

Задание 2.

Баг-репорт: Неверный HTTP-статус при обновлении несуществующего ресурса

1. Идентификационная информация:

- **ID проекта:** Go REST API
- **Приоритет:** Средний (Medium)
- **Серьезность:** Средняя (Major)
- **Окружение:** master ветка
- **Тестер:** Сизоненко Валерия
- **Дата:** 05.10.2025

2. Краткое описание (Summary):

Метод обновления продукта возвращает статус `200 OK` при попытке обновить несуществующий продукт, что вводит в заблуждение. Ожидается статус `404 Not Found`.

3. Описание (Description):

Эндпоинт `PUT` обновления продукта некорректно обрабатывает запросы к несуществующим продуктам по id. Вместо того чтобы сообщить клиенту, что запрашиваемый ресурс не найден, сервер возвращает успешный статус `200 OK`. При этом никаких изменений в системе не происходит — продукт не создается и не обновляется, что нарушает принцип идемпотентности `PUT`.

Ожидаемый результат:

Система должна возвращать ошибку `404 Not Found`, когда клиент пытается обновить ресурс с идентификатором, который не существует в системе.

- **HTTP Status Code:** `404 Not Found`
- **Response Body:** Четкое сообщение об ошибке (например, `{"error": "Product not found"}`).

Фактический результат:

Система возвращает статус `200 OK`, создавая ложное впечатление об успешном выполнении операции, хотя на самом деле запрашиваемый продукт не был найден и никаких действий не произведено.

- **HTTP Status Code:** 200 OK
- **Response Body:** С переданными данными

4. Шаги для воспроизведения (Steps to Reproduce):

1. Убедится, что продукт с ID 99999 не существует в системе (например, через GET /api/product/99999).
2. Открыть инструмент для тестирования API (например, Postman).
3. Выполнить PUT запрос на эндпоинт: http://localhost:8010/api/product/99999
4. Установить Headers: Content-Type: application/json
5. В Body передать любой валидный JSON для обновления, например:

```
1 {  
2   "name": "Обновленное название",  
3   "price": 2000  
4 }
```

6. Отправить запрос.
7. Попробовать получить обновленный продукт через GET /api/product/99999 , чтобы убедиться, что он не был создан.

Баг-репорт: Отсутствие валидации обязательного поля price в методе создания продукта

1. Идентификационная информация:

- **ID проекта:** Go REST API
- **Приоритет:** Высокий (High)
- **Серьезность:** Средняя (Major)
- **Окружение:** master
- **Тестер:** Сизоненко Валерия
- **Дата:** 05.10.2025

2. Краткое описание (Summary):

Метод POST /api/product создает продукт со значением цены 0 , если обязательное поле price не передано в запросе, вместо возврата ошибки валидации.

3. Описание (Description):

Эндпоинт создания продукта не выполняет проверку на наличие обязательного поля `price` в теле запроса. При его отсутствии система некорректно присваивает значение по умолчанию `0`, что позволяет создавать продукты с нулевой стоимостью. Это нарушает бизнес-логику, так как цена продукта является критически важным атрибутом.

Ожидаемый результат:

Система должна отклонять запросы на создание продукта с отсутствующим полем `price`. Должен возвращаться ответ с ошибкой валидации (HTTP 400 Bad Request) и четким сообщением о том, какое именно поле является обязательным.

- **HTTP Status Code:** `400 Bad Request`
- **Response Body:** Сообщение об ошибке валидации (например, `{"error": "Field 'price' is required"}` или `Или другая ошибка о том, что поле обязательное`).

Фактический результат:

Система принимает запрос без поля `price`, создает запись в базе данных со значением `0` и возвращает успешный ответ.

- **HTTP Status Code:** `201 Created`
- **Response Body:** Содержит объект созданного продукта, где `price: 0`.

```
1 {  
2   "id": 123,  
3   "name": "Тестовый продукт",  
4   "price": 0  
5 }
```

4. Шаги для воспроизведения (Steps to Reproduce):

1. Открыть инструмент для тестирования API (например, Postman).
2. Выполнить **POST** запрос на эндпоинт: `http://localhost:8010/api/product`
3. Установить **Headers**: `Content-Type: application/json`
4. В **Body** передать следующий JSON, намеренно опустив поле `price`:

```
1 {
```

```
2  "name": "Тестовый продукт"
3  }
```

5. Отправить запрос.

Баг-репорт: Отсутствие валидации обязательного поля `name` в методе создания продукта

1. Идентификационная информация:

- **ID проекта:** Go REST API
- **Приоритет:** Высокий (High)
- **Серьезность:** Средняя (Major)
- **Окружение:** master
- **Тестер:** Сизоненко Валерия
- **Дата:** 05.10.2025

2. Краткое описание (Summary):

Метод `POST /api/product` создает продукт с пустым значением названия, если обязательное поле `name` не передано в запросе, вместо возврата ошибки валидации.

3. Описание (Description):

Эндпоинт создания продукта не выполняет проверку на наличие обязательного поля `name` в теле запроса. При его отсутствии система некорректно присваивает пустое значение (`null` или пустую строку), что позволяет создавать продукты без названия. Это нарушает целостность данных и бизнес-логику, так как название продукта является фундаментальным атрибутом для его идентификации.

Ожидаемый результат:

Система должна отклонять запросы на создание продукта с отсутствующим полем `name`. Должен возвращаться ответ с ошибкой валидации (HTTP 400 Bad Request) и четким сообщением о том, что поле `name` является обязательным.

- **HTTP Status Code:** `400 Bad Request`
- **Response Body:** Сообщение об ошибке валидации (например, `{"error": "Field 'name' is required"}` или `Или другая ошибка с описанием, что поле обязательное`).

Фактический результат:

Система принимает запрос без поля `name`, создает запись в базе данных с пустым названием и возвращает успешный ответ.

- **HTTP Status Code:** `201 Created`
- **Response Body:** Содержит объект созданного продукта, где `name` является `""` (пустая строка).

```
1 {  
2   "id": 124,  
3   "name": "",  
4   "price": 1500  
5 }
```

4. Шаги для воспроизведения (Steps to Reproduce):

1. Открыть инструмент для тестирования API (например, Postman).
2. Выполнить **POST** запрос на эндпоинт: `http://localhost:8010/api/product`
3. Установить **Headers**: `Content-Type: application/json`
4. В **Body** передать следующий JSON, намеренно опустив поле `name`:

```
1 {  
2   "price": 1500  
3 }
```

5. Отправить запрос.

Баг-репорт: Неверный HTTP-статус при обновлении несуществующего ресурса

1. Идентификационная информация:

- **ID проекта:** Go REST API
- **Приоритет:** Средний (Medium)
- **Серьезность:** Средняя (Major)
- **Окружение:** master ветка

- **Тестер:** Сизоненко Валерия

- **Дата:** 05.10.2025

2. Краткое описание (Summary):

Метод обновления продукта возвращает статус **200 OK** при попытке обновить несуществующий продукт, что вводит в заблуждение. Ожидается статус **404 Not Found**.

3. Описание (Description):

Эндпоинт обновления продукта (предположительно **PUT** или **PATCH**) обрабатывает запросы к несуществующим ID продуктов некорректно. Вместо того чтобы сообщить клиенту, что запрашиваемый ресурс не найден, сервер возвращает успешный статус **200 OK**. При этом никаких изменений в системе не происходит — продукт не создается и не обновляется, что нарушает принцип идиоматичности **PUT** и семантику HTTP-кодов.

Ожидаемый результат:

Система должна возвращать ошибку **404 Not Found**, когда клиент пытается обновить ресурс с идентификатором, который не существует в системе.

- **HTTP Status Code:** **404 Not Found**
- **Response Body:** Четкое сообщение об ошибке (например, **{"error": "Product not found"}**).

Фактический результат:

Система возвращает статус **200 OK**, создавая ложное впечатление об успешном выполнении операции, хотя на самом деле запрашиваемый продукт не был найден и никаких действий не произведено.

- **HTTP Status Code:** **200 OK**
- **Response Body:** Обновленные данные

4. Шаги для воспроизведения (Steps to Reproduce):

1. Убедитесь, что продукт с ID **99999** не существует в системе (например, через **GET** **/api/product/99999**).
2. Открыть инструмент для тестирования API (например, Postman).
3. Выполнить **PUT** (или **PATCH**) запрос на эндпоинт: **http://localhost:8010/api/product/99999**
4. Установить **Headers:** **Content-Type: application/json**

5. В **Body** передать любой валидный JSON для обновления, например:

```
▼  
1 {  
2   "name": "Обновленное название",  
3   "price": 2000  
4 }
```

6. Отправить запрос.

7. Попробовать получить обновленный продукт через `GET /api/product/99999`, чтобы убедиться, что он не был создан.