

Мобильная разработка

Ionic + Angular + .NET

Задача

В этом задании вы реализуете простое приложение “Список задач” с взаимодействием с API. В репозитории https://github.com/Nastya-golubkova/mobile_dev_hw заготовлен шаблон на Ionic + Angular (фронтенд) и .NET (бэкенд), который вам предстоит доработать. Подробности см. в README репозитория.

1 Основное задание

Реализовать мини-приложение “Список задач” на одной странице: форма добавления задачи и список уже добавленных задач. Список задач хранится в памяти на бэкенде (без базы данных).

1.1 Backend (.NET)

Добавить два эндпоинта. Задачи хранятся в статическом списке в памяти.

- GET `/api/tasks` — вернуть список всех задач.
- POST `/api/tasks` — добавить новую задачу вида `{ id, title }`.

1.2 Frontend

На главной странице:

- поле ввода текста задачи и кнопка “Добавить”, которая отправляет POST `/api/tasks`;
- список уже добавленных задач, полученный через GET `/api/tasks`;
- после успешного добавления задачи список на странице обновляется (например, повторным запросом GET `/api/tasks` или локальным добавлением элемента в массив).

1.3 Definition of done

Для демонстрации работы приложения вам предлагается записать видео экрана.

- корректные HTTP-запросы к обоим эндпоинтам через `HttpClient`;
- отображение списка задач на экране;
- добавление новой задачи и появление её в списке работают без перезагрузки страницы.

Бонусное задание

Поднять PostgreSQL в Docker и заменить хранение задач в памяти на хранение в базе данных. Подробности см. в README репозитория.

1. Написать `docker-compose.yml` с сервисом `postgres` (образ `postgres`, переменные окружения для пользователя, пароля и имени базы данных, `volume` для сохранения данных между перезапусками).
2. В backend-проекте подключить `Npgsql` и `Entity Framework Core`, создать сущность `TaskItem` и `DbContext`, сгенерировать и применить миграцию.
3. Заменить in-memory список на работу через `DbContext`. Сигнатуры эндпоинтов при этом не меняются.