



Техническое задание: Сервис локализации для игр



Цель тестового задания

Создать сервис для управления локализациями игровых проектов. Интерфейс представляет собой таблицу (аналогично Google Sheets), где:

- строки — ключи локализаций,
- столбцы — языки,
- на пересечении — переводы.



Что нужно сделать

В рамках тестовой версии необходимо реализовать следующие функции:

1. Таблица переводов

- Таблица отображает список ключей (вертикально) и языков (горизонтально).
- Пересечение ключа и языка содержит перевод.
- Поддержка горизонтального и вертикального скролла
- Поддержка пагинации, таблица может быть разбита на куски, нужно реализовать постраничный показ

2. Добавление языка

- кнопка «Добавить язык».
- При клике открывается попап с выпадающим списком языков (подгружается с бэкенда).

- Пользователь выбирает язык → жмёт кнопку «Добавить» → в таблицу добавляется новый столбец.
- **Важно:** язык можно выбрать только из доступных (русский, английский, турецкий — временно защиты на бэке).

3. Добавление ключа

- кнопка «Добавить ключ» добавляет новую строку в таблицу.
- При клике открывается попап с вводом, где нужно указать ключ перевода
- Пользователь вписывает ключ → жмёт кнопку «Добавить» → в таблицу добавляется новая строка с ключом.

4. Редактирование переводов

- Каждая ячейка в таблице редактируемая.
- При клике в ячейку — активируется режим ввода.
- При потере фокуса — сохранение нового значения (по API).

Опционально и будет плюсом

- Поиск по ключу и отображение найденного результата
- Удаление ключей из таблицы
- Удаление языков из таблицы
- Тесты для тестирования
- Отдельное окно для добавления языка в базу для выбора пользователем (мини админка)



Frontend

✓ Обязательные технологии:

- **Next.js** (v13+/v14+, **app** routing) или **React**
- **Axios** — HTTP-клиент

- **Tailwind CSS** — стилизация

+ Дополнительно (поощряется):

- **React Hook Form** — удобная работа с формами
 - **React Query** — работа с API и кэшированием
 - **shadcn/ui** — библиотека UI-компонентов
 - **Zod** — схема и валидация данных на клиенте
-



Backend



Обязательные технологии:

- **ASP.NET Core 8**
- **Entity Framework Core (Code-First)**
- **PostgreSQL**
- **Controllers**

+ Дополнительно (поощряется):

- **FluentValidation** — валидация входящих данных
- **FluentResults** — для унифицированного возврата результатов



Архитектура (одно из двух):

- **Монолит**

- **Модульный монолит.** В случае модульного монолита вынесите каждый модуль в отдельную библиотеку классов (например: главный web api проект и к нему подключается модуль GameLocalization)
-



Запуск через Docker Compose

- Команда: `docker-compose up --build`
 - Сервисы:
 - Frontend
 - Backend (ASP.NET Core 8)
 - База данных (PostgreSQL)
-



Что ожидается от кандидата

От разработчика ожидается реализация

♦ Backend

- Грамотное проектирование **REST API**:
 - Чистая структура маршрутов (CRUD + специализированные endpoints).
 - Удобные форматы ошибок и сообщений.
 - Поддержка версионирования (желательно).
- Продуманная архитектура:
 - Монолит или модульный монолит с четким разделением ответственности.
 - Использование **слоистой архитектуры (контроллеры, сервисы, репозитории)** (желательно)..
- Грамотное проектирование **базы данных**:

- Учет масштабируемости (например, добавление новых языков).
- Минимизация дублирования.
- Возможность фильтрации и поиска по ключам/языкам.

♦ Frontend

- Работа с **табличными структурами**, виртуализация данных при большом количестве строк/столбцов.
- Аккуратная и **читаемая структура компонентов** с логическим делением.

♦ DevOps & запускаемость

- Проект должен быть легко запускаем **через Docker Compose**:
 - Стандартизированные **Dockerfile** для фронта и бэка.
 - Настроенный **docker-compose.yml**, который поднимает весь стек.
- Описание необходимых переменных окружения и инструкций по запуску.

♦ Общие компетенции

- Умение писать **чистый, читаемый и поддерживаемый код**.
- Навыки **самостоятельной проработки архитектуры** при минимальном надзоре.
- Приветствуется опыт работы с локализацией/табличными UI/внутренними админками.

♦ Плюс

- Краткое описание того что можно было бы улучшить, например взять другую базу данных для хранения. И как в целом прокачать сервис

✅ Что прислать

- Ссылка на гитхаб репозиторий

- Опционально: короткий скринкаст или деплой- с с ы л к а

✓ Дедлайн и сроки

Базовая часть — дата согласуется. Если нужно больше времени — предупредить заранее.

✓ Примеры страниц

Q Search keys...

Rows per page: 10 1-10 of 20 < >

≡ Key	🇺🇸 English	🇩🇪 German	🇮🇹 Italian	🇫🇷 French	+	🗑 Remove
ui_Menu	Menu	Speisekarte	Menù	Menu		–
ui_Settings	Settings	Einstellungen	Impostazioni	Paramètres		–
ui_Volume	Volume	Volumen	Volume	Volume		–
ui_Battle Pass	Battle Pass	Schlachtpass	Passaggio di battaglia	Pass de combat		–
ui_Market	Market	Markt	Mercato	Marché		–
ui_Play	Play	Spielen	Giocare	Jouer		–
ui_Disconnect	Disconnect	Trennen	Disconnetti	Déconnecter		–
ui_Sale	Sale	Verkauf	Vendita			–
ui_Guild	Guild	Gilde	Gilda	Guilde		–
ui_Friends	Friends	Freunde	Amici	Amis		–
+ Add						