# Экспериментальная часть

Для проверки корректности выполненной работы были проведены следующие виды тестирования:

1. тестирование корректности работы WEB API методом черного ящика;
2. модульное тестирование;
3. кроссбраузерное тестирование интерфейсов пользовательской части.

## Тестирование корректности работы WEB API методом черного ящика

Тестирование методом черного ящика – это функциональное и нефункциональное тестирование без доступа к внутренней структуре компонентов системы. Метод тестирования «черного ящика» – процедура получения и выбора тестовых случаев на основе анализа спецификации (функциональной или нефункциональной), компонентов или системы без ссылки на их внутреннее устройство.

В рамках данного тестирования были проверены следующие контроллеры WEB API:

1. messages;
2. users;
3. contacts;
4. files;
5. delivery-services;
6. delivery-queues;
7. accesses;
8. sales.

Сигнатуры и результаты тестов представлены в таблице 1.1.

Таблица 1.1.

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Входные данные | Реальный результат | Ожидаемый результат |

## Модульное тестирование

Модульное тестирование, или юнит-тестирование (англ. unit testing) — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы.

Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

В рамках модульного тестирования были созданы новые проекты DeliveryRely.ServiceLayer.Tests и DeliveryRely.Domain.Tests.

В проекте Delivery.Domain.ServiceLayer.Tests содержатся тесты для методов класса Hash и других вспомогательных классов, таких как AutoMapper. Код тестов представлен в листинге 1.

В проекте DeliveryRely.Domain.Tests содержатся тесты для методов получения (методы Get…), добавления (методы Insert…), изменения (методы Update…) и удаления (методы Delete…) класса EFRepository. Код тестов представлен в листинге 2.

## Кроссбраузерное тестирование интерфейсов пользовательской части

Тестирование кроссбраузерности — вид тестирования, направленный на поддержку и правильное полное отображение программного продукта в разных браузерах, мобильных устройствах, планшетах, экранах различного размера.

Для упрощения проведения кроссбраузерного тестирования использовалось стороннее ПО Xenocode Browser Sandbox. Эта программа позволяет работать практически в любом браузере без всякой установки.

Список проверяемых браузеров:

1. Google Chrome 89.0;
2. Safari 14;
3. Mozilla Firefox 91.6 ESR;
4. Opera 12.

В каждом из указанных браузеров были открыты следующие интерфейсы:

1. страница авторизации;
2. главная страница;
3. список сервисов доставки;
4. список сообщений;
5. список контактов;
6. страница создания сообщения.

### Google Chrome

Внешний вид интерфейсов в Google Chrome 89.0 представлен на рис. 1.1 – 1.6.

***Рис. 1.1. Страница авторизации в Google Chrome***

***Рис. 1.2. Главная страница в Google Chrome***

***Рис. 1.3. Список сервисов доставки в Google Chrome***

***Рис. 1.4. Список сообщений в Google Chrome***

***Рис. 1.5. Список контактов в Google Chrome***

***Рис. 1.6. Страница создания сообщения в Google Chrome***

### Safari

Внешний вид интерфейсов в Safari 14 представлен на рис. 1.7 – 1.12.

***Рис. 1.7. Страница авторизации в Safari***

***Рис. 1.8. Главная страница в Safari***

***Рис. 1.9. Список сервисов доставки в Safari***

***Рис. 1.10. Список сообщений в Safari***

***Рис. 1.11. Список контактов в Safari***

***Рис. 1.12. Страница создания сообщения в Safari***

### Mozilla Firefox

Внешний вид интерфейсов в Mozilla Firefox 91.6 ESR представлен на рис. 1.13 – 1.18.

***Рис. 1.13. Страница авторизации в Safari***

***Рис. 1.14. Главная страница в Mozilla Firefox***

***Рис. 1.15. Список сервисов доставки в Mozilla Firefox***

***Рис. 1.16. Список сообщений в Mozilla Firefox***

***Рис. 1.17. Список контактов в Mozilla Firefox***

***Рис. 1.18. Страница создания сообщения в Mozilla Firefox***

### Opera

Внешний вид интерфейсов в Opera 12 представлен на рис. 1.19 – 1.24.

***Рис. 1.19. Страница авторизации в Opera***

***Рис. 1.20. Главная страница в Opera***

***Рис. 1.21. Список сервисов доставки в Opera***

***Рис. 1.22. Список сообщений в Opera***

***Рис. 1.23. Список контактов в Opera***

***Рис. 1.24. Страница создания сообщения в Opera***