API смены учетных данных пользователя

Оглавление

API предназначено для выполнения сценария смены учетных данных пользователя (логин, пароль) самим пользователем.

Данное API используется из мобильного или SPA-приложения и доступно для использования через Интернет.

Для смены учетных данных другой системой через доверенное межсерверное обращение используйте API управления пользователями.

Сценарий смены учетных данных

Предусловия

- Пользователь аутентифицирован в системе
- Пользователь открыл Web- или мобильное приложение (далее называется как "Приложение")

Сценарий

1. Пользователь нажимает на ссылку "сменить пароль" в Приложении
   a. Приложение отправляет запрос на старт сценария смены учетных данных и получение execution
   b. Сервер отвечает описанием формы смены учетных данных.
   c. Приложение отображает форму смены учетных данных
2. Пользователь вводит данные в форму: текущий пароль, новый логин, новый пароль
   a. Приложение проверяет введенные данные
   b. Приложение отправляет запрос на смену учетных данных
   c. Сервер в зависимости от настроек проверяет выполнение парольных политик
   d. Сервер проверяет необходимость подтверждения операции с помощью 2FA в зависимости от настроек.
      i. Если требуется подтверждение вторым фактором, сервер запускает вложенный сценария проверки ОТП (в данном документе не описан), после успешного прохождения проверки продолжает основной сценарий.
   e. Сервер меняет учетные данные пользователя
   f. Сервер находит и прерывает все сессии данного пользователя, кроме текущей.
   g. Сервер отвечает подтверждением успешности смены данных

Постусловия

- Установлен новый пароль в БД UIDM, таблица Credentials (только при использовании собственной БД UIDM)
- Установлен новый пароль во внешнем хранилище учетных записей (только при использовании внешнего хранилища учетных записей)
- В БД Аудита запротоколировано событие `sso.credentials_change.success`
- Все другие сессии данного пользователя, кроме текущей, прерваны

Замечания по работе сценария

1. Все запросы должны быть выполнены в приведенной последовательности, так как параметр `execution` из каждого ответа используется как входной параметр в последующих запросах.

ВАЖНО

Возвращаемое значение ‹execution› в каждом из запросов к UIDM может обновляться. Необходимо использовать самое актуальное значение.

Старт сценария

Для начала сценария создания привязки необходимо отправить запрос в UIDM на `/sso/oauth2/access_token`. В ответе будет содержаться параметр ‹execution›, который необходимо включить в следующий запрос к UIDM.

Так же в ответе содержится описание формы ввода идентификатора пользователя, который восстанавливает пароль.

Формат запроса

```
POST /sso/auth/change-credentials

Host: <sso_host>
Accept: application/json
Content-Type: application/x-www-form-urlencoded

client_id=<client_id>&
access_token=<access_token>
```

Параметры

- ‹sso_host› - базовый адрес сервера UIDM, например sso.rooxteam.com
- ‹client_id› - идентификатор клиента, от имени которого выполняется смена учетных данных, например selfcare

Формат успешного ответа

В ответе содержится JSON объект, содержащий описание формы, а так же ‹execution›, который необходимо включить в следующий запрос к UIDM.

```
HTTP/1.1 200 OK

Content-Type: application/json;charset=UTF-8
```

```
{
  "execution": <execution_value>,
```

```
  "view": {
    "username": <username>
  },
  "form": {
    "name": "credentialsForm",
    "fields": {
      "password": {
        "constraints": [
          {
            "name": "ConfigurablePattern",
            "value": <formParamValue>
          },
          {
            "name": "ConfigurableMaxSize"
          },
          {
            "name": "ConfigurableMinSize"
          }
        ]
      },
      "newUsername": {
        "constraints": [
          {
            "name": "ConfigurableMaxSize"
          },
          {
            "name": "ConfigurablePattern"
          },
          {
            "name": "ConfigurableMinSize"
          }
        ]
      },
      "newPasswordBody": {
        "constraints": [
          {
            "name": "ConfigurableMaxSize"
          },
          {
            "name": "ConfigurableMinSize"
          },
          {
            "name": "ConfigurablePattern"
          }
        ]
      }
    },
    "errors": []
  },
  "serverUrl": <ignore>,
  "step": "enter_credentials"
}
```

Поля ответа

- execution_value - значение параметра ‹execution› для следующего запроса к UIDM
- описание формы может содержать в себе описание различных полей и накладываемых на них ограничений. Если ограничение не накладывается, параметры name и value непосредственно определяют набор ограничений. Если ограничение не задано, параметр value может отсутствовать
- step - обозначение текущего шага сценария, в данном случае отображается форма ввода учетных данных

Состав формы говорит, что следует отобразить форму ввода учетных данных и передать полученные данные в следующем запросе

Передача новых учетных данных

UI отображает форму ввода учетных данных.

Пользователь вводит данные, после чего приложение отправляет запрос на сервер.

Формат запроса

```
POST /sso/auth/change-credentials

Host: <sso_host>
Accept: application/json
Content-Type: application/x-www-form-urlencoded


execution=<execution>&
_eventId=next&
password=<oldPassword>&
newPasswordBody=<newPassword>>&
username=<username>
```

Параметры запроса

- execution - значение равно значению поля ‹execution› полученному из ответа на предыдущий запрос к API
- oldPassworde - предыдущий пароль пользователя
- newPassword - новый пароль
- username - имя пользователя. Может содержать новое имя пользователя, если необходимо его поменять.

Формат успешного ответа

```
HTTP/1.1 200 OK

Content-Type: application/json;charset=UTF-8
```

```
{
  "step": "redirect",
  "location": "/sso/auth/complete"
}
```