



Центр сертификатов доступа

Aladdin Enterprise Certificate Authority Certified Edition

Руководство администратора. Часть 6. Описание методов REST API
Центра регистрации Aladdin Enterprise Registration Authority

Издание	RU.АЛДЕ.03.01.020
Документ	RU.АЛДЕ.03.01.020 32 01-6
Версия	2.2.1
Листов	51
Дата	17.06.2025

Авторские права, товарные знаки, ограничения

Данный документ, включая подбор и расположение иллюстраций и материалов в нём, является объектом авторских прав и охраняется в соответствии с законодательством Российской Федерации. Обладателем исключительных авторских и имущественных прав является АО «Аладдин Р.Д.».

Использование этих материалов любым способом без письменного разрешения правообладателя запрещено и может повлечь ответственность, предусмотренную законодательством РФ. При перепечатке и использовании данных материалов либо любой их части ссылки на АО «Аладдин Р.Д.» обязательны.

Владельцем зарегистрированных товарных знаков "Аладдин", Aladdin, JaCarta, JMS, JAS, Secret Disk, SecurLogon, "Крипто БД", логотипов и правообладателем исключительных прав на их дизайн и использование, патентов на соответствующие продукты является АО «Аладдин Р.Д.».

Названия прочих технологий, продуктов, компаний, упоминающиеся в данном документе, могут являться товарными знаками своих законных владельцев.

Ограничение ответственности

Информация, приведённая в данном документе, предназначена исключительно для ознакомления и не является исчерпывающей. Состав продуктов, компонент, их функции, характеристики, версии, доступность и пр. могут быть изменены АО «Аладдин Р.Д.» без предварительного уведомления.

АО «Аладдин Р.Д.» не гарантирует ни отсутствия ошибок в данном документе, ни того, что описанное программное обеспечение (ПО) не содержит дефектов, будет работать в произвольно выбранных условиях и при этом удовлетворять всем требованиям, которые могут быть к нему предъявлены.

АО «Аладдин Р.Д.» не гарантирует работоспособность нелегально полученного программного обеспечения. Нелегальное использование программного обеспечения и документации на него преследуется по закону.

Все указанные данные о характеристиках продуктов основаны на международных или российских стандартах и результатах тестирования, полученных в независимых тестовых или сертификационных лабораториях, либо на принятых в компании методиках. В данном документе АО «Аладдин Р.Д.» не предоставляет никаких ни явных, ни подразумеваемых гарантий.

АО «Аладдин Р.Д.» НЕ НЕСЁТ ОТВЕТСТВЕННОСТИ (КАК В СИЛУ ДОГОВОРА, ГРАЖДАНСКОГО ПРАВОНАРУШЕНИЯ, ВКЛЮЧАЯ ХАЛАТНОСТЬ, ТАК И В ЛЮБОЙ ИНОЙ ФОРМЕ) ПЕРЕД ВАМИ ИЛИ ЛЮБОЙ ТРЕТЬЕЙ СТОРОНОЙ ЗА ЛЮБЫЕ ПОТЕРИ ИЛИ УБЫТКИ (ВКЛЮЧАЯ КОСВЕННЫЕ, ФАКТИЧЕСКИЕ ИЛИ ПОБОЧНЫЕ УБЫТКИ), ВКЛЮЧАЯ БЕЗ ОГРАНИЧЕНИЙ ЛЮБЫЕ ПОТЕРИ ИЛИ УБЫТКИ ПРИБЫЛЬНОСТИ БИЗНЕСА, ПОТЕРЮ ДОХОДНОСТИ ИЛИ РЕПУТАЦИИ, УТРАЧЕННУЮ ИЛИ ИСКАЖЁННУЮ ИНФОРМАЦИЮ ИЛИ ДОКУМЕНТАЦИЮ ВСЛЕДСТВИЕ ИСПОЛЬЗОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И/ИЛИ ЛЮБОГО КОМПОНЕНТА ОПИСАННОГО ПРОДУКТА, ДАЖЕ ЕСЛИ АО «Аладдин Р.Д.» БЫЛО ПИСЬМЕННО УВЕДОМЛЕНО О ВОЗМОЖНОСТИ ПОДОБНЫХ УБЫТКОВ.

Государственное регулирование и экспортный контроль

Описываемый в данном документе продукт (или продукты) может являться или содержать в себе средство криптографической защиты информации (СКЗИ), являющееся предметом экспортного контроля.

Вы соглашаетесь с тем, что продукт не будет поставляться, передаваться или экспортироваться в какую-либо страну, а также использоваться каким-либо противоречащим закону образом.

Вы гарантируете, что будете соблюдать накладываемые на экспорт и реэкспорт продукта ограничения.

Сведения, приведённые в данном документе, актуальны на дату его публикации.

© АО «Аладдин Р.Д.», 1995–2025. Все права защищены

Лицензионное соглашение

Пожалуйста, внимательно прочитайте данное лицензионное соглашение прежде, чем использовать содержимое данного комплекта и/или прежде, чем загружать или устанавливать программное обеспечение.

Все указания по использованию программного обеспечения, предоставляемые Закрытым акционерным обществом "Аладдин Р. Д." (или любым его дочерним предприятием – каждое из них упоминаемое как "компания"), подчиняются и будут подчиняться условиям, оговоренным в данном соглашении. Загружая данное программное обеспечение (как определено далее по тексту) и/или устанавливая данное программное обеспечение на Ваш компьютер и/или используя данное программное обеспечение иным способом, Вы принимаете данное соглашение и соглашаетесь с его условиями.

Если Вы не согласны с данным соглашением, не загружайте и/или не устанавливайте данное программное обеспечение и незамедлительно (не позднее 7 (семи) дней с даты ознакомления с настоящим текстом) верните этот продукт в АО «Аладдин Р.Д.», удалите данное программное обеспечение и все его части со своего компьютера и не используйте его никоим образом.

Настоящее лицензионное соглашение (далее "Соглашение") является договором, заключенным между Вами (физическим или юридическим лицом) – конечным пользователем (далее "Пользователь") – и АО «Аладдин Р.Д.» (далее "Компания") относительно передачи неисключительного права на использование настоящего программного обеспечения, являющегося интеллектуальной собственностью Компании.

Права и собственность

ДАННОЕ СОГЛАШЕНИЕ НЕ ЯВЛЯЕТСЯ СОГЛАШЕНИЕМ О ПРОДАЖЕ.

Программное обеспечение, включая все переработки, исправления, модификации, дополнения, обновления и/или усовершенствования к нему (далее по всему тексту и любой его части определяемое как Программное обеспечение или ПО), и связанная с ним документация предназначена НЕ ДЛЯ ПРОДАЖИ и является и остаётся исключительной собственностью Компании.

Все права на интеллектуальную собственность (включая, без ограничений, авторские права, коммерческую тайну, товарные знаки, и т.д.), подтверждённые или включённые в приложенные/взаимосвязанные/имеющие отношение к данному руководству, данные, содержащиеся в нём, а также все права на ПО являются и будут являться собственностью исключительно Компании.

Данное соглашение не передаёт Вам права на Программное обеспечение, а лишь предоставляет ограниченное право на использование, которое подлежит отмене согласно условиям данного Соглашения. Ничто в данном Соглашении не подтверждает отказ Компании от прав на интеллектуальную собственность по какому бы то ни было законодательству.

Лицензия

Компания настоящим предоставляет Вам, а Вы получаете индивидуальное, неисключительное и отзываемое ограниченное право на использование данного ПО только в форме исполняемого кода, как описано в прилагаемой к ПО технической/эксплуатационной документации, и только в соответствии с условиями данного

Соглашения:

Вы можете установить ПО и использовать его на компьютерах, расположенных в пределах Вашего предприятия, как описано в соответствующей технической/эксплуатационной документации ПО и в настоящем соглашении.

Вы можете добавить/присоединить Программное обеспечение к программам для мобильных устройств с единственной целью, описанной в данном Соглашении. Принимая условия настоящего соглашения, Вы соглашаетесь:

- не использовать, не модифицировать и не выдавать сублицензии на данное Программное обеспечение и любое другое ПО Компании, за исключением явных разрешений в данном Соглашении;
- не модифицировать, не демонтировать, не декомпилировать, не реконструировать, не видоизменять и не расширять данное Программное обеспечение и не пытаться раскрыть (получить) исходные коды данного Программного обеспечения;
- не помещать данное Программное обеспечение на сервер с возможностью доступа к нему третьих лиц через открытую сеть;
- не использовать какие бы то ни было резервные или архивные копии данного Программного обеспечения (или позволять кому-либо ещё использовать такие копии) с любой иной целью, кроме замены его оригинального экземпляра в случае его разрушения или наличия дефектов.

Требования к использованию

Программное обеспечение должно использоваться и обслуживаться строго в соответствии с описаниями и инструкциями Компании, приведёнными в данном и других документах Компании, в том числе на портале онлайн документации для разработчиков Компании (<http://developer.aladdin-rd.ru/>).

Использование ПО

Пользователь вправе:

- воспроизводить ПО путём записи его в память электронно-вычислительных машин Пользователя, ограниченное правом инсталляции, копирования и запуска программ для ЭВМ;
- встраивать ПО любым способом в продукты и решения Пользователя;
- распространять ПО любым способом исключительно в составе продуктов и решений Пользователя.

При использовании и распространении ПО Пользователь обязан руководствоваться действующим законодательством Российской Федерации и международным законодательством, учитывая ограничения и дополнительные требования, которые могут возникать в связи с экспортом шифровальных (криптографических) средств с территории Российской Федерации и импортом таких средств в другие страны. В частности, ограничения и дополнительные требования могут возникать при распространении ПО через магазины приложений, содержащие различные приложения для мобильных устройств.

Условия использования, изложенные в настоящем соглашении, действуют в отношении всего содержимого ПО, в частности в отношении:

- дизайна (графики, расположения элементов оформления и т.п.);
- всех иных элементов, в том числе изображений, фонограмм, текстов.

Получаемые Пользователем неисключительные имущественные права не включают права на передачу третьим лицам каких-либо прав на встраивание, воспроизведение, распространение и использование программ для ЭВМ не в составе продуктов и решений Пользователя.

Компания сохраняет за собой все исключительные права на ПО и входящие в него компоненты, включая права на предоставление неисключительных и исключительных прав третьим лицам.

Пользователь вправе осуществлять использование ПО в пределах, предусмотренных настоящим Соглашением, исключительно на территории Российской Федерации

Обслуживание и поддержка

Компания не несёт обязательств по предоставлению поддержки, обслуживания, модификации или выходу новых релизов ПО.

Ограниченная гарантия

Компания гарантирует, что программное обеспечение с момента приобретения его Вами в течение 12 (двенадцати) месяцев будет функционировать в полном соответствии с его технической/эксплуатационной документацией, при условии, что ПО будет использоваться на компьютерном аппаратном обеспечении и с операционной системой, для которой оно было разработано.

Отказ от гарантии

Компания не гарантирует, что программное обеспечение будет соответствовать Вашим желаниям и требованиям, или что его работа будет бесперебойной или безошибочной. В объёме, предусмотренном законодательством РФ, компания открыто отказывается от всех гарантий, не оговоренных здесь, от всех иных подразумеваемых гарантий. Ни один из дилеров, дистрибьюторов, продавцов, агентов или сотрудников компании не уполномочен производить модификации, расширения или дополнения к данной гарантии.

Если Вы произвели какие-либо модификации ПО или любой из его частей во время гарантийного периода, ПО подверглось повреждению, неосторожному или неправильному обращению, если Вы нарушили любое из условий настоящего Соглашения, то гарантия, упомянутая выше в разделе 5, будет немедленно прекращена.

Гарантия недействительна, если ПО используется в сочетании с иным аппаратным и/или программным обеспечением, отличным от описанных в технической/эксплуатационной документации, или используется на компьютере с любым установленным нелегальным программным обеспечением.

Ограничение возмещения

В случае нарушения гарантии, оговоренной выше, Компания может по собственному усмотрению:

- заменить ПО, если это не противоречит вышеупомянутому ограничению гарантии;
- возместить стоимость, выплаченную Вами за ПО.

Гарантийные требования должны быть выставлены в письменном виде в течение гарантийного периода, но не позднее 7 (семи) дней с момента обнаружения дефекта, и содержать в себе подтверждения, удовлетворяющие Компанию. Всё ПО (все экземпляры, имеющиеся у Вас) должно быть возвращено Компании и отправлено возвращающей стороной с оплаченной стоимостью перевозки и, при необходимости, страховки. Экземпляры ПО должны быть отправлены с копией платёжных документов и накладных.

Исключение косвенных убытков

Стороны признают, что Программное обеспечение не может быть полностью лишено ошибок. Компания не несёт ответственности (как в силу договора, гражданского правонарушения, включая халатность, так и в любой иной форме) перед Вами или любой третьей стороной за любые потери или убытки (включая косвенные, фактические, побочные или потенциальные убытки), включая, без ограничений, любые потери или убытки прибыльности бизнеса, потерю доходности или репутации, утраченную или искажённую информацию или документацию вследствие какого-либо использования данного программного обеспечения и/или любой компоненты данного ПО, даже если компания письменно уведомлена о возможности подобных убытков.

Ограничение ответственности

В случае если, несмотря на условия данного соглашения, компания признана ответственной за убытки на основании каких-либо дефектов или несоответствия программного обеспечения Вашим ожиданиям, полная ответственность за каждый экземпляр дефектного программного обеспечения не будет превышать суммы, выплаченной вами АО «Аладдин Р.Д.» за это ПО.

Прекращение действия соглашения

В случае невыполнения Вами условий данного Соглашения действие Вашей лицензии и настоящего Соглашения будет прекращено.

После прекращения действия данного Лицензионного соглашения:

- лицензия, предоставленная Вам данным Соглашением, прекращает своё действие, и Вы после её прекращения не сможете продолжать дальнейшее использование данного Программного обеспечения и других лицензионных Продуктов;
- вы немедленно вернёте в Компанию все экземпляры ПО и все копии такого и/или сотрёте/удалите любую информацию, содержащуюся в электронном виде.

Применимое законодательство

Данное Соглашение должно быть истолковано и определено в соответствии с законодательством Российской Федерации (за исключением конфликта применения правовых норм), и только российский суд уполномочен осуществлять правосудие в любых конфликтах и спорах, вытекающих из данного Соглашения. Невозможность для любой из сторон воспользоваться любым из прав, предоставленных ей по данному Соглашению, или принять меры против другой стороны в случае любого нарушения своих обязательств по Соглашению не должно рассматриваться как отказ этой стороны от последующего понуждения к признанию своих прав или совершению последующих действий в случае дальнейших нарушений.

Государственное регулирование и экспортный контроль

Вы соглашаетесь с тем, что ПО не будет Вами поставляться, передаваться или экспортироваться в какую-либо страну, а также использоваться каким-либо противоречащим закону и условиям настоящего соглашения образом. ПО является предметом дополнительного экспортного контроля, относящегося к Вам или Вашей юрисдикции. Вы гарантируете, что будете соблюдать накладываемые ограничения на экспорт и резэкспорт ПО.

Разное

Настоящее Соглашение представляет собой полное соглашение, относящееся к данной лицензии, и может быть изменено только посредством письменного соглашения, подписанного обеими сторонами. Если выполнение какого-либо условия настоящего Соглашения представляется невозможным, такое условие будет скорректировано только в пределах, обеспечивающих возможность выполнения данного условия.

Я ПРОЧИТАЛ И ПОНЯЛ НАСТОЯЩЕЕ ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ И СОГЛАСЕН ВЫПОЛНЯТЬ ВСЕ ЕГО УСЛОВИЯ.

Я ПРИНИМАЮ ДАННОЕ ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ ЦЕЛИКОМ.

ЕСЛИ Я НЕ ПРИНИМАЮ ЭТО ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ ИЛИ ХОТЯ БЫ ОДИН ИЗ ЕГО ПУНКТОВ, ТО ДАННОЕ ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ НЕ ВСТУПАЕТ В СИЛУ, И Я ОБЯЗУЮСЬ НЕ УСТАНОВЛИВАТЬ И НЕ ИСПОЛЬЗОВАТЬ ДАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Содержание

1 Методы аутентификации	6
1.1 Метод аутентификации по сертификату	6
1.1.1 Пример использования метода для аутентификации по сертификату доступа:.....	6
1.1.2 Пример ответа на запрос к данному методу:.....	6
1.2 Метод аутентификации по Kerberos-ticket	7
1.3 Метод аутентификации по логину и паролю.....	7
1.4 Метод обновления маркера доступа.....	8
1.4.1 Пример использования метода:.....	9
1.4.2 Пример ответа на запрос к данному методу:.....	10
2 Методы работы с заявками	11
2.1 Метод создания новой заявки на выпуск сертификата с закрытым ключом (PKCS#12).....	11
2.2 Метод создания новой заявки на основании запроса PKCS#10.....	12
2.3 Метод получения заявки по идентификатору	14
2.4 Метод получения заявки по внешнему ключу.....	17
2.5 Метод поиска заявок.....	18
2.6 Метод получения данных о выпущенном сертификате заявки по ее идентификатору.....	20
3 Методы экспорта файлов.....	23
3.1 Метод получения запроса на сертификат по идентификатору заявки	23
3.2 Метод получения сертификата по идентификатору заявки	23
3.3 Метод получения цепочки сертификатов по идентификатору заявки	24
3.4 Метод получения контейнера pkcs12 по идентификатору заявки.....	24
3.5 Метод получения сертификата издателя сертификата по заявке	25
3.6 Метод получения цепочки сертификатов издателя сертификата по заявке.....	25
3.7 Метод получения CRL издателя сертификата по заявке	26
4 Методы работы с субъектами.....	27
4.1 Метод поиска субъектов.....	27
4.2 Метод получения субъекта по идентификатору	28
5 Методы работы с шаблонами.....	30
5.1 Метод поиска шаблонов.....	30
5.2 Метод получения шаблона по идентификатору	31
6 Методы работы с сертификатами	34
6.1 Метод отзыва сертификата по идентификатору заявки.....	34
7 Диаграмма последовательности получения сертификата по запросу PKCS#10.....	35
7.1 Диаграмма последовательности получения сертификата по запросу PKCS#10.....	35
8 Спецификация OpenAPI	39
Обозначения и сокращения.....	50

1 МЕТОДЫ АУТЕНТИФИКАЦИИ

1.1 Метод аутентификации по сертификату (устаревший)

POST API – Аутентификация с помощью сертификата	
Данный метод должен позволять выполнять аутентификацию в eCA-RA по сертификату доступа администратора или оператора подключенного eCA-CA.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv2%5D%20Контроллер%3A%20Авторизации/signInByX509	
URL – /x509-provider-service/api/v2/public/auth/sign-in/x509	
Query	
-	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

1.1.1 Пример использования метода для аутентификации по сертификату доступа:

```
sudo curl --request POST -k --cert cert.crt.pem --cert-type PEM --key test.key.pem --key-type PEM 'https://localhost/x509-provider-service/api/v2/public/auth/sign-in/x509'
```

В примере указаны:

- cert.crt.pem - сертификат;
- test.key.pem - закрытый ключ.

Ключ и сертификат можно получить из контейнера PKCS#12 администратора или оператора, используя утилиту openssl:

```
openssl pkcs12 -in INITIAL_ADMIN.p12 -out test.key.pem -nocerts -nodes
openssl pkcs12 -in INITIAL_ADMIN.p12 -out cert.crt.pem -clcerts -nokeys
```

1.1.2 Пример ответа на запрос к данному методу:

```
{
  "status": 200,
  "data": {
    "token": "eyJhbGciOiJSUzUxMiJ9.eyJzZXNzaW9uSWQiOiJiYzZmMDQ5ZC1hM2EwLTQ4ZjQtYjE0Ny02YmYzNmRjYjZhNGIiLCJpYXQiOiJlMDE5ODU2MjYsImV4cCI6MTcwMTk4NTgwNn0uYX7pbvPHtP8tQ0vg9upBR7AlYXbgwr8XX1tldcgcySzehA6qi2mJGE-MlMf0iNeAoY0L_QJ7YVsB_080yTwcw2E6ipsTRI1rRbq1VHKw4xTNcVh2UVCXXTxz4WaYw3uSkNJfPPLgs7wyxyAlu3Is4d7JuiCSgONoj-rArjam_ccTuEHJnHp7PTnrDLOUNm8elu-VkP9qvQPqLb0P_K7-XLp3lQrWOZWMPM4imldiA70fBD4QeNiruQjYHK4ENbqCARIMiPnV0GFTz7JmNLLP1liFrnOgHhUUhp6JylKOTKxddLgUjTU03BmucUwAl_ccThZZPzpwNLd6Cx67X00g",
    "refresh": "8b73de98-721d-4d0b-98f7-f19745272f0c"
  }
}
```

В результате успешной аутентификации сервер возвращает cookies-файл с маркером доступа и токеном обновления. Для дальнейшей аутентификации используется данный маркер доступа (см. примеры использования методов далее).

Маркер доступа необходимо передать в http-header «token», либо приложить к запросу полученный cookies-файл с маркером доступа (http-header «cookie»).

1.2 Метод аутентификации по Kerberos-ticket (устаревший)

POST API – Аутентификация по Kerberos-ticket	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv2%5D%20Контроллер%3A%20Авторизации/kerberosSignIn_1	
URL – /kerberos-provider-service/api/v2/public/auth/sign-in/kerberos	
Query	
-	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

В результате авторизации сервер возвращает cookies-файл с маркером доступа и токеном обновления. Для дальнейшей аутентификации используется данный маркер доступа.

Маркер доступа необходимо передать в http-header «token», либо приложить к запросу полученный cookies-файл с маркером доступа (http-header «cookie»).

1.3 Метод аутентификации по логину и паролю (устаревший)

POST API – Аутентификация по логину и паролю	
URL – /kerberos-provider-service/api/v2/public/auth/sign-in/ldap	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv2%5D%20Контроллер%3A%20Авторизации/ldapSignIn_1	
Query	
-	
Request	
{	
username (string),	Имя пользователя
password (string)	Пароль пользователя
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

В результате авторизации сервер возвращает cookies-файл с маркером доступа и токеном обновления. Для дальнейшей аутентификации используется данный маркер доступа.

Маркер доступа необходимо передать в http-header «token», либо приложить к запросу полученный cookies-файл с маркером доступа (http-header «cookie»).

1.4 Метод аутентификации по сертификату

POST API – Аутентификация с помощью сертификата	
Данный метод должен позволять выполнять аутентификацию в Центре регистрации Aladdin eRA по сертификату доступа администратора или оператора подключенного Центра сертификации Aladdin eCA.	
URL – /security-service/api/v2/public/auth/sign-in/x509	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Авторизации/signInByX509_1	
Query	
-	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

В результате успешной аутентификации сервер возвращает cookies-файл с маркером доступа и токеном обновления. Для дальнейшей аутентификации используется данный маркер доступа (см. примеры использования методов далее). Маркер доступа необходимо передать в http-header «token», либо приложить к запросу полученный cookies-файл с маркером доступа (http-header «cookie»)

1.5 Метод аутентификации по Kerberos-ticket

POST API – Аутентификация по Kerberos-ticket	
URL – /security-service/api/v2/public/auth/sign-in/ldap	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Авторизации/kerberosSignIn	
Query	
-	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

В результате успешной аутентификации сервер возвращает cookies-файл с маркером доступа и токеном обновления. Для дальнейшей аутентификации используется данный маркер доступа (см. примеры использования методов далее). Маркер доступа необходимо передать в http-header «token», либо приложить к запросу полученный cookies-файл с маркером доступа (http-header «cookie»)

1.6 Метод аутентификации по логину и паролю

POST API – Аутентификация по логину и паролю	
Данный метод должен позволять выполнять аутентификацию в Центре регистрации Aladdin eRA по сертификату доступа администратора или оператора подключенного Центра сертификации Aladdin eCA.	
URL – /security-service/api/v2/public/auth/sign-in/kerberos	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Авторизации/ldapSignIn	
Query	
-	
Request	
{	
username (string),	Имя пользователя
password (string)	Пароль пользователя
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

В результате успешной аутентификации сервер возвращает cookies-файл с маркером доступа и токеном обновления. Для дальнейшей аутентификации используется данный маркер доступа (см. примеры использования методов далее). Маркер доступа необходимо передать в http-header «token», либо приложить к запросу полученный cookies-файл с маркером доступа (http-header «cookie»)

1.7 Метод обновления маркера доступа

PUT API – Обновления токена доступа	
Метод должен быть доступен администратору и оператору	
URL – /security-service/api/v2/public/auth/refresh-token	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Авторизации/refreshToken	
Query	
-	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

1.7.1 Пример использования метода:

```
curl --location -k --request POST 'https://172.17.152.213/security-service/api/v2/public/auth/refresh-token' --header 'Cookie:
```

```
token=eyJhbGciOiJSUzUxMiJ9.eyJzZXNzaW9uSWQiOiI2YTZhNmM4Yi0zYWQyLTQwYjMtYWI5NS05NDgxNmZlYjBkZGQiLCJpYXQiOiJlMjE3MDIwMjEzNzEsImV4cCI6MTcwMjAyMTU1M30.CUO3CaTt7JsIAg8LYMlURx0tetbsPw5iHT33i14JwSAos8DmNFSelylnfEHNlb2U8Q0G1KvJlhm84FqWR45RgBuvvsQkgzzR5aDDiVzLhhpkbPGqAUbILmFdiqLmDokyS0SyybUO4-SSfueV8lzo5aF6ly80c2cWpwRrRQgNhBDdc1LgjACmdD29urmBqlQsFV6AFRdlN6l3EZkP2ragZInQQ1lM9D4e5d3mlugCP7pJ2bp39pOVLLWC1czusPfs9tDoIyigkZBkSTi7xPYAWUdghzBfc--eK1kepoWilV5r4hln4bT6grZ7Yyi-J5JLVycuGOzkle3ByIcBExmFwtQ", "refresh": "3071f779-f4f7-47db-91a4-13acfc50005" }
```

1.7.2 Пример ответа на запрос к данному методу:

```
{ "status": 200, "data": { "token": "eyJhbGciOiJSUzUxMiJ9.eyJzZXNzaW9uSWQiOiI2YTZhNmM4Yi0zYWQyLTQwYjMtYWI5NS05NDgxNmZlYjBkZGQiLCJpYXQiOiJlMjE3MDIwMjEzNzEsImV4cCI6MTcwMjAyMTU1M30.CUO3CaTt7JsIAg8LYMlURx0tetbsPw5iHT33i14JwSAos8DmNFSelylnfEHNlb2U8Q0G1KvJlhm84FqWR45RgBuvvsQkgzzR5aDDiVzLhhpkbPGqAUbILmFdiqLmDokyS0SyybUO4-SSfueV8lzo5aF6ly80c2cWpwRrRQgNhBDdc1LgjACmdD29urmBqlQsFV6AFRdlN6l3EZkP2ragZInQQ1lM9D4e5d3mlugCP7pJ2bp39pOVLLWC1czusPfs9tDoIyigkZBkSTi7xPYAWUdghzBfc--eK1kepoWilV5r4hln4bT6grZ7Yyi-J5JLVycuGOzkle3ByIcBExmFwtQ", "refresh": "3071f779-f4f7-47db-91a4-13acfc50005" } }
```

2 МЕТОДЫ РАБОТЫ С ЗАЯВКАМИ

2.1 Метод создания новой заявки на выпуск сертификата с закрытым ключом (PKCS#12)

POST API – Создание новой заявки на выпуск сертификата в контейнере PKCS#12	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для УЗ с ролью «Администратор» или «Оператор» с использованием данного метода должно быть доступно создание заявок для субъектов ресурсных систем из eCA-CA, к которому подключен eCA-RA.	
URL – /tasks-service/api/v2/public/tasks/pkcs12	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/registerPkcs12	
Query	
{	
certificateAuthorityId (uuid), [опционально]	ID Центра сертификации, на котором должен быть выпущен сертификат по заявке. Если для данного параметра не указано значение, выпуск будет осуществляться на Центре сертификации подключенного eCA-CA, который определен в используемом шаблоне (поле «templateId»). Если в используемом шаблоне не определен конкретный центр сертификации (указано значение «Любой»), выпуск будет осуществляться на активном на момент создания заявки Центре сертификации подключенного eCA-CA.
subjectId (uuid), [опционально]	ID субъекта¹. Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь».
templateId (uuid), [обязательно]	ID шаблона сертификата ² .
foreignKey (string) [опционально]	Внешний ключ заявки. Внешний ключ должен быть уникальным.³
}	
Request	
{	
subjectDN: { (enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[] },	Поля разделенного имени субъекта, которое должно попасть в сертификат. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
subjectAltName: { (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, }	Поля альтернативного имени субъекта, которое должно попасть в сертификат. В формате key-value. Где key – один из

¹ Субъекты eCA-CA и eCA-RA имеют сквозную идентификацию.

² Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

³ При указании в поле «foreignKey» значения, аналогичного значению в поле «foreignKey» у уже существующей в базе данных программы заявки, при использовании метода будет возвращена ошибка с кодом 400.

KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[] },	перечисленных в enum параметров, а value – значение параметра
keyBits (integer),	Длина ключа
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключевой пары сертификата
password (string)	Пароль контейнера
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ⁴
type (enum: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (enum: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ⁵
certificateId(uuid),	Идентификатор сертификата. Заполняется по итогам успешной отработки заявки.
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string)	Внешний ключ заявки
}	

2.2 Метод создания новой заявки на основании запроса PKCS#10

2.2.1 Метод создания новой заявки на основании запроса PKCS#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

POST API – Создание новой заявки на основании запроса PKCS#10	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для УЗ с ролью «Администратор» или «Оператор» с использованием данного метода должно быть доступно создание заявок для субъектов ресурсных систем из eCA-CA, к которому подключен eCA-RA.	
URL – /tasks-service/api/v2/public/tasks/pkcs10	
Swagger: external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20задачи/registerPkcs10AsFileDescription	
Query	
{	

⁴ Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

⁵ Субъекты eCA-CA и eCA-RA имеют сквозную идентификацию.

<code>certificateAuthorityId (uuid), [опционально]</code>	ID Центра сертификации, на котором должен быть выпущен сертификат по заявке. Если для данного параметра не указано значение, выпуск будет осуществляться на Центре сертификации подключенного eCA-CA, который определен в используемом шаблоне (поле «templateId»). Если в используемом шаблоне не определен конкретный центр сертификации (указано значение «Любой»), выпуск будет осуществляться на активном на момент создания заявки Центре сертификации подключенного eCA-CA.
<code>subjectId (uuid), [опционально]</code>	ID субъекта ⁶ . Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь». Может быть указан при использовании метода учетной записью с ролью «Администратор». ⁷
<code>templateId (uuid), [обязательно]</code>	ID шаблона сертификата ⁸ .
<code>foreignKey (string) [опционально]</code>	Внешний ключ заявки. Внешний ключ должен быть уникальным. ⁹
<code>}</code>	
Request	
<code>{</code>	
<code>request (MultipartFile),</code>	PKCS#10 файл запроса на сертификат
<code>subjectName: { (enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[] }, [опционально]</code>	Поля отличительного имени субъекта, которое должно попасть в сертификат. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра. Значения полей, указанные в subjectName, переопределяют значения соответствующих полей SDN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.
<code>subjectAltName: { (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, }</code>	Поля альтернативного имени субъекта, которое должно попасть в сертификат. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра.

⁶ Субъекты eCA-CA и eCA-RA имеют сквозную идентификацию.

⁷ Если при создании администратором заявки на сертификат на основании запроса во входных параметрах метода не указан subjectId, ПО eCA-RA должно на основании указанного в запросе значения Common name автоматически определять (или создавать) получателя сертификата по заявке в соответствии со следующими правилами:

- если в списке субъектов присутствует один субъект, имеющий Common name аналогичный указанному в запросе, получателем сертификата по заявке должен считаться данный субъект;
- если в списке субъектов присутствует несколько субъектов, имеющих Common name аналогичный указанному в запросе, получателем сертификата по заявке из данного перечня субъектов должен считаться субъект с самой поздней датой последнего редактирования;
- если в списке субъектов отсутствуют субъекты, имеющие Common name аналогичный указанному в запросе, eCA-RA должно создавать в eCA-CA локальный субъект на основании запроса.

⁸ Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

⁹ При указании в поле «foreignKey» значения, аналогичного значению в поле «foreignKey» у уже существующей в базе данных программы заявки, при использовании метода будет возвращена ошибка с кодом 400.

<pre> XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[] } </pre>	Необязательный параметр, задается в случае невозможности поместить данные в PKCS10. Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ¹⁰
type (enum: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (enum: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ¹¹
certificateId (string),	Идентификатор сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string)	Внешний ключ заявки
}	

2.2.2 Метод создания новой заявки на основании запроса PKCS#10 (формат запроса на сертификат из входных параметров – application/json)

POST API – Создание новой заявки на основании запроса PKCS#10	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для УЗ с ролью «Администратор» или «Оператор» с использованием данного метода должно быть доступно создание заявок для субъектов ресурсных систем из eCA-CA, к которому подключен eCA-RA.	
URL – /tasks-service/api/v2/public/tasks/pkcs10	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/registerPkcs10AsFileDescription	
Query	
{	

¹⁰ Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

¹¹ Субъекты eCA-CA и eCA-RA имеют сквозную идентификацию.

<code>certificateAuthorityId (uuid), [опционально]</code>	ID Центра сертификации, на котором должен быть выпущен сертификат по заявке. Если для данного параметра не указано значение, выпуск будет осуществляться на Центре сертификации подключенного eCA-CA, который определен в используемом шаблоне (поле « <code>templateId</code> »). Если в используемом шаблоне не определен конкретный центр сертификации (указано значение «Любой»), выпуск будет осуществляться на активном на момент создания заявки Центре сертификации подключенного eCA-CA.
<code>subjectId (uuid), [опционально]</code>	ID субъекта ¹² . Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь». Может быть указан при использовании метода учетной записью с ролью «Администратор». ¹³
<code>templateId (uuid), [обязательно]</code>	ID шаблона сертификата ¹⁴ .
<code>foreignKey (string) [опционально]</code>	Внешний ключ заявки. Внешний ключ должен быть уникальным ¹⁵
<code>}</code>	
Request	
<code>{</code>	
<code>request: {</code>	Запрос на сертификат
<code>contentType (string) [опционально],</code>	Тип загружаемого файла (HTTP MediaType) - application/octet-stream)
<code>fileName (string) [опционально],</code>	Имя загружаемого файла
<code>data (string:binary)</code>	Содержимое PEM файла запроса на сертификат (массив байт в Base64)
<code>},</code>	

¹² Субъекты eCA-CA и eCA-RA имеют сквозную идентификацию.

¹³ Если при создании администратором заявки на сертификат на основании запроса во входных параметрах метода не указан `subjectId`, ПО eCA-RA должно на основании указанного в запросе значения `Common name` автоматически определять (или создавать) получателя сертификата по заявке в соответствии со следующими правилами:

- если в списке субъектов присутствует один субъект, имеющий `Common name` аналогичный указанному в запросе, получателем сертификата по заявке должен считаться данный субъект;
- если в списке субъектов присутствует несколько субъектов, имеющих `Common name` аналогичный указанному в запросе, получателем сертификата по заявке из данного перечня субъектов должен считаться субъект с самой поздней датой последнего редактирования;
- если в списке субъектов отсутствуют субъекты, имеющие `Common name` аналогичный указанному в запросе, eCA-RA должно создавать в eCA-CA локальный субъект на основании запроса.

¹⁴ Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

¹⁵ При указании в поле «`foreignKey`» значения, аналогичного значению в поле «`foreignKey`» у уже существующей в базе данных программы заявки, при использовании метода будет возвращена ошибка с кодом 400.

<pre>subjectName: { (enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[] }, [опционально]</pre>	Поля отличительного имени субъекта, которое должно попасть в сертификат. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра. Значения полей, указанные в subjectName, переопределяют значения соответствующих полей SDN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.
<pre>subjectAltName: { (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[] }</pre>	Поля альтернативного имени субъекта, которое должно попасть в сертификат. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра. Необязательный параметр, задается в случае невозможности поместить данные в PKCS10. Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.
}	
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ¹⁶
type (enum: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (enum: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ¹⁷
certificateId (string),	Идентификатор сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string)	Внешний ключ заявки
}	

2.3 Метод получения заявки по идентификатору

GET API – Получение заявки по идентификатору
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно получение только тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ.

¹⁶ Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

¹⁷ Субъекты eCA-CA и eCA-RA имеют сквозную идентификацию.

Для УЗ с ролью «Оператор» с использованием данного метода должно быть доступно получение заявок, созданных данной УЗ, а также заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного еСА-СА. Для УЗ с ролью «Администратор» с использованием данного метода должно быть доступно получение любых заявок.	
URL – /tasks-service/api/v2/public/tasks/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/findById	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ¹⁸
certificateAuthorityId (uuid),	Идентификатор сертификата издателя
type (string: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (string: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ¹⁹
certificateId (uuid),	ID сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string),	Внешний ключ заявки
templateName (string),	Имя шаблона сертификата
msUpn (string)	UPN субъекта
}	

2.4 Метод получения заявки по внешнему ключу

GET API – Получение заявки по внешнему ключу
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно получение только тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ. Для УЗ с ролью «Оператор» с использованием данного метода должно быть доступно получение заявок, созданных данной УЗ, а также заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного еСА-СА.

¹⁸ Шаблоны еСА-СА и еСА-РА имеют сквозную идентификацию.

¹⁹ Субъекты еСА-СА и еСА-РА имеют сквозную идентификацию.

Для УЗ с ролью «Администратор» с использованием данного метода должно быть доступно получение любых заявок.	
URL – /tasks-service/api/v2/public/tasks/foreign-key/{foreignKey}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/findByForeignKey	
Query	
{	
foreignKey (string)	Внешний ключ заявки
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ²⁰
certificateAuthorityId (uuid),	Идентификатор сертификата издателя
type (string: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (string: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ²¹
certificateId (uuid),	ID сертификата. Заполняется по итогам успешной обработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string),	Внешний ключ заявки
templateName (string),	Имя шаблона сертификата
msUpn (string)	UPN субъекта
}	

2.5 Метод поиска заявок

GET API – Поиск заявок
Для УЗ с ролью «Пользователь» в ответе данного метода должны возвращаться только те заявки, получателем сертификата по которым является субъект, связанный с данной УЗ. Для УЗ с ролью «Оператор» в ответе данного метода должны возвращаться заявки, созданные данной УЗ, а также заявки для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного еСА-СА. Для УЗ с ролью «Администратор» в ответе данного метода должны возвращаться все существующие заявки.
URL – /tasks-service/api/v2/public/tasks

²⁰ Шаблоны еСА-СА и еСА-РА имеют сквозную идентификацию.

²¹ Субъекты еСА-СА и еСА-РА имеют сквозную идентификацию.

Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/findAll	
Query	
{	
sortDirection (string),	Направления сортировки (ASC;DESC)
sortBy (string[]),	Список полей, к которым применяется сортировка
pageOffset (integer),	Смещение от начала списка (пагинация)
pageLimit (integer),	Ограничение на размер выборки (пагинация)
search (string),	Полнотекстовый поиск
foreignKeys (string[]),	Фильтр: список внешних ключей
tempalteIds (uuid[]),	Фильтр: список ID шаблонов
creatorIds (uuid[]),	Фильтр: список ID УЗ для фильтрации
types (enum[]: PKCS10, PKCS12, SCEP, WSTEP, ON_TOKEN, UNKNOWN),	Фильтр: список типов заявки
statuses (enum[]: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN)	Фильтр: список статусов заявки
certificateAuthorityId (string[]),	Фильтр: список ID Центров сертификации
serialnumber (string),	Поиск заявки по серийному номеру сертификата
fingerprint (string[])	Фильтр: список отпечатков сертификатов
subjectKeyIdentifier (string[])	Фильтр: Идентификатор ключа субъекта
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ²²
certificateAuthorityId (uuid),	Идентификатор сертификата издателя
type (string: PKCS10, PKCS12, SCEP, WSTEP, ON_TOKEN, UNKNOWN),	Тип заявки
status (string: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ²³
certificateId (uuid),	ID сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string),	Внешний ключ заявки
templateName (string),	Имя шаблона сертификата

²² Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

²³ Субъекты eCA-CA и eCA-RA имеют сквозную идентификацию.

msUpn (string)	UPN субъекта
}	

2.6 Метод получения данных о выпущенном сертификате заявки по ее идентификатору

GET API – Получение данных о выпущенном сертификате заявки по ее идентификатору	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно получение данных о выпущенных сертификатах только по тем заявкам, получателем сертификата по которым является субъект, связанный с данной УЗ. Для УЗ с ролью «Оператор» с использованием данного метода должно быть доступно получение данных о выпущенных сертификатах только по созданным данным оператором заявкам, а также по заявкам для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного еСА-СА. Для УЗ с ролью «Администратор» с использованием данного метода должно быть доступно получение данных о выпущенных сертификатах по любым заявкам.	
URL – /middleware-service/api/v2/public/tasks/{id}/certificate	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/findCertificateByTaskId	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (uuid),	Идентификатор сертификата
chain: {	Цепочка сертификатов (рекурсивный объект)
id (uuid),	Идентификатор сертификата
name (string),	Имя сертификата (на основе CN)
subjectDN: {	Имя субъекта сертификата
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
issuer: {...}	Издатель сертификата (вложенный объект)
},	
serialnumber (string),	Серийный номер сертификата
fingerprint (string),	Фингерпринт сертификата
name (string),	Имя сертификата (на основе CN)
issuerId (uuid),	Идентификатор издателя сертификата
issuerDN: {	Имя субъекта издателя сертификата
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE,	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра

BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]	
},	
subjectDN: {	Имя субъекта сертификата
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectAltName: {	Альтернативное имя субъекта сертификата
(enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[]	Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата ЦС
hashAlgorithm (enum: SHA1, SHA256, SHA384, SHA512, SHA3_256, SHA3_384, SHA3_512, RSASSA_PSS, MD5, GOST_R_34_11_2012, UNKNOWN),	Алгоритм подписи сертификата ЦС
keyBits (int32),	Длина ключа сертификата ЦС
templateId (uuid),	Идентификатор шаблона ²⁴
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип сертификата
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant),	Дата окончания действия сертификата (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN),	Статус сертификата
actions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
publicKey (string),	Открытый ключ
certificateType (string),	Тип сертификата (X.509)
version (int32),	Версия сертификата
subjectKeyIdentifier (string),	Идентификатор ключа сертификата
authorityKeyIdentifier (string)	Идентификатор ключа издателя сертификата
keyUsages: {	Назначение ключа сертификата

²⁴ Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

id (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCIPHERMENT, DATA_ENCIPHERMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCIPHER_ONLY, DECIPHER_ONLY, UNKNOWN),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (enum: ECU_PKIX_ANY_EXTENDED_KEY_USAGE, CSN_369791_TLS_CLIENT, CSN_369791_TLS_SERVER, CLIENT_AUTHENTICATION, CODE_SIGNING, EAP_OVER_LAN, EAP_OVER_PPP, ETSI_TSL_SIGNING, EMAIL_PROTECTION, ICAO_DEVIATION_LIST_SIGNING, ECU_INTEL_AMT, INTERNET_KEY_EXCHANGE_FOR_IPSEC, KERBEROS_CLIENT_AUTHENTICATION, ECU_KRB_PKINIT_KDC, MS_COMMERCIAL_CODE_SIGNING, MS_DOCUMENT_SIGNING, MS_EFS_RECOVERY, MS_ENCRYPTED_FILE_SYSTEM, MS_INDIVIDUAL_CODE_SIGNING, MS_SMART_CARD_LOGON, OCSP_SIGNER, ECU_ADOBE_PDF_SIGNING, PIV_CARD_AUTHENTICATION, SCVP_CLIENT, SCVP_SERVER, SIP_DOMAIN, ECU_PKIX_SSH_CLIENT, SSH_SERVER, SERVER_AUTHENTICATION, TIME_STAMPING, ICAO_MASTER_LIST_SIGNING, UNKNOWN),	Перечисление расширенного использования ключа
value (string),	Наименование элемента
oid (string),	OID назначения
description (string)	Описание использования ключа
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
}	

3 МЕТОДЫ ЭКСПОРТА ФАЙЛОВ

3.1 Метод получения запроса на сертификат по идентификатору заявки

GET API – Получение запроса на сертификат по идентификатору заявки	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно получение запроса на сертификат только из тех заявок на сертификат на основании запроса, получателем сертификата по которым является субъект, связанный с данной УЗ. Для УЗ с ролью «Оператор» с использованием данного метода должно быть доступно получение запроса на сертификат только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA. Для УЗ с ролью «Администратор» с использованием данного метода должно быть доступно получение запроса на сертификат из любых заявок на сертификат на основании запроса. Для успешного получения запроса на сертификат из заявки она должна иметь тип «PKCS10».	
URL – /export-service/api/v2/public/export/task/{id}/request	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findRequestByTaskId	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	
Response	
ResponseEntity->byte[]	

3.2 Метод получения сертификата по идентификатору заявки

GET API – Получение сертификата по идентификатору заявки	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно получение сертификатов только из тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ. Для УЗ с ролью «Оператор» с использованием данного метода должно быть доступно получение сертификата только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA. Для УЗ с ролью «Администратор» с использованием данного метода должно быть доступно получение сертификатов из любых заявок. Для успешного получения сертификата из заявки она должна иметь статус «COMPLETED».	
URL – /export-service/api/v2/public/export/task/{id}/certificate	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findCertificateByTaskId_1	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	

Response
ResponseEntity->byte[]

3.3 Метод получения цепочки сертификатов по идентификатору заявки

GET API – Получение цепочки сертификатов по идентификатору заявки	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно получение цепочки сертификатов только из тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ. Для УЗ с ролью «Оператор» с использованием данного метода должно быть доступно получение цепочки сертификатов только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA. Для УЗ с ролью «Администратор» с использованием данного метода должно быть доступно получение цепочки сертификатов из любых заявок. Для успешного получения цепочки сертификатов из заявки она должна иметь статус «COMPLETED».	
URL – /export-service/api/v2/public/export/task/{id}/chain	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findChainByTaskId	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	
Response	
ResponseEntity->byte[]	

3.4 Метод получения контейнера pkcs12 по идентификатору заявки

GET API – Получение контейнера pkcs12 по идентификатору заявки	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно получение контейнера pkcs12 только из тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ. Для УЗ с ролью «Оператор» с использованием данного метода должно быть доступно получение контейнера pkcs12 только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA. Для УЗ с ролью «Администратор» с использованием данного метода должно быть доступно получение контейнера pkcs12 из любых заявок. Для успешного получения контейнера pkcs12 из заявки она должна иметь статус «COMPLETED» и тип «PKCS12».	
URL – /export-service/api/v2/public/export/task/{id}/pkcs12	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findP12ByTaskId	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	

Response
ResponseEntity->byte[]

3.5.3.5 Метод получения сертификата издателя сертификата по заявке

GET API – Получение сертификата издателя сертификата по заявке	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно получение сертификата издателя только из тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ. Для УЗ с ролью «Оператор» с использованием данного метода должно быть доступно получение сертификата издателя только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA. Для УЗ с ролью «Администратор» с использованием данного метода должно быть доступно получение сертификата издателя из любых заявок. Для успешного получения сертификата издателя из заявки она должна иметь статус «COMPLETED».	
URL – /export-service/api/v2/public/export/task/{id}/issuer/certificate	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findIssuerCertificateByTaskId	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	
Response	
ResponseEntity->byte[]	

3.6 Метод получения цепочки сертификатов издателя сертификата по заявке

GET API – Получение цепочки сертификатов издателя сертификата по заявке	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно получение цепочки сертификатов издателя только из тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ. Для УЗ с ролью «Оператор» с использованием данного метода должно быть доступно получение цепочки сертификатов издателя только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA. Для УЗ с ролью «Администратор» с использованием данного метода должно быть доступно получение цепочки сертификатов издателя из любых заявок. Для успешного получения цепочки сертификатов издателя из заявки она должна иметь статус «COMPLETED».	
URL – /export-service/api/v2/public/export/task/{id}/issuer/chain	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findIssuerChainByTaskId	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	

Response
ResponseEntity->byte[]

3.7 Метод получения CRL издателя сертификата по заявке

GET API – Получение CRL издателя сертификата по заявке	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно получение CRL издателя только из тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ. Для УЗ с ролью «Оператор» с использованием данного метода должно быть доступно CRL издателя только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA. Для УЗ с ролью «Администратор» с использованием данного метода должно быть доступно получение CRL издателя из любых заявок. Для успешного получения CRL издателя из заявки она должна иметь статус «COMPLETED».	
URL – /export-service/api/v2/public/export/task/{id}/issuer/crl	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findIssuerCrlByTaskId	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	
Response	
ResponseEntity->byte[]	

4 МЕТОДЫ РАБОТЫ С СУБЪЕКТАМИ

4.1 Метод поиска субъектов

GET API – Поиск субъектов	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно только получение данных субъекта, связанный с данной УЗ. Для УЗ с ролью «Оператор» или «Администратор» с использованием данного метода должно быть доступно получение данных любого субъекта подключенного еСА-СА.	
URL – /middleware-service/api/v2/public/subjects	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/findById	
Query	
{	
id (UUID[]), [опционально]	ID субъекта ²⁵
resourceId (UUID[]), [опционально]	ID ресурсной системы
securityGroupId (UUID[]), [опционально]	ID группы безопасности
search (string), [опционально]	Полнотекстовый поиск (имя субъекта)
isConnected (boolean), [опционально]	Флаг: субъект подключен к ресурсной системе
isBlocked (boolean), [опционально]	Флаг: субъект заблокирован в ресурсной системе
pageLimit (integer), [опционально]	Ограничение на размер выборки (пагинация)
pageOffset (integer), [опционально]	Смещение от начала списка (пагинация)
sortBy (string[]), [опционально]	Список полей, к которым применяется сортировка
sortDirection (string) [опционально]	Направления сортировки (ASC;DESC)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта ²⁶
commonName (string),	Имя субъекта
distinguishedName (string),	Расположение субъекта в ресурсной системе
resource: {	Ресурсная система
id (uuid),	ID ресурсной системы
commonName (string),	Имя ресурсной системы
distinguishedName (string)	BaseDN точки подключения к ресурсной системе
},	
subjectName: {	Имя субъекта
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L,	Поля разделенного имени субъекта

²⁵ Субъекты еСА-СА и еСА-РА имеют сквозную идентификацию.

²⁶ Субъекты еСА-СА и еСА-РА имеют сквозную идентификацию.

ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): {	
values (string[]),	Значения компонента
editable (boolean),	Флаг: компонент доступен для редактирования
}	
},	
subjectAltName: {	Альтернативное имя субъекта
(enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): {	Поля альтернативного имени субъекта
values (string[]),	Значения компонента
editable (boolean),	Флаг: компонент доступен для редактирования
}	
},	
isConnected (boolean),	Флаг: субъект подключен к ресурсной системе
isBlocked (boolean),	Флаг: субъект заблокирован в ресурсной системе
modify (instant),	Время изменения (ISO 8601)
certificatesCount (integer),	Количество действующих сертификатов у субъекта
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

4.2 Метод получения субъекта по идентификатору

GET API – Получение субъекта по идентификатору	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно только получение данных субъекта, связанный с данной УЗ. Для УЗ с ролью «Оператор» или «Администратор» с использованием данного метода должно быть доступно получение данных любого субъекта подключенного еСА-СА.	
URL – /middleware-service/api/v2/public/subjects/{id}	
Swagger: : https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20субъекты/findById_2	
Query	
{	
id (UUID)	ID субъекта ²⁷
}	
Request	
-	
Response	
{	Ответ JSON в HTTP-body

²⁷ Субъекты еСА-СА и еСА-РА имеют сквозную идентификацию.

id (UUID),	ID субъекта ²⁸
commonName (string),	Имя субъекта
distinguishedName (string),	Расположение субъекта в ресурсной системе
resource: {	Ресурсная система
id (uuid),	ID ресурсной системы
commonName (string),	Имя ресурсной системы
distinguishedName (string)	BaseDN точки подключения к ресурсной системе
},	
subjectName: {	Имя субъекта
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): {	Поля разделенного имени субъекта
values (string[]),	Значения компонента
editable (boolean),	Флаг: компонент доступен для редактирования
}	
},	
subjectAltName: {	Альтернативное имя субъекта
(enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): {	Поля альтернативного имени субъекта
values (string[]),	Значения компонента
editable (boolean),	Флаг: компонент доступен для редактирования
}	
},	
isConnected (boolean),	Флаг: субъект подключен к ресурсной системе
isBlocked (boolean),	Флаг: субъект заблокирован в ресурсной системе
modify (instant),	Время изменения (ISO 8601)
certificatesCount (integer),	Количество действующих сертификатов у субъекта
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

²⁸ Субъекты eCA-CA и eCA-RA имеют сквозную идентификацию.

5 МЕТОДЫ РАБОТЫ С ШАБЛОНАМИ

5.1 Метод поиска шаблонов

GET API – Поиск шаблонов	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно получение только тех шаблонов, которые указаны в правилах выпуска с режимом обработки «Автоматический выпуск» или «Ручная обработка» для ассоциированного с текущей учетной записью субъекта. Для УЗ с ролью «Оператор» с использованием данного метода должно быть доступно получение только тех шаблонов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA. Для УЗ с ролью «Администратор» с использованием данного метода должно быть доступно получение любых шаблонов.	
URL – /middleware-service/api/v2/public/templates	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20шаблоны/findAll_1	
Query	
{	
pageLimit (integer), [опционально]	Ограничение на размер выборки (пагинация)
pageOffset (integer), [опционально]	Смещение от начала списка (пагинация)
sortBy (string[]), [опционально]	Список полей, к которым применяется сортировка
sortDirection (string), [опционально]	Направления сортировки (ASC;DESC)
types (enum[]: EMBEDDED, CLONED, IMPORTED, UNKNOWN), [опционально]	Тип шаблона
endEntityType (enum[]: USER, DEVICE, ROOT_CA, SUB_CA, UNKNOWN) [опционально],	Тип субъекта
certificateType (enum[]: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN), [опционально]	Тип выпускаемого сертификата
search (string), [опционально]	Полнотекстовый поиск по имени шаблона
removed (boolean), [опционально]	Флаг: шаблон удален
id (UUID[]), [опционально]	ID шаблона
notId (UUID[]) [опционально]	Исключая ID шаблона
keyAlgorithm (enum[]: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN) [опционально]	Фильтр: алгоритм ключа включен в шаблоне ²⁹
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID шаблона ³⁰
name (string),	Имя шаблона
type (enum: EMBEDDED, CLONED, IMPORTED, UNKNOWN),	Тип шаблона

²⁹ В случае использования множественных значений для фильтра «keyAlgorithm» в ответе метода будут содержаться шаблоны, в которых включен хотя бы один алгоритм из перечня, указанного в данном фильтре.

³⁰ Шаблоны ЦС Aladdin eCA и ЦР Aladdine eCA имеют сквозную идентификацию.

certificateType (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип выпускаемого сертификата
certificateAuthorityId (UUID),	ID ЦС, который должен использоваться при выпуске сертификата по данному шаблону. Шаблоны, у которых в поле «Центр сертификации» указано «Любой», будут иметь в данном поле значение NULL
certificateCount (int64),	Число выпущенных по шаблону сертификатов
removed (boolean),	Флаг: шаблон удален
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

5.2 Метод получения шаблона по идентификатору

GET API – Получение шаблона по идентификатору	
Для УЗ с ролью «Пользователь» с использованием данного метода должно быть доступно получение только тех шаблонов, которые указаны в правилах выпуска с режимом обработки «Автоматический выпуск» или «Ручная обработка» для ассоциированного с текущей учетной записью субъекта. Для УЗ с ролью «Оператор» с использованием данного метода должно быть доступно получение только тех шаблонов, доступ к которым данному оператору предоставлен по правилам доступа подключенного еСА-СА. Для УЗ с ролью «Администратор» с использованием данного метода должно быть доступно получение любых шаблонов.	
URL – /middleware-service/api/v2/public/templates/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20шаблоны/findById_1	
Query	
{	
templateId (UUID)	ID шаблона ³¹
}	
Request	
-	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID шаблона ³²
name (string),	Имя шаблона
validity (int64),	Время действия выпускаемого сертификата (мс)
certificateType (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип выпускаемого сертификата
type (enum: EMBEDDED, CLONED, IMPORTED, UNKNOWN),	Тип шаблона
certificateAuthorityId (UUID),	ID ЦС, который должен использоваться при выпуске сертификата по данному шаблону. Шаблоны, у которых в поле «Центр сертификации» указано «Любой», будут иметь в данном поле значение NULL
removed (boolean),	Флаг: шаблон удален

³¹ Шаблоны еСА-СА и еСА-РА имеют сквозную идентификацию.

³² Шаблоны еСА-СА и еСА-РА имеют сквозную идентификацию.

rsa: {	Описание RSA-криптографии
use (boolean),	Флаг: RSA-ключи доступны для шаблона
minLength (int32),	Минимальная длина RSA-ключа
lengths (int32[])	Доступные длины RSA-ключа
},	
ecdsa: {	Описание ESDCA-криптографии
use (boolean),	Флаг: ESDCA -ключи доступны для шаблона
minLength (int32),	Минимальная длина ESDCA -ключа
lengths (int32[])	Доступные длины ESDCA -ключа
},	
gost: {	Описание ГОСТ-криптографии
use (boolean),	Флаг: ГОСТ -ключи доступны для шаблона
minLength (int32),	Минимальная длина ГОСТ -ключа
lengths (int32[])	Доступные длины ГОСТ -ключа
},	
keyUsages: {	Назначение ключа сертификата
critical (boolean),	Флаг: расширение критическое
values (enum[:DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPT, DATA_ENCRYPT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY, UNKNOWN])	Значение расширения
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
critical (boolean),	Флаг: расширение критическое
values (string[])	Значение расширения (OIDs)
},	
policies: {	Политики сертификата
critical (boolean),	Флаг: расширение критическое
values (string[])	Значение расширения (OIDs)
},	
subjectDN: [{	Имя субъекта сертификата
index (int32),	Индекс (для сортировки, по умолчанию - 0)
name (string),	Имя компонента
description (string),	Описание компонента
required (boolean),	Флаг: обязателен к заполнению
validation (boolean),	Флаг: валидация значения
modifiable (boolean),	Флаг: доступен к редактированию
regex (string),	Регулярное значение для валидации значения
defaultValue (string),	Значение по умолчанию
alert (string),	Предупреждение о неудачной валидации значения
code (enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN,	Код компонента

DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN)	
}},	
subjectAltName: [{	Расширенное имя субъекта сертификата
index (int32),	Индекс (для сортировки, по умолчанию – 0)
name (string),	Имя компонента
description (string),	Описание компонента
required (boolean),	Флаг: обязателен к заполнению
validation (boolean),	Флаг: валидация значения
modifiable (boolean),	Флаг: доступен к редактированию
regex (string),	Регулярное значение для валидации значения
defaultValue (string),	Значение по умолчанию
alert (string),	Предупреждение о неудачной валидации значения
code (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN),	Код компонента
generalName (int32),	Идентификатор компонента в RFC
oid (string)	OID компонента в RFC
}},	
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

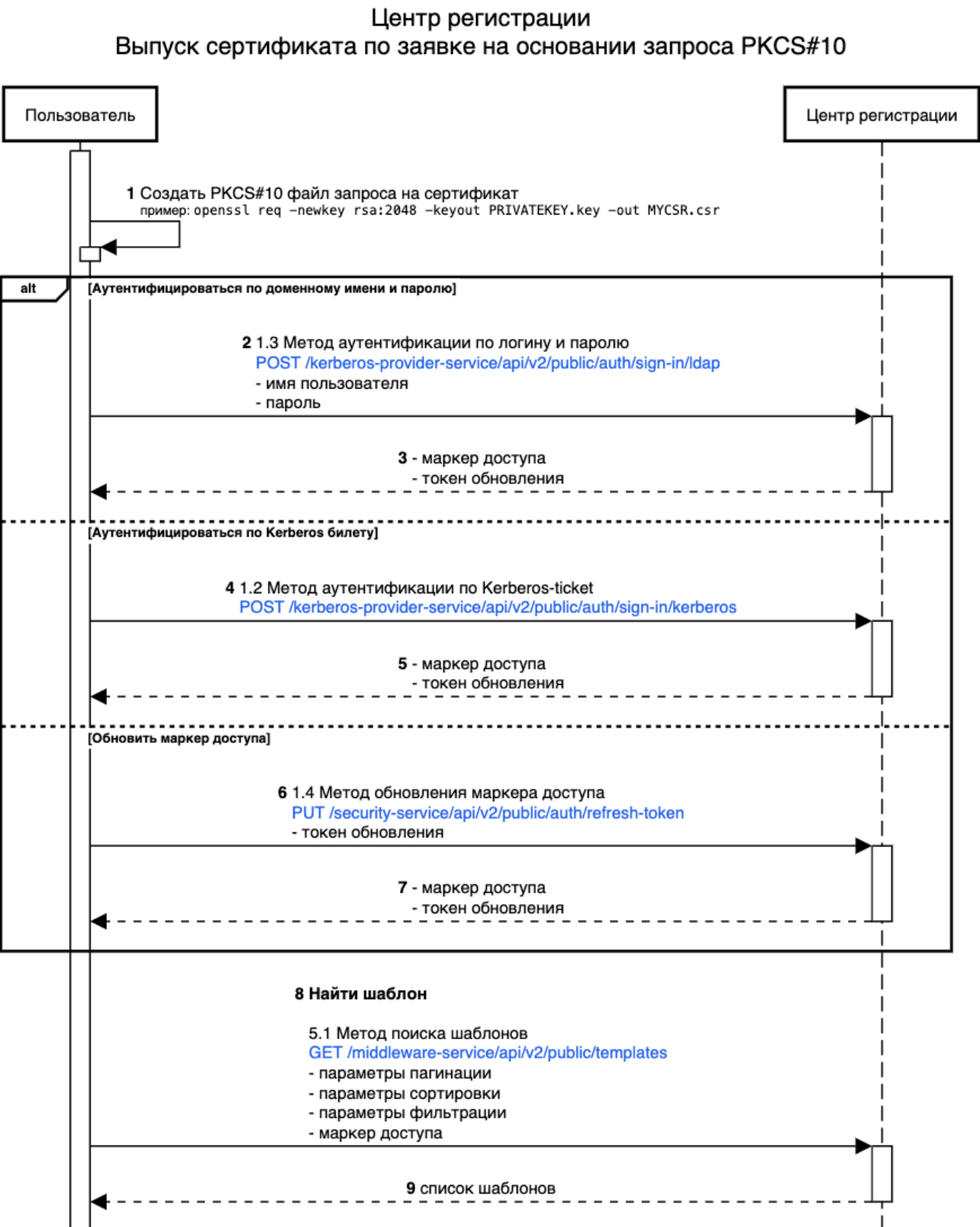
6 МЕТОДЫ РАБОТЫ С СЕРТИФИКАТАМИ

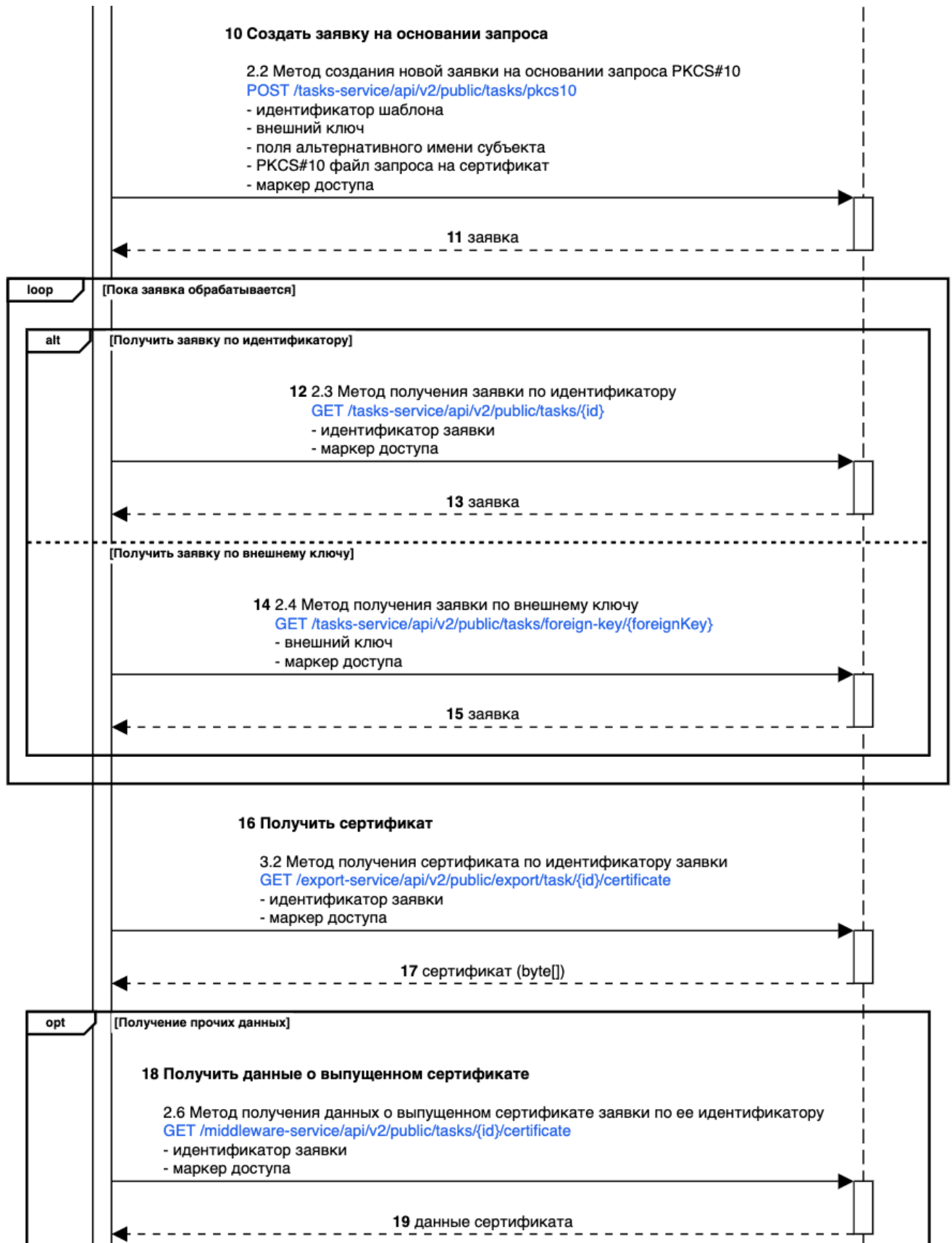
6.1 Метод отзыва сертификата по идентификатору заявки

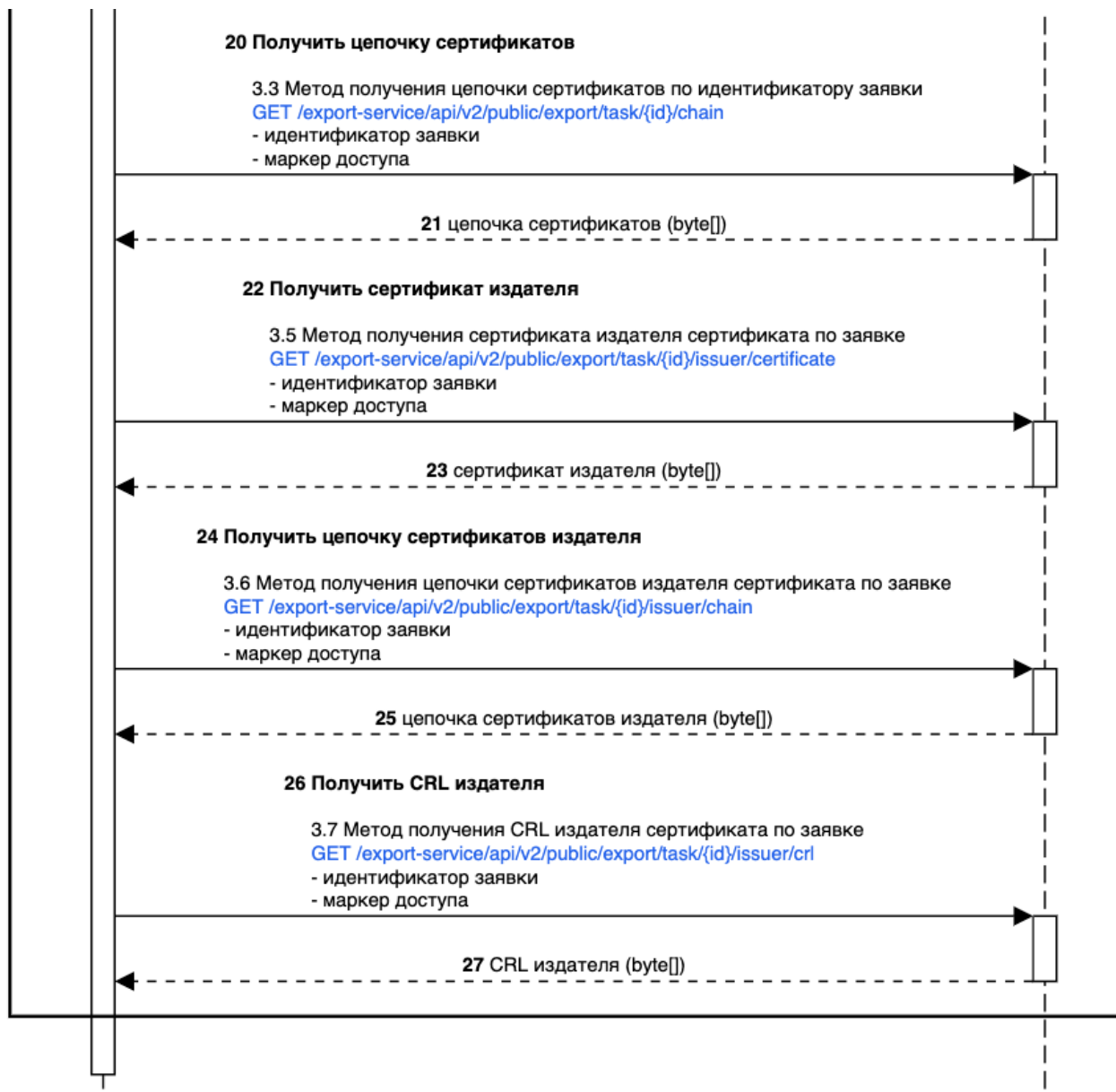
POSTT API – Отзыв сертификата по идентификатору заявки	
Для УЗ с ролью «Пользователь» с использованием данного метода должен быть доступен отзыв сертификатов только из тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ. Для УЗ с ролью «Оператор» с использованием данного метода должен быть доступен отзыв сертификатов из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного Центра сертификации Aladdin eCA. Для УЗ с ролью «Администратор» с использованием данного метода должен быть доступен отзыв сертификатов из любых заявок. Для успешного получения сертификата из заявки она должна иметь статус «COMPLETED».	
URL – /tasks-service/api/v2/public/tasks/{id}/certificate/revoke	
Swagger: https://{имя_хоста}/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/revokeByTaskId	
Query	
{	
id (uuid)	ID заявки
}	
Request	
{	
reason (enum: UNSPECIFIED, KEY_COMPROMISE, CA_COMPROMISE, AFFILIATION_CHANGED, SUPERSEDED, CESSATION_OF_OPERATION, REMOVE_FROM_CRL, PRIVILEGE_WITHDRAWN, AA_COMPROMISE, UNKNOWN)	Причина отзыва
comment (string)	Комментарий
}	
Response	
–	

7 ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ ПОЛУЧЕНИЯ СЕРТИФИКАТА ПО ЗАПРОСУ PKCS#10

7.1 Диаграмма последовательности получения сертификата по запросу PKCS#10







Для получения сертификата по запросу PKCS#10 под учетной записью с ролью «Получатель сертификатов» следует выполнить шаги, представленные на диаграмме последовательности выше. Краткое описание представлено ниже:

- предварительно подготовить файл запроса (шаг 1) – `request`;
- аутентифицироваться на ЦР (шаг 2-5);
 - можно аутентифицироваться по доменному имени и паролю (шаги 2-3);
 - или с использованием Kerberos (шаги 4-5)
- при истечении маркера доступа следует произвести его обновление, используя токен обновления (шаги 6-7);
- все последующие запросы используют маркер доступа;
- найти шаблон сертификата (шаги 8-9) – получить его идентификатор `templateId`;
- создать заявку на основании запроса (шаги 10-11), используются параметры:
 - идентификатор шаблона – `templateId`;
 - файл запроса PKCS#10 – `request`;
 - поля альтернативного имени субъекта;
 - внешний ключ – `foreignKey`.
- в результате будет получен идентификатор заявки – `taskId`;

- проверить статус заявки, отправив запрос на получение заявки по идентификатору `taskId` (шаги 12-13) или по внешнему ключу `foreignKey` (шаги 14-15); проверять следует периодически, пока заявка не перейдет в один из конечных статусов: Выполнена, Отклонена или Отменена;

- сценарий завершается, если заявка перешла в статус Отклонена или Отменена;

- если заявка перешла в статус Выполнена, то по ней был выпущен сертификат и его следует скачать (шаги 18-19), используя `taskId`;

- опционально по идентификатору заявки `taskId` можно получить следующую информацию:

- информацию о выпущенном сертификате (шаги 18-19);
- цепочку сертификатов (шаги 20-21);
- сертификат издателя (шаги 22-23);
- цепочку сертификатов издателя (шаги 24-25);
- и CRL издателя (шаги 26-27).

8 СПЕЦИФИКАЦИЯ ОРЕНАРИ

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "AECA RA external integration service API",
    "description": "API to AECA RA external integration service",
    "termsOfService": "http://example.com/terms/",
    "contact": {
      "name": "Product Service API Team",
      "url": "http://example.com/team",
      "email": "apiteam@example.com"
    },
    "license": {
      "name": "Apache 2.0",
      "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
    },
    "version": "2.0.0",
    "servers": [
      {
        "url": "",
        "description": "Service URL"
      }
    ],
    "security": [
      {
        "X-API-KEY": []
      },
      {
        "token": []
      },
      {
        "X-AECA-Principal-ID": []
      },
      {
        "X-AECA-Is-Account": []
      },
      {
        "X-AECA-Principal-Name": []
      },
      {
        "X-AECA-User-Role": []
      },
      {
        "X-AECA-User-Authorities": []
      },
      {
        "X-User-Session-Id": []
      }
    ],
    "tags": [
      {
        "name": "Контроллер: шаблоны",
        "description": "REST API: шаблоны"
      },
      {
        "name": "[v2] Контроллер: Авторизации",
        "description": "Интерфейс описания Api: Авторизации"
      },
      {
        "name": "Контроллер экспорта",
        "description": "API методов по работе с экспорта"
      },
      {
        "name": "Контроллер: Задачи",
        "description": "REST API: Задачи"
      },
      {
        "name": "Контроллер: Авторизации",
        "description": "Интерфейс описания Api: Авторизации"
      },
      {
        "name": "Контроллер: субъекты",
        "description": "REST API: субъекты"
      }
    ],
    "paths": {
      "/x509-provider-service/api/v2/public/auth/sign-in/x509": {
        "post": {
          "tags": [
            "[v2] Контроллер: Авторизации"
          ],
          "summary": "Аутентификация x509",
          "description": "POST метод REST API: Аутентификация x509",
          "operationId": "signInByX509",
          "responses": {
            "200": {
              "description": "Данные о токене",
              "content": {
                "*/*": {
                  "schema": {
                    "$ref": "#/components/schemas/ItemResponseTokenResponsePublic"
                  }
                }
              },
              "500": {
              "description": "Внутренняя ошибка",
              "content": {
                "*/*": {
                  "schema": {
                    "$ref": "#/components/schemas/ErrorResponse"
                  }
                }
              }
            }
          },
          "/tasks-service/api/v2/public/tasks/pkcs12": {
            "post": {
              "tags": [
                "Контроллер: Задачи"
              ],
              "summary": "Регистрация новой заявки в формате pkcs#12",
              "description": "POST метод REST API: Регистрация новой заявки в формате pkcs#12",
              "operationId": "registerPkcs12",
              "parameters": [
                {
                  "name": "certificateAuthorityId",
                  "in": "query",
                  "description": "ID ЦС",
                  "schema": {
                    "type": "string",
                    "format": "uuid"
                  }
                },
                {
                  "name": "templateId",
                  "in": "query",
                  "description": "ID шаблона",
                  "schema": {
                    "type": "string",
                    "format": "uuid"
                  }
                },
                {
                  "name": "subjectId",
                  "in": "query",
                  "description": "ID субъекта",
                  "schema": {
                    "type": "string",
                    "format": "uuid"
                  }
                },
                {
                  "name": "foreignKey",
                  "in": "query",
                  "description": "Внешний ключ",
                  "schema": {
                    "type": "string"
                  }
                }
              ],
              "requestBody": {
                "content": {
                  "application/json": {
                    "schema": {
                      "$ref": "#/components/schemas/TaskRegisterPkcs12RequestPublic"
                    }
                  },
                  "required": true
                },
                "responses": {
                  "200": {
                    "description": "Заявка зарегистрирована",
                    "content": {
                      "*/*": {
                        "schema": {
                          "$ref": "#/components/schemas/ItemResponseTaskRegisterPkcs12ResponsePublic"
                        }
                      }
                    },
                    "500": {
                    "description": "Внутренняя ошибка",
                    "content": {
                      "*/*": {
                        "schema": {
                          "$ref": "#/components/schemas/ErrorResponse"
                        }
                      }
                    }
                  }
                },
                "/tasks-service/api/v2/public/tasks/pkcs10": {
                  "post": {
                    "tags": [
                      "Контроллер: Задачи"
                    ],
                    "summary": "Регистрация новой заявки в формате pkcs#10",
                    "description": "POST метод REST API: Регистрация новой заявки в формате pkcs#10",
                    "operationId": "registerPkcs10AsFileDescription_1",
                    "parameters": [
                      {
                        "name": "certificateAuthorityId",
                        "in": "query",
                        "description": "ID ЦС",
                        "schema": {
                          "type": "string",
                          "format": "uuid"
                        }
                      },
                      {
                        "name": "templateId",
                        "in": "query",
                        "description": "ID шаблона",
                        "schema": {
                          "type": "string",
                          "format": "uuid"
                        }
                      },
                      {
                        "name": "subjectId",
                        "in": "query",
                        "description": "ID субъекта",
                        "schema": {
                          "type": "string",
                          "format": "uuid"
                        }
                      },
                      {
                        "name": "foreignKey",
                        "in": "query",
                        "description": "Внешний ключ",
                        "schema": {
                          "type": "string"
                        }
                      }
                    ],
                    "requestBody": {
                      "content": {
                        "application/json": {
                          "schema": {
                            "$ref": "#/components/schemas/TaskRegisterPkcs10RequestPublicFileDescription"
                          }
                        },
                        "multipart/form-data": {
                          "schema": {
                            "$ref": "#/components/schemas/TaskRegisterPkcs10RequestPublicMultipartFile"
                          }
                        },
                        "required": true
                      },
                      "responses": {
                        "200": {
                          "description": "Заявка зарегистрирована",
                          "content": {
                            "*/*": {
                              "schema": {
                                "$ref": "#/components/schemas/ItemResponseTaskRegisterPkcs10ResponsePublic"
                              }
                            }
                          },
                          "500": {
                          "description": "Внутренняя ошибка",
                          "content": {
                            "*/*": {
                              "schema": {
                                "oneOf": [
                                  {
                                    "$ref": "#/components/schemas/ItemResponseTaskRegisterPkcs10ResponsePublic"
                                  },
                                  {
                                    "$ref": "#/components/schemas/ErrorResponse"
                                  }
                                ]
                              }
                            }
                          }
                        }
                      },
                      "/security-service/api/v2/public/auth/refresh-token": {
                        "post": {
                          "tags": [
                            "Контроллер: Авторизации"
                          ],
                          "summary": "Обновление токена",
                          "description": "POST метод REST API: Обновление токена",
                          "operationId": "refreshToken",
                          "responses": {
                            "200": {
                              "description": "Данные о токене",
                              "content": {
                                "*/*": {
                                  "schema": {
                                    "$ref": "#/components/schemas/ItemResponseTokenResponse"
                                  }
                                }
                              },
                              "500": {
                              "description": "Внутренняя"

```

АО «Аладдин РД», 1995–2025 г. Руководство администратора. Часть 6. Описание методов REST API Центра регистрации Aladdin Enterprise Registration Authority

Стр. 40 / 51


```

шаблона", "schema": {"type": "array", "items": {"type": "string", "enum": ["EMBEDDED", "CLONED", "IMPORT
ED", "UNKNOWN"]}}}, {"name": "certificateType", "in": "query", "description": "Фильтр: по типу
выпускаемого сертификата", "schema": {"type": "string", "description": "Перечисление типов
сертификата", "enum": ["CERTIFICATE", "ROOT_CA", "SUB_CA", "UNKNOWN"]}}, {"name": "search", "in": "quer
y", "description": "Фильтр: поиск по имени", "schema": {"type": "string", "default": "Name"}}, {"name": "removed", "in": "query", "descriptio
n": "Фильтр: поиск удаленного шаблона", "schema": {"type": "boolean", "default": false}}, {"name": "id", "in": "query", "description":
"Фильтр: поиск по списку идентификаторов", "schema": {"type": "array", "items": {"type": "string", "format": "uuid"}}}, {"name":
"notId", "in": "query", "description": "Фильтр: поиск по списку идентификаторов (исключающий)", "schema": {"type": "array", "items": {"type": "string", "format": "uuid"}}}, {"name": "k
eyAlgorithm", "in": "query", "description": "Фильтр: алгоритм ключа сертификата", "schema": {"type": "array", "items": {"type": "string", "description": "Перечисление
алгоритмов ключа сертификата", "enum": ["RSA", "ECDSA", "GOST_R_34_10_2012", "UNKNOWN"]}}}, {"responses": {"200": {"des
cription": "Список успешно получен", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/CollectionRespo
nseTemplateItemResponsePublic"}}}}, "400": {"description": "Ошибка получения шаблона", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/ErrorResponse"
}}}}, "500": {"description": "Внутренняя ошибка", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/ErrorResponse"
}}}}}, {"middleware-service/api/v2/public/templates/{templateId}": {"get": {"tags": ["Контроллер:
шаблоны"], "summary": "Получить шаблон по идентификатору", "description": "Get метод REST API
получения шаблона по идентификатору", "operationId": "findById_1", "parameters": [{"name": "templateId", "in": "path", "des
cription": "Идентификатор шаблона", "required": true, "schema": {"type": "string", "format": "uuid"}, "example": 1}], "responses":
{"200": {"description": "Шаблон успешно получен", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/ItemResponseTem
plateResponsePublic"}}}}, "400": {"description": "Ошибка получения шаблона", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/ErrorResponse"
}}}}, "500": {"description": "Внутренняя ошибка", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/ErrorResponse"
}}}}}, {"middleware-service/api/v2/public/tasks/{id}/certificate": {"get": {"tags": ["Контроллер:
Задачи"], "summary": "Получить сертификат по идентификатору заявки", "description": "GET метод REST
API: получить сертификат по идентификатору заявки", "operationId": "findCertificateByTaskId", "parameters": [{"name": "id", "in": "path", "descri
ption": "Идентификатор заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}, {"responses": {"200": {"desc
ription": "Сертификат успешно получен", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ItemResponseCertificateRespo
nsePublic"}}}}, "500": {"description": "Внутренняя ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"
}}}}}}}, {"middleware-service/api/v2/public/subjects": {"get": {"tags": ["Контроллер: субъекты"], "summary": "Поиск
субъектов", "description": "GET метод REST API: поиск субъектов", "operationId": "findAll_2", "parameters": [{"name": "id", "in": "query", "description": "Фи
льтр: ID субъекта", "schema": {"type": "array", "items": {"type": "string", "format": "uuid"}}, {"name": "resour
ceId", "in": "query", "description": "Фильтр: ID ресурсной системы", "schema": {"type": "array", "items": {"type": "string", "format": "uuid"}}, {"name": "securit
yGroupId", "in": "query", "description": "Фильтр: ID группы безопасности", "schema": {"type": "array", "items": {"type": "string", "format": "uuid"}}, {"name": "or
ganizationalUnitId", "in": "query", "description": "Фильтр: ID подразделения", "schema": {"type": "array", "items": {"type": "string", "format": "uuid"}}, {"name": "s
earch", "in": "query", "description": "Фильтр: полнотекстовый поиск по имени", "schema": {"type": "string", "default": "Name"}}, {"name": "isConnected", "in": "query", "descri
ption": "Фильтр: подключение субъекта к ресурсной системе", "schema": {"type": "boolean"}}, {"name": "isBlocked", "in": "query", "description": "Фильтр:
блокировка субъекта в ресурсной системе", "schema": {"type": "boolean"}}, {"name": "pageLimit", "in": "query", "description": "Пагинаци
я: ограничение на размер списка", "schema": {"type": "integer", "format": "int32"}}, {"name": "pageOffset", "in": "query", "descr
iption": "Пагинация: смещение от начала

```

```

списка", "schema": {"type": "integer", "format": "int32"}}, {"name": "sortBy", "in": "query", "description": "Сортировка: имя поля, по которому выполнять сортировку", "schema": {"type": "array", "items": {"type": "string"}}, {"name": "sortDirection", "in": "query", "description": "Сортировка: направление сортировки", "schema": {"type": "string"}}, {"responses": {"200": {"description": "Список найденных субъектов", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/CollectionResponseSubjectResponsePublic"}}}}, {"500": {"description": "Внутренняя ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, {"middleware-service/api/v2/public/subjects/{subjectId}": {"get": {"tags": ["Контроллер: субъекты"], "summary": "Поиск субъекта по ID", "description": "GET метод REST API: поиск субъекта по ID", "operationId": "findById_2", "parameters": [{"name": "subjectId", "in": "path", "description": "ID субъекта", "required": true, "schema": {"type": "string", "format": "uuid"}}], "responses": {"200": {"description": "Найденный субъект", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ItemResponseSubjectResponsePublic"}}}}, {"404": {"description": "Субъект с указанным ID не существует", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}, {"500": {"description": "Внутренняя ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, {"external-integration-service/api/version": {"get": {"tags": ["version-controller"], "summary": "Получение версии модуля", "operationId": "getApiVersion", "responses": {"200": {"description": "Версия модуля", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/GetApiVersionResponse"}}}}}}, {"export-service/api/v2/public/export/task/{taskId}/request": {"get": {"tags": ["Контроллер экспорта"], "summary": "Получение запроса на сертификат по идентификатору заявки", "description": "GET метод REST API: Получение запроса на сертификат", "operationId": "findRequestByTaskId", "parameters": [{"name": "taskId", "in": "path", "description": "Идентификатор заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}], "responses": {"200": {"description": "Файл запроса на сертификат", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, {"500": {"description": "Внутренняя ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, {"export-service/api/v2/public/export/task/{taskId}/pkcs12": {"get": {"tags": ["Контроллер экспорта"], "summary": "Получение контейнера pkcs12 по идентификатору заявки", "description": "GET метод REST API: Получение контейнера pkcs12", "operationId": "findPkcs12ByTaskId", "parameters": [{"name": "taskId", "in": "path", "description": "Идентификатор заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}], "responses": {"200": {"description": "Файл контейнера pkcs12", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, {"500": {"description": "Внутренняя ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, {"export-service/api/v2/public/export/task/{taskId}/issuer/crl": {"get": {"tags": ["Контроллер экспорта"], "summary": "Получение crl издателя по идентификатору заявки", "description": "GET метод REST API: Получение crl издателя по идентификатору заявки", "operationId": "findIssuerCrlByTaskId", "parameters": [{"name": "taskId", "in": "path", "description": "Идентификатор заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}], "responses": {"200": {"description": "Файл crl", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, {"500": {"description": "Внутренняя ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, {"export-service/api/v2/public/export/task/{taskId}/issuer/chain": {"get": {"tags": ["Контроллер экспорта"], "summary": "Получение цепочки сертификатов издателя по идентификатору заявки", "description": "GET метод REST API: Получение цепочки сертификатов издателя", "operationId": "findIssuerChainByTaskId", "parameters": [{"name": "taskId", "in": "path", "description": "Идентификатор заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}], "responses": {"200": {"description": "Файл цепочки сертификатов", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, {"500": {"description": "Внутренняя ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, {"export-service/api/v2/public/export/task/{taskId}/issuer/certificate": {"get": {"tags": ["Контроллер

```

```

экспорта"], "summary": "Получение сертификата издателя по идентификатору
заявки", "description": "GET метод REST API: Получение сертификата
издателя", "operationId": "findIssuerCertificateByTaskId", "parameters": [{"name": "taskId", "in": "p
ath", "description": "Идентификатор
заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}], "responses": {"200": {"desc
ription": "Файл
сертификата", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, "500": {"descripti
on": "Внутренняя
ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, "/expor
t-service/api/v2/public/export/task/{taskId}/chain": {"get": {"tags": ["Контроллер
экспорта"], "summary": "Получение цепочки сертификатов по идентификатору
заявки", "description": "GET метод REST API: Получение цепочки
сертификатов", "operationId": "findChainByTaskId", "parameters": [{"name": "taskId", "in": "path", "de
scription": "Идентификатор
заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}], "responses": {"200": {"desc
ription": "Файл
цепочки
сертификатов", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, "500": {"descripti
on": "Внутренняя
ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, "/expor
t-service/api/v2/public/export/task/{taskId}/certificate": {"get": {"tags": ["Контроллер
экспорта"], "summary": "Получение сертификата по идентификатору заявки", "description": "GET метод
REST API: Получение
сертификата", "operationId": "findCertificateByTaskId_1", "parameters": [{"name": "taskId", "in": "pa
th", "description": "Идентификатор
заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}], "responses": {"200": {"desc
ription": "Файл
сертификата", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, "500": {"descripti
on": "Внутренняя
ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, "compo
nents": {"schemas": {"Data": {"type": "object", "properties": {"id": {"type": "object", "description": "
Идентификатор сущности, работа с которым привела к
ошибке"}, "fieldName": {"type": "string", "description": "Имя поля, которое не прошло
валидацию"}, "fieldValue": {"type": "object", "description": "Значение поля, которое не прошло
валидацию"}, "causedBy": {"type": "string", "description": "Сообщение, чем вызвано
исключение"}, "description": "Объект с деталями
ошибки"}, "ErrorResponse": {"type": "object", "properties": {"status": {"type": "integer", "format": "i
nt32"}, "code": {"type": "integer", "description": "Код ошибки xxxууу, где xxx – код сервиса, ууу –
внутренний код ошибки", "format": "int32"}, "message": {"type": "string", "description": "Описание
ошибки"}, "data": {"$ref": "#/components/schemas/Data"}}, "description": "Стандартизированный
объект ответа сервиса, содержащий информацию об
ошибке"}, "ItemResponseTokenResponsePublic": {"type": "object", "properties": {"status": {"type": "in
teger", "format": "int32"}, "data": {"$ref": "#/components/schemas/TokenResponsePublic"}}, "descript
ion": "Стандартизированный объект ответа сервиса, содержащий один единственный
элемент"}, "TokenResponsePublic": {"type": "object", "properties": {"token": {"type": "string", "descr
iption": "Токен пользователя"}, "refresh": {"type": "string", "description": "Токен обновления
пользователя"}, "description": "Объект ответа сервиса: данные о
токене"}, "TaskRegisterPkcs12RequestPublic": {"type": "object", "properties": {"subjectDN": {"type":
"object", "additionalProperties": {"type": "array", "description": "Subject
DN", "items": {"type": "string", "description": "Subject
DN"}}, "description": "Subject
DN"}, "subjectAltName": {"type": "object", "additionalProperties": {"type": "array", "description": "S
ubject
AltName", "items": {"type": "string", "description": "Subject
AltName"}}, "description": "Subject
AltName"}, "keyBits": {"type": "integer", "description": "Размерность
ключа", "format": "int32"}, "keyAlgorithm": {"type": "string", "description": "Перечисление
алгоритмов
ключа
сертификата", "enum": ["RSA", "ECDSA", "GOST_R_34_10_2012", "UNKNOWN"]}, "password": {"type": "string"
, "description": "Пароль от контейнера"}, "description": "Запрос на регистрацию новой заявки в
формате
pkcs#12"}, "ItemResponseTaskRegisterPkcs12ResponsePublic": {"type": "object", "properties": {"statu
s": {"type": "integer", "format": "int32"}, "data": {"$ref": "#/components/schemas/TaskRegisterPkcs12
ResponsePublic"}}, "description": "Стандартизированный объект ответа сервиса, содержащий один
единственный
элемент"}, "TaskRegisterPkcs12ResponsePublic": {"type": "object", "properties": {"id": {"type": "stri
ng", "description": "Идентификатор

```

```

заявки", "format": "uuid", "templateId": { "type": "string", "description": "Идентификатор
шаблона", "format": "uuid", "type": { "type": "string", "description": "Тип
заявки", "enum": ["PKCS10", "PKCS12", "ON_TOKEN", "SCEP", "WSTEP", "UNKNOWN"] }, "status": { "type": "stri
ng", "description": "Статус
заявки", "enum": ["NEW", "FOR_APPROVE", "ISSUE_ERROR", "REJECTED", "COMPLETED", "CANCELLED", "AWAITING
_IMPORT", "UNKNOWN"] }, "creatorId": { "type": "string", "description": "Идентификатор пользователя,
создавшего заявку", "format": "uuid", "subjectId": { "type": "string", "description": "Идентификатор
субъекта", "format": "uuid", "certificateId": { "type": "string", "description": "Идентификатор
сертификата", "format": "uuid", "name": { "type": "string", "description": "Название
активного
сертификата", "updated": { "type": "string", "description": "Время
последнего
обновления
заявки", "format": "date-time", "created": { "type": "string", "description": "Время
создания
заявки", "format": "date-time", "foreignKey": { "type": "string", "description": "Внешний
ключ" } }, "description": "Ответ
на
регистрацию
новой
заявки
в
формате
pkcs#12", "FileDescription": { "type": "object", "properties": { "contentType": { "type": "string", "fi
leName": { "type": "string", "data": { "type": "string", "format": "byte" }, "description": "Файл
запроса", "TaskRegisterPkcs10RequestPublicFileDescription": { "type": "object", "properties": { "req
uest": { "$ref": "#/components/schemas/FileDescription", "subjectName": { "type": "object", "addition
alProperties": { "type": "array", "description": "Отличительное
имя
субъекта", "items": { "type": "string", "description": "Отличительное
имя
субъекта" }, "description": "Отличительное
имя
субъекта", "subjectAltName": { "type": "object", "additionalProperties": { "type": "array", "descripti
on": "Дополнительное имя субъекта", "items": { "type": "string", "description": "Дополнительное имя
субъекта" }, "description": "Дополнительное имя субъекта" }, "description": "Запрос на регистрацию
новой
заявки
в
формате
pkcs#10", "ItemResponseTaskRegisterPkcs10ResponsePublic": { "type": "object", "properties": { "statu
s": { "type": "integer", "format": "int32", "data": { "$ref": "#/components/schemas/TaskRegisterPkcs10
ResponsePublic" }, "description": "Стандартизированный объект ответа сервиса, содержащий один
единственный
элемент", "TaskRegisterPkcs10ResponsePublic": { "type": "object", "properties": { "id": { "type": "stri
ng", "description": "Идентификатор
заявки", "format": "uuid", "templateId": { "type": "string", "description": "Идентификатор
шаблона", "format": "uuid", "type": { "type": "string", "description": "Тип
заявки", "enum": ["PKCS10", "PKCS12", "ON_TOKEN", "SCEP", "WSTEP", "UNKNOWN"] }, "status": { "type": "stri
ng", "description": "Статус
заявки", "enum": ["NEW", "FOR_APPROVE", "ISSUE_ERROR", "REJECTED", "COMPLETED", "CANCELLED", "AWAITING
_IMPORT", "UNKNOWN"] }, "creatorId": { "type": "string", "description": "Идентификатор пользователя,
создавшего заявку", "format": "uuid", "subjectId": { "type": "string", "description": "Идентификатор
субъекта", "format": "uuid", "certificateId": { "type": "string", "description": "Идентификатор
сертификата", "format": "uuid", "name": { "type": "string", "description": "Название
активного
сертификата", "updated": { "type": "string", "description": "Время
последнего
обновления
заявки", "format": "date-time", "created": { "type": "string", "description": "Время
создания
заявки", "format": "date-time", "foreignKey": { "type": "string", "description": "Внешний
ключ" } }, "description": "Ответ
на
регистрацию
новой
заявки
в
формате
pkcs#10", "TaskRegisterPkcs10RequestPublicMultipartFile": { "type": "object", "properties": { "reque
st": { "type": "string", "format": "binary", "subjectName": { "type": "object", "additionalProperties":
{ "type": "array", "description": "Отличительное
имя
субъекта", "items": { "type": "string", "description": "Отличительное
имя
субъекта" }, "description": "Отличительное
имя
субъекта", "subjectAltName": { "type": "object", "additionalProperties": { "type": "array", "descripti
on": "Дополнительное имя субъекта", "items": { "type": "string", "description": "Дополнительное имя
субъекта" }, "description": "Дополнительное имя субъекта" }, "description": "Запрос на регистрацию
новой
заявки
в
формате
pkcs#10", "ItemResponseTokenResponse": { "type": "object", "properties": { "status": { "type": "integer
", "format": "int32", "data": { "$ref": "#/components/schemas/TokenResponse" }, "description": "Станд
артизированный объект ответа сервиса, содержащий один единственный
элемент", "TokenResponse": { "type": "object", "properties": { "token": { "type": "string", "description
": "Токен
пользователя", "refresh": { "type": "string", "description": "Токен
обновления
пользователя" }, "description": "Объект
ответа
сервиса:
данные
о
токене", "CollectionResponseTaskFindUIResponsePublic": { "type": "object", "properties": { "status":
{ "type": "integer", "format": "int32", "data": { "$ref": "#/components/schemas/DataTaskFindUIRespons
ePublic" }, "description": "Стандартизированный объект ответа сервиса, содержащий коллекцию
элементов", "DataTaskFindUIResponsePublic": { "type": "object", "properties": { "range": { "$ref": "#/c
omponents/schemas/Range", "items": { "type": "array", "description": "Коллекция
элементов", "items": { "$ref": "#/components/schemas/TaskFindUIResponsePublic" } }, "description": "П

```

```

"полезная": {"type": "object", "properties": {"count": {"type": "integer", "description": "Количество элементов", "format": "int64"}, "offset": {"type": "integer", "description": "Ограничение количества элементов", "format": "int64"}, "limit": {"type": "integer", "description": "Общее число страниц", "format": "int32"}, "pageNumber": {"type": "integer", "description": "Номер страницы выборки", "format": "int32"}, "description": "Данные по выборке"}, "TaskFindUIResponsePublic": {"type": "object", "properties": {"id": {"type": "string", "description": "Идентификатор заявки", "format": "uuid"}, "templateId": {"type": "string", "description": "Идентификатор шаблона", "format": "uuid"}, "certificateAuthorityId": {"type": "string", "description": "Идентификатор сертификата издателя", "format": "uuid"}, "type": {"type": "string", "description": "Тип заявки", "enum": ["PKCS10", "PKCS12", "ON_TOKEN", "SCEP", "WSTEP", "UNKNOWN"]}, "status": {"type": "string", "description": "Статус заявки", "enum": ["NEW", "FOR_APPROVE", "ISSUE_ERROR", "REJECTED", "COMPLETED", "CANCELLED", "AWAITING_IMPORT", "UNKNOWN"]}, "creatorId": {"type": "string", "description": "Идентификатор пользователя, создавшего заявку", "format": "uuid"}, "subjectId": {"type": "string", "description": "Идентификатор субъекта", "format": "uuid"}, "certificateId": {"type": "string", "description": "Идентификатор сертификата", "format": "uuid"}, "name": {"type": "string", "description": "Название активного сертификата", "format": "string"}, "updated": {"type": "string", "description": "Время последнего обновления заявки", "format": "date-time"}, "created": {"type": "string", "description": "Время создания заявки", "format": "date-time"}, "foreignKey": {"type": "string", "description": "Внешний ключ"}, "templateName": {"type": "string", "description": "Имя шаблона"}, "msUpn": {"type": "string", "description": "MS UPN"}, "description": "Объект ответа сервиса, содержащий данные об ответе"}, "ItemResponseTaskFindUIResponsePublic": {"type": "object", "properties": {"status": {"type": "integer", "format": "int32"}, "data": {"$ref": "#/components/schemas/TaskFindUIResponsePublic"}}, "description": "Стандартизированный объект ответа сервиса, содержащий один единственный элемент"}, "CollectionResponseTemplateItemResponsePublic": {"type": "object", "properties": {"status": {"type": "integer", "format": "int32"}, "data": {"$ref": "#/components/schemas/DataTemplateItemResponsePublic"}}, "description": "Стандартизированный объект ответа сервиса, содержащий коллекцию элементов"}, "DataTemplateItemResponsePublic": {"type": "object", "properties": {"range": {"$ref": "#/components/schemas/Range"}, "items": {"type": "array", "description": "Коллекция элементов", "items": {"$ref": "#/components/schemas/TemplateItemResponsePublic"}}, "description": "Полезная нагрузка с коллекцией элементов"}, "TemplateItemResponsePublic": {"type": "object", "properties": {"id": {"type": "string", "description": "Идентификатор шаблона", "format": "uuid"}, "name": {"type": "string", "description": "Наименование шаблона"}, "certificateType": {"type": "string", "description": "Перечисление типов сертификата", "enum": ["CERTIFICATE", "ROOT_CA", "SUB_CA", "UNKNOWN"]}, "type": {"type": "string", "description": "Тип шаблона", "enum": ["EMBEDDED", "CLONED", "IMPORTED", "UNKNOWN"]}, "certificateCount": {"type": "integer", "description": "Количество выпущенных сертификатов", "format": "int64"}, "removed": {"type": "boolean", "description": "Удаленный шаблон"}, "created": {"type": "string", "description": "Дата создания шаблона", "format": "date-time"}, "updated": {"type": "string", "description": "Дата обновления шаблона", "format": "date-time"}, "description": "Объект ответа сервиса, содержащий данные элемента коллекции шаблонов"}, "CertificatePoliciesPublic": {"type": "object", "properties": {"critical": {"type": "boolean", "description": "Считать критическим"}, "values": {"type": "array", "description": "Перечень политик", "items": {"type": "string", "description": "Перечень политик"}}, "description": "Класс политик сертификата"}, "ExtendedKeyUsagePublic": {"type": "object", "properties": {"critical": {"type": "boolean", "description": "Считать критическим"}, "values": {"type": "array", "description": "Значения (oids)", "items": {"type": "string", "description": "Значения (oids)"}}, "description": "Класс расширенного использования ключа"}, "ItemResponseTemplateResponsePublic": {"type": "object", "properties": {"status": {"type": "integer", "format": "int32"}, "data": {"$ref": "#/components/schemas/TemplateResponsePublic"}}, "description": "Стандартизированный объект ответа сервиса, содержащий один единственный элемент"}, "KeyAlgorithmPublic": {"type": "object", "properties": {"use": {"type": "boolean", "description": "Используется ли"}, "minLength": {"type": "integer", "description": "Минимальная длина ключа", "format": "int32"}, "lengths": {"type": "array", "description": "Длины алгоритма с учетом минимальной длины", "items": {"type": "integer", "description": "Длины алгоритма с учетом минимальной длины", "format": "int32"}}, "description": "Объект сервиса, содержащего данные

```

```

алгоритма                                                                                                   ключа
сертификата"}, {"KeyUsagePublic": {"type": "object", "properties": {"critical": {"type": "boolean", "de
scription": "Считать
критическим"}, {"values": {"type": "array", "description": "Значения", "items": {"type": "string", "desc
ription": "Перечисление                                                                                                   использования
ключа", "enum": ["DIGITAL_SIGNATURE", "NON_REPUDIATION", "KEY_ENCIPHERMENT", "DATA_ENCIPHERMENT", "K
EY_AGREEMENT", "KEY_CERT_SIGN", "CRL_SIGN", "ENCIPHER_ONLY", "DECIPHER_ONLY", "UNKNOWN"]}}}, {"descri
ption": "Класс                                                                                                   использования
ключа"}, {"SubjectAltNameFieldPublic": {"required": ["code", "extendable", "generalName", "modifiable
", "name", "required", "validation"], "type": "object", "properties": {"index": {"type": "integer", "des
cription": "Индекс                                                                                                   поля                                                                                                   имени
субъекта", "format": "int32"}, "name": {"type": "string", "description": "Полное имя поля имени
субъекта"}, {"description": {"type": "string", "description": "Описание поля имени
субъекта"}, {"required": {"type": "boolean", "description": "Обязательный
ли"}, {"validation": {"type": "boolean", "description": "Используется ли
валидация"}, {"modifiable": {"type": "boolean", "description": "Редактируемый
ли"}, {"extendable": {"type": "boolean", "description": "Флаг: расширяемое
поле"}, {"regex": {"type": "string", "description": "Регулярное
выражение"}, {"defaultValue": {"type": "string", "description": "Значение по
умолчанию"}, {"alert": {"type": "string", "description": "Предупреждающее сообщение поля имени
субъекта"}, {"code": {"type": "string", "description": "Перечисление полей Subject
AltName", "enum": ["RFC822NAME", "DNS_NAME", "IPADDRESS", "DIRECTORY_NAME", "UNIFORM_RESOURCE_ID", "R
EGISTERED_ID", "MS_UPN", "MS_GUID", "KRB5PRINCIPAL", "PERMANENT_IDENTIFIER", "XMPP_ADDR", "SRV_NAME"
, "SUBJECT_IDENTIFICATION_METHOD", "UNKNOWN"]}, {"generalName": {"type": "integer", "description": "Ид
ентификатор имени в RFC", "format": "int32"}, {"oid": {"type": "string", "description": "OID в
RFC"}}, {"description": "Класс поля дополнительного имени
субъекта"}, {"SubjectNameFieldPublic": {"required": ["code", "extendable", "modifiable", "name", "requ
ired", "validation"], "type": "object", "properties": {"index": {"type": "integer", "description": "Инд
екс поля имени субъекта", "format": "int32"}, "name": {"type": "string", "description": "Полное имя
поля имени субъекта"}, {"description": {"type": "string", "description": "Описание поля имени
субъекта"}, {"required": {"type": "boolean", "description": "Обязательный
ли"}, {"validation": {"type": "boolean", "description": "Используется ли
валидация"}, {"modifiable": {"type": "boolean", "description": "Редактируемый
ли"}, {"extendable": {"type": "boolean", "description": "Флаг: расширяемое
поле"}, {"regex": {"type": "string", "description": "Регулярное
выражение"}, {"defaultValue": {"type": "string", "description": "Значение по
умолчанию"}, {"alert": {"type": "string", "description": "Предупреждающее сообщение поля имени
субъекта"}, {"code": {"type": "string", "description": "Перечисление полей Subject
DN", "enum": ["CN", "UID", "E", "EMAILADDRESS", "MAIL", "SERIALNUMBER", "GIVENNAME", "INITIALS", "SURNAM
E", "OU", "O", "L", "ST", "DC", "C", "UNSTRUCTUREDADDRESS", "UNSTRUCTUREDNAME", "POSTALCODE", "BUSINESSC
ATEGORY", "TELEPHONENUMBER", "PSEUDONYM", "POSTALADDRESS", "STREET", "NAME", "T", "DN", "DESCRIPTION",
"INN", "OGRN", "OGRNIP", "SNILS", "INNLE", "UNKNOWN"]}}}, {"description": "Класс поля отличительного
имени
субъекта"}, {"TemplateResponsePublic": {"type": "object", "properties": {"id": {"type": "string", "desc
ription": "Идентификатор
шаблона", "format": "uuid"}, "name": {"type": "string", "description": "Наименование
шаблона"}, {"validity": {"type": "integer", "description": "Срок действия
сертификата", "format": "int64"}, {"certificateType": {"type": "string", "description": "Перечисление
типов
сертификата", "enum": ["CERTIFICATE", "ROOT_CA", "SUB_CA", "UNKNOWN"]}, {"type": {"type": "string", "des
cription": "Тип
шаблона", "enum": ["EMBEDDED", "CLONED", "IMPORTED", "UNKNOWN"]}, {"removed": {"type": "boolean", "descr
iption": "Удаленный
шаблон"}, {"rsa": {"$ref": "#/components/schemas/KeyAlgorithmPublic"}, {"ecdsa": {"$ref": "#/component
s/schemas/KeyAlgorithmPublic"}, {"gost": {"$ref": "#/components/schemas/KeyAlgorithmPublic"}, {"keyU
sage": {"$ref": "#/components/schemas/KeyUsagePublic"}, {"extendedKeyUsage": {"$ref": "#/components/
schemas/ExtendedKeyUsagePublic"}, {"policies": {"$ref": "#/components/schemas/CertificatePoliciesP
ublic"}, {"subjectDN": {"type": "array", "description": "Отличительное имя
субъекта", "items": {"$ref": "#/components/schemas/SubjectNameFieldPublic"}}, {"subjectAltName": {"t
ype": "array", "description": "Дополнительное имя
субъекта", "items": {"$ref": "#/components/schemas/SubjectAltNameFieldPublic"}}, {"created": {"type"
: "string", "description": "Дата создания шаблона", "format": "date-
time"}, {"updated": {"type": "string", "description": "Дата обновления шаблона", "format": "date-
time"}}, {"description": "Объект ответа сервиса, содержащий данные

```

```

шаблона"}, "CertificateChainPublic": {"type": "object", "properties": {"id": {"type": "string", "description": "Уникальный идентификатор сертификата", "format": "uuid"}, "name": {"type": "string", "description": "Имя сертификата (формируется из CN, разделитель \"_\""}, "subjectDN": {"type": "object", "additionalProperties": {"type": "array", "description": "Subject DN", "items": {"type": "string", "description": "Subject DN"}}, "description": "Subject DN"}, "issuer": {"$ref": "#/components/schemas/CertificateChainPublic"}}, "description": "POJO: цепочка сертификатов"}, "CertificateDownloadActionsPublic": {"type": "object", "properties": {"p12": {"type": "boolean", "description": "Флаг: выгрузка контейнера p12"}, "pem": {"type": "boolean", "description": "Флаг: выгрузка сертификата"}, "csr": {"type": "boolean", "description": "Флаг: выгрузка запроса сертификата"}}, "description": "POJO: Действия выгрузки у сертификата"}, "CertificateResponsePublic": {"type": "object", "properties": {"id": {"type": "string", "description": "Уникальный идентификатор сертификата", "format": "uuid"}, "chain": {"$ref": "#/components/schemas/CertificateChainPublic"}, "serialnumber": {"type": "string", "description": "Серийный номер сертификата"}, "fingerprint": {"type": "string", "description": "Фингерпринт сертификата"}, "name": {"type": "string", "description": "Имя сертификата (формируется из CN, разделитель \"_\""}, "issuerId": {"type": "string", "description": "Уникальный идентификатор издающего сертификата", "format": "uuid"}, "issuerDN": {"type": "object", "additionalProperties": {"type": "array", "description": "Issuer DN", "items": {"type": "string", "description": "Issuer DN"}}, "description": "Issuer DN"}, "subjectDN": {"type": "object", "additionalProperties": {"type": "array", "description": "Subject DN", "items": {"type": "string", "description": "Subject DN"}}, "description": "Subject DN"}, "subjectAltName": {"type": "object", "additionalProperties": {"type": "array", "description": "Subject AltName", "items": {"type": "string", "description": "Subject AltName"}}, "description": "Subject AltName"}, "keyAlgorithm": {"type": "string", "description": "Перечисление алгоритмов ключа сертификата", "enum": ["RSA", "ECDSA", "GOST_R_34_10_2012", "UNKNOWN"]}, "hashAlgorithm": {"type": "string", "description": "Перечисление Hash алгоритмов", "enum": ["SHA1", "SHA256", "SHA384", "SHA512", "SHA3_256", "SHA3_384", "SHA3_512", "RSASSA_PSS", "MD5", "GOST_R_34_11_2012", "UNKNOWN"]}, "keyBits": {"type": "integer", "description": "Размерность ключа", "format": "int32"}, "templateId": {"type": "string", "description": "Идентификатор шаблона, по которому выпущен сертификат", "format": "uuid"}, "templateName": {"type": "string", "description": "Название шаблона, по которому выпущен сертификат"}, "type": {"type": "string", "description": "Перечисление типов сертификата", "enum": ["CERTIFICATE", "ROOT_CA", "SUB_CA", "UNKNOWN"]}, "validFrom": {"type": "string", "description": "Дата начала действия сертификата", "format": "date-time"}, "validTo": {"type": "string", "description": "Дата окончания действия сертификата", "format": "date-time"}, "status": {"type": "string", "description": "Перечисление: статус сертификата", "enum": ["ACTIVE", "HOLD", "REVOKE", "REQUEST", "UNKNOWN"]}, "actions": {"$ref": "#/components/schemas/CertificateDownloadActionsPublic"}, "publicKey": {"type": "string", "description": "Публичный ключ сертификата"}, "certificateType": {"type": "string", "description": "Тип сертификата"}, "version": {"type": "integer", "description": "Версия сертификата", "format": "int32"}, "subjectKeyIdentifier": {"type": "string", "description": "Идентификатор ключа сертификата"}, "authorityKeyIdentifier": {"type": "string", "description": "Идентификатор ключа издателя"}, "keyUsages": {"type": "array", "description": "Разрешенные использования ключа", "items": {"$ref": "#/components/schemas/KeyUsageDictResponsePublic"}}, "extendedKeyUsages": {"type": "array", "description": "Разрешенные расширенные использования ключа", "items": {"$ref": "#/components/schemas/ExtendedKeyUsageDictResponsePublic"}}, "ca": {"type": "boolean", "description": "Флаг: сертификат ЦС"}, "aiaUrls": {"type": "array", "description": "Точки распространения AIA", "items": {"type": "string", "description": "Точки распространения AIA"}}, "ocspUrls": {"type": "array", "description": "URL сервисов OCSP", "items": {"type": "string", "description": "URL сервисов OCSP"}}, "crlUrls": {"type": "array", "description": "Точки распространения CRL", "items": {"type": "string", "description": "Точки распространения CRL"}}, "deltaCrlUrls": {"type": "array", "description": "Точки распространения Delta CRL", "items": {"type": "string", "description": "Точки распространения Delta CRL"}}, "description": "Ответ сервиса: данные"

```

```

сертификата"}, "ExtendedKeyUsageDictResponsePublic": {"type": "object", "properties": {"id": {"type":
: "string", "description": "Перечисление расширенного использования
ключа", "enum": ["EKU_PKIX_ANY_EXTENDED_KEY_USAGE", "CSN_369791_TLS_CLIENT", "CSN_369791_TLS_SERVE
R", "CLIENT_AUTHENTICATION", "CODE_SIGNING", "EAP_OVER_LAN", "EAP_OVER_PPP", "ETSI_TSL_SIGNING", "EM
AIL_PROTECTION", "ICAO_DEVIATION_LIST_SIGNING", "EKU_INTEL_AMT", "INTERNET_KEY_EXCHANGE_FOR_IPSEC
", "KERBEROS_CLIENT_AUTHENTICATION", "EKU_KRB_PKINIT_KDC", "MS_COMMERCIAL_CODE_SIGNING", "MS_DOCUM
ENT_SIGNING", "MS_EFS_RECOVERY", "MS_ENCRYPTED_FILE_SYSTEM", "MS_INDIVIDUAL_CODE_SIGNING", "MS_SMA
RT_CARD_LOGON", "OCSP_SIGNER", "EKU_ADOBE_PDF_SIGNING", "PIV_CARD_AUTHENTICATION", "SCVP_CLIENT", "
SCVP_SERVER", "SIP_DOMAIN", "EKU_PKIX_SSH_CLIENT", "SSH_SERVER", "SERVER_AUTHENTICATION", "TIME_STA
MPING", "ICAO_MASTER_LIST_SIGNING", "UNKNOWN"]}, "value": {"type": "string", "description": "Наименов
ание элемента"}, "oid": {"type": "string", "description": "Идентификатор
объекта"}, "description": {"type": "string", "description": "Описание использования
ключа"}}, "description": "Объект ответа сервиса, содержащий данные расширенного использования
ключа"}, "ItemResponseCertificateResponsePublic": {"type": "object", "properties": {"status": {"type
": "integer", "format": "int32"}, "data": {"$ref": "#/components/schemas/CertificateResponsePublic"}
}, "description": "Стандартизированный объект ответа сервиса, содержащий один единственный
элемент"}, "KeyUsageDictResponsePublic": {"type": "object", "properties": {"id": {"type": "string", "d
escription": "Перечисление использования
ключа", "enum": ["DIGITAL_SIGNATURE", "NON_REPUDIATION", "KEY_ENCIPHERMENT", "DATA_ENCIPHERMENT", "K
EY_AGREEMENT", "KEY_CERT_SIGN", "CRL_SIGN", "ENCIPHER_ONLY", "DECIPHER_ONLY", "UNKNOWN"]}, "value": {
"type": "string", "description": "Наименование
элемента"}, "description": {"type": "string", "description": "Описание использования
ключа"}}, "description": "Объект ответа сервиса, содержащий данные использования
ключа"}, "CollectionResponseSubjectResponsePublic": {"type": "object", "properties": {"status": {"ty
pe": "integer", "format": "int32"}, "data": {"$ref": "#/components/schemas/DataSubjectResponsePublic
"}}, "description": "Стандартизированный объект ответа сервиса, содержащий коллекцию
элементов"}, "DataSubjectResponsePublic": {"type": "object", "properties": {"range": {"$ref": "#/comp
onents/schemas/Range"}, "items": {"type": "array", "description": "Коллекция
элементов"}, "items": {"$ref": "#/components/schemas/SubjectResponsePublic"}}, "description": "Поле
зная нагрузка с коллекцией
элементов"}, "ResourceItemResponsePublic": {"type": "object", "properties": {"id": {"type": "string",
"description": "ID ресурсной
системы", "format": "uuid"}, "commonName": {"type": "string", "description": "Общее наименование (CN)
ресурсной системы"}, "distinguishedName": {"type": "string", "description": "Отличающиеся имя
ресурсной системы"}, "description": "Ответ: ресурсная система
(сокращенный)"}, "SubjectNameFieldItemResponsePublic": {"type": "object", "properties": {"values": {
"type": "array", "description": "Значения атрибута
субъекта", "items": {"type": "string", "description": "Значения атрибута
субъекта"}}, "editable": {"type": "boolean", "description": "Флаг: редактируемый
атрибут"}, "description": "Ответ: данные атрибута
субъекта"}, "SubjectResponsePublic": {"type": "object", "properties": {"id": {"type": "string", "descr
iption": "ID субъекта", "format": "uuid"}, "commonName": {"type": "string", "description": "Общее
наименование (CN) субъекта"}, "distinguishedName": {"type": "string", "description": "Отличающиеся
наименование (DN)
субъекта"}, "resource": {"$ref": "#/components/schemas/ResourceItemResponsePublic"}, "subjectName"
: {"type": "object", "additionalProperties": {"$ref": "#/components/schemas/SubjectNameFieldItemRes
ponsePublic"}, "description": "Отличительное имя
субъекта"}, "subjectAltName": {"type": "object", "additionalProperties": {"$ref": "#/components/sche
mas/SubjectNameFieldItemResponsePublic"}, "description": "Дополнительное имя
субъекта"}, "isConnected": {"type": "boolean", "description": "Флаг: подключение субъекта к
ресурсной системе"}, "isBlocked": {"type": "boolean", "description": "Флаг: блокировка субъекта в
ресурсной системе"}, "modify": {"type": "string", "description": "Дата и время обновления в
ресурсной системе", "format": "date-
time"}, "certificatesCount": {"type": "integer", "description": "Количество
сертификатов", "format": "int64"}, "updated": {"type": "string", "description": "Дата и время
обновления", "format": "date-time"}, "created": {"type": "string", "description": "Дата и время
создания", "format": "date-time"}}, "description": "Ответ:
субъект"}, "ItemResponseSubjectResponsePublic": {"type": "object", "properties": {"status": {"type":
"integer", "format": "int32"}, "data": {"$ref": "#/components/schemas/SubjectResponsePublic"}}, "des
cription": "Стандартизированный объект ответа сервиса, содержащий один единственный
элемент"}, "GetApiVersionResponse": {"type": "object", "properties": {"status": {"type": "integer", "f
ormat": "int32"}, "data": {"$ref": "#/components/schemas/VersionResponse"}}, "VersionResponse": {"t
ype": "object", "properties": {"version": {"type": "string", "description": "Версия
модуля"}}, "description": "Объект ответа, содержащий версию модуля"}, "securitySchemes": {"X-API-

```



```
KEY":{"type":"apiKey","description":"API Key","name":"X-API-KEY","in":"header"},"token":{"type":"apiKey","description":"Token","name":"token","in":"header"},"X-AECA-Principal-ID":{"type":"apiKey","description":"UUID","name":"X-AECA-Principal-ID","in":"header"},"X-AECA-Is-Account":{"type":"apiKey","description":"Is Account","name":"X-AECA-Is-Account","in":"header"},"X-AECA-Principal-Name":{"type":"apiKey","description":"Principal name","name":"X-AECA-Principal-Name","in":"header"},"X-AECA-User-Role":{"type":"apiKey","description":"Role","name":"X-AECA-User-Role","in":"header"},"X-AECA-User-Authorities":{"type":"apiKey","description":"Authorities","name":"X-AECA-User-Authorities","in":"header"},"X-User-Session-Id":{"type":"apiKey","description":"Session ID","name":"X-User-Session-Id","in":"header"}}}}
```

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ОС	–	Операционная система
ПО	–	Программное обеспечение
ЦР		Центр регистрации
ЦС	–	Центр сертификации
API	–	Application Programming Interface
CRL	–	Certificate Revocation List
AIA	–	Authority Information Access
URL	–	Uniform Resource Locator

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]