



Инструкция по установке экземпляра программного обеспечения “Kolmogorov AI. Continuity.”, предоставленного для проведения экспертной проверки

Оглавление

Оглавление	1
Введение	2
О Kolmogorov.ai	2
Назначение документа	2
Системные требования к ПО «Kolmogorov.ai Continuity»	2
Архитектура и компоненты ПО	3
Компоненты	3
Архитектура	3
Пререквизиты	4
Установка компонент	4
Registry	4
Список скриптов установки (helm-чартов)	5
Keycloak	5
values.yaml	5
install	7
Continuity	7
values.yaml	7
install	8
Continuity Frontend	8
values.yaml	8
install	9
Приложение 1. Настройка Keycloak	9
Создание realm	9
Создание client	10
Настройка client	11
Настройка Mapper	12
Создание ролей	13
Добавление ролей пользователям	13

ООО «Дата Сапиенс»

<https://kolmogorov.ai/>

ИНН 9701181979

ОГРН 1217700358083



Введение

Данная версия документа подготовлена для проведения экспертной проверки установки экземпляра ПО.

Установка ПО «Kolmogorov.ai Continuity» осуществляется методом развертывания на кластере Kubernetes с использованием helm. Образы и скрипты для установки находятся в репозитории ***registry.datasapience.ru***.

Project Name: ***russian-registry***

Helm Charts: ***continuity, continuity-ui***.

В описанном в данном документе процессе установки экземпляра ПО через Helm необходимо использовать следующую учетную запись:

Пользователь (<your-name>): ***ds-russian-registry***

Пароль <your-pword>: ***avX4JsFkxvKNB5a9j62XTdfvqLUhziXb***

Демонстрационный стенд с развернутым экземпляром ПО расположен по адресу:

<https://continuity-prod.demo.datasapience.ru/project>

Учетная запись для пользовательского доступа:

Пользователь: ***demo***

Пароль: ***JM6{hrnxE5!^@bq}***

Контакты технического специалиста:

Ростислав Изимов

Телефон: +7 (977) 889-00-20

E-mail: rostislav.izimov@glowbyteconsulting.com

Дополнительный e-mail: contact@kolmogorov.ai

О Kolmogorov.ai

Программное обеспечение «Kolmogorov.ai Continuity» предназначено для решения задач управления процессами жизненного цикла моделей машинного обучения и продвинутой аналитики.

Назначение документа

В настоящем документе описаны системные требования, а также порядок установки и настройки компонент программного обеспечения «Kolmogorov.ai Continuity».

Системные требования к ПО «Kolmogorov.ai Continuity»

Оперативная память	Не менее 8 Гб
Частота процессора	Не менее 2.3 ГГц
Количество ядер	Не менее 4
Свободное место на HDD	Не менее 100 Гб
Скорость интернета на входящий и исходящий трафик	Не менее 50 Мбит
Операционная система	Рекомендуемая – Ubuntu 20.04 LTS Поддерживаемая – любая *nix-ОС с поддержкой Kubernetes

Архитектура и компоненты ПО

Компоненты

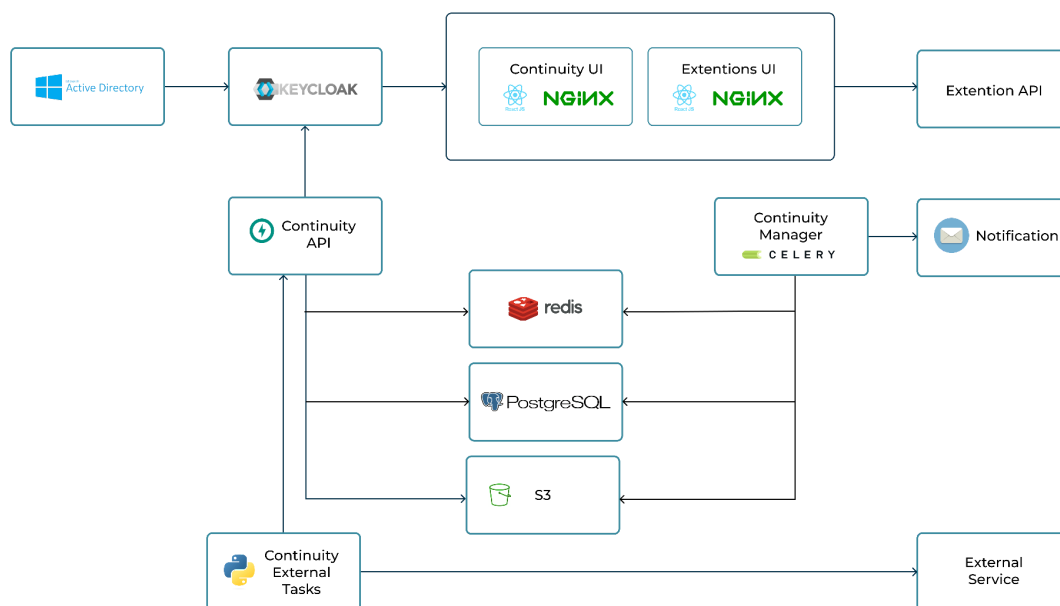
- **Keycloak.** Сервис аутентификации. Работает по принципу OAuth2.
- **PostgreSQL.** База данных для хранения метаданных.
- **Continuity Web.** Интерфейс для работы с системой из браузера.
- **Continuity API.** Интерфейс для работы с API системы.
- **Continuity Celery.** Обработчик background задач.

Интеграционные компоненты:

- **AD. LDAP.** Для dev установки можно не использовать (создавать локальных пользователей на keycloak).
- **S3.** сервис для хранения файлов.

Архитектура

Архитектура решения представлена на схеме ниже.



Прerequisites

Описанный в настоящем документе процесс установки предполагает выполнение следующих прerequisites:

- Kubernetes,
- Helm,
- Kubernetes Ingress,
- Возможность подключения внешнего registry (описано ниже),
- Доступ к <https://github.com>.

Установка компонент

Registry

Детальную инструкцию подключения внешнего registry см. также в документации Kubernetes:

<https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/>

Необходимо через kubectl добавить секреты для docker registry. В коде ниже необходимо заменить **<your-name>** и **<your-pword>** на полученные ранее реквизиты доступа к **registry.datasapience.ru**.

(Как указано во введении:

Пользователь (**<your-name>**): **ds-russian-registry**

Пароль **<your-pword>**: **avX4JsFkxvKNB5a9j62XTdfvqLUhziXb**

```
kubectl create secret docker-registry regcred \  
  --docker-server=registry.datasapience.ru \  
  --docker-username=<your-name> \  
  --docker-password=<your-pword>
```

Helm registry:

```
helm repo add ds \  
  https://registry.datasapience.ru/chartrepo/russian-registry \  
  --username <your-name> \  
  --password <your-pword>  
  
helm repo update  
  
helm search repo ds
```

Список скриптов установки (helm-чартов)

Установка компонент ПО осуществляется последовательным запуском helm-чартов:

1. keycloak;
2. continuity-backend (включает в себе чарты redis, postgres);
3. continuity-frontend.

Keycloak

Helm chart: <https://github.com/codecentric/helm-charts/tree/master/charts/keycloak>.

Для работы с компонентой keycloak необходимо настроить ingress. Пример конфигурации:

values.yaml

```
keycloakUser: "admin"  
keycloakPassword: "admin"  
  
databaseUser: "keycloak-user"  
databasePassword: "dbpassword"  
  
# replicas: 1  
  
postgresql:  
  resources:  
    requests:
```

```
    memory: "512Mi"
    cpu: "250m"
  limits:
    memory: "1024Mi"
    cpu: "500m"

resources:
  requests:
    memory: "512Mi"
    cpu: "250m"
  limits:
    memory: "1024Mi"
    cpu: "1000m"

extraEnv: |
- name: JAVA_OPTS
  value: >-
    -XX:+UseContainerSupport
    -XX:MaxRAMPercentage=50.0
    -Djava.net.preferIPv4Stack=true
    -Djboss.modules.system.pkgs=$JBASS_MODULES_SYSTEM_PKGS
    -Djava.awt.headless=true
    -Dkeycloak.profile.feature.upload_scripts=enabled
- name: KEYCLOAK_LOGLEVEL
  value: INFO
- name: PROXY_ADDRESS_FORWARDING
  value: "true"

extraEnvFrom: |
- secretRef:
  name: '{{ include "keycloak.fullname" . }}-cred'
- secretRef:
  name: '{{ include "keycloak.fullname" . }}-db'

secrets:
  cred:
    stringData:
      KEYCLOAK_USER: '{{ .Values.keycloakUser }}'
      KEYCLOAK_PASSWORD: '{{ .Values.keycloakPassword }}'
  db:
    stringData:
      DB_USER: '{{ .Values.databaseUser }}'
      DB_PASSWORD: '{{ .Values.databasePassword }}'

ingress:
  enabled: true
  ingressClassName: "nginx"
  servicePort: http
  annotations:
    ingress.kubernetes.io/ssl-redirect: "true"
    nginx.ingress.kubernetes.io/ssl-redirect: "true"
    nginx.ingress.kubernetes.io/proxy-body-size: "128k"
    nginx.ingress.kubernetes.io/server-snippet: |
      more_set_headers "Access-Control-Allow-Origin: $http_origin";
      location ~* /auth/realms/[^/]+/metrics {
        return 403;
```

```
}
rules:
  - host: "my.keycloak.local"
    paths:
      - path: /
        pathType: Prefix
tls:
  - hosts:
    - "my.keycloak.local"
      secretName: dev-wildcard
console:
  enabled: true
  ingressClassName: "nginx"
  annotations:
    ingress.kubernetes.io/ssl-redirect: "true"
    nginx.ingress.kubernetes.io/ssl-redirect: "true"
    nginx.ingress.kubernetes.io/proxy-body-size: "128k"
  rules:
    - host: "my.keycloak.local"
      paths:
        - path: /auth/admin/
          pathType: Prefix
  tls:
    - hosts:
      - "my.keycloak.local"
        secretName: dev-wildcard
```

install

```
helm repo add codecentric https://codecentric.github.io/helm-charts
helm repo update
helm upgrade --install keycloak codecentric/keycloak --values values.yaml
```

Keycloak будет доступен по адресу **<https://my.keycloak.local>**

Настройку сервиса **keycloak** см. в [Приложении 1](#).

Continuity

Helm chart в репозитории registry.datasapience.ru: **ds/continuity**.

values.yaml

```
helm show values --version 2.1.0 ds/continuity > values.yaml
```

Минимальная настройка values.yaml:

- **image:**
 - **service.image** - registry.datasapience.ru/russian-registry/continuity/continuity

- **service.image.tag** - 2.1.0
- **service.init.containers[0].image** - registry.datasapience.ru/russian-registry/continuity/continuity
- **service.init.containers[0].tag** - 2.1.0
- **service_celery.image** - registry.datasapience.ru/russian-registry/continuity/continuity
- **service_celery.tag** - 2.1.0
- **service.ingress:**
 - **host** - continuity
 - **baseDomain** - your-domain
 - **tls.secretName** - continuity
- **service.secrets** (меняет secret с названием **continuity-backend-secret**)
 - "CONTINUITY_K8S_NAMESPACE": "continuity-qa",
 - "CONTINUITY_ROOT_PATH": "/api",
 - "CONTINUITY_S3_BUCKET": "kolmogorov",
 - "CONTINUITY_S3_PASSWORD": "s3_secret_key",
 - "CONTINUITY_S3_PREFIX": "continuity",
 - "CONTINUITY_S3_URL": "https://storage.yandexcloud.net",
 - "CONTINUITY_S3_USERNAME": "s3_access_key",
 - "KEYCLOAK_ADMIN_PASSWORD": "admin_password",
 - "KEYCLOAK_ADMIN_USERNAME": "admin_username",
 - "KEYCLOAK_CLIENT_ID": "client",
 - "KEYCLOAK_CLIENT_SECRET": "client_secret",
 - "KEYCLOAK_REALM": "realm",
 - "KEYCLOAK_URL": "https://keycloak/auth",

install

```
helm install continuity ds/continuity --version 2.1.0 -f values.yaml
```

Continuity Frontend

Helm chart в репозитории registry.datasapience.ru:: **ds/continuity-ui**.

values.yaml

```
helm show values --version 2.1.0 ds/continuity-ui > values.yaml
```

Минимальная настройка values.yaml:

- **image:**
 - **service.image** - registry.datasapience.ru/russian-registry/continuity/continuity-ui

- **service.image.tag** - 2.1.0
- **service.ingress:**
 - **host** - continuity
 - **baseDomain** - your-domain
 - **tls.secretName** - continuity
- **service.secrets** (меняет secret с названием **continuity-ui-secret**):
 - **KEYCLOAK_URL** - адрес keycloak.
 - **KEYCLOAK_REALM** - keycloak realm.
 - **KEYCLOAK_CLIENT_ID** - keycloak realm client.
 - **KEYCLOAK_CLIENT_SECRET** - keycloak realm client secret.

install

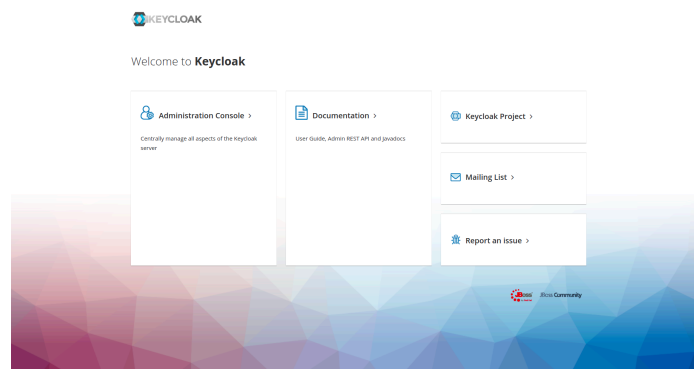
```
helm install continuity-ui ds/continuity-ui --version 2.1.0 -f values.yaml
```

Приложение 1. Настройка Keycloak

В данном приложении описана минимальная настройка необходимая для функционирования компонент ПО.

Создание realm

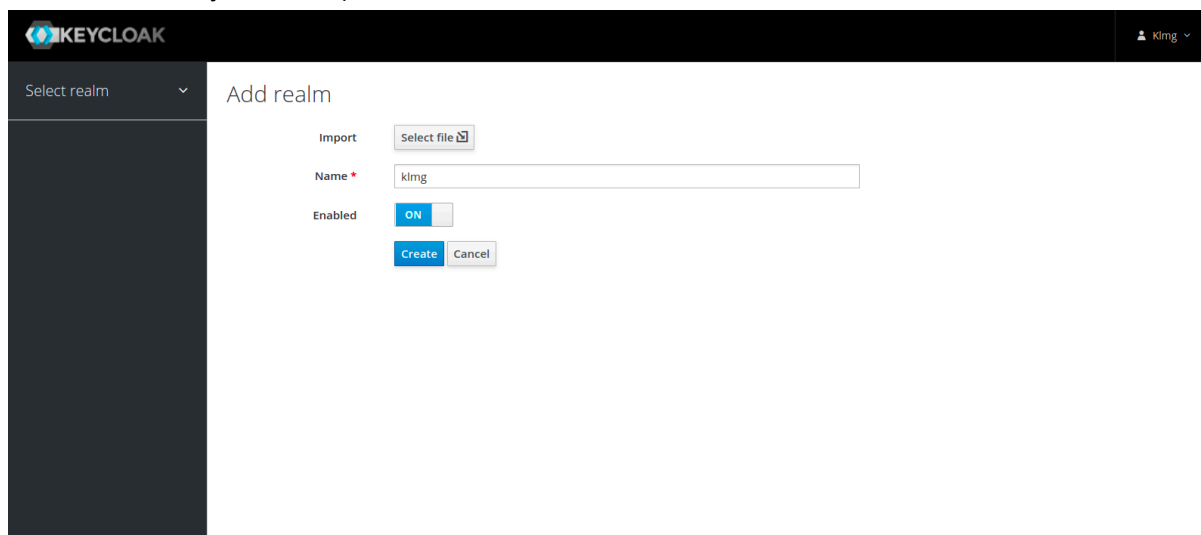
1. Перейти в консоль администратора Keycloak.



2. Нажать **Add realm**.



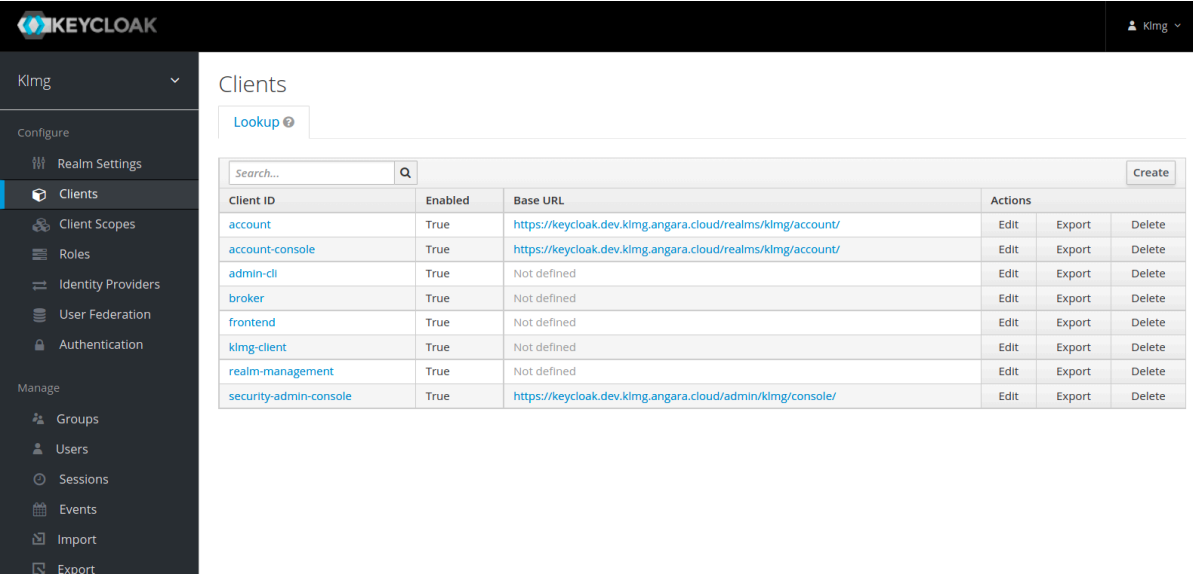
3. Ввести название (по дефолту название должно быть **kimg**, если вы используете другое название не забудьте поправить переменные в continuity-backend, continuity-frontend).



4. Нажать **Create**.

Создание client

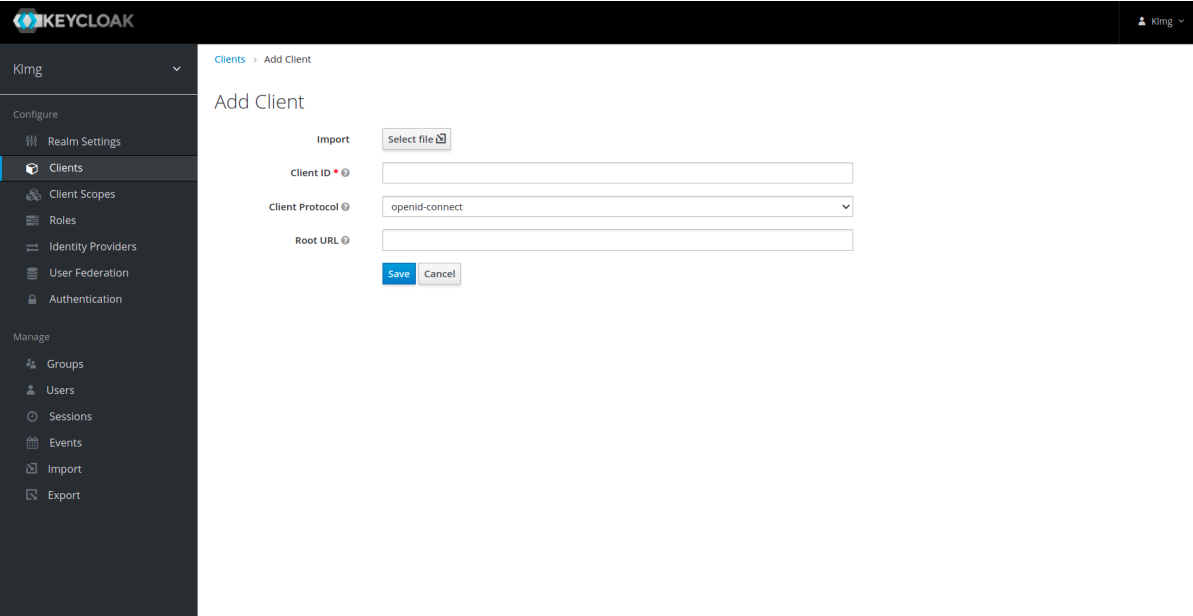
1. Зайти на вкладку **Clients**.



The screenshot shows the Keycloak Admin Console interface. The left sidebar contains navigation options under 'Configure' (Realm Settings, Clients, Client Scopes, Roles, Identity Providers, User Federation, Authentication) and 'Manage' (Groups, Users, Sessions, Events, Import, Export). The main content area is titled 'Clients' and features a search bar and a 'Create' button. Below is a table listing existing clients.

Client ID	Enabled	Base URL	Actions		
account	True	https://keycloak.dev.kimg.angara.cloud/realm/kimg/account/	Edit	Export	Delete
account-console	True	https://keycloak.dev.kimg.angara.cloud/realm/kimg/account/	Edit	Export	Delete
admin-cli	True	Not defined	Edit	Export	Delete
broker	True	Not defined	Edit	Export	Delete
frontend	True	Not defined	Edit	Export	Delete
kimg-client	True	Not defined	Edit	Export	Delete
realm-management	True	Not defined	Edit	Export	Delete
security-admin-console	True	https://keycloak.dev.kimg.angara.cloud/admin/kimg/console/	Edit	Export	Delete

2. Нажать кнопку **Create** (в правом верхнем углу).
3. Ввести **Client ID**. (по дефолту client_id должно быть **frontend**, если вы используете другое название не забудьте поправить переменные в continuity-backend, continuity-frontend).



The screenshot shows the 'Add Client' form in the Keycloak Admin Console. The form includes an 'Import' button with a file selection icon, a 'Client ID' input field, a 'Client Protocol' dropdown menu set to 'openid-connect', and a 'Root URL' input field. At the bottom of the form are 'Save' and 'Cancel' buttons.

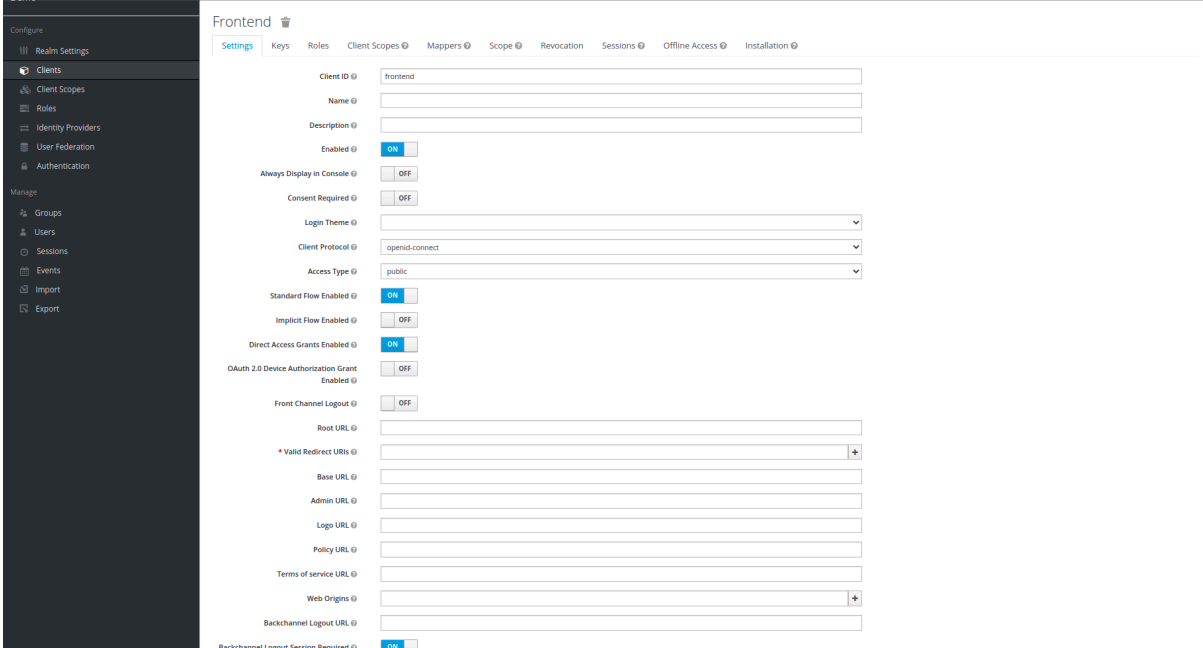
4. Нажать кнопку **Save**.

Настройка client

В списке client выбрать созданный client. Далее перейти на страницу конфигурации клиента.

В настройках необходимо заполнить два обязательных параметра.

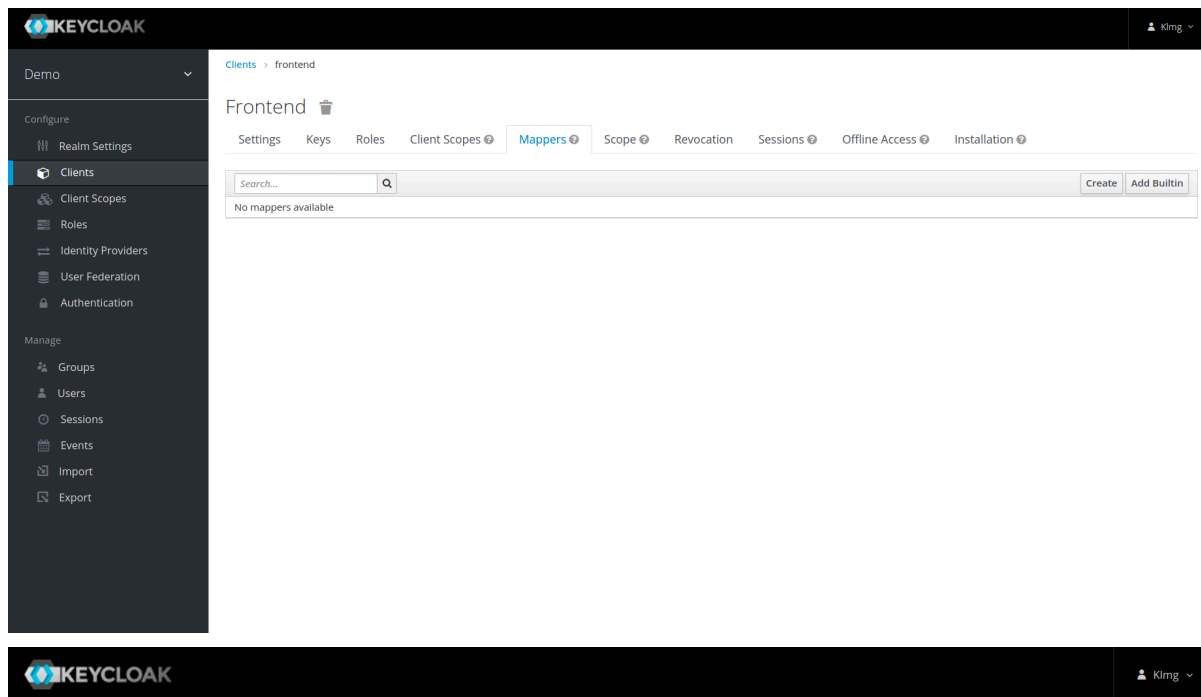
- **Valid Redirect URIs** - список адресов откуда возможно авторизация (здесь необходимо указать адрес Continuity Web, Continuity API) (**внимание: в конце каждого redirect url необходимо добавить /***)
- **Web Origins** - настройка cors. Для простоты конфигурации можно написать *



Настройка Mapper

На странице конфигурации client перейти на вкладку **Mappers**.

1. Нажать кнопку **Create**.
2. Создать новый mapper, как показано на втором скриншоте.
3. Нажать кнопку **Save**.



Создание ролей

1. Зайти на вкладку **Roles**.
2. Нажать кнопку **Add Role**.
3. Ввести значение **Role Name - continuity_admin**.

Добавление ролей пользователям

1. Зайти на вкладку **Users**.

ООО «Дата Сапиенс»

<https://kolmogorov.ai/>

ИНН 9701181979

ОГРН 1217700358083



2. Выбрать нужного пользователя.
3. На странице с настройками пользователя перейти на вкладку Role Mappings.
4. Добавить пользователю роль **continuity_admin**.