

Завдання
Етапу Кваліфікації
Dev Challenge 10
Front-end JS



1. Завдання

Реалізувати пошук найоптимальнішого і най неоптимальнішого шляху з пункту А до пункту В.

Опис додатку:

використовуючи Open Street Maps API або інші мапи публічного доступу, реалізувати можливість пошуку найкоротшого та, бажано, найдовшого маршруту з точки А до точки В.

Користувач повинен мати можливість:

- 1) вибрати точку А та точку В, бажано зупинки метро, але можете використовувати і інші види транспортних сполучень у вигляді випадючих списків;
- 2) автоматичного вибору найближчого кінцевого вузла при кліку по мапі;
- 3) пошуку як найкоротшого, так і найдовшого маршрутів. Координати вузлів (зупинок транспорту) можете зберігати у форматі json або txt., зберігати файли з координатами на сервері (Node.js).

2. Формат представлення результатів

2.1. Рішення вивантажувати в особистому кабінеті на сайті в **ОДНОМУ** файлі-архіві з назвою у форматі **“Прізвище Ім’я.zip”**, у якому мають бути наступні файли:

- 1) вихідні файли проекту підготовлені для збірки (gulp-проект, webpack і т.п.);
- 2) зібрану версію проекту, яку можна просто відкрити в браузері.

2.2. Організатори та судді залишають за собою право дискваліфікувати роботу учасника, якщо робота:

- 1) містить будь-яку вказівку на ім’я, прізвище, електронну пошту, компанію, адресу чи інші персональні дані учасника;
- 2) виконана у іншому форматі, ніж вказано у завданні;

- 3) виконана з допомогою сторонніх осіб, а не учасником особисто.

3. Обмеження та критерії оцінювання

3.1. Обмеження реалізації завдання:

- 1) Допускається використання будь-яких бібліотек загального використання. Бажаним стеком є Node.js (може містити логіку розрахунків оптимального маршруту та дані про вузли A, B, C, D ... Z), Angular.js, Underscore.js, bootstrap, будь-яка з Open Maps API.
- 2) Не допускається використання бібліотек, які реалізують алгоритми пошуку

3.2. За наступний реалізований функціонал нараховуватимуться бали:

- 1) Можливість вибирати точку A та B за допомогою випадających списків.
- 2) Можливість вибирати точку A та B за допомогою кліку по карті (алгоритм сам знайде найближчу станцію до місця кліку мишки по карті).
- 3) Підтримка Node.js серверної частини з можливістю додавати нові координати вузлів.
- 4) Реалізація алгоритму пошуку найкоротшого та найдовшого маршруту на стороні клієнта або сервера.
- 5) Можливість вибору найкоротшого та найдовшого маршруту з вузла A до вузла B.
- 6) Вивести результати у довільній формі. Або прокласти маршрут по мапі (червоним кольором), або поставити "прапорці" A та B на мапі, буде вітатися ваш варіант.

4. Експерти



Євген Вовк

Senior Software Engineer
@EPAM Systems



Андрій Шумада

Fullstack js developer
@Debitoor/Ciklum



Максим Крамаренко

Front-End Technical Lead
@Zoomdata/Ciklum

5. Контакти

- 5.1. Рішення необхідно вивантажити у особистому кабінеті на сайті devchallenge.it за необхідною номінацією **до 18 вересня, 23:59 (EEST)**. Після вичерпання часу можливість вивантажити роботи на сайт буде автоматично заблокована.
- 5.2. Питання та уточнення **щодо змісту завдання** ви можете задати за адресою box@devchallenge.it.
- 5.3. Судді ігноруватимуть питання, які не стосуються завдання Чемпіонату.
- 5.4. Час роботи саппорту: **щодня 10:00 - 21:00**.
- 5.5. **Оголошення півфіналістів відбудеться 19 вересня.**

Генеральний партнер сезону:

PROFFSTORE

