**Модульные тесты (тест бэка, где у нас один контроллер с одним экшеном)**

**Группа 1.1 (валидация входных данных):**

1. Если *card\_number* содержит не 16 цифр (регулярка совпадение - /^[0-9]{13}$/), возвращает код 400, сообщение об ошибке содержит “Неверно заполнено поле номер карты”
2. Если *card\_number* не передано, возвращает код 400, сообщение об ошибке содержит “Неверно заполнено поле номер карты”
3. Если *card\_holder* содержит символы, кроме латинских букв, дефисов и пробела между ними (регулярка совпадение - /^([A-Za-z-]{3, })\s([A-Za-z-]{3, })$/), возвращает код 400, сообщение об ошибке содержит “Неверно заполнено поле владелец”
4. Если *card\_holder* не передано, возвращает код 400, сообщение об ошибке содержит “Неверно заполнено поле владелец”
5. Если *card\_expiration* не удовлетворяет условиям регулярки /^(0[1-9]|1[0-2])\/?([0-9]{2})$/, возвращает код 400, сообщение об ошибке содержит “Неверно заполнено поле действительно до”
6. Если *card\_expiration* не передано, возвращает код 400, сообщение об ошибке содержит “Неверно заполнено поле действительно до”
7. Если *cvv* содержит не 3 цифры (^[0-9]{3}$/), возвращает код 400, сообщение об ошибке содержит “Неверно заполнено поле CVV”
8. Если *cvv* не передано, возвращает код 400, сообщение об ошибке содержит “Неверно заполнено поле CVV”
9. Если *order\_number* содержит более 16 символов (не выполняется условие mb\_strlen(trim($*order\_number)*) <= 16), возвращает код 400, сообщение об ошибке содержит “Неверно заполнено поле номер заказа”
10. Если *order\_number* не передано, возвращает код 400, сообщение об ошибке содержит “Неверно заполнено поле номер заказа”
11. Если *sum* не является целым числом или числом с дробной частью, отделенной запятой (регулярка - ?([0-9]\*[,])?[0-9]+), возвращает код 400, сообщение об ошибке содержит “Неверно заполнено поле сумма заказа”
12. Если *sum* не передано, сообщение об ошибке содержит “Неверно заполнено поле сумма заказа”

**Группа 1.2 (Связь с сервисом А и репозиторием):**

**!!!Важно!!! Запускать тесты группы 2 только при 100% прохождении тестов группы 1.**

1. Все поля переданы корректно (берем условия из 1.1.1 - 1.1.12), подменяем сервис А (фейк 1 - тот, который не может списать деньги и возвращает 403), возвращает код 403, сообщение об ошибке содержит “Ошибка оплаты”
2. Все поля переданы корректно (берем условия из 1.1.1 - 1.1.12), подменяем сервис А (фейк 2 - тот, который может списать деньги и возвращает 200). Подменяем репозиторий (фейк 1 - всегда возвращает код 400 и ошибку “Оплата прошла, неверная сумма, обратитесь к продавцу”). Возвращает код 400, содержит “Оплата прошла, неверная сумма, обратитесь к продавцу”.
3. Все поля переданы корректно (берем условия из 1.1.1 - 1.1.12), подменяем сервис А (фейк 2 - тот, который может списать деньги и возвращает 200). Подменяем репозиторий (фейк 2 - всегда возвращает код 200 и сообщение “Оплата успешно прошла”). Возвращает код 200, содержит “Оплата успешно прошла”.

**Интеграционные тесты**

**Группа 2.1 (фронт-бэк)**

1. Если card\_number содержит не 16 цифр (регулярка совпадение - /^[0-9]{13}$/), после получения ответа от бэка рамка вокруг поля “Номер карты” становится красной.
2. Если *card\_number* не передано, после получения ответа от бэка рамка вокруг поля “Номер карты” становится красной.
3. Если *card\_holder* содержит символы, кроме латинских букв, дефисов и пробела между ними (регулярка совпадение - /^([A-Za-z-]{3, })\s([A-Za-z-]{3, })$/), после получения ответа от бэка рамка вокруг поля “Владелец” становится красной.
4. Если *card\_holder* не передано, после получения ответа от бэка рамка вокруг поля “Владелец” становится красной.
5. Если *card\_expiration* не удовлетворяет условиям регулярки /^(0[1-9]|1[0-2])\/?([0-9]{2})$/, после получения ответа от бэка рамка вокруг поля “действительно до” становится красной.
6. Если *card\_expiration* не передано, после получения ответа от бэка рамка вокруг поля “действительно до” становится красной.
7. Если *cvv* содержит не 3 цифры (^[0-9]{3}$/), после получения ответа от бэка рамка вокруг поля “CVV” становится красной.
8. Если *cvv* не передано, после получения ответа от бэка рамка вокруг поля “CVV” становится красной.
9. Если *order\_number* содержит более 16 символов (не выполняется условие mb\_strlen(trim($*order\_number)*) <= 16), после получения ответа от бэка рамка вокруг поля “Номер заказа” становится красной.
10. Если *order\_number* не передано, после получения ответа от бэка рамка вокруг поля “Номер заказа” становится красной.
11. Если *sum* не является целым числом или числом с дробной частью, отделенной запятой (регулярка - ?([0-9]\*[,])?[0-9]+), после получения ответа от бэка рамка вокруг поля “Сумма заказа” становится красной.
12. Если *sum* не передано, после получения ответа от бэка рамка вокруг поля “Сумма заказа” становится красной.
13. Подменяем на бэке сервис А (фейк 1 - тот, который не может списать деньги и возвращает 403). Если заполнили все поля верно (см. условия 2.1.1 - 2.1.12), после получения ответа от бэка выдается всплывающее окно “Ошибка оплаты”.
14. Подменяем на бэке сервис А (фейк 2 - тот, который может списать деньги и возвращает 200), ставим сумму заказа, передаваемую в репозиторий не равной, приходящей с фронта. Если заполнили все поля верно (см. условия 2.1.1 - 2.1.12), после получения ответа от бэка выдается всплывающее окно “Оплата прошла, неверная сумма, обратитесь к продавцу”.
15. Подменяем на бэке сервис А (фейк 2 - тот, который может списать деньги и возвращает 200). Если заполнили все поля верно (см. условия 2.1.1 - 2.1.12), после получения ответа от бэка выдается всплывающее окно “Оплата прошла успешно!” и нас редиректит на страницу оплаченного заказа.

**Группа 2.2 (бэк-сервис А)**

1. Устанавливаем адрес сервиса А (тестовый стенд партнера для интеграционного тестирования). Пингуем их сервис, ожидаем получить ответ, что пакеты прошли.
2. Вызываем метод бека валидными данными, удовлетворяющие условиям 1.1.1 - 1.1.12, но с параметром *card\_expiration* 01/01. Ожидаем код 403 и сообщение об ошибке, содержащее “Некорректные данные карты”.
3. Вызываем метод бека валидными данными, удовлетворяющие условиям 1.1.1 - 1.1.12, с реальными данными какой-нибудь карты, у которой по данным партнера (владельца сервиса А) есть баланс >= суммы покупке (баланс в их контуре для интеграционного тестирования). Ожидаем код 200 и ответ, содержащий “Оплата прошла успешно”.

**Группа 2.3 (бэк-репозиторий)**

1. Все поля переданы корректно (берем условия из 1.1.1 - 1.1.12), подменяем сервис А (фейк 2 - тот, который может списать деньги и возвращает 200), подменяем сумму, передаваемую в репозиторий на меньшую, чем сумма заказа. Возвращает код 400, содержит “Оплата прошла, неверная сумма, обратитесь к продавцу”.
2. Все поля переданы корректно (берем условия из 1.1.1 - 1.1.12), подменяем сервис А (фейк 2 - тот, который может списать деньги и возвращает 200), подменяем сумму, передаваемую в репозиторий на большую, чем сумма заказа. Возвращает код 400, содержит “Оплата прошла, неверная сумма, обратитесь к продавцу”.
3. Все поля переданы корректно (берем условия из 1.1.1 - 1.1.12), подменяем сервис А (фейк 2 - тот, который может списать деньги и возвращает 200), подменяем сумму, передаваемую в репозиторий на 0,00. Возвращает код 400, содержит “Оплата прошла, неверная сумма, обратитесь к продавцу”.
4. Все поля переданы корректно (берем условия из 1.1.1 - 1.1.12), подменяем сервис А (фейк 2 - тот, который может списать деньги и возвращает 200). Возвращает код 200, содержит “Оплата успешно прошла”.

**Системные тесты**

**Группа 3.1**

1. Настроили свою интеграционную тестовую среду, проверили доступность партнеров (сервис А, репозиторий), получили ответ от фронта, что он тоже вас видит.
2. Открыли фронт, попробовали отправить пустую форму. Получили красные рамки вокруг каждого из полей.
3. Ввели везде неверные данные (не удовлетворяет условиям 1.1.1 - 1.1.12). Попробовали отправить. Получили красные рамки вокруг каждого из полей.
4. Ввели все данные корректно (с точки зрения 1.1.1 - 1.1.12), но это данные несуществующей карты (с точки зрения тестового контура сервиса А). Получили сообщение “Некорректные данные карты”.
5. Ввели все данные корректно (с точки зрения 1.1.1 - 1.1.12), но ввели несуществующий номер заказа (с точки зрения тестового контура репозитория), получили сообщение “Оплата прошла, неверная сумма, обратитесь к продавцу”.
6. Ввели все данные корректно (с точки зрения 1.1.1 - 1.1.12), карта существует, баланс на ней >= сумме заказа, номер заказа верный, сумма заказа совпадает с суммой заказа в репозитории. Попробовали отправить. Получаем всплывающее окно “Оплата прошла успешно!” и нас редиректит на страницу оплаченного заказа.