ИНФОРМАЦИОННАЯ СИСТЕМА
«УНИВЕРСАЛЬНАЯ СИСТЕМА ДОСТУПА»
(ИС УСД)

Инструкция по организации взаимодействия с внешними информационными системами

Листов 41

АННОТАЦИЯ

Настоящий документ содержит сведения, необходимые для использования программного интерфейса сервера авторизации ИС УСД для организации взаимодействия ИС УСД и внешних государственных и корпоративных информационных систем.

Сервер авторизации ИС УСД (СА ИС УСД) организован на базе программного комплекса «Сервер авторизации АВЕСТ» (AvOAuth).

В документе излагаются сведения, необходимые разработчикам для использования программного интерфейса СА ИС УСД при организации взаимодействия ИС УСД и внешних информационных систем.

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

1.1 Назначение

ИС УСД предназначена для идентификации, аутентификации пользователей внешних ИС и предоставления пользователями согласия (авторизации) на передачу своих идентификационных данных ИС УСД.

СА ИС УСД предоставляет приложениям внешних ИС программные интерфейсы для выполнения процедур идентификации, аутентификации и авторизации, для выработки ЭЦП и других доступных функций.

Функции СА ИС УСД:

– аутентификация пользователей с использованием протоколов аутентификации, перечень протоколов аутентификации является настраиваемым;

– получение согласия пользователя на передачу своих идентификационных данных в зарегистрированную информационную систему (авторизация информационной системы);

– предоставление доступа к идентификационным данным пользователя в рамках сеанса пользователя (идентификация пользователя);

– управление аутентифицированными сеансами пользователя;

– предоставление доступа к дополнительным программным интерфейсам:

- выработка ЭЦП пользователем с использованием доступного средства ЭЦП.

1.2 Условия применения

1.2.1 Регистрация информационных систем

Доступ к программному интерфейсу СА ИС УСД предоставляется по протоколу Oauth2.

Предоставление пользователям внешних ИС сервисов ИС УСД осуществляется через соответствующие приложения внешних ИС, которые в свою очередь взаимодействуют с СА ИС УСД с обеспечением мер безопасности - аутентификация СА ИС УСД и поточное шифрование данных.

Доступ к программному интерфейсу предоставляется внешним информационным системам после их регистрации на СА ИС УСД.

Доступ зарегистрированных информационных систем к идентификационным данным пользователей осуществляется с использованием сертифицированных средств криптографической защиты информации по протоколу согласно СТБ 34.101.65 (TLS.СТБ[1]).

СА ИС УСД реализует программный интерфейс при сетевых подключениях на заданный сетевой порт. Доступ к сетевому порту СА ИС УСД должен быть разрешен в настройках сетевого оборудования владельца информационной системы и владельца СА ИС УСД.

1.3 Настройка рабочего места пользователя внешней ИС

Рабочее место пользователя внешней ИС функционирует под управлением ОС Windows.

На рабочем месте пользователя внешней ИС должны быть предварительно установлены и настроены сертифицированные средства криптографической защиты информации – программный комплекс (ПК) «Комплект абонента ГосСУОК» при использовании протокола TLS согласно СТБ 34.101.65, а также интернет–браузер Internet Explorer.

Доступ пользователей внешних ИС к сервисам идентификации, аутентификации СА ИС УСД осуществляется из браузера Internet Explorer:

– с использованием методов защиты информации, встроенных в интернет–браузеры клиентов; при этом способе предварительная настройка

[1] Сокращенное обозначение "TLS СТБ” означает, что протокол TLS организуется в соответствии с требованиями СТБ 34.101.65-2014 "Информационные технологии и безопасность. Протокол защиты транспортного уровня (TLS)".

рабочего места клиента не выполняется, используются механизмы защиты информации, предоставляемые браузером по умолчанию;

– с использованием криптографической защиты информации по протоколу согласно СТБ 34.101.65.

В зависимости от используемого протокола аутентификации, СА ИС УСД предоставляет доступ к пользовательскому интерфейсу по протоколу TLS согласно СТБ 34.101.65. Для доступа к пользовательскому интерфейсу СА ИС УСД интернет-браузер должен поддерживать данный протокол.

1.3.1. Установка комплекта абонента ГосСУОК

ПК «Комплект абонента ГосСУОК» предназначен для работы на персональном компьютере общего назначения, функционирующем под управлением ОС MS Windows:

– Windows 2003 Server (x32, x64) SP1 или выше;

– Windows XP SP3 (x32);

– Windows XP SP2 (x64);

– Windows 7 (x32, x64);

– Windows 8 (x32, x64);

– Windows 8.1 (x32, x64);

– Windows 2008 R1 Server (x32, x64);

– Windows 2008 R2 Server (x64);

– Windows 2012 Server (x64);

– Windows 2012 R2 Server (x64);

– Windows 10 (x32, x64).

ПК «Комплект абонента ГосСУОК» включает в себя:

– Программное средство криптографической защиты информации «Криптопровайдер Avest CSP BEL» (AvCSPBEL);

– Программный комплекс «Персональный менеджер сертификатов АВЕСТ» (AvPCM);

– Программный комплекс «JCE Provider АВЕСТ» (AvJCEProv).

ПК «Комплект абонента ГосСУОК» совместно с сертификатом открытого ключа и атрибутным сертификатом, сконфигурированный для установки с помощью AvPKISetup, передается пользователям на диске, флеш-носителе или иным способом (порядок определяется РУЦ ГосСУОК, издающим сертификаты и выдающим ПО).

Каждое окно объединенного инсталлятора AvPKISetup снабжено пояснительными надписями, с которыми необходимо ознакомиться и которым необходимо точно следовать.

В любой момент установку можно прервать, нажав кнопку «Отмена».

Для начала установки ПО необходимо запустить файл AvPKISetup2.exe.

Более подробно порядок установки ПК «Комплект абонента ГосСУОК» описан в инструкции по установке, расположенной в папке Docs на носителе, содержащем ПК «Комплект абонента ГосСУОК».

Установка ПК «Комплект абонента ГосСУОК» требуется в следующих случаях:

1. При аутентификации пользователя по протоколу TLS согласно СТБ 34.101.65 с использованием клиентского сертификата на USB-токене:

- сертификаты открытых ключей, которые могут быть использованы для аутентификации, должны быть установлены в справочник «Личные»;

2. При аутентификации пользователя по протоколу TLS согласно СТБ 34.101.65 с использованием клиентского сертификата на USB-токене и с предоставлением атрибутного сертификата:

- сертификаты открытых ключей, которые могут быть использованы для аутентификации, должны быть установлены в справочник «Личные»;

- атрибутные сертификаты, которые могут использоваться при аутентификации, должны быть установлены в справочник «Атрибутные сертификаты».

3. При аутентификации пользователя по протоколу TLS согласно СТБ 34.101.65 с использованием Mobile-ID с клиентским сертификатом.

Персональный менеджер сертификатов может использоваться для установки сертификата корневого УЦ в доверенные для данного компьютера.

1.3.2. Настройка браузера Internet Explorer

Перед началом работы с ИС необходимо настроить браузер Internet Explorer, выполнив следующие действия:

– Добавить адреса *usd.nces.by* и *iusd.nces.by*, в список надежных сайтов: меню *Свойства/ Безопасность/Надежные сайты/Добавить*.

– Зайти в меню *Свойства браузера/Дополнительно*, убедиться, что в разделе Безопасность снята отметка SSL 3.0 и установлены отметки TLS 1.0, TLS 1.1 и TLS 1.2.

Если сертификат открытого ключа сервера авторизации выпущен в другой ИОК, то следует установить доверие к используемой ИОК.

1.4 Настройка сервера внешней ИС

Сервер внешней ИС предназначен для работы на персональном компьютере функционирующим под управлением ОС MS Windows:

– Windows 2003 Server (x32, x64) SP1 или выше;

– Windows 2008 R1 Server (x32, x64);

– Windows 2008 R2 Server (x64);

– Windows 2012 Server (x64);

– Windows 2012 R2 Server (x64).

и ОС Linux:

– Ubuntu Linux, начиная с версии 12.10.

Для обеспечения защищённого канала по протоколу TLS согласно СТБ 34.101.65 при взаимодействии клиентских приложений с сервером авторизации, на сервере внешней ИС может быть установлен программный комплекс «JCE Provider АВЕСТ» AvJCEProv из состава ПК «Комплект абонента ГосСУОК».

При необходимости, на сервере внешней ИС могут использоваться другие сертифицированные средства криптографической защиты информации, обеспечивающие взаимодействие по протоколу TLS согласно СТБ 34.101.65 и совместимость по интерфейсам и форматам данных с СА ИС УСД.

2. ХАРАКТЕРИСТИКИ СЕРВЕРА АВТОРИЗАЦИИ ИС УСД

2.1 Возможности по аутентификации пользователей

Сервер авторизации поддерживает следующие криптографические протоколы аутентификации пользователей:

– Аутентификация по протоколу TLS согласно СТБ 34.101.65 с использованием клиентского сертификата на СКЗИ пользователя.

– Аутентификация по протоколу TLS согласно СТБ 34.101.65 с использованием клиентского сертификата на СКЗИ пользователя и с предоставлением атрибутного сертификата.

– Аутентификация по протоколу Mobile-ID с использованием клиентского сертификата на SIM с функцией ЭЦП.

Параметры протоколов аутентификации приведены в таблице 1.

Таблица 1. Протоколы аутентификации и их параметры

Протокол аутентификации	Идентификатор протокола	Протокол / URL подключения к серверу авторизации	Настройка браузера и рабочего места
TLS с клиентским сертификатом	certificate	TLS.СТБ / https://usd.nces.by	Требуется
TLS с клиентским и атрибутным сертификатами	attribute	TLS.СТБ / https://usd.nces.by	Требуется
Mobile-ID с клиентским сертификатом	phone	TLS.RFC / https://iusd.nces.by	Не требуется
Mobile-ID с клиентским сертификатом	phone	TLS.СТБ / https://usd.nces.by	Требуется

2.2 Возможности по выработке ЭЦП

ИС УСД предоставляет программный интерфейс для выработки ЭЦП с использованием средства ЭЦП, используемого пользователем в аутентифицированном сеансе. ЭЦП вырабатывается по алгоритму, указанному в сертификате открытого ключа пользователя: поддерживаются СТБ 1176.2 (режим с функцией хэширования согласно СТБ 34.101.31) и СТБ 34.101.45.

СА ИС УСД после выработки ЭЦП средством ЭЦП пользователя осуществляет формирование электронного документа согласно СТБ 34.101.23, п.п. 8.2, 8.3 (тип `SignedData`).

3. РЕГИСТРАЦИЯ ПРИЛОЖЕНИЙ ПАНЕЛИ АДМИНИСТРАТОРА ИС УСД

Веб-приложения (сайт или мобильная версия сайта) или автономные приложения (настольные или мобильные) для доступа к API сервера авторизации должны быть предварительно зарегистрированы на сервере авторизации.

Для регистрации владелец информационной системы предоставляет сведения о сетевых адресах информационной системы, на которые должен осуществляться возврат данных от сервера авторизации, например, для web-приложений (сайт или мобильная версия сайта):

`RETURN_URL1=https://service1.company.by/oauth/callback`

`RETURN_URL2=https://service2.company.by/oauth/callback`

и/или автономных приложений (настольных или мобильных), например, `RETURN_URL3=oauth2://company/application` либо допускается иной протокол при реализации взаимодействия между мобильным приложением и web-сервером.

После регистрации владельцу информационной системы предоставляются следующие параметры:

- уникальный идентификатор приложения (пример идентификатора приложения: `CLIENT_ID=NMGB7AIyAu2OJcAIzaS29PCTDrtzyh1YpALQIx)`;
- секретный ключ приложения (пример секретного ключа: `CLIENT_SECRET=0e63f0103970d33ce545db1cab29d783)`;
- сведения об API сервера авторизации:
  - сетевой адрес API сервера авторизации, доступ предоставляется по протоколу TLS согласно СТБ 34.101.65 с односторонней аутентификацией сервера:

    `REQUEST_URL=https://usd.nces.by`

- адрес страницы с пользовательским интерфейсом протоколов аутентификации, доступ предоставляется по протоколу TLS согласно СТБ 34.101.65 с односторонней аутентификацией сервера:
  `REQUEST_AUTH_WEBGUI_URL=https://usd.nces.by/oauth/authorize`
- адрес страницы с пользовательским интерфейсом протоколов аутентификации, доступ предоставляется по протоколу TLS согласно RFC 5246 с односторонней аутентификацией сервера:
  `REQUEST_AUTH_WEBGUI_URL=https://iusd.nces.by/oauth/authorize`

– сведения о сервере, предоставляющем Signature API:

- сетевой адрес сервера Signature API, доступ предоставляется по протоколу TLS согласно СТБ 34.101.65 с односторонней аутентификацией сервера:
  `SIGN_REQUEST_URL=https://usd.nces.by/api`
- адрес страницы с пользовательским интерфейсом протокола выработки ЭЦП, доступ предоставляется по протоколу TLS согласно СТБ 34.101.65 с односторонней аутентификацией сервера:
  `REQUEST_SIGN_WEBGUI_URL=https://usd.nces.by/api`
- адрес страницы с пользовательским интерфейсом протоколов аутентификации, доступ предоставляется по протоколу TLS согласно RFC 5246 с односторонней аутентификацией сервера:
  `REQUEST_SIGN_WEBGUI_URL=https://iusd.nces.by/api`

4. ПРОТОКОЛЫ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ

4.1 Авторизация доступа к API

Авторизация доступа приложения поставщика электронных услуг к серверу авторизации осуществляется после проверки секретного ключа приложения `CLIENT_SECRET`. Секретный ключ приложения `CLIENT_SECRET` передается как параметр POST-запроса.

Авторизация обращения к данным и ресурсам пользователя осуществляется путем включения в HTTP-запрос заголовка `Authorization` согласно RFC 7235 и указанием в нем билета доступа с помощью схемы `Bearer` согласно RFC 6750, например:

```
Authorization: Bearer
533bacf01e11f55b536a565b57531ac114461ae8736d6506a3
```

Для защиты секретного ключа приложения и билета доступа при передаче на сервер авторизации должен использоваться протокол TLS согласно СТБ 34.101.65, поставщик электронных услуг должен установить доверие к цепочке сертификатов сервера авторизации, т.е. сертификат Корневого УЦ ГосСУОК добавить в справочник доверенных УЦ, Сертификат Республиканского УЦ ГосСУОК поместить в справочник промежуточных центров сертификации, проимпортировать соответствующие СОС.

4.2 Общий порядок обращения к API

Приложение внешней ИС – поставщика электронных услуг пользователям должно выполнить следующую последовательность шагов:

1. Выполнить запрос на начало операции аутентификации пользователя: в браузере пользователя должен быть открыт URL сервера авторизации, который реализует заданный протокол аутентификации.
2. Ожидать получение сведений о завершении аутентификации: браузер пользователя будет перенаправлен на URL

REQUEST_AUTH_WEBGUI_URL, указанный при регистрации. При перенаправлении в качестве параметра будет передан код авторизации.

3. Обменять полученный код доступа на билет доступа с помощью соответствующей функции по URL REQUEST_URL: в ходе запроса выполняется авторизация приложения по доступу к API.
4. Получить идентификационные данные и ресурсы пользователя с помощью соответствующей функции по URL REQUEST_URL, предоставив билет доступа.

4.3 Аутентификация пользователей web-приложений

Аутентификация пользователей web-приложений производится в два шага:

- Получение кода авторизации.
- Обмен кода авторизации на билет доступа.

4.3.1 Получение кода авторизации

Web-приложение для аутентификации пользователя должно перенаправить браузер пользователя на соответствующий сетевой адрес сервера авторизации:

```
REQUEST_AUTH_WEBGUI_URL?client_id=<уникальный идентификатор
приложения>&
         response_type=code&
         state=<произвольная строка>&
         authentication=<идентификатор протокола>&
         redirect_uri=<адрес возврата для запроса>&
         scope=<список идентификаторов запрашиваемых параметров>
```

Значения параметров:

Параметр	**Описание**
	Обязательные параметры
`client_id`	Значение уникального идентификатора приложения, полученное при регистрации
`response_type`	Для приложений разрешенным является только значение `code`
`state`	Произвольная строка, в ответе сервера для параметра `state` будет указано переданное значение.
`authentication`	Идентификатор протокола аутентификации (см. раздел 2)
`redirect_uri`	Значение одного из сетевых адресов возврата для запроса авторизации `RETURN_URL`, заданных при регистрации
`scope`	Список идентификаторов запрашиваемых ресурсов и данных пользователя
	Дополнительные параметры
`force_reauth`	Значение `true` для обязательного повтора аутентификации
`attribute`	Объектный идентификатор (OBJECT IDENTIFIER) требуемого атрибута, обязательный параметр при использовании протоколов аутентификации `attribute` и `phone`; строка с текстовым представлением объектного идентификатора.

Значения параметров должны быть переданы согласно спецификации GET-запроса в закодированном виде согласно требованиям формата application/x-www-form-urlencoded.

Параметр `state` позволяет приложению организовывать уникальные сеансы работы пользователя, передавая в `state` идентификатор или данные сеанса пользователя в вызывающем приложении. Состав символов параметра

state определяется разработчиком в целях достижения уникальности. Рекомендуется длина параметра до 256 символов.

В ходе подключения к серверу авторизации выполняется проверка подлинности сервера авторизации по протоколу TLS, затем пользователю будет отображена страница сервера авторизации с предложением ввести данные для выбранного протокола аутентификации и пройти аутентификацию.

В случае успешного прохождения аутентификации пользователю будет отображена страница сервера авторизации с предложением разрешить (авторизовать) приложению доступ к своим ресурсам и персональным данным.

Далее браузер пользователя будет перенаправлен на URL `redirect_uri` с GET запросом с передачей следующих параметров:

- `code` – значение кода доступа; обязательный параметр; строка;
- `state` – значение контекста, из которого осуществлен вызов операции аутентификации; необязательный параметр, присутствует только если был указан в запросе на начало операции аутентификации пользователя; строка.

Значение кода доступа code может быть использовано только *один раз* и только *в течение 30 секунд* после его выпуска сервером авторизации.

В случае отмены аутентификации пользователем браузер пользователя будет перенаправлен на URL `redirect_uri` с GET запросом с передачей следующего параметра:

- `execute` – идентификатор причины отмены аутентификации; обязательный параметр; строка со значением `cancel`;
- `state` – значение контекста, из которого осуществлен вызов операции аутентификации; необязательный параметр, присутствует только если был указан в запросе на начало операции аутентификации пользователя; строка.

В случае ошибки при обращении к серверу авторизации браузер пользователя будет перенаправлен на URL `redirect_uri` с GET запросом с передачей следующих параметров:

- `error` – идентификатор ошибки; строка;
- `error_description` – текст на английском языке с кратким описанием ошибки; строка;
- `state` – значение контекста, из которого осуществлен вызов операции аутентификации; необязательный параметр, присутствует только если был указан в запросе на начало операции аутентификации пользователя; строка.

4.3.2 Обмен кода авторизации на билет доступа

Для получения билета доступа информационная система должна отправить POST-запрос на URL:

```
https://usd.nces.by/oauth/token
```

Примечание: REQUEST_URL только по TLS.СТБ.

Значения параметров:

Параметр	**Описание**
`client_id`	Уникальный идентификатор приложения, полученный при регистрации
`client_secret`	Секретный ключ приложения, полученный при регистрации
`redirect_uri`	Адрес возврата RETURN_URL, заданный при регистрации
`grant_type`	Значение `authorization_code`, указывающее на использование кода авторизации
`code`	Значение кода авторизации

Значения параметров должны быть переданы согласно спецификации POST-запроса в закодированном виде согласно требованиям формата application/x-www-form-urlencoded (должен присутствовать заголовок `Content-type` со значением `application/x-www-form-urlencoded`).

Возвращаемое значение является JSON-структурой и включается в HTTP-сообщение в закодированном согласно требованиям application/json виде.

Могут возвращаться следующие виды HTTP сообщений:

- HTTP сообщение с кодом 200: запрос успешно обработан.

  Содержимое: JSON-структура с билетом доступа.
- HTTP сообщение с кодом 400: в запросе отсутствуют обязательные параметры, имеют неверный формат или неверные значения.

  Содержимое: JSON-структура с кодом ошибки.
- HTTP сообщение с кодом 401: необходима авторизация

  Заголовки:

  `WWW-Authenticate: Bearer realm="api", error="invalid_token"`

  Содержимое: JSON-структура с кодом ошибки.
- HTTP сообщение с кодом 503: сервер временно недоступен.

JSON-структура с билетом доступа включает поля:

- `access_token` – строка со значением билета доступа;
- `expires_in` – число, указывающее срок действия билета доступа, в секундах;
- `scope` – идентификатор протокола аутентификации и значения `scope`, указанные при запросе для получения кода авторизации, через пробел.

Пример ответа с билетом доступа:

```
{"access_token": "533bacf01e11f55b536a565b57531ac114461ae8736d6506a3", "expires_in": 3600, "scope": "sign"}
```

Билет доступа прекращает действие по истечению срока действия или в течение срока действия: при смене пользователем данных авторизации (пароля).

4.4 Аутентификация пользователей автономных приложений

Автономное приложение (настольное или мобильное) для аутентификации пользователя должно запустить браузер пользователя и направить его на соответствующий сетевой адрес сервера авторизации:

```
REQUEST_AUTH_WEBGUI_URL?client_id=<уникальный идентификатор
приложения>&
          response_type=token&
          state=<произвольная строка>&
          authentication=<идентификатор протокола>&
          redirect_uri=<адрес возврата для запроса >&
          scope=<список идентификаторов запрашиваемых параметров>
```

Значения параметров:

Параметр	**Описание**
	Обязательные параметры
`client_id`	Значение уникального идентификатора приложения, полученное при регистрации
`response_type`	Для приложений разрешенным является только значение `token`
`state`	Произвольная строка, в ответе сервера для параметра `state` будет указано переданное значение.
`authentication`	Идентификатор протокола аутентификации (см. раздел 2)
`redirect_uri`	Значение одного из сетевых адресов возврата для запроса авторизации `RETURN_URL`, заданных при регистрации

scope	Список идентификаторов запрашиваемых ресурсов и данных пользователя
	Дополнительные параметры
force_reauth	Значение true для обязательного повтора аутентификации
attribute	Объектный идентификатор (OBJECT IDENTIFIER) требуемого атрибута, обязательный параметр при использовании протоколов аутентификации attribute и phone; строка с текстовым представлением объектного идентификатора.

Значения параметров должны быть переданы согласно спецификации GET-запроса в закодированном виде согласно требованиям формата application/x-www-form-urlencoded.

В ходе подключения к серверу авторизации выполняется проверка подлинности сервера авторизации по протоколу TLS, затем пользователю будет отображена страница сервера авторизации с предложением ввести данные для выбранного протокола аутентификации и пройти аутентификацию.

В случае успешного прохождения аутентификации пользователю будет отображена страница сервера авторизации с предложением разрешить (авторизовать) приложению доступ к своим ресурсам и персональным данным.

Далее браузер пользователя будет перенаправлен по локальному адресу возврата RETURN_URL, указанному при регистрации приложения, с передачей билета доступа к персональным данным:

```
RETURN_AUTH_URL#access_token=<билет доступа>&
                expires_in=<время жизни билета доступа в секундах>&
                token_type=bearer&
                state=<значение параметра state, переданное в запросе>
```

Значения возвращаемых параметров:

Параметр	Описание
access_token	Значение билета доступа
expires_in	Время жизни билета в секундах
token_type	Тип выданного билета: всегда принимает значение «bearer»
state	Значение параметра state из исходного запроса

4.5 Доступ к ресурсам пользователя

- Идентификаторы ресурсов.
- Запрос ресурсов пользователя.

4.5.1 Идентификаторы ресурсов

Ресурсы пользователя включают *универсальные сведения* о пользователе или возможные операции, доступные для любой информационной системы. Поддерживаемые значения идентификаторов ресурсов для переменной `scope`:

- `sign` – доступ к API протокола выработки электронной цифровой подписи.

4.5.2 Запрос ресурсов пользователя

Для получения ресурсов пользователя необходимо отправить POST-запрос на URL:

```
https://usd.nces.by/oauth/resource
```

Примечание: REQUEST_URL только по TLS.СТБ.

Предъявление билета доступа осуществляется путем включения в HTTP-запрос заголовка `Authorization` согласно RFC 7235 и указанием в нем билета доступа с помощью схемы `Bearer` согласно RFC 6750.

Могут возвращаться следующие виды HTTP сообщений:

- HTTP сообщение с кодом 200: данные пользователя получены. Содержимое: JSON-структура с данными.
- HTTP сообщение с кодом 401: необходима авторизация

  Заголовки:

  `WWW-Authenticate: Bearer realm="api", error="invalid_token"`

  Содержимое: JSON-структура с кодом ошибки.
- HTTP сообщение с кодом 503: сервер временно не доступен.

Если доступ к идентификационным данным был разрешен, то будет возвращена JSON–структура следующего вида:

```
{"success": "true", "data": JSON}
```

Поле `data` содержит JSON-структуру с полями, содержащими идентификационные данные пользователя:

- `guid` – строка с уникальным идентификатором пользователя на сервере авторизации;
- `time_created` – дата создания учетной записи пользователя на сервере авторизации;
- `url` – строка с URL личного кабинета пользователя на сервере авторизации;
- `name` – строка с ФИО пользователя;
- `birth_date` – строка с датой рождения пользователя формата DD.MM.YYYY;
- `phone` – строка с номером телефона пользователя (в международном формате), может отсутствовать;
- `email` – строка с email пользователя, может отсутствовать;
- `cert` – JSON–структура с сертификатом в кодировке PEM и с данными, извлеченными из сертификата (для удобства пользования), поля структуры:
  - `pem`: строка с сертификатом в формате PEM;

  - `version`: строка с версией формата сертификата;
  - `serialNum`: строка с номером сертификата в десятичном виде;
  - `serialHex`: строка с номером сертификата в шестнадцатеричном виде;
  - `issuerName`: строка с наименованием издателя сертификата;
  - `issuer`: JSON–структура, содержащая элементы наименования издателя сертификата;
  - `subjectName`: строка с наименованием владельца сертификата;
  - `subject`: JSON–структура, содержащая элементы наименования владельца сертификата;
  - `publicKeyAlgorithm`: строка с OID открытого ключа;
  - `signatureAlgorithm`: строка с OID алгоритма ЭЦП, использованного УЦ;
  - `validity`: JSON–структура, содержащая срок действия сертификата, поля структуры:
    - `start`: дата выпуска сертификата в формате согласно ISO 8601;
    - `end`: дата завершения действия сертификата в формате согласно ISO 8601;
    - `remain`: число дней до завершения действия сертификата.
- `attrcert`[2] – JSON–структура с атрибутным сертификатом в кодировке PEM и с данными, извлеченными из сертификата (для удобства пользования), поля структуры:
  - `pem`: строка с атрибутным сертификатом в формате PEM.
  - `serialHex`: строка с номером сертификата в шестнадцатеричном виде;
  - `serialNum`: строка с номером сертификата в десятичном виде;

[2] При условии предъявления пользователем при аутентификации атрибутного сертификата.

- `issuer`: JSON–структура, содержащая элементы наименования издателя сертификата;
- `issuerName`: строка с наименованием издателя сертификата;
- `validity`: JSON–структура, содержащая срок действия сертификата, поля структуры:
  - `start`: дата выпуска сертификата в формате согласно ISO 8601;
  - `end`: дата завершения действия сертификата в формате согласно ISO 8601;
  - `remain`: число дней до завершения действия сертификата.
- `signatureAlgorithm`: строка с OID алгоритма ЭЦП, использованного УЦ.

4.6 Завершение аутентифицированного сеанса

Для завершения аутентифицированного сеанса пользователя и отмены разрешения на получение персональных данных необходимо отправить POST-запрос на URL:

```
https://usd.nces.by/oauth/revoke
```

Примечание: REQUEST_URL только по TLS.СТБ.

Значения параметров:

Параметр	**Описание**
`client_id`	Уникальный идентификатор приложения, полученный при регистрации
`client_secret`	Секретный ключ приложения, полученный при регистрации
`token`	Значение билета доступа

Значения параметров должны быть переданы согласно спецификации POST-запроса в закодированном виде согласно требованиям формата application/x-www-form-urlencoded (должен присутствовать заголовок `Content-type` со значением `application/x-www-form-urlencoded`).

Если по переданному билета доступа сеанс был завершен, то будет возвращен статус HTTP ответа 200.

В случае ошибки будет возвращен ответ в формате JSON следующего вида:

```
{"error": "invalid_request", "error_description": "Missing token parameter"}
```

4.7 Ошибки аутентификации

В ходе аутентификации пользователя могут возвращаться следующие идентификаторы ошибок:

- `invalid_request` – согласно RFC 6750;
- `invalid_client` – согласно RFC 6750;
- `invalid_grant` – согласно RFC 6750;
- `unauthorized_client` – согласно RFC 6750;
- `unsupported_grant_type` – согласно RFC 6750;
- `invalid_scope` - согласно RFC 6750.

Схема JSON-структуры с описанием ошибки:

```
  {
   "type": "object"
   "$schema": "http://json-schema.org/draft-04/schema#",
   "properties": {
      "error": {
         "type": "string",
         "enum": [ "invalid_request",
                   "invalid_client",
                   "invalid_grant",
                   "unauthorized_client",
                   "unsupported_grant_type",
                   "invalid_scope",
         ],
         "desciption": "Текстовый идентификатор ошибки"
      },
      "error_description": {
         "type": "string",
         "desciption": "Краткое описание ошибки на английском языке в
кодировке ASCII"
      },
      "error_uri": {
         "type": "string",
         "desciption": "URL с подробным описанием ошибки"
      }
   },
   "required": [
      "error"
   ],
}
```

5. ПРОТОКОЛЫ ВЫРАБОТКИ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ

5.1 Общее описание

При необходимости выработки ЭЦП и формирования электронных документов поставщики электронных услуг могут использовать сервис ЭЦП, предоставляющий Signature API.

Signature API является RESTful API, которое позволяет поставщику электронных услуг выработать ЭЦП на любом средстве ЭЦП пользователя путем выполнения последовательности HTTP-запросов к серверу ЭЦП.

Для использования Signature API поставщик электронных услуг должен провести аутентификацию пользователя - владельца средства ЭЦП - на OAuth-сервере, обслуживающем сервер ЭЦП. При аутентификации пользователя поставщик электронных услуг должен запросить разрешение на доступ к функциям выработки ЭЦП пользователя. Доступ к функциям Signature API будет предоставлен сервером ЭЦП при предъявлении билета доступа, выпущенного сервером авторизации.

5.2 Авторизация доступа к API

Доступ к функциям Signature API будет предоставлен сервером ЭЦП при предъявлении билета доступа, выпущенного сервером авторизации, обслуживающим сервер ЭЦП.

При получении билета доступа должно запрашиваться разрешение на доступ к функциям выработки ЭЦП пользователя: параметр `scope` должен включать значение `sign`.

Предъявление билета доступа осуществляется путем включения в HTTP-запрос заголовка `Authorization` согласно RFC 7235 и указанием в нем билета доступа с помощью схемы `Bearer` согласно RFC 6750, например:

```
Authorization: Bearer
533bacf01e11f55b536a565b57531ac114461ae8736d6506a3
```

5.3 Общий порядок обращения к API

Поставщик электронных услуг должен выполнить следующую последовательность шагов:

1. Опционально: вычислить значение функции хэширования от содержания электронного документа.

2. Выполнить запрос на инициирование операции выработки ЭЦП, передав в качестве параметров хэш-значение или содержание документа, идентификатор функции хэширования и URL, по которому будет ожидать получение результата выработки ЭЦП: выполняется однократно.

3. Выполнить запрос на начало операции выработки ЭЦП: в браузере пользователя должен быть открывает URL, полученный в ответ на запрос на инициирование операции выработки ЭЦП, для взаимодействия пользователя и сервера ЭЦП.

4. Ожидать получение сведений о завершении выработки ЭЦП: браузер пользователя будет перенаправлен на указанный URL.

5. Выполнить запрос на получение значения ЭЦП: выполняется однократно.

5.4 Запрос на инициирование операции выработки ЭЦП

Для инициирования операции выработки ЭЦП информационная система должна отправить POST-запрос на URL:

```
https://usd.nces.by/sign/v1
```

Примечание: SIGN_REQUEST_URL только по TLS.СТБ.

5.4.1 Отправка хэш-значения документа

Значения параметров должны быть переданы согласно спецификации POST-запроса, в закодированном виде согласно требованиям формата application/x-www-form-urlencoded.

Параметры запроса:

- `hash` – значение функции хэширования от содержания электронного документа, обязательный параметр; строка с шестнадцатеричным представлением хэш-значения.
- `hashAlgOid` – объектный идентификатор (OBJECT IDENTIFIER) функции хэширования, обязательный параметр; строка с текстовым представлением объектного идентификатора.
- `eventId` – идентификатор операции выработки ЭЦП, необязательный параметр; строка из не более 6 цифр.
- `returnUrl` – URL, по которому поставщик электронных услуг будет ожидать получение результата выработки ЭЦП, обязательный параметр; URL может включать заполнители `{id}` для идентификатора операции и `{hash}` для значения функции хэширования, вместо которых будут подставлены значения из данной операции. Если в адресе отсутствуют заполнители, то оба параметра будут добавлены в параметры запроса.

Примеры использования заполнителей:

```
returnUrl=http://test.org/{id}/{hash} ->
    http://test.org/9007199254740991/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
returnUrl=http://test.org/sign/{hash} ->
    http://test.org/sign/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAA
returnUrl=http://test.org/sign/{id} ->
    http://test.org/sign/9007199254740991
returnUrl=http://test.org?myid=10 ->
    http://test.org?myid=10&id=9007199254740991&hash=AAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
returnUrl=http://test.org ->
    http://test.org?id=9007199254740991&hash=AAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
returnUrl=http://test.org# ->
    http://test.org#id=9007199254740991&hash=AAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

5.4.2 Отправка содержания документа

Значения параметров должны быть переданы согласно спецификации POST-запроса в закодированном виде согласно требованиям формата multipart/form-data.

Параметры запроса:

- `file` – файл с данными для подписи, обязательный параметр.
- `hashAlgOid` – объектный идентификатор (OBJECT IDENTIFIER) функции хэширования, обязательный параметр; строка с текстовым представлением объектного идентификатора.
- `eventId` – идентификатор операции выработки ЭЦП, необязательный параметр; строка из не более 6 цифр.
- `returnUrl` – URL, по которому поставщик электронных услуг будет ожидать получение результата выработки ЭЦП, обязательный параметр; URL может включать заполнители `{id}` для идентификатора операции и `{hash}` для значения функции хэширования, вместо которых будут подставлены значения из данной операции. Если в адресе отсутствуют заполнители, то оба параметра будут добавлены в параметры запроса.

Примеры использования заполнителей:

```
returnUrl=http://test.org/{id}/{hash} ->
    http://test.org/9007199254740991/AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
returnUrl=http://test.org/sign/{hash} ->
    http://test.org/sign/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAA
```

```
returnUrl=http://test.org/sign/{id} ->
    http://test.org/sign/9007199254740991
returnUrl=http://test.org?myid=10 ->
    http://test.org?myid=10&id=9007199254740991&hash=AAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
returnUrl=http://test.org ->
    http://test.org?id=9007199254740991&hash=AAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
returnUrl=http://test.org# ->
    http://test.org#id=9007199254740991&hash=AAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

5.4.3 Возвращаемый результат

После завершения выработки ЭЦП браузер пользователя будет перенаправлен на URL `returnUrl` с GET запросом, указывая, что операция выработки ЭЦП завершена.

Возвращаемое значение является JSON-структурой и включается в HTTP-сообщение в закодированном согласно требованиям application/json виде.

Могут возвращаться следующие виды HTTP сообщений:

- HTTP сообщение с кодом 201: успешное инициирование операции выработки ЭЦП
    - Заголовки:

      ```
      Location: [значение URL для запроса для получения
      сведений о ходе выработки ЭЦП]
      ```

    - Содержимое: JSON-структура с параметрами операции выработки ЭЦП:
        - `id` – идентификатор операции выработки ЭЦП;
        - `progressUrl` – URL, который поставщик электронных услуг должен открыть в браузере пользователя для начала операции выработки ЭЦП. **Примечание**: перенаправление браузера пользователя на полученный адрес лучше осуществлять средствами сервера со статусом HTTP ответа 302.

Например, после получения указанного ниже HTTP-сообщения, поставщик электронных услуг должен перенаправить браузер пользователя (в новом окне или popup-окне) на URL `https://usd.nces.by/api/sign/progress/9007199254740991` для взаимодействия пользователя и сервера ЭЦП и использовать URL`https://usd.nces.by/api/sign/v1/9007199254740991` для выполнения запросов на получение сведений о ходе выработки ЭЦП.

```
HTTP/1.1 201 Created
Content-type: application/json
Location: https://usd.nces.by/api/sign/v1/9007199254740991

{
   "id" : 9007199254740991,
   "progressUrl" :
"https://usd.nces.by/api/sign/progress/9007199254740991"
}
```

- HTTP сообщение с кодом 400: в запросе отсутствуют обязательные параметры, имеют неверный формат или неверные значения
  - Содержимое: JSON-структура с кодом ошибки.
  - Пример HTTP ответа в случае использования билета доступа, запрошенного без указания `scope=sign`:

```
HTTP/1.1 400 Bad Request
Content-type: application/json
{
   "error" : "invalid_request"
}
```

- HTTP сообщение с кодом 401: необходима авторизация либо переданный билет не действителен
  - Заголовки:

```
WWW-Authenticate: Bearer realm="api", error="invalid_token"
```

  - Содержимое: JSON-структура с кодом ошибки.

- Пример HTTP сообщения в случае использования неверное билета доступа:

```
HTTP/1.1 401 Unauthorized
Content-type: application/json
{
   "error" : "invalid_token"
}
```

- HTTP сообщение с кодом 403: билет доступа не включает права доступа к запрашиваемому API
  - Заголовки:

```
WWW-Authenticate: Bearer realm="api",
error="insufficient_scope", scope="sign"
```

  - Содержимое: JSON-структура с кодом ошибки.
  - Пример HTTP сообщения в случае использования неверное билета доступа:

```
HTTP/1.1 403 Forbidden
Content-type: application/json
{
   "error" : "insufficient_scope"
}
```

- HTTP сообщение с кодом 500: внутренняя ошибка сервера при обработке запроса. Возникает в случае невозможности проверить выработанную ЭЦП из-за отсутствия действующего списка отозванных сертификатов, доступа к OCSP-сервису, доверия к корневым сертификатам и т.п.
  - Содержимое: JSON-структура с кодом ошибки.
  - Пример HTTP ответа:

```
HTTP/1.1 500 Internal Server Error
Content-type: application/json
{
   "error" : "server_error"
}
```

- HTTP сообщение с кодом 503: сервер временно не доступен.
  - Содержимое: нет.

5.5 Запрос на начало операции выработки ЭЦП

URL `progressUrl`, возвращенный в HTTP-сообщении в результате успешного выполнения запроса на инициирование выработки ЭЦП, должен быть открыт в браузере пользователя для прямого взаимодействия пользователя и сервера ЭЦП.

Примечание: перенаправление браузера пользователя на полученный адрес лучше осуществлять средствами сервера со статусом HTTP ответа 302.

В окне будет отображен пользовательский интерфейс сервера ЭЦП с запросом подтверждения операции выработки ЭЦП с идентификатором `eventId`, указанным поставщиком электронных услуг в запросе на инициирование выработки ЭЦП.

Поставщик электронных услуг должен отображать пользователю идентификатор `eventId`. Идентификатор `eventId` будет отображен владельцу SIM на мобильном устройстве. Для контроля за выполняемыми операциями владелец SIM должен сравнивать значения идентификаторов сеанса работы `eventId`, отображаемые поставщиком электронных услуг, сервером ЭЦП и SIM.

5.6 Запрос на получение результата выработки ЭЦП

Для получения статуса выработки ЭЦП информационная система должна отправить GET-запрос на URL:

```
https://usd.nces.by/sign/v1/{id}
```

Примечание: SIGN_REQUEST_URL только по TLS.СТБ.

Вместо `{id}` подставляется значение `id`, возвращенное в HTTP-сообщении в результате успешного выполнения запроса на инициирование выработки ЭЦП.

Вместо самостоятельного формирования URI может использоваться URL, указанный в заголовке Location HTTP-сообщения, которое было возвращено в результате успешного выполнения запроса на инициирование выработки ЭЦП.

Параметры запроса: нет.

Возвращаемое значение является JSON-структурой и включается в HTTP-сообщение в закодированном согласно требованиям application/json виде.

Могут возвращаться следующие виды HTTP сообщений:

- HTTP сообщение с кодом 200: сведения получены
  - Содержимое: JSON-структура со сведениями о ходе выработки ЭЦП:
    - `status` – текстовый идентификатор сведений о ходе выработки ЭЦП;
    - `response` – JSON-структура со значением ЭЦП (base64-представление CMSv3 подписанного сообщения, включающего сертификат ЭЦП подписанта);
  - Пример HTTP ответа в случае ожидания выработки ЭЦП:

```
HTTP/1.1 200 OK
Content-type: application/json
{
   "status" : "waiting"
}
```

  - Пример HTTP ответа в случае завершения выработки ЭЦП:

```
HTTP/1.1 200 OK
Content-type: application/json
{
   "status" : "success",
   "response" : {
      "signature" :
"MIIFUgYJKoZIhvcNAQcCoIIFQzCCBT8CAQMxEDAOBgoqhiQCAQEBAQI
BBQAwCwYJKoZIhvcNA....."
   }
}
```

- HTTP сообщение с кодом 400: аналогично запросу на инициирование выработки ЭЦП.
- HTTP сообщение с кодом 401: аналогично запросу на инициирование выработки ЭЦП.
- HTTP сообщение с кодом 403: аналогично запросу на инициирование выработки ЭЦП.
- HTTP сообщение с кодом 404: запрос с указанным идентификатором не найден
- Содержимое: нет.
- HTTP сообщение с кодом 500: аналогично запросу на инициирование выработки ЭЦП.
- HTTP сообщение с кодом 503: аналогично запросу на инициирование выработки ЭЦП.

5.7 Запрос на отмену операции выработки ЭЦП

Для отмены операции выработки ЭЦП информационная система должна отправить DELETE-запрос на URL:

```
https://usd.nces.by/sign/v1/{id}
```

Примечание: SIGN_REQUEST_URL только по TLS.СТБ.

Вместо `{id}` подставляется значение `id`, возвращенное в HTTP-сообщении в результате успешного выполнения запроса на инициирование выработки ЭЦП.

Параметры запроса: нет.

Возвращаемое значение является JSON-структурой и включается в HTTP-сообщение в закодированном согласно требованиям application/json виде.

Могут возвращаться следующие виды HTTP сообщений:

- HTTP сообщение с кодом 204: операция отменена
  - Содержимое: нет.
  - Пример HTTP ответа в случае успешной отмены операции выработки ЭЦП:

```
HTTP/1.1 204 No Content
```

- HTTP сообщение с кодом 401: аналогично запросу на инициирование выработки ЭЦП.
- HTTP сообщение с кодом 403: аналогично запросу на инициирование выработки ЭЦП.
- HTTP сообщение с кодом 404: запрос с указанным идентификатором не найден
- Содержимое: нет.
- HTTP сообщение с кодом 500: аналогично запросу на инициирование выработки ЭЦП.
- HTTP сообщение с кодом 503: аналогично запросу на инициирование выработки ЭЦП.

5.8 Типы данных

5.8.1 Параметры операции выработки ЭЦП

Схема JSON-структуры с параметрами операции выработки ЭЦП:

```
{
   "type": "object"
   "$schema": "http://json-schema.org/draft-04/schema#",
   "properties": {
      "id": {
         "type": "integer",
         "desciption": "Числовой идентификатор операции выработки
ЭЦП"
      },
      "progressUrl": {
         "type": "string",
         "desciption": "URL, который поставщик электронных услуг
должен открыть в браузере пользователя для взаимодействия
пользователя и сервера ЭЦП"
      }
   },
   "required": [
      "id",
```

```
      "progressUrl"
  ],
}
```

5.8.2 Сведения о ходе выработки ЭЦП

Схема JSON-структуры со сведениями о ходе выработки ЭЦП:

```
{
   "type": "object"
   "$schema": "http://json-schema.org/draft-04/schema#",
   "properties": {
      "status": {
         "type": "string",
         "enum": ["waiting", "success", "cancelled", "timed_out"],
         "desciption": "Текстовый идентификатор сведений о ходе
выработки ЭЦП"
      }
      "response": {
         "type": "object"
         "properties": {
            "signature": {
               "type": "string",
               "desciption": "Base64-представление CMSv3
подписанного сообщения включающего сертификат ЭЦП подписанта"
            }
         },
         "required": [
            "signature"
         ],
      },
   },
   "required": [
      "status"
   ]
}
```

Текстовые идентификаторы сведений о ходе выработки ЭЦП:

- `waiting` – выполняется выработка ЭЦП, выработка не завершена;

- `success` – ЭЦП выработана и получена;
- `cancelled` – пользователь не одобрил операцию выработки ЭЦП;
- `timed_out` – за отведенное время значение ЭЦП от пользователя не получено.

5.9 Ошибки выработки ЭЦП

В ходе выработки ЭЦП могут возвращаться следующие идентификаторы ошибок:

- `server_error` – внутренняя ошибка сервера;
- `unauthorized` – билет доступа не предоставлен;
- `invalid_request` – параметры запроса переданы неверно (согласно RFC 6750);
- `invalid_token` – предоставлен неверный билет доступа (согласно RFC 6750);
- `insufficient_scope` – билет доступа не включает права доступа к запрашиваемому API (согласно RFC 6750).

Схема JSON-структуры с ошибкой:

```
{
   "type": "object"
   "$schema": "http://json-schema.org/draft-04/schema#",
   "properties": {
      "error": {
         "type": "string",
         "enum": [ "invalid_request",
                   "server_error",
                   "invalid_token",
                   "insufficient_scope",
                   "unauthorized"
         ],
         "desciption": "Текстовый идентификатор ошибки"
      },
      "error_description": {
         "type": "string",
```

```
         "desciption": "Краткое описание ошибки на английском языке
в кодировке ASCII"
      },
      "error_uri": {
         "type": "string",
         "desciption": "URL с подробным описанием ошибки"
      }
   },
   "required": [
      "error"
   ],
}
```