**Текст выступления: Keycloak - интеграция с брокерами и федерация идентификации**

1. Введение

**1.1. Что такое Keycloak? Основные возможности**

Добрый день, коллеги! Сегодня я расскажу вам о Keycloak и его возможностях по интеграции с брокерами идентификации и федерации.

Итак, что такое Keycloak? Если говорить простыми словами, Keycloak — это решение с открытым исходным кодом для управления доступом и идентификацией пользователей. Представьте себе, что у вас есть "швейцар" или "охранник", который стоит перед входом в ваше приложение и проверяет, кто может войти, а кто нет.

Основные возможности Keycloak:

Единый вход (SSO): пользователю достаточно авторизоваться один раз, чтобы получить доступ ко всем приложениям

Централизованное управление пользователями: создание, удаление и изменение пользователей в одном месте

Двухфакторная аутентификация: дополнительный уровень защиты через SMS, приложения-аутентификаторы и т.д.

Социальный вход: возможность входа через учетные записи Google, Facebook, GitHub и других провайдеров

Поддержка современных протоколов: OpenID Connect, OAuth 2.0, SAML

Настраиваемые страницы входа: кастомизация под фирменный стиль компании

API для управления: возможность программного управления всеми аспектами системы

Keycloak можно представить как конструктор Lego для построения системы безопасности — вы выбираете только те блоки, которые вам нужны.

**1.2. Зачем нужна интеграция с брокерами и федерация идентификации?**

Теперь давайте разберемся, зачем нам может понадобиться интеграция с брокерами и федерация идентификации.

Представьте, что в вашей компании уже есть Active Directory или другая система для управления учетными записями. При этом вы внедряете новые приложения и не хотите, чтобы сотрудники создавали отдельные учетные записи для каждого приложения. Вот тут-то и приходят на помощь брокеры и федерация.

Брокеры идентификации позволяют вам делегировать аутентификацию внешним провайдерам. Например, вместо создания новой учетной записи пользователь может войти через свою корпоративную учетную запись Microsoft или через Google.

Федерация идентификации идет еще дальше и позволяет синхронизировать учетные данные между Keycloak и существующими хранилищами пользователей, такими как LDAP или Active Directory.

**Active Directory (AD)** — это **сервис управления и хранения информации о пользователях, компьютерах и других ресурсах в компьютерных сетях**. Он предоставляет централизованное хранилище для всех этих данных, позволяя организовывать и контролировать доступ к ним. [1](https://www.roksis.ru/articles/active-directory/)

AD организует информацию в виде иерархической структуры, что упрощает администрирование и поиск ресурсов.

**Некоторые возможности Active Directory:**

* **Централизованное управление пользователями и группами**. Позволяет администраторам создавать и управлять учётными записями пользователей и групп в сети. Это облегчает процесс аутентификации и авторизации пользователей, а также управление их доступом к ресурсам.
* **Управление безопасностью**. С помощью AD администраторы могут определять политики безопасности и применять их к пользователям и компьютерам в сети. [1](https://www.roksis.ru/articles/active-directory/)
* **Централизованное управление ресурсами**. Активный каталог позволяет управлять сетевыми ресурсами, такими как принтеры, файловые серверы, базы данных и другие. [1](https://www.roksis.ru/articles/active-directory/)
* **Интеграция с другими службами и приложениями**. AD интегрируется с другими службами и приложениями, такими как электронная почта, серверы приложений, системы управления базами данных и другие. Это позволяет пользователям использовать единые учётные данные для доступа к различным ресурсам и упрощает управление различными сервисами

Коротко говоря: **AD** - это база данных служб каталогов, а **LDAP** - один из протоколов, которые вы можете использовать для общения с ней. LDAP - это протокол, а Active Directory - это сервер.

Зачем это нужно:

Удобство для пользователей: им не нужно запоминать много паролей

Безопасность: централизованное управление доступом и политиками безопасности

Снижение нагрузки на администраторов: не нужно дублировать управление учетными записями

Соответствие корпоративным политикам: централизованный контроль за доступом

**1.3. Примеры реальных кейсов, где применяется такая интеграция**

Разберем несколько реальных примеров, где интеграция с брокерами и федерация идентификации действительно решают бизнес-задачи:

Пример 1: Крупная корпорация

У компании есть множество внутренних приложений и существующая система Active Directory. С помощью Keycloak и федерации идентификации сотрудники используют свои корпоративные учетные данные для доступа ко всем системам. При увольнении сотрудника блокировка его учетной записи в AD автоматически блокирует доступ ко всем приложениям.

Пример 2: Сервис для партнеров

Компания создала портал для партнеров, но не хочет управлять учетными записями всех партнеров. С помощью брокеров идентификации партнеры могут входить через свои корпоративные учетные записи, а компания предоставляет им доступ к нужным ресурсам.

Пример 3: B2C приложение

Приложение для широкой публики, которое позволяет пользователям входить через социальные сети (Google, Facebook) с помощью брокеров идентификации в Keycloak. Это упрощает процесс регистрации и увеличивает конверсию.

Пример 4: Образовательное учреждение

Университет использует Keycloak для интеграции со своей существующей системой учетных записей студентов и предоставления доступа к образовательным ресурсам, библиотеке и другим сервисам через единый вход.

***2. Основы аутентификации и авторизации***

**2.1. Разница между аутентификацией и авторизацией**

Прежде чем двигаться дальше, давайте разберемся с базовыми понятиями, которые часто путают.

Аутентификация — это процесс подтверждения личности пользователя. Проще говоря, это ответ на вопрос "Кто вы?". Когда вы вводите логин и пароль, система проверяет, действительно ли вы тот, за кого себя выдаете.

Авторизация — это процесс определения, что пользователь имеет право делать после того, как он аутентифицирован. Это ответ на вопрос "Что вам разрешено?". Например, обычный пользователь может просматривать данные, но не может их изменять, а администратор имеет полный доступ.

Представьте, что аутентификация — это проверка паспорта на входе в здание, а авторизация — это определение, в какие комнаты человек может войти внутри этого здания.

В Keycloak эти процессы разделены:

Аутентификация управляется через провайдеры аутентификации, брокеры и федерацию

Авторизация управляется через роли, группы и разрешения

**2.2. Протоколы SAML, OpenID Connect и OAuth2: сравнение**

Keycloak поддерживает несколько протоколов аутентификации и авторизации. Давайте разберем их простыми словами.

OAuth 2.0 — это протокол авторизации, который позволяет приложению получить ограниченный доступ к учетной записи пользователя на другом сервисе. Например, когда приложение просит разрешение использовать ваш аккаунт Google для входа.

**Ключевые понятия OAuth 2.0:**

Токены доступа: временные "пропуска", дающие право на определенные действия

Области действия (scopes): определяют, к каким данным есть доступ

Потоки (flows): различные сценарии получения токенов

OpenID Connect (OIDC) — это слой идентификации поверх OAuth 2.0. Если OAuth 2.0 отвечает за авторизацию ("что можно делать"), то OIDC добавляет аутентификацию ("кто вы"). OIDC добавляет ID-токен, который содержит информацию о пользователе.

SAML (Security Assertion Markup Language) — более старый, но все еще широко используемый протокол для единого входа. SAML работает через обмен XML-документами между поставщиком услуг (SP) и поставщиком идентификации (IdP).

Сравнение протоколов:

OAuth 2.0: фокусируется на авторизации, проще в реализации, популярен в веб и мобильных приложениях

OpenID Connect: расширяет OAuth 2.0, добавляя идентификацию, использует JSON Web Tokens (JWT)

SAML: более сложный, использует XML, часто применяется в корпоративных средах и устаревших системах

Keycloak поддерживает все эти протоколы, что делает его универсальным решением.

**2.3. Что такое Identity Brokering?**

Identity Brokering (брокер идентификации) — это механизм, который позволяет Keycloak делегировать аутентификацию внешнему провайдеру идентификации.

Простыми словами, это как если бы на входе в здание охранник говорил: "У меня нет вашего паспорта в базе, но я признаю пропуск из соседнего здания. Покажите его, и я вас пропущу".

В контексте Keycloak это означает, что пользователь может войти в ваше приложение, используя учетную запись из Google, Facebook, Microsoft, другого экземпляра Keycloak или любого поставщика, поддерживающего OpenID Connect или SAML.

Как это работает:

Пользователь пытается войти в приложение

Keycloak предлагает список доступных провайдеров идентификации

Пользователь выбирает провайдера (например, Google)

Keycloak перенаправляет пользователя на страницу входа выбранного провайдера

После успешной аутентификации у провайдера пользователь возвращается в Keycloak

Keycloak создает локальную сессию и перенаправляет пользователя в приложение

Преимущества использования брокеров:

Пользователи могут использовать существующие учетные записи

Нет необходимости управлять паролями для этих пользователей

Можно получать дополнительную информацию о пользователе от провайдера

**2.4. Что такое Identity Federation?**

Identity Federation (федерация идентификации) — это механизм, который позволяет Keycloak использовать внешние хранилища пользователей для аутентификации.

В отличие от брокера, где аутентификация полностью делегируется внешнему провайдеру, при федерации Keycloak напрямую взаимодействует с внешним хранилищем пользователей, таким как LDAP или Active Directory.

Представьте, что вместо создания копии базы данных пользователей, Keycloak просто "подключается" к существующей базе и проверяет учетные данные там.

Ключевые особенности федерации:

Пользователи могут аутентифицироваться напрямую в Keycloak, используя свои учетные данные из внешней системы

Атрибуты пользователей могут синхронизироваться между Keycloak и внешним хранилищем

Можно настроить кэширование для улучшения производительности

Возможна как полная, так и частичная синхронизация пользователей

Федерация особенно полезна в корпоративных средах, где уже есть централизованное управление пользователями через Active Directory или LDAP.

***3. Брокеры идентификации в Keycloak***

**3.1. Архитектура брокера в Keycloak**

Давайте разберемся, как устроены брокеры идентификации в Keycloak. Архитектура здесь достаточно простая, но гибкая.

В центре схемы находится Keycloak, который выступает в роли посредника между вашими приложениями и внешними провайдерами идентификации.

Ключевые компоненты архитектуры:

Identity Provider Interface: унифицированный интерфейс для всех типов провайдеров

Identity Provider SPI: механизм для подключения различных типов провайдеров

Маппер атрибутов: преобразует атрибуты из внешних провайдеров в формат Keycloak

Хранилище токенов: сохраняет токены от внешних провайдеров для дальнейшего использования

Правила связывания аккаунтов: определяют, как связывать внешние учетные записи с локальными

Keycloak поддерживает множество типов провайдеров идентификации:

Социальные: Google, Facebook, GitHub, Twitter, LinkedIn

Корпоративные: Azure AD, SAML v2.0, OpenID Connect

Специализированные: OpenShift, BitBucket и другие

Важно понимать, что Keycloak может одновременно выступать как в роли поставщика услуг (Service Provider), так и в роли поставщика идентификации (Identity Provider) в зависимости от настройки.

**3.2. Настройка Identity Brokering в Keycloak**

Теперь давайте посмотрим, как настроить брокер идентификации в Keycloak. Я покажу общий процесс, который применим для большинства провайдеров.

Войдите в административную консоль Keycloak

Откройте браузер и перейдите по адресу: http://<your-keycloak-host>/auth/admin/

Войдите, используя учетные данные администратора

Выберите нужный Realm

В верхнем левом углу находится выпадающий список с доступными Realm'ами

Выберите Realm, в котором хотите настроить брокер

Примечание: Realm — это изолированная среда в Keycloak, содержащая свои настройки, пользователей и приложения

Перейдите в раздел Identity Providers

В левом меню найдите раздел "Identity Providers"

Нажмите на него, чтобы увидеть список доступных провайдеров

Добавьте нового провайдера

Нажмите кнопку "Add provider" справа

Выберите тип провайдера из выпадающего списка (например, Google, GitHub, SAML v2.0 и т.д.)

Настройте базовые параметры провайдера

Alias: уникальное имя для этого провайдера в системе

Display Name: имя, которое будет отображаться на странице входа

Enabled: включает или отключает этот провайдер

Hide on Login Page: если включено, провайдер не будет отображаться на странице входа, но будет доступен через прямую ссылку

Укажите параметры аутентификации

Для OAuth2/OIDC провайдеров: Client ID и Client Secret

Для SAML провайдеров: метаданные или конфигурация конечных точек

Настройте дополнительные параметры (опционально)

Trust Email: доверять ли email, полученному от провайдера

Store Tokens: сохранять ли токены от провайдера

Account Linking Only: использовать только для связывания, а не для входа

First Login Flow: поток аутентификации для первого входа

Post Login Flow: поток аутентификации после входа

Настройте маппинг атрибутов (опционально)

Перейдите на вкладку "Mappers"

Нажмите "Create" для добавления нового маппера

Настройте отображение атрибутов провайдера на атрибуты Keycloak

Сохраните конфигурацию

Нажмите кнопку "Save" внизу страницы

После этого брокер идентификации будет настроен и появится на странице входа в виде дополнительной кнопки, позволяющей войти через выбранного провайдера.

**3.3. Подключение внешнего IdP (Google, GitHub, Azure AD)**

Давайте рассмотрим конкретные примеры настройки популярных внешних провайдеров идентификации.

Подключение Google

Создайте проект в Google Cloud Platform

Перейдите на console.cloud.google.com

Создайте новый проект или выберите существующий

Перейдите в раздел "APIs & Services" > "Credentials"

Настройте OAuth 2.0 Client ID

Нажмите "Create Credentials" и выберите "OAuth client ID"

Выберите тип приложения "Web application"

Укажите имя

В поле "Authorized redirect URIs" добавьте:

http://<your-keycloak-host>/auth/realms/<your-realm>/broker/google/endpoint

Нажмите "Create"

Запишите Client ID и Client Secret

Настройте Google IdP в Keycloak

В административной консоли Keycloak перейдите в "Identity Providers"

Выберите "Google" из списка

Укажите Client ID и Client Secret из Google Console

Включите опцию "Trust Email", так как Google проверяет email пользователей

Сохраните настройки

Подключение GitHub

Зарегистрируйте новое OAuth приложение в GitHub

Перейдите в свой профиль GitHub > Settings > Developer settings > OAuth Apps

Нажмите "New OAuth App"

Заполните форму:

Application name: любое понятное имя

Homepage URL: URL вашего приложения

Authorization callback URL:

http://<your-keycloak-host>/auth/realms/<your-realm>/broker/github/endpoint

Нажмите "Register application"

Запишите Client ID и сгенерируйте Client Secret

Настройте GitHub IdP в Keycloak

В административной консоли Keycloak перейдите в "Identity Providers"

Выберите "GitHub" из списка

Укажите Client ID и Client Secret из GitHub

Сохраните настройки

Подключение Azure AD

Зарегистрируйте приложение в Azure Portal

Перейдите в Azure Portal > Azure Active Directory > App registrations

Нажмите "New registration"

Укажите имя приложения

В качестве Redirect URI укажите:

http://<your-keycloak-host>/auth/realms/<your-realm>/broker/microsoft/endpoint

Нажмите "Register"

Запишите Application (client) ID

Создайте Client Secret

В зарегистрированном приложении перейдите в "Certificates & secrets"

Нажмите "New client secret"

Укажите описание и срок действия

Нажмите "Add"

Немедленно запишите значение секрета (оно будет показано только один раз)

Настройте Azure AD IdP в Keycloak

В административной консоли Keycloak перейдите в "Identity Providers"

Выберите "Microsoft" или "OpenID Connect" из списка

Если выбрали "Microsoft":

Укажите Client ID и Client Secret из Azure

Если выбрали "OpenID Connect":

Укажите Client ID и Client Secret из Azure

Укажите "Authorization URL": https://login.microsoftonline.com/{tenant-id}/oauth2/v2.0/authorize

Укажите "Token URL": https://login.microsoftonline.com/{tenant-id}/oauth2/v2.0/token

Укажите "User Info URL": https://graph.microsoft.com/oidc/userinfo

Замените {tenant-id} на ID вашего тенанта Azure AD

Сохраните настройки

Проверка работы

После настройки любого из этих провайдеров:

Выйдите из консоли администратора

Перейдите на страницу входа Keycloak

Вы должны увидеть кнопку входа через настроенного провайдера

Нажмите на эту кнопку и пройдите процесс аутентификации у провайдера

После успешной аутентификации вы будете перенаправлены обратно в Keycloak

**3.4. Разбор работы потока аутентификации через брокер**

Давайте подробно разберем, как работает аутентификация через брокер в Keycloak. Это поможет лучше понять процесс и решать возможные проблемы.

Полный поток аутентификации через брокера:

Инициация входа

Пользователь переходит на страницу входа Keycloak

Пользователь нажимает на кнопку внешнего провайдера (например, "Войти через Google")

Перенаправление к провайдеру

Keycloak создает запрос аутентификации к провайдеру

Пользователь перенаправляется на страницу входа провайдера

URL перенаправления содержит параметры:

client\_id: идентификатор клиента, зарегистрированного у провайдера

redirect\_uri: URL, на который провайдер должен вернуть пользователя

scope: запрашиваемые разрешения (например, "openid profile email")

state: случайная строка для предотвращения CSRF-атак

response\_type: тип ответа (обычно "code")

Аутентификация у провайдера

Пользователь вводит свои учетные данные у провайдера

Провайдер аутентифицирует пользователя

Провайдер просит подтвердить разрешения для приложения (если это первый вход)

Возврат в Keycloak с кодом авторизации

Провайдер перенаправляет пользователя на redirect\_uri

В URL содержится код авторизации и state

Keycloak проверяет state для защиты от CSRF-атак

Обмен кода на токены

Keycloak отправляет запрос к провайдеру с кодом авторизации

Провайдер проверяет код и возвращает токены:

access\_token: для доступа к API провайдера

id\_token: содержит информацию о пользователе (при OIDC)

refresh\_token: для обновления access\_token

Получение информации о пользователе

Keycloak использует access\_token для запроса информации о пользователе

Для OIDC: информация может содержаться в id\_token или запрашивается через userinfo endpoint

Для OAuth2: информация запрашивается через API провайдера

Для SAML: информация содержится в SAML-утверждении

Создание или обновление пользователя в Keycloak

Keycloak проверяет, существует ли пользователь с таким же email/username

Если пользователь уже существует и настроена опция связывания аккаунтов, происходит связывание

Если пользователя нет, создается новый пользователь (если разрешено)

Атрибуты пользователя обновляются согласно настроенным маппингам

Применение потоков аутентификации

Для первого входа: применяется First Login Flow

После входа: применяется Post Login Flow

Потоки могут включать:

Подтверждение электронной почты

Принятие условий использования

Запрос дополнительной информации

Настройку многофакторной аутентификации

Создание сессии Keycloak

Keycloak создает локальную сессию для пользователя

Генерируются собственные токены Keycloak для приложений

Токены от внешнего провайдера сохраняются (если настроено)

Перенаправление в приложение

Пользователь перенаправляется в исходное приложение

Приложение получает доступ к токенам Keycloak

Процесс аутентификации завершен

Важные моменты для понимания:

Keycloak всегда создает локального пользователя (если разрешено) или связывает аккаунты

Приложения взаимодействуют только с Keycloak, а не с внешним провайдером

Токены внешнего провайдера могут быть сохранены и использованы для доступа к его API

**3.5. Роли и маппинг атрибутов при использовании брокеров**

Когда пользователь аутентифицируется через внешнего провайдера, Keycloak получает информацию о нем, включая различные атрибуты и, возможно, роли. Чтобы эта информация была полезной, необходимо настроить маппинг атрибутов и ролей.

Маппинг атрибутов

Маппинг атрибутов позволяет преобразовывать атрибуты, полученные от внешнего провайдера, в атрибуты пользователя Keycloak. Вот как это настроить:

Перейдите в настройки провайдера идентификации

В консоли администратора выберите Identity Providers

Выберите нужного провайдера

Перейдите на вкладку "Mappers"

Создайте новый маппер

Нажмите "Create"

Выберите тип маппера:

Attribute Importer: копирует атрибут от провайдера в Keycloak

Hardcoded Attribute: устанавливает фиксированное значение атрибута

Hardcoded Role: назначает фиксированную роль

Attribute to Role: назначает роль на основе значения атрибута

Username Template Importer: устанавливает имя пользователя по шаблону

Настройте маппер

Для типа Attribute Importer:

Name: имя маппера (например, "Email")

Sync Mode: режим синхронизации (обычно "Inherit" или "Force")

User Attribute Name: имя атрибута в Keycloak (например, "email")

Attribute Name: имя атрибута у провайдера (например, "email")

Attribute Friendly Name: дружественное имя атрибута (для SAML)

Claim (для OIDC/OAuth2): путь к значению в токене (например, "email")

Сохраните маппер

Нажмите "Save"

Примеры полезных маппингов:

Маппинг email

Тип: Attribute Importer

User Attribute Name: email

Claim: email

Маппинг имени и фамилии

Тип: Attribute Importer

User Attribute Name: firstName

Claim: given\_name

Тип: Attribute Importer

User Attribute Name: lastName

Claim: family\_name

Маппинг роли на основе группы

Тип: Attribute to Role

Attribute Name: groups

Attribute Value: admin

Role: admin

Маппинг username

Тип: Username Template Importer

Template: ${CLAIM.preferred\_username}\_{ALIAS}

Маппинг ролей

Если внешний провайдер предоставляет информацию о ролях пользователя, вы можете маппить эти роли на роли в Keycloak:

\*\*

Настройте маппер ролей

В настройках провайдера идентификации перейдите на вкладку "Mappers"

Нажмите "Create"

Выберите тип маппера "Role Importer"

Настройте параметры маппера ролей

Name: имя маппера (например, "Azure AD Roles")

Sync Mode: режим синхронизации (обычно "Inherit" или "Force")

Claim: имя утверждения, содержащего роли (например, "roles", "groups" или "resource\_access.${client\_id}.roles")

Claim Value Type: тип значения (String или JSON)

Role: оставьте пустым для импорта всех ролей или укажите конкретную роль для проверки

Client ID: укажите ID клиента, если роли нужно импортировать как клиентские роли, или оставьте пустым для realm-ролей

Настройте дополнительные параметры

Add to ID token: добавлять ли роли в ID токен

Add to access token: добавлять ли роли в access токен

Add to userinfo: добавлять ли роли в endpoint userinfo

Сохраните маппер

Нажмите "Save"

Примеры маппинга ролей для разных провайдеров:

GitHub

GitHub не предоставляет роли напрямую, но можно использовать организации или команды

Тип: Attribute to Role

Attribute Name: organizations

Attribute Value: название\_организации

Role: соответствующая\_роль

Azure AD

Тип: Role Importer

Claim: roles или wids (для групп безопасности)

Claim Value Type: String

Google

Google не предоставляет роли, но можно использовать дополнительные атрибуты

Тип: Attribute to Role

Attribute Name: hd (домен Google Workspace)

Attribute Value: ваш\_домен

Role: соответствующая\_роль

SAML провайдер

Тип: Role Importer

Attribute Name: http://schemas.microsoft.com/ws/2008/06/identity/claims/role

Attribute Friendly Name: role

Стратегии маппинга ролей:

Прямое маппирование: роли провайдера напрямую соответствуют ролям Keycloak

Пример: роль "admin" у провайдера = роль "admin" в Keycloak

Префиксное маппирование: добавляйте префикс к ролям для разделения источников

Пример: роль "admin" у провайдера = роль "azure\_admin" в Keycloak

Для этого используйте маппер с шаблоном: {ALIAS}\_

{CLAIM.role}

Условное маппирование: назначайте роли на основе атрибутов

Пример: если department="IT", то назначить роль "it\_support"

Используйте несколько мапперов Attribute to Role

Фиксированные роли: всем пользователям от определенного провайдера назначается фиксированный набор ролей

Используйте Hardcoded Role Mapper

**3.6. Проблемы и их решения при настройке брокеров**

При настройке брокеров идентификации могут возникать различные проблемы. Давайте рассмотрим наиболее распространенные из них и способы их решения.

**Проблема 1**: Ошибка при перенаправлении

Симптомы: После аутентификации у внешнего провайдера пользователь видит ошибку "Invalid redirect\_uri" или не возвращается в Keycloak.

Решения:

Проверьте, правильно ли настроен Redirect URI в консоли провайдера. Он должен точно соответствовать формату:

http://<your-keycloak-host>/auth/realms/<your-realm>/broker/<provider-alias>/endpoint

Убедитесь, что имя хоста и протокол (http/https) совпадают

Проверьте настройку Redirect URI в Keycloak, если провайдер требует точного соответствия

**Проблема 2:** Ошибка "User not found" или "Invalid user credentials"

Симптомы: После успешной аутентификации у провайдера Keycloak выдает ошибку о том, что пользователь не найден.

Решения:

Проверьте настройку "First Login Flow" - по умолчанию используется "first broker login", который может требовать дополнительных действий

Убедитесь, что в настройках брокера разрешено автоматическое создание пользователей (не включена опция "Account Linking Only")

Проверьте, правильно ли настроены маппинги для email и username

Проверьте, нет ли конфликта с существующим пользователем (например, с тем же email)

**Проблема 3:** Отсутствие атрибутов пользователя

Симптомы: Пользователь успешно аутентифицируется, но некоторые атрибуты (имя, email и т.д.) не отображаются в Keycloak.

Решения:

Проверьте, запрашиваете ли вы нужные scopes при настройке провайдера (например, "profile email" для OAuth/OIDC)

Проверьте, правильно ли настроены маппинги атрибутов

Включите логирование для отладки и посмотрите, какие данные приходят от провайдера

Для OIDC провайдеров проверьте содержимое ID токена и ответа userinfo endpoint

**Проблема 4:** Отсутствие ролей

Симптомы: Роли пользователя от внешнего провайдера не импортируются в Keycloak.

Решения:

Проверьте, предоставляет ли провайдер информацию о ролях в токене

Убедитесь, что правильно настроен маппинг ролей

Для OIDC/OAuth2 проверьте, что запрошен правильный scope для получения ролей

Для Azure AD может потребоваться дополнительная настройка манифеста приложения

**Проблема 5:** Проблемы с сессиями и токенами

Симптомы: Пользователь входит через брокер, но быстро теряет сессию или не может получить доступ к приложениям.

Решения:

Проверьте настройки сессий в Keycloak (Realm Settings > Sessions)

Убедитесь, что часы на сервере Keycloak и на сервере провайдера синхронизированы

Проверьте настройки SSO (Single Sign-On) в Realm Settings

Для SAML проверьте, что настроены правильные алгоритмы подписи и шифрования

**Проблема 6:** Пользователи видят множество провайдеров на странице входа

Симптомы: На странице входа отображаются все настроенные провайдеры, что может запутать пользователей.

Решения:

Используйте опцию "Hide on Login Page" для провайдеров, которые не должны быть видны всем

Настройте клиент Keycloak с параметром "kc\_idp\_hint", чтобы перенаправлять пользователей сразу к нужному провайдеру

Кастомизируйте тему входа, чтобы выделить предпочтительные методы входа

**Проблема 7:** Пользователи не могут связать существующие аккаунты

Симптомы: Пользователи не могут связать свой существующий аккаунт в Keycloak с аккаунтом внешнего провайдера.

Решения:

Проверьте, настроен ли правильно "First Login Flow"

Убедитесь, что пользователи знают, как управлять связанными аккаунтами через Account Console

Настройте кастомный поток аутентификации, который будет предлагать связывание аккаунтов

Общие рекомендации по отладке:

Включите отладочное логирование

В standalone/configuration/standalone.xml настройте уровень логирования:

xmlCopy<logger category="org.keycloak.broker">

<level name="DEBUG"/>

</logger>

Используйте инструменты разработчика браузера

Отслеживайте HTTP-запросы и ответы

Анализируйте редиректы и параметры URL

Проверяйте токены

Используйте jwt.io для декодирования JWT токенов

Проверяйте содержимое ID токена и access токена

Тестируйте постепенно

Начните с базовой настройки и одного маппинга

Постепенно добавляйте маппинги и проверяйте их работу

Обратитесь к документации провайдера

Каждый провайдер имеет свои особенности и требования к настройке

Проверьте, требуются ли дополнительные разрешения для доступа к определенным атрибутам

***4. Федерация идентификации в Keycloak***

**4.1. Что такое федерация учетных записей?**

Федерация учетных записей (User Federation) в Keycloak — это механизм, который позволяет подключаться к внешним хранилищам пользователей и использовать их для аутентификации и управления учетными записями.

В отличие от брокеров идентификации, где Keycloak делегирует аутентификацию внешнему сервису, при федерации Keycloak напрямую взаимодействует с базой данных пользователей и проверяет их учетные данные.

Простыми словами, федерация учетных записей — это "мост" между Keycloak и существующими системами управления пользователями в вашей организации.

Ключевые особенности федерации учетных записей:

Прямая аутентификация

Пользователи вводят свои учетные данные непосредственно в форму входа Keycloak

Keycloak проверяет эти данные в подключенном хранилище

Синхронизация данных

Keycloak может импортировать пользователей из внешнего хранилища

Данные пользователей могут синхронизироваться периодически или по запросу

Локальное кэширование

Для повышения производительности Keycloak может кэшировать данные пользователей

Можно настроить, как долго данные сохраняются в кэше

Управление паролями

Keycloak может проверять пароли во внешнем хранилище

В некоторых случаях Keycloak может обновлять пароли во внешнем хранилище

Частичная федерация

Можно комбинировать локальных пользователей Keycloak с федерированными пользователями

Можно федерировать только часть атрибутов пользователей

Когда использовать федерацию учетных записей:

Федерация учетных записей особенно полезна в следующих случаях:

У вас есть существующая база пользователей (LDAP, Active Directory)

Вы хотите централизованно управлять учетными записями

Вам нужна синхронизация между несколькими системами

Вы хотите избежать дублирования учетных данных

Вы внедряете SSO, но не хотите мигрировать пользователей

**Разница между федерацией и брокерами:**

Федерация учетных записей

Брокеры идентификации

Пользователи вводят учетные данные в Keycloak

Пользователи вводят учетные данные на странице внешнего провайдера

Keycloak проверяет учетные данные напрямую

Keycloak делегирует проверку учетных данных

Нужен прямой доступ к хранилищу пользователей

Нужен только API-доступ к провайдеру

Хорошо для существующих корпоративных каталогов

Хорошо для социальных и других публичных

IdPФокус на синхронизации данныхФокус на делегировании аутентификации

**4.2. Поддерживаемые механизмы федерации: LDAP, Kerberos и др.**

Keycloak поддерживает несколько механизмов федерации учетных записей. Давайте рассмотрим основные из них.

1. LDAP/Active Directory

LDAP (Lightweight Directory Access Protocol) — это открытый протокол для доступа к службам каталогов, таким как Active Directory от Microsoft. Это наиболее распространенный механизм федерации в Keycloak.

Особенности:

Полная поддержка синхронизации атрибутов пользователей

Маппинг групп LDAP на группы и роли Keycloak

Поддержка различных схем LDAP

Возможность настройки поиска пользователей

Поддержка разных стратегий импорта и синхронизации

Возможность управления паролями (если поддерживается LDAP-сервером)

Поддерживаемые реализации:

Microsoft Active Directory

Red Hat Directory Server

OpenLDAP

ApacheDS

Oracle Directory Server

IBM Tivoli Directory Server

И другие совместимые с LDAP серверы

**2. Kerberos**

Kerberos — это протокол аутентификации, который позволяет узлам доказывать свою подлинность через незащищенную сеть. Часто используется в Windows-средах через Active Directory.

Особенности:

Интеграция с Windows-доменами

Поддержка SSO для настольных приложений

Аутентификация без необходимости вводить пароль (если пользователь уже вошел в домен)

Обычно используется совместно с LDAP-федерацией

Типы интеграции:

SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism) для веб-приложений

GSSAPI (Generic Security Services API) для прямой интеграции

**3. Пользовательские провайдеры федерации**

Keycloak позволяет создавать собственные провайдеры федерации через SPI (Service Provider Interface).

Возможности:

Интеграция с любыми источниками данных

Настройка пользовательской логики аутентификации

Реализация специфичных для организации правил

Доступ к базам данных, API и другим сервисам

Примеры пользовательских провайдеров:

Провайдеры для устаревших баз данных пользователей

Интеграция с пользовательскими REST API

Федерация с несколькими источниками данных одновременно

**4. Хранилища пользователей (User Storage SPI)**

Это более новый и гибкий способ федерации, который заменяет старые провайдеры федерации пользователей. Позволяет создавать адаптеры для любых хранилищ пользователей.

Особенности:

Более гибкий API

Поддержка кэширования

Возможность реализации только части функциональности

Лучшая производительность

Более четкое разделение ответственности

***5. Прочие механизмы федерации***

SSSD (System Security Services Daemon): используется в Linux-системах для интеграции с различными системами идентификации

DB Federation: федерация с внешними базами данных (обычно реализуется через User Storage SPI)

REST Federation: федерация через REST API (обычно реализуется через User Storage SPI)

Выбор механизма федерации

При выборе механизма федерации учитывайте следующие факторы:

Существующая инфраструктура: какие системы управления пользователями уже используются

Требования к производительности: сколько у вас пользователей и как часто они аутентифицируются

Потребности в синхронизации: какие данные и как часто нужно синхронизировать

Требования к безопасности: нужна ли поддержка сложных политик паролей, MFA и т.д.

Техническая экспертиза: насколько сложные механизмы вы готовы поддерживать

В большинстве корпоративных сред наиболее распространен сценарий федерации с LDAP/Active Directory, часто в сочетании с Kerberos для SSO настольных приложений.

4.3. Подключение Keycloak к Active Directory/LDAP

Теперь давайте пошагово разберем, как настроить федерацию пользователей между Keycloak и Active Directory/LDAP.

Шаг 1: Подготовка информации о LDAP/AD

Прежде чем начать настройку, нужно собрать следующую информацию:

Адрес сервера: hostname или IP-адрес LDAP-сервера

Порт: обычно 389 для незащищенного соединения, 636 для LDAPS

Тип сервера: Microsoft Active Directory, Red Hat Directory Server, и т.д.

Учетные данные для подключения: DN и пароль пользователя с правами чтения каталога

Base DN: базовый DN для поиска пользователей (например, dc=example,dc=com)

Фильтры поиска: как искать пользователей и группы

Шаг 2: Настройка федерации в Keycloak

Войдите в административную консоль Keycloak

Откройте браузер и перейдите по адресу: http://<your-keycloak-host>/auth/admin/

Войдите, используя учетные данные администратора

Выберите Realm

В верхнем левом углу находится выпадающий список с доступными Realm'ами

Выберите Realm, для которого настраиваете федерацию

Перейдите в раздел User Federation

В левом меню найдите раздел "User Federation"

Нажмите на него, чтобы открыть список доступных провайдеров

Добавьте новый LDAP-провайдер

Нажмите на выпадающий список "Add provider" и выберите "ldap"

Откроется форма настройки LDAP-провайдера

Настройте базовые параметры подключения

Console Display Name: имя для отображения в консоли (например, "Corporate LDAP")

Priority: приоритет (важно, если у вас несколько провайдеров)

Import Users: включите для импорта пользователей в Keycloak

Edit Mode: режим редактирования (READONLY, WRITABLE, UNSYNCED)

Sync Registrations: включите, если новые пользователи должны создаваться в LDAP

Vendor: выберите тип сервера (Active Directory, Other для Generic LDAP)

Настройте соединение с LDAP

Connection URL: URL подключения к LDAP (например, ldap://ldap.example.com:389)

Enable StartTLS: включите для защиты соединения через StartTLS

Use Truststore SPI: как использовать хранилище доверенных сертификатов

Connection Pooling: включите для повышения производительности

Connection Timeout: таймаут соединения

Read Timeout: таймаут чтения

Настройте аутентификацию LDAP

Bind Type: тип привязки (simple, none, или пользовательский)

Bind DN: DN для привязки (например, cn=ldap-user,dc=example,dc=com)

Bind Credential: пароль для привязки

Настройте параметры поиска пользователей

Users DN: базовый DN для поиска пользователей (например, ou=users,dc=example,dc=com)

Username LDAP attribute: атрибут LDAP для username (обычно sAMAccountName для AD, uid для других)

RDN LDAP attribute: атрибут RDN (обычно cn или uid)

UUID LDAP attribute: атрибут для уникального идентификатора (обычно objectGUID для AD, entryUUID для других)

User Object Classes: классы объектов пользователей (обычно person, organizationalPerson, user для AD)

User LDAP Filter: дополнительный фильтр для поиска пользователей (например, (!(userAccountControl:1.2.840.113556.1.4.803:=2)) для AD, чтобы исключить отключенных пользователей)

Настройте кэширование и синхронизацию

Cache Policy: политика кэширования (NO\_CACHE, DEFAULT, EVICT\_DAILY и т.д.)

Batch Size: размер пакета для синхронизации

Periodic Full Sync: включите для периодической полной синхронизации

Full Sync Period: периодичность полной синхронизации (в секундах)

Periodic Changed Users Sync: включите для синхронизации только изменившихся пользователей

Changed Users Sync Period: периодичность синхронизации изменений (в секундах)

Сохраните настройки

Нажмите "Save" для сохранения конфигурации

Keycloak проверит соединение с LDAP

Шаг 3: Настройка маппингов атрибутов

После создания провайдера LDAP нужно настроить маппинги атрибутов:

Перейдите к настройкам созданного провайдера

В списке провайдеров найдите созданный LDAP-провайдер

Нажмите на его имя, чтобы открыть настройки

Перейдите на вкладку "Mappers"

Нажмите на вкладку "Mappers" вверху

Вы увидите список существующих маппингов

Создайте новый маппер

Нажмите "Create" для добавления нового маппера

Выберите тип маппера:

user-attribute-ldap-mapper: отображает атрибут LDAP на атрибут пользователя

username-ldap-mapper: отображает атрибут LDAP на имя пользователя

role-ldap-mapper: отображает группы LDAP на роли

И другие в зависимости от потребностей

Настройте маппер атрибутов

Для user-attribute-ldap-mapper:

Name: имя маппера (например, "email")

User Model Attribute: имя атрибута в Keycloak (например, "email")

LDAP Attribute: имя атрибута в LDAP (например, "mail")

Read Only: только чтение или возможность записи

Always Read Value From LDAP: всегда ли читать значение из LDAP

Is Mandatory In LDAP: обязательный ли атрибут в LDAP

Создайте маппер для основных атрибутов

Обычно нужны маппинги для:

Имя пользователя (username)

Email

Имя и фамилия (firstName, lastName)

Другие необходимые атрибуты (например, телефон, отдел и т.д.)

Настройте маппинг групп (опционально)

Создайте маппер типа "group-ldap-mapper"

Настройте соответствие между группами LDAP и группами Keycloak

Укажите базовый DN для групп и другие параметры

Шаг 4: Запуск синхронизации

После настройки всех маппингов запустите синхронизацию:

Вернитесь к настройкам LDAP-провайдера

В левом меню выберите "User Federation"

Нажмите на имя настроенного LDAP-провайдера

Запустите синхронизацию

Найдите кнопки "Sync all users" и "Sync changed users" внизу страницы

Нажмите "Sync all users" для первой полной синхронизации

Дождитесь завершения процесса

Проверьте результаты

Перейдите в раздел "Users" в левом меню

Убедитесь, что пользователи из LDAP были импортированы

Проверьте, что атрибуты отображаются правильно

Шаг 5: Проверка работы

Проверьте, что федерация работает корректно:

Войдите как пользователь из LDAP

Выйдите из консоли администратора

Перейдите на страницу входа

Введите учетные данные пользователя из LDAP

Проверьте атрибуты и группы

После входа перейдите в Account Console

Убедитесь, что атрибуты пользователя отображаются правильно

Проверьте, что группы и роли назначены корректно

Типичные проблемы и их решения

Не удается подключиться к LDAP

Проверьте адрес сервера и порт

Убедитесь, что учетные данные для привязки корректны

Проверьте, что сервер LDAP доступен с сервера Keycloak

Пользователи не импортируются

Проверьте базовый DN для поиска пользователей

Убедитесь, что LDAP-фильтр не исключает пользователей

Проверьте, правильно ли указаны классы объектов пользователей

Атрибуты не отображаются

Проверьте настройки маппингов атрибутов

Убедитесь, что атрибуты существуют в LDAP

Проверьте права доступа к атрибутам

Пользователи не могут войти

Проверьте, правильно ли настроен атрибут username

Убедитесь, что пароли хранятся в поддерживаемом формате

Проверьте, что пользователи не заблокированы в LDAP

Проблемы с производительностью

Настройте кэширование

Увеличьте значение Batch Size для больших каталогов

Оптимизируйте LDAP-фильтры

**4.4. Сценарий использования Kerberos с Keycloak**

Kerberos — это протокол сетевой аутентификации, который позволяет узлам безопасно доказывать свою подлинность через незащищенную сеть. Он особенно популярен в Windows-средах через Active Directory. Давайте рассмотрим, как настроить интеграцию Kerberos с Keycloak для реализации Single Sign-On.

Почему стоит использовать Kerberos с Keycloak?

Бесшовный SSO: пользователи, вошедшие в домен Windows, могут автоматически аутентифицироваться в веб-приложениях, защищенных Keycloak, без повторного ввода учетных данных.

Централизованное управление: администраторы могут управлять учетными записями пользователей в одном месте (Active Directory), а Keycloak будет использовать эти учетные данные.

Повышенная безопасность: Kerberos предлагает высокий уровень безопасности с шифрованием и защитой от атак повторного воспроизведения.

Корпоративная совместимость: идеально подходит для организаций, уже использующих Active Directory.

Как работает интеграция Kerberos с Keycloak?

Настройка сервер-принципала (Service Principal):

На стороне Active Directory необходимо создать сервисный принципал для Keycloak.

Генерация keytab-файла, содержащего криптографические ключи для Keycloak.

Настройка Keycloak:

Включение Kerberos-аутентификации в Keycloak.

Указание пути к keytab-файлу и настройка параметров Kerberos.

Настройка LDAP/AD федерации пользователей.

Настройка браузеров:

Для работы Kerberos требуется настройка браузеров пользователей для поддержки интегрированной аутентификации Windows (SPNEGO).

Процесс аутентификации:

Пользователь входит в домен Windows.

При обращении к защищенному Keycloak ресурсу, Keycloak запрашивает Kerberos-аутентификацию.

Браузер автоматически отправляет Kerberos-билет.

Keycloak проверяет билет и создает сессию без запроса пароля.

Пример потока аутентификации с Kerberos

Пользователь авторизуется в Windows-домене (получает TGT-билет от KDC).

Пользователь открывает веб-приложение, защищенное Keycloak.

Keycloak отправляет HTTP-заголовок с запросом SPNEGO-аутентификации.

Браузер запрашивает сервисный билет для службы Keycloak от контроллера домена.

Браузер отправляет полученный билет в Keycloak.

Keycloak проверяет билет с помощью своего keytab-файла.

При успешной проверке Keycloak ищет пользователя в федерации LDAP/AD.

Пользователь получает доступ к приложению без ввода пароля.

Типичные сложности и их решения

Проблемы с DNS: Kerberos чувствителен к правильному разрешению имен. Убедитесь, что DNS настроен корректно.

Настройка SPN: Часто возникают проблемы с неправильно настроенным Service Principal Name. Проверьте командой setspn -L username.

Настройка браузеров: В Firefox и Chrome нужно вручную добавить сайты в список доверенных для Kerberos-аутентификации.

Разные домены: Если Keycloak и Active Directory находятся в разных доменах, требуется настройка доверительных отношений.

Пример команд для настройки

bashCopy# Создание принципала для службы Keycloak в Active Directory

ktpass -princ HTTP/keycloak.example.com@EXAMPLE.COM -mapuser keycloak@EXAMPLE.COM -pass Password123 -crypto AES256-SHA1 -ptype KRB5\_NT\_PRINCIPAL -out keycloak.keytab

# Проверка keytab-файла

klist -k keycloak.keytab

# Настройка параметров Kerberos в Keycloak (примерный конфиг в standalone.xml)

<spi name="userFederationProvider">

<provider name="kerberos" enabled="true">

<properties>

<property name="kerberosRealm" value="EXAMPLE.COM"/>

<property name="serverPrincipal" value="HTTP/keycloak.example.com@EXAMPLE.COM"/>

<property name="keyTab" value="/path/to/keycloak.keytab"/>

<property name="debug" value="true"/>

</properties>

</provider>

</spi>

Рекомендации по внедрению

Начните с тестовой среды: Настройте интеграцию в отдельной тестовой среде перед применением в производстве.

Ведите подробный журнал: Включите детальное логирование для отладки проблем с Kerberos.

Учитывайте ограничения: Kerberos-аутентификация работает не во всех браузерах и требует настройки со стороны клиента.

Запланируйте альтернативный метод: Всегда имейте резервный метод аутентификации на случай проблем с Kerberos.