ (qualified/unknown/nonqualified); <i>certfielderror</i> – код ошибки квалифицированности, если нет, то равен 0; <i>accreditstatus</i> – статус аккредитации УЦ; <i>qualifiedstatus</i> – статус квалифицированности подписи (true/false); <i>nonqualstrings</i> – описание ошибки квалифицированности
"verifyresult"	Флаг математической корректности подписи
"signaturetime"	Время создания подписи

Пример ответа:

```
{
  "signinfosex": [
    {
      "signerchain": {
        "ischaincorrect": false,
        "chainelements": [
          {
            "certificate":
"MIIF3DCCBYmgAwIBAgIPVdi9W0CBGvloMqzAzLiXMAoGCCqFAwcBAQMCMIIIBPTEdMBSGCSqGSIB3D
QEJARYOcmVzcEBnYXotaXMucnUxXzAJBgNVBAYTA1JVMR4wHAYDVQQHDBXQoS3Qn9C10YLQtdGA0LH
Rg9GA0LMxJzAlBgNVBAoM.....HtCT0JDQl9CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoTELMak
```

```
GA1UECwwCSVQxZTBjBgNVBAMMXNCi0LXRgdGC0L7QstGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0Y\/  
RgNCwINCQMB4XDITixMDEyMDEyNTEyOVVoXDITixMDEzMDEyNTEyOVowDTELMAkGA1UEAwwCQ1MwZjAfb  
ggqhQMHAQEBAATATBgcqhQMCAiQABggqhQMHAQECAgNDAARAny2cjA21D74NTQ7V3hI1e\/xq2zLS\  
SCe7CBEMmoOWKTzXnlBqxuLslNTXz3nyQim0Rd0HXImkazUgo0\/BsdY0kYu9ocsAYA8kw==",  
    "revokeinfo": {  
        "isselfsignerroot": false,  
        "revstatus": "unrevoked"  
    }  
},  
{  
    "certificate":  
"MIIDwzCCA3CgAwIBAgIPE+cAC8x0Awk\/FTcAAADoMAoGCCqFAwcbAQMCMIIIBPTEdMBsGCSqGSIB3  
DQEJARYOcmVzcEBnYXotaXMucnUxCzAJBgNVBAYTAlJVMR4wHAYDVQQHDBXQoS3Qn9C10YLQtdGA0L  
HRg9GA0LMxJzAlBgNVBAoMH.....tCT0JDQl9CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoTELMA  
kGA1UECwwCSVQxZTBjBgNVBAMMXNCi0LXRgdGC0L7QstGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0Y\  
RjtGJ0LjQuSDRhtC10L3RgtGAINCT0JDQl9CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoSAyMDEyMQ8wD  
QYDVQQFEwYxOTgwOTYxQTA\/00YE7A==",  
    "revokeinfo": {  
        "isselfsignerroot": false,  
        "revstatus": "undertermined"  
    },  
    "errorcode": "Нет доверия к сертификату центра  
сертификации. Установите сертификат в хранилище доверенных корневых  
сертификатов центров сертификации. "  
    }  
},  
    "chainerrors": [  
        "Сертификат или один из сертификатов в цепочке издан  
недоверенным центром сертификации. ",  
        "Ошибка проверки доказательств действительности  
сертификата на момент создания ЭП."  
    ]  
},  
    "deathtime": "29-12-2030 23:01:25",  
    "signatureindex": 0,  
    "verifycertificateresult": false,  
    "signalg": "ГОСТ Р 34.11-2012\/34.10-2012 256 бит",  
    "signtype": [  
        "Basic",  
        "Advanced"  
    ],  
    "qualifyinfo": {  
        "caqualifiedstatus": "unknown",  
        "certfielderror": 0,  
        "accreditstatus": "unknown",  
        "qualifiedstatus": false  
    },  
    "verifyresult": true,
```

```
    "signaturetime": "29-01-2021 08:59:57"  
  }  
]  
}
```

4.9 Извлечение данных

Для извлечения данных используется функция «*cryptverifier*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/cryptverifier>

JSON:

```
{  
  "source":  
  {  
    "rawdata":  
    {  
      "data": ""  
    },  
    "file":  
    {  
      "input": "",  
      "output": ""  
    }  
  },  
  "decryptparam":  
  {  
    "certificate": "",  
    "pincode": ""  
  }  
}
```

"source"	Набор параметров для ввода исходных данных подписи. Содержит параметры <i>rawdata</i> и <i>file</i> .
"rawdata"	Набор параметров для работы с данными. Содержит параметры: <i>data</i> – обрабатываемая информация, закодированная в BASE64
"file"	Набор параметров для работы с файлами. Содержит параметры: <i>input</i> – путь к входному файлу (подпись),

	<p><i>output</i> – путь к выходному файлу. Путь задается в формате C:/Folder/File</p>
"decryptparam"	<p>Набор параметров для извлечения данных. Содержит параметры: <i>certificate</i> – сертификат получателя в формате BASE64. <i>pincode</i> – PIN-код к контейнеру с закрытым ключом сертификата получателя (если параметр не задан - в процессе выполнения операции потребуется ввод PIN-кода)</p>

Пример запроса:

```
{
  "source":
  {
    "rawdata":
    {
      "data": "MIAGCSqGSIB3DQEHA6CAMIACAQAxggJ8MIICeAIBADCCAVkwggFEMR0wGwYJKoZIhvcNAQ
kBFg5yZXNwQGdheilpcy5ydTELMakGA1UEBhMCU1UxHjAcBgNVBACMFdChLdCf0LXRgtC10YDQsdGD
0YDQszEnMCUGA1UECgwe0JPQkNCX0JjQndCk0J7QoNCC0KHQldCg0JLQmNChMQswCQYDVQQQLDAJVD
FsMGoGA1UEAwxj0KLCX0JjQnd.....Ck0J7QoNCC0KHQldCg0JLQmNChIDIwMTIguU3Ryb25nM
Q8wDQYDVQQFEwYxOTgwOTYxQTA/+14utVSLvqsNAxOLQ6VxQetcucNrKK/HNzsDIYPA/SHS29A312p
kiyXGLBABZQC/Z8tfVulqzPWdiak6134ohrlX112FIfIo7NKeQiGuXtKAV4nhxZ1BfIK+WcNADzZf
sb8ECMbd911KROUwMIAGCSqGSIB3DQEHATAdbgYqhQMCAhUwEwQIVZG30gWagKUGByqFAwICHwGggA
QFOSN9YMAAAAAAAAAAAAAA"

    },
    "decryptparam":
    {
      "certificate": "MIIGgzCCBe+gAwIBAgIPVdhTPrxVSRrVsJ1tr31WMAoGCCqFAwCBAQMDMIIBRDE
dMBsGCSqGSIB3DQEJARYOcmVzceBnYXotaXMucnUxCzAJBgNVBAYTA1JVMR4wHAYDVQQHDBXQoS3Qn
9C10YLQtdGA0LHRg9GA0LMxJzAlBgNVBAoMHtCT0JDQ19CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoTE
LMakGA1UECwwCSVQxbDBqBgNVBAMMY.....9Ci0LXRgdGC0L7QstGL0Lkg0YPQtNC+0YHRgtC
+0LLQtdGA0Y/+0L3RiNGC0LDQtNGC0YHQutCw0Y8sINC0LjEwLCDQu9C40YLQtdGA0LAg0JCCD3vnA
AvMTgMJPxU3AAABbzAMBgNVHRMBAf8EAjAAMAoGCCqFAwCBAQMDA4GBAAIcQ0qn5i/7437vYpEjxtq
Ni2TVcj15PFS0nu5chre47GjCiCuJYZgo0pWTMEP83FABvjSZukv3eatp71jWEzkLBL0Lu8XCIq+1r
SghirZzE6KQeVDeY9BRuosh18N/QkFv9vDlw5ufiQ+gdpXl3FBihBfE7jmFLVE3hdMYUrC",
      "pincode": "12345678",
      "verifyparam":
      {
        "getdata" : true
      }
    }
  }
}
```

На выходе в результате успешного выполнения операции извлечения данных, в случае представления данных в виде *rawdata*, возвращается блок извлеченных данных в формате JSON:

```
{
  "resultdata": ""
}
```

"resultdata"	Извлеченные данные в формате BASE64
--------------	-------------------------------------

В случае представления данных в виде *file* результат выполнения запишется в файл *output*, код ответа «200».

Пример ответа:

```
{
  "resultdata": "aGVsbG8="
}
```

4.10 Хеширование

Для хеширования данных используется функция «*hash*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/hash>

JSON:

```
{
  "source":
  {
    "data": "",
    "file": ""
  },
  "hashalg": ""
}
```

"source"	Набор параметров для ввода исходных данных для хеширования. Содержит параметры <i>data</i> или <i>file</i>
"data"	Набор параметров для работы с данными. Содержит параметры: <i>data</i> – обрабатываемая информация, закодированная в BASE64

"file"	Путь к файлу, для которого необходимо получить хэш-значение. Путь задается в формате C:/Folder/File
"hashalg"	Используемый алгоритм хэширования

Пример запроса:

```
{
  "source":
  {
    "data": "MTIzNDU2Nzg5UXdlcnR5"
  },
  "hashalg": "1.2.643.7.1.1.2.2"
}
```

Результатом успешного выполнения операции является код состояния запроса «200» и ответ в формате JSON:

```
{
  "hashvalue": ""
}
```

"hashvalue"	Хэш-значение от данных
-------------	------------------------

Пример ответа:

```
{
  "hashvalue": "5CK6wAЕkdBk4HCP31K5Np2WMM0V+N5+10MXBNb2itZI="
}
```

4.11 Подпись хеша и создание отдельной подписи

Для подписи хеша используется функция «*signhash*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/signhash>

JSON:

```
{
  "hashvalue": "",
  "certificateinfo":
  {
```

```

    "certificate": "",
    "pincode": ""
  },
  "parameters":
  {
    "comment": "",
    "includetimestamp":
    {
      "advanced": bool,
      "tspaddress": ""
    }
  }
}

```

"hashvalue"	Подписываемое хэш-значение в формате BASE64
"certificateinfo"	Набор параметров для работы с подписью
"certificate"	Сертификат, с использованием которого будет создаваться подпись в формате BASE64
"pincode"	PIN-код для доступа к контейнеру
"parameters"	<p>Дополнительные (опциональные) параметры для создания подписи.</p> <p>Содержит параметры:</p> <p><i>comment</i> –комментарий к подписи,</p> <p><i>includetimestamp</i> – параметры для создания усовершенствованной ЭП,</p> <p><i>advanced</i> – включение/отключение создания усовершенствованной подписи (<i>true/false</i>),</p> <p><i>tspaddress</i> – адрес службы штампов времени.</p> <hr/> <p>Перед началом использования приложения «Litoria Crypto Service 2» необходимо убедиться в наличии корневого сертификата для цепочки сертификатов службы штампов времени в хранилище сертификатов «Доверенные корневые центры сертификации».</p> <hr/> <p>Используется опционально</p>

Пример запроса:

```

{
  "hashvalue": "5CK6wAEkdBk4HCP31K5Np2WMM0V+N5+10MXBNb2itZI=",
  "certificateinfo":

```

```
{  
  
  "certificate": "MIIGgzCCBe+gAwIBAgIPVdhTPrXVSRrVsJ1tr3lWMAoGCCqFAwcBAQMDMIIBRDE  
dMBsGCSqGSIB3DQEJARYOcmVzcEBnYXotaXMucnUxCzAJBgNVBAYTAlJVMR4wHAYDVQQQHDBXQoS3Qn  
9C10YLQtdGA0LHRg9GA0LMxJzAlBgNVBAoMHtCT0JDQl9CY0J3QpNce0KDQnNCh0JXQoNCS0JjQoTE  
LMAkGA1UECwwCSVQxbDBqBgNVBAMMY9Ci0LXRgdGC0L7QstGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0  
Y/+0LLRi9C5INGD0LTQvtGB0YLQvtC.....y0LXRgNGP0Y7RidC40Lkg0YbQtdC90YLRgCDQk  
9CQ0JfQmNCd0KTQntCg0JzQodCV0KDQktCY0KEgMjAxMiBTdHJvbmcxDzANBgNVBAUTBjE5ODA5NjF  
BMD8GA1UECQw40YPQuy4g0JrRgNC+S0nu5chre47GjCiCuJYZgo0pWTMEP83FAbvjSZukv3eatp71j  
WEzkLbLoLu8XCIq+1rSghirZzE6KQeVDeY9BRuosh18N/QkFv9vDlwW5ufiQ+gdpXl3FBihBfe7jmF  
LVE3hdMYUrC",  
    "pincode": "12345678"  
  },  
  "parameters":  
  {  
    "comment": "Подпись hash",  
    "includetimestamp":  
    {  
      "tspaddress": "http://testca.gaz-is.ru/tsp.srf"  
    }  
  }  
}
```

В результате успешного выполнения операции, возвращается код состояния запроса «200» и ответ в формате JSON:

```
{  
  "signdata": ""  
}
```

"signdata"

Подписанные данные в формате BASE64

Пример ответа:

```
{  
  "signdata":  
  "MIIVYQYJKoZIhvcNAQcCoIIVUjCCFU4CAQExDDAKBggqhqMHAQECAzALBgkqhkiG9w0BBwGgggaHM  
IIGgzCCBe+++0YHRgtC+0LLQtdGA0Y\ /CY0KExDzANBgNVBAUTBjE5ODA5NjFBMD8GA1UECQw40YPQ  
uy4g0JrRgNC+0L3RiNGC0LDQtNGC0YHQtC.....w0Y8sINC0LjEwLCDQu9C40YLQtdGA0LAg  
0JACD3vnAAvMTgMJPxU3AAAGfzAIBgYqhqMCAhMEQJnyAPx4WLGfvQ0W\ /nnyfYogZfD3Dw1j9tgay  
uthqWyoQwkN\ /DsARHFdkYyeMyoZmDjJjNG7kUjCDwznbEvNy4s="
```

4.12 Присоединение подписи

Для присоединения подписи используется функция «*attachsignature*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/attachsignature>

JSON:

```
{
  "source":
  {
    "rawdata":
    {
      "data": "",
      "detacheddata": ""
    },
    "file":
    {
      "originfile": "",
      "detachedsign": "",
      "output": ""
    }
  }
}
```

"source"	Набор параметров для ввода исходных данных для подписи. Содержит параметры <i>rawdata</i> или <i>file</i>
"rawdata"	Набор параметров для работы с данными. Содержит параметры: <i>data</i> – обрабатываемая информация, закодированная в BASE64. <i>detacheddata</i> – исходные данные для подписи в формате BASE64 (используется при работе с отделенной подписью)
"file"	Набор параметров для работы с файлами. Содержит параметры: <i>detachedsign</i> – путь к входному файлу (подпись), <i>originfile</i> – путь к файлу с исходными данными в случае добавления/обновления отделенной подписи, <i>output</i> – путь к выходному файлу. Путь задается в формате C:/Folder/File

Пример запроса:

```
{
  "source":
```

```
{
  "rawdata":
  {
    "data": "SGVsbG8sIHdvcnQhISE=",
    "detacheddata":
    "MIAGCSqGSIb3DQEHAqCAMIACAQExDDAKBggqhQMHAQECAjCABgkqhkiG9w0BBwEAAKCCBeAwggXcMIIFiaADAgECAg9V2L1bQIEa+.....+lJTEOz1Wmg8+G5cIaiZwX9JohqyGgxiIr77hrQezmZ65I6EzozR+oc4c42DVI25QvESjQjBAMA4GA1UdDwEB/wQEAWIAxjAdBgNVHQ4EFgQUv4oZDGyZiHTc4AAVWlm9aPttVgYwDwYDVR0TAQH/BAUwAwEB/zAKBggqhQMHAQEDAgNBABPlnOQ1Si3/fHqKzWMwZL9tqgLaXgPhqLNULEV+f/giFWX0hfC2p0WzCBPavKLOhaOWAi3iR4oEsQoH/9DmBOwAAAAAAAA="
  }
}
```

Результатом успешного выполнения операции является код состояния запроса «200» и ответ в формате JSON:

```
{
  "resultdata": ""
}
```

"resultdata"	Набор данных – присоединенная подпись в формате BASE64
--------------	--

В случае представления данных в виде *file* результат выполнения запишется в файл *output*, код ответа «200».

Пример ответа:

```
{
  "resultdata":
  "MIAGCSqGSIb3DQEHAqCAMIACAQExDDAKBggqhQMHAQECAjCABgkqhkiG9w0BBwGggCSABA5IZWxsbywgd29yZCEhIQAAAAAAAAKCCBeAwggXcMIIFiaADAgECAg9V2L1bQIEa+G5cIaiZwX9JohqyGgxiIr77hrQezmZ65I6EzozR+oc4c42DVI25Q.....vESjQjBAMA4GA1UdDwEB/wQEAWIAxjAdBgNVHQ4EFgQUv4oZDGyZiHTc4AAVWlm9aPttVgYwDwYDVR0TAQH/BAUwAwEB/zAKBggqhQMHAQEDAgNBABPlnOQ1Si3/fHqKzWMwZL9tqgLaXgPhqLNULEV+f/giFWX0hfC2p0WzCBPavKLOhaOWAi3iR4oEsQoH/9DmBOwAAAAAAAA="
}
```

4.13 DVCS-запрос и разбор квитанции

Для создания DVCS-запроса и разбора квитанции с целью проверки электронной подписи используется функция «*dvcs*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/dvcs>

JSON:

```
{
  "serviceaddress": "",
  "transactionid": "",
  "source":
  {
    "file":
    {
      "signfile": "",
      "originfile": "",
      "m4dfile": "",
      "m4ddetachedcmsfile": ""
    },
    "rawdata":
    {
      "signdata": "",
      "detacheddata": "",
      "m4ddata": "",
      "m4ddetachedcms": ""
    }
  },
  "userinfo":
  {
    "certificate": "",
    "pincode": "",
    "username": "",
    "password": ""
  }
  "getdvc": bool
}
```

"serviceaddress"	Адрес сервиса DVCS в формате « <i>http://<HOST>:PORT/Api/Dvc</i> »
"transactionid"	Параметр для задания уникального имени транзакции. Используется опционально
"source"	Набор параметров для ввода исходных данных для подписи. Содержит параметры <i>rawdata</i> или <i>file</i>
"rawdata"	Набор параметров для работы с данными. Содержит параметры: <i>signdata</i> – обрабатываемая информация, закодированная в BASE64; <i>detacheddata</i> – исходные данные для подписи в формате BASE64 (используется при работе с отделенной подписью);

	<p><i>m4ddata</i> – возможность проверять подпись совместно с МЧД (может использоваться только с ПК «Litoria DVCS» версии 6.1.0.3 и выше, проверка возможна только для данных) (используется опционально);</p> <p><i>m4ddetachedcms</i> – отделённый cms для МЧД (используется опционально).</p>
"file"	<p>Набор параметров для работы с файлами. Содержит параметры:</p> <p><i>signfile</i> – путь к входному файлу (подпись),</p> <p><i>originfile</i> – путь к файлу с исходными данными в случае добавления/обновления отделенной подписи. (путь задается в формате C:/Folder/File)</p> <p><i>m4dfile</i> – данные МЧД (используется опционально);</p> <p><i>m4ddetachedcmsfile</i> – отделённый cms для МЧД (используется опционально).</p>
"userinfo"	<p>Информация для доступа к сервису DVCS. Содержит параметры:</p> <p><i>certificate</i> - сертификат для подписи запроса в формате BASE64, или <i>username</i> и <i>password</i> – логин и пароль пользователя. При использовании с ПК «Litoria DVCS» версии 6.0 и выше в полях <i>username</i> и <i>password</i> необходимо указать токен доступа, предварительно созданный и сохраненный в настройках пользователя ПК «Litoria DVCS».</p>
"pincode"	<p>PIN-код к контейнеру с сертификатом, который привязан к учетной записи пользователя ДТС</p>
"getdvc"	<p>Получать ли квитанцию в ответе. Принимаемые значения: <i>true/false</i></p> <p>Используется опционально</p>

Пример запроса:

```
{
  "serviceaddress": "http://10.72.14.151/api/dvc",
  "getdvc": true,
  "getdata": true,
  "source":
  {
    "rawdata":
    {
      "signdata": "MIAGCSqGS Ib3DQEHAqCAMIACAQEExDDAKBggqhQMHAQECAzCABgkqhkiG9w0BBwGggC
SABA5IZWxsbywgd29yZCEhIQAAAAAAAKCCBocwggADMIIF76ADAgECAg9V2FM++0YDQvNGB0LXRgNC
y0LjRgSIgKNCy0LXRgNGB0LjQuCAxLjApDF/QodC10YDRgtC40YtQuNC60LDRgiDRgdC+0L7RgtCy0
```

```
LXRgtGB0YLQstC40Y8g0KTQodC.....RINCg0L7RgdGB0LjQuCDQodCkLzEyNC0yNTQwINC+/
/BgNVBAkMONGD0LsuINCa0YDQvtC90YjRgtCw0LTRgtGB0LrQsNGPLCDQtC4xMCwg0LvQuNGC0LXRg
NCwINCQAg9V2FM+tdVJGtWwnW2vfVYwCgYIKoUDBwEBAQIEgYBWoogTABQszN1Ys1NuJJ16ytFvBdd
qRZSLO1m5e0sVZHmqLzwnQ9cK18bYGKblf5KnNBGAf7L9kkvxBcILm7R+Qc++Riy562lDuFj0oz2t7
JMz16dmQBmTubAxwW9H7UMMKJj8tVhsHX6cu8Csn3T8UksEi7Apgwmbc8zfEjigSQAAAAAAAAA=="
"m4ddata": "PD95bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGlucz0iV01ORE9XUy0xMjUxIj8+PNTg6es
gwxw8c/w7uM9ItbP0NAg1M3RINDUIFsxLjE1LjBdIDC5fDx1O7w7D0iNS4wMSIgyOTU4OnrPSJPT
19ET1ZFTF83ODA1Xzc4MDVfNzgzODAxNzk2ODc4MDUwMTAwMV8yMDIyMDQxOF81NDdhMTBmOS04N2V
iLTQ0NDEtOTNmNS01OWE0YjUzYmMxN2QiPgoJPMtu6vPs5e3yIMrNxD0iMTEExMMDxMCIgyu7kzc49I
jc4MDUiPgoJCTzR4sTu4iDE4PLgzeD3PSIwMS4wNC4yMDIyIiDN7uzE7uLl8D0iNTQ3YTEwZjktODd
lYi00NDQxLTkzZjUtNt1hNGI1.....M2JjMTdkIj4KCQkJPMtg8uDO6u7tPjMwLjA0LjIwMjI
8L8Tg8uDO6u7tPgoJCTwv0eLE7uI+/+NpZ25lZFNpZ25hdHVyZVByb3BlcnRpZXM+CiAgICAgIDwve
GFkZXM6U2lnbmVkuUHJvcGVydGllcz4KICAgIDwveGFkZXM6UXVhbGlmeWluZ1Byb3BlcnRpZXM+CiA
gPC9kczpPYmplY3Q+CjwvZHM6U2lnbmF0dXJlPjwv1ODp6z4="
}
},
"userinfo":
{
"certificate": "MIIGZDCCBc2gAwIBAgIPVdklozX+gjimkPAWoG0iMA0GCSqGSIb3DQEBBQUAMI
BPDEdMBsGCSqGSIb3DQEJARYOcmVzceBnYXotaXMucnUxXzAjbG9uYm95bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGlucz0iV01ORE9XUy0xMjUxIj8+PNTg6es
S3Qn9C10YLQtdGA0LHRg9GA0LMxJzAlBgNVBAoMHtCT0JDQl9CY0J3QpNCe0KDQnNCh0JXQoNCS0Jj
QoTELMakGA1UECwwCSVQxZDBiBgN.....VBAMW9Ci0LXRgdGC0L7QstGL0Lkg0YPQtNC+0YH
RgtC++cAC8x0Awk/FTcAAAIYMAwGA1UdEwEB/wQMAAwDQYJKoZIhvcNAQEFBQADgYEABbegG18m7t
/adDyW9VITHzmTeanA+9LwN6K0TBV+2eWmeF97fUkv78bA1s570o06uT+HzG4xJWK4u50QCpqUrnmh
CAMEzEcN99d7LTva66lwxZoZ0xjY94oBOnH710pzuUWYOfNOn95iw5EUpaHkz+K/zaH4Ri46yXmLoz
m/eRw="
}
}
```

Результатом успешного выполнения операции является квитанция от доверенной третьей стороны в формате JSON:

```
{
  "dvcinfo":
  {
    "dvcddata": "",
    "dvcexpire": "",
    "signinfo":
    {
      "deathtime": "",
      "signalg": "",
      "signatureindex": ,
      "signaturetime": "",
      "signerchain":
      {
        "chainelements": [],
        "ischaincorrect": bool
      },
      "signtype": [ "" ],
    }
  }
}
```

```

        "verifycertificateresult": bool,
        "verifyresult": bool
    },
    "signinfosex": [
        {
            "deathtime": "",
            "signalg": "",
            "signatureindex": ,
            "signaturetime": "",
            "signerchain":
                {
                    "chainelements": [],
                    "ischaincorrect": bool
                },
            "signtype": [ "", "" ],
            "verifycertificateresult": bool,
            "qualifyinfo": {
                "caqualifiedstatus": "",
                "certfielderror": "",
                "accreditstatus": "",
                "qualifiedstatus": bool
                "nonqualstrings": []
            },
            "verifyresult": bool
        }
    ],
    "m4dinfo":
        {
            "permissions": [],
            "status": "",
            "statustext": ""
        }
    }
}

```

"dvcinfo"	Информация об ответе доверенной третьей стороны
"dvcddata"	Квитанция DVCS в формате BASE64 (при условии, что в запросе был установлен флаг «getdvc»)
"dvceexpire"	Срок действия созданной квитанции (при условии, что в запросе был установлен флаг «getdvc»)
"signinfo"	Информация о подписи квитанции
"deathtime"	Срок действия подписи квитанции
"signalg"	Алгоритм подписи квитанции
"signatureindex"	Индекс (порядковый номер) подписи квитанции

"signaturetime"	Время создания подписи квитанции
"signerchain"	Информация о цепочке сертификатов подписи квитанции
"chainelements"	Информация обо всех сертификатах в цепочке подписи квитанции
"ischaincorrect"	Результат проверки корректности цепочки сертификатов подписи квитанции (true/false)
"signtype"	Тип подписи квитанции
"verifycertificateresult"	Результат проверки действительности сертификата подписи квитанции (true/false)
"verifyresult"	Результат проверки математической корректности подписи квитанции (true/false)
"status"	Статус ответа DVCS: 0 - Неизвестно 1 - Проверка успешна 2 - Ошибка формирования запроса 3 - Ошибка подписи запроса клиентским сертификатом 4 - Отсутствует соединение с сервисом 5 - Ответ не соответствует ожидаемому ответу сервиса 6 - Во время обработки запроса на сервере произошла ошибка 7 - Отсутствие доверия к сертификату сервиса 8 - Целостность DVC нарушена 9 - Полученный ответ не соответствует запросу
"statustext"	Дополнительные текстовые сообщения сервера ДТС
"signinfosex"	Информация о всех подписях, отправленных на проверку в ДТС
"deathtime"	Срок действия проверяемой подписи
"signalg"	Алгоритм проверяемой подписи
"signatureindex"	Индекс (порядковый номер) проверяемой подписи
"signaturetime"	Время создания проверяемой подписи
"signerchain"	Информация о цепочке сертификатов проверяемой подписи
"chainelements"	Информация обо всех сертификатах в цепочке проверяемой подписи

"ischaincorrect"	Результат проверки корректности цепочки сертификатов проверяемой подписи (true/false)
"signtype"	Тип проверяемой подписи
"verifycertificateresult"	Результат проверки действительности сертификата проверяемой подписи (true/false)
"verifyresult"	Результат проверки математической корректности проверяемой подписи (true/false)
"m4dinfo"	Информация о машиночитаемой доверенности. Содержит параметры: <i>permissions</i> - коды полномочий представителя, на которого выдана МЧД; <i>statustext</i> – дополнительные текстовые сообщения о МЧД; <i>status</i> – статус ответа МЧД.
"qualifyinfo"	Набор параметров проверки подписи на квалифицированность: <i>caqualifiedstatus</i> – статус квалифицированности УЦ (qualified/unknown/nonqualified); <i>certfielderror</i> – код ошибки квалифицированности, если нет, то равен 0; <i>accreditstatus</i> – статус аккредитации УЦ; <i>qualifiedstatus</i> – статус квалифицированности подписи (true/false); <i>nonqualstrings</i> – описание ошибки квалифицированности

Пример ответа:

```
{
  "dvcinfo": {
    "signinfo": {
      "signerchain": {
        "ischaincorrect": true,
        "chainelements": [
          {
            "certificate":
"MIIGtTCCBiGgAwIBAgIPVdmtm3NAN\BnrBUKvjyUMAoGCCqFAwcBAQMDMIIBRDEdMBsGCSqGSIb3
DQEJARYOcmVzceBnYXotaXMucnUxCzAJBgNVBAYTA1JVMR4wHAYDVQQHDBXQoS3Qn9C10YLQtdGA0L
HRg9GA0LMxJzAlBqNVBAoMhtCT0JDQ19CY0J3QpNce0KDQnNCh0JXQoNCS0JjQoTELMaKGA1UECwwC
SVQxbDBqBgNVBAMMY9Ci0LXRgdGC0L.....7QstGL0Lkg0YPQtNC+0YHRgtC+0LLQt dGA0Y\
+0LLQt dGA0Y\ /RjtGJ0LjQuSDRhtC10L3RgtGAINCT0JDQ19CY0J3QpNce0KDQnNCh0JXQoNCS0JjQ
oSAYMDEyIFN0cm9uZzEPMA0GA1UEBRMGMTk4MDk2MUEwPwYDVQQJDDjRg9C7LiDQmtGA0L7QvdGI0Y
```

```

LQsNC00YLRgdC60LDRjywg0LQuMTAsINC70LjRgtC10YDQsCDQkIIPe+cAC8xOAwk\FTcAAAFvMA8
GA1UdEwEB\wQFMAMBAf8wCgYIKoUDBwEBAwMDgYEAuwsAbT7QjHCIQscbEprUmmkcGPriixyw7yMm
1\Xa1Bx\CV31Vv4E805DjU14ctnOAF7XE4AuPcy9UUGCjOUif4XmTGw91xxDAsr9LvbvmuHSdoOY
WqlhQvv503b5R9upvD8K76BisohLWHHSECpr4qSccefyCFjbidusuMxRn+E4=",
    "revokeinfo": {
        "isselfsignerroot": false,
        "revstatus": "unrevoked",
        "ocspinfo": {
            "ocspresptime": "2022-06-17T06:31:44Z",
            "revocationstatus": -1,
            "serialnumber":
"55D9AD9B734037F067ADB50A563C94",
            "ocspchain": {
                "ischaincorrect": true,
                "chainelements": [
                    {
                        "certificate":
"MIIH6TCCB1WgAwIBAgIIPe+cAC8xOAwk\FTcAAAFxMAoGCCqFAwcBAQMDMIIBRDEdMBsGCSqGSib3
DQEJARYOcmVzcEBnYXotaXMucnUxCzAJBgNVBAYTA1JVMR4wHAYDVQQHDBXQoS3Qn9C10YLQtdGA0L
HRg9GA0LMxJzAlBgNVBAoMHtCT0JDQ.....l9CY0J3QpNce0KDQnNCh0JXQoNCS0JjQoTELMA
kGA1UECwwCSVQxbDBqBgNVBAMMY9Ci0LXRgdGC0L7QstGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0Y\
\BAUwAwEB\zAKBggqhQMHAQEDAwOBgQAimhqzwcl0rZtsl074q1ldNNmS6gZRrYU3yl4ln5Q3kni
S7+VY+aBhhBL\a8P3WkuWr\uvg60WT3Eackc5ohI3yEXAOE7DVbVtaBZjmZ7XcKa9t89nlmlgKT5
7hVj0dqidV2QX+zBV8opbQ53\b4bN6CVLAsWIiHPsESIh5KzGUA==",
                        "revokeinfo": {
                            "isselfsignerroot": false,
                            "revstatus": "unrevoked",
                            "crlinfos": {
                                "revocationstatus":
"unrevoked",
                                "crlinfo": [
                                    {
                                        "expirydate":
"2022-06-20T07:27:34Z",
                                        "crlrawdata":
"MIIJ4DCCUwCAQEwCgYIKoUDBwEBAwMwggFEMR0wGwYJKoZIhvcNAQkBFg5yZXNwQGdheilpcy5yd
TELMakGA1UEBhmCU1UxHjAcBgNVBACmFdChLdCf0LXRgtC10YDQsdGD0YDQszEnMCUGA1UECgwe0JP
QkNCX0JjQndCk0J7QoNCc0KHQldCg0JLQmNChMQswCQYDVQQLEDAJJDVDFsMG0GA1UEAwxj0KLQtdGB0
YLQvtCy0YvQuSDRg9C00L7RgdGC0L7QstC10YDRj9G00YnQuNC5INGG0LXQvdGC0YAg0JFPkNCX0Jj
QndCk0J7QoNCc0KHQldCg0JLQmNChIDIWMTIGU3Ryb25nMQ8wDQYDVQQFEwYxOTgwOTYxQTA\++0Y
HRgtC+0LLQtdGA0Y\RjtGJ0LjQuSDRht.....C10L3RgtGAINCT0JDQl9CY0J3QpNce0KDQn
NCh0JXQoNCS0JjQoSAYMDEyIFN0cm9uZzEPMA0GA1UEBRMGMTk4MDk2MUEwPwYDVQQJDDjRg9C7LiD
QmtGA0L7QvdGI0YLQsNC00YLRgdC60LDRjywg0LQuMTAsINC70LjRgtC10YDQsCDQkIIPe+cAC8xOA
wk\FTcAAAFvMAoGCCqFAwcBAQMDA4GBACs71CDFckA+1N16jwW90xCTTGpRZJxR73A1yOr9L8PZ3n
cNQqMrwzL1EG9rBs26sdfjFe2DDhQyKIwGX9C2u+nq1YejPxa9xwMy\tHHSLV5AUD1SpQ0GuSKqOu
MGlzndoZcZkM66MEy\L8fDVBGZjSwKoPZ\Oa5frEWBxuFV3J",
                                "effectivedate":
"2022-06-09T04:27:34Z",

```



```
    }
  },
  {
    "certificate":
"MIIEVDCCA8CgAwIBAgIPE+cAC8x0Awk\ /FTcAAAFvMAoGCCqFAwcBAQMDMIIBRDEdMBsGCSqGSIb3
DQEJARYOcmVzceBnYXotaXMucnUxCzAJBgNVBAYTAlJVMR4wHAYDVQQHDBXQoS3Qn9C10YLQtdGA0L
HRg9GA0LMxJzAlBgNVBAoMHtCT0JDQl9CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoTELMAkGA1UECwwC
SVQxbDBqBgNVBAMMY9Ci0LXRgdGC0L7.....QstGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0Y\ /
\ /Mtk7VBx3oZKTDg2JBNxZo0bkAN3lD5LEvrVJy4inrqfNUHx0LAmgkklA2VzmTeObo+thtDQgFjb9
VaZemdD5L3uzzoL9N7kpo9W1yJLASzxfMX2qfWXz5LWlEalmv0d7qHuUohdtAI1WVSq4B2HhuwTJoS
Foz8wPTALBgNVHQ8EBAMCAcYwHQYDVR0OBBYEFH+B8MkLKgrim0pUkgOaLA3D2JYEMA8GA1UdEwEB\
/wQFMAMBAf8wCgYIKoUDBwEBAwMDgYEA107nYRg2F9TGLCM6Csc1Fbwve2yAoysLaOYSzuPVsaAZ9y
G6C\ /3ty705\ /LxdCKBSTctzZ8FTLEItgrJ33ax5gQEhYWhW3edHRaIoCRs3ZwqgWmV\ /yqCG2Ha40
6TEKfXEjQZtmxlnQaoJUPVHvWvaJ+srNwDgmjI9GhSWQ+KVlf8=",
    "revokeinfo": {
      "isselfsignerroot": true,
      "revstatus": "undertermined"
    }
  }
],
"deathtime": "22-11-2022 09:35:41",
"signatureindex": 0,
"verifycertificateresult": true,
"signalg": "GOST R 34.11-2012\ /34.10-2012 512 bit",
"signtype": [
  "Basic"
],
"qualifyinfo": {
  "caqualifiedstatus": "notqualified",
  "certfielderror": 464910,
  "accreditstatus": "unknown",
  "qualifiedstatus": false,
  "nonqualstrings": [
    "УЦ издателя не является аккредитованным",
    "Не удалось определить тип сертификата для проверки
квалифицированности (он не является ЮЛ, ИП, ФИЗ, ЮЛ иностранного
государства)",
    "Значение ОГРН\ /ОГРНИП владельца сертификата имеет
некорректный формат",
    "Сертификат не содержит ИНН, либо значение ИНН имеет
некорректный формат",
    "В сертификате отсутствует информация о наименовании
аккредитованного УЦ, который выдал сертификат, либо значение имеет
некорректный формат",
    "В сертификате отсутствует информация о местонахождении
аккредитованного УЦ, который выдал сертификат, либо значение имеет
```

```
некорректный формат",
    "В сертификате отсутствует информация о наименовании
средства ЭП владельца квалифицированного сертификата, либо это средство не
является сертифицированным",
    "В сертификате отсутствует информация о классе средств ЭП
владельца квалифицированного сертификата",
    "В сертификате отсутствует информация о типе идентификации
заявителя"
]
},
"verifyresult": true,
"signaturetime": "17-06-2022 09:31:42"
},
"statustext": "Проверка проведена",
"status": 1
},
"signinfosex": [
{
    "signerchain": {
        "ischaincorrect": true,
        "chainelements": [
            {
                "certificate":
"MIIG1DCCBoGgAwIBAgIPVdopRm+fb6ouwIGYNBHDMAoGCCqFAwcbAQMCMII BPTEdMBsGCSqGSIb3D
QEJARYOcmVzcEBnYXotaXMucnUxCzAJBgNVBAYTALJVMR4wHAYDVQQQHDBXQoS3Qn9C10YLQtdGA0LH
Rg9GA0LMxJzAlBgNVBAoMHtCT0JDQl9CY0J3QpNcE0KDQnNCh0JXQoNCS0JjQoTELMAkGAlUECwwCS
VQxZTBjBgNVBAMMXNCi0LXRgdGC0L7Q.....stGL0Lkg0YPQtNC+0YHRgtC+0LLQtdGA0Y\ /R
jtGJ0LjQuSDRhtC10L3RgtGAINCT0JDQl9CY0J3QpNcE0KDQnNCh0JXQoNCS0JjQoSAyMDEyMQ8wDQ
YDVQQFEwYxOTgwOTYxQTA\ /XDTIyMDQyODE1MzkzMl0XDTIzMDQyODE1MzkzMlowgcsxFjAUBgUqhQ
NkAxILMDEyMjgxMTMzOTAxGjAYBggqhQMDgQMBARIMMTEwODAlODg3NTM5MR8wHQYJKoZIHvcNAQkB
FhBhbGV4ZWV2QG1haWwuY29tMQswCQYDVQQGEwJSVTEYMBYGA1UEAwwP0KLQtdGB0YIgoJzQp9CUMS
gwJgYDVQQqDB\ /+cAC8x0Awk\ /FTcAAADoMAwGA1UdEwEB\ /wQMAAwCgYIKoUDBwEBAwIDQQC0YDE
2oEtIoD94uLiGJGKmbzraLHWKESzeiZ\ /AEby5kYwC438UfLL5hnr4tgCH4IQMgN\ /voe2RN6pnjdD
ukvbQ",
                "revokeinfo": {
                    "isselfsignerroot": false,
                    "revstatus": "unrevoked",
                    "ocspinfo": {
                        "ocsppresptime": "2022-06-17T06:31:42Z",
                        "revocationstatus": -1,
                        "serialnumber":
"55DA29466F9F07AA2EC081983411C3"
                    }
                }
            }
        ]
    }
},
{
    "certificate":
"MIIDwzCCA3CgAwIBAgIPe+cAC8x0Awk\ /\ /BAQDAgDGMB0GA1UdDgQWBBS\ /ihkMbJmIdNzgABVaW
```

```
b1o+21WBjAPBgNVHRMBAf8EBTADAQH\ /M.....AoGCCqFAwcBAQMCA0EAE+Wc5CVKLf98eorN
YzBkv22qAtpeA+Gos24sRX5\ /+CIVZfSF8LanRbMIE9q8os6Fo5YCLeJHigSxCgf\ /00YE7A==",
    "revokeinfo": {
        "isselfsignerroot": false,
        "revstatus": "undertermined"
    }
}
]
},
"deathtime": "28-04-2023 18:39:32",
"signatureindex": 0,
"verifycertificateresult": true,
"signalg": "GOST R 34.11-2012\ /34.10-2012 256 bit",
"signtype": [
    "Basic"
],
"verifyresult": true,
"signaturetime": "28-04-2022 18:56:29",
"m4dinfo": {
    "permissions": [
        1,
        3,
        5,
        7,
        9,
        11,
        99
    ],
    "statustext": "Действительна",
    "status": 1
}
}
]
}
```

4.14 Обращение к базе данных

4.14.1 Добавление информации в базу данных

Для добавления информации в базу данных необходимо использовать функцию «*signstorage*», имеющую следующий синтаксис:

Content-Type: *application/json*

Метод: *PUT*

URL: <http://localhost:8080/signstorage>

JSON:

```
{
  "source":
  {
    "rawdata":
    {
      "signdata": "",
      "detacheddata": ""
    }
  },
  "uuid": ""
}
```

"rawdata"	Набор параметров для работы с данными
"signdata"	Подпись в формате BASE64
"detacheddata"	Оригинал данных (используется в случае работы с отделенной подписью)
"uuid"	Уникальный номер транзакции (опциональный параметр, если не задан – генерируется автоматически). Используется опционально

Пример запроса:

```
{
  "source":
  {
    "rawdata":
    {
      "signdata": "MIAGCSqGSIB3DQEHAqCAMIACAQExDDAKBggqhqMHAQECAzCABgkqhkiG9w0BBwGggC
SABAtIZWxsbyx3b3JkIQAAAAAAAAAKCCBogwgggEMIIF8KADAgECAg9V2GlQzof40Hgi7scLTWkwCgYI
KoUDBwEBAwMwggFEMR0wGwYJKoZIhvcNAQkBFg5yZXNwQGdhei1pcy5ydTELMakGA1UEBhMCU1UxHj
AcBgNVBACmFdChLdCf0LXRgtC10YDQsdGD0YDQszEnMCUGA1UECgwe0JPQkNCX0JjQndCk0J7QoNCc
0KHQldCg0JLQmNChMQswCQYDVQQLDAJVV.....DFsMGoGA1UEAwxj0KLQtdGB0YLQvtCy0YvQ
uSDRg9C00L7RgdGC0L7QstC10YDRj9G00YnQuNC5INGG0LXQvdGC0YAg0JPQkNCX0JjQndCk0J7QoN
Cc0KHQldCg0JLQmNChIDIwMTIwMzRyYb25nMQ8wDQYDVQQFEwYxOTgwOTYxQTA/NB4Iu7HC01pMAoGC
CqFAwCBAQECEBGAFAFPiXyP+z5REoq6rpmPs7j11U+lprmhW5zQQHkqrm629RqUVj6hWEzD0S7Y4cKge
5wd5K1fw5wZ7k12Cu2ndljlpqfTdtvm8IOn6PSDPakG+uBi5aXFVnrDiWLplipoY/I6Xbr2nwYUbw7
mDlsOt5b3/lZGdAxGypzSn1wESLQsAAAAAAAAA=",
      "detacheddata": "SGVsbG8sd29yZCEhIQ=="
    }
  }
}
```

В результате успешного выполнения операции в БД помещаются подписанные данные,

а записи присваивается заданный идентификационный номер «*uuid*». Ответ в формате JSON выглядит следующим образом:

```
{
  "md5hash": "",
  "uuid": ""
}
```

"md5hash"	Хэш-значение помещенной в БД записи
"uuid"	Уникальный номер транзакции

При добавлении подписанных данных в БД, осуществляется проверка подписи:

- если подпись усиленная (не усовершенствованная), она автоматически усовершенствуется и добавляется в БД уже в статусе УЭП.
- если подпись усовершенствованная (УЭП), сравнивается количество дней до её истечения со значением, указанным в параметре «*limitdays*». Если срок жизни (количество дней до истечения) меньше указанного в параметре, подпись продляется еще раз и становится архивной.

Пример ответа:

```
{
  "md5hash": "5Lf8BiNHZxHI\ /OUlo6md+w==",
  "uuid": "d6a63651-67ac-435b-82c9-5c22025c3ced"
}
```

4.14.2 Поиск информации в базе данных

Для поиска информации в созданной базе данных необходимо использовать функцию «*signstorage*», имеющую следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/signstorage>

JSON:

```
{
  "uuid" : "",
  "md5hash" : "",
  "detacheddata": "",
  "getdata" : bool
}
```

"uuid"	Уникальный номер транзакции
"md5hash"	Хэш-значение от подписи, хранящееся в БД
"detacheddata"	Оригинал данных (используется в случае работы с отдельной подписью)
"getdata"	Получать ли данные по завершению операции (данные представлены в формате BASE64) Принимаемые значения: <i>true/false</i> . Используется опционально

Параметры "uuid", "md5hash", "detacheddata" могут использоваться как по отдельности, так и все вместе.

Пример запроса:

```
{
  "uuid" : "d6a63651-67ac-435b-82c9-5c22025c3ced",
  "getdata" : true
}
```

В случае если флаг «getdata» в запросе был установлен в true, при использовании в поиске параметров «uuid» и «md5hash» возвращается запись из БД в формате JSON:

```
{
  "signdata": [
    "<Подпись_1>",
    "<Подпись_2>",
    ...,
    "<Подпись_n>"
  ],
  "detacheddata": ""
}
```

"signdata"	Массив подписей
"detacheddata"	Оригинал данных для отделённой подписи

Пример ответа:

```
{
  "signdata": [
    "MIAGCSqGSIB3DQEHAAQCAMIACAQExKTAKBggqhkjQMHAQECAzAbBgsqhkjG9w0BCRACMDemMAoGCCqFA
wcBAQICMIAGCSqGSIB3DQEHAAoIIIGiDCCBoQwggXwoAMCAQICD1XYaVDOh\ /jQeCLuxwtNaTAKBg
gqhkjQMHAQEDAzCCAUXHTAbBgkqhkiG9w0BCQEWdnlc3BAZ2F6LWlzlNj1MQswCQYDVQQGEwJSVTEe
```

```
MBwGA1UEBwwV0KEt0J\ / .....+0L3RiNGC0LDQtNGC0YHQutCw0Y8sINC0LjEwLCDQu9C40YL
QtdGA0LAg0JCCD3vnAAvMTgMJPxU3AAABbzAPBgNVHRMBAf8EBTADAQH\ /MAoGCCqFAwCBAQMDA4GB
ACKaGrPByXStm2yU7viqXV002ZLqBlGthTfKXiWflDeSeJLv5Vj5oGGEEv9rw\ /daS5av+6+DrRZPc
RpyRzmiEjfIRcA4TsNVtW1oFmOZntdwpr23z2eWbWApPnuFWPR2qJ1XZBf7MFXYiltDnf9vhs3oJUs
CxYiIc+wRIiHkrMZQAAAAAAAA"
    ],
    "detacheddata": "SGVsbG8sd29yZCE="
}
```

В случае если флаг «getdata» в запросе был установлен в false, при выполнении поиска по параметру «detacheddata» возвращается запись из БД в формате JSON:

```
{
  "signdata": [
    "<Подпись_1>",
    "<Подпись_2>",
    ...,
    "<Подпись_n>"
  ]
}
```

"signdata"	Массив подписей
------------	-----------------

Пример ответа:

```
{
  "signdata": [
    "MIAGCSqGSIB3DQEHAqCAMIACAQEExKTAkBggqhQMHAQECAzAbBgsqhkig9w0BCRACMDEMMAoGCCqFA
wcBAQICMIAGCSqGSIB3DQEHAQAoIIGiDCCBoQwggXwoAMCAQICD1XYaVDOh\ /jQeCLuxwtNaTAKBg
gqhQMHAQEDAzCCAUXHTAbBgkqhkiG9w0BCQEWdnJlc3BAZ2F6LWlzlNj1MQswCQYDVQGEwJSVTEe
MBwGA1UEBwwV0KEt0J\ /QtdGC0.....LXRgNCx0YPRgNCzMScwJQYDVQKDB7Qk9CQ0JfQmNC
d0KTQntCg0JzQodCV0KDQktCY0KEExCzAJBgNVBAsMAklUMWwwagYDVQDDGPQotC10YHRgtC+0LLRi
9C5INGD0LTQvtGB0YLQvtCy0LXRgNGP0Y7RidC40Lkg0YbQtdC90YLRgCDQk9CQ0JfQmNCd0KTQntC
g0JzQodCV0KDQktCY0KEgMjAxMiBTdHJvbmcxDzANBgNVBAUTBjE5ODA5NjFBMD8GA1UECQw40YPQu
y4g0JrRgNC\ /QtdGC0LXRgNCx0YPRgNCzMScwJQYDVQKDB7Qk9CQ0JfQmNCd0KTQntCg0JzQodCV0
KDQktCY0KEExCzAJBgNVBAsMAklUMWwwagYDVQDDGPQotC10YHRgtC+\ /daS5av+6+DrRZPcRpyRzm
iEjfIRcA4TsNVtW1oFmOZntdwpr23z2eWbWApPnuFWPR2qJ1XZBf7MFXYiltDnf9vhs3oJUsCxYiIc
+wRIiHkrMZQAAAAAAAA"
  ]
}
```

4.15 Проверка соответствия ДТС квитанции данным

Для проверки соответствия ДТС квитанции данным используется функция «dvscscompliance», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/dvcscompliance>

JSON:

```
{
  "source":
  {
    "rawdata":
    {
      "signdata": "",
      "detacheddata": ""
    },
    "file":
    {
      "signfile": "",
      "originfile": ""
    }
  },
  "dvccdata": ""
}
```

"source"	Набор параметров для ввода исходных данных для проверки соответствия с квитанцией. Содержит параметры <i>rawdata</i> и <i>file</i>
"rawdata"	Набор параметров для работы с данными. Содержит параметры: <i>signdata</i> – обрабатываемая информация, закодированная в BASE64. <i>detacheddata</i> – исходные данные для подписи в формате BASE64 (используется при работе с отделенной подписью)
"file"	Набор параметров для работы с файлами. Содержит параметры: <i>signfile</i> – путь к входному файлу (подпись), <i>originfile</i> – путь к файлу с исходными данными в случае применения отделенной подписи. Путь задается в формате <i>C:/Folder/File</i>
"dvccdata"	ДТС квитанция, закодированная в BASE64

Результатом выполнения операции проверки соответствия ДТС квитанции данным является код состояния запроса «200» и ответ, содержащий результат проверки в формате JSON:

```
{
  "dvcccompliance": bool
}
```

`"dvccompliance"`

Результат соответствия ДТС квитанции данным (true/false)

4.16 Аутентификация запросов по токену безопасности

После запуска ПО «Litoria Crypto Service 2» при обращении к любому контроллеру (например, при выполнении активации или настроек) для доступа к основным функциям сервиса по токену безопасности в файле конфигурации необходимо задать следующие параметры: токен безопасности и установить флаг для включения его использования.

```
main:
{
  securitytokenenable = false;
  securitytoken = "";
};
```

Изначально флаг «*securitytokenenable*» имеет значение «false». Для включения использования аутентификации по токену нужно поставить значение флага «true» внутри конфигурационного файла и перезапустить сервис. Извне изменять эту настройку нельзя.

После включения режима аутентификации по токену безопасности необходимо указывать его в любом JSON запросе к сервису.

Для POST, PUT запросов в корне:

```
{
  "securitytoken": "",
  "param1": 1,
  "param2": 2
}
```

Для GET запросов нужно преобразовать значение токена безопасности с помощью url encode (например, <https://www.urlencoder.org/>).

Примеры запросов GET:

```
http://127.0.0.1:8089/certs?storename=MY&securitytoken=GH%26JEge12mG9ay88z%23VQ
http://127.0.0.1:8089/coreinfo?securitytoken=GH%26JEge12mG9ay88z%23VQ
```

4.17 Получение отчета о проверке подписи в ДТС

Для получения отчета о проверке ЭП через ДТС в формате html используется функция «*dvcreport*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/dvcreport>

JSON:

```
{
  "source":
  {
    "rawdata":
    {
      "data": ""
    },
    "file":
    {
      "input": "",
      "output": "",
    }
  }
}
```

"source"	Набор параметров для ввода исходных данных для получения отчета. Содержит параметры <i>rawdata</i> и <i>file</i>
"rawdata"	Набор параметров для работы с данными. Содержит параметры: <i>data</i> – обрабатываемая информация (dvc-квитанция), закодированная в BASE64.
"file"	Набор параметров для работы с файлами. Содержит параметры: <i>input</i> – путь к входному файлу (dvc-квитанция в DER-кодировке), <i>output</i> – путь к файлу, в который будет сохранен отчет о проверки ЭП из указанной dvc-квитанции (лучше указывать с расширением .html). Путь задается в формате <i>C:/Folder/File</i>

Пример запроса:

```
{
  "source":
  {
    "rawdata":
    {
      "data": "MIInygYJKoZIhvcNAQcCoIIInuzCCJ7cCAQMxCTAHBgUrDgMCGjCCHfIGCyqGS Ib3DQeJEA  
EIoIId4QSChd0wgh3ZMCIKAQIYDzIwMjExMDE5MTQzOTM0WqAMoAoGA1UEMKADAQH//Q19CY0J3QpN  
Ce0KQDQnNCh0JXQoNCS0JjQoTELMaKGA1UECwwCSVQxZDBiBgNVBAMMw9Ci0LXRgdGC0L7QstGL0Lkg  
0YPQtNC+0YHRgtC+0LLQtdGA0Y/RjtGJ0LjQuSDRhtC10L3RgtGAINCT0JDQ19CY0J3QpNCE0KQDnN  
Ch0JXQoNCS0JjQoSBSU0ExDzANBgNVBAUTBjE5ODA5NjFBMD8GA1UECQw40YPQuY4g0JrRgNC+0L3R  
iNGC0LDQtNGC0YHQutCw0Y8sINC0LjEwLCDQu9C40YLQtdGA0LAg0JACDwjWwcC1dM8o1frwDPO9Yz  
AHBgUrDgMCGqCBjDAABgkqhkiG9w0BC.....QMxDQYLKoZIhvcNAQkQAQgWHAYJKoZIhvcNAQ  
kFMQ8XDTIxMTAxOTE0Mzkn1owIwYJKoZIhvcNAQkEMRYEFKwBtDgQi7YYdvz1nDoMJD1gRP+tMCsG  
CyqGS Ib3DQeJEAIMMRwwGjAYMBYEFDRwyBEOai0iIMiSDpIPbcJWsY4dMA0GCSqGS Ib3DQeEBAQUABI  
GAGBZ0H7MRr05Z3JZBfCKSR9nLBNCEVLkGkjdcUw1B60ero1vMQpOew1IkRT45nyRqMX5dl//q88WL  
B+dRL/Mf1rKmdEzBfZVK8aGN83Z2wijVDx+oGTr2D1LZSGp6gTxxOYJ0AuNk3inZT3jhQtYPjCpKtP
```

```
0mOb1VBsC2J7GON40="
  }
}
}
```

Результатом выполнения операции создания отчета о проверке ЭП через ДТС в случае представления квитанции в виде *rawdata* является код состояния запроса «200» и ответ, содержащий блок данных отчета в формате JSON:

```
{
  "resultdata": ""
}
```

"resultdata"

Отчет о проверке ЭП через ДТС в формате BASE64

В случае представления данных в виде *file* результат выполнения операции запишется в файл *output*, код ответа «200».

Пример ответа:

```
{
  "resultdata":
  "JVBERi0xLjQKJcOiW6MKMSAwIG9iago8PAovVG10bGUgKkKL0NyZWf0b3IgcKkKL1Byb2R1Y2VyI
  Cj+\wBRANQAIAAA1AC4AMQA1AC4AMykKL0NyZWf0aW9uRGF0ZSAoRDoyMDIyMDYwOTEwMjA0NiswMy
  cwMCCpCj4++UgMDAwMDAgbiAKMDAwMD.....AwMNjI5OSAwwMDAwMCBUiAowMDAwMDI3NDg1ID
  AwMDAwIG4gCjAwMDAwMjU3MDAgMDAwMDAgbiAKdHJhaWxlcg08PAovU216ZSAyOSAKL0luZm8gMSAw
  IFIKL1Jvb3QgMiAwIFIKPj4Kc3RhcncR4cmVmCjI3NjQ2IAolJUVPRgo="
}
```

4.18 Получение информации о машиночитаемой доверенности

Для получения информации о машиночитаемой доверенности используется функция «*m4dinfo*», имеющая следующий синтаксис:

Content-Type: *application/json*

Метод: *POST*

URL: <http://localhost:8080/m4dinfo>

JSON:

```
{
  "source":
  {
    "file":
    {
      "input": "",
      "originfile": "",
      "m4dfile": "",
      "m4ddetachedcmsfile": ""
    }
  }
}
```

```

    },
    "rawdata":
    {
        "m4ddata": "",
        "m4ddetachedcms": ""
    }
},
}

```

"source"	Набор параметров для ввода исходных данных для подписи. Содержит параметры <i>rawdata</i> или <i>file</i>
"rawdata"	Набор параметров для работы с данными. Содержит параметры: <i>data</i> – обрабатываемая информация, закодированная в BASE64. <i>m4ddata</i> – возможность проверять подпись совместно с МЧД (может использоваться только с ПК «Litoria DVCS» версии 6.1.0.3 и выше, проверка возможна только для данных) <i>m4ddetachedcms</i> – отделённый cms для МЧД (используется опционально).
"file"	Набор параметров для работы с файлами. Содержит параметры: <i>input</i> – путь к входному файлу (подпись), <i>originfile</i> – путь к файлу с исходными данными в случае добавления/обновления отделенной подписи, <i>m4dfile</i> – данные МЧД (используется опционально); <i>m4ddetachedcmsfile</i> – отделённый cms для МЧД (используется опционально).

Примеры передаваемой информации в виде данных:

- в запросе содержится подпись и МЧД, подписанная в формате XAdES:

```

{
  "source": {
    "rawdata": {
      "data":
      "MIAGCSqGSIB3DQEHAqCAMIACAQExDDAKBggqhQMHAQECAjCABgkqhkiG9w0BBwGggCSABDDQryDQv
      9C+0LzQvdGOINGH0YPQtNC90L7QtSDQvNCz0L3QvtCy0LXQvdGM0LUuLi4AAAAAACgggbLMIIGxzC
      CBnSgAwIBAgIPVduYNMITcDIbgNK+0YHRgtC+0LLQtDGA0Y/RjtGJ0LjQuSDRhtC10L3RgtGAINCT0
      JDQ19CY0J3QpNCe0KDQnNCh0JXQoNCS0JjQoSAYMDEyMQ8wDQYDVQQFEwYxOTgwOTYxQTA/BgNVBAk
      MONGD0LsuINCa0YDQvtC90YjRgtC.....w0LTRgtGB0LrQsNGPLCDQtC4xMCwg0LvQuNGC0LX
      RgNCwINCQA9V25g0whNwMhuA0r5wbAQwCgYIKoUDBwEBAQEeQIBQkqZbswER+x1Fn0GCWmnkVK0u3
      YXJAeEDYFINmasTxJ9PT2qz0au5JcLyaEStKDHhvTwUGfNq7N9v8xT+zjQAAAAAAA=",
      "m4ddata":

```



```
y4M7q7u0+MjMuMDYuMjAyNDwvxODy4M7q7u0+++TE7uo9Ijc4MC0wNjgiLz48L9Hi5eTU6OfLPjwv0
eLP8OXkPjwv0eLT78/w5eQ+PM/u50/o8eDt8iDU4Ozo6+j/PSLPzsPQxCHNzskiIMjs/z0iwMvFytH
AzctQIiDO8vfl8fLi7j0izsvFw87CyNciLz48L8Tu6vPs5e3yPjwv1ODp6z4=",
    "m4ddetachedcms":
"MIAGCSqGSIB3DQEHAqCAMIACAQExDDAKBggqhQMHAQECAjCABgkqhkiG9w0BBwEAAKCCCDowggg2M
IIH46ADAgECAg9V25J1jiVg9XU/+0YDQvNGB0LXRgNCy0LjRgSiGKNCy0LXRgNGB0LjQuCAxLjApDF
/QodC10YDRgtC40YTQuNC60LDRgiDRgdC+0L7RgtCy0LXRgtGB0YLQstC40Y8g0KTQodCRINCg0L7R
gdGB0LjQuCDQodCkLzEyNC0yNTQwINC+0YI.....gMTUuMDEuMjAxNQwG0J3QtdGCMB0GA1Ud
DgQWBQR+8z3BM/+0YHRgtC+0LLQtdGA0Y/RjtGJ0LjQuSDRhtC10L3RgtGAINCT0JDQ19CY0J3Qp
NCe0KDQnNCh0JXQoNCS0JjQoSAYMDEyMQ8wDQYDVQQFEWYxOTgwOTYxQTA/BgNVBAkMONGD0LsuINC
a0YDQvtC90YjRgtCw0LTRgtGB0LrQsNGPLCDQtC4xMCwg0LvQuNGC0LXRgNCwINCQAg9V25J1jiVg9
XU/IoOuxkIwCgYIKoUDBwEBAQEEOGD8WGh5D1g32oLEDulQKYdE6wabPeVk/LC5o+67dMYdMr3ZRIX
OQZrBLV3+UKjPDUT15hq6QTZpiTn1MyIMzb8AAAAA="
    }
  }
}
```

Примеры передаваемой информации в виде файлов:

- в запросе содержится подпись и МЧД, подписанная в формате XAdES:

```
{
  "source": {
    "file": {
      "input": "D:/document_to_sign_withM4D.txt.p7s",
      "originfile": "D:/document_to_sign_withM4D.txt",
      "m4dfile": "D:/CorrectM4Dtsp.sign.xml"
    }
  }
}
```

- в запросе содержится только МЧД, подписанная в формате XAdES:

```
{
  "source": {
    "file": {
      "m4dfile": "D:/CorrectM4D.sign.xml"
    }
  }
}
```

- в запросе содержится подпись, неподписанная МЧД и отделенная ЭП в формате CAdES для МЧД:

```
{
  "source": {
    "file": {
      "input": "D:/document_to_sign_withM4D.txt.p7s",
      "originfile": "D:/document_to_sign_withM4D.txt",
      "m4dfile": "D:/CorrectM4D.xml",

```

```
"m4ddetachedcmsfile": "D:/CorrectM4D.xml.p7s"
}
}
}
```

Результатом успешного выполнения операции является ответ в формате JSON:

```
{
  "m4dinfo": {
    "delegate": {
      "givenname": "",
      "surname": "",
      "inn": "",
      "snils": ""
    },
    "principal": {
      "ogrn": "",
      "givenname": "",
      "surname": "",
      "organization": "",
      "inn": "",
      "snils": ""
    },
    "number": "",
    "permissions": [],
    "statustext": "",
    "timenotafter": "",
    "timenotbefore": "",
    "status": 0
  }
}
```

"m4dinfo"	Информация о машиночитаемой доверенности
"delegate"	Информация о представителе
"givenname"	Имя и отчество предствителя/доверителя
"surname"	Фамилия представителя/доверителя
"inn"	ИНН представителя/доверителя
"snils"	СНИЛС представителя/доверителя
"principal"	Информация о доверителе
"ogrn"	ОГРН доверителя
"organization"	Организация доверителя
"number"	Номер МЧД

"permissions"	Коды полномочий представителя, на которого выдана МЧД
"statustext"	Дополнительные текстовые сообщения о МЧД
"timenotafter"	Время окончания действия МЧД
"timenotbefore"	Время начала действия МЧД
"status"	Статус ответа МЧД

Пример ответа:

```
{
  "m4dinfo": {
    "delegate": {
      "givenname": "АНАТОЛИЙ АНАТОЛЬЕВИЧ",
      "surname": "ИВАНОВ",
      "inn": "000001350991",
      "snils": "00000816025"
    },
    "principal": {
      "ogrn": "0000047050199",
      "givenname": "АЛЕКСАНДР АЛЕКСАНДРОВИЧ",
      "surname": "ПЕТРОВ",
      "organization": "ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ ККК",
      "inn": "000009054911",
      "snils": "00000785978"
    },
    "number": "1234563",
    "permissions": [
      21,
      22
    ],
    "statustext": "Неопределено",
    "timenotafter": "2024-06-22T21:00:00Z",
    "timenotbefore": "2023-06-22T21:00:00Z",
    "status": 0
  }
}
```

5 Возвращаемые значения и коды ответов

Общие ошибки		
Код ответа	Ошибка	Возможная причина
400		Синтаксическая ошибка в запросе
501		Использование любого метода, отличного от указанного для данной операции
401	{ "licenseinfo": { "message": "Триальная лицензия "Litoria Crypto Service" истекла. Для продолжения работы активируйте лицензию либо обратитесь в службу поддержки email: support@gaz-is.ru, Телефон: (812) 677-20-50" } }	Временный срок использования сервиса без лицензии истек. Необходимо выполнить активацию сервиса.
500	{ "error": "Не удалось сохранить настройки в config-файл." }	Синтаксическая ошибка в запросе.
400	{"error": "Отсутствует поле запроса либо данные не заданы: '<поле_запроса>'"} }	Отсутствие параметра "поле_запроса"/пустое значение параметра "поле_запроса"
400	{ "error": "Отсутствуют данные в одном или нескольких полях запроса: '<поле1>', '<поле2>', ..." }	Пустые значения параметров блока
400	{ "error": "Неверный формат json." }	Пустое значение логического параметра
500	{ "errors": ["Не найден указанный модуль.\r\n"] }	Неверный PIN-код для операции
500	{ "errors": ["Не удалось установить сертификат в хранилище сертификатов.", "Неверный набор данных используемого сертификата."], }	Нарушенная структура BASE64-представления сертификата

	<pre>}</pre>	
500	<pre>{ "errors": ["Объект или свойство не найдено.\r\n"] }</pre>	Попытка удалить сертификат, которого нет в хранилище
Запрос на сертификат		
400	<pre>{ "error": "Отсутствуют данные в одном или нескольких полях запроса: '<поле1>', '<поле2>', ..." }</pre>	Пустые значения параметров блока
500	<pre>{ "errors": ["Объект уже существует. "], }</pre>	Повторяющееся значение параметра "containername"
500	<pre>{ "errors": [], }</pre>	Неверное значение параметра "corename"
500	<pre>{ "errors": ["Указаны неправильные флаги. "], }</pre>	Неверное значение параметра "keysize"
500	<pre>{ "errors": ["Данные запроса на сертификат имеют некорректный формат.", "Имя субъекта имеет некорректный формат", "Ошибка кодирования объектного идентификатора."], }</pre>	Некорректное значение параметра "requestfields"
500	<pre>{ "errors": ["Произошла ошибка в процессе формирования asn.1 структур расширений сертификата.", "Ошибка формирования расширения Enhanced Key Usage.", "Ошибка кодирования одной из целей использования сертификата.", "Ошибка кодирования объектного идентификатора."], }</pre>	Пустое/неверное значение параметра "extkeyusage"

Подпись данных		
500	{ "errors": ["Не удалось получить штамп времени на значение подписи или он некорректен."] }	Неверно указан адрес службы штампов времени
500	{ "errors": ["Ошибка создания ЭП.", "Не удалось сформировать архивный штамп времени и добавить его в обрабатываемую подпись."] }	Неверно указан адрес службы штампов времени для продления подписи
500	{ "errors": ["Для инициализации механизма хэширования передан алгоритм, не являющийся алгоритмом хэширования."] }	Неверный OID алгоритма хэширования
400	{ "error": "Файл не является криптографическим сообщением, либо структура сообщения была нарушена." }	Нарушена структура данных
Подпись хэш-значений и присоединение отделинной подписи		
500	{ "errors": ["Ошибка инициализации данных для хэширования файла.", "Для инициализации механизма хэширования передан алгоритм, не являющийся алгоритмом хэширования."] }	Указан неподдерживаемый алгоритм хэширования
500	{ "errors": ["Ошибка подписи хэш-значения.", "Ошибка инициализации параметров создания ЭП.", "Неверный набор данных сертификата подписчика."] }	Нарушена структура сертификата

500	<pre>{ "errors": ["Ошибка подписи хэш-значения.", "Не удалось получить штамп времени на значение подписи или он некорректен."] }</pre>	Неверно указан адрес службы штампов времени
500	<pre>{ "errors": ["Ошибка инициализации преобразования отделенной ЭП в присоединенную."] }</pre>	Отделённая подпись не соответствует присоединяемым данным
DVCS		
500	<pre>{ "errors": ["Цепочка сертификата подписавшего лица составлена некорректно.", "Сертификат или один из сертификатов в цепочке издан недоверенным центром сертификации. ", "Статус отзыва сертификата или одного из сертификатов в цепочке не определен. ", "Ошибка проверки целостности OCSP-ответа.", "Цепочка проверяемого сертификата составлена некорректно.", "Сертификат или один из сертификатов в цепочке издан недоверенным центром сертификации. ", "Статус отзыва сертификата или одного из сертификатов в цепочке не определен. "] }</pre>	Отсутствуют некоторые сертификаты в цепочке сертификатов
500	<pre>{ "errors": ["Ошибка проверки подписи данных протокола DVCS.", "Ошибка анализа входного документа"] }</pre>	Формат подписи, не поддерживаемый сервером ДТС
400	<pre>{ "error": "Данные не являются подписью." }</pre>	Проверяемые значения не являются подписанными данными
400	<pre>{ "error": "Отсутствует аутентифицирующая информация пользователя." }</pre>	Отсутствует информация для аутентификации пользователя на сервере ДТС

500	<pre>{ "errorinfo": { "statusstring": ["Пользователь отсутствует в системе"], "errorcode": 6 }, "transactionid": "cp-{2885667b-e294-48df-9951-b78946a9c834}", "status": 2 }</pre>	Введены данные пользователя, не зарегистрированного в системе ДТС
500	<pre>{ "errors": ["Ошибка подписи данных протокола DVCS."] }</pre>	Некорректный сертификат
500	<pre>{ "errors": ["Действие было отменено пользователем.\r\n", "Ошибка подписи данных протокола DVCS."] }</pre>	Действие отменено пользователем
400	<pre>{ "error": "Файл не содержит подписи." }</pre>	Указанный файл не содержит подписанных данных, или они не корректны
400	<pre>{ "error": "Данные не являются подписью." }</pre>	Используемая структура данных не является подписью
400	<pre>{ "error": "Одна или несколько подписей некорректны." }</pre>	Некорректная подпись
500	<pre>{ "errors": ["Не удалось подключиться к хосту или прокси серверу. Проверьте, не блокирует ли брандмауэр."] }</pre>	Отсутствует соединение с сервером ДТС
Работа С БД		
400	<pre>{ "error": "В архиве электронных подписей отсутствуют записи, соответствующие Вашему запросу." }</pre>	В БД отсутствуют искомые данные

Перечень сокращений

CSP	–	Cryptographic Service Provider
DVC	–	Data Validation and Certification
DVCS	–	Data Validation and Certification Server
OID	–	Object Identifier
TSP	–	Time-Stamp Protocol
VSD	–	Validation of digitally Signed Document
БД	–	База Данных
ДТС	–	Доверенная Третья Сторона
ПО	–	Программное обеспечение
ЭП	–	Электронная Подпись