



Центр сертификатов доступа

Aladdin Enterprise Certificate Authority Certified Edition

Руководство администратора. Часть 3.
Описание методов REST API
«Центра сертификации Aladdin Enterprise Certification Authority»

Изделие	RU.АЛДЕ.03.01.020
Документ	RU.АЛДЕ.03.01.020 32 01-3
Версия	2.4
Листов	190
Дата	28.05.2026

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—2026. 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 предназначается 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.), подтвержденные или включенные 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](#).

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 возникнуть 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 представление 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 нелицензионным 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 незамедлительно вернёте в Компанию все экземпляры ПО и все копии такового 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 реэкспорт 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 УСТАНАВЛИВАТЬ И НЕ ИСПОЛЬЗОВАТЬ ДАННОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.

Содержание

1	Описание методов REST API версии 2	10
1.1	Методы идентификации и аутентификации.....	10
1.1.1	Метод идентификации и аутентификации по сертификату доступа	10
1.1.2	Метод обновления маркера доступа	10
1.2	Методы работы с субъектами.....	10
1.2.1	Метод поиска субъектов	10
1.2.2	Метод получения субъекта по идентификатору	11
1.2.3	Метод создания и изменения субъекта	12
1.2.4	Методы создания и изменения субъекта на основании запроса pkcs#10	14
1.2.5	Метод удаления субъекта.....	16
1.3	Методы работы с шаблонами сертификатов.....	16
1.3.1	Метод поиска шаблонов	17
1.3.2	Метод получения шаблона по идентификатору	18
1.4	Методы работы с Центрами сертификации.....	19
1.4.1	Метод получения активного ЦС	19
1.4.2	Метод получения ЦС по идентификатору	21
1.4.3	Метод получения Центров сертификации.....	23
1.5	Методы работы с сертификатами	25
1.5.1	Метод выпуска сертификата в контейнере pkcs#12	25
1.5.2	Методы выпуска сертификата по запросу pkcs#10.....	27
1.5.3	Методы валидации запроса pkcs#10	31
1.5.4	Метод поиска сертификатов.....	33
1.5.5	Метод получения сертификата по идентификатору	35
1.5.6	Метод получения сертификата по серийному номеру	37
1.5.7	Метод получения сертификата по его отпечатку	39
1.5.8	Метод отзыва (приостановки) сертификата по идентификатору.....	41
1.5.9	Метод активации сертификата по идентификатору	42
1.5.10	Метод публикации сертификата в РС по идентификатору	42
1.6	Методы экспорта файлов.....	43
1.6.1	Метод получения сертификата по идентификатору сертификата	43
1.6.2	Метод получения запроса на сертификат по идентификатору сертификата	43
1.6.3	Метод получения цепочки сертификата по идентификатору сертификата	43
1.6.4	Метод получения контейнера PKCS #12 по идентификатору сертификата	44
1.6.5	Метод получения сертификата Центра сертификации по идентификатору Центра сертификации	44
1.6.6	Метод получения цепочки сертификатов Центра сертификации по идентификатору Центра сертификации	44
1.6.7	Метод получения CRL по идентификатору Центра сертификации	45
1.6.8	Метод получения DeltaCRL по идентификатору Центра сертификации.....	45
1.7	Методы работы с точками распространения	45
1.7.1	Метод генерации и публикации CRL по идентификатору Центра сертификации ..	45
1.7.2	Метод генерации и публикации CRL по идентификатору Центра сертификации (устаревший).....	46
1.8	Методы работы с точками подключения и ресурсными системами	46
1.8.1	Метод поиска зарегистрированных ресурсных систем	46
1.8.2	Метод получения ресурсной системы по идентификатору	46
1.8.3	Метод полной синхронизации ресурсной системы	47
1.8.4	Метод поиска точек подключения	47
1.8.5	Метод получения точки подключения по идентификатору	48
1.8.6	Метод частичной синхронизации точки подключения	48
1.9	Метод получения версии сервиса внешних интеграций	49
1.10	Методы работы с Syslog-серверами.....	49

1.10.1	Метод поиска Syslog-серверов.....	49
1.10.2	Метод получения Syslog-сервера по идентификатору.....	49
1.10.3	Метод создания Syslog-сервера	50
1.10.4	Метод обновления Syslog-сервера.....	50
1.10.5	Метод деактивации Syslog-сервера	50
1.10.6	Метод активации Syslog-сервера.....	51
1.10.7	Метод удаления Syslog-сервера.....	51
2	Описание методов REST API версии 3	52
2.1	Методы идентификации и аутентификации.....	52
2.1.1	Метод идентификации и аутентификации по сертификату доступа	52
2.1.2	Метод обновления маркера доступа	52
2.1.3	Метод обновления последней активности учётной записи.....	52
2.1.4	Метод аутентификации по Kerberos-ticket.....	53
2.1.5	Метод аутентификации по логину и паролю	53
2.2	Методы работы с лицензией.....	53
2.2.1	Метод получения информации о возможности использования сторонних ключевых носителей в соответствии с параметрами лицензии.....	53
2.3	Методы работы с субъектами.....	54
2.3.1	Метод поиска субъектов	54
2.3.2	Метод получения субъекта по идентификатору	55
2.3.3	Метод создания и изменения субъекта	56
2.3.4	Методы создания и изменения субъекта на основании запроса pkcs#10	57
2.3.5	Метод удаления субъекта.....	60
2.3.6	Метод поиска идентификаторов субъектов.....	60
2.4	Методы работы с шаблонами сертификатов.....	61
2.4.1	Метод поиска шаблонов	61
2.4.2	Метод получения шаблона по идентификатору	62
2.5	Методы работы с Центрами сертификации.....	64
2.5.1	Метод получения активного Центра сертификации	64
2.5.2	Метод получения Центра сертификации по идентификатору	65
2.5.3	Метод получения Центров сертификации.....	67
2.6	Методы работы с сертификатами	70
2.6.1	Метод выпуска сертификата в контейнере pkcs#12	70
2.6.2	Методы выпуска сертификата по запросу pkcs#10.....	72
2.6.3	Методы перевыпуска сертификата по запросу pkcs#10.....	76
2.6.4	Методы валидации запроса pkcs#10	80
2.6.5	Метод поиска сертификатов.....	83
2.6.6	Метод получения сертификата по идентификатору	85
2.6.7	Метод получения сертификата по серийному номеру	87
2.6.8	Метод получения сертификата по его отпечатку.....	89
2.6.9	Метод отзыва (приостановки) сертификата по идентификатору.....	91
2.6.10	Метод активации сертификата по идентификатору	92
2.6.11	Метод публикации сертификата в РС по идентификатору	92
2.6.12	Метод расшифровки контейнера сертификата.....	92
2.6.13	Метод расшифровки сертификата.....	93
2.6.14	Метод расшифровки запроса на сертификат	95
2.7	Методы экспорта файлов.....	96
2.7.1	Метод получения сертификата по идентификатору сертификата	96
2.7.2	Метод получения запроса на сертификат по идентификатору сертификата	97
2.7.3	Метод получения цепочки сертификата по идентификатору сертификата	97
2.7.4	Метод получения контейнера PKCS#12 по идентификатору сертификата	97
2.7.5	Метод получения сертификата Центра сертификации по идентификатору.....	98
2.7.6	Метод получения цепочки сертификатов Центра сертификации по идентификатору	98

2.7.7	Метод получения CRL по идентификатору Центра сертификации	98
2.7.8	Метод получения DeltaCRL по идентификатору Центра сертификации	99
2.8	Методы работы с точками распространения	99
2.8.1	Метод генерации и публикации CRL по идентификатору Центра сертификации ..	99
2.8.2	Метод генерации и публикации CRL по идентификатору Центра сертификации (устаревший)	99
2.9	Методы работы с точками подключения и ресурсными системами	100
2.9.1	Метод поиска зарегистрированных ресурсных систем	100
2.9.2	Метод получения ресурсной системы по идентификатору	100
2.9.3	Метод полной синхронизации ресурсной системы	101
2.9.4	Метод поиска идентификаторов ресурсных систем	101
2.9.5	Метод поиска точек подключения к ресурсной системе	101
2.9.6	Метод получения точки подключения по идентификатору	102
2.9.7	Метод частичной синхронизации точки подключения	103
2.10	Методы получения информации о сервисах	103
2.10.1	Методы получения информации о сервисе безопасности (security-service) ...	103
2.10.2	Методы получения информации о сервисе лицензий (license-service)	105
2.10.3	Методы получения информации о сервисе журнала событий (logs-service) ...	107
2.10.4	Методы получения информации о сервисе сертификатов (certificate-authority-service)	109
2.10.5	Методы получения информации о сервисе настроек (settings-service)	111
2.10.6	Методы получения информации о сервисе хранения данных (storage-service)	113
2.10.7	Методы получения информации о сервисе оповещения пользователей (event-delivery-service)	115
2.10.8	Методы получения информации о сервисе внешних интеграций (external-integration-service)	117
2.11	Методы работы с Syslog-серверами	119
2.11.1	Метод поиска Syslog-серверов	119
2.11.2	Метод получения Syslog-сервера по идентификатору	119
2.11.3	Метод создания Syslog-сервера	120
2.11.4	Метод обновления Syslog-сервера	120
2.11.5	Метод деактивации Syslog-сервера	120
2.11.6	Метод активации Syslog-сервера	121
2.11.7	Метод удаления Syslog-сервера	121
2.12	Методы работы с учетными записями	122
2.12.1	Метод поиска учетных записей	122
2.12.2	Метод получения учетной записи по идентификатору	122
2.12.3	Метод получения учетной записи по отпечатку сертификата	122
2.12.4	Метод получения учетной записи по идентификатору субъекта	123
2.13	Методы работы с издателями	124
2.13.1	Метод поиска издателей	124
2.14	Методы работы с группами безопасности	125
2.14.1	Метод поиска групп безопасности	125
2.14.2	Метод получения группы безопасности по идентификатору	125
2.15	Методы работы с центрами валидации	127
2.15.1	Метод проверки доступности центра валидации	127
2.15.2	Метод регистрации центра валидации	127
2.15.3	Метод удаления центра валидации	128
2.15.4	Метод создания службы OCSP	128
2.15.5	Метод удаления службы OCSP	128
3	Описание Prometheus-метрик для REST API версии 3	129
3.1	Базовые метрики сервиса	129
3.1.1	Время запуска	129

3.2	Метрики диска	129
3.3	Метрики исполнителей (Thread Pools)	129
3.3.1	taskExecutor (пул асинхронных задач)	129
3.3.2	taskScheduler (пул планировщика задач)	129
3.4	Метрики пула подключений к БД (HikariCP)	130
3.4.1	Основные метрики пула	130
3.5	Метрики HTTP-клиента	130
3.5.1	Активные клиентские запросы	130
3.5.2	Завершенные клиентские запросы	130
3.6	Метрики HTTP-сервера	131
3.6.1	Активные серверные запросы	131
3.6.2	Завершенные серверные запросы	131
3.7	JDBC-метрики (альтернативное представление HikariCP)	131
3.8	Метрики JVM (Java Virtual Machine)	131
3.8.1	Общая информация	131
3.8.2	Буферы	131
3.8.3	Классы	131
3.8.4	Компиляция	132
3.8.5	Сборка мусора	132
3.8.6	Память (выделенная)	132
3.8.7	Память (максимальная)	132
3.8.8	Память (после сборки мусора)	132
3.8.9	Память (используемая)	132
3.8.10	Потоки	132
3.9	Метрики логирования (Logback)	132
3.10	Метрики процесса	132
3.11	Метрики Spring Data Repository	133
3.12	Метрики безопасности (Spring Security)	133
3.12.1	Активная авторизация	133
3.12.2	Завершенная авторизация	133
3.12.3	Счетчики прохождения фильтров безопасности (часть 1)	133
3.12.4	Активные фильтры безопасности	134
3.12.5	Счетчики прохождения фильтров безопасности (часть 2)	134
3.12.6	Время выполнения фильтров	134
3.12.7	Счетчики прохождения фильтров безопасности (часть 3)	135
3.12.8	Защищенные запросы	135
3.12.9	Незащищенные запросы	135
3.13	Системные метрики CPU	135
3.14	Метрики планировщика задач	136
3.14.1	Активные задачи	136
3.14.2	Завершенные задачи	136
3.15	Метрики Tomcat-сессий	136
4	Описание методов REST API версии 4	137
4.1	Методы работы с субъектами	137
4.1.1	Метод поиска субъектов	137
4.1.2	Метод получения субъекта по идентификатору	138
4.1.3	Метод создания и изменения субъекта	139
4.1.4	Методы создания и изменения субъекта на основании запроса pkcs#10	140
4.2	Методы работы с шаблонами сертификатов	143
4.2.1	Метод поиска шаблонов	143
4.2.2	Метод получения шаблона по идентификатору	144
4.3	Методы работы с Центрами сертификации	147
4.3.1	Метод получения активного ЦС	147
4.3.2	Метод получения ЦС по идентификатору	148

4.3.3 Метод получения Центров сертификации	150
4.4 Методы работы с сертификатами	153
4.4.1 Метод выпуска сертификата в контейнере pkcs#12	153
4.4.2 Методы выпуска сертификата по запросу pkcs#10.....	154
4.4.3 Методы перевыпуска сертификата по запросу pkcs#10.....	158
4.4.4 Методы валидации запроса pkcs#10	163
4.4.5 Метод поиска сертификатов.....	166
4.4.6 Метод получения сертификата по идентификатору	168
4.4.7 Метод получения сертификата по серийному номеру	169
4.4.8 Метод получения сертификата по его отпечатку	171
4.4.9 Метод расшифровки контейнера сертификата	173
4.4.10 Метод расшифровки сертификата.....	174
4.4.11 Метод расшифровки запроса на сертификат	176
4.4.12 Метод выпуска короткоживущего (short-lived, throwaway) сертификата в контейнере PKCS#12	177
4.4.13 Методы выпуска короткоживущего (short-lived, throwaway) сертификата на основании запроса PKCS#10	178
4.5 Методы работы с точками подключения.....	180
4.5.1 Метод поиска точек подключения	180
4.5.2 Метод получения точки подключения по идентификатору	181
4.5.3 Метод частичной синхронизации точки подключения	181
4.6 Методы работы с Syslog-серверами	181
4.6.1 Метод поиска Syslog-серверов.....	181
4.6.2 Метод получения Syslog-сервера по идентификатору	182
4.6.3 Метод создания Syslog-сервера	182
4.6.4 Метод обновления Syslog-сервера	182
4.6.5 Метод деактивации Syslog-сервера.....	183
4.6.6 Метод активации Syslog-сервера.....	183
4.6.7 Метод удаления Syslog-сервера	183
5 Дополнительные возможности	184
6 Диаграмма последовательности получения сертификата по запросу PKCS№10.....	185
Обозначения и сокращения	189
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	190

1 ОПИСАНИЕ МЕТОДОВ REST API ВЕРСИИ 2

1.1 Методы идентификации и аутентификации

1.1.1 Метод идентификации и аутентификации по сертификату доступа

POST API – Аутентификация с помощью сертификата	
URL – x509-provider-service/api/v2/public/auth/sign-in/x509	
Swagger: отсутствует, так как метод реализован на уровне прокси (отсутствует в реализации сервиса)	
Query -	
Request -	
Response ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

1.1.2 Метод обновления маркера доступа

POST API – Обновления токена доступа	
Метод доступен администратору и оператору	
URL – security-service/api/v2/public/auth/refresh-token	
Swagger: https://HOST/security-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Авторизации/refreshToken_1	
Query -	
Request -	
Response ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

1.2 Методы работы с субъектами

1.2.1 Метод поиска субъектов

GET API – Поиск субъектов	
Метод доступен администратору и оператору. В ответе для оператора содержатся только те субъекты, на просмотр или управление которых ему предоставлены полномочия.	
В ответе данного метода будут отсутствовать субъекты, у которых присутствуют атрибуты «Role» (ROLE), «Дата рождения» (DATEOFBIRTH) или «Место рождения» (PLACEOFBIRTH). Данные атрибуты поддерживаются в публичном API начиная с версии v4.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv2%5D%20Контроллер%3A%20субъекты/findAll_1	
URL – subjects-service/api/v2/public/subjects	
Query {	
search (string) [опционально],	Полнотекстовый поиск (имя субъекта)
isBlocked (boolean) [опционально],	Флаг: субъект заблокирован в ресурсной системе
isConnected (boolean) [опционально],	Флаг: субъект подключен к ресурсной системе
id (UUID[]) [опционально],	ID субъекта

securityGroupId (UUID[])[опционально],	ID группы безопасности
organizationalUnitId (UUID[])[опционально],	ID структурной единицы
resourceId (UUID[])[опционально],	ID ресурсной системы
sortDirection (string)[опционально],	Направления сортировки (ASC;DESC)
sortBy (string[])[опционально],	Список полей, к которым применяется сортировка
pageOffset (integer)[опционально],	Смещение от начала списка (пагинация)
pageLimit (integer)[опционально]	Ограничение на размер выборки (пагинация)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
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): {	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): {	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
certificatesCount (integer),	Количество действующих сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.2.2 Метод получения субъекта по идентификатору

GET API – Получение субъекта по идентификатору
Метод доступен администратору и оператору при наличии полномочий на просмотр или управление субъектом, идентификатор которого передается во входных параметрах
При попытке получения субъекта, у которого присутствуют атрибуты «Role» (ROLE), «Дата рождения» (DATEOFBIRTH) или «Место рождения» (PLACEOFBIRTH), данный метод вернет ошибку с кодом 400 и сообщением

«Запрошенный объект не поддерживается данной версией API». Данные атрибуты поддерживаются в публичном API начиная с версии v4.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%3A%20субъекты/findById_1	
URL – subjects-service/api/v2/public/subjects/{id}	
Query	
{	
id (UUID)	ID субъекта
}	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
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): {	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): {	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
certificatesCount (integer),	Количество действующих сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.2.3 Метод создания и изменения субъекта

PUT API – Создания и изменения субъекта
Метод доступен администратору и оператору при наличии полномочий на управление субъектами
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%3A%20субъекты/update
URL – subjects-service/api/v2/public/subjects

Query	
Request	
{	
id (UUID) [опционально],	Идентификатор субъекта
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): string[] } [опционально],	Поля разделенного имени субъекта
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): string[] } [опционально]	Поля альтернативного имени субъекта
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
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): {	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): {	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
certificatesCount (integer),	Количество действующих сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)

created (instant)	Время создания (ISO 8601)
}	

1.2.4 Методы создания и изменения субъекта на основании запроса pkcs#10

1.2.4.1 Метод создания и изменения субъекта на основании запроса pkcs#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

PUT API – Создания и изменения субъекта на основании запроса pkcs#10 (multipart/form-data)	
Метод доступен администратору и оператору при наличии полномочий на управление субъектами	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv2%5D%20Контроллер%3A%20субъекты/updateByPkcs10AsMultipartFile_1	
URL – subjects-service/api/v2/public/subjects/pkcs10	
Query	
-	
Request	
{	
id (UUID) [опционально],	Идентификатор субъекта
request (binary),	Файл запроса на сертификат (см. пример использования метода ниже). Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
subjectAltName: { (enum: RFC822NAME, DNS_NAME, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD): string[] } [опционально]	Поля альтернативного имени субъекта
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
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): {	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,	Поля альтернативного имени субъекта

MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD): {	
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
certificatesCount (integer),	Количество действующих сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.2.4.2 Метод создания и изменения субъекта на основании запроса pkcs#10 (формат запроса на сертификат из входных параметров – application/json)

Примечание: отличается от метода выше форматом передаваемого во входных параметрах запроса на сертификат. В данном методе на вход поступает содержимое файла запроса на сертификат в формате PEM (содержимое в Base64).

PUT API – Создания и изменения субъекта на основании запроса pkcs#10 (application/json)	
Метод доступен администратору и оператору при наличии полномочий на управление субъектами	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv2%5D%20Контроллер%3A%20субъекты/updateByPkcs10AsMultipartFile_1	
URL – subjects-service/api/v2/public/subjects/pkcs10	
Query	
-	
Request	
{	
Id (UUID) [опционально],	Идентификатор субъекта
request: {	Файл запроса на сертификат
contentType (string),	Тип загружаемого файла (HTTP MediaType)
fileName (string),	Имя загружаемого файла
data (byte[])	Содержимое PEM файла запроса на сертификат (массив байт в Base64) – см. пример использования метода ниже. Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE-----").
},	
subjectAltName: {	Поля альтернативного имени субъекта
(enum: RFC822NAME, IPADDRESS, DNS_NAME, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD): string[]	
} [опционально]	
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
commonName (string)	Имя субъекта
distinguishedName (string),	Расположение субъекта в ресурсной системе
resource: {	Ресурсы
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): {	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): {	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
certificatesCount (integer),	Количество действующих сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.2.5 Метод удаления субъекта

DELETE API – Удаление субъекта	
Метод доступен администратору и оператору при наличии полномочий на управление субъектами локальной PC	
URL – subjects-service/api/v2/public/subjects/{id}	
Swagger: https://HOST/subjects-service/swagger/swagger-ui/index.html#/Контроллер%3A%20субъекты/deleteById	
Query {(id – обязательный параметр)}	
id (UUID)	Идентификатор субъекта
}	
Request -	
Response -	

1.3 Методы работы с шаблонами сертификатов

Сопоставление идентификаторов шаблонов	
Идентификатор 1.2.0	Идентификатор 2.x
100001	9129245a-eaad-4ebc-a2a4-8845ac0336fb
100002	af3b0355-1798-4c64-98f7-a9c70407db1c
100003	bf2dac0a-f05f-49dd-95b4-e50691489b6a
100004	aa03e458-50cd-46b8-82cd-d5612ed3b647

Сопоставление идентификаторов шаблонов	
Идентификатор 1.2.0	Идентификатор 2.x
100005	aac2e49b-9c8e-4869-80c1-eef526ba75ab
100006	059a38f5-f345-4275-b79f-e7e6cc3cbb68
100007	08c66f99-218a-46ef-bdee-6a2b3b26a4f1
100008	0c234243-18cf-4c05-b699-537731b2436f
100009	11ec34a4-d03e-4059-92f0-9c09b08bffeaa
100010	18d9bd4e-6f15-423f-8137-ac8416ad6874

1.3.1 Метод поиска шаблонов

GET API – Поиск шаблонов	
<p>Метод доступен администратору и оператору.</p> <p>В ответе для оператора содержатся только те шаблоны, на использование которых ему предоставлены полномочия.</p> <p>Данный метод не возвращает шаблоны, в которых указан конкретный ЦС – в ответе будут содержаться только шаблоны со значением «Любой» в параметре «Центр сертификации».</p> <p>В ответе данного метода будут отсутствовать шаблоны, у которых присутствуют поля «Role» (ROLE), «Дата рождения» (DATEOFBIRTH) или «Место рождения» (PLACEOFBIRTH). Данные атрибуты поддерживаются в публичном API начиная с версии v4.</p>	
URL – templates-service/api/v2/public/templates	
Swagger: https://HOST/templates-service/swagger/swagger-ui/index.html#/Контроллер%3A%20шаблоны/findAll_2	
Query	
{	
types (enum[]: EMBEDDED, CLONED, IMPORTED) [опционально],	Тип шаблона
certificateType (enum[]: CERTIFICATE, ROOT CA, SUB CA) [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
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID шаблона
name (string),	Имя шаблона
type (enum: EMBEDDED, CLONED, IMPORTED),	Тип шаблона
certificateType (enum: CERTIFICATE, ROOT CA, SUB CA),	Тип выпускаемого сертификата
certificateCount (int64),	Число выпущенных по шаблону сертификатов
removed (boolean),	Флаг: шаблон удален
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.3.2 Метод получения шаблона по идентификатору

GET API – Получение шаблона по идентификатору	
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на использование шаблона, идентификатор которого передается во входных параметрах. <p>Данный метод не возвращает шаблон, в котором указан конкретный ЦС – по идентификатору может быть получен только шаблон со значением «Любой» в параметре «Центр сертификации». Для других шаблонов ответ метода будет иметь код 400 с сообщением «Запрошенный объект не поддерживается данной версией API».</p> <p>При попытке получения шаблона, у которого присутствуют поля «Role» (ROLE), «Дата рождения» (DATEOFBIRTH) или «Место рождения» (PLACEOFBIRTH), данный метод вернет ошибку с кодом 400 и сообщением «Запрошенный объект не поддерживается данной версией API». Данные поля поддерживаются в публичном API начиная с версии v4.</p>	
URL – templates-service/api/v2/public/templates/{id}	
Swagger: https://HOST/templates-service/swagger/swagger-ui/index.html#/Контроллер%3A%20шаблоны/findById_2	
Query	
{	
id (UUID)	ID шаблона
}	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID шаблона
name (string),	Имя шаблона
type (enum: EMBEDDED, CLONED, IMPORTED),	Тип шаблона
certificateType (enum: CERTIFICATE, ROOT_CA, SUB_CA),	Тип выпускаемого сертификата
removed (boolean),	Флаг: шаблон удален
validity (int64),	Время действия выпускаемого сертификата (мс)
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_ENCIPHERMENT, DATA_ENCIPHERMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCIPHER_ONLY, DECIPHER_ONLY])	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
defaultValue (string),	Значение по умолчанию
regex (string),	Регулярное значение для валидации значения
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, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE)	Код компонента
}],	
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
defaultValue (string),	Значение по умолчанию
regex (string),	Регулярное значение для валидации значения
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),	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)
}	

1.4 Методы работы с Центрами сертификации

1.4.1 Метод получения активного ЦС

GET API – Получение активного ЦС	
Метод доступен администратору и оператору.	
Если активным является центр сертификации, у которого криптопровайдером алгоритма ГОСТ Р 34.10-2012 является Aladdin JCP, данный метод ошибку с кодом 400 и сообщением «Запрошенный объект не поддерживается данной версией API».	
Если активным является центр сертификации, у которого в SDN или в SDN издателя его сертификата присутствуют компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) или «PLACEOFBIRTH» (место рождения), данный метод вернет ошибку с кодом 400 и сообщением «Запрошенный объект не поддерживается данной версией API». Данные компоненты поддерживаются в публичном API начиная с версии v4.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html/#/%5Bv2%5D%20Контроллер%3A%20Центры%20сертификации/active	
URL – certificate-authority-service/api/v2/public/certificate-authorities/active	
Query	
-	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID ЦС

isActive (boolean),	Флаг: активный ЦС
active (boolean),	Флаг: активный ЦС
isManagement (boolean),	Флаг: технологический ЦС
management (boolean),	Флаг: технологический ЦС
certificate: {	Сертификат ЦС
id (UUID),	Идентификатор сертификата ЦС
issuerId (UUID),	Идентификатор издателя сертификата ЦС
issuerFingerprint (string),	Фингерпринт издателя сертификата ЦС
serialnumber (string),	Серийный номер сертификата ЦС
fingerprint (string),	Фингерпринт сертификата ЦС
issuerDN: {	Имя субъекта издателя сертификата ЦС
(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): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectDN: {	Имя субъекта сертификата ЦС
(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): 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 ЦС
(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): string[]	Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
name (string),	Имя сертификата ЦС (на основе CN)
templateId (UUID),	Идентификатор шаблона
templateName (string),	Имя шаблона
validFrom (instant),	Дата начала действия сертификата ЦС (ISO 8601)
validTo (instant),	Дата окончания действия сертификата ЦС (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST),	Статус сертификата ЦС
isValid (boolean),	Флаг: сертификат ЦС действует
isExpired (boolean),	Флаг: сертификат ЦС истек
actions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int32),	Код причины отзыва
value (string)	Значение причины отзыва
},	
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
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата ЦС
keyBits (int32),	Длина ключа сертификата ЦС
subjectKeyIdentifier (string),	Идентификатор ключа сертификата ЦС
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
},	

chain: {	Цепочка сертификатов ЦС (рекурсивный объект)
id (UUID),	Идентификатор сертификата
name (string),	Имя сертификата (на основе CN)
subjectDN: {	Имя субъекта сертификата ЦС
(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): 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)
},	
certificateCount (int64),	Число выпущенных сертификатов
title (string),	Отображаемое имя ЦС
cryptographyProviders: {	Конфигурация криптопровайдеров алгоритмов ЦС
(enum: RSA, ECDSA, GOST R 34 10 2012): {	Название алгоритма
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
primaryCryptographyProvider: {	Криптопровайдер закрытого ключа
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
isAvailable (boolean),	Флаг: Доступность ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.4.2 Метод получения ЦС по идентификатору

GET API – Получение ЦС по идентификатору	
Метод доступен администратору.	
При попытке получения центра сертификации, у которого криптопровайдером алгоритма ГОСТ Р 34.10-2012 является Aladdin JCP, данный метод ошибку с кодом 400 и сообщением «Запрошенный объект не поддерживается данной версией API».	
При попытке получения центра сертификации, у которого в SDN или в SDN издателя его сертификата присутствуют компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) или «PLACEOFBIRTH» (место рождения), данный метод вернет ошибку с кодом 400 и сообщением «Запрошенный объект не поддерживается данной версией API». Данные компоненты поддерживаются в публичном API начиная с версии v4.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%3A%20Центры%20сертификации/findById_6	
URL – certificate-authority-service/api/v2/public/certificate-authorities/{id}	
Query	
{	
id (UUID)	ID ЦС
}	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID ЦС
isActive (boolean),	Флаг: активный ЦС
active (boolean),	Флаг: активный ЦС
isManagement (boolean),	Флаг: технологический ЦС

management (boolean),	Флаг: технологический ЦС
certificate: {	Сертификат ЦС
id (UUID),	Идентификатор сертификата ЦС
issuerId (UUID),	Идентификатор издателя сертификата ЦС
issuerFingerprint (string),	Фингерпринт издателя сертификата ЦС
serialnumber (string),	Серийный номер сертификата ЦС
fingerprint (string),	Фингерпринт сертификата ЦС
issuerDN: {	Имя субъекта издателя сертификата ЦС
(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): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectDN: {	Имя субъекта сертификата ЦС
(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): 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 ЦС
(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): string[]	Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
name (string),	Имя сертификата ЦС (на основе CN)
templateId (UUID),	Идентификатор шаблона
templateName (string),	Имя шаблона
validFrom (instant),	Дата начала действия сертификата ЦС (ISO 8601)
validTo (instant),	Дата окончания действия сертификата ЦС (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST),	Статус сертификата ЦС
isValid (boolean),	Флаг: сертификат ЦС действует
isExpired (boolean),	Флаг: сертификат ЦС истек
actions: {	Доступные действия по выгрузке
pl2 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int32),	Код причины отзыва
value (string)	Значение причины отзыва
},	
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
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата ЦС
keyBits (int32),	Длина ключа сертификата ЦС
subjectKeyIdentifier (string),	Идентификатор ключа сертификата ЦС
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
},	
chain: {	Цепочка сертификатов ЦС (рекурсивный объект)
id (UUID),	Идентификатор сертификата

name (string),	Имя сертификата (на основе CN)
subjectDN: {	Имя субъекта сертификата ЦС
(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): 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)
},	
certificateCount (int64),	Число выпущенных сертификатов
title (string),	Отображаемое имя ЦС
cryptographyProviders: {	Конфигурация криптопровайдеров алгоритмов ЦС
(enum: RSA, ECDSA, GOST_R_34_10_2012): {	Название алгоритма
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
primaryCryptographyProvider: {	Криптопровайдер закрытого ключа
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
isAvailable (boolean),	Флаг: Доступность ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.4.3 Метод получения Центров сертификации

GET API – Получение ЦС	
Метод доступен администратору.	
В ответе метода не будут содержаться центры сертификации, у которых криптопровайдером алгоритма ГОСТ Р 34.10-2012 является Aladdin JCP.	
В ответе метода будут отсутствовать центры сертификации, у которых в SDN или в SDN издателей их сертификатов присутствуют компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) или «PLACEOFBIRTH» (место рождения). Данные компоненты поддерживаются в публичном API начиная с версии v4.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%3A%20Центры%20сертификации/findAll_6	
URL – certificate-authority-service/api/v2/public/certificate-authorities	
Query	
{	
status (enum[:ACTIVE, HOLD, REVOKE, REQUEST] [опционально],	Статус сертификата ЦС
type (enum[: CERTIFICATE1, ROOT_CA, SUB_CA] [опционально],	Тип сертификата ЦС
search (string) [опционально],	Полнотекстовый поиск по имени ЦС
isManagement (boolean) [опционально],	Флаг: технологический ЦС
isActive (boolean) [опционально],	Флаг: активный ЦС
isValid (boolean) [опционально],	Флаг: сертификат ЦС действителен
isExpired (boolean) [опционально],	Флаг: сертификат ЦС истек
notIds (UUID[]) [опционально],	Исключая ID ЦС
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)

1 Тип `CERTIFICATE` является общим для всех словарей типов сертификатов в программе. При использовании данного метода указание данного типа также доступно, однако сертификаты ЦС с данным типом отсутствуют, соответственно не будут найдены и возвращены в ответе.

sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID ЦС
isActive (boolean),	Флаг: активный ЦС
active (boolean),	Флаг: активный ЦС
isManagement (boolean),	Флаг: технологический ЦС
management (boolean),	Флаг: технологический ЦС
certificate: {	Сертификат ЦС
id (UUID),	Идентификатор сертификата ЦС
issuerId (UUID),	Идентификатор издателя сертификата ЦС
issuerFingerprint (string),	Фингерпринт издателя сертификата ЦС
serialnumber (string),	Серийный номер сертификата ЦС
fingerprint (string),	Фингерпринт сертификата ЦС
issuerDN: {	Имя субъекта издателя сертификата ЦС
(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): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key - один из перечисленных в enum параметров, а value - значение параметра
},	
subjectDN: {	Имя субъекта сертификата ЦС
(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): 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 ЦС
(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): string[]	Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key - один из перечисленных в enum параметров, а value - значение параметра
},	
name (string),	Имя сертификата ЦС (на основе CN)
templateId (UUID),	Идентификатор шаблона
templateName (string),	Имя шаблона
validFrom (instant),	Дата начала действия сертификата ЦС (ISO 8601)
validTo (instant),	Дата окончания действия сертификата ЦС (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST),	Статус сертификата ЦС
isValid (boolean),	Флаг: сертификат ЦС действует
isExpired (boolean),	Флаг: сертификат ЦС истек
actions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва

number (int32),	Код причины отзыва
value (string)	Значение причины отзыва
},	
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
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата ЦС
keyBits (int32),	Длина ключа сертификата ЦС
subjectKeyIdentifier (string),	Идентификатор ключа сертификата ЦС
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
},	
chain: {	Цепочка сертификатов ЦС (рекурсивный объект)
id (UUID),	Идентификатор сертификата
name (string),	Имя сертификата (на основе CN)
subjectDN: {	Имя субъекта сертификата ЦС
(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): 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)
},	
certificateCount (int64),	Число выпущенных сертификатов
title (string),	Отображаемое имя ЦС
cryptographyProviders: {	Конфигурация криптопровайдеров алгоритмов ЦС
(enum: RSA, ECDSA, GOST_R_34_10_2012): {	Название алгоритма
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
primaryCryptographyProvider: {	Криптопровайдер закрытого ключа
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
isAvailable (boolean),	Флаг: Доступность ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.5 Методы работы с сертификатами

1.5.1 Метод выпуска сертификата в контейнере pkcs#12

POST API – Выпуск сертификата в контейнере pkcs#12
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление сертификатами субъектов и использование шаблона, идентификаторы которых передаются во входных параметрах. <p>Использование данного метода оператором для создания сертификатов для учетных записей запрещено.</p> <p>В используемом шаблоне должна быть включена опция «Выпуск сертификатов с закрытым ключом (PKCS#12)», иначе метод вернет сообщение об ошибке с кодом 400 и текстом «Выпуск сертификатов с закрытым ключом (PKCS#12) недоступен по данному шаблону».</p>

<p>Указываемый во входных параметрах пароль от контейнера должен соответствовать требованиям регулярного выражения по шаблону, иначе метод вернет сообщение об ошибке с кодом 400 и текстом «Пароль не соответствует регулярному выражению, указанному в шаблоне».</p>	
<p>Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv2%5D%20Контроллер%3A%20сертификаты/enrollByCald</p>	
<p>URL – <code>certificate-authority-service/api/v2/public/certificates/enroll/{cald}</code></p>	
<p>Query</p>	
<pre>{ caId (UUID), subjectId (UUID) [обязателен, если не указан userId], userId (UUID) [обязателен, если не указан subjectId] }</pre>	<p>ID ЦС</p> <p>ID субъекта</p> <p>ID учетной записи</p>
<p>Request</p>	
<pre>{ templateId (UUID), 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): string[] }, 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): string[] }, keyBits (integer), keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN), password (string) }</pre>	<p>Идентификатор шаблона¹</p> <p>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. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p> <p>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. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p> <p>Длина ключа</p> <p>Алгоритм ключевой пары сертификата</p> <p>Пароль контейнера</p>
<p>Response</p>	
<p>ResponseEntity -> ItemResponse -> {</p>	
<pre> id (UUID), downloadActions: { p12 (boolean), csr (boolean), pem (boolean) }, fingerprint (string), serialnumber (string), templateId (UUID), templateName (string), name (string), issuerDN: { (enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME,</pre>	<p>Ответ JSON в HTTP-body</p> <p>ID сертификата</p> <p>Доступные действия по загрузке</p> <p>Флаг: выгрузка pkcs12</p> <p>Флаг: выгрузка pkcs10</p> <p>Флаг: выгрузка сертификата</p> <p>Фингерпринт шаблона</p> <p>Серийный номер сертификата</p> <p>ID шаблона</p> <p>Имя шаблона</p> <p>Имя сертификата (на основе CN)</p> <p>Поля разделенного имени субъекта издателя из сертификата. В формате key-value.</p>

¹ Шаблоны в eCA-CA 2.2 содержат поле «Центр сертификации», определяющее ЦС, на котором должен быть издан сертификат. В случае, если для указанного в поле «templateId» шаблона задан ЦС, отличный от указанного в поле «cald», ответ данного метода будет иметь код 500 и будет содержать сообщение об ошибке «Шаблон {идентификатор шаблона} не может быть использован для выпуска сертификата на центре сертификации {идентификатор центра сертификации из поля «cald»}. При использовании шаблона, в котором в поле «Центр сертификации» указано значение «Любой», выпуск сертификата будет происходить на ЦС, указанном в поле «cald».

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): string[]	Где key – один из перечисленных в enum параметров, а value – значение параметра
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): string[]	Поля разделенного имени субъекта из сертификата. 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, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD): string[]	Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant)	Дата окончания действия сертификата (ISO 8601)
}	

1.5.2 Методы выпуска сертификата по запросу pkcs#10

1.5.2.1 Выпуск сертификата по запросу pkcs#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

POST API – Выпуск сертификата в по запросу pkcs#10 (multipart/form-data)	
Метод доступен:	
<ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление сертификатами субъектов и использование шаблона, идентификаторы которых передаются во входных параметрах. 	
Использование данного метода оператором для создания сертификатов для учетных записей запрещено.	
URL – certificate-authority-service/api/v2/public/certificates/enroll/{caId}/pkcs10	
Swagger: https://HOST/certificate-authority-service/swagger/swagger-ui/index.html#/Контроллер%3A%20сертификаты/enrollRequestByCald_1_1	
Query	
{	
caId (UUID),	ID ЦС
subjectId (UUID) [обязателен, если не указан accountId],	ID субъекта
accountId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	

templateId (UUID),	Идентификатор шаблона ¹
request (binary),	<p>Файл запроса на сертификат (см. пример использования метода ниже).</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» значения полей запроса на сертификат должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p> <p>Допустимые форматы запроса на сертификат:</p> <ul style="list-style-type: none"> • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE-----").
<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): string[] } [опционально]</pre>	<p>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.</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p>
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID сертификата
downloadActions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
fingerprint (string),	Фингерпринт шаблона
serialnumber (string),	Серийный номер сертификата
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
name (string),	Имя сертификата (на основе CN)
<pre>issuerDN: { (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): string[] },</pre>	<p>Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра</p>
<pre>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): string[] },</pre>	<p>Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра</p>

¹ Шаблоны в eCA-CA 2.2 содержат поле «Центр сертификации», определяющее ЦС, на котором должен быть издан сертификат. В случае, если для указанного в поле «templateId» шаблона задан ЦС, отличный от указанного в поле «cald», ответ данного метода будет иметь код 500 и будет содержать сообщение об ошибке «Шаблон {идентификатор шаблона} не может быть использован для выпуска сертификата на центре сертификации {идентификатор центра сертификации из поля «cald»}. При использовании шаблона, в котором в поле «Центр сертификации» указано значение «Любой», выпуск сертификата будет происходить на ЦС, указанном в поле «cald».

<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): string[] }, validFrom (instant), validTo (instant) }</pre>	<p>Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра</p> <p>Дата начала действия сертификата (ISO 8601)</p> <p>Дата окончания действия сертификата (ISO 8601)</p>
--	--

1.5.2.2 Выпуск сертификата по запросу pkcs#10 (формат запроса на сертификат из входных параметров – application/json)

Примечание: отличается от метода выше форматом передаваемого во входных параметрах запроса на сертификат. В данном методе на вход поступает содержимое файла запроса на сертификат в формате PEM (содержимое в Base64).

POST API – Выпуск сертификата в по запросу pkcs#10 (application/json)	
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление сертификатами субъектов и использование шаблона, идентификаторы которых передаются во входных параметрах. <p>Использование данного метода оператором для создания сертификатов для учетных записей запрещено.</p>	
URL – certificate-authority-service/api/v2/public/certificates/enroll/{cald}/pkcs10	
Swagger: https://HOST/certificate-authority-service/swagger/swagger-ui/index.html#/Контроллер%3A%20сертификаты/enrollRequestByCald_1_1	
Query	
{	
caId (UUID),	ID ЦС
subjectId (UUID) [обязателен, если не указан accountId],	ID субъекта
accountId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона ¹
request: {	Запрос на сертификат
contentType(string),	Тип загружаемого файла (HTTP MediaType)
fileName (string),	Имя загружаемого файла
data (byte[])	Содержимое PEM файла запроса на сертификат (массив байт в Base64) – см. пример использования метода ниже. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» значения полей запроса на сертификат должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с

¹ Шаблоны в eCA-CA 2.2 содержат поле «Центр сертификации», определяющее ЦС, на котором должен быть издан сертификат. В случае, если для указанного в поле «templateId» шаблона задан ЦС, отличный от указанного в поле «cald», ответ данного метода будет иметь код 500 и будет содержать сообщение об ошибке «Шаблон {идентификатор шаблона} не может быть использован для выпуска сертификата на центре сертификации {идентификатор центра сертификации из поля «cald»}. При использовании шаблона, в котором в поле «Центр сертификации» указано значение «Любой», выпуск сертификата будет происходить на ЦС, указанном в поле «cald».

	регулярными выражениями полей) значения, не соответствующие атрибутам субъекта. Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE-----").
},	
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): string[] } [опционально]	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. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.
}	
Response ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID сертификата
downloadActions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
fingerprint (string),	Фингерпринт шаблона
serialnumber (string),	Серийный номер сертификата
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
name (string),	Имя сертификата (на основе CN)
issuerDN: { (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): string[] },	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
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): string[] },	Поля разделенного имени субъекта из сертификата. 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, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD): string[] },	Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant)	Дата окончания действия сертификата (ISO 8601)
}	

1.5.3 Методы валидации запроса pkcs#10

1.5.3.1 Метод валидации запроса pkcs#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

POST API – Валидация запроса pkcs#10 (multipart/form-data)	
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление сертификатами субъектов и использование шаблона, идентификаторы которых передаются во входных параметрах. <p>Использование данного метода оператором для валидации запросов на сертификат для учетных записей запрещено.</p>	
URL – certificate-authority-service/api/v2/public/certificates/validate/{caId}/pkcs10	
Swagger: https://HOST/certificate-authority-service/swagger/swagger-ui/index.html#/Контроллер%3A%20сертификаты/validateAsFileDescription_1_1	
Query	
{	
caId (UUID)	ID ЦС
subjectId (UUID) [обязателен, если не указан accountId],	ID субъекта
accountId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона
request (binary),	<p>Файл запроса на сертификат.</p> <p>Допустимые форматы запроса на сертификат:</p> <ul style="list-style-type: none"> • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE-----").
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): string[] }	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
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
name (string),	Имя сертификата (на основе CN)
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
subjectId (UUID)	ID субъекта (может отсутствовать, если в Query указан accountId, а не subjectId)
valid (boolean),	Флаг: запрос прошел валидацию
subjectNames: [{	Компоненты имени субъекта сертификата
fieldName (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),	Имя компонента
fieldValue (string),	Значение компонента
required (boolean),	Флаг: компонент обязательный
additional (boolean),	Флаг: компонент дополнительный
valid (boolean),	Флаг: компонент прошел валидацию
message (string)	Дополнительное сообщение
}],	

subjectAltNames: [{	Компоненты расширенного имени субъекта сертификата
fieldName (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),	Имя компонента
fieldValue (string),	Значение компонента
required (boolean),	Флаг: компонент обязательный
additional (boolean),	Флаг: компонент дополнительный
valid (boolean),	Флаг: компонент прошел валидацию
message (string)	Дополнительное сообщение
}]	
}	

1.5.3.2 Метод валидации запроса pkcs#10 (формат запроса на сертификат из входных параметров – application/json)

Примечание: отличается от метода выше форматом передаваемого во входных параметрах запроса на сертификат. В данном методе на вход поступает содержимое файла запроса на сертификат в формате PEM (содержимое в Base64).

POST API – Валидация запроса pkcs#10 (application/json)	
Метод доступен:	
<ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление сертификатами субъектов и использование шаблона, идентификаторы которых передаются во входных параметрах. 	
Использование данного метода оператором для валидации запросов на сертификат для учетных записей запрещено.	
URL – certificate-authority-service/api/v2/public/certificates/validate/{caId}/pkcs10	
Swagger: https://HOST/certificate-authority-service/swagger/swagger-ui/index.html#/Контроллер%3A%20сертификаты/validateAsFileDescription_1_1	
Query	
{	
caId (UUID)	ID ЦС
subjectId (UUID) [обязателен, если не указан accountId],	ID субъекта
accountId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона
request: {	Файл запроса на сертификат
contentType (string),	Тип загружаемого файла (HTTP MediaType)
fileName (string),	Имя загружаемого файла
data (byte[])	Содержимое загружаемого файла (массив байт). Допустимые форматы запроса на сертификат: <ul style="list-style-type: none"> • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
},	
subjectAltName: {	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
(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) :	
string[]	

}	
}	
Response ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
name (string),	Имя сертификата (на основе CN)
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
subjectId (UUID)	ID субъекта (может отсутствовать, если в Query указан accountId, а не subjectId)
valid (boolean),	Флаг: запрос прошел валидацию
subjectNames: [{	Компоненты имени субъекта сертификата
fieldName (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),	Имя компонента
fieldValue (string),	Значение компонента
required (boolean),	Флаг: компонент обязательный
additional (boolean),	Флаг: компонент дополнительный
valid (boolean),	Флаг: компонент прошел валидацию
message (string)	Дополнительное сообщение
}],	
subjectAltNames: [{	Компоненты расширенного имени субъекта сертификата
fieldName (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),	Имя компонента
fieldValue (string),	Значение компонента
required (boolean),	Флаг: компонент обязательный
additional (boolean),	Флаг: компонент дополнительный
valid (boolean),	Флаг: компонент прошел валидацию
message (string)	Дополнительное сообщение
}]	
}	

1.5.4 Метод поиска сертификатов

GET API – Поиск сертификатов	
Метод доступен администратору и оператору.	
В ответе для оператора содержатся только сертификаты субъектов, к которым ему предоставлен доступ.	
В ответе метода не будут содержаться сертификаты, в которых в SDN или в SDN их издателей присутствуют компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) или «PLACEOFBIRTH» (место рождения). Данные компоненты поддерживаются в публичном API начиная с версии v4.	
URL – certificate-authority-service/api/v2/public/certificates	
Swagger: https://HOST/certificate-authority-service/swagger/swagger-ui/index.html#/Контроллер%3A%20сертификаты/findAll_1	
Query {	
search (string) [опционально],	Полнотекстовый поиск (имя или серийный номер)
issuerId (UUID) [опционально],	ID сертификата издателя

templateId (UUID) [опционально],	ID шаблона
status (enum[: ACTIVE, HOLD, REVOKE, REQUEST] [опционально],	Статус сертификата
type (enum[:CERTIFICATE, ROOT_CA, SUB_CA] [опционально],	Тип сертификата
revocationReason (enum[:UNSPECIFIED, KEY_COMPROMISE, CA_COMPROMISE, AFFILIATION_CHANGED, SUPERSEDED, CESSATION_OF_OPERATION, CERTIFICATE_HOLD, REMOVE_FROM_CRL, PRIVILEGE_WITHDRAWN, AA_COMPROMISE) [опционально],	Причина отзыва
notRevocationReason (enum[:UNSPECIFIED, KEY_COMPROMISE, CA_COMPROMISE, AFFILIATION_CHANGED, SUPERSEDED, CESSATION_OF_OPERATION, CERTIFICATE_HOLD, REMOVE_FROM_CRL, PRIVILEGE_WITHDRAWN, AA_COMPROMISE) [опционально],	Исключая причину отзыва
revocationDateFrom (instant) [опционально],	Дата отзыва (начало)
revocationDateTo (instant) [опционально],	Дата отзыва (окончание)
hasRevocationReason (boolean) [опционально],	Флаг: наличие причины отзыва
hasRequest (boolean) [опционально],	Флаг: наличие pkcs10
hasCA (boolean) [опционально],	Флаг: наличие ЦС
isManagementCA (boolean) [опционально],	Флаг: технологический ЦС
isValid (boolean) [опционально],	Флаг: сертификат действует
isExpired (boolean) [опционально],	Флаг: сертификат истек
validFromFrom (instant) [опционально],	Дата начала действия (начало)
validFromTo (instant) [опционально],	Дата начала действия (окончание)
validToFrom (instant) [опционально],	Дата окончания действия (начало)
validToTo (instant),	Дата окончания действия (окончание)
subjectId (UUID[]),	ID субъекта
userId (UUID[]),	ID учетной записи
serialnumber (string[]),	Серийный номер
fingerprint (string[]),	Отпечаток
notId (UUID[]),	Исключая ID сертификата
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)
}	
Request -	
Response ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	Идентификатор сертификата
issuerId (UUID),	Идентификатор издателя сертификата
issuerFingerprint (string),	Фингерпринт издателя сертификата
serialnumber (string),	Серийный номер сертификата
fingerprint (string),	Фингерпринт 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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectDN: {	Имя субъекта сертификата
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS,	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра

UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE): string[]	
},	
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): string[]	Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
name (string),	Имя сертификата (на основе CN)
templateId (UUID),	Идентификатор шаблона
templateName (string),	Имя шаблона
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant),	Дата окончания действия сертификата (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST),	Статус сертификата ЦС
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
actions: {	Доступные действия по выгрузке
pl2 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
revocation: {	Сведения об отзыве сертификата
date (instant),	Дата отзыва
number (int32),	Код причины отзыва
value (string)	Значение причины отзыва
},	
hashAlgorithm (enum: SHA1, SHA256, SHA384, SHA512, SHA3_256, SHA3_384, SHA3_512, RSASSA_PSS, MD5, GOST_R_34_11_2012, UNKNOWN),	Алгоритм подписи сертификата
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата
keyBits (int4),	Длина ключа сертификата
subjectKeyIdentifier (string),	Идентификатор ключа сертификата
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.5.5 Метод получения сертификата по идентификатору

GET API – Получение сертификата по идентификатору	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах.	
При попытке получения сертификата, у которого в SDN или в SDN его издателя присутствуют компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) или «PLACEOFBIRTH» (место рождения), данный метод вернет ошибку с кодом 400 и сообщением «Запрошенный объект не поддерживается данной версией API». Данные компоненты поддерживаются в публичном API начиная с версии v4.	
URL – certificate-authority-service/api/v2/public/certificates/{id}	
Swagger: https://HOST/certificate-authority-service/swagger/swagger-ui/index.html#/Контроллер%3A%20сертификаты/getByld	
Query	
{	
id (UUID)	ID сертификата
}	
Request	
-	

Response	Ответ JSON в HTTP-body
ResponseEntity -> CollectionResponse -> {	
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): 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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key - один из перечисленных в enum параметров, а value - значение параметра
},	
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): 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): 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
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA),	Тип сертификата
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant),	Дата окончания действия сертификата (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST),	Статус сертификата
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
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

authorityKeyIdIdentifier (string)	Идентификатор ключа издателя сертификата
keyUsages: {	Назначение ключа сертификата
id (uuid),	Идентификатор элемента
code (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPTHERMENT, DATA_ENCRYPTHERMENT, KEY AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPTHER_ONLY, DECRYPTHER_ONLY),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (uuid),	Идентификатор назначения
code (enum: EKU_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, EKU_INTEL_AMT, INTERNET_KEY_EXCHANGE_FOR_IPSEC, KERBEROS_CLIENT_AUTHENTICATION, EKU_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, EKU_ADOBE_PDF_SIGNING, PIV_CARD_AUTHENTICATION, SCVP_CLIENT, SCVP_SERVER, SIP_DOMAIN, EKU_PKIX_SSH_CLIENT, SSH_SERVER, SERVER_AUTHENTICATION, TIME_STAMPING, ICAO_MASTER_LIST_SIGNING),	Перечисление расширенного использования ключа
value (string),	Наименование элемента
oid (string),	OID назначения
description (string)	Описание использования ключа
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int4),	Код причины отзыва
value (string)	Значение причины отзыва
},	
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
userId (uuid),	Идентификатор субъекта
subjectId (uuid),	Идентификатор субъекта
created (instant)	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
}	

1.5.6 Метод получения сертификата по серийному номеру

GET API – Получение сертификата по его серийному номеру
<p>Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, серийный номер которого передается во входных параметрах.</p> <p>При попытке получения сертификата, у которого в SDN или в SDN его издателя присутствуют компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) или «PLACEOFBIRTH» (место рождения), данный метод вернет ошибку с кодом 400 и сообщением «Запрошенный объект не поддерживается данной версией API». Данные компоненты поддерживаются в публичном API начиная с версии v4.</p>
URL – certificate-authority-service/api/v2/public/certificates/serialNumber/{serialNumber}
Swagger: https://HOST/certificate-authority-service/swagger/swagger-ui/index.html#/Контроллер%3A%20сертификаты/getBySerialNumber

Query	
{	
serialnumber (String)	Серийный номер сертификата (формат: 40 символов, нижний регистр)
}	
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): 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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key - один из перечисленных в enum параметров, а value - значение параметра
},	
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): 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): 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
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA),	Тип сертификата
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant),	Дата окончания действия сертификата (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST),	Статус сертификата
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
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
id (uuid),	Идентификатор элемента
code (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPTMENT, DATA_ENCRYPTMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (uuid),	Идентификатор элемента
code (enum: EKU_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, EKU_INTEL_AMT, INTERNET_KEY_EXCHANGE_FOR_IPSEC, KERBEROS_CLIENT_AUTHENTICATION, EKU_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, OSCP_SIGNER, EKU_ADOBE_PDF_SIGNING, PIV_CARD_AUTHENTICATION, SCVP_CLIENT, SCVP_SERVER, SIP_DOMAIN, EKU_PKIX_SSH_CLIENT, SSH_SERVER, SERVER_AUTHENTICATION, TIME_STAMPING, ICAO_MASTER_LIST_SIGNING),	Перечисление расширенного использования ключа
value (string),	Наименование элемента
oid (string),	OID назначения
description (string)	Описание использования ключа
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int4),	Код причины отзыва
value (string)	Значение причины отзыва
},	
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
userId (uuid),	Идентификатор субъекта
subjectId (uuid),	Идентификатор субъекта
created (instant)	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
}	

1.5.7 Метод получения сертификата по его отпечатку

GET API – Получение сертификата по его отпечатку
<p>Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, отпечаток которого передается во входных параметрах.</p> <p>При попытке получения сертификата, у которого в SDN или в SDN его издателя присутствуют компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) или «PLACEOFBIRTH» (место рождения), данный метод вернет ошибку с кодом 400</p>

и сообщением «Запрошенный объект не поддерживается данной версией API». Данные компоненты поддерживаются в публичном API начиная с версии v4.	
URL – certificate-authority-service/api/v2/public/certificates/fingerprint/{fingerprint}	
Swagger: https://HOST/certificate-authority-service/swagger/swagger-ui/index.html#/Контроллер%3A%20сертификаты/getByFingerprint_1	
Query	
{	
fingerprint (String)	Отпечаток сертификата
}	
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): 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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
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): 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): 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
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA),	Тип сертификата
validFrom (instant),	Дата начала действия сертификата (ISO 8601)

validTo (instant),	Дата окончания действия сертификата (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST),	Статус сертификата
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
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
id (uuid),	Идентификатор элемента
code (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPTMENT, DATA_ENCRYPTMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (uuid),	Идентификатор элемента
code (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),	Перечисление расширенного использования ключа
value (string),	Наименование элемента
oid (string),	OID назначения
description (string)	Описание использования ключа
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int4),	Код причины отзыва
value (string)	Значение причины отзыва
},	
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
userId (uuid),	Идентификатор субъекта
subjectId (uuid),	Идентификатор субъекта
created (instant)	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
}	

1.5.8 Метод отзыва (приостановки) сертификата по идентификатору

PUT API – Отзыв (приостановка) сертификата по идентификатору

Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – certificate-authority-service/api/v2/public/certificates/{id}/revoke	
Swagger: https://HOST/certificate-authority-service/swagger/swagger-ui/index.html#/Контроллер%3A%20сертификаты/revokeById_2	
Query	
{	
id (UUID)	ID сертификата
}	
Request	
{	
reason (enum: UNSPECIFIED, KEY_COMPROMISE, CA_COMPROMISE, AFFILIATION_CHANGED, SUPERSEDED, CESSATION_OF_OPERATION, CERTIFICATE_HOLD, REMOVE_FROM_CRL, PRIVILEGE_WITHDRAWN, AA_COMPROMISE)	Причина отзыва (приостановки)
}	
Response	
–	
При указании значения «CERTIFICATE_HOLD» в параметре reason сертификат будет приостановлен, а не отозван.	

1.5.9 Метод активации сертификата по идентификатору

PUT API – Активация сертификата по идентификатору	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – certificate-authority-service/api/v2/public/certificates/{id}/reactivate	
Swagger: https://HOST/certificate-authority-service/swagger/swagger-ui/index.html#/Контроллер%3A%20сертификаты/reactivateById_1	
Query	
{	
id (UUID)	ID сертификата
}	
Request	
–	
Response	
–	

1.5.10 Метод публикации сертификата в РС по идентификатору

PUT API – Публикация сертификата в ресурсную систему	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – certificate-authority-service/api/v2/public/certificates/{id}/publish	
Swagger: https://HOST/certificate-authority-service/swagger/swagger-ui/index.html#/Контроллер%3A%20сертификаты/publishById_1	
Query	
{	
id (uuid)	Идентификатор сертификата
}	

Request	
-	
Response	
-	

1.6 Методы экспорта файлов

1.6.1 Метод получения сертификата по идентификатору сертификата

GET API – Получение сертификата по идентификатору сертификата	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – export-service/api/v2/public/export/certificates/{certificateId}/certificate	
Swagger: https://HOST/export-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findCertificateById	
Query	
{	
certificateId (UUID)	ID сертификата
}	
Request	
-	
Response	
ResponseEntity->byte[]	

1.6.2 Метод получения запроса на сертификат по идентификатору сертификата

GET API – Получение запроса на сертификат по идентификатору сертификата	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – export-service/api/v2/public/export/certificates/{certificateId}/pkcs10	
Swagger: https://HOST/export-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findRequestById_2	
Query	
{	
certificateId (UUID)	ID сертификата
}	
Request	
-	
Response	
ResponseEntity->byte[]	

1.6.3 Метод получения цепочки сертификата по идентификатору сертификата

GET API – Получение цепочки сертификатов по идентификатору сертификата	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – export-service/api/v2/public/export/certificates/{certificateId}/chain	
Swagger: https://HOST/export-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findChainById_1	
Query	
{	
certificateId (UUID)	ID сертификата

}	
Request	
-	
Response	
ResponseEntity->byte[]	

1.6.4 Метод получения контейнера PKCS #12 по идентификатору сертификата

GET API – Получение контейнера pkcs12 по идентификатору сертификата	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – export-service/api/v2/public/export/certificates/{certificateId}/pkcs12	
Swagger: https://HOST/export-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findP12ById_1	
Query	
{	
certificateId (UUID)	ID сертификата
}	
Request	
-	
Response	
ResponseEntity->byte[]	

1.6.5 Метод получения сертификата Центра сертификации по идентификатору Центра сертификации

GET API – Получение сертификата по идентификатору Центра сертификации	
Метод доступен администратору	
URL – export-service/api/v2/public/export/certificate-authorities/{caId}/certificate	
Swagger: https://HOST/export-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findCertificateByCald_2	
Query	
{	
caId (UUID)	ID ЦС
}	
Request	
-	
Response	
ResponseEntity->byte[]	

1.6.6 Метод получения цепочки сертификатов Центра сертификации по идентификатору Центра сертификации

GET API – Получение цепочки сертификатов по идентификатору Центра сертификации	
Метод доступен администратору	
URL – export-service/api/v2/public/export/certificate-authorities/{caId}/chain	
Swagger: https://HOST/export-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findChainByCald_2	
Query	
{	
caId (UUID)	ID ЦС
}	

Request
-
Response
ResponseEntity->byte[]

1.6.7 Метод получения CRL по идентификатору Центра сертификации

GET API – Получение CRL по идентификатору Центра сертификации
Метод доступен администратору и оператору
URL – export-service/api/v2/public/export/certificate-authorities/{cald}/crl
Swagger: https://HOST/export-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findCrlByCald_1
Query
-
Request
-
Response
ResponseEntity->byte[]

1.6.8 Метод получения DeltaCRL по идентификатору Центра сертификации

GET API – Получение DeltaCRL по идентификатору Центра сертификации
Метод доступен администратору и оператору
URL – export-service/api/v2/public/export/certificate-authorities/{cald}/delta-crl
Swagger: https://HOST/export-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findDeltaCrlByCald_1
Query
{
caId (UUID)
ID ЦС
}
Request
-
Response
ResponseEntity->byte[]

1.7 Методы работы с точками распространения

1.7.1 Метод генерации и публикации CRL по идентификатору Центра сертификации

POST API – Генерация и публикация CRL по идентификатору Центра сертификации
Метод доступен администратору
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv2%5D%20Контроллер%3A%20Конфигурации%20CRL%20для%20ЦС/generate
URL – publisher-service/api/v2/public/certificate-authorities/{cald}/crl-configuration/generate
Query
{
id (UUID)
ID ЦС
}
Request
-
Response
-

1.7.2 Метод генерации и публикации CRL по идентификатору Центра сертификации (устаревший)

POST API – Генерация и публикация CRL по идентификатору Центра сертификации	
Данный метод является устаревшим и будет исключен из публичного API в последующих версиях.	
Метод доступен администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%3A%20Конфигурации%20CRL%20для%20ЦС/generateDeprecated	
URL – publisher-service/api/v2/ui/certificate-authorities/{cald}/crl-configuration/generate	
Query	
{	
id (UUID)	ID ЦС
}	
Request	
-	
Response	
-	

1.8 Методы работы с точками подключения и ресурсными системами

1.8.1 Метод поиска зарегистрированных ресурсных систем

GET API – Поиск ресурсных систем	
Метод доступен администратору и оператору. В ответе для оператора содержатся только те ресурсные системы, к субъектам которых ему предоставлен доступ.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%3A%20ресурсы%20системы/findAll_2	
URL – subjects-service/api/v2/public/resources	
Query	
{	
id (uuid) [опционально],	Фильтр: ID ресурсной системы
securityGroupId (uuid) [опционально],	Фильтр: ID группы безопасности
subjectId (uuid) [опционально],	Фильтр: ID субъекта
organizationalUnitId (uuid) [опционально],	Фильтр: ID подразделения
search (string),	Фильтр: полнотекстовый поиск по отображаемому имени
isConnected (boolean),	Фильтр: подключенная ресурсная система
isDefault (boolean)	Фильтр: ресурсная система по умолчанию
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID ресурсной системы
commonName (string),	Имя ресурсной системы
distinguishedName (string),	DN ресурсной системы
subjectsCount (int64),	Количество субъектов ресурсной системы
isConnected (boolean),	Флаг: ресурсная система подключена
isDefault (boolean),	Флаг: локальная ресурсная система
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.8.2 Метод получения ресурсной системы по идентификатору

GET API – Получение ресурсной системы по идентификатору

Метод доступен администратору и оператору при наличии доступа к субъектам РС, идентификатор которой передается во входных параметрах	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%3A%20ресурсные%20системы/findById_2	
URL – subjects-service/api/v2/public/resources/{id}	
Query	
{	
id (uuid) [опционально]	ID ресурсной системы
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID ресурсной системы
commonName (string),	Имя ресурсной системы
distinguishedName (string),	DN ресурсной системы
subjectsCount (int64),	Количество субъектов ресурсной системы
isConnected (boolean),	Флаг: ресурсная система подключена
isDefault (boolean),	Флаг: локальная ресурсная система
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.8.3 Метод полной синхронизации ресурсной системы

PUT API – Полная синхронизация РС	
Метод доступен администратору и оператору при наличии полномочий на субъектов данной РС	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%3A%20ресурсные%20системы/synchronize	
URL – ldap-service/api/v2/public/resources/{resourceId}/synchronize	
Query	
{	
resourceId (UUID)	ID ресурсной системы
}	
Request	
-	
Response	
-	

1.8.4 Метод поиска точек подключения

GET API – Поиск точек подключения к РС	
Метод доступен администратору и оператору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%3A%20Точки%20подключения%20ресурсных%20систем/findAll_4	
URL – ldap-service/api/v2/public/connection-points	
Query	
{	
id (uuid) [опционально],	Фильтр: ID точки подключения
resourceId (uuid) [опционально],	Фильтр: ID ресурсной системы
search (string),	Фильтр: полнотекстовый поиск по отображаемому имени
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body

id (UUID),	ID точки подключения
title (string),	Имя точки подключения
domainType (enum: SAMBA_DC, MS_AD, RED_ADM, FREE_IPA, ALD_PRO),	Тип точки подключения ¹
connectionAddress (string),	Адрес (хост) подключения
useTls (boolean),	Флаг: использовать TLS при подключении
baseDn (string),	BaseDN точки подключения
username (string),	Имя пользователя ресурсной системы
status (string)	Статус точки подключения
resourceId (UUID),	ID ресурсной системы
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.8.5 Метод получения точки подключения по идентификатору

GET API – Получение точки подключения по идентификатору	
Метод доступен администратору и оператору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv2%5D%20Контроллер%3A%20Точки%20подключения%20ресурсных%20систем/findById_4	
URL – <code>Idap-service/api/v2/public/connection-points/{id}</code>	
Query	
{	
id (uuid)	ID точки подключения
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID точки подключения
title (string),	Имя точки подключения
domainType (enum: SAMBA_DC, MS_AD, RED_ADM, FREE_IPA, ALD_PRO),	Тип точки подключения ²
connectionAddress (string),	Адрес (хост) подключения
useTls (boolean),	Флаг: использовать TLS при подключении
baseDn (string),	BaseDN точки подключения
username (string),	Имя пользователя ресурсной системы
status (string)	Статус точки подключения
resourceId (UUID),	ID ресурсной системы
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.8.6 Метод частичной синхронизации точки подключения

PUT API – Частичная синхронизация точки подключения	
Метод доступен администратору и оператору при наличии полномочий. У оператора есть возможность выполнять синхронизацию с теми внешними ресурсными системами, к субъектам которых у него есть доступ.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv2%5D%20Контроллер%3A%20Точки%20подключения%20ресурсных%20систем/synchronize_1	
URL – <code>Idap-service/api/v2/public/connection-points/{pointId}/synchronize</code>	
Query	

¹ В eCA-CA 2.4 реализована поддержка работы с ресурсной системой «Альт Домен», однако в API v2 при получении точек подключения к данной PC ее тип в поле «domainType» указывается как «SAMBA_DC». В API v3 в поле «domainType» для точек подключения к ресурсной системе «Альт Домен» указывается значение «ALT_DOMAIN».

В eCA-CA 2.4 реализована поддержка работы с ресурсной системой «ROSA Dynamic Directory», однако в API v2 при получении точек подключения к данной PC ее тип в поле «domainType» указывается как «FREE_IPA». Начиная с API v4 в поле «domainType» для точек подключения к ресурсной системе «ROSA Dynamic Directory» указывается значение «ROSA_DD».

² В eCA-CA 2.4 реализована поддержка работы с ресурсной системой «Альт Домен», однако в API v2 при получении точек подключения к данной PC ее тип в поле «domainType» указывается как «SAMBA_DC». В API v3 в поле «domainType» для точек подключения к ресурсной системе «Альт Домен» указывается значение «ALT_DOMAIN».

В eCA-CA 2.4 реализована поддержка работы с ресурсной системой «ROSA Dynamic Directory», однако в API v2 при получении точек подключения к данной PC ее тип в поле «domainType» указывается как «FREE_IPA». Начиная с API v4 в поле «domainType» для точек подключения к ресурсной системе «ROSA Dynamic Directory» указывается значение «ROSA_DD».

{	
pointId (UUID)	ID точки подключения
}	
Request	
-	
Response	
-	

1.9 Метод получения версии сервиса внешних интеграций

GET API – Получение версии сервиса внешних интеграций	
Метод по умолчанию доступен неаутентифицированному пользователю. В конфигурационном файле имеется возможность отключения доступа неаутентифицированному пользователю. При этом аутентифицированному пользователю с любой ролью метод остаётся доступным.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/version-controller/getApiVersion	
URL – external-integration-service/api/version	
Query	
-	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
version (string)	Версия сервиса внешних интеграций
}	

1.10 Методы работы с Syslog-серверами

1.10.1 Метод поиска Syslog-серверов

GET API – Поиск Syslog-серверов	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv2%5D%20Контроллер%3A%20Syslog%20сервера/findAll_3	
URL – logs-service/api/v2/public/syslog	
Query	
-	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID Syslog-сервера
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP),	Протокол Syslog-сервера
active (boolean)	Флаг: состояние настройки публикации событий
}	

1.10.2 Метод получения Syslog-сервера по идентификатору

GET API – Получение Syslog-сервера по идентификатору	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv2%5D%20Контроллер%3A%20Syslog%20сервера/findById_3	
URL – logs-service/api/v2/public/syslog/{id}	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
-	

Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID Syslog-сервера
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP),	Протокол Syslog-сервера
active (boolean)	Флаг: состояние настройки публикации событий
}	

1.10.3 Метод создания Syslog-сервера

POST API – Создание Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%3A%20Syslog%20сервера/create	
URL – logs-service/api/v2/public/syslog	
Query	
-	
Request	
{	
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP)	Протокол Syslog-сервера
}	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID Syslog-сервера
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP),	Протокол Syslog-сервера
active (boolean)	Флаг: состояние настройки публикации событий
}	

1.10.4 Метод обновления Syslog-сервера

PUT API – Обновление Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%3A%20Syslog%20сервера/updateById	
URL – logs-service/api/v2/public/syslog/{id}	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
{	
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP)	Протокол Syslog-сервера
}	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID Syslog-сервера
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP),	Протокол Syslog-сервера
active (boolean)	Флаг: состояние настройки публикации событий
}	

1.10.5 Метод деактивации Syslog-сервера

PATCH API – Деактивация Syslog-сервера	
Метод доступен только администратору	

Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv2%5D%20Контроллер%3A%20Syslog%20сервера/deactivate	
URL – logs-service/api/v2/public/syslog/{id}/deactivate	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
–	
Response	
–	

1.10.6 Метод активации Syslog-сервера

PATCH API – Активация Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv2%5D%20Контроллер%3A%20Syslog%20сервера/activate	
URL – logs-service/api/v2/public/syslog/{id}/activate	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
–	
Response	
–	

1.10.7 Метод удаления Syslog-сервера

DELETE API – Удаление Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv2%5D%20Контроллер%3A%20Syslog%20сервера/deleteById_1	
URL – logs-service/api/v2/public/syslog/{id}	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
–	
Response	
–	

2 ОПИСАНИЕ МЕТОДОВ REST API ВЕРСИИ 3

2.1 Методы идентификации и аутентификации

2.1.1 Метод идентификации и аутентификации по сертификату доступа

POST API – Аутентификация с помощью сертификата	
URL – x509-provider-service/api/v3/public/auth/sign-in/x509	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv3%5D%20Контроллер%3A%20Авторизации/signInByX509	
Query -	
Request -	
Response ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

2.1.2 Метод обновления маркера доступа

POST API – Обновление токена доступа	
Метод доступен администратору и оператору	
URL – security-service/api/v3/public/auth/refresh-token	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/[v3] Контроллер%3A Авторизации/refreshToken	
Query -	
Request -	
Response ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

2.1.3 Метод обновления последней активности учётной записи

PUT API – Обновление последней активности учетной записи по ее идентификатору	
Метод доступен только администратору	
URL – security-service/api/v3/public/accounts/{accountId}/activity	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv3%5D%20Контроллер%3A%20Учетные%20записи/updateActivity	
Query {(id – обязательный параметр)}	
id (UUID)	Идентификатор учетной записи
}	
Request -	
Response -	

2.1.4 Метод аутентификации по Kerberos-ticket

POST API – Аутентификация по Kerberos-ticket	
URL – security-service/api/v3/public/auth/sign-in/kerberos	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv3%5D%20Контроллер%3A%20Авторизации/kerberosSignIn	
Query –	
Request –	
Response {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

2.1.5 Метод аутентификации по логину и паролю

POST API – Аутентификация по логину и паролю	
URL – security-service/api/v3/public/auth/sign-in/ldap	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv3%5D%20Контроллер%3A%20Авторизации/ldapSignIn	
Query –	
Request {	
username (string),	Имя пользователя
password (string)	Пароль пользователя
}	
Response {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

2.2 Методы работы с лицензией

2.2.1 Метод получения информации о возможности использования сторонних¹ ключевых носителей в соответствии с параметрами лицензии

GET API – Метод получения информации о возможности использования сторонних ключевых носителей в соответствии с параметрами лицензии
Метод доступен только администратору. Если по текущей лицензии разрешено использование сторонних ключевых носителей, ответ метода будет иметь код 204. Если по текущей лицензии разрешено использование сторонних ключевых носителей, ответ метода будет иметь код 403 и сообщение «Использование сторонних токенов запрещено лицензией».
URL – license-service/api/v3/public/restrictions/third-party-tokens

¹ Отличных от JaCarta.

Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A Лицензионные ограничения/checkThirdPartyTokens
Query -
Request -
Response -

2.3 Методы работы с субъектами

2.3.1 Метод поиска субъектов

GET API – Поиск субъектов	
Метод доступен администратору и оператору. В ответе для оператора содержатся только те субъекты, на просмотр или управление которых ему предоставлены полномочия. В ответе данного метода в поле «subjectName» атрибуты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения) в случае их наличия у субъекта будут указаны как «UNKNOWN». Данные атрибуты поддерживаются в публичном API начиная с версии v4.	
URL – subjects-service/api/v3/public/subjects	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A субъекты/findAll_2	
Query	
{	
search (string) [опционально],	Полнотекстовый поиск (имя субъекта)
isBlocked (boolean) [опционально],	Флаг: субъект заблокирован в ресурсной системе
isConnected (boolean) [опционально],	Флаг: субъект подключен к ресурсной системе
id (UUID[]) [опционально],	ID субъекта
notId (UUID[]) [опционально],	Исключая ID субъекта
securityGroupId (UUID[]) [опционально],	ID группы безопасности
resourceId (UUID[]) [опционально],	ID ресурсной системы
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
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,	Поля разделенного имени субъекта

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
certificatesCount (integer),	Количество сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
sid (string)	SID субъекта
}	

2.3.2 Метод получения субъекта по идентификатору

GET API – Получение субъекта по идентификатору	
Метод доступен администратору и оператору при наличии полномочий на просмотр или управление субъектом, идентификатор которого передается во входных параметрах.	
В ответе данного метода в поле «subjectName» атрибуты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения) в случае их наличия у субъекта будут указаны как «UNKNOWN». Данные атрибуты поддерживаются в публичном API начиная с версии v4.	
URL – subjects-service/api/v3/public/subjects/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A субъекты/findById_2	
Query	
{	
id (UUID)	ID субъекта
}	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
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,	Поля разделенного имени субъекта

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
certificatesCount (integer),	Количество сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
sid (string)	SID субъекта
}	

2.3.3 Метод создания и изменения субъекта

PUT API – Создание и изменение субъекта	
Метод доступен администратору и оператору при наличии полномочий на управление субъектами	
URL – subjects-service/api/v3/public/subjects	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A субъекты/update	
Query	
-	
Request	
{	
id (UUID) [опционально],	Идентификатор субъекта
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],	Поля разделенного имени субъекта
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[] } [опционально]	Поля альтернативного имени субъекта
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
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
certificatesCount (integer),	Количество сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
sid (string)	SID субъекта
}	

2.3.4 Методы создания и изменения субъекта на основании запроса pkcs#10

2.3.4.1 Метод создания и изменения субъекта на основании запроса pkcs#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

PUT API – Создание и изменение субъекта на основании запроса pkcs#10 (multipart/form-data)	
Метод доступен администратору и оператору при наличии полномочий на управление субъектами	
URL – subjects-service/api/v3/public/subjects/pkcs10	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A субъекты/updateByPkcs10AsMultipartFile_1	
Query	
-	
Request	
{	
Id (UUID) [опционально],	Идентификатор субъекта
request (binary),	Файл запроса на сертификат (см. пример использования метода ниже). Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").

<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[] } [опционально] </pre>	Поля альтернативного имени субъекта
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
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
certificatesCount (integer),	Количество сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
sid (string)	SID субъекта
}	

2.3.4.2 Метод создания и изменения субъекта на основании запроса pkcs#10 (формат запроса на сертификат из входных параметров – application/json)

Примечание: отличается от метода выше форматом передаваемого во входных параметрах запроса на сертификат. В данном методе на вход поступает содержимое файла запроса на сертификат в формате PEM (содержимое в Base64).

PUT API – Создание и изменение субъекта на основании запроса pkcs#10 (application/json)
Метод доступен администратору и оператору при наличии полномочий на управление субъектами
URL – subjects-service/api/v3/public/subjects/pkcs10

Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A субъекты/updateByPkcs10AsMultipartFile_1	
Query	
-	
Request	
{	
Id (UUID) [опционально],	Идентификатор субъекта
request: {	Файл запроса на сертификат
contentType (string) [опционально],	Тип загружаемого файла (HTTP MediaType) - application/octet-stream)
fileName (string) [опционально],	Имя загружаемого файла
data (string:binary)	Содержимое PEM файла запроса на сертификат (массив байт в Base64) - см. пример использования метода ниже. Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
},	
subjectAltName: {	Поля альтернативного имени 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): string[]	
} [опционально]	
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
commonName (string)	Имя субъекта
distinguishedName (string),	Расположение субъекта в ресурсной системе
resource: {	Ресурсы
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
certificatesCount (integer),	Количество сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)

sid (string)	SID субъекта
}	

2.3.5 Метод удаления субъекта

DELETE API – Удаление субъекта	
Метод доступен администратору и оператору при наличии полномочий на локальную PC	
URL – subjects-service/api/v3/public/subjects/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A субъекты/deleteById	
Query {(id – обязательный параметр)}	
id (UUID)	Идентификатор субъекта
}	
Request –	
Response –	

2.3.6 Метод поиска идентификаторов субъектов

GET API – Поиск ID субъектов	
Метод доступен администратору и оператору	
URL – subjects-service/api/v3/public/subjects/ids	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A субъекты/findAllAsIds	
Query {	
search (string) [опционально],	Полнотекстовый поиск (имя субъекта)
isBlocked (boolean) [опционально],	Флаг: субъект заблокирован в ресурсной системе
isConnected (boolean) [опционально],	Флаг: субъект подключен к ресурсной системе
id (UUID[]) [опционально],	ID субъекта
notId (UUID[]) [опционально],	Исключая ID субъекта
securityGroupId (UUID[]) [опционально],	ID группы безопасности
resourceId (UUID[]) [опционально],	ID ресурсной системы
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	
Request –	
Response ResponseEntity -> CollectionResponse -> {	
items (UUID)	ID субъекта
}	

2.4 Методы работы с шаблонами сертификатов

Сопоставление идентификаторов шаблонов	
Идентификатор 1.2.0	Идентификатор 2.x
100001	9129245a-eaad-4ebc-a2a4-8845ac0336fb
100002	af3b0355-1798-4c64-98f7-a9c70407db1c
100003	bf2dac0a-f05f-49dd-95b4-e50691489b6a
100004	aa03e458-50cd-46b8-82cd-d5612ed3b647
100005	aac2e49b-9c8e-4869-80c1-eef526ba75ab
100006	059a38f5-f345-4275-b79f-e7e6cc3cbb68
100007	08c66f99-218a-46ef-bdee-6a2b3b26a4f1
100008	0c234243-18cf-4c05-b699-537731b2436f
100009	11ec34a4-d03e-4059-92f0-9c09b08bffeaa
100010	18d9bd4e-6f15-423f-8137-ac8416ad6874

2.4.1 Метод поиска шаблонов

GET API – Поиск шаблонов	
Метод доступен администратору и оператору. В ответе для оператора содержатся только те шаблоны, на использование которых ему предоставлены полномочия	
URL – templates-service/api/v3/public/templates	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3/Контроллер%3А шаблоны/findAll	
Query	
{	
types (enum[]: EMBEDDED, CLONED, IMPORTED, UNKNOWN) [опционально],	Тип шаблона
certificateType (enum[]: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN) [опционально],	Тип выпускаемого сертификата
endEntityType (enum[]: USER, DEVICE, ROOT_CA, SUB_CA, UNKNOWN) [опционально],	Тип субъекта
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 ¹
extendedKeyUsage (string[]) [опционально],	Фильтр: расширенное использование ключа
isCertificateAuthorityIdEmpty (boolean) [опционально],	Фильтр: ID издающего ЦС не задан
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body

¹ 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.

id (UUID),	ID шаблона
name (string),	Имя шаблона
type (enum: EMBEDDED, CLONED, IMPORTED, UNKNOWN),	Тип шаблона
certificateType (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип выпускаемого сертификата
certificateAuthorityId (UUID),	ID ЦС, который должен использоваться при выпуске сертификата по данному шаблону
endEntityType (enum: USER, DEVICE, ROOT_CA, SUB_CA, UNKNOWN),	Тип субъекта
certificateCount (int64),	Число выпущенных по шаблону сертификатов
removed (boolean),	Флаг: шаблон удален
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
appendSubjectSid (boolean)	Флаг: включать SID субъекта в сертификат
}	

2.4.2 Метод получения шаблона по идентификатору

GET API – Получение шаблона по идентификатору	
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на использование шаблона, идентификатор которого передается во входных параметрах. <p>В ответе данного метода в поле «subjectDN» компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения) в случае их наличия в шаблоне будут указаны как «UNKNOWN». Данные компоненты поддерживаются в публичном API начиная с версии v4.</p>	
URL – templates-service/api/v3/public/templates/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A шаблоны/findById	
Query	
{	
id (UUID)	ID шаблона
}	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID шаблона
name (string),	Имя шаблона
type (enum: EMBEDDED, CLONED, IMPORTED, UNKNOWN),	Тип шаблона
certificateType (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип выпускаемого сертификата
certificateAuthorityId (UUID),	ID ЦС, который должен использоваться при выпуске сертификата по данному шаблону
endEntityType (enum: USER, DEVICE, ROOT_CA, SUB_CA, UNKNOWN),	Тип субъекта
removed (boolean),	Флаг: шаблон удален
validity (int64),	Время действия выпускаемого сертификата (мс)
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_ENCIPHERMENT, DATA_ENCIPHERMENT, KEY AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCIPHER_ONLY, DECIPHER_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
defaultValue (string),	Значение по умолчанию
regex (string),	Регулярное значение для валидации значения
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, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN)	Код компонента
}],	
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
defaultValue (string),	Значение по умолчанию
regex (string),	Регулярное значение для валидации значения
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)
appendSubjectSid (boolean),	Флаг: включать SID субъекта в сертификат
publication (boolean)	Флаг: публиковать сертификат в PC
}	

2.5 Методы работы с Центрами сертификации

2.5.1 Метод получения активного Центра сертификации

GET API – Получение активного ЦС	
<p>Метод доступен администратору и оператору.</p> <p>Если активным является центр сертификации, у которого криптопровайдером алгоритма ГОСТ Р 34.10-2012 является Aladdin JCP, в ответе метода данный криптопровайдер будет указан как «UNKNOWN».</p> <p>В ответе данного метода в полях «subjectDN» и «issuerDN» компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения) в случае их наличия в SDN запрашиваемого ЦС или издателя сертификата данного ЦС будут указаны как «UNKNOWN». Данные компоненты поддерживаются в публичном API начиная с версии v4.</p>	
URL – certificate-authority-service/api/v3/public/certificate-authorities/active	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A Центры сертификации/active	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
ResponseEntity -> ItemResponse -> {	
id (UUID),	ID ЦС
isActive (boolean),	Флаг: активный ЦС
active (boolean),	Флаг: активный ЦС
isManagement (boolean),	Флаг: технологический ЦС
management (boolean),	Флаг: технологический ЦС
certificate: {	Сертификат ЦС
id (UUID),	Идентификатор сертификата ЦС
issuerId (UUID),	Идентификатор издателя сертификата ЦС
issuerFingerprint (string),	Фингерпринт издателя сертификата ЦС
serialnumber (string),	Серийный номер сертификата ЦС
fingerprint (string),	Фингерпринт сертификата ЦС
issuerDN: {	Имя субъекта издателя сертификата ЦС
(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 издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectDN: {	Имя субъекта сертификата ЦС
(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 ЦС
(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
},	
name (string),	Имя сертификата ЦС (на основе CN)
templateId (UUID),	Идентификатор шаблона

templateName (string),	Имя шаблона
validFrom (instant),	Дата начала действия сертификата ЦС (ISO 8601)
validTo (instant),	Дата окончания действия сертификата ЦС (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN),	Статус сертификата ЦС
isValid (boolean),	Флаг: сертификат ЦС действует
isExpired (boolean),	Флаг: сертификат ЦС истек
actions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int32),	Код причины отзыва
value (string)	Значение причины отзыва
},	
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
keyAlgorithm (enum: RSA, ECDSA, GOST R 34 10 2012, UNKNOWN),	Алгоритм ключа сертификата ЦС
keyBits (int32),	Длина ключа сертификата ЦС
subjectKeyIdentifier (string),	Идентификатор ключа сертификата ЦС
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
},	
chain: {	Цепочка сертификатов ЦС (рекурсивный объект)
id (UUID),	Идентификатор сертификата
name (string),	Имя сертификата (на основе CN)
subjectDN: {	Имя субъекта сертификата ЦС
(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)
},	
certificateCount (int64),	Число выпущенных сертификатов
title (string),	Отображаемое имя ЦС
cryptographyProviders: {	Конфигурация криптопровайдеров алгоритмов ЦС
(enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN): {	Название алгоритма
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO, UNKNOWN),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
primaryCryptographyProvider: {	Криптопровайдер закрытого ключа
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO, UNKNOWN),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
isAvailable (boolean),	Флаг: Доступность ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

2.5.2 Метод получения Центра сертификации по идентификатору

GET API – Получение ЦС по идентификатору
Метод доступен администратору.
Если с помощью данного метода будет запрошен центр сертификации, у которого криптопровайдером алгоритма ГОСТ Р 34.10-2012 является Aladdin JCP, в ответе метода данный криптопровайдер будет указан как «UNKNOWN».

В ответе данного метода в полях «subjectDN» и «issuerDN» компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения) в случае их наличия в SDN запрашиваемого ЦС или издателя сертификата данного ЦС будут указаны как «UNKNOWN». Данные компоненты поддерживаются в публичном API начиная с версии v4.	
URL – certificate-authority-service/api/v3/public/certificate-authorities/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A Центры сертификации/findById_14	
Query	
{	
id (UUID)	ID ЦС
}	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID ЦС
isActive (boolean),	Флаг: активный ЦС
active (boolean),	Флаг: активный ЦС
isManagement (boolean),	Флаг: технологический ЦС
management (boolean),	Флаг: технологический ЦС
certificate: {	Сертификат ЦС
id (UUID),	Идентификатор сертификата ЦС
issuerId (UUID),	Идентификатор издателя сертификата ЦС
issuerFingerprint (string),	Фингерпринт издателя сертификата ЦС
serialnumber (string),	Серийный номер сертификата ЦС
fingerprint (string),	Фингерпринт сертификата ЦС
issuerDN: {	Имя субъекта издателя сертификата ЦС
(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 издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectDN: {	Имя субъекта сертификата ЦС
(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 ЦС
(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
},	
name (string),	Имя сертификата ЦС (на основе CN)
templateId (UUID),	Идентификатор шаблона
templateName (string),	Имя шаблона
validFrom (instant),	Дата начала действия сертификата ЦС (ISO 8601)
validTo (instant),	Дата окончания действия сертификата ЦС (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN),	Статус сертификата ЦС
isValid (boolean),	Флаг: сертификат ЦС действует
isExpired (boolean),	Флаг: сертификат ЦС истек

actions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int32),	Код причины отзыва
value (string)	Значение причины отзыва
},	
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
keyAlgorithm (enum: RSA, ECDSA, GOST R 34 10 2012, UNKNOWN),	Алгоритм ключа сертификата ЦС
keyBits (int32),	Длина ключа сертификата ЦС
subjectKeyIdentifier (string),	Идентификатор ключа сертификата ЦС
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
},	
chain: {	Цепочка сертификатов ЦС (рекурсивный объект)
id (UUID),	Идентификатор сертификата
name (string),	Имя сертификата (на основе CN)
subjectDN: {	Имя субъекта сертификата ЦС
(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)
},	
certificateCount (int64),	Число выпущенных сертификатов
title (string),	Отображаемое имя ЦС
cryptographyProviders: {	Конфигурация криптопровайдеров алгоритмов ЦС
(enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN): {	Название алгоритма
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO, UNKNOWN),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
primaryCryptographyProvider: {	Криптопровайдер закрытого ключа
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO, UNKNOWN),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
isAvailable (boolean),	Флаг: Доступность ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

2.5.3 Метод получения Центров сертификации

GET API – Получение ЦС
<p>Метод доступен администратору</p> <p>Для центров сертификации, у которых криптопровайдером алгоритма ГОСТ Р 34.10-2012 является Aladdin JCP, в ответе метода данный криптопровайдер будет указан как «UNKNOWN».</p> <p>В ответе данного метода в полях «subjectDN» и «issuerDN» компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения) в случае их наличия в SDN запрашиваемого ЦС или издателя сертификата данного ЦС будут указаны как «UNKNOWN». Данные компоненты поддерживаются в публичном API начиная с версии v4.</p> <p>URL – certificate-authority-service/api/v3/public/certificate-authorities</p>

Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A Центры сертификации/findAll_15	
Query	
{	
status (enum[]:ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN) [опционально],	Статус сертификата ЦС
type (enum[]: CERTIFICATE1, ROOT_CA, SUB_CA, UNKNOWN) [опционально],	Тип сертификата ЦС
search (string) [опционально],	Полнотекстовый поиск по имени ЦС
isManagement (boolean) [опционально],	Флаг: технологический ЦС
isActive (boolean) [опционально],	Флаг: активный ЦС
isValid (boolean) [опционально],	Флаг: сертификат ЦС действителен
isExpired (boolean) [опционально],	Флаг: сертификат ЦС истек
Id (UUID[]) [опционально],	Id ЦС
notIds (UUID[]) [опционально],	Исключая ID ЦС
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
endEntityType (enum[]: ROOT_CA, SUB_CA, USER, DEVICE, UNKNOWN) [опционально],	Тип субъекта
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID ЦС
isActive (boolean),	Флаг: активный ЦС
active (boolean),	Флаг: активный ЦС
isManagement (boolean),	Флаг: технологический ЦС
management (boolean),	Флаг: технологический ЦС
certificate: {	Сертификат ЦС
id (UUID),	Идентификатор сертификата ЦС
issuerId (UUID),	Идентификатор издателя сертификата ЦС
issuerFingerprint (string),	Фингерпринт издателя сертификата ЦС
serialnumber (string),	Серийный номер сертификата ЦС
fingerprint (string),	Фингерпринт сертификата ЦС
issuerDN: {	Имя субъекта издателя сертификата ЦС
(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 издателя из сертификата. В формате key-value. Где key - один из перечисленных в enum параметров, а value - значение параметра
},	
subjectDN: {	Имя субъекта сертификата ЦС
(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

1 Тип `CERTIFICATE` является общим для всех словарей типов сертификатов в программе. При использовании данного метода указание данного типа также доступно, однако сертификаты ЦС с данным типом отсутствуют, соответственно не будут найдены и возвращены в ответе.

},	
subjectAltName: {	Альтернативное имя субъекта сертификата ЦС
(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
},	
name (string),	Имя сертификата ЦС (на основе CN)
templateId (UUID),	Идентификатор шаблона
templateName (string),	Имя шаблона
validFrom (instant),	Дата начала действия сертификата ЦС (ISO 8601)
validTo (instant),	Дата окончания действия сертификата ЦС (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN),	Статус сертификата ЦС
isValid (boolean),	Флаг: сертификат ЦС действует
isExpired (boolean),	Флаг: сертификат ЦС истек
actions: {	Доступные действия по выгрузке
pl2 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int32),	Код причины отзыва
value (string)	Значение причины отзыва
},	
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
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата ЦС
keyBits (int32),	Длина ключа сертификата ЦС
subjectKeyIdentifier (string),	Идентификатор ключа сертификата ЦС
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
},	
chain: {	Цепочка сертификатов ЦС (рекурсивный объект)
id (UUID),	Идентификатор сертификата
name (string),	Имя сертификата (на основе CN)
subjectDN: {	Имя субъекта сертификата ЦС
(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)
},	
certificateCount (int64),	Число выпущенных сертификатов
title (string),	Отображаемое имя ЦС
cryptographyProviders: {	Конфигурация криптопровайдеров алгоритмов ЦС
(enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN): {	Название алгоритма
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO, UNKNOWN),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
primaryCryptographyProvider: {	Криптопровайдер закрытого ключа
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO, UNKNOWN),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
isAvailable (boolean),	Флаг: Доступность ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)

}	
---	--

2.6 Методы работы с сертификатами

2.6.1 Метод выпуска сертификата в контейнере pkcs#12

POST API – Выпуск сертификата в контейнере pkcs#12	
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление сертификатами субъектов и использование шаблона, идентификаторы которых передаются во входных параметрах. <p>Использование данного метода оператором для создания сертификатов для учетных записей запрещено.</p> <p>В используемом шаблоне должна быть включена опция «Выпуск сертификатов с закрытым ключом (PKCS#12)», иначе метод вернет сообщение об ошибке с кодом 400 и текстом «Выпуск сертификатов с закрытым ключом (PKCS#12) недоступен по данному шаблону».</p> <p>Указываемый во входных параметрах пароль от контейнера должен соответствовать требованиям регулярного выражения по шаблону, иначе метод вернет сообщение об ошибке с кодом 400 и текстом «Пароль не соответствует регулярному выражению, указанному в шаблоне».</p>	
URL – certificate-authority-service/api/v3/public/certificates/enroll/{cald}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A сертификаты/enrollByCald	
Query	
{	
caId (UUID),	ID ЦС
subjectId (UUID) [обязателен, если не указан userId],	ID субъекта
userId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона ¹
subjectDN: {	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. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам 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): string[]	
},	

¹ Шаблоны в eCA-CA 2.2 содержат поле «Центр сертификации» (поле «certificateAuthorityId» в API v3), определяющее ЦС, на котором должен быть издан сертификат. В случае, если для указанного в поле «templateId» шаблона задан ЦС, отличный от указанного в поле «cald», ответ данного метода будет иметь код 500 и будет содержать сообщение об ошибке «Шаблон {идентификатор шаблона} не может быть использован для выпуска сертификата на центре сертификации {идентификатор центра сертификации из поля «cald»}. При использовании шаблона, в котором в поле «Центр сертификации» указано значение «Любой», выпуск сертификата будет происходить на ЦС, указанном в поле «cald».

<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>	<p>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. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p>
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 сертификата
downloadActions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
fingerprint (string),	Фингерпринт шаблона
serialnumber (string),	Серийный номер сертификата
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
name (string),	Имя сертификата (на основе CN)
issuerDN: { (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[] },	<p>Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра</p>
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[] },	<p>Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра</p>
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[] },	<p>Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра</p>
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant)	Дата окончания действия сертификата (ISO 8601)
}	

2.6.2 Методы выпуска сертификата по запросу pkcs#10

2.6.2.1 Выпуск сертификата по запросу pkcs#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

POST API – Выпуск сертификата по запросу pkcs#10 (multipart/form-data)	
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление сертификатами субъектов и использование шаблона, идентификаторы которых передаются во входных параметрах. <p>Использование данного метода оператором для создания сертификатов для учетных записей запрещено.</p>	
URL – certificate-authority-service/api/v3/public/certificates/enroll/{cald}/pkcs10	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A сертификаты/enrollRequestByCald_1	
Query	
{	
caId (UUID),	ID ЦС
subjectId (UUID) [обязателен, если не указан userId],	ID субъекта
userId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона ¹
request (binary),	<p>Файл запроса на сертификат (см. пример использования метода ниже).</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» значения полей запроса на сертификат должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p> <p>Допустимые форматы запроса на сертификат:</p> <ul style="list-style-type: none"> • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
<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>	<p>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.</p> <p>Значения полей, указанные в subjectName, переопределяют значения соответствующих полей SDN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта.</p> <p>Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p>

¹ Шаблоны в eCA-CA 2.2 содержат поле «Центр сертификации» (поле «certificateAuthorityId» в API v3), определяющее ЦС, на котором должен быть издан сертификат. В случае, если для указанного в поле «templateId» шаблона задан ЦС, отличный от указанного в поле «cald», ответ данного метода будет иметь код 500 и будет содержать сообщение об ошибке «Шаблон {идентификатор шаблона} не может быть использован для выпуска сертификата на центре сертификации {идентификатор центра сертификации из поля «cald»}. При использовании шаблона, в котором в поле «Центр сертификации» указано значение «Любой», выпуск сертификата будет происходить на ЦС, указанном в поле «cald».

<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[] } [опционально] </pre>	<p>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.</p> <p>Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p>
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID сертификата
downloadActions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
fingerprint (string),	Фингерпринт шаблона
serialnumber (string),	Серийный номер сертификата
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
name (string),	Имя сертификата (на основе CN)
issuerDN: {	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
(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[]	
},	
subjectDN: {	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
(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[]	
},	
subjectAltName: {	Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
(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[]	
},	
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant)	Дата окончания действия сертификата (ISO 8601)
}	

2.6.2.2 Выпуск сертификата по запросу pkcs#10 (формат запроса на сертификат из входных параметров – application/json)

Примечание: отличается от метода выше форматом передаваемого во входных параметрах запроса на сертификат. В данном методе на вход поступает содержимое файла запроса на сертификат в формате PEM (содержимое в Base64).

POST API – Выпуск сертификата по запросу pkcs#10 (application/json)	
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление сертификатами субъектов и использование шаблона, идентификаторы которых передаются во входных параметрах. <p>Использование данного метода оператором для создания сертификатов для учетных записей запрещено.</p>	
URL – certificate-authority-service/api/v3/public/certificates/enroll/{caId}/pkcs10	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A сертификаты/enrollRequestByCald_1	
Query	
{	
caId (UUID) ,	ID ЦС
subjectId (UUID) [обязателен, если не указан userId],	ID субъекта
userId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID) ,	Идентификатор шаблона ¹
request: {	Запрос на сертификат
contentType(string) [опционально],	Тип загружаемого файла (HTTP MediaType) – application/octet-stream)
fileName (string) [опционально],	Имя загружаемого файла
data (string:binary)	Содержимое PEM файла запроса на сертификат (массив байт в Base64) – см. пример использования метода ниже. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» значения полей запроса на сертификат должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта. Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
},	

¹ Шаблоны в eCA-CA 2.2 содержат поле «Центр сертификации» (поле «certificateAuthorityId» в API v3), определяющее ЦС, на котором должен быть издан сертификат. В случае, если для указанного в поле «templateId» шаблона задан ЦС, отличный от указанного в поле «caId», ответ данного метода будет иметь код 500 и будет содержать сообщение об ошибке «Шаблон {идентификатор шаблона} не может быть использован для выпуска сертификата на центре сертификации {идентификатор центра сертификации из поля «caId»}. При использовании шаблона, в котором в поле «Центр сертификации» указано значение «Любой», выпуск сертификата будет происходить на ЦС, указанном в поле «caId».

<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>	<p>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.</p> <p>Значения полей, указанные в subjectName, переопределяют значения соответствующих полей SDN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p>
<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[] } [опционально] </pre>	<p>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.</p> <p>Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p>
<pre> } </pre>	
<p>Response</p> <pre> ResponseEntity -> ItemResponse -> { </pre>	<p>Ответ JSON в HTTP-body</p>
<pre> id (UUID), </pre>	<p>ID сертификата</p>
<pre> downloadActions: { </pre>	<p>Доступные действия по выгрузке</p>
<pre> pl2 (boolean), </pre>	<p>Флаг: выгрузка pkcs12</p>
<pre> csr (boolean), </pre>	<p>Флаг: выгрузка pkcs10</p>
<pre> pem (boolean) </pre>	<p>Флаг: выгрузка сертификата</p>
<pre> }, </pre>	
<pre> fingerprint (string), </pre>	<p>Фингерпринт шаблона</p>
<pre> serialnumber (string), </pre>	<p>Серийный номер сертификата</p>
<pre> templateId (UUID), </pre>	<p>ID шаблона</p>
<pre> templateName (string), </pre>	<p>Имя шаблона</p>
<pre> name (string), </pre>	<p>Имя сертификата (на основе CN)</p>
<pre> issuerDN: { (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[] }, </pre>	<p>Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра</p>
<pre> 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[] }, </pre>	<p>Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра</p>

<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[] }, validFrom (instant), validTo (instant) }</pre>	<p>Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра</p> <p>Дата начала действия сертификата (ISO 8601)</p> <p>Дата окончания действия сертификата (ISO 8601)</p>
---	--

2.6.3 Методы перевыпуска сертификата по запросу pkcs#10

2.6.3.1 Перевыпуск сертификата по запросу pkcs#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

POST API – Перевыпуск сертификата по запросу pkcs#10 (multipart/form-data)	
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление сертификатами субъектов и использование шаблона, идентификаторы которых передаются во входных параметрах. <p>Использование данного метода оператором для создания сертификатов для учетных записей запрещено.</p> <p>Метод позволяет перевыпустить сертификат по запросу, по которому ранее уже был выпущен сертификат.</p> <p>При использовании метода проверяется наличие в базе данных программы выпущенного сертификата, имеющего «Subject Key Identifier» аналогичный указанному в запросе на сертификат из входных параметров. Если такой сертификат не будет найден, данный метод осуществит выпуск нового сертификата по запросу аналогично методу выпуска сертификата по запросу (см. раздел 5.2).</p> <p>Если сертификат с аналогичным указанному в запросе на сертификат «Subject Key Identifier» будет найден, программа:</p> <ol style="list-style-type: none"> 1) проверит статус найденного сертификата. Если срок действия данного сертификата истек, запрос пользователя будет отклонен. 2) проверит соответствие значений в полях SDN и SAN, указанных в запросе (или во входных параметрах метода), значениям в полях SDN и SAN в найденном сертификате. При соответствии значений будет осуществлен выпуск сертификата по запросу на сертификат из входных параметров метода, иначе запрос пользователя будет отклонен. 	
URL – certificate-authority-service/api/v3/public/certificates/renewal/{caId}/pkcs10	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3/Контроллер%3A%20сертификаты/renewalRequestByCald_1	
Query	
{	
caId (UUID),	ID ЦС
subjectId (UUID) [обязателен, если не указан userId],	ID субъекта
userId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона
request (binary),	Файл запроса на сертификат.

	<p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» значения полей запроса на сертификат должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта. Допустимые форматы запроса на сертификат:</p> <ul style="list-style-type: none"> • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE-----").
<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>	<p>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.</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p>
<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[] } [опционально]</pre>	<p>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. Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p>
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID сертификата
downloadActions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
fingerprint (string),	Фингерпринт шаблона
serialnumber (string),	Серийный номер сертификата
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
name (string),	Имя сертификата (на основе CN)
issuerDN: {	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
(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[]),	

<pre>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[] },</pre>	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
<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 из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant)	Дата окончания действия сертификата (ISO 8601)
}	

2.6.3.2 Перевыпуск сертификата по запросу pkcs#10 (формат запроса на сертификат из входных параметров – application/json)

Примечание: отличается от метода выше форматом передаваемого во входных параметрах запроса на сертификат. В данном методе на вход поступает содержимое файла запроса на сертификат в формате PEM (содержимое в Base64).

<p>POST API – Перевыпуск сертификата в по запросу pkcs#10 (application/json)</p> <p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление сертификатами субъектов и использование шаблона, идентификаторы которых передаются во входных параметрах. <p>Использование данного метода оператором для создания сертификатов для учетных записей запрещено.</p> <p>Метод позволяет перевыпустить сертификат по запросу, по которому ранее уже был выпущен сертификат.</p> <p>При использовании метода проверяется наличие в базе данных программы выпущенного сертификата, имеющего «Subject Key Identifier» аналогичный указанному в запросе на сертификат из входных параметров. Если такой сертификат не будет найден, данный метод осуществит выпуск нового сертификата по запросу аналогично методу выпуска сертификата по запросу. Если сертификат с аналогичным указанному в запросе на сертификат «Subject Key Identifier» будет найден, программа:</p> <ol style="list-style-type: none"> 1) проверит статус найденного сертификата. Если срок действия данного сертификата истек, запрос пользователя будет отклонен. 2) проверит соответствие значений в полях SDN и SAN, указанных в запросе (или во входных параметрах метода), значениям в полях SDN и SAN в найденном сертификате. При соответствии значений будет осуществлен выпуск сертификата по запросу на сертификат из входных параметров метода, иначе запрос пользователя будет отклонен. <p>URL – certificate-authority-service/api/v3/public/certificates/renewal/{cald}/pkcs10</p> <p>Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A сертификаты/renewalRequestByCald_1</p>

Query	
{	
caId (UUID),	ID ЦС
subjectId (UUID) [обязателен, если не указан userId],	ID субъекта
userId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона
request: {	Запрос на сертификат
contentType(string) [опционально],	Тип загружаемого файла (HTTP MediaType) - application/octet-stream)
fileName (string) [опционально],	Имя загружаемого файла
data (string:binary)	Содержимое PEM файла запроса на сертификат (массив байт в Base64) – см. пример использования метода ниже. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» значения полей запроса на сертификат должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта. Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
},	
subjectName: {	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. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам 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): string[]	
}, [опционально]	
subjectAltName: {	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. Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам 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): string[]	
} [опционально]	
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID сертификата
downloadActions: {	Доступные действия по выгрузке

p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
fingerprint (string),	Фингерпринт шаблона
serialnumber (string),	Серийный номер сертификата
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
name (string),	Имя сертификата (на основе CN)
issuerDN: { (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 издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
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 из сертификата. 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, 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
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant)	Дата окончания действия сертификата (ISO 8601)
}	

2.6.4 Методы валидации запроса pkcs#10

2.6.4.1 Метод валидации запроса pkcs#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

POST API – Валидация запроса pkcs#10 (multipart/form-data)
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление сертификатами субъектов и использование шаблона, идентификаторы которых передаются во входных параметрах. <p>Использование данного метода оператором для валидации запросов на сертификат для учетных записей запрещено.</p>
URL – certificate-authority-service/api/v3/public/certificates/validate/{cald}/pkcs10
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A сертификаты/validate_1

Query	
{	
caId (UUID)	ID ЦС
subjectId (UUID) [обязателен, если не указан accountId],	ID субъекта
accountId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона
request (binary),	Файл запроса на сертификат. Допустимые форматы запроса на сертификат: <ul style="list-style-type: none"> • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE-----").
subjectName: { <ul style="list-style-type: none"> (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]	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.
subjectAltName: { <ul style="list-style-type: none"> (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[] } [опционально]	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. 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
name (string),	Имя сертификата (на основе CN)
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
subjectId (UUID)	ID субъекта (может отсутствовать, если в Query указан accountId, а не subjectId)
valid (boolean),	Флаг: запрос прошел валидацию
subjectNames: [{	Компоненты имени субъекта сертификата
fieldName (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),	Имя компонента
fieldValue (string),	Значение компонента
required (boolean),	Флаг: компонент обязательный
additional (boolean),	Флаг: компонент дополнительный
valid (boolean),	Флаг: компонент прошел валидацию
message (string)	Дополнительное сообщение
}],	
subjectAltNames: [{	Компоненты расширенного имени субъекта сертификата
fieldName (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),	Имя компонента
fieldValue (string),	Значение компонента
required (boolean),	Флаг: компонент обязательный

additional (boolean),	Флаг: компонент дополнительный
valid (boolean),	Флаг: компонент прошел валидацию
message (string)	Дополнительное сообщение
}}	
}	

2.6.4.2 Метод валидации запроса pkcs#10 (формат запроса на сертификат из входных параметров – application/json)

Примечание: отличается от метода выше форматом передаваемого во входных параметрах запроса на сертификат. В данном методе на вход поступает содержимое файла запроса на сертификат в формате PEM (содержимое в Base64).

POST API – Валидация запроса pkcs#10 (application/json)	
Метод доступен:	
<ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление сертификатами субъектов и использование шаблона, идентификаторы которых передаются во входных параметрах. 	
Использование данного метода оператором для валидации запросов на сертификат для учетных записей запрещено.	
URL – certificate-authority-service/api/v3/public/certificates/validate/{caId}/pkcs10	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3/Контроллер%3A%20сертификаты/validate_1	
Query	
{	
caId (UUID)	ID ЦС
subjectId (UUID) [обязателен, если не указан accountId],	ID субъекта
accountId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона
request: {	Файл запроса на сертификат
contentType (string),	Тип загружаемого файла (HTTP MediaType) - application/octet-stream)
fileName (string),	Имя загружаемого файла
data (string:binary)	Содержимое загружаемого файла (массив байт в Base64). Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE-----").
},	
subjectName: {	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.
(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[]	Значения полей, указанные в subjectName, переопределяют значения соответствующих полей SDN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.
}, [опционально]	

<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[] } [опционально] </pre>	<p>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.</p> <p>Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.</p>
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
name (string),	Имя сертификата (на основе CN)
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
subjectId (UUID)	ID субъекта (может отсутствовать, если в Query указан accountId, а не subjectId)
valid (boolean),	Флаг: запрос прошел валидацию
subjectNames: [{	Компоненты имени субъекта сертификата
fieldName (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),	Имя компонента
fieldValue (string),	Значение компонента
required (boolean),	Флаг: компонент обязательный
additional (boolean),	Флаг: компонент дополнительный
valid (boolean),	Флаг: компонент прошел валидацию
message (string)	Дополнительное сообщение
}],	
subjectAltNames: [{	Компоненты расширенного имени субъекта сертификата
fieldName (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),	Имя компонента
fieldValue (string),	Значение компонента
required (boolean),	Флаг: компонент обязательный
additional (boolean),	Флаг: компонент дополнительный
valid (boolean),	Флаг: компонент прошел валидацию
message (string)	Дополнительное сообщение
}]	
}	

2.6.5 Метод поиска сертификатов

GET API – Поиск сертификатов
<p>Метод доступен администратору и оператору.</p> <p>В ответе для оператора содержатся только сертификаты субъектов, к которым ему предоставлен доступ.</p> <p>В ответе данного метода в полях «subjectDN» и «issuerDN» компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения) в случае их наличия в сертификате будут указаны как «UNKNOWN». Данные компоненты поддерживаются в публичном API начиная с версии v4.</p>
URL – certificate-authority-service/api/v3/public/certificates

Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A сертификаты/findAll_14	
Query	
{	
search (string) [опционально],	Полнотекстовый поиск (имя или серийный номер)
issuerId (UUID) [опционально],	ID сертификата издателя
templateId (UUID) [опционально],	ID шаблона
status (enum[: ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN] [опционально],	Статус сертификата
type (enum[:CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN] [опционально],	Тип сертификата
revocationReason (enum[:UNSPECIFIED, KEY_COMPROMISE, CA_COMPROMISE, AFFILIATION_CHANGED, SUPERSEDED, CESSATION_OF_OPERATION, CERTIFICATE_HOLD, REMOVE_FROM_CRL, PRIVILEGE_WITHDRAWN, AA_COMPROMISE, UNKNOWN] [опционально],	Причина отзыва
notRevocationReason (enum[:UNSPECIFIED, KEY_COMPROMISE, CA_COMPROMISE, AFFILIATION_CHANGED, SUPERSEDED, CESSATION_OF_OPERATION, CERTIFICATE_HOLD, REMOVE_FROM_CRL, PRIVILEGE_WITHDRAWN, AA_COMPROMISE, UNKNOWN] [опционально],	Исключая причину отзыва
revocationDateFrom (instant) [опционально],	Дата отзыва (начало)
revocationDateTo (instant) [опционально],	Дата отзыва (окончание)
hasRevocationReason (boolean) [опционально],	Флаг: наличие причины отзыва
hasRequest (boolean) [опционально],	Флаг: наличие pkcs10
hasCA (boolean) [опционально],	Флаг: наличие ЦС
isManagementCA (boolean) [опционально],	Флаг: технологический ЦС
isValid (boolean) [опционально],	Флаг: сертификат действует
isExpired (boolean) [опционально],	Флаг: сертификат истек
validFromFrom (instant) [опционально],	Дата начала действия (начало)
validFromTo (instant) [опционально],	Дата начала действия (окончание)
validToFrom (instant) [опционально],	Дата окончания действия (начало)
validToTo (instant) [опционально],	Дата окончания действия (окончание)
updatedFrom (instant) [опционально],	Дата обновления сертификата (начало)
updatedTo (instant) [опционально],	Дата обновления сертификата (конец)
subjectId (UUID[]) [опционально],	ID субъекта
userId (UUID[]) [опционально],	ID учетной записи
serialnumber (string[]) [опционально],	Серийный номер
fingerprint (string[]) [опционально],	Отпечаток
subjectKeyIdentifier (string[]) [опционально],	Идентификатор ключа субъекта
notId (UUID[]) [опционально],	Исключая ID сертификата
endEntityType(string[]) (enum[:ROOT_CA, SUB_CA, USER, DEVICE, UNKNOWN] [опционально],	Фильтр: Тип субъекта
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	Идентификатор сертификата
issuerId (UUID),	Идентификатор издателя сертификата
issuerFingerprint (string),	Фингерпринт издателя сертификата
serialnumber (string),	Серийный номер сертификата
fingerprint (string),	Фингерпринт сертификата

issuerDN: { (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[] },	Имя субъекта издателя сертификата Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
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[] },	Имя субъекта сертификата Поля разделенного имени субъекта из сертификата. 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, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[] },	Альтернативное имя субъекта сертификата Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
name (string),	Имя сертификата (на основе CN)
templateId (UUID),	Идентификатор шаблона
templateName (string),	Имя шаблона
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 ЦС
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
actions: { p12 (boolean), csr (boolean), pem (boolean) },	Доступные действия по выгрузке Флаг: выгрузка pkcs12 Флаг: выгрузка pkcs10 Флаг: выгрузка сертификата
revocation: { date (instant), number (int32), value (string) },	Сведения об отзыве сертификата Дата отзыва Код причины отзыва Значение причины отзыва
hashAlgorithm (enum: SHA1, SHA256, SHA384, SHA512, SHA3_256, SHA3_384, SHA3_512, RSASSA_PSS, MD5, GOST_R_34_11_2012, UNKNOWN),	Алгоритм подписи сертификата
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата
keyBits (int4),	Длина ключа сертификата
subjectKeyIdentifier (string),	Идентификатор ключа сертификата
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

2.6.6 Метод получения сертификата по идентификатору

GET API – Получение сертификата по идентификатору
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах. В ответе данного метода в полях «subjectDN» и «issuerDN» компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения)

в случае их наличия в сертификате будут указаны как «UNKNOWN». Данные компоненты поддерживаются в публичном API начиная с версии v4.	
URL – certificate-authority-service/api/v3/public/certificates/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A сертификаты/findById_13	
Query	
{	
id (UUID)	ID сертификата
}	
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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
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
templateName (string),	Имя шаблона

type (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип сертификата
endEntityType (enum: ROOT_CA, SUB_CA, USER, DEVICE, UNKNOWN),	Тип субъекта
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
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
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
subjectKeyIdIdentifier (string),	Идентификатор ключа сертификата
authorityKeyIdIdentifier (string)	Идентификатор ключа издателя сертификата
keyUsages: {	Назначение ключа сертификата
id (uuid),	Идентификатор элемента
code (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 (uuid),	Идентификатор назначения
value (string),	Наименование элемента
oid (string),	OID назначения
description (string),	Описание использования ключа
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
default (boolean)	Флаг: расширенное использование по умолчанию
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int4),	Код причины отзыва
value (string)	Значение причины отзыва
},	
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
userId (uuid),	Идентификатор субъекта
subjectId (uuid),	Идентификатор субъекта
created (instant)	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
}	

2.6.7 Метод получения сертификата по серийному номеру

GET API – Получение сертификата по его серийному номеру
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, серийный номер которого передается во входных параметрах. В ответе данного метода в полях «subjectDN» и «issuerDN» компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения) в случае их наличия в сертификате будут указаны как «UNKNOWN». Данные компоненты поддерживаются в публичном API начиная с версии v4.
URL – certificate-authority-service/api/v3/public/certificates/serialNumber/{serialNumber}
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A сертификаты/getBySerialNumber

Query	
{	
serialnumber (string)	Серийный номер сертификата (формат: 40 символов, нижний регистр)
}	
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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key - один из перечисленных в enum параметров, а value - значение параметра
},	
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
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип сертификата
endEntityType (enum: ROOT_CA, SUB_CA, USER, DEVICE, UNKNOWN),	Тип субъекта
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
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
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
id (uuid),	Идентификатор элемента
code (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPTMENT, DATA_ENCRYPTMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY, UNKNOWN),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (uuid),	Идентификатор элемента
value (string),	Наименование элемента
oid (string),	OID назначения
description (string)	Описание использования ключа
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
default (boolean)	Флаг: расширенное использование по умолчанию
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int4),	Код причины отзыва
value (string)	Значение причины отзыва
},	
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
userId (uuid),	Идентификатор субъекта
subjectId (uuid),	Идентификатор субъекта
created (instant)	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
}	

2.6.8 Метод получения сертификата по его отпечатку

GET API – Получение сертификата по его отпечатку	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, отпечаток которого передается во входных параметрах.	
В ответе данного метода в полях «subjectDN» и «issuerDN» компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения) в случае их наличия в сертификате будут указаны как «UNKNOWN». Данные компоненты поддерживаются в публичном API начиная с версии v4.	
URL – certificate-authority-service/api/v3/public/certificates/fingerprint/{fingerprint}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A сертификаты/getByFingerprint_1	
Query	
{	
fingerprint (String)	Отпечаток сертификата
}	
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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key - один из перечисленных в enum параметров, а value - значение параметра
},	
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
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип сертификата
endEntityType (enum: ROOT_CA, SUB_CA, USER, DEVICE, UNKNOWN),	Тип субъекта
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
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
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
id (uuid),	Идентификатор элемента
code (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 (uuid),	Идентификатор элемента
value (string),	Наименование элемента
oid (string),	OID назначения
description, (string)	Описание использования ключа
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
default (boolean)	Флаг: расширенное использование по умолчанию
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int4),	Код причины отзыва
value (string)	Значение причины отзыва
},	
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
userId (uuid),	Идентификатор субъекта
subjectId (uuid),	Идентификатор субъекта
created (instant)	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
}	

2.6.9 Метод отзыва (приостановки) сертификата по идентификатору

PUT API – Отзыв (приостановка) сертификата по идентификатору	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – certificate-authority-service/api/v3/public/certificates/{id}/revoke	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A сертификаты/revokeById	
Query	
{	
id (UUID)	ID сертификата
}	
Request	
{	
reason (enum: UNSPECIFIED, KEY_COMPROMISE, CA_COMPROMISE, AFFILIATION_CHANGED, SUPERSEDED, CESSATION_OF_OPERATION, CERTIFICATE_HOLD, REMOVE_FROM_CRL, PRIVILEGE_WITHDRAWN, AA_COMPROMISE, UNKNOWN)	Причина отзыва (приостановки)
}	
Response	
-	
При указании значения «CERTIFICATE_HOLD» в параметре reason сертификат будет приостановлен, а не отозван.	

2.6.10 Метод активации сертификата по идентификатору

PUT API – Активация сертификата по идентификатору	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – certificate-authority-service/api/v3/public/certificates/{id}/reactivate	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A сертификаты/reactivateById	
Query	
{	
id (UUID)	ID сертификата
}	
Request	
–	
Response	
–	

2.6.11 Метод публикации сертификата в РС по идентификатору

PUT API – Активация сертификата по идентификатору	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – certificate-authority-service/api/v3/public/certificates/{id}/reactivate	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A сертификаты/reactivateById	
Query	
{	
id (UUID)	ID сертификата
}	
Request	
–	
Response	
–	

2.6.12 Метод расшифровки контейнера сертификата

POST API – Расшифровка контейнера сертификата	
Метод доступен администратору. В ответе данного метода в полях «subjectDN» и «issuerDN» компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения) в случае их наличия в каком-либо сертификате из состава контейнера будут указаны как «UNKNOWN». Данные компоненты поддерживаются в публичном API начиная с версии v4.	
URL – certificate-authority-service/api/v3/public/parse/pkcs12	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A контейнеры сертификатов/parse	
Query	
Request	
{	
container: {	Файл контейнера
contentType (string) [опционально],	Тип загружаемого файла (HTTP MediaType) – application/octet-stream)
fileName (string) [опционально],	Имя загружаемого файла
data (string:binary)	Содержимое загружаемого файла (массив байт в Base64)

<code>},</code>	
<code>password(string),</code>	Пароль от контейнера
<code>templateId (UUID)</code>	Идентификатор шаблона
<code>}</code>	
Response	
<code>{</code>	Ответ JSON в HTTP-body
<code>serialnumber (string),</code>	Серийный номер сертификата
<code>fingerprint (string),</code>	Фингерпринт сертификата
<code>name (string),</code>	Имя сертификата (на основе CN)
<code>issuerDN: {</code>	Имя субъекта издателя сертификата
<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, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]</code>	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
<code>}</code>	
<code>subjectDN: {</code>	Имя субъекта сертификата
<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, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]</code>	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
<code>}</code>	
<code>subjectAltName: {</code>	Альтернативное имя субъекта сертификата
<code>(enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPF_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[]</code>	Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
<code>validFrom (instant),</code>	Дата начала действия сертификата (ISO 8601)
<code>validTo (instant),</code>	Дата окончания действия сертификата (ISO 8601)
<code>privateKey: {</code>	Файл закрытого ключа
<code> contentType (string),</code>	Тип загружаемого файла (HTTP MediaType)
<code> fileName (string),</code>	Имя загружаемого файла
<code> data (string:binary)</code>	Содержимое загружаемого файла (массив байт в Base64)
<code>}</code>	
<code>certificate: {</code>	Файл сертификата
<code> contentType (string),</code>	Тип загружаемого файла (HTTP MediaType)
<code> fileName (string),</code>	Имя загружаемого файла
<code> data (string:binary)</code>	Содержимое загружаемого файла (массив байт в Base64)
<code>}</code>	
<code>keyUsages: {</code>	Назначение ключа сертификата
<code> id (uuid),</code>	Идентификатор элемента
<code> code (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPT, DATA_ENCRYPT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY, UNKNOWN),</code>	Перечисление использования ключа
<code> value (string),</code>	Наименование элемента
<code> description (string)</code>	Описание использования ключа
<code>}</code>	
<code>extendedKeyUsages: {</code>	Расширенное назначение ключа сертификата
<code> id (uuid),</code>	Идентификатор назначения
<code> value (string),</code>	Наименование элемента
<code> oid (string),</code>	OID назначения
<code> description (string),</code>	Описание использования ключа
<code> updated (instant),</code>	Время обновления (ISO 8601)
<code> created (instant),</code>	Время создания (ISO 8601)
<code> default (boolean)</code>	Флаг: расширенное использование по умолчанию
<code>}</code>	Описание OID
<code>ca (boolean)</code>	Флаг: сертификат ЦС
<code>}</code>	

2.6.13 Метод расшифровки сертификата

POST API – Расшифровка сертификата
Метод доступен администратору.

В ответе данного метода в полях «subjectDN» и «issuerDN» компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения) в случае их наличия в сертификате будут указаны как «UNKNOWN». Данные компоненты поддерживаются в публичном API начиная с версии v4.	
URL – certificate-authority-service/api/v3/public/parse/pem	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A контейнеры сертификатов/parse_2	
Query	
Request	
{	
request: {	Файл сертификата
contentType (string) [опционально],	Тип загружаемого файла (HTTP MediaType) - application/octet-stream
fileName (string) [опционально],	Имя загружаемого файла
data (string:binary)	Содержимое загружаемого файла (массив байт в Base64)
}	
Response	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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
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
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип сертификата
endEntityType (enum: ROOT_CA, SUB_CA, USER, DEVICE, UNKNOWN),	Тип субъекта
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
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
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
id (uuid),	Идентификатор элемента
code (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPTMENT, DATA_ENCRYPTMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY, UNKNOWN),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (uuid),	Идентификатор назначения
value (string),	Наименование элемента
oid (string),	OID назначения
description (string),	Описание использования ключа
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
default (boolean)	Флаг: расширенное использование по умолчанию
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int4),	Код причины отзыва
value (string)	Значение причины отзыва
},	
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
userId (uuid),	Идентификатор субъекта
subjectId (uuid),	Идентификатор субъекта
created (instant)	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
}	

2.6.14 Метод расшифровки запроса на сертификат

POST API – Расшифровка запроса на сертификат
Метод доступен администратору. В ответе данного метода в поле «subjectDN» компоненты «ROLE» (роль), «DATEOFBIRTH» (дата рождения) и «PLACEOFBIRTH» (место рождения) в случае их наличия в запросе будут указаны как «UNKNOWN». Данные компоненты поддерживаются в публичном API начиная с версии v4.
URL – certificate-authority-service/api/v3/public/parse/pkcs10
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A контейнеры сертификатов/parse_1

Query	
Request	
{	
request: {	Файл сертификата
contentType (string)	Тип загружаемого файла (HTTP MediaType) - application/octet-stream
[опционально],	
fileName (string) [опционально],	Имя загружаемого файла
data (string:binary)	Содержимое загружаемого файла (массив байт в Base64). Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
}	
subjectName: {	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.
(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[]	Значения полей, указанные в subjectName, переопределяют значения соответствующих полей SDN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.
}, [опционально]	
subjectAltName: {	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.
(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[]	Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.
} [опционально]	
}	
Response	
{	Ответ JSON в HTTP-body
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
},	
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
keyBits (int32),	Длина ключа сертификата ЦС
subjectKeyIdentifier (string)	Идентификатор ключа сертификата
}	

2.7 Методы экспорта файлов

2.7.1 Метод получения сертификата по идентификатору сертификата

GET API – Получение сертификата по идентификатору сертификата

Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – export-service/api/v3/public/export/certificates/{certificateId}/certificate	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3] Контроллер экспорта/findCertificateById	
Query	
{	
certificateId (UUID)	ID сертификата
}	
Request	
–	
Response	
ResponseEntity->byte[]	

2.7.2 Метод получения запроса на сертификат по идентификатору сертификата

GET API – Получение запроса на сертификат по идентификатору сертификата	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – export-service/api/v3/public/export/certificates/{certificateId}/pkcs10	
Swagger: https://HOST/ external-integration-service/swagger/swagger-ui/index.html#/v3] Контроллер экспорта/findRequestById	
Query	
{	
certificateId (UUID)	ID сертификата
}	
Request	
–	
Response	
ResponseEntity->byte[]	

2.7.3 Метод получения цепочки сертификата по идентификатору сертификата

GET API – Получение цепочки сертификатов по идентификатору сертификата	
Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – export-service/api/v3/public/export/certificates/{certificateId}/chain	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3] Контроллер экспорта/findChainById	
Query	
{	
certificateId (UUID)	ID сертификата
}	
Request	
–	
Response	
ResponseEntity->byte[]	

2.7.4 Метод получения контейнера PKCS#12 по идентификатору сертификата

GET API – Получение контейнера pkcs12 по идентификатору сертификата

Метод доступен администратору и оператору при наличии полномочий на управление сертификатом, идентификатор которого передается во входных параметрах	
URL – export-service/api/v3/public/export/certificates/{certificateId}/pkcs12	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3] Контроллер экспорта/findP12ById	
Query	
{	
certificateId (UUID)	ID сертификата
}	
Request	
–	
Response	
ResponseEntity->byte[]	

2.7.5 Метод получения сертификата Центра сертификации по идентификатору

GET API – Получение сертификата по идентификатору Центра сертификации	
Метод доступен администратору	
URL – export-service/api/v3/public/export/certificate-authorities/{caId}/certificate	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3] Контроллер экспорта/findCertificateByCald	
Query	
{	
caId (UUID)	ID ЦС
}	
Request	
–	
Response	
ResponseEntity->byte[]	

2.7.6 Метод получения цепочки сертификатов Центра сертификации по идентификатору

GET API – Получение цепочки сертификатов по идентификатору Центра сертификации	
Метод доступен администратору	
URL – export-service/api/v3/public/export/certificate-authorities/{caId}/chain	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3] Контроллер экспорта/findChainByCald	
Query	
{	
caId (UUID)	ID ЦС
}	
Request	
–	
Response	
ResponseEntity->byte[]	

2.7.7 Метод получения CRL по идентификатору Центра сертификации

GET API – Получение CRL по идентификатору Центра сертификации	
Метод доступен администратору и оператору	
URL – export-service/api/v3/public/export/certificate-authorities/{caId}/crl	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3] Контроллер экспорта/findCrlByCald	

Query
-
Request
-
Response
ResponseEntity->byte[]

2.7.8 Метод получения DeltaCRL по идентификатору Центра сертификации

GET API – Получение DeltaCRL по идентификатору Центра сертификации	
Метод доступен администратору и оператору	
URL – export-service/api/v3/public/export/certificate-authorities/{caId}/delta-crl	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3] Контроллер экспорта/findDeltaCrlByCald	
Query	
{	
caId (UUID)	ID ЦС
}	
Request	
-	
Response	
ResponseEntity->byte[]	

2.8 Методы работы с точками распространения

2.8.1 Метод генерации и публикации CRL по идентификатору Центра сертификации

POST API – Генерация и публикация CRL по идентификатору Центра сертификации	
Метод доступен администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3] Контроллер%3A Конфигурации CRL для ЦС/generate	
URL – publisher-service/api/v3/public/certificate-authorities/{caId}/crl-configuration/generate	
Query	
{	
id (UUID)	ID ЦС
}	
Request	
-	
Response	
-	

2.8.2 Метод генерации и публикации CRL по идентификатору Центра сертификации (устаревший)

POST API – Генерация и публикация CRL по идентификатору Центра сертификации	
Данный метод является устаревшим и будет исключен из публичного API в последующих версиях.	
Метод доступен администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3] Контроллер%3A Конфигурации CRL для ЦС/generateDeprecated	
URL – publisher-service/api/v3/ui/certificate-authorities/{caId}/crl-configuration/generate	
Query	
{	
id (UUID)	ID ЦС
}	
Request	
-	

Response

-

2.9 Методы работы с точками подключения и ресурсными системами

2.9.1 Метод поиска зарегистрированных ресурсных систем

GET API – Поиск ресурсных систем	
Метод доступен администратору и оператору. В ответе для оператора содержатся только те ресурсные системы, к субъектам которых ему предоставлен доступ.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3Контроллер%3A ресурсные системы/findAll_4	
URL – subjects-service/api/v3/public/resources	
Query	
{	
id (uuid) [опционально],	Фильтр: ID ресурсной системы
securityGroupId (uuid) [опционально],	Фильтр: ID группы безопасности
subjectId (uuid) [опционально],	Фильтр: ID субъекта
search (string) [опционально],	Фильтр: полнотекстовый поиск по отображаемому имени
isConnected(boolean) [опционально],	Фильтр: подключенная ресурсная система
isDefault (boolean) [опционально],	Фильтр: ресурсная система по умолчанию
status (string) [опционально],	Фильтр: статус ресурсной системы
inQueue (boolean) [опционально],	Фильтр: ресурсная система в очереди
lastSynchronizationSuccessDate (instant) [опционально]	Фильтр: дата и время последней синхронизации
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID ресурсной системы
commonName (string),	Имя ресурсной системы
distinguishedName (string),	DN ресурсной системы
subjectsCount (int64),	Количество субъектов ресурсной системы
isConnected (boolean),	Флаг: ресурсная система подключена
isDefault (boolean),	Флаг: локальная ресурсная система
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
status (string),	Статус ресурсной системы
inQueue (boolean),	Флаг: ресурсная система в очереди
lastSynchronizationSuccessDate (instant)	Дата и время последней синхронизации
}	

2.9.2 Метод получения ресурсной системы по идентификатору

GET API – Получение ресурсной системы по идентификатору	
Метод доступен администратору и оператору при наличии доступа к субъектам РС, идентификатор которой передается во входных параметрах	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3Контроллер%3A ресурсные системы/findById_4	
URL – subjects-service/api/v3/public/resources/{id}	
Query	
{	
id (uuid) [опционально]	ID ресурсной системы
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID ресурсной системы
commonName (string),	Имя ресурсной системы
distinguishedName (string),	DN ресурсной системы

subjectsCount (int64),	Количество субъектов ресурсной системы
isConnected (boolean),	Флаг: ресурсная система подключена
isDefault (boolean),	Флаг: локальная ресурсная система
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
status (string),	Статус ресурсной системы
inQueue (boolean),	Флаг: ресурсная система в очереди
lastSynchronizationSuccessDate (instant)	Дата и время последней синхронизации
}	

2.9.3 Метод полной синхронизации ресурсной системы

PUT API – Полная синхронизация РС	
Метод доступен администратору и оператору при наличии полномочий на субъектов данной РС	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A ресурсные системы/synchronize	
URL – https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 ldap-service/api/v3/public/resources/{resourceId}/synchronize	
Query	
{	
resourceId (UUID)	ID ресурсной системы
}	
Request	
-	
Response	
-	

2.9.4 Метод поиска идентификаторов ресурсных систем

GET API – Поиск ID ресурсных систем	
Метод доступен администратору и оператору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A словарь субъектов/findResources	
URL – https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 subjects-service/api/v3/public/subjects/dict/resources	
Query	
{	
id (string[]) [опционально],	Фильтр: ID субъекта
notId (string[]) [опционально],	Фильтр: исключая ID субъекта
resourceId (string[]) [опционально],	Фильтр: ID ресурсной системы
securityGroupId (string[]) [опционально],	Фильтр: ID группы безопасности
search (string),	Фильтр: полнотекстовый поиск по имени
isConnected (boolean),	Фильтр: подключение субъекта к ресурсной системе
isBlocked (boolean)	Фильтр: блокировка субъекта в ресурсной системе
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID ресурсной системы
value (string)	Имя ресурсной системы
}	

2.9.5 Метод поиска точек подключения к ресурсной системе

GET API – Поиск точек подключения к РС	
Метод доступен администратору и оператору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 Контроллер%3A Точки подключения ресурсных систем/findAll_12	
URL – https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3 ldap-service/api/v3/public/connection-points	
Query	
{	
id (uuid) [опционально],	Фильтр: ID точки подключения

resourceId (uuid) [опционально],	Фильтр: ID ресурсной системы
search (string),	Фильтр: полнотекстовый поиск по отображаемому имени
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально],	Ограничение на размер выборки (пагинация)
inQueue (boolean) [опционально]	Фильтр: точка подключения в очереди
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID точки подключения
title (string),	Имя точки подключения
domainType(enum: SAMBA_DC, MS_AD, RED_ADM, FREE_IPA, ALD_PRO, ALT_DOMAIN, UNKNOWN),	Тип точки подключения ¹
connectionAddress (string),	Адрес (хост) подключения
useTls (boolean),	Флаг: использовать TLS при подключении
baseDn(string),	BaseDN точки подключения
username (string),	Имя пользователя ресурсной системы
status (string)	Статус точки подключения
resourceId (UUID),	ID ресурсной системы
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
inQueue (boolean)	Флаг: точка подключения в очереди
}	

2.9.6 Метод получения точки подключения по идентификатору

GET API – Получение точки подключения по идентификатору	
Метод доступен администратору и оператору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3	
Контроллер%3A Точки подключения ресурсных систем/findById_11	
URL – ldap-service/api/v3/public/connection-points/{id}	
Query	
{	
id (uuid)	ID точки подключения
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID точки подключения
title (string),	Имя точки подключения
domainType(enum: SAMBA_DC, MS_AD, RED_ADM, FREE_IPA, ALD_PRO, ALT_DOMAIN, UNKNOWN),	Тип точки подключения ²
connectionAddress (string),	Адрес (хост) подключения
useTls (boolean),	Флаг: использовать TLS при подключении
baseDn(string),	BaseDN точки подключения
username (string),	Имя пользователя ресурсной системы
status (string)	Статус точки подключения
resourceId (UUID),	ID ресурсной системы
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
inQueue (boolean)	Флаг: точка подключения в очереди
}	

¹ В eCA-CA 2.4 реализована поддержка работы с ресурсной системой «ROSA Dynamic Directory», однако в API v3 при получении точек подключения к данной PC ее тип в поле «domainType» указывается как «FREE_IPA». Начиная с API v4 в поле «domainType» для точек подключения к ресурсной системе «ROSA Dynamic Directory» указывается значение «ROSA_DD».

² В eCA-CA 2.4 реализована поддержка работы с ресурсной системой «ROSA Dynamic Directory», однако в API v3 при получении точек подключения к данной PC ее тип в поле «domainType» указывается как «FREE_IPA». Начиная с API v4 в поле «domainType» для точек подключения к ресурсной системе «ROSA Dynamic Directory» указывается значение «ROSA_DD».

2.9.7 Метод частичной синхронизации точки подключения

PUT API – Частичная синхронизация точки подключения	
Метод доступен администратору и оператору при наличии полномочий. У оператора имеется возможность выполнять синхронизацию с теми внешними ресурсными системами, к субъектам которых у него есть доступ	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3Контроллер%3А Точки подключения ресурсных систем/synchronize_1	
URL – ldap-service/api/v3/public/connection-points/{pointId}/synchronize	
Query	
{	
pointId (UUID)	ID точки подключения
}	
Request	
-	
Response	
-	

2.10 Методы получения информации о сервисах

2.10.1 Методы получения информации о сервисе безопасности (security-service)

2.10.1.1 Метод получения эндпоинтов для запроса информации о сервисе безопасности (security-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе безопасности	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – security-service/actuator	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/security-service/actuator",	URL эндпоинта, который возвращает эндпоинты для запроса информации о сервисе, где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
},	
"health": {	
"href": "http://HOST/security-service/actuator/health",	URL эндпоинта, который возвращает информацию о состоянии сервиса (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
},	
"health-path": {	
"href": "http://HOST/security-service/actuator/health/{*path}",	URL зарезервированного эндпоинта под будущие реализации
"templated": true	Флаг наличия переменной в URL
},	
"info": {	

"href": "http://HOST/security-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/security-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

2.10.1.2 Метод получения информации о состоянии сервиса безопасности (security-service)

GET – Получение информации о состоянии сервиса безопасности	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – security-service/actuator/health	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
status (enum: UP, DOWN, OUT_OF_SERVICE, UNKNOWN)	Статус (состояние) сервиса безопасности. Возможные значения:
	- UP – работает;
	- DOWN – не работает;
	- OUT_OF_SERVICE – выключен;
	- UNKNOWN – нет информации.
}	

2.10.1.3 Метод получения информации о сервисе безопасности (security-service)

GET – Получение информации о сервисе безопасности	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – security-service/actuator/info	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

2.10.1.4 Метод получения Prometheus-метрик сервиса безопасности (security-service)

GET – Получение Prometheus-метрик сервиса безопасности	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – security-service/actuator/prometheus	
Swagger: -	
Query	-
Request	-
Response	
Метод возвращает метрики сервиса в формате Prometheus (text/plain)	

2.10.2 Методы получения информации о сервисе лицензий (license-service)

2.10.2.1 Метод получения эндпоинтов для запроса информации о сервисе лицензий (license-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе лицензий	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – license-service/actuator	
Swagger: -	
Query	-
Request	-
Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/license-service/actuator",	
"templated": false	
},	
"health": {	
"href": "http://HOST/license-service/actuator/health",	
"templated": false	
},	
"health-path": {	
"href": "http://HOST/license-service/actuator/health/{*path}",	
"templated": true	
},	
"info": {	
"href": "http://HOST/license-service/actuator/info",	

"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/license-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

2.10.2.2 Метод получения информации о состоянии сервиса лицензий (license-service)

GET – Получение информации о состоянии сервиса лицензий	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – license-service/actuator/health	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
status (enum: UP, DOWN, OUT_OF_SERVICE, UNKNOWN)	Статус (состояние) сервиса безопасности. Возможные значения:
	- UP – работает;
	- DOWN – не работает;
	- OUT_OF_SERVICE – выключен;
	- UNKNOWN – нет информации.
}	

2.10.2.3 Метод получения информации о сервисе лицензий (license-service)

GET – Получение информации о сервисе лицензий	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – license-service/actuator/info	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

2.10.2.4 Метод получения Prometheus-метрик сервиса лицензий (license-service)

GET – Получение Prometheus-метрик сервиса лицензий	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – license-service/actuator/prometheus	
Swagger: -	
Query	-
Request	-
Response	
Метод возвращает метрики сервиса в формате Prometheus (text/plain)	

2.10.3 Методы получения информации о сервисе журнала событий (logs-service)

2.10.3.1 Метод получения эндпоинтов для запроса информации о сервисе журнала событий (logs-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе журнала событий	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – logs-service/actuator	
Swagger: -	
Query	-
Request	-
Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/logs-service/actuator",	URL эндпоинта, который возвращает эндпоинты для запроса информации о сервисе, где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
},	
"health": {	
"href": "http://HOST/logs-service/actuator/health",	URL эндпоинта, который возвращает информацию о состоянии сервиса (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
},	
"health-path": {	
"href": "http://HOST/logs-service/actuator/health/{*path}",	URL зарезервированного эндпоинта под будущие реализации
"templated": true	Флаг наличия переменной в URL
},	
"info": {	
"href": "http://HOST/logs-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-CA

"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/logs-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

2.10.3.2 Метод получения информации о состоянии сервиса журнала событий (logs-service)

GET – Получение информации о состоянии сервиса журнала событий	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – logs-service/actuator/health	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
status (enum: UP, DOWN, OUT_OF_SERVICE, UNKNOWN)	Статус (состояние) сервиса безопасности. Возможные значения:
	- UP – работает;
	- DOWN – не работает;
	- OUT_OF_SERVICE – выключен;
	- UNKNOWN – нет информации.
}	

2.10.3.3 Метод получения информации о сервисе журнала событий (logs-service)

GET – Получение информации о сервисе журнала событий	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – logs-service/actuator/info	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

2.10.3.4 Метод получения Prometheus-метрик сервиса журнала событий (logs-service)

GET – Получение Prometheus-метрик сервиса журнала событий	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – logs-service/actuator/prometheus	
Swagger: -	
Query	-
Request	-
Response	
Метод возвращает метрики сервиса в формате Prometheus (text/plain)	

2.10.4 Методы получения информации о сервисе сертификатов (certificate-authority-service)

2.10.4.1 Метод получения эндпоинтов для запроса информации о сервисе сертификатов (certificate-authority-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе сертификатов	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – certificate-authority-service/actuator	
Swagger: -	
Query	-
Request	-
Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/certificate-authority-service /actuator",	
"templated": false	
},	
"health": {	
"href": "http://HOST/certificate-authority-service/actuator/health",	
"templated": false	
},	
"health-path": {	
"href": "http://HOST/certificate-authority-service/actuator/health/{*path}",	
"templated": true	
},	
"info": {	

"href": "http://HOST/certificate-authority-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/certificate-authority-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

2.10.4.2 Метод получения информации о состоянии сервиса сертификатов (certificate-authority-service)

GET – Получение информации о состоянии сервиса сертификатов	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – certificate-authority-service/actuator/health	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
status (enum: UP, DOWN, OUT_OF_SERVICE, UNKNOWN)	Статус (состояние) сервиса безопасности. Возможные значения:
	- UP – работает;
	- DOWN – не работает;
	- OUT_OF_SERVICE – выключен;
	- UNKNOWN – нет информации.
}	

2.10.4.3 Метод получения информации о сервисе сертификатов (certificate-authority-service)

GET – Получение информации о сервисе сертификатов	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – certificate-authority-service/actuator/info	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

2.10.4.4 Метод получения Prometheus-метрик сервиса сертификатов (certificate-authority-service)

GET – Получение Prometheus-метрик сервиса сертификатов	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – certificate-authority-service/actuator/prometheus	
Swagger: -	
Query	-
Request	-
Response	
Метод возвращает метрики сервиса в формате Prometheus (text/plain)	

2.10.5 Методы получения информации о сервисе настроек (settings-service)

2.10.5.1 Метод получения эндпоинтов для запроса информации о сервисе настроек (settings-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе настроек	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – settings-service/actuator	
Swagger: -	
Query	-
Request	-
Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/settings-service/actuator",	
"templated": false	
},	
"health": {	
"href": "http://HOST/settings-service/actuator/health",	
"templated": false	
},	
"health-path": {	
"href": "http://HOST/settings-service/actuator/health/{*path}",	
"templated": true	
},	
"info": {	

"href": "http://HOST/settings-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/settings-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

2.10.5.2 Метод получения информации о состоянии сервиса настроек (settings-service)

GET – Получение информации о состоянии сервиса настроек	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – settings-service/actuator/health	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
status (enum: UP, DOWN, OUT_OF_SERVICE, UNKNOWN)	Статус (состояние) сервиса безопасности. Возможные значения:
	- UP – работает;
	- DOWN – не работает;
	- OUT_OF_SERVICE – выключен;
	- UNKNOWN – нет информации.
}	

2.10.5.3 Метод получения информации о сервисе настроек (settings-service)

GET – Получение информации о сервисе настроек	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – settings-service/actuator/info	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

2.10.5.4 Метод получения Prometheus-метрик сервиса настроек (settings-service)

GET – Получение Prometheus-метрик сервиса настроек	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – settings-service/actuator/prometheus	
Swagger: -	
Query	-
Request	-
Response	
Метод возвращает метрики сервиса в формате Prometheus (text/plain)	

2.10.6 Методы получения информации о сервисе хранения данных (storage-service)

2.10.6.1 Метод получения эндпоинтов для запроса информации о сервисе хранения данных (storage-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе хранения данных	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – storage-service/actuator	
Swagger: -	
Query	-
Request	-
Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/storage-service/actuator",	URL эндпоинта, который возвращает эндпоинты для запроса информации о сервисе, где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
},	
"health": {	
"href": "http://HOST/storage-service/actuator/health",	URL эндпоинта, который возвращает информацию о состоянии сервиса (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
},	
"health-path": {	
"href": "http://HOST/storage-service/actuator/health/{*path}",	URL зарезервированного эндпоинта под будущие реализации
"templated": true	Флаг наличия переменной в URL
},	
"info": {	

"href": "http://HOST/storage-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/storage-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

2.10.6.2 Метод получения информации о состоянии сервиса хранения данных (storage-service)

GET – Получение информации о состоянии сервиса хранения данных	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – storage-service/actuator/health	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
status (enum: UP, DOWN, OUT_OF_SERVICE, UNKNOWN)	Статус (состояние) сервиса безопасности. Возможные значения:
	- UP – работает;
	- DOWN – не работает;
	- OUT_OF_SERVICE – выключен;
	- UNKNOWN – нет информации.
}	

2.10.6.3 Метод получения информации о сервисе хранения данных (storage-service)

GET – Получение информации о сервисе хранения данных	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – storage-service/actuator/info	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

2.10.6.4 Метод получения Prometheus-метрик сервиса хранения данных (storage-service)

GET – Получение Prometheus-метрик сервиса хранения данных	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – storage-service/actuator/prometheus	
Swagger: -	
Query	
-	
Request	
-	
Response	
Метод возвращает метрики сервиса в формате Prometheus (text/plain)	

2.10.7 Методы получения информации о сервисе оповещения пользователей (event-delivery-service)

2.10.7.1 Метод получения эндпоинтов для запроса информации о сервисе оповещения пользователей (event-delivery-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе оповещения пользователей	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – event-delivery-service/actuator	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/event-delivery-service/actuator",	
"templated": false	
},	
"health": {	
"href": "http://HOST/event-delivery-service/actuator/health",	
"templated": false	
},	
"health-path": {	
"href": "http://HOST/event-delivery-service/actuator/health/{*path}",	
"templated": true	
},	
"info": {	

"href": "http://HOST/event-delivery-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/event-delivery-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

2.10.7.2 Метод получения информации о состоянии сервиса оповещения пользователей (event-delivery-service)

GET – Получение информации о состоянии сервиса оповещения пользователей	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – event-delivery-service/actuator/health	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
status (enum: UP, DOWN, OUT_OF_SERVICE, UNKNOWN)	Статус (состояние) сервиса безопасности. Возможные значения:
	- UP – работает;
	- DOWN – не работает;
	- OUT_OF_SERVICE – выключен;
	- UNKNOWN – нет информации.
}	

2.10.7.3 Метод получения информации о сервисе оповещения пользователей (event-delivery-service)

GET – Получение информации о сервисе оповещения пользователей	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – event-delivery-service/actuator/info	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

2.10.7.4 Метод получения Prometheus-метрик сервиса оповещения пользователей (event-delivery-service)

GET – Получение Prometheus-метрик сервиса оповещения пользователей	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – event-delivery-service/actuator/prometheus	
Swagger: -	
Query	-
Request	-
Response	
Метод возвращает метрики сервиса в формате Prometheus (text/plain)	

2.10.8 Методы получения информации о сервисе внешних интеграций (external-integration-service)

2.10.8.1 Метод получения эндпоинтов для запроса информации о сервисе внешних интеграций (external-integration-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе внешних интеграций	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – external-integration-service/actuator	
Swagger: -	
Query	-
Request	-
Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/external-integration-service/actuator",	
"templated": false	
},	
"health": {	
"href": "http://HOST/external-integration-service/actuator/health",	
"templated": false	
},	
"health-path": {	
"href": "http://HOST/external-integration-service/actuator/health/{*path}",	
"templated": true	
},	
"info": {	

"href": "http://HOST/external-integration-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/external-integration-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-CA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

2.10.8.2 Метод получения информации о состоянии сервиса внешних интеграций (external-integration-service)

GET – Получение информации о состоянии сервиса внешних интеграций	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – external-integration-service/actuator/health	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
status (enum: UP, DOWN, OUT_OF_SERVICE, UNKNOWN)	Статус (состояние) сервиса безопасности. Возможные значения:
	- UP – работает;
	- DOWN – не работает;
	- OUT_OF_SERVICE – выключен;
	- UNKNOWN – нет информации.
}	

2.10.8.3 Метод получения информации о сервисе внешних интеграций (external-integration-service)

GET – Получение информации о сервисе внешних интеграций	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – external-integration-service/actuator/info	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

2.10.8.4 Метод получения Prometheus-метрик сервиса внешних интеграций (external-integration-service)

GET – Получение Prometheus-метрик сервиса внешних интеграций	
Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:	
<ul style="list-style-type: none"> если параметр имеет значение «true», доступ к методу будет разрешен только аутентифицированным пользователям; если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – external-integration-service/actuator/prometheus	
Swagger: -	
Query	
-	
Request	
-	
Response	
Метод возвращает метрики сервиса в формате Prometheus (text/plain)	

2.11 Методы работы с Syslog-серверами

2.11.1 Метод поиска Syslog-серверов

GET API – Поиск Syslog-серверов	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3Контроллер%3ASyslog сервера/findAll 10	
URL – logs-service/api/v3/public/syslog	
Query	
-	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID Syslog-сервера
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP, UNKNOWN),	Протокол Syslog-сервера
active (boolean)	Флаг: состояние настройки публикации событий
}	

2.11.2 Метод получения Syslog-сервера по идентификатору

GET API – Получение Syslog-сервера по идентификатору	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3Контроллер%3ASyslog сервера/findByld 9	
URL – logs-service/api/v3/public/syslog/{id}	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
-	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID Syslog-сервера

host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP, UNKNOWN),	Протокол Syslog-сервера
active (boolean)	Флаг: состояние настройки публикации событий
}	

2.11.3 Метод создания Syslog-сервера

POST API – Создание Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3	
Контроллер%3A Syslog сервера/create	
URL – logs-service/api/v3/public/syslog	
Query	
-	
Request	
{	
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP, UNKNOWN)	Протокол Syslog-сервера
}	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID Syslog-сервера
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP, UNKNOWN),	Протокол Syslog-сервера
active (boolean)	Флаг: состояние настройки публикации событий
}	

2.11.4 Метод обновления Syslog-сервера

PUT API – Обновление Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3	
Контроллер%3A Syslog сервера/updateById	
URL – logs-service/api/v3/public/syslog/{id}	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
{	
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP, UNKNOWN)	Протокол Syslog-сервера
}	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID Syslog-сервера
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP, UNKNOWN),	Протокол Syslog-сервера
active (boolean)	Флаг: состояние настройки публикации событий
}	

2.11.5 Метод деактивации Syslog-сервера

PATCH API – Деактивация Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3	
Контроллер%3A Syslog сервера/deactivate	

URL – logs-service/api/v3/public/syslog/{id}/deactivate	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
–	
Response	
–	

2.11.6 Метод активации Syslog-сервера

PATCH API – Активация Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#[v3]	
Контроллер%3A Syslog сервера/activate	
URL – logs-service/api/v3/public/syslog/{id}/activate	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
–	
Response	
–	

2.11.7 Метод удаления Syslog-сервера

DELETE API – Удаление Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#[v3]	
Контроллер%3A Syslog сервера/deleteById_2	
URL – logs-service/api/v3/public/syslog/{id}	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
–	
Response	
–	

2.12 Методы работы с учетными записями

2.12.1 Метод поиска учетных записей

GET API – Поиск учетных записей	
Метод доступен администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3	
Контроллер%3A Учетные записи/findAll_9	
URL – security-service/api/v3/public/accounts	
Query	
{	
id (uuid[]) [опционально],	Фильтр: ID учетной записи
notId (uuid[]) [опционально],	Фильтр: ID учетной записи (исключающий)
logins (uuid[]) [опционально],	Фильтр: логин учетной записи
search (string) [опционально]	Фильтр: полнотекстовый поиск по отображаемому имени
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID учетной записи
login (string),	Логин учетной записи
principalName (string),	Отображаемое учетной записи
role (enum: ADMINISTRATOR, OPERATOR, UNKNOWN),	Роль учетной записи
status (enum: ACTIVE, BLOCKED, UNKNOWN),	Статус учетной записи
created (instant),	Время создания (ISO 8601)
isLinked (boolean)	Флаг: связь с субъектом из ресурсной системы
}	

2.12.2 Метод получения учетной записи по идентификатору

GET API – Получение учетной записи по идентификатору	
Метод доступен администратору, а также оператору для получения данных о его собственной учетной записи	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3	
Контроллер%3A Учетные записи/findById_8	
URL – security-service/api/v3/public/accounts/{id}	
Query	
{	
id (uuid)	ID учетной записи
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID учетной записи
login (string),	Логин учетной записи
principalName (string),	Отображаемое учетной записи
role (enum: ADMINISTRATOR, OPERATOR, UNKNOWN),	Роль учетной записи
status (enum: ACTIVE, BLOCKED, UNKNOWN),	Статус учетной записи
created (instant),	Время создания (ISO 8601)
isLinked (boolean)	Флаг: связь с субъектом из ресурсной системы
}	

2.12.3 Метод получения учетной записи по отпечатку сертификата

GET API – Получение учетной записи по отпечатку сертификата
--

Метод доступен администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#[v3]Контроллер%3A Учетные записи/getByFingerprint	
URL – security-service/api/v3/public/accounts/fingerprint/{fingerprint}	
Query	
{	
fingerprint (string)	Отпечаток сертификата
}	
Request	
–	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID учетной записи
login (string),	Логин учетной записи
principalName (string),	Отображаемое учетной записи
role (enum: ADMINISTRATOR, OPERATOR, UNKNOWN),	Роль учетной записи
status (enum: ACTIVE, BLOCKED, UNKNOWN),	Статус учетной записи
created (instant),	Время создания (ISO 8601)
isLinked (boolean)	Флаг: связь с субъектом из ресурсной системы
}	

2.12.4 Метод получения учетной записи по идентификатору субъекта

GET API – Получение учетной записи по идентификатору субъекта	
Метод доступен администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#[v3]Контроллер%3A Учетные записи/findBySubjectId	
URL – security-service/api/v3/public/accounts/subject/{subjectId}	
Query	
{	
subjectId (string)	Идентификатор субъекта
}	
Request	
–	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID учетной записи
login (string),	Логин учетной записи
principalName (string),	Отображаемое учетной записи
role (enum: ADMINISTRATOR, OPERATOR, UNKNOWN),	Роль учетной записи
status (enum: ACTIVE, BLOCKED, UNKNOWN),	Статус учетной записи
created (instant),	Время создания (ISO 8601)
isLinked (boolean)	Флаг: связь с субъектом из ресурсной системы
}	

2.13 Методы работы с издателями

2.13.1 Метод поиска издателей

GET API – Поиск учетных записей	
Метод доступен администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3	
Контроллер%3A Издатели/findAll_8	
URL – settings-service/api/v3/public/issuer	
Query	
{	
authorityKeyIdentifier (string),	Идентификатор ключа издателя
subjectKeyIdentifier (string),	Идентификатор ключа сертификата ЦС
isActive (boolean),	Флаг: активный ЦС
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID издателя
caId (UUID),	ID ЦС
validFrom (instant),	Дата начала действия сертификата ЦС (ISO 8601)
validTo (instant),	Дата окончания действия сертификата ЦС (ISO 8601)
name (string),	Имя сертификата ЦС (на основе CN)
title (string),	Отображаемое имя ЦС
subjectKeyIdentifier (string),	Идентификатор ключа сертификата ЦС
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата ЦС
active (boolean)	Флаг: активный ЦС
}	

2.14 Методы работы с группами безопасности

2.14.1 Метод поиска групп безопасности

GET API – Поиск групп безопасности	
Метод доступен администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3Контроллер%3A группы безопасности/findAll_3	
URL – <code>subjects-service/api/v3/public/security-groups</code>	
Query	
<code>{</code>	
<code>sortDirection (string) [опционально],</code>	Направления сортировки (ASC;DESC)
<code>sortBy (string[]) [опционально],</code>	Список полей, к которым применяется сортировка
<code>pageOffset (integer) [опционально],</code>	Смещение от начала списка (пагинация)
<code>pageLimit (integer) [опционально],</code>	Ограничение на размер выборки (пагинация)
<code>id (UUID[]) [опционально],</code>	ID группы безопасности
<code>notId (UUID[]) [опционально],</code>	Исключая ID группы безопасности
<code>resourceId (UUID) [опционально],</code>	ID ресурсной системы
<code>subjectId (uuid) [опционально],</code>	Идентификатор субъекта
<code>search (string) [опционально],</code>	Фильтр: полнотекстовый поиск
<code>isActive (boolean)</code>	Флаг: активная для назначения прав группа безопасности
<code>}</code>	
Request	
-	
Response	
<code>ResponseEntity -> CollectionResponse -> {</code>	Ответ JSON в HTTP-body
<code>id (UUID),</code>	ID группы безопасности
<code>commonName (string),</code>	Имя группы безопасности
<code>distinguishedName (string),</code>	DN группы безопасности
<code>resource: {</code>	Ресурсная система
<code>id (UUID),</code>	ID ресурсной системы
<code>commonName (string),</code>	Имя ресурсной системы
<code>distinguishedName (string),</code>	DN ресурсной системы
<code>},</code>	
<code>isActive (boolean),</code>	Флаг: активная для назначения прав группа безопасности
<code>modify (instant),</code>	Дата и время обновления в ресурсной системе (ISO 8601)
<code>updated (instant),</code>	Время обновления (ISO 8601)
<code>created (instant)</code>	Время создания (ISO 8601)
<code>}</code>	

2.14.2 Метод получения группы безопасности по идентификатору

GET API – Получение группы безопасности по идентификатору	
Метод доступен администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v3Контроллер%3A группы безопасности/findById_3	
URL – <code>subjects-service/api/v3/public/security-groups/{id}</code>	
Query	
<code>{</code>	
<code>id (uuid) [опционально]</code>	ID группы безопасности
<code>}</code>	
Request	
-	
Response	
<code>{</code>	Ответ JSON в HTTP-body
<code>id (UUID),</code>	ID группы безопасности
<code>commonName (string),</code>	Имя группы безопасности
<code>distinguishedName (string),</code>	DN группы безопасности

resource: {	Ресурсная система
id (UUID),	ID ресурсной системы
commonName (string),	Имя ресурсной системы
distinguishedName (string),	DN ресурсной системы
},	
isActive (boolean),	Флаг: активная для назначения прав группа безопасности
modify (instant),	Дата и время обновления в ресурсной системе (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

2.15 Методы работы с центрами валидации

2.15.1 Метод проверки доступности центра валидации

PUT API – Метод проверки доступности центра валидации	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv3%5D%20Контроллер%3A%20центры%20валидации/heartbeat	
URL – validation-authority-service/api/v3/public/validation-authorities/{id}/heartbeat	
Query	
{	
id (uuid)	ID центра валидации
}	
Request	
-	
Response	
-	

2.15.2 Метод регистрации центра валидации

POST API – Регистрация центра валидации	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv3%5D%20Контроллер%3A%20центры%20валидации/create	
URL – validation-authority-service/api/v3/public/validation-authorities/{id}	
Query	
{	
id (uuid)	ID центра валидации
}	
Request	
{	
certificateAuthorityId (UUID),	ID центра сертификации
instanceId (UUID),	ID экземпляра центра валидации
hostname (string),	Адрес хоста eCA-VA
httpsPort (int),	HTTPS порт доступа к Центру валидации
httpPort (int),	HTTP порт доступа к Центру валидации
crl: {	Информация о точке распространения CRL
distribution (string),	URL распространения
publication (string),	URL публикации
info (string)	URL получения информации о точке
},	
deltacrl: {	Информация о точке распространения Delta CRL
distribution (string),	URL распространения
publication (string),	URL публикации
info (string)	URL получения информации о точке
},	
aia: {	Информация о точке распространения AIA
distribution (string),	URL распространения
publication (string),	URL публикации
info (string)	URL получения информации о точке
}	
}	
Response	
{	Ответ JSON в HTTP-body
id (uuid),	ID центра валидации
certificateAuthorityId (UUID),	ID центра сертификации
instanceId (UUID),	ID экземпляра центра валидации
hostname (string),	Адрес хоста eCA-VA
isLegacy (boolean),	Флаг: ЦВ старой версии
lastUpdate (instant),	Время последнего подключения (ISO 8601)
created (instant),	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
}	

2.15.3 Метод удаления центра валидации

DELETE API – Удаление центра валидации	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv3%5D%20Контроллер%3A%20центры%20валидации/delete	
URL – validation-authority-service/api/v3/public/validation-authorities/{id}	
Query	
{	
id (uuid)	ID центра валидации
}	
Request	
–	
Response	
–	

2.15.4 Метод создания службы OCSP

POST API – Создание службы OCSP	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/Контроллер%3A%20Службы%20OCSP%20Центров%20валидации/createOcsByVald	
URL – validation-authority-service/api/v3/public/validation-authorities/{vald}/ocsp-services	
Query	
{	
id (uuid)	ID центра валидации
}	
Request	
{	
url (string),	URL службы
info (string)	URL получения информации о службе
}	
Response	
{	Ответ JSON в HTTP-body
id (uuid),	ID службы
caId (uuid),	ID центра сертификации
vald (uuid),	ID центра валидации
url (string),	URL службы
info (string),	URL получения информации о службе
vaHost (string),	Адрес хоста eCA-VA
priority(int),	Приоритет службы
children {вложенные объекты с аналогичной структурой}	Дочерние службы OCSP
created (instant),	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
active (boolean)	Флаг: служба OCSP активна
}	

2.15.5 Метод удаления службы OCSP

DELETE API – Удаление службы OCSP	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/Контроллер%3A%20Службы%20OCSP%20Центров%20валидации/deleteOcsByVald	
URL – validation-authority-service/api/v3/public/validation-authorities/{vald}/ocsp-services	
Query	
{	
id (uuid)	ID центра валидации
}	
Request	
–	
Response	
–	

3 ОПИСАНИЕ PROMETHEUS-МЕТРИК ДЛЯ REST API ВЕРСИИ 3

3.1 Базовые метрики сервиса

3.1.1 Время запуска

- `application_ready_time_seconds{main_application_class="..."}` gauge. Время, за которое сервис стал готов обслуживать запросы (в секундах). Метка «`main_application_class`» содержит имя основного класса сервиса.
- `application_started_time_seconds{main_application_class="..."}` gauge. Время, затраченное на запуск сервиса (в секундах). Метка «`main_application_class`» содержит имя основного класса сервиса.

3.2 Метрики диска

- `disk_free_bytes{path="..."}` gauge. Свободное место на диске, в котором располагается сервис (в байтах). Метка «`path`» указывает путь к сервису в файловой системе.
- `disk_total_bytes{path="..."}` gauge. Общий объем диска, в котором располагается сервис (в байтах). Метка «`path`» указывает путь к сервису в файловой системе.

3.3 Метрики исполнителей (Thread Pools)

3.3.1 `taskExecutor` (пул асинхронных задач)

- `executor_active_threads{name="taskExecutor"}` gauge. Количество потоков, прямо сейчас выполняющих задачи.
- `executor_completed_tasks_total{name="taskExecutor"}` counter. Сколько задач уже выполнено с момента запуска.
- `executor_pool_core_threads{name="taskExecutor"}` gauge. Минимальное количество потоков, которое пул старается поддерживать.
- `executor_pool_max_threads{name="taskExecutor"}` gauge. Максимальное количество потоков, которое может быть создано.
- `executor_pool_size_threads{name="taskExecutor"}` gauge. Сколько потоков сейчас существует в пуле.
- `executor_queue_remaining_tasks{name="taskExecutor"}` gauge. Количество свободных мест в очереди задач без блокировки.
- `executor_queued_tasks{name="taskExecutor"}` gauge. Количество задач, ожидающих в очереди на выполнение.

3.3.2 `taskScheduler` (пул планировщика задач)

- `executor_active_threads{name="taskScheduler"}` gauge. Количество потоков, прямо сейчас выполняющих запланированные задачи.
- `executor_completed_tasks_total{name="taskScheduler"}` counter. Общее количество уже завершенных запланированных задач.
- `executor_pool_core_threads{name="taskScheduler"}` gauge. Базовый (core) размер пула потоков.
- `executor_pool_max_threads{name="taskScheduler"}` gauge. Максимально допустимый размер пула.
- `executor_pool_size_threads{name="taskScheduler"}` gauge. Текущее количество потоков в пуле.
- `executor_queue_remaining_tasks{name="taskScheduler"}` gauge. Количество свободных мест в очереди запланированных задач.

- `executor_queued_tasks{name="taskScheduler"}` gauge. Количество задач, ожидающих в очереди на выполнение.

3.4 Метрики пула подключений к БД (HikariCP)

3.4.1 Основные метрики пула

- `hikaricp_connections{pool="..."}` gauge. Общее количество подключений в пуле.
- `hikaricp_connections_acquire_seconds_count{pool="..."}` counter. Количество операций получения подключения из пула.
- `hikaricp_connections_acquire_seconds_sum{pool="..."}` counter. Суммарное время получения подключений (в секундах).
- `hikaricp_connections_acquire_seconds_max{pool="..."}` gauge. Максимальное время получения подключения (в секундах).
- `hikaricp_connections_active{pool="..."}` gauge. Количество активных подключений.
- `hikaricp_connections_creation_seconds_count{pool="..."}` counter. Количество созданных подключений.
- `hikaricp_connections_creation_seconds_sum{pool="..."}` counter. Суммарное время создания подключений (в секундах).
- `hikaricp_connections_creation_seconds_max{pool="..."}` gauge. Максимальное время создания подключения (в секундах).
- `hikaricp_connections_idle{pool="..."}` gauge. Количество простаивающих подключений.
- `hikaricp_connections_max{pool="..."}` gauge. Максимальный размер пула.
- `hikaricp_connections_min{pool="..."}` gauge. Минимальный размер пула.
- `hikaricp_connections_pending{pool="..."}` gauge. Количество потоков, ожидающих подключение.
- `hikaricp_connections_timeout_total{pool="..."}` counter. Количество таймаутов при получении подключения.
- `hikaricp_connections_usage_seconds_count{pool="..."}` counter. Количество операций использования подключений.
- `hikaricp_connections_usage_seconds_sum{pool="..."}` counter. Суммарное время использования подключений (в секундах).
- `hikaricp_connections_usage_seconds_max{pool="..."}` gauge. Максимальное время использования одного подключения (в секундах).

3.5 Метрики HTTP-клиента

3.5.1 Активные клиентские запросы

- `http_client_requests_active_seconds_count{client_name="...", exception="...", method="...", outcome="...", status="...", uri="..."}` counter. Количество активных исходящих запросов.
- `http_client_requests_active_seconds_sum{client_name="...", exception="...", method="...", outcome="...", status="...", uri="..."}` counter. Суммарное время активных исходящих запросов.
- `http_client_requests_active_seconds_max{client_name="...", exception="...", method="...", outcome="...", status="...", uri="..."}` gauge. Максимальное время активного исходящего запроса.

3.5.2 Завершенные клиентские запросы

- `http_client_requests_seconds_count{client_name="...", error="...", exception="...", method="...", outcome="...", status="...", uri="..."}` counter. Количество исходящих HTTP-запросов.

- `http_client_requests_seconds_sum{client_name="...", error="...", exception="...", method="...", outcome="...", status="...", uri="..."}` counter. Суммарное время выполнения исходящих запросов (в секундах).
- `http_client_requests_seconds_max{client_name="...", error="...", exception="...", method="...", outcome="...", status="...", uri="..."}` gauge. Максимальное время выполнения исходящего запроса.

3.6 Метрики HTTP-сервера

3.6.1 Активные серверные запросы

- `http_server_requests_active_seconds_count{exception="...", method="...", outcome="...", status="...", uri="..."}` counter. Количество активных входящих запросов.
- `http_server_requests_active_seconds_sum{exception="...", method="...", outcome="...", status="...", uri="..."}` counter. Суммарное время активных запросов.
- `http_server_requests_active_seconds_max{exception="...", method="...", outcome="...", status="...", uri="..."}` gauge. Максимальное время активного запроса.

3.6.2 Завершенные серверные запросы

- `http_server_requests_seconds_count{error="...", exception="...", method="...", outcome="...", status="...", uri="..."}` counter. Количество входящих HTTP-запросов.
- `http_server_requests_seconds_sum{error="...", exception="...", method="...", outcome="...", status="...", uri="..."}` counter. Суммарное время обработки входящих запросов (в секундах).
- `http_server_requests_seconds_max{error="...", exception="...", method="...", outcome="...", status="...", uri="..."}` gauge. Максимальное время обработки входящего запроса (в секундах).

3.7 JDBC-метрики (альтернативное представление HikariCP)

- `jdbcn_connections_active{name="dataSource"}` gauge. Количество активных подключений.
- `jdbcn_connections_idle{name="dataSource"}` gauge. Количество простаивающих подключений.
- `jdbcn_connections_max{name="dataSource"}` gauge. Максимальный размер пула.
- `jdbcn_connections_min{name="dataSource"}` gauge. Минимальный размер пула.

3.8 Метрики JVM (Java Virtual Machine)

3.8.1 Общая информация

- `jvm_info{runtime="...", vendor="...", version="..."}` gauge. Информация о версии JVM (значение всегда 1, метки содержат детали).

3.8.2 Буферы

- `jvm_buffer_count_buffers{id="..."}` gauge. Количество буферов в пуле. Метка `id` указывает тип буфера (`direct` или `mapped`).
- `jvm_buffer_memory_used_bytes{id="..."}` gauge. Память, используемая буферами (в байтах).
- `jvm_buffer_total_capacity_bytes{id="..."}` gauge. Общая емкость буферов (в байтах).

3.8.3 Классы

- `jvm_classes_loaded_classes` gauge. Количество загруженных классов.
- `jvm_classes_unloaded_classes_total` counter. Общее количество выгруженных классов.

3.8.4 Компиляция

- `jvm_compilation_time_ms_total{compiler="..."}` counter. Общее время, затраченное на JIT-компиляцию (в миллисекундах).

3.8.5 Сборка мусора

- `jvm_gc_live_data_size_bytes` gauge. Размер данных в long-lived heap после последней сборки мусора.
- `jvm_gc_max_data_size_bytes` gauge. Максимальный размер "долгоживущей" области (Old Generation) в байтах.
- `jvm_gc_memory_allocated_bytes_total` counter. Объем памяти, выделенной в молодом поколении после сборки мусора.
- `jvm_gc_memory_promoted_bytes_total` counter. Объем памяти, продвинутой из молодого поколения в старое.
- `jvm_gc_overhead` gauge. Процент времени CPU, затраченного на сборку мусора (значение от 0 до 1).

3.8.6 Память (выделенная)

- `jvm_memory_committed_bytes{area="...", id="..."}` gauge. Объем памяти, гарантированно доступный JVM (в байтах). Метка `area` указывает область (heap или nonheap), метка `id` указывает конкретный пул памяти.

3.8.7 Память (максимальная)

- `jvm_memory_max_bytes{area="...", id="..."}` gauge. Максимальный объем памяти, который может использовать JVM (в байтах).

3.8.8 Память (после сборки мусора)

- `jvm_memory_usage_after_gc{area="heap", pool="long-lived"}` gauge. Процент использования long-lived области после последней сборки мусора (значение от 0 до 1).

3.8.9 Память (используемая)

- `jvm_memory_used_bytes{area="...", id="..."}` gauge. Используемая память (в байтах) по областям heap и non-heap.

3.8.10 Потоки

- `jvm_threads_daemon_threads` gauge. Количество потоков-демонов.
- `jvm_threads_live_threads` gauge. Текущее количество живых потоков.
- `jvm_threads_peak_threads` gauge. Пиковое количество потоков с момента запуска.
- `jvm_threads_started_threads_total` counter. Общее количество запущенных потоков.
- `jvm_threads_states_threads{state="..."}` gauge. Количество потоков в каждом состоянии (runnable, waiting, timed-waiting, blocked, new, terminated).

3.9 Метрики логирования (Logback)

- `logback_events_total{level="..."}` counter. Количество событий лога по уровням: debug, error, info, trace, warn.

3.10 Метрики процесса

- `process_cpu_time_ns_total` counter. Процессорное время, использованное процессом JVM (в наносекундах).
- `process_cpu_usage` gauge. Загрузка ЦП процессом JVM (значение от 0 до 1).

- process_files_max_files gauge. Максимальное количество файловых дескрипторов.
- process_files_open_files gauge. Количество открытых файловых дескрипторов.
- process_start_time_seconds gauge. Время запуска процесса в формате Unix timestamp.
- process_uptime_seconds gauge. Время работы процесса с момента запуска (в секундах).

3.11 Метрики Spring Data Repository

- spring_data_repository_invocations_seconds_count{exception="...", method="...", repository="...", state="..."} counter. Количество вызовов методов репозитория.
- spring_data_repository_invocations_seconds_sum{exception="...", method="...", repository="...", state="..."} counter. Суммарное время выполнения методов репозитория (в секундах).
- spring_data_repository_invocations_seconds_max{exception="...", method="...", repository="...", state="..."} gauge. Максимальное время выполнения метода репозитория (в секундах).

3.12 Метрики безопасности (Spring Security)

3.12.1 Активная авторизация

- spring_security_authorizations_active_seconds_count{spring_security_authentication_type="...", spring_security_authorization_decision="...", spring_security_object="..."} counter. Количество активных проверок авторизации.
- spring_security_authorizations_active_seconds_sum{...} counter. Суммарное время активных проверок (в секундах).
- spring_security_authorizations_active_seconds_max{...} gauge. Максимальное время активной проверки.

3.12.2 Завершенная авторизация

- spring_security_authorizations_seconds_count{error="...", spring_security_authentication_type="...", spring_security_authorization_decision="...", spring_security_object="..."} counter. Количество проверок авторизации.
- spring_security_authorizations_seconds_sum{...} counter. Суммарное время проверок авторизации (в секундах).
- spring_security_authorizations_seconds_max{...} gauge. Максимальное время проверки авторизации.

3.12.3 Счетчики прохождения фильтров безопасности (часть 1)

- spring_security_filterchains_[FilterName]_after_total{security_reached_filter_section="after", spring_security_filterchain_position="...", spring_security_filterchain_size="...", spring_security_reached_filter_name="none"} counter. Количество запросов, прошедших после выполнения фильтра.
- spring_security_filterchains_[FilterName]_before_total{security_reached_filter_section="before", spring_security_filterchain_position="...", spring_security_filterchain_size="...", spring_security_reached_filter_name="none"} counter. Количество запросов, прошедших перед выполнением фильтра.

Примечание: [FilterName] заменяется на имя конкретного фильтра (например, AescaAuthenticationExceptionHandler, ApiKeyAuthenticationFilter, UserPrincipalAuthenticationFilter и др.). Набор фильтров зависит от конфигурации безопасности конкретного сервиса.

3.12.4 Активные фильтры безопасности

- `spring_security_filterchains_active_seconds_count{security_security_reached_filter_section="...", spring_security_filterchain_position="...", spring_security_filterchain_size="...", spring_security_reached_filter_name="..."}` counter. Количество активных выполнений фильтров безопасности.
- `spring_security_filterchains_active_seconds_sum{...}` counter. Суммарное время активных выполнений фильтров (в секундах).
- `spring_security_filterchains_active_seconds_max{...}` gauge. Максимальное время активного выполнения фильтра.

3.12.5 Счетчики прохождения фильтров безопасности (часть 2)

- `spring_security_filterchains_authentication_anonymous_after_total{...}` counter. Количество прохождений после фильтра `authentication_anonymous`.
- `spring_security_filterchains_authentication_anonymous_before_total{...}` counter. Количество прохождений перед фильтром `authentication_anonymous`.
- `spring_security_filterchains_authorization_after_total{...}` counter. Количество прохождений после фильтра `authorization`.
- `spring_security_filterchains_authorization_before_total{...}` counter. Количество прохождений перед фильтром `authorization`.
- `spring_security_filterchains_context_async_after_total{...}` counter. Количество прохождений после фильтра `context_async`.
- `spring_security_filterchains_context_async_before_total{...}` counter. Количество прохождений перед фильтром `context_async`.
- `spring_security_filterchains_context_holder_after_total{...}` counter. Количество прохождений после фильтра `context_holder`.
- `spring_security_filterchains_context_holder_before_total{...}` counter. Количество прохождений перед фильтром `context_holder`.
- `spring_security_filterchains_context_servlet_after_total{...}` counter. Количество прохождений после фильтра `context_servlet`.
- `spring_security_filterchains_context_servlet_before_total{...}` counter. Количество прохождений перед фильтром `context_servlet`.
- `spring_security_filterchains_header_after_total{...}` counter. Количество прохождений после фильтра `header`.
- `spring_security_filterchains_header_before_total{...}` counter. Количество прохождений перед фильтром `header`.
- `spring_security_filterchains_logout_after_total{...}` counter. Количество прохождений после фильтра `logout`.
- `spring_security_filterchains_logout_before_total{...}` counter. Количество прохождений перед фильтром `logout`.
- `spring_security_filterchains_requestcache_after_total{...}` counter. Количество прохождений после фильтра `requestcache`.
- `spring_security_filterchains_requestcache_before_total{...}` counter. Количество прохождений перед фильтром `requestcache`.

3.12.6 Время выполнения фильтров

- `spring_security_filterchains_seconds_count{error="...", security_security_reached_filter_section="...", spring_security_filterchain_position="...", spring_security_filterchain_size="...", spring_security_reached_filter_name="..."}` counter. Количество выполнений фильтров безопасности.

- `spring_security_filterchains_seconds_sum{...}` counter. Суммарное время выполнения фильтров (в секундах).
- `spring_security_filterchains_seconds_max{...}` gauge. Максимальное время выполнения фильтра (в секундах).

3.12.7 Счетчики прохождения фильтров безопасности (часть 3)

- `spring_security_filterchains_session_management_after_total{...}` counter. Количество прохождений после фильтра `session_management`.
- `spring_security_filterchains_session_management_before_total{...}` counter. Количество прохождений перед фильтром `session_management`.
- `spring_security_filterchains_session_urlencoding_after_total{...}` counter. Количество прохождений после фильтра `session_urlencoding`.
- `spring_security_filterchains_session_urlencoding_before_total{...}` counter. Количество прохождений перед фильтром `session_urlencoding`.

3.12.8 Защищенные запросы

- `spring_security_http_secured_requests_active_seconds_count` counter. Количество активных защищенных запросов.
- `spring_security_http_secured_requests_active_seconds_sum` counter. Суммарное время активных защищенных запросов (в секундах).
- `spring_security_http_secured_requests_active_seconds_max` gauge. Максимальное время активного защищенного запроса.
- `spring_security_http_secured_requests_seconds_count{error="..."}` counter. Количество защищенных HTTP-запросов.
- `spring_security_http_secured_requests_seconds_sum{error="..."}` counter. Суммарное время обработки защищенных запросов (в секундах).
- `spring_security_http_secured_requests_seconds_max{error="..."}` gauge. Максимальное время обработки защищенного запроса.

3.12.9 Незащищенные запросы

- `spring_security_http_unsecured_requests_active_seconds_count` counter. Количество активных незащищенных запросов.
- `spring_security_http_unsecured_requests_active_seconds_sum` counter. Суммарное время активных незащищенных запросов (в секундах).
- `spring_security_http_unsecured_requests_active_seconds_max` gauge. Максимальное время активного незащищенного запроса.
- `spring_security_http_unsecured_requests_seconds_count{error="..."}` counter. Количество незащищенных HTTP-запросов.
- `spring_security_http_unsecured_requests_seconds_sum{error="..."}` counter. Суммарное время обработки незащищенных запросов (в секундах).
- `spring_security_http_unsecured_requests_seconds_max{error="..."}` gauge. Максимальное время обработки незащищенного запроса.

3.13 Системные метрики CPU

- `system_cpu_count` gauge. Количество процессоров/ядер, доступных JVM.
- `system_cpu_usage` gauge. Общая загрузка ЦП системы (значение от 0 до 1).
- `system_load_average_1m` gauge. Средняя нагрузка на систему за 1 минуту.

3.14 Метрики планировщика задач

3.14.1 Активные задачи

- `tasks_scheduled_execution_active_seconds_count{code_function="...", code_namespace="...", exception="...", outcome="..."}` counter. Количество активных выполнений запланированных задач.
- `tasks_scheduled_execution_active_seconds_sum{...}` counter. Суммарное время активных выполнений.
- `tasks_scheduled_execution_active_seconds_max{...}` gauge. Максимальное время активного выполнения.

3.14.2 Завершенные задачи

- `tasks_scheduled_execution_seconds_count{code_function="...", code_namespace="...", error="...", exception="...", outcome="..."}` counter. Количество выполнений запланированных задач.
- `tasks_scheduled_execution_seconds_sum{...}` counter. Суммарное время выполнения запланированных задач (в секундах).
- `tasks_scheduled_execution_seconds_max{...}` gauge. Максимальное время выполнения запланированной задачи.

3.15 Метрики Tomcat-сессий

- `tomcat_sessions_active_current_sessions` gauge. Текущее количество активных HTTP-сессий.
- `tomcat_sessions_active_max_sessions` gauge. Максимальное количество одновременных активных сессий.
- `tomcat_sessions_alive_max_seconds` gauge. Максимальное время жизни сессии.
- `tomcat_sessions_created_sessions_total` counter. Общее количество созданных сессий.
- `tomcat_sessions_expired_sessions_total` counter. Количество истекших сессий.
- `tomcat_sessions_rejected_sessions_total` counter. Количество отклоненных сессий.

4 ОПИСАНИЕ МЕТОДОВ REST API ВЕРСИИ 4

4.1 Методы работы с субъектами

4.1.1 Метод поиска субъектов

GET API – Поиск субъектов	
Метод доступен администратору и оператору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20субъекты/findAll_2	
URL – subjects-service/api/v4/public/subjects	
Query	
{	
search (string) [опционально],	Полнотекстовый поиск (имя субъекта)
isBlocked (boolean) [опционально],	Флаг: субъект заблокирован в ресурсной системе
isConnected (boolean) [опционально],	Флаг: субъект подключен к ресурсной системе
id (UUID[]) [опционально],	ID субъекта
notId (UUID[]) [опционально],	Исключая ID субъекта
securityGroupId (UUID[]) [опционально],	ID группы безопасности
resourceId (UUID[]) [опционально],	ID ресурсной системы
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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
certificatesCount (integer),	Количество сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
sid (string)	SID субъекта
}	

4.1.2 Метод получения субъекта по идентификатору

GET API – Получение субъекта по идентификатору	
Метод доступен администратору и оператору при наличии полномочий на управление субъектом, идентификатор которого передается во входных параметрах	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv4%5D%20Контроллер%3A%20субъекты/findById_3	
URL – subjects-service/api/v4/public/subjects/{id}	
Query	
{	
id (UUID)	ID субъекта
}	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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
certificatesCount (integer),	Количество сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)

sid (string)	SID субъекта
}	

4.1.3 Метод создания и изменения субъекта

PUT API – Создание и изменение субъекта	
Метод доступен администратору и оператору при наличии полномочий	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20субъекты/update	
URL – subjects-service/api/v4/public/subjects	
Query	
-	
Request	
{	
id (UUID) [опционально],	Идентификатор субъекта
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] } [опционально],	Поля разделенного имени субъекта
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[] } [опционально]	Поля альтернативного имени субъекта
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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,	Поля альтернативного имени субъекта

SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): {	
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
certificatesCount (integer),	Количество сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
sid (string)	SID субъекта
}	

4.1.4 Методы создания и изменения субъекта на основании запроса pkcs#10

4.1.4.1 Метод создания и изменения субъекта на основании запроса pkcs#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

PUT API – Создание и изменение субъекта на основании запроса pkcs#10 (multipart/form-data)	
Метод доступен администратору и оператору при наличии полномочий	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html/#/%5Bv4%5D%20Контроллер%3A%20субъекты/updateByPkcs10AsMultipartFile_1	
URL – subjects-service/api/v4/public/subjects/pkcs10	
Query	-
Request	{
id (UUID) [опционально],	Идентификатор субъекта
request (binary),	Файл запроса на сертификат. Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE-----").
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[] } [опционально]	Поля альтернативного имени субъекта
}	
Response	Ответ JSON в HTTP-body
ResponseEntity -> ItemResponse -> {	
id (UUID),	ID субъекта
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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
certificatesCount (integer),	Количество сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
sid (string)	SID субъекта
}	

4.1.4.2 Метод создания и изменения субъекта на основании запроса pkcs#10 (формат запроса на сертификат из входных параметров – application/json)

Примечание: отличается от метода выше форматом передаваемого во входных параметрах запроса на сертификат. В данном методе на вход поступает содержимое файла запроса на сертификат в формате PEM (содержимое в Base64).

PUT API – Создание и изменение субъекта на основании запроса pkcs#10 (application/json)	
Метод доступен администратору и оператору при наличии полномочий	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv4%5D%20Контроллер%3A%20субъекты/updateByPkcs10AsMultipartFile_1	
URL – subjects-service/api/v4/public/subjects/pkcs10	
Query	
-	
Request	
{	
id (UUID) [опционально],	Идентификатор субъекта
request: {	Файл запроса на сертификат
contentType (string) [опционально],	Тип загружаемого файла (HTTP MediaType) – application/octet-stream)
fileName (string) [опционально],	Имя загружаемого файла
data (string:binary)	Содержимое PEM файла запроса на сертификат (массив байт в Base64). Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
},	

<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[] } [опционально] </pre>	Поля альтернативного имени субъекта
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта
commonName (string)	Имя субъекта
distinguishedName (string),	Расположение субъекта в ресурсной системе
resource: {	Ресурсы
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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
certificatesCount (integer),	Количество сертификатов
modify (instant),	Время изменения (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
sid (string)	SID субъекта
}	

4.2 Методы работы с шаблонами сертификатов

4.2.1 Метод поиска шаблонов

GET API – Поиск шаблонов	
Метод доступен администратору и оператору. В ответе для оператора содержатся только те шаблоны, на использование которых ему предоставлены полномочия.	
URL – templates-service/api/v4/public/templates	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv4%5D%20Контроллер%3A%20шаблоны/findAll_5	
Query	
{	
types (enum[]: EMBEDDED, CLONED, IMPORTED, UNKNOWN) [опционально],	Тип шаблона
certificateType (enum[]: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN) [опционально],	Тип выпускаемого сертификата
endEntityType (enum[]: USER, DEVICE, ROOT_CA, SUB_CA, UNKNOWN) [опционально],	Тип субъекта
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 ¹
extendedKeyUsage (string[]) [опционально],	Фильтр: расширенное использование ключа
isCertificateAuthorityIdEmpty (boolean) [опционально],	Фильтр: ID издающего ЦС не задан
issuePkcs12Allow (boolean) [опционально],	Фильтр: Шаблон позволяет выпуск PKCS#12
deprecated (boolean) [опционально],	Фильтр: Устаревший шаблон
keyUsage (enum[]: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPT, DATA_ENCRYPT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY, UNKNOWN) [опционально],	Фильтр: использование ключа
keyUsageMatchAll (boolean) [опционально],	Фильтр: шаблон должен содержать все указанные значения использования ключа
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID шаблона
name (string),	Имя шаблона
type (enum: EMBEDDED, CLONED, IMPORTED, UNKNOWN),	Тип шаблона
certificateType (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип выпускаемого сертификата
certificateAuthorityId (UUID),	ID ЦС, который должен использоваться при выпуске сертификата по данному шаблону
endEntityType (enum: USER, DEVICE, ROOT_CA, SUB_CA, UNKNOWN),	Тип субъекта
certificateCount (int64),	Число выпущенных по шаблону сертификатов
removed (boolean),	Флаг: шаблон удален
updated (instant),	Время обновления (ISO 8601)

¹ 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.

created (instant),	Время создания (ISO 8601)
appendSubjectSid (boolean),	Флаг: включать SID субъекта в сертификат
pkcs12Policies: {	Политики выпуска PKCS#12
passwordRegex (string),	Регулярное выражение для пароля PKCS#12
issueAllow(boolean)	Флаг возможности выпуска PKCS#12
},	
subjectValidationEnabled(boolean),	Флаг: контролировать соответствие полей в сертификате атрибутам субъекта
deprecated (boolean)	Флаг: Устаревший шаблон
}	

4.2.2 Метод получения шаблона по идентификатору

GET API – Получение шаблона по идентификатору	
Метод доступен:	
<ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на использование шаблона, идентификатор которого передается во входных параметрах. 	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv4%5D%20Контроллер%3A%20шаблоны/findById	
URL – templates-service/api/v4/public/templates/{id}	
Query	
{	
id (UUID)	ID шаблона
}	
Request	-
Response	Ответ JSON в HTTP-body
ResponseEntity -> ItemResponse -> {	
id (UUID),	ID шаблона
name (string),	Имя шаблона
type (enum: EMBEDDED, CLONED, IMPORTED, UNKNOWN),	Тип шаблона
certificateType (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип выпускаемого сертификата
certificateAuthorityId (UUID),	ID ЦС, который должен использоваться при выпуске сертификата по данному шаблону
endEntityType (enum: USER, DEVICE, ROOT_CA, SUB_CA, UNKNOWN),	Тип субъекта
removed (boolean),	Флаг: шаблон удален
validity (int64),	Время действия выпускаемого сертификата (мс)
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_ENIPHERMENT, DATA_ENIPHERMENT, KEY AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENIPHER_ONLY, DECIPHER_ONLY, UNKNOWN)	Значение расширения

<code>},</code>	
<code>extendedKeyUsages: {</code>	Расширенное назначение ключа сертификата
<code>critical (boolean),</code>	Флаг: расширение критическое
<code>values (string[])</code>	Значение расширения (OIDs)
<code>},</code>	
<code>policies: {</code>	Политики сертификата
<code>critical (boolean),</code>	Флаг: расширение критическое
<code>values (string[])</code>	Значение расширения (OIDs)
<code>},</code>	
<code>subjectDN: [{</code>	Имя субъекта сертификата
<code>index (int32),</code>	Индекс (для сортировки, по умолчанию – 0)
<code>name (string),</code>	Имя компонента
<code>description (string),</code>	Описание компонента
<code>required (boolean),</code>	Флаг: обязателен к заполнению
<code>validation (boolean),</code>	Флаг: валидация значения
<code>modifiable (boolean),</code>	Флаг: доступен к редактированию
<code>defaultValue (string),</code>	Значение по умолчанию
<code>regex (string),</code>	Регулярное значение для валидации значения
<code>alert (string),</code>	Предупреждение о неудачной валидации значения
<code>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, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN)</code>	Код компонента
<code>}],</code>	
<code>subjectAltName: [{</code>	Расширенное имя субъекта сертификата
<code>index (int32),</code>	Индекс (для сортировки, по умолчанию – 0)
<code>name (string),</code>	Имя компонента
<code>description (string),</code>	Описание компонента
<code>required (boolean),</code>	Флаг: обязателен к заполнению
<code>validation (boolean),</code>	Флаг: валидация значения
<code>modifiable (boolean),</code>	Флаг: доступен к редактированию
<code>defaultValue (string),</code>	Значение по умолчанию
<code>regex (string),</code>	Регулярное значение для валидации значения
<code>alert (string),</code>	Предупреждение о неудачной валидации значения
<code>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),</code>	Код компонента
<code>generalName (int32),</code>	Идентификатор компонента в RFC
<code>oid (string)</code>	OID компонента в RFC
<code>}],</code>	
<code>updated (instant),</code>	Время обновления (ISO 8601)
<code>created (instant),</code>	Время создания (ISO 8601)
<code>appendSubjectSid (boolean),</code>	Флаг: включать SID субъекта в сертификат
<code>publication (boolean),</code>	Флаг: публиковать сертификат в PC
<code>pkcs12Policies: {</code>	Политики выпуска PKCS#12
<code>passwordRegex (string),</code>	Регулярное значение для валидации пароля PKCS#12
<code>issueAllow (boolean)</code>	Флаг: Шаблон позволяет выпуск PKCS#12
<code>},</code>	
<code>allowThrowaway (boolean),</code>	Флаг: Шаблон позволяет throwaway-сертификата
<code>appendIssuerSignTool (boolean),</code>	Флаг: Включать сведения о средствах ЭП и УЦ издателя
<code>issuerSignTool: {</code>	
<code>signTool (string),</code>	Средство электронной подписи
<code>caTool (string),</code>	Средство УЦ
<code>signToolCert (string),</code>	Заключение на средство ЭП
<code>caToolCert (string)</code>	Заключение на средство УЦ
<code>},</code>	
<code>appendSubjectSignTool (boolean),</code>	Флаг: Включать сведения о средстве ЭП владельца

subjectSignTool: {	Средство электронной подписи владельца
signTool (string)	Средство электронной подписи владельца
},	
subjectValidationEnabled(boolean),	Флаг: контролировать соответствие полей в сертификате атрибутам субъекта
deprecated (boolean),	Флаг: Устаревший шаблон
version (int64)	Версия шаблона
}	

4.3 Методы работы с Центрами сертификации

4.3.1 Метод получения активного ЦС

GET API – Получение активного ЦС	
Метод доступен администратору и оператору.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20Центры%20сертификации/active	
URL – certificate-authority-service/api/v4/public/certificate-authorities/active	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
ResponseEntity -> ItemResponse -> {	
id (UUID),	ID ЦС
isActive (boolean),	Флаг: активный ЦС
active (boolean),	Флаг: активный ЦС
isManagement (boolean),	Флаг: технологический ЦС
management (boolean),	Флаг: технологический ЦС
certificate: {	Сертификат ЦС
id (UUID),	Идентификатор сертификата ЦС
issuerId (UUID),	Идентификатор издателя сертификата ЦС
issuerFingerprint (string),	Фингерпринт издателя сертификата ЦС
serialnumber (string),	Серийный номер сертификата ЦС
fingerprint (string),	Фингерпринт сертификата ЦС
issuerDN: {	Имя субъекта издателя сертификата ЦС
(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectDN: {	Имя субъекта сертификата ЦС
(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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 ЦС
(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
},	
name (string),	Имя сертификата ЦС (на основе CN)
templateId (UUID),	Идентификатор шаблона
templateName (string),	Имя шаблона
validFrom (instant),	Дата начала действия сертификата ЦС (ISO 8601)
validTo (instant),	Дата окончания действия сертификата ЦС (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN),	Статус сертификата ЦС
isValid (boolean),	Флаг: сертификат ЦС действует
isExpired (boolean),	Флаг: сертификат ЦС истек
actions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10

pem (boolean)	Флаг: выгрузка сертификата
},	
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int32),	Код причины отзыва
value (string)	Значение причины отзыва
},	
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
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата ЦС
keyBits (int32),	Длина ключа сертификата ЦС
subjectKeyIdentifier (string),	Идентификатор ключа сертификата ЦС
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
},	
chain: {	Цепочка сертификатов ЦС (рекурсивный объект)
id (UUID),	Идентификатор сертификата
name (string),	Имя сертификата (на основе CN)
subjectDN: {	Имя субъекта сертификата ЦС
(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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)
},	
certificateCount (int64),	Число выпущенных сертификатов
title (string),	Отображаемое имя ЦС
cryptographyProviders: {	Конфигурация криптопровайдеров алгоритмов ЦС
(enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN): {	Название алгоритма
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO, ALADDIN_JCP, UNKNOWN),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
primaryCryptographyProvider: {	Криптопровайдер закрытого ключа
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO, ALADDIN_JCP, UNKNOWN),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
isAvailable (boolean),	Флаг: Доступность ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

4.3.2 Метод получения ЦС по идентификатору

GET API – Получение ЦС по идентификатору	
Метод доступен администратору.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20Центры%20сертификации/findById_15	
URL – certificate-authority-service/api/v4/public/certificate-authorities/{id}	
Query	
{	
id (UUID)	ID ЦС
}	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body

id (UUID),	ID ЦС
isActive (boolean),	Флаг: активный ЦС
active (boolean),	Флаг: активный ЦС
isManagement (boolean),	Флаг: технологический ЦС
management (boolean),	Флаг: технологический ЦС
certificate: {	Сертификат ЦС
id (UUID),	Идентификатор сертификата ЦС
issuerId (UUID),	Идентификатор издателя сертификата ЦС
issuerFingerprint (string),	Фингерпринт издателя сертификата ЦС
serialnumber (string),	Серийный номер сертификата ЦС
fingerprint (string),	Фингерпринт сертификата ЦС
issuerDN: {	Имя субъекта издателя сертификата ЦС
(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectDN: {	Имя субъекта сертификата ЦС
(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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 ЦС
(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
},	
name (string),	Имя сертификата ЦС (на основе CN)
templateId (UUID),	Идентификатор шаблона
templateName (string),	Имя шаблона
validFrom (instant),	Дата начала действия сертификата ЦС (ISO 8601)
validTo (instant),	Дата окончания действия сертификата ЦС (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN),	Статус сертификата ЦС
isValid (boolean),	Флаг: сертификат ЦС действует
isExpired (boolean),	Флаг: сертификат ЦС истек
actions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int32),	Код причины отзыва
value (string)	Значение причины отзыва
},	
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
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата ЦС
keyBits (int32),	Длина ключа сертификата ЦС
subjectKeyIdentifier (string),	Идентификатор ключа сертификата ЦС
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
},	
chain: {	Цепочка сертификатов ЦС (рекурсивный объект)
id (UUID),	Идентификатор сертификата
name (string),	Имя сертификата (на основе CN)
subjectDN: {	Имя субъекта сертификата ЦС

(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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)
},	
certificateCount (int64),	Число выпущенных сертификатов
title (string),	Отображаемое имя ЦС
cryptographyProviders: {	Конфигурация криптопровайдеров алгоритмов ЦС
(enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN): {	Название алгоритма
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO, ALADDIN_JCP, UNKNOWN),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
primaryCryptographyProvider: {	Криптопровайдер закрытого ключа
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO, ALADDIN_JCP, UNKNOWN),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
isAvailable (boolean),	Флаг: Доступность ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

4.3.3 Метод получения Центров сертификации

GET API – Получение ЦС	
Метод доступен администратору.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20Центры%20сертификации/find All_15	
URL – certificate-authority-service/api/v4/public/certificate-authorities	
Query	
{	
status (enum[]:ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN) [опционально],	Статус сертификата ЦС
type (enum[]: CERTIFICATE1, ROOT_CA, SUB_CA, UNKNOWN) [опционально],	Тип сертификата ЦС
search (string) [опционально],	Полнотекстовый поиск по имени ЦС
isManagement (boolean) [опционально],	Флаг: технологический ЦС
isActive (boolean) [опционально],	Флаг: активный ЦС
isValid (boolean) [опционально],	Флаг: сертификат ЦС действителен
isExpired (boolean) [опционально],	Флаг: сертификат ЦС истек
id (UUID[]) [опционально],	Id ЦС
notIds (UUID[]) [опционально],	Исключая ID ЦС
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
endEntityType (enum[]: ROOT_CA, SUB_CA, USER, DEVICE, UNKNOWN) [опционально],	Тип субъекта
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	
Request	
-	

1 Тип `CERTIFICATE` является общим для всех словарей типов сертификатов в программе. При использовании данного метода указание данного типа также доступно, однако сертификаты ЦС с данным типом отсутствуют, соответственно не будут найдены и возвращены в ответе.

Response	Ответ JSON в HTTP-body
ResponseEntity -> CollectionResponse -> {	
id (UUID),	ID ЦС
isActive (boolean),	Флаг: активный ЦС
active (boolean),	Флаг: активный ЦС
isManagement (boolean),	Флаг: технологический ЦС
management (boolean),	Флаг: технологический ЦС
certificate: {	Сертификат ЦС
id (UUID),	Идентификатор сертификата ЦС
issuerId (UUID),	Идентификатор издателя сертификата ЦС
issuerFingerprint (string),	Фингерпринт издателя сертификата ЦС
serialnumber (string),	Серийный номер сертификата ЦС
fingerprint (string),	Фингерпринт сертификата ЦС
issuerDN: {	Имя субъекта издателя сертификата ЦС
(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectDN: {	Имя субъекта сертификата ЦС
(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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 ЦС
(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
},	
name (string),	Имя сертификата ЦС (на основе CN)
templateId (UUID),	Идентификатор шаблона
templateName (string),	Имя шаблона
validFrom (instant),	Дата начала действия сертификата ЦС (ISO 8601)
validTo (instant),	Дата окончания действия сертификата ЦС (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN),	Статус сертификата ЦС
isValid (boolean),	Флаг: сертификат ЦС действует
isExpired (boolean),	Флаг: сертификат ЦС истек
actions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean),	Флаг: выгрузка сертификата
},	
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int32),	Код причины отзыва
value (string),	Значение причины отзыва
},	
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
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата ЦС

keyBits (int32),	Длина ключа сертификата ЦС
subjectKeyIdentifier (string),	Идентификатор ключа сертификата ЦС
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
},	
chain: {	Цепочка сертификатов ЦС (рекурсивный объект)
id (UUID),	Идентификатор сертификата
name (string),	Имя сертификата (на основе CN)
subjectDN: {	Имя субъекта сертификата ЦС
(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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)
},	
certificateCount (int64),	Число выпущенных сертификатов
title (string),	Отображаемое имя ЦС
cryptographyProviders: {	Конфигурация криптопровайдеров алгоритмов ЦС
(enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN): {	Название алгоритма
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO, ALADDIN_JCP, UNKNOWN),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
primaryCryptographyProvider: {	Криптопровайдер закрытого ключа
cryptographyProvider (enum: DEFAULT, CRYPTO_PRO, ALADDIN_JCP, UNKNOWN),	Название криптопровайдера
isAvailable (boolean),	Флаг: Доступность криптопровайдера
},	
isAvailable (boolean),	Флаг: Доступность ЦС
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

4.4 Методы работы с сертификатами

4.4.1 Метод выпуска сертификата в контейнере pkcs#12

POST API – Выпуск сертификата в контейнере pkcs#12	
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление субъектами и использование шаблона, идентификаторы которых передаются во входных параметрах. <p>Использование данного метода оператором для создания сертификатов для учетных записей запрещено.</p>	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20сертификаты/enrollByCald	
URL – certificate-authority-service/api/v4/public/certificates/enroll/{cald}	
Query	
{	
caId (UUID),	ID ЦС
subjectId (UUID) [обязателен, если не указан userId],	ID субъекта
userId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона
subjectDN: {	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.
(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.
},	
subjectAltName: {	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.
(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[]	При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.
},	
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 сертификата
downloadActions: {	Доступные действия по выгрузке

p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
fingerprint (string),	Фингерпринт шаблона
serialnumber (string),	Серийный номер сертификата
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
name (string),	Имя сертификата (на основе CN)
issuerDN: { (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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] },	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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: { (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
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant)	Дата окончания действия сертификата (ISO 8601)
}	

4.4.2 Методы выпуска сертификата по запросу pkcs#10

4.4.2.1 Выпуск сертификата по запросу pkcs#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

POST API – Выпуск сертификата по запросу pkcs#10 (multipart/form-data)	
Метод доступен:	
<ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление субъектами и использование шаблона, идентификаторы которых передаются во входных параметрах. 	
Использование данного метода оператором для создания сертификатов для учетных записей запрещено.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20сертификаты/enrollRequestByCald_1	
URL – certificate-authority-service/api/v4/public/certificates/enroll/{cald}/pkcs10	
Query	
{	
caId (UUID),	ID ЦС

subjectId (UUID) [обязателен, если не указан userId],	ID субъекта
userId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request {	
templateId (UUID),	Идентификатор шаблона
request (binary),	<p>Файл запроса на сертификат.</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» значения полей запроса на сертификат должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта. Допустимые форматы запроса на сертификат:</p> <ul style="list-style-type: none"> • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] }, [опционально]	<p>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.</p> <p>Значения полей, указанные в subjectName, переопределяют значения соответствующих полей SDN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p>
subjectAltName: { (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPF_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[] } [опционально]	<p>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.</p> <p>Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p>
}	
Response ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID сертификата
downloadActions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
fingerprint (string),	Фингерпринт шаблона

serialnumber (string),	Серийный номер сертификата
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
name (string),	Имя сертификата (на основе CN)
issuerDN: { (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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] },	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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: { (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
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant)	Дата окончания действия сертификата (ISO 8601)
}	

4.4.2.2 Выпуск сертификата по запросу pkcs#10 (формат запроса на сертификат из входных параметров – application/json)

Примечание: отличается от метода выше форматом передаваемого во входных параметрах запроса на сертификат. В данном методе на вход поступает содержимое файла запроса на сертификат в формате PEM (содержимое в Base64).

POST API – Выпуск сертификата по запросу pkcs#10 (application/json)	
Метод доступен:	
<ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление субъектами и использование шаблона, идентификаторы которых передаются во входных параметрах. 	
Использование данного метода оператором для создания сертификатов для учетных записей запрещено.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv4%5D%20Контроллер%3A%20сертификаты/enrollRequestByCald_1	
URL – certificate-authority-service/api/v4/public/certificates/enroll/{cald}/pkcs10	
Query	
{	
caId (UUID),	ID ЦС
subjectId (UUID) [обязателен, если не указан userId],	ID субъекта
userId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	

Request {	
templateId(UUID),	Идентификатор шаблона
request: {	Запрос на сертификат
contentType(string) [опционально],	Тип загружаемого файла (HTTP MediaType) - application/octet-stream)
fileName (string) [опционально],	Имя загружаемого файла
data (string:binary)	Содержимое PEM файла запроса на сертификат (массив байт в Base64). При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» значения полей запроса на сертификат должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта. Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
},	
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] }, [опционально]	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. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.
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[] } [опционально]	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. Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.
}	
Response ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID сертификата
downloadActions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
fingerprint (string),	Фингерпринт шаблона
serialnumber (string),	Серийный номер сертификата

templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
name (string),	Имя сертификата (на основе CN)
issuerDN: { (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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] },	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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: { (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
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant)	Дата окончания действия сертификата (ISO 8601)
}	

4.4.3 Методы перевыпуска сертификата по запросу pkcs#10

4.4.3.1 Перевыпуск сертификата по запросу pkcs#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

POST API – Перевыпуск сертификата по запросу pkcs#10 (multipart/form-data)
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление субъектами и использование шаблона, идентификаторы которых передаются во входных параметрах. <p>Использование данного метода оператором для создания сертификатов для учетных записей запрещено.</p> <p>Метод позволяет перевыпустить сертификат по запросу, по которому ранее уже был выпущен сертификат.</p> <p>При использовании метода проверяется наличие в базе данных программы выпущенного сертификата, имеющего «Subject Key Identifier» аналогичный указанному в запросе на сертификат из входных параметров. Если такой сертификат не будет найден, данный метод осуществит выпуск нового сертификата по запросу аналогично методу выпуска сертификата по запросу.</p> <p>Если сертификат с аналогичным указанному в запросе на сертификат «Subject Key Identifier» будет найден, программа:</p> <ol style="list-style-type: none"> 1) проверит статус найденного сертификата. Если срок действия данного сертификата истек, запрос пользователя будет отклонен.

<p>2) проверит соответствие значений в полях SDN и SAN, указанных в запросе (или во входных параметрах метода), значениям в полях SDN и SAN в найденном сертификате. При соответствии значений будет осуществлен выпуск сертификата по запросу на сертификат из входных параметров метода, иначе запрос пользователя будет отклонен.</p>	
<p>Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20сертификаты/renewalRequestByCald_1</p>	
<p>URL – certificate-authority-service/api/v4/public/certificates/renewal/{cald}/pkcs10</p>	
<p>Query</p>	
<pre>{ caId (UUID), subjectId (UUID) [обязателен, если не указан userId], userId (UUID) [обязателен, если не указан subjectId] }</pre>	<p>ID ЦС</p> <p>ID субъекта</p> <p>ID учетной записи</p>
<p>Request</p>	
<pre>templateId (UUID), request (binary),</pre>	<p>Идентификатор шаблона</p> <p>Файл запроса на сертификат. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» значения полей запроса на сертификат должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта. Допустимые форматы запроса на сертификат:</p> <ul style="list-style-type: none"> • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE-----").
<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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] }, [опционально]</pre>	<p>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. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p>
<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[] } [опционально]</pre>	<p>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. Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта. При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с</p>

	регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID сертификата
downloadActions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
fingerprint (string),	Фингерпринт шаблона
serialnumber (string),	Серийный номер сертификата
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
name (string),	Имя сертификата (на основе CN)
issuerDN: { (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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] },	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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: { (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPF_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
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant)	Дата окончания действия сертификата (ISO 8601)
}	

4.4.3.2 Перевыпуск сертификата по запросу pkcs#10 (формат запроса на сертификат из входных параметров – application/json)

Примечание: отличается от метода выше форматом передаваемого во входных параметрах запроса на сертификат. В данном методе на вход поступает содержимое файла запроса на сертификат в формате PEM (содержимое в Base64).

POST API – Перевыпуск сертификата в по запросу pkcs#10 (application/json)
Метод доступен: – администратору;

<p>– оператору при наличии полномочий на управление субъектами и использование шаблона, идентификаторы которых передаются во входных параметрах.</p> <p>Использование данного метода оператором для создания сертификатов для учетных записей запрещено.</p>	
<p>Метод позволяет перевыпустить сертификат по запросу, по которому ранее уже был выпущен сертификат.</p> <p>При использовании метода проверяется наличие в базе данных программы выпущенного сертификата, имеющего «Subject Key Identifier» аналогичный указанному в запросе на сертификат из входных параметров. Если такой сертификат не будет найден, данный метод осуществит выпуск нового сертификата по запросу аналогично методу выпуска сертификата по запросу.</p> <p>Если сертификат с аналогичным указанному в запросе на сертификат «Subject Key Identifier» будет найден, программа:</p> <p>1) проверит статус найденного сертификата. Если срок действия данного сертификата истек, запрос пользователя будет отклонен.</p> <p>2) проверит соответствие значений в полях SDN и SAN, указанных в запросе (или во входных параметрах метода), значениям в полях SDN и SAN в найденном сертификате. При соответствии значений будет осуществлен выпуск сертификата по запросу на сертификат из входных параметров метода, иначе запрос пользователя будет отклонен.</p>	
<p>Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20сертификаты/renewalRequestByCald_1</p>	
<p>URL – certificate-authority-service/api/v4/public/certificates/renewal/{cald}/pkcs10</p>	
Query	
{	
caId (UUID),	ID ЦС
subjectId (UUID) [обязателен, если не указан userId],	ID субъекта
userId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона
request: {	Запрос на сертификат
contentType(string) [опционально],	Тип загружаемого файла (HTTP MediaType) - application/octet-stream)
fileName (string) [опционально],	Имя загружаемого файла
data (string:binary)	Содержимое PEM файла запроса на сертификат (массив байт в Base64). При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» значения полей запроса на сертификат должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта. Допустимые форматы запроса на сертификат:
	<ul style="list-style-type: none"> • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
},	
subjectName: {	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.
(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN):	Значения полей, указанные в subjectName, переопределяют значения соответствующих полей SDN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.
string[]	
}, [опционально]	

	При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам 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[] } [опционально] </pre>	<p>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.</p> <p>Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.</p> <p>При использовании шаблона с включенной опцией «Контролировать соответствие полей в сертификате атрибутам субъекта» указываемые значения полей должны соответствовать значениям аналогичных атрибутов субъекта. Отключение данной опции в шаблоне позволяет записывать в поля сертификата любые (в соответствии с регулярными выражениями полей) значения, не соответствующие атрибутам субъекта.</p>
}	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID сертификата
downloadActions: {	Доступные действия по выгрузке
pl2 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
fingerprint (string),	Фингерпринт шаблона
serialnumber (string),	Серийный номер сертификата
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
name (string),	Имя сертификата (на основе CN)
<pre> issuerDN: { (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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] }, </pre>	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
<pre> 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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] }, </pre>	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
<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 из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
validFrom (instant),	Дата начала действия сертификата (ISO 8601)

validTo (instant)	Дата окончания действия сертификата (ISO 8601)
}	

4.4.4 Методы валидации запроса pkcs#10

4.4.4.1 Метод валидации запроса pkcs#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

POST API – Валидация запроса pkcs#10 (multipart/form-data)	
Метод доступен:	
<ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление субъектами и использование шаблона, идентификаторы которых передаются во входных параметрах. 	
Использование данного метода оператором для валидации запросов на сертификат для учетных записей должно быть запрещено.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv4%5D%20Контроллер%3A%20сертификаты/validate_1	
URL – certificate-authority-service/api/v4/public/certificates/validate/{caId}/pkcs10	
Query	
{	
caId (UUID)	ID ЦС
subjectId (UUID) [обязателен, если не указан accountId],	ID субъекта
accountId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона
request (binary),	Файл запроса на сертификат. Допустимые форматы запроса на сертификат: <ul style="list-style-type: none"> • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE-----").
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] }, [опционально]	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.
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[] } [опционально]	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. 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
name (string),	Имя сертификата (на основе CN)
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
subjectId (UUID)	ID субъекта (может отсутствовать, если в Query указан accountId, а не subjectId)

valid (boolean),	Флаг: запрос прошел валидацию
subjectNames: [{	Компоненты имени субъекта сертификата
fieldName (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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN),	Имя компонента
fieldValue (string),	Значение компонента
required (boolean),	Флаг: компонент обязательный
additional (boolean),	Флаг: компонент дополнительный
valid (boolean),	Флаг: компонент прошел валидацию
message (string)	Дополнительное сообщение
}],	
subjectAltNames: [{	Компоненты расширенного имени субъекта сертификата
fieldName (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),	Имя компонента
fieldValue (string),	Значение компонента
required (boolean),	Флаг: компонент обязательный
additional (boolean),	Флаг: компонент дополнительный
valid (boolean),	Флаг: компонент прошел валидацию
message (string)	Дополнительное сообщение
}]	
}	

4.4.4.2 Метод валидации запроса pkcs#10 (формат запроса на сертификат из входных параметров – application/json)

Примечание: отличается от метода выше форматом передаваемого во входных параметрах запроса на сертификат. В данном методе на вход поступает содержимое файла запроса на сертификат в формате PEM (содержимое в Base64).

POST API – Валидация запроса pkcs#10 (application/json)	
Метод доступен:	
<ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на управление субъектами и использование шаблона, идентификаторы которых передаются во входных параметрах. 	
Использование данного метода оператором для валидации запросов на сертификат для учетных записей запрещено.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5Bv4%5D%20Контроллер%3A%20сертификаты/validate_1	
URL – certificate-authority-service/api/v4/public/certificates/validate/{caId}/pkcs10	
Query	
{	
caId (UUID)	ID ЦС
subjectId (UUID) [обязателен, если не указан accountId],	ID субъекта
accountId (UUID) [обязателен, если не указан subjectId]	ID учетной записи
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона
request: {	Файл запроса на сертификат

contentType (string),	Тип загружаемого файла (HTTP MediaType) - application/octet-stream)
fileName (string),	Имя загружаемого файла
data (string:binary)	Содержимое загружаемого файла (массив байт в Base64). Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
},	
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] }, [опционально]	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.
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[] } [опционально]	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. 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
name (string),	Имя сертификата (на основе CN)
templateId (UUID),	ID шаблона
templateName (string),	Имя шаблона
subjectId (UUID)	ID субъекта (может отсутствовать, если в Query указан accountId, а не subjectId)
valid (boolean),	Флаг: запрос прошел валидацию
subjectNames: [{	Компоненты имени субъекта сертификата
fieldName (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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN),	Имя компонента
fieldValue (string),	Значение компонента
required (boolean),	Флаг: компонент обязательный
additional (boolean),	Флаг: компонент дополнительный
valid (boolean),	Флаг: компонент прошел валидацию
message (string)	Дополнительное сообщение
}],	
subjectAltNames: [{	Компоненты расширенного имени субъекта сертификата
fieldName (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),	Имя компонента
fieldValue (string),	Значение компонента
required (boolean),	Флаг: компонент обязательный
additional (boolean),	Флаг: компонент дополнительный
valid (boolean),	Флаг: компонент прошел валидацию
message (string)	Дополнительное сообщение
}]	
}	

4.4.5 Метод поиска сертификатов

GET API – Поиск сертификатов	
Метод доступен администратору и оператору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20сертификаты/findAll_14	
URL – certificate-authority-service/api/v4/public/certificates	
Query	
{	
search (string) [опционально],	Полнотекстовый поиск (имя или серийный номер)
issuerId (UUID) [опционально],	ID сертификата издателя
templateId (UUID) [опционально],	ID шаблона
status (enum[]: ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN) [опционально],	Статус сертификата
type (enum[]: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN) [опционально],	Тип сертификата
revocationReason (enum[]: UNSPECIFIED, KEY_COMPROMISE, CA_COMPROMISE, AFFILIATION_CHANGED, SUPERSEDED, CESSATION_OF_OPERATION, CERTIFICATE_HOLD, REMOVE_FROM_CRL, PRIVILEGE_WITHDRAWN, AA_COMPROMISE, UNKNOWN) [опционально],	Причина отзыва
notRevocationReason (enum[]: UNSPECIFIED, KEY_COMPROMISE, CA_COMPROMISE, AFFILIATION_CHANGED, SUPERSEDED, CESSATION_OF_OPERATION, CERTIFICATE_HOLD, REMOVE_FROM_CRL, PRIVILEGE_WITHDRAWN, AA_COMPROMISE, UNKNOWN) [опционально],	Исключая причину отзыва
revocationDateFrom (instant) [опционально],	Дата отзыва (начало)
revocationDateTo (instant) [опционально],	Дата отзыва (окончание)
hasRevocationReason (boolean) [опционально],	Флаг: наличие причины отзыва
hasRequest (boolean) [опционально],	Флаг: наличие pkcs10
hasCA (boolean) [опционально],	Флаг: наличие ЦС
isManagementCA (boolean) [опционально],	Флаг: технологический ЦС
isValid (boolean) [опционально],	Флаг: сертификат действует
isExpired (boolean) [опционально],	Флаг: сертификат истек
validFromFrom (instant) [опционально],	Дата начала действия (начало)
validFromTo (instant) [опционально],	Дата начала действия (окончание)
validToFrom (instant) [опционально],	Дата окончания действия (начало)
validToTo (instant) [опционально],	Дата окончания действия (окончание)
updatedFrom (instant) [опционально],	Дата обновления сертификата (начало)
updatedTo (instant) [опционально],	Дата обновления сертификата (конец)
subjectId (UUID[]) [опционально],	ID субъекта
userId (UUID[]) [опционально],	ID учетной записи
serialnumber (string[]) [опционально],	Серийный номер
fingerprint (string[]) [опционально],	Отпечаток
subjectKeyIdentifier (string[]) [опционально],	Идентификатор ключа субъекта
notId (UUID[]) [опционально],	Исключая ID сертификата
endEntityType(string[]) (enum[]: ROOT_CA, SUB_CA, USER, DEVICE, UNKNOWN) [опционально],	Фильтр: Тип субъекта
sortDirection (string) [опционально],	Направления сортировки (ASC;DESC)
sortBy (string[]) [опционально],	Список полей, к которым применяется сортировка
pageOffset (integer) [опционально],	Смещение от начала списка (пагинация)
pageLimit (integer) [опционально]	Ограничение на размер выборки (пагинация)
}	

Request -	
Response ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	Идентификатор сертификата
issuerId (UUID),	Идентификатор издателя сертификата
issuerFingerprint (string),	Фингерпринт издателя сертификата
serialnumber (string),	Серийный номер сертификата
fingerprint (string),	Фингерпринт 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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key - один из перечисленных в enum параметров, а value - значение параметра
},	
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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
},	
name (string),	Имя сертификата (на основе CN)
templateId (UUID),	Идентификатор шаблона
templateName (string),	Имя шаблона
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 ЦС
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
actions: {	Доступные действия по выгрузке
pl2 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
revocation: {	Сведения об отзыве сертификата
date (instant),	Дата отзыва
number (int32),	Код причины отзыва
value (string)	Значение причины отзыва
},	
hashAlgorithm (enum: SHA1, SHA256, SHA384, SHA512, SHA3_256, SHA3_384, SHA3_512, RSASSA_PSS, MD5, GOST_R_34_11_2012, UNKNOWN),	Алгоритм подписи сертификата
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата
keyBits (int4),	Длина ключа сертификата
subjectKeyIdentifier (string),	Идентификатор ключа сертификата
authorityKeyIdentifier (string),	Идентификатор ключа издателя сертификата
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

4.4.6 Метод получения сертификата по идентификатору

GET API – Получение сертификата по идентификатору	
Метод доступен администратору и оператору при наличии полномочий	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20сертификаты/findById_14	
URL – certificate-authority-service/api/v4/public/certificates/{id}	
Query	
{	
id (UUID)	ID сертификата
}	
Request	
-	
Response	Ответ JSON в HTTP-body
ResponseEntity -> CollectionResponse -> {	
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип сертификата
endEntityType (enum: ROOT_CA, SUB_CA, USER, DEVICE, UNKNOWN),	Тип субъекта
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
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
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
id (uuid),	Идентификатор элемента
code (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPTMENT, DATA_ENCRYPTMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY, UNKNOWN),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (uuid),	Идентификатор назначения
value (string),	Наименование элемента
oid (string),	OID назначения
description (string),	Описание использования ключа
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
default (boolean)	Флаг: расширенное использование по умолчанию
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int4),	Код причины отзыва
value (string)	Значение причины отзыва
},	
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
userId (uuid),	Идентификатор субъекта
subjectId (uuid),	Идентификатор субъекта
created (instant)	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
}	

4.4.7 Метод получения сертификата по серийному номеру

GET API – Получение сертификата по его серийному номеру	
Метод доступен администратору и оператору при наличии полномочий	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20сертификаты/getBySerialNumber	
URL – certificate-authority-service/api/v4/public/certificates/serialNumber/{serialNumber}	
Query	
{	

serialnumber (string)	Серийный номер сертификата (формат: 40 символов, нижний регистр)
}	
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key - один из перечисленных в enum параметров, а value - значение параметра
},	
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип сертификата
endEntityType (enum: ROOT_CA, SUB_CA, USER, DEVICE, UNKNOWN),	Тип субъекта
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
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
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
id (uuid),	Идентификатор элемента
code (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPTMENT, DATA_ENCRYPTMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY, UNKNOWN),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (uuid),	Идентификатор элемента
value (string),	Наименование элемента
oid (string),	OID назначения
description, (string)	Описание использования ключа
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
default (boolean)	Флаг: расширенное использование по умолчанию
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int4),	Код причины отзыва
value (string)	Значение причины отзыва
},	
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
userId (uuid),	Идентификатор субъекта
subjectId (uuid),	Идентификатор субъекта
created (instant)	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
}	

4.4.8 Метод получения сертификата по его отпечатку

GET API – Получение сертификата по его отпечатку	
Метод доступен администратору и оператору при наличии полномочий	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20сертификаты/getByFingerprint_1	
URL – certificate-authority-service/api/v4/public/certificates/fingerprint/{fingerprint}	
Query	
{	
fingerprint (String)	Отпечаток сертификата
}	
Request	-
Response	Ответ JSON в HTTP-body
ResponseEntity -> CollectionResponse -> {	
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,	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum

POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key - один из перечисленных в enum параметров, а value - значение параметра
},	
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип сертификата
endEntityType (enum: ROOT_CA, SUB_CA, USER, DEVICE, UNKNOWN),	Тип субъекта
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
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
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
id (uuid),	Идентификатор элемента
code (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPTMENT, DATA_ENCRYPTMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY, UNKNOWN),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа

},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (uuid),	Идентификатор элемента
value (string),	Наименование элемента
oid (string),	OID назначения
description, (string)	Описание использования ключа
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
default (boolean)	Флаг: расширенное использование по умолчанию
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int4),	Код причины отзыва
value (string)	Значение причины отзыва
},	
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
userId (uuid),	Идентификатор субъекта
subjectId (uuid),	Идентификатор субъекта
created (instant)	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
}	

4.4.9 Метод расшифровки контейнера сертификата

POST API – Расшифровка контейнера сертификата	
Метод доступен администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20контейнеры%20сертификатов/parse	
URL – certificate-authority-service/api/v4/public/parse/pkcs12	
Query	
Request	
{	
container: {	Файл контейнера
contentType (string) [опционально],	Тип загружаемого файла (HTTP MediaType) - application/octet-stream)
fileName (string) [опционально],	Имя загружаемого файла
data (string:binary)	Содержимое загружаемого файла (массив байт в Base64)
},	
password(string),	Пароль от контейнера
templateId (UUID)	Идентификатор шаблона
}	
Response	Ответ JSON в HTTP-body
{	
serialnumber (string),	Серийный номер сертификата
fingerprint (string),	Фингерпринт сертификата
name (string),	Имя сертификата (на основе CN)
issuerDN: {	Имя субъекта издателя 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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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, KRBSPRINCIPAL, PERMANENT_IDENTIFIER,	Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key –

XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[]	один из перечисленных в enum параметров, а value – значение параметра
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant),	Дата окончания действия сертификата (ISO 8601)
privateKey: {	Файл закрытого ключа
contentType (string),	Тип загружаемого файла (HTTP MediaType)
fileName (string),	Имя загружаемого файла
data (string:binary)	Содержимое загружаемого файла (массив байт в Base64)
},	
certificate: {	Файл сертификата
contentType (string),	Тип загружаемого файла (HTTP MediaType)
fileName (string),	Имя загружаемого файла
data (string:binary)	Содержимое загружаемого файла (массив байт в Base64)
},	
keyUsages: {	Назначение ключа сертификата
id (uuid),	Идентификатор элемента
code (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 (uuid),	Идентификатор назначения
value (string),	Наименование элемента
oid (string),	OID назначения
description (string),	Описание использования ключа
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
default (boolean)	Флаг: расширенное использование по умолчанию
},	Описание OID
ca (boolean)	Флаг: сертификат ЦС
}	

4.4.10 Метод расшифровки сертификата

POST API – Расшифровка сертификата	
Метод доступен администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20контейнеры%20сертификатов/parse_2	
URL – certificate-authority-service/api/v4/public/parse/pem	
Query	
Request	
{	
request: {	Файл сертификата
contentType (string) [опционально],	Тип загружаемого файла (HTTP MediaType) – application/octet-stream)
fileName (string) [опционально],	Имя загружаемого файла
data (string:binary)	Содержимое загружаемого файла (массив байт в Base64)
}	
Response	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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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, XMPF_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
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип сертификата
endEntityType (enum: ROOT_CA, SUB_CA, USER, DEVICE, UNKNOWN),	Тип субъекта
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
isValid (boolean),	Флаг: сертификат действует
isExpired (boolean),	Флаг: сертификат истек
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
id (uuid),	Идентификатор элемента
code (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPTMENT, DATA_ENCRYPTMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY, UNKNOWN),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (uuid),	Идентификатор назначения
value (string),	Наименование элемента
oid (string),	OID назначения
description (string),	Описание использования ключа
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
default (boolean)	Флаг: расширенное использование по умолчанию

},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
revocation: {	Сведения об отзыве сертификата ЦС
date (instant),	Дата отзыва
number (int4),	Код причины отзыва
value (string)	Значение причины отзыва
},	
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
userId (uuid),	Идентификатор субъекта
subjectId (uuid),	Идентификатор субъекта
created (instant)	Время создания (ISO 8601)
updated (instant),	Время обновления (ISO 8601)
}	

4.4.11 Метод расшифровки запроса на сертификат

POST API – Расшифровка запроса на сертификат	
Метод доступен администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv4%5D%20Контроллер%3A%20контейнеры%20сертификатов/parse_1	
URL – certificate-authority-service/api/v4/public/parse/pkcs10	
Query	
Request	
{	
request: {	Файл сертификата
contentType (string) [опционально],	Тип загружаемого файла (HTTP MediaType) – application/octet-stream)
fileName (string) [опционально],	Имя загружаемого файла
data (string:binary)	Содержимое загружаемого файла (массив байт в Base64). Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE---").
}	
subjectName: {	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.
(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN):	
string[]	
}, [опционально]	
subjectAltName: {	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. Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов 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):	
string[]	
} [опционально]	
}	
Response	
{	Ответ JSON в HTTP-body
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,	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра

DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	
},	
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
keyBits (int32),	Длина ключа сертификата ЦС
subjectKeyIdentifier (string)	Идентификатор ключа сертификата
}	

4.4.12 Метод выпуска короткоживущего (short-lived, throwaway) сертификата в контейнере PKCS#12

POST API – Выпуск короткоживущего (short-lived, throwaway) сертификата в контейнере PKCS#12	
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на использование шаблона, идентификатор которого передается во входных параметрах. <p>Выпуск короткоживущего (short-lived, throwaway) сертификата с помощью данного метода будет выполняться успешно только при соблюдении следующих условий:</p> <ul style="list-style-type: none"> – выпуск данного вида сертификатов разрешен в соответствии с параметрами лицензии, примененной в ПО еСА-СА. В противном случае будет возвращено сообщение об ошибке с кодом 403 и текстом «Выпуск короткоживущих (short-lived, throwaway) сертификатов недоступен в соответствии с параметрами лицензии»; – для выпуска используется шаблон с включенной опцией «Короткоживущий (short-lived, throwaway) сертификат». В противном случае будет возвращено сообщение об ошибке с кодом 400 и текстом «Выпуск короткоживущих (short-lived, throwaway) сертификатов недоступен по данному шаблону»; – для выпуска используется шаблон с включенной опцией «Выпуск сертификатов с закрытым ключом (PKCS#12)», иначе метод вернет сообщение об ошибке с кодом 400 и текстом «Выпуск сертификатов с закрытым ключом (PKCS#12) недоступен по данному шаблону». – указываемый во входных параметрах пароль от контейнера соответствует требованиям регулярного выражения по шаблону, иначе метод вернет сообщение об ошибке с кодом 400 и текстом «Пароль не соответствует регулярному выражению, указанному в шаблоне». <p>Успешно выпущенный с использованием данного метода сертификат:</p> <ul style="list-style-type: none"> – не будет сохранен в базе данных ПО еСА-СА; – не будет содержать записей о точках распространения CRL, Delta CRL и службах OCSP, однако может содержать записи о точках AIA; – не будет иметь владельца-субъекта, и, как следствие, не повлияет на лицензируемое количество субъектов с действующими сертификатами. 	
URL – certificate-authority-service/api/v4/public/certificates/enroll/{caId}/throwaway	
Query	
{	
caId (UUID)	ID ЦС
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона
subjectDN: {	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
(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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]	
},	

<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
keyBits (integer),	Длина ключа
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключевой пары сертификата
password (string)	Пароль контейнера
}	
Response	
Файл контейнера PKCS#12 (byte)	

4.4.13 Методы выпуска короткоживущего (short-lived, throwaway) сертификата на основании запроса PKCS#10

4.4.13.1 Выпуск короткоживущего (short-lived, throwaway) сертификата на основании запроса PKCS#10 (формат запроса на сертификат из входных параметров – multipart/form-data)

POST API – Выпуск короткоживущего (short-lived, throwaway) сертификата на основании запроса PKCS#10	
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на использование шаблона, идентификатор которого передается во входных параметрах. <p>Выпуск короткоживущего (short-lived, throwaway) сертификата с помощью данного метода будет выполняться успешно только при соблюдении следующих условий:</p> <ul style="list-style-type: none"> – выпуск данного вида сертификатов разрешен в соответствии с параметрами лицензии, примененной в ПО еСА-СА. В противном случае будет возвращено сообщение об ошибке с кодом 403 и текстом «Выпуск короткоживущих (short-lived, throwaway) сертификатов недоступен в соответствии с параметрами лицензии»; – для выпуска используется шаблон с включенной опцией «Короткоживущий (short-lived, throwaway) сертификат». В противном случае будет возвращено сообщение об ошибке с кодом 400 и текстом «Выпуск короткоживущих (short-lived, throwaway) сертификатов недоступен по данному шаблону». <p>Успешно выпущенный с использованием данного метода сертификат:</p> <ul style="list-style-type: none"> – не будет сохранен в базе данных ПО еСА-СА; – не будет содержать записей о точках распространения CRL, Delta CRL и службах OCSP, однако может содержать записи о точках AIA; – не будет иметь владельца-субъекта, и, как следствие, не повлияет на лицензируемое количество субъектов с действующими сертификатами. 	
URL – certificate-authority-service/api/v4/public/certificates/enroll/{caId}/pkcs10/throwaway	
Query	
{	
caId (UUID)	ID ЦС
}	
Request	
{	
templateId (UUID),	Идентификатор шаблона
request (binary),	<p>Файл запроса на сертификат.</p> <p>Допустимые форматы запроса на сертификат:</p> <ul style="list-style-type: none"> • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE-----").

<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, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] }, [опционально] </pre>	<p>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.</p> <p>Значения полей, указанные в subjectName, переопределяют значения соответствующих полей SDN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.</p>
<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[] } [опционально] </pre>	<p>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.</p> <p>Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.</p>
<p>Response</p> <p>Файл цепочки сертификатов выпущенного сертификата (byte)</p>	

4.4.13.2 Выпуск короткоживущего (short-lived, throwaway) сертификата на основании запроса PKCS#10 (формат запроса на сертификат из входных параметров – application/json)

Примечание: отличается от метода выше форматом передаваемого во входных параметрах запроса на сертификат. В данном методе на вход поступает содержимое файла запроса на сертификат в формате PEM (содержимое в Base64).

<p>POST API – Выпуск короткоживущего (short-lived, throwaway) сертификата на основании запроса PKCS#10</p>	
<p>Метод доступен:</p> <ul style="list-style-type: none"> – администратору; – оператору при наличии полномочий на использование шаблона, идентификатор которого передается во входных параметрах. <p>Выпуск короткоживущего (short-lived, throwaway) сертификата с помощью данного метода будет выполняться успешно только при соблюдении следующих условий:</p> <ul style="list-style-type: none"> – выпуск данного вида сертификатов разрешен в соответствии с параметрами лицензии, примененной в ПО eCA-CA. В противном случае будет возвращено сообщение об ошибке с кодом 403 и текстом «Выпуск короткоживущих (short-lived, throwaway) сертификатов недоступен в соответствии с параметрами лицензии»; – для выпуска используется шаблон с включенной опцией «Короткоживущий (short-lived, throwaway) сертификат». В противном случае будет возвращено сообщение об ошибке с кодом 400 и текстом «Выпуск короткоживущих (short-lived, throwaway) сертификатов недоступен по данному шаблону». <p>Успешно выпущенный с использованием данного метода сертификат:</p> <ul style="list-style-type: none"> – не будет сохранен в базе данных ПО eCA-CA; – не будет содержать записей о точках распространения CRL, Delta CRL и службах OCSP, однако может содержать записи о точках AIA; – не будет иметь владельца-субъекта, и, как следствие, не повлияет на лицензируемое количество субъектов с действующими сертификатами. 	
<p>URL – certificate-authority-service/api/v4/public/certificates/enroll/{caId}/pkcs10/throwaway</p>	
<p>Query</p> <pre>{ caId (UUID) }</pre>	<p>ID ЦС</p>
<p>Request</p> <pre>{ templateId (UUID),</pre>	
	<p>Идентификатор шаблона</p>

<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). Допустимые форматы запроса на сертификат: • PEM; • PEM без хидера и футера ("-----BEGIN CERTIFICATE-----" и "-----END CERTIFICATE-----").
<code>},</code>	
<code> subjectName: {</code> <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, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[]</code> <code> }, [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: {</code> <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): string[]</code> <code> } [опционально]</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. Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.
<code>}</code>	
Response Файл цепочки сертификатов выпущенного сертификата (byte)	

4.5 Методы работы с точками подключения

4.5.1 Метод поиска точек подключения

GET API – Поиск точек подключения к PC	
Метод доступен администратору и оператору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v4	
Контроллер%3A Точки подключения ресурсных систем/findAll_12	
URL – ldap-service/api/v4/public/connection-points	
Query	
<code>{</code>	
<code> id (uuid) [опционально],</code>	Фильтр: ID точки подключения
<code> resourceId (uuid) [опционально],</code>	Фильтр: ID ресурсной системы
<code> search (string),</code>	Фильтр: полнотекстовый поиск по отображаемому имени
<code> sortDirection (string) [опционально],</code>	Направления сортировки (ASC;DESC)
<code> sortBy (string[]) [опционально],</code>	Список полей, к которым применяется сортировка
<code> pageOffset (integer) [опционально],</code>	Смещение от начала списка (пагинация)
<code> pageLimit (integer) [опционально],</code>	Ограничение на размер выборки (пагинация)
<code> inQueue (boolean) [опционально]</code>	Фильтр: точка подключения в очереди
<code>}</code>	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
<code> id (UUID),</code>	ID точки подключения
<code> title (string),</code>	Имя точки подключения
<code> domainType(enum: SAMBA_DC, MS_AD, RED_ADM, FREE_IPA, ALD_PRO, ALT_DOMAIN, ROSA_DD, UNKNOWN),</code>	Тип точки подключения
<code> connectionAddress (string),</code>	Адрес (хост) подключения
<code> useTls (boolean),</code>	Флаг: использовать TLS при подключении
<code> baseDn(string),</code>	BaseDN точки подключения
<code> username (string),</code>	Имя пользователя ресурсной системы
<code> status (string)</code>	Статус точки подключения

resourceId (UUID),	ID ресурсной системы
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
inQueue (boolean)	Флаг: точка подключения в очереди
}	

4.5.2 Метод получения точки подключения по идентификатору

GET API – Получение точки подключения по идентификатору	
Метод доступен администратору и оператору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v4/Контроллер%3А Точки подключения ресурсных систем/findById_11	
URL – Idap-service/api/v4/public/connection-points/{id}	
Query	
{	
id (uuid)	ID точки подключения
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	
id (UUID),	ID точки подключения
title (string),	Имя точки подключения
domainType(enum: SAMBA_DC, MS_AD, RED_ADM, FREE_IPA, ALD_PRO, ALT_DOMAIN, ROSA_DD, UNKNOWN),	Тип точки подключения
connectionAddress (string),	Адрес (хост) подключения
useTls (boolean),	Флаг: использовать TLS при подключении
baseDn(string),	BaseDN точки подключения
username (string),	Имя пользователя ресурсной системы
status (string)	Статус точки подключения
resourceId (UUID),	ID ресурсной системы
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
inQueue (boolean)	Флаг: точка подключения в очереди
}	

4.5.3 Метод частичной синхронизации точки подключения

PUT API – Частичная синхронизация точки подключения	
Метод доступен администратору и оператору при наличии полномочий. У оператора имеется возможность выполнять синхронизацию с теми внешними ресурсными системами, к субъектам которых у него есть доступ.	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v5Bv4%5D%20Контроллер%3А%20Точки%20подключения%20ресурсных%20систем/synchronize	
URL – Idap-service/api/v4/public/connection-points/{pointId}/synchronize	
Query	
{	
pointId (UUID)	ID точки подключения
}	
Request	
-	
Response	
-	

4.6 Методы работы с Syslog-серверами

4.6.1 Метод поиска Syslog-серверов

GET API – Поиск Syslog-серверов	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/v5Bv4%5D%20Контроллер%3А%20Syslog%20сервера/findAll_3	
URL – logs-service/api/v4/public/syslog	
Query	
-	
Request	

-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID Syslog-сервера
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP, TCP_TLS, UNKNOWN),	Протокол Syslog-сервера
active (boolean)	Флаг: состояние настройки публикации событий
}	

4.6.2 Метод получения Syslog-сервера по идентификатору

GET API – Получение Syslog-сервера по идентификатору	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv4%5D%20Контроллер%3A%20Syslog%20сервера/findById	
URL – logs-service/api/v4/public/syslog/{id}	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
-	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID Syslog-сервера
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP, TCP_TLS, UNKNOWN),	Протокол Syslog-сервера
active (boolean)	Флаг: состояние настройки публикации событий
}	

4.6.3 Метод создания Syslog-сервера

POST API – Создание Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv4%5D%20Контроллер%3A%20Syslog%20сервера/create_1	
URL – logs-service/api/v4/public/syslog	
Query	
-	
Request	
{	
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP, TCP_TLS, UNKNOWN)	Протокол Syslog-сервера
}	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID Syslog-сервера
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP, TCP_TLS, UNKNOWN),	Протокол Syslog-сервера
active (boolean)	Флаг: состояние настройки публикации событий
}	

4.6.4 Метод обновления Syslog-сервера

PUT API – Обновление Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv4%5D%20Контроллер%3A%20Syslog%20сервера/updateById	
URL – logs-service/api/v4/public/syslog/{id}	
Query	
{	
id (uuid)	ID Syslog-сервера
}	

Request	
{	
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP, TCP_TLS, UNKNOWN)	Протокол Syslog-сервера
}	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID Syslog-сервера
host (string),	Имя хоста Syslog-сервера
port (int32),	Порт Syslog-сервера
protocol (enum: UDP, TCP, TCP_TLS, UNKNOWN),	Протокол Syslog-сервера
active (boolean)	Флаг: состояние настройки публикации событий
}	

4.6.5 Метод деактивации Syslog-сервера

PATCH API – Деактивация Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv4%5D%20Контроллер%3A%20Syslog%20сервера/deactivate	
URL – logs-service/api/v4/public/syslog/{id}/deactivate	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
-	
Response	
-	

4.6.6 Метод активации Syslog-сервера

PATCH API – Активация Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv4%5D%20Контроллер%3A%20Syslog%20сервера/activate	
URL – logs-service/api/v4/public/syslog/{id}/activate	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
-	
Response	
-	

4.6.7 Метод удаления Syslog-сервера

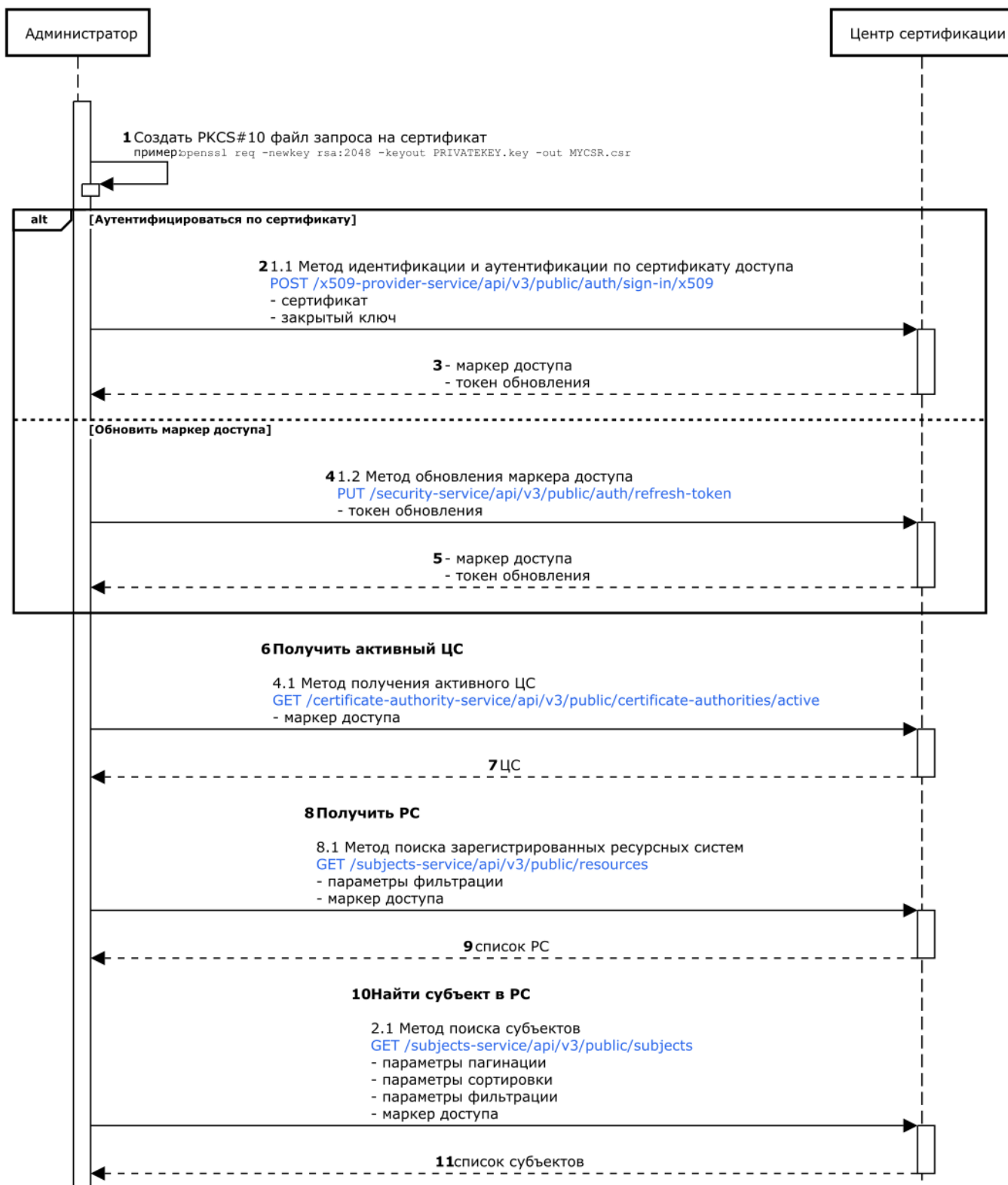
DELETE API – Удаление Syslog-сервера	
Метод доступен только администратору	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv4%5D%20Контроллер%3A%20Syslog%20сервера/deleteById	
URL – logs-service/api/v4/public/syslog/{id}	
Query	
{	
id (uuid)	ID Syslog-сервера
}	
Request	
-	
Response	
-	

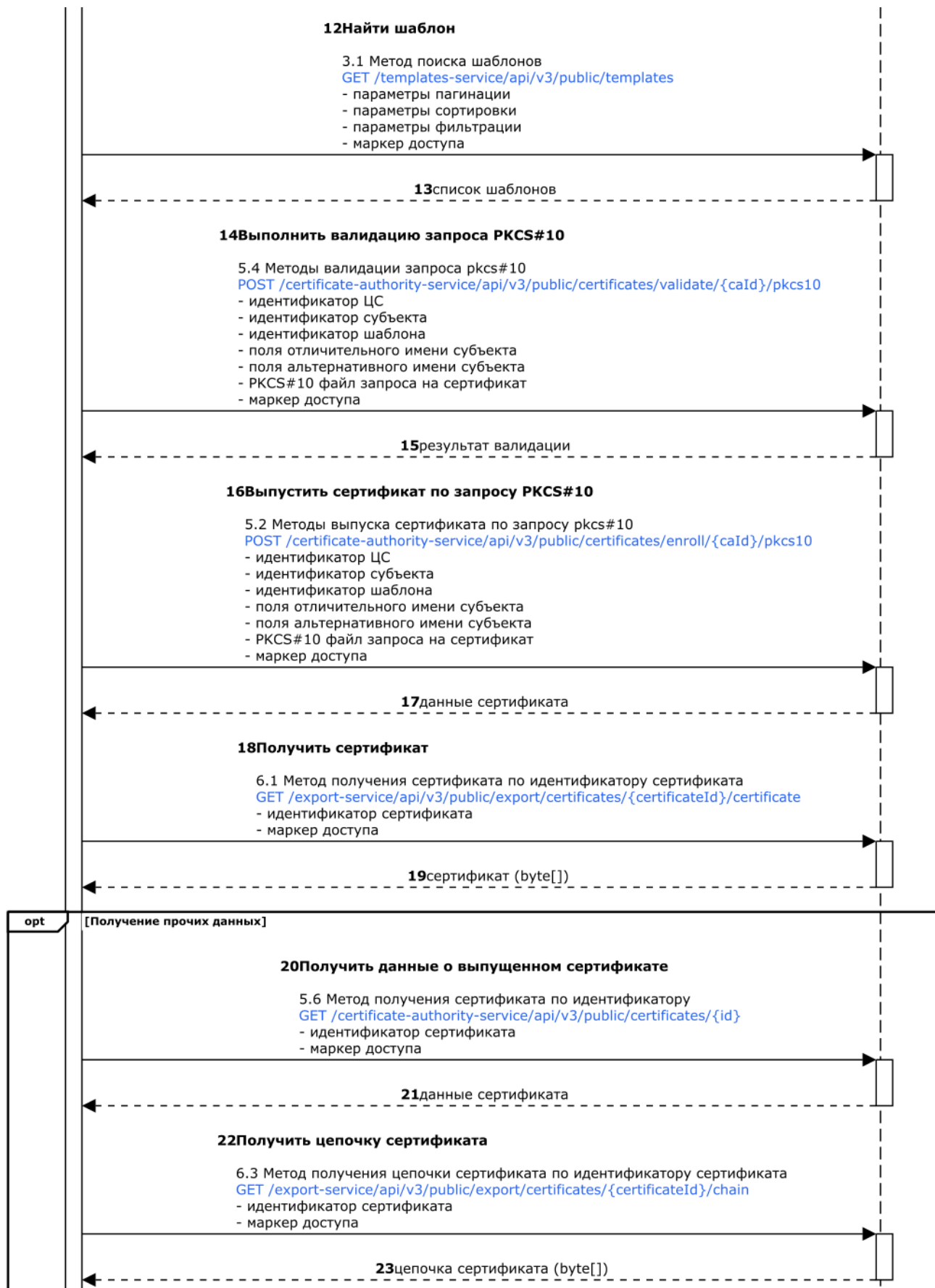
5 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

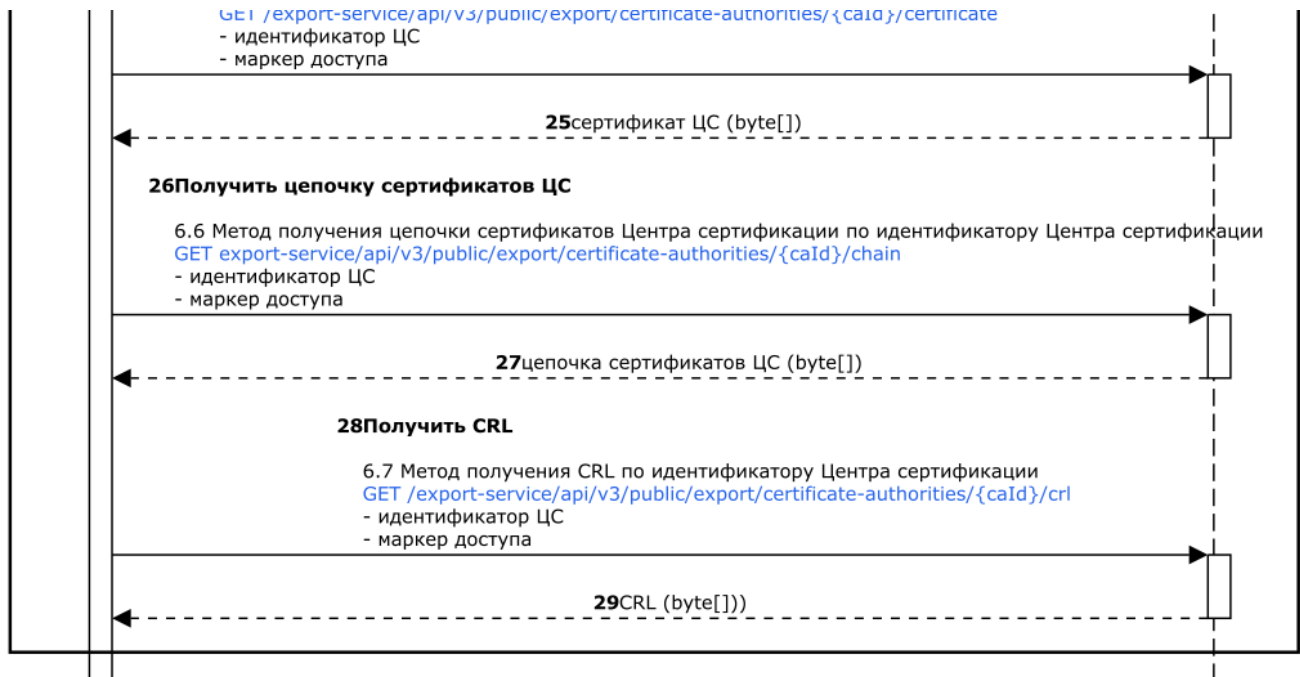
Выполнение пользователем с ролью «Администратор» методов от имени другой учётной записи

Все методы REST API версии 3 за исключением метода идентификации и аутентификации по сертификату доступа (см. 2.1.1) позволяют пользователям с ролью «Администратор» использовать их от имени любой учётной записи eCA-CA. Для этого необходимо передавать идентификатор учётной записи в заголовке «X-User-Context» HTTP-запроса при вызове методов.

6 ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ ПОЛУЧЕНИЯ СЕРТИФИКАТА ПО ЗАПРОСУ PKCS#10







Для получения сертификата по запросу PKCS#10 следует выполнить шаги, представленные на диаграмме последовательности выше. Краткое описание представлено ниже:

- предварительно подготовить файл запроса (шаг 1) – `request`;
- аутентифицироваться на ЦС (шаг 2-3);
- при истечении маркера доступа следует произвести его обновление, используя токен обновления (шаги 4-5);
- все последующие запросы используют маркер доступа;
- после этого следует получить активный ЦС (шаги 6-7), далее будет использован его идентификатор – `caId`;
- получить идентификатор ресурсной системы, в которой находит субъект, для которого выпускается сертификат (шаги 8-9) – `resourceId`;
- найти субъект, для которого выпускается сертификат (шаги 10-11), при этом для поиска используется идентификатор ресурсной системы `resourceId` – получить его идентификатор `subjectId`;
- найти шаблон сертификата (шаги 12-13) – получить его идентификатор `templateId`;
- выполнить валидацию запроса (шаги 14-15), используются параметры:
 - идентификатор ЦС – `caId`;
 - идентификатор субъекта – `subjectId`;
 - идентификатор шаблона – `templateId`;
 - файл запроса PKCS#10 – `request`;
 - поля отличного имени субъекта;
 - поля альтернативного имени субъекта.
- при отсутствии ошибок выпустить сертификат по запросу PKCS#10 (шаги 16-17), параметры такие же, как при валидации запроса, в результате будет идентификатор сертификата – `certificateId`;
- после этого скачать выпущенный сертификат (шаги 18-19), используя `certificateId`;
- опционально можно получить информацию о выпущенном сертификате, цепочку сертификатов, используя идентификатор сертификата – `certificateId` (шаги 20-23).

- а также сертификат ЦС, цепочку сертификатов ЦС и CRL, используя идентификатор ЦС – `caId` (шаги 24-29).

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ЦС	–	Центр сертификации
API	–	Authority Information Access (это программный интерфейс приложений, набор инструкций, который позволяет разным приложениям общаться между собой)
CRL	–	Certificate Revocation List (список отзыва сертификатов)
AIA	–	Authority Information Access (это расширение сертификата X.509, которое содержит информацию о доступе к сертификату центра сертификации)
URL	–	Uniform Resource Locator (адрес ресурса в сети Интернет)
CN	–	Common Name (это поле в цифровых сертификатах (SSL/TLS), которое используется для указания доменного имени или имени хоста, для которых предназначен сертификат)
HTTP	–	Hyper Text Transfer Protocol (сетевой протокол прикладного уровня, который изначально предназначался для получения с серверов гипертекстовых документов в формате HTML)
OCSP	–	Online Certificate Status Protocol (интернет-протокол, используемый для получения статуса отзыва цифрового сертификата X.509)
JSON	–	JavaScript Object Notation (текстовый формат обмена данными, основанный на JavaScript)
SID	–	Security Identifier (идентификатор безопасности , структура данных переменной длины, которая идентифицирует учётную запись пользователя, группы, службы, домена или компьютера)
ISO	–	International Organization for Standardization (международная организация, занимающаяся выпуском стандартов)
OID	–	Object identifier (это уникальный идентификатор, который идентифицирует объекты и атрибуты внутри инфраструктуры)

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]