



Центр сертификатов доступа

Aladdin Enterprise Certificate Authority Certified Edition

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

Издание	RU.АЛДЕ.03.01.020
Документ	RU.АЛДЕ.03.01.020 32 01-6
Версия	2.4
Листов	90
Дата	28.05.2026

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 доработки, 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 приложения/взаимосвязанные/имеющие отношение к данному руководству, данные, содержащиеся в нём, а также все права на ПО являются и будут являться собственностью исключительно Компании.

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

Лицензия

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

Соглашения:

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

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

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

Требования к использованию

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

Использование ПО

Пользователь вправе:

- воспроизводить ПО путём записи его в память электронно-вычислительных машин Пользователя, ограниченное правом инсталляции, копирования и запуска программ для ЭВМ;
 - встраивать ПО любым способом в продукты и решения Пользователя;
 - распространять ПО любым способом исключительно в составе продуктов и решений Пользователя.
- При использовании и распространении ПО Пользователь обязан руководствоваться действующим законодательством Российской Федерации и международным законодательством, учитывая ограничения и дополнительные требования, которые могут возникать в связи с экспортом шифровальных (криптографических) средств с территории Российской Федерации и импортом таких средств в другие страны. В частности, ограничения и дополнительные требования могут возникать при распространении ПО через магазины приложений, содержащие различные приложения для мобильных устройств.
- Условия использования, изложенные в настоящем соглашении, действуют в отношении всего содержимого ПО, в частности в отношении:

- дизайна (графики, расположения элементов оформления и т.п.);
 - всех иных элементов, в том числе изображений, фонограмм, текстов.
- Получаемые Пользователем неисключительные имущественные права не включают права на передачу третьим лицам каких-либо прав на встраивание, воспроизведение, распространение и использование программ для ЭВМ не в составе продуктов и решений Пользователя.
- Компания сохраняет за собой все исключительные права на ПО и входящие в него компоненты, включая права на предоставление неисключительных и исключительных прав третьим лицам.

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

Обслуживание и поддержка

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

Ограниченная гарантия

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

Отказ от гарантии

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

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

Гарантия недействительна, если ПО используется в сочетании с иным аппаратным и/или программным обеспечением, отличным от описанных в технической/эксплуатационной документации, или используется на компьютере с любым установленным 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	6
1.1 Методы аутентификации.....	6
1.1.1 Метод аутентификации по сертификату (устаревший)	6
1.1.2 Метод аутентификации по Kerberos-ticket (устаревший)	6
1.1.3 Метод аутентификации по логину и паролю (устаревший).....	7
1.1.4 Метод аутентификации по сертификату.....	7
1.1.5 Метод аутентификации по Kerberos-ticket	8
1.1.6 Метод аутентификации по логину и паролю	8
1.1.7 Метод обновления маркера доступа	9
1.2 Методы работы с заявками.....	10
1.2.1 Метод создания новой заявки на выпуск сертификата с закрытым ключом (PKCS#12).....	10
1.2.2 Метод создания новой заявки на основании запроса PKCS#10.....	12
1.2.3 Метод получения заявки по идентификатору.....	18
1.2.4 Метод получения заявки по внешнему ключу	19
1.2.5 Метод поиска заявок	20
1.2.6 Метод получения данных о выпущенном сертификате заявки по ее идентификатору	22
1.3 Методы экспорта файлов.....	27
1.3.1 Метод получения запроса на сертификат по идентификатору заявки	27
1.3.2 Метод получения сертификата по идентификатору заявки.....	27
1.3.3 Метод получения цепочки сертификатов по идентификатору заявки	28
1.3.4 Метод получения контейнера pkcs12 по идентификатору заявки	29
1.3.5 Метод получения сертификата издателя сертификата по заявке.....	29
1.3.6 Метод получения цепочки сертификатов издателя сертификата по заявке	30
1.3.7 Метод получения CRL издателя сертификата по заявке.....	30
1.4 Методы работы с субъектами.....	32
1.4.1 Метод поиска субъектов.....	32
1.4.2 Метод получения субъекта по идентификатору.....	33
1.5 Методы работы с шаблонами	35
1.5.1 Метод поиска шаблонов.....	35
1.5.2 Метод получения шаблона по идентификатору	36
1.6 Методы работы с сертификатами.....	38
1.6.1 Метод отзыва сертификата по идентификатору заявки.....	38
1.7 Методы получения информации о сервисах	39
1.7.1 Методы получения информации о сервисе безопасности (security-service)	39
1.7.2 Методы получения информации о сервисе заявок (tasks-service).....	41
1.7.3 Методы получения информации о сервисе журнала событий (logs-service).....	43
1.7.4 Методы получения информации о сервисе правил выпуска (policies-service)	45
1.7.5 Методы получения информации о сервисе интеграции с центром сертификации (ca-adapter-service)	47
1.7.6 Методы получения информации о сервисе настроек (settings-service).....	49

1.7.7 Методы получения информации о сервисе промежуточного взаимодействия (middleware-service) ..	50
1.7.8 Методы получения информации о сервисе хранения данных (storage-service)	52
1.7.9 Методы получения информации о сервисе экспорта данных (export-service)	54
1.7.10 Методы получения информации о SCEP-сервисе (scep-service)	56
1.7.11 Методы получения информации о сервисе внешних интеграций (external-integration-service)	58
1.7.12 Методы получения информации о WSTEP-сервисе (wstep-enrollment-service)	60
1.7.13 Описание Prometheus-метрик сервисов	62
1.8 Метод получения версии сервиса внешних интеграций	69
2 Описание методов REST API версии 3	70
2.1 Методы работы с заявками	70
2.1.1 Метод создания новой заявки на выпуск сертификата с закрытым ключом (PKCS#12)	70
2.1.2 Методы создания новой заявки на основании запроса PKCS#10	71
2.1.3 Метод получения данных сертификата, выпущенного по заявке, по идентификатору заявки	75
2.2 Методы работы с субъектами	78
2.2.1 Метод поиска субъектов	78
2.2.2 Метод получения субъекта по идентификатору	80
2.3 Методы работы с шаблонами	82
2.3.1 Метод получения шаблона по идентификатору	82
3 Диаграмма последовательности получения сертификата по запросу PKCS#10	85
Обозначения и сокращения	89

1 ОПИСАНИЕ МЕТОДОВ REST API ВЕРСИИ 2

1.1 Методы аутентификации

1.1.1 Метод аутентификации по сертификату (устаревший)

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

1.1.2 Метод аутентификации по Kerberos-ticket (устаревший)

POST API – Аутентификация по Kerberos-ticket	
В случае аутентификации по реквизитам субъекта домена, к которому подключен eCA-RA, для которого существует связанная незаблокированная учетная запись в eCA-CA, вход в eCA-RA осуществляется под именем данной УЗ с соответствующими пользователю правами.	
URL – /kerberos-provider-service/api/v2/public/auth/sign-in/kerberos	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5Bv2%5D%20Контроллер%3A%20Авторизации/kerberosSignIn_1	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
ResponseEntity -> ItemResponse -> {	
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

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

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

1.1.3 Метод аутентификации по логину и паролю (устаревший)

POST API – Аутентификация по логину и паролю	
В случае аутентификации по реквизитам субъекта домена, к которому подключен eCA-RA, для которого существует связанная незаблокированная учетная запись в eCA-CA, вход в eCA-RA осуществляется под именем данной УЗ с соответствующими пользователю правами.	
URL – /kerberos-provider-service/api/v2/public/auth/sign-in/ldap	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%3A%20Авторизации/ldapSignIn_1	
Query -	
Request {	
username (string),	Имя пользователя
password (string)	Пароль пользователя
}	
Response ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

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

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

1.1.4 Метод аутентификации по сертификату

POST API – Аутентификация с помощью сертификата	
Данный метод позволяет выполнять аутентификацию в eCA-RA по сертификату доступа администратора или оператора подключенного eCA-CA.	
URL – /security-service/api/v2/public/auth/sign-in/x509	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Авторизации/signInByX509_1	
Query -	

Request	
-	
Response	Ответ JSON в HTTP-body
ResponseEntity -> ItemResponse -> {	
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

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

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

1.1.5 Метод аутентификации по Kerberos-ticket

POST API – Аутентификация по Kerberos-ticket	
В случае аутентификации по реквизитам субъекта домена, к которому подключен eCA-RA, для которого существует связанная незаблокированная учетная запись в eCA-CA, вход в eCA-RA осуществляется под именем данной УЗ с соответствующими пользователю правами.	
URL – /security-service/api/v2/public/auth/sign-in/kerberos	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Авторизации/kerberosSignIn	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
ResponseEntity -> ItemResponse -> {	
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

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

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

1.1.6 Метод аутентификации по логину и паролю

POST API – Аутентификация по логину и паролю
--

В случае аутентификации по реквизитам субъекта домена, к которому подключен еCA-RA, для которого существует связанная незаблокированная учетная запись в еCA-CA, вход в еCA-RA осуществляется под именем данной УЗ с соответствующими пользователю правами.	
URL – /security-service/api/v2/public/auth/sign-in/ldap	
Swagger:https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Авторизации/ldapSignIn	
Query	
-	
Request	
{	
username (string),	Имя пользователя
password (string)	Пароль пользователя
}	
Response	Ответ JSON в HTTP-body
ResponseEntity -> ItemResponse -> {	
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

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

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

1.1.7 Метод обновления маркера доступа

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

1.2 Методы работы с заявками

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

POST API – Создание новой заявки на выпуск сертификата в контейнере PKCS#12	
<p>Для УЗ с ролью «Пользователь» с использованием данного метода доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью.</p> <p>Для УЗ с ролью «Администратор» или «Оператор» с использованием данного метода доступно создание заявок для субъектов ресурсных систем из еCA-CA, к которому подключен еCA-RA.</p> <p>Если в рамках создания заявки осуществляется вызов метода создания субъекта¹ публичного API еCA-CA, в заголовке «X-User-Context» будет передан идентификатор УЗ оператора, от имени которого осуществляется попытка создания заявки. В случае, если у оператора нет полномочий на создание нового локального субъекта, в ответе метода создания заявки будет возвращаться ошибка с кодом 403 и текстом «Создание заявки недоступно. Отсутствуют полномочия на создание новых субъектов».</p>	
URL – /tasks-service/api/v2/public/tasks/pkcs12	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/registerPkcs12	
Query	
{	
certificateAuthorityId (uuid), [опционально]	<p>ID Центра сертификации, на котором должен быть выпущен сертификат по заявке.</p> <p>Если для данного параметра не указано значение, выпуск будет осуществляться на Центре сертификации подключенного еCA-CA, который определен в используемом шаблоне (поле «templateId»).</p> <p>Если в используемом шаблоне не определен конкретный центр сертификации (указано значение «Любой»), выпуск будет осуществляться на активном на момент создания заявки Центре сертификации подключенного еCA-CA.</p>
subjectId (uuid), [опционально]	<p>ID субъекта².</p> <p>Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь».</p>
templateId (uuid), [обязательно]	ID шаблона сертификата ³ .
foreignKey (string) [опционально]	<p>Внешний ключ заявки.</p> <p>Внешний ключ должен быть уникальным.⁴</p>
}	
Request	
{	

¹ В случае, если во входных параметрах метода создания заявки не указан ID субъекта и не найден субъект по CN, указанному во входных параметрах метода.

² Субъекты еCA-CA и еCA-RA имеют сквозную идентификацию.

³ Шаблоны еCA-CA и еCA-RA имеют сквозную идентификацию.

⁴ При указании в поле «foreignKey» значения, аналогичного значению в поле «foreignKey» у уже существующей в базе данных программы заявки, при использовании метода будет возвращена ошибка с кодом 400.

<pre>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, которое должно попасть в сертификат. 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, 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 ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ¹
type (enum: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (enum: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ²
certificateId(uuid),	Идентификатор сертификата. Заполняется по итогам успешной отработки заявки.
name (string),	Common Name сертификата

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

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

updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string)	Внешний ключ заявки
}	

1.2.2 Метод создания новой заявки на основании запроса PKCS#10

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

POST API – Создание новой заявки на основании запроса PKCS#10	
<p>Для УЗ с ролью «Пользователь» с использованием данного метода доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью.</p> <p>Для УЗ с ролью «Администратор» или «Оператор» с использованием данного метода доступно создание заявок для субъектов ресурсных систем из еCA-CA, к которому подключен еCA-RA.</p> <p>Если в рамках создания заявки осуществляется вызов метода создания субъекта¹ публичного API еCA-CA, в заголовке «X-User-Context» будет передан идентификатор УЗ оператора, от имени которого осуществляется попытка создания заявки. В случае, если у оператора нет полномочий на создание нового локального субъекта, в ответе метода создания заявки будет возвращаться ошибка с кодом 403 и текстом «Создание заявки недоступно. Отсутствуют полномочия на создание новых субъектов».</p>	
URL – /tasks-service/api/v2/public/tasks/pkcs10	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/registerPkcs10AsFileDescription	
Query	
{	
certificateAuthorityId (uuid), [опционально]	<p>ID Центра сертификации, на котором должен быть выпущен сертификат по заявке.</p> <p>Если для данного параметра не указано значение, выпуск будет осуществляться на Центре сертификации подключенного еCA-CA, который определен в используемом шаблоне (поле «templateId»).</p> <p>Если в используемом шаблоне не определен конкретный центр сертификации (указано значение «Любой»), выпуск будет осуществляться на активном на момент создания заявки Центре сертификации подключенного еCA-CA.</p>

¹ В случае, если во входных параметрах метода создания заявки не указан ID субъекта и не найден субъект по CN, указанному во входных параметрах метода.

subjectId (uuid), [опционально]	ID субъекта ¹ . Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь». Может быть указан при использовании метода учетной записью с ролью «Администратор». ²
templateId (uuid), [обязательно]	ID шаблона сертификата ³ .
foreignKey (string) [опционально]	Внешний ключ заявки. Внешний ключ должен быть уникальным. ⁴
}	
Request	
{	
request (MultipartFile),	PKCS#10 файл запроса на сертификат
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 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.

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

² Если при создании администратором заявки на сертификат на основании запроса во входных параметрах метода не указан subjectId, ПО eCA-RA должно на основании указанного в запросе значения Common name автоматически определять (или создавать) получателя сертификата по заявке в соответствии со следующими правилами:

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

³ Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

⁴ При указании в поле «foreignKey» значения, аналогичного значению в поле «foreignKey» у уже существующей в базе данных программы заявки, при использовании метода будет возвращена ошибка с кодом 400.

<pre> 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.</p> <p>Необязательный параметр, задается в случае невозможности поместить данные в PKCS10.</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>ResponseEntity -> ItemResponse -> {</p>	<p>Ответ JSON в HTTP-body</p>
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ¹
type (enum: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (enum: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ²
certificateId (string),	Идентификатор сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string)	Внешний ключ заявки

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

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

}

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

POST API – Создание новой заявки на основании запроса PKCS#10	
<p>Для УЗ с ролью «Пользователь» с использованием данного метода доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью.</p> <p>Для УЗ с ролью «Администратор» или «Оператор» с использованием данного метода доступно создание заявок для субъектов ресурсных систем из еCA-CA, к которому подключен еCA-RA.</p> <p>Если в рамках создания заявки осуществляется вызов метода создания субъекта¹ публичного API еCA-CA, в заголовке «X-User-Context» будет передан идентификатор УЗ оператора, от имени которого осуществляется попытка создания заявки. В случае, если у оператора нет полномочий на создание нового локального субъекта, в ответе метода создания заявки будет возвращаться ошибка с кодом 403 и текстом «Создание заявки недоступно. Отсутствуют полномочия на создание новых субъектов».</p>	
URL – /tasks-service/api/v2/public/tasks/pkcs10	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/registerPkcs10AsFileDescription	
Query	
{	
certificateAuthorityId (uuid), [опционально]	<p>ID Центра сертификации, на котором должен быть выпущен сертификат по заявке.</p> <p>Если для данного параметра не указано значение, выпуск будет осуществляться на Центре сертификации подключенного еCA-CA, который определен в используемом шаблоне (поле «templateId»).</p> <p>Если в используемом шаблоне не определен конкретный центр сертификации (указано значение «Любой»), выпуск будет осуществляться на активном на момент создания заявки Центре сертификации подключенного еCA-CA.</p>
subjectId (uuid), [опционально]	<p>ID субъекта².</p> <p>Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь».</p>

¹ В случае, если во входных параметрах метода создания заявки не указан ID субъекта и не найден субъект по CN, указанному во входных параметрах метода.

² Субъекты еCA-CA и еCA-RA имеют сквозную идентификацию.

	Может быть указан при использовании метода учетной записью с ролью «Администратор». ¹
templateId (uuid), [обязательно]	ID шаблона сертификата ² .
foreignKey (string) [опционально]	Внешний ключ заявки. Внешний ключ должен быть уникальным. ³
}	
Request	
{	
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)
},	
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 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

¹ Если при создании администратором заявки на сертификат на основании запроса во входных параметрах метода не указан subjectId, ПО eCA-RA должно на основании указанного в запросе значения Common name автоматически определять (или создавать) получателя сертификата по заявке в соответствии со следующими правилами:

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

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

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

	соответствуют значениям атрибутов субъекта.
<pre> subjectAltName: { (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[] } </pre>	<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>Необязательный параметр, задается в случае невозможности поместить данные в PKCS10.</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>ResponseEntity -> ItemResponse -> {</p>	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ¹
type (enum: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (enum: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ²
certificateId (string),	Идентификатор сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)

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

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

created (instant),	Время создания заявки (ISO 8601)
foreignKey (string)	Внешний ключ заявки
}	

1.2.3 Метод получения заявки по идентификатору

GET API – Получение заявки по идентификатору	
<p>Для УЗ с ролью «Пользователь» с использованием данного метода доступно получение только тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ.</p> <p>Для УЗ с ролью «Оператор» с использованием данного метода доступно получение заявок, созданных данной УЗ, а также заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного еСА-СА.</p> <p>Для УЗ с ролью «Администратор» с использованием данного метода доступно получение любых заявок.</p>	
URL – /tasks-service/api/v2/public/tasks/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/findById	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	
Response	Ответ JSON в HTTP-body
ResponseEntity -> CollectionResponse -> {	
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ¹
certificateAuthorityId (uuid),	Идентификатор сертификата издателя
type (string: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (string: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки

¹ Шаблоны еСА-СА и еСА-РА имеют сквозную идентификацию.

creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ¹
certificateId (uuid),	ID сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string),	Внешний ключ заявки
templateName (string),	Имя шаблона сертификата
msUpn (string)	UPN субъекта
}	

1.2.4 Метод получения заявки по внешнему ключу

GET API – Получение заявки по внешнему ключу	
<p>Для УЗ с ролью «Пользователь» с использованием данного метода доступно получение только тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ.</p> <p>Для УЗ с ролью «Оператор» с использованием данного метода доступно получение заявок, созданных данной УЗ, а также заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного еСА-СА.</p> <p>Для УЗ с ролью «Администратор» с использованием данного метода доступно получение любых заявок.</p>	
URL – /tasks-service/api/v2/public/tasks/foreign-key/{foreignKey}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/findByForeignKey	
Query	
{	
foreignKey (string)	Внешний ключ заявки
}	
Request	
-	
Response	Ответ JSON в HTTP-body
ResponseEntity -> CollectionResponse -> {	
id (uuid),	ID заявки

¹ Субъекты еСА-СА и еСА-РА имеют сквозную идентификацию.

templateId (uuid),	ID шаблона сертификата ¹
certificateAuthorityId (uuid),	Идентификатор сертификата издателя
type (string: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (string: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ²
certificateId (uuid),	ID сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string),	Внешний ключ заявки
templateName (string),	Имя шаблона сертификата
msUpn (string)	UPN субъекта
}	

1.2.5 Метод поиска заявок

GET API – Поиск заявок
<p>Для УЗ с ролью «Пользователь» в ответе данного метода возвращаются только те заявки, получателем сертификата по которым является субъект, связанный с данной УЗ.</p> <p>Для УЗ с ролью «Оператор» в ответе данного метода возвращаются заявки, созданные данной УЗ, а также заявки для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного еСА-СА.</p> <p>Для УЗ с ролью «Администратор» в ответе данного метода возвращаются все существующие заявки.</p>
URL – /tasks-service/api/v2/public/tasks
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/findAll
<p>Query</p> <pre>{</pre>

¹ Шаблоны еСА-СА и еСА-РА имеют сквозную идентификацию.

² Субъекты еСА-СА и еСА-РА имеют сквозную идентификацию.

sortDirection (string),	Направления сортировки (ASC;DESC)
sortBy (string[]),	Список полей, к которым применяется сортировка
pageOffset (integer),	Смещение от начала списка (пагинация)
pageLimit (integer),	Ограничение на размер выборки (пагинация)
search (string),	Полнотекстовый поиск
foreignKeys (string[]),	Фильтр: список внешних ключей
tempaltelds (uuid[]),	Фильтр: список ID шаблонов
creatorIds (uuid[]),	Фильтр: список ID УЗ для фильтрации
types (enum[]: PKCS10, PKCS12, SCEP, WSTEP, ON_TOKEN, UNKNOWN),	Фильтр: список типов заявки
statuses (enum[]: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Фильтр: список статусов заявки
certificateAuthorityId (string[]),	Фильтр: список ID Центров сертификации
serialnumber (string),	Поиск заявки по серийному номеру сертификата
fingerprint (string[]),	Фильтр: список отпечатков сертификатов
subjectKeyIdentifier (string[])	Фильтр: Идентификатор ключа субъекта
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ¹
certificateAuthorityId (uuid),	Идентификатор сертификата издателя
type (string: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (string: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки

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

creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ¹
certificateId (uuid),	ID сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string),	Внешний ключ заявки
templateName (string),	Имя шаблона сертификата
msUpn (string)	UPN субъекта
}	

1.2.6 Метод получения данных о выпущенном сертификате заявки по ее идентификатору

GET API – Получение данных о выпущенном сертификате заявки по ее идентификатору	
<p>Для УЗ с ролью «Пользователь» с использованием данного метода доступно получение данных о выпущенных сертификатах только по тем заявкам, получателем сертификата по которым является субъект, связанный с данной УЗ.</p> <p>Для УЗ с ролью «Оператор» с использованием данного метода доступно получение данных о выпущенных сертификатах только по созданным данным оператором заявкам, а также по заявкам для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA.</p> <p>Для УЗ с ролью «Администратор» с использованием данного метода доступно получение данных о выпущенных сертификатах по любым заявкам.</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 v3.</p>	
URL – /middleware-service/api/v2/public/tasks/{id}/certificate	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20задачи/findCertificateByTaskId	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body

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

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,	Поля разделенного имени субъекта из сертификата. В формате key-value. Где

O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]	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 ¹
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип сертификата
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant),	Дата окончания действия сертификата (ISO 8601)
status (enum: ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN),	Статус сертификата
actions: {	Доступные действия по выгрузке

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

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 (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPTHERMENT, DATA_ENCRYPTHERMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPTHER_ONLY, DECRYPTHER_ONLY, UNKNOWN),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (enum: ECU_PKIX_ANY_EXTENDED_KEY_USAGE, CSN_369791_TLS_CLIENT, CSN_369791_TLS_SERVER, CLIENT_AUTHENTICATION, CODE_SIGNING, EAP_OVER_LAN, EAP_OVER_PPP, ETSI_TSL_SIGNING, EMAIL_PROTECTION, ICAO_DEVIATION_LIST_SIGNING, ECU_INTEL_AMT, INTERNET_KEY_EXCHANGE_FOR_IPSEC, KERBEROS_CLIENT_AUTHENTICATION, ECU_KRB_PKINIT_KDC, MS_COMMERCIAL_CODE_SIGNING, MS_DOCUMENT_SIGNING, MS_EFS_RECOVERY, MS_ENCRYPTED_FILE_SYSTEM, MS_INDIVIDUAL_CODE_SIGNING, MS_SMART_CARD_LOGON,	Перечисление расширенного использования ключа

OCSP_SIGNER, ECU_ADOBE_PDF_SIGNING, PIV_CARD_AUTHENTICATION, SCVP_CLIENT, SCVP_SERVER, SIP_DOMAIN, ECU_PKIX_SSH_CLIENT, SSH_SERVER, SERVER_AUTHENTICATION, TIME_STAMPING, ICAO_MASTER_LIST_SIGNING, UNKNOWN),	
value (string),	Наименование элемента
oid (string),	OID назначения
description (string)	Описание использования ключа
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
}	

1.3 Методы экспорта файлов

1.3.1 Метод получения запроса на сертификат по идентификатору заявки

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

1.3.2 Метод получения сертификата по идентификатору заявки

GET API – Получение сертификата по идентификатору заявки	
<p>Для УЗ с ролью «Пользователь» с использованием данного метода доступно получение сертификатов только из тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ.</p> <p>Для УЗ с ролью «Оператор» с использованием данного метода доступно получение сертификата только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного еСА-СА.</p> <p>Для УЗ с ролью «Администратор» с использованием данного метода доступно получение сертификатов из любых заявок.</p> <p>Для успешного получения сертификата из заявки она должна иметь статус «COMPLETED».</p>	
URL – /export-service/api/v2/public/export/task/{id}/certificate	

Swagger:https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findCertificateByTaskId_1	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	
Response	
ResponseEntity->byte[]	

1.3.3 Метод получения цепочки сертификатов по идентификатору заявки

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

1.3.4 Метод получения контейнера pkcs12 по идентификатору заявки

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

1.3.5 Метод получения сертификата издателя сертификата по заявке

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

Query	
{	
id (uuid)	ID заявки
}	
Request	
-	
Response	
ResponseEntity->byte[]	

1.3.6 Метод получения цепочки сертификатов издателя сертификата по заявке

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

1.3.7 Метод получения CRL издателя сертификата по заявке

GET API – Получение CRL издателя сертификата по заявке
--

Для УЗ с ролью «Пользователь» с использованием данного метода доступно получение CRL издателя только из тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ.

Для УЗ с ролью «Оператор» с использованием данного метода доступно CRL издателя только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA.

Для УЗ с ролью «Администратор» с использованием данного метода доступно получение CRL издателя из любых заявок.

Для успешного получения CRL издателя из заявки она должна иметь статус «COMPLETED».

URL – /export-service/api/v2/public/export/task/{id}/issuer/crl

Swagger: <https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%20экспорта/findIssuerCrlByTaskId>

Query

{

id (uuid)

ID заявки

}

Request

-

Response

ResponseEntity->byte[]

1.4 Методы работы с субъектами

1.4.1 Метод поиска субъектов

GET API – Поиск субъектов	
Для УЗ с ролью «Пользователь» с использованием данного метода доступно только получение данных субъекта, связанный с данной УЗ.	
Для УЗ с ролью «Оператор» или «Администратор» с использованием данного метода доступно получение данных любого субъекта подключенного eCA-CA.	
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 v3.	
URL – /middleware-service/api/v2/public/subjects	
Swagger:https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20субъекты/findAll_2	
Query	
{	
id (UUID[]), [опционально]	ID субъекта ¹
resourceId (UUID[]), [опционально]	ID ресурсной системы
securityGroupId (UUID[]), [опционально]	ID группы безопасности
search (string), [опционально]	Полнотекстовый поиск (имя субъекта)
isConnected (boolean), [опционально]	Флаг: субъект подключен к ресурсной системе
isBlocked (boolean), [опционально]	Флаг: субъект заблокирован в ресурсной системе
pageLimit (integer), [опционально]	Ограничение на размер выборки (пагинация)
pageOffset (integer), [опционально]	Смещение от начала списка (пагинация)
sortBy (string[]), [опционально]	Список полей, к которым применяется сортировка
sortDirection (string) [опционально]	Направления сортировки (ASC;DESC)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта ²
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,	Поля разделенного имени субъекта

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

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

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

1.4.2 Метод получения субъекта по идентификатору

GET API – Получение субъекта по идентификатору	
Для УЗ с ролью «Пользователь» с использованием данного метода доступно только получение данных субъекта, связанный с данной УЗ.	
Для УЗ с ролью «Оператор» или «Администратор» с использованием данного метода доступно получение данных любого субъекта подключенного еCA-CA.	
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 v3.	
URL – /middleware-service/api/v2/public/subjects/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20субъекты/findById_2	
Query	
{	
id (UUID)	ID субъекта ¹
}	
Request	
-	

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

Response	Ответ JSON в HTTP-body
{	
id (UUID),	ID субъекта ¹
commonName (string),	Имя субъекта
distinguishedName (string),	Расположение субъекта в ресурсной системе
resource: {	Ресурсная система
id (uuid),	ID ресурсной системы
commonName (string),	Имя ресурсной системы
distinguishedName (string)	BaseDN точки подключения к ресурсной системе
},	
subjectName: {	Имя субъекта
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): {	Поля разделенного имени субъекта
values (string[]),	Значения компонента
editable (boolean),	Флаг: компонент доступен для редактирования
}	
},	
subjectAltName: {	Альтернативное имя субъекта
(enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): {	Поля альтернативного имени субъекта
values (string[]),	Значения компонента
editable (boolean),	Флаг: компонент доступен для редактирования
}	
},	
isConnected (boolean),	Флаг: субъект подключен к ресурсной системе
isBlocked (boolean),	Флаг: субъект заблокирован в ресурсной системе
modify (instant),	Время изменения (ISO 8601)
certificatesCount (integer),	Количество действующих сертификатов у субъекта
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

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

1.5 Методы работы с шаблонами

1.5.1 Метод поиска шаблонов

GET API – Поиск шаблонов	
<p>Для УЗ с ролью «Пользователь» с использованием данного метода доступно получение только тех шаблонов, которые указаны в правилах выпуска с режимом обработки «Автоматический выпуск» или «Ручная обработка» для ассоциированного с текущей учетной записью субъекта.</p> <p>Для УЗ с ролью «Оператор» с использованием данного метода доступно получение только тех шаблонов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA.</p> <p>Для УЗ с ролью «Администратор» с использованием данного метода доступно получение любых шаблонов.</p>	
URL – /middleware-service/api/v2/public/templates	
Swagger:https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20шаблоны/findAll_1	
Query	
{	
pageLimit (integer), [опционально]	Ограничение на размер выборки (пагинация)
pageOffset (integer), [опционально]	Смещение от начала списка (пагинация)
sortBy (string[]), [опционально]	Список полей, к которым применяется сортировка
sortDirection (string), [опционально]	Направления сортировки (ASC;DESC)
types (enum[]: EMBEDDED, CLONED, IMPORTED, UNKNOWN), [опционально]	Тип шаблона
endEntityType (enum[]: USER, DEVICE, ROOT_CA, SUB_CA, UNKNOWN) [опционально],	Тип субъекта
certificateType (enum[]: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN), [опционально]	Тип выпускаемого сертификата
search (string), [опционально]	Полнотекстовый поиск по имени шаблона
removed (boolean), [опционально]	Флаг: шаблон удален
id (UUID[]), [опционально]	ID шаблона
notId (UUID[]), [опционально]	Исключая ID шаблона
keyAlgorithm (enum[]: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN) [опционально]	Фильтр: алгоритм ключа включен в шаблоне ¹
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID шаблона ²
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.

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.

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

	Шаблоны, у которых в поле «Центр сертификации» указано «Любой», будут иметь в данном поле значение NULL.
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)
}	

1.5.2 Метод получения шаблона по идентификатору

GET API – Получение шаблона по идентификатору	
<p>Для УЗ с ролью «Пользователь» с использованием данного метода доступно получение только тех шаблонов, которые указаны в правилах выпуска с режимом обработки «Автоматический выпуск» или «Ручная обработка» для ассоциированного с текущей учетной записью субъекта.</p> <p>Для УЗ с ролью «Оператор» с использованием данного метода доступно получение только тех шаблонов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA.</p> <p>Для УЗ с ролью «Администратор» с использованием данного метода доступно получение любых шаблонов.</p> <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 v3.</p>	
URL – /middleware-service/api/v2/public/templates/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20шаблоны/findById_1	
Query	
{	
templateId (UUID)	ID шаблона ¹
}	
Request	
-	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID шаблона ²
name (string),	Имя шаблона
validity (int64),	Время действия выпускаемого сертификата (мс)
certificateType (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип выпускаемого сертификата
type (enum: EMBEDDED, CLONED, IMPORTED, UNKNOWN),	Тип шаблона
certificateAuthorityId (UUID),	ID ЦС, который должен использоваться при выпуске сертификата по данному шаблону Шаблоны, у которых в поле «Центр сертификации» указано «Любой», будут иметь в данном поле значение NULL
endEntityType (enum: USER, DEVICE, ROOT_CA, SUB_CA, UNKNOWN),	Тип субъекта
removed (boolean),	Флаг: шаблон удален
rsa: {	Описание RSA-криптографии

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

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

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	Значение расширения
(enum[]):DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPTHERMENT, DATA_ENCRYPTHERMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPTHER_ONLY, DECRYPTHER_ONLY, UNKNOWN)	
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
critical (boolean),	Флаг: расширение критическое
values (string[])	Значение расширения (OIDs)
},	
policies: {	Политики сертификата
critical (boolean),	Флаг: расширение критическое
values (string[])	Значение расширения (OIDs)
},	
subjectDN: [{	Имя субъекта сертификата
index (int32),	Индекс (для сортировки, по умолчанию – 0)
name (string),	Имя компонента
description (string),	Описание компонента
required (boolean),	Флаг: обязателен к заполнению
validation (boolean),	Флаг: валидация значения
modifiable (boolean),	Флаг: доступен к редактированию
regex (string),	Регулярное значение для валидации значения
defaultValue (string),	Значение по умолчанию
alert (string),	Предупреждение о неудачной валидации значения
code (enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, 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
regex (string),	Регулярное значение для валидации значения
defaultValue (string),	Значение по умолчанию
alert (string),	Предупреждение о неудачной валидации значения
code (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN),	Код компонента
generalName (int32),	Идентификатор компонента в RFC
oid (string)	OID компонента в RFC
}},	
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.6 Методы работы с сертификатами

1.6.1 Метод отзыва сертификата по идентификатору заявки

POST API – Отзыв сертификата по идентификатору заявки	
<p>Для УЗ с ролью «Пользователь» с использованием данного метода доступен отзыв сертификатов только из тех заявок, получателем сертификата по которым является субъект, связанный с данной УЗ.</p> <p>Для УЗ с ролью «Оператор» с использованием данного метода доступен отзыв сертификатов из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного еСА-СА.</p> <p>Для УЗ с ролью «Администратор» с использованием данного метода доступен отзыв сертификатов из любых заявок.</p> <p>Для успешного получения сертификата из заявки она должна иметь статус «COMPLETED».</p>	
URL – /tasks-service/api/v2/public/tasks/{id}/certificate/revoke	
Swagger:	
Query	
{	
id (uuid)	ID заявки
}	
Request	
{	
reason (enum: UNSPECIFIED, KEY_COMPROMISE,	Причина отзыва

CA_COMPROMISE, AFFILIATION_CHANGED, SUPERSEDED, CESSATION_OF_OPERATION, REMOVE_FROM_CRL, PRIVILEGE_WITHDRAWN, AA_COMPROMISE, UNKNOWN),	
comment (string)	Комментарий
}	
Response	
-	

1.7 Методы получения информации о сервисах

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

1.7.1.1 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-RA
"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-RA
"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-RA
"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-RA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

1.7.1.2 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.
}	

1.7.1.3 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
}	
}	

1.7.1.4 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), описание которых приведено в Приложении Б.	

1.7.2 Методы получения информации о сервисе заявок (tasks-service)

1.7.2.1 Метод получения эндпоинтов для запроса информации о сервисе заявок (tasks-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе заявок	
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 – tasks-service/actuator	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
" _links": {	
"self": {	
"href": "http://HOST/tasks-service/actuator",	URL эндпоинта, который возвращает эндпоинты для запроса информации о сервисе, где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health": {	
"href": "http://HOST/tasks-service/actuator/health",	URL эндпоинта, который возвращает информацию о состоянии сервиса (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health-path": {	
"href": "http://HOST/tasks-service/actuator/health/{*path}",	URL зарезервированного эндпоинта под будущие реализации
"templated": true	Флаг наличия переменной в URL
},	
"info": {	
"href": "http://HOST/tasks-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/tasks-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
}	

}	
}	

1.7.2.2 Метод получения информации о состоянии сервиса заявок (tasks -service)

GET – Получение информации о состоянии сервиса заявок	
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 – tasks-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.
}	

1.7.2.3 Метод получения информации о сервисе заявок (tasks-service)

GET – Получение информации о сервисе заявок	
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 – tasks-service/actuator/info	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

1.7.2.4 Метод получения Prometheus-метрик сервиса заявок (tasks-service)

GET – Получение Prometheus-метрик сервиса заявок	
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 – tasks-service/actuator/prometheus	
Swagger: -	

–	Query	
–	Request	
	Response	
Метод возвращает метрики сервиса в формате Prometheus (text/plain), описание которых приведено в Приложении Б.		

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

1.7.3.1 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-RA
"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-RA
"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-RA
"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-RA
"templated": false		Флаг наличия переменной в URL
}		
}		
}		

1.7.3.2 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: <ul style="list-style-type: none"> - UP - работает; - DOWN - не работает; - OUT_OF_SERVICE - выключен; - UNKNOWN - нет информации.
}	

1.7.3.3 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
}	
}	

1.7.3.4 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), описание которых приведено в Приложении Б.

1.7.4 Методы получения информации о сервисе правил выпуска (policies-service)

1.7.4.1 Метод получения эндпоинтов для запроса информации о сервисе правил выпуска (policies-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе правил выпуска	
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 – policies-service/actuator	
Swagger: -	
Query	
-	
Request	
-	
Response	Ответ JSON в HTTP-body
{	
" links": {	
"self": {	
"href": "http://HOST/policies-service/actuator",	URL эндпоинта, который возвращает эндпоинты для запроса информации о сервисе, где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health": {	
"href": "http://HOST/policies-service/actuator/health",	URL эндпоинта, который возвращает информацию о состоянии сервиса (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health-path": {	
"href": "http://HOST/policies-service/actuator/health/{*path}",	URL зарезервированного эндпоинта под будущие реализации
"templated": true	Флаг наличия переменной в URL
},	
"info": {	
"href": "http://HOST/policies-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/policies-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

1.7.4.2 Метод получения информации о состоянии сервиса правил выпуска (policies service)

GET – Получение информации о состоянии сервиса правил выпуска	
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 – policies-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.
}	

1.7.4.3 Метод получения информации о сервисе правил выпуска (policies-service)

GET – Получение информации о сервисе правил выпуска	
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 – policies-service/actuator/info	
Swagger: -	
Query	
–	
Request	
–	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

1.7.4.4 Метод получения Prometheus-метрик сервиса правил выпуска (policies-service)

GET – Получение Prometheus-метрик сервиса правил выпуска	
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 – policies-service/actuator/prometheus	
Swagger: -	
Query	
–	
Request	
–	
Response	
Метод возвращает метрики сервиса в формате Prometheus (text/plain), описание которых приведено в Приложении Б.	

1.7.5 Методы получения информации о сервисе интеграции с центром сертификации (ca-adapter-service)

1.7.5.1 Метод получения эндпоинтов для запроса информации о сервисе интеграции с центром сертификации (ca-adapter-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе интеграции с центром сертификации	
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 – ca-adapter-service/actuator	
Swagger: -	
Query	-
Request	-
Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/ca-adapter-service/actuator",	
"templated": false	
},	
"health": {	
"href": "http://HOST/ca-adapter-service/actuator/health",	
"templated": false	
},	
"health-path": {	
"href": "http://HOST/ca-adapter-service/actuator/health/{*path}",	
"templated": true	
},	
"info": {	
"href": "http://HOST/ca-adapter-service/actuator/info",	
"templated": false	
},	
"prometheus": {	
"href": "http://HOST/ca-adapter-service/actuator/prometheus",	
"templated": false	
}	
}	
}	

1.7.5.2 Метод получения информации о состоянии сервиса интеграции с центром сертификации (ca-adapter-service)

GET – Получение информации о состоянии сервиса интеграции с центром сертификации	
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 – ca-adapter-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.
}	

1.7.5.3 Метод получения информации о сервисе интеграции с центром сертификации (ca-adapter-service)

GET – Получение информации о сервисе интеграции с центром сертификации	
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 – ca-adapter-service/actuator/info	
Swagger: -	
Query	
Request	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

1.7.5.4 Метод получения Prometheus-метрик сервиса интеграции с центром сертификации (ca-adapter-service)

GET – Получение Prometheus-метрик сервиса интеграции с центром сертификации	
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 – ca-adapter-service/actuator/prometheus	
Swagger: -	
Query	
Request	
Response	Метод возвращает метрики сервиса в формате Prometheus (text/plain), описание которых приведено в Приложении Б.

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

1.7.6.1 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",	URL эндпоинта, который возвращает эндпоинты для запроса информации о сервисе, где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health": {	
"href": "http://HOST/settings-service/actuator/health",	URL эндпоинта, который возвращает информацию о состоянии сервиса (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health-path": {	
"href": "http://HOST/settings-service/actuator/health/{*path}",	URL зарезервированного эндпоинта под будущие реализации
"templated": true	Флаг наличия переменной в URL
},	
"info": {	
"href": "http://HOST/settings-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"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-RA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

1.7.6.2 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
<pre>{ status (enum: UP, DOWN, OUT_OF_SERVICE, UNKNOWN) }</pre>	<p>Статус (состояние) сервиса безопасности. Возможные значения:</p> <ul style="list-style-type: none"> - UP - работает; - DOWN - не работает; - OUT_OF_SERVICE - выключен; - UNKNOWN - нет информации.

1.7.6.3 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
<pre>{ "application": { name (string) version (string) } }</pre>	
	Название сервиса
	Версия сервиса

1.7.6.4 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), описание которых приведено в Приложении Б.

1.7.7 Методы получения информации о сервисе промежуточного взаимодействия (middleware-service)

1.7.7.1 Метод получения эндпоинтов для запроса информации о сервисе промежуточного взаимодействия (middleware-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе промежуточного взаимодействия
--

<p>Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:</p> <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 – middleware-service/actuator	
Swagger: -	
Query	
–	
Request	
–	
Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/middleware-service/actuator",	URL эндпоинта, который возвращает эндпоинты для запроса информации о сервисе, где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health": {	
"href": "http://HOST/middleware-service/actuator/health",	URL эндпоинта, который возвращает информацию о состоянии сервиса (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health-path": {	
"href": "http://HOST/middleware-service/actuator/health/{*path}",	URL зарезервированного эндпоинта под будущие реализации
"templated": true	Флаг наличия переменной в URL
},	
"info": {	
"href": "http://HOST/middleware-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/middleware-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

1.7.7.2 Метод получения информации о состоянии сервиса промежуточного взаимодействия (middleware-service)

<p>GET – Получение информации о состоянии сервиса промежуточного взаимодействия</p> <p>Доступность метода для неаутентифицированного пользователя зависит от значения параметра «actuator_authenticate» конфигурационного файла:</p> <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 – middleware-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:

	<ul style="list-style-type: none"> - UP - работает; - DOWN - не работает; - OUT_OF_SERVICE - выключен; - UNKNOWN - нет информации.
}	

1.7.7.3 Метод получения информации о сервисе промежуточного взаимодействия (middleware-service)

GET – Получение информации о сервисе промежуточного взаимодействия	
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 – middleware-service/actuator/info	
Swagger: -	
Query	
Request	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

1.7.7.4 Метод получения Prometheus-метрик сервиса промежуточного взаимодействия (middleware-service)

GET – Получение Prometheus-метрик сервиса промежуточного взаимодействия (middleware-service)	
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 – middleware-service/actuator/prometheus	
Swagger: -	
Query	
Request	
Response	Метод возвращает метрики сервиса в формате Prometheus (text/plain), описание которых приведено в Приложении Б.

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

1.7.8.1 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-RA
"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-RA
"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-RA
"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-RA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

1.7.8.2 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: <ul style="list-style-type: none"> UP – работает; DOWN – не работает; OUT_OF_SERVICE – выключен; UNKNOWN – нет информации.
}	

1.7.8.3 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
}	
}	

1.7.8.4 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), описание которых приведено в Приложении Б.	

1.7.9 Методы получения информации о сервисе экспорта данных (export-service)

1.7.9.1 Метод получения эндпоинтов для запроса информации о сервисе экспорта данных (export-service)

GET – Получение списка доступных эндпоинтов для запроса информации о сервисе экспорта 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 – export-service/actuator	
Swagger: -	
Query	
Request	

Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/export-service/actuator",	URL эндпоинта, который возвращает эндпоинты для запроса информации о сервисе, где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health": {	
"href": "http://HOST/export-service/actuator/health",	URL эндпоинта, который возвращает информацию о состоянии сервиса (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health-path": {	
"href": "http://HOST/export-service/actuator/health/{*path}",	URL зарезервированного эндпоинта под будущие реализации
"templated": true	Флаг наличия переменной в URL
},	
"info": {	
"href": "http://HOST/export-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/export-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
}	
}	

1.7.9.2 Метод получения информации о состоянии сервиса экспорта данных (export-service)

GET – Получение информации о состоянии сервиса экспорта 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 – export-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: <ul style="list-style-type: none"> UP – работает; DOWN – не работает; OUT_OF_SERVICE – выключен; UNKNOWN – нет информации.
}	

1.7.9.3 Метод получения информации о сервисе экспорта данных (export-service)

GET – Получение информации о сервисе экспорта 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;

<ul style="list-style-type: none"> если параметр имеет значение «false» (по умолчанию), доступ к методу будет разрешен без аутентификации. 	
URL – export-service/actuator/info	
Swagger: -	
Query	
Request	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	
version (string)	
}	
}	

1.7.9.4 Метод получения Prometheus-метрик сервиса экспорта данных (export-service)

GET – Получение Prometheus-метрик сервиса экспорта 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 – export-service/actuator/prometheus	
Swagger: -	
Query	
Request	
Response	Метод возвращает метрики сервиса в формате Prometheus (text/plain), описание которых приведено в Приложении Б.

1.7.10 Методы получения информации о SCEP-сервисе (scep-service)

1.7.10.1 Метод получения эндпоинтов для запроса информации о SCEP-сервисе (scep-service)

GET – Получение списка доступных эндпоинтов для запроса информации о SCEP-сервисе	
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 – scep-service/actuator	
Swagger: -	
Query	
Request	
Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/scep-service/actuator",	URL эндпоинта, который возвращает эндпоинты для запроса информации о сервисе, где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	

"health": {	
"href": "http://HOST/scep-service/actuator/health",	URL эндпоинта, который возвращает информацию о состоянии сервиса (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health-path": {	
"href": "http://HOST/scep-service/actuator/health/{*path}",	URL зарезервированного эндпоинта под будущие реализации
"templated": true	Флаг наличия переменной в URL
},	
"info": {	
"href": "http://HOST/scep-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"prometheus": {	
"href": "http://HOST/scep-service/actuator/prometheus",	URL эндпоинта, который возвращает метрики сервиса в формате Prometheus (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

1.7.10.2 Метод получения информации о состоянии SCEP-сервиса (scep -service)

GET – Получение информации о состоянии SCEP-сервиса	
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 – scep-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: <ul style="list-style-type: none"> UP – работает; DOWN – не работает; OUT_OF_SERVICE – выключен; UNKNOWN – нет информации.
}	

1.7.10.3 Метод получения информации о SCEP-сервисе (scep -service)

GET – Получение информации о SCEP-сервисе	
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 – scep-service/actuator/info	
Swagger: -	
Query	
–	
Request	
–	

Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

1.7.10.4 Метод получения Prometheus-метрик SCEP-сервиса (scep -service)

GET – Получение Prometheus-метрик SCEP-сервиса	
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 – scep-service/actuator/prometheus	
Swagger: -	
Query	
–	
Request	
–	
Response	
Метод возвращает метрики сервиса в формате Prometheus (text/plain), описание которых приведено в Приложении Б.	

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

1.7.11.1 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",	URL эндпоинта, который возвращает эндпоинты для запроса информации о сервисе, где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health": {	
"href": "http://HOST/external-integration-service/actuator/health",	URL эндпоинта, который возвращает информацию о состоянии сервиса (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"templated": false	Флаг наличия переменной в URL
},	
"health-path": {	

"href": "http://HOST/external-integration-service/actuator/health/{*path}",	URL зарезервированного эндпоинта под будущие реализации
"templated": true	Флаг наличия переменной в URL
},	
"info": {	
"href": "http://HOST/external-integration-service/actuator/info",	URL эндпоинта, который возвращает информацию о сервисе (подробнее см. ниже), где HOST – адрес хоста eCA-RA
"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-RA
"templated": false	Флаг наличия переменной в URL
}	
}	
}	

1.7.11.2 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.
}	

1.7.11.3 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
}	

--	--

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

GET – Получение Prometheus-метрик сервиса внешних интеграций (external-integration-service)	
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), описание которых приведено в Приложении Б.	

1.7.12 Методы получения информации о WSTEP-сервисе (wstep-enrollment-service)

1.7.12.1 Метод получения эндпоинтов для запроса информации о WSTEP-сервисе (wstep-enrollment-service)

GET – Получение списка доступных эндпоинтов для запроса информации о WSTEP-сервисе	
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 – wstep-enrollment-service/actuator	
Swagger: -	
Query	-
Request	-
Response	Ответ JSON в HTTP-body
{	
"links": {	
"self": {	
"href": "http://HOST/wstep-enrollment-service/actuator",	
"templated": false	
},	
"health": {	
"href": "http://HOST/wstep-enrollment-service/actuator/health",	
"templated": false	
},	
"health-path": {	
"href": "http://HOST/wstep-enrollment-service/actuator/health/{*path}",	
"templated": true	
},	
"info": {	
"href": "http://HOST/wstep-enrollment-service/actuator/info",	
"templated": false	
}	

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

1.7.12.2 Метод получения информации о состоянии WSTEP-сервисе (wstep-enrollment-service)

GET – Получение информации о состоянии WSTEP-сервисе	
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 – wstep-enrollment-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: <ul style="list-style-type: none"> – UP – работает; – DOWN – не работает; – OUT_OF_SERVICE – выключен; – UNKNOWN – нет информации.
}	

1.7.12.3 Метод получения информации о WSTEP-сервисе (wstep-enrollment-service)

GET – Получение информации о WSTEP-сервисе	
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 – wstep-enrollment-service/actuator/info	
Swagger: -	
Query	
–	
Request	
–	
Response	Ответ JSON в HTTP-body
{	
"application": {	
name (string)	Название сервиса
version (string)	Версия сервиса
}	
}	

1.7.12.4 Метод получения Prometheus-метрик WSTEP-сервиса (wstep-enrollment-service)

GET – Получение Prometheus-метрик WSTEP-сервиса	
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 – wstep-enrollment-service/actuator/prometheus	
Swagger: -	
-	Query
-	Request
Response	
Метод возвращает метрики сервиса в формате Prometheus (text/plain), описание которых приведено в Приложении Б.	

1.7.13 Описание Prometheus-метрик сервисов

1.7.13.1 Базовые метрики сервиса

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.

1.7.13.2 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.

1.7.13.3 Метрики исполнителей (Thread Pools)

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.

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.

1.7.13.4 Метрики пула подключений к БД (HikariCP)

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).

1.7.13.5 5 Метрики HTTP-клиента

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.

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.

1.7.13.6 Метрики HTTP-сервера

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.

Завершённые 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).

1.7.13.7 JDBC-метрики (альтернативное представление HikariCP):

- `jdbc_connections_active{name="dataSource"} gauge`. Количество активных подключений.
- `jdbc_connections_idle{name="dataSource"} gauge`. Количество простаивающих подключений.
- `jdbc_connections_max{name="dataSource"} gauge`. Максимальный размер пула.
- `jdbc_connections_min{name="dataSource"} gauge`. Минимальный размер пула.

1.7.13.8 Метрики JVM (Java Virtual Machine)

Общая информация:

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

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).

Классы:

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

Компиляция:

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

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).

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.

Память (максимальная):

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

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).

Память (используемая):

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

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).

1.7.13.9 Метрики логирования (Logback):

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

1.7.13.10 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).

1.7.13.11 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).

1.7.13.12 Метрики безопасности (Spring Security)

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.

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.

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, `AescaAuthenticationExceptionFilter`, `ApiKeyAuthenticationFilter`, `UserPrincipalAuthenticationFilter` и др.). Набор фильтров зависит от конфигурации безопасности конкретного сервиса.

Активные фильтры безопасности:

- `spring_security_filterchains_active_seconds_count{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.

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`.

Время выполнения фильтров:

- `spring_security_filterchains_seconds_count{error="...", 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).

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`.

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.

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.

1.7.13.13 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.

1.7.13.14 Метрики планировщика задач

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.

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.

1.7.13.15 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.

1.8 Метод получения версии сервиса внешних интеграций

GET API – Получение версии 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. 	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5Bv2%5D%20Контроллер%20версии%20модуля/getApiVersion	
URL – external-integration-service/api/version	
Query	
-	
Request	
-	
Response	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
version (string)	Версия сервиса внешних интеграций
}	

2 ОПИСАНИЕ МЕТОДОВ REST API ВЕРСИИ 3

2.1 Методы работы с заявками

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

POST API – Создание новой заявки на выпуск сертификата в контейнере PKCS#12	
<p>Для УЗ с ролью «Получатель сертификата» с использованием данного метода доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью.</p> <p>Для УЗ с ролью «Администратор» или «Оператор» с использованием данного метода доступно создание заявок для субъектов ресурсных систем eCA-CA.</p>	
URL – /tasks-service/api/v3/public/tasks/pkcs12	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html/#/%5BAPI%20V3%5D%20Контроллер%3A%20Задачи/registerPkcs12	
Query	
{	
certificateAuthorityId (uuid), [опционально]	ID Центра сертификации, на котором должен быть выпущен сертификат по заявке. Если для данного параметра не указано значение, выпуск будет осуществляться на Центре сертификации подключенного eCA-CA, который определен в используемом шаблоне (поле «templateId»). Если в используемом шаблоне не определен конкретный центр сертификации (указано значение «Любой»), выпуск будет осуществляться на активном на момент создания заявки Центре сертификации подключенного eCA-CA.
subjectId (uuid), [опционально]	ID субъекта ¹ . Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь».
templateId (uuid), [обязательно]	ID шаблона сертификата ² .
foreignKey (string) [опционально]	Внешний ключ заявки. Внешний ключ должен быть уникальным.
}	
Request	
{	
subjectDN: { (enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] },	Поля разделенного имени субъекта, которое должно попасть в сертификат. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра

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

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

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, 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 ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ¹
type (enum: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (enum: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ²
certificateId(uuid),	Идентификатор сертификата. Заполняется по итогам успешной обработки заявки.
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string)	Внешний ключ заявки
}	

2.1.2 Методы создания новой заявки на основании запроса PKCS#10

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

POST API – Создание новой заявки на основании запроса PKCS#10
Для УЗ с ролью «Получатель сертификата» с использованием данного метода доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для УЗ с ролью «Администратор» или «Оператор» с использованием данного метода доступно создание заявок для субъектов ресурсных систем eCA-CA.
URL – /tasks-service/api/v3/public/tasks/pkcs10

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

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

Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5BAPI%20V3%5D%20Контроллер%3A%20Задачи/registerPkcs10AsFileDescription	
Query {	
certificateAuthorityId (uuid), [опционально]	ID Центра сертификации, на котором должен быть выпущен сертификат по заявке. Если для данного параметра не указано значение, выпуск будет осуществляться на Центре сертификации подключенного eCA-CA, который определен в используемом шаблоне (поле «templateId»). Если в используемом шаблоне не определен конкретный центр сертификации (указано значение «Любой»), выпуск будет осуществляться на активном на момент создания заявки Центре сертификации подключенного eCA-CA.
subjectId (uuid), [опционально]	ID субъекта ¹ . Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь». Может быть указан при использовании метода учетной записью с ролью «Администратор». ²
templateId (uuid), [обязательно]	ID шаблона сертификата ³ .
foreignKey (string) [опционально]	Внешний ключ заявки. Внешний ключ должен быть уникальным.
}	
Request {	
request (MultipartFile),	PKCS#10 файл запроса на сертификат
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,	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

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

² Если при создании администратором заявки на сертификат на основании запроса во входных параметрах метода не указан subjectId, ПО eCA-RA на основании указанного в запросе значения Common name автоматически определяет (или создаёт) получателя сертификата по заявке в соответствии со следующими правилами:

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

³ Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

INNLE, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] , [опционально]	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[] }	Поля альтернативного имени субъекта, которое должно попасть в сертификат. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра. Необязательный параметр, задается в случае невозможности поместить данные в PKCS10. Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.
}	
Response ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ¹
type (enum: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (enum: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ²
certificateId (string),	Идентификатор сертификата. Заполняется по итогам успешной обработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string)	Внешний ключ заявки
}	

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

POST API – Создание новой заявки на основании запроса PKCS#10
Для УЗ с ролью «Получатель сертификата» с использованием данного метода доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для УЗ с ролью «Администратор» или «Оператор» с использованием данного метода доступно создание заявок для субъектов ресурсных систем eCA-CA.
URL – /tasks-service/api/v3/public/tasks/pkcs10

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

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

Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5BAPI%20V3%5D%20Контроллер%3A%20Задачи/registerPkcs10AsFileDescription	
Query {	
certificateAuthorityId (uuid), [опционально]	ID Центра сертификации, на котором должен быть выпущен сертификат по заявке. Если для данного параметра не указано значение, выпуск будет осуществляться на Центре сертификации подключенного еCA-CA, который определен в используемом шаблоне (поле «templateId»). Если в используемом шаблоне не определен конкретный центр сертификации (указано значение «Любой»), выпуск будет осуществляться на активном на момент создания заявки Центре сертификации подключенного еCA-CA.
subjectId (uuid), [опционально]	ID субъекта ¹ . Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь». Может быть указан при использовании метода учетной записью с ролью «Администратор». ²
templateId (uuid), [обязательно]	ID шаблона сертификата ³ .
foreignKey (string) [опционально]	Внешний ключ заявки. Внешний ключ должен быть уникальным.
}	
Request {	
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)
},	

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

² Если при создании администратором заявки на сертификат на основании запроса во входных параметрах метода не указан subjectId, ПО еCA-RA на основании указанного в запросе значения Common name автоматически определяет (или создаёт) получателя сертификата по заявке в соответствии со следующими правилами:

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

³ Шаблоны еCA-CA и еCA-RA имеют сквозную идентификацию.

<code>subjectName: { (enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, ROLE, UNKNOWN): string[] }, [опционально]</code>	<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>
<code>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[] }</code>	<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. Необязательный параметр, задается в случае невозможности поместить данные в PKCS10. Значения полей, указанные в subjectAltName, переопределяют значения соответствующих полей SAN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.</p>
}	
Response ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ¹
type (enum: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (enum: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ²
certificateId (string),	Идентификатор сертификата. Заполняется по итогам успешной обработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string)	Внешний ключ заявки
}	

2.1.3 Метод получения данных сертификата, выпущенного по заявке, по идентификатору заявки

GET API – Получение данных сертификата, выпущенного по заявке, по идентификатору заявки

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

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

<p>Для УЗ с ролью «Получатель сертификата» с использованием данного метода доступно получение данных о выпущенных сертификатах только по тем заявкам, получателем сертификата по которым является субъект, связанный с данной УЗ.</p> <p>Для УЗ с ролью «Оператор» с использованием данного метода доступно получение данных о выпущенных сертификатах только по созданным данным оператором заявкам, а также по заявкам для субъектов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA.</p> <p>Для УЗ с ролью «Администратор» с использованием данного метода доступно получение данных о выпущенных сертификатах по любым заявкам.</p>	
URL – /middleware-service/api/v3/public/tasks/{id}/certificate	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5BAPI%20V3%5D%20Контроллер%3A%20Задачи/findCertificateByTaskId	
Query	
{	
id (uuid)	ID заявки
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (uuid),	Идентификатор сертификата
chain: {	Цепочка сертификатов (рекурсивный объект)
id (uuid),	Идентификатор сертификата
name (string),	Имя сертификата (на основе CN)
subjectDN: {	Имя субъекта сертификата
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, 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: { (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
},	
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 ¹
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 действителен
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 (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCIPHERMENT, DATA_ENCIPHERMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCIPHER_ONLY, DECIPHER_ONLY, UNKNOWN),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (enum: 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,	Перечисление расширенного использования ключа

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

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, UNKNOWN),	
value (string),	Наименование элемента
oid (string),	OID назначения
description (string),	Описание использования ключа
created (instant),	Дата создания ECU (ISO 8601)
updated (instant),	Дата создания ECU (ISO 8601)
default (boolean)	Флаг: ECU 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 УЗ
subjectId (uuid),	Идентификатор субъекта
created (instant),	Время создания (ISO 8601)
updated (instant)	Время обновления (ISO 8601)
}	

2.2 Методы работы с субъектами

2.2.1 Метод поиска субъектов

GET API – Поиск субъектов	
Для УЗ с ролью «Получатель сертификата» с использованием данного метода доступно только получение данных субъекта, связанный с данной УЗ.	
Для УЗ с ролью «Оператор» или «Администратор» с использованием данного метода доступно получение данных любого субъекта подключенного eCA-CA.	
URL – /middleware-service/api/v3/public/subjects	
Swagger:https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5BAPI%20V3%5D%20Контроллер%3A%20субъекты/findAll_1	
Query	
{	
id (UUID[]), [опционально]	ID субъекта ¹
resourceId (UUID[]), [опционально]	ID ресурсной системы
securityGroupId (UUID[]), [опционально]	ID группы безопасности
search (string), [опционально]	Полнотекстовый поиск (имя субъекта)

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

isConnected (boolean), [опционально]	Флаг: субъект подключен к ресурсной системе
isBlocked (boolean), [опционально]	Флаг: субъект заблокирован в ресурсной системе
pageLimit (integer), [опционально]	Ограничение на размер выборки (пагинация)
pageOffset (integer), [опционально]	Смещение от начала списка (пагинация)
sortBy (string[]), [опционально]	Список полей, к которым применяется сортировка
sortDirection (string) [опционально]	Направления сортировки (ASC;DESC)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта ¹
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
modify (instant),	Время изменения (ISO 8601)
certificatesCount (integer),	Количество действующих сертификатов у субъекта
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)

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

sid (string)	SID субъекта
}	

2.2.2 Метод получения субъекта по идентификатору

GET API – Получение субъекта по идентификатору	
Для УЗ с ролью «Получатель сертификата» с использованием данного метода доступно только получение данных субъекта, связанный с данной УЗ. Для УЗ с ролью «Оператор» или «Администратор» с использованием данного метода доступно получение данных любого субъекта подключенного eCA-CA.	
URL – /middleware-service/api/v3/public/subjects/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#!/%5BAPI%20V3%5D%20Контроллер%3A%20субъекты/findById_2	
Query	
{	
id (UUID)	ID субъекта ¹
}	
Request	
-	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID субъекта ²
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

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

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

}	
},	
isConnected (boolean),	Флаг: субъект подключен к ресурсной системе
isBlocked (boolean),	Флаг: субъект заблокирован в ресурсной системе
modify (instant),	Время изменения (ISO 8601)
certificatesCount (integer),	Количество действующих сертификатов у субъекта
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
sid (string)	SID субъекта
}	

2.3 Методы работы с шаблонами

2.3.1 Метод получения шаблона по идентификатору

GET API – Получение шаблона по идентификатору	
Для УЗ с ролью «Получатель сертификата» с использованием данного метода доступно получение только тех шаблонов, которые указаны в правилах выпуска с режимом обработки «Автоматический выпуск» или «Ручная обработка» для ассоциированного с текущей учетной записью субъекта.	
Для УЗ с ролью «Оператор» с использованием данного метода доступно получение только тех шаблонов, доступ к которым данному оператору предоставлен по правилам доступа подключенного eCA-CA.	
Для УЗ с ролью «Администратор» с использованием данного метода доступно получение любых шаблонов.	
URL – /middleware-service/api/v3/public/templates/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5BAPI%20V3%5D%20Контроллер%3A%20шаблоны/findById_1	
Query	
{	
templateId (UUID)	ID шаблона ¹
}	
Request	
-	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID шаблона ²
name (string),	Имя шаблона
validity (int64),	Время действия выпускаемого сертификата (мс)
certificateType (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип выпускаемого сертификата
type (enum: EMBEDDED, CLONED, IMPORTED, UNKNOWN),	Тип шаблона
endEntityType (enum: USER, DEVICE, ROOT_CA, SUB_CA, UNKNOWN),	Тип субъекта
removed (boolean),	Флаг: шаблон удален
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,	Значение расширения

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

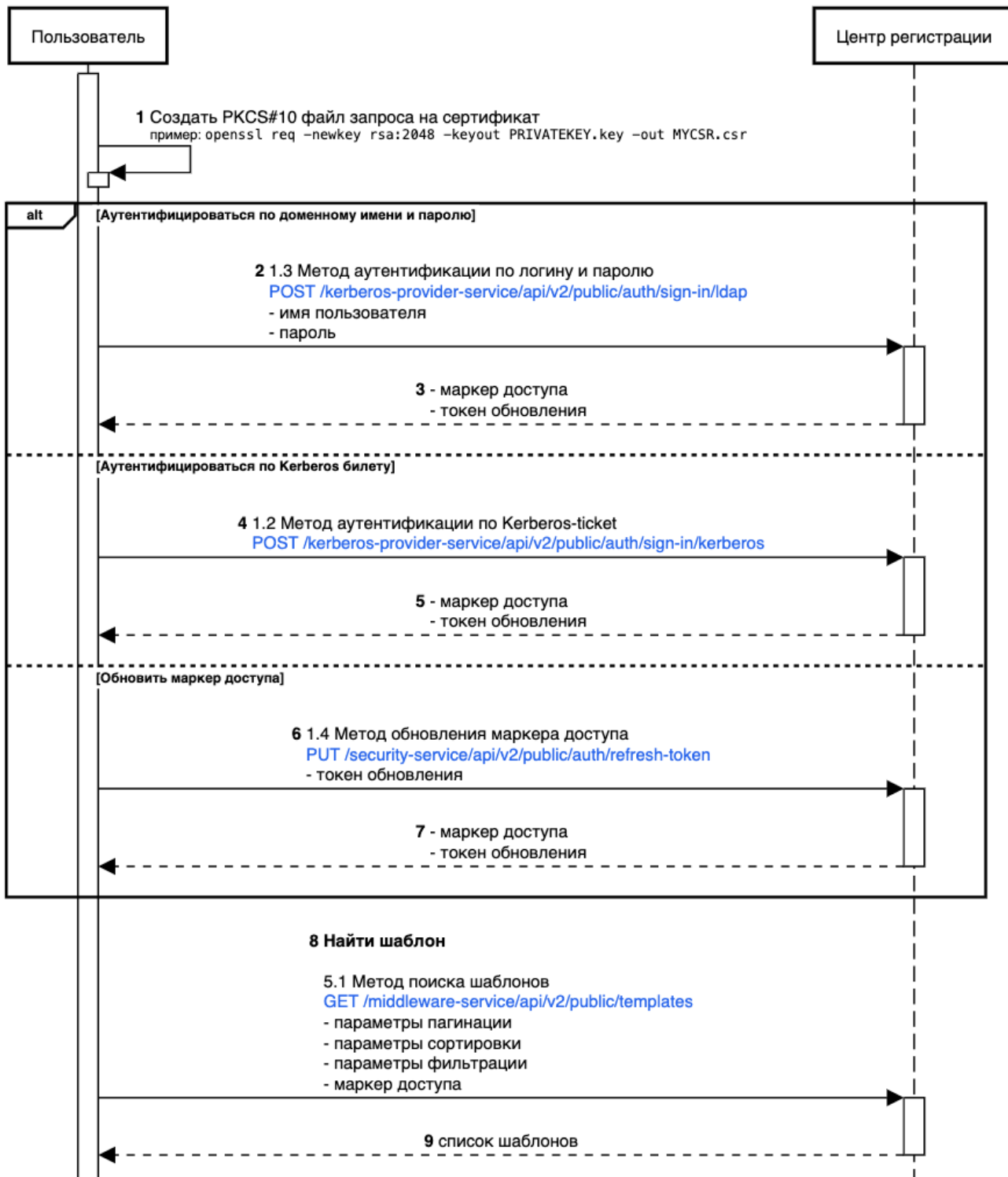
² Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

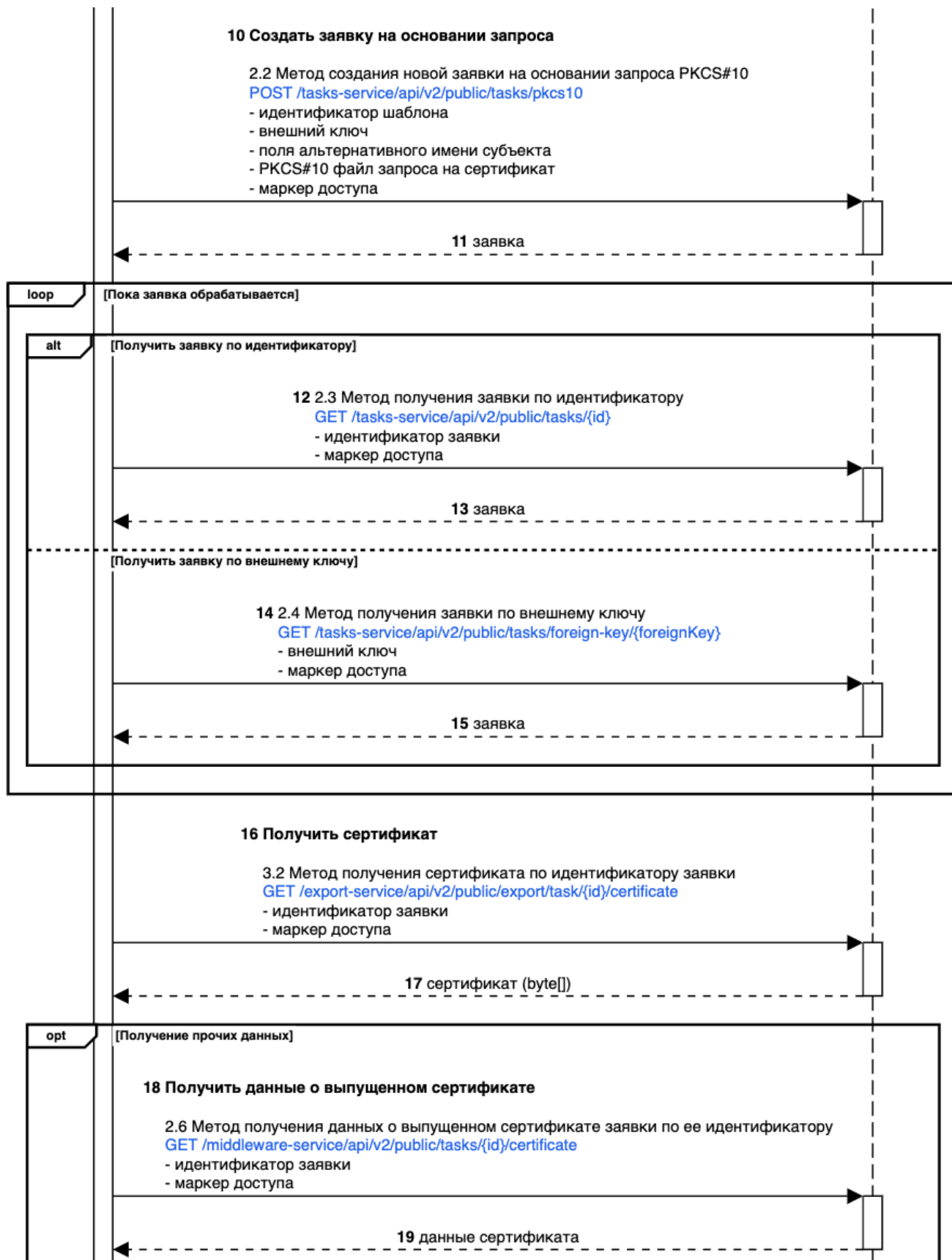
ENCIPHER_ONLY, DECIPHER_ONLY, UNKNOWN)	
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
critical (boolean),	Флаг: расширение критическое
values (string[])	Значение расширения (OIDs)
},	
policies: {	Политики сертификата
critical (boolean),	Флаг: расширение критическое
values (string[])	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
extendable (boolean),	Флаг: расширяемое поле
regex (string),	Регулярное значение для валидации значения
defaultValue (string),	Значение по умолчанию
alert (string),	Предупреждение о неудачной валидации значения
code (enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, ROLE, 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
extendable (boolean),	Флаг: расширяемое поле
regex (string),	Регулярное значение для валидации значения
defaultValue (string),	Значение по умолчанию
alert (string),	Предупреждение о неудачной валидации значения
code (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN),	Код компонента
generalName (int32),	Идентификатор компонента в RFC
oid (string)	OID компонента в RFC
}],	
updated (instant),	Время обновления (ISO 8601)
created (instant),	Время создания (ISO 8601)
certificateAuthorityId (UUID),	ID ЦС, который должен использоваться при выпуске сертификата по данному шаблону

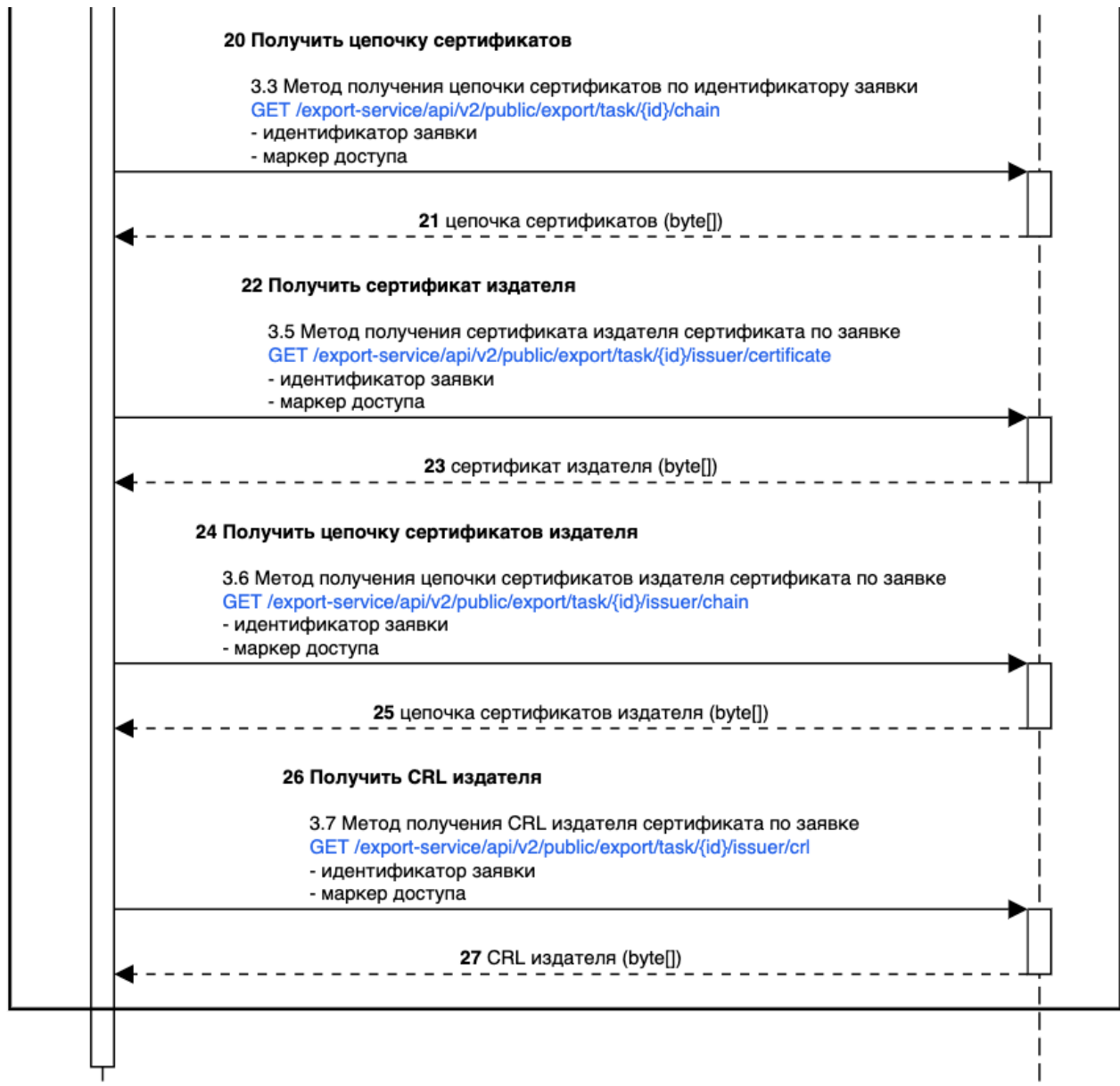
	Шаблоны, у которых в поле «Центр сертификации» указано «Любой», будут иметь в данном поле значение NULL
appendSubjectSid (boolean),	Флаг: включать SID субъекта в сертификат
publication (boolean)	Флаг: публиковать сертификат в РС
}	

3 ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ ПОЛУЧЕНИЯ СЕРТИФИКАТА ПО ЗАПРОСУ PKCS#10

Центр регистрации
Выпуск сертификата по заявке на основании запроса PKCS#10







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

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

- проверить статус заявки, отправив запрос на получение заявки по идентификатору `taskId` (шаги 12-13) или по внешнему ключу `foreignKey` (шаги 14-15); проверять следует периодически, пока заявка не перейдет в один из конечных статусов: Выполнена, Отклонена или Отменена;
- сценарий завершается, если заявка перешла в статус Отклонена или Отменена;
- если заявка перешла в статус Выполнена, то по ней был выпущен сертификат и его следует скачать (шаги 18-19), используя `taskId`;
- опционально по идентификатору заявки `taskId` можно получить следующую информацию:
 - информацию о выпущенном сертификате (шаги 18-19);
 - цепочку сертификатов (шаги 20-21);
 - сертификат издателя (шаги 22-23);
 - цепочку сертификатов издателя (шаги 24-25);
 - и CRL издателя (шаги 26-27).

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ОС	—	Операционная система
ПО	—	Программное обеспечение
ЦР	—	Центр регистрации
ЦС	—	Центр сертификации
API	—	Application Programming Interface
CRL	—	Certificate Revocation List
AIA	—	Authority Information Access
URL	—	Uniform Resource Locator
eCA-CA	—	Программный комплекс «Центр сертификации Aladdin Enterprise Certification Authority»
eCA-VA	—	Программный комплекс «Центр валидации Aladdin Enterprise Validation Authority»
eCA-RA	—	Программный комплекс «Центр регистрации Aladdin Enterprise Registration Authority»

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]