

# Как проверить CRUD операции

---

## Обзор

---

В проекте реализованы **CRUD операции** для всех 5 модулей:

- **Create** (Создание) - POST запросы ✓
- **Read** (Чтение) - GET запросы ✓
- **Update** (Обновление) - *будет добавлено при необходимости*
- **Delete** (Удаление) - *будет добавлено при необходимости*

**Текущая реализация:** Create и Read для всех модулей

## Способ 1: Через Swagger UI

---

### Шаг 1: Запустите сервер

```
cd backend  
python run.py
```

## **Шаг 2: Откройте Swagger UI**

Откройте в браузере: **http://localhost:8000/docs**

## **Шаг 3: Тестирование операций**

### **READ операции (GET)**

#### **1. Получить все записи:**

- Найдите нужный модуль (например, "Material Types")
- Раскройте GET /api/material-types
- Нажмите "Try it out" ◦ Нажмите "Execute" ◦ Проверьте ответ - должны вернуться все записи из БД

#### **2. Получить запись по ID:**

- Раскройте GET /api/material-types/{material\_type\_id}
- Нажмите "Try it out" ◦ Введите ID (например, 1) ◦ Нажмите "Execute"

### **CREATE операции (POST)**

#### **1. Создать новую запись:**

- Раскройте POST /api/material-types
- Нажмите "Try it out" ◦ Заполните форму JSON ◦ Нажмите "Execute"

## Способ 2: Через curl (Терминал)

---

### READ операции

```
# Получить все типы материалов
curl http://localhost:8000/api/material-types

# Получить тип материала по ID
curl http://localhost:8000/api/material-types/1

# Получить все типы продукции
curl http://localhost:8000/api/product-types

# Получить все цеха
curl http://localhost:8000/api/workshops

# Получить всю продукцию
curl http://localhost:8000/api/products

# Получить все связи продукции и цехов
curl http://localhost:8000/api/product-workshops
```

## **CREATE операции**

```
# Создать новый тип материала curl -X POST
http://localhost:8000/api/material-types \
-H "Content-Type: application/json" \
-d '{
    "name": "Новый материал",
    "loss_percentage": 3.5
}'

# Создать новый тип продукции curl -X POST
http://localhost:8000/api/product-types \
-H "Content-Type: application/json" \
-d '{
    "name": "Новый тип продукции",
    "coefficient": 1.2
}'

# Создать новый цех curl -X POST
http://localhost:8000/api/workshops \
-H "Content-Type: application/json" \
-d '{
    "name": "Новый цех",
    "workshop_type": "Производственный",
    "staff_count": 10
}'

# Создать новую продукцию curl -X POST
http://localhost:8000/api/products \
-H "Content-Type: application/json" \
-d '{
    "name": "Новый продукт",
    "product_type_id": 1,
    "article": "NP-001",
    "min_price": 1000.0,
    "main_material": "Дерево"
}'

# Создать новую связь продукции и цеха curl -X POST
http://localhost:8000/api/product-workshops \
-H "Content-Type: application/json" \
-d '{
```

```
        "product_id": 1,  
        "workshop_id": 1,  
        "production_time_hours": 5.5  
    }'
```

## Способ 3: Через Python скрипт

### Использование готового скрипта

```
cd backend  
python test_api.py
```

## Проверка всех модулей

### Модуль 1: Material Types (Типы материалов)

#### READ:

- GET /api/material-types - получить все
- GET /api/material-types/{id} - получить по ID

#### CREATE:

- POST /api/material-types - создать новый

### Модуль 2: Product Types (Типы продукции)

#### READ:

- GET /api/product-types - получить все

- GET /api/product-types/{id} - получить по ID

#### **CREATE:**

- POST /api/product-types - создать новый

## **Модуль 3: Workshops (Цеха)**

#### **READ:**

- GET /api/workshops - получить все
- GET /api/workshops/{id} - получить по ID

#### **CREATE:**

- POST /api/workshops - создать новый

## **Модуль 4: Products (Продукция)** | **Важно:** product\_type\_id

должен существовать в БД! **Модуль 5: Product Workshops**

### **(Связь продукции и цехов)**

#### **Важно:**

- product\_id должен существовать в БД!
- workshop\_id должен существовать в БД!
- Пара (product\_id, workshop\_id) должна быть уникальной!

## **Проверка обработки ошибок**

---

## Ошибка 404 - Запись не найдена

```
curl http://localhost:8000/api/material-types/999
```

### Ожидаемый ответ:

```
{
  "detail": "Material type not found"
}
```

## Ошибка 400 - Дублирование уникального поля

```
curl -X POST http://localhost:8000/api/material-types \
-H "Content-Type: application/json" \
-d '{"name": "Дерево", "loss_percentage": 5.0}'
```

### Ожидаемый ответ:

```
{
  "detail": "Material type with this name already exists"
}
```

## Чек-лист проверки CRUD

---

## **Material Types**

- GET все записи - возвращает список

---
- GET по ID - возвращает одну запись

---
- POST создание - создает новую запись

---
- POST дублирование - возвращает ошибку 400

---
- GET несуществующий ID - возвращает ошибку 404

## **Product Types**

- GET все записи

---
- GET по ID

---
- POST создание

---
- POST дублирование

---
- GET несуществующий ID

## **Workshops**

- GET все записи

---
- GET по ID

---
- POST создание

---

POST дублирование

---

GET несуществующий ID

## **Products**

GET все записи

---

GET по ID

---

POST создание

---

POST с неверным product\_type\_id - ошибка 400

---

GET несуществующий ID

## **Product Workshops**

GET все записи

---

GET по ID

---

POST создание

---

POST с неверным product\_id - ошибка 400

---

POST с неверным workshop\_id - ошибка 400

---

POST дублирование связи - ошибка 400

---

GET несуществующий ID

## Полезные команды

---

- **Проверка здоровья API:** curl http://localhost:8000/health
- **Информация об API:** curl http://localhost:8000/
- **Swagger UI:** http://localhost:8000/docs

