

Цель тестового

Проверить, обладают ли кандидаты базовыми знаниями, необходимыми для обучения:

- ✅ Умение работать с Go.
 - ✅ Понимание базового CRUD.
 - ✅ Знание Fiber (или способность разобраться в нём).
 - ✅ Работа с PostgreSQL.
-

Задание

Описание проекта

Необходимо разработать **REST API** для управления задачами (TODO-лист).

API должно позволять:

- **Создавать** задачу.
 - **Читать** список задач.
 - **Обновлять** задачу.
 - **Удалять** задачу.
-

Требования к реализации

1. Используемый стек:

- Go + Fiber
- PostgreSQL (через `pgx`)
- Среда выполнения – локальная (Docker не обязателен)

2. Структура БД

Создайте таблицу `tasks` с полями:

- `id SERIAL PRIMARY KEY`
- `title TEXT NOT NULL`
- `description TEXT`
- `status TEXT CHECK (status IN ('new', 'in_progress', 'done')) DEFAULT 'new'`
- `created_at TIMESTAMP DEFAULT now()`
- `updated_at TIMESTAMP DEFAULT now()`

1. Реализовать API-эндпоинты:

- ✓ `POST /tasks` – создание задачи.
- ✓ `GET /tasks` – получение списка всех задач.
- ✓ `PUT /tasks/:id` – обновление задачи.
- ✓ `DELETE /tasks/:id` – удаление задачи.

2. Дополнительные требования:

- Корректная обработка ошибок.
- Код должен быть читаемым и структурированным.

3. Будет плюсом:

- Использование Docker
 - Наличие документации
 - Наличие миграций
-

Как сдавать задание

1. Опубликовать код в **GitHub/GitLab** (открытый или приватный репозиторий).
 2. Прислать ссылку на репозиторий.
 3. В `README.md` кратко описать запуск (если нужны доп. настройки, например `.env`).
 4. Предоставить скриншоты работоспособности сервиса, а именно:
 - Скриншот успешного запроса (Postman или Insomnia)
 - Скриншот наличия записей в базе данных
-

Критерии оценки

- ✓ Работоспособность API (CRUD операции).
 - ✓ Качество кода и логичность структуры.
 - ✓ Корректная работа с PostgreSQL.
 - ✓ Грамотная обработка ошибок.
-

Дедлайн

Кандидату даётся **3 дня** на выполнение задания.