



Инструкция по установке и настройке решения

## ОГЛАВЛЕНИЕ

---

1	АННОТАЦИЯ.....	4
2	ВВЕДЕНИЕ.....	5
3	КОМПОНЕНТЫ.....	6
4	ИНТЕГРИРУЕМЫЕ СИСТЕМЫ.....	7
5	ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ.....	8
5.1	ТРЕБОВАНИЯ К ОБОРУДОВАНИЮ.....	8
5.2	ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ.....	8
6	УСТАНОВКА КОМПОНЕНТОВ.....	9
6.1	СУБД.....	9
6.2	INTRY FILES.....	9
6.3	INTRY SSO.....	10
6.4	INTRY API.....	11
6.5	INTRY UI.....	12
6.6	ШЛЮЗ.....	12
6.7	INTRY JOBS.....	13
7	НАСТРОЙКА ПАРАМЕТРОВ.....	14
7.1	НАСТРОЙКА INTRY SSO (KEYCLOAK).....	14
7.2	НАСТРОЙКА ШЛЮЗА (NGINX).....	27
7.3	НАСТРОЙКА INTRY FILES (MINIO).....	27
7.3.1	Создание бакета.....	27
7.3.2	Создание ключей доступа.....	28

## СПИСОК ТАБЛИЦ

---

ТАБЛИЦА 1	ОСНОВНЫЕ КОМПОНЕНТЫ INTRY.....	6
ТАБЛИЦА 2	ИНТЕГРИРУЕМЫЕ СИСТЕМЫ.....	7
ТАБЛИЦА 3	ТРЕБОВАНИЯ К ОБОРУДОВАНИЮ.....	8
ТАБЛИЦА 4	ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ.....	8

## СПИСОК ИЛЛЮСТРАЦИЙ

---

РИСУНОК 1	СОЗДАНИЕ REALM (ОКРУЖЕНИЯ).....	14
-----------	---------------------------------	----

РИСУНОК 2 СТРАНИЦА СОЗДАНИЯ REALM (ОКРУЖЕНИЯ) .....	15
РИСУНОК 3 СТРАНИЦА СОЗДАННОГО REALM.....	15
РИСУНОК 4 СПИСОК КЛИЕНТОВ ДЛЯ REALM INTRY .....	16
РИСУНОК 5 СОЗДАНИЕ КЛИЕНТА INTRY_SPA В REALM INTRY (СТР 1) .....	16
РИСУНОК 6 СОЗДАНИЕ КЛИЕНТА INTRY_SPA В REALM INTRY (СТР 2) .....	17
РИСУНОК 7 СОЗДАНИЕ КЛИЕНТА INTRY_SPA В REALM INTRY (СТР 3).....	18
РИСУНОК 8 СТРАНИЦА РЕДАКТИРОВАНИЯ СОЗДАННОГО КЛИЕНТА INTRY_SPA .....	19
РИСУНОК 9 СТРАНИЦА CLIENT SCOPES .....	20
РИСУНОК 10 СТРАНИЦА СОЗДАНИЯ CLIENT SCOPE.....	20
РИСУНОК 11 СТРАНИЦА СПИСКА CLIENT SCOPES ПОСЛЕ СОЗДАНИЯ INTRY_API.....	21
РИСУНОК 12 СТРАНИЦА CLIENT SCOPE INTRY_API .....	21
РИСУНОК 13 СТРАНИЦА ВЫБОРА ТИПА КОНФИГУРИРОВАНИЯ ДЛЯ CLIENT SCOPE INTRY_API.....	22
РИСУНОК 14 СТРАНИЦА КОНФИГУРИРОВАНИЯ ПРИВЯЗКИ AUDIENCE ДЛЯ CLIENT SCOPE INTRY_API.....	22
РИСУНОК 15 ДОБАВЛЕНИЕ SCOPE INTRY_API В КЛИЕНТ INTRY_SPA (РИС 1) .....	23
РИСУНОК 16 ДОБАВЛЕНИЕ SCOPE INTRY_API В КЛИЕНТ INTRY_SPA (РИС 2) .....	23
РИСУНОК 17 ДОБАВЛЕННЫЙ SCOPE В СПИСКЕ .....	24
РИСУНОК 18 СОЗДАНИЕ ПОДКЛЮЧЕНИЯ К ACTIVE DIRECTORY .....	24
РИСУНОК 19 СТРАНИЦА НАСТРОЙКИ LDAP ПОДКЛЮЧЕНИЯ (ЧАСТЬ1).....	25
РИСУНОК 20 СТРАНИЦА НАСТРОЙКИ LDAP ПОДКЛЮЧЕНИЯ (ЧАСТЬ2).....	26
РИСУНОК 21 ДОБАВЛЕННЫЕ LDAP ПОДКЛЮЧЕНИЯ В ФЕДЕРАЦИИ ПОЛЬЗОВАТЕЛЕЙ.....	26
РИСУНОК 22 СТРАНИЦА СОЗДАНИЯ БАКЕТА .....	27
РИСУНОК 23 СТРАНИЦА СОЗДАННОГО БАКЕТА .....	28
РИСУНОК 24 СТРАНИЦА СПИСКА ПОЛЬЗОВАТЕЛЕЙ .....	28
РИСУНОК 25 СТРАНИЦА СОЗДАНИЯ ПОЛЬЗОВАТЕЛЯ.....	28
РИСУНОК 26 СТРАНИЦА СПИСКА КЛЮЧЕЙ ДОСТУПА ДЛЯ ПОЛЬЗОВАТЕЛЯ .....	29
РИСУНОК 27 СТРАНИЦА СОЗДАНИЯ КЛЮЧА ДОСТУПА.....	29
РИСУНОК 28 ПРОВОДНИК ПО ДАННЫМ В БАКЕТЕ.....	30

## 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	<a href="https://nginx.org/">https://nginx.org/</a>
2.	Intry UI	Angular	11	<a href="https://angular.io/cli">https://angular.io/cli</a>
3.	Intry API	.NET	6	<a href="https://github.com/dotnet/aspnetcore#readme">https://github.com/dotnet/aspnetcore#readme</a>
4.	Intry SSO	Keycloak	21.1.1	<a href="https://www.keycloak.org/">https://www.keycloak.org/</a>
5.	Intry Jobs	.NET	6	<a href="https://github.com/dotnet/aspnetcore#readme">https://github.com/dotnet/aspnetcore#readme</a>
6.	Intry Files	MinIO	RELEASE.2023-04-07T05-28-58Z	<a href="https://min.io/">https://min.io/</a>
7.	СУБД	PostgreSQL	15.2	<a href="https://www.postgresql.org/">https://www.postgresql.org/</a>

## 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](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](https://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="e17b67t3HOc1DPqA4ZYp" \
  -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@sswOrd;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](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@sswOrd;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».

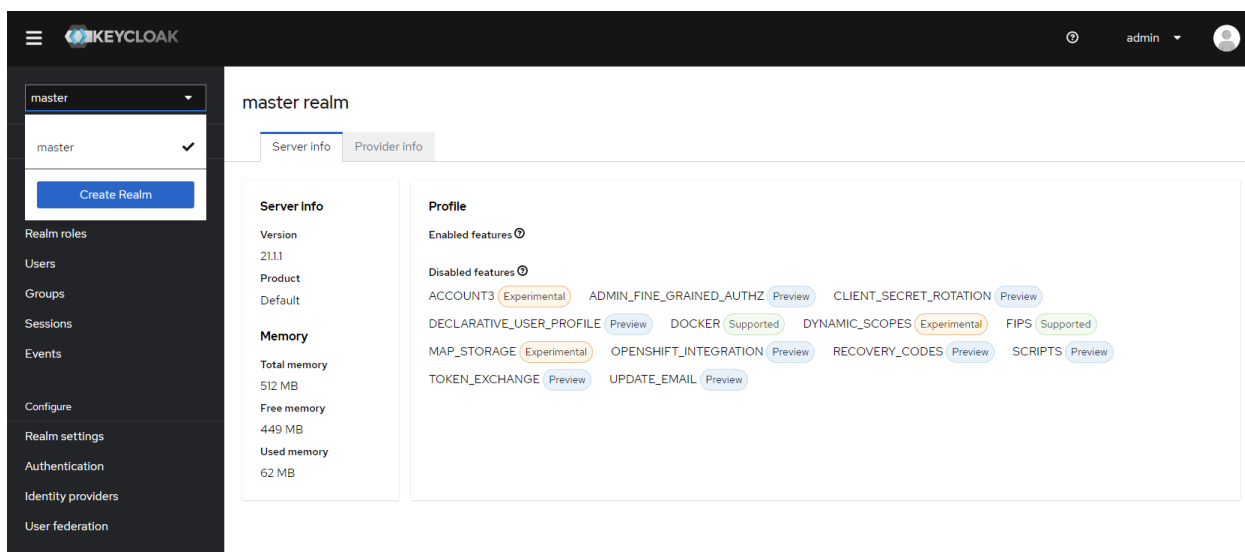


Рисунок 1 Создание Realm (окружения)

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

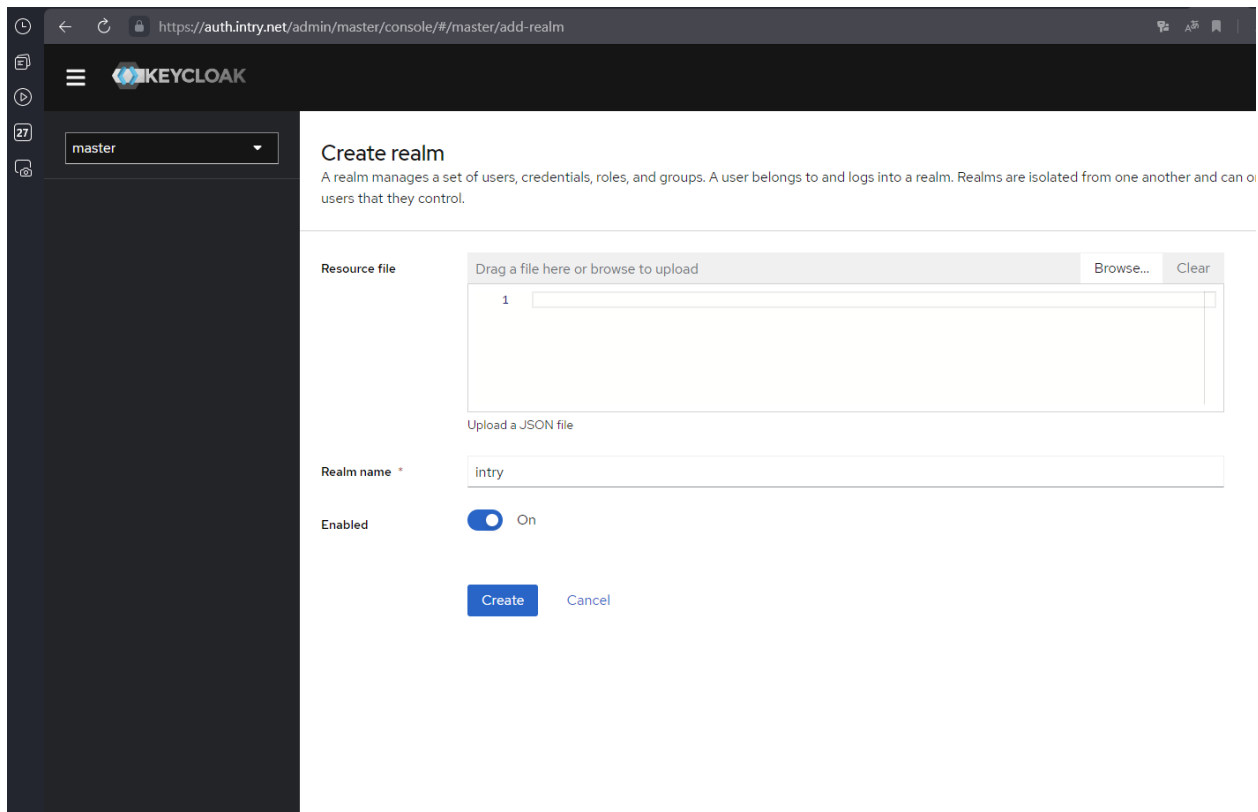


Рисунок 2 Страница создания Realm (окружения)

Созданный Realm будет отображаться в выпадающем меню.

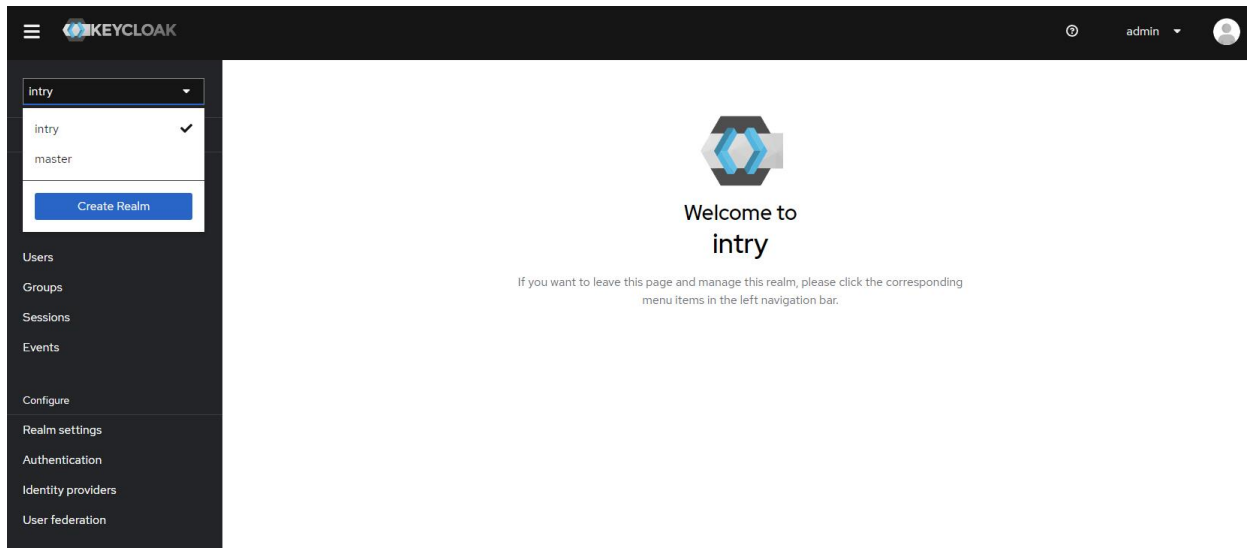


Рисунок 3 Страница созданного Realm

## 2. Создать и настроить клиент

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

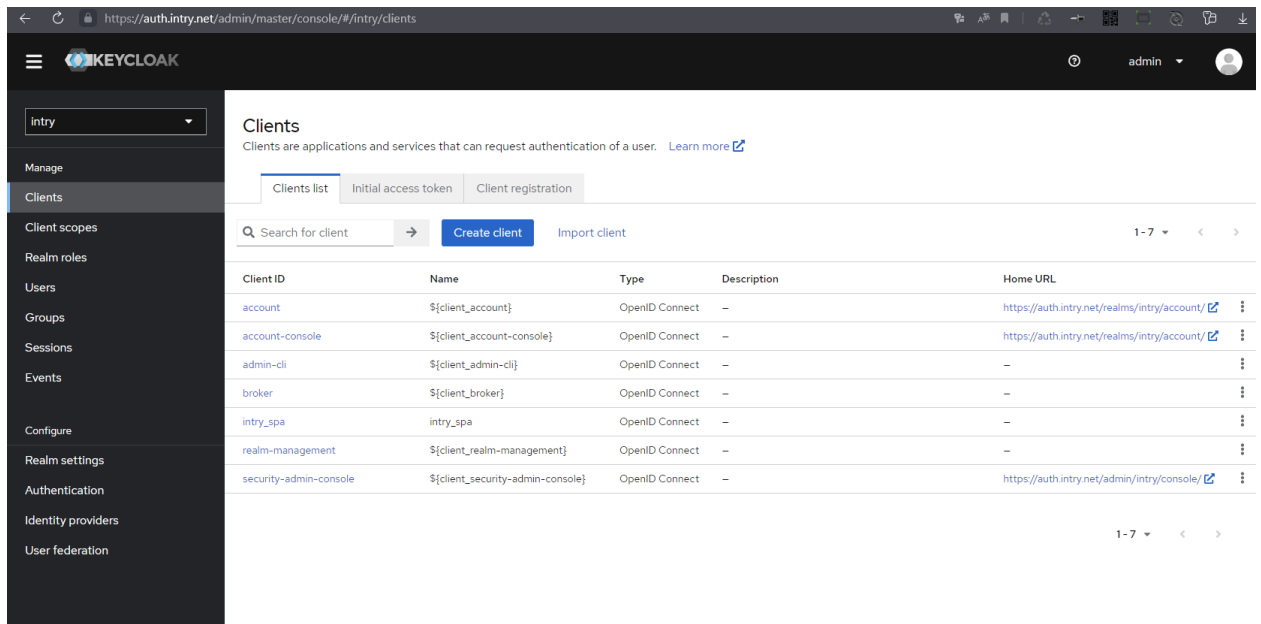


Рисунок 4 Список клиентов для realm Intry

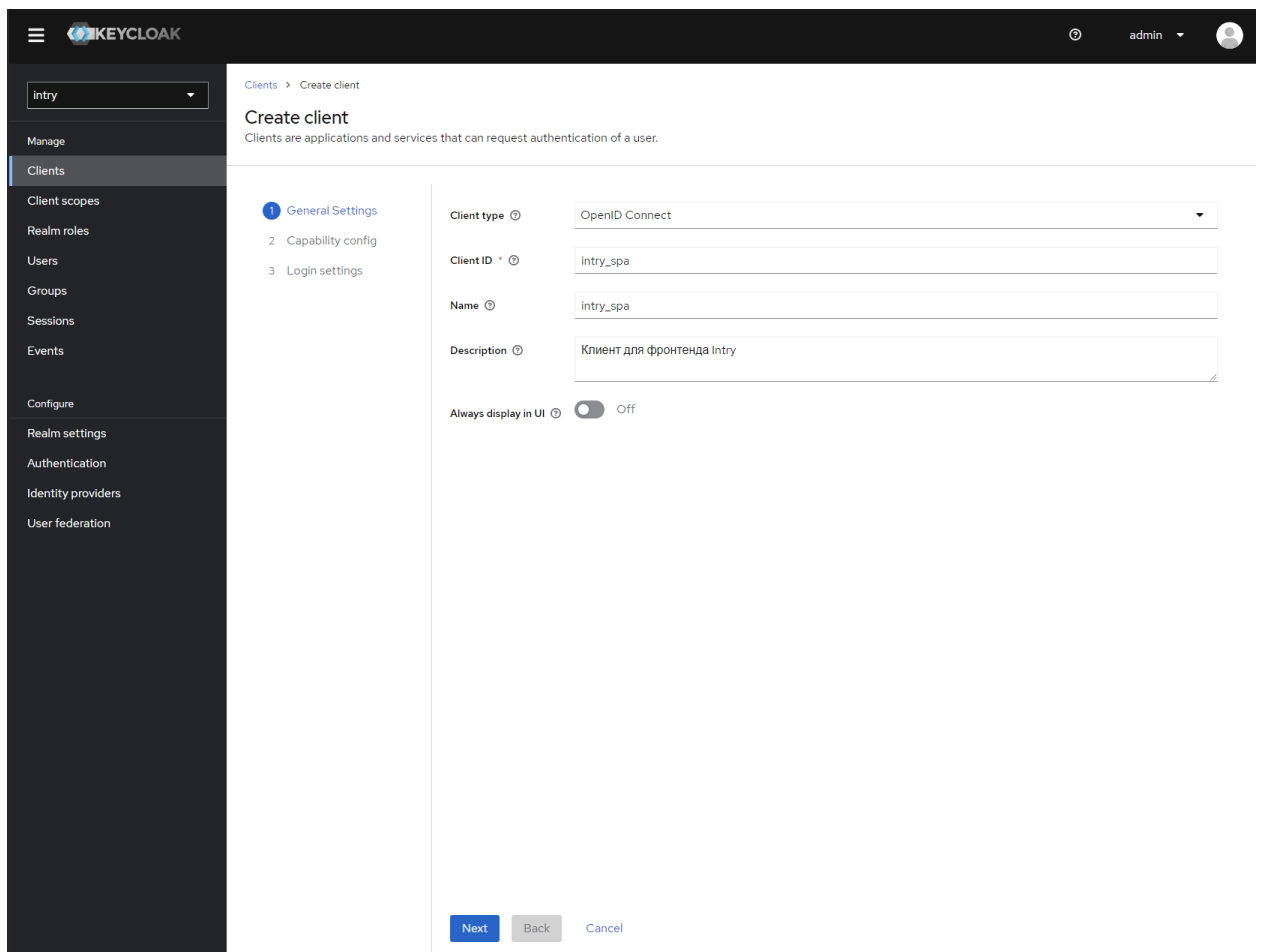


Рисунок 5 Создание клиента intry\_spa в realm Intry (стр 1)



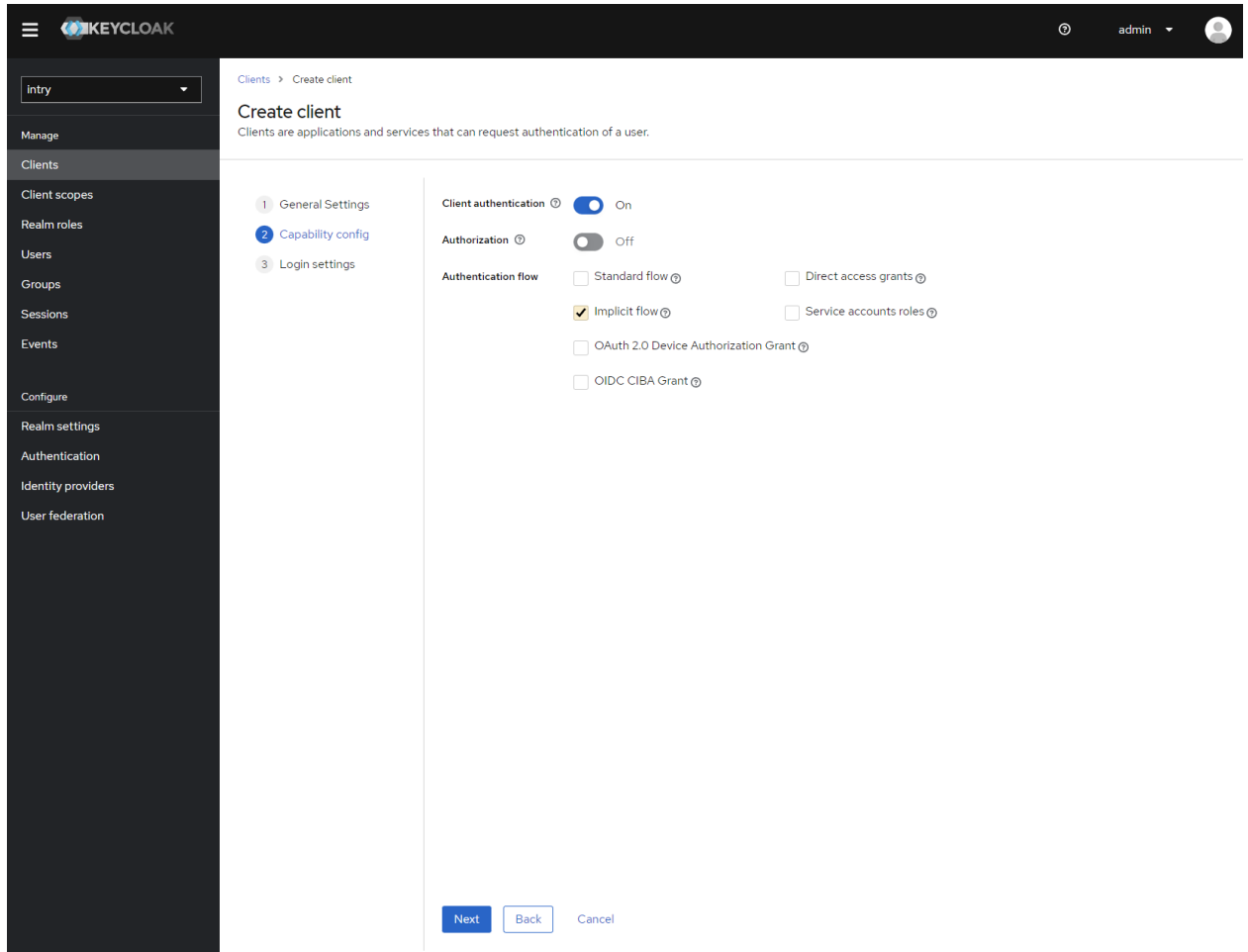


Рисунок 6 Создание клиента *intry\_spa* в realm *Intry* (стр 2)

The screenshot shows the Keycloak administration interface for the 'intry' realm. The left sidebar contains navigation options: Manage, Clients, Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main content area is titled 'Create client' and includes a breadcrumb 'Clients > Create client'. Below the title, it states 'Clients are applications and services that can request authentication of a user.' The 'Login settings' tab is selected, showing a list of configuration options: 'Root URL', 'Home URL', 'Valid redirect URIs' (with four entries: 'https://<ui-url>/silent-renew.html', 'https://<ui-url>/auth-callback', 'https://<api-url>/swagger/oauth2-redirect.html', and 'https://<api-url>/signin-oidc'), 'Valid post logout redirect URIs' (with one entry: 'https://<ui-url>'), and 'Web origins' (with one entry: '\*'). At the bottom, there are 'Save', 'Back', and 'Cancel' buttons.

Рисунок 7 Создание клиента *intry\_spa* в realm *Intry* (стр 3)

The screenshot displays the Keycloak Admin Console interface. On the left is a dark sidebar with navigation options: Manage, Clients, Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main content area is titled 'Clients > Client details' and shows the configuration for a client named 'intry\_spa' (OpenID Connect). The client is currently 'Enabled'. Below this, there are tabs for 'Settings', 'Keys', 'Credentials', 'Roles', 'Client scopes', 'Sessions', and 'Advanced'. The 'Settings' tab is active, showing 'General Settings' and 'Access settings'. In 'General Settings', the 'Client ID' and 'Name' are both 'intry\_spa', and the 'Description' is 'Клиент для фронтенда intry'. The 'Always display in UI' toggle is turned off. The 'Access settings' section includes fields for 'Root URL', 'Home URL', 'Valid redirect URIs' (with a list of four URLs and an 'Add valid redirect URIs' button), 'Valid post logout redirect URIs' (with one URL and an 'Add valid post logout redirect URIs' button), 'Web origins' (with a '\*' value and an 'Add web origins' button), and 'Admin URL'. A 'Jump to section' sidebar on the right lists: General Settings, Access settings, Capability config, Login settings, and Logout settings. At the bottom of the settings area are 'Save' and 'Revert' buttons.

Рисунок 8 Страница редактирования созданного клиента intry\_spa

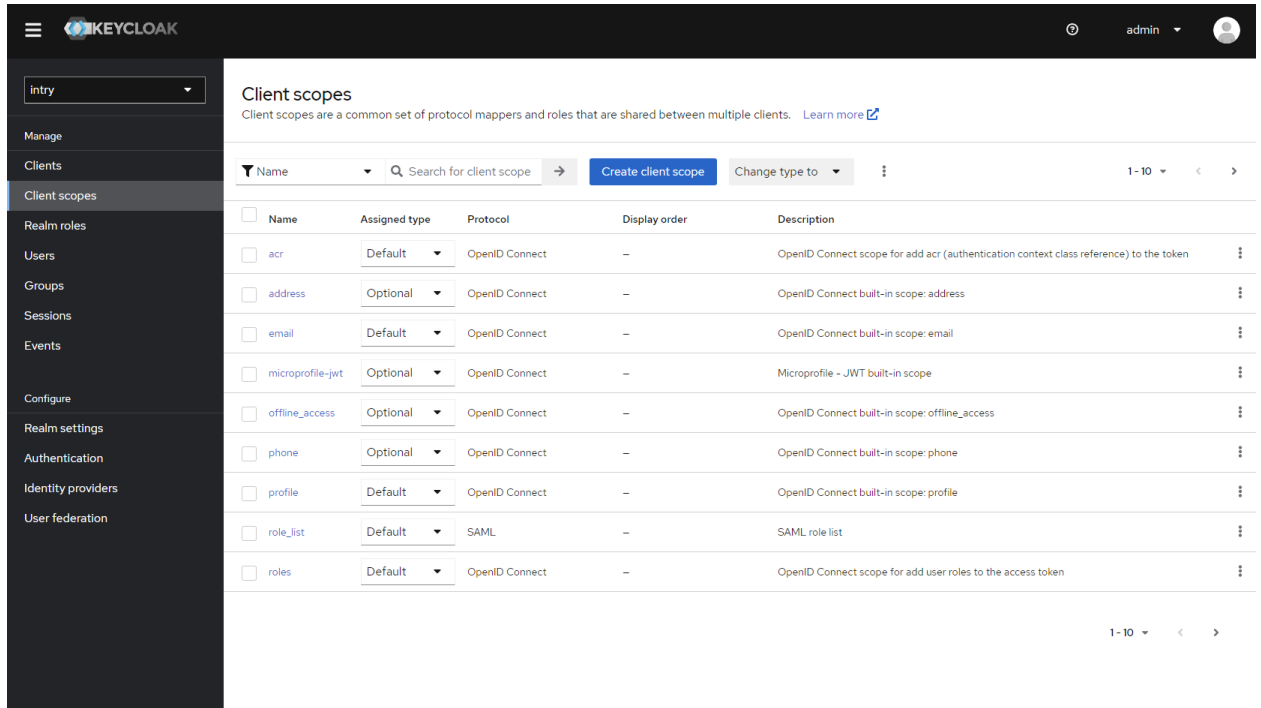


Рисунок 9 Страница client scopes

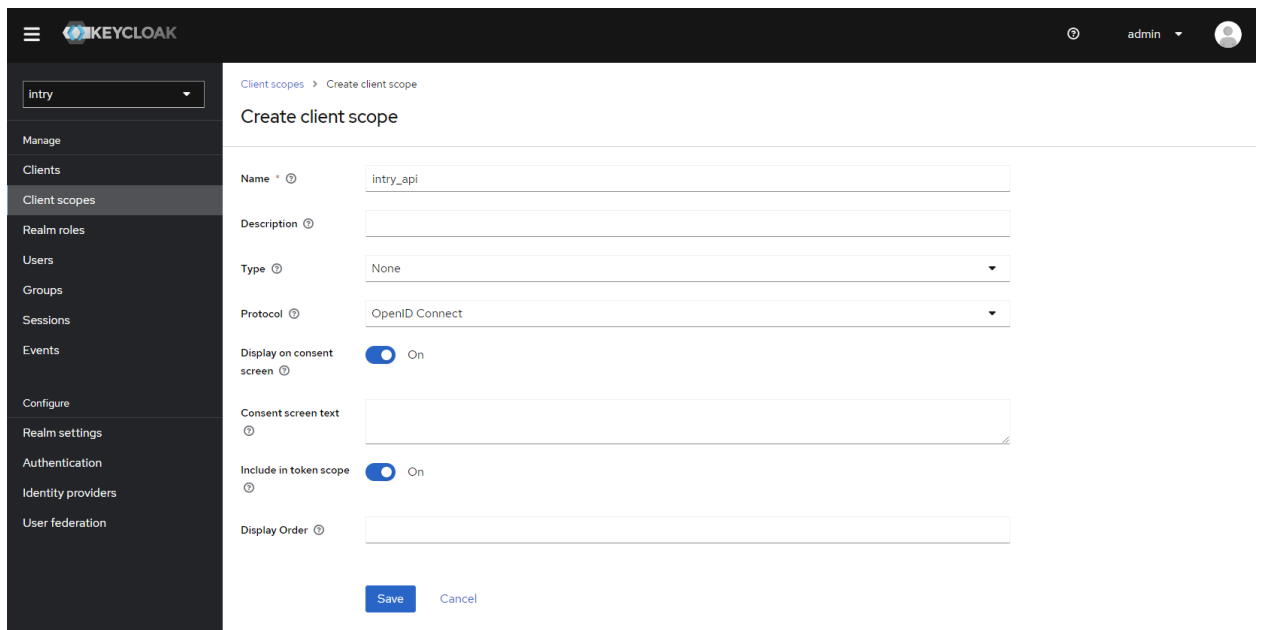


Рисунок 10 Страница создания client scope

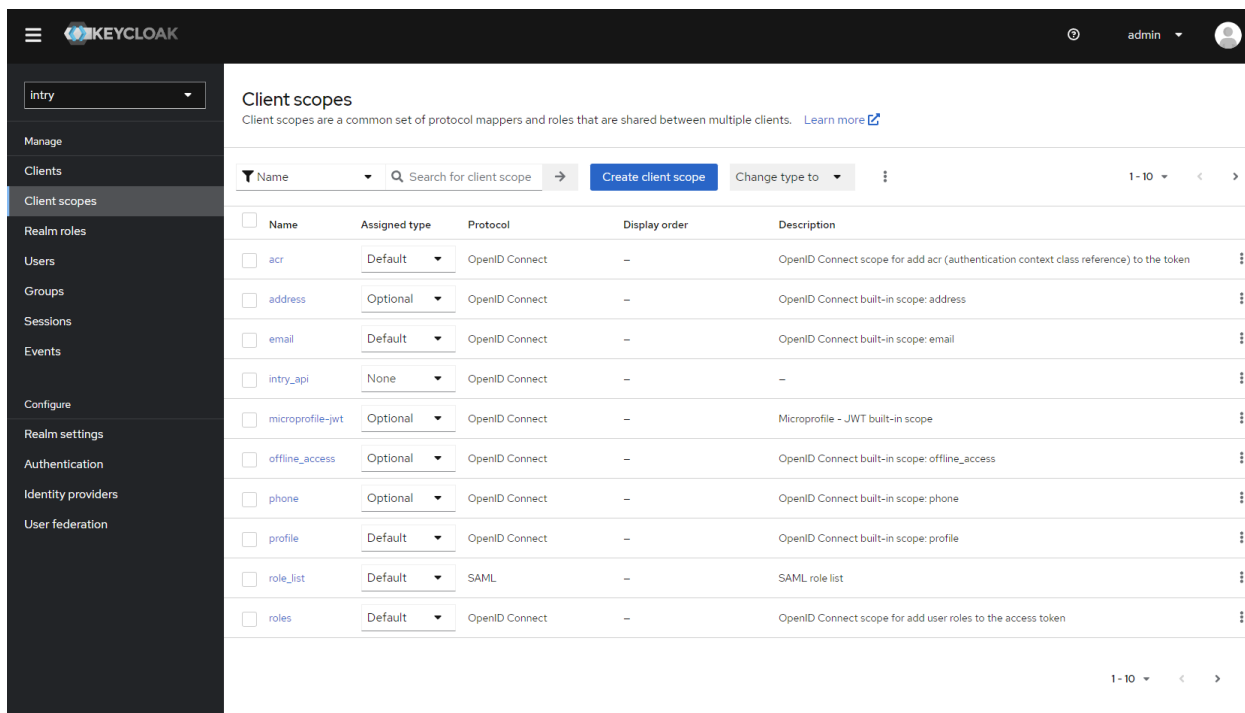


Рисунок 11 Страница списка client scopes после создания intry\_api

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.

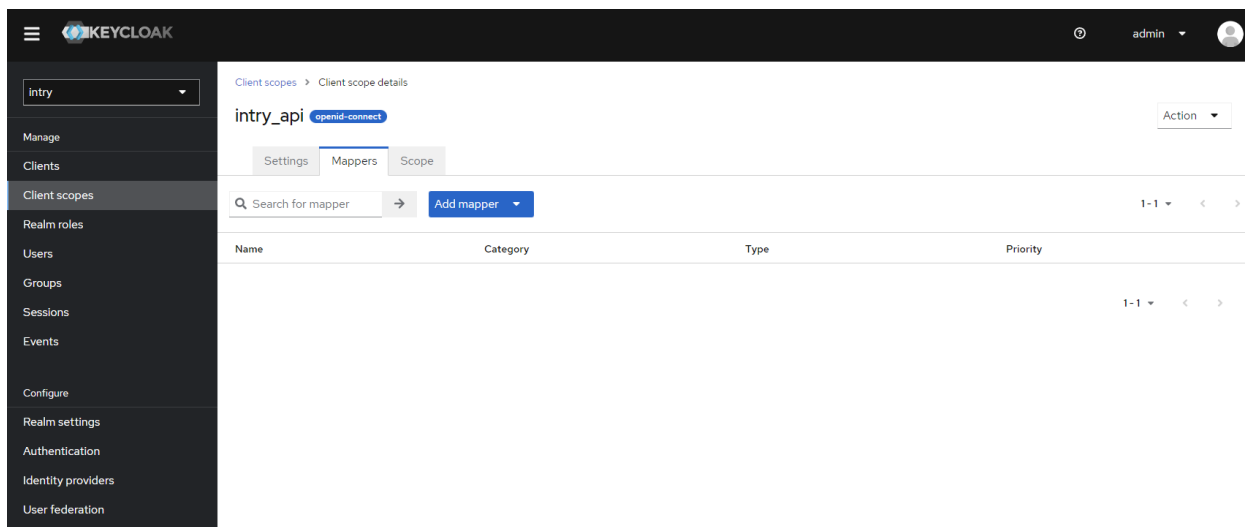


Рисунок 12 Страница client scope intry\_api

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

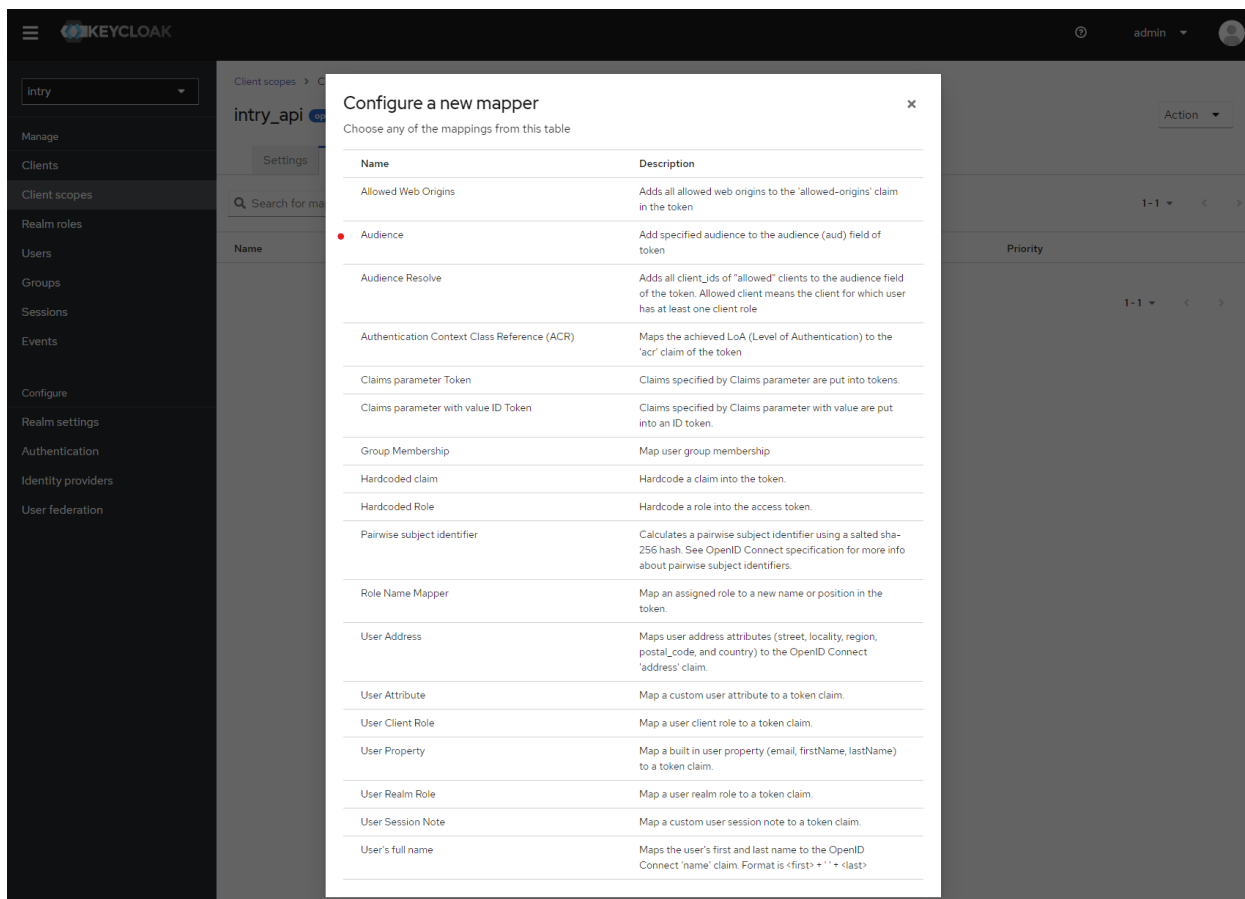


Рисунок 13 Страница выбора типа конфигурирования для client scope intry\_api

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

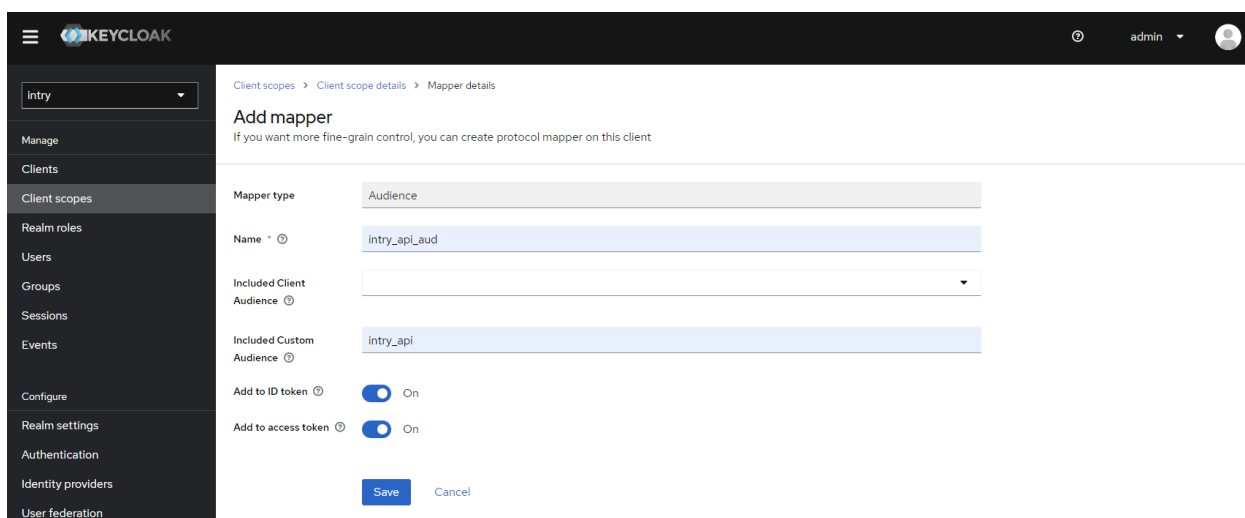


Рисунок 14 Страница конфигурирования привязки Audience для client scope intry\_api

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.

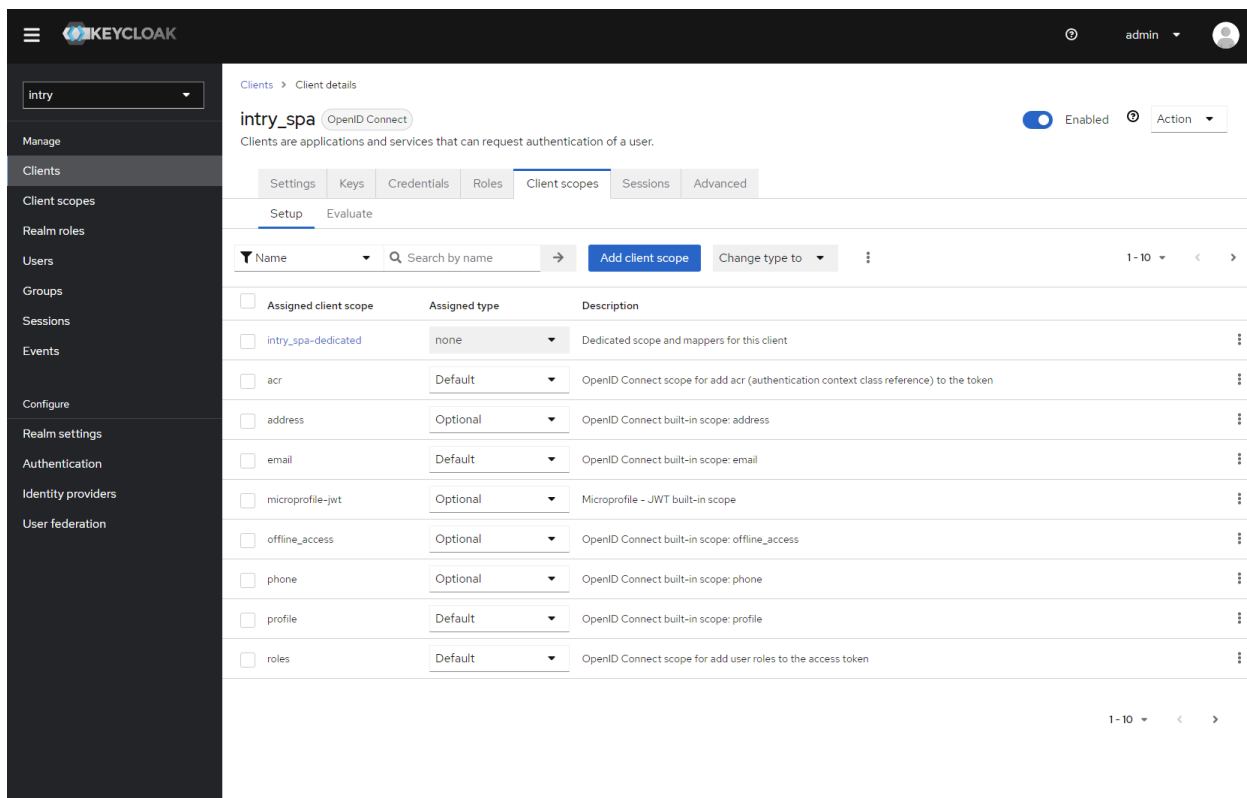


Рисунок 15 Добавление scope intry\_api в клиент intry\_spa (рис 1)

После нажатия кнопки «Add client scope», во всплывающем окне необходимо добавить требуемый scope – intry\_api.

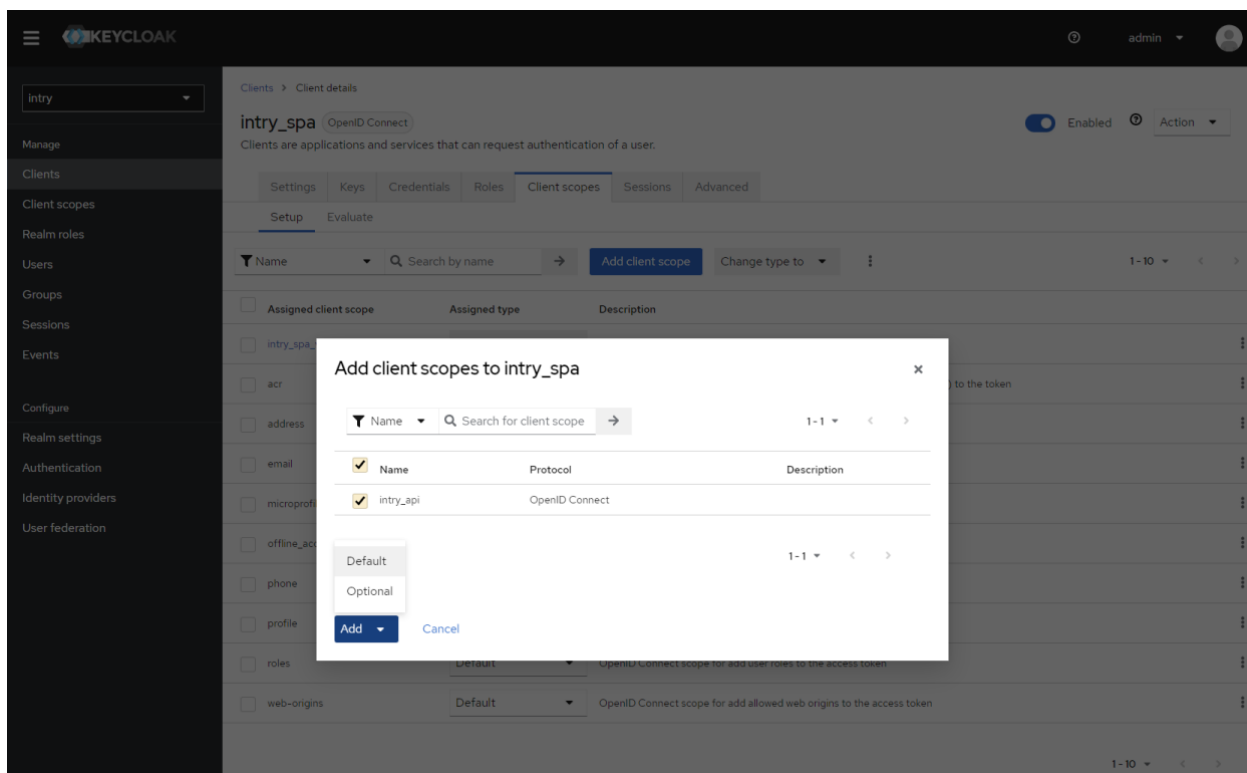


Рисунок 16 Добавление scope intry\_api в клиент intry\_spa (рис 2)

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

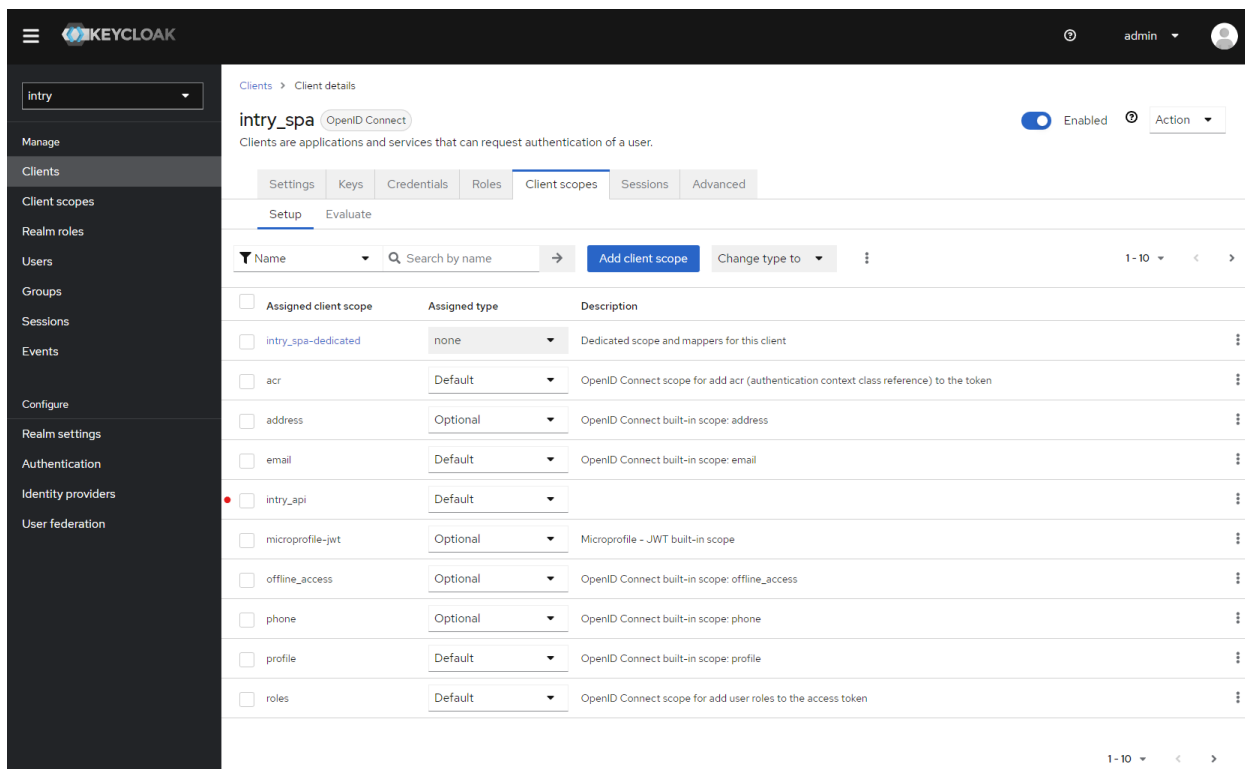


Рисунок 17 Добавленный scope в списке

### 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.

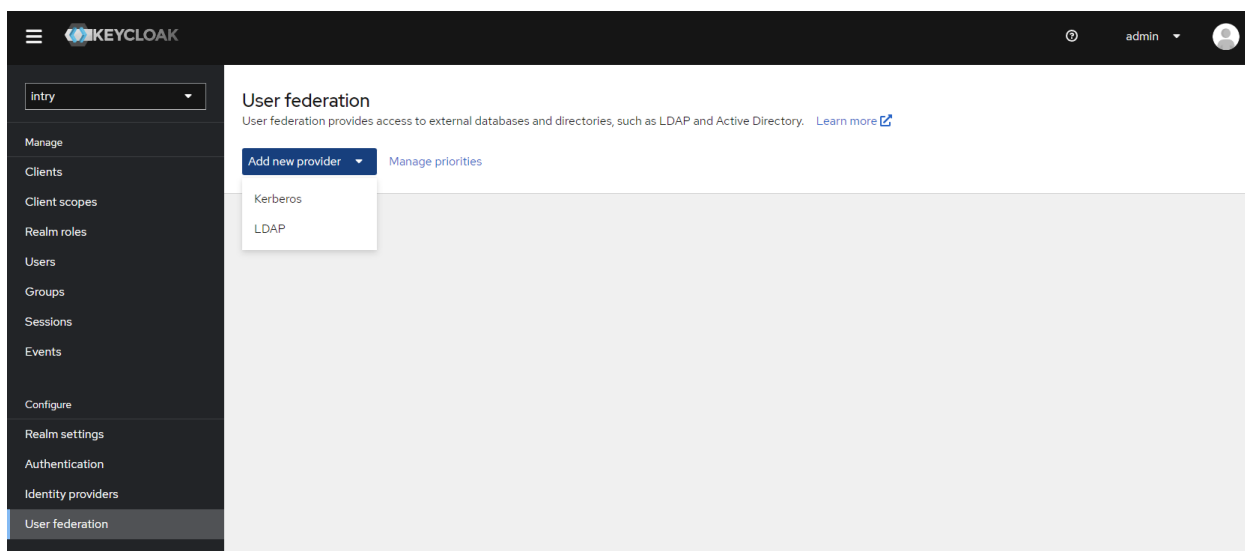


Рисунок 18 Создание подключения к Active Directory



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.

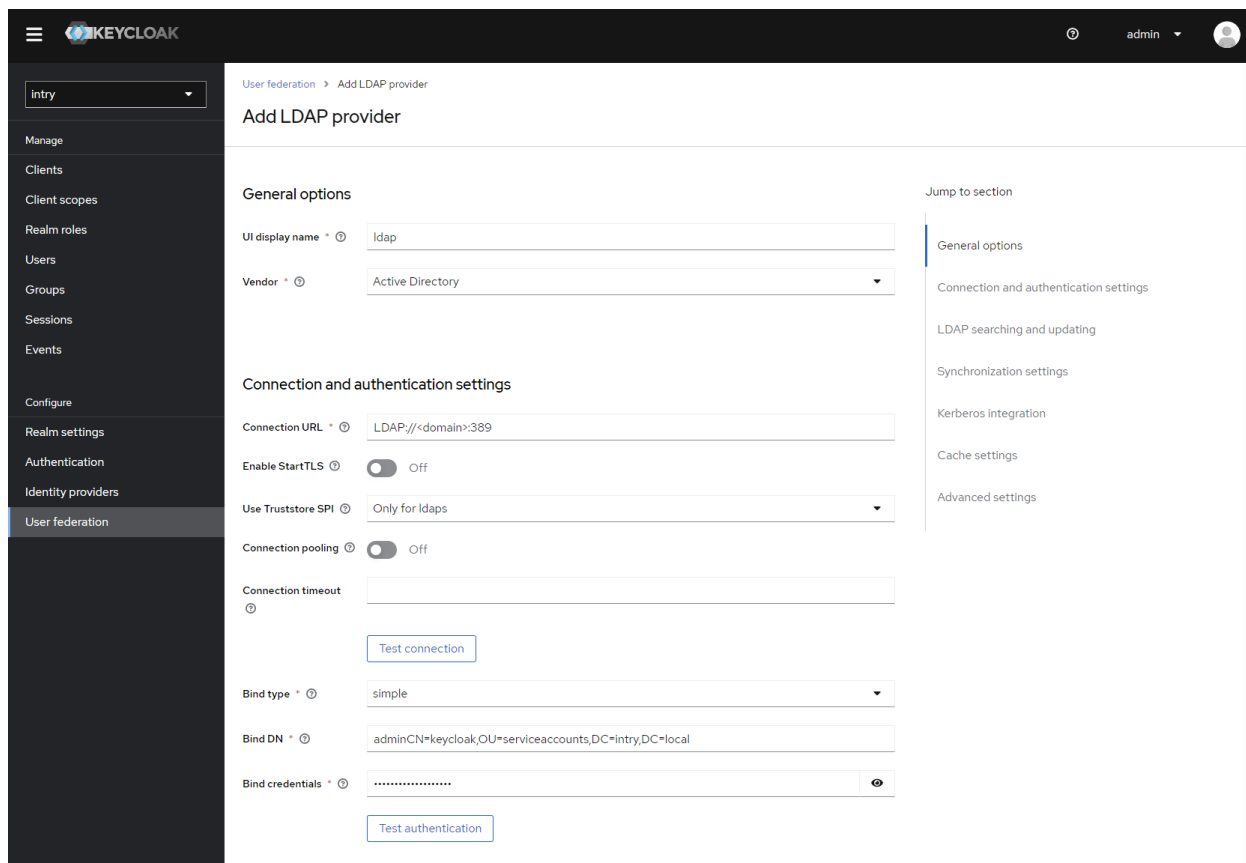


Рисунок 19 Страница настройки LDAP подключения (часть1)

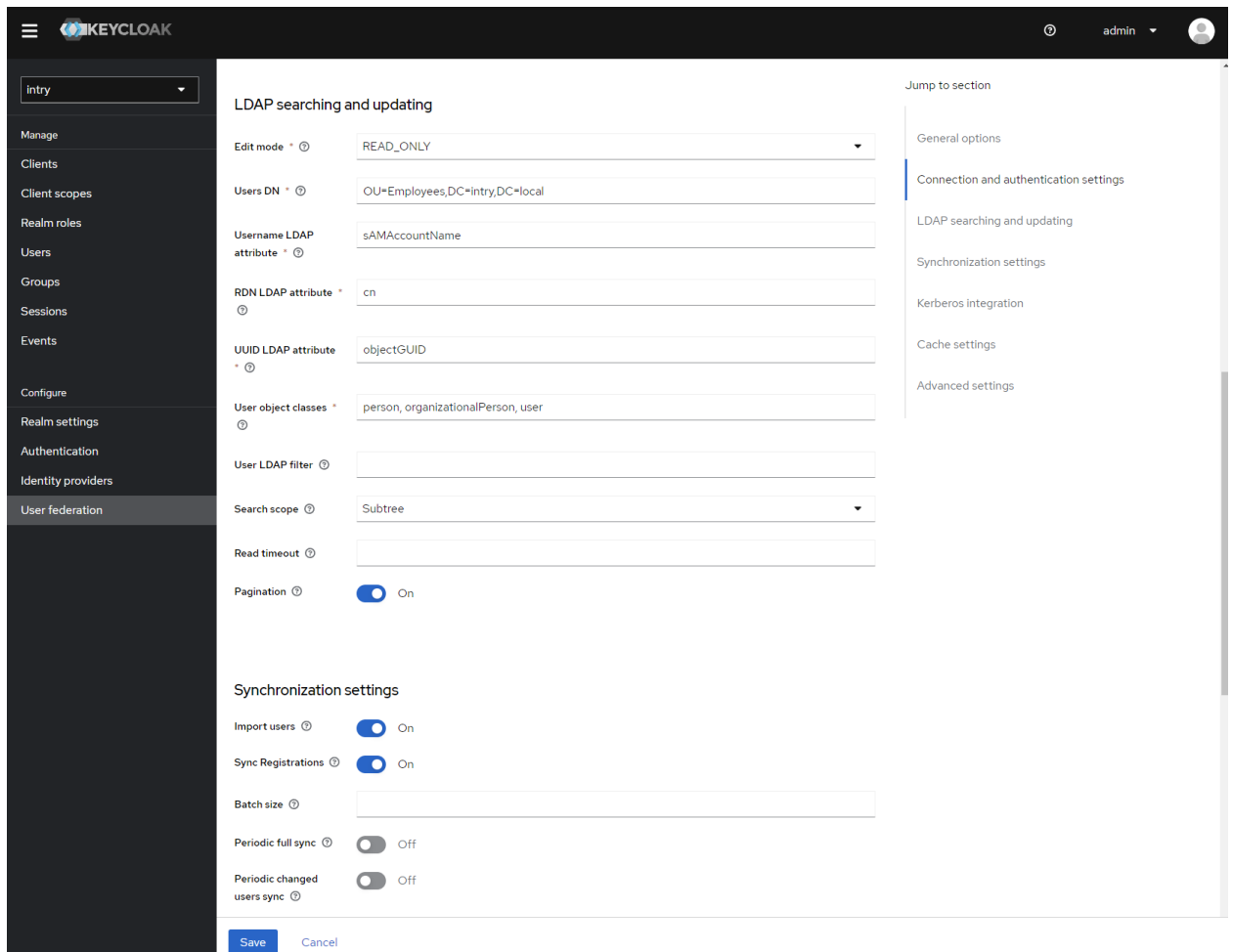


Рисунок 20 Страница настройки LDAP подключения (часть2)

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

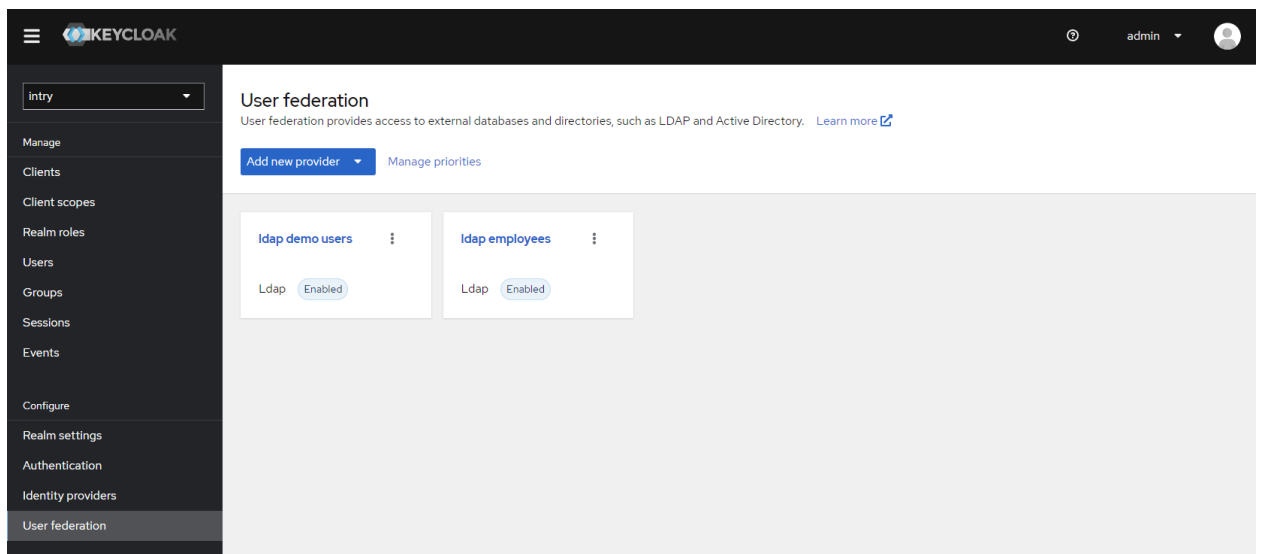


Рисунок 21 Добавленные LDAP подключения в федерации пользователей

## 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»

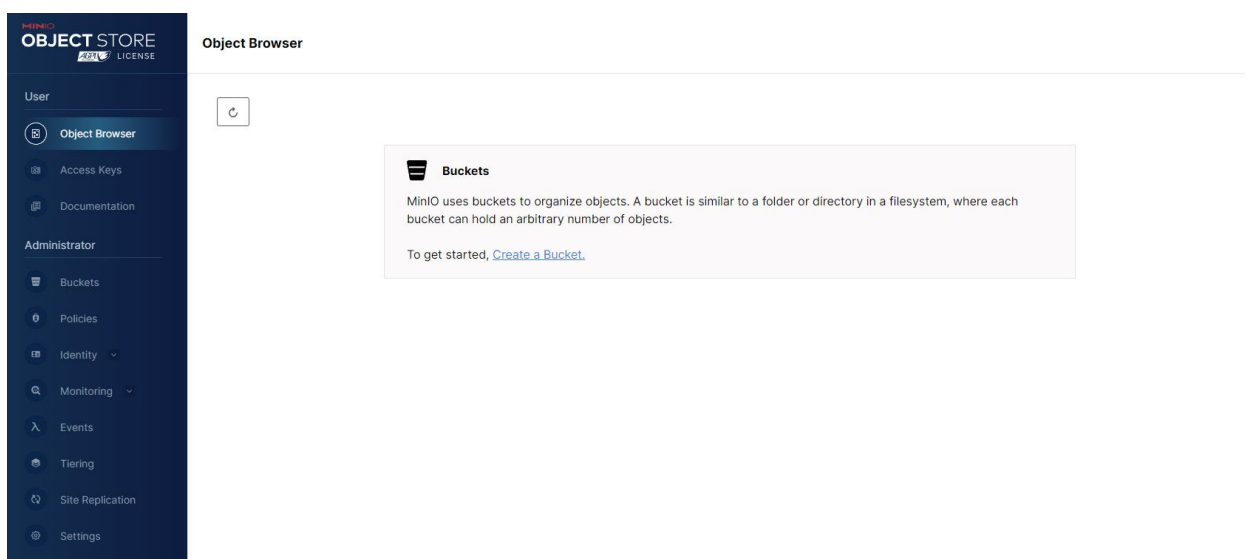


Рисунок 22 Страница создания бакета

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

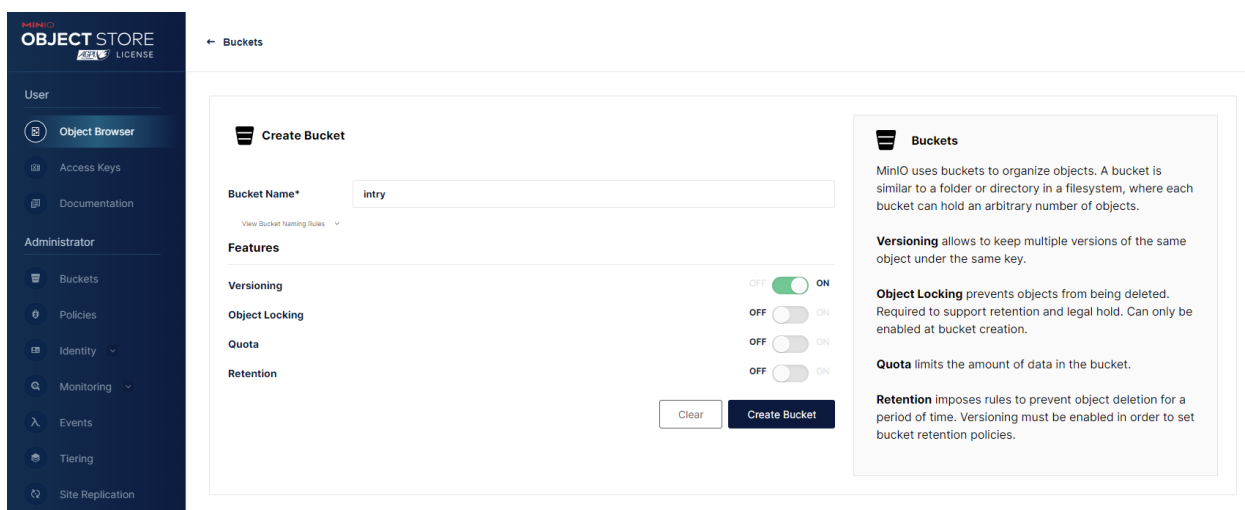


Рисунок 23 Страница созданного бакета

## 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 типичным подключением к s3-совместимому хранилищу.

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

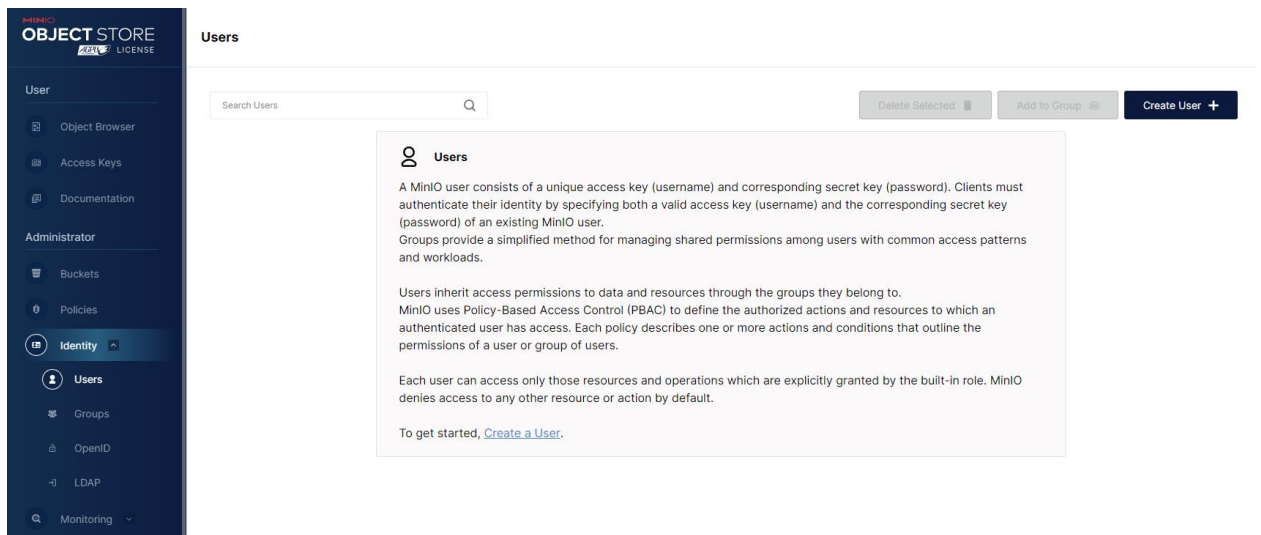


Рисунок 24 Страница списка пользователей

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

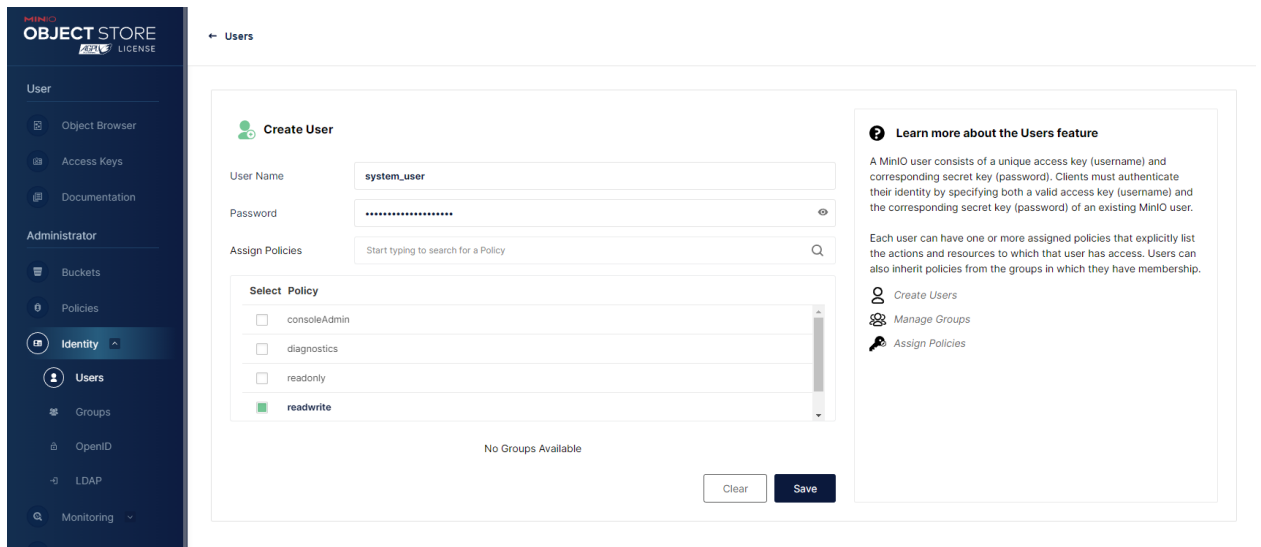


Рисунок 25 Страница создания пользователя

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

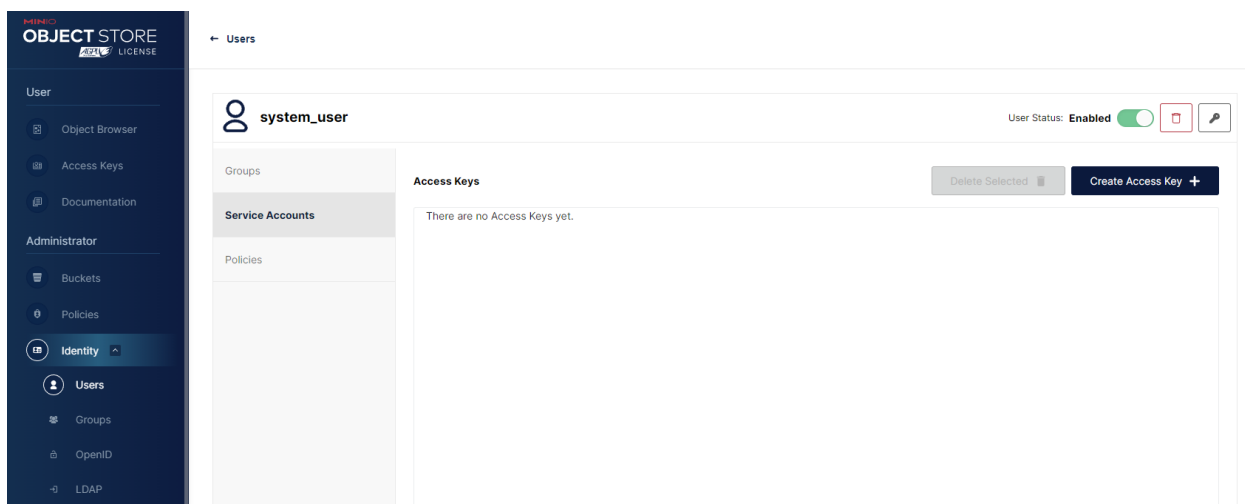


Рисунок 26 Страница списка ключей доступа для пользователя

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

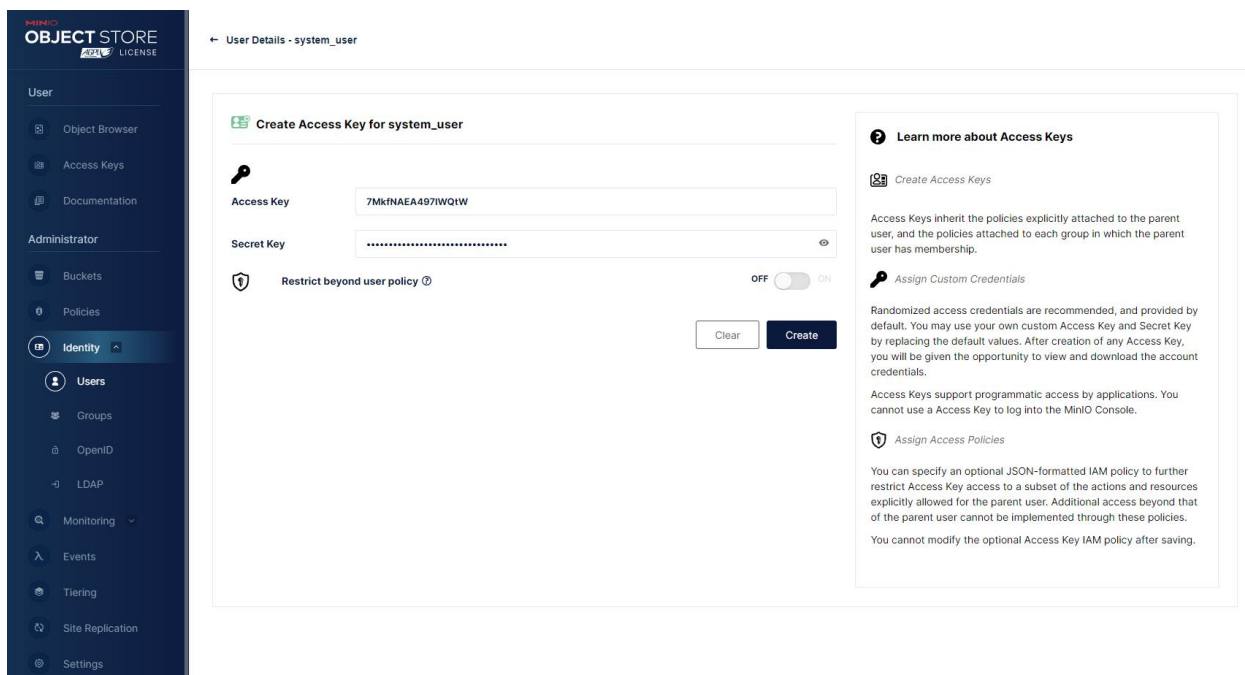


Рисунок 27 Страница создания ключа доступа

В дальнейшем можно просматривать и администрировать файлы через данный интерфейс.

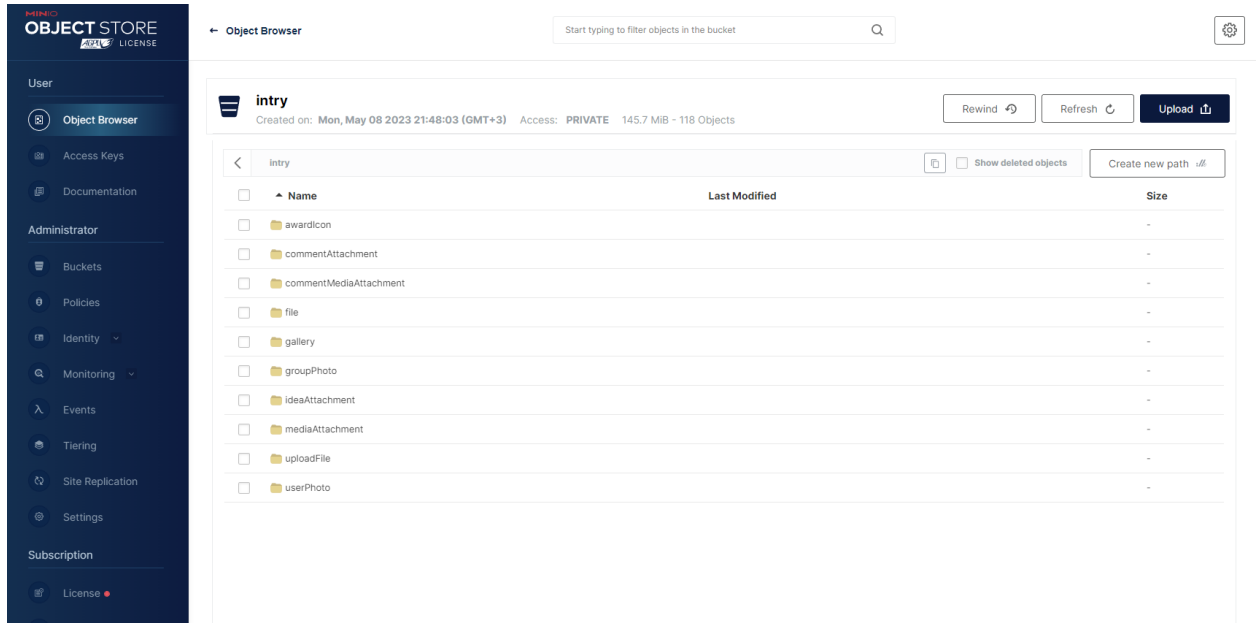


Рисунок 28 Проводник по данным в бакете