

Веб-тестирование

Пример использования техник тест-дизайна. Таблица принятия решений.

Заявку на получение кредита может подать заемщик старше 25 лет. Если он уже является клиентом банка, его заявка будет рассмотрена в индивидуальном порядке, и сумма кредита будет выдана выше. Если заемщик является клиентом банка, его заработная плата выше 25 тыс. руб. и он имеет в собственности какую-либо недвижимость, то он может рассчитывать на сумму кредита выше 1 000 000 руб. Если у заемщика-клиента банка заработная плата больше 25 тыс. руб., но в собственности нет недвижимости, то он может получить кредит в размере от 500 000 до 1 000 000 руб. Если заработная плата заемщика-клиента банка ниже 25 тыс. руб., но в его собственности есть недвижимость, то он может рассчитывать на кредит в сумме от 100 000 до 500 000 руб. Если заработная плата заемщика-клиента банка составляет меньше 25 тыс. руб. и у него в собственности нет недвижимости, то максимальная сумма кредита, на которую он может рассчитывать – 100 000 руб. Если заемщик не является клиентом банка, имеет доход выше 25 тыс. руб. и недвижимость в собственности, то ему может быть выдан кредит в размере от 500 000 до 1 000 000 руб. Если заемщик – не клиент банка, его заработная плата ниже 25 тыс. руб. и в собственности есть недвижимость, то сумма кредита – до 100 000 руб. Если заемщик не клиент банка, имеет заработную плату выше 25 тыс. руб., но не имеет недвижимости в собственности, то сумма кредита составит от 100 000 до 500 000 тыс. руб. Если заемщик не клиент банка с заработной платой меньше 25 тыс. руб. и без недвижимости – в кредите ему должно быть отказано.

	1 вариант	2 вариант	3 вариант	4 вариант	5 вариант	6 вариант	7 вариант	8 вариант
Старше 25 лет	+	+	+	+	+	+	+	+
Клиент банка	+	+	+	+	-	-	-	-
Зп выше 25 тыс/мес	+	+	-	-	+	-	+	-
Недвижимость в собственности	+	-	+	-	+	+	-	-
В рублях	<1 000 000	500 000 - 1 000 000	100 000 - 500 000	Max 100 000	500 000 - 1 000 000	Max 100 000	100 000 - 500 000	Отказать в заеме

Пример баг-репорта, открытый тестовый стенд

[BUG] Система принимает неверный логин и продолжает вход

Серьезность / Критичность: Высокая (High)

Приоритет: Высокий (P1)

Тестовый стенд / Окружение:

- Ссылка: <https://idemo.bspb.ru/welcome>
- ОС: **iOS**
- Браузер: Chrome, версия 133.0.6943.128
- Устройство: **MacBook Air**

Шаги воспроизведения:

1. Открыть сайт <https://idemo.bspb.ru/welcome>.
2. В поле **Логин** ввести: **demo** (некорректное значение).
3. В поле **Пароль** ввести: **demogh** (некорректное значение).
4. Нажать кнопку входа.

Ожидаемый результат (ОР):

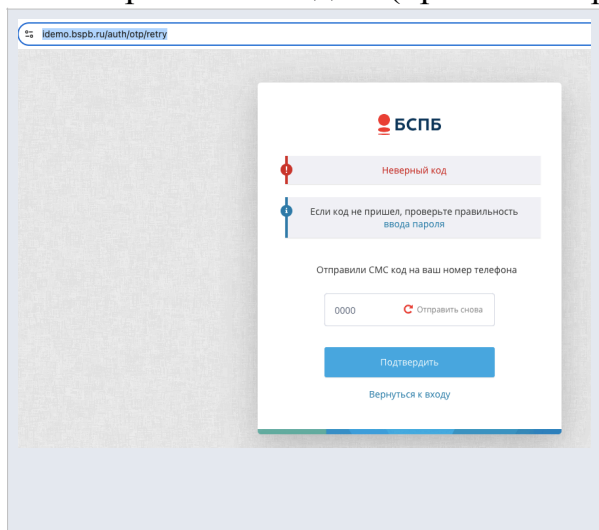
- Появляется информационное сообщение: **“Неверно введен логин и/или пароль”**.
- Система **не** позволяет продолжить вход.

Фактический результат (ФР):

- Система **принимает неверный логин** и продолжает вход, несмотря на некорректные данные.

Дополнительная информация:

- Тестовые данные:
 - Логин: **demo**
 - Пароль: **demogh**
- Скриншот / видео: (приложите файл или ссылку)



- Логи

API

Коллекция Postman: <https://.postman.co/workspace/My-Workspace~dffd081c-b865-4f6f-8501-59ce344aa29b/collection/41481443-37ed2a58-76bc-497c-8731-98ec6dfb894c?action=share&creator=41481443>

Тестовый стенд: книжный магазин

Отчет по тестированию API Booking

Дата тестирования: 7.02.2025

Версия API: v2

Количество запросов: 7

1. Обзор коллекции

Коллекция охватывает полный CRUD-цикл работы с бронированиями:

- Создание токена аутентификации
- Создание/чтение/обновление/удаление бронирований
- Визуализация данных ответов

2. Детальный анализ эндпоинтов

2.1 Аутентификация

POST /auth

- Назначение: Получение токена для авторизованных операций
- Тело запроса: username/password
- Особенности: Используется в других запросах через Cookie

2.2 Операции с бронированиями

Метод	Endpoint	Назначение	Особенности
-------	----------	------------	-------------

GET	/booking	Получение списка ID бронирований	Визуализация "No Bookings Available"
-----	----------	----------------------------------	--------------------------------------

GET	/booking/{id}	Получение деталей бронирования	Красивая табличная визуализация данных
-----	---------------	--------------------------------	--

POST	/booking	Создание бронирования	Пример тела: данные клиента, даты, дополнительные услуги
------	----------	-----------------------	--

PUT	/booking/{id}	Полное обновление бронирования	Требует токена в Cookie
-----	---------------	--------------------------------	-------------------------

PATCH	/booking/{id}	Частичное обновление	Изменение только имени/фамилии
-------	---------------	----------------------	--------------------------------

DELETE	/booking/{id}	Удаление бронирования	Требует авторизации
--------	---------------	-----------------------	---------------------

3. Примеры запросов

Создание бронирования (POST):

```
json
```

```
{  
  "firstname": "Amy",  
  "lastname": "Green",  
}
```

```
"totalprice": 122,  
"depositpaid": true,  
"bookingdates": {  
  "checkin": "2025-02-23",  
  "checkout": "2025-10-23"  
},  
"additionalneeds": "Breakfast"  
}  
Частичное обновление (PATCH):  
json
```

```
{  
  "firstname": "Alise",  
  "lastname": "Green"  
}
```

4. Визуализация данных

Коллекция содержит скрипты для отображения данных в табличном формате:

- Стилизованные таблицы CSS
- Условное форматирование (например, для depositpaid)
- Ховер-эффекты для строк

5. Рекомендации по тестированию

1. Позитивные тесты:

- Все CRUD-операции с валидными данными
- Проверка граничных значений дат
- Проверка разных additionalneeds

2. Негативные тесты:

- Попытки операций без токена
- Невалидные форматы дат
- Отрицательные значения price
- Попытки изменить несуществующее бронирование

3. Дополнительные проверки:

- Валидация структуры ответов
- Проверка времени ответа
- Тестирование параллельных запросов

6. Статистика коллекции

- GET: 2 запроса
- POST: 2 запроса
- PUT: 1 запрос
- PATCH: 1 запрос
- DELETE: 1 запрос