# Bills-service

Часть микро сервиса, в которой собраны все операции связанные со счетами, добавление операций, категорий, подкатегорий.

## Микро сервис содержит:

### База данных:

**Bills**

### Список Таблиц:

|  |  |  |
| --- | --- | --- |
| **Название таблицы** | **Что хранится** | **Примечания** |
| bill | Хранит список счетов каждого пользователя приложения | Когда пользователь создает новый счет, данные счета сохраняются |
| category | Хранит список категорий всех пользователей | Необходим, когда пользователь создает операцию. Если пользователь хочет пометить в какой категории произведена операция, то он может выбрать одну из доступных категорий или добавить свою категорию |
| subcategory | Хранит список подкатегорий всех пользователей | Необходим, когда пользователь создает операцию. Если пользователь хочет пометить в какой подкатегории произведена операция, то он может выбрать одну из доступных подкатегорий или добавить свою категорию.  Подкатегория обязательно связана с категорией! |
| operations | Хранит список всех операций конкретного пользователя | Все подробности операции |

### API:

#### <http://.../bills-service/bill>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API** | **метод** | **описание** | **авторизация** | **JSON** |
| / | POST | Метод добавления нового счета | Требуется | {  "billNumber":1,  "startSum":0.0,  "userUuid":"11111",  "description":"Bill of bank"  } |
| /allBills/{userUuid} | GET | Метод для получения всех счетов конкретного пользователя | Требуется |  |
| /allBills/details/{userUuid} | GET | Метод для получения всех счетов конкретного пользователя с подробностями и датами | Требуется |  |
| /{userUuid}/{billUuid} | DELETE | Метод, который удаляет конкретный счет конкретного пользователя | Требуется |  |
| /{userUuid}/{billUuid} | PUT | Метод, который меняет конкретный счет конкретного пользователя | Требуется |  |

#### <http://.../bills-service/category>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API** | **метод** | **описание** | **авторизация** | **JSON** |
| / | POST | Метод добавления новой категории | Требуется | {  "categoryName":"buying",  "userUuid":"11111",  "type":"OUTCOME",  "decryption":"shop1"  } |
| /{uuid}/{userUuid} | DELETE | Удаляет конкретную категорию конкретного пользователя | Требуется |  |

#### <http://.../bills-service/subcategory>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API** | **метод** | **описание** | **авторизация** | **JSON** |
| / | POST | Метод добавления новой субкатегории | Требуется | {  "subCategoryName":"buying eat",  "userUuid":"11111",  "category":{  "categoryName":"buying",  "categoryUuid": "22222"  },  "type":"OUTCOME",  "decryption":"buying apple"  } |
| /{uuid}/{userUuid} | DELETE | Удаляет конкретную субкатего -рию конкретного пользователя | Требуется |  |

#### http://.../bills-service/operations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API** | **метод** | **описание** | **авторизация** | **JSON** |
| / | POST | Метод добавления новой операции | Требуется | {  "userUuid":"11111",  "billUuid":"44444",  "categoryUuid":"22222",  "sum":78.0,  "currency":"RUB",  "type":"OUTCOME",  "decryption":"buying for week"  } |
| /{userUuid} | GET | Метод получения всех операций пользователя | Требуется |  |
| /{operationUuid}/{userUuid}/{type} | DELETE | Метод удаления конкретной операции конкретного пользователя | Требуется |  |
| /{operationUuid}/{userUuid}/{type} | PUT | Метод изменения конкретной операции конкретного пользователя | Требуется | { "billUuid":"44444",  "categoryUuid":"22222", "sum":53.6, "currency":"RUB"  } |

### **Описание и логика методов:**

#### **Добавление нового счета:**

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 |  |