Инструкция по установке и настройке решения

2023

Оглавление

Список таблиц

Список иллюстраций

1 Аннотация

Настоящий документ описывает процедуру установки и настройки интранет-решения Intry.

2 Введение

Данная инструкция предназначена для развёртывания решения Intry с использованием Docker (https://www.docker.com)

3 Компоненты

Таблица 1 Основные компоненты Intry

№	Название	Программное обеспечение	Версия	Источник
1.	Шлюз	nginx	1.23	https://nginx.org/
2.	Intry UI	Angular	11	https://angular.io/cli
3.	Intry API	.NET	6	https://github.com/dotnet/aspnetcore#readme
4.	Intry SSO	Keycloak	21.1.1	https://www.keycloak.org/
5.	Intry Jobs	.NET	6	https://github.com/dotnet/aspnetcore#readme
6.	Intry Files	MinIO	RELEASE.2023-04-07T05-28-58Z	https://min.io/
7.	СУБД	PostgreSQL	15.2	https://www.postgresql.org/

4 Интегрируемые системы

Таблица 2 Интегрируемые системы

Название	Назначение
SMTP-сервер	Используется для отправки email-писем.
Active Directory	Используется как один из провайдеров для аутентификации. Опционально.

5 Технические требования

Требования к аппаратным и программным характеристикам определяются исходя из количества одновременно действующих пользователей, их сценариям взаимодействия с решением, а также требований к отказоустойчивости.

Ниже приведены минимальные технические требования для работы продукта Intry в Docker, число пользователей системы составляет не более 100 человек, без учета отказоустойчивости.

5.1 Требования к оборудованию

Таблица 3 Требования к оборудованию

Компонент	ЦПУ, ядер	ОЗУ, МБ	Диск, ГБ
Шлюз	0.1	512	20
Intry UI	0.1	32	20
Intry API	0.5	1024	20
Intry SSO	1	1024	20
Intry Jobs	1	2048	20
Intry Files	2*	2048*	100*
СУБД	2	2048	50

* - Зависит от количества файлов загружаемых в систему

5.2 Требования к программному обеспечению

В настоящей инструкции операционной системой хост-сервера является Ubuntu 22.04.1 LTS https://releases.ubuntu.com/jammy/. При необходимости можно использовать любую операционную систему, поддерживающую инфраструктуру Docker https://docs.docker.com/engine/install/.

Таблица 4 Требования к программному обеспечению

Компонент	Образы Docker
Шлюз	Nginx 1.23
Intry UI	Angular 11, intry-ui v-2648
Intry API	.NET 6, intry-api v-2649
Intry SSO	Keycloak 21.1.1
Intry Jobs	.NET 6, intry-jobs v-2649
Intry Files	MinIO RELEASE.2023-04-07T05-28-58Z
СУБД	PostgreSQL 15.2

6 Установка компонентов

6.1 СУБД

Необходим для хранения реляционных данных приложения.

Переменные окружения:

Название	Значение	Описание
POSTGRESS_PASSWORD	<пароль>	Пароль системного пользователя приложения, под которым приложение подключается к БД

Последовательность действий:

- Установить докер контейнер из образа https://hub.docker.com/_/postgres
- Включить контейнер с образом в общую сеть (**docker network connect intry postgres**)

Пример:

```
docker run -d -p 5432:5432 --name postgres \
    -e POSTGRES_PASSWORD=P@ssw0rd \
    postgres:15.2-alpine
```

6.2 Intry Files

Необходим для хранения бинарных данных (файлы, видео, изображения). Используется последняя версия.

Размер диска и RAM сильно зависит от планируемого общего количества файлов и их размера.

Переменные окружения:

Название	Значение	Описание
MINIO_ROOT_USER	<Сгенерировать>	Имя системного пользователя для доступа к интерфейсу приложения
MINIO_ROOT_PASSWORD	<Сгенерировать>	Пароль системного пользователя
MINIO_REGION	us-east-1	Необходим для нормального функционирования библиотеки AWS по работе с S3-совместимыми хранилищами.

Последовательность действий:

- установить докер контейнер из образа quay.io/minio/minio, инструкция https://min.io/docs/minio/container/index.html.

- Включить контейнер в общую сеть (**docker network connect intry minio**)

Пример:

```
docker run -d -p 9000:9000 -p 9090:9090 --user $(id -u):$(id -g) --name minio \
      -e "MINIO_ROOT_USER=admin" \
      -e "MINIO_ROOT_PASSWORD=P@ssw0rd" \
      -e "MINIO_REGION=us-east-1" \
      -v ${HOME}/minio/data:/data quay.io/minio/minio server /data --console-address ":9090"
```

6.3 INTRY SSO

Необходимо для обеспечения аутентификации пользователей.

Переменные окружения:

Название	Значение	Описание
KEYCLOAK_ADMIN	admin	Встроенная учётная запись администратора SSO
KEYCLOAK_ADMIN_PASSWORD	<пароль>	Пароль встроенной учётной записи администратора
KC_DB	postgres	Провайдер хранилища для SSO, используется PostgreSQL
KC_DB_URL_HOST	postgres	Хост
KC_DB_URL_PORT	5432	Порт, стандартный
KC_DB_URL_DATABASE	keycloak	Название БД SSO
KC_DB_USERNAME	keycloak_writer	Учётная запись БД SSO
KC_DB_PASSWORD	<пароль>	Пароль для подключения к БД SSO

Последовательность действий:

- Установить докер контейнер из образа **quay.io/keycloak/keycloak** согласно инструкции https://www.keycloak.org/getting-started/getting-started-docker
- Включить контейнер в общую сеть (**docker network connect intry keycloak**)

Пример:

```
docker run -p 30003:8080 --name keycloak \
      -e KEYCLOAK_ADMIN=admin \
      -e KEYCLOAK_ADMIN_PASSWORD="eI7b67t3HOc1DPqA4ZYp" \
      -e PROXY_ADDRESS_FORWARDING=true \
```

```
-e KC_PROXY=edge \
-e KC_DB=postgres \
-e KC_DB_URL_HOST=postgres \
-e KC_DB_URL_PORT=5432 \
-e KC_DB_URL_DATABASE=keycloak \
-e KC_DB_USERNAME=postgres \
-e KC_DB_PASSWORD="T9swqGps5fzpG0Wp5zey" \
-d quay.io/keycloak/keycloak:21.1.1 start
```

6.4 Intry API

Переменные окружения:

Название	Значение	Описание
AuthConfiguration__Authority	<keycloak_url>/realms/intry	Абсолютная ссылка к сервису аутентификации.
SwaggerConfiguration__Authority	<keycloak_url>/realms/intry	Абсолютная ссылка к сервису аутентификации.
AWS__BucketName	intry	Название бакета. Можно изменять.
AWS__ServiceURL	<minio_url>:9000	Абсолютная ссылка к сервису s3-совместимого хранилища Minio.
AWS__AccessKey	<ключ>	Ключ учётной записи в Minio для внутренних коммуникаций между API и Minio
AWS__SecretKey	<секрет>	Секрет учётной записи в Minio для внутренних коммуникаций между API и Minio
ASPNETCORE_ENVIRONMENT	Production	Переменная, определяющая тип окружения (продуктивное, тестовое)
ConnectionStrings__Intry	Host=postgres;Database=intry;Username=postgres;Password=<Password>;Port=5432;	Строка подключения к БД

Последовательность действий:

- Установить докер контейнер из образа **registry.intry.net:5000/intry-api:v-2648**
- Включить контейнер в общую сеть (**docker network connect intry intry-api**)

Пример:

```
docker run -d -p 30001:80 --restart unless-stopped --name intry-api \
```

-e "ASPNETCORE_ENVIRONMENT= Production" \

-e "ConnectionStrings__Intry"="Host=postgres;Database=intry;Username=postgres;Password=P@ssw0rd;Port=5432;" registry.intry.net:5000/intry-api:v-2646

6.5 Intry UI

Переменные окружения:

Название	Значение	Описание
API_URL	<ссылка к API>	Абсолютная ссылка к сервису API.
AUTHORITY_URL	<keycloak_url>/realms/intry	Абсолютная ссылка к сервису аутентификации.

Последовательность действий:

- Установить докер контейнер из образа **registry.intry.net:5000/intry-ui:v-2647**
- Включить контейнер в общую сеть (**docker network connect intry intry-app**)

Пример:

**docker run -d -p 30000:80 --restart unless-stopped --name intry-ui **

-e "environment=pg" registry.intry.net:5000/intry-ui:v-2648

6.6 Шлюз

Последовательность действий:

- Установить докер контейнер из образа https://hub.docker.com/_/nginx
- Включить контейнер в общую сеть (**docker network connect intry nginx**)

Пример:

**docker run -d -p 80:80 -p 443:443 --privileged --restart unless-stopped --name nginx **

**-v ./nginx/conf/:/etc/nginx/conf.d/:ro **

**-v ./certbot/www:/var/www/certbot/:ro **

nginx

Примечания:

- *В рамках развёртывания предполагается использование **certbot** для получения сертификатов для доступа по **https.***
- *Публикация сервисов по небезопасному протоколу **http** не рассматривается в данной инструкции.*
- *Если имеются уже выпущенные собственные сертификаты, то можно использовать их и пропустить шаги для выпуска сертификатов letsencrypt через certbot.*

6.7 Intry Jobs

Переменные окружения:

Название	Значение	Описание
ConnectionStrings__Intry	Host=postgres;Database=intry;Username=postgres;Password=<Password>;Port=5432;	Строка подключения к БД
ASPNETCORE_ENVIRONMENT	Production	Переменная, определяющая тип окружения (продуктивное, тестовое)

Последовательность действий:

- Установить докер контейнер из образа **registry.intry.net:5000/intry-jobs:v-2649**
- Включить контейнер в общую сеть (**docker network connect intry intry-jobs**)

Пример:

```
docker run -d -p 30004:80 --restart unless-stopped --name intry-jobs \
        -e "ASPNETCORE_ENVIRONMENT= Production" \
        -e
"ConnectionStrings__Intry"="Host=postgres;Database=intry;Username=postgres;
Password=P@ssw0rd;Port=5432;" registry.intry.net:5000/intry-jobs:v-2649
```

7 Настройка параметров

7.1 Настройка Intry SSO (Keycloak)

Для сервиса единого входа (Single Sign-On) необходимо выполнить первоначальные настройки.

1. Создать Realm

В выпадающем меню нажать кнопку «Create realm».

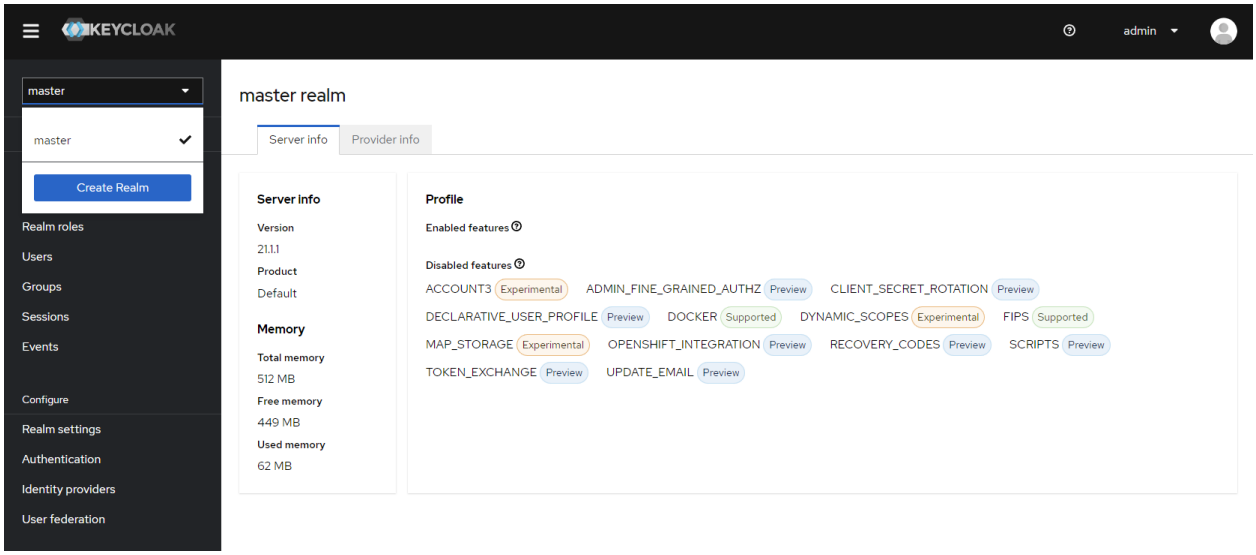

Рисунок 1 Создание Realm (окружения)

На странице создания realm указать название (**intry**) и передвинуть переключатель в **Enabled**.

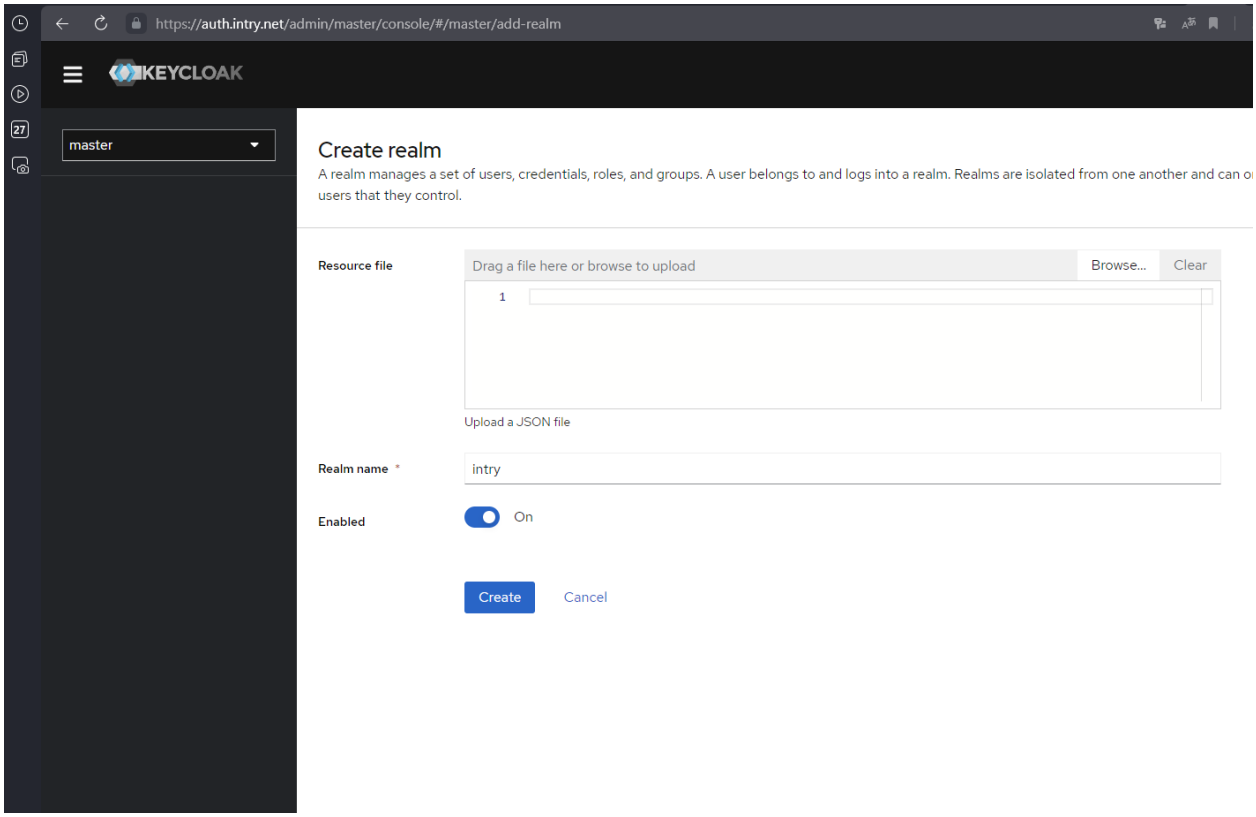

Рисунок 2 Страница создания Realm (окружения)

Созданный **Realm** будет отображаться в выпадающем меню.

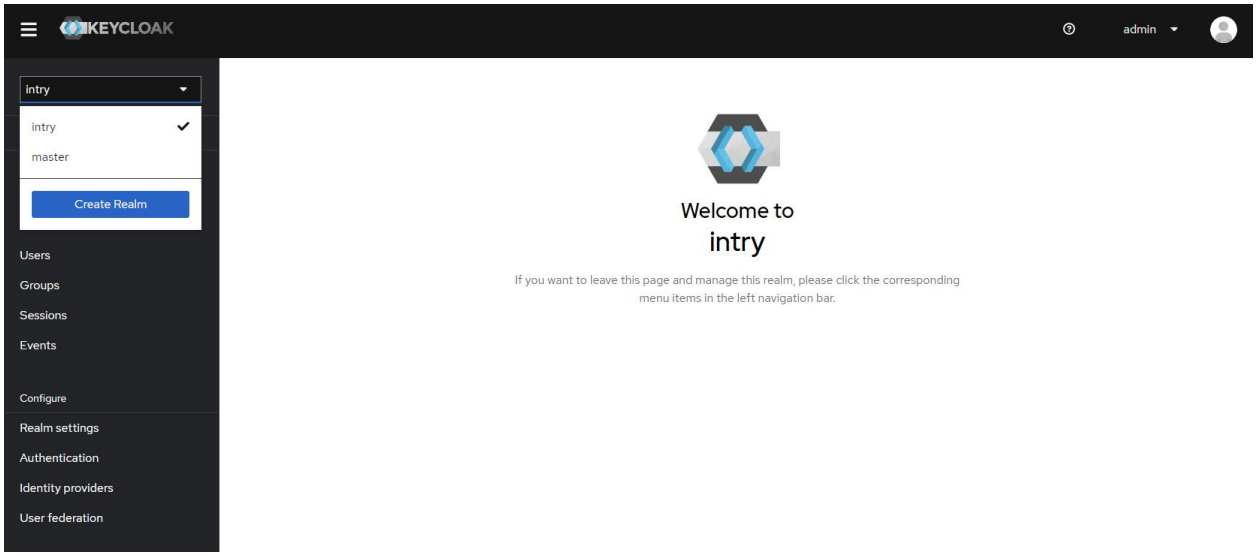

Рисунок 3 Страница созданного Realm

2. Создать и настроить клиент

Для этого необходимо перейти на страницу **Clients** и там нажать на кнопку **Create client.**

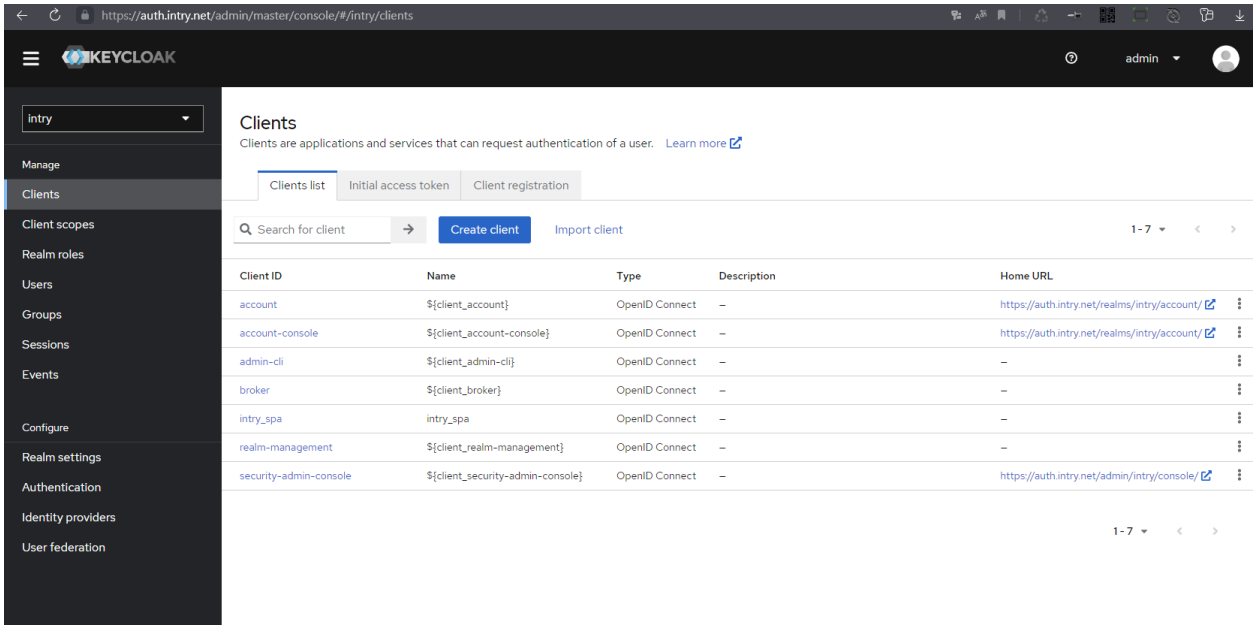

Рисунок 4 Список клиентов для realm Intry

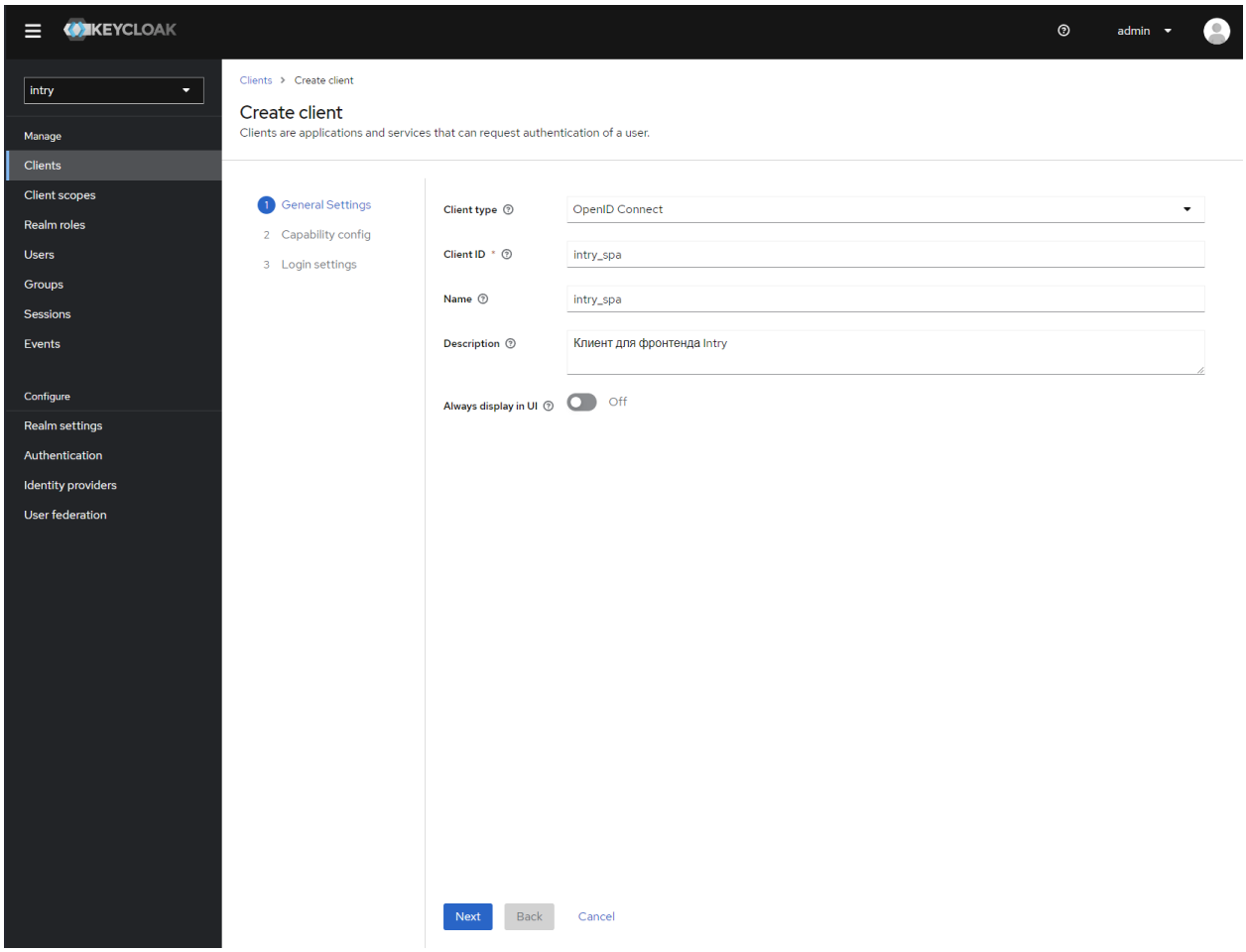

Рисунок 5 Создание клиента intry_spa в realm Intry (стр 1)

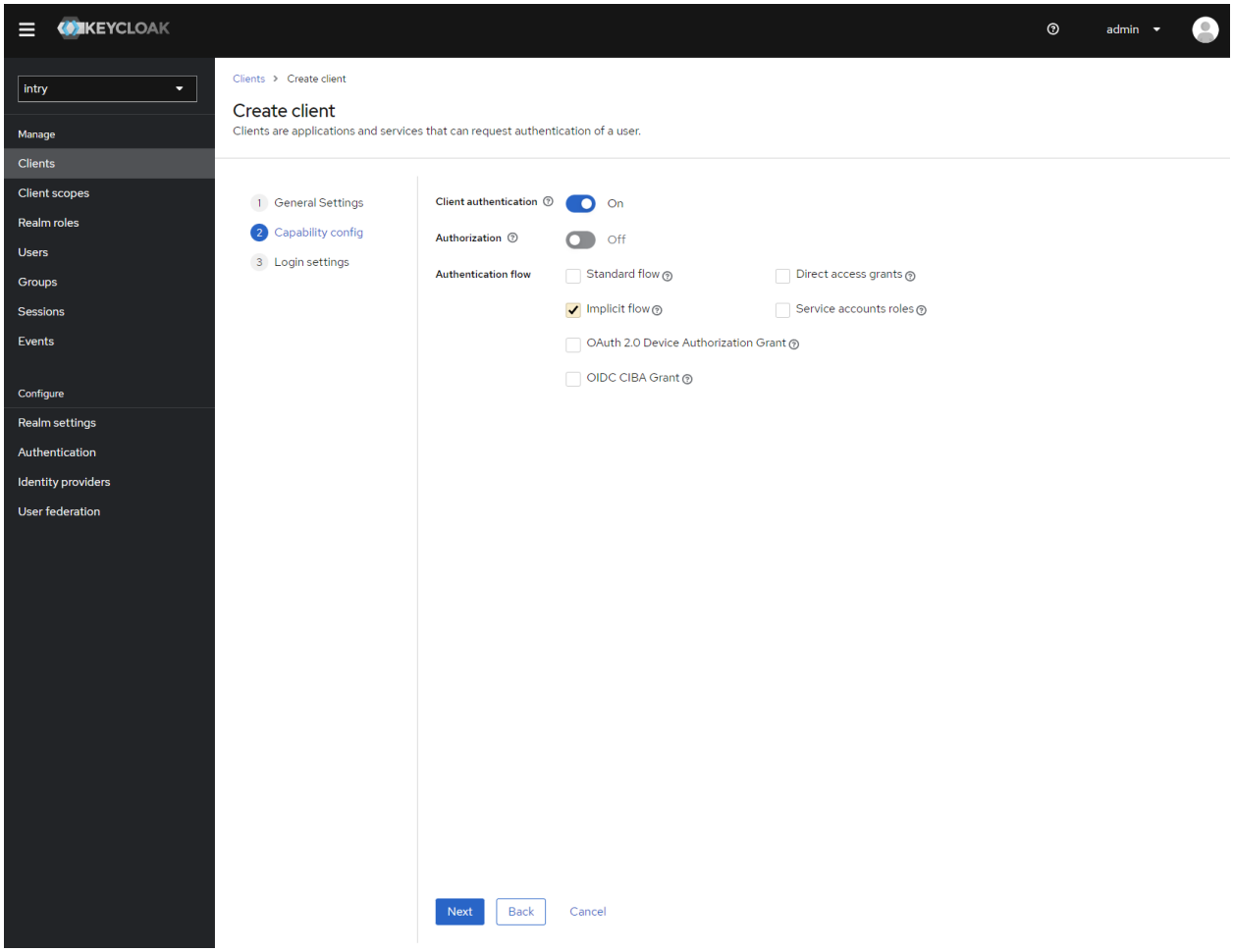

Рисунок 6 Создание клиента intry_spa в realm Intry (стр 2)

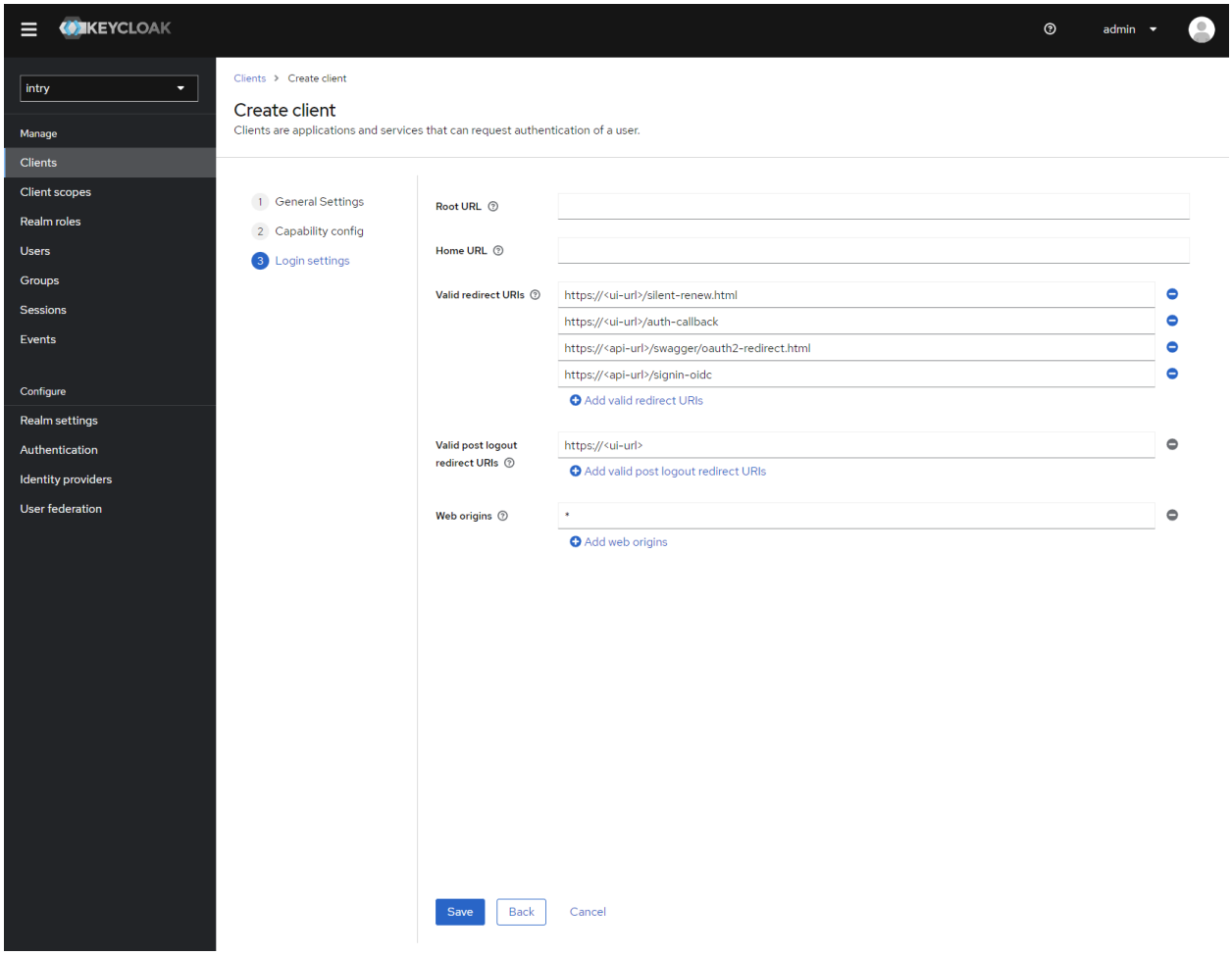

Рисунок 7 Создание клиента intry_spa в realm Intry (стр 3)

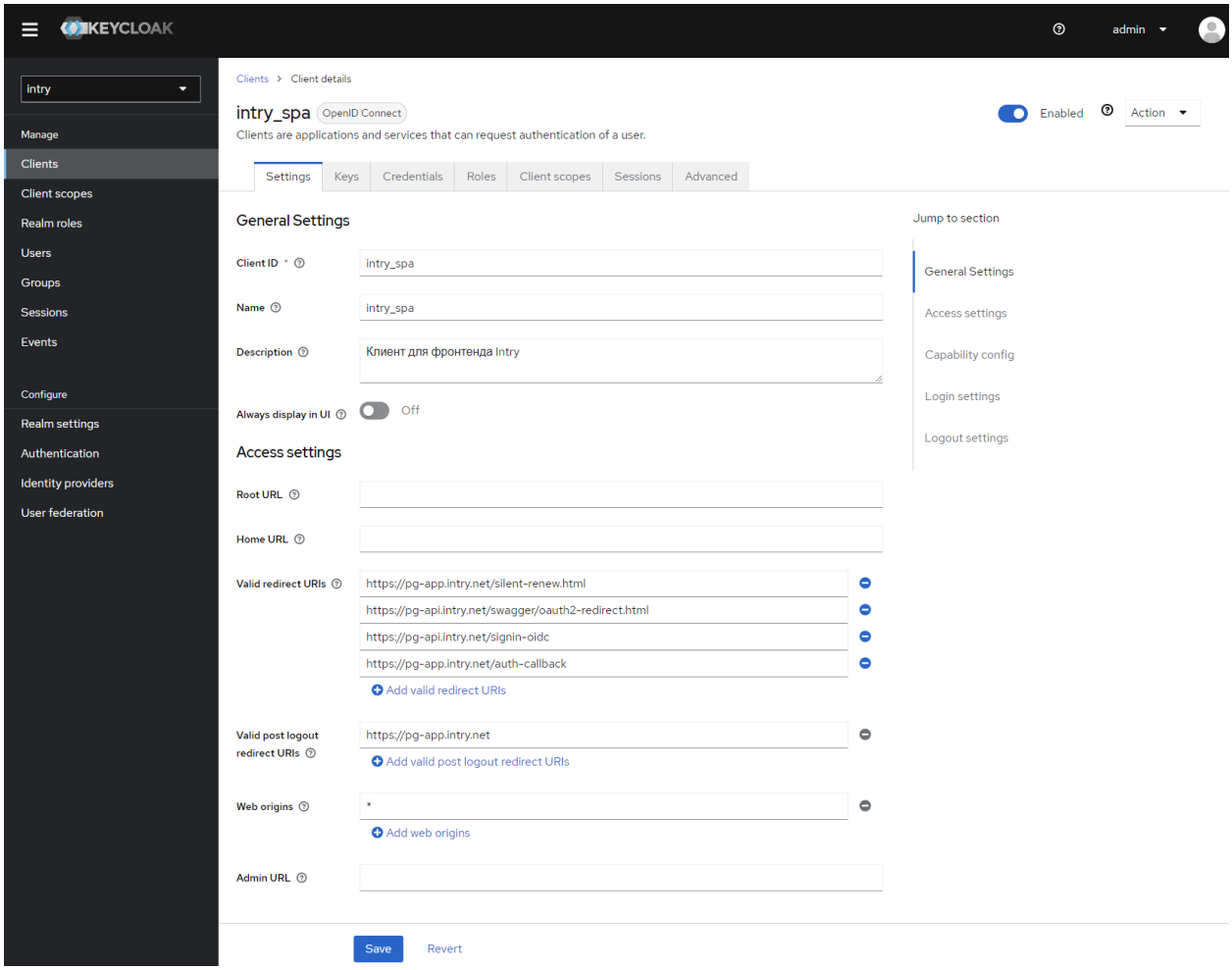

Рисунок 8 Страница редактирования созданного клиента intry_spa

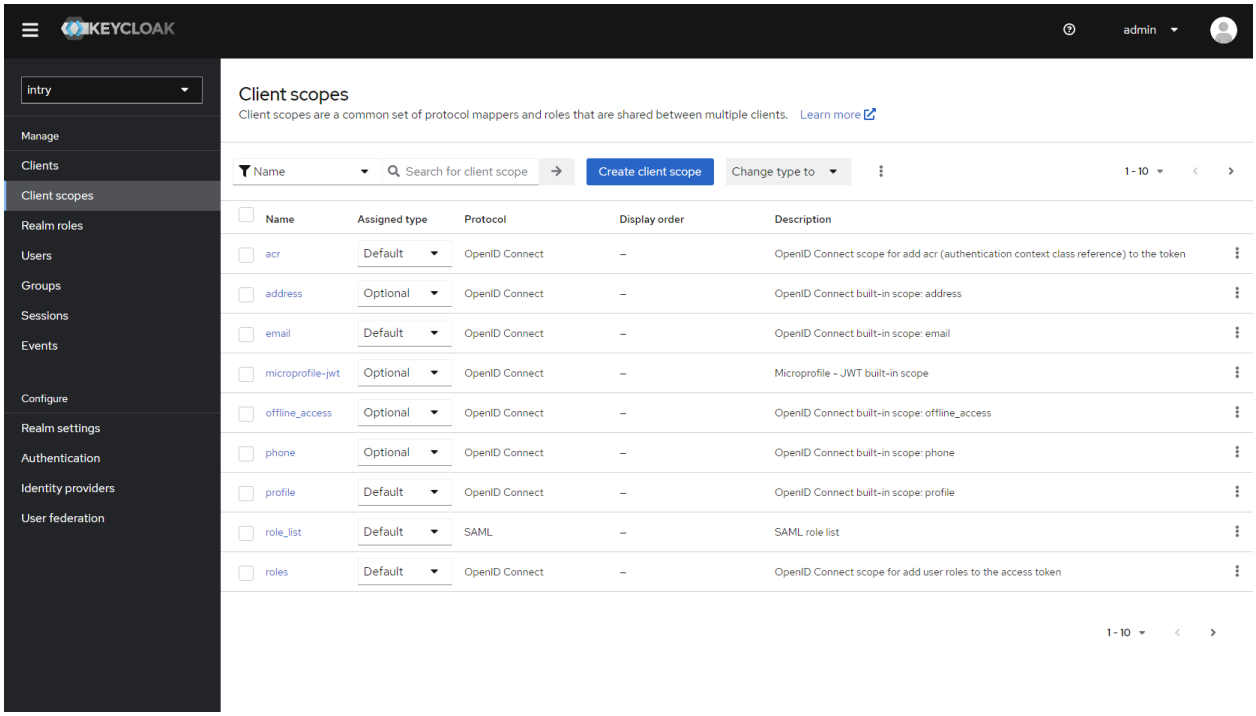

Рисунок 9 Страница client scopes

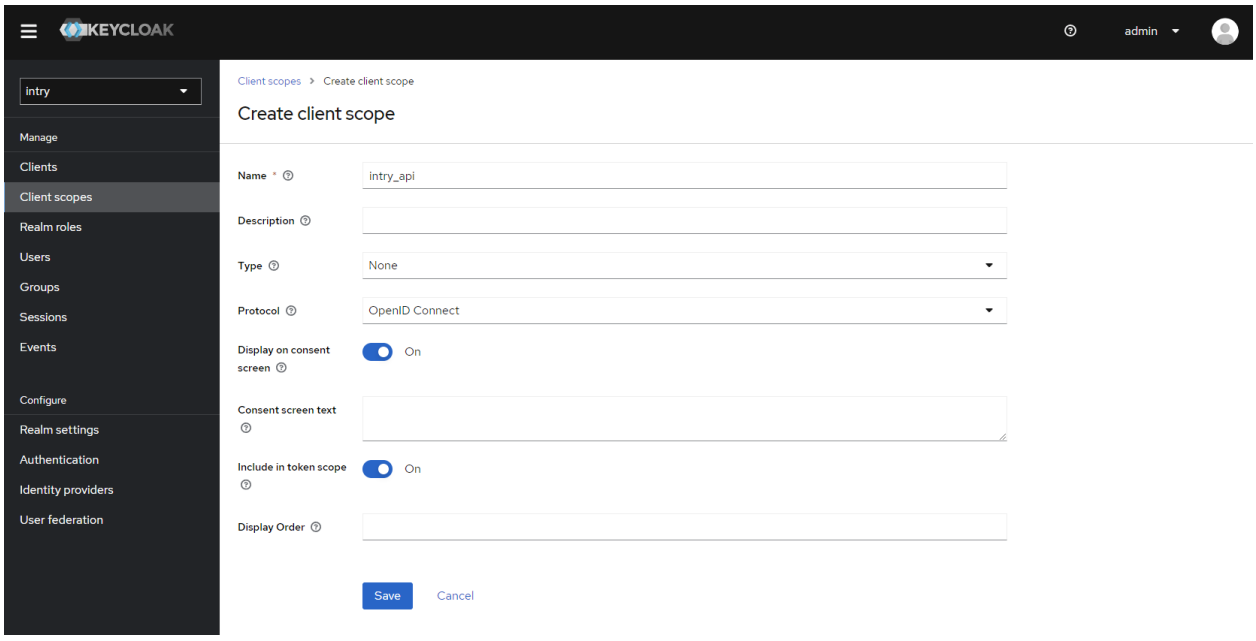

Рисунок 10 Страница создания client scope

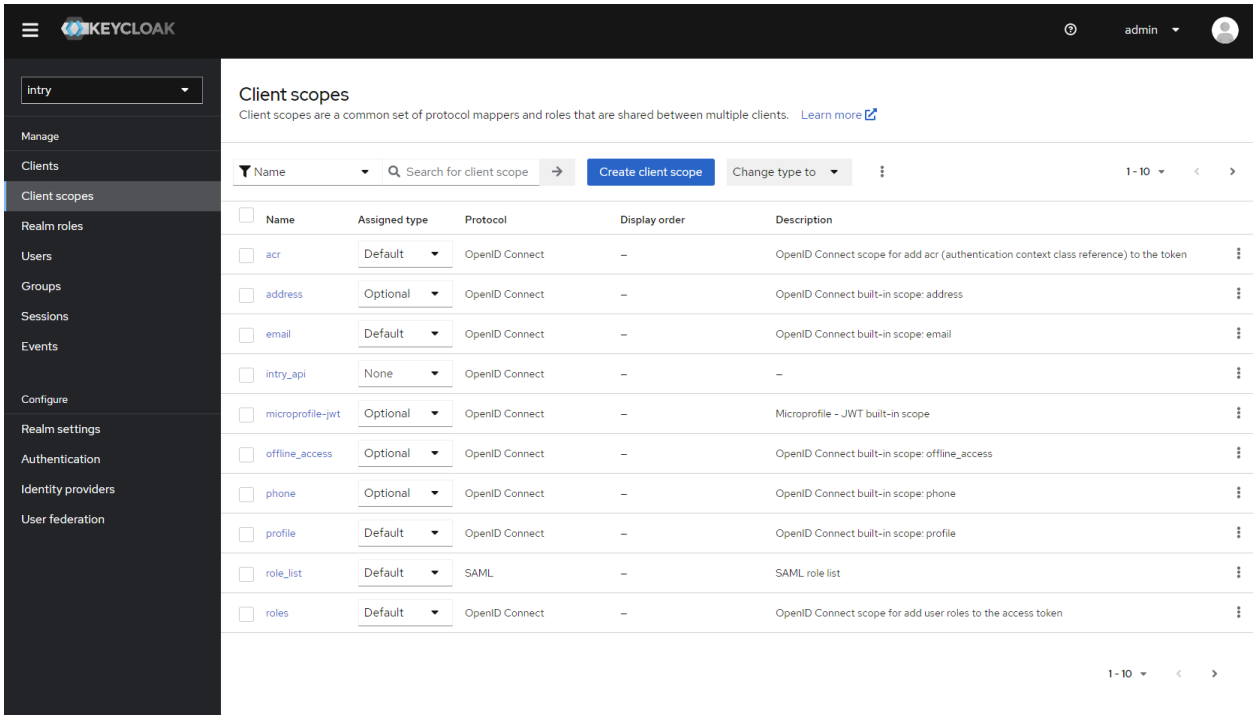

Рисунок 11 Страница списка client scopes после создания intry_api

Для созданного **Client Scope** под названием **intry_api** необходимо создать привязку к audience. Это является необходимой защитой клиента и **API** – токен выданный сервисом будет валиден только для **API**.

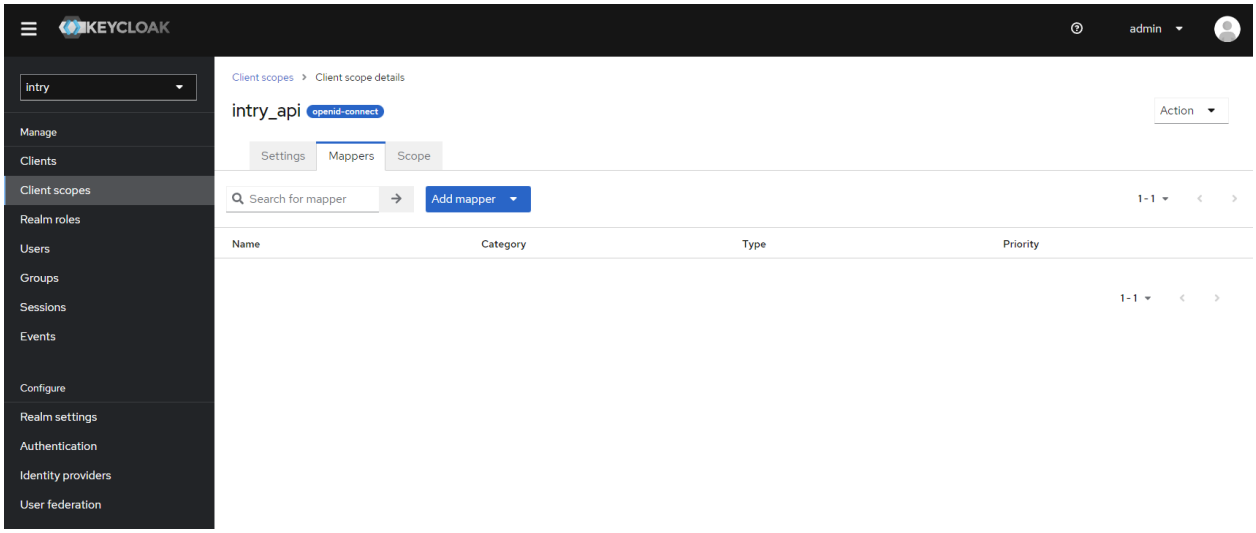

Рисунок 12 Страница client scope intry_api

При нажатии «**Add mapper**» будет открыто всплывающее окно создания конфигурации привязки для созданного client scope.

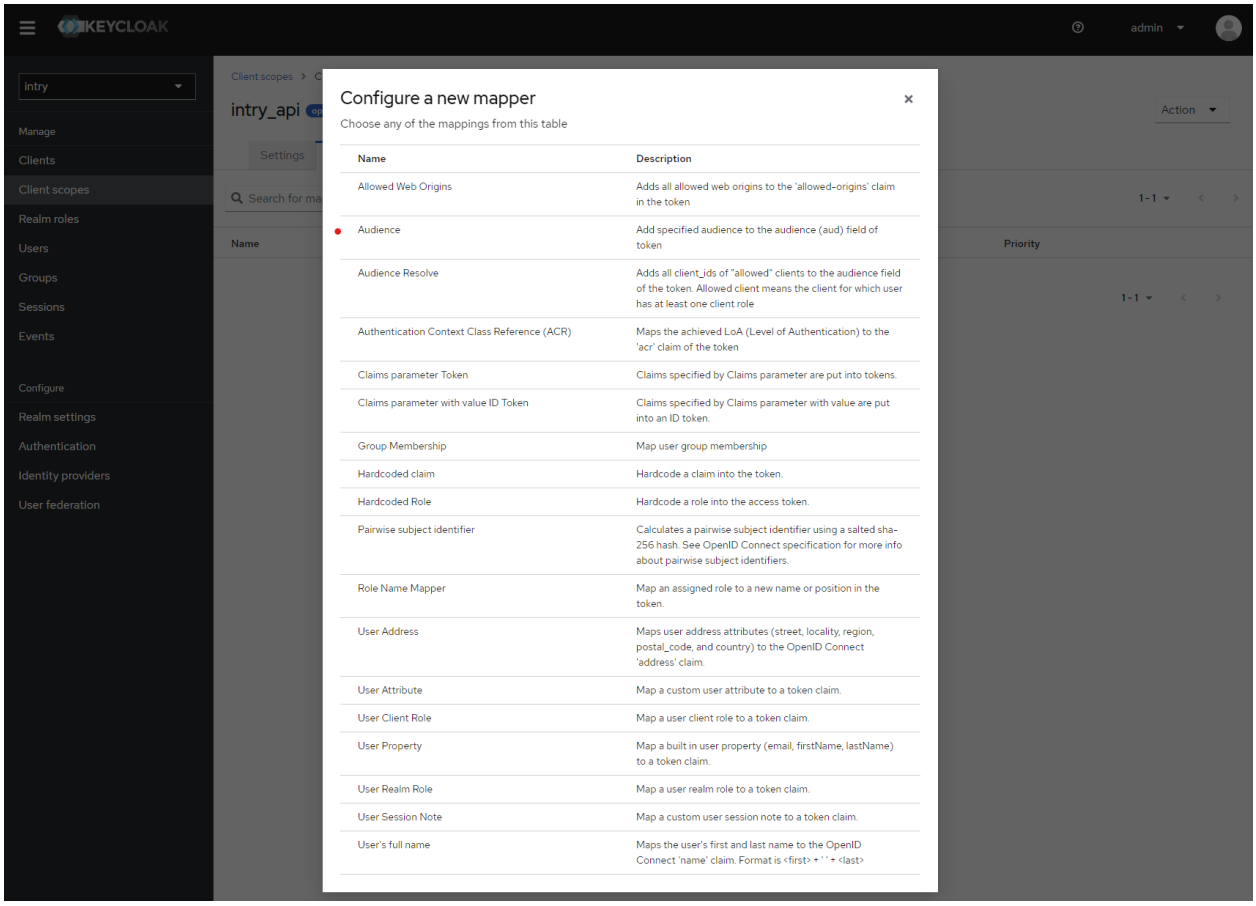

Рисунок 13 Страница выбора типа конфигурированиия для client scope intry_api

Необходимо заполнить как на картинке ниже. Важный момент – это поле Included Custom Audience, в него необходимо внести значение **intry_api**.

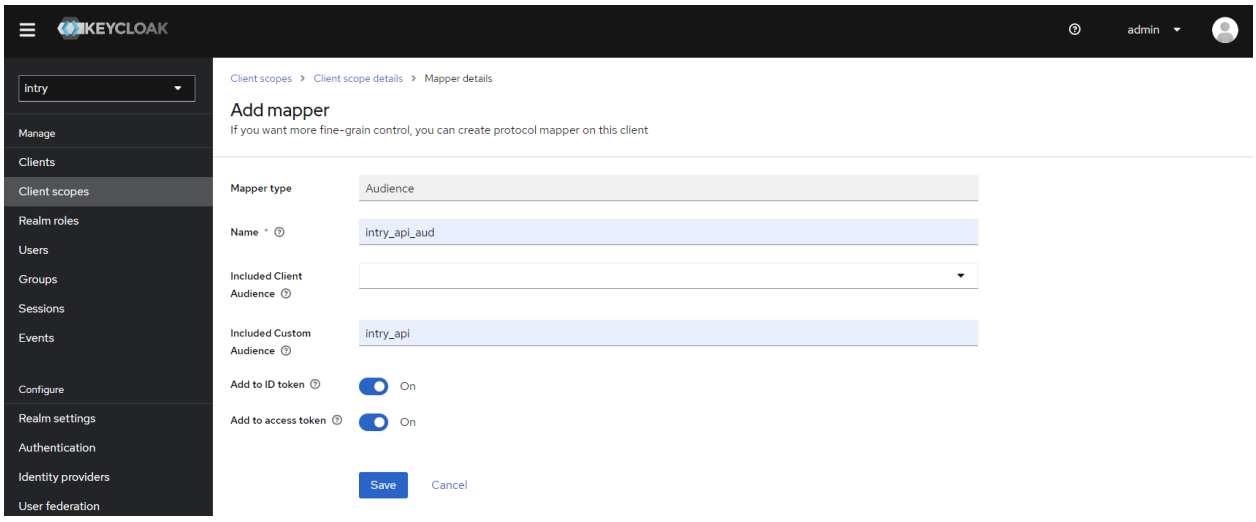

Рисунок 14 Страница конфигурированиия привязки Audience для client scope intry_api

Затем необходимо добавить созданный и настроенный client scope **intry_api** в клиент **intry_spa**. Это выполняется на вкладке **Client scopes** на странице отображения клиента **intry_spa**.

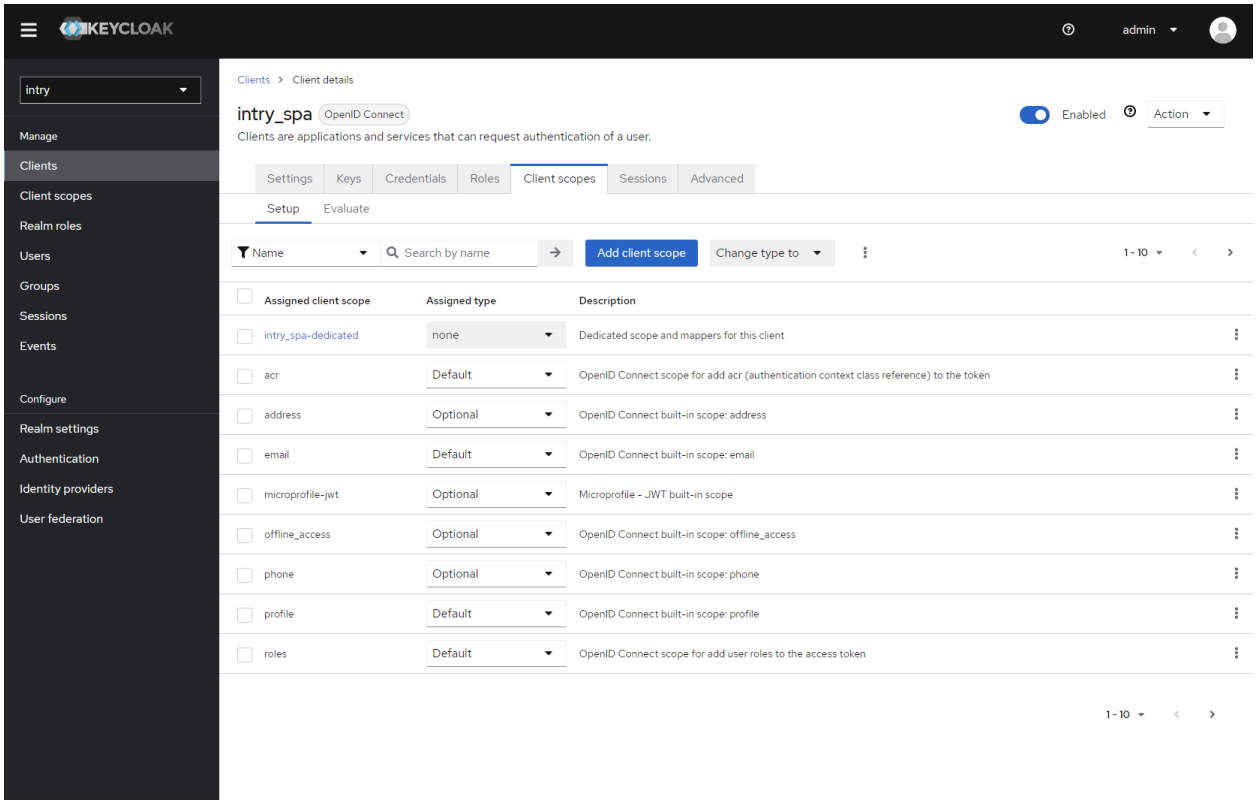

Рисунок 15 Добавление scope intry_api в клиент intry_spa (рис 1)

После нажатия кнопки «**Add client scope**», во всплывающем окне необходимо добавить требуемый scope – **intry_api**.

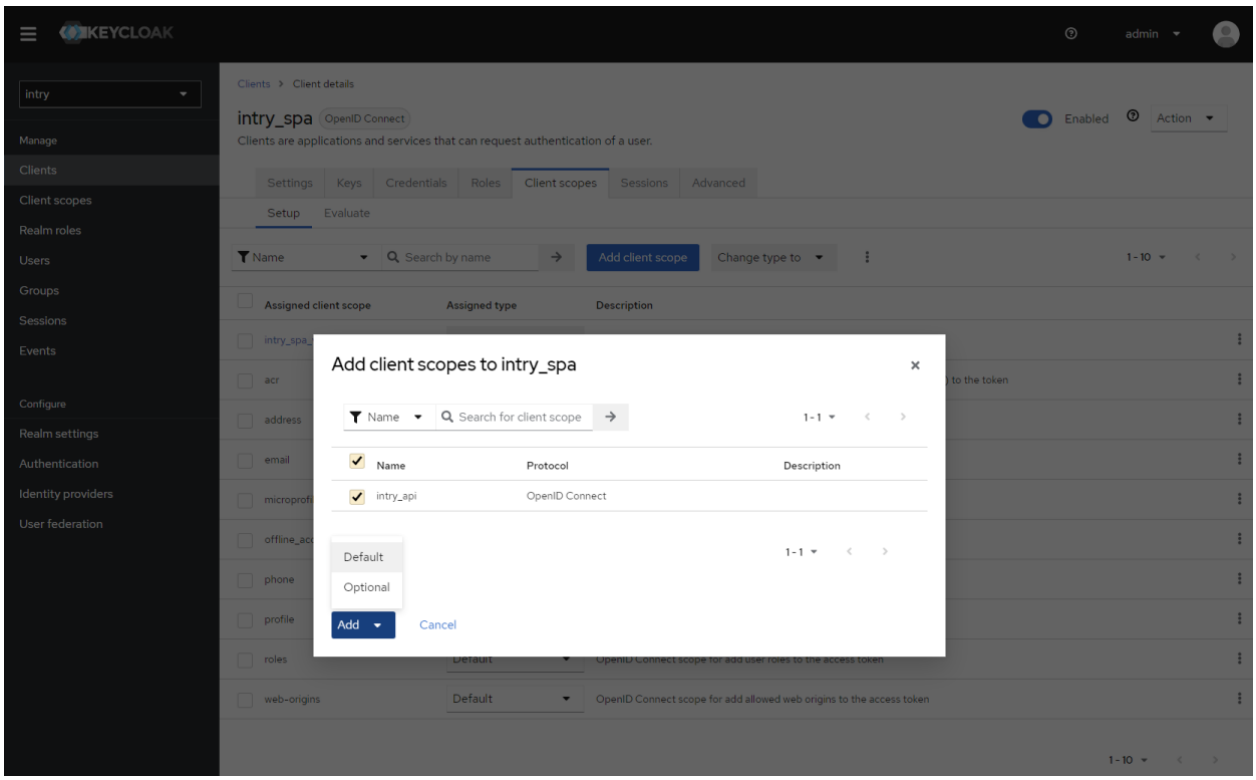

Рисунок 16 Добавление scope intry_api в клиент intry_spa (рис 2)

Добавленный scope **intry_api** обязательно будет отображаться на странице списка scope для клиента **intry_spa**.

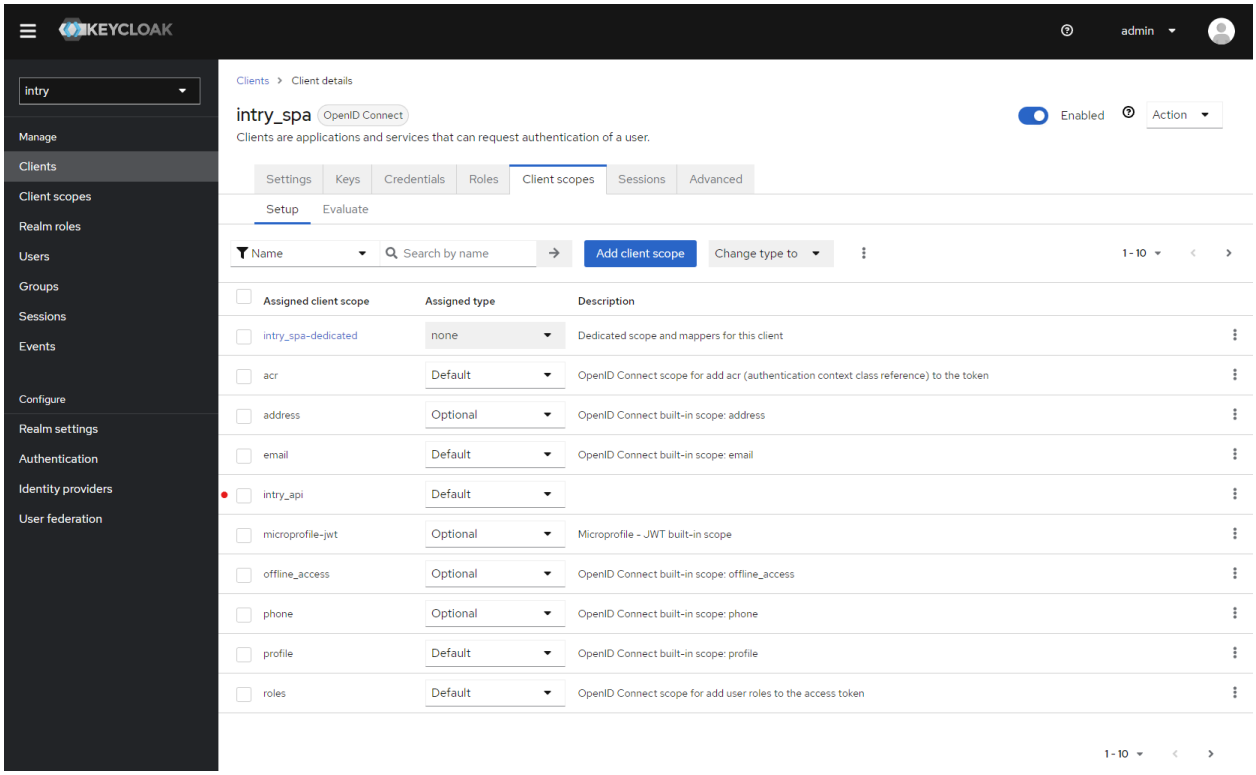

Рисунок 17 Добавленный scope в списке

3. Добавление подключения к Active Directory (опционально)

При наличии служб Active Directory в компании, либо любой иной службы директорий, поддерживающих LDAP протокол, возможно подключить федерацию для синхронизации данных по пользователям.

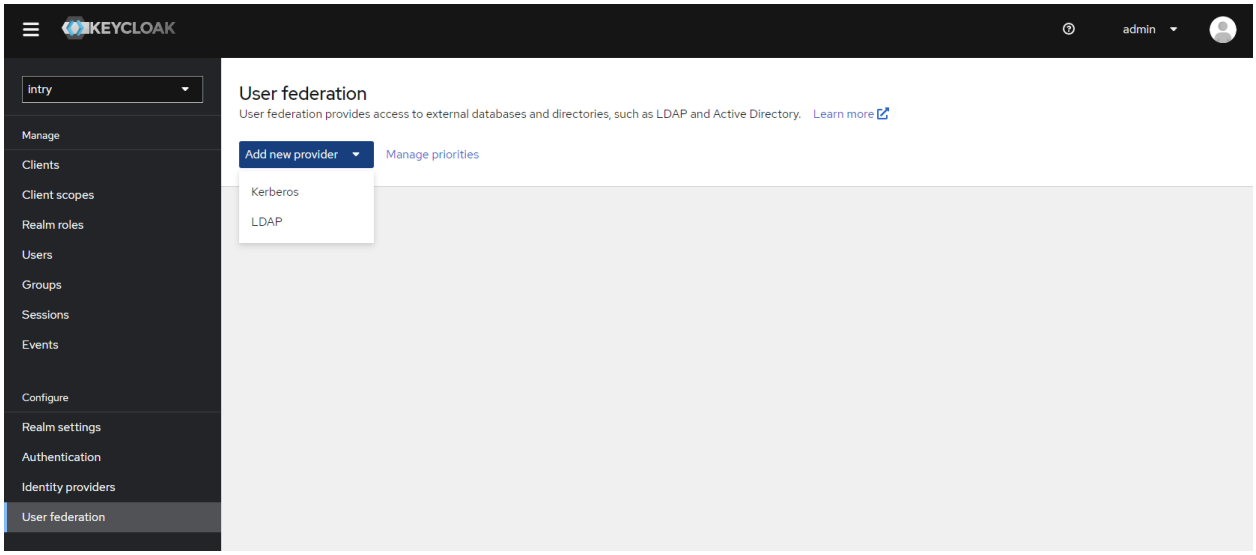

Рисунок 18 Создание подключения к Active Directory

Пользователи, синхронизированные из LDAP, смогут входить на портал без дополнительных настроек и создания их вручную.

На странице создания LDAP подключения заполняются необходимые параметры. Протоколы используются стандартные, поэтому примеры настройки можно найти в общем доступе, в том числе и на сайте open source поставщика (https://www.keycloak.org/documentation.html).

Рабочий пример, также, указан ниже на скриншотах.

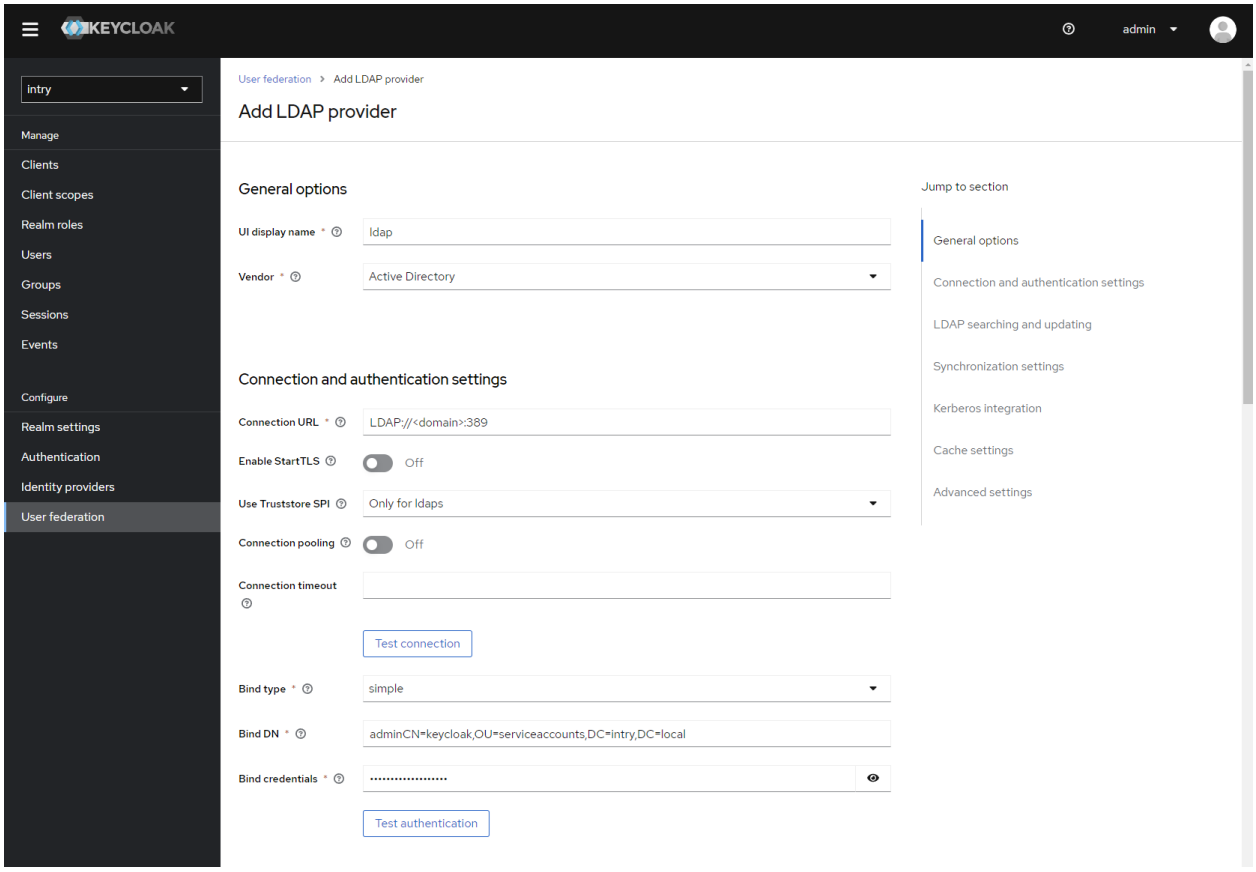

Рисунок 19 Страница настройки LDAP подключения (часть1)

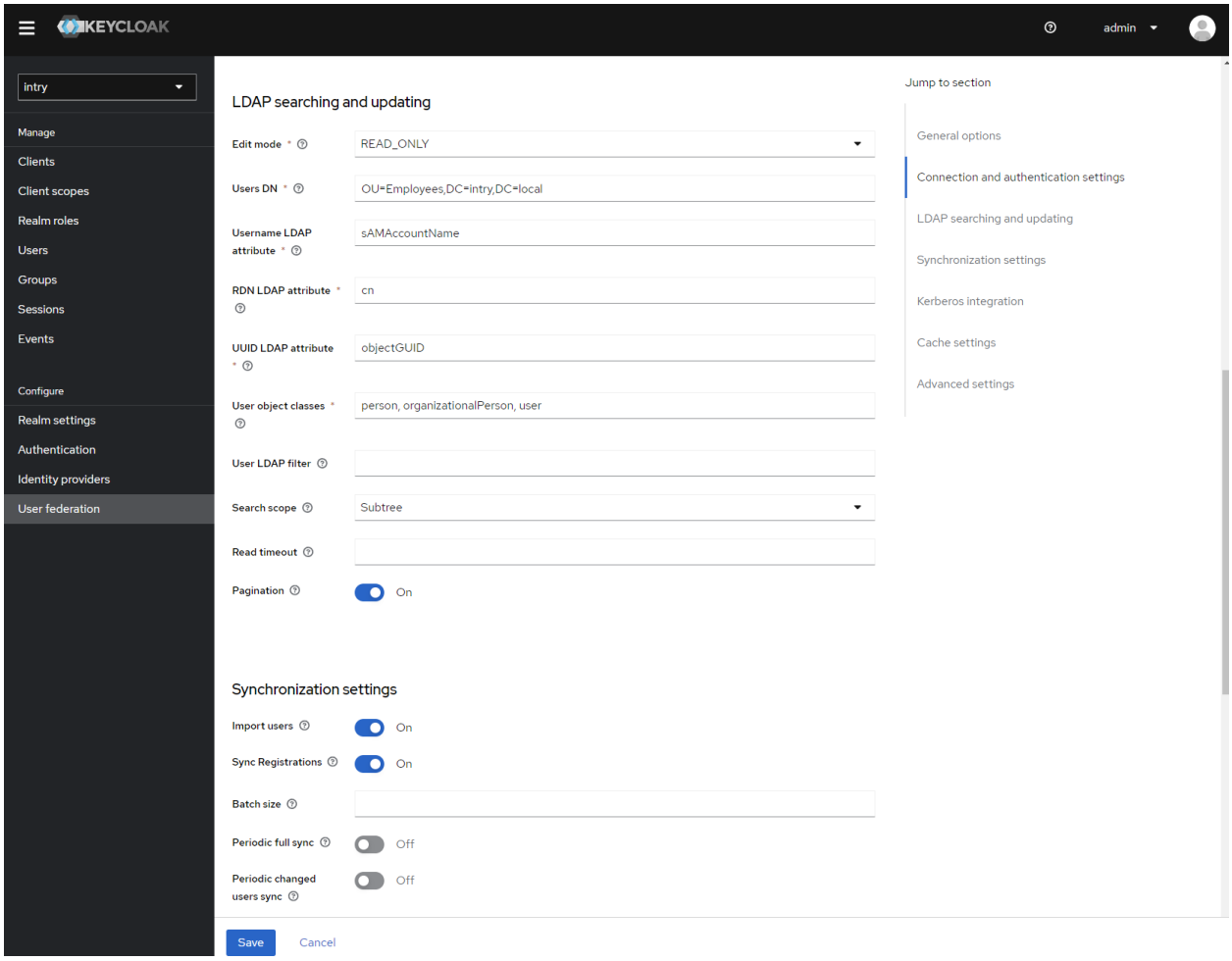

Рисунок 20 Страница настройки LDAP подключения (часть2)

После добавления LDAP провайдера – он будет отображаться на странице **User federation**.

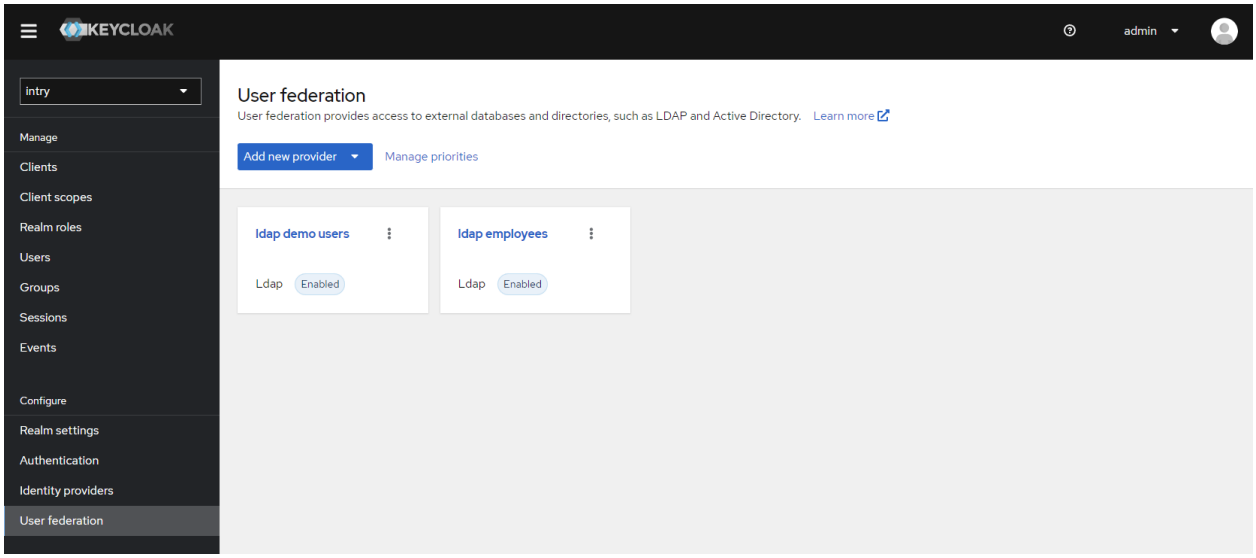

Рисунок 21 Добавленые LDAP подключения в федерации пользователей

7.2 Настройка шлюза (nginx)

Для успешного маршрутизация запросов необходимо выполнить настройки для службы nginx.

1. Необходимо скопировать настройки для сайтов в папку /nginx/conf
2. Выпустить ssl-сертификат или использовать имеющийся

7.3 Настройка Intry Files (MinIO)

Для выполнения настроек необходимо перейти на сайт MinIO. По умолчанию это <minio_url>:9090

7.3.1 Создание бакета

Далее следует создать бакет и назвать его **intry.**

Для этого необходимо нажать «**Create bucket**»

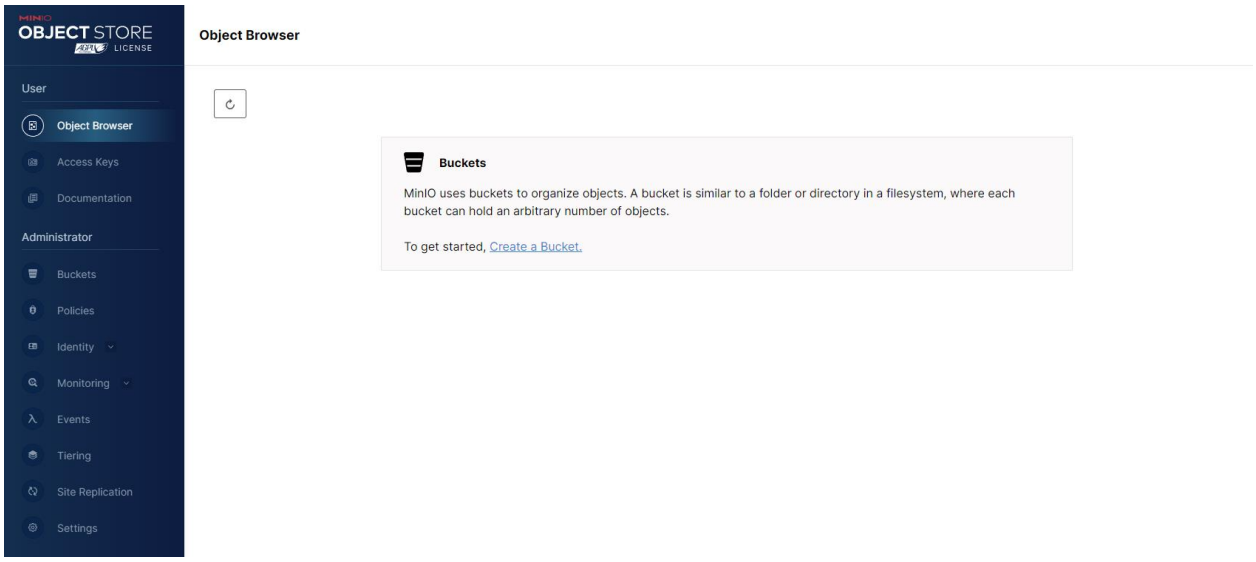

Рисунок 22 Страница создания бакета

На следующей странице ввести в **BucketName** значение «**intry**», включить версионность и нажать «**Create bucket**».

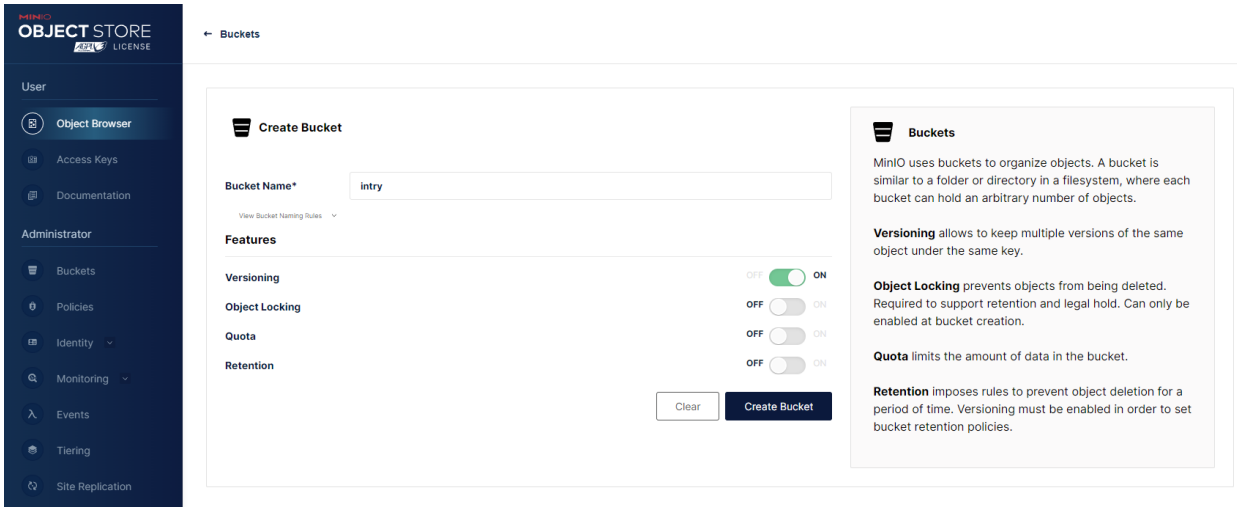

Рисунок 23 Страница созданного бакета

7.3.2 Создание ключей доступа

Приложение API подключается к MinIO по 9000 порту с использованием access_key_id и secret_access_key, что является типичными подключением к s3-совместимогу хранилищу.

Ключ доступа можно создать глобально либо привязанными к конкретному пользователю. Мы рекомендуем создать «системного пользователя», выдать ему права и затем сгенерировать ключ.

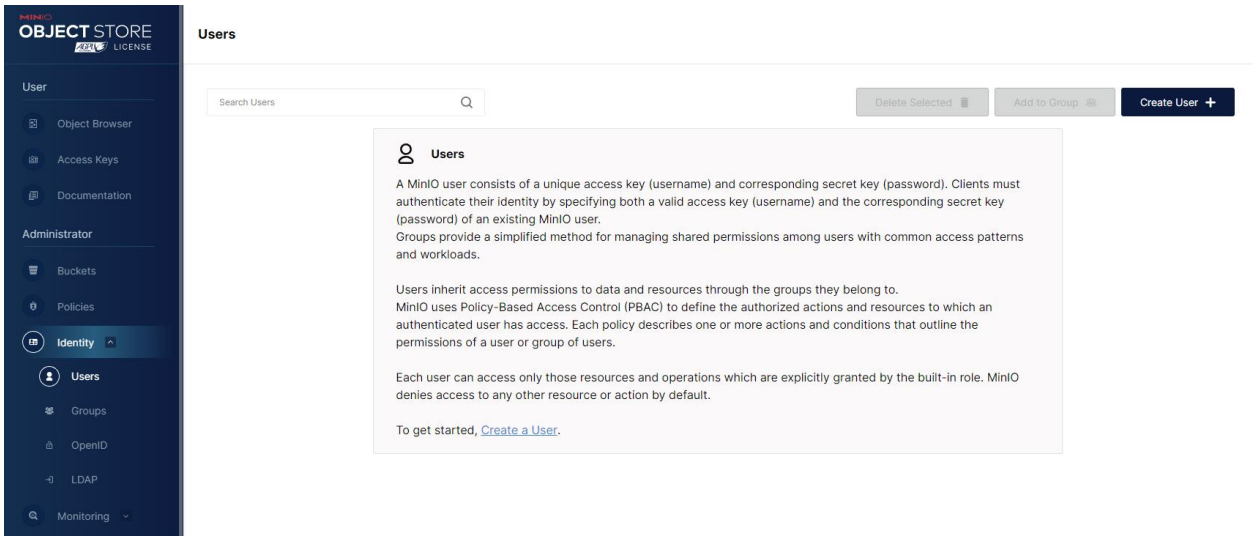

Рисунок 24 Страница списка пользователей

На странице создания пользователя указать **User Name** (имя), **Password** (пароль) и отметить права **readwrite**.

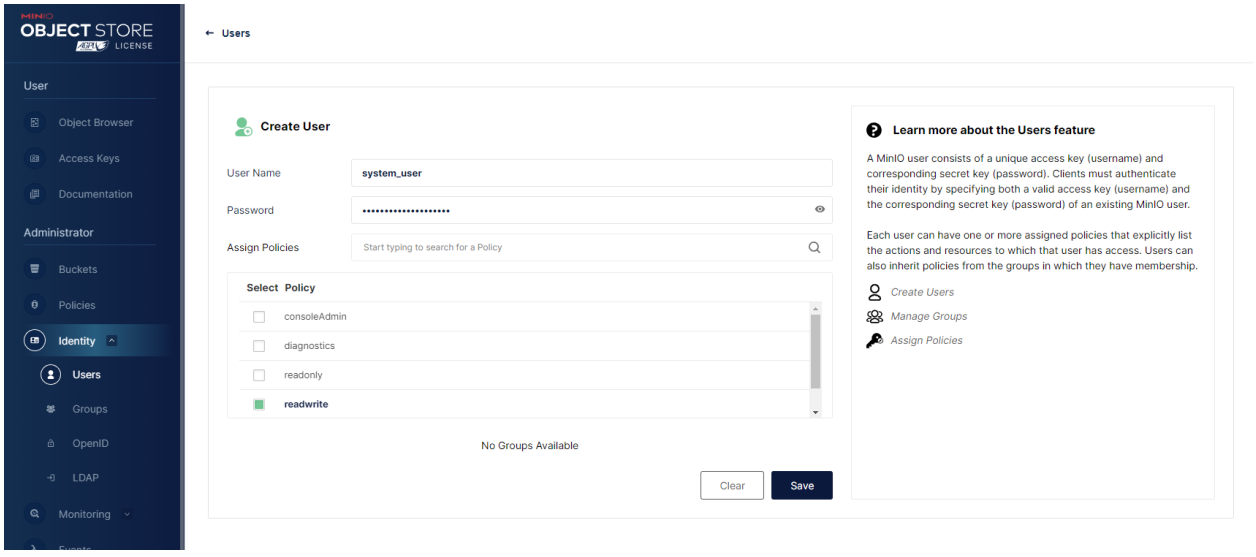

Рисунок 25 Страница создания пользователя

После создания пользователя необходимо в него перейти и выбрать **Service Accounts**. На данной странице нажать **Create Access Key**.

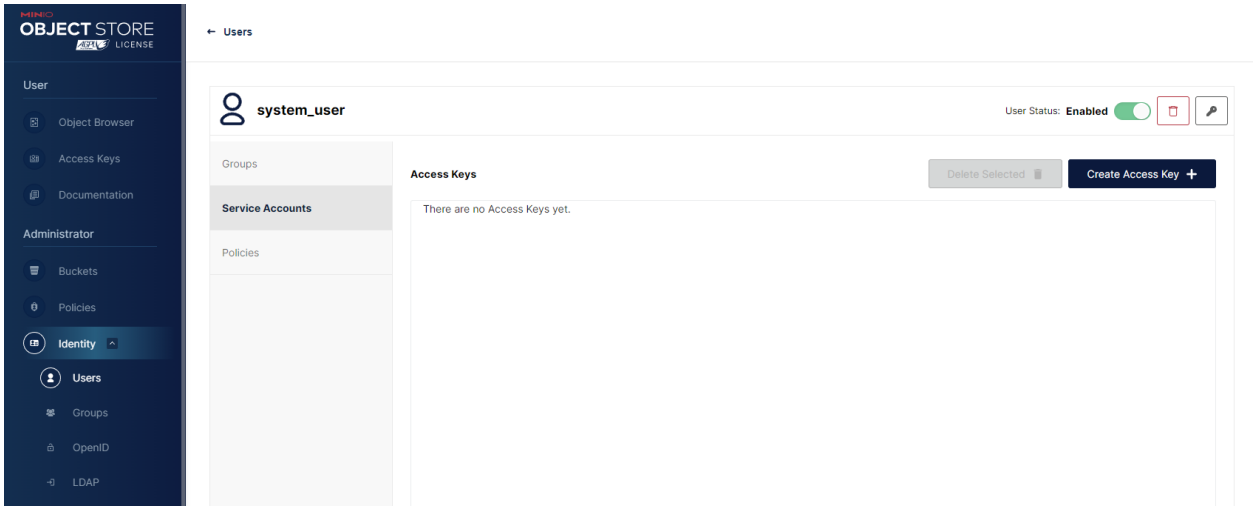

Рисунок 26 Страница списка ключей доступа для пользователя

Ключ и секрет сгенерируются автоматически. Их необходимо будет записать, т. к. секрет после создания посмотреть невозможно.

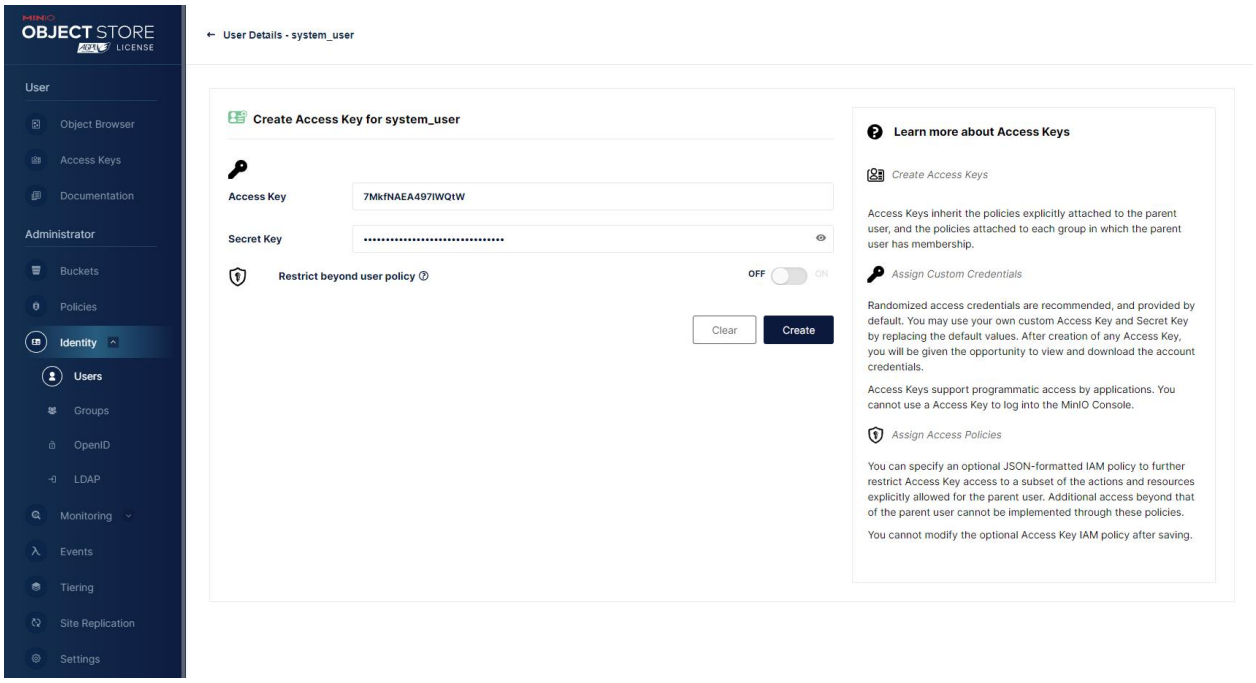

Рисунок 27 Страница создания ключа доступа

В дальнейшем можно просматривать и администрировать файлы через данный интерфейс.

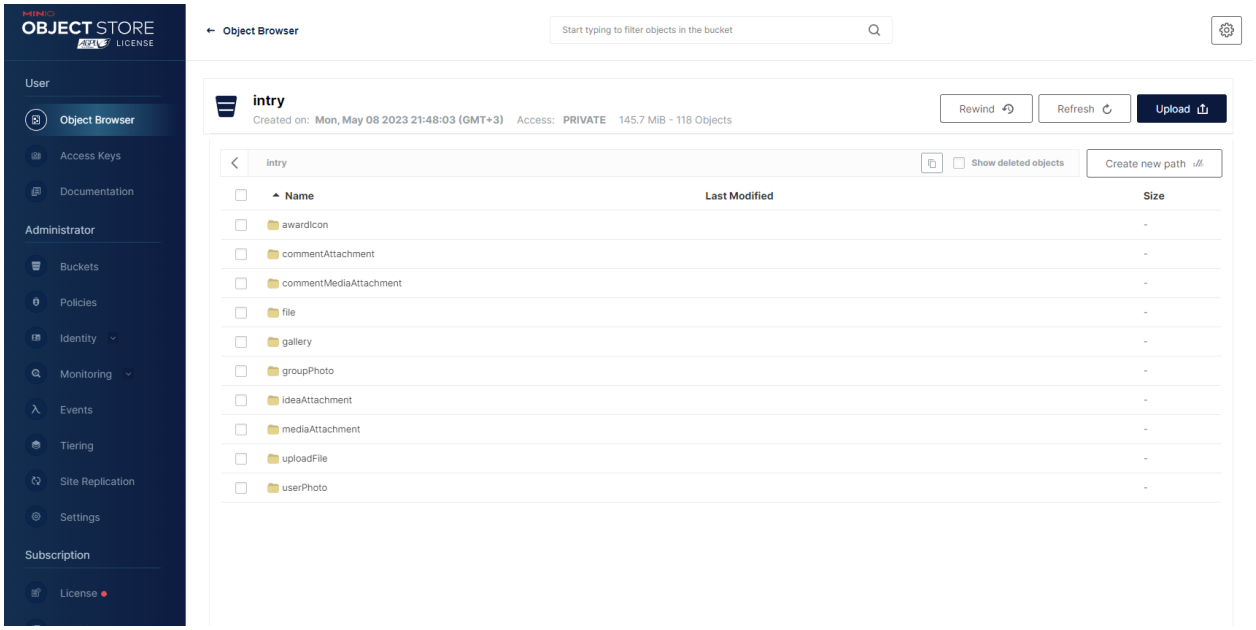

Рисунок 28 Проводник по данным в бакете