# Users-service

Часть микро сервиса, в которой собраны все операции с пользователем.

## Микро сервис содержит:

### База данных:

**users**

### Список Таблиц:

|  |  |  |
| --- | --- | --- |
| **Название таблицы** | **Что хранится** | **Примечания** |
| users | Хранится список пользователей | [Когда пользователь регистрируется, его данные добавляются в таблицу](#_Описание_и_логика) |
| password-reset-token | Хранятся токены, необходимые для изменения пароля | [Токены отправляются пользователю в письме, когда он хочет сменить свой пароль](#_Смена_пароля:) |

### API:

http://.../users-service/users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API** | **метод** | **описание** | **авторизация** | **JSON** |
| / | POST | [Метод для регистрации нового пользователя](#_Регистрация_нового_пользователя:) | Не требуется | {  "firstName":"User",  "lastName":"Khr",  "email":"nu@mail.co”,  "password":"123",  "lang":"RUS",  "myCountry":"ISRAEL"  } |
| /login | POST | Метод входа в приложение |  | {  "firstName":"User",  "password":"123"  } |
| /create-admin | GET | [Метод, необходим для scheduler-service](#_Создание_супер_администратора:) | Не требуется | Данные доступны в свойствах аппликации. |
| /delete-admin/{adminId} | DELETE | Метод, который удаляет супер администратора | Не требуется |  |
| /{id} | GET | Получение данных пользователя | Требуется |  |
| /{id} | PUT | [Обновление данных пользователя](#_Обновление_своих_данных:) | Требуется | {  "firstName":"User",  "lastName": "Khr",  "lang":"RUS",  "myCountry":"ISRAEL"  } |
| /{id} | DELETE | [Удаление пользователя](#_Удаление_учетных_данных) | Требуется |  |
| / | GET | [Получение данных всех пользователей системы](#_Получение_всех_пользователей:) | Требуется |  |
| /email-verification | GET | Метод для подтверждения электронной почты. Вызывается, когда пользователь нажимает на ссылку в письме подтверждения своего электронного адреса | Требуется |  |
| /password-reset-request | POST | [Метод для смены пароля. Первый шаг – пользователь вводит свой электронный адрес, и получает на него письмо с токеном для изменения пароля](#_Смена_пароля:) | Не требуется | {  "email":"nu@mail.co”,  } |
| /password-reset | POST | [Второй шаг метода «обновить пароль»](#_Смена_пароля:) | Не требуется | {  "password":"123",  "token": "пеУ№пвапл0"  } |
| **Hystrix – методы, отправляют запрос в send-email-service** | | | | |
| /verification-email | POST | Запрос на оправку письма для подтверждения почты | | |
| /password-reset-request | POST | Запрос на отправку письма для смены пароля | | |
| /clean-not-confirmed-mail | POST | Запрос на отправку письма о том, что база данных отчищена от пользователей с неподтвержденным e-mail | | |
| /error-mail | POST | Запрос на отправку письма об ошибке | | |

### **Описание и логика методов:**

#### **Регистрация нового пользователя:**

При регистрации новый пользователь вводит необходимые данные:

* имя (firstName),
* фамилия(lastName),
* электронный адрес(email),
* пароль(password),
* язык интерфейса(lang)
* страну проживания (myCountry)

язык интерфейса – это язык, на котором будет отображаться интерфейс приложения

страна проживания нужна для того, чтобы в дальнейшем, когда будет реализована функция обмена валюты пользователь сможет «поменять» свою валюту исходя из официального курса той страны, которая указана в пункте «страна проживания».

Пользователю на выбор доступны языки:

* RUS (по умолчанию)
* ENG
* HEB

В бесплатной версии будет возможность выбрать страну:

* ISRAEL (по умолчанию)
* UKRAINE

Но в платной версии можно связаться с администратором и попросить о добавлении необходимой пользователю страны.

Электронный адрес должен быть релевантным. После регистрации на него будет отправлено письмо для подтверждения и окончания регистрации. Пользователи с неподтвержденным e-mail будут считаться незарегистрированными.

Раз в неделю (в воскресенье в 00:00 «чистильщик» и удалять из базы данных всех пользователей, у которых не подтверждена регистрация.

После того, как пользователь введет свои данные и нажмет кнопку «зарегистрироваться» его данные будут добавлены в базу данных, а на почту придёт письмо для подтверждения регистрации. В письме будет ссылка, на которую пользователь должен нажать и подтвердить регистрацию.

После того как пользователь подтвердит регистрацию, он сможет зайти в приложение под своим логином, используя пароль.

#### **Обновление своих данных:**

Пользователь может обновить свои данные. А именно:

* имя (firstName),
* фамилия(lastName),
* язык интерфейса(lang)
* страну проживания (myCountry)

#### **Удаление учетных данных**

#### **Получение всех пользователей:**

Администратор может посмотреть список всех пользователей, зарегистрированных в приложении вместе с их учетными данными.

#### **Смена пароля:**

Если пользователь забыл пароль – он может нажать на кнопку «забыл пароль». В этом случае на его электронную почту придет письмо с временным токеном и ссылкой для изменения пароля.

Пользователь может пройти по ссылке из письма и изменить свой пароль в базе данных.

#### **Добавление члена семьи:**

Когда пользователь регистрируется в приложении он может выбрать, хочет ли он быть «независимым пользователем» или хочет присоединиться к семье.

Если пользователь хочет присоединиться к семье, он должен в списке пользователей выбрать того пользователя, в чью семью он хочет присоединиться.

Пользователю «хозяину семьи» будет отправлено письмо с вопросом – действительно ли этот пользователь является «членом семьи». Если да, и «хозяин семьи» готов открыть свои счета для пользователя, пославшего запрос – он должен просто перейти по указанной в письме ссылке.

Если «хозяин семьи» подтвердил «родство», то «родственник» сможет получать доступ ко всем счетам и операциям, которые открыты для данной семьи.

### Дополнительные методы:

##### Создание суперадминистратора:

Метод, который вызывается при каждом запуске задач из scheduler-service. Перед тем, как начинать выполнять задачу, сервис создает суперадминистратора с введенными заранее данными, и таким образом получает пароль и идентификатор администратора, необходимые для того, что бы создать релевантный токен.

Перед тем, как создавать суперадминистратора, метод проверят – если такой пользователь есть в базе данных, то новый не создается.

### **Структура таблиц:**

#### users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column | Datatype | PK | NN | AI | Defoult |
| id | BIGINT | + | + | + |  |
| email | VARCHAR(120) |  | + |  |  |
| email\_verification\_status | BIT(1) |  | + |  |  |
| email\_verification\_token | VARCHAR(255) |  |  |  | NULL |
| encrypted\_password | VARCHAR(255) |  | + |  |  |
| first\_name | VARCHAR(50) |  | + |  |  |
| last\_name | VARCHAR(50) |  | + |  |  |
| user\_id | VARCHAR(255) |  | + |  |  |
| lang | VARCHAR(255) |  | + |  |  |
| my\_country | VARCHAR(255) |  |  |  | NULL |

#### password-reset-token

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column | Datatype | PK | NN | AI | Defoult |
| id | BIGINT | + | + | + |  |
| token | VARCHAR(255) |  |  |  | NULL |
| user\_id | BIT(1) |  | + |  | NULL |

### **Application properties:**

|  |  |
| --- | --- |
| Свойство | Описание |
| server.port |  |
| spring.application.name |  |