

Отчёт о юнит-тестировании Vkatun API

1. Информация о тестировании

- Название проекта: Vkatun
- Версия API: 1.0.1
- Дата тестирования: 22 апреля 2025
- Тестировщик: Васильев Артём Александрович
- Метод тестирования: Юнит-тестирование через Go testing + Testify

2. Описание тестирования

Проведено юнит-тестирование основных обработчиков Vkatun API: работа с метриками, резюме, пользователями.

Использованы моки базы данных для изоляции тестируемой логики.

Покрыты следующие сценарии:

- Позитивные тесты

3. Результаты тестирования

Тест 1: Получение метрик

Параметр	Описание
Название теста	Получение метрик /metrics
Цель	Проверить успешное получение агрегированных метрик
Функция	GetMetrics
Предусловия	Авторизация под пользователем-администратором
Тип тестирования	Позитивный
Данные запроса	Заголовок Authorization с токеном
Шаги выполнения	1. Подготовить мок-ответ от БД 2. Отправить GET-запрос на /metrics
Ожидаемый результат	Code 200. Метрики получены
Проверка	Сравнить полученные метрики с ожидаемыми
Полученный результат	Метрики корректны. Code 200. Метрики получены

Тест 2: Обновление метрик

Параметр	Описание
Название теста	Обновление метрик /metrics/update
Цель	Проверить успешное обновление метрик через POST-запрос
Функция	UpdateMetrics
Предусловия	Авторизация под пользователем-администратором
Тип тестирования	Позитивный
Данные запроса	JSON: обновление метрик
Шаги выполнения	1. Подготовить мок-ответ от БД 2. Отправить POST-запрос на /metrics/update
Ожидаемый результат	Code 200. Метрики обновлены
Проверка	Проверить выполнение операции без ошибок
Полученный результат	Обновление прошло успешно. Code 200. Метрики обновлены

Тест 3: Получение резюме по ID

Параметр	Описание
Название теста	Получение резюме по ID /resume/{id}
Цель	Проверить успешное получение резюме
Функция	GetResumeByID
Предусловия	Резюме с заданным ID существует
Тип тестирования	Позитивный
Данные запроса	ID в URL
Шаги выполнения	1. Подготовить мок-ответ от БД 2. Отправить GET-запрос
Ожидаемый результат	Code 200. Резюме получено
Проверка	Проверить заголовок и содержимое тела ответа
Полученный результат	Резюме корректно. Code 200. Резюме получено

Тест 4: Удаление резюме

Параметр	Описание
Название теста	Удаление резюме /resume/{id}/delete
Цель	Проверить успешное удаление резюме
Функция	DeleteResume
Предусловия	Резюме с заданным ID существует
Тип тестирования	Позитивный
Данные запроса	ID в URL
Шаги выполнения	1. Подготовить мок-ответ от БД 2. Отправить DELETE-запрос
Ожидаемый результат	Code 200. Резюме удалено
Проверка	Проверить статус ответа
Полученный результат	Удаление прошло успешно. Code 200. Резюме удалено

Тест 5: Редактирование полного резюме

Параметр	Описание
Название теста	Редактирование резюме /resume/{id}/edit
Цель	Проверить успешное обновление всех полей резюме
Функция	EditResume
Предусловия	Резюме с заданным ID существует
Тип тестирования	Позитивный
Данные запроса	JSON: все поля резюме
Шаги выполнения	1. Подготовить мок-ответ от БД 2. Отправить POST-запрос
Ожидаемый результат	Code 200. Резюме обновлено
Проверка	Проверить успешное завершение операции
Полученный результат	Обновление прошло успешно. Code 200. Резюме обновлено

4. Выявленные дефекты

Дефектов не обнаружено в ходе юнит-тестирования основных функций.

5. Общая оценка

- Юнит-тесты проходят успешно
- Логика API работает корректно при изолированном тестировании
- Покрытие основных бизнес-сценариев достигнуто