

Мессенджер и ВКС

Управление пользователями без контроллера домена

Оглавление

Назначение документа	3
Дополнительная документация	3
Управление пользователями через CLI-утилиту <code>users.py</code>	3
Управление пользователями через <code>kscli</code> -утилиту	6
Управление пользователями через веб-интерфейс сервиса Keycloak	8
Настройка доступа в окружение администратора	8

Назначение документа

В инструкции описано управления пользователями при отсутствии контроллера домена. Документ предназначен для использования администраторами организации.

Примечание

Ранее Мессенджер и ВКС назывался Myteam, что находит отражение в технических моментах (например, команды в консоли).

Дополнительная документация

[Инструкция по интеграции с контроллером домена по протоколу LDAP](#) — в документе описано управление параметрами синхронизации LDAP.

[Инструкция по установке на одну виртуальную машину /Инструкция по установке кластера](#) — в документах описана настройка доступа в окружение администратора в процессе установки Мессенджер и ВКС.

Управление пользователями через CLI-утилиту users.py

Чтобы добавить, удалить или обновить пользователей, минуя контроллер домена, можно использовать специальную CLI-утилиту users.py:

Команды

```
usage: users.py [-h] [-c CONFIG_FILE] --cmd COMMAND [-m MAX_USERS]
               [--users [N [N ...]]]
optional arguments:
  -h, --help            show this help message and exit
  -c CONFIG_FILE        Path to users list
  --cmd COMMAND         Command: add, update, list, delete, search
  -m MAX_USERS          Max users count for list and search
  --users [N [N ...]]  user's emails for search
```

- Показать список всех пользователей (в том числе и пользователей из контроллера домена):

```
/usr/local/bin/users.py --cmd list
```

Количество пользователей, отображаемых командой `list`, ограничивается параметром `-m`. По умолчанию выводится 100 пользователей. Вывод большого количества пользователей может занять длительное время, поэтому рекомендуется использовать команду `search`, которая позволяет выдавать список пользователей поиском по подстроке.

- Показать список всех пользователей (в том числе и пользователей из контроллера домена), у которых в адресе почты, фамилии или имени есть требуемая подстрока (в данном примере две подстроки `user01` и `user02`). Количество пользователей ограничивается параметром `-m`.

```
/usr/local/bin/users.py --cmd search --users user01 user02
```

- Добавить пользователей:

```
/usr/local/bin/users.py --cmd add -c ./users.yaml
```

Прежде чем отдавать команду на добавление пользователей, необходимо в любой удобной папке создать файл `users.yaml` и заполнить его данными пользователей (описание формата YAML-файла представлено ниже).

- Обновить пользователей:

```
/usr/local/bin/users.py --cmd update -c ./users.yaml
```

- Удалить пользователей:

```
/usr/local/bin/users.py --cmd delete -c ./users.yaml
```

или

```
/usr/local/bin/users.py --cmd delete --users user01@vkteams.example.com
```

- Вывод списка пользователей поиском по подстроке (количество пользователей ограничивается параметром `-m`):

```
// найти всех пользователей, у которых в полях login, lastName,  
// firstName присутствует слово Petrov  
users.py --cmd search --users Petrov.
```

Форматы файла `users.yaml`

При добавлении пользователей по умолчанию ищется файл `./users.yaml` (можно изменить параметром `-c`).

1. Основной формат файла `users.yaml` (объект `users` имеет тип `Array`):

```
users:  
- 'user1@mail.ru'  
- 'Surname user2@bk.ru'  
- 'Name Surname user3@list.ru'  
- 'Surname Name MiddleName user4@inbox.ru'
```

Варианты описания пользователя:

- email
- lastName email
- firstName lastName email
- lastName firstName middleName email

В базе пользователей нет понятия отчества, поэтому оно не обрабатывается. В варианте email (только адрес почты), lastName (фамилия) и firstName (имя) будут заполнены именем до знака @ в адресе почты.

Пример

```
users:  
- 'user1@icq.com'  
- 'Ivanov user2@icq.com'  
- 'Alexander Petrov user3@icq.com'  
- 'Alexander Ivanov Ivanovich user4@icq.com'
```

2. Расширенный формат файла users.yaml (объект users имеет тип Hash):

```
users:
  user01@vkteams.example.com:
    email: user01@vkteams.example.com #почта
    firstName: FirstName01 #имя
    lastName: LastName01 #фамилия
    enabled: false #заблокирован ли пользователь
    attributes:
      mobile: +000000000 #телефон
      department: Test Department #отдел
      title: Job title N1 #должность
      middleName: MiddleName01 #отчество
      memberOf: ["VIP1"] #член группы VIP1
  user02@vkteams.example.com:
    email: user02@vkteams.example.com
    firstName: FirstName02
    lastName: LastName02
    enabled: false
    attributes:
      mobile: +000000001
      department: Test Department
      title: Job title N2
      middleName: MiddleName02
      memberOf: ["VIP1"]
```

При использовании расширенного формата .yaml-файла, username должен совпадать с email. В примере выше это user01@vkteams.example.com.

Примечание

Возможно использование утилиты `users.py` без использования YAML-файла. Вместо этого можно использовать параметр `--users`, например:

```
users.py --cmd delete --users user01@vkteams.example.com user02@vkteams.example.com
users.py --cmd search --users user01@vkteams.example.com user02@vkteams.example.com
```

Обратите внимание, что при использовании команд `add` и `update` пользователи будут созданы без фамилии и имени (`lastName`, `firstName`).

Управление пользователями через ksccli-утилиту

Ниже представлены команды для управления пользователями при помощи ksccli-утилиты:

- Поиск информации по пользователю с использованием основных атрибутов пользователя:

```
ksccli user search <email> или <имя|фамилия>
```

- Поиск информации по пользователю только по почте:

```
kccli user get <email>
```

- Добавить пользователя:

```
kccli user add <email>
```

При добавлении пользователя из консоли пользователи будут созданы без фамилии, имени и отчества (lastName, firstName и middleName) — они будут назначены автоматически. Добавить пользователя с заданными фамилией, именем и отчеством возможно при помощи YAML-файла (см. описание ниже).

- Удалить пользователя:

```
kccli user delete <email>
```

- Добавить пользователя при помощи YAML-файла:

Предварительно создайте в любой удобной директории файл users.yaml и заполните его данными пользователей.

Расширенный формат файла users.yaml (объект users имеет тип Hash):

```
users:
  user01@vkteams.example.com:
    email: user01@vkteams.example.com #почта
    firstName: FirstName01 #имя
    lastName: LastName01 #фамилия
    enabled: false #заблокирован ли пользователь
    attributes:
      mobile: +000000000 #телефон
      department: Test Department #отдел
      title: Job title N1 #должность
      middleName: MiddleName01 #отчество
      memberOf: ["VIP1"] #член группы VIP1
  user02@vkteams.example.com:
    email: user02@vkteams.example.com
    firstName: FirstName02
    lastName: LastName02
    enabled: false
    attributes:
      mobile: +000000001
      department: Test Department
      title: Job title N2
      middleName: MiddleName02
      memberOf: ["VIP1"]
```

После создания users.yaml выполните в консоли команду:

```
kccli user add -f <имя файла>
```

- Удалить пользователя при помощи YAML-файла:

```
kccli user delete -f <имя YAML-файла>
```

Управление пользователями через веб-интерфейс сервиса Keycloak

Инструкции см. в разделе [Авторизация в системе](#).

Необходимо учесть, что веб-интерфейс не поддерживает batch-режим, поэтому каждым пользователем придется управлять отдельно, что неудобно при обработке большого количества пользователей.

Настройка доступа в окружение администратора

Доступ в окружение администратора настраивается при установке системы и описан в документации по установке:

- [Установка на одну виртуальную машину](#).
- [Установка кластера](#).

Доступ в окружение администратора проверяется пересечением атрибута **memberOf** в файле `users.yaml` с перечнем групп, указанных в секции **otp_permission** конфигурационного файла инсталляции `/usr/local/etc/premsetup/defaults.yaml`.

Добавить этот атрибут к существующему пользователю можно при помощи команды `--cmd update`, с использованием расширенного формата `users.yaml`.

```
users:
  user01@vkteams.example.com:
    attributes:
      memberOf: [ 'myteam-admin' ]
```

 Технический писатель: Белова Ирина

 12 ноября 2025 г.