Обраний ендпойнт: GET /store/order/{orderId}

|  |  |  |  |
| --- | --- | --- | --- |
| № тесту | Назва тесту | Вхідні дані | Очікуваний результат |
| 1 | Отримання існуючого замовлення | orderId = 12345 (існуючий) | 200 OK, тіло відповіді містить id: 12345 |
| 2 | Отримання неіснуючого замовлення | orderId = 99999999 (не існує) | 404 Not Found |
| 3 | Запит без orderId | orderId не передано | 404 Not Found |
| 4 | Передача orderId як тексту | orderId = «orderId» | 404 Not Found або 400 Bad Request |
| 5 | Перевірка швидкості відповіді | orderId = 12345 | Відповідь приходить менше ніж за 1 секунду |
| 6 | Тестування максимально великого значення | orderId = 99999999999999 | 500 Internal Server Error або 400 Bad Request |
| 7 | Перевірка структури відповіді | orderId = 12345 | Всі поля присутні у відповіді |
| 8 | SQL-ін’єкція в поле orderId | orderId = "1 OR 1=1" | 404 Not Found або 400 Bad Request |
| 9 | Нестандартні символи в orderId | orderId = "@#$%^&\*()" | 404 Not Found або 400 Bad Request |
| 10 | Стрес-тест (1000 запитів) | orderId = 12345 | Жодних 500 Internal Server Error, усі відповіді < 1s |

**Test Case № 1**

**Title:** Успішне створення замовлення

**Description:** Перевірити, що API дозволяє створити замовлення, повертаючи коректну відповідь.

**Preconditions:**

* Сервер доступний.
* Користувач надсилає коректні дані.

Кроки до відтворення:

1.Надіслати POST-запит до {{host}}/store/order з наступними параметрами: json Copy Edit

**Expected Result:**

* Статус-код відповіді: **200 OK**.
* У відповіді міститься об’єкт замовлення з переданими даними.
* Замовлення збережене у базі даних.

### ****Test Case № 2****

**Title:** Отримання інформації про існуюче замовлення

**Description:** Переконатися, що API повертає коректні дані про замовлення за orderId.

**Preconditions:**

* У базі даних існує замовлення з orderId = {{orderId}}.

Кроки до відтворення:

1. Надіслати GET-запит до {{host}}/store/order/{{orderId}}.

**Expected Result:**

 Статус-код відповіді: **200 OK**.

 У тілі відповіді містяться дані замовлення (orderId, petId, status, quantity).

### ****Test Case № 3****

**Title:** Запит до неіснуючого orderId

**Description:** Переконатися, що API правильно обробляє запит до неіснуючого замовлення.

**Preconditions:**

* У базі немає замовлення з orderId = 999999.

Кроки для відтворення

1. Надіслати GET-запит до {{host}}/store/order/{{orderId}}.

**Expected Result:**

* Статус-код відповіді: **404 Not Found**.
* У відповіді міститься повідомлення Order not found.

### ****Test Case № 4****

**Title:** Видалення існуючого замовлення

**Description:** Переконатися, що API успішно видаляє замовлення за його orderId.

**Preconditions:**

* У базі даних існує замовлення з orderId = {{orderId}}.

**Steps:**

1. Надіслати DELETE-запит до {{host}}/store/order/{{orderId}}.

**Expected Result:**

* Статус-код відповіді: **200 OK**.
* Замовлення видалене з бази даних.

### ****Test Case № 5****

**Title:** Перевірка видалення замовлення

**Description:** Переконатися, що після видалення GET /store/order/{{orderId}} повертає 404.

**Preconditions:**

* Замовлення orderId = {{orderId}} було видалене в попередньому тесті.

**Steps:**

1. Надіслати GET-запит до {{host}}/store/order/{{orderId}}.

**Expected Result:**

* Статус-код відповіді: **404 Not Found**.
* Замовлення більше не існує в базі.

Начало формы

Конец формы