Задание 1.

Решение:

1. Открыть консоль разработчика (F12).

2. Перезагрузить страницу.

3. Заполнить все поля валидными данными.

4. Нажать кнопку «Войти»/

На что можно обратить внимание далее:

1. Если Network-элемент не был создан вовсе (т.е. Не произошло запроса на бэкенд) - баг на фронтенде.

2. Если Network-элемент был создан и ответ сервера:

2.1) 400: с фронтенда уходит невалидное тело на бэк;

2.2) 404: с фронтенда уходит запрос на невалидный url;

2.3) 500: ошибка на бэке (не 100% на бэке, тк мы можем отправлять просто невалидный серверу запрос, который он не умеет обрабатывать, но все же) - идем смотреть логи сервиса. Если докер: docker logs -f service\_name , если это демон: journalctl -xefu service\_name ; проверка статуса активности БД; 2.4) 403: у нас нет доступа к авторизации на данном сервисе, нужно смотреть логи сервиса авторизации;

2.5) Если запрос есть , но никакого статус-кода нет - у нас где-то на уровне прокси / днс-сервера / где-либо еще на сетевом уровне проблема.

3. Если Network-элемент создан и статус код - ОК , но все равно ничего не поменялось - посмотреть на вкладке console, есть ли ошибки исполнения js кода.

4. Если и тут все ок - проверить, активен ли у нас тротлинг в консоли (хотя в таком случае у нас бы даже страница не загрузилась).

Задание 2.

Решение:

**1**. Тест-кейс объясняющий порядок тестирования системы:

Pre\_conditions:

1. Установить Postman (по ТЗ можем имитировать действия сервиса BH1 – т.е. слать http-запрос в модули).

2. Активен эмулятор логики системы BH2.

3. Доступна и активна БД интеграционной шины.

4. Модули (1, 2, 3), отвечающие за валидацию, логгированием отправку данных, активны.

Test\_Data:

1. Создано валидное тело в формате XML\_1 для отправки из BH1 (из postman)

(Условие: нам известно, какие данные мы слать можем и мы априори способны отделять валидные тела для от невалидных при отправке).

2. Создано невалидное (отсутствует какой-либо ключ или добавлен лишний) тело XML\_1.

Подробно тест-кейс:

|  |  |
| --- | --- |
| Шаги | Ожидаемы результат |
| 1. Отправить тело XML\_1 из подготовленных данных на эндпойнт | Ответ сервера: 200 ОК, тело ответа в формате XML\_1 |
| 2. Вызвать лог модуля 1 | В логах шины модуля 1 есть информация о получении XML\_1 ; (возможно есть информация об успешной конвертации) ; есть информация об отправке тела в модуль 2 |
| 3. Вызвать лог модуля 2 | В логах шины модуля 2 присутствует преобразованное из XML\_1 в XML\_2 тело запроса ; Есть информация об успешной отправке в БД и в модуль 3 |
| 4. Запрос в БД: select <id\_name> from journal | БД вернула строку с ID из запроса (sending\_status: Success) |
| 5. Вызов логов модуля 3 | В логах шины модуля 3 присутствует преобразованное из XML\_1 в XML\_2 тело запроса ; (Возможно есть информация о конвертации в формат XML\_3), также есть информация об отправке тела в систему BH2 |
| 6. Запрос логов у эмулятора BH2 | Эмулятор BH2 вернул логи, в которых можно найти успешно принятое тело из шага 6 и сообщение об отправке ответа в модуль 3 в формате XML\_3 |
| 7. Запрос в БД логов от системы BH2: select <id\_message> from status\_messages | БД вернула строку с ID из запроса (status\_message: Success) |

**2**. Возможные тестовые сценарии, представленные в чек-листе:

Чек-лист:

1. Отправка валидного тела запроса BH1 -> BH2 (здесь возможно несколько вариаций бизнес-сценариев).

2. Отправка невалидного тела запроса BH1-> BH2.

3. Отправка данных не XML формата.

4. Отправка валидного тела запроса при отключенном эмуляторе BH2.

5. Отправка валидного тела запроса при отключенном одном из модулей интеграционной шины.

6. Отправка валидного тела запроса при недоступной БД, контактирующей с модулем 2.

7. Отправка модулю 1 XML\_2.

8. Отправка модулю 1 XML\_3.