



Финальный проект Мария

Ссылка на задачу	<input checked="" type="checkbox"/> JUS-234: Финальный проект [to do]
Статус страницы	В работе
Описание	Проектная документация к тестированию веб-приложения Bank Saint-Petersburg

Описание фичей, к которым предоставлены требования и описания требований.

▼ Описание фичей

1. Экран авторизации: ввод логина demo, пароль 0000.
2. Код из смс - 0000.
3. Главный экран.
4. Экран “Выписка со счета в валюте (CNY), период 12.01.2025 - 19.01.2025” в разделе “Счета”.
5. Экран “Переписка с банком”.
6. Экран “Персональные предложения”, блок “The best banner DOMS Banner 3”.
7. Экран “Кредиты”, нажать “Оформить кредитный продукт”, раздел “кредитная карта, нажать на кнопку “Оформить кредитный продукт”.
8. Экран “Обмен валюты”, раздел “Валюта”.

▼ Требования

Требования

1. Логин: от 5 до 10 символов, латиница, нижний регистр.
- 1.1. Пароль: от 7 до 15 символов, латиница, нижний регистр и верхний регистр, цифры.
- 1.2. При вводе корректных значений осуществляется переход к окну входа через SMS.
- 1.3. При трех неудачных попытках аккаунт блокируется на 1 час.
2. Номер для смс должен быть верифицирован в настройках аккаунта.
- 2.1. Код должен состоять из цифр, не более 4 знаков.
- 2.3. Повторная отправка кода осуществляется каждые 60 минут.
- 2.4. При 3 неудачных попытках входа аккаунт блокируется на 1 час.
- 2.5. При успешном вводе кода и нажатии кнопки «Войти» осуществляется переход на главную экран личного кабинета пользователя.

3. Чтобы перейти с главной страницы веб-приложения на страницу "Выписка со счета в валюте (CNY)", наведите на раздел "Счета", затем нажмите на кнопку "Выписка". Отобразилось поле, в котором отобразилась полная информация о конкретном счете.

3.1. Чтобы осуществить связь с банком через личный кабинет, необходимо нажать на иконку "Конверт" (связь с банком) на главной странице. Отображается экран, на котором возможно отправить сообщение банку.

3.2. Чтобы перейти с главной страницы веб-приложения на страницу «Персональные предложения», нажмите на значок в виде подарка в правом верхнем углу. Отображается экран с «Персональные предложения» банка. При клике на блок DOMS Banner 3 отображается информация о выбранном подарке.

3.3. Чтобы выбрать счет списания и счет зачисления, а также чтобы произошла автоматическая конвертация валюты, необходимо:

а) Нажать на раздел "Валюта" ;

б) выбрать раздел "Простой режим" под заголовком "Обмен валюты".

3.3. Чтобы оформить кредитную карту, надо кликнуть на раздел "Кредиты", далее нажать на кнопку "Оформить кредитный продукт".

Тест-кейсы

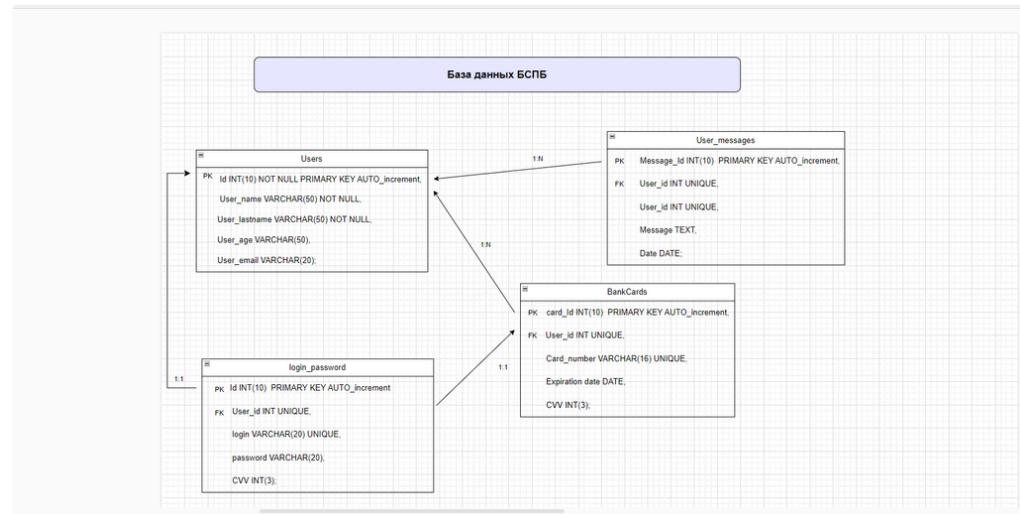
Описание	Тест-кейс
Выписка со счета в валюте (CNY), период 12.01.2025 - 19.01.2025	JUS-860: Выписка со счета в валюте (CNY), период 12.01.2025 - 19.01.2025 DONE
Связь с банком через сообщения в ЛК	JUS-859: Связь с банком через сообщения в ЛК DONE
Авторизация в интернет банке БСПБ	JUS-858: Авторизация в интернет банке БСПБ TO DO

Баг-репорты

Описание	Баг-репорт
Не открывается DOMS Banner 3 во вкладке "Персональные предложения"	JUS-864: Не открывается DOMS Banner 3 во вкладке «Персональные предложения» ГОТОВО
Не происходит автоматическая конвертация в поле "Сумма зачисления" при вводе суммы на экране обмен валюты	JUS-863: Не происходит автоматическая конвертация в поле "Сумма зачисления" при вводе суммы на экране обмен валюты ГОТОВО
При попытке оформить кредитную карту отображается сообщение "В данный момент подача заявки невозможна" при попытке оформить кредитную карту	JUS-865: Отображается информация "В данный момент подача заявки невозможна" при попытке оформить кредитную карту ГОТОВО

База данных

SQL



База данных банка “БСПБ”

- Создала базу данных bank_operations, затем создала таблицу Users, в таблицу внесла данные(id, имя пользователя, фамилию, возраст, емайл).

```

mysql> CREATE DATABASE bank_operations;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| bank_operations |
| information_schema |
| mysql |
| performance_schema |
| qwerty |
| students |
| sys |
| test |
+-----+
8 rows in set (0.00 sec)

mysql> USE bank_operations;
Database changed
mysql> CREATE TABLE Users
    -> (id int(10) PRIMARY KEY AUTO_INCREMENT,
    -> User_name VARCHAR(20),
    -> User_lastname VARCHAR(20),
    -> age VARCHAR(15),
    -> email VARCHAR(20) UNIQUE);
Query OK, 0 rows affected, 1 warning (0.02 sec)

mysql> CREATE TABLE BankCards
    
```

- Создала таблицу BankCards, содержащая поля (id, id пользователя, номер карты, дату окончания срока действия карты, код cvv), затем с помощью внешнего ключа соединила поле user_id с полем user_id таблицы Users.

```

mysql> show tables;
Empty set (0.00 sec)

mysql> CREATE TABLE Users
    -> (User_id INT PRIMARY KEY AUTO_INCREMENT,
    -> user_name VARCHAR(20) NOT NULL,
    -> user_lastname VARCHAR(20) NOT NULL,
    -> age VARCHAR(15) NOT NULL,
    -> email VARCHAR(20) UNIQUE);
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE BankCards
    -> (card_id INT PRIMARY KEY AUTO_INCREMENT,
    -> user_id INT,
    -> card_number VARCHAR(16) UNIQUE,
    -> expiration_date DATE,
    -> CVV INT(3),
    -> FOREIGN KEY (user_id) REFERENCES Users(user_id));
Query OK, 0 rows affected, 1 warning (0.02 sec)

mysql> CREATE TABLE login_password
    -> (id INT PRIMARY KEY AUTO_INCREMENT)
    
```

- Далее, создала таблицу login_password, содержащая (id, id пользователя, логин, пароль). С помощью внешнего ключа соединяла поле user_id с полем user_id таблицы BankCards.

- Создала таблицу User_messages, содержащую поля(id, id пользователя, сообщение, дату).

С помощью внешнего ключа соединяла поле user_id с полем user_id таблицы Users.

```

MySQL> DESC BankCards;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| card_id | int | NO | PRI | NULL | auto_increment |
| user_id | int | YES | UNI | NULL |
| card_number | varchar(16) | YES | UNI | NULL |
| expiration_date | date | YES | UNI | NULL |
| is_active | tinyint | YES | NULL | |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> CREATE TABLE login_password
-> (id INT PRIMARY KEY AUTO_INCREMENT,
-> user_id int,
-> login VARCHAR(20) UNIQUE,
-> password VARCHAR(20),
-> FOREIGN KEY (user_id) REFERENCES BankCards(user_id));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE User_Messages
-> (message_id INT PRIMARY KEY AUTO_INCREMENT,
-> user_id INT UNIQUE,
-> message TEXT,
-> date DATETIME,
-> FOREIGN KEY (user_id) REFERENCES Users(user_id));
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_bank_operations |
+-----+
| bankcards
| login_password
| user_messages
| users
+-----+
4 rows in set (0.00 sec)

mysql> 

```

5. В таблицу BankCards и login_password внесла значения.

```

MySQL> 
+-----+-----+-----+-----+-----+
| expiration_date | date | YES | | NULL | |
| CVV | int | YES | | NULL | |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> INSERT INTO BankCards VALUES
-> ('55109', '1', '2202456787982', '2028-12-01', '234');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO BankCards VALUES
-> ('22109', '2', '2208996789792', '2029-09-08', '897'),
-> ('45678', '3', '22085167834', '2026-01-18', '189'),
-> ('12345', '4', '2202456787982', '2026-11-22', '117'),
-> ('22167', '5', '2202772821197', '2026-05-27', '594');
Query OK, 4 rows affected (0.01 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> DESC login_password;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id | int | NO | PRI | NULL | auto_increment |
| user_id | int | YES | UNI | NULL |
| login | varchar(20) | YES | UNI | NULL |
| password | varchar(20) | YES | | NULL | |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> INSERT INTO login_password VALUES
-> (1, 1, 'marivesna', 'marvds5'),
-> (2, 2, 'qwerty', 'sandinol1'),
-> (3, 3, 'grisho9', 'siren1'),
-> (4, 4, 'kuzh777', 'g5gl2xmq'),
-> (5, 5, 'valya', 'logval');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> 

```

6. В таблицу user_messages внесла значения.

```

MySQL> SHOW TABLES;
+-----+
| Tables_in_bank_operations |
+-----+
| bankcards
| login_password
| user_messages
| users
+-----+
4 rows in set (0.00 sec)

mysql> DESC user_messages;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| message_id | int | NO | PRI | NULL | auto_increment |
| user_id | int | YES | UNI | NULL |
| message | text | YES | | NULL | |
| date | datetime | YES | | NULL | |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> INSERT INTO user_messages VALUES
-> (1, 1, 'Добрый день! Хочу открыть счет в евро.', '2025-02-12 12:20:19'),
-> (2, 2, 'Хочу сделать выписку со счета.', '2025-02-11 11:28:05'),
-> (3, 3, 'Нужно проверить баланс.', '2025-02-12 15:26:45'),
-> (4, 4, 'Перекличите пожалуйста на кредитный отдел.', '2025-02-11 09:54:45'),
-> (5, 5, 'Хочу заблокировать карту.', '2025-02-10 16:18:22');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> 

```

7. В таблицу users внесла значения.

```

mysql> DESC Users;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| user_id | int | NO | PRI | NULL | auto_increment |
| user_name | varchar(20) | NO | | NULL | |
| user_lastname | varchar(20) | NO | | NULL | |
| age | varchar(15) | NO | | NULL | |
| email | varchar(20) | YES | UNI | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> INSERT INTO Users VALUES
-> (1,'Egorova', 'Maria', '36', 'mariavesna@yandex.ru'),
-> (2,'Ivanova', 'Elena', '38', 'elen99@mail.ru'),
-> (3,'Zaharov', 'Grigorii', '40', 'grisha555@gmail.com'),
-> (4, 'Kravtsov', 'Danil', '25', 'danka@yandex.ru'),
-> (5, 'Loginov', 'Valentin', '30', 'valval@mail.ru');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql>

```

8. Вывела все таблицы с данными.

```

MySQL 8.4 Command Line Cli  +  -
+-----+-----+-----+-----+-----+
| users |          ↓          |
+-----+-----+-----+-----+-----+
0 rows in set (0.00 sec)

mysql> SELECT * FROM BankCards;
+-----+-----+-----+-----+
| card_id | user_id | card_number | expiration_date | CVV |
+-----+-----+-----+-----+
| 18924 | 4 | 2201557889100 | 2026-11-22 | 117 |
| 22189 | 2 | 2208996789792 | 2029-09-08 | 897 |
| 22167 | 5 | 2202772824107 | 2028-05-27 | 554 |
| 45678 | 3 | 2208521167834 | 2026-01-10 | 189 |
| 55189 | 1 | 2202456787982 | 2028-12-01 | 234 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM users;
+-----+-----+-----+-----+-----+
| user_id | user_name | user_lastname | age | email |
+-----+-----+-----+-----+
| 1 | Egorova | Maria | 36 | mariavesna@yandex.ru |
| 2 | Ivanova | Elena | 38 | elen99@mail.ru |
| 3 | Zaharov | Grigorii | 40 | grisha555@gmail.com |
| 4 | Kravtsov | Danil | 25 | danka@yandex.ru |
| 5 | Loginov | Valentin | 30 | valval@mail.ru |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM login_password;
+-----+-----+-----+
| id | user_id | login | password |
+-----+-----+-----+
| 1 | 1 | mariivesna | marvd5s |
| 2 | 2 | queryt | sandinol |
| 3 | 3 | grishb99 | siren1 |
| 4 | 4 | krvz777 | g5g12xwq |
| 5 | 5 | valya | logval |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM user_messages;

```

```

MySQL 8.4 Command Line Cli  +  -
+-----+-----+-----+-----+-----+
| 55189 | 1 | 2202456787982 | 2028-12-01 | 234 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM users;
+-----+-----+-----+-----+-----+
| user_id | user_name | user_lastname | age | email |
+-----+-----+-----+-----+
| 1 | Egorova | Maria | 36 | mariavesna@yandex.ru |
| 2 | Ivanova | Elena | 38 | elen99@mail.ru |
| 3 | Zaharov | Grigorii | 40 | grisha555@gmail.com |
| 4 | Kravtsov | Danil | 25 | danka@yandex.ru |
| 5 | Loginov | Valentin | 30 | valval@mail.ru |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

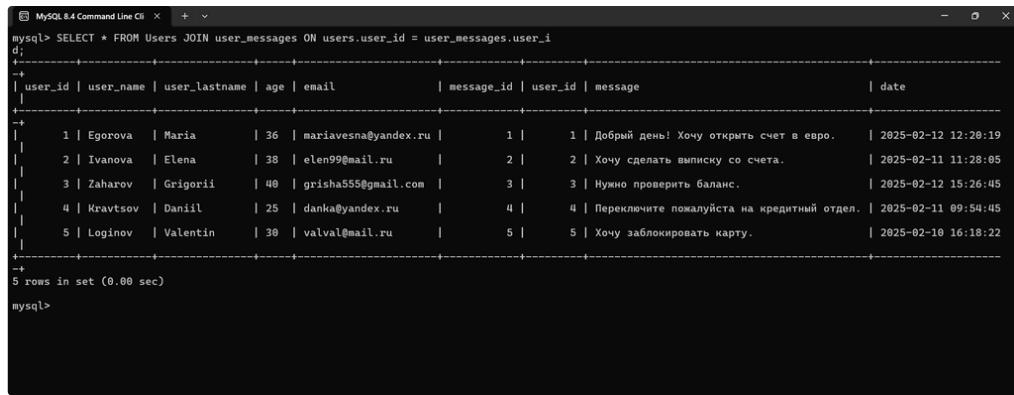
mysql> SELECT * FROM login_password;
+-----+-----+-----+
| id | user_id | login | password |
+-----+-----+-----+
| 1 | 1 | mariivesna | marvd5s |
| 2 | 2 | queryt | sandinol |
| 3 | 3 | grishb99 | siren1 |
| 4 | 4 | krvz777 | g5g12xwq |
| 5 | 5 | valya | logval |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM user_messages;
+-----+-----+-----+-----+
| message_id | user_id | message | date |
+-----+-----+-----+-----+
| 1 | 1 | Добрый день! Хочу открыть счет в евро. | 2025-02-12 12:28:19 |
| 2 | 2 | Хочу сделать выписку со счета. | 2025-02-12 13:08:05 |
| 3 | 3 | Нужно проверить баланс. | 2025-02-12 15:10:45 |
| 4 | 4 | Помогите с переводом карты на кредитный отдел. | 2025-02-11 09:54:45 |
| 5 | 5 | Хочу заблокировать карту. | 2025-02-18 16:18:22 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>

```

9. Далее, я объединила таблицу user_messages с таблицей users по общему признаку user_id.



MySQL 8.4 Command Line Cli

```
mysql> SELECT * FROM Users JOIN user_messages ON users.user_id = user_messages.user_id;
+-----+-----+-----+-----+-----+-----+-----+
| user_id | user_name | user_lastname | age | email | message_id | user_id | message | date |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Egorova | Maria | 36 | mariavesna@yandex.ru | 1 | 1 | Добрый день! Хочу открыть счет в евро. | 2025-02-12 12:29:19 |
| 2 | Ivanova | Elena | 38 | elen99@mail.ru | 2 | 2 | Хочу сделать выписку со счета. | 2025-02-11 11:28:05 |
| 3 | Zaharov | Grigorii | 40 | grisha555@gmail.com | 3 | 3 | Нужно проверить баланс. | 2025-02-12 15:26:45 |
| 4 | Kravtsov | Daniil | 25 | danka@yandex.ru | 4 | 4 | Переключите пожалуйста на кредитный отдел. | 2025-02-11 09:54:45 |
| 5 | Loginov | Valentin | 30 | valval@mail.ru | 5 | 5 | Хочу заблокировать карту. | 2025-02-10 16:18:22 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

mysql>

❖ Развертывание MySQL-контейнера через Docker

1. Установка Docker.

Можно скачать Docker на официальном сайте [Get Docker](#)

2. Проверка установки Docker.

После установки нужно проверить, работает ли Docker, с помощью следующей команды в терминале (эта команда должна вывести версию Docker):

docker --version

3. Запуск MySQL-контейнера.

Для того чтобы запустить контейнер с MySQL, нужно выполнить следующую команду:

docker run --name mysql-container -e MariaSQL_ROOT_PASSWORD=... -d -p 3306:3306 mysql:latest.

Параметры команды:

--name mysql-container: задает имя контейнера (можно выбрать любое имя).

-e MariaSQL_ROOT_PASSWORD= ...: задает пароль для пользователя root. Нужно заменить yourpassword на безопасный пароль.

-d: запускает контейнер в фоновом режиме (detached mode).

-p 3306:3306: проксирует порт 3306 на локальной машине к порту 3306 в контейнере (это стандартный порт MySQL).

mysql:latest: указывает, что мы используем последнюю версию MySQL.

4. Проверка работы контейнера

После запуска контейнера можно проверить его статус с помощью команды (мы должны увидеть наш контейнер в списке работающих контейнеров):

docker ps

5. Подключение к MySQL.

Для подключения к MySQL в контейнере можно использовать команду:

```
docker exec -it mysql-container mysql -u root -p
```

После выполнения этой команды нам будет предложено ввести пароль (который мы указали ранее).

6. Управление контейнером.

Остановить контейнер:

```
docker stop mysql-container
```

Запустить контейнер снова:

```
docker start mysql-container
```

Удалить контейнер (если он остановлен):

```
docker rm mysql-container
```

Можно также использовать другие переменные окружения для настройки MySQL, например:

MYSQL_DATABASE: создаёт указанную базу данных при запуске контейнера.

MYSQL_USER и **MYSQL_PASSWORD**: создаёт нового пользователя с указанным паролем и правами на указанную базу данных.

Пример команды с дополнительными параметрами:

```
docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=yourpassword -e  
MYSQL_DATABASE=mydatabase -e MYSQL_USER=myuser -e MYSQL_PASSWORD=mypassword -d -p  
3306:3306 mysql:latest
```

```
C:\Users\User>docker exec -it MariaSQL /bin/bash
bash-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.2.0 MySQL Community Server - GPL

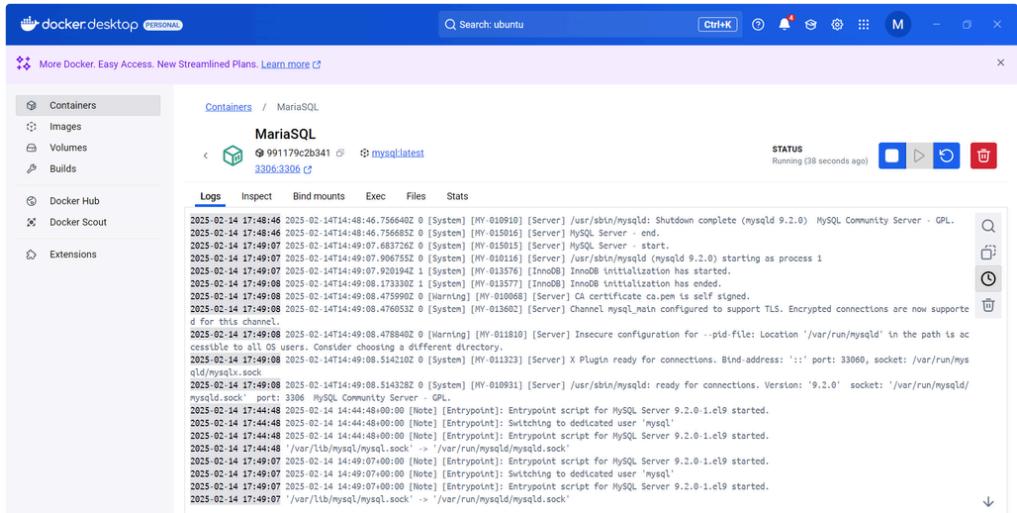
Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+

```



Ресурс

[Swagger UI](#)

Составить и выполнить 4 запроса: GET, POST, PUT, DELETE.

Коллекция Petstore в Postman нужна для тестирования REST API Swagger Petstore.

POST - создать пользователя;

GET - получить пользователя;

GET - зарегистрировать пользователя в системе;

PUT - обновить данные;

DELETE - удалить пользователя.

Запуск коллекции из командной строки

Запускать коллекции Postman из командной строки можно с помощью инструмента под названием Newman. Newman — это CLI-инструмент для Postman, который позволяет выполнять запросы и тесты коллекций Postman.

- Скачать Node.js (скачать можно с официального сайта [Node.js — Run JavaScript Everywhere](#))
 - После установки Node.js нужно установить Newman с помощью npm (менеджер пакетов Node.js). Нужно ввести следующую команду в терминале (эта команда установит Newman глобально на вашей системе):
- npm install -g newman**

3. Экспорт коллекции Postman:

Перед тем как запустить коллекцию, нужно ее экспорттировать из Postman:

- Открыть Postman.
- Найти коллекцию, которую хотим экспорттировать, в левой панели.
- Щелкнуть правой кнопкой мыши по коллекции и выбрать "Export".
- Выбрать нужный формат (обычно JSON) и сохранить файл на вашем компьютере.

4. Запуск коллекции с помощью Newman.

Выполните следующую команду в терминале, заменив your-collection.json на путь к экспортированному файлу:

newman run your-collection.json

newman run "C:\Maria\Final_Project.postman_collection111.json" -e

"C:\Maria\User.postman_environment11.json" - прямая команда к запуску моей коллекции в терминале

5. Проверка результатов.

После завершения выполнения команды в терминале мы увидим результаты запуска коллекции, включая информацию о пройденных и непройденных тестах.

Информация о запуске коллекции:

Your environment has been set up for using Node.js 20.17.0 (x64) and npm.

```
Windows PowerShell x + v
PS C:\Users\User> newman run "C:\Maria\Final_Project.postman_collection111.json" -e "C:\Maria\User.postman_environment11.json"
newman
Final_Project
+ Создать пользователя
  POST https://petstore.swagger.io/v2/user [200 OK, 3878, 715ms]
  1. Content-Type is present
+ Получить пользователя
  GET https://petstore.swagger.io/v2/user/Maria [200 OK, 5088, 170ms]
+ Зарегистрировала в системе
  GET https://petstore.swagger.io/v2/user/login [200 OK, 4718, 144ms]
+ Обновить данные
  PUT https://petstore.swagger.io/v2/user/Maria [200 OK, 3878, 135ms]
+ Удалить пользователя
  DELETE https://petstore.swagger.io/v2/user/Maria [200 OK, 3738, 137ms]

          executed    failed
iterations           1        0
requests              5        0
test-scripts          1        0
prerequest-scripts   0        0
assertions            1        1

total run duration: 1685ms
total data received: 429B (approx)
average response time: 260ms [min: 135ms, max: 715ms, s.d.: 227ms]
```

▼ JSON

info.docx

Bash Script

✓ Bash Script.

Для создания скрипта необходимо:

1. Перейти в директорию Final-Project_QA
2. Создать папку Bash scripts, перейти в нее
3. Создать файл Bash-scripts.sh
4. Перейти в редактор vim - vim Bash-scripts.sh
5. С помощью команды echo выводим текст
6. Обозначаем переменные a_1, a_2, a_3
7. С помощью команды read терминал считывает строку

```
echo "Введите первое целое число для сложения"
read a_1
echo "Введите второе целое число для сложения"
read a_2
echo "Прибавьте цифру 1 для сложения"
read a_3
echo "Результат = $(($a_1 + $a_2 + $a_3))"
```

8. Далее, запускаем скрипт с помощью команды sh (sh Bash-scripts.sh)

```
rm: cannot remove '/etc/post-install/01-devices.post': Permission denied
rm: cannot remove '/etc/post-install/03-mtab.post': Permission denied
rm: cannot remove '/etc/post-install/06-windows-files.post': Permission denied
rm: cannot remove '/etc/post-install/99-post-install-cleanup.post': Permission denied
UserDESKTOP-DTHR9IK MINGW64 ~ (develop)
$ cd Final-Project_QA
UserDESKTOP-DTHR9IK MINGW64 ~/Final-Project_QA (develop)
$ ls
Bash-scripts/ Bash_command/ Bug-reports/ Charles/ Client_server_architecture/ Postman/ README.md SQL/ Test-cases/
UserDESKTOP-DTHR9IK MINGW64 ~/Final-Project_QA (develop)
$ cd Bash-scripts
UserDESKTOP-DTHR9IK MINGW64 ~/Final-Project_QA/Bash-scripts (develop)
$ ls
Bash-scripts.sh
UserDESKTOP-DTHR9IK MINGW64 ~/Final-Project_QA/Bash-scripts (develop)
$ sh Bash-scripts.sh
Введите любое целое число для сложения
10
Введите второе целое число для сложения
12
Прибавьте цифру 1 для сложения
1
Результат = 16
UserDESKTOP-DTHR9IK MINGW64 ~/Final-Project_QA/Bash-scripts (develop)
$
```



✓ Bash_command

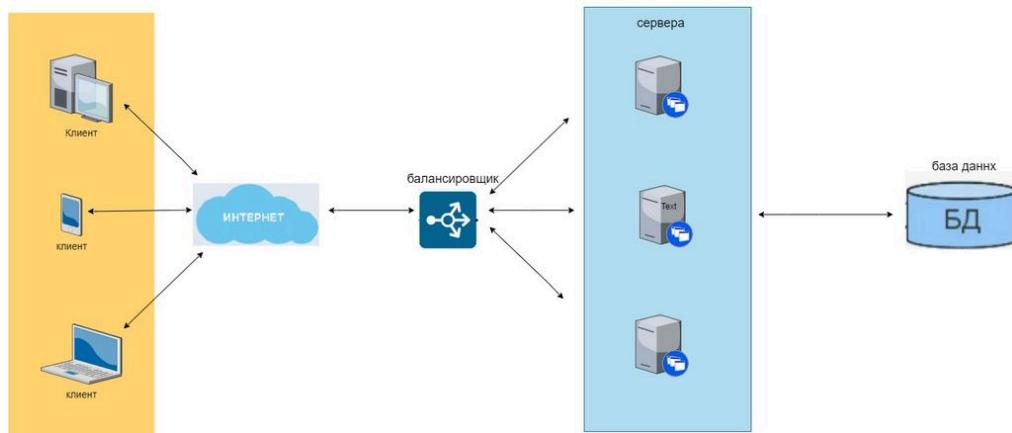
Bash команды:

1. pwd – выводит текущий рабочий каталог
2. cd Final-Project_QA/ - перейти в Final-Project_QA/
3. git checkout -b develop - создает новую ветку develop и переходит на нее
4. git branch - посмотреть на какой ветке
5. mkdir Bash-scripts - создать новую папку Bash-scripts
6. cd Bash-scripts - перейти в папку Bash-scripts
7. touch Bash-scripts.sh - создать новый файл Bash-scripts.sh
8. ls - проверить наличие файла
9. vim Bash-scripts.sh - перейти в редактор vim
10. echo - выводит текст
11. read - читает строку и использует для присвоения значений переменной
12. sh Bash-scripts.sh - запускает скрипт Bash-scripts.sh
13. git add - добавляет изменения в Гитхаб
14. git commit -m - создает коммит с комментарием
15. git push origin develop - передает изменения в Гитхаб
16. cd ~ - выход из прошлой директории
17. cd cd Final-Project_QA/ - перейти директорию Final-Project_QA/
18. mkdir Test - создает папку Test
19. touch Test1.txt - создает файл Test1.txt
20. mkdir Test2 - создает папку Test2
21. mv ~/Final-Project_QA/Test/Test1.txt ~/Final-Project_QA/Test2 - перемещает файл из папки Test в папку Test2
22. ls -a - показывает скрытые файлы
23. rm -r Test1.txt - удаляет файл Test1.txt

Клиент-серверная архитектура

client_server

Клиент -серверная архитектура



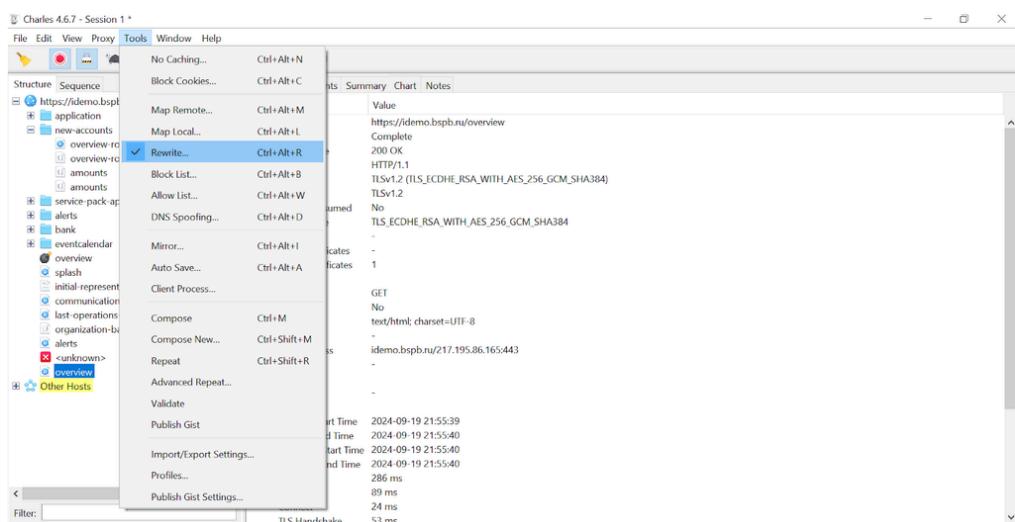
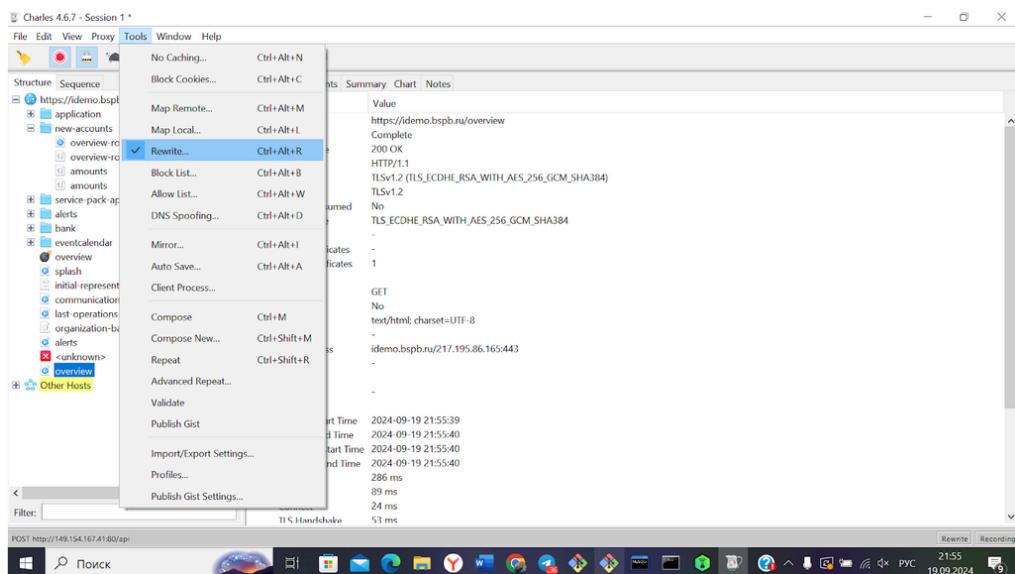
Charles

Rewrite, Breakpoint, Map Local

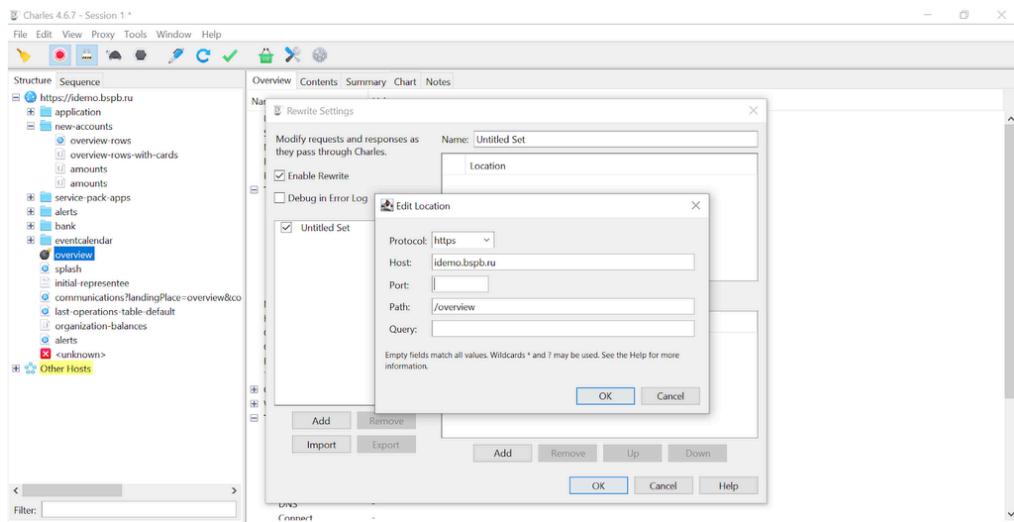
Charles

1. Подмена статус кода с помощью “Rewrite”.

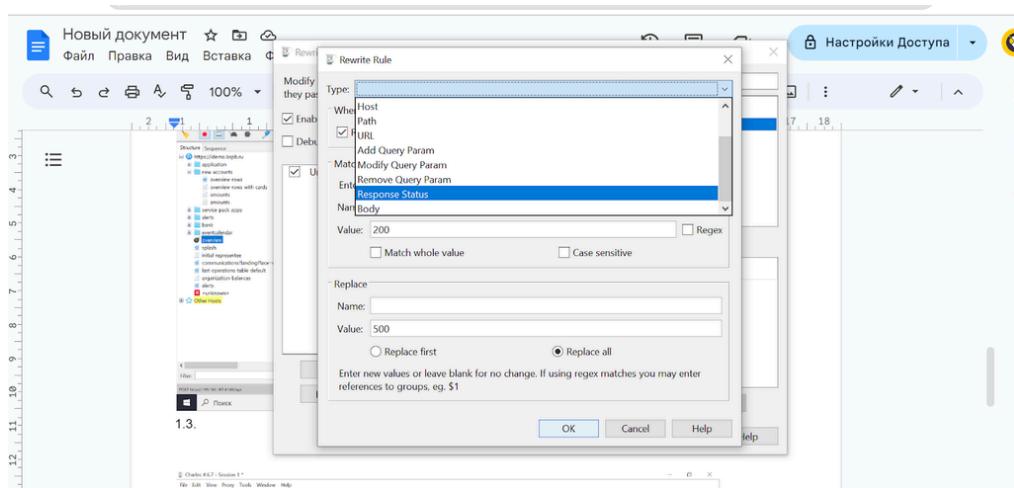
1.1. Открыть Tools, далее Rewrite.



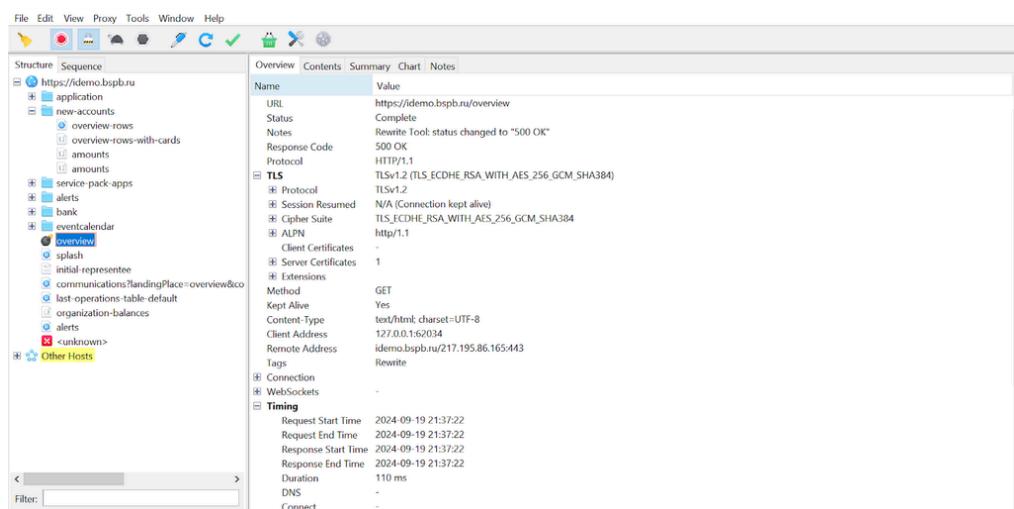
1.2. Поставить галочку «Enable Rewrite», затем в открывшемся окне под левым полем нажать «Add», а дальше нажать «Add» под правым верхним полем.



1.3. Указать тип редактирования запроса, поставить значения. В данном примере поменять статус код с 200 на 500. Далее нажать OK и еще раз OK.



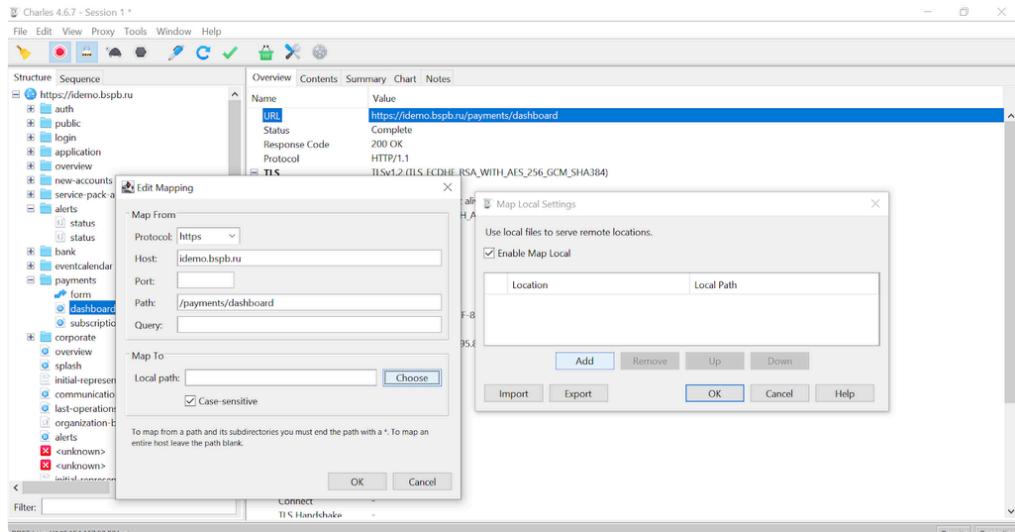
1.4. Обновить страницу веб-приложения с данным запросом. Получить изменение в статус-коде. Код поменялся на 500.



2. Подмена тела ответа с помощью “Map local”

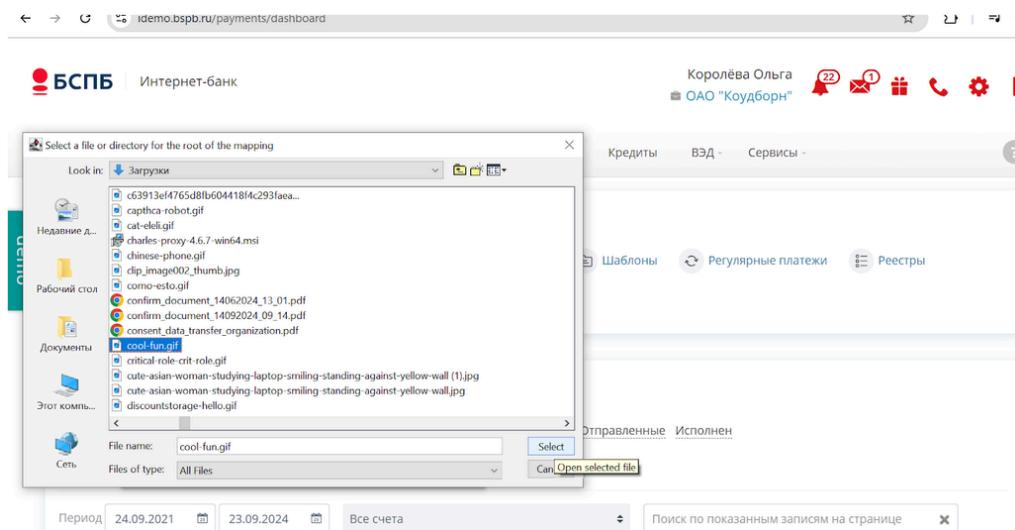
2.1. Открыть запрос на сайте банка СПб, выбрать Tools, далее Map local.

2.2. Далее указать host (Url запроса), указать путь.



2.3. Далее выбрать локальный путь. В примере выбрана гифка cool-fun.gif.

2.4. Нажать Select.



2.5. Вернуться на сайт банка СПб на страницу запроса и обновить её.

3. Подмена хедера с помощью функции “Breakpoint”

Данная функция позволяет точнее работать с запросами, вешая на них так называемый «Breakpoint» (точку остановки).

Ниже показано, как через настройки добавить запрос в список Breakpoint. После повторного получения указанного вами запроса в программе откроется отдельная сессия, в которой можно спокойно работать с ним.

3.1. Открыть страницу сайта, с которой будет работать.

Главная Счета - Платежи Зарплата - Карты - Эквайринг - Депозиты Кредиты ВЭД - Сервисы -

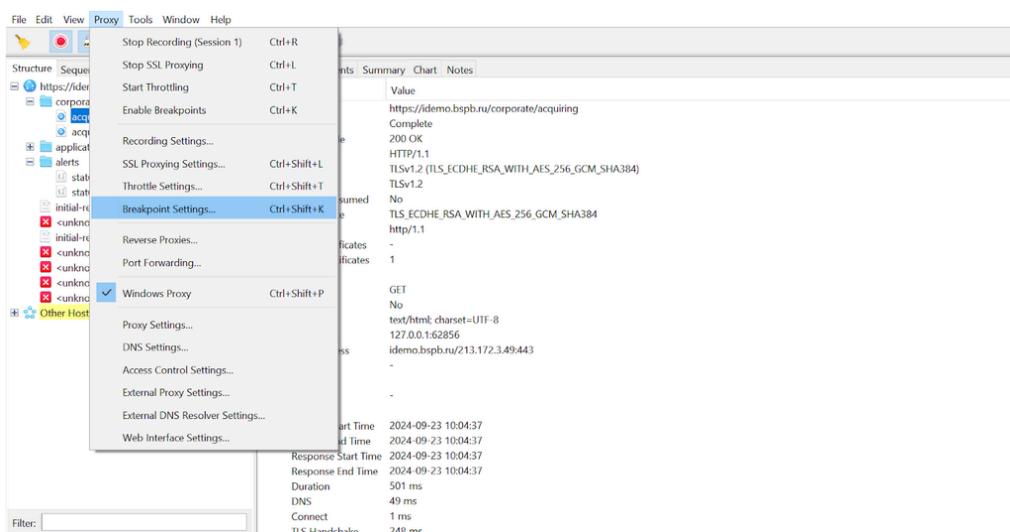
Задать вопрос об эквайринге

Эквайринг

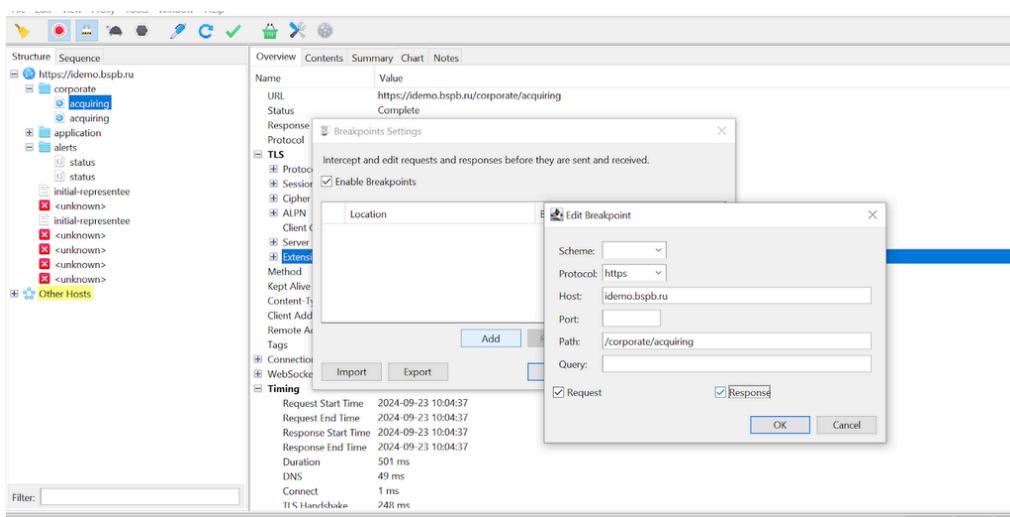
demo Договоры Терминалы Заявления История операций

Номер договора	Тип договора	Комиссия	Статус	Дата заключения	Количество терминалов	
08-1-04-2437 ООО "Джапан Мотор Сервис"	Интернет-эквайринг	2.1%	Подписан	20.07.2000	15	<button>Список терминалов</button>
08-1-01-0003 Магазин "Все и тапочки"	Торговый эквайринг	2.4%	Подписан	20.07.2012	15	<button>Список терминалов</button> <button>Подключить новый</button>
08-1-01-0019 Бар у Пальчика	Торговый эквайринг	2.4%	Подписан	20.07.2015	15	<button>Список терминалов</button> <button>Подключить новый</button>
08-1-01-0005 Магазин "Пятёрочка"	Торговый эквайринг	2.4%	Действует	20.07.2012	15	<button>Список терминалов</button> <button>Подключить новый</button>
08-1-01-0007 Магазин "Семёночка"	Торговый эквайринг	2.4%	Действует	20.07.2012	15	<button>Список терминалов</button> <button>Подключить новый</button>

3.2. Включить “Breakpoint Setting”.



3.3. Указать путь.



3.4. Обязательно указать метод (GET).



3.5. Включился Breakpoint.

The screenshot shows a browser's developer tools Network tab with a breakpoint set for a request to `https://idemo.bspb.ru/corporate/acquiring`. The 'Edit Request' dialog is open, displaying the following details:

Name	Value
URL	<code>https://idemo.bspb.ru/corporate/acquiring</code>
Status	Sent request. Waiting for response...
Response Code	-
Protocol	HTTP/1.1
TLS	TLSv1.2 (TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384)
Protocol	TLSv1.2
Session Resumed	No
Cipher Suite	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
ALPN	http/1.1
Client Certificates	-
Server Certificates	1
Extensions	GET
Method	No
Kept Alive	-
Content-Type	127.0.0.1:63840
Client Address	idemo.bspb.ru/213.172.3.49:443
Remote Address	Breakpoints
Tags	-
Connection	-
WebSockets	-
Origin	-
Version	-
Protocol	-
Extensions	-
Messages Sent	-

3.6. Можно поменять\добавить Header. В поле выбираем Edit Request “TEST”.

3.7. Замена статус-кода на 500 TEST. После обновления Response code поменялся на 500 TEST.



✓ Github Actions

1. Проверка доступности хоста “`https://www.ozon.ru`”.

```
name: CI
on:
  push:
    branches:
      - develop
jobs:
  check-host:
    runs-on: ubuntu-latest
    steps:
      - name: Check Host_Availability "https://www.ozon.ru"
        run: |
          if curl --output /dev/null --silent --head --fail "https://www.ozon.ru"; then
            echo "https://www.ozon.ru is reachable"
          else
            echo echo "https://www.ozon.ru is not reachable"
            exit 1
          fi
```

2. Запуск скрипта Bash-scripts.sh.

```
jobs:
  check-host:
    steps:
      - name: Check Host_Availability "https://www.ozon.ru"
        run: |
          if curl --output /dev/null --silent --head --fail "https://www.ozon.ru"; then
            echo "https://www.ozon.ru is reachable"
          else
            echo echo "https://www.ozon.ru is not reachable"
            exit 1
          fi

  run-script:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v2
      - name: Запуск скрипта
        run: chmod +x Bash-scripts/*.sh
```

Mariavesna01 / Final-Project_QA

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Triggered via push 4 minutes ago
Mariavesna01 pushed => 7ab44c3 develop

Status Success Total duration 11s Artifacts

blank.yml

Triggered via push 4 minutes ago
Mariavesna01 pushed => 7ab44c3 develop

check-host 2s

run-script 2s