

## КОММЕНТАРИИ К РЕАЛИЗАЦИИ ПРОЕКТА

### Конфигурация

в качестве рабочей указана база *bks* (свойство *spring.datasource.url*). Возможно, использовать и указать свою.

Необходимо заменить на локальные значения свойств *spring.datasource.username* и *spring.datasource.password*

Также указать локальные значения свойств для связи с брокером сообщений - *spring.rabbitmq.host*, *spring.rabbitmq.username*, *spring.rabbitmq.password*

Лог приложения формируется как */log/bks.log* в корне проекта (свойство *logging.file.name*)

За исключением базы все необходимое создается при старте приложения автоматически:

- таблицы (и сиквенс) согласно скриптам Liquibase

- иксчендж и очередь в RabbitClient согласно значениям свойств *app.client\_exchange* и *app.client\_queue* в *application.properties*

Переименовал таблицу *User* в *Client* во избежание возможных конфликтов с одноименным зарезервированным SQL словом. В дальнейшем буду говорить именно о клиентах.

Конечно, хранить клиентские пароли в базе необходимо в зашифрованном виде, но поскольку в задаче это явно не указано, то...

### Функционирование

Для демонстрации таска 'Создание нового пользователя должно происходить с помощью принятия сообщения из RabbitMQ' при старте приложения формируется DTO тестового клиента, затем сериализуется, публикуется в иксчендж Rabbit MQ, десериализуется при чтении из очереди и обрабатывается сервисом регистрации нового клиента (см. метод *run* класса *BksCrmApplication*).

```
2023-03-27 01:38:30.227 INFO 11828 --- [pool-2-thread-4] com.bks.BksCrmApplication : зарегистрирован новый клиент: ClientDto(id=null, name=Andy, birthdate=2005-11-07, password=new_client_pass)
```

Также имеется REST точка создания нового клиента.

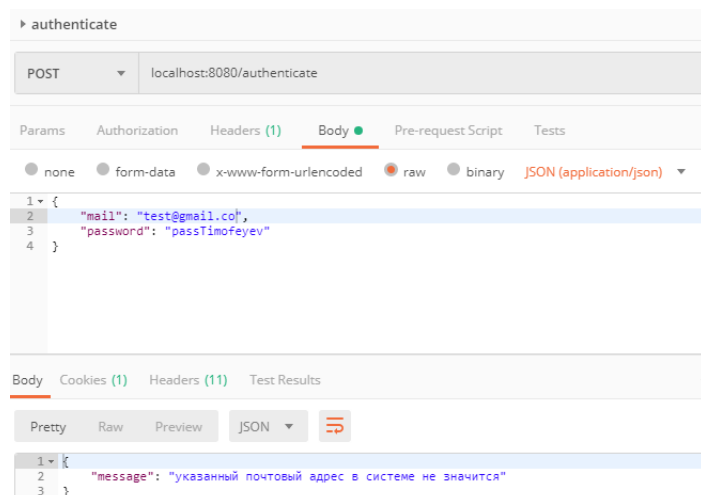
**Результат периодического инкрементирования баланса** каждого клиента (до достижения уровня в 207% от стартовой величины) выводится в лог с округлением до двух знаков после десятичной точки

```
2023-03-27 00:15:44.701 INFO 8952 --- [BKS-1] com.bks.service.AccountServiceImpl : client ID - 1, current balance - 51.23, changed at - 2023/03/27 00:15:44
2023-03-27 00:15:44.702 INFO 8952 --- [BKS-1] com.bks.service.AccountServiceImpl : client ID - 2, current balance - 59.98, changed at - 2023/03/27 00:15:44
2023-03-27 00:15:44.702 INFO 8952 --- [BKS-1] com.bks.service.AccountServiceImpl : client ID - 3, current balance - 140.37, changed at - 2023/03/27 00:15:44
2023-03-27 00:16:13.998 INFO 8952 --- [BKS-2] com.bks.service.AccountServiceImpl : client ID - 1, current balance - 56.35, changed at - 2023/03/27 00:16:13
2023-03-27 00:16:13.999 INFO 8952 --- [BKS-2] com.bks.service.AccountServiceImpl : client ID - 2, current balance - 65.98, changed at - 2023/03/27 00:16:13
2023-03-27 00:16:13.999 INFO 8952 --- [BKS-2] com.bks.service.AccountServiceImpl : client ID - 3, current balance - 154.41, changed at - 2023/03/27 00:16:13
```

Приложение находится под управлением Spring Security, поэтому для доступа к REST точкам и Swagger необходимо авторизоваться.

Из предложенных вариантов выбран вариант аутентификации по комбинации mail + password

В приложении формируются пользовательские исключения, поэтому в случае ошибочной комбинации mail + password вас приветствуют, например, таким сообщением:



В случае верной комбинации формируется JWT:

authenticate

POSTlocalhost:8080/authenticate

SendSave

ParamsAuthorizationHeaders (1)BodyPre-request ScriptTests

noneform-datax-www-form-urlencodedrawbinaryJSON (application/json)

Beautiful

1 {  
2 "mail": "test@gmail.com",  
3 "password": "passTimofeyev"  
4 }

client

email\_data

	id	date_of_birth	name	password	id	mail	client_id
1	1	2000-01-01	Ivan Ivanov	passivanov	1	test@mail.com	1
2	2	2000-02-02	Pavel Pavlov	passPavlov	2	test@yahoo.com	2
3	3	2000-03-03	Timofey Timofeyev	passTimofeyev	3	test@gmail.com	3
4	4	2005-11-07	Andy	new_client_pass			

BodyCookies (1)Headers (11)Test Results

Status: 200 OKTime: 105 msSize: 499 BSaveDownload

PrettyRawPreviewJSON

```
1 {  
2 "jwt": "eyJhbGciOiJIUzI1NiIsInp0eSI6Ij02ZXN0QmdYbWV5SIsImV4OiI6MTY3OTkwNjg0MSwiaWQiOiJ0eXN0QmdYbWV5LW1-cmgAPYnJXpNkA5ciR1UE9fb-Lh1khFVYI48siqCM"  
3 }
```

REST точки требуют указания JWT в заголовке *Authorization* - в противном случае статус отклика 403:

get client by phone

GETlocalhost:8080/api/client/getbyphone/phone

Send

ParamsAuthorizationHeaders (1)BodyPre-request ScriptTests

KEY	VALUE	DESCRIPTION
phone	79207865434	Description
Key	Value	Description

BodyCookies (1)Headers (10)Test Results

Status: 403 ForbiddenTime: 37 msSize: 312 B

PrettyRawPreviewAuto

1

get client by phone

GETlocalhost:8080/api/client/getbyphone/phone

Send

ParamsAuthorizationHeaders (1)BodyPre-request ScriptTests

Authorization

KEY	VALUE	DESCRIPTION
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInp0eSI6Ij02ZXN0QmdYbWV5SIsImV4OiI6MTY3OTkwNjg0MSwiaWQiOiJ0eXN0QmdYbWV5LW1-cmgAPYnJXpNkA5ciR1UE9fb-Lh1khFVYI48siqCM	
Key	Value	Description

BodyCookies (1)Headers (11)Test Results

Status: 200 OKTime: 75 msSize: 433 BSave

PrettyRawPreviewJSON

```
1 {  
2 "id": 3,  
3 "name": "Timofey Timofeyev",  
4 "birthdate": "2000-03-03",  
5 "password": "passTimofeyev"  
6 }
```

В приложении реализованы следующие REST точки:

- без пагинации

api/client/getbyemail/{mail}

api/client/getbyphone/{phone}

- с пагинацией

api/client/getbyname(частичное совпадение)

api/client/getbybirthdate (моложе указанной даты)

get client by birthdate

GET localhost:8080/api/client/getbybirthdate?birthdate=02.02.2000&page=0&size=2

Params Authorization Headers (1) Body Pre-request Script Tests

KEY	VALUE	DESCRIPTION
Authorization	Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ0ZXN0QGdtYWlsL...	
Key	Value	Description

Body Cookies (1) Headers (11) Test Results Status: 200 OK Time: 101 ms Size: 624

Pretty Raw Preview JSON

```

1 {
2   "content": [
3     {
4       "id": 3,
5       "name": "Timofey Timofeyev",
6       "birthdate": "2000-03-03",
7       "password": "pass1timofeyev"
8     },
9     {
10      "id": 4,
11      "name": "Andy",
12      "birthdate": "2005-11-07",
13      "password": "new_client_pass"
14    }
15  ],
16  "pageable": {
17    "sort": {
18      "empty": true,
19      "sorted": false,
20      "unsorted": true
21    },
22    "offset": 0,
23    "pageNumber": 0,
24    "pageSize": 2,
25    "paged": true,
26    "unpaged": false
27  },
28  "last": true,
29  "totalPages": 1,
30  "totalElements": 2,
31  "number": 0,
32  "size": 2,
33  "sort": {
34    "empty": true,
35    "sorted": false,
36    "unsorted": true
37  },
38  "first": true,
39  "numberOfElements": 2,
40  "empty": false
41 }

```

- api/client/registerclient; регистрации нового клиента

register client

POST localhost:8080/api/client/registerclient Send

Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code

none form-data x-www-form-urlencoded raw binary JSON (application/json)

```

1 {
2   "name": "Потан",
3   "password": "passПотанович",
4   "birthdate": "06.06.2020"
5 }

```

Body Cookies (1) Headers (11) Test Results Status: 201 Created Time: 87 ms Size: 440 B Save

Pretty Raw Preview JSON

```

1 {
2   "id": 5,
3   "name": "Потан",
4   "birthdate": "2020-06-06",
5   "password": "passПотанович"
6 }

```

- api/client/moneytransfer; перевода средств между счетами

Критические сообщения при трансфере средств между счетами формируются в следующих ситуациях:

- ошибочный id отправителя
- ошибочный id получателя
- недостаточно средств на счете отправителя
- значение, передаваемое между счетами, является infinite либо NAN, что приведет к NumberFormatException при конвертации в BigDecimal

например:

money transfer

PUT localhost:8080/api/client/moneytransfer?sourceId=1&recipientId=2&amount=50.3 Send

Params Authorization Headers (1) Body Pre-request Script Tests Cookies Code

KEY	VALUE	DESCRIPTION
sourceId	1	
recipientId	2	
amount	50.3	
Key	Value	Description

Body Cookies (1) Headers (11) Test Results Status: 404 Not Found Time: 143 ms Size: 432 B Save

Pretty Raw Preview JSON

```
1 {  
2   "message": "у отправителя недостаточно средств"  
3 }
```

К слову, структура JWT:

JWT

Debugger Libraries Introduction Ask

Crafted by auth0 by Okta

Algorithm HS256

Encoded PASTE A TOKEN HERE

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256"  
}
```

PAYLOAD: DATA

```
{  
  "sub": "test@gmail.com",  
  "exp": 1679908386,  
  "iat": 1679872386  
}
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret  
)  
☐ secret base64 encoded
```

Signature Verified

SHARE JWT

Прошу прощения, что клейм USER\_ID оставил на возможное будущее. Токену назначено время жизни (10 ч).

Описание REST API доступно по адресу <http://localhost:8080/v3/api-docs> при работающем сервере будучи авторизованным.

В дополнение к этому документу передаются:

- коллекция REST запросов (с параметрами) для импорта в Postman. Необходимо лишь исправить header Authorization в соответствии с полученным значением JWT
- пример лога

Код проекта находится по адресу

<https://github.com/hyperborean72/bks>