Аутентификация с использованием ЕСИА

Оглавление

RooX UIDM поддерживает аутентификацию с использованием внешней системы хранения учётных данных пользователей — Единой системы идентификации и аутентификации (ЕСИА).

ЕСИА

> Единая система идентификации и аутентификации (федеральная государственная информационная система «Единая система идентификации и аутентификации в инфраструктуре, обеспечивающей информационно-технологическое взаимодействие информационных систем, используемых для предоставления государственных и муниципальных услуг в электронной форме»).

ЕСИА является федеральной государственной информационной системой, в которой содержатся персональные данные граждан Российской Федерации, а также иные сведения (о статусе индивидуального предпринимателя, о юридических лицах и др.). ЕСИА используется в первую очередь для доступа граждан к сервисам государственных услуг и иных взаимодействий в электронной форме граждан с органами государственной власти.

RooX UIDM позволяет пользователю с учётной записью в ЕСИА получить доступ к защищаемому ресурсу при выполнении пользователем аутентификации в ЕСИА.

Использование в RooX UIDM идентификации и аутентификации при помощи ЕСИА:

- освобождает пользователя от необходимости хранить логин и пароль от учётной записи RooX UIDM, достаточно знать логин и пароль от учётной записи ЕСИА. Существует возможность беспарольной аутентификации пользователей (аутентификация только с использованием ЕСИА, когда учётная запись RooX UIDM не содержит установленного пароля);
- сохраняет за пользователем возможность аутентифицироваться в RooX UIDM с использованием других средств (логин и пароль, сертификат электронной подписи и других);
- обеспечивает заказчика достоверными сведениями о пользователе: ФИО, паспортными данными, СНИЛС (в случае, если учётная запись пользователя обладает статусом «подтверждённая»), и роли такого пользователя в юридических лицах;
- позволяет назначать пользователю полномочия в RooX UIDM в соответствии с его ролью в юридическом лице (в соответствии со сведениями ЕСИА).

Степень достоверности сведений о пользователе, содержащихся в учётной записи ЕСИА, выражается в статусе такой учётной записи:

Таблица 1. Статусы (уровни) учётных записей физических лиц в ЕСИА

упрощённая (непроверенная) учётная запись	минимальный набор сведений о пользователе
стандартная учётная запись	данные о пользователе сверены с государственными информационными ресурсами
подтверждённая учётная запись	данные о пользователе сверены с государственными информационными ресурсами, а личность пользователя подтверждена одним из доступных способов подтверждения

Достоверность сведений подтверждённой учётной записи ЕСИА обеспечивается проверкой сведений, предоставленных пользователем, органами государственной власти.

RooX UIDM не содержит заранее установленных требований к статусу учётной записи пользователя в ЕСИА, с использованием которой будет производиться аутентификация в RooX UIDM. Ограничения статуса учётной записи пользователя, аутентифицирующегося с использованием ЕСИА, устанавливаются в соответствии с требованиями заказчика.

ВАЖНО

Токены доступа, выпускаемые RooX UIDM, содержат в себе свойство (клейм) `amr`, сообщающее об уровне доверия к способу аутентификации пользователя. Уровень доверия при аутентификации с использованием

ЕСИА выше, чем уровень доверия к аутентификации с использованием логина и пароля RooX UIDM).

На основании этого уровня доверия защищаемые системы могут принимать решения о разграничении доступа пользователей к своим данным.

Нормативные документы Единой системы идентификации и аутентификации

Нормативные документы ЕСИА размещены на сайте Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации в разделе «Документы» по направлению Единая система идентификации и аутентификации (ЕСИА).

Состав наборов данных, предоставляемых ЕСИА в соответствии с заданными областями доступа (scope), содержится в актуальной версии Методических рекомендаций по использованию Единой системы идентификации и аутентификации, размещённых на сайте Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации в разделе «Документы» по направлению Единая система идентификации и аутентификации (ЕСИА).

Требования к безопасности при использовании аутентификации с использованием ЕСИА

Корректность обработки статуса учётной записи ЕСИА

Учётная запись пользователя в ЕСИА может находиться в одном из трёх статусов, соответствующих уровню доверия к содержащимся в ней данным.

Необходимо убедиться, что учётные записи ЕСИА с различным статусом логически корректно обрабатываются защищаемой системой.

Использование сертификата электронной подписи для подписания запросов к ЕСИА

Запросы экземпляра RooX UIDM к ЕСИА должны быть подписаны закрытым ключом сертификата электронной подписи. Статус сертификата электронной подписи проверяется ЕСИА. Сертификаты, выпущенные неаккредитованными удостоверяющими центрами (самоподписанные), не могут быть использованы для подписания запросов к ЕСИА.

Владельцу защищаемой системы необходимо обеспечить выпуск сертификата электронной подписи юридического лица.

Соблюдение требований по защите персональных данных

Данные, полученные из ЕСИА, являются персональными.

Владелец защищаемой системы должен убедиться, что выполняются требования к хранению персональных данных, в том числе возможность отзыва согласия пользователя на обработку персональных данных, а также другие положения федерального закона от 27.07.2006 № 152-ФЗ «О персональных данных».

Контроль доступа бывших сотрудников организации

В случае использования модели доступа с организациями необходимо убедиться, что защищаемая система корректно отслеживает и обрабатывает случаи изменения в ЕСИА ролей и отношений между организациями и их сотрудниками (например, бывший генеральный директор не должен иметь возможности подавать платежные поручения).

Сценарии аутентификации с использованием ЕСИА

Системный сценарий аутентификации с использованием ЕСИА

- Пользователь обращается к защищаемому ресурсу.
- Пользователь выполняет переход в ЕСИА для аутентификации.
- Сервис аутентификации ЕСИА аутентифицирует пользователя.
- Сервис аутентификации ЕСИА получает согласие пользователя на предоставление RooX UIDM его данных.
- Сервис аутентификации ЕСИА перенаправляет пользователя обратно в RooX UIDM и передает авторизационный код.
- RooX UIDM формирует запрос с использованием авторизационного кода на получение токена идентификации.
- RooX UIDM получает ответ, содержащий необходимый токен идентификации.
- RooX UIDM проводит проверку токена идентификации.
- При успешной проверке токена идентификации пользователь считается аутентифицированным.

Пользовательский сценарий аутентификации с использованием ЕСИА

- Пользователь обращается к защищаемому ресурсу.
- Пользователю отображается экран аутентификации с кнопкой «Войти при помощи ЕСИА».
- Пользователь нажимает кнопку и переходит на сайт ЕСИА к экрану аутентификации в ЕСИА.
- Пользователь успешно выполняет аутентификацию в ЕСИА.
- Пользователь автоматически перенаправляется и получает доступ к защищаемому ресурсу.

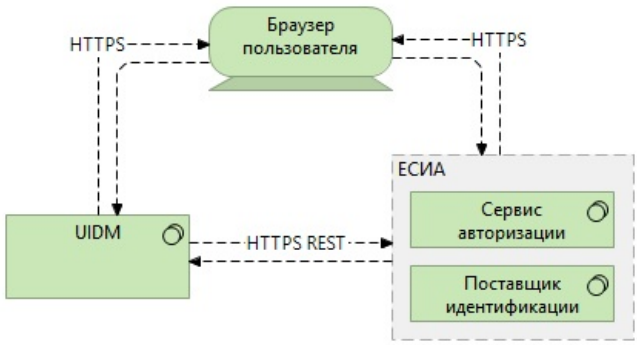

Рисунок 1. Схема взаимодействия RooX UIDM и ЕСИА при аутентификации

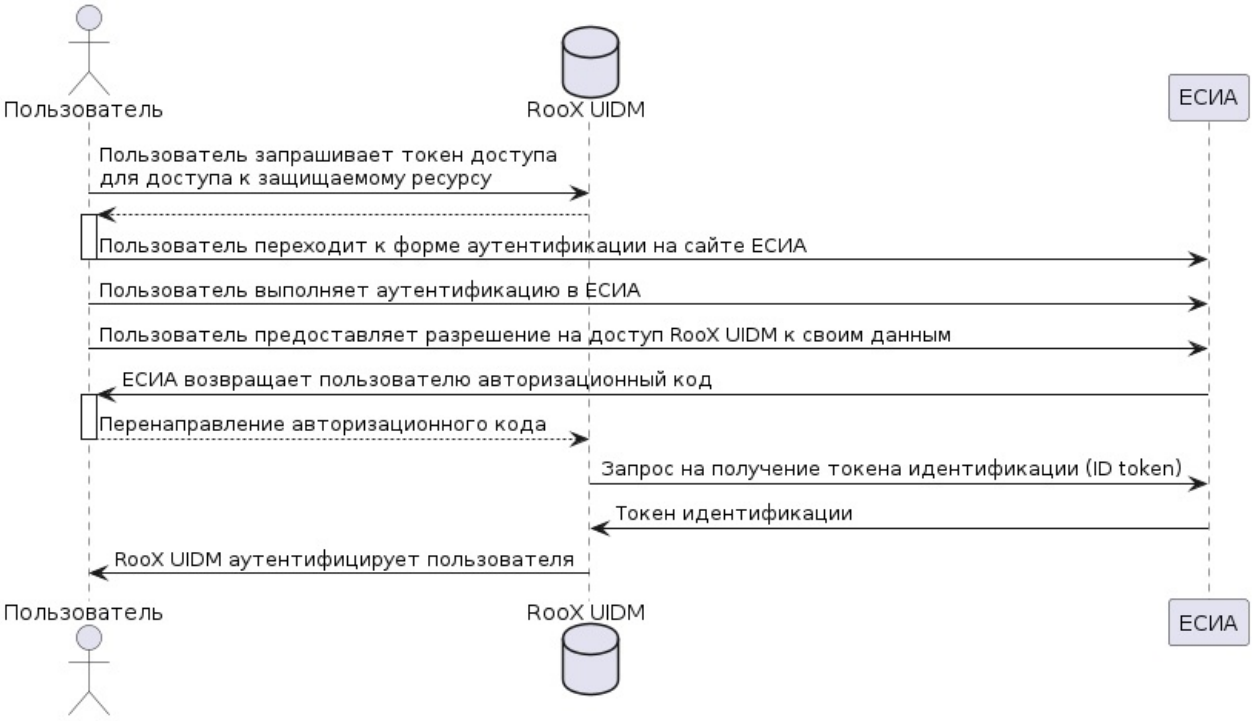

Рисунок 2. Диаграмма аутентификации с использованием ЕСИА

Модели данных

Сведения о пользователе, загруженные RooX UIDM из ЕСИА, хранятся в виде строк формата JSON в полях `profile` и `userinfo` таблицы `external_user` БД Roox UIDM.

Внешние привязки пользователя хранятся в таблице `partner_mappings` БД RooX UIDM.

Внешние привязки пользователя к ЕСИА могут быть получены с использованием API управления привязками.

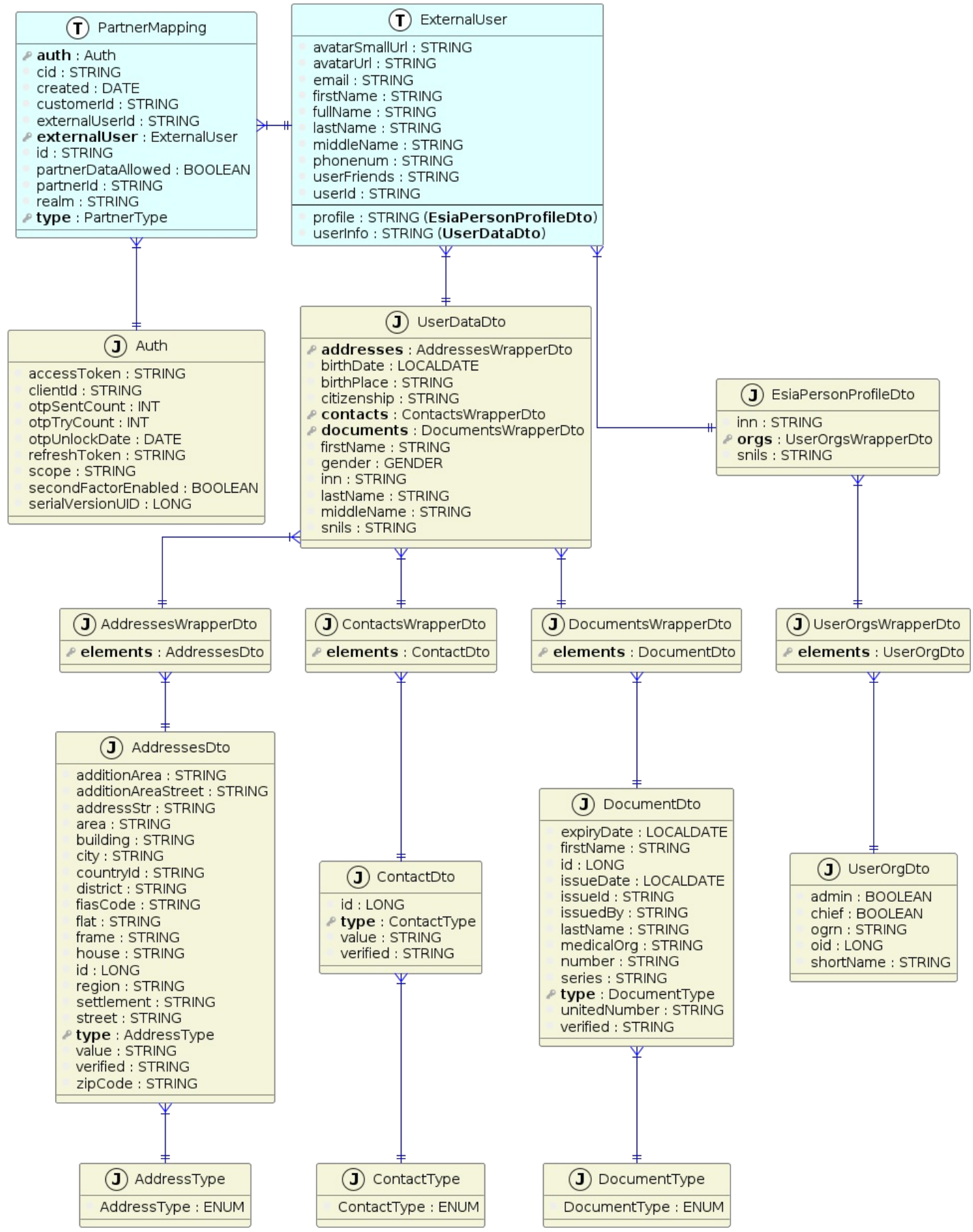

Рисунок 3. Общий вид модели данных ЕСИА

Модель данных PartnerMapping

Атрибут	Тип данных	Описание	Дополнительная информация
`auth`	Auth	Данные аутентификации	Инициализируется при создании экземпляра класса: `new Auth()`

`cid`	String	`cid` пользователя (из Claim)	
`created`	Date	Дата и время создания привязки	
`customerId`	String	Идентификатор пользователя в RooX UIDM	
`externalUserId`	String	Идентификатор внешнего пользователя	
`externalUser`	ExternalUser	Блок информации о пользователе на стороне ЕСИА или социальной сети	Инициализируется при создании экземпляра класса: `new ExternalUser()`
`id`	String	Уникальный идентификатор привязки. Используется, например, для удаления	
`partnerDataAllowed`	boolean	Признак, разрешил ли пользователь сохранять социальные данные	Инициализируется при создании экземпляра класса: `false`
`partnerId`	String	Идентификатор системы, в которой создана привязка. Для ЕСИА и социальных сетей содержит условное имя	
`realm`	String	Пространство пользователей	
`type`	PartnerType	Тип привязки. Всегда содержит `social`,	Значения: `social` (social network), `extid` (external Identity provider)

Модель данных ExternalUser

Атрибут	Тип данных	Описание	Дополнительная информация
`avatarSmallUrl`	String	Ссылка на уменьшенный аватар пользователя	
`avatarUrl`	String	Ссылка на аватар пользователя	

`email`	String	Адрес электронной почты	
`firstName`	String	Имя пользователя в ЕСИА или социальной сети	
`fullName`	String	Полное имя пользователя в ЕСИА или социальной сети	
`lastName`	String	Фамилия пользователя в ЕСИА или социальной сети	
`middleName`	String	Отчество пользователя в ЕСИА или социальной сети	
`phonenum`	String	Номер телефона	Передается в формате +7(XXX)XXXXXXX, где X — цифра
`profile`	String	Данные профиля	Записанный в строку JSON-объект. В случае ЕСИА используется модель EsiaPersonProfileDto
`userFriends`	String	Список друзей пользователя в социальной сети	
`userId`	String	Идентификатор пользователя в ЕСИА или социальной сети	
`userInfo`	String	Информация о пользователе	Записанный в строку JSON-объект. В случае ЕСИА используется модель UserDataDto

Модель данных EsiaPersonProfileDto

Таблица 2. EsiaPersonProfileDto — базовый тип для профиля данных организаций из ЕСИА

Атрибут	Тип данных	Описание
`inn`	String	10-я копия ИНН
`orgs`	Collection‹ UserOrgDto ›	Сведения об организациях из профиля ЕСИА

`snils`	String	10-я копия СНИЛС

Модель данных UserOrgDto

Атрибут	Тип данных	Описание	Дополнительная информация
`admin`	Boolean	Признак администратора	`true` | `false`
`chief`	Boolean	Признак руководителя	`true` | `false`
`ogrn`	String	Номер ОГРН организации	
`oid`	Long	Идентификатор	Внутренний идентификатор объекта, в том числе пользователя
`shortName`	String	Краткое наименование организации (сокращение/ аббревиатура)	

Модель данных Auth

Атрибут	Тип данных	Описание	Дополнительная информация
`accessToken`	String	Токен авторизации	Выданный токен доступа
`clientId`	String	Идентификатор клиента	`selfcare` (например)
`otpSentCount`	int	Количество отправок одноразового пароля	
`otpTryCount`	int	Количество попыток ввода одноразового пароля	
`otpUnlockDate`	Date	Дата разблокировки отправки одноразовых кодов	
`refreshToken`	String	Обновление токена	Выданный токен обновления доступа
`scope`	String	Список разрешенных scope для использования от имени пользователя, возможные значения задаются конфигурацией	scope (в формате JSON Array)

`secondFactorEnabled`	boolean	Признак подключения второго фактора аутентификации	Включен (да / нет)
`serialVersionUID`	long		

Модель данных UserDataDto

Таблица 3. UserDataDto — базовый тип для профиля данных пользователя из ЕСИА

Атрибут	Тип данных	Описание
`addresses`	AddressesDto[]	Перечень адресов
`birthDate`	LocalDate	Дата рождения (`dd.MM.yyyy`)
`birthPlace`	String	Место рождения
`citizenship`	String	Гражданство
`contacts`	ContactsDto[]	Перечень контактов
`documents`	DocumentsDto[]	Перечень документов, удостоверяющих личность
`firstName`	String	Имя
`gender`	Gender	Пол (`MALE` | `FEMALE`)
`inn`	String	ИНН (номер ИНН физического лица)
`lastName`	String	Фамилия
`middleName`	String	Отчество
`snils`	String	СНИЛС (номер страхового свидетельства)

Модель данных ContactsDto

Атрибут	Тип данных	Описание
`id`	Long	Идентификатор контакта
		–

`type`	ContactType	Тип контакта
`value`	String	Значение
`verified`	String	Статус проверки

Модель данных ContactType

enum	Тип контакта
`NA`	Не определено
`MBT`	Мобильный телефон
`PHN`	Городской телефон
`EML`	Электронная почта

Модель данных AddressesDto

Атрибут	Тип данных	Описание
`additionAreaStreet`	String	Улица на доп. территории
`additionArea`	String	Доп. территория
`addressStr`	String	Наименование улицы
`area`	String	Область
`building`	String	Номер строения
`city`	String	Город
`countryId`	String	Идентификатор страны
`district`	String	Внутригородской район
`fiasCode`	String	Код ФИАС
`flat`	String	Номер квартиры
`frame`	String	Корпус
`house`	String	Номер дома

house	String	Номер дома
id	Long	Идентификатор
region	String	Регион
settlement	String	Посёлок
street	String	Улица
type	AddressType	Тип адреса
value	String	Значение
verified	String	Статус проверки
zipCode	String	Почтовый индекс

Модель данных AddressType

Тип данных	Значения	Описание
enum		Тип адреса
	PLV	Адрес места проживания
	PTA	Адрес временной регистрации
	PRG	Адрес постоянной регистрации
	NA	Отсутствует

Модель данных DocumentDto

Атрибут	Тип данных	Описание
expiryDate	LocalDate	Срок действия документа
firstName	String	Имя
id	Long	Идентификатор
issueDate	LocalDate	Дата выдачи
issueId	String	Код подразделения

issueId	String	Код подразделения
issuedBy	String	Кем выдан (наименование организации)
lastName	String	Фамилия
medicalOrg	String	Наименование медицинской организации
number	String	Номер документа
series	String	Серия
type	DocumentType	Тип документа
unitedNumber	String	Единый номер полиса (ЕНП)
verified	String	Статус проверки

Модель данных DocumentType

Тип данных	Значения	Описание
enum		Тип документа
	NA	не определено
	RF_PASSPORT	Паспорт гражданина РФ
	FID_DOC	Документ иностранного гражданина
	RF_DRIVING_LICENSE	Водительское удостоверение
	MLTR_ID	Военный билет
	FRGN_PASS	Заграничный паспорт
	MDCL_PLCY	полис ОМС
	RF_BRTH_CERT	Свидетельство о рождении (Россия)
	FID_BRTH_CERT	Свидетельство о рождении (другая страна)
	OLD_BRTH_CERT	Свидетельство о рождении (СССР)

Настройка функциональности аутентификации с использованием ЕСИА

Интеграция с ЕСИА выполняется с использованием протоколов OAuth2 и OIDC (OpenID Connect), что делает интеграцию с ЕСИА аналогичной с социальными сетями, использующими эти же протоколы аутентификации и авторизации.

Требования к каналам связи

Для работы функциональности аутентификации с использованием ЕСИА необходимо обеспечить доступность прохождения пакетов в порт `443` до серверов с именами `https://esia-portal1.test.gosuslugi.ru` (тестовый контур) и `https://esia.gosuslugi.ru` (продуктовый контур).

Компоненты, обеспечивающие функциональность аутентификации с использованием ЕСИА

Компоненты RooX UIDM, обеспечивающие функциональность аутентификации с использованием ЕСИА

- `sso-server` (основной сервис аутентификации: выполняет аутентификацию, предоставляет API для аутентификации, самообслуживания и администрирования)
- `federation-webapi` (сервис данных федеративной аутентификации, хранит привязки пользователя к сторонним сервисам и предоставляет API к ним)

Требования к содержимому баз данных для работоспособности функциональности аутентификации с использованием ЕСИА

Для работоспособности функциональности аутентификации с использованием ЕСИА в базе данных сервиса `federation-webapi` в таблице `social_partner` должна существовать запись со значениями, указанными в таблице Значения полей записи таблицы `social_partner` базы данных сервиса `federation-webapi`.

Таблица 4. Значения полей записи таблицы `social_partner` базы данных сервиса `federation-webapi`, необходимые для функциональности аутентификации с использованием ЕСИА

Поле	Значение
`partner_id`	`esia`
`application_id`	Идентификатор экземпляра RooX UIDM, присваиваемый в ходе регистрации его подключения к ЕСИА
`enabled`	`true`
`iframe`	`false` (поле является устаревшим)
`website`	`true`
`request_scope`	Разделенный пробелами перечень областей доступа (scope) ЕСИА из списка предоставляемых ЕСИА

	наборов данных о пользователе (см. Методические указания по использованию Единой системы идентификации и аутентификации). Области доступа определяют сведения, возвращаемые ЕСИА (при наличии разрешения RooX UIDM на доступ к ним)
`social_network_id`	`esia`
`api_url`	Адрес ЕСИА, на который выполняется переход пользователя для аутентификации

Настройки RooX UIDM, влияющие на функциональность аутентификации с использованием ЕСИА

Настройки, необходимые для работы функциональности интеграции UIDM с Единой государственной системой идентификации и аутентификации

`com.rooxteam.federation.esia.app.id`

> Описание
>
> > Идентификатор экземпляра UIDM, выдаваемый при его регистрации в ЕСИА (внутренний идентификатор огранизации в ЕСИА). Значение этой настройки должно совпадать со значением настройки `com.rooxteam.sso.esia.clientId`

`com.rooxteam.sso.esia.users.scopes`

> Описание
>
> > Список разделённых пробелом областей действия (областей доступа, scope), сведения из которых запрашиваются в ЕСИА. ЕСИА использует области действия в качестве указателей на запрашиваемые данные
>
> Возможные значения
>
> > Список предоставляемых ЕСИА областей действия (областей доступа, scope) содержится в Методических рекомендациях по использованию Единой системы идентификации и аутентификации, размещённых на сайте Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации в разделе в разделе «Документы» по направлению Единая система идентификации и аутентификации (ЕСИА)

`com.rooxteam.sso.esia.oauth2.authorization.request.url`

> Описание
>
> > URL ЕСИА для направления кода аутентификации и обмена его на токен доступа (для авторизации по протоколу OAuth2)

`com.rooxteam.sso.esia.clientId`

> Описание
>
> > Идентификатор экземпляра UIDM, выдаваемый при его регистрации в ЕСИА (внутренний идентификатор огранизации в ЕСИА). Значение этой настройки должно совпадать со значением настройки `com.rooxteam.federation.esia.app.id`

`com.rooxteam.sso.esia.cert`

> Описание

> Сертификат электронной подписи, используемый для подписания запросов к ЕСИА. Такой сертификат предварительно регистрируется в ЕСИА и привязывается к учётной записи системы-клиента в ЕСИА

`com.rooxteam.sso.esia.private.key`

> Описание
>
> > Закрытый (приватный, непубличный) ключ ключевой пары сертификата электронной подписи, используемого для подписи запросов к ЕСИА

`com.rooxteam.sso.esia.secretProvider`

> Описание
>
> > Отсутствует

Фронтэнд-реализация аутентификации с использованием ЕСИА

Для начала сценария аутентификации или создания учётной записи пользователя RooX UIDM с помощью учетной записи ЕСИА необходимо обратиться к API RooX UIDM с запросом токена доступа. Из полученного ответа необходимо извлечь URL-адрес (поле `view.esiaRequestUri`), по которому пользователь должен будет перейти на сайт ЕСИА для аутентификации.

НАПОМИНАЕМ

В составе ответа на запрос токена доступа RooX UIDM в поле `esiaRequestUri` возвращает реальный адрес тестового или продуктового контура ЕСИА. При локальной разработке клиентского приложения может потребоваться подмена адреса ЕСИА из поля `esiaRedirectUri` на адрес локальной «заглушки».

Для открытия модального окна ЕСИА можно использовать следующую функцию:

```
window.open(esiaRequestUri, 'Data', 'fullscreen=yes');
```

На странице начала сценария аутентификации необходимо присвоить переменной `window.esiaAuth` функцию (callback), которая будет вызывана после успешной авторизации пользователя в ЕСИА:

Пример

```
const encodeSocialParams = (params) => {
  let result = '';
  for (let param in params) {
    if (params.hasOwnProperty(param)) {
      result += ${param}=${params[param]}&;
    }
  }
  result = result.substr(0, result.length - 1);
  return encodeURIComponent(result);
};

const EsiaAuth = (code) => {
  const socialData = Base64.encode(encodeSocialParams({ code }), false, true);
  // Далее происходит вызов асинхронной функции, которой передаётся объект
  // `socialData` в RooX UIDM вместе с `_eventId: 'esia'` для перехода на следующий
  // шаг сценария
```

```
};
```

Сценарии аутентификации с использованием ЕСИА

ВАЖНО

Пользователь, выполняющий аутентификацию с использованием ЕСИА, должен обладать учётной записью в RooX UIDM.

Сценарии аутентификации с использованием ЕСИА предполагают, что у пользователя существует учётная запись в RooX UIDM, к которой выполняется привязка.

Создание учётной записи RooX UIDM (регистрации) в ходе сценария аутентификации с использованием ЕСИА **не предусмотрено**.

Во время выполнения сценария при направлении запросов в каждом последующем шаге сценария необходимо передавать контекст предыдущего шага — параметр `execution`.

Сценарий аутентификации в RooX UIDM с использованием ЕСИА

Начало сценария аутентификации

Запрос токена доступа (начало сценария аутентификации)

```
POST https://example.com/sso/oauth2/access_token
```

```
client_id=mlk&
client_secret=password&
realm=%2Fcustomer&
grant_type=urn%3Aroox%3Aparams%3Aoauth%3Agrant-type%3Am2m&
service=dispatcher
```

`<client_id>`

> Идентификатор приложения-клиента. Возможные значения зависят от конфигурации.

`<client_secret>`

> Пароль приложения-клиента. Возможные значения зависят от конфигурации.

`<realm>`

> Группа пользователей RooX UIDM. Всегда используется значение `%2Fcustomer`, которое является uri-encoded значением `/customer`.

`<grant_type>`

> Способ авторизации пользователя:
>
> `urn:roox:params:oauth:grant-type:m2m`

> для сценариев получения токена доступа, токена обновления доступа, токена автоматического входа.

`client_credentials`

> для сценария получения приложением-клиентом системного токена.

`<service>`

> Используемый сервис (цепочка аутентификации).

Ответ на запрос токена доступа (начало сценария аутентификации)


```
{
  "execution": "<…>",
  "form": {
    "errors": [],
    "fields": {
      "password": {
        "constraints": [
          {
            "attributes": {
              "max": 1024,
              "min": 1
            },
            "name": "Size"
          },
          {
            "name": "NotNull"
          }
        ]
      },
      "username": {
        "constraints": [
          {
            "attributes": {
              "enabledConfigurationKey": "com.rooxteam.sso.webflow.realm.
{realm}.simple_login_form.login.pattern_enabled",
              "max": 10,
              "min": 10,
              "skip": "(^[^9]+)|([^0-9])"
            },
            "name": "ConfigurableFilteredSize"
          },
          {
            "name": "NotNull"
          }
        ]
      }
    },
    "name": "loginForm"
  },
  "serverUrl": "https://example.com:443/sso/auth/login-widget-router",
  "step": "auth_form",
  "view": {
    "autologin": "skipped",
    "esiaAppId": "771F01",
    "esiaRedirectUri": "/esia_callback.jsp",
    "esiaRequestScopesAsArray": [
      "http://esia.gosuslugi.ru/usr_inf http://esia.gosuslugi.ru/usr_avt"
    ],
    "esiaRequestUri": "https://esia-portal1.test.gosuslugi.ru/aas/oauth2/ac?
state=17e66864-9202-4758-9f96-
```

```
bf0c212cbbda&client_id=OPG&response_type=code&access_type=offline&scope=http%3A%2F%2Fesia.
gosuslugi.ru%2Fusr_inf&timestamp=2022.10.09+22%3A36%3A44+%2B0000&redirect_uri=https%3A%2F%
2Fexample.com%3A443%2Fsso%2Fesia_callback.jsp&client_secret=<…>&display=popup",
    "isBlocked": false,
    "ssoUrl": "https://example.com:443/sso",
  }
}
```

Если пользователь нажимает кнопку **Вход с помощью ЕСИА**, необходимо выполнить переход на адрес, переданный в поле `esiaRequestUri`.

После аутентификации пользователя в учётной записи ЕСИА ЕСИА возвращает пользователю **код авторизации**, который должен быть передан RooX UIDM.

Запрос токена доступа после выполнения аутентификации пользователя в ЕСИА

Запрос токена доступа после выполнения пользователм аутентификации в ЕСИА

```
POST https://example.com/sso/oauth2/access_token
```

```
client_id=mlk&
client_secret=password&
realm=%2fcustomer&
grant_type=urn%3aroox%3aparams%3aoauth%3agrant-type%3am2m&
service=esia&
_eventid=esia&
execution=<…>&
socialdata=<…>
```

`<client_id>`

> Идентификатор приложения-клиента. Возможные значения зависят от конфигурации.

`<client_secret>`

> Пароль приложения-клиента. Возможные значения зависят от конфигурации.

`<realm>`

> Группа пользователей RooX UIDM. Всегда используется значение `%2Fcustomer`, которое является uri-encoded значением `/customer`.

`<grant_type>`

> Способ авторизации пользователя:
>
> `urn:roox:params:oauth:grant-type:m2m`
>
> > для сценариев получения токена доступа, токена обновления доступа, токена автоматического входа.
> >
> > `client_credentials`
> >
> > для сценария получения приложением-клиентом системного токена.

`<service>`

> Используемый сервис (цепочка аутентификации).

`<_eventId>`

> Идентификатор действия (перехода к следующему состоянию сценария). Определяется отдельно для каждого состояния.

`<execution>`

> Идентификатор предсессии аутентификации. В каждом конкретном запросе для продолжения сценария это значение должно быть равно значению поля `execution` из предыдущего ответа сервера на запрос к API.

`<socialData>`

> Результат (список параметров) попытки аутентификации пользователя, полученный от стороннего ресурса и закодированный установленным способом.
>
> Сторонний ресурс возвращает результат попытки аутентификации пользователя в виде JSON. Для получения списка параметров в требуемой форме необходимо представить ответ сервиса в формате `x-www-form-urlencoded`, представить полученную строку в кодировке UTF-8, взять битовое представление строки в кодировке UTF-8, закодировать битовое представление в Base64 (см. пример кода для клиентского веб-приложения).

Ответ об успешной аутентификации с выпуском токенов


```
{
  "access_token": "8db408d1-6a60-4a87-8ef2-5fb0649c53c0",
  "mpt": "65636054-bf02-41ea-889b-e18fc79df488",
  "refresh_token": "8a3bedc0-7f4f-4cb9-8d57-052fcf47a4f6",
  "mpt_expires_in": 59,
  "refresh_expires_in": 599,
  "scope": [
    "cn",
    "externalIdpToken",
    "impersonator"
  ],
  "claims": {
    "cn": "79876543210"
  },
  "token_type": "Bearer",
  "old_token": "8db408d1-6a60-4a87-8ef2-5fb0649c53c0",
  "expires_in": 59,
  "JWTToken":
"eyJraWQiOiJwb2xpY3ljb250ZXh0dG9rZW46andzIiwiY3R5IjoiSldUIiwidHlwIjoiSldUIiwiYWxnIjoiSFMyN
TYifQ.eyJzdWIiOiJiaXNfX19fXzE5OTkxMDAxNTQ5OTg0MyIsImFtciI6ImRldmljZV9zdG9yZWRfdG9rZW4iLCJp
c3MiOiJtbGsiLCJ0b2tlbk5hbWUiOiJpZF90b2tlbiIsImNuIjoiNzk4NzY1NDMyMTAiLCJhdWQiOiJtbGsiLCJhdG
giOjE2NjUzNTUwNzMsImF6cCI6Im1sayIsImNsYWltcyI6eyJjbiI6Ijc5ODc2NTQzMjEwIn0sInJlYWxtIjoiXC9j
dXN0b21lciIsInNjb3BlcyI6WyJpbXBlcnNvbmF0b3IiLCJjbiIsImV4dGVybmFsSWRwVG9rZW4iXSwiZXhwIjoxNj
cwNTM5MDczLCJ0b2tlblR5cGUiOiJFbWJlZGRlZEpXVFRva2VuIiwiaWF0IjoxNjY1MzU1MDczfQ.YbdAHAjgNVVvq
0Xh25d-0E8gzsaMemNWUQe8aTRFqs4"
}
```


Результат сценария аутентификации

В результате успешного выполнения сценария аутентификации с использованием ЕСИА выпущены:

- токен доступа `8db408d1-6a60-4a87-8ef2-5fb0649c53c0`,

- токен обновления доступа `8a3bedc0-7f4f-4cb9-8d57-052fcf47a4f6`,
- мультиплатформенный токен `65636054-bf02-41ea-889b-e18fc79df488`,
- JWT токен.

Сценарий первичной аутентификации в RooX UIDM с использованием ЕСИА (с созданием привязки учётной записи RooX UIDM к идентификатору пользователя ЕСИА)

Начало сценария аутентификации

Запрос токена доступа (начало сценария аутентификации)

```
POST https://example.com/sso/oauth2/access_token
```

```
client_id=mlk&
client_secret=password&
realm=%2Fcustomer&
grant_type=urn%3Aroox%3Aparams%3Aoauth%3Agrant-type%3Am2m&
service=dispatcher
```

`<client_id>`

> Идентификатор приложения-клиента. Возможные значения зависят от конфигурации.

`<client_secret>`

> Пароль приложения-клиента. Возможные значения зависят от конфигурации.

`<realm>`

> Группа пользователей RooX UIDM. Всегда используется значение `%2Fcustomer`, которое является uri-encoded значением `/customer`.

`<grant_type>`

> Способ авторизации пользователя:
>
> `urn:roox:params:oauth:grant-type:m2m`
>
> > для сценариев получения токена доступа, токена обновления доступа, токена автоматического входа.
>
> `client_credentials`
>
> > для сценария получения приложением-клиентом системного токена.

`<service>`

> Используемый сервис (цепочка аутентификации).

Ответ на запрос токена доступа (начало сценария аутентификации)

В ответе на первичный запрос токена доступа возвращается форма ввода логина и пароля.

Поскольку аутентификация с использованием ЕСИА технически идентична аутентификации с использованием любого другого провайдера подлинности по протоколам OAuth2 и OpenID Connect, в составе ответа RooX UIDM возвращаются параметры и для других провайдеров подлинности.

```
HTTP/1.1 200 OK
```


```
{
  "execution": "f16011ef-4c1c-4de2-a6c3-45e4497023f8…",
  "form": {
    "errors": [],
    "fields": {
      "password": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "attributes": {
              "max": 1024,
              "min": 1
            },
            "name": "Size"
          }
        ]
      },
      "username": {
        "constraints": [
          {
            "attributes": {
              "enabledConfigurationKey": "com.rooxteam.sso.webflow.realm.
{realm}.simple_login_form.login.pattern_enabled",
              "max": 10,
              "min": 10,
              "skip": "(^[^9]+)|([^0-9])"
            },
            "name": "ConfigurableFilteredSize"
          },
          {
            "name": "NotNull"
          }
        ]
      }
    },
    "name": "loginForm"
  },
  "serverUrl": "https://example.com:443/sso/auth/login-widget-router",
  "step": "auth_form",
  "view": {
    "autologin": "skipped",
    "esiaAppId": "771F01",
    "esiaRedirectUri": "/esia_callback.jsp",
    "esiaRequestScopesAsArray": [
      "http://esia.gosuslugi.ru/usr_inf http://esia.gosuslugi.ru/usr_avt"
    ],
    "esiaRequestUri": "https://esia-portal1.gosuslugi.ru/aas/oauth2/ac?state=6d896d3a-
f1a7-440a-9c0d-
2008260cfebe&client_id=OPG&response_type=code&access_type=offline&scope=http%3A%2F%2Fesia.
gosuslugi.ru%2Fusr_inf&timestamp=2022.10.03+18%3A23%3A26+%2B0000&redirect_uri=https%3A%2F%
```

```
2Fexample.com%3A443%2Fsso%2Fesia_callback.jsp&client_secret=<…>&display=popup",
    "isBlocked": false,
    "ssoUrl": "https://example.com:443/sso",
  }
}
```

`esiaRequestUri`

> Адрес, на который необходимо перенаправить пользователя для выполнения им аутентификации в ЕСИА. В составе ссылки передаётся параметр адреса (`redirect_uri`), на который необходимо будет перенаправить пользователя после завершения им аутентификации. Также в составе ссылки указывается, какой объём данных (`scope`) будет возвращён из ЕСИА (в данном случае — персональные данные пользователя, `usr_inf`).

Если пользователь нажимает кнопку **Вход с помощью ЕСИА**, необходимо выполнить переход на адрес, переданный в поле `esiaRequestUri` .

После аутентификации пользователя в учётной записи ЕСИА ЕСИА возвращает пользователю **код авторизации**, который должен быть передан RooX UIDM.

Создание привязки к ЕСИА

После получения пользователем кода авторизации ЕСИА клиентское приложение направляет запрос на создание привязки учётной записи пользователя RooX UIDM к идентификатору пользователя ЕСИА.

Запрос создания привязки к ЕСИА

```
POST https://example.com/sso/oauth2/access_token
```

```
client_id=mlk&
client_secret=password&
realm=%2Fcustomer&
grant_type=urn%3Aroox%3Aparams%3Aoauth%3Agrant-type%3Am2m&
service=esia&
_eventId=esia&
execution=<…>&
socialData=<…>
```

`<client_id>`

> Идентификатор приложения-клиента. Возможные значения зависят от конфигурации.

`<client_secret>`

> Пароль приложения-клиента. Возможные значения зависят от конфигурации.

`<realm>`

> Группа пользователей RooX UIDM. Всегда используется значение `%2Fcustomer` , которое является uri-encoded значением `/customer` .

`<grant_type>`

> Способ авторизации пользователя:
>
> `urn:roox:params:oauth:grant-type:m2m`
>
> > для сценариев получения токена доступа, токена обновления доступа, токена автоматического входа.
>
> `client_credentials`
>
> > для сценария получения приложением-клиентом системного токена.

`<service>`

> Используемый сервис (цепочка аутентификации).

`<_eventId>`

> Идентификатор действия (перехода к следующему состоянию сценария). Определяется отдельно для каждого состояния.

`<execution>`

> Идентификатор предсессии аутентификации. В каждом конкретном запросе для продолжения сценария это значение должно быть равно значению поля `execution` из предыдущего ответа сервера на запрос к API.

`<socialData>`

> Результат (список параметров) попытки аутентификации пользователя, полученный от стороннего ресурса и закодированный установленным способом.
>
> Сторонний ресурс возвращает результат попытки аутентификации пользователя в виде JSON. Для получения списка параметров в требуемой форме необходимо представить ответ сервиса в формате `x-www-form-urlencoded`, представить полученную строку в кодировке UTF-8, взять битовое представление строки в кодировке UTF-8, закодировать битовое представление в Base64 (см. пример кода для клиентского веб-приложения).

Код авторизации, полученный пользователем от ЕСИА и переданный RooX UIDM, обменивается RooX UIDM в ЕСИА на токен идентификации пользователя. Из токена идентификации извлекаются сведения об аутентифицировавшемся в ЕСИА пользователе.

Ответ на запрос создания привязки к ЕСИА

```
HTTP/1.1 200 OK
```

```
{
  "execution": "<…>",
  "form": {
    "errors": [],
    "fields": {
      "password": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "attributes": {
              "max": 1024,
```

```
              "min": 1
            },
            "name": "Size"
          }
        ]
      },
      "username": {
        "constraints": [
          {
            "attributes": {
              "enabledConfigurationKey": "com.rooxteam.sso.webflow.realm.
{realm}.simple_login_form.login.pattern_enabled",
              "max": 10,
              "min": 10,
              "skip": "(^[^9]+)|([^0-9])"
            },
            "name": "ConfigurableFilteredSize"
          },
          {
            "name": "NotNull"
          }
        ]
      }
    },
    "name": "loginForm"
  },
  "serverUrl": "https://example.com:443/sso/auth/login-widget-router",
  "step": "auth_form",
  "view": {
    "esiaAppId": "771F01",
    "esiaRedirectUri": "/esia_callback.jsp",
    "esiaRequestScopesAsArray": [
      "http://esia.gosuslugi.ru/usr_inf http://esia.gosuslugi.ru/usr_avt"
    ],
    "esiaRequestUri": "https://esia-portal1.gosuslugi.ru/aas/oauth2/ac?state=d54b0e6b-
fecf-4b3a-8bf8-
e8ca865658ad&client_id=OPG&response_type=code&access_type=offline&scope=http%3A%2F%2Fesia.
gosuslugi.ru%2Fusr_inf&timestamp=2022.10.03+18%3A23%3A38+%2B0000&redirect_uri=https%3A%2F%
2Fexample.com%3A443%2Fsso%2Fesia_callback.jsp&client_secret=<…>&display=popup",
    "firstName": "Имя030",
    "fullName": "Имя030 Отчество030 Фамилия030",
    "isBlocked": false,
    "socialNetworkId": "esia",
    "ssoUrl": "https://example.com:443/sso",
  }
}
```


`execution`

> Идентификатор предсессии аутентификации.

Аутентификация пользователя в учётной записи RooX UIDM для создания привязки к этой учётной записи

Запрос аутентификации в учётной записи RooX UIDM с именем учётной записи `9876543210` и паролем `password`

```
POST https://example.com/sso/oauth2/access_token
```

```
client_id=mlk&
client_secret=password&
realm=%2Fcustomer&
grant_type=urn%3Aroox%3Aparams%3Aoauth%3Agrant-type%3Am2m&
service=dispatcher&
_eventId=next&
username=9876543210&
password=password&
execution=<…>
```

`<client_id>`

> Идентификатор приложения-клиента. Возможные значения зависят от конфигурации.

`<client_secret>`

> Пароль приложения-клиента. Возможные значения зависят от конфигурации.

`<realm>`

> Группа пользователей RooX UIDM. Всегда используется значение `%2Fcustomer`, которое является uri-encoded значением `/customer`.

`<grant_type>`

> Способ авторизации пользователя:
>
> `urn:roox:params:oauth:grant-type:m2m`
>
> > для сценариев получения токена доступа, токена обновления доступа, токена автоматического входа.
>
> `client_credentials`
>
> > для сценария получения приложением-клиентом системного токена.

`<service>`

> Используемый сервис (цепочка аутентификации).

`<_eventId>`

> Идентификатор действия (перехода к следующему состоянию сценария). Определяется отдельно для каждого состояния.

`<username>`

> Логин пользователя.

`<password>`

> Пароль пользователя.

`<execution>`

> Идентификатор предсессии аутентификации. В каждом конкретном запросе для продолжения сценария это значение должно быть равно значению поля `execution` из предыдущего ответа сервера на запрос к API.

Ответ об успешной аутентификации в учётной записи RooX UIDM с именем учётной записи `9876543210` и паролем `password` и привязке идентификатора ЕСИА

```
HTTP/1.1 200 OK
```


```
{
  "execution": "<…>",
  "view": {
    "socialNetworkId": "esia",
    "firstName": "Имя030",
    "fullName": "Имя030 Отчество030 Фамилия030",
    "step": "attach_form"
  },
  "form": {
    "name": "attachForm",
    "fields": {},
    "errors": []
  },
  "serverUrl": "https://example.com:443/sso/auth/social-attach",
  "step": "show_attach_form"
}
```


Запрос токена доступа после выполнения привязки учётной записи RooX UIDM к идентификатору ЕСИА

Запрос токена доступа

```
POST https://example.com/sso/oauth2/access_token
```

```
client_id=mlk&
client_secret=password&
realm=%2Fcustomer&
grant_type=urn%3Aroox%3Aparams%3Aoauth%3Agrant-type%3Am2m&
service=dispatcher&
_eventId=next&
execution=<…>
```

Ответ на запрос токена доступа

```
HTTP/1.1 200 OK
```


```
{
  "access_token": "9eb8528f-90a5-41d8-8739-cf0bc4ef9be8",
```

```
  "mpt": "0cd7a297-c195-4cd3-9c8d-7d11f34a3ce3",
  "refresh_token": "95d28a74-bd27-4cae-80f8-326c6d1e9246",
  "mpt_expires_in": 59,
  "refresh_expires_in": 599,
  "scope": [
    "cn",
    "externalIdpToken",
    "impersonator"
  ],
  "claims": {
    "cn": "79876543210"
  },
  "token_type": "Bearer",
  "old_token": "9eb8528f-90a5-41d8-8739-cf0bc4ef9be8",
  "expires_in": 59,
  "JWTToken": "<…>"
}
```


Результат сценария аутентификации

В результате успешного выполнения сценария аутентификации с использованием ЕСИА выпущены:

- токен доступа `9eb8528f-90a5-41d8-8739-cf0bc4ef9be8`,
- токен обновления доступа `95d28a74-bd27-4cae-80f8-326c6d1e9246`,
- мультиплатформенный токен `0cd7a297-c195-4cd3-9c8d-7d11f34a3ce3`,
- JWT-токен.

Управление привязками учётной записи пользователя к ЕСИА

Получение списка привязок учётной записи пользователя к ЕСИА

Запрос поиска привязки к ЕСИА

```
GET https://example.com/sso/federation-webapi-2.0/customers/@me/partnerMappings
```

`@me`

> Псевдоидентификатор, указывающий, что запрос выполняется от имени текущего авторизованного пользователя. Пользователь определяется на основе токена доступа.

Запрос возвращает список всех существующих привязок учётной записи RooX UIDM к идентификаторам в ЕСИА, социальных сетях или сервисах.

Успешный ответ о наличии привязки к ЕСИА

```
HTTP/1.1 200 OK
```


```
[
```

```
  {
    "id": "sso_____8a5f0c46-0f1e-4a5a-a970-c4e66a0fe3c0",
    "type": "social",
    "partnerId": "esia",
    "externalUserId": "1000321821",
    "externalUser": {
      "userId": "1000321821",
      "userInfo": "
{\"firstName\":\"Имя030\",\"lastName\":\"Фамилия030\",\"middleName\":\"Отчество030\",\"sni
ls\":\"000-000-600
30\",\"gender\":\"M\",\"citizenship\":\"RUS\",\"birthDate\":\"30.01.1998\",\"birthPlace\":
\"Общая тестовая УЗ! НЕ изменяйте данные! НЕ изменяйте пароль!\"}",
      "firstName": "Имя030",
      "lastName": "Фамилия030",
      "middleName": "Отчество030",
      "fullName": "Имя030 Отчество030 Фамилия030",
      "profile": "{\"snils\":\"000-000-600 30\"}"
    },
    "auth": {
      "clientId": "771F01",
      "accessToken": "<…>",
      "otpSentCount": 0,
      "otpTryCount": 0
    },
    "created": "2022-10-03T18:24:38Z",
    "customerId": "bis_____199910015499843",
    "partnerDataAllowed": true,
    "messagePostingAllowed": false,
    "updated": "2022-10-03T18:24:38Z",
    "realm": "customer",
    "enabled": true
  }
]
```


В каждом из элементов возвращаемого списка привязок:

- поле `id` указывает на идентификатор привязки в RooX UIDM,
- поле `partnerId` указывает на ЕСИА, социальную сеть или сервис,
- поле `externalUserId` указывает на идентификатор пользователя RooX UIDM в ЕСИА, социальной сети или сервисе.

Удаление привязки учётной записи RooX UIDM к ЕСИА

Запрос удаления привязок к ЕСИА из учётной записи пользователя в RooX UIDM

```
DELETE https://example.com/sso/federation-webapi-2.0/customers/@me/partnerMappings?
partnerId=esia
```

Будут удалены привязки текущего пользователя (`@me`) к сервису ЕСИА.

Успешный ответ на запрос удаления привязок к ЕСИА из учётной записи пользователя в RooX UIDM

```
HTTP/1.1 200 OK
```

```
[
  {
    "id": "sso_____8a5f0c46-0f1e-4a5a-a970-c4e66a0fe3c0",
    "type": "social",
    "partnerId": "esia",
    "externalUserId": "1000303233",
    "externalUser": {
      "userId": "1000303233",
      "userInfo": "
{\"firstName\":\"Денис\",\"lastName\":\"Фамилия006\",\"middleName\":\"Отчество006\",\"snil
s\":\"000-000-600
06\",\"inn\":\"335669961450\",\"gender\":\"M\",\"citizenship\":\"RUS\",\"birthDate\":\"13.
07.1985\",\"birthPlace\":\"Челябинск\"}",
      "firstName": "Денис",
      "lastName": "Фамилия006",
      "middleName": "Отчество006",
      "fullName": "Денис Отчество006 Фамилия006",
      "profile": "{\"inn\":\"335669961450\",\"snils\":\"000-000-600 06\"}"
    },
    "auth": {
      "clientId": "771F01",
      "accessToken": "<…>",
      "secondFactorEnabled": false,
      "otpSentCount": 0,
      "otpTryCount": 0
    },
    "created": "2022-10-09T22:37:35Z",
    "customerId": "bis_____199910015499843",
    "partnerDataAllowed": true,
    "messagePostingAllowed": false,
    "updated": "2022-10-09T22:37:53Z",
    "realm": "customer",
    "enabled": true
  }
]
```


В результате запроса возвращается список удалённых привязок.

В случае отсутствия привязок к заданными характеристиками возвращается пустой список (`[]`).

Получение информации о токене (интроспекция токена)

Некоторые сведения о токене доступа можно получить, направив запрос на эндпоинт `/sso/oauth2/tokeninfo` с предоставлением токена, информацию о котором необходимо получить.

Запрос сведений о токене с предоставлением токена

```
POST https://example.com/sso/oauth2/tokeninfo?access_token=9eb8528f-90a5-41d8-8739-
cf0bc4ef9be8
```

Успешный ответ на запрос сведений о токене

```
HTTP/1.1 200 OK
```

```
{
  "sub": "bis_____199910015499843",
  "auth_level": "5",
  "amr": [
    "urn:uidm:esia:pwd"
  ],
  "roles": [
    "ROLE_CUSTOMER"
  ],
  "ext_sub": "199910015499843",
  "cn": "9876543210",
  "token_type": "Bearer",
  "client_id": "mlk",
  "access_token": "9eb8528f-90a5-41d8-8739-cf0bc4ef9be8",
  "executionId": "4326ced9-5d03-40c4-9448-aed3a6e6cea2",
  "scope": [
    "impersonator",
    "cn",
    "externalIdpToken"
  ],
  "auth_time": 1664821479,
  "realm": "/customer",
  "authType": "social_esia",
  "expires_in": 59,
  "jti": "9eb8528f-90a5-41d8-8739-cf0bc4ef9be8"
}
```


API получения сведений о пользователях из внешних источников данных

RooX UIDM посредством External Identity Provider API предоставляет возможности поиска:

- учетной записи пользователя по его идентификатору в ЕСИА,
- учетной записи пользователя по предоставленному им токену доступа.

Если ответ RooX UIDM о наличии привязок учётной записи пользователя к ЕСИА положителен, то External Identity Provider API может использоваться клиентским приложением для получения дополнительных данных о пользователе из ЕСИА.

События, протоколируемые в СУБД аудита в ходе сценария аутентификации с использованием ЕСИА

`sso.auth.success`

> Успешная попытка аутентификации пользователя в UIDM

`sso.auth.fail`

> Неудачная попытка аутентификации пользователя в UIDM

`sso.auth.revoke`

> Принудительное завершение сессии

`sso.auth.logout`

> Инвалидация сессии пользователя

`sso.auth.auto.fail`

> Неуспешная попытка аутентификации с использованием токена автоматического входа

`sso.auth.autologin.token.created`

> Успешная аутентификация с использованием токена автоматического входа (с последующим выпуском токена доступа, токена обновления доступа)

`sso.auth.token.lifetime.end`

> Завершение сессии по таймауту

`sso.auth.increase.success`

> Успешное повышение уровня доступа в текущей сессии

`sso.auth.increase.fail`

> Неуспешная попытка повышения уровня доступа в текущей сессии

`sso.api.session.delete.success`

> Удаление сессии через API управления сессиями (`/sessionlist`) или АРМ Контакт-Центра

`sso.api.session.delete.fail`

> Неуспешное удаление сессии через API управления сессиями (`/sessionlist`) или АРМ Контакт-Центра

`sso.auth.preauth.success`

> Попытки аутентификации пользователей с использованием логина и пароля в UIDM, которые завершились успешно (сам процесс продолжается; следующим шагом можеть быть ввод одноразового пароля). Токен доступа еще не выпущен, так что это событие можно использовать только для предварительных действий, например «прогрева» кеша бизнес-данных.

`webapi.social.mapping.create.success`

> Успешное создание привязки

`webapi.social.mapping.create.fail`

> Неуспешное создание привязки

`webapi.social.mapping.delete.success`

> Успешное удаление привязки

`webapi.social.mapping.delete.fail`

> Неуспешное удаление привязки

Выборка из БД аудита событый, относящихся к функциональности аутентификации с использованием ЕСИА

При помощи SQL-запроса можно произвести выборку из БД аудита событий, относящихся к успешной аутентификации или другим событиям после заданных даты и времени.

```
SELECT "authtype",
       Count(*),
       service,
       url,
       To_char("timeend", 'YYYY-MM')
FROM   "operationaudit"
WHERE  "name" = 'sso.auth.get_access_token.success'
       AND "timeend" > '2022-08-01 00:00:00'
GROUP  BY "authtype",
          service,
          url,
          To_char("timeend", 'YYYY-MM')
ORDER  BY To_char("timeend", 'YYYY-MM'),
          "authtype",
          service;
```

В поле `authType` результата выполнения такого запроса будет указано на способ аутентификации, использовавшийся для выпуска токена доступа (для события `sso.auth.get_access_token.success` в примере будут выбраны в том числе способы аутентификации `esia`, `login-by-esia`, `registration-by-esia`, `social-esia`, `social-validate`).

Мониторинг и метрики функциональности аутентификации с использованием ЕСИА

Адреса сбора метрик

Общий вид запроса метрик

```
POST https://{sso_host}:{prometheus_port}/{query}
```

`{sso_host}`

> Имя хоста или IP-адрес сервера RooX UIDM.
>
> Конечная точка мониторинга не доступна на балансировщике нагрузки; используйте внутренние сетевые адреса.

В каждом запросе метрик необходимо использовать Basic-авторизацию:

```
Authorization: Basic dGVzdGxvZ2luOnRlc3RwYXNzd29yZA==
```

Адреса сбора метрик работы сервисов `sso-server` и `federation-webapi` (порты указаны по умолчанию)

`sso-server`	`http://{sso_host}:15012`
`federation-webapi`	`http://{sso_host}:13212`
`{sso_host}`	Имя хоста или IP-адрес сервера RooX UIDM. Конечная точка мониторинга не доступна на балансировщике нагрузки; используйте

внутренние сетевые адреса.

Порты по умолчанию, по которым доступны метрики, указаны в статье о мониторинге и метриках, и могут быть изменены настройками RooX UIDM.

Метрики функциональности аутентификации с использованием ЕСИА

Работа API аутентификации

Имя метрики

`rx_authnz_endpoint`

Устаревшее имя метрики

`com.rooxteam.monitoring.filter.StatisticalMonitoringFilter...*`

Метрика отображает время ответа API аутентификации и авторизации в миллисекундах (мс).

Таблица 5. Срезы

Label	Срез	Примеры значений
`apigroup`	Тип API	`authentication`, `authorization`
`api`	Тип API	`oauth1`, `oauth2`, `mlk`
`status`	Результат операции	`success`, `error`