

**Система управления идентификацией и доступом «AtomID»**

**РУКОВОДСТВО АДМИНИСТРАТОРА**

**часть 3**

**Служба авторизации**

|  |  |  |
| --- | --- | --- |
| **УТВЕРЖДАЮ** |  | **УТВЕРЖДАЮ** |
| Директор департамента  информационных технологий  ГК «Росатом» |  | Должность |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.М. Абакумов |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО |
| «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2022\_ г. |  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2022\_ г. |

**Система управления идентификацией и доступом «AtomID»**

**РУКОВОДСТВО АДМИНИСТРАТОРА часть 3**

**Служба авторизации**

**RU.ЕРТД.00101-01 32 01**

на 65 листах

**Версия 1.2**

**Москва**

**2022**

**Лист согласования**

|  |  |
| --- | --- |
| Должность и организация | ФИО |
| Должность  Отдел  Предприятие | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Ф.И.О./  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022\_г. |
| Должность  Отдел  Предприятие | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Ф.И.О. /  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022\_г. |
| Должность  Отдел  Предприятие | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Ф.И.О. /  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022\_г. |
| Должность  Отдел  Предприятие | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Ф.И.О. /  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022\_г. |
| Должность  Отдел  Предприятие | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Ф.И.О. /  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022\_г. |
| Должность  Отдел  Предприятие | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Ф.И.О. /  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022\_г. |
| Должность  Отдел  Предприятие | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Ф.И.О./  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022\_г. |

Лист регистрации изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Версия** | **Дата** | **Автор** | **Вносимые изменения** |
| 1.0 | 20.05.2022 | Шубинская О.А. | Создание документа |
| 1.1 | 26.05.2022 | Шубинская О.А. | Дополнение в части сервисов авторизации |
| 1.2 | 03.06.2022 | Шубинская О.А. | Форматирование, корректировка нумерации |
|  |  |  |  |

**Содержание**

[Лист регистрации изменений 7](#_Toc104485792)

[Аннотация 9](#_Toc104485793)

[Термины и определения 10](#_Toc104485794)

[1 Общие сведения 12](#_Toc104485795)

[1.1 Взаимодействие с пользователями/администраторами и смежными системами АтомID 12](#_Toc104485796)

[2 Служба авторизации 14](#_Toc104485797)

[2.1 Обзор служб авторизации 14](#_Toc104485798)

[2.2 Управление серверами ресурсов 21](#_Toc104485799)

[2.3 Управление ресурсами и областями безопасности 26](#_Toc104485800)

[2.4 Управление политиками 29](#_Toc104485801)

[2.5 Управление разрешениями 45](#_Toc104485802)

[2.6 Оценка и тестирование политик 49](#_Toc104485803)

[2.7 Сервис авторизации 50](#_Toc104485804)

Аннотация

Настоящий документ является руководством администратора системы управления идентификацией и доступом «AtomID» (далее- AtomID, система).

Документ предназначен для Системных администраторов, обслуживающих систему, которые должны иметь образование в области системного администрирования и в области защиты информации соответственно.

В документе приводится описание:

* Общих сведений о системе;
* Описание работы службы авторизации.

Термины и определения

В настоящем документе применялись следующие термины:

|  |  |  |
| --- | --- | --- |
| Администратор | — | пользователь, уполномоченный выполнять некоторые действия по администрированию системы защиты информации АтомID в рамках функциональных обязанностей |
| Идентификатор | — | обезличенное программное представление уникальных идентификационных данных субъекта доступа, позволяющее выполнить процесс его идентификации. Идентификатором в Изделии является имя пользователя субъекта доступа |
| Пользователь | — | лицо, которому разрешено выполнять некоторые действия (операции) по обработке информации в информационной системе или использующее результаты ее функционирования |
| Роль | — | предопределенная совокупность правил, устанавливающих допустимое взаимодействие между субъектом и объектом доступа |
| Сервер | — | выделенный или специализированный компьютер, предоставляющий вычислительные ресурсы для функционирования программного обеспечения |
| ID | — | (англ. identifier) уникальный признак объекта, позволяющий отличать его от других объектов, то есть идентифицировать |
| JWT | — | (англ. JSON web-token) веб-токен JSON |
| OAuth2 | — | открытый протокол (схема) авторизации, обеспечивающий предоставление третьей стороне ограниченный доступ к защищённым ресурсам пользователя без передачи ей (третьей стороне) логина и пароля |
| LDAP | — | (англ. lightweight directory access protocol) протокол хранения данных об организации, каталогах, пользователях, позволяющий осуществлять аутентификацию |
| PAP | — | (англ. policy administration point) пункт администрирования политики |
| PDP | — | (англ. policy decision point) пункт принятия решений по политике |
| PEP | — | (англ. policy enforcement point) пункт обеспечения соблюдения политики |
| PIP | — | (англ. policy information point) информационный центр по вопросам политики |
| RBAC | — | (англ. role-based access control) управление доступом на основе ролей |
| RESTful | — | (англ. representational state transfer) это способ написания сервисов с использованием архитектуры REST. RESTful services предоставляет ресурсы для определения целей взаимодействия с клиентами |
| RPT | — | (англ. Requesting Party Token) токен запрашиваемой стороны |
| URI | — | (англ. uniform resource identifier) унифицированный (единообразный) идентификатор ресурса |

1. Общие сведения

Система управления идентификацией и доступом АтомID является программным средством обеспечения безопасности информационных технологий. Система применяется для защиты информации в значимых объектах критической информационной инфраструктуры 3 категории, в государственных информационных системах 3 класса защищенности, в автоматизированных системах управления производственными и технологическими процессами 3 класса защищенности, а также для обеспечения до 3 уровня защищенности персональных данных в информационных системах, для которых к актуальным отнесены угрозы 3-го типа.

Взаимодействие с пользователями/администраторами и смежными системами АтомID

AtomID реализует технологию единого входа (Single Sign-On) для управления идентификацией и доступом пользователей.

Общая схема работы АтомID представлена на рисунке ниже (Рисунок 1).

В АтомID реализованы следующие функции по защите информации:

* идентификация и аутентификация пользователей;
* ролевое разграничение доступа;
* регистрация событий безопасности;
* обеспечение доступности информации.

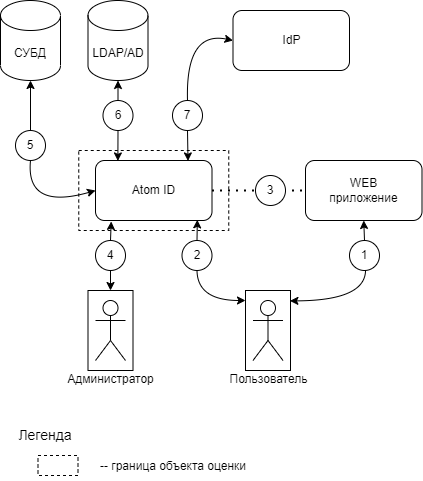


Рисунок 1 - Схема работы АтомID

АтомID в соответствии с схемой работы, представленной на рисунке (Рисунок 1), имеет следующие возможности по взаимодействию с пользователями/администраторами и смежными системами:

1. Администратор может настраивать AtomID с использованием веб-интерфейса АтомID или административного API (4);
2. Учётные записи пользователей могут быть созданы вручную администратором, путем импорта из каталогов LDAP/AD (6), в ходе самостоятельной регистрации;
3. Собственные данные (УЗ, группы, настройки клиентов и пр) AtomID хранит в СУБД (5);
4. Пользователь обращается к защищаемому приложению (1). Если пользователь не прошел аутентификацию, происходит перенаправление на AtomID (2);
5. Пользователь автоматически переходит в веб-интерфейс AtomID. Далее, в зависимости от настроек АтомID, пользователь может ввести свои логин и пароль, осуществить вход через стороннего провайдера идентификации по протоколам SAML 2.0, OIDC или SPNEGO/Kerberos (7);
6. В случае успешной аутентификации, происходит обмен служебной информацией между AtomID и веб-приложением (3), пользователь становится авторизованным.
7. Служба авторизации
   1. Обзор служб авторизации

AtomID поддерживает детализированные политики авторизации и способен комбинировать различные механизмы контроля доступа, такие как:

* управление доступом на основе атрибутов (ABAC);
* управление доступом на основе ролей (RBAC);
* пользовательский контроль доступа (UAC);
* управление доступом на основе контекста (CBAC);
* управление доступом на основе правил;

1. использование JavaScript;

* контроль доступа на основе времени;
* поддержка пользовательских механизмов управления доступом (ACMs) через интерфейс поставщика услуг (SPI).

AtomID основан на наборе административных пользовательских интерфейсов и RESTful API и предоставляет необходимые средства для создания разрешений для защищенных ресурсов и областей, связывания этих разрешений с политиками авторизации и обеспечения выполнения решений об авторизации в ваших приложениях и службах.

Серверы ресурсов (приложения или службы, обслуживающие защищенные ресурсы) обычно полагаются на некоторую информацию, чтобы решить, следует ли предоставлять доступ к защищенному ресурсу. Для серверов ресурсов на основе RESTful эта информация обычно получается из токена безопасности, отправляемого в качестве токена-носителя при каждом запросе к серверу. Для веб-приложений, которые полагаются на сеанс для аутентификации пользователей, эта информация обычно хранится в сеансе пользователя и извлекается оттуда для каждого запроса.

Часто серверы ресурсов принимают решения об авторизации только на основе управления доступом на основе ролей (RBAC), где роли, предоставленные пользователю, пытающемуся получить доступ к защищенным ресурсам, сверяются с ролями, сопоставленными с этими же ресурсами.

Роли используются приложениями и имеют есть несколько ограничений:

* ресурсы и роли тесно связаны, и изменения ролей (например, добавление, удаление или изменение контекста доступа) могут повлиять на несколько ресурсов;
* изменения в требованиях к безопасности могут повлечь за собой глубокие изменения в коде приложения, отражающие эти изменения;
* в зависимости от размера вашего приложения управление ролями может стать сложным и подверженным ошибкам;
* это не самый гибкий механизм контроля доступа. Роли не отражают того, кто вы есть, и не содержат контекстуальной информации. Если была предоставлена роль, у вас есть хотя бы некоторый доступ.

Учитывая, что сегодня необходимо принимать во внимание разнородные среды, в которых пользователи распределены по разным регионам, с различными локальными политиками, используют разные устройства с высоким спросом на обмен информацией, службы авторизации AtomID могут улучшить возможности авторизации ваших приложений и служб, предоставляя:

* защиту ресурсов с использованием детализированных политик авторизации и различных механизмов контроля доступа;
* централизованное управление ресурсами, разрешениями и политиками;
* централизованный пункт принятия решений по политике;
* безопасность REST включает в себя множество служб авторизации на основе REST;
* рабочие процессы авторизации и доступ, управляемый пользователем;
* инфраструктуру, помогающую избежать репликации кода между проектами (и повторного развертывания), и быстро адаптироваться к изменениям в ваших требованиях к безопасности.
  + 1. Архитектура

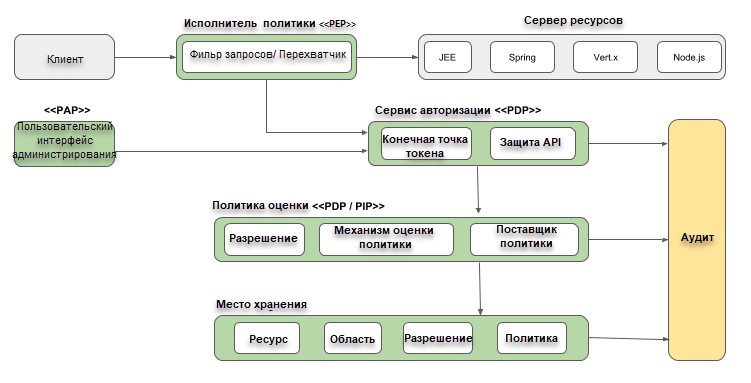


Рисунок 2 - Архитектура

С точки зрения проектирования Службы авторизации основаны на четко определенном наборе шаблонов авторизации, обеспечивающих эти возможности (Рисунок 2):

1. Пункт администрирования политики (PAP). Предоставляет набор пользовательских интерфейсов на основе консоли администрирования AtomID для управления серверами ресурсов, ресурсами, областями, разрешениями и политиками. Часть этого также выполняется удаленно с помощью API защиты;
2. Пункт принятия решений по политике (PDP). Предоставляет распространяемую точку принятия решений о политике, куда отправляются запросы на авторизацию, и политики оцениваются в соответствии с запрашиваемыми разрешениями. Дополнительные сведения приведены в разделе 2.5 «Управление разрешениями»;
3. Пункт обеспечения соблюдения политики (PEP). Предоставляет реализации для различных сред с целью фактического применения решений об авторизации на стороне сервера ресурсов. AtomID предоставляет некоторые встроенные средства обеспечения соблюдения политики;
4. Информационный центр по вопросам политики (PIP). Ввиду использования в качестве основы сервер аутентификации AtomID, существует возможность получения атрибутов из удостоверений и среды выполнения во время оценки политик авторизации.
   * + 1. Процесс авторизации

Три основных процесса определяют необходимые шаги для понимания того, как использовать AtomID для обеспечения детальной авторизации ваших приложений:

* управление ресурсами;
* управление разрешениями и политиками;
* обеспечение соблюдения политики.

**Управление ресурсами.**

Управление ресурсами включает в себя шаги (Рисунок 3) для определения того, что защищается.

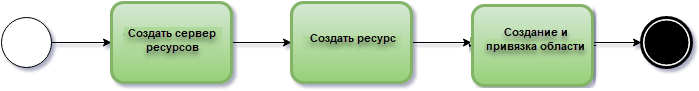


Рисунок 3 – Шаги для определения объекта защиты

Во-первых, нужно указать AtomID, который вы хотите защитить, который обычно представляет собой веб-приложение или набор из одной или нескольких служб. Дополнительные сведения о серверах ресурсов приведены в разделе 2.1.2 «Терминология».

Управление серверами ресурсов (Рисунок 4) осуществляется с помощью консоли администрирования AtomID. Там существует возможность включить любое зарегистрированное клиентское приложение в качестве сервера ресурсов и начать управлять ресурсами и областями, которые необходимо защитить.



Рисунок 4 – Управление серверами ресурсов

Ресурс может быть веб-страницей, ресурсом RESTful, файлом в вашей файловой системе, EJB и так далее. Они могут представлять группу ресурсов (точно так же, как класс в Java), или они могут представлять один и конкретный ресурс.

Например, может быть ресурс банковского счета, представляющий все банковские счета, и есть возможность использовать его для определения политик авторизации, общих для всех банковских счетов. Но должна быть возможность определения конкретных политик для учетной записи (экземпляра ресурса, принадлежащего клиенту), где только владельцу разрешен доступ к некоторой информации или выполнение операции.

Ресурсами можно управлять с помощью консоли администрирования AtomID или API защиты. В последнем случае серверы ресурсов могут удаленно управлять своими ресурсами.

Области обычно представляют действия, которые могут быть выполнены с ресурсом, но этим они не ограничиваются. Вы также можете использовать области для представления одного или нескольких атрибутов внутри ресурса.

**Управление разрешениями и политиками.**

После того, как определены сервер ресурсов и список ресурсов, которые нужно защитить, необходимо настроить разрешения и политики.

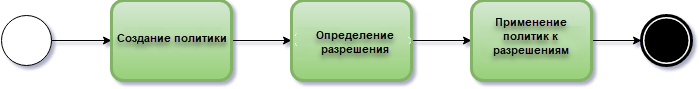


Рисунок 5 - Шаги по определению требований к безопасности и доступу

Этот процесс включает в себя все необходимые шаги (Рисунок 5) для фактического определения требований к безопасности и доступу, которые управляют ресурсами.

Политики определяют условия, которые должны быть выполнены для доступа или выполнения операций с чем-либо (ресурсом или областью действия), но они не привязаны к тому, что они защищают. Они являются общими и могут быть повторно использованы для создания разрешений или более сложных политик.

Например, чтобы разрешить доступ к группе ресурсов только пользователям, которым предоставлена роль «Премиум-пользователь», вы можете использовать RBAC (Управление доступом на основе ролей).

AtomID предоставляет несколько встроенных типов политик (и их соответствующих поставщиков политик), охватывающих наиболее распространенные механизмы контроля доступа. Существует возможность создавать политики на основе правил, написанных с использованием JavaScript.

После создания политики, необходимо приступить к определению разрешений. Разрешения связаны с ресурсом, который они защищают. На данном шаге необходимо указать, объект защиты (ресурс или область), и политики, которые должны быть выполнены, чтобы предоставить или отклонить разрешение.

**Обеспечение соблюдения политики.**

Обеспечение соблюдения политики включает в себя необходимые шаги для фактического применения решений об авторизации к серверу ресурсов. Это достигается путем включения точки применения политики или PEP (Рисунок 6) на сервере ресурсов, которая способна взаимодействовать с сервером авторизации, запрашивать данные авторизации и управлять доступом к защищенным ресурсам на основе решений и разрешений, возвращаемых сервером.

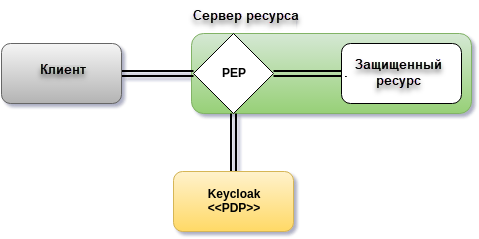


Рисунок 6 – Обеспечение соблюдения политики

AtomID предоставляет некоторые встроенные реализации средств обеспечения соблюдения политик, которые можно использовать для защиты приложений в зависимости от платформы, на которой они работают.

* + - 1. Службы авторизации

Службы авторизации состоят из следующих конечных точек RESTful:

* конечная точка токена;
* конечная точка управления ресурсами;
* конечная точка управления разрешениями.

Каждая из этих служб предоставляет определенный API, охватывающий различные этапы, связанные с процессом авторизации.

**Конечная точка токена.**

Клиенты OAuth2 (например, интерфейсные приложения) могут получать токены доступа с сервера, используя конечную точку токена, и использовать эти же токены для доступа к ресурсам, защищенным сервером ресурсов (например, внутренним службам). Аналогичным образом, службы авторизации AtomID предоставляют расширения для OAuth2, позволяющие выдавать токены доступа на основе обработки всех политик, связанных с запрашиваемым ресурсом (ресурсами) или областью (областями).

Это означает, что серверы ресурсов могут принудительно предоставлять доступ к своим защищенным ресурсам на основе разрешений, предоставленных сервером и удерживаемых токеном доступа. В службах авторизации AtomID токен доступа с разрешениями называется «Токеном запрашивающей стороны» или сокращенно RPT.

Дополнительные ресурсы:

* получение разрешений.

**API защиты.**

API защиты представляет собой набор UMA-совместимых операций по предоставлению конечных точек для серверов ресурсов, которые помогают им управлять своими ресурсами, областями, разрешениями и политиками, связанными с ними. Только серверам ресурсов разрешен доступ к этому API, для которого также требуется область uma\_protection.

Операции, предоставляемые API защиты, могут быть организованы в две основные группы:

* управление ресурсами:

1. создание ресурса;
2. удаление ресурса;
3. поиск по идентификатору;
4. запрос;

* управление разрешениями:

1. выдача тикетов на получение разрешений.

По умолчанию включено удаленное управление ресурсами. Для изменения удаленного управления ресурсами необходимо в консоли администрирования AtomID установить разрешение управления ресурсами только через консоль.

При использовании протокола UMA выдача разрешений с помощью Protection API является важной частью всего процесса авторизации. Как описано в следующем разделе, они представляют разрешения, запрашиваемые клиентом и отправляемые на сервер для получения окончательного токена со всеми разрешениями, предоставленными во время оценки разрешений и политик, связанных с запрашиваемыми ресурсами и областями.

Дополнительные ресурсы:

* API защиты.
  + 1. Терминология

Ниже перечислены термины и понятия, введенные службами авторизации AtomID:

1. Сервер ресурсов. Согласно терминологии OAuth2, сервер ресурсов — это сервер, на котором размещены защищенные ресурсы, и который способен принимать запросы на защищенные ресурсы и отвечать на них.

Серверы ресурсов обычно опираются на некоторую информацию, чтобы решить, следует ли предоставлять доступ к защищенному ресурсу. Для серверов ресурсов на основе RESTful эта информация обычно передается в токене безопасности, который обычно отправляется в качестве токена-носителя вместе с каждым запросом к серверу. Веб-приложения, которые полагаются на сеанс для аутентификации пользователей, обычно хранят эту информацию в сеансе пользователя и извлекают ее оттуда для каждого запроса.

В AtomID любое конфиденциальное клиентское приложение может выступать в качестве сервера ресурсов. Ресурсы этого клиента и их соответствующие области защищены и регулируются набором политик авторизации.

1. Ресурс.Ресурс — это часть ресурсов приложения и организации. Это может быть набор из одной или нескольких конечных точек, классический веб-ресурс (например, HTML-страница), и прочее. В терминологии политики авторизации ресурс — это защищаемый объект.

Каждый ресурс имеет уникальный идентификатор, который может представлять отдельный ресурс или набор ресурсов. Например, вы можете управлять ресурсом банковского счета, который представляет и определяет набор политик авторизации для всех банковских счетов. Но у вас также может быть другой ресурс банковского счета с другим именем, который представляет собой единый ресурс, принадлежащий одному клиенту, имеющий свой собственный набор политик авторизации.

1. Область.Область действия ресурса — это ограниченная степень доступа, которую можно выполнить к ресурсу. В терминологии политики авторизации область действия — это один из потенциально многих действий, которые могут логически применяться к ресурсу.

Обычно оно указывает, что можно сделать с данным ресурсом. Примерами областей являются просмотр, редактирование, удаление и так далее. Однако область применения также может быть связана с конкретной информацией, предоставляемой ресурсом. В этом случае это может быть ресурс проекта и область затрат, где область затрат используется для определения конкретных политик и разрешений для пользователей на доступ к стоимости проекта.

1. Разрешение.Разрешение связывает защищаемый объект с политиками, которые необходимо оценить, чтобы определить, предоставлен ли доступ. Например:

X МОЖЕТ СДЕЛАТЬ Y НА РЕСУРСЕ Z,

где X представляет одного или нескольких пользователей, роли или группы или их комбинацию. Вы также можете использовать здесь утверждения и контекст;

Y представляет действие, которое должно быть выполнено, например, запись, просмотр и так далее;

Z представляет защищенный ресурс, например, «/accounts».

AtomID предоставляет платформу для создания целого ряда стратегий разрешений, начиная от простых и заканчивая очень сложными динамическими разрешениями на основе правил.

Это обеспечивает гибкость и помогает:

* сокращению затрат на рефакторинг кода и управление разрешениями;
* поддержке более гибкой модели безопасности, помогающей легко адаптироваться к изменениям в требованиях к безопасности;
* вносению изменений во время выполнения (приложения обеспечивает только защищенность ресурсов и областей, но не о то, как они защищены).

1. Политика. Политика определяет условия, которые должны быть выполнены для предоставления доступа к объекту. В отличие от разрешений, необходимо указывать не защищаемый объект, а скорее условия, которые должны быть выполнены для доступа к данному объекту (например, ресурс, область действия или и то, и другое). Политики тесно связаны с различными механизмами контроля доступа (ACM), которые можно использовать для защиты ресурсов. С помощью политик существует возможность реализовать стратегии управления доступом на основе атрибутов (ABAC), управления доступом на основе ролей (RBAC), управления доступом на основе контекста или любой их комбинации.

AtomID использует концепцию политик и то, как они определяются, при предоставлении концепции агрегированных политик, где существует возможность создания «политики политик» и контролировать поведение оценки. Вместо написания одной большой политики со всеми условиями, которые должны быть выполнены для доступа к данному ресурсу, реализация политик в службах авторизации AtomID следует методу «разделяй и властвуй». Другими словами, существует возможность создавать отдельные политики, затем повторно использовать их с разными разрешениями и создавать более сложные политики, комбинируя отдельные политики.

1. Поставщик политики.Поставщики политик — это реализации определенных типов политик. AtomID предоставляет встроенные политики, поддерживаемые соответствующими поставщиками политик, и существует возможность создавать свои собственные типы политик для поддержки ваших конкретных требований.

AtomID предоставляет SPI (Интерфейс поставщика услуг), который вы можете использовать для подключения собственных реализаций поставщика политик.

1. Тикет на получение разрешения. Тикет на получение разрешения — это специальный тип токена, определенный спецификацией Управляемого пользователем доступа (UMA), который предоставляет непрозрачную структуру, форма которой определяется сервером авторизации. Эта структура представляет ресурсы и/или области, запрашиваемые клиентом, контекст доступа, а также политики, которые должны быть применены к запросу данных авторизации (токен запрашивающей стороны [RPT]).

В UMA тикет разрешений имеют решающее значение для поддержки обмена данными между пользователями, а также обмена данными между пользователями и организациями. Использование тикетов разрешений для рабочих процессов авторизации позволяет использовать различные сценарии от простых до сложных, в которых владельцы ресурсов и серверы ресурсов имеют полный контроль над своими ресурсами на основе детализированных политик, регулирующих доступ к этим ресурсам.

В рабочем процессе UMA тикеты разрешений выдаются сервером авторизации серверу ресурсов, который возвращает тикет разрешений клиенту, пытающемуся получить доступ к защищенному ресурсу. Как только клиент получает тикет, он может сделать запрос на RPT (конечный токен, содержащий данные авторизации), отправив тикет обратно на сервер авторизации.

* 1. Управление серверами ресурсов

Согласно спецификации OAuth2, сервер ресурсов — это сервер, на котором размещаются защищенные ресурсы и который способен принимать запросы защищенных ресурсов и отвечать на них.

В AtomID серверам ресурсов предоставляется платформа для обеспечения детальной авторизации для защищенных ресурсов, где решения об авторизации могут приниматься на основе различных механизмов контроля доступа.

Любое клиентское приложение может быть настроено для поддержки детализированных разрешений. Поступая таким образом, концептуально клиентское приложение становится сервером ресурсов.

* + 1. Создание клиентского приложения

Первым шагом для включения служб авторизации AtomID является создание клиентского приложения, которое вы хотите превратить в сервер ресурсов.

Процедура:

1. В левой части меню выберите раздел «Клиенты» (Рисунок 7).

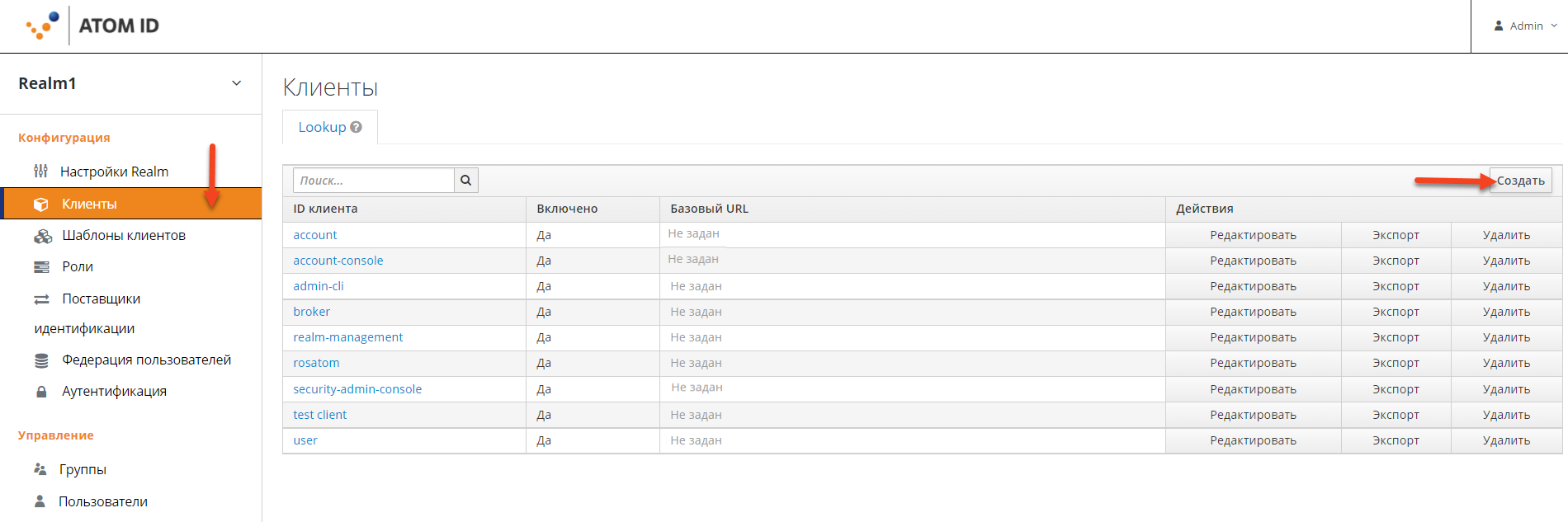


Рисунок 7 - Переход в раздел «Клиенты»

1. На открывшейся странице нажмите кнопку «Создать».

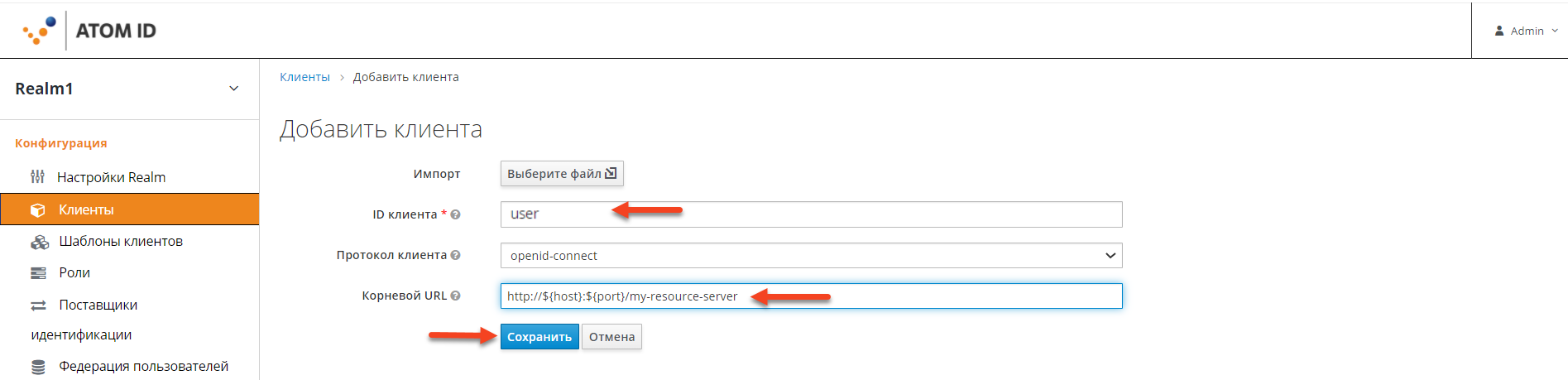


Рисунок 8 - Форма «Добавить клиента»

1. Введите идентификатор клиента в поле «ID клиента» (Рисунок 8). Например, User.
2. Введите корневой URL-адрес вашего приложения в поле «Корневой URL» (Рисунок 8).

Например, http://${host}:${port}/my-resource-server

1. Нажмите кнопку «Сохранить» (Рисунок 8).

Клиент будет создан, и откроется страница настроек клиента (Рисунок 9).

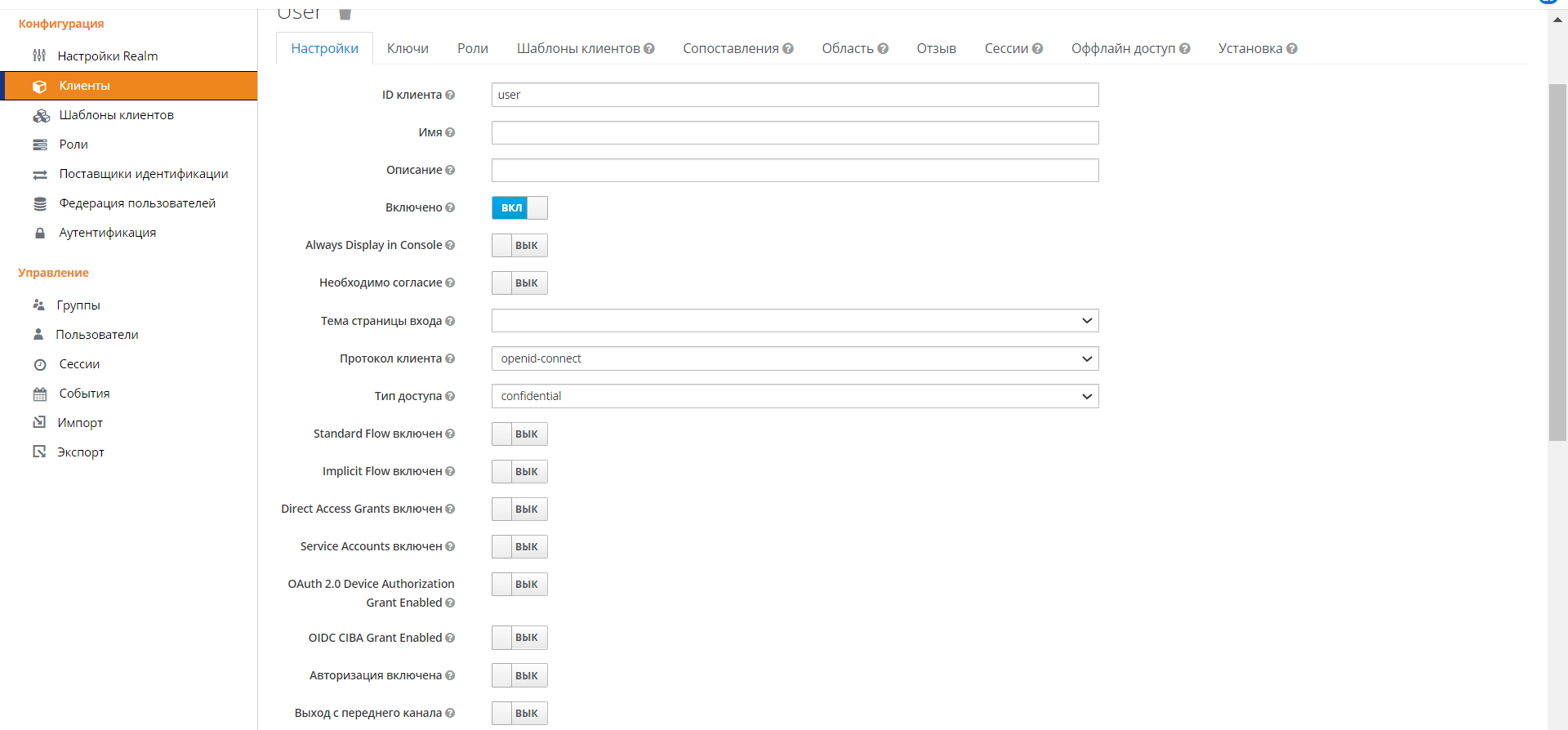


Рисунок 9 - Настройки клиента

* + 1. Включение служб авторизации

Чтобы превратить клиентское приложение OIDC в сервер ресурсов и включить детальную авторизацию, в поле «Тип доступа» выберите значение «Confidential» (Рисунок 10), установите переключатель «Авторизация включена» в положение «Включено» (Рисунок 10), а затем нажмите кнопку «Сохранить».

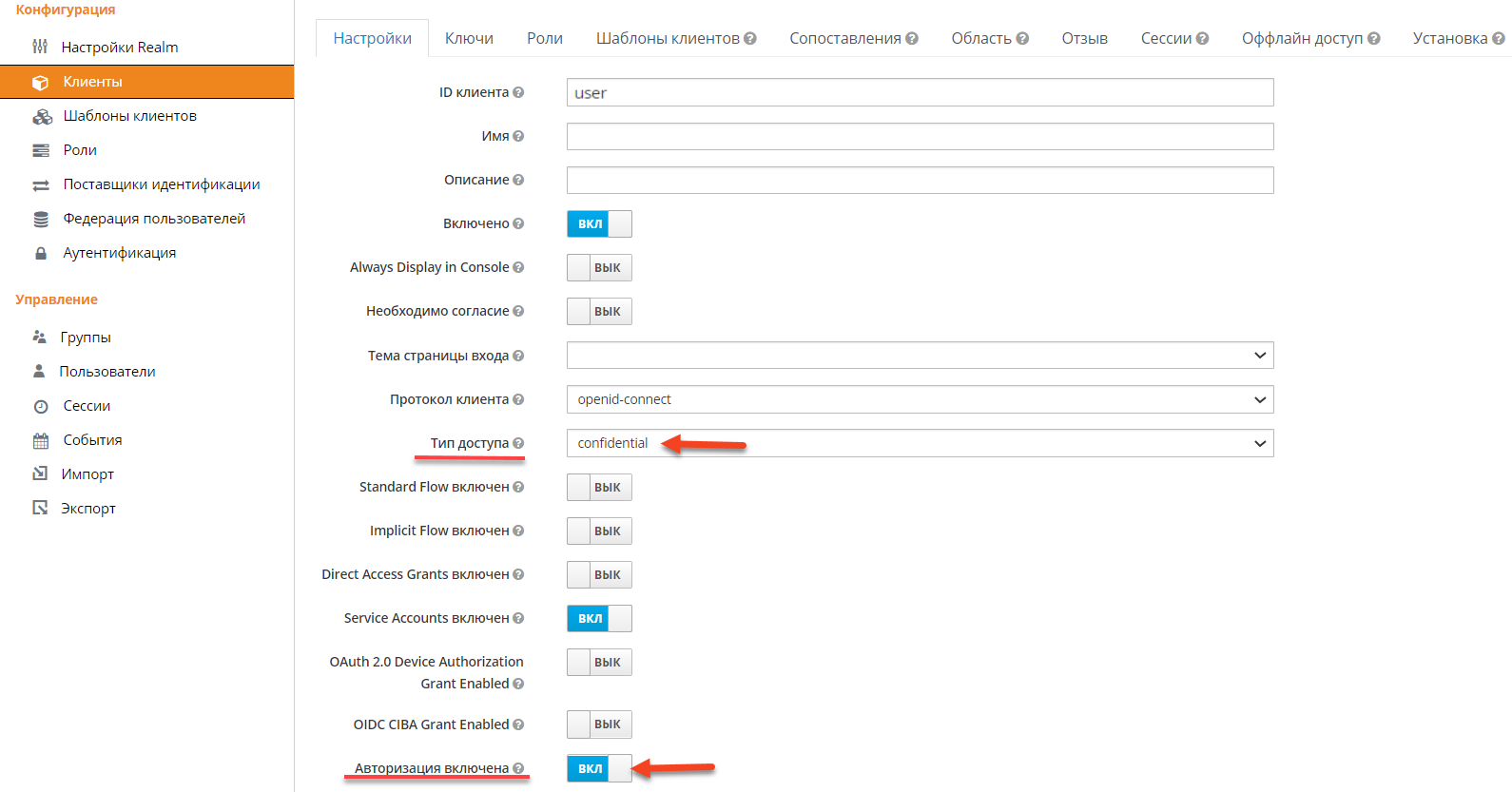


Рисунок 10 - Включение авторизации в настройке клиента

Для этого клиента отобразится новая вкладка «Авторизация». Перейдите на вкладку «Авторизация», после чего отобразится страница, представленная на рисунке ниже (Рисунок 11).

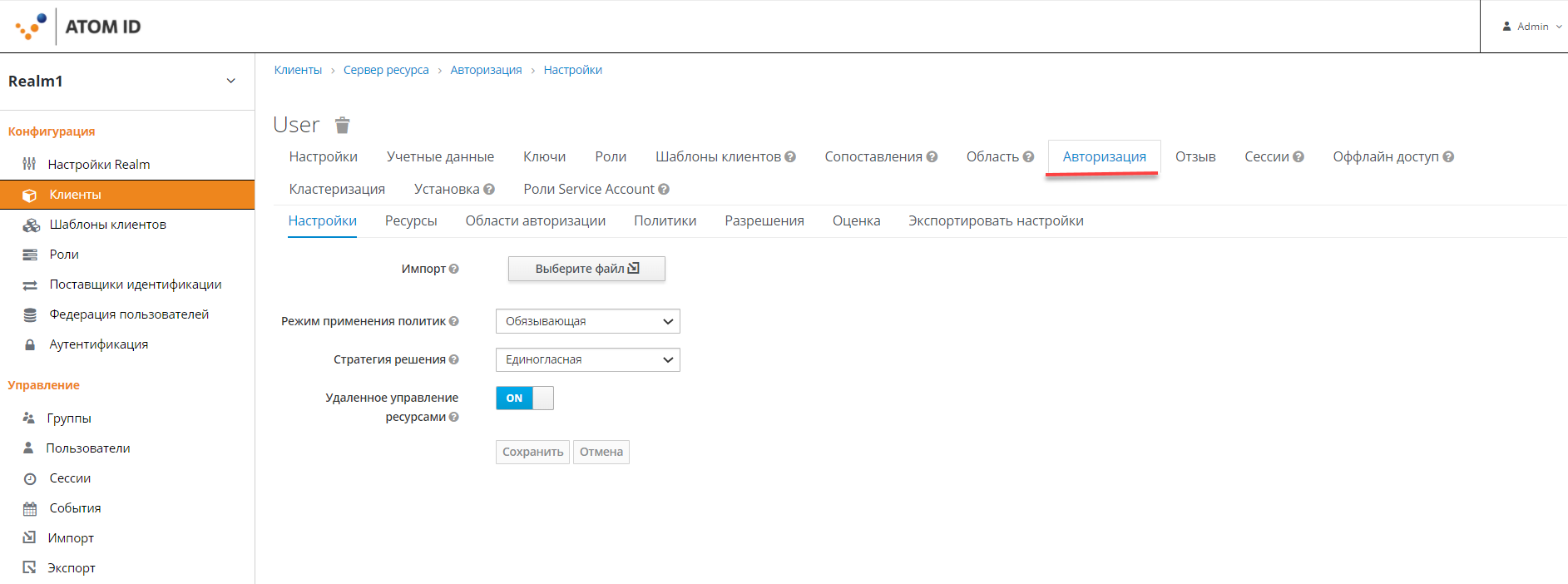


Рисунок 11 - Вкладка «Авторизация»

Вкладка «Авторизация» содержит дополнительные вложенные вкладки, описывающие различные шаги, которым необходимо следовать, чтобы защитить ресурсы приложения. Каждая вкладка рассматривается отдельно в конкретном разделе этой документации. Краткое описание каждого из них:

1. Настройки **-** общие настройки для вашего сервера ресурсов. Дополнительные сведения об этой странице приведены в разделе 2.2.2.1 «Настройки сервера ресурсов»;
2. Ресурсы **-** управление ресурсами приложения;
3. Область авторизации **-** управление областью авторизации;
4. Политики **-** управление политиками авторизации и определение условий, которые должны быть выполнены для предоставления разрешения;
5. Разрешения **-** управление разрешениями для ваших защищенных ресурсов и областей, установка связи разрешения с созданными политиками;
6. Оценка - воспроизведение запросов на авторизацию и просмотр результата оценки определенных разрешений и политик авторизации;
7. Экспорт настроек - Экспорт настроек атворизации в JSON-файл.
   * + 1. Настройки сервера ресурсов

На вкладке «Настройки» сервера ресурсов есть возможность настроить режим принудительного применения политики, установить разрешение удаленного управления ресурсами и экспорта параметров конфигурации авторизации с помощью выбора нужных значений в следующих полях:

1. Поле «Режим применения политики» - указывает, как применяются политики при обработке запросов на авторизацию, отправляемых на сервер. Содержит следующие значения:

* значение «Обязывающая» (значение режима по умолчанию) - запросы отклоняются по умолчанию, даже если нет политики, связанной с данным ресурсом;
* значение «Разрешающая» - запросы разрешены, даже если нет политики, связанной с данным ресурсом;
* значение «Отключено» - отключение оценки всех политик и разрешение доступа ко всем ресурсам.

1. Поле «Стратегия решения» - указывает способ, которым механизм оценки политики решает следует ли предоставлять ресурс или область, на основе результатов всех оцененных разрешений. Содержит следующие значения:

* значение «Утвердительная» - означает, что по крайней мере одно разрешение должно быть оценено как положительное решение, чтобы предоставить доступ к ресурсу и его областям;
* значение «Единогласная» - означает, что все разрешения должны быть оценены как положительное решение, чтобы окончательное решение также было положительным.

В качестве примера, если два разрешения для одного и того же ресурса или области находятся в конфликте (одно из них предоставляет доступ, а другое запрещает доступ), разрешение на ресурс или область будет предоставлено, если выбранная стратегия является «Утвердительной». В противном случае однократное отклонение любого разрешения также приведет к отказу в доступе к ресурсу или области.

1. Поле «Удаленное управление ресурсами» - указывает, может ли сервер ресурсов удаленно управлять ресурсами. Если значение равно false, ресурсами можно управлять только из консоли администрирования.
   * 1. Конфигурация по умолчанию

Когда вы создаете сервер ресурсов, AtomID создает конфигурацию по умолчанию для вашего вновь созданного сервера ресурсов.

Конфигурация по умолчанию состоит из:

1. Защищенный ресурс по умолчанию, представляющий все ресурсы в вашем приложении;
2. Политика, которая всегда предоставляет доступ к ресурсам, защищенным этой политикой;
3. Разрешение, которое управляет доступом ко всем ресурсам на основе политики по умолчанию.
   * + 1. Защищенный ресурс

Защищенный ресурс по умолчанию называется «Default Resource» (ресурсом по умолчанию), и отображается на вкладке «Ресурсы» (Рисунок 12).

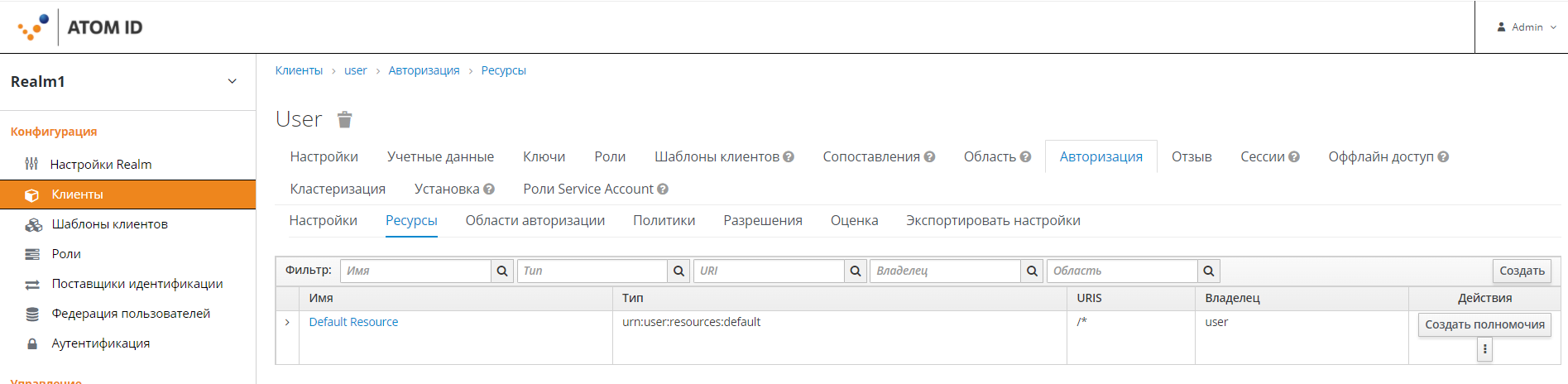


Рисунок 12 - Вложенная вкладка «Ресурсы»

Этот ресурс определяет «Тип», а именно urn:user:resources:default и URI /\*. Здесь поле «Тип» определяет шаблон подстановочных знаков, который указывает AtomID, что этот ресурс представляет все пути в вашем приложении. Другими словами, при включении принудительного применения политики для приложения все разрешения, связанные с ресурсом, будут проверены перед предоставлением доступа.

Тип, упомянуказанный выше, определяет значение, которое можно использовать для создания типизированных разрешений на ресурсы, которые должны быть применены к ресурсу по умолчанию или любому другому ресурсу, который необходимо создать с использованием того же типа.

* + - 1. Политика

Политика по умолчанию называется политикой только из области, для просмотра ее необходимо перейти на вкладку «Политики» (Рисунок 13).

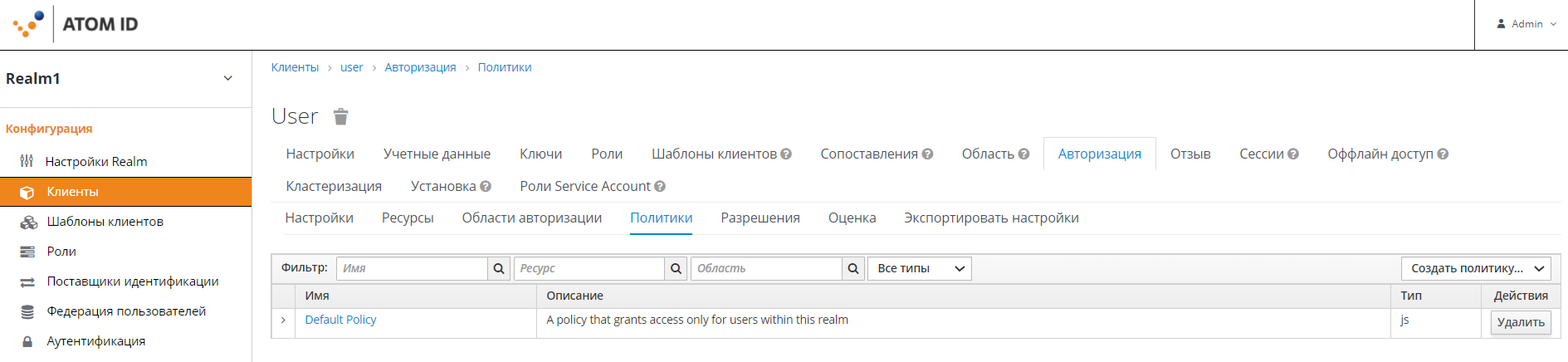


Рисунок 13 - Вложенная вкладка «Политики»

Эта политика представляет собой политику на основе JavaScript, определяющую условие, которое всегда предоставляет доступ к ресурсам, защищенным этой политикой. Если вы нажмете на эту политику, вы увидите, что она определяет правило следующим образом (Рисунок 14):



Рисунок 14 – Код JavaScript, определяющий условия для выбранной политики

* + - 1. Разрешение

Разрешение называется «Default Permission» (разрешением по умолчанию), для его просмотра необходимо перейти на вкладку «Разрешения» (Рисунок 15).

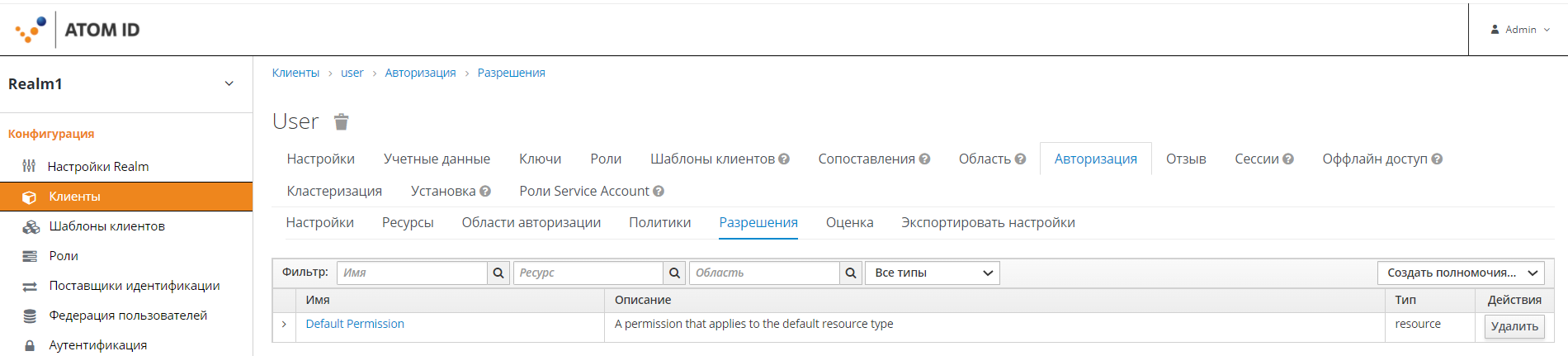


Рисунок 15 - Вложенная вкладка «Разрешения»

Это разрешение является разрешением на основе ресурсов, определяющим набор из одной или нескольких политик, которые применяются ко всем ресурсам заданного типа.

* + - 1. Изменение конфигурации по умолчанию

Для изменения конфигурации по умолчанию, необходимо удалить определения ресурсов, политики или разрешений по умолчанию, создав свои собственные.

Ресурс по умолчанию создается в приложении с использованием шаблона /\* или с помощью URI, который сопоставляется с любым ресурсом. Перед созданием ресурсов, разрешений и политики, необходимо убедиться, что конфигурация по умолчанию не противоречит установленным настройкам.

Конфигурация по умолчанию определяет ресурс, который сопоставляется со всеми путями в приложении. Для записи разрешения на созданные ресурсы, необходимо удалить «Ресурс по умолчанию» или изменить его поля URI на более конкретные пути в приложении. В противном случае политика, связанная с ресурсом по умолчанию (который по умолчанию всегда предоставляет доступ), позволит AtomID предоставлять доступ к любому защищенному ресурсу.

* 1. Управление ресурсами и областями безопасности

Управление ресурсами является простым и универсальным. После создания сервера ресурсов можно приступить к созданию ресурсов и областей, которые необходимо защитить. Ресурсами и областями можно управлять, перейдя на вкладки «Ресурсы» и "»Области авторизации» соответственно.

* + 1. Просмотр ресурсов

На вкладке «Ресурсы» (Рисунок 16) отображается список ресурсов, связанных с сервером ресурсов.

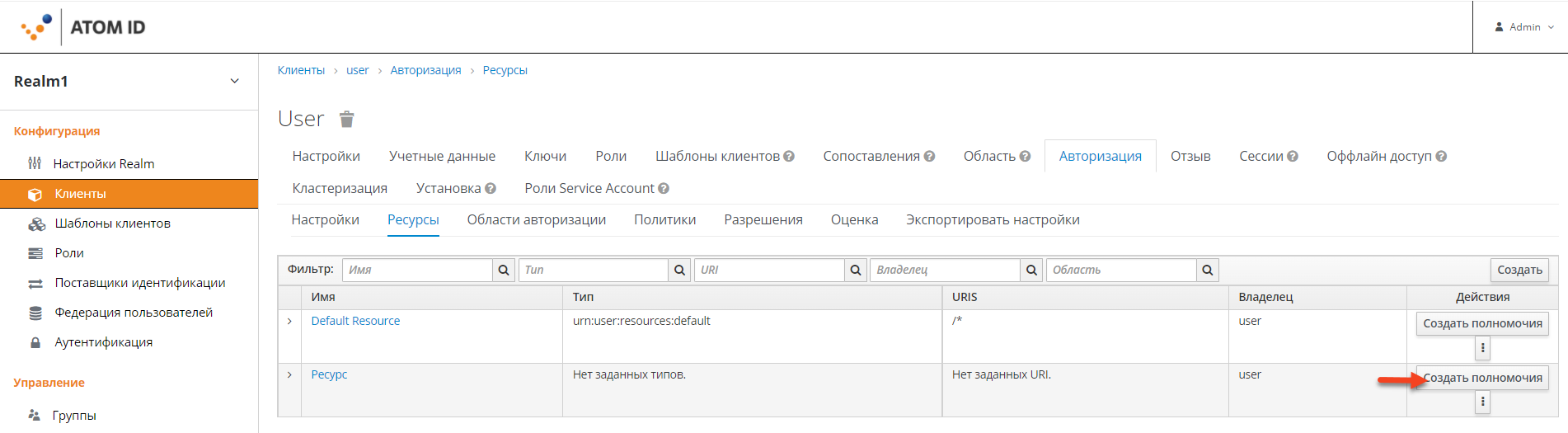


Рисунок 16 - Вложенная вкладка «Ресурсы»

Список ресурсов содержит информацию о защищенных ресурсах, таких как:

* Тип;
* URIS;
* Владелец;
* Связанные области, если таковые имеются;
* Связанные разрешения.

Из этого списка есть возможность напрямую создать разрешение, нажав кнопку «Создать полномочия» для ресурса (Рисунок 16), для которого необходимо создать разрешение.

Прежде чем создавать разрешения для ресурсов, нужно убедиться, что политики, которые планируется связать с этим разрешением, определены.

* + 1. Создание ресурсов

Создание ресурса является простым и универсальным процессом. Основная задача — это детализация создаваемых вами ресурсов. Другими словами, ресурсы могут быть созданы для представления набора из одного или нескольких ресурсов, и способ их определения имеет решающее значение для управления разрешениями.

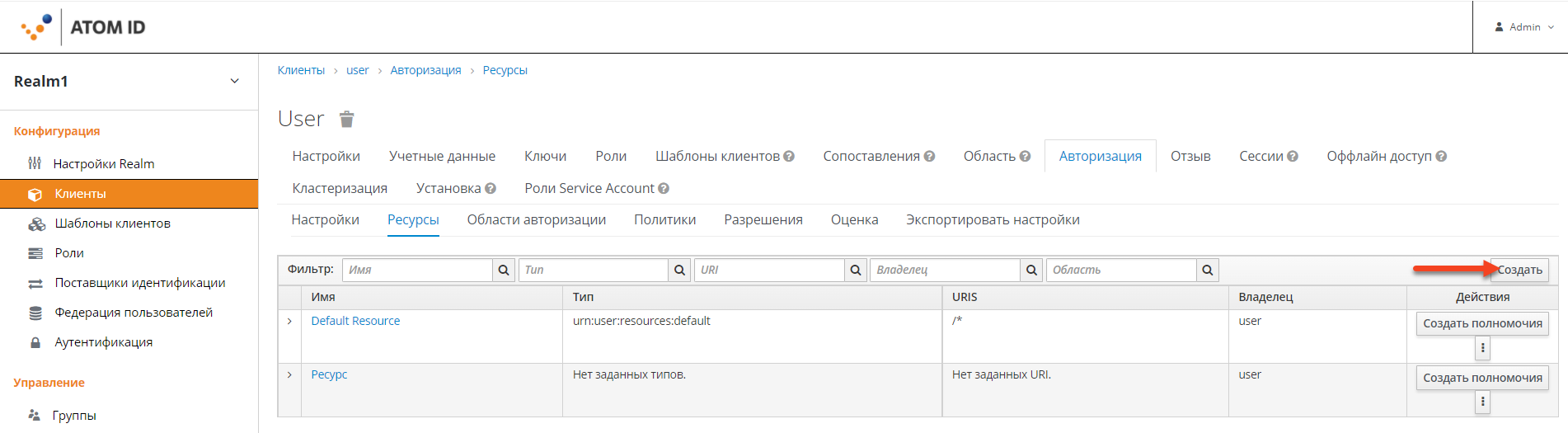


Рисунок 17 - Кнопка «Создать» для создания ресурса

Чтобы создать новый ресурс, нажмите кнопку «Создать» (Рисунок 17) в правом верхнем углу списка ресурсов.

В результате чего откроется форма «Добавить ресурс» (Рисунок 18).

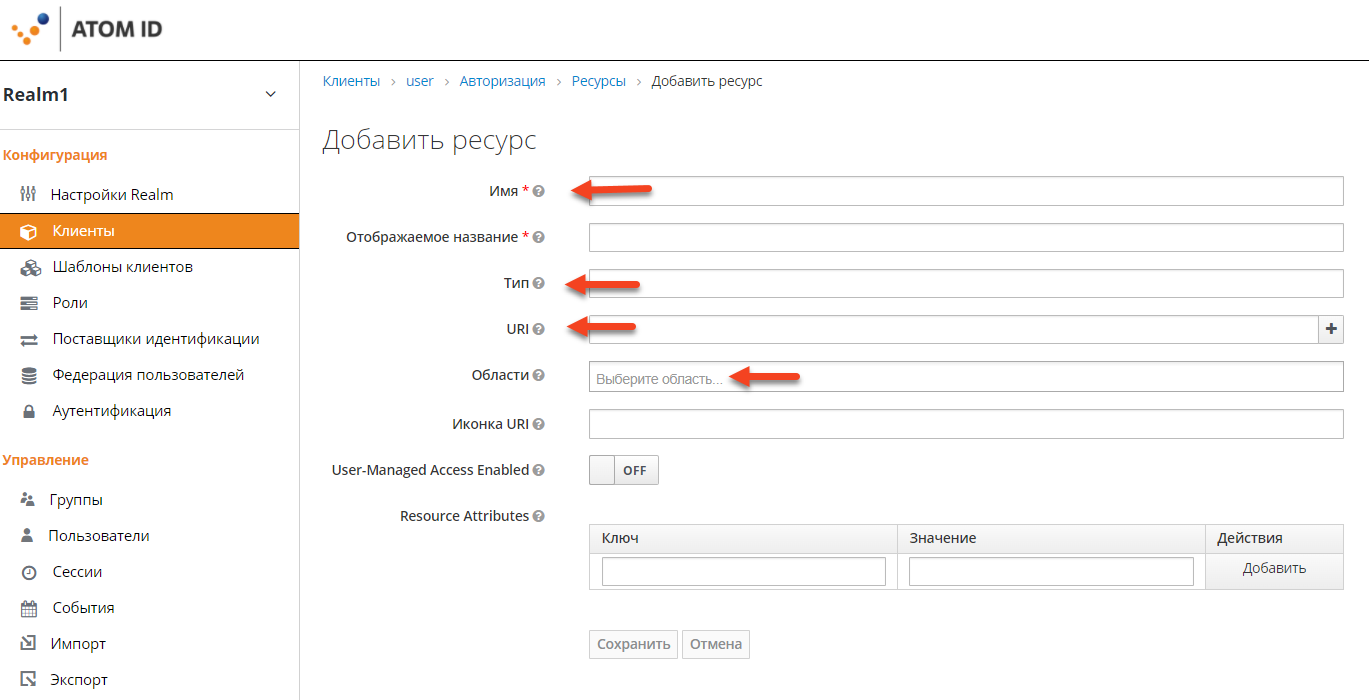


Рисунок 18 - Создание нового ресурса

В AtomID ресурс определяет небольшой набор информации, который является общим для различных типов ресурсов, таких как:

* поле «Имя» – должно быть уникальным, удобным к прочтению, содержащим описание создаваемого ресурса;
* поле «Тип» - строка, однозначно идентифицирующая тип набора из одного или нескольких ресурсов. Тип представляет собой строку, используемую для группировки различных экземпляров ресурсов. Например, тип по умолчанию для автоматически создаваемого ресурса - urn:resource-server-name:resources:default;
* поле «URIS» - URI, который предоставляет местоположения/адреса для ресурса. Для HTTP-ресурсов URI обычно представляют собой относительные пути, используемые для обслуживания этих ресурсов;
* поле «Области» - Одна или несколько областей для связи с ресурсом.
  + - 1. Атрибуты ресурсов

Ресурсы могут иметь связанные с ними атрибуты. Эти атрибуты можно использовать для предоставления дополнительной информации о ресурсе и для предоставления дополнительной информации политикам при оценке разрешений, связанных с ресурсом.

Каждый атрибут представляет собой пару «Ключ-значение», где значение может быть набором из одной или нескольких строк. Для атрибута можно определить несколько значений, разделив каждое значение запятой.

* + - 1. Типизированные ресурсы

Поле типа ресурса можно использовать для группировки различных ресурсов вместе, чтобы их можно было защитить с помощью общего набора разрешений.

* + - 1. Владельцы ресурсов

У ресурсов также есть владелец. По умолчанию ресурсы принадлежат серверу ресурсов.

Ресурсы также могут быть связаны с пользователями, поэтому есть существует возможность создания разрешения на основе владельца ресурса. Например, только владельцу ресурса разрешено удалять или обновлять данный ресурс.

* + - 1. Удаленное управление ресурсами

Управление ресурсами также доступно через API защиты, что позволяет серверам ресурсов удаленно управлять своими ресурсами.

При использовании API защиты серверы ресурсов могут быть реализованы для управления ресурсами, принадлежащими их пользователям. В этом случае вы можете указать идентификатор пользователя, чтобы настроить ресурс как принадлежащий определенному пользователю.

AtomID предоставляет серверам ресурсов полный контроль над их ресурсами. В последствии должна быть возможность разрешать пользователям управлять своими собственными ресурсами, а также утверждать запросы на авторизацию и управлять разрешениями, особенно при использовании протокола UMA.

* 1. Управление политиками

Политики определяют условия, которые должны быть выполнены перед предоставлением доступа к объекту.

Процедура:

1. Перейдите на вложенную вкладку «Политика» () для просмотра всех политик, связанных с сервером ресурсов. На этой вкладке отображается список ранее созданных политик, а также находятся кнопки для создания и редактирования политик.

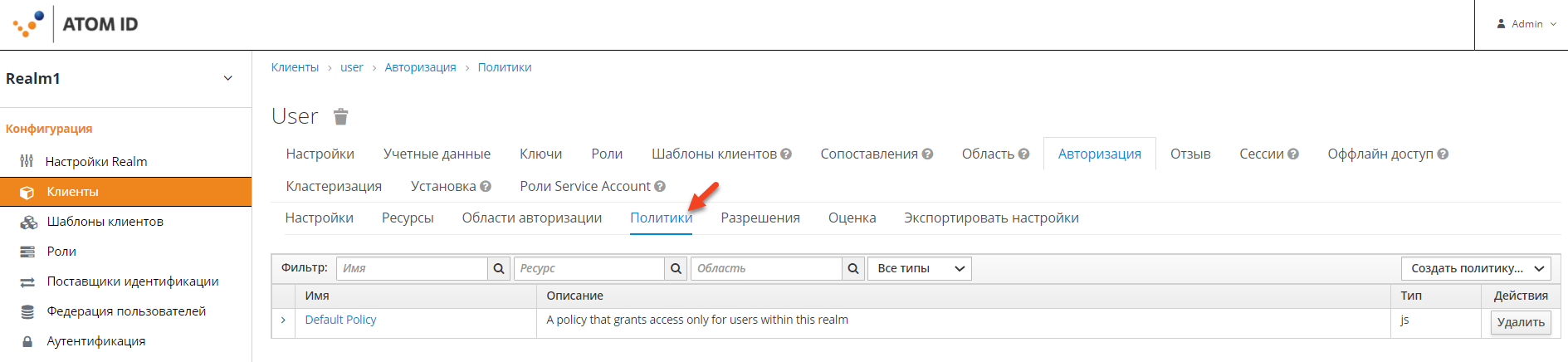


Рисунок 19 - Вложенная вкладка «Политики» для ресурса

1. Чтобы создать новую политику, нажмите кнопку «Создать политику» в правом верхнем углу таблицы со списком политик, затем из выпадающего списка выберите тип создаваемой политики (Рисунок 20).

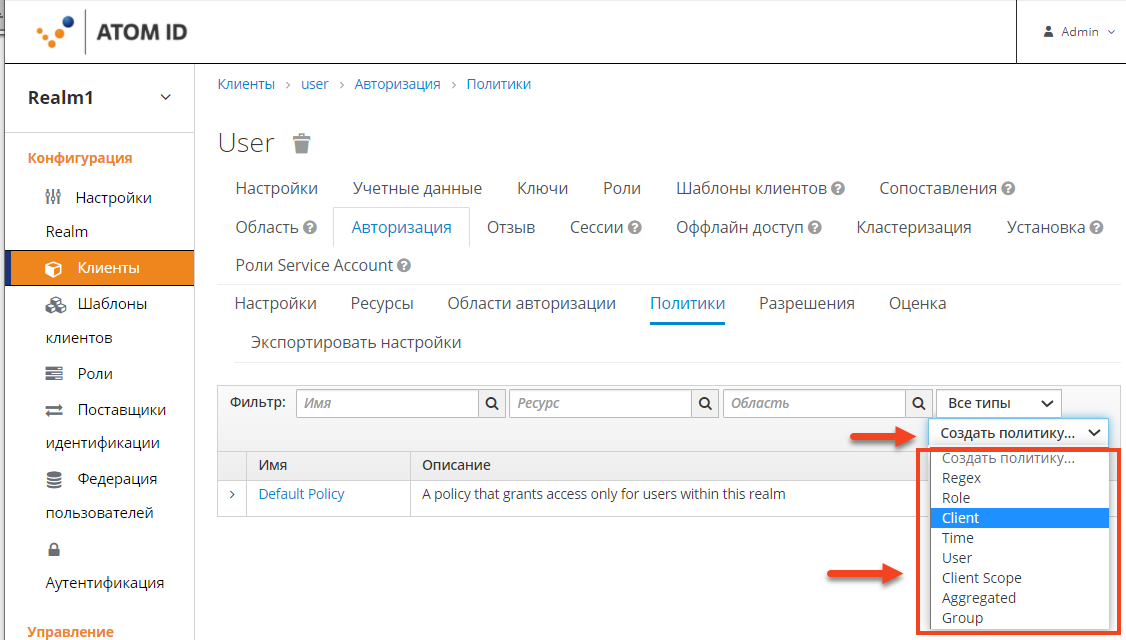


Рисунок 20 - Кнопка «Создать политику»

Подробные сведения о каждом типе политики описаны в этом разделе.

* + 1. Политика для пользователей

Указанный тип политики необходимо использовать с целью определения условий для разрешений, при которых одному или нескольким пользователям разрешен доступ к объекту.

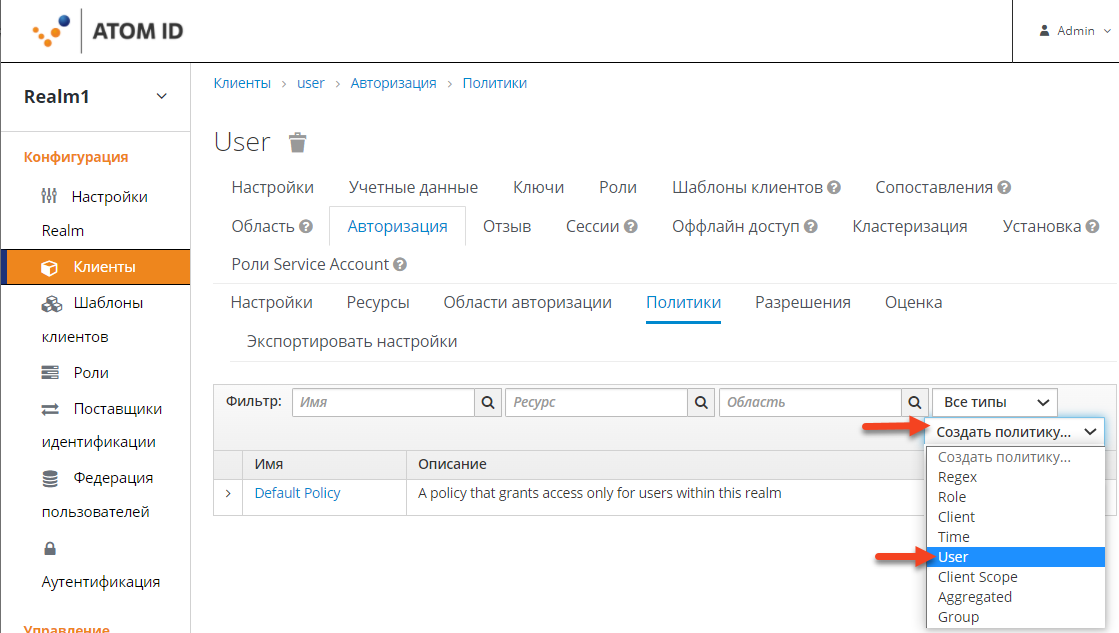


Рисунок 21 - Создание политики для пользователей

Чтобы создать новую пользовательскую политику, выберите значение «User» (Рисунок 21) в выпадающем списке при нажатии на кнопку «Создать политику». В результате чего откроется форма «Добавить политику пользователей» (Рисунок 22).

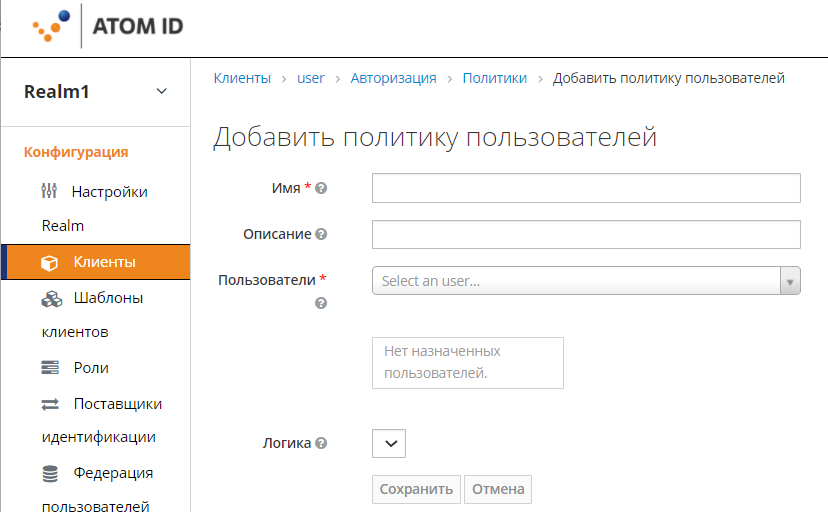


Рисунок 22 - Форма «Добавить политику пользователей»

* + - 1. Состав

Форма «Добавить политику пользователей» имеет следующй состав полей:

* поле «Имя» - должно быть уникальным, удобным к прочтению, однозначно идентифицирующим политику. Рекомендуется использовать имена, которые тесно связаны с бизнесом и требованиями безопасности, чтобы было легче их идентифицировать;
* поле «Описание» - строка, содержащая подробную информацию об этой политике;
* поле «Пользователи» - указывает, каким пользователям предоставляется доступ в соответствии с создаваемой политикой;
* поле «Логика» - логика применения создаваемой политики после оценки других условий. Имеет значения «Позитивная»/»Негативная».
  + 1. Политика для ролей

Указанный тип политики (Рисунок 23) необходимо использовать для определения условий для разрешений, при которых доступ к объекту разрешен одной или нескольким ролям.

По умолчанию роли, добавленные в эту политику, не указаны как требуемые, и политика предоставит доступ в случае, если пользователю, запрашивающему доступ, была предоставлена любая из этих ролей. Конкретная роль может быть указана по мере необходимости, если нужно применить определенную роль. Дополнительно существует возможность комбинировать обязательные и необязательные роли, независимо от того, являются ли они ролями области или клиента.

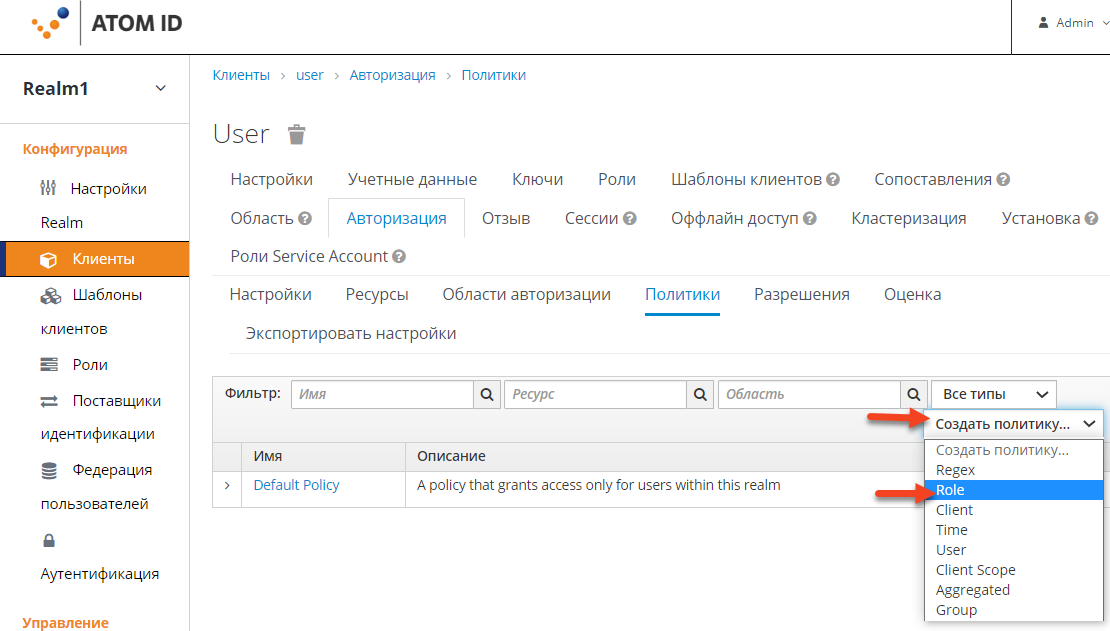


Рисунок 23 - Создание политики для роли

Политики ролей могут быть полезны, когда требуется более ограниченное управление доступом на основе ролей (RBAC), где для предоставления доступа к объекту необходимо применять определенные роли. Например, для принудительной установки факта, что пользователь должен дать согласие на разрешение клиентскому приложению (которое действует от имени пользователя) получать доступ к ресурсам пользователя. Существует возможность использования сопоставления области видимости клиента AtomID для включения страниц согласия или принудительного предоставления клиентам явного указания области видимости при получении токенов доступа с сервера AtomID.

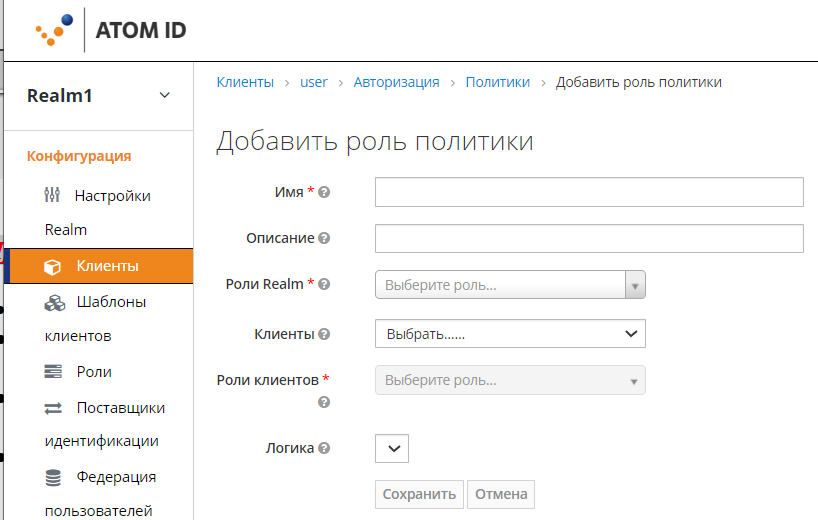


Рисунок 24 - Форма «Добавить роль политики»

Чтобы создать новую политику на основе ролей, выберите значение «Role» (Рисунок 23) в выпадающем списке при нажатии на кнопку «Создать политику». В результате чего откроется форма «Добавить роль политики» (Рисунок 24).

* + - 1. Состав

Форма «Добавить роль политики» имеет следующй состав полей:

* поле «Имя» - должно быть уникальным, удобным к прочтению, однозначно описывающим политику. Рекомендуется использовать имена, которые тесно связаны с бизнесом и требованиями безопасности, чтобы было легче их идентифицировать;
* поле «Описание» - строка, содержащая подробную информацию об этой политике;
* поли «Роли Realm» - указывает какие роли разрешены этой политикой;
* поле «Роли клиентов» - указывает, какие роли клиентов разрешены этой политикой. Чтобы включить это поле, необходимо сначала выбрать клиента в поле «Клиент»;
* поле «Логика» - логика применения создаваемой политики после оценки других условий. Имеет значения «Позитивная»/»Негативная».
  + - 1. Определение роли по мере необходимости

При создании политики на основе ролей существует возможность указания конкретной роли по мере необходимости. При этом политика предоставляет доступ только в том случае, если пользователю, запрашивающему доступ, были предоставлены все необходимые роли. Как роли области, так и роли клиента могут быть так настроены.

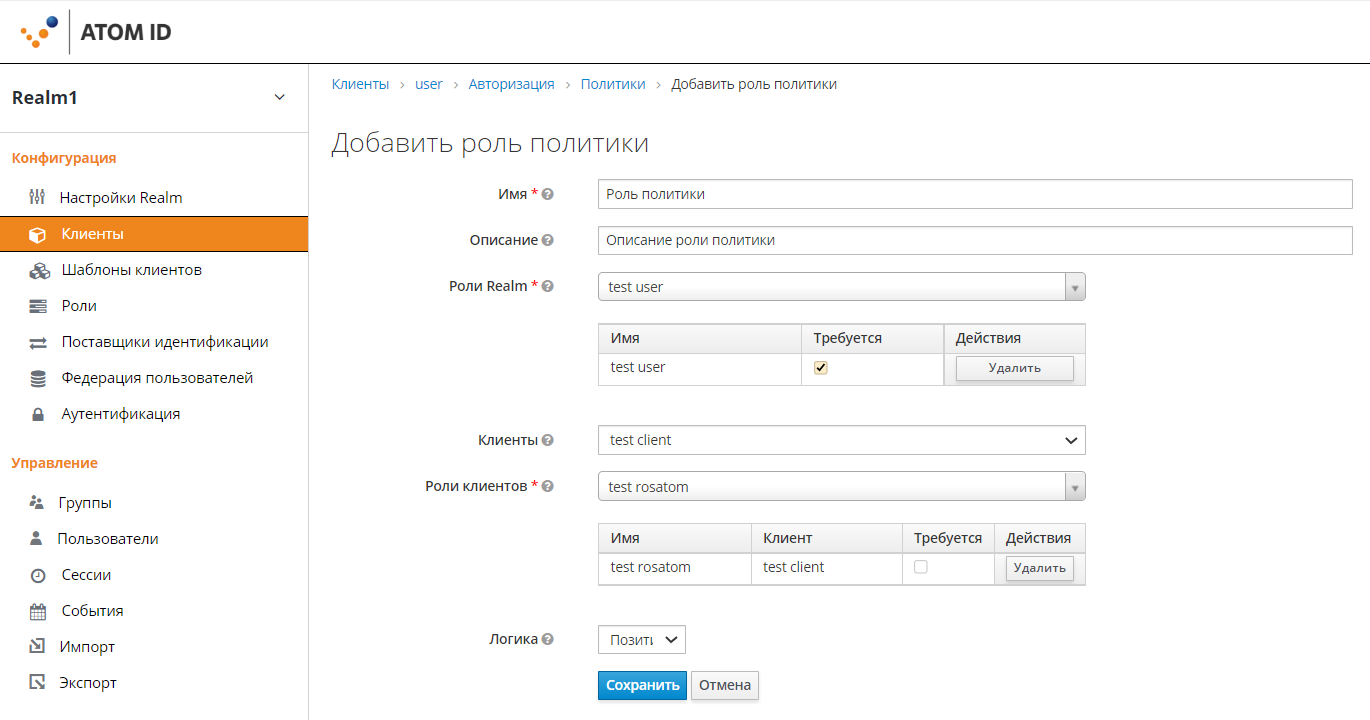


Рисунок 25 – Пример заполнения формы добавления роли политики

Чтобы указать требуемую роль, установите флажок «Требуется» ((Рисунок 25) для роли, которую необходимо настроить в соответствии с требованиями.

Обязательные роли могут быть полезны, когда политика определяет несколько ролей, но только часть из них является обязательной. В этом случае существует возможность объединить роли области и клиента, чтобы включить еще более детализированную модель управления доступом на основе ролей (RBAC) для приложения. Например, могут быть политики, специфичные для клиента, и может потребоваться определенная роль клиента, связанная с этим клиентом, или необходимо обеспечить, чтобы доступ предоставлялся только при наличии определенной роли области. Также существует возможность комбинирования оба подхода в рамках одной и той же политики.

* + 1. Политика для определения временных условий

Указанный тип политики используется для определения временных условий для разрешений.

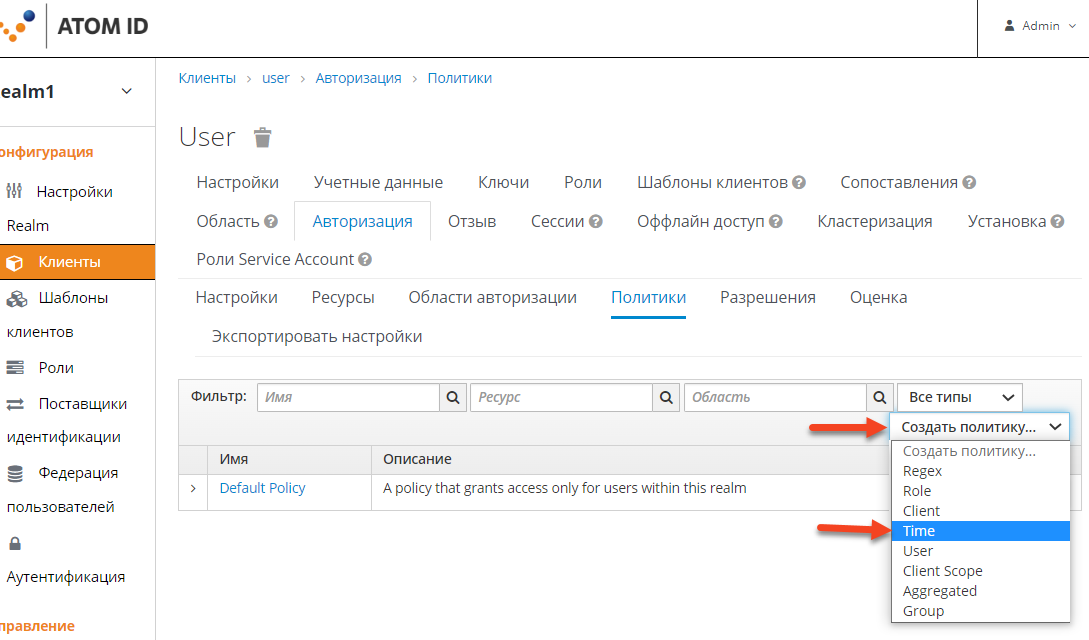


Рисунок 26 - Создание политики для определения временных условий

Чтобы создать новую политику, основанную на времени, выберите значение «Time» (Рисунок 26) в выпадающем списке при нажатии кнопки «Создать политику».

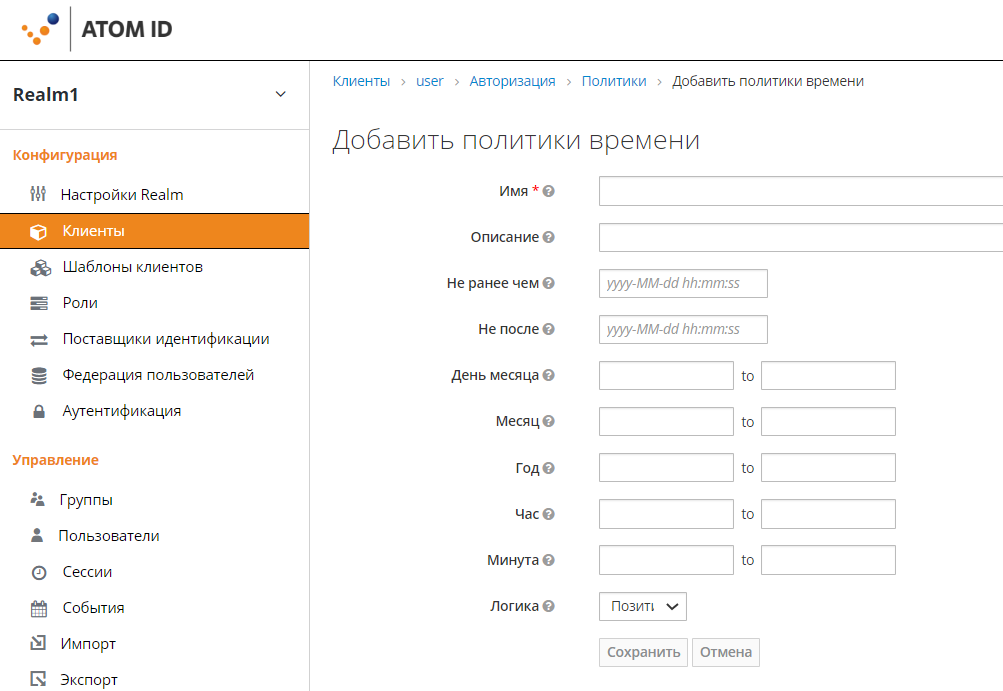


Рисунок 27 - Форма «Добавить политики времени»

В результате чего откроется форма «Добавить политики времени» (Рисунок 27).

* + - 1. Состав

Форма «Добавить политики времени» имеет следующй состав полей:

* поле «Имя» - должно быть уникальным, удобным к прочтению, однозначно описывающим политику. Рекомендуется использовать имена, которые тесно связаны с бизнесом и требованиями безопасности, чтобы было легче их идентифицировать;
* поле «Описание» - строка, содержащая подробную информацию об этой политике;
* поли «Не ранее чем» - определяет время, до которого доступ не должен предоставляться. Разрешение предоставляется только в том случае, если текущая дата/время позже или равно этому значению
* поле «Не после» - определяет время, по истечении которого доступ не должен предоставляться. Разрешение предоставляется только в том случае, если текущая дата/время раньше или равно этому значению;
* поле «День месяца» - определяет день месяца, в который должен быть предоставлен доступ. Также допустимо указывать диапазон дат. В этом случае разрешение предоставляется только в том случае, если текущий день месяца находится между двумя указанными значениями или равен им;
* поле «Месяц» - определяет месяц, в течение которого должен быть предоставлен доступ. Также допустимо указывать диапазон месяцев. В этом случае разрешение предоставляется только в том случае, если текущий месяц находится между двумя указанными значениями или равен им;
* поле «Год» - определяет год, в котором должен быть предоставлен доступ. Также допустимо указывать диапазон лет. В этом случае разрешение предоставляется только в том случае, если текущий год находится между двумя указанными значениями или равен им;
* поле «Час» - определяет час, в течение которого должен быть предоставлен доступ. Также допустимо указывать диапазон часов. В этом случае разрешение предоставляется только в том случае, если текущий час находится между двумя указанными значениями или равен им;
* поле «Минута» - Определяет минуту, в течение которой должен быть предоставлен доступ. Также допустимо указывать диапазон минут. В этом случае разрешение предоставляется только в том случае, если текущая минута находится между двумя указанными значениями или равна им;
* поле «Логика» - логика применения создаваемой политики после оценки других условий. Имеет значения «Позитивная»/»Негативная».

Доступ предоставляется только в том случае, если выполнены все условия. AtomID выполнит действие логики «AND» на основе результата каждого условия.

* + 1. Агрегированная политика

AtomID позволяет создавать «политику политик», концепцию, называемую агрегированием политик, что предполагает использование агрегирования политик для повторного использования существующих политик с целью создания более сложных полититик, а также обеспечения большей независимости ваших разрешений от политик, которые оцениваются во время обработки запросов на авторизацию.

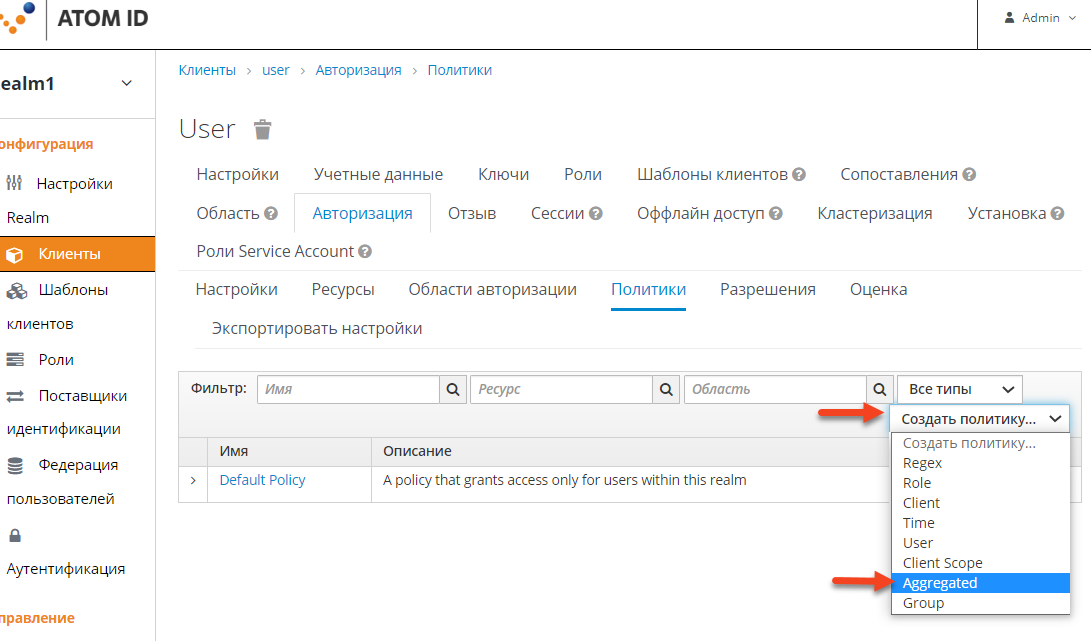


Рисунок 28 - Создание агрегированной политики

Чтобы создать новую агрегированную политику, выберите значение «Агрегированный» (Рисунок 28) в выпадающем списке при нажатии кнопки «Создать политику». В результате чего откроется форма «Добавить совокупную политику» (Рисунок 29).

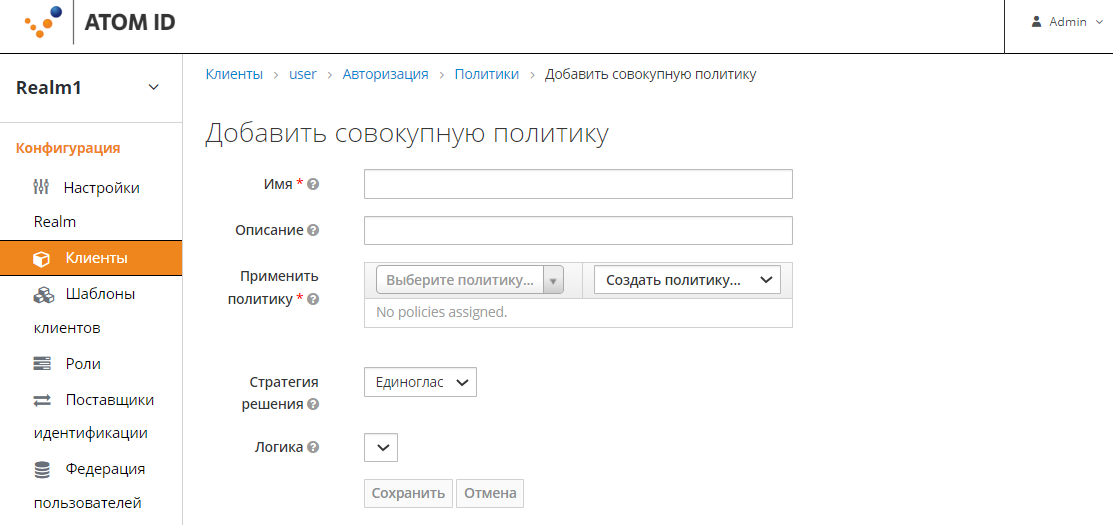


Рисунок 29 – Форма «Добавить совокупную политику»

Предположим, у имеется ресурс, доступ к которому могут получить только пользователи из AtomID.org домена и из определенного диапазона IP-адресов. Существует возможность создать единую политику с обоими условиями, повторно использовав доменную часть этой политики для применения к разрешениям, которые работают независимо от исходной сети.

Существует возможность создания отдельной политики как для условий домена, так и для условий сети, а также создания третьей политики на основе комбинации этих двух политик. С помощью агрегированной политики существует возможность свободно комбинировать другие политики, а затем применять новую агрегированную политику к любому разрешению.

При создании агрегированных политик необходимо помнить, что это не ввод циклической ссылки или зависимости между политиками. Если обнаружена циклическая зависимость, отсутствует возможность создания или обновления политики.

* + - 1. Состав

Форма «Добавить политики времени» имеет следующй состав полей:

* поле «Имя» - должно быть уникальным, удобным к прочтению, однозначно описывающим политику. Рекомендуется использовать имена, которые тесно связаны с бизнесом и требованиями безопасности, чтобы было легче их идентифицировать;
* поле «Описание» - строка, содержащая подробную информацию об этой политике;
* поле «Применить политику» - определяет одной или нескольких политик для связи с агрегированной политикой. Чтобы связать политику необходимо либо выбрать существующую политику, либо создать новую, выбрав тип политики, которую нужно создать.
* поле «Стратегия решения» - стратегия принятия решения;
* поле «Логика» - логика применения создаваемой политики после оценки других условий. Имеет значения «Позитивная»/»Негативная».
  + - 1. Стратегия принятия решений для агрегированных политик

При создании агрегированных политик существует возможность определения стратегии принятия решений, которая будет использоваться для определения окончательного решения на основе результатов каждой политики.

Стратегия принятия решений имеет следующие значения:

* «Единогласная» - стратегия по умолчанию, если она не указана. В этом случае все политики должны быть оценены в положительное решение, чтобы окончательное решение также было положительным;
* «Утвердительная» - по крайней мере одна политика должна дать оценку положительному решению, чтобы окончательное решение также было положительным;
* «Консенсусная» - количество положительных решений должно быть больше, чем количество отрицательных решений. Если количество положительных и отрицательных решений одинаково, окончательное решение будет отрицательным.
  + 1. Политика клиента

Этот тип политики можно использовать для определения условий для разрешений, при которых одному или нескольким клиентам разрешен доступ к объекту.

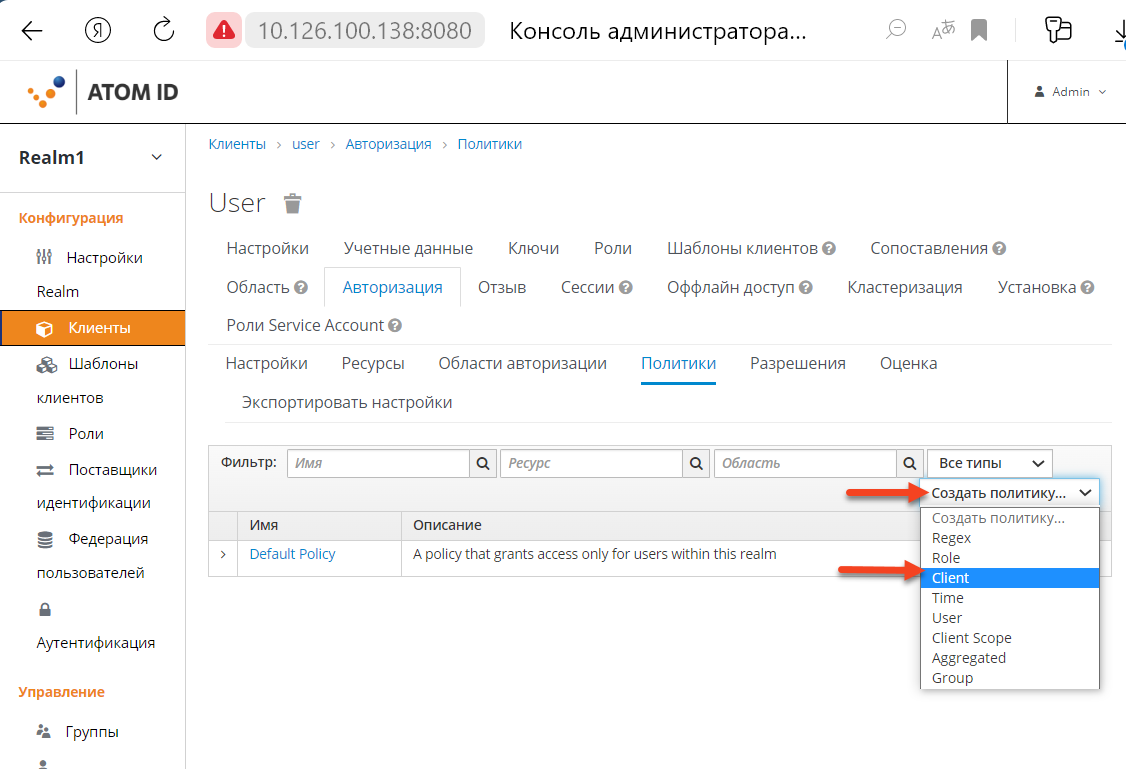


Рисунок 30 - Добавление политики клиента

Чтобы создать новую политику на основе клиента, выберите «Client» (Рисунок 30) в выпадающем списке при нажатии кнопки «Создать политику».

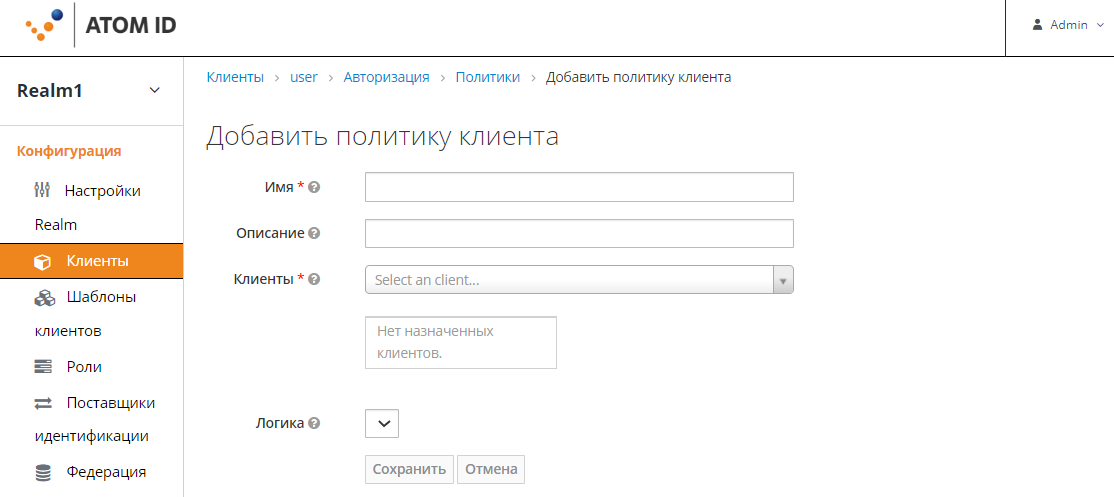


Рисунок 31 - Форма «Добавить политику клиента»

В результате чего откроется форма «Добавить политику клиента» (Рисунок 31).

* + - 1. Состав

Форма «Добавить политику клиента» имеет следующй состав полей:

* поле «Имя» - должно быть уникальным, удобным к прочтению, однозначно описывающим политику. Рекомендуется использовать имена, которые тесно связаны с бизнесом и требованиями безопасности, чтобы было легче их идентифицировать;
* поле «Описание» - строка, содержащая подробную информацию об этой политике;
* поле «Клиенты» - указывает, каким клиентам предоставляется доступ в соответствии с этой политикой;
* поле «Логика» - логика применения создаваемой политики после оценки других условий. Имеет значения «Позитивная»/»Негативная».
  + 1. Политика для групп

Этот тип политики можно использовать с целью определения условий для разрешений, при которых из одной или нескольких групп (и их иерархий) разрешен доступ к объекту.

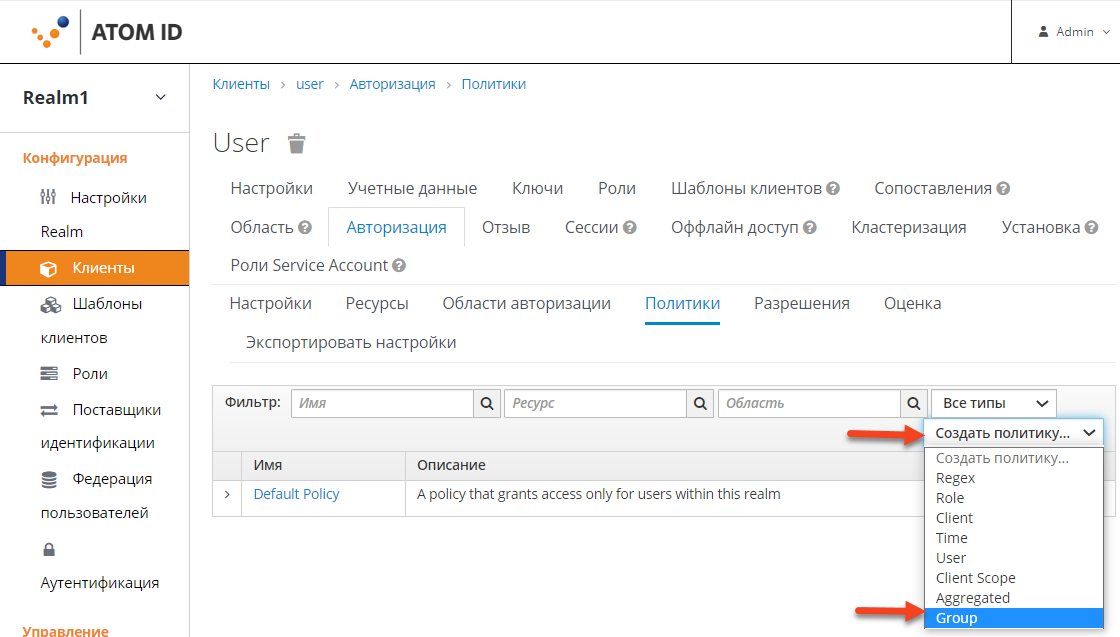


Рисунок 32 - Добавление политики для групп

Чтобы создать новую политику для групп (Рисунок 32), выберите пункт «Group» в выпадающем списке при нажатии кнопки «Создать политику».

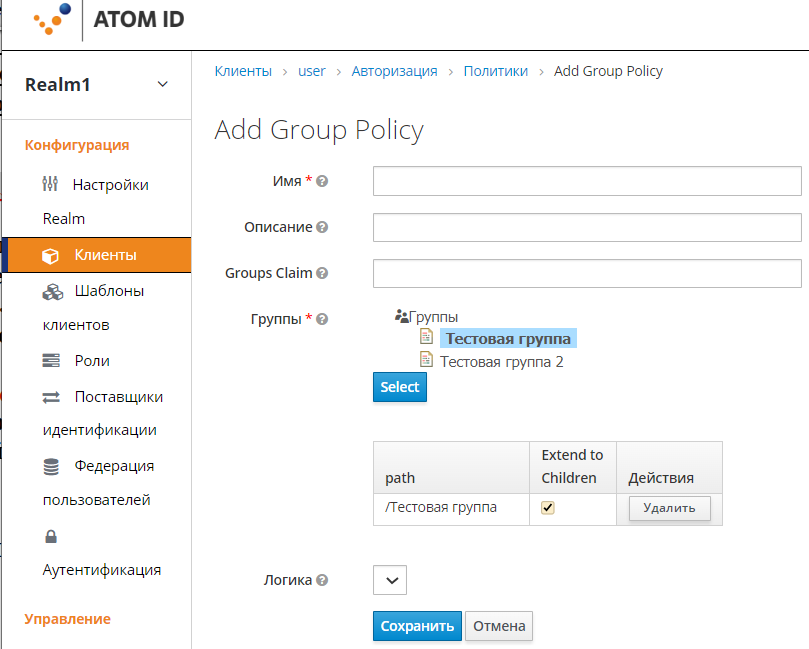


Рисунок 33 - Форма «Add Group Policy»

В результате чего откроется форма «Add Group Policy» (Рисунок 33).

* + - 1. Состав

Форма «Add Group Policy» имеет следующй состав полей:

* поле «Имя» - должно быть уникальным, удобным к прочтению, однозначно описывающим политику. Рекомендуется использовать имена, которые тесно связаны с бизнесом и требованиями безопасности, чтобы было легче их идентифицировать;
* поле «Описание» - строка, содержащая подробную информацию об этой политике;
* поле «Groups Claim» - указывает утвержденное имя, содержащее имена групп и/или пути. Обычно запросы на авторизацию обрабатываются на основе идентификационного токена или токена доступа, ранее выданного клиенту, действующему от имени какого-либо пользователя. Если он определен, токен должен включать утверждение, из которого эта политика будет получать группы, членом которых является пользователь. Если не определено, группы пользователей будут получены из конфигурации вашей области;
* поле «Группы» - позволяет выбрать группы, к которым должна применяться эта политика при оценке разрешений. После добавления группы имеется возможность расширить доступ к дочерним элементам группы, установив флажок «Extend to Children». В ином случае ограничения доступа применяются только к выбранной группе;
* поле «Logic» - логика применения создаваемой политики после оценки других условий. Имеет значения «Позитивная»/»Негативная».
  + - 1. Расширение доступа к дочерним группам

По умолчанию, при добавлении группы в вышеуказанную политику, ограничения доступа будут применяться только к членам выбранной группы.

При некоторых обстоятельствах может потребоваться разрешить доступ не только к самой группе, но и к любой дочерней группе в иерархии. Для любой добавленной группы существует возможность установить флажок «Extend to Children» (Рисунок 34), чтобы расширить доступ к дочерним группам.

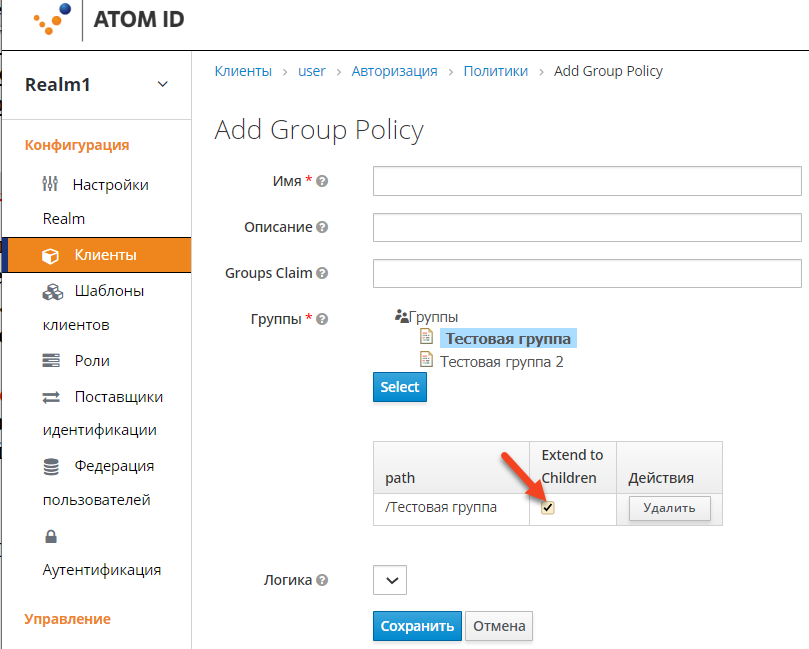


Рисунок 34 - Флажок «Extend to children»

В приведенном выше примере политика предоставляет доступ любому пользователю, входящему в нее, или любому из ее дочерних элементов.

* + 1. Политика для клиентской области

Этот тип политики можно использовать с целью определения условий для ваших разрешений, при которых доступ к объекту разрешен набору из одной или нескольких клиентских областей.

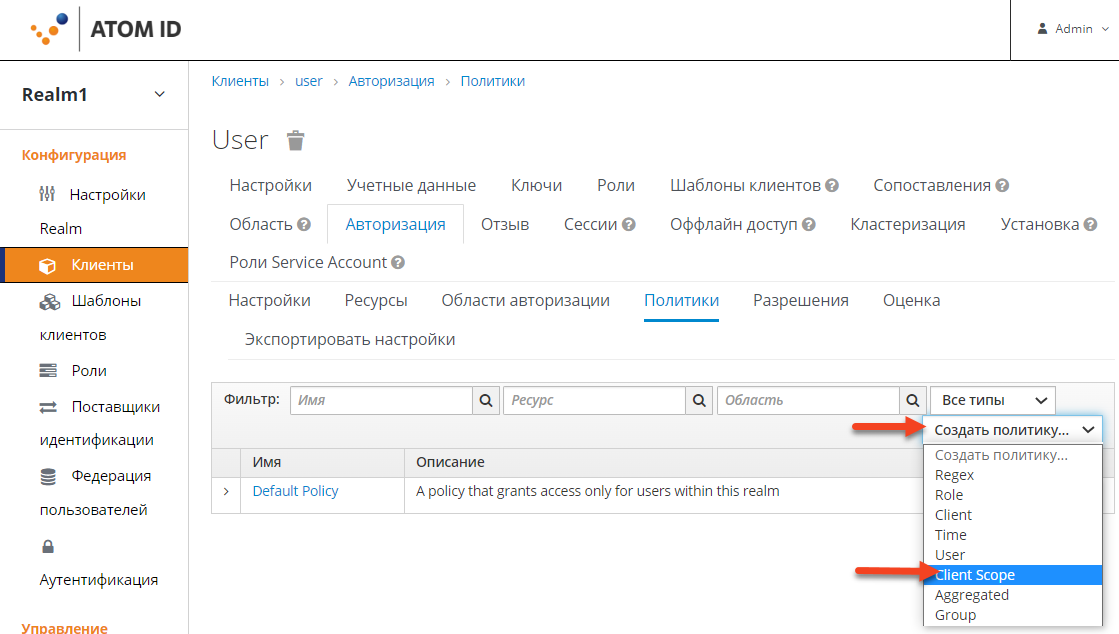


Рисунок 35 - Создание политики для клиентской области

По умолчанию клиентские области, добавленные в эту политику, не указаны как требуемые, и политика предоставит доступ, если клиенту, запрашивающему доступ, была предоставлена любая из этих клиентских областей. Для применения определенной клиентской области необходимо указать конкретную клиентскую область по мере необходимости.

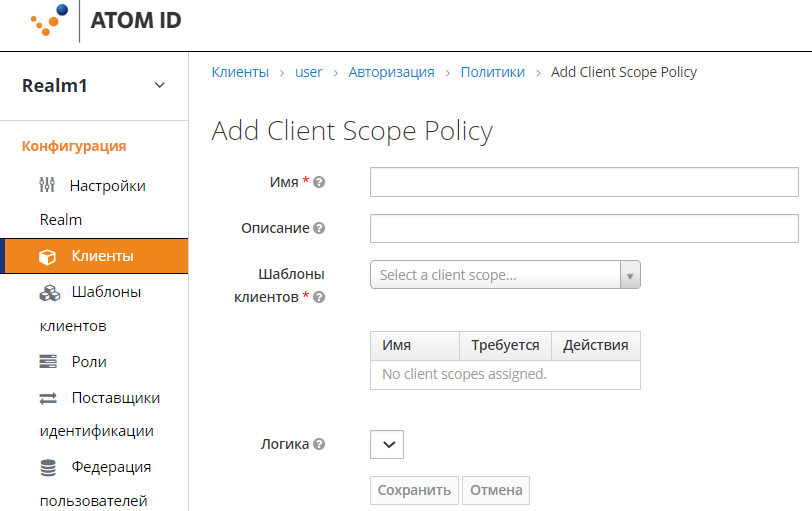


Рисунок 36 - Форма «Add Client Scope Policy»

Чтобы создать новую политику на основе клиентской области, выберите «Client Scope» (Рисунок 35) в выпадающем списке при нажатии кнопки «Создать политику».

В результате чего откроется форма «Add Client Scope Policy» (Рисунок 36)/

* + - 1. Состав

Форма «Add Client Scope Policy» имеет следующй состав полей:

* поле «Имя» - должно быть уникальным, удобным к прочтению, однозначно описывающим политику. Рекомендуется использовать имена, которые тесно связаны с бизнесом и требованиями безопасности, чтобы было легче их идентифицировать;
* поле «Описание» - строка, содержащая подробную информацию об этой политике;
* поле «Шаблоны клиентов» - указывает, какие клиентские области разрешены этой политикой;
* поле «Logic» - логика применения создаваемой политики после оценки других условий. Имеет значения «Позитивная»/»Негативная».
  + - 1. Определение клиентской области по мере необходимости

При создании политики на основе клиентской области существует возможность указать конкретную клиентскую область по мере необходимости, при этом политика предоставит доступ только в том случае, если клиенту, запрашивающему доступ, были предоставлены все необходимые клиентские области.

Чтобы указать требуемую клиентскую область, установите флаг «Требуется» для клиентской области, которую необходимо настроить в соответствии с требованиями.

Обязательные клиентские области могут быть полезны, когда политика определяет несколько клиентских областей, но только подмножество из них является обязательным.

* + 1. Политика, основанная на регулярных выражений

Этот тип политики можно использовать с целью определения условий регулярных выражений для ваших разрешений.

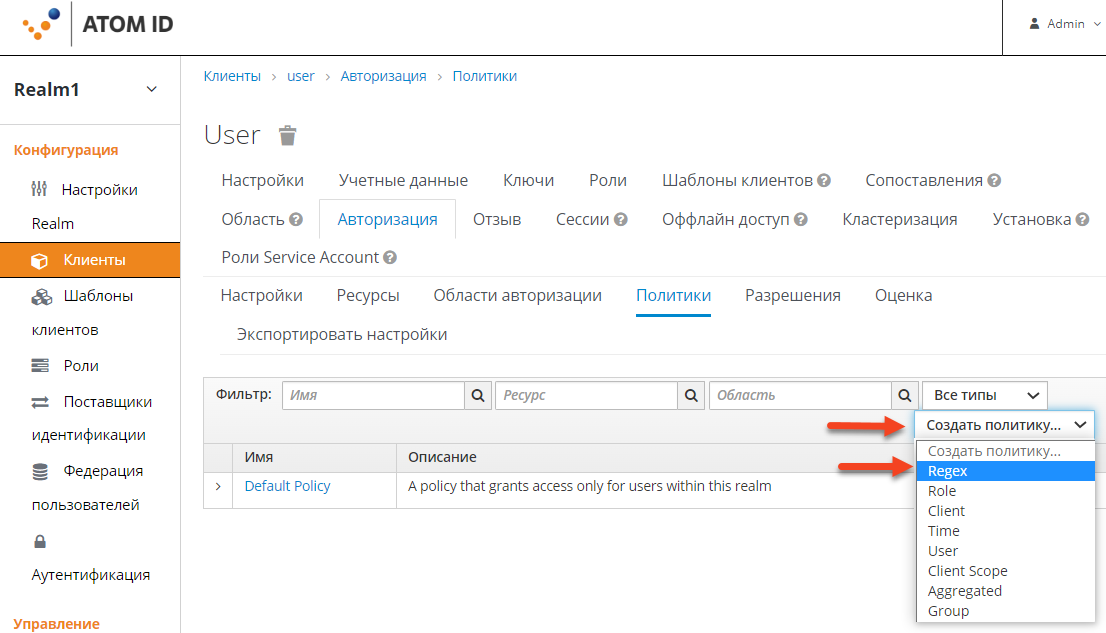


Рисунок 37 - Создание полики, основанной на регулярных выражениях

Чтобы создать новую политику на основе регулярных выражений, выберите «Regex» (Рисунок 37) в выпадающем списке при нажатии кнопки «Создать политику».

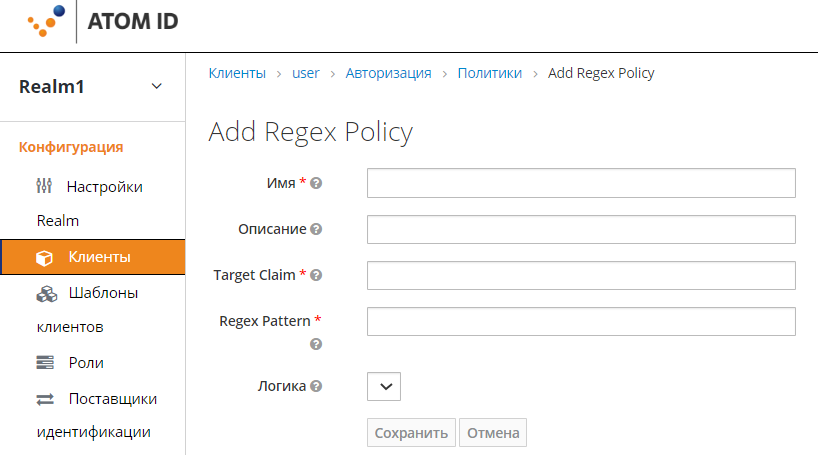


Рисунок 38 - Форма «Add Regex Policy»

В результате чего откроется форма «Add Regex Policy» (Рисунок 38).

* + - 1. Состав

Форма «Add Client Scope Policy» имеет следующй состав полей:

* поле «Имя» - должно быть уникальным, удобным к прочтению, однозначно описывающим политику. Рекомендуется использовать имена, которые тесно связаны с бизнесом и требованиями безопасности, чтобы было легче их идентифицировать;
* поле «Описание» - строка, содержащая подробную информацию об этой политике;
* поле «Target Claim» - указывает наименование целевого требований токена;
* поле «Regex Pattern» - задает шаблон регулярных выражений;
* поле «Logic» - логика применения создаваемой политики после оценки других условий. Имеет значения «Позитивная»/»Негативная».
  + 1. Позитивная и негативная логика

Политики могут быть настроены с положительной или отрицательной логикой. Существует возможность использовать этот параметр, чтобы определить, следует ли сохранить результат политики таким, какой он есть, или отменить его.

Например, для создания политики, в которой доступ должен предоставляться только пользователям, которым не предоставлена определенная роль, необходимо создать политику на основе ролей, используя эту роль, и установить для поля «Логика» значение «Отрицательное». При выборе значения «Положительная», что является значением по умолчанию, результат политики будет сохранен таким, какой он есть.

* 1. Управление разрешениями

Разрешение связывает защищаемый объект и политики, которые необходимо оценить, чтобы решить, следует ли предоставлять доступ.

После создания ресурсов, которые нужно защитить, и политик, которые планируется использовать для защиты этих ресурсов, необходимо приступить к управлению разрешениями.

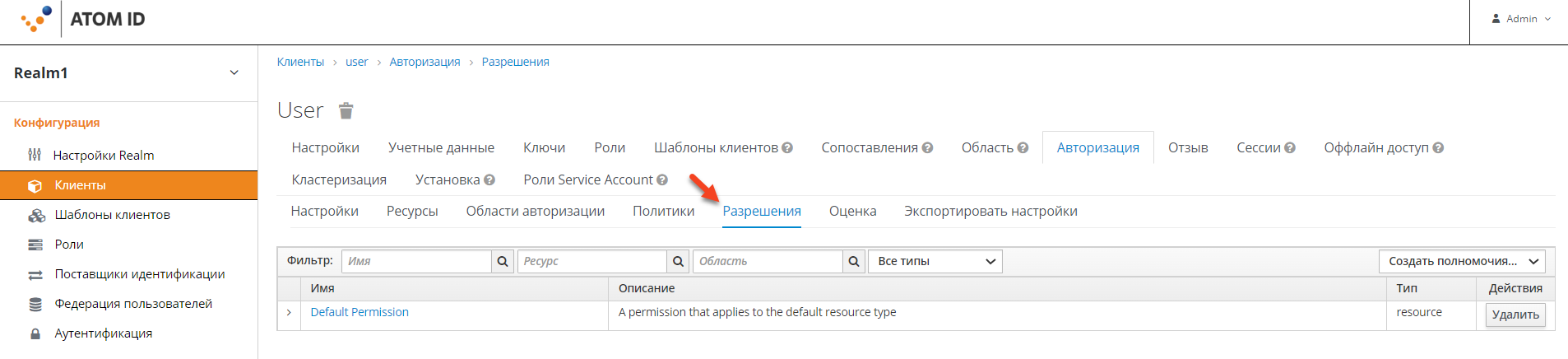


Рисунок 39 - Вкладка «Разрешения»

Чтобы управлять разрешениями, перейдите на вкладку «Разрешения» (Рисунок 39) при редактировании сервера ресурсов.

Разрешения могут быть созданы для защиты двух основных типов объектов:

* Ресурсы;
* Области применения.

Чтобы создать разрешение, выберите тип разрешения, который вы хотите создать, из выпадающего списка при нажатии кнопки «Создать полномочия». В следующих разделах эти два типа объектов описываются более подробно.

* + 1. Создание разрешения на основе ресурсов

Разрешение на основе ресурсов определяет набор из одного или нескольких ресурсов для защиты с использованием одной или нескольких политик авторизации.

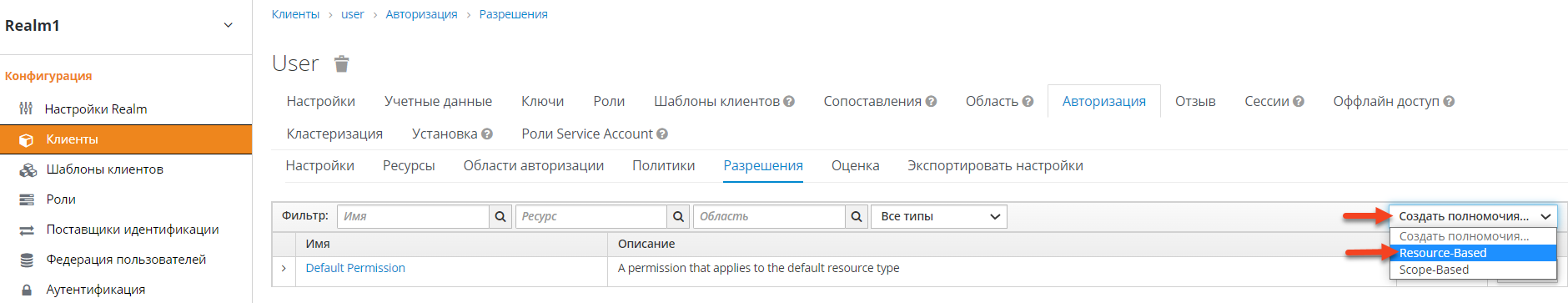


Рисунок 40 - Создание новых полномочий

Чтобы создать новое разрешение на основе ресурсов, нажмите кнопку «Создать полномочия», затем из выпадающего списка выберите значение «Resource Based» (Рисунок 40) в выпадающем списке.

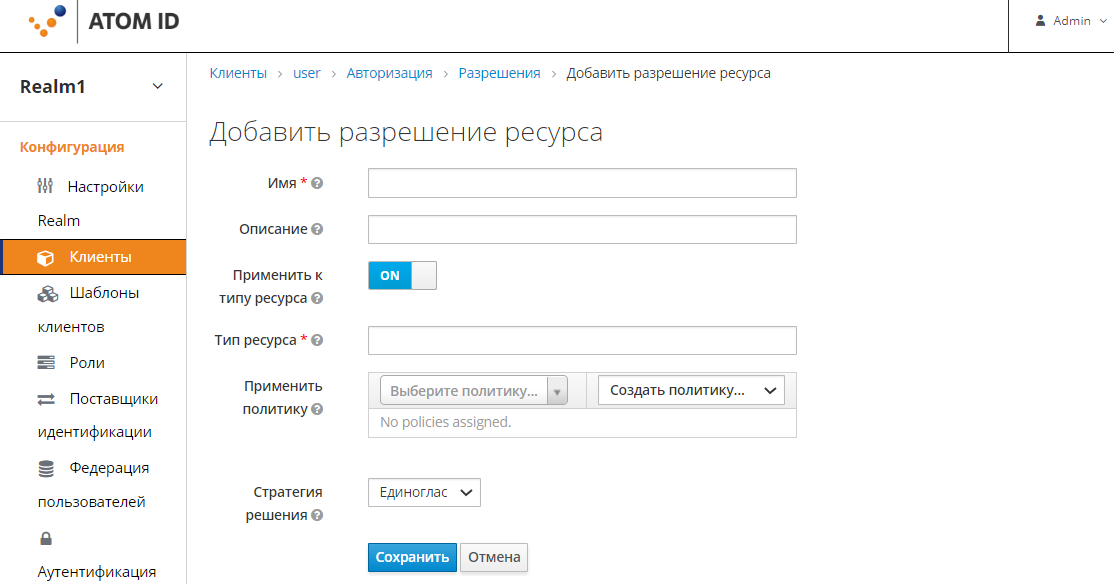


Рисунок 41 - Форма «Добавить разрешение ресурса»

В результате чего откроется форма «Добавить разрешение ресурса» (Рисунок 41).

* + - 1. Состав

Форма «Добавить разрешение ресурса» имеет следующй состав полей:

* поле «Имя» - должно быть уникальным, удобным к прочтению, однозначно описывающим политику. Рекомендуется использовать имена, которые тесно связаны с бизнесом и требованиями безопасности, чтобы было легче их идентифицировать;
* поле «Описание» - строка, содержащая подробную информацию об этой политике;
* поле «Применить к типу ресурса» - указывает, применяется ли разрешение ко всем ресурсам с заданным типом. При выборе этого поля будет предложено ввести тип ресурса для защиты;
* поле «Тип ресурса» - определяет тип ресурса для защиты. Когда это разрешение определено, оно оценивается для всех ресурсов, соответствующих этому типу;
* поле «Ресурсы» - определяет один или несколько ресурсов для защиты;
* поле «Применить политику» - определяет одну или нескольких политик для связи с разрешением. Чтобы связать политику, необходимо либо выбрать существующую политику, либо создать новую, выбрав тип политики, которую необходимо создать;
* поле «Стратегия решения» - стратегия принятия решения для разрешения.
  + - 1. Введенное разрешение на использование ресурсов

Разрешения на ресурсы также можно использовать для определения политик, которые должны применяться ко всем ресурсам с заданным типом. Эта форма разрешения на основе ресурсов может быть полезна, когда у вас есть ресурсы, разделяющие общие требования и ограничения доступа.

Часто ресурсы в приложении могут быть классифицированы (или типизированы) на основе данных, которые они инкапсулируют, или функциональности, которую они предоставляют. Например, финансовое приложение может управлять различными банковскими счетами, каждый из которых принадлежит определенному клиенту. Хотя это разные банковские счета, они имеют общие требования к безопасности и ограничения, которые глобально определяются банковской организацией. С помощью введенных разрешений на ресурсы вы можете определить общие политики, которые будут применяться ко всем банковским счетам, например:

* «Только владелец может управлять своей учетной записью»;
* «Разрешать доступ только из страны и/или региона владельца»;
* «Принудительное применение определенного метода проверки подлинности».

Чтобы создать типизированное разрешение на ресурс (Рисунок 42), переведите переключатель «Применить к типу ресурса» в положение «Включено» при создании нового разрешения на основе ресурсов. Если для параметра «Применить к типу ресурса» установлено значение «Включено», укажите тип, который необходимо защитить, а также политики, которые должны применяться для управления доступом ко всем ресурсам с указанным типом.

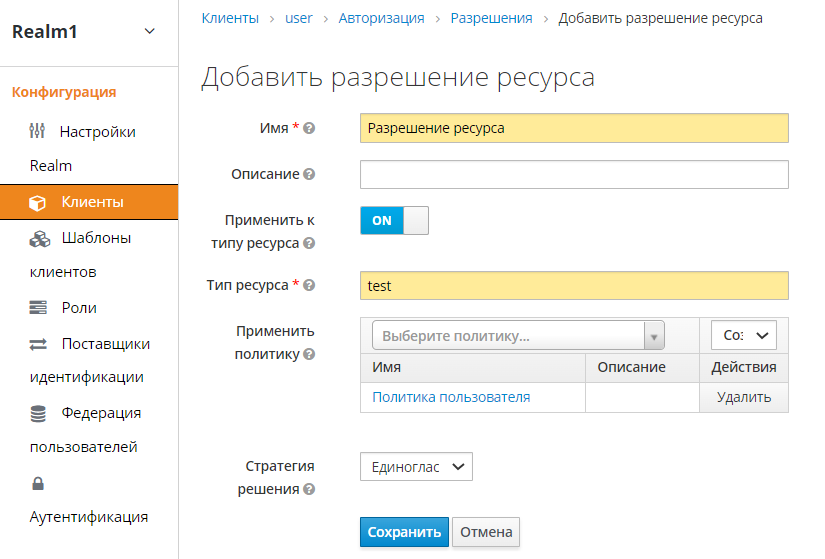


Рисунок 42 - Пример типизированного разрешения на использование ресурсов

* + 1. Создание разрешений области

Разрешение на основе области определяет одну или несколько областей для защиты с использованием одной или нескольких политик авторизации. В отличие от разрешений на основе ресурсов, можно использовать указанный тип разрешений для создания разрешений не только для ресурса, но и для связанных с ним областей, обеспечивая большую детализацию при определении разрешений, управляющих ресурсами, и действий, которые могут выполняться с ними.

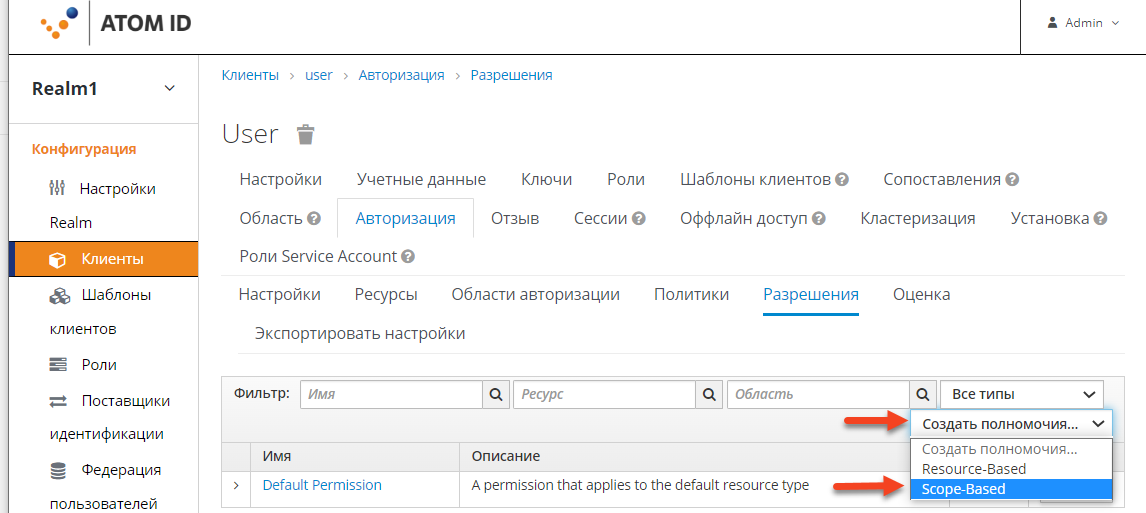


Рисунок 43 - Создание разрешений на основе области

Чтобы создать новое разрешение на основе области, выберите «Scope-Based» (Рисунок 43) в выпадающем списке при нажатии кнопки «Создать полномочия».

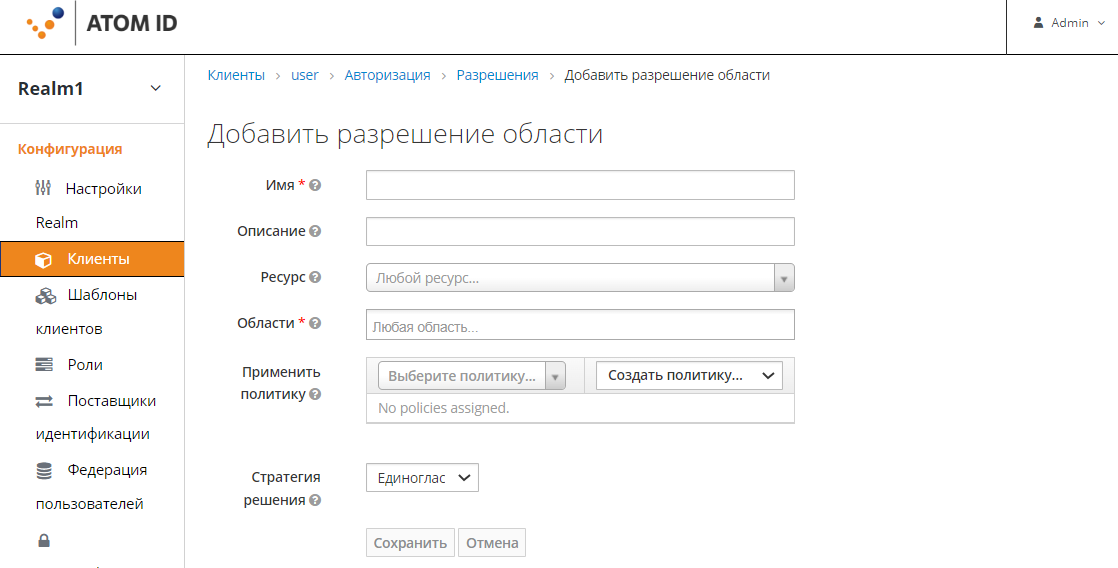


Рисунок 44 - Добавление разрешение области

В результате чего откроется форма «Добавить разрешение области» (Рисунок 44).

* + - 1. Состав

Форма «Добавить разрешение ресурса» имеет следующй состав полей:

* поле «Имя» - должно быть уникальным, удобным к прочтению, однозначно описывающим политику. Рекомендуется использовать имена, которые тесно связаны с бизнесом и требованиями безопасности, чтобы было легче их идентифицировать;
* поле «Описание» - строка, содержащая подробную информацию об этой политике;
* поле «Ресурс» - ограничивает области действия областями, связанными с выбранным ресурсом. Если не выбрано ни одно, доступны все области;
* поле «Области» - определяет одну или несколько областей для защиты;
* поле «Применить политику» - определяет одну или нескольких политик для связи с разрешением. Чтобы связать политику, необходимо либо выбрать существующую политику, либо создать новую, выбрав тип политики, которую необходимо создать;
* поле «Стратегия решения» - стратегия принятия решения для разрешения.
  + 1. Стратегии принятия решений политики

При связывании политик с разрешением существует возможность определения стратегии принятия решений, чтобы указать, как оценивать результаты связанных политик для определения доступа.

Стратегии принятия решений имеет следующие значения:

* «Единоглассная» - стратегия по умолчанию, если она не указана. В этом случае все политики должны быть оценены в положительное решение, чтобы окончательное решение также было положительным;
* «Утвердительная» - как минимум одна политика должна быть оценена как положительное решение, чтобы окончательное решение также было положительным;
* «Консенсусная» - количество положительных решений должно быть больше, чем количество отрицательных решений. Если количество положительных и отрицательных решений равно, то окончательное решение будет отрицательным.
  1. Оценка и тестирование политик

При разработке политик существует возможность имитировать запросы на авторизацию, с целью проверить, как оцениваются ваши политики.

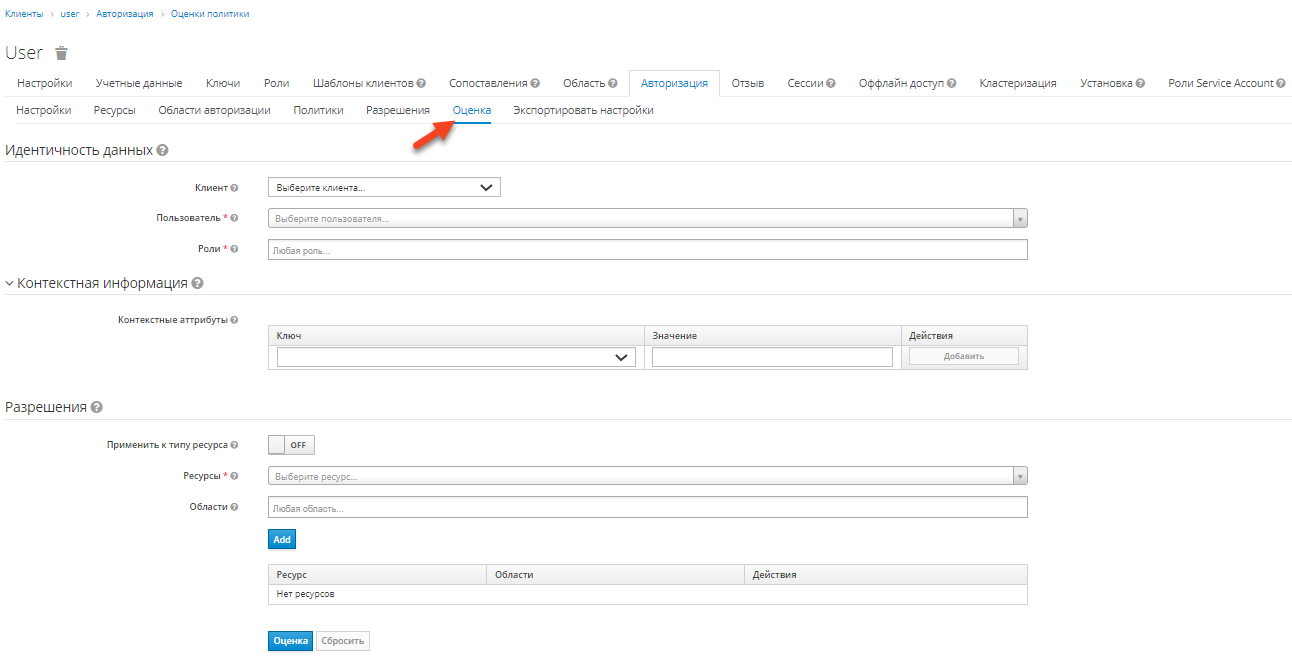


Рисунок 45 - Вкладка «Оценка», инструмент оценки политики

Для получения доступа к инструменту оценки политики, откройте вкладку «Оценка» (Рисунок 45) при редактировании сервера ресурсов. Затем необходимо указать различные входные данные для имитации реальных запросов на авторизацию и проверки действия политик.

* + 1. Предоставление идентификационной информации

Фильтры идентификационной информации можно использовать для указания пользователя, запрашивающего разрешения.

* + 1. Предоставление контекстуальной информации

Фильтры контекстной информации можно использовать для определения дополнительных атрибутов контекста оценки, чтобы политики могли получать эти же атрибуты.

* + 1. Предоставление разрешений

Фильтры разрешений можно использовать для создания запроса на авторизацию. Существует возможность запроса разрешения для набора из одного или нескольких ресурсов и областей.

Для имитирования запросов авторизации на основе всех защищенных ресурсов и областей, нажмите кнопку «Add» (Рисунок 46), не указывая никаких ресурсов или областей.

При указании нужного значения, нажмите кнопку «Оценка» (Рисунок 46).

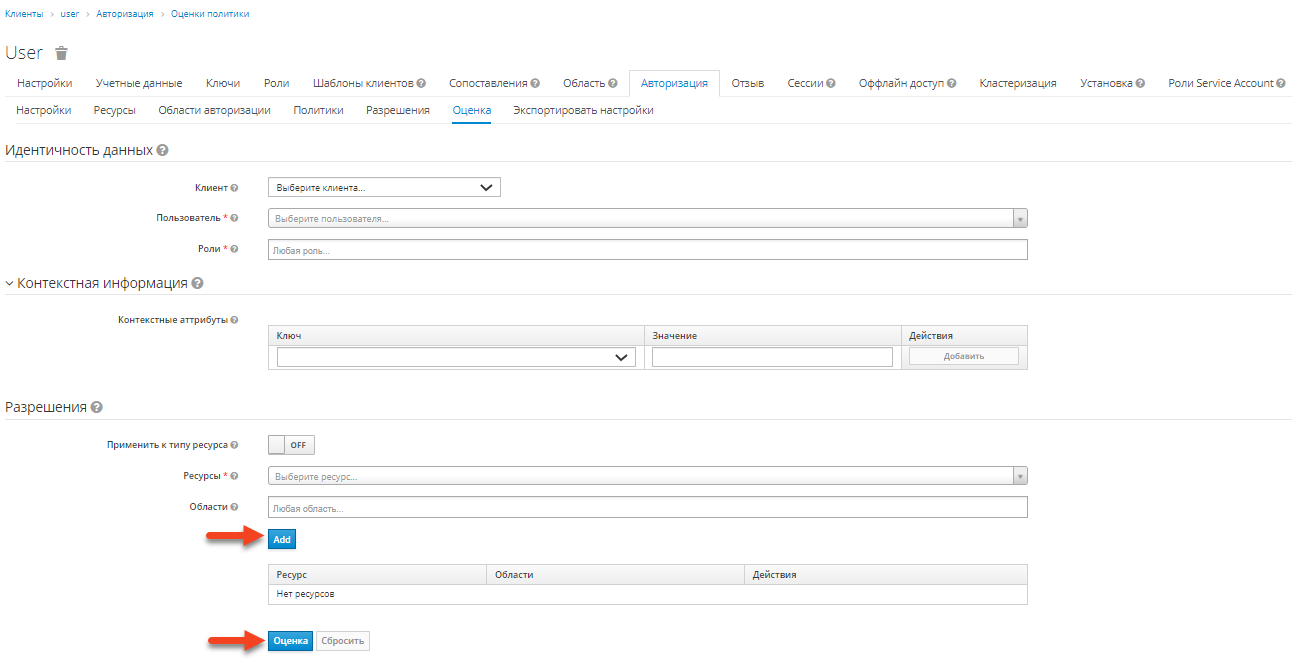


Рисунок 46 - Кнопки «Добавить», кнопка «Оценка»

* 1. Сервис авторизации

Службы авторизации AtomID построены на основе хорошо известных стандартов, таких как спецификации OAuth2 и управляемого пользователем доступа.

Клиенты OAuth2 (например, интерфейсные приложения) могут получать токены доступа с сервера, используя конечную точку токена, и использовать эти же токены для доступа к ресурсам, защищенным сервером ресурсов (например, к внутренним службам). Аналогичным образом, службы авторизации AtomID предоставляют расширения для OAuth2, позволяющие выдавать токены доступа на основе обработки всех политик, связанных с запрашиваемым ресурсом (ресурсами) или областью (областями). Это означает, что серверы ресурсов могут принудительно предоставлять доступ к своим защищенным ресурсам на основе разрешений, предоставленных сервером и удерживаемых токеном доступа. В службах авторизации AtomID токен доступа с разрешениями называется «Токеном запрашивающей стороны» или сокращенно RPT.

В дополнение к выдаче RPT, служба авторизации AtomID также предоставляет набор конечных точек RESTful, которые позволяют серверам ресурсов управлять своими защищенными ресурсами, областями, разрешениями и политиками, помогая разработчикам расширять или интегрировать эти возможности в свои приложения для поддержки детальной авторизации.

* + 1. Обнаружение конечных точек служб авторизации и метаданных

AtomID предоставляет документ обнаружения, из которого клиенты могут получить всю необходимую информацию для взаимодействия со службами авторизации AtomID, включая местоположения и возможности конечных точек.

Документ об обнаружении можно получить из:

curl -X GET \

http://${host}:${port}/realms/${realm}/.well-known/uma2-configuration

где ${host}:${port} – это имя хоста (или IP-адрес) и порт, где запущен AtomID и ${realm} – имя области AtomID.

В результате чего направляется следующий ответ:

{

// some claims are expected here

// these are the main claims in the discovery document about Authorization Services endpoints location

"token\_endpoint": "http://${host}:${port}/realms/${realm}/protocol/openid-connect/token",

"token\_introspection\_endpoint": "http://${host}:${port}/realms/${realm}/protocol/openid-connect/token/introspect",

"resource\_registration\_endpoint": "http://${host}:${port}/realms/${realm}/authz/protection/resource\_set",

"permission\_endpoint": "http://${host}:${port}/realms/${realm}/authz/protection/permission",

"policy\_endpoint": "http://${host}:${port}/realms/${realm}/authz/protection/uma-policy"

}

Каждая из этих конечных точек предоставляет определенный набор возможностей:

1. token\_endpoint. Конечная точка токена, совместимая с OAuth2, которая поддерживает urn:ietf:params:oauth:grant-type:uma-ticket для определения типа. Через эту конечную точку клиенты могут отправлять запросы на авторизацию и получать RPT со всеми разрешениями, предоставленными AtomID;
2. token\_introspection\_endpoint. Через эту конечную точку клиенты могут отправлять запросы на авторизацию и получать RPT со всеми разрешениями, предоставленными AtomID;
3. resource\_registration\_endpoint. Конечная точка регистрации ресурсов, совместимая с UMA, которую серверы ресурсов могут использовать для управления своими защищенными ресурсами и областями. Эта конечная точка обеспечивает операции создания, чтения, обновления и удаления ресурсов и областей в AtomID;
4. permission\_endpoint. Конечная точка разрешений, совместимая с UMA, которую серверы ресурсов могут использовать для управления тикетами разрешений. Эта конечная точка обеспечивает операции создания, чтения, обновления и удаления тикетов разрешений в AtomID.
   * 1. Получение разрешений

Чтобы получить разрешения от AtomID, необходимо отправить запрос на авторизацию конечной точке токена. В результате AtomID оценит все политики, связанные с запрашиваемым ресурсом (ресурсами) и областью (областями), и выдаст RPT со всеми разрешениями, предоставленными сервером.

Клиентам разрешено отправлять запросы на авторизацию в конечную точку токена, используя следующие параметры:

1. grant\_type. Этот параметр является обязательным. Должен быть: urn:ietf:params:oauth:grant-type:uma-ticket;
2. ticket. Этот параметр является необязательным. Самый последний тикет разрешения, полученный клиентом в рамках процесса авторизации UMA;
3. claim\_token. Этот параметр является необязательным. Строка, представляющая дополнительные утверждения, которые должны учитываться сервером при оценке разрешений для запрашиваемого ресурса (ресурсов) и области (областей). Этот параметр позволяет клиентам отправлять утверждения в AtomID. Для получения более подробной информации обо всех поддерживаемых форматах токенов смотри информацию о параметре claim\_token\_format;
4. claim\_token\_format. Этот параметр является необязательным. Строка, указывающая формат токена, указанного в параметре claim\_token. AtomID поддерживает два формата токенов:
5. urn:ietf:params:oauth:token-type:jwt - указывает, что параметр claim\_token ссылается на маркер доступа;
6. https://openid.net/specs/openid-connect-core-1\_0.html#IDToken - указывает, что параметр claim\_token ссылается на токен идентификатора подключения OpenID;
7. Rpt. Этот параметр является необязательным. Ранее выданный RPT, разрешения которого также должны быть оценены и добавлены в новый. Этот параметр позволяет клиентам, имеющим RPT, выполнять инкрементную авторизацию, при которой разрешения добавляются по требованию;
8. permission. Этот параметр является необязательным. Строка, представляющая набор из одного или нескольких ресурсов и областей, к которым клиент ищет доступ. Этот параметр может быть определен несколько раз, чтобы запросить разрешение для нескольких ресурсов и областей. Этот параметр является расширением для urn:ietf:params:oauth:grant-type:uma-ticket, определяющим тип разрешения, чтобы позволить клиентам отправлять запросы на авторизацию без тикетов. Формат строки должен быть: RESOURCE\_ID#SCOPE\_ID. Например: Resource A#Scope A, Resource A#Scope A, Scope B, Scope C, Resource A, #Scope A;
9. audience. Этот параметр является необязательным. Идентификатор клиента сервера ресурсов, к которому клиент запрашивает доступ. Этот параметр является обязательным в случае, если определен параметр разрешения. Это служит подсказкой для AtomID, чтобы указать контекст, в котором должны оцениваться разрешения;
10. response\_include\_resource\_name. Этот параметр является необязательным. Логическое значение, указывающее серверу, следует ли включать имена ресурсов в разрешения RPT. Если значение false, то включается только идентификатор ресурса;
11. response\_permissions\_limit. Этот параметр является необязательным. Целое число N, которое определяет ограничение на количество разрешений, которые может иметь RPT. При использовании вместе с параметром rpt в RPT будут сохранены только последние N запрошенных разрешений;
12. submit\_request. Этот параметр является необязательным. Логическое значение, указывающее, должен ли сервер создавать тикет разрешений для ресурсов и областей, на которые ссылается тикет разрешения. Этот параметр действует только в том случае, если используется вместе с параметром ticket как часть процесса авторизации UMA;
13. response\_mode. Этот параметр является необязательным. Строковое значение, указывающее, как сервер должен отвечать на запросы авторизации. Этот параметр особенно важен, когда необходимо либо общее решение, либо разрешения, предоставленные сервером, вместо стандартного ответа OAuth2. Возможными значениями являются:
14. decision - указывает, что ответы от сервера должны представлять только общее решение, возвращая JSON со следующим форматом:

{

'result': true

}

Если запрос авторизации не сопоставляется ни с каким разрешением, вместо этого возвращается код состояния HTTP 403;

1. permissions - указывает, что ответы от сервера должны содержать любые разрешения, предоставленные сервером, возвращая JSON со следующим форматом:

[

{

'rsid': 'My Resource'

'scopes': ['view', 'update']

},

...

]

Если запрос авторизации не сопоставляется ни с каким разрешением, вместо этого возвращается код состояния HTTP 403.

Пример запроса авторизации, когда клиент запрашивает доступ к двум ресурсам, защищенным сервером ресурсов:

curl -X POST \

http://${host}:${port}/realms/${realm}/protocol/openid-connect/token \

-H "Authorization: Bearer ${access\_token}" \

--data "grant\_type=urn:ietf:params:oauth:grant-type:uma-ticket" \

--data "audience={resource\_server\_client\_id}" \

--data "permission=Resource A#Scope A" \

--data "permission=Resource B#Scope B"

Пример запроса авторизации, когда клиент запрашивает доступ к любому ресурсу и области, защищенной сервером ресурсов.

curl -X POST \

http://${host}:${port}/realms/${realm}/protocol/openid-connect/token \

-H "Authorization: Bearer ${access\_token}" \

--data "grant\_type=urn:ietf:params:oauth:grant-type:uma-ticket" \

--data "audience={resource\_server\_client\_id}"

Пример запроса авторизации, когда клиент запрашивает доступ к ресурсу, защищенному UMA, после получения тикета разрешения от сервера ресурсов в рамках процесса авторизации:

curl -X POST \

http://${host}:${port}/realms/${realm}/protocol/openid-connect/token \

-H "Authorization: Bearer ${access\_token}" \

--data "grant\_type=urn:ietf:params:oauth:grant-type:uma-ticket" \

--data "ticket=${permission\_ticket}

Если процесс оценки AtomID приводит к выдаче разрешений, он выдает RPT, с которым он связал разрешения:

AtomID отвечает клиенту с помощью RPT:

HTTP/1.1 200 OK

Content-Type: application/json

...

{

"access\_token": "${rpt}",

}

Ответ от сервера такой же, как и любой другой ответ от конечной точки токена при использовании какого-либо другого типа гранта. RPT может быть получен из параметра ответа access\_token. Если клиент не авторизован, AtomID отвечает кодом состояния HTTP 403:

AtomID отклоняет запрос на авторизацию:

HTTP/1.1 403 Forbidden

Content-Type: application/json

...

{

"error": "access\_denied",

"error\_description": "request\_denied"

}

* + - 1. Методы аутентификации клиента

Клиенты должны пройти аутентификацию на конечной точке токена, чтобы получить RPT. При использовании urn:ietf:params:oauth:grant-type:uma-ticket типа предоставления, клиенты могут использовать любой из этих методов проверки подлинности:

1. Токен на предъявителя. Клиенты должны отправлять токен в качестве учетных данных на предъявителя в заголовке авторизации HTTP на конечную точку токена.

Пример: запрос авторизации с использованием токена доступа для аутентификации в конечной точке токена:

curl -X POST \

http://${host}:${port}/realms/${realm}/protocol/openid-connect/token \

-H "Authorization: Bearer ${access\_token}" \

--data "grant\_type=urn:ietf:params:oauth:grant-type:uma-ticket"

Этот метод особенно полезен, когда клиент действует от имени пользователя. В этом случае токен на предъявителя — это токен доступа, ранее выданный AtomID какому-либо клиенту, действующему от имени пользователя (или от своего имени). Разрешения будут оцениваться с учетом контекста доступа, представленного токеном доступа. Например, если токен доступа был выдан Клиенту A, действующему от имени Пользователя A, разрешения будут предоставлены в зависимости от ресурсов и областей, к которым Пользователь A имеет доступ;

1. Учетные данные клиента. Клиенты могут использовать любой из методов аутентификации клиентов, поддерживаемых AtomID. Например, client\_id/client\_secret или JWT.

Пример: запрос авторизации с использованием идентификатора клиента и секретного ключа клиента для аутентификации в конечной точке токена:

curl -X POST \

http://${host}:${port}/realms/${realm}/protocol/openid-connect/token \

-H "Authorization: Basic cGhvdGg6L7Jl13RmfWgtkk==pOnNlY3JldA==" \

--data "grant\_type=urn:ietf:params:oauth:grant-type:uma-ticket"

* + - 1. Предъявление требований

При получении разрешений с сервера существует возможность отправлять произвольные утверждения, чтобы эти утверждения были доступны для политик при оценке разрешений.

При получении разрешения от сервера без использования тикета разрешения (поток UMA), существует возможность отправить запрос на авторизацию конечной точке токена следующим образом:

curl -X POST \

http://${host}:${port}/realms/${realm}/protocol/openid-connect/token \

--data "grant\_type=urn:ietf:params:oauth:grant-type:uma-ticket" \

--data "claim\_token=ewogICAib3JnYW5pemF0aW9uIjogWyJhY21lIl0KfQ==" \

--data "claim\_token\_format=urn:ietf:params:oauth:token-type:jwt" \

--data "client\_id={resource\_server\_client\_id}" \

--data "client\_secret={resource\_server\_client\_secret}" \

--data "audience={resource\_server\_client\_id}"

Параметр claim\_token ожидает JSON в кодировке BASE64 с форматом, аналогичным приведенному ниже примеру:

{

"organization" : ["acme"]

}

Формат ожидает одно или несколько утверждений, где значение для каждого утверждения должно быть массивом строк.

**Предъявление требований с использованием UMA.**

Для получения более подробной информации о том, как отправлять заявки при использовании UMA и тикетов разрешений, пожалуйста, ознакомьтесь с API разрешений.

* + 1. Доступ, управляемый пользователем (UMA)

Службы авторизации AtomID основаны на управляемом пользователем доступе или сокращенно UMA. UMA — это спецификация, которая расширяет возможности OAuth2 следующими способами:

1. Конфиденциальность. В настоящее время конфиденциальность пользователей становится серьезной проблемой, поскольку все больше и больше данных и устройств доступны и подключены к облаку. С помощью UMA и AtomID серверы ресурсов могут расширить свои возможности, чтобы улучшить защиту своих ресурсов в отношении конфиденциальности пользователей, где разрешения предоставляются на основе политик, определенных пользователем;
2. Авторизация между сторонами. Владельцы ресурсов (например, обычные конечные пользователи) могут управлять доступом к своим ресурсам и разрешать другим сторонам (например, обычным конечным пользователям) доступ к этим ресурсам. Это отличается от OAuth2, где согласие дается клиентскому приложению, действующему от имени пользователя, при этом владельцам ресурсов UMA разрешается давать согласие на доступ другим пользователям полностью асинхронным образом;
3. Совместное использование ресурсов. Владельцам ресурсов разрешено управлять разрешениями на свои ресурсы и решать, кто и как может получить доступ к определенному ресурсу. Затем AtomID может выступать в качестве службы управления общим доступом, с помощью которой владельцы ресурсов могут управлять своими ресурсами.

AtomID — это сервер авторизации, совместимый с UMA 2.0, который предоставляет большинство возможностей UMA.

В качестве примера рассмотрим пользователя (владельца ресурса), использующего услугу Интернет-банка (сервер ресурсов) для управления своим банковским счетом (ресурсом). Допустим этот пользователь решает открыть свой банковский счет другому пользователю (запрашивающая сторона), специалисту по бухгалтерскому учету. Но у запрашивающей стороны должен быть доступ только на просмотр счета.

Как сервер ресурсов, Служба интернет-банка должна быть способна защитить банковский счет владельца счета. Для этого он использует конечную точку регистрации ресурса AtomID для создания ресурса на сервере, представляющего банковский счет пользователя.

В этот момент, если второй пользователь (запрашивающая сторона) попытается получить доступ к банковскому счету владельца, доступ будет запрещен. Служба интернет-банка определяет несколько политик по умолчанию для банковских счетов. Один из них заключается в том, что только владельцу разрешен доступ к его банковскому счету.

Однако услуга интернет-банкинга в отношении конфиденциальности пользователя-владельца счета также позволяет изменять определенные политики для банковского счета. Одна из этих политик, которую возможно изменить, — это определить, каким людям разрешено просматривать ее банковский счет. Для этого Услуга интернет-банкинга полагается на AtomID, чтобы предоставить владельцу счета пространство, где он может выбирать отдельных лиц и операции (или данные), к которым им разрешен доступ. В любое время владелец счета может отозвать доступ или предоставить дополнительные разрешения другому пользователю

* + - 1. Процесс авторизации

В UMA процесс авторизации начинается, когда клиент пытается получить доступ к серверу ресурсов, защищенному UMA.

Сервер ресурсов, защищенный UMA, ожидает токен-носитель в запросе, где токен является RPT. Когда клиент запрашивает ресурс на сервере ресурсов без RPT:

Клиент запрашивает защищенный ресурс без отправки RPT:

curl -X GET \

http://${host}:${port}/my-resource-server/resource/1bfdfe78-a4e1-4c2d-b142-fc92b75b986f

Сервер ресурсов отправляет ответ обратно клиенту с тикетом разрешения и параметром as\_uri с указанием местоположения сервера AtomID, куда должен быть отправлен тикет для получения RPT.

Сервер ресурсов отвечает с тикетом разрешения:

HTTP/1.1 401 Unauthorized

WWW-Authenticate: UMA realm="${realm}",

as\_uri="https://${host}:${port}/realms/${realm}",

ticket="016f84e8-f9b9-11e0-bd6f-0021cc6004de"

Тикет разрешения — это специальный тип токена, выдаваемый AtomID Permission API. Они представляют запрашиваемые разрешения (например, ресурсы и области), а также любую другую информацию, связанную с запросом. Только серверам ресурсов разрешено создавать эти токены.

Теперь, когда у клиента есть тикет разрешения, а также местоположение сервера AtomID, клиент может использовать документ обнаружения для получения местоположения конечной точки токена и отправки запроса на авторизацию.

Клиент отправляет запрос на авторизацию конечной точке токена для получения RPT:

curl -X POST \

http://${host}:${port}/realms/${realm}/protocol/openid-connect/token \

-H "Authorization: Bearer ${access\_token}" \

--data "grant\_type=urn:ietf:params:oauth:grant-type:uma-ticket" \

--data "ticket=${permission\_ticket}

Если процесс оценки AtomID приводит к выдаче разрешений, он выдает RPT, с которым он связал разрешения:

AtomID отвечает клиенту с помощью RPT:

HTTP/1.1 200 OK

Content-Type: application/json

...

{

"access\_token": "${rpt}",

}

Ответ от сервера такой же, как и любой другой ответ от конечной точки токена при использовании какого-либо другого типа гранта. RPT может быть получен из параметра ответа access\_token. В случае, если клиент не авторизован для получения разрешений AtomID отвечает кодом состояния HTTP 403:

AtomID отклоняет запрос на авторизацию:

HTTP/1.1 403 Forbidden

Content-Type: application/json

...

{

"error": "access\_denied",

"error\_description": "request\_denied"

}

* + - 1. Отправка запросов на получение разрешений

В рамках процесса авторизации клиентам сначала необходимо получить тикеты разрешения от сервера ресурсов, защищенного UMA, чтобы обменять его на RPT в конечной точке токена AtomID.

По умолчанию AtomID отвечает кодом состояния HTTP 403 и ошибкой request\_denied в случае, если клиенту не может быть выдан RPT.

AtomID отклоняет запрос на авторизацию:

HTTP/1.1 403 Forbidden

Content-Type: application/json

...

{

"error": "access\_denied",

"error\_description": "request\_denied"

}

Такой ответ подразумевает, что AtomID не смог выдать RPT с разрешениями, представленными тикетом разрешений.

В некоторых ситуациях клиентские приложения могут запустить асинхронный поток авторизации и позволить владельцу запрашиваемых ресурсов решить, следует ли предоставлять доступ или нет. Для этого клиенты могут использовать параметр запроса submit\_request вместе с запросом авторизации к конечной точке токена:

curl -X POST \

http://${host}:${port}/realms/${realm}/protocol/openid-connect/token \

-H "Authorization: Bearer ${access\_token}" \

--data "grant\_type=urn:ietf:params:oauth:grant-type:uma-ticket" \

--data "ticket=${permission\_ticket} \

--data "submit\_request=true"

При использовании параметра submit\_request AtomID сохранит запрос разрешения для каждого ресурса, доступ к которому был запрещен. После создания владельцы ресурсов могут проверять свою учетную запись и управлять запросами разрешений.

Данная функциональность может интерпретироваться как кнопка запроса доступа в приложении, где пользователи могут запрашивать у других пользователей доступ к своим ресурсам.

* + 1. Защита API

API защиты предоставляет UMA-совместимый набор конечных точек, обеспечивающих:

1. Управление ресурсами. С помощью этой конечной точки серверы ресурсов могут удаленно управлять своими ресурсами и позволяют исполнителям политик запрашивать у сервера ресурсы, которые нуждаются в защите;
2. Управление разрешениями. В протоколе UMA серверы ресурсов обращаются к этой конечной точке для создания тикетов разрешений. AtomID также предоставляет конечные точки для управления состоянием разрешений и запросов на разрешени;
3. API политики. AtomID использует API защиты UMA, чтобы позволить серверам ресурсов управлять разрешениями для своих пользователей. В дополнение к API ресурсов и разрешений AtomID предоставляет API политики, с помощью которого серверы ресурсов могут устанавливать разрешения для ресурсов от имени своих пользователей.

Важным требованием к этому API является то, что только серверам ресурсов разрешен доступ к его конечным точкам с использованием специального токена доступа OAuth2, называемого токеном API защиты (PAT). В UMA PAT — это токен с областью действия uma\_protection.

* + - 1. Что такое PAT и как его получить

Токен API защиты (PAT) — это специальный токен доступа OAuth2 с областью действия, определенной как uma\_protection. Когда вы создаете сервер ресурсов, AtomID автоматически создает роль uma\_protection для соответствующего клиентского приложения и связывает ее с учетной записью службы клиента.

Учетная запись службы, предоставленная с ролью uma\_protection, приведена на рисунке ниже (Рисунок 47).

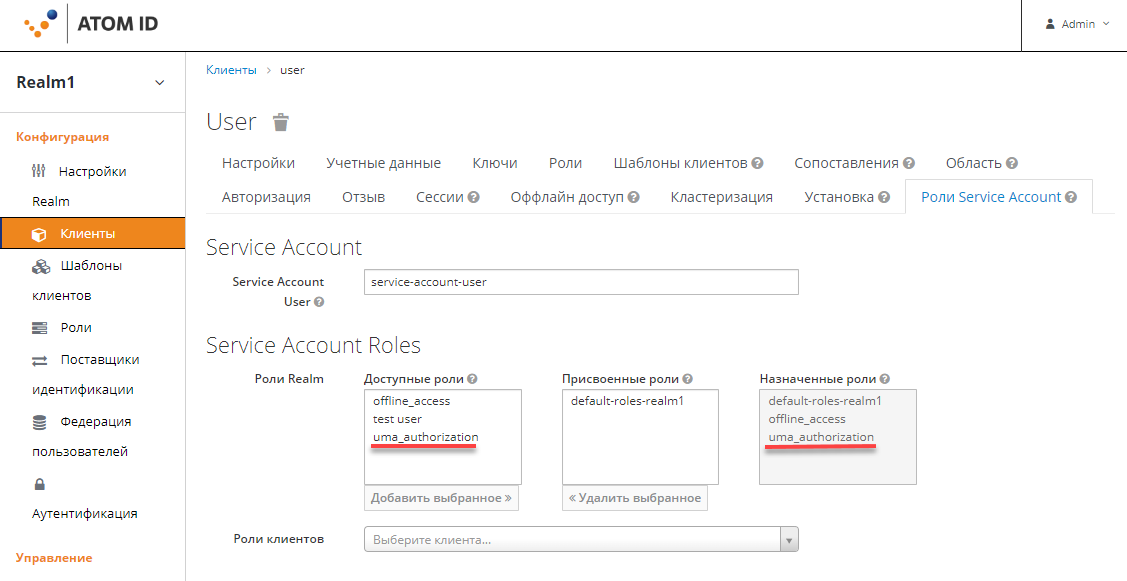


Рисунок 47 - Учетная запись с ролью uma\_protection

Серверы ресурсов могут получать PAT от AtomID, как и любой другой токен доступа OAuth2. Например, с помощью curl:

curl -X POST \

-H "Content-Type: application/x-www-form-urlencoded" \

-d 'grant\_type=client\_credentials&client\_id=${client\_id}&client\_secret=${client\_secret}' \

"http://localhost:8080/realms/${realm\_name}/protocol/openid-connect/token"

В приведенном выше примере используется тип предоставления client\_credentials для получения разрешения от сервера. В результате сервер возвращает ответ, аналогичный следующему:

{

"access\_token": ${PAT},

"expires\_in": 300,

"refresh\_expires\_in": 1800,

"refresh\_token": ${refresh\_token},

"token\_type": "bearer",

"id\_token": ${id\_token},

"not-before-policy": 0,

"session\_state": "ccea4a55-9aec-4024-b11c-44f6f168439e"

}

AtomID может аутентифицировать клиентское приложение различными способами. Для простоты здесь используется тип предоставления client\_credentials, для которого требуются client\_id и client\_secret. К выбору доступен любой поддерживаемый метод проверки подлинности.

* + - 1. Управление ресурсами

Серверы ресурсов могут управлять своими ресурсами удаленно, используя конечную точку, совместимую с UMA:

http://${host}:${port}/realms/${realm\_name}/authz/protection/resource\_set

Указанная конечная точка обеспечивает операции, описанные следующим образом:

* создание описания набора ресурсов: POST /resource\_set;
* просмотр описания набора ресурсов: GET /resource\_set/{\_id};
* обновление описания набора ресурсов: PUT /resource\_set/{\_id};
* удаление описания набора ресурсов: УДАЛИТЬ /resource\_set/{\_id};
* список описаний набора ресурсов: GET /resource\_set.

Дополнительные сведения о каждой из этих операций приведено в разделе «API регистрации ресурсов UMA».

**Создание ресурса.**

Чтобы создать ресурс, необходимо отправить HTTP POST-запрос следующим образом:

curl -v -X POST \

http://${host}:${port}/realms/${realm\_name}/authz/protection/resource\_set \

-H 'Authorization: Bearer '$pat \

-H 'Content-Type: application/json' \

-d '{

"name":"Tweedl Social Service",

"type":"http://www.example.com/rsrcs/socialstream/140-compatible",

"icon\_uri":"http://www.example.com/icons/sharesocial.png",

"resource\_scopes":[

"read-public",

"post-updates",

"read-private",

"http://www.example.com/scopes/all"

]

}'

По умолчанию владельцем ресурса является сервер ресурсов. Если необходимо определить другого владельца, например конкретного пользователя, тогда отправьте запрос следующим образом:

curl -v -X POST \

http://${host}:${port}/realms/${realm\_name}/authz/protection/resource\_set \

-H 'Authorization: Bearer '$pat \

-H 'Content-Type: application/json' \

-d '{

"name":"Alice Resource",

"owner": "alice"

}'

Владелец свойства может быть задан с помощью имени пользователя или идентификатора пользователя.

**Создание ресурсов, управляемых пользователем.**

По умолчанию ресурсы, созданные с помощью Protection API, не могут управляться владельцами ресурсов через Службу учетных записей пользователей.

Чтобы создать ресурсы и разрешить владельцам ресурсов управлять этими ресурсами, необходимо задать свойство ownerManagedAccess следующим образом:

curl -v -X POST \

<http://$>{host}:${port}/realms/${realm\_name}/authz/protection/resource\_set \

-H 'Authorization: Bearer '$pat \

-H 'Content-Type: application/json' \

-d '{

«name»:»Alice Resource»,

«owner»: «alice»,

«ownerManagedAccess»: true

}'

**Обновление ресурсов.**

Чтобы обновить существующий ресурс, необходимо отправить HTTP-запрос PUT следующим образом:

curl -v -X PUT \

http://${host}:${port}/realms/${realm\_name}/authz/protection/resource\_set/{resource\_id} \

-H 'Authorization: Bearer '$pat \

-H 'Content-Type: application/json' \

-d '{

"\_id": "Alice Resource",

"name":"Alice Resource",

"resource\_scopes": [

"read"

]

}'

**Удаление ресурсов.**

Чтобы удалить существующий ресурс, необходимо отправить HTTP-запрос на удаление следующим образом:

curl -v -X DELETE \

http://${host}:${port}/realms/${realm\_name}/authz/protection/resource\_set/{resource\_id} \

-H 'Authorization: Bearer '$pat

**Запрос ресурсов.**

Чтобы выполнить запрос ресурсов по идентификатору, необходимо отправить HTTP-запрос на HTTP GET-запрос следующим образом:

http://${host}:${port}/realms/${realm\_name}/authz/protection/resource\_set/{resource\_id}

Чтобы выполнить запрос ресурсов по имени, необходимо отправить HTTP GET-запрос следующим образом:

http://${host}:${port}/realms/${realm\_name}/authz/protection/resource\_set?name=Alice Resource

По умолчанию фильтр по параметру «Имя» будет сопоставлять любой ресурс с заданным шаблоном. Чтобы ограничить запрос возвратом только ресурсов с точным совпадением, необходимо использовать:

http://${host}:${port}/realms/${realm\_name}/authz/protection/resource\_set?name=Alice Resource&exactName=true

Чтобы запросить ресурсы с заданным uri, необходимо отправить HTTP GET-запрос следующим образом:

http://${host}:${port}/realms/${realm\_name}/authz/protection/resource\_set?uri=/api/alice

Чтобы запросить ресурсы с указанием владельца, необходимо отправить HTTP GET-запрос следующим образом:

http://${host}:${port}/realms/${realm\_name}/authz/protection/resource\_set?owner=alice

Чтобы запросить ресурсы с заданным типом, необходимо отправить HTTP GET-запрос следующим образом:

http://${host}:${port}/realms/${realm\_name}/authz/protection/resource\_set?type=albums

Чтобы запросить ресурсы с заданной областью действия, необходимо отправить HTTP GET-запрос следующим образом:

http://${host}:${port}/realms/${realm\_name}/authz/protection/resource\_set?scope=read

При запросе к серверу разрешений необходимо использовать параметры first и max results, чтобы ограничить результат.

* + - 1. Управление тикетами разрешений

Серверы ресурсов, использующие протокол UMA, могут использовать определенную конечную точку для управления запросами разрешений. Эта конечная точка обеспечивает поток, совместимый с UMA, для регистрации запросов на получение разрешений и получения тикета разрешения:

http://${host}:${port}/realms/${realm\_name}/authz/protection/permission

Тикет разрешения — это специальный тип токена безопасности, представляющий запрос разрешения. В соответствии со спецификацией UMA, тикет разрешения является дескриптором корреляции, который передается от сервера авторизации к серверу ресурсов, от сервера ресурсов к клиенту и, в конечном счете, от клиента обратно к серверу авторизации, чтобы позволить серверу авторизации оценить правильные политики для применения к запросу данных авторизации.

В большинстве случаев нет необходимости иметь дело с этой конечной точкой напрямую. AtomID предоставляет средство обеспечения соблюдения политики, которое включает UMA для сервера ресурсов, чтобы он мог получать тикет разрешения от сервера авторизации, возвращать этот тикет клиентскому приложению и применять решения об авторизации на основе конечного токена запрашивающей стороны (RPT).

Процесс получения тикетов разрешений от AtomID выполняется серверами ресурсов, а не обычными клиентскими приложениями, где тикеты разрешений получаются, когда клиент пытается получить доступ к защищенному ресурсу без необходимых разрешений для доступа к ресурсу. Выдача тикетов разрешений является важным аспектом при использовании UMA, поскольку это позволяет серверам ресурсов:

* получать от клиентов данные, связанных с ресурсами, защищенными сервером ресурсов;
* регистрироваться в запросах авторизации AtomID, которые, в свою очередь, могут быть использованы позже в рабочих процессах для предоставления доступа на основе согласия владельца ресурса;
* разделение серверов ресурсов и серверов авторизации и получать ими разрешение защищать ресурсы и управлять ими с помощью разных серверов авторизации.

Это лишь некоторые из преимуществ, предоставляемых UMA, в то время как другие аспекты UMA сильно зависят от тикетов разрешений, особенно в отношении конфиденциальности и контролируемого пользователями доступа к их ресурсам.

**Создание тикета разрешений.**

Чтобы создать тикет разрешения, необходимо отправить HTTP POST-запрос следующим образом:

curl -X POST \

http://${host}:${port}/realms/${realm\_name}/authz/protection/permission \

-H 'Authorization: Bearer '$pat \

-H 'Content-Type: application/json' \

-d '[

{

"resource\_id": "{resource\_id}",

"resource\_scopes": [

"view"

]

}

]'

При создании билетов существует возможность отправлять произвольные требования и связывать эти требования с тикетом:

curl -X POST \

http://${host}:${port}/realms/${realm\_name}/authz/protection/permission \

-H 'Authorization: Bearer '$pat \

-H 'Content-Type: application/json' \

-d '[

{

"resource\_id": "{resource\_id}",

"resource\_scopes": [

"view"

],

"claims": {

"organization": ["acme"]

}

}

]'

Где эти утверждения будут доступны для политик при оценке разрешений для ресурса и области (областей), связанных с тикетом разрешения.

**Другие конечные точки, не совместимые с UMA.**

**Создание тикетов разрешений.**

Чтобы предоставить разрешения для определенного ресурса с идентификатором {resource\_id} пользователю с идентификатором {user\_id}, как владельцу ресурса отправьте HTTP POST-запрос следующим образом:

curl -X POST \

http://${host}:${port}/realms/${realm\_name}/authz/protection/permission/ticket \

-H 'Authorization: Bearer '$access\_token \

-H 'Content-Type: application/json' \

-d '{

"resource": "{resource\_id}",

"requester": "{user\_id}",

"granted": true,

"scopeName": "view"

}'

**Получение тикетов разрешений.**

curl http://${host}:${port}/realms/${realm\_name}/authz/protection/permission/ticket \

-H 'Authorization: Bearer '$access\_token

Необходимо использовать любой из нижепредставленных параметров запроса:

* scopeId;
* resourceId;
* owner;
* requester;
* granted;
* returnNames;
* first;
* max.

**Обновление тикета разрешения.**

curl -X PUT \

http://${host}:${port}/realms/${realm\_name}/authz/protection/permission/ticket \

-H 'Authorization: Bearer '$access\_token \

-H 'Content-Type: application/json' \

-d '{

"id": "{ticket\_id}"

"resource": "{resource\_id}",

"requester": "{user\_id}",

"granted": false,

"scopeName": "view"

}'

**Удаление тикета разрешений.**

curl -X DELETE http://${host}:${port}/realms/${realm\_name}/authz/protection/permission/ticket/{ticket\_id} \

-H 'Authorization: Bearer '$access\_token

* + - 1. Управление разрешениями на ресурсы с помощью API политики

AtomID использует API защиты UMA, чтобы позволить серверам ресурсов управлять разрешениями для своих пользователей. В дополнение к API ресурсов и разрешений AtomID предоставляет API политики, с помощью которых серверы ресурсов могут устанавливать разрешения для ресурсов от имени своих пользователей.

API политики доступен по адресу:

http://${host}:${port}/realms/${realm\_name}/authz/protection/uma-policy/{resource\_id}

Этот API защищен токеном на предъявителя, который должен репрезентовать согласие, предоставленное пользователем серверу ресурсов для управления разрешениями от его имени. Токен-предъявитель может быть обычным токеном доступа, полученным из конечной точки токена с помощью:

* Владелец ресурса-Пароль-Учетные данные-Тип предоставления;
* Обмена токенами. Чтобы обменять токен доступа, предоставленный некоторому клиенту (общедоступному клиенту), на токен, где аудитория является сервером ресурсов.

**Связывание разрешения с ресурсом.**

Чтобы связать разрешение с определенным ресурсом, необходимо отправить HTTP POST-запрос следующим образом:

curl -X POST \

http://localhost:8180/realms/photoz/authz/protection/uma-policy/{resource\_id} \

-H 'Authorization: Bearer '$access\_token \

-H 'Cache-Control: no-cache' \

-H 'Content-Type: application/json' \

-d '{

"name": "Any people manager",

"description": "Allow access to any people manager",

"scopes": ["read"],

"roles": ["people-manager"]

}'

В приведенном выше примере создается и связывается новое разрешение с ресурсом, представленным resource\_id, где любому пользователю с ролью people-manager должна быть предоставлена область с доступом «Чтение».

Существует возможность создавать политики, используя другие механизмы контроля доступа, например, используя группы:

curl -X POST \

http://localhost:8180/realms/photoz/authz/protection/uma-policy/{resource\_id} \

-H 'Authorization: Bearer '$access\_token \

-H 'Cache-Control: no-cache' \

-H 'Content-Type: application/json' \

-d '{

"name": "Any people manager",

"description": "Allow access to any people manager",

"scopes": ["read"],

"groups": ["/Managers/People Managers"]

}'

или конкретный клиент:

curl -X POST \

http://localhost:8180/realms/photoz/authz/protection/uma-policy/{resource\_id} \

-H 'Authorization: Bearer '$access\_token \

-H 'Cache-Control: no-cache' \

-H 'Content-Type: application/json' \

-d '{

"name": "Any people manager",

"description": "Allow access to any people manager",

"scopes": ["read"],

"clients": ["my-client"]

}'

Также можно установить любую комбинацию этих механизмов контроля доступа.

Чтобы обновить существующее разрешение, необходимо отправить HTTP PUT -запрос следующим образом:

curl -X PUT \

http://localhost:8180/realms/photoz/authz/protection/uma-policy/{permission\_id} \

-H 'Authorization: Bearer '$access\_token \

-H 'Content-Type: application/json' \

-d '{

"id": "21eb3fed-02d7-4b5a-9102-29f3f09b6de2",

"name": "Any people manager",

"description": "Allow access to any people manager",

"type": "uma",

"scopes": [

"album:view"

],

"logic": "POSITIVE",

"decisionStrategy": "UNANIMOUS",

"owner": "7e22131a-aa57-4f5f-b1db-6e82babcd322",

"roles": [

"user"

]

}'

**Удаление разрешения.**

Чтобы удалить разрешение, связанное с ресурсом, необходимо отправить HTTP-запрос на удаление следующим образом:

curl -X DELETE \

http://localhost:8180/realms/photoz/authz/protection/uma-policy/{permission\_id} \

-H 'Authorization: Bearer '$access\_token

**Запрос разрешения.**

Чтобы запросить разрешения, связанные с ресурсом, необходимо отправить HTTP GET -запрос следующим образом:

http://${host}:${port}/realms/${realm}/authz/protection/uma-policy?resource={resource\_id}

Чтобы запросить разрешения, указанные в его имени, необходимо отправить HTTP GET -запрос следующим образом:

http://${host}:${port}/realms/${realm}/authz/protection/uma-policy?name=Any people manager

Чтобы запросить разрешения, связанные с определенной областью, необходимо отправить HTTP GET -запрос следующим образом:

http://${host}:${port}/realms/${realm}/authz/protection/uma-policy?scope=read

Чтобы запросить все разрешения, необходимо отправить HTTP GET -запрос следующим образом:

http://${host}:${port}/realms/${realm}/authz/protection/uma-policy?scope=read

Чтобы запросить все разрешения, необходимо отправить HTTP GET -запрос следующим образом:

http://${host}:${port}/realms/${realm}/authz/protection/uma-policy

При запросе к серверу разрешений используйте параметры first и max results, чтобы ограничить результат.

* + 1. Токен запрашивающей стороны

Токен запрашивающей стороны (RPT) — это веб-токен JSON (JWT), подписанный цифровой подписью с использованием веб-подписи JSON (JWS). Токен создается на основе токена доступа OAuth2, ранее выданного AtomID конкретному клиенту, действующему от имени пользователя или от своего имени.

При расшифровке RPT отображается ориентировочно следующая информация:

{

"authorization": {

"permissions": [

{

"resource\_set\_id": "d2fe9843-6462-4bfc-baba-b5787bb6e0e7",

"resource\_set\_name": "Hello World Resource"

}

]

},

"jti": "d6109a09-78fd-4998-bf89-95730dfd0892-1464906679405",

"exp": 1464906971,

"nbf": 0,

"iat": 1464906671,

"sub": "f1888f4d-5172-4359-be0c-af338505d86c",

"typ": "kc\_ett",

"azp": "hello-world-authz-service"

}

С помощью этого токена существует возможность получить все разрешения, предоставленные сервером, из требованийразрешений.

Дополнительно, разрешения напрямую связаны с ресурсами/областями, которые подлежат защите, и полностью отделены от методов контроля доступа, которые использовались для фактического предоставления и выдачи этих же разрешений.

* + - 1. Самоанализ запрашиваемой стороны

Иногда может потребоваться провести самоанализ токена запрашивающей стороны (RPT), чтобы проверить его действительность или получить разрешения внутри токена для принудительного выполнения решений об авторизации на стороне сервера ресурсов.

Существует два основных варианта использования, в которых необходимо использование самоанализа токенов:

1. Когда клиентским приложениям необходимо запросить действительность токена для получения нового с теми же или дополнительными разрешениями;
2. При принудительном исполнении решений об авторизации на стороне сервера ресурсов, особенно когда ни один из встроенных средств обеспечения соблюдения политик не подходит для приложения.
   * + 1. Получение информации о RPT

Самоанализ токена – это конечная точка, совместимая с самоанализом токена OAuth2, с которой существует возможность получать информацию о RPT.

http://${host}:${port}/realms/${realm\_name}/protocol/openid-connect/token/introspect

Чтобы проанализировать RPT с использованием этой конечной точки, необходимо отправить запрос на сервер следующим образом:

curl -X POST \

-H "Authorization: Basic aGVsbG8td29ybGQtYXV0aHotc2VydmljZTpzZWNyZXQ=" \

-H "Content-Type: application/x-www-form-urlencoded" \

-d 'token\_type\_hint=requesting\_party\_token&token=${RPT}' \

"http://localhost:8080/realms/hello-world-authz/protocol/openid-connect/token/introspect"

Приведенный запрос использует за основу HTTP и передает учетные данные клиента (идентификатор клиента и секрет) для аутентификации клиента, пытающегося проанализировать токен, но также есть возможность использовать любой другой метод аутентификации клиента, поддерживаемый AtomID.

Конечная точка самоанализа использует два параметра:

* token\_type\_hint. Используйте requesting\_party\_token в качестве значения для этого параметра, который указывает о потребности провести самоанализ RPT;
* token. Используйте строку токена в том виде, в каком она была возвращена сервером во время процесса авторизации, в качестве значения для этого параметра.

В результате будет следующий ответ сервера:

{

"permissions": [

{

"resource\_id": "90ccc6fc-b296-4cd1-881e-089e1ee15957",

"resource\_name": "Hello World Resource"

}

],

"exp": 1465314139,

"nbf": 0,

"iat": 1465313839,

"aud": "hello-world-authz-service",

"active": true

}

Если RPT не активен, возвращается следующий ответ:

{

"active": false

}

* + - 1. О необходимости вызывать сервер каждый раз, когда необходимо самостоятельно провести самоанализ RPT

Нет необходимости вызывать сервер каждый раз, когда необходимо самостоятельно провести самоанализ RPT. Как и обычный токен доступа, выдаваемый сервером AtomID, RPTS также использует спецификацию веб-токен JSON (JWT) в качестве формата по умолчанию.

Если есть необходимость проверить эти токены без вызова удаленной конечной точки самоанализа, существует возможность расшифровать RPT и запросить его достоверность локально. После расшифровки токена также можно использовать разрешения внутри токена для принудительного выполнения решений об авторизации. По сути, это то, что делают разработчики политики.

Необходимо выполнить следующие проверки:

* проверить подписи RPT (на основе открытого ключа области);
* запроса на валидность токена основывался на его утверждениях exp, iat и aud.

Дополнительные ресурсы

* веб-токен JSON (JWT);
* исполнители политики.