



Центр сертификатов доступа

Aladdin Enterprise Certificate Authority Certified Edition KG

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

Изделие	33714370.03.01.001
Документ	33714370.03.01.001
Версия	2.3.0
Листов	59
Дата	30.05.2025

СОДЕРЖАНИЕ

1 Описание методов REST API версии 2.....	4
1.1 Методы аутентификации	4
1.1.1 Метод аутентификации по сертификату (устаревший)	4
1.1.2 Метод аутентификации по Kerberos-ticket (устаревший)	5
1.1.3 Метод аутентификации по логину и паролю (устаревший)	5
1.1.4 Метод аутентификации по сертификату	6
1.1.5 Метод аутентификации по Kerberos-ticket	6
1.1.6 Метод аутентификации по логину и паролю	6
1.1.7 Метод обновления маркера доступа	7
1.2 Методы работы с заявками	8
1.2.1 Метод создания новой заявки на выпуск сертификата с закрытым ключом (PKCS#12)	8
1.2.2 Метод создания новой заявки на основании запроса PKCS#10	10
1.2.3 Метод получения заявки по идентификатору	14
1.2.4 Метод получения заявки по внешнему ключу	15
1.2.5 Метод поиска заявок	16
1.2.6 Метод получения данных о выпущенном сертификате заявки по ее идентификатору	17
1.3 Методы экспорта файлов	20
1.3.1 Метод получения запроса на сертификат по идентификатору заявки	20
1.3.2 Метод получения сертификата по идентификатору заявки	20
1.3.3 Метод получения цепочки сертификатов по идентификатору заявки	21
1.3.4 Метод получения контейнера pkcs12 по идентификатору заявки	21
1.3.5 3.5 Метод получения сертификата издателя сертификата по заявке	22
1.3.6 Метод получения цепочки сертификатов издателя сертификата по заявке	22
1.3.7 Метод получения CRL издателя сертификата по заявке	23
1.4 Методы работы с субъектами	23
1.4.1 Метод поиска субъектов	23
1.4.2 Метод получения субъекта по идентификатору	25
1.5 Методы работы с шаблонами	26
1.5.1 Метод поиска шаблонов	26
1.5.2 Метод получения шаблона по идентификатору	27
2 Описание методов REST API версии 3.....	30
2.1 Методы работы с заявками	30
2.1.1 Метод создания новой заявки на выпуск сертификата с закрытым ключом (PKCS#12)	30
2.1.2 Метод создания новой заявки на основании запроса PKCS#10	32
2.1.3 Метод получения данных о выпущенном сертификате заявки по ее идентификатору	36
2.2 Методы работы с субъектами	39

2.2.1 Метод поиска субъектов	39
2.2.2 Метод получения субъекта по идентификатору.....	40
2.3 Методы работы с шаблонами.....	42
2.3.1 Метод получения шаблона по идентификатору.....	42
3 Диаграмма последовательности получения сертификата по запросу PKCS#10.....	45
3.1 Диаграмма последовательности получения сертификата по запросу PKCS#10.....	45
4 Спецификация OpenAPI	49
Обозначения и сокращения.....	58

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

Пример использования метода для аутентификации по сертификату доступа:

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

В примере указаны:

`cert.crt.pem` – сертификат;

`test.key.pem` – закрытый ключ.

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

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

Пример ответа на запрос к данному методу:

```
{
  "status": 200,
  "data": {
    "token": "eyJhbGciOiJSUzUxMiJ9.eyJzZXNzaW9uSWQiOiJiYzZwMDQ5ZC1hM2EwLTQ4ZjQtYjE0Ny02YmYzNmRjYjZlZG9iLjE3MDE5ODU2MjYsImV4cCI6MTcwMTk4NTgwNn0.aX7pbvPHtP8tQ0vg9upBR7AlYXbgwr8XX1tlcdggcySzehA6qi2mJGE-MlMf0iNeAOy0L_QJ7YVsB_080yTcw2E6ipstRI1 rRbq1VHKw4xTncVh2UVCXXTxz4WaYw3uSkNjfpPLgs7wyxyAlu3Is4d7JuicSgONoj-rArjam_ccTuEHJnHp7PTnrDLOUNm8elu-VkP9qvQPqLb0P_K7-XLp3lQrWOZWMPPM4imldiA70fBD4QeNiruQjY HK4ENbqCARIMiPnV0GFTz7JMnLLPlliFrnOgHhUUhp6JylKOTKxdLgUjTU03BmucUwAl_ccThZZPzpwNLd6Cx67X00g",
    "refresh": "8b73de98-721d-4d0b-98f7-f19745272f0c"
  }
}
```

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

1.1.2 Метод аутентификации по Kerberos-ticket (устаревший)

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

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

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

1.1.3 Метод аутентификации по логину и паролю (устаревший)

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

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

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

1.1.4 Метод аутентификации по сертификату

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

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

1.1.5 Метод аутентификации по Kerberos-ticket

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

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

1.1.6 Метод аутентификации по логину и паролю

POST API – Аутентификация по логину и паролю	
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	
ResponseEntity -> ItemResponse -> {	Ответ JSON в HTTP-body
token (string),	Маркер доступа
refresh (string)	Токен обновления
}	

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

1.1.7 Метод обновления маркера доступа

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

Пример использования метода:

```
curl --location -k --request POST 'https://172.17.152.213/security-  
service/api/v2/public/auth/refresh-token' --header 'Cookie:  
token=eyJhbGciOiJSUzUxMiJ9.eyJzZXNzaW9uSWQiOiI2YTZhNmM4Yi0zYWQyLTQwYjMtYWI5NS05NDgxNmZlYjBkZGQ  
iLCJpYXQiOiE3MDIwMjEyNzEsImV4cCI6MTcwMjAyMTQ1MX0uLJUCXBCuiClMWFhMVNoYy8peli1YgdDw4eQ-v4Blp5mY  
RcOxS0swGipUGZ_0cTrV3QyYjVXxrtY9CHTM2virdnvU7xp_4lASug5MRUCjOhsWnNfigu21bDoFbK7jQPDDdG5LfCxfswq  
VAXrY-G4BPst8azPdY-CVtsjXIPyUmkL-vo7_7b4Prs2OwNeewYn_9KsdHXV56XJHEo4nBM0wses830Q813pFe7LpL  
XNuQ8SvekEr9QKGSspBgYT3qC9D_57NhWzEgKqVRsLxk4f4p8x7TI4zIsFI1v1VaqwrnAiDnElwxFHNnvXVDuyxh54g6Bk  
LfoHj3DGnCiijopPCTi pw; refresh=7aaad966-b417-418e-b2bb-63574ddcc8d1'
```

Пример ответа на запрос к данному методу:

```
{ "status":200,"data":{"token":"eyJhbGciOiJSUzUxMiJ9.eyJzZXNzaW9uSWQiOiI2YTZhNmM4Yi0zYWQyLTQwYjMtYWI5NS05NDgxNmZlYjBkZGQiLCJpYXQiOiE3MDIwMjEzNzMsImV4cCI6MTcwMjAyMTU1M30.CUO3CaTt7JsIAg8LYMlURx0tetbsPw5iHT33i14JwSAos8DmNfSelylnfEHNlb2U8Q0G1KvJlhm84FqWR45RgBuvsQkgzzR5aDDiVzLhhpkbPGqAUbILmFdiqLmDokyS0SyybUO4-SSfueV8lzo5aF6ly80c2 cWpwRrRQgNhBDdc1LgJACmdD29urmBq1QsFV6AFRdlN6l3EZkP2ragZInQQ1lM9D4e5d3mlugCP7pJ2bp39pOVLWC1czusPfs9tDoIyigkZBkSti7xPYAUdghzBfc--eK1kepoWilv5r4hln4bT6grZ7Yyi-J5JLVycuGOzkle3ByIcBExmFwtQ","refresh":"3071f779-f4f7-47db-91a4-13acfc50005"}}
```

1.2 Методы работы с заявками

Внимание! Технологический Центр сертификации использовать для выпуска сертификатов запрещено.

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	
Для получателей сертификатов с использованием данного метода должно быть доступно только создание заявок. Для администратора или оператора с использованием данного метода должно быть доступно создание заявок для субъектов ресурсных систем из Центра сертификации Aladdin eCA, к которому подключен Центр регистрации Aladdin eRA.	
URL – /tasks-service/api/v2/public/tasks/pkcs12	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20задачи/registerPkcs12	
Query	
{	
certificateAuthorityId (uuid), [опционально]	ID Центра сертификации, на котором должен быть выпущен сертификат по заявке. Если для данного параметра не указано значение, выпуск будет осуществляться на Центре сертификации, 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.
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. ³
}	

¹ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

² Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA 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.

Request	
{	
subjectDN: { (enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[] },	Поля разделенного имени субъекта, которое должно попасть в сертификат. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
subjectAltName: { (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, 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 ⁴
type (enum: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (enum: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ⁵
certificateId (uuid),	Идентификатор сертификата. Заполняется по итогам успешной отработки заявки.
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string)	Внешний ключ заявки
}	

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

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

1.2.2 Метод создания новой заявки на основании запроса PKCS#10

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 получателей сертификатов с использованием данного метода должно быть доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для администраторов и операторов с использованием данного метода должно быть доступно создание заявок для субъектов ресурсных систем из Центра сертификации Aladdin eCA, к которому подключен Центр регистрации Aladdin eRA.</p>	
URL – /tasks-service/api/v2/public/tasks/pkcs10	
Swagger: 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, 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.</p>
subjectId (uuid), [опционально]	<p>ID субъекта⁶.</p> <p>Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь».</p> <p>Может быть указан при использовании метода учетной записью с ролью «Администратор».⁷</p>
templateId (uuid), [обязательно]	ID шаблона сертификата ⁸ .
foreignKey (string) [опционально]	<p>Внешний ключ заявки.</p> <p>Внешний ключ должен быть уникальным.⁹</p>
}	

⁶ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

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

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

⁸ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

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

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.
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 ¹⁰
type (enum: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (enum: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ¹¹
certificateId (string),	Идентификатор сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string)	Внешний ключ заявки
}	

¹⁰ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

¹¹ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

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 получателей сертификатов с использованием данного метода должно быть доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для администраторов и операторов с использованием данного метода должно быть доступно создание заявок для субъектов ресурсных систем из Центра сертификации Aladdin eCA, к которому подключен Центр регистрации Aladdin eRA.</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]	ID Центра сертификации, на котором должен быть выпущен сертификат по заявке. Если для данного параметра не указано значение, выпуск будет осуществляться на Центре сертификации, 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.
subjectId (uuid), [опционально]	ID субъекта ¹² . Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь». Может быть указан при использовании метода учетной записью с ролью «Администратор». ¹³
templateId (uuid), [обязательно]	ID шаблона сертификата ¹⁴ .
foreignKey (string) [опционально]	Внешний ключ заявки. Внешний ключ должен быть уникальным ¹⁵
}	
Request	

¹² Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

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

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

¹⁴ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

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

{	
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 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 ¹⁶
type (enum: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (enum: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ¹⁷
certificateId (string),	Идентификатор сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)

¹⁶ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

¹⁷ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

foreignKey (string)	Внешний ключ заявки
}	

1.2.3 Метод получения заявки по идентификатору

GET API – Получение заявки по идентификатору	
<p>Для получателя сертификатов с использованием данного метода должно быть доступно получение только тех заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для операторов с использованием данного метода должно быть доступно получение заявок, созданных данной учетной записью, а также заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть 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	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ¹⁸
certificateAuthorityId (uuid),	Идентификатор сертификата издателя
type (string: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (string: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ¹⁹
certificateId (uuid),	ID сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string),	Внешний ключ заявки
templateName (string),	Имя шаблона сертификата
msUpn (string)	UPN субъекта
}	

¹⁸ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

¹⁹ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

1.2.4 Метод получения заявки по внешнему ключу

GET API – Получение заявки по внешнему ключу	
<p>Для получателя сертификатов с использованием данного метода должно быть доступно получение только тех заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для операторов с использованием данного метода должно быть доступно получение заявок, созданных данной учетной записью, а также заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть 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	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ²⁰
certificateAuthorityId (uuid),	Идентификатор сертификата издателя
type (string: PKCS10, PKCS12, ON_TOKEN, SCEP, WSTEP, UNKNOWN),	Тип заявки
status (string: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки
subjectId (uuid),	ID субъекта ²¹
certificateId (uuid),	ID сертификата. Заполняется по итогам успешной 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
}	

²⁰ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

²¹ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

1.2.5 Метод поиска заявок

GET API – Поиск заявок	
<p>Для получателя сертификатов в ответе данного метода должны возвращаться только те заявки, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для операторов в ответе данного метода должны возвращаться заявки, созданные данной учетной записью, а также заявки для субъектов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора в ответе данного метода должны возвращаться 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	
Query	
{	
sortDirection (string),	Направления сортировки (ASC;DESC)
sortBy (string[]),	Список полей, к которым применяется сортировка
pageOffset (integer),	Смещение от начала списка (пагинация)
pageLimit (integer),	Ограничение на размер выборки (пагинация)
search (string),	Полнотекстовый поиск
foreignKeys (string[]),	Фильтр: список внешних ключей
templateIds (uuid[]),	Фильтр: список ID шаблонов
creatorIds (uuid[]),	Фильтр: список ID УЗ для фильтрации
types (enum[]: PKCS10, PKCS12, SCEP, WSTEP, ON_TOKEN, UNKNOWN),	Фильтр: список типов заявки
statuses (enum[]: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN)	Фильтр: список статусов заявки
certificateAuthorityId (string[]),	Фильтр: список ID Центров сертификации
serialnumber (string),	Поиск заявки по серийному номеру сертификата
fingerprint (string[])	Фильтр: список отпечатков сертификатов
subjectKeyIdentifier (string[])	Фильтр: Идентификатор ключа субъекта
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (uuid),	ID заявки
templateId (uuid),	ID шаблона сертификата ²²
certificateAuthorityId (uuid),	Идентификатор сертификата издателя
type (string: PKCS10, PKCS12, SCEP, WSTEP, ON_TOKEN, UNKNOWN),	Тип заявки
status (string: NEW, FOR_APPROVE, ISSUE_ERROR, REJECTED, AWAITING_IMPORT, COMPLETED, CANCELLED, UNKNOWN),	Статус заявки
creatorId (uuid),	ID учетной записи автора заявки

²² Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

subjectId (uuid),	ID субъекта ²³
certificateId (uuid),	ID сертификата. Заполняется по итогам успешной отработки заявки
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string),	Внешний ключ заявки
templateName (string),	Имя шаблона сертификата
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 получателя сертификатов с использованием данного метода должно быть доступно получение данных о выпущенных сертификатах только по тем заявкам, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для оператора с использованием данного метода должно быть доступно получение данных о выпущенных сертификатах только по созданным данным оператором заявкам, а также по заявкам для субъектов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть доступно получение данных о выпущенных сертификатах по любым заявкам.</p> <p>В ответе данного метода в полях «subjectDN» и «issuerDN» компоненты «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
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,	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра

²³ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]	
},	
issuer: {...}	Издатель сертификата (вложенный объект)
},	
serialnumber (string),	Серийный номер сертификата
fingerprint (string),	Фингерпринт сертификата
name (string),	Имя сертификата (на основе CN)
issuerId (uuid),	Идентификатор издателя сертификата
issuerDN: {	Имя субъекта издателя сертификата
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectDN: {	Имя субъекта сертификата
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectAltName: {	Альтернативное имя субъекта сертификата
(enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[]	Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата ЦС
hashAlgorithm (enum: SHA1, SHA256, SHA384, SHA512, SHA3_256, SHA3_384, SHA3_512, RSASSA_PSS, MD5, GOST_R_34_11_2012, UNKNOWN),	Алгоритм подписи сертификата ЦС
keyBits (int32),	Длина ключа сертификата ЦС
templateId (uuid),	Идентификатор шаблона ²⁴
templateName (string),	Имя шаблона
type (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип сертификата
validFrom (instant),	Дата начала действия сертификата (ISO 8601)
validTo (instant),	Дата окончания действия сертификата (ISO 8601)

²⁴ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

status (enum: ACTIVE, HOLD, REVOKE, REQUEST, UNKNOWN),	Статус сертификата
actions: {	Доступные действия по выгрузке
p12 (boolean),	Флаг: выгрузка pkcs12
csr (boolean),	Флаг: выгрузка pkcs10
pem (boolean)	Флаг: выгрузка сертификата
},	
publicKey (string),	Открытый ключ
certificateType (string),	Тип сертификата (X.509)
version (int32),	Версия сертификата
subjectKeyIdentifier (string),	Идентификатор ключа сертификата
authorityKeyIdentifier (string)	Идентификатор ключа издателя сертификата
keyUsages: {	Назначение ключа сертификата
id (enum: DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCRYPTMENT, DATA_ENCRYPTMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY, UNKNOWN),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (enum: EKU_PKIX_ANY_EXTENDED_KEY_USAGE, CSN_369791_TLS_CLIENT, CSN_369791_TLS_SERVER, CLIENT_AUTHENTICATION, CODE_SIGNING, EAP_OVER_LAN, EAP_OVER_PPP, ETSI_TSL_SIGNING, EMAIL_PROTECTION, ICAO_DEVIATION_LIST_SIGNING, EKU_INTEL_AMT, INTERNET_KEY_EXCHANGE_FOR_IPSEC, KERBEROS_CLIENT_AUTHENTICATION, EKU_KRB_PKINIT_KDC, MS_COMMERCIAL_CODE_SIGNING, MS_DOCUMENT_SIGNING, MS_EFS_RECOVERY, MS_ENCRYPTED_FILE_SYSTEM, MS_INDIVIDUAL_CODE_SIGNING, MS_SMART_CARD_LOGON, OCSP_SIGNER, EKU_ADOBE_PDF_SIGNING, PIV_CARD_AUTHENTICATION, SCVP_CLIENT, SCVP_SERVER, SIP_DOMAIN, EKU_PKIX_SSH_CLIENT, SSH_SERVER, SERVER_AUTHENTICATION, TIME_STAMPING, ICAO_MASTER_LIST_SIGNING, UNKNOWN),	Перечисление расширенного использования ключа
value (string),	Наименование элемента
oid (string),	OID назначения
description (string)	Описание использования ключа
},	Описание OID
ca (boolean),	Флаг: сертификат ЦС
aiaUrls (string[]),	URL AIA
ocspUrls (string[]),	URL OCSP
crlUrls (string[]),	URL CRL
deltaCrlUrls (string[]),	URL Delta CRL
}	

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 получателя сертификатов с использованием данного метода должно быть доступно получение запроса на сертификат только из тех заявок на сертификат на основании запроса, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для Оператора с использованием данного метода должно быть доступно получение запроса на сертификат только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа. Для Администратора с использованием данного метода должно быть доступно получение запроса на сертификат из любых заявок на сертификат на основании запроса. Для успешного получения запроса на сертификат из заявки она должна иметь тип «PKCS10».</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 получателя сертификатов с использованием данного метода должно быть доступно получение сертификатов только из тех заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для оператора с использованием данного метода должно быть доступно получение сертификата только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть доступно получение сертификатов из любых заявок. 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 получателя сертификатов с использованием данного метода должно быть доступно получение цепочки сертификатов только из тех заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для оператора с использованием данного метода должно быть доступно получение цепочки сертификатов только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть доступно получение цепочки сертификатов из любых заявок. Для успешного получения цепочки сертификатов из заявки она должна иметь статус «COMPLETED».</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 получателя сертификатов с использованием данного метода должно быть доступно получение контейнера pkcs12 только из тех заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для оператора с использованием данного метода должно быть доступно получение контейнера pkcs12 только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть доступно получение контейнера pkcs12 из любых заявок. Для успешного получения контейнера pkcs12 из заявки она должна иметь статус «COMPLETED» и тип «PKCS12».</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 3.5 Метод получения сертификата издателя сертификата по заявке

GET API – Получение сертификата издателя сертификата по заявке	
<p>Для получателя сертификатов с использованием данного метода должно быть доступно получение сертификата издателя только из тех заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для оператора с использованием данного метода должно быть доступно получение сертификата издателя только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть доступно получение сертификата издателя из любых заявок. Для успешного получения сертификата издателя из заявки она должна иметь статус «COMPLETED».</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 получателя сертификатов с использованием данного метода должно быть доступно получение цепочки сертификатов издателя только из тех заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для оператора с использованием данного метода должно быть доступно получение цепочки сертификатов издателя только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть доступно получение цепочки сертификатов издателя из любых заявок. Для успешного получения цепочки сертификатов издателя из заявки она должна иметь статус «COMPLETED».</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	
<p>Для получателя сертификатов с использованием данного метода должно быть доступно получение CRL издателя только из тех заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для оператора с использованием данного метода должно быть доступно CRL издателя только из созданных данным оператором заявок, а также из заявок для субъектов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть доступно получение CRL издателя из любых заявок. Для успешного получения CRL издателя из заявки она должна иметь статус «COMPLETED».</p>	
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	
<p>Для получателя сертификатов с использованием данного метода должно быть доступно только получение данных субъекта, связанный с данной учетной записью. Для оператора или администратора с использованием данного метода должно быть доступно получение данных любого субъекта.</p> <p>В ответе данного метода в поле «subjectName» атрибуты «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.</p>	
URL – /middleware-service/api/v2/public/subjects	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20Задачи/findById	
Query	
{	
id (UUID[]), [опционально]	ID субъекта ²⁵
resourceId (UUID[]), [опционально]	ID ресурсной системы
securityGroupId (UUID[]), [опционально]	ID группы безопасности
search (string), [опционально]	Полнотекстовый поиск (имя субъекта)
isConnected (boolean), [опционально]	Флаг: субъект подключен к ресурсной системе
isBlocked (boolean), [опционально]	Флаг: субъект заблокирован в ресурсной системе

²⁵ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

pageLimit (integer), [опционально]	Ограничение на размер выборки (пагинация)
pageOffset (integer), [опционально]	Смещение от начала списка (пагинация)
sortBy (string[]), [опционально]	Список полей, к которым применяется сортировка
sortDirection (string) [опционально]	Направления сортировки (ASC;DESC)
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID субъекта ²⁶
commonName (string),	Имя субъекта
distinguishedName (string),	Расположение субъекта в ресурсной системе
resource: {	Ресурсная система
id (uuid),	ID ресурсной системы
commonName (string),	Имя ресурсной системы
distinguishedName (string)	BaseDN точки подключения к ресурсной системе
},	
subjectName: {	Имя субъекта
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, 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)

²⁶ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

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	
<p>Для получателя сертификатов с использованием данного метода должно быть доступно только получение данных субъекта, связанный с данной учетной записью. Для оператора или администратора с использованием данного метода должно быть доступно получение данных любого субъекта.</p> <p>В ответе данного метода в поле «subjectName» атрибуты «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.</p>	
URL – /middleware-service/api/v2/public/subjects/{id}	
Swagger: : https://HOST/external-integration-service/swagger/swagger-ui/index.html#/Контроллер%3A%20субъекты/findById_2	
Query	
{	
id (UUID)	ID субъекта ²⁷
}	
Request	
-	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID субъекта ²⁸
commonName (string),	Имя субъекта
distinguishedName (string),	Расположение субъекта в ресурсной системе
resource: {	Ресурсная система
id (uuid),	ID ресурсной системы
commonName (string),	Имя ресурсной системы
distinguishedName (string)	BaseDN точки подключения к ресурсной системе
},	
subjectName: {	Имя субъекта
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): {	Поля разделенного имени субъекта
values (string[]),	Значения компонента
editable (boolean),	Флаг: компонент доступен для редактирования
}	
},	
subjectAltName: {	Альтернативное имя субъекта
(enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID,	Поля альтернативного имени субъекта

²⁷ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

²⁸ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): {	
values (string[]),	Значения компонента
editable (boolean),	Флаг: компонент доступен для редактирования
}	
},	
isConnected (boolean),	Флаг: субъект подключен к ресурсной системе
isBlocked (boolean),	Флаг: субъект заблокирован в ресурсной системе
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)
}	

1.5 Методы работы с шаблонами

Внимание! Технологический Центр сертификации использовать для выпуска сертификатов запрещено.

1.5.1 Метод поиска шаблонов

GET API – Поиск шаблонов	
<p>Для получателя сертификатов с использованием данного метода должно быть доступно получение только тех шаблонов, которые указаны в правилах выпуска с режимом обработки «Автоматический выпуск» или «Ручная обработка» для ассоциированного с текущей учетной записью субъекта. Для оператора с использованием данного метода должно быть доступно получение только тех шаблонов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть 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 ²⁹
}	
Request	
-	
Response	
ResponseEntity -> CollectionResponse -> {	Ответ JSON в HTTP-body
id (UUID),	ID шаблона ³⁰
name (string),	Имя шаблона
type (enum: EMBEDDED, CLONED, IMPORTED, UNKNOWN),	Тип шаблона
certificateType (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип выпускаемого сертификата
certificateAuthorityId (UUID),	ID ЦС, который должен использоваться при выпуске сертификата по данному шаблону. Шаблоны, у которых в поле «Центр сертификации» указано «Любой», будут иметь в данном поле значение NULL
certificateCount (int64),	Число выпущенных по шаблону сертификатов
removed (boolean),	Флаг: шаблон удален
updated (instant),	Время обновления (ISO 8601)
created (instant)	Время создания (ISO 8601)
}	

1.5.2 Метод получения шаблона по идентификатору

GET API – Получение шаблона по идентификатору	
<p>Для получателя сертификатов с использованием данного метода должно быть доступно получение только тех шаблонов, которые указаны в правилах выпуска с режимом обработки «Автоматический выпуск» или «Ручная обработка» для ассоциированного с текущей учетной записью субъекта. Для оператора с использованием данного метода должно быть доступно получение только тех шаблонов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть доступно получение любых шаблонов.</p> <p>В ответе данного метода в поле «subjectDN» компоненты «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 ³¹
}	
Request	

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

³⁰ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

³¹ Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

-	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID шаблона ³²
name (string),	Имя шаблона
validity (int64),	Время действия выпускаемого сертификата (мс)
certificateType (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип выпускаемого сертификата
type (enum: EMBEDDED, CLONED, IMPORTED, UNKNOWN),	Тип шаблона
certificateAuthorityId (UUID),	ID ЦС, который должен использоваться при выпуске сертификата по данному шаблону. Шаблоны, у которых в поле «Центр сертификации» указано «Любой», будут иметь в данном поле значение NULL
removed (boolean),	Флаг: шаблон удален
rsa: {	Описание RSA-криптографии
use (boolean),	Флаг: RSA-ключи доступны для шаблона
minLength (int32),	Минимальная длина RSA-ключа
lengths (int32[])	Доступные длины RSA-ключа
},	
ecdsa: {	Описание ESDCA-криптографии
use (boolean),	Флаг: ESDCA -ключи доступны для шаблона
minLength (int32),	Минимальная длина ESDCA -ключа
lengths (int32[])	Доступные длины ESDCA -ключа
},	
gost: {	Описание ГОСТ-криптографии
use (boolean),	Флаг: ГОСТ -ключи доступны для шаблона
minLength (int32),	Минимальная длина ГОСТ -ключа
lengths (int32[])	Доступные длины ГОСТ -ключа
},	
keyUsages: {	Назначение ключа сертификата
critical (boolean),	Флаг: расширение критическое
values (enum[:DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCIPHERMENT, DATA_ENCIPHERMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCIPHER_ONLY, DECIPHER_ONLY, UNKNOWN])	Значение расширения
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
critical (boolean),	Флаг: расширение критическое
values (string[])	Значение расширения (OIDs)
},	
policies: {	Политики сертификата
critical (boolean),	Флаг: расширение критическое
values (string[])	Значение расширения (OIDs)

³² Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

<code>},</code>	
<code>subjectDN: [{</code>	Имя субъекта сертификата
<code>index (int32),</code>	Индекс (для сортировки, по умолчанию – 0)
<code>name (string),</code>	Имя компонента
<code>description (string),</code>	Описание компонента
<code>required (boolean),</code>	Флаг: обязателен к заполнению
<code>validation (boolean),</code>	Флаг: валидация значения
<code>modifiable (boolean),</code>	Флаг: доступен к редактированию
<code>extendable (boolean),</code>	Флаг: расширяемое поле
<code>regex (string),</code>	Регулярное значение для валидации значения
<code>defaultValue (string),</code>	Значение по умолчанию
<code>alert (string),</code>	Предупреждение о неудачной валидации значения
<code>code (enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN)</code>	Код компонента
<code>}],</code>	
<code>subjectAltName: [{</code>	Расширенное имя субъекта сертификата
<code>index (int32),</code>	Индекс (для сортировки, по умолчанию – 0)
<code>name (string),</code>	Имя компонента
<code>description (string),</code>	Описание компонента
<code>required (boolean),</code>	Флаг: обязателен к заполнению
<code>validation (boolean),</code>	Флаг: валидация значения
<code>modifiable (boolean),</code>	Флаг: доступен к редактированию
<code>regex (string),</code>	Регулярное значение для валидации значения
<code>defaultValue (string),</code>	Значение по умолчанию
<code>alert (string),</code>	Предупреждение о неудачной валидации значения
<code>code (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN),</code>	Код компонента
<code>generalName (int32),</code>	Идентификатор компонента в RFC
<code>oid (string)</code>	OID компонента в RFC
<code>}],</code>	
<code>updated (instant),</code>	Время обновления (ISO 8601)
<code>created (instant)</code>	Время создания (ISO 8601)
<code>}</code>	

2 ОПИСАНИЕ МЕТОДОВ REST API ВЕРСИИ 3

2.1 Методы работы с заявками

Внимание! Технологический Центр сертификации использовать для выпуска сертификатов запрещено.

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	
Для получателей сертификатов с использованием данного метода должно быть доступно только создание заявок. Для администратора или оператора с использованием данного метода должно быть доступно создание заявок для субъектов ресурсных систем из Центра сертификации Aladdin eCA, к которому подключен Центр регистрации Aladdin eRA.	
URL – /tasks-service/api/v2/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, 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.
subjectId (uuid), [опционально]	ID субъекта ³³ . Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь».
templateId (uuid), [обязательно]	ID шаблона сертификата ³⁴ .
foreignKey (string) [опционально]	Внешний ключ заявки. Внешний ключ должен быть уникальным. ³⁵
}	

³³ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

³⁴ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

³⁵ 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.

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, UNKNOWN): string[] },	Поля разделенного имени субъекта, которое должно попасть в сертификат. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
subjectAltName: { (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, 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 ³⁶
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 ³⁷
certificateId (uuid),	Идентификатор сертификата. Заполняется по итогам успешной отработки заявки.
name (string),	Common Name сертификата
updated (instant),	Время обновления заявки (ISO 8601)
created (instant),	Время создания заявки (ISO 8601)
foreignKey (string)	Внешний ключ заявки
}	

³⁶ Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

³⁷ Субъекты eCA-CA и eCA-RA имеют сквозную идентификацию.

2.1.2 Метод создания новой заявки на основании запроса PKCS#10

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 получателей сертификатов с использованием данного метода должно быть доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для администраторов и операторов с использованием данного метода должно быть доступно создание заявок для субъектов ресурсных систем из Центра сертификации Aladdin eCA, к которому подключен Центр регистрации Aladdin eRA.</p>	
URL – /tasks-service/api/v2/public/tasks/pkcs10	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/%5BAPI%20V3%5D%20Контроллер%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, 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.</p>
subjectId (uuid), [опционально]	<p>ID субъекта³⁸.</p> <p>Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь».</p> <p>Может быть указан при использовании метода учетной записью с ролью «Администратор».³⁹</p>
templateId (uuid), [обязательно]	ID шаблона сертификата ⁴⁰ .
foreignKey (string) [опционально]	<p>Внешний ключ заявки.</p> <p>Внешний ключ должен быть уникальным.⁴¹</p>
}	

³⁸ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

³⁹ 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.

⁴⁰ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

⁴¹ 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.

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, DATEOFBIRTH, PLACEOFBIRTH, UNKNOWN): string[] }, [опционально]	Поля отличительного имени субъекта, которое должно попасть в сертификат. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра. Значения полей, указанные в subjectName, переопределяют значения соответствующих полей SDN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.
subjectAltName: { (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[] }	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 ⁴²
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 ⁴³
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
}	

⁴² Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

⁴³ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

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 получателей сертификатов с использованием данного метода должно быть доступно только создание заявок, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для администраторов и операторов с использованием данного метода должно быть доступно создание заявок для субъектов ресурсных систем из Центра сертификации Aladdin eCA, к которому подключен Центр регистрации Aladdin eRA.</p>	
URL – /tasks-service/api/v2/public/tasks/pkcs10	
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, 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.
subjectId (uuid), [опционально]	ID субъекта ⁴⁴ . Должен отсутствовать при использовании метода учетной записью с ролью «Пользователь». Может быть указан при использовании метода учетной записью с ролью «Администратор». ⁴⁵
templateId (uuid), [обязательно]	ID шаблона сертификата ⁴⁶ .
foreignKey (string) [опционально]	Внешний ключ заявки. Внешний ключ должен быть уникальным ⁴⁷

⁴⁴ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

⁴⁵ 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.

⁴⁶ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

⁴⁷ 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.

}	
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, DATEOFBIRTH, PLACEOFBIRTH, UNKNOWN): string[] }, [опционально]	Поля отличительного имени субъекта, которое должно попасть в сертификат. В формате key-value. Где key - один из перечисленных в enum параметров, а value - значение параметра. Значения полей, указанные в subjectName, переопределяют значения соответствующих полей SDN запроса на сертификат, при условии, что они соответствуют значениям атрибутов субъекта.
subjectAltName: { (enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[] }	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 ⁴⁸
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 ⁴⁹
certificateId (string),	Идентификатор сертификата. Заполняется по итогам успешной отработки заявки

⁴⁸ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

⁴⁹ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

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 о выпущенном сертификате заявки по ее идентификатору

GET API – Получение данных о выпущенном сертификате заявки по ее идентификатору	
<p>Для получателя сертификатов с использованием данного метода должно быть доступно получение данных о выпущенных сертификатах только по тем заявкам, получателем сертификата по которым является субъект, связанный с данной учетной записью. Для оператора с использованием данного метода должно быть доступно получение данных о выпущенных сертификатах только по созданным данным оператором заявкам, а также по заявкам для субъектов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть доступно получение данных о выпущенных сертификатах по любым заявкам.</p>	
URL – /middleware-service/api/v2/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, UNKNOWN): string[]	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
issuer: {...}	Издатель сертификата (вложенный объект)
},	
serialnumber (string),	Серийный номер сертификата
fingerprint (string),	Фингерпринт сертификата
name (string),	Имя сертификата (на основе CN)
issuerId (uuid),	Идентификатор издателя сертификата

issuerDN: {	Имя субъекта издателя сертификата
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, UNKNOWN): string[]	Поля разделенного имени субъекта издателя из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectDN: {	Имя субъекта сертификата
(enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, UNKNOWN): string[]	Поля разделенного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
subjectAltName: {	Альтернативное имя субъекта сертификата
(enum: RFC822NAME, DNS_NAME, IPADDRESS, DIRECTORY_NAME, UNIFORM_RESOURCE_ID, REGISTERED_ID, MS_UPN, MS_GUID, KRB5PRINCIPAL, PERMANENT_IDENTIFIER, XMPP_ADDR, SRV_NAME, SUBJECT_IDENTIFICATION_METHOD, UNKNOWN): string[]	Поля альтернативного имени субъекта из сертификата. В формате key-value. Где key – один из перечисленных в enum параметров, а value – значение параметра
},	
keyAlgorithm (enum: RSA, ECDSA, GOST_R_34_10_2012, UNKNOWN),	Алгоритм ключа сертификата ЦС
hashAlgorithm (enum: SHA1, SHA256, SHA384, SHA512, SHA3_256, SHA3_384, SHA3_512, RSASSA_PSS, MD5, GOST_R_34_11_2012, UNKNOWN),	Алгоритм подписи сертификата ЦС
keyBits (int32),	Длина ключа сертификата ЦС
templateId (uuid),	Идентификатор шаблона ⁵⁰
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

⁵⁰ Шаблоны Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

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_ENCRYPTMENT, DATA_ENCRYPTMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCRYPT_ONLY, DECRYPT_ONLY, UNKNOWN),	Перечисление использования ключа
value (string),	Наименование элемента
description (string)	Описание использования ключа
},	
extendedKeyUsages: {	Расширенное назначение ключа сертификата
id (enum: ECU_PKIX_ANY_EXTENDED_KEY_USAGE, CSN_369791_TLS_CLIENT, CSN_369791_TLS_SERVER, CLIENT_AUTHENTICATION, CODE_SIGNING, EAP_OVER_LAN, EAP_OVER_PPP, ETSI_TSL_SIGNING, EMAIL_PROTECTION, ICAO_DEVIATION_LIST_SIGNING, ECU_INTEL_AMT, INTERNET_KEY_EXCHANGE_FOR_IPSEC, KERBEROS_CLIENT_AUTHENTICATION, ECU_KRB_PKINIT_KDC, MS_COMMERCIAL_CODE_SIGNING, MS_DOCUMENT_SIGNING, MS_EFS_RECOVERY, MS_ENCRYPTED_FILE_SYSTEM, MS_INDIVIDUAL_CODE_SIGNING, MS_SMART_CARD_LOGON, OCSP_SIGNER, ECU_ADOBE_PDF_SIGNING, PIV_CARD_AUTHENTICATION, SCVP_CLIENT, SCVP_SERVER, SIP_DOMAIN, ECU_PKIX_SSH_CLIENT, SSH_SERVER, SERVER_AUTHENTICATION, TIME_STAMPING, ICAO_MASTER_LIST_SIGNING, UNKNOWN),	Перечисление расширенного использования ключа
value (string),	Наименование элемента
oid (string),	OID назначения
description (string)	Описание использования ключа
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 получателя сертификатов с использованием данного метода должно быть доступно только получение данных субъекта, связанный с данной учетной записью. Для оператора или администратора с использованием данного метода должно быть доступно получение данных любого субъекта.	
URL – /middleware-service/api/v2/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 ⁵¹
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 ⁵²
commonName (string),	Имя субъекта
distinguishedName (string),	Расположение субъекта в ресурсной системе
resource: {	Ресурсная система
id (uuid),	ID ресурсной системы
commonName (string),	Имя ресурсной системы

⁵¹ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

⁵² Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

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, 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)
sid (string)	SID субъекта
}	

2.2.2 Метод получения субъекта по идентификатору

GET API – Получение субъекта по идентификатору	
Для получателя сертификатов с использованием данного метода должно быть доступно только получение данных субъекта, связанный с данной учетной записью. Для оператора или администратора с использованием данного метода должно быть доступно получение данных любого субъекта.	
URL – /middleware-service/api/v2/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 ⁵³

⁵³ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

}	
Request	
-	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID субъекта ⁵⁴
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, 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)
sid (string)	SID субъекта
}	

⁵⁴ Субъекты Центра сертификации Aladdin eCA и Центра регистрации Aladdin eRA имеют сквозную идентификацию.

2.3 Методы работы с шаблонами

Внимание! Технологический Центр сертификации использовать для выпуска сертификатов запрещено.

2.3.1 Метод получения шаблона по идентификатору

GET API – Получение шаблона по идентификатору	
Для получателя сертификатов с использованием данного метода должно быть доступно получение только тех шаблонов, которые указаны в правилах выпуска с режимом обработки «Автоматический выпуск» или «Ручная обработка» для ассоциированного с текущей учетной записью субъекта. Для оператора с использованием данного метода должно быть доступно получение только тех шаблонов, доступ к которым данному оператору предоставлен по правилам доступа. Для администратора с использованием данного метода должно быть доступно получение любых шаблонов.	
URL – /middleware-service/api/v2/public/templates/{id}	
Swagger: https://HOST/external-integration-service/swagger/swagger-ui/index.html#/5BAPI%20V3%5D%20Контроллер%3A%20шаблоны/findById_1	
Query	
{	
templateId (UUID)	ID шаблона ⁵⁵
}	
Request	
-	
Response	
{	Ответ JSON в HTTP-body
id (UUID),	ID шаблона ⁵⁶
name (string),	Имя шаблона
validity (int64),	Время действия выпускаемого сертификата (мс)
certificateType (enum: CERTIFICATE, ROOT_CA, SUB_CA, UNKNOWN),	Тип выпускаемого сертификата
type (enum: EMBEDDED, CLONED, IMPORTED, UNKNOWN),	Тип шаблона
certificateAuthorityId (UUID),	ID ЦС, который должен использоваться при выпуске сертификата по данному шаблону. Шаблоны, у которых в поле «Центр сертификации» указано «Любой», будут иметь в данном поле значение NULL
removed (boolean),	Флаг: шаблон удален
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

⁵⁵ Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

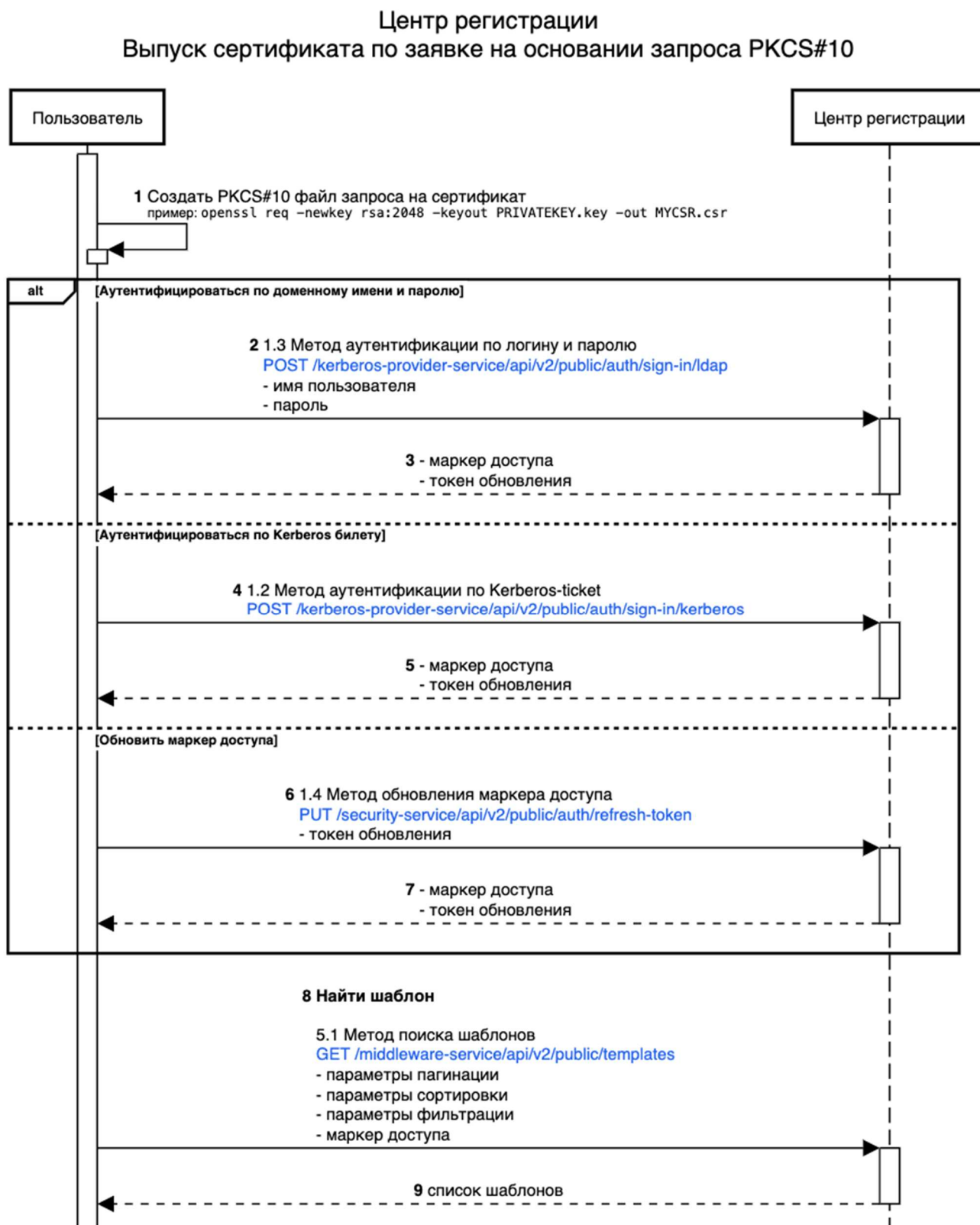
⁵⁶ Шаблоны eCA-CA и eCA-RA имеют сквозную идентификацию.

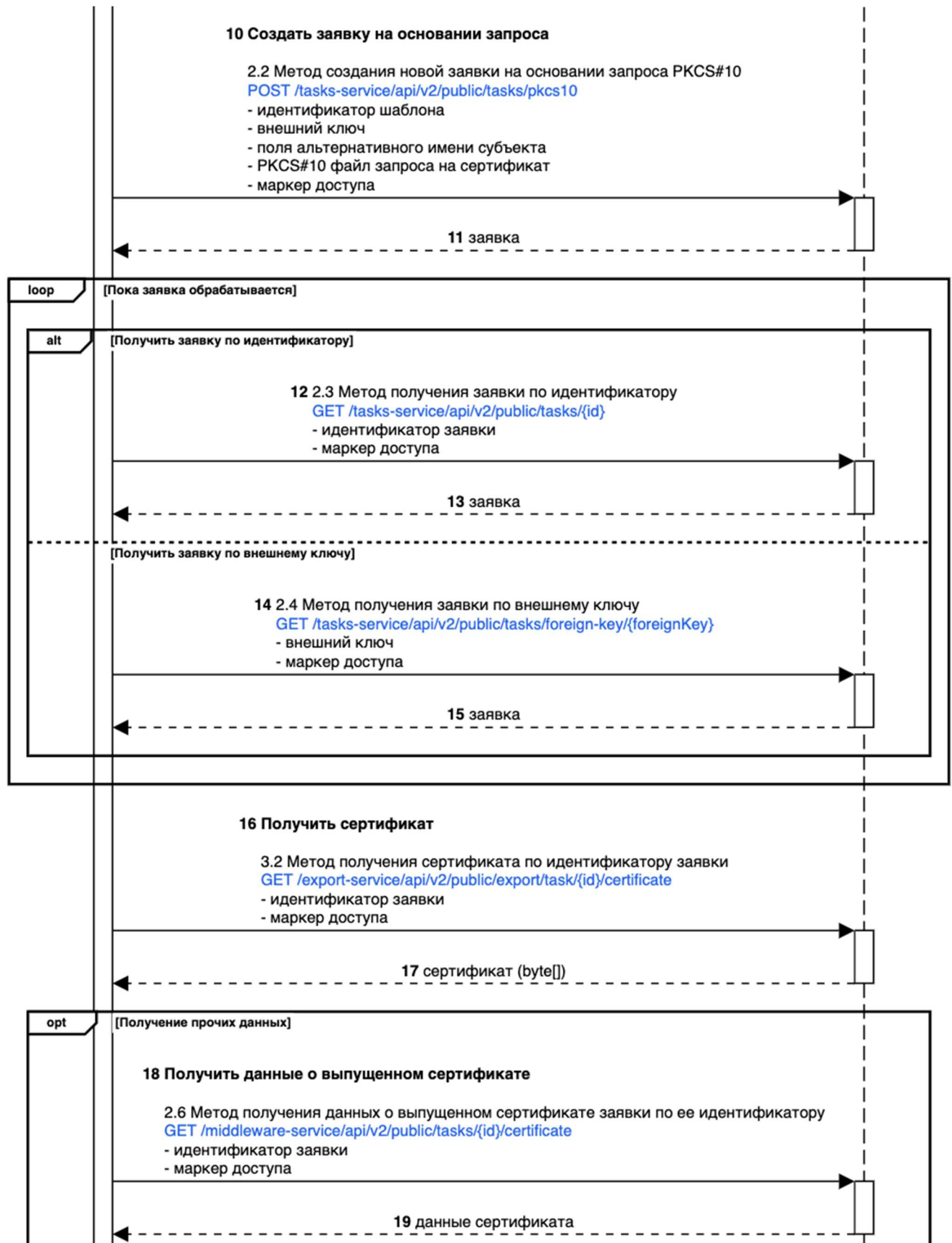
<code>},</code>	
<code>gost: {</code>	Описание ГОСТ-криптографии
<code>use (boolean),</code>	Флаг: ГОСТ -ключи доступны для шаблона
<code>minLength (int32),</code>	Минимальная длина ГОСТ -ключа
<code>lengths (int32[])</code>	Доступные длины ГОСТ -ключа
<code>},</code>	
<code>keyUsages: {</code>	Назначение ключа сертификата
<code>critical (boolean),</code>	Флаг: расширение критическое
<code>values (enum[:DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCIPHERMENT, DATA_ENCIPHERMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, ENCIPHER_ONLY, DECIPHER_ONLY, UNKNOWN])</code>	Значение расширения
<code>},</code>	
<code>extendedKeyUsages: {</code>	Расширенное назначение ключа сертификата
<code>critical (boolean),</code>	Флаг: расширение критическое
<code>values (string[])</code>	Значение расширения (OIDs)
<code>},</code>	
<code>policies: {</code>	Политики сертификата
<code>critical (boolean),</code>	Флаг: расширение критическое
<code>values (string[])</code>	Значение расширения (OIDs)
<code>},</code>	
<code>subjectDN: [{</code>	Имя субъекта сертификата
<code>index (int32),</code>	Индекс (для сортировки, по умолчанию - 0)
<code>name (string),</code>	Имя компонента
<code>description (string),</code>	Описание компонента
<code>required (boolean),</code>	Флаг: обязателен к заполнению
<code>validation (boolean),</code>	Флаг: валидация значения
<code>modifiable (boolean),</code>	Флаг: доступен к редактированию
<code>extendable (boolean),</code>	Флаг: расширяемое поле
<code>extendable (boolean),</code>	Флаг: расширяемое поле
<code>regex (string),</code>	Регулярное значение для валидации значения
<code>defaultValue (string),</code>	Значение по умолчанию
<code>alert (string),</code>	Предупреждение о неудачной валидации значения
<code>code (enum: CN, UID, E, EMAILADDRESS, MAIL, SN, GIVENNAME, INITIALS, SURNAME, OU, O, L, ST, DC, C, UNSTRUCTUREDADDRESS, UNSTRUCTUREDNAME, POSTALCODE, BUSINESSCATEGORY, TELEPHONENUMBER, PSEUDONYM, POSTALADDRESS, STREET, NAME, T, DN, DESCRIPTION, INN, OGRN, OGRNIP, SNILS, INNLE, DATEOFBIRTH, PLACEOFBIRTH, UNKNOWN)</code>	Код компонента
<code>}},</code>	
<code>subjectAltName: [{</code>	Расширенное имя субъекта сертификата
<code>index (int32),</code>	Индекс (для сортировки, по умолчанию - 0)
<code>name (string),</code>	Имя компонента
<code>description (string),</code>	Описание компонента
<code>required (boolean),</code>	Флаг: обязателен к заполнению

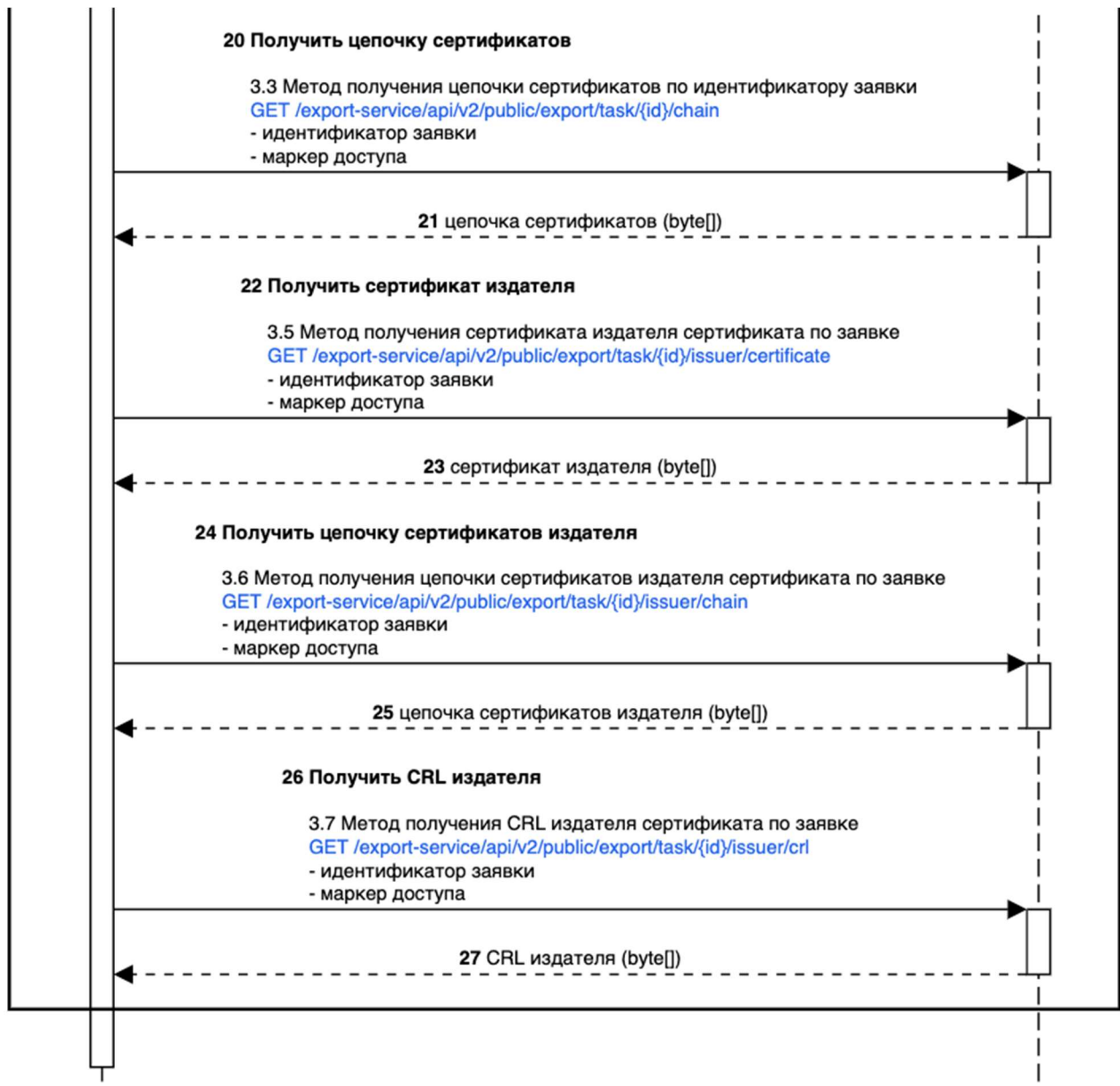
validation (boolean),	Флаг: валидация значения
modifiable (boolean),	Флаг: доступен к редактированию
extendable (boolean),	Флаг: расширяемое поле
regex (string),	Регулярное значение для валидации значения
defaultValue (string),	Значение по умолчанию
alert (string),	Предупреждение о неудачной валидации 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

3.1 Диаграмма последовательности получения сертификата по запросу PKCS#10







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

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

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

4 СПЕЦИФИКАЦИЯ ОРЕНАРИ

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "AECA RA external integration service API",
    "description": "API to AECA RA external integration service",
    "termsOfService": "http://example.com/terms/",
    "contact": {
      "name": "Product Service API Team",
      "url": "http://example.com/team",
      "email": "apiteam@example.com"
    },
    "license": {
      "name": "Apache 2.0",
      "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
    },
    "version": "2.0.0",
    "servers": [
      {
        "url": "",
        "description": "Service URL"
      }
    ],
    "security": [
      {
        "X-API-KEY": []
      },
      {
        "token": []
      },
      {
        "X-AECA-Principal-ID": []
      },
      {
        "X-AECA-Is-Account": []
      },
      {
        "X-AECA-Principal-Name": []
      },
      {
        "X-AECA-User-Role": []
      },
      {
        "X-AECA-User-Authorities": []
      },
      {
        "X-User-Session-Id": []
      },
      {
        "tags": [
          {
            "name": "Контроллер: шаблоны",
            "description": "REST API: шаблоны"
          },
          {
            "name": "Контроллер: Авторизации",
            "description": "Интерфейс описания Api: Авторизации"
          },
          {
            "name": "Контроллер: экспорта",
            "description": "API методов по работе с экспорта"
          },
          {
            "name": "Контроллер: Задачи",
            "description": "REST API: Задачи"
          },
          {
            "name": "Контроллер: Авторизации",
            "description": "Интерфейс описания Api: Авторизации"
          },
          {
            "name": "Контроллер: субъекты",
            "description": "REST API: субъекты"
          }
        ]
      }
    ],
    "paths": {
      "/x509-provider-service/api/v2/public/auth/sign-in/x509": {
        "post": {
          "tags": [
            "v2"
          ],
          "summary": "Аутентификация x509",
          "description": "POST метод REST API: Аутентификация x509",
          "operationId": "signInByX509",
          "responses": {
            "200": {
              "description": "Данные о токене",
              "content": {
                "*/*": {
                  "schema": {
                    "$ref": "#/components/schemas/ItemResponseTokenResponsePublic"
                  }
                }
              }
            },
            "500": {
              "description": "Внутренняя ошибка",
              "content": {
                "*/*": {
                  "schema": {
                    "$ref": "#/components/schemas/ErrorResponse"
                  }
                }
              }
            }
          }
        }
      },
      "/tasks-service/api/v2/public/tasks/pkcs12": {
        "post": {
          "tags": [
            "Контроллер: Задачи"
          ],
          "summary": "Регистрация новой заявки в формате pkcs#12",
          "description": "POST метод REST API: Регистрация новой заявки в формате pkcs#12",
          "operationId": "registerPkcs12",
          "parameters": [
            {
              "name": "certificateAuthorityId",
              "in": "query",
              "description": "ID ЦС",
              "schema": {
                "type": "string",
                "format": "uuid"
              }
            },
            {
              "name": "templateId",
              "in": "query",
              "description": "ID шаблона",
              "schema": {
                "type": "string",
                "format": "uuid"
              }
            },
            {
              "name": "subjectId",
              "in": "query",
              "description": "ID субъекта",
              "schema": {
                "type": "string",
                "format": "uuid"
              }
            },
            {
              "name": "foreignKey",
              "in": "query",
              "description": "Внешний ключ",
              "schema": {
                "type": "string"
              }
            }
          ],
          "requestBody": {
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/TaskRegisterPkcs12RequestPublic"
                }
              }
            },
            "required": true
          },
          "responses": {
            "200": {
              "description": "Заявка зарегистрирована",
              "content": {
                "*/*": {
                  "schema": {
                    "$ref": "#/components/schemas/ItemResponseTaskRegisterPkcs12ResponsePublic"
                  }
                }
              }
            },
            "500": {
              "description": "Внутренняя ошибка",
              "content": {
                "*/*": {
                  "schema": {
                    "$ref": "#/components/schemas/ErrorResponse"
                  }
                }
              }
            }
          }
        }
      },
      "/tasks-service/api/v2/public/tasks/pkcs10": {
        "post": {
          "tags": [
            "Контроллер: Задачи"
          ],
          "summary": "Регистрация новой заявки в формате pkcs#10",
          "description": "POST метод REST API: Регистрация новой заявки в формате pkcs#10",
          "operationId": "registerPkcs10AsFileDescription_1",
          "parameters": [
            {
              "name": "certificateAuthorityId",
              "in": "query",
              "description": "ID ЦС",
              "schema": {
                "type": "string",
                "format": "uuid"
              }
            },
            {
              "name": "templateId",
              "in": "query",
              "description": "ID шаблона",
              "schema": {
                "type": "string",
                "format": "uuid"
              }
            },
            {
              "name": "subjectId",
              "in": "query",
              "description": "ID субъекта",
              "schema": {
                "type": "string",
                "format": "uuid"
              }
            },
            {
              "name": "foreignKey",
              "in": "query",
              "description": "Внешний ключ",
              "schema": {
                "type": "string"
              }
            }
          ],
          "requestBody": {
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/TaskRegisterPkcs10RequestPublicFileDescription"
                }
              },
              "multipart/form-data": {
                "schema": {
                  "$ref": "#/components/schemas/TaskRegisterPkcs10RequestPublicMultipartFile"
                }
              }
            },
            "required": true
          },
          "responses": {
            "200": {
              "description": "Заявка зарегистрирована",
              "content": {
                "*/*": {
                  "schema": {
                    "$ref": "#/components/schemas/ItemResponseTaskRegisterPkcs10ResponsePublic"
                  }
                }
              }
            },
            "500": {
              "description": "Внутренняя ошибка",
              "content": {
                "*/*": {
                  "schema": {
                    "oneOf": [
                      {
                        "$ref": "#/components/schemas/ItemResponseTaskRegisterPkcs10ResponsePublic"
                      },
                      {
                        "$ref": "#/components/schemas/ErrorResponse"
                      }
                    ]
                  }
                }
              }
            }
          }
        }
      },
      "/security-service/api/v2/public/auth/refresh-token": {
        "post": {
          "tags": [
            "Контроллер: Авторизации"
          ],
          "summary": "Обновление токена",
          "description": "POST метод REST API: Обновление токена",
          "operationId": "refreshToken",
          "responses": {
            "200": {
              "description": "Данные о токене",
              "content": {
                "*/*": {
                  "schema": {
                    "$ref": "#/components/schemas/ItemResponseTokenResponse"
                  }
                }
              }
            },
            "500": {
              "description": "Внутренняя ошибка",
              "content": {
                "*/*": {
                  "schema": {
                    "$ref": "#/components/schemas/ErrorResponse"
                  }
                }
              }
            }
          }
        }
      },
      "/tasks-service/api/v2/public/tasks": {
        "get": {
          "tags": [
            "Контроллер: Задачи"
          ],
          "summary": "Поиск заявок",
          "description": "GET метод REST API: поиск заявок",
          "operationId": "findAll",
          "parameters": [
            {
              "name": "pageLimit",
              "in": "query",
              "description": "Пагинация: ограничение на размер списка",
              "schema": {
                "type": "integer",
                "format": "int32"
              }
            },
            {
              "name": "pageOffset",
              "in": "query",
              "description": "Пагинация: смещение от начала списка",
              "schema": {
                "type": "integer",
                "format": "int32"
              }
            },
            {
              "name": "sortBy",
              "in": "query",
              "description": "Сортировка: имя поля, по которому выполнять"
            }
          ]
        }
      }
    }
  }
}
```

```

"sortDirection", "in": "query", "description": "Сортировка: направление сортировки", "schema": {"type": "string"}}, {"name": "search", "in": "query", "description": "Фильтр: идентификатор заявки или название сертификата", "schema": {"type": "string"}}, {"name": "templateIds", "in": "query", "description": "Фильтр: идентификаторы шаблонов", "schema": {"type": "array", "items": {"type": "string", "format": "uuid"}}}, {"name": "creatorIds", "in": "query", "description": "Фильтр: идентификаторы пользователей", "schema": {"type": "array", "items": {"type": "string", "format": "uuid"}}}, {"name": "types", "in": "query", "description": "Фильтр: типы заявок", "schema": {"type": "array", "items": {"type": "string", "enum": ["PKCS10", "PKCS12", "ON_TOKEN", "SCEP", "WSTEP", "UNKNOWN"]}}}, {"name": "statuses", "in": "query", "description": "Фильтр: статусы заявок", "schema": {"type": "array", "items": {"type": "string", "enum": ["NEW", "FOR_APPROVE", "ISSUE_ERROR", "REJECTED", "COMPLETED", "CANCELLED", "AWAITING_IMPORT", "UNKNOWN"]}}}, {"name": "foreignKeys", "in": "query", "description": "Фильтр: внешние ключи", "schema": {"type": "array", "items": {"type": "string"}}}], "responses": {"200": {"description": "Заявки успешно получены", "content": {"*/": {"schema": {"$ref": "#/components/schemas/CollectionResponseTaskFindUIResponsePublic"}}}}, "500": {"description": "Внутренняя ошибка", "content": {"*/": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}, "/tasks-service/api/v2/public/tasks/{id}": {"get": {"tags": ["Контроллер: Задачи"], "summary": "Получение заявки по идентификатору", "description": "GET метод REST API: Получение заявки по идентификатору", "operationId": "findById", "parameters": [{"name": "id", "in": "path", "description": "Идентификатор заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}}], "responses": {"200": {"description": "Заявка", "content": {"*/": {"schema": {"$ref": "#/components/schemas/ItemResponseTaskFindUIResponsePublic"}}}}, "404": {"description": "Заявка не найдена", "content": {"*/": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}, "500": {"description": "Внутренняя ошибка", "content": {"*/": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}, "/tasks-service/api/v2/public/tasks/foreign-key/{foreignKey}": {"get": {"tags": ["Контроллер: Задачи"], "summary": "Получение заявки по внешнему ключу", "description": "GET метод REST API: Получение заявки по внешнему ключу", "operationId": "findByForeignKey", "parameters": [{"name": "foreignKey", "in": "path", "description": "Внешний ключ", "required": true, "schema": {"type": "string"}}}], "responses": {"200": {"description": "Заявка", "content": {"*/": {"schema": {"$ref": "#/components/schemas/ItemResponseTaskFindUIResponsePublic"}}}}, "404": {"description": "Заявка не найдена", "content": {"*/": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}, "500": {"description": "Внутренняя ошибка", "content": {"*/": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}, "/middleware-service/api/v2/public/templates": {"get": {"tags": ["Контроллер: шаблоны"], "summary": "Получить список шаблонов с фильтрацией, пагинацией и сортировкой", "description": "GET метод REST API получения списка шаблонов с фильтрацией, пагинацией и сортировкой", "operationId": "findAll_1", "parameters": [{"name": "pageLimit", "in": "query", "description": "Пагинация: ограничение на размер списка", "schema": {"type": "integer", "format": "int32"}}, {"name": "pageOffset", "in": "query", "description": "Пагинация: смещение от начала списка", "schema": {"type": "integer", "format": "int32"}}, {"name": "sortBy", "in": "query", "description": "Сортировка: имя поля, по которому выполнять сортировку", "schema": {"type": "array", "items": {"type": "string"}}}, {"name": "sortDirection", "in": "query", "description": "Сортировка: направление сортировки", "schema": {"type": "string"}}, {"name": "types", "in": "query", "description": "Фильтр: по типам шаблона", "schema": {"type": "array", "items": {"type": "string", "enum": ["EMBEDDED", "CLONED", "IMPORTED", "UNKNOWN"]}}}, {"name": "certificateType", "in": "query", "description": "Фильтр: по типу выпускаемого сертификата", "schema": {"type": "string", "description": "Перечисление типов сертификата", "enum": ["CERTIFICATE", "ROOT_CA", "SUB_CA", "UNKNOWN"]}}, {"name": "search", "in": "query", "description": "Фильтр: поиск по имени", "schema": {"type": "string", "default": "Name"}}, {"name": "removed", "in": "query", "description": "Фильтр: поиск удаленного шаблона", "schema": {"type": "boolean", "default": false}}, {"name": "id", "in": "query", "description": "Фильтр: поиск по списку идентификаторов", "schema": {"type": "array", "items": {"type": "string", "format": "uuid"}}}, {"name": "notId", "in": "query", "description": "Фильтр: поиск по списку идентификаторов (исключающий)", "schema": {"type": "array", "items": {"type": "string", "format": "uuid"}}}, {"name": "keyAlgorithm", "in": "query", "description": "Фильтр: алгоритм ключа сертификата", "schema": {"type": "array", "items": {"type": "string", "description": "Перечисление алгоритмов ключа сертификата", "enum": ["RSA", "ECDSA", "GOST_R_34_10_2012", "UNKNOWN"]}}}], "responses": {"200": {"description": "Список успешно получен", "content": {"application/json": {"schema": {"$ref": "#/components/schemas/CollectionResponseTaskFindUIResponsePublic"}}}}},

```

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

```

scription": "Идентификатор
заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}, "responses": {"200": {"desc
ription": "Файл
запроса
на
сертификат", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, "500": {"descriptio
n": "Внутренняя
ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, "/expor
t-service/api/v2/public/export/task/{taskId}/pkcs12": {"get": {"tags": ["Контроллер
экспорта"], "summary": "Получение контейнера pkcs12 по идентификатору заявки", "description": "GET
метод
REST
API:
Получение
контейнера
pkcs12", "operationId": "findP12ByTaskId", "parameters": [{"name": "taskId", "in": "path", "descriptio
n": "Идентификатор
заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}, "responses": {"200": {"desc
ription": "Файл
контейнера
pkcs12", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, "500": {"description": "
Внутренняя
ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, "/expor
t-service/api/v2/public/export/task/{taskId}/issuer/crl": {"get": {"tags": ["Контроллер
экспорта"], "summary": "Получение crl издателя по идентификатору заявки", "description": "GET метод
REST
API:
Получение
crl
издателя
по
идентификатору
заявки", "operationId": "findIssuerCrlByTaskId", "parameters": [{"name": "taskId", "in": "path", "desc
ription": "Идентификатор
заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}, "responses": {"200": {"desc
ription": "Файл
crl", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, "500": {"description": "Вну
тренняя
ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, "/expor
t-service/api/v2/public/export/task/{taskId}/issuer/chain": {"get": {"tags": ["Контроллер
экспорта"], "summary": "Получение цепочки сертификатов издателя по идентификатору
заявки", "description": "GET метод
REST
API:
Получение
цепочки
сертификатов
издателя", "operationId": "findIssuerChainByTaskId", "parameters": [{"name": "taskId", "in": "path", "
description": "Идентификатор
заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}, "responses": {"200": {"desc
ription": "Файл
цепочки
сертификатов", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, "500": {"descript
ion": "Внутренняя
ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, "/expor
t-service/api/v2/public/export/task/{taskId}/issuer/certificate": {"get": {"tags": ["Контроллер
экспорта"], "summary": "Получение сертификата издателя по идентификатору
заявки", "description": "GET метод
REST
API:
Получение
сертификата
издателя", "operationId": "findIssuerCertificateByTaskId", "parameters": [{"name": "taskId", "in": "p
ath", "description": "Идентификатор
заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}, "responses": {"200": {"desc
ription": "Файл
сертификата", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, "500": {"descripti
on": "Внутренняя
ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, "/expor
t-service/api/v2/public/export/task/{taskId}/chain": {"get": {"tags": ["Контроллер
экспорта"], "summary": "Получение цепочки сертификатов по идентификатору
заявки", "description": "GET метод
REST
API:
Получение
цепочки
сертификатов", "operationId": "findChainByTaskId", "parameters": [{"name": "taskId", "in": "path", "de
scription": "Идентификатор
заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}, "responses": {"200": {"desc
ription": "Файл
цепочки
сертификатов", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, "500": {"descript
ion": "Внутренняя
ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, "/expor
t-service/api/v2/public/export/task/{taskId}/certificate": {"get": {"tags": ["Контроллер
экспорта"], "summary": "Получение сертификата по идентификатору заявки", "description": "GET метод
REST
API:
Получение
сертификата", "operationId": "findCertificateByTaskId_1", "parameters": [{"name": "taskId", "in": "pa
th", "description": "Идентификатор
заявки", "required": true, "schema": {"type": "string", "format": "uuid"}}, "responses": {"200": {"desc
ription": "Файл
сертификата", "content": {"*//*": {"schema": {"type": "string", "format": "byte"}}}}, "500": {"descripti
on": "Внутренняя
ошибка", "content": {"*//*": {"schema": {"$ref": "#/components/schemas/ErrorResponse"}}}}}}, "compo
nents": {"schemas": {"Data": {"type": "object", "properties": {"id": {"type": "object", "description": "
Идентификатор
сущности,
работа
с
которым
привела
к
ошибке"}, "fieldName": {"type": "string", "description": "Имя поля, которое не прошло
валидацию"}, "fieldValue": {"type": "object", "description": "Значение поля, которое не прошло
валидацию"}, "causedBy": {"type": "string", "description": "Сообщение,
чем
вызвано
исключение"}, "description": "Объект
с
детальми
ошибки"}, "ErrorResponse": {"type": "object", "properties": {"status": {"type": "integer", "format": "i

```

```
nt32"},"code":{"type":"integer","description":"Код ошибки хххууу, где ххх - код сервиса, ууу -
внутренний код ошибки"},"format":"int32"},"message":{"type":"string","description":"Описание
ошибки"},"data":{"$ref":"#/components/schemas/Data"},"description":"Стандартизированный
объект ответа сервиса, содержащий информацию об
ошибке"},"ItemResponseTokenResponsePublic":{"type":"object","properties":{"status":{"type":"in
teger","format":"int32"},"data":{"$ref":"#/components/schemas/TokenResponsePublic"},"descript
ion":"Стандартизированный объект ответа сервиса, содержащий один единственный
элемент"},"TokenResponsePublic":{"type":"object","properties":{"token":{"type":"string"},"descr
iption":"Токен пользователя"},"refresh":{"type":"string"},"description":"Токен обновления
пользователя"},"description":"Объект ответа сервиса: данные о
токене"},"TaskRegisterPkcs12RequestPublic":{"type":"object","properties":{"subjectDN":{"type":
"object","additionalProperties":{"type":"array","description":"Subject
DN"},"items":{"type":"string","description":"Subject
DN"},"description":"Subject
DN"},"subjectAltName":{"type":"object","additionalProperties":{"type":"array","description":"S
ubject
AltName"},"items":{"type":"string","description":"Subject
AltName"},"description":"Subject
AltName"},"keyBits":{"type":"integer","description":"Размерность
ключа"},"format":"int32"},"keyAlgorithm":{"type":"string","description":"Перечисление
алгоритмов
ключа
сертификата"},"enum":["RSA","ECDSA","GOST_R_34_10_2012","UNKNOWN"]},"password":{"type":"string"
},"description":"Пароль от контейнера"},"description":"Запрос на регистрацию новой заявки в
формате
pkcs#12"},"ItemResponseTaskRegisterPkcs12ResponsePublic":{"type":"object","properties":{"statu
s":{"type":"integer","format":"int32"},"data":{"$ref":"#/components/schemas/TaskRegisterPkcs12
ResponsePublic"},"description":"Стандартизированный объект ответа сервиса, содержащий один
единственный
элемент"},"TaskRegisterPkcs12ResponsePublic":{"type":"object","properties":{"id":{"type":"stri
ng"},"description":"Идентификатор
заявки"},"format":"uuid"},"templateId":{"type":"string","description":"Идентификатор
шаблона"},"format":"uuid"},"type":{"type":"string","description":"Тип
заявки"},"enum":["PKCS10","PKCS12","ON_TOKEN","SCEP","WSTEP","UNKNOWN"]},"status":{"type":"stri
ng"},"description":"Статус
заявки"},"enum":["NEW","FOR_APPROVE","ISSUE_ERROR","REJECTED","COMPLETED","CANCELLED","AWAITING
_IMPORT","UNKNOWN"]},"creatorId":{"type":"string","description":"Идентификатор пользователя,
создавшего заявку"},"format":"uuid"},"subjectId":{"type":"string","description":"Идентификатор
субъекта"},"format":"uuid"},"certificateId":{"type":"string","description":"Идентификатор
сертификата"},"format":"uuid"},"name":{"type":"string","description":"Название
активного
сертификата"},"updated":{"type":"string","description":"Время
последнего
обновления
заявки"},"format":"date-time"},"created":{"type":"string","description":"Время
создания
заявки"},"format":"date-time"},"foreignKey":{"type":"string","description":"Внешний
ключ"},"description":"Ответ
на
регистрацию
новой
заявки
в
формате
pkcs#12"},"FileDescription":{"type":"object","properties":{"contentType":{"type":"string"},"fi
leName":{"type":"string"},"data":{"type":"string","format":"byte"},"description":"Файл
запроса"},"TaskRegisterPkcs10RequestPublicFileDescription":{"type":"object","properties":{"req
uest":{"$ref":"#/components/schemas/FileDescription"},"subjectName":{"type":"object","addition
alProperties":{"type":"array","description":"Отличительное
имя
субъекта"},"items":{"type":"string","description":"Отличительное
имя
субъекта"},"description":"Отличительное
имя
субъекта"},"subjectAltName":{"type":"object","additionalProperties":{"type":"array","descripti
on":"Дополнительное имя субъекта"},"items":{"type":"string","description":"Дополнительное имя
субъекта"},"description":"Дополнительное имя субъекта"},"description":"Запрос на регистрацию
новой
заявки
в
формате
pkcs#10"},"ItemResponseTaskRegisterPkcs10ResponsePublic":{"type":"object","properties":{"statu
s":{"type":"integer","format":"int32"},"data":{"$ref":"#/components/schemas/TaskRegisterPkcs10
ResponsePublic"},"description":"Стандартизированный объект ответа сервиса, содержащий один
единственный
элемент"},"TaskRegisterPkcs10ResponsePublic":{"type":"object","properties":{"id":{"type":"stri
ng"},"description":"Идентификатор
заявки"},"format":"uuid"},"templateId":{"type":"string","description":"Идентификатор
шаблона"},"format":"uuid"},"type":{"type":"string","description":"Тип
заявки"},"enum":["PKCS10","PKCS12","ON_TOKEN","SCEP","WSTEP","UNKNOWN"]},"status":{"type":"stri
ng"},"description":"Статус
заявки"},"enum":["NEW","FOR_APPROVE","ISSUE_ERROR","REJECTED","COMPLETED","CANCELLED","AWAITING
_IMPORT","UNKNOWN"]},"creatorId":{"type":"string","description":"Идентификатор пользователя,
создавшего заявку"},"format":"uuid"},"subjectId":{"type":"string","description":"Идентификатор
субъекта"},"format":"uuid"},"certificateId":{"type":"string","description":"Идентификатор
сертификата"},"format":"uuid"},"name":{"type":"string","description":"Название
активного
сертификата"},"updated":{"type":"string","description":"Время
последнего
обновления
заявки"},"format":"date-time"},"created":{"type":"string","description":"Время
создания
заявки"},"format":"date-time"},"foreignKey":{"type":"string","description":"Внешний
ключ"},"description":"Ответ
на
регистрацию
новой
заявки
в
формате
pkcs#10"},"TaskRegisterPkcs10RequestPublicMultipartFile":{"type":"object","properties":{"reque
st":{"type":"string","format":"binary"},"subjectName":{"type":"object","additionalProperties":
```

```
{
  "type": "array", "description": "Отличительное имя субъекта",
  "items": {
    "type": "string", "description": "Отличительное имя субъекта"
  }, "description": "Отличительное имя субъекта",
  "subjectAltName": {
    "type": "object", "additionalProperties": {
      "type": "array", "description": "Дополнительное имя субъекта",
      "items": {
        "type": "string", "description": "Дополнительное имя субъекта"
      }, "description": "Дополнительное имя субъекта"
    }, "description": "Запрос на регистрацию новой заявки в формате pkcs#10"
  }, "ItemResponseTokenResponse": {
    "type": "object", "properties": {
      "status": {
        "type": "integer", "format": "int32", "data": {
          "$ref": "#/components/schemas/TokenResponse"
        }, "description": "Стандартизированный объект ответа сервиса, содержащий один единственный элемент"
      }, "TokenResponse": {
        "type": "object", "properties": {
          "token": {
            "type": "string", "description": "Токен пользователя"
          }, "refresh": {
            "type": "string", "description": "Токен обновления пользователя"
          }, "description": "Объект ответа сервиса: данные о токене"
        }, "CollectionResponseTaskFindUIResponsePublic": {
          "type": "object", "properties": {
            "status": {
              "type": "integer", "format": "int32", "data": {
                "$ref": "#/components/schemas/DataTaskFindUIResponsePublic"
              }, "description": "Стандартизированный объект ответа сервиса, содержащий коллекцию элементов"
            }, "DataTaskFindUIResponsePublic": {
              "type": "object", "properties": {
                "range": {
                  "$ref": "#/components/schemas/Range"
                }, "items": {
                  "type": "array", "description": "Коллекция элементов",
                  "items": {
                    "$ref": "#/components/schemas/TaskFindUIResponsePublic"
                  }, "description": "Полезная нагрузка с коллекцией элементов"
                }, "Range": {
                  "type": "object", "properties": {
                    "count": {
                      "type": "integer", "description": "Всего элементов",
                      "format": "int64", "offset": {
                        "type": "integer", "description": "Количество пропущенных элементов",
                        "format": "int64", "limit": {
                          "type": "integer", "description": "Ограничение количества элементов в выборке",
                          "format": "int32", "totalPages": {
                            "type": "integer", "description": "Общее число страниц",
                            "format": "int32", "pageNumber": {
                              "type": "integer", "description": "Номер страницы выборки",
                              "format": "int32"
                            }, "description": "Данные по выборке"
                            }, "TaskFindUIResponsePublic": {
                              "type": "object", "properties": {
                                "id": {
                                  "type": "string", "description": "Идентификатор заявки",
                                  "format": "uuid", "templateId": {
                                    "type": "string", "description": "Идентификатор шаблона",
                                    "format": "uuid", "certificateAuthorityId": {
                                      "type": "string", "description": "Идентификатор сертификата издателя",
                                      "format": "uuid", "type": {
                                        "type": "string", "description": "Тип заявки",
                                        "enum": ["PKCS10", "PKCS12", "ON_TOKEN", "SCEP", "WSTEP", "UNKNOWN"]
                                      }, "status": {
                                        "type": "string", "description": "Статус заявки",
                                        "enum": ["NEW", "FOR_APPROVE", "ISSUE_ERROR", "REJECTED", "COMPLETED", "CANCELLED", "AWAITING_IMPORT", "UNKNOWN"]
                                      }, "creatorId": {
                                        "type": "string", "description": "Идентификатор пользователя, создавшего заявку",
                                        "format": "uuid", "subjectId": {
                                          "type": "string", "description": "Идентификатор субъекта",
                                          "format": "uuid", "certificateId": {
                                            "type": "string", "description": "Идентификатор сертификата",
                                            "format": "uuid", "name": {
                                              "type": "string", "description": "Название активного сертификата",
                                              "updated": {
                                                "type": "string", "description": "Время последнего обновления заявки",
                                                "format": "date-time", "created": {
                                                  "type": "string", "description": "Время создания заявки",
                                                  "format": "date-time", "foreignKey": {
                                                    "type": "string", "description": "Внешний ключ",
                                                    "templateName": {
                                                      "type": "string", "description": "Имя шаблона",
                                                      "msUpn": {
                                                        "type": "string", "description": "MS UPN"
                                                      }, "description": "Объект ответа сервиса, содержащий данные об заявке"
                                                    }, "ItemResponseTaskFindUIResponsePublic": {
                                                      "type": "object", "properties": {
                                                        "status": {
                                                          "type": "integer", "format": "int32", "data": {
                                                            "$ref": "#/components/schemas/TaskFindUIResponsePublic"
                                                          }, "description": "Стандартизированный объект ответа сервиса, содержащий один единственный элемент"
                                                        }, "CollectionResponseTemplateItemResponsePublic": {
                                                          "type": "object", "properties": {
                                                            "status": {
                                                              "type": "integer", "format": "int32", "data": {
                                                                "$ref": "#/components/schemas/DataTemplateItemResponsePublic"
                                                              }, "description": "Стандартизированный объект ответа сервиса, содержащий коллекцию элементов"
                                                            }, "DataTemplateItemResponsePublic": {
                                                              "type": "object", "properties": {
                                                                "range": {
                                                                  "$ref": "#/components/schemas/Range"
                                                                }, "items": {
                                                                  "type": "array", "description": "Коллекция элементов",
                                                                  "items": {
                                                                    "$ref": "#/components/schemas/TemplateItemResponsePublic"
                                                                  }, "description": "Полезная нагрузка с коллекцией элементов"
                                                                }, "TemplateItemResponsePublic": {
                                                                  "type": "object", "properties": {
                                                                    "id": {
                                                                      "type": "string", "description": "Идентификатор шаблона",
                                                                      "format": "uuid", "name": {
                                                                        "type": "string", "description": "Наименование шаблона",
                                                                        "certificateType": {
                                                                          "type": "string", "description": "Перечисление типов сертификата",
                                                                          "enum": ["CERTIFICATE", "ROOT_CA", "SUB_CA", "UNKNOWN"]
                                                                        }, "type": {
                                                                          "type": "string", "description": "Тип шаблона",
                                                                          "enum": ["EMBEDDED", "CLONED", "IMPORTED", "UNKNOWN"]
                                                                        }, "certificateCount": {
                                                                          "type": "integer", "description": "Количество выпущенных сертификатов",
                                                                          "format": "int64", "removed": {
                                                                            "type": "boolean", "description": "Удаленный шаблон",
                                                                            "created": {
                                                                              "type": "string", "description": "Дата создания шаблона",
                                                                              "format": "date-time", "updated": {
                                                                                "type": "string", "description": "Дата обновления шаблона",
                                                                                "format": "date-time", "description": "Объект ответа сервиса, содержащий данные элемента коллекции шаблонов"
                                                                              }, "CertificatePoliciesPublic": {
                                                                                "type": "object", "properties": {
                                                                                  "critical": {
                                                                                    "type": "boolean", "description": "Считать критическим",
                                                                                    "values": {
                                                                          "type": "array", "description": "Перечень политик",
                                                                          "items": {
                                                                            "type": "string", "description": "Перечень политик"
                                                                          }, "description": "Класс сертификата",
                                                                          "ExtendedKeyUsagePublic": {
                                                                            "type": "object", "properties": {
                                                                              "critical": {
                                                                                "type": "boolean", "description": "Считать критическим",
                                                                                "values": {
                                                                      "type": "array", "description": "Значения"

```

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

```
time"}}, "description": "Объект ответа сервиса, содержащий данные шаблона", "CertificateChainPublic": {"type": "object", "properties": {"id": {"type": "string", "description": "Уникальный идентификатор сертификата", "format": "uuid"}, "name": {"type": "string", "description": "Имя сертификата (формируется из CN, разделитель \"_\" \" \")", "subjectDN": {"type": "object", "additionalProperties": {"type": "array", "description": "Subject DN", "items": {"type": "string", "description": "Subject DN"}}, "description": "Subject DN"}, "issuer": {"$ref": "#/components/schemas/CertificateChainPublic"}}, "description": "ПОЖО: цепочка сертификатов", "CertificateDownloadActionsPublic": {"type": "object", "properties": {"p12": {"type": "boolean", "description": "Флаг: выгрузка контейнера p12"}, "pem": {"type": "boolean", "description": "Флаг: выгрузка сертификата"}, "csr": {"type": "boolean", "description": "Флаг: выгрузка запроса сертификата"}, "description": "ПОЖО: Действия выгрузки у сертификата"}, "CertificateResponsePublic": {"type": "object", "properties": {"id": {"type": "string", "description": "Уникальный идентификатор сертификата", "format": "uuid"}, "chain": {"$ref": "#/components/schemas/CertificateChainPublic"}, "serialnumber": {"type": "string", "description": "Серийный номер сертификата"}, "fingerprint": {"type": "string", "description": "Фингерпринт сертификата"}, "name": {"type": "string", "description": "Имя сертификата (формируется из CN, разделитель \"_\" \" \")", "issuerId": {"type": "string", "description": "Уникальный идентификатор издающего сертификата", "format": "uuid"}, "issuerDN": {"type": "object", "additionalProperties": {"type": "array", "description": "Issuer DN", "items": {"type": "string", "description": "Issuer DN"}}, "description": "Issuer DN"}, "subjectDN": {"type": "object", "additionalProperties": {"type": "array", "description": "Subject DN", "items": {"type": "string", "description": "Subject DN"}}, "description": "Subject DN"}, "subjectAltName": {"type": "object", "additionalProperties": {"type": "array", "description": "Subject AltName", "items": {"type": "string", "description": "Subject AltName"}}, "description": "Subject AltName"}, "keyAlgorithm": {"type": "string", "description": "Перечисление алгоритмов ключа сертификата", "enum": ["RSA", "ECDSA", "GOST_R_34_10_2012", "UNKNOWN"]}, "hashAlgorithm": {"type": "string", "description": "Перечисление Hash алгоритмов", "enum": ["SHA1", "SHA256", "SHA384", "SHA512", "SHA3_256", "SHA3_384", "SHA3_512", "RSASSA_PSS", "MD5", "GOST_R_34_11_2012", "UNKNOWN"]}, "keyBits": {"type": "integer", "description": "Размерность ключа", "format": "int32"}, "templateId": {"type": "string", "description": "Идентификатор шаблона, по которому выпущен сертификат", "format": "uuid"}, "templateName": {"type": "string", "description": "Название шаблона, по которому выпущен сертификат"}, "type": {"type": "string", "description": "Перечисление типов сертификата", "enum": ["CERTIFICATE", "ROOT_CA", "SUB_CA", "UNKNOWN"]}, "validFrom": {"type": "string", "description": "Дата начала действия сертификата", "format": "date-time"}, "validTo": {"type": "string", "description": "Дата окончания действия сертификата", "format": "date-time"}, "status": {"type": "string", "description": "Перечисление: статус сертификата", "enum": ["ACTIVE", "HOLD", "REVOKE", "REQUEST", "UNKNOWN"]}, "actions": {"$ref": "#/components/schemas/CertificateDownloadActionsPublic"}, "publicKey": {"type": "string", "description": "Публичный ключ сертификата"}, "certificateType": {"type": "string", "description": "Тип сертификата"}, "version": {"type": "integer", "description": "Версия сертификата", "format": "int32"}, "subjectKeyIdentifier": {"type": "string", "description": "Идентификатор ключа сертификата"}, "authorityKeyIdentifier": {"type": "string", "description": "Идентификатор ключа издателя"}, "keyUsages": {"type": "array", "description": "Разрешенные использования ключа", "items": {"$ref": "#/components/schemas/KeyUsageDictResponsePublic"}}, "extendedKeyUsages": {"type": "array", "description": "Разрешенные расширенные использования ключа", "items": {"$ref": "#/components/schemas/ExtendedKeyUsageDictResponsePublic"}}, "ca": {"type": "boolean", "description": "Флаг: сертификат ЦС"}, "aiaUrls": {"type": "array", "description": "Точки распространения AIA", "items": {"type": "string", "description": "Точки распространения AIA"}}, "ocspUrls": {"type": "array", "description": "URL сервисов OCSP", "items": {"type": "string", "description": "URL сервисов OCSP"}}, "crlUrls": {"type": "array", "description": "Точки распространения CRL", "items": {"type": "string", "description": "Точки распространения CRL"}}, "deltaCrlUrls": {"type": "array", "description": "Точки распространения Delta CRL", "items": {"type": "string", "description": "Точки распространения Delta CRL"}}, "description": "Ответ сервиса: данные сертификата", "ExtendedKeyUsageDictResponsePublic": {"type": "object", "properties": {"id": {"type": "string", "description": "Перечисление расширенного использования ключа", "enum": ["EKU_PKIX_ANY_EXTENDED_KEY_USAGE", "CSN_369791_TLS_CLIENT", "CSN_369791_TLS_SERVER", "CLIENT_AUTHENTICATION", "CODE_SIGNING", "EAP_OVER_LAN", "EAP_OVER_PPP", "ETSI_TSL_SIGNING", "EMAIL_PROTECTION", "ICAO_DEVIATION_LIST_SIGNING", "EKU_INTEL_AMT", "INTERNET_KEY_EXCHANGE_FOR_IPSEC", "KERBEROS_CLIENT_AUTHENTICATION", "EKU_KRB_PKINIT_KDC", "MS_COMMERCIAL_CODE_SIGNING", "MS_DOCUMENT_SIGNING", "MS_EFS_RECOVERY", "MS_ENCRYPTED_FILE_SYSTEM", "MS_INDIVIDUAL_CODE_SIGNING", "MS_SMART_CARD_LOGON", "OCSP_SIGNER", "EKU_ADOBE_PDF_SIGNING", "PIV_CARD_AUTHENTICATION", "SCVP_CLIENT", "
```



```

SCVP_SERVER", "SIP_DOMAIN", "EKU_PKIX_SSH_CLIENT", "SSH_SERVER", "SERVER_AUTHENTICATION", "TIME_STAMPING", "ICAO_MASTER_LIST_SIGNING", "UNKNOWN"]], "value": {"type": "string", "description": "Наименование элемента"}, "oid": {"type": "string", "description": "Идентификатор объекта"}, "description": {"type": "string", "description": "Описание использования ключа"}}, {"description": "Объект ответа сервиса, содержащий данные расширенного использования ключа"}, {"ItemResponseCertificateResponsePublic": {"type": "object", "properties": {"status": {"type": "integer", "format": "int32"}, "data": {"$ref": "#/components/schemas/CertificateResponsePublic"}}, "description": "Стандартизированный объект ответа сервиса, содержащий один единственный элемент"}, {"KeyUsageDictResponsePublic": {"type": "object", "properties": {"id": {"type": "string", "description": "Перечисление использования ключа", "enum": ["DIGITAL_SIGNATURE", "NON_REPUDIATION", "KEY_ENCIPHERMENT", "DATA_ENCIPHERMENT", "KEY_AGREEMENT", "KEY_CERT_SIGN", "CRL_SIGN", "ENCIPHER_ONLY", "DECIPHER_ONLY", "UNKNOWN"]}, "value": {"type": "string", "description": "Наименование элемента"}, "description": {"type": "string", "description": "Описание использования ключа"}}, {"description": "Объект ответа сервиса, содержащий данные использования ключа"}, {"CollectionResponseSubjectResponsePublic": {"type": "object", "properties": {"status": {"type": "integer", "format": "int32"}, "data": {"$ref": "#/components/schemas/DataSubjectResponsePublic"}}, "description": "Стандартизированный объект ответа сервиса, содержащий коллекцию элементов"}, {"DataSubjectResponsePublic": {"type": "object", "properties": {"range": {"$ref": "#/components/schemas/Range"}, "items": {"type": "array", "description": "Коллекция элементов"}, "items": {"$ref": "#/components/schemas/SubjectResponsePublic"}}, "description": "Поле значения нагрузки с коллекцией элементов"}, {"ResourceItemResponsePublic": {"type": "object", "properties": {"id": {"type": "string", "description": "ID ресурсной системы", "format": "uuid"}, "commonName": {"type": "string", "description": "Общее наименование (CN) ресурсной системы"}, "distinguishedName": {"type": "string", "description": "Отличающиеся имя ресурсной системы"}, "description": "Ответ: ресурсная система (сокращенный)"}, {"SubjectNameFieldItemResponsePublic": {"type": "object", "properties": {"values": {"type": "array", "description": "Значения атрибута субъекта", "items": {"type": "string", "description": "Значения атрибута субъекта"}}, "editable": {"type": "boolean", "description": "Флаг: редактируемый атрибут"}, "description": "Ответ: данные атрибута субъекта"}, {"SubjectResponsePublic": {"type": "object", "properties": {"id": {"type": "string", "description": "ID субъекта", "format": "uuid"}, "commonName": {"type": "string", "description": "Общее наименование (CN) субъекта"}, "distinguishedName": {"type": "string", "description": "Отличающееся наименование (DN) субъекта"}, "resource": {"$ref": "#/components/schemas/ResourceItemResponsePublic"}, "subjectName": {"type": "object", "additionalProperties": {"$ref": "#/components/schemas/SubjectNameFieldItemResponsePublic"}}, "description": "Отличительное имя субъекта"}, "subjectAltName": {"type": "object", "additionalProperties": {"$ref": "#/components/schemas/SubjectNameFieldItemResponsePublic"}}, "description": "Дополнительное имя субъекта"}, "isConnected": {"type": "boolean", "description": "Флаг: подключение субъекта к ресурсной системе"}, "isBlocked": {"type": "boolean", "description": "Флаг: блокировка субъекта в ресурсной системе"}, "modify": {"type": "string", "description": "Дата и время обновления в ресурсной системе", "format": "date-time"}, "certificatesCount": {"type": "integer", "description": "Количество сертификатов", "format": "int64"}, "updated": {"type": "string", "description": "Дата и время обновления", "format": "date-time"}, "created": {"type": "string", "description": "Дата и время создания", "format": "date-time"}}, {"description": "Ответ: субъект"}, {"ItemResponseSubjectResponsePublic": {"type": "object", "properties": {"status": {"type": "integer", "format": "int32"}, "data": {"$ref": "#/components/schemas/SubjectResponsePublic"}}, "description": "Стандартизированный объект ответа сервиса, содержащий один единственный элемент"}, {"GetApiVersionResponse": {"type": "object", "properties": {"status": {"type": "integer", "format": "int32"}, "data": {"$ref": "#/components/schemas/VersionResponse"}}, {"VersionResponse": {"type": "object", "properties": {"version": {"type": "string", "description": "Версия модуля"}}, {"description": "Объект ответа, содержащий версию модуля"}}, {"securitySchemes": {"X-API-KEY": {"type": "apiKey", "description": "API Key", "name": "X-API-KEY", "in": "header"}, "token": {"type": "apiKey", "description": "Token", "name": "token", "in": "header"}, "X-AECA-Principal-ID": {"type": "apiKey", "description": "UUID", "name": "X-AECA-Principal-ID", "in": "header"}, "X-AECA-Is-Account": {"type": "apiKey", "description": "Is Account", "name": "X-AECA-Is-Account", "in": "header"}, "X-AECA-Principal-Name": {"type": "apiKey", "description": "Principal name", "name": "X-AECA-Principal-Name", "in": "header"}, "X-AECA-User-Role": {"type": "apiKey", "description": "Role", "name": "X-AECA-User-Role", "in": "header"}, "X-AECA-User-Authorities": {"type": "apiKey", "description": "Authorities", "name": "X-AECA-User-Authorities", "in": "header"}, "X-User-Session-Id": {"type": "apiKey", "description": "Session ID", "name": "X-User-Session-Id", "in": "header"}}}}

```

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

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

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]