

ОПТИМІЗАЦІЯ МАРШРУТИЗАЦІЇ НА ОСНОВІ ГЕОКОДУВАННЯ ДАНИХ

Автор Шевченко В. О.

(Науковий керівник – д.т.н., професор Яненко О. П.)

Більшість програм, які використовують наприклад кур'єри для доставки замовлення, розраховують маршрути з точки А в точку Б без прив'язки до поточного часу, вони просто розраховують дистанцію та час перебування на маршруті.

Необхідна інформація для аналізу дорожньої ситуації для побудови оптимального маршруту розміщена в Google Maps Api, а саме в Directions Api в поєднанні з Geocoding Api. За наявності токена, дані ресурси можливо використовувати безкоштовно по 1000 запитів на день, що достатньо для тестування програмного забезпечення.

Алгоритм роботи модуля динамічної побудови оптимальних маршрутів базується на парсингу отриманої інформації з Google Maps Api.

По протоколу HTTP, робиться GET запит з заданими параметрами. Запит на GOOGLE має такий вигляд <https://maps.googleapis.com/maps/api/directions/output?parameters>, в якому

- output може приймати параметри : JSON, XML ;
- parameters може бути наступним :
 - origin – текстове значення довготи або широти точки початку маршруту;
 - destination - текстове значення довготи або широти точки кінцевої точки маршруту.

Це два основних параметрів, для отримання інформації від Google Maps Api, ще є багато параметрів які будуть формуватися в залежності від запиту клієнта, наприклад, якщо клієнт буде використовувати велосипед для переміщення, то будувати маршрут йому треба з врахуванням наявності велодоріжок на маршруті, для цього використаємо параметр mode = bicycling.

Google сформує відповідь в вигляді JSON, яку ми отримуємо. Проаналізуємо потрібну інформацію і сформуємо результат в форматі JSON, який далі відправляємо клієнту де формується маршрут і відображається отримана інформація. Нижче (рис.1.) наведено приклад запиту HTTP, який розраховує маршрут з Чикаго (штат Іллінойс) в Лос-Анджелес (штат Каліфорнія) за допомогою двох проміжних точок: Джоуплін (штат Міссурі) і Оклахома-Сіті (штат Оклахома).

В наведеному прикладі виконується запит в форматі JSON : https://maps.googleapis.com/maps/api/directions/json?origin=Chicago,IL&destination=Los+Angeles,CA&waypoints=Joplin,MO|Oklahoma+City,OK&key=YOUR_API_KEY. Також можна запросити запит в форматі XML. Нижче наведено приклад JSON відповіді від Google в форматі JSON.

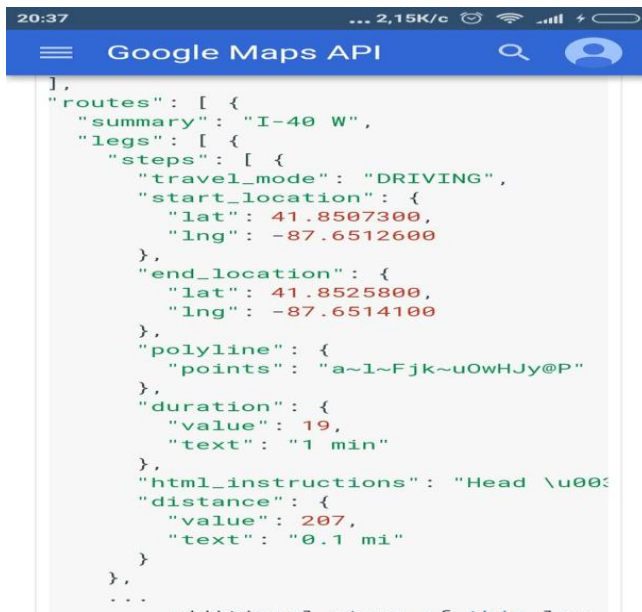


Рисунок 1. Формат JSON

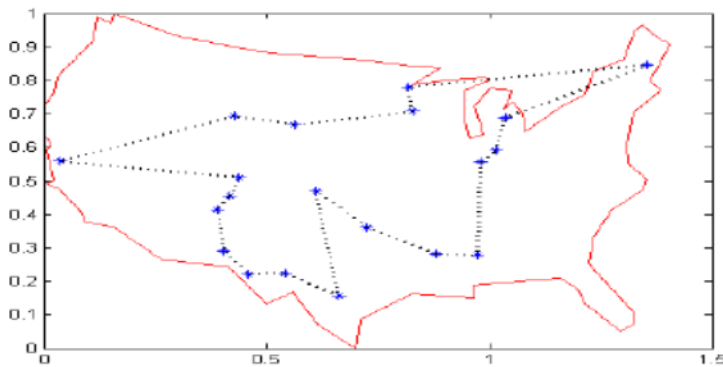


Рисунок 2. Побудова маршруту

Оскільки отримані маршрути можуть бути досить детальними, повторювані елементи у відповідях не приводяться.

На рис.2 зображено лише маленьку частину отриманої інформації. Маючи цю інформацію і використовуючи всі можливості Java та фреймворків динамічно розраховується і маршрут з точки А в точку Б.

Алгоритм розрахунку відстані

1. Авторизація/реєстрація на ресурсі.

2. Вибір точки відправлення і точки прибуття (збоку фронт-енд) , передача їх на бек-енд з використанням технології ajax.

3. Процес геокодування даних (використовуємо Google Maps Geocoding api).

4. Розрахунок відстані по формулі.

5. Передача координат, результату відстані, часу на фронт-енд.

6. Відображення отриманої інформації на карті з використанням Google Maps Api.

Перелік посилань

1. Документація Google Maps API, Distance Matrix API . [електронний ресурс] <https://developers.google.com/maps>
2. Яндекс.Карты. [електронний ресурс] [://yandex.ru/maps](https://yandex.ru/maps)
3. Мацуда К., Лі Р. WebGL: Програмування тривимірної графіки.

Анотація

Розглянемо можливість розробки програмного продукту, який буде динамічно розраховувати маршрути враховуючи наявність пробок на дорожі, режими поїздки: то може бути автомобільна, велосипедна, піша чи з використання транспорту загального призначення поїздки.

Ключові слова: геокодування, Java, Google Maps Api, Directions Api, Spring

Abstract

Consider the possibility of developing a software product that will dynamically calculate routes taking into account the presence of traffic jams on the road, travel modes: it may be automotive, cycling, walking or using general purpose travel.

Keywords: geocoding, Java, Google Maps Api, Directions Api, Spring