



KOLMOGOROV

# **Программное обеспечение «Kolmogorov.ai Continuity»**

**Руководство по установке**

## Оглавление

<b>Оглавление</b>	<b>1</b>
<b>Введение</b>	<b>2</b>
О Kolmogorov.ai	2
Назначение документа	2
<b>Системные требования к ПО «Kolmogorov.ai Continuity»</b>	<b>2</b>
<b>Архитектура и компоненты ПО</b>	<b>3</b>
Компоненты	3
Архитектура	3
<b>Пререквизиты</b>	<b>4</b>
<b>Установка компонент</b>	<b>4</b>
Registry	4
Список скриптов установки (helm-чартов)	5
<b>Keycloak</b>	<b>5</b>
values.yaml	5
install	7
Continuity	7
values.yaml	7
install	8
Continuity Frontend	8
values.yaml	8
install	9
<b>Приложение 1. Настройка Keycloak</b>	<b>9</b>
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}***

e-mail: [contact@kolmogorov.ai](mailto: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

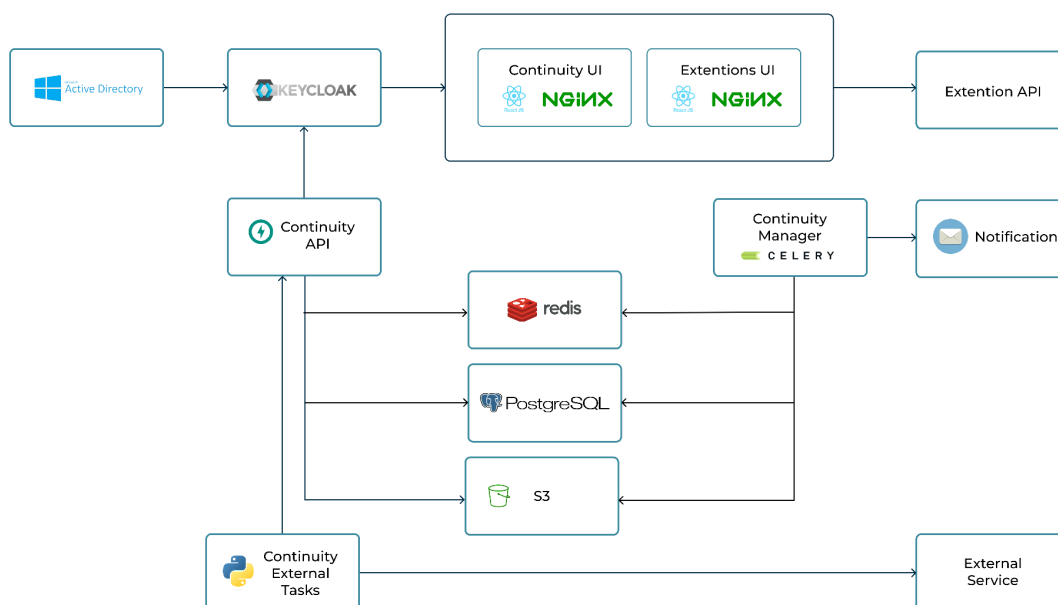
- **Keycloak.** Сервис аутентификации. Работает по принципу OAuth2.
- **PostgreSQL.** База данных для хранения метаданных.
- **Continuity Web.** Интерфейс для работы с системой из браузера.
- **Continuity API.** Интерфейс для работы с API системы.
- **Continuity Celery.** Обработчик background задач.

### 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

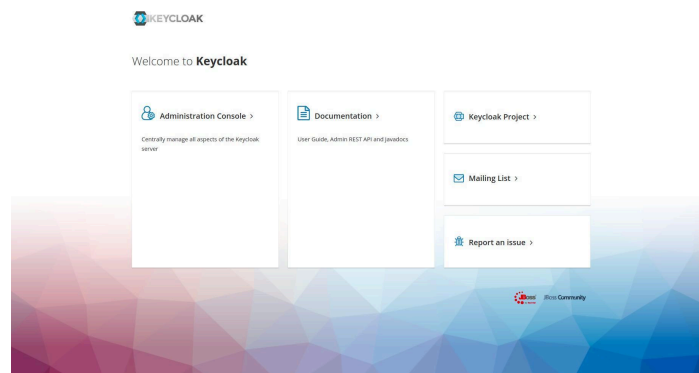
```
helm install continuity-ui ds/continuity-ui --version 2.1.0 -f values.yaml
```

## Приложение 1. Настройка Keycloak

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

### Создание realm

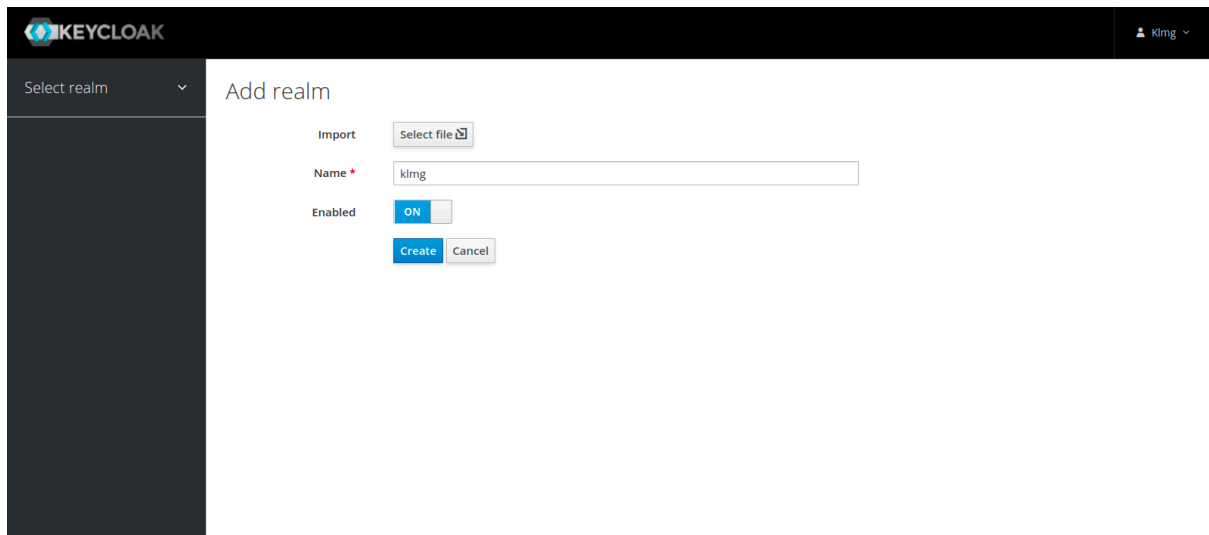
1. Перейти в консоль администратора Keycloak.



2. Нажать **Add realm**.



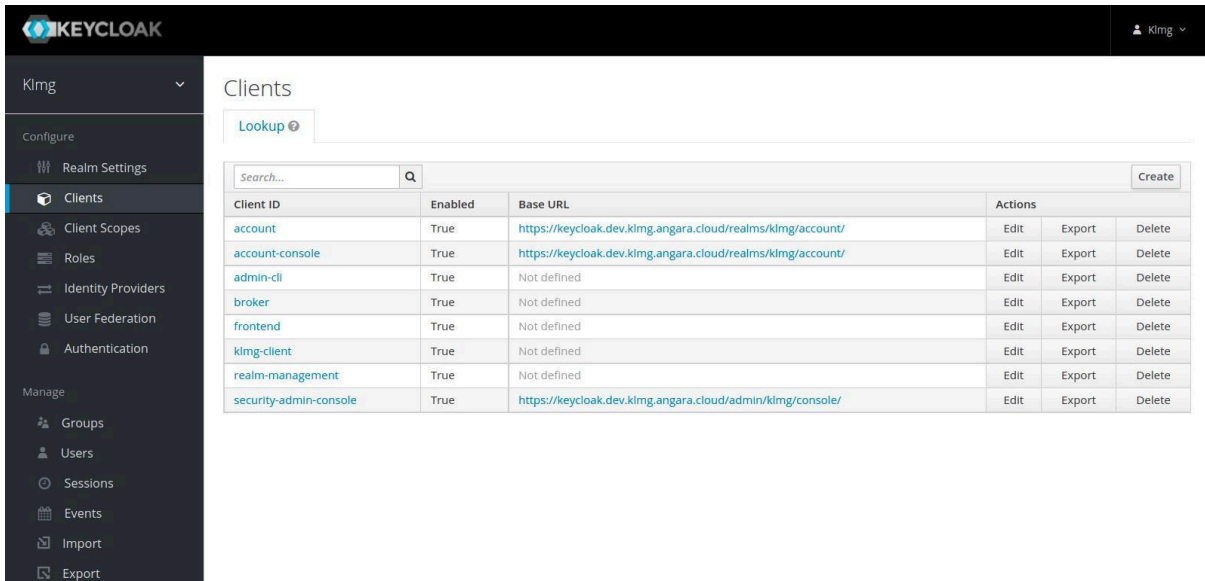
3. Ввести название (по дефолту название должно быть **kimg**, если вы используете другое название не забудьте поправить переменные в `continuity-backend`, `continuity-frontend`).



4. Нажать **Create**.

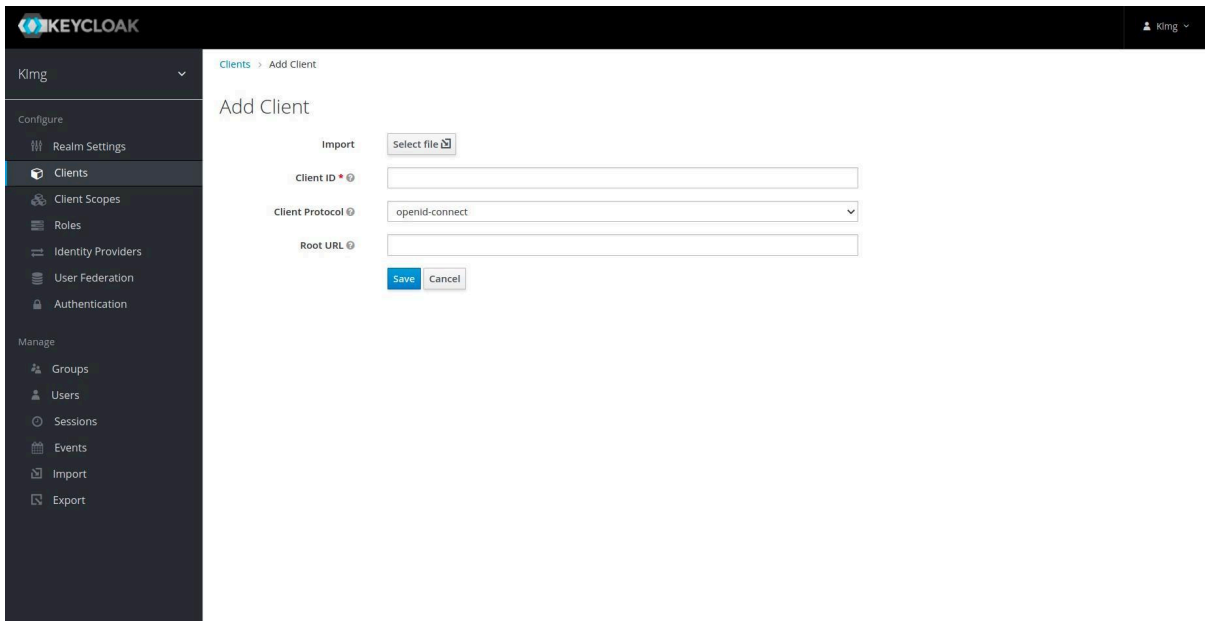
## Создание **client**

1. Зайти на вкладку **Clients**.



Client ID	Enabled	Base URL	Actions		
account	True	<a href="https://keycloak.dev.kimg.angara.cloud/realm/kimg/account/">https://keycloak.dev.kimg.angara.cloud/realm/kimg/account/</a>	Edit	Export	Delete
account-console	True	<a href="https://keycloak.dev.kimg.angara.cloud/realm/kimg/account/">https://keycloak.dev.kimg.angara.cloud/realm/kimg/account/</a>	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	<a href="https://keycloak.dev.kimg.angara.cloud/admin/kimg/console/">https://keycloak.dev.kimg.angara.cloud/admin/kimg/console/</a>	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).



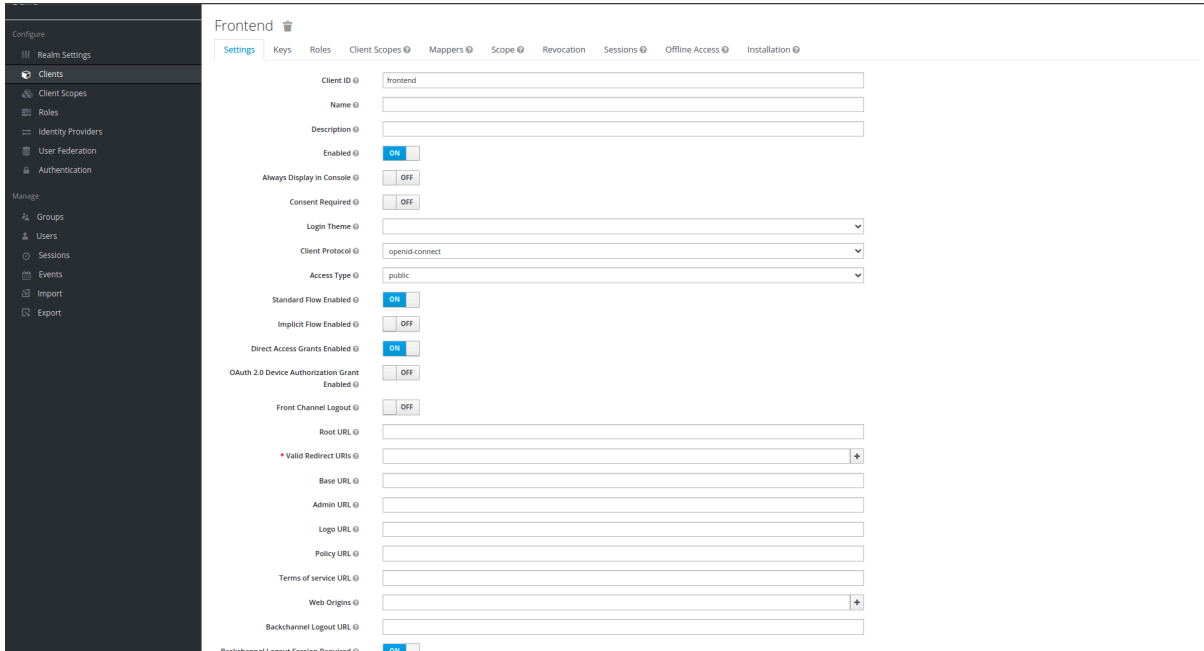
4. Нажать кнопку **Save**.

## 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 \*



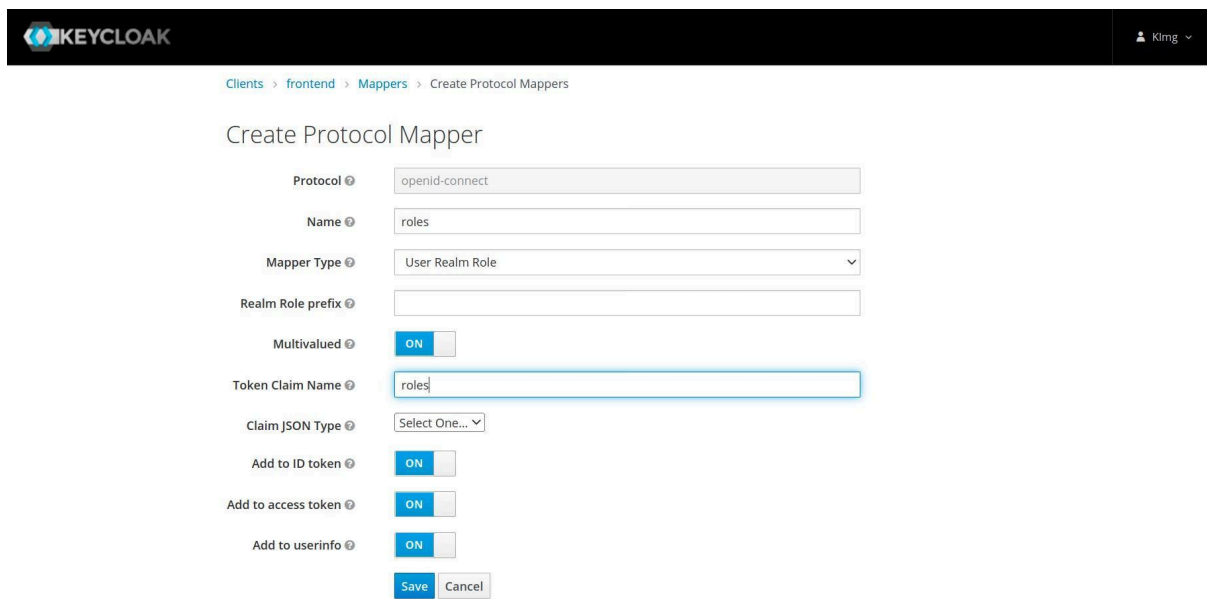
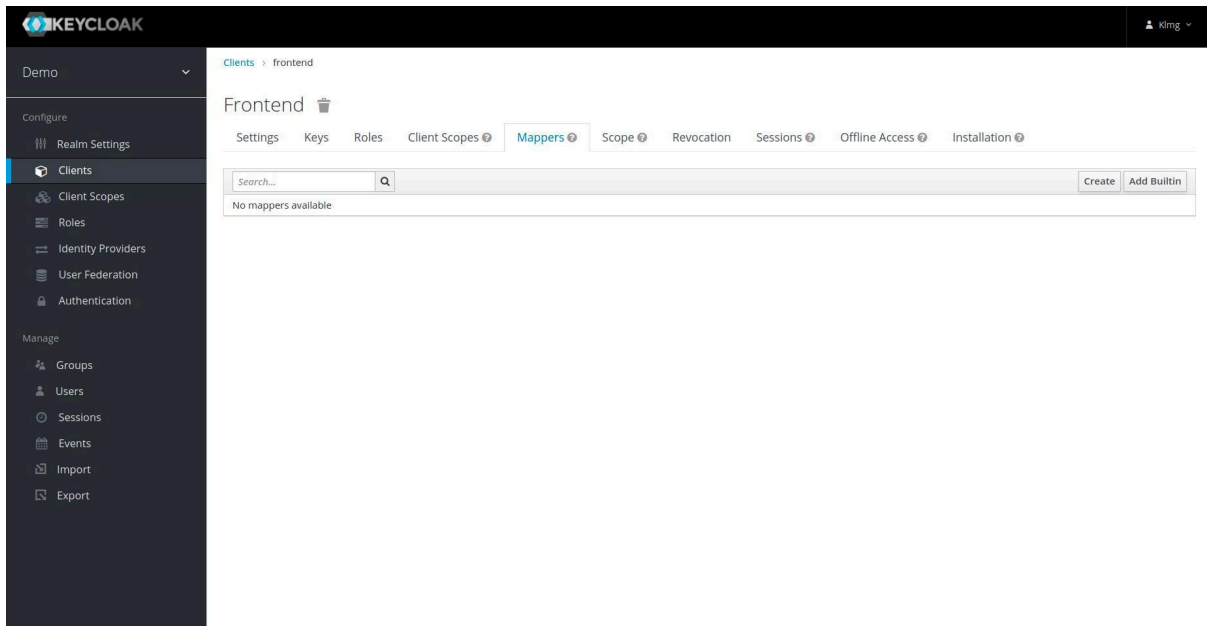
The screenshot shows the configuration page for a client named "Frontend" in Keycloak. The page is divided into several sections: "Settings", "Keys", "Roles", "Client Scopes", "Mappers", "Scope", "Revocation", "Sessions", "Offline Access", and "Installation". The "Settings" section is currently active and contains the following fields and options:

- Client ID: frontend
- Name: [empty]
- Description: [empty]
- Enabled: ON
- Always Display in Console: OFF
- Consent Required: OFF
- Login Theme: [dropdown]
- Client Protocol: openid-connect
- Access Type: public
- Standard Flow Enabled: ON
- Implicit Flow Enabled: OFF
- Direct Access Grants Enabled: ON
- OAuth 2.0 Device Authorization Grant Enabled: OFF
- Front Channel Logout: OFF
- Root URL: [empty]
- Valid Redirect URIs: [dropdown with asterisk]
- Base URL: [empty]
- Admin URL: [empty]
- Logo URL: [empty]
- Policy URL: [empty]
- Terms of service URL: [empty]
- Web Origins: [dropdown with asterisk]
- Backchannel Logout URL: [empty]
- Backchannel Logout Enabled: ON

## 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 Сapiенс»

<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**.