

# Что тестировать в каждом методе API

Краткая выжимка урока про CRUD — твой контекст перед практикой

## CRUD и CRUD-L

- **Create** — создать ( `POST` )
- **Read** — прочитать ( `GET` )
- **Update** — обновить ( `PUT` / `PATCH` )
- **Delete** — удалить ( `DELETE` )
- **L** (в CRUD-L) — List: списков в базе в готовом виде нет, их собирает API как посредник между хранилищем и системами

## Параметры vs часть пути

| Параметр <code>?status=candidate</code>         | Часть пути <code>/users/:id</code>            |
|-------------------------------------------------|-----------------------------------------------|
| Фильтрация, сортировка, пагинация               | Идентификация конкретного ресурса             |
| Логика обработки в коде                         | Роутинг эндпоинта                             |
| Пустой результат — валидный <code>200 OK</code> | Ресурс не найден — <code>404 Not Found</code> |

## Тестирование POST

- Валидация схемы входящего запроса (JSON-схема)
- Обязательность полей и типы данных
- Валидация схемы ответа и статус-кода ( `201 Created` )
- Попытка создания полного дубликата записи (если дубликаты не прописаны в доке — уточнить и всё равно проверить)
- Обработка некорректного тела запроса ( `400 Bad Request` )
- Проверка прав доступа и авторизации (права на создание)

## Тестирование GET

- Получение по ID: существующий ( 200 OK ) vs отсутствующий ( 404 Not Found )
- Получение списка: фильтрация, сортировка, пагинация
- Валидация структуры и типов данных в ответе
- Изоляция данных (разграничение доступа между пользователями)
- Поведение при большом объёме записей
- Кэш: удали ресурс и запроси снова — как быстро обновляется ответ

## Тестирование PATCH

- Частичное обновление (передача только изменённых полей)
- Валидация типов данных и ограничений обновляемых полей
- Защита системных и неизменяемых полей (ID, дата создания)
- Обработка некорректных значений ( 400 Bad Request )
- Проверка прав доступа на редактирование ( 403 Forbidden )

## Тестирование DELETE

- Удаление существующего ресурса ( 200 OK или 204 No Content )
- Повторное удаление ресурса ( 404 Not Found )
- Каскадное удаление (автоматическое удаление связанных данных)
- Проверка прав доступа на удаление ресурса
- Поведение зависимых ресурсов после удаления родительского
- Soft delete: ресурс может помечаться удалённым и не выдаваться в GET



### Это твой контекст перед практикой

Каждый метод в каждом API уникален, но по HTTP-методу сразу понятно, что проверять. Теперь используй этот чек-лист в задачах практикума.