

«Национальный исследовательский университет
«Высшая школа экономики»
Лицей

Индивидуальная выпускная работа

ИТ-ПРОЕКТ: Your Train

<https://github.com/Ilya-konstantinov/YourTrain/>

Выполнил: Константинов Илья Игоревич

Группа: 11И2

Москва – 2023

1. Введение

Раньше я имел небольшой опыт разработки telegram ботов, так чтобы был знаком с концепцией, однако до этого никогда не работал с серьёзными асинхронными задачами. Такая область была выбрана, потому что я сам езжу на электричках и знаю о существующей проблеме и неудобствах, с ней связанных.

2. Проблемное поле

Каждый человек, который пользовался электричками, сталкивался с отсутствием подходящего сервиса. Электрички очень непостоянны, и каждый, кто на них ездил, сталкивался с трудностями поиска подходящего маршрута. При чем многие ищут электричку каждый день, поэтому их непостоянность становится насущной проблемой. Мой проект и призван помочь каждому, кто пользуется электричками, решить эти проблемы.

3. Образ продукта

- Узнать ближайшую электричку от станции до станции на определенную дату

Основная цель проекта – сервис для расписания, было необходимо предоставить удобный сервис с исчерпывающим функционалом.

- Выгрузка данных пользователя о электричках в формате JSON

Мне, как пользователю многих сервисов, не нравится, что я не знаю какие данные хранит обо мне сервис, поэтому в своем сервисе мне хотелось дать пользователю манипулировать данными, которые «я» о нем храню.

- Оставить отзыв о работе сервиса

Никакой сервис не создаётся хорошим сразу – если пользователь поймет, что он может улучшить приложение, то может оставить комментарий, который поступит напрямую администратору.

- Узнавать новости о выбранной целевой электричке

У электричек есть еще одна черта – иногда у привычной электрички меняется расписание, а когда ты не смотришь на неё каждый день, а просто «знаешь» её расписание, то подобные внезапные изменения могут сильно испортить жизнь. Чтобы такого не было, бот сам отслеживает такие изменения за вас и уведомляет об изменениях заранее.

4. Сделанный продукт

- «Сценарий 1» был реализован полностью. Пользователь может манипулировать данными о себе, а также писать отзывы.
- «Сценарий 2» был полностью реализован. Пользователь получает полноценную информацию от бота.
- «Сценарий 3» был полностью реализован. Пользователь меняет параметры на свое усмотрение без ошибок.
- «Сценарий 4» был полностью реализован. Бот обрабатывает запросы, ловит ошибочные выражения и параметры.
- «Сценарий 5» был полностью реализован. Пользователь может изменить параметры запроса и увидеть необходимый маршрут после.
- «Сценарий 6» был не полностью реализован. Пункт «б», на изменение существующего пути, на практике оказался бессмысленным и тяжелым в реализации, было принято решение от него отказаться.
- «Сценарий 7» был не полностью реализован, по той же причине, что и прошлый сценарий, так как они связаны. По той же причине, что и пункт 6.6 был не реализован пункт «3», «5» и «б».

5. Бэкенд

Базой данных был выбран MySQL сервер. Основные таблицы – users, path_cache, req_cache, station_cache – хранят сохранённые параметры запросов, данные о пользователе, а также копии информации о станциях и путях. В качестве “primary keys” были выбраны уже существующие id из «телеграмма» или те, что хранит в себе API.

6. Средства разработки

В качестве основного фреймворка был выбран aiogram, который позволяет удобно работать со всеми методами API Telegram. Для проектирования БД был выбран dbdesigner, в последствии для непосредственной работы с базой данных были использованы возможности PyCharm Professional.

Так же проект использует такие фреймворки для асинхронной работы, как aioschedule для расписания уведомлений о сохранённых путях и asyncio для асинхронной работы в принципе.

7. Этапы работы

- А. Знакомство и обработка запросов API – API, с которым я работаю, не имеет документации. Этот этап был пройден, но потребовал нестандартных навыков и знаний работы сети интернет.
- В. Преобразование данных API в прикладную форму – Типы данных, в которые преобразовывались сырые запросы к API, изменялись в процессе написания сервиса, но основные идеи и методы были написаны на данном этапе.
- С. Реализация функционала без интерфейса – На данном этапе были изучены базовые принципы работы с aiogram и взаимодействия с пользователем.
- Д. Связь проекта с БД – Были реализованы сохранения запросов и путей с последующей модификаций данных.
- Е. Реализация интерфейса – Полноценное знакомство с возможностями aiogram и Finite-state machine.
- Ф. Полировка проекта – Обработка непредсказуемого поведения пользователя и очищение проекта от всех возможных багов.

8. Рефлексия

В процессе работы были замечены ошибки проектирования используемых фреймворков, которые требовали погружения в исходный код и чтения его

неполной документации. Так же в процессе написания проект несколько раз менял архитектуру в связи со сложной иерархией взаимодействия классов и методов.

В будущем развития проекта, можно улучшить взаимодействие с сохранёнными запросами, добавить возможность уведомлять об сохранённых маршрутах по пользовательскому расписанию. Также можно добавить inline запросы, которые позволяют делать запросы без захода в меню бота.

В процессе работы были получены знания и навыки асинхронной работы Python и Telegram. Эти навыки расширили мои возможности написания кода, а также возможности использования Python.

Для лучшего масштабирования проекта можно изменить типы хранения некоторых данных, которые на данном этапе необязательно хранить вне оперативной памяти.

9. Итог

На данный момент это самый большой проект, который я писал в одиночку и самый интересный проект, написанный мной. В дальнейшем хотелось бы развивать проект дальше, масштабировать используя возможности Docker и, возможно, сделать его массовым, передав права одной из компаний, занимающихся обслуживанием электричек.