Neoflex
IT-решения для бизнеса

Разработка NFO

ТЕСТОВОЕ ЗАДАНИЕ

Подготовлено:
Компания «Неофлекс»
Россия, 127015, Москва, ул. Вятская, 35, стр. 4
Тел./факс: +7(495) 984-25-13, 984-27-90
http://www.neoflex.ru

Москва
2020

Функциональные требования

Необходимо реализовать систему перевода. Система перевода будет представлять из себя сервис или сервисы, которые будут представлять возможность пользователю делать перевод предложения. Фронтовую часть реализовывать не нужно. Реализуются следующий сценарии:

Основной сценарий:

- Пользователь вызывает вводит на форму предложение, выбирает текущий язык и язык перевода
- С фронта вызывается api бэковой системы
- Бэковая система валидирует сообщение и предложение (разрешенные символы **{ } ! , . : [ ] -**). При невалидном запросе см. Альтернативные сценарии
- Бэковая система разбивает предложения на части (в качестве разделителя используется пробел)
- Бэковая система для каждой части предложения, убирает с концов знаки пунктуации выполняет по преобразованному слову поиск в реляционной БД перевод, в переведенное слово добавляет убранные знаки пунктуации
- Бэковая система собирает из переведенных частей предложение и возвращает на фронт

Альтернативный сценарий1:

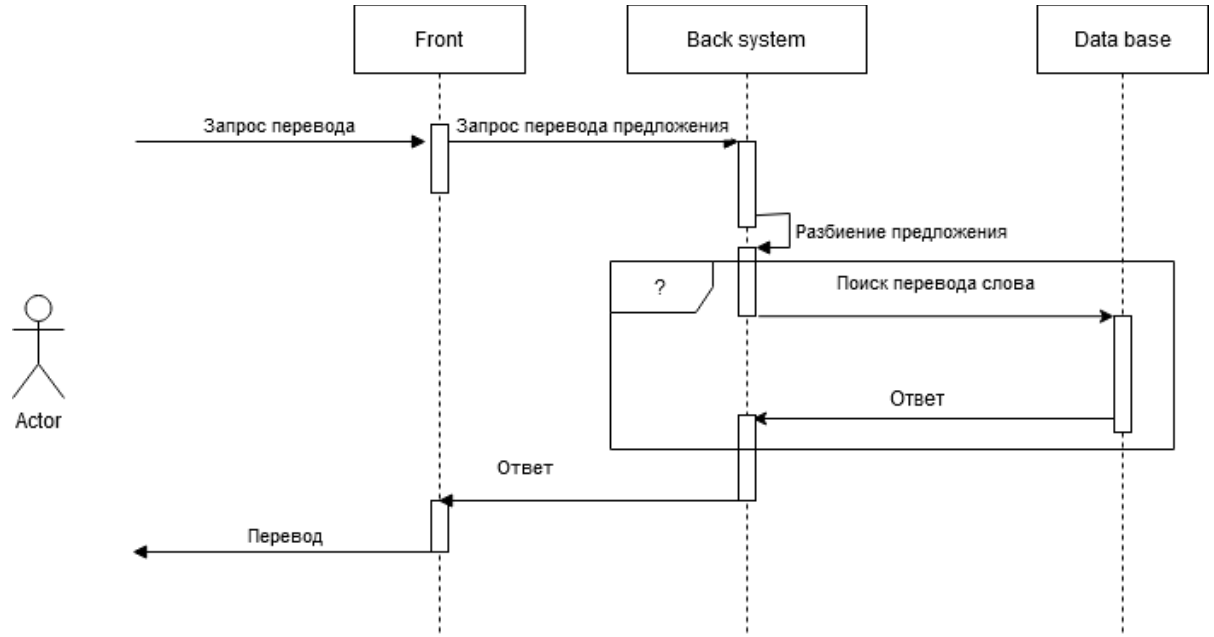
- Пользователь вызывает вводит на форму предложение, выбирает текущий язык и язык перевода, при этом вводит запрещенные символы
- С фронта вызывается api бэковой системы

- Бэковая система определяет, что предложение содержит запрещенные символы и возвращает на фронт ошибку

Альтернативный сценарий1:

- На бэковую систему пришло сообщение с некорректным типом запроса, в теле отсутствуют обязательные поля или указан некорректный url
- Бэковая система возвращает на фронт ошибку.

Верхнеуровневая диаграмма последовательности



Нефункциональные требования

- Кол-во слов в предложении не более 100000.
- Время обработки запроса не более 1 мин.

Требования к реализации

- Бэковая часть реализуется на языке Java или Kotlin
- Бэковая система может быть реализована в виде монолитного веб приложения или одного, или нескольких сервисов в виде standalone java приложений
- Можно использовать стек Spring или Java EE
- Реляционная БД либо PostgreSQS, либо MySQL
- Для сборки можно использовать maven или gradle
- Можно использовать брокеры сообщений Kafka или RabbitMQ
- Структура БД и тестовый набор данных должны проливаться при старте системы автоматически
- Будет плюсом если система будет собираться в наборы докер образов и полностью подниматься с помощью docker-compose файл
- Исходный код выкладывается на github