

# API изменения пользовательских настроек одноразового пароля

## Оглавление

- # Авторизация
  - # Авторизация через системный токен
  - # Авторизация через пользовательский токен
- # Поддерживаемые имена параметров
- # Получение значений параметров
  - # Получение единственного значения
  - # Получение всех доступных значений
- # Установка единственного значения параметра
  - # Запрос изменения или установки значения параметра
  - # Запрос сброса настройки в значение по умолчанию
- # Установка нескольких параметров одним запросом
- # Пример положительного ответа
- # Пример отрицательного ответа
- # Специальные сценарии
  - # Включение уведомлений для пользователя при выключении второго фактора (и наоборот)

API управления конфигурацией одноразового пароля (OTP) предназначен для настройки пользователем параметров запроса OTP-кодов.

### ВАЖНО

Все методы сервиса поддерживают @me-нотацию для ID пользователя. При этом в качестве ID будет использоваться имя пользователя из переданного для авторизации токена.

## # Авторизация

Сервис поддерживает авторизацию двумя способами: через системный токен и через пользовательский (только редактирование собственного пользователя @me).

### Авторизация через системный токен

Другие сервисы могут авторизоваться через системный токен SSO. Такой токен позволяет редактировать все записи сервиса.

### ВАЖНО

Наличие пользователя с запрошенным ID не проверяется. Отсутствие валидного пользователя в БД UIDM является допустимым, поэтому предоставление корректного значения — задача вызывающей стороны.

Пример работы с системным токеном:

```
GET /sso/api/settings/{principalId}/otp/{settingName}  
Accept: application/json  
Authorization: Bearer sso_1.0_{token}
```

- principalId — идентификатор пользователя, для которого запрашивается настройка
- settingName — имя настройки
- token — системный токен SSO

## Авторизация через пользовательский токен

Виджеты могут авторизоваться с помощью пользовательского токена. При такой авторизации для работы будет доступен только псевдо-ID пользователя @me, дающий доступ к редактированию только данных пользователя, которому был выдан этот токен.

```
GET /sso/api/settings/@me/otp/{settingName}  
Accept: application/json  
Authorization: Bearer sso_1.0_{token}
```

- settingName — имя настройки
- token — токен пользователя

## # Поддерживаемые имена параметров

UIDM поддерживает следующие настройки одноразового пароля (OTP):

- otp.social.mapping.login.enabled — включен ли одноразовый пароль (OTP) для аутентификации с использованием социальной сети;
- otp.social.mapping.attach.enabled — включен ли одноразовый пароль (OTP) для создания связи с учётной записью социальной сети.
- otp.social.mapping.reattach.enabled — включен ли одноразовый пароль (OTP) для изменения связи с учётной записью социальной сети.
- otp.login.enabled — включен ли одноразовый пароль (OTP) для аутентификации по логину и паролю;
- otp.action.enabled — включен ли одноразовый пароль (OTP) под операцию.

## # Получение значений параметров

Предоставляется возможность запросить как значение единственного параметра, так и значение всех параметров для заданного пользователя.

**ВАЖНО**

Для отсутствующих параметров будет возвращено значение по умолчанию.

## Получение единственного значения

```
GET /sso/api/settings/{principalId}/otp/{settingName}  
Accept: application/json  
Authorization: Bearer sso_1.0_{token}
```

- principalId — идентификатор пользователя, для которого запрашивается настройка
- settingName — имя настройки

### Пример ответа:

```
HTTP/1.1 200 OK  
Content-Type: application/json  
true
```

- settingName — имя настройки

## Получение всех доступных значений

```
GET /sso/api/settings/{principalId}/otp  
Accept: application/json  
Authorization: Bearer sso_1.0_{token}
```

- principalId — идентификатор пользователя, для которого запрашиваются настройки

### Пример ответа:

```
HTTP/1.1 200 OK  
Content-Type: application/json
```

```
{  
  "{settingName1}": true,  
  "{settingName2}": false,  
  "{settingName3}": true  
}
```

- settingNameX — имена настроек

## # Установка единственного значения параметра

Для управления единичным значением параметра следует использовать следующие запросы:

### Запрос изменения или установки значения параметра

```
PUT /sso/api/settings/{principalId}/otp/{settingName}  
Accept: application/json  
Authorization: Bearer sso_1.0_{token}  
Content-Type: application/json  
  
true
```

- principalId — идентификатор пользователя, для которого выставляется настройка
- settingName — имя настройки

В качестве значения передается выражение типа boolean.

### Запрос сброса настройки в значение по умолчанию

```
DELETE /sso/api/settings/{principalId}/otp/{settingName}  
Accept: application/json  
Authorization: Bearer sso_1.0_{token}
```

- principalId — идентификатор пользователя, для которого выставляется настройка
- settingName — имя настройки

## # Установка нескольких параметров одним запросом

Для управления несколькими параметрами за один запрос необходимо отправить запрос в формате JSON Patch:

```
PATCH /sso/api/settings/{principalId}/otp  
Accept: application/json  
Authorization: Bearer sso_1.0_{token}  
Content-Type: application/json-patch+json
```

```
[  
  {  
    "op": "add",  
    "path": "/{settingName1}",  
    "value": true  
  },  
  {  
    "op": "replace",
```

```
[{"path": "/{settingName2}",
  "value": false
},
{
  "op": "remove",
  "path": "/{settingName3}"
}]
```

- principalId — идентификатор пользователя, для которого выставляется настройка
- settingNameX — имя настройки

## # Пример положительного ответа

При корректной обработке сервис вернет ответ с кодом 204.

```
HTTP/1.1 204 No Content
```

## # Пример отрицательного ответа

При возникновении ошибки в ходе работы сервиса, а также в случае передачи неверных параметров, сервис вернет соответствующий код ошибки с пояснением, например:

```
HTTP/1.1 404 Not Found
Content-Type: application/json
```

```
{
  "error": {
    "code": 404,
    "message": "Principal not found"
  }
}
```

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
```

```
{
  "error": {
    "code": 400,
    "message": "Unexpected operation 'move' supplied in JSON Patch"
  }
}
```

```
}  
  
HTTP/1.1 500 Internal Server Error  
Content-Type: application/json
```

```
{  
  "error": {  
    "code": 500,  
    "message": "Service was unable to handle the request"  
  }  
}
```

## # Специальные сценарии

### Включение уведомлений для пользователя при выключении второго фактора (и наоборот)

Специальный сценарий при котором пользователю включаются уведомления о входе через SMS при выключении второго фактора при входе. А также наоборот — выключение уведомлений при включении двухфакторной аутентификации.

Данный сценарий выполняется при использовании метода

```
PUT /sso/api/settings/{principalId}/otp/otp.login.enabled
```

#### ВАЖНО

В целях безопасности, использование методов DELETE и PATCH должно быть заблокировано через настройки Gateway (haproxy)

Технически сценарий делает следующее: при смене принципалу настройки `otp.login.enabled`, устанавливает ему настройку `com.rooxteam.event-notifier.default.enabled` (таблица PrincipalProperty) противоположную той что устанавливается в `otp.login.enabled` (таблица OtpPrincipalConfiguration).

Для включения сценария, воспользуйтесь следующей конфигурацией:

```
# описание: Включает нотификацию когда выключается для для пользователя второй фактор.  
# тип: Boolean  
# значение по-умолчанию: false  
com.rooxteam.sso.enable_notification_when_otp_disabled=true
```

