API управления пользователями

Оглавление

Используется для управления учетными записями, ролевой моделью, блокировками от лица серверной системы или автоматизационных скриптов.

Примеры

- создание нового пользователя по сигналу из ESB
- управление пользователями из APM (когда APM реализовано как серверное приложение, а не SPA)
- синхронизация пользователей между серверными системами (в сторону "к-RooX UIDM")
- массовое заведение пользователей скриптом при первоначальном внедрении

ПРЕДУПРЕЖДАЕМ

API не предназначено для использования с клиентских устройств и не должно быть доступно через Интернет.

Конечная точка

```
POST https://{sso_host}/sso/provision/principals
```

Обозначения и общие требования

- {{sso_host}} - базовый адрес сервера RooX UIDM, например https://sso.rooxteam.com
- В случае успешного выполения запроса, HTTP статус ответа будет 20X, в случае проблем, код статуса будет - 4XX-5XX, в теле ответа будет описание ошибки.
- При запросах к API ошибки со статусом 503 всегда приходят в виде HTML.
- При выполнении запроса требуется авторизация. Могут использоваться токены доступа, полученные с помощью OAuth2 Client Credentials или Basic авторизация.

Методы API

Создание учетной записи

Пример запроса

ВАЖНО

В некоторых версиях продукта **msisdn** и **login** в теле JSON запроса, являются обязательными полями.

```
POST /sso/provision/principals
Host: {{sso_host}}
Content-Type: application/json
Accept: application/json
```

```
{
  "externalId": "123",
  "msisdn": "9211234567",
  "fd": "2015-02-18T12:00:00.000+00:00",
  "person": {
    "firstNameNat": "John",
    "lastNameNat": "Doe",
    "patronymicNameNat": "Alex",
    "displayNameNat": "John Alex Doe",
    "genericRelations": [
      {
        "target": {
          "@c": ".Contact",
          "contactType": "email",
          "address": "example@example.com"
        }
      },
      {
        "target": {
          "@c": ".Contact",
          "contactType": "phone",
          "address": "9211234567"
        }
      }
    ]
  },
  "credentials": [
    {
      "login": "9211234567",
      "password": "b59c67bf196a4758191e42f76670ceba"
    }
  ],
  "extendedAttributes": {
    "IMEI": "12345678901234567",
    "IMSI": "123456789012345",
    "ICCID": "1234567890",
    "externalFd": "2015-02-18T12:00:00.000+00:00",
    "baseServiceBlocked": true,
    "allowRobots": true
  },
  "blocked": true,
```

```
  "blockedTo": "2015-02-18T12:00:00.000+00:00",
  "blockedReasonId": "1",
  "networkAuthenticationType": "AUTO"
}
```

- externalId - (опционально) уникальный идентификатор пользователя во внешней системе. Если передано, ID пользователя в сервере RooX UIDM будет создан на основе полученного значения. Если поле отсутствует, уникальный идентификатор будет сгенерирован сервером sso
- msisdn - (опционально) строковое поле с номером телефона длиной 10 символов, только цифры
- fd - (опционально) дата и время изменения профиля во внешней системе в формате UTC (в формате дата-время ISO 8601)
- person - (опционально) описание физ.лица/пользователя
  - firstNameNat - (опционально) симя пользователя, строковое значение, максимальная длина 255 символов
  - lastNameNat - (опционально) сфамилия пользователя, строковое значение, максимальная длина 255 символов
  - patronymicNameNat - (опционально) сотчество пользователя, строковое значение, максимальная длина 255 символов
  - displayNameNat - (опционально) сполное имя пользователя, если нет отдельных полей, строковое значение, максимальная длина 255 символов
- genericRelations/target/@c: .Contact - (опционально) структура и массив для задания контактных данных пользователя
  - contactTyoe - (обязательно) тип контакта. Допустимые значения:
    - `email` для адреса электронной почты
    - `phone` для номера телефона
  - address - (обязательно). Значение адреса (email или номер телефона), в зависимости от contactType, строка не более 1000 символов
- credentials - (обязательно) структура с описанием учетных данных
  - login - (обязательно) строковое поле с логином пользователя. В некоторых версиях продукта должен совпадать с msisdn
  - password - хеш пароля пользователя, строковое значение. Алгоритм хеширования может указываться с помощью префикса, непосредственно предшествующего значению хеша. В зависимости от версии продукта поддерживаются следующие префиксы:
    - `{md5}` - хеш md5. Является вариантом по умолчанию, если префикс не указан.
    - `{srp6a}` - хеш и алгоритм srp6a
    - `{bcrypt}` - хеш bcrypt
    - `{resetrequired}` - специальное значение, после которого не требуется передачи хеш пароля. Указывает на то, что пароль не установлен, вход по логину-паролю невозможен и пользователь должен пройти процедуру сброса/восстановления пароля.
- extendedAttributes - (опционально). Дополнительные атрибуты пользователя, могут передаваться произвольные атрибуты с общим объемом данных не более 2000 символов. Существует договоренность об использовании следующих специальных атрибутов, все являются опциональными:
  - IMEI - идентификатор мобильного оборудования, строка не более 20 симолов
  - IMSI - идентификатор мобильного пользователя, строка не более 20 симолов
  - ICCID - индентификатор SIM-карты, строка не более 20 симолов
  - externalFd - (**устаревший**) дата и время изменения профиля во внешней системе в формате UTC (в формате дата-время ISO 8601)

- baseServiceBlocked - признак блокировки базовой услуги (true - базовая услуга заблокирована, false - разблокирована)
- allowRobots - разрешение на вход роботам (true - вход роботам разрешен, false - запрещен)

- blocked - (опционально) признак блокировки (true - учетная запись заблокирована, false - разблокирована)
- blockedTo - (опционально) дата, время в формате UTC определяет срок окончания блокировки. Если передать дату пустой ("") или null, то учетная запись будет заблокирована бессрочно до явной разблокировки
- blockedReasonId - (опционально) код причины блокировки (в рамках проекта должен быть согласовано свой справочник кодов)
- networkAuthenticationType - (опционально, поддерживается только в версиях продукта для мобильных операторов) разрешение автоматического логина на основании данных мобильной сети (AUTO - автологин разрешен, NONE - автологин запрещен)

ВАЖНО

Параметр "externalFd" является устаревшим, вместо него следует использовать параметр fd. Использование обоих параметров одновременно недопустимо.

ВАЖНО

У одного пользователя не может быть два объекта Contact с одинаковым contactType

Электронную почту и дополнительный номер телефона пользователя можно добавить с помощью поля genericRelations объекта person.

genericRelations представляет из себя список данных с указанием типа через параметр @c.

В случае успешного создания учетной записи, возвращается пустой ответ с HTTP статусом 201. В зависимости от версии продукта ответ может задержать дополнительно следующие HTTP-заголовки: `Location` - содержит ID созданной учетной записи в формате `/sso/provision/principals/<присвоенный ID>` `X-Login-Location` - URL, при переходе на который произойдет автоматическая аутентификация пользлвателя без дополнительного запроса учетных данных.

Пример ответа:

```
HTTP/1.1 201 Created
Location: /sso/provision/principals/sso_____89ba7445-5c3c-4d4a-8ca8-2166ad14756a
X-Login-Location: https://somedomain.com/sso/UI/Login?
org=customer&service=uidm&ForceAuth=true&goto=https%3A%2F%2Fsomedomain.com%2Fsso%2Foauth2%
2Fauthorize%3Fredirect_uri%3Dhttps%3A%2F%2Focb-msb.demo.rooxteam.com%2Foauth2-
consumer%2Fauthorize%26realm%3D%2Fcustomer%26response_type%3Dcode%26client_id%3Dsmeportal%
26state%3Dgoto%253Dhttps%253A%252F%252Focb-
msb.demo.rooxteam.com%252Fapp&iPlanetAuthToken=eyJlbmMiOiJBMTI4R0NNIiwiYWxnIjoiUlNBLU9BRVA
ifQ.bxH_gJtLddod33_dFQZpkp2j1qA3J-
2TTjphHEnG62qPfXkQb9xtIIctE3JetagMbBA1g_jWpP2jQ7_y9oqzsg.4Ye68JOCoPCjp7Djj6HojorptrrwoJWh.
SxLTcDOHh4aVfy2SHe_MxxdOLaG-
pt63yiEMnHO39nDOmCef_CmyveIPyfpT7esKMfaF3LddPPzGEnLk67sgAbs9Ooltl5wRAfzDT7k6kNUZ_hif9xU1HZ
FgHIV6vZEeOsoXW4VOnj8PQ0faoYjfZyobU7Xa_hoXTXzfgrqLU3D7QxfkdCDUqn4OCqfDPYsvr0EDhsEuNoyggyvw
V--qk_U44iRxjyvBP1eTaDmVbzuMN9-AOPz-Nmy2cLVQcXkVnS7AvatSr6nwfJ-
F7uVnwT1uWZHsiOhHoFvOh74Nqdxlk6ZtRluipz6AeiIIQ-2H-
1Ul2JFpLnThN7OY8nJjOOpogwQCk1FKBNcTvyyXKvFvvmDlOw2_0pnGt9568jzoPVrIQNFNypzY2AIIX7F3FxQveyL
It_XEne70HyZoxCYt3e-0L-
9_7fW6uBHV2opcbtSY1hQbaOBs_Q0BkZOhoiqXcVrj_BI7JFAX7tQl54K27gTifV4Kt9aPlxiIGq-rtTbJURc2f-
r5LHB8X80OavSDcmU.8lQ1OZAop1YGhWeaATajOQ
Content-Length: 0
```

Изменение учетной записи

Изменение учетной записи происходит с помощью запроса HTTP PATCH. Для идентификации учетной записи в URL операции можно использовать один из следующих вариантов: * `uid` - уникальный идентификатор пользователя в сервере RooX UIDM * `msisdn` - (только в версиях продукта для телеком) номер телефона пользователя * `msisdn, externalId` - (только в версиях продукта для телеком) номер телефона пользователя и уникальный идентификатор пользователя во внешней системе. Семантически эквивалентен идентификации записи по номеру телефона.

ВАЖНО

Параметр "externalFd" является устаревшим, вместо него следует использовать параметр fd. Использование обоих параметров одновременно недопустимо.

Тело запроса должно содержать список операций по изменению согласно RFC 6902:

> The "add" operation performs one of the following functions,
> depending upon what the target location references:
> o If the target location specifies an array index, a new value is
> inserted into the array at the specified index.
> o If the target location specifies an object member that does not
> already exist, a new member is added to the object.
> o If the target location specifies an object member that does exist,
> that member's value is replaced.
> The operation object MUST contain a "value" member whose content
> specifies the value to be added.
> For example:
> { "op": "add", "path": "/a/b/c", "value": ["foo", "bar"] }

— RFC6902 пункт 4.1.

> The "remove" operation removes the value at the target location.
> The target location MUST exist for the operation to be successful.
> For example:
> { "op": "remove", "path": "/a/b/c" }
> If removing an element from an array, any elements above the
> specified index are shifted one position to the left.

— RFC6902 пункт 4.2.

> The "replace" operation replaces the value at the target location
> with a new value. The operation object MUST contain a "value" member
> whose content specifies the replacement value.
> The target location MUST exist for the operation to be successful.
> For example:
> { "op": "replace", "path": "/a/b/c", "value": 42 }
> This operation is functionally identical to a "remove" operation for
> a value, followed immediately by an "add" operation at the same
> location with the replacement value.

— RFC6902 пункт 4.3.

В случае успешного изменения учетной записи, возвращается пустой ответ с HTTP статусом 204

Изменение объектов principal, person и credentials

Примеры запросов

1. Идентификация учетной записи по msisdn и externalId

```
PATCH /sso/provision/principals?msisdn=9211234567&externalId=123
Host: {{sso_host}}
Content-Type: application/json-patch+json
Accept: application/json
```

```
[
  {
    "op": "remove",
    "path": "/path/to/field"
  },
  {
    "op": "add",
    "path": "/path/to/field",
    "value": [
      "foo",
      "bar"
    ]
  },
  {
    "op": "replace",
    "path": "/path/to/field",
    "value": 42
  }
]
```

- msisdn - строковое поле с номером телефона длиной 10 символов, только цифры
- /path/to/field - это путь до поля, через символ /, для выбора элемента массива указывается его индекс начиная с 0 (например: /person/firstNameNat, /credentials/0/password, /extendedAttributes/IMEI)
- externalId - уникальный идентификатор пользователя во внешней системе

1. Идентификация учетной записи по uid, смена хеша пароля

```
PATCH /sso/provision/principals?uid=sso_____21429dc3-a63d-48f0-8e55-c29d44390aaf
Host: {{sso_host}}
Accept: application/json
Content-Type: application/json-patch+json
```

```
[
  {
    "op": "replace",
    "path": "/credentials/0/password",
    "value": "{bcrypt}$2a$10$BJR5oTGKQuekpqxl62PjfupVv6vY8cK3IX1MA.zeBDQisgXBWVl1q"
  }
]
```

- uid - уникальный идентификатор пользователя на сервере RooX UIDM

Изменение контактных данных (объект Contact)

Пример запроса

```
PATCH /sso/provision/contacts?msisdn=9211234567&principal.externalId=123&contactType=email
Host: {{sso_host}}
Content-Type: application/json-patch+json
Accept: application/json
```

```
[
  {
    "op": "remove",
    "path": "/address"
  },
  {
    "op": "add",
    "path": "/address",
    "value": "example@example.com"
  },
  {
    "op": "replace",
    "path": "/address",
    "value": "example@example.com"
  }
]
```

- msisdn - строковое поле с номером телефона длиной 10 символов, только цифры
- contactType - строковое поле с типом контакта
- principal.externalId - уникальный идентификатор пользователя во внешней системе

На поле address с типом (contactType) "phone" действуют такие же ограничения как и на msisdn

ВАЖНО

У одного пользователя не может быть два объекта Contact с одинаковым contactType

Изменение msisdn

Изменение msisdn у активной учетной записи запрещено. Для изменения msisdn, необходимо сначала удалить текущую активную учетную запись, после чего, создать новую запись с таким же идентификатором пользователя из внешней системы (externalId) и новым msisdn.

Удаление учетной записи

Удаление учетной записи происходит с помощью HTTP запроса DELETE. Для идентификации учетной записи в URL операции можно использовать один из следующих вариантов: * `uid` - уникальный идентификатор пользователя в RooX UIDM * `msisdn` - (только в версиях продукта для телеком) номер телефона пользователя * `msisdn, externalId` - (только в версиях продукта для телеком) номер телефона пользователя и уникальный

идентификатор пользователя во внешней системе. Семантически эквивалентен идентификации записи по номеру телефона.

Пример запроса

```
DELETE /sso/provision/principals?msisdn=9211234567&externalId=123
Host: {{sso_host}}
Accept: application/json
```

- msisdn - строковое поле с номером телефона длиной 10 символов, только цифры
- externalId - уникальный идентификатор пользователя во внешней системе

В случае успешного удаления учетной записи, возвращается пустой ответ с HTTP статусом 204

Блокировка/Разблокировка учетной записи

Блокирование/Разблокирование учетной записи происходит путем изменения полей blocked и blockedTo, запросом на изменение

- blocked - признак блокировки (true - учетная запись заблокирована, false - разблокирована)
- blockedTo - дата, время в формате UTC, без временной зоны - определяет срок блокировки. Разблокировка происходит при первой аутентификации после указанного времени. Для бессрочной блокировки нужно передать null или пустую строку для blockedTo, тогда учетная запись может быть разблокирована только вручную, администратором системы.

Пример запроса

```
PATCH /sso/provision/principals?msisdn=9211234567&externalId=123
Host: {{sso_host}}
Content-Type: application/json-patch+json
Accept: application/json
```

```
[
  {
    "op": "replace",
    "path": "/blocked",
    "value": true
  },
  {
    "op": "replace",
    "path": "/blockedTo",
    "value": "2015-02-18T12:00:00.000+00:00"
  },
  {
    "op": "replace",
    "path": "/blockedReasonId",
    "value": "2"
  }
]
```

- msisdn - строковое поле с номером телефона длиной 10 символов, только цифры
- blocked - тип boolean, возможные значения: true или false
- blockedTo - строковое значение даты и времени в формате UTC (в формате дата-время ISO 8601)
- blockedReasonId - код причины блокировки из справочника (справочник будет предоставлен отдельно)
- externalId - уникальный идентификатор пользователя во внешней системе

В случае успешной блокировки/разблокировки возвращается пустой ответ с HTTP статусом 204

Обработка ошибок

Если операция завершилась неуспешно - будет возвращен соответствующий HTTP статус и описание ошибки следующего вида в теле ответа. Все HTTP коды кроме **20X** следует расценивать как ошибки provisioning-а, логировать код.

Примеры ответов с ошибкой

```
{
  "error": {
    "code": 400,
    "message": "RX_SSO_PROVIS_9004: principal should have property 'msisdn'"
  }
}
```

```
{
  "error": {
    "code": 400,
    "message": "RX_SSO_PROVIS_9004: credentials should have property 'login'"
  }
}
```

```
{
  "error": {
    "code": 404,
    "message": "RX_SSO_PROVIS_9001: User with msisdn '9211234567' not found"
  }
}
```

```
{
  "error": {
    "code": 400,
    "message": "RX_SSO_PROVIS_9003: Invalid JSON PATCH format"
  }
}
```

```
{
  "error": {
    "code": 400,
    "message": "RX_SSO_PROVIS_9002: Principal format error. Unrecognized field
'wrong_property'"
  }
}
```



```
{
  "error": {
    "code": 409,
    "message": "User with msisdn '9211234567' already exists"
  }
}
```