

GTFSで何ができる？

ーこれまでの受賞作とGTFSで遊べることー

東日本旅客鉄道株式会社

羽田野湧太

自己紹介

ー学生時代に交通×ITの虜となり鉄道会社のIT部門へー



東日本旅客鉄道株式会社
イノベーション戦略本部
(JR東日本情報システム出向)
Digital & Data
Innovation Center

羽田野 湧太
IT・デジタル採用
社会人3年目

高校

2016年 武蔵野大学第二回数理工学コンテスト最優秀賞
(羽田空港へのダイヤ改正乗客シミュレーション)
2018年 朝ラッシュ時のダイヤに関する卒業研究

大学

2019年 第2回東京公共交通オープンデータチャレンジ 優秀賞
2020年 第3回東京公共交通オープンデータチャレンジ 審査員特別賞
2022年 第4回東京公共交通オープンデータチャレンジ 最優秀賞
2022年 卒業論文 Multi-objective Searching on Multimodal Transportation Networks*
2022 13th International Congress on Advanced Applied Informatics Winter (IIAI-AAI-Winter)

入社

2022年 4月 仙台支社研修
2022年 8月 池袋駅 (改札・みどりの窓口)

2023年 8月 JR東日本情報システム出向
10月 都知事杯オープンデータチャレンジ 技術賞
2023年11月 JR東日本 Digital & Data Innovation Center

2024年 3月 アーバンデータチャレンジ 銀賞・銅賞・GTFS特別賞・
国土交通データプラットフォーム特別賞

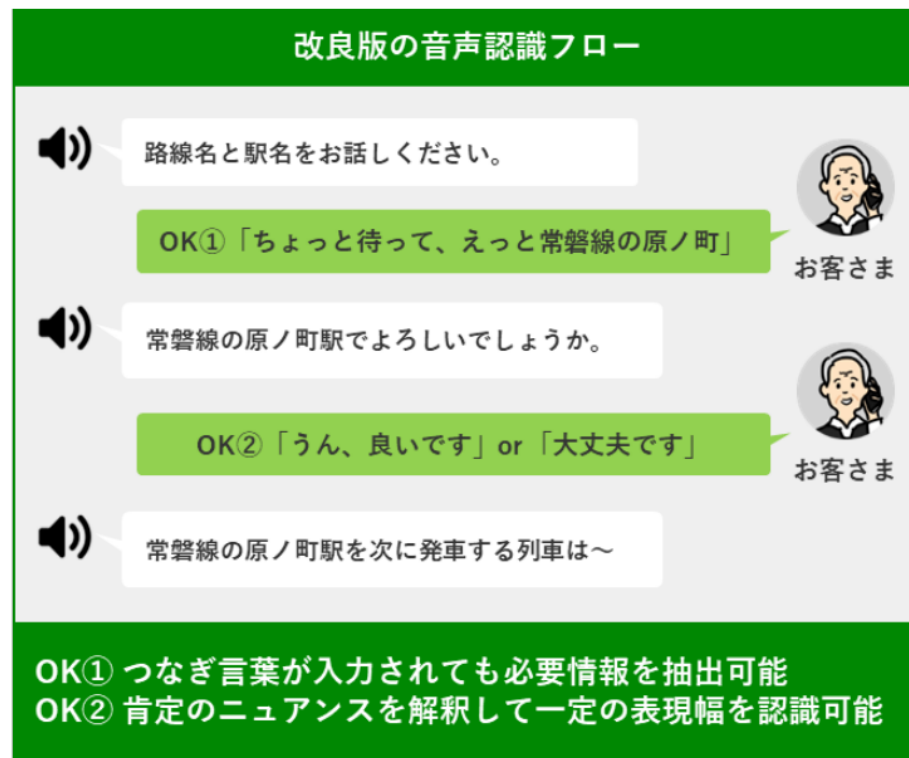
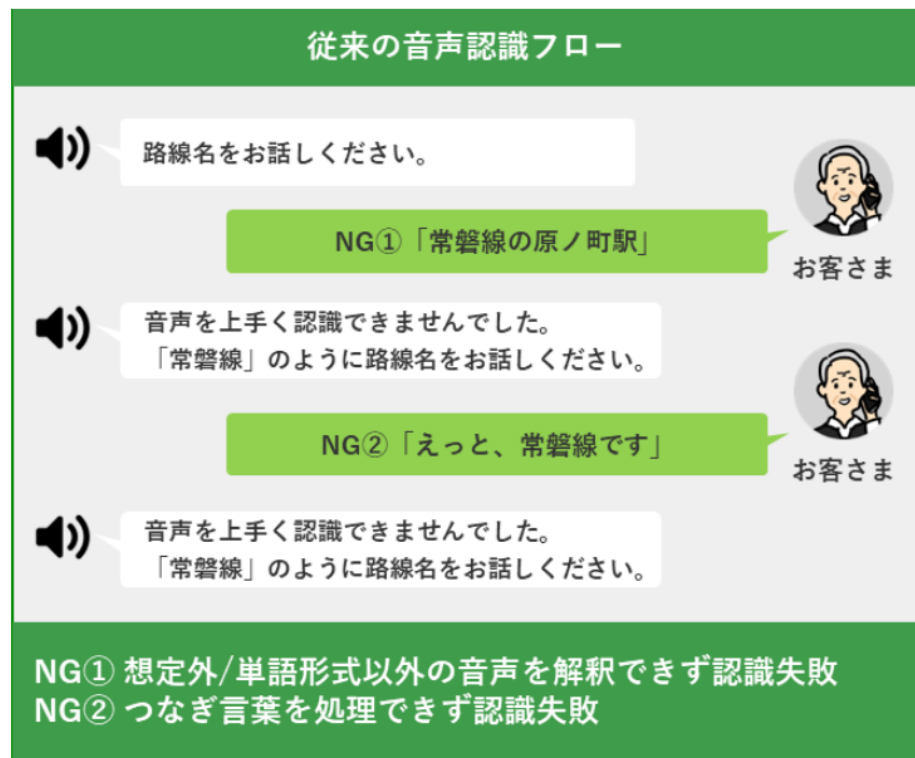
現在の業務

— 鉄道会社における内製開発 —

実証実験中！！

生成AIを活用した電話による運行情報案内サービス「どこトレダイヤル（☎ 050-3355-6421）」開発

※常磐線・水戸線・水郡線の一部区間に対応



趣味

－ Hackathonや乗り物にたくさん乗る旅行－

Hackathonなどで趣味開発



都知事杯
OpenData
Hackathon2023

UrbanDataChallenge2023

鉄道などでの移動全般



これまでの受賞作

—経路検索の見せ方を変えてアプリ化—

第2回 優秀賞（チーム）

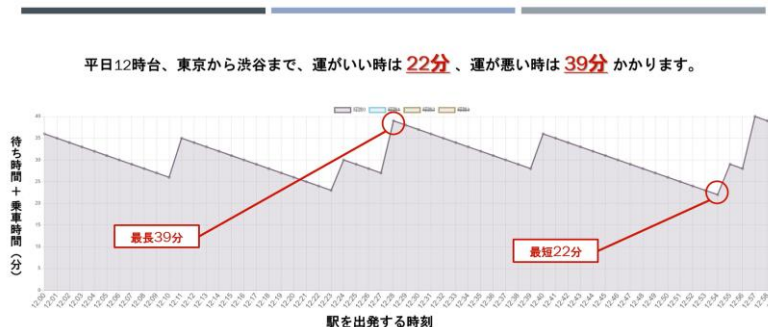
羽田野湧太、村松波、玉井亮央

鶴見→大船



リアルタイム遅延考慮経路検索

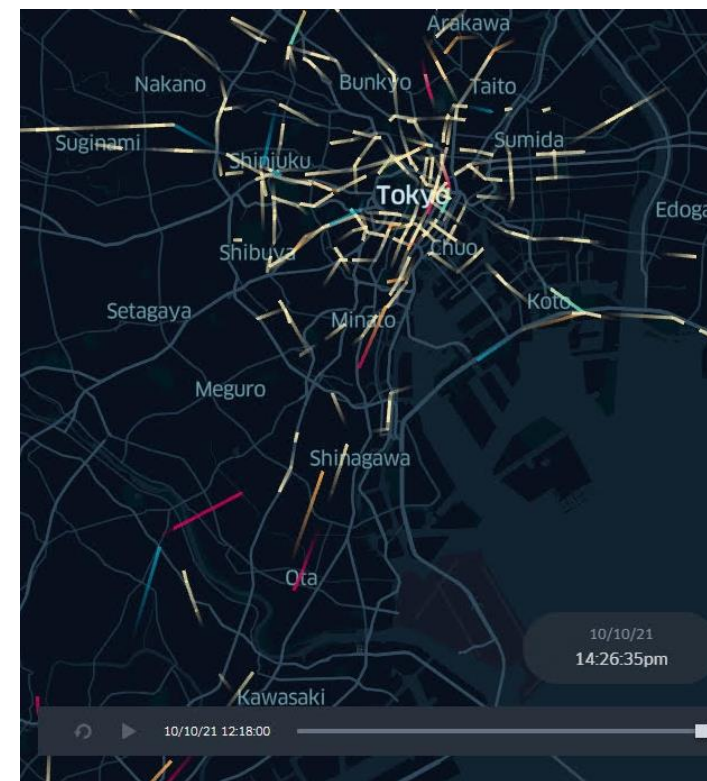
第3回 審査員特別賞（個人）



いつ出発すると早いか可視化

第4回 最優秀賞（チーム）

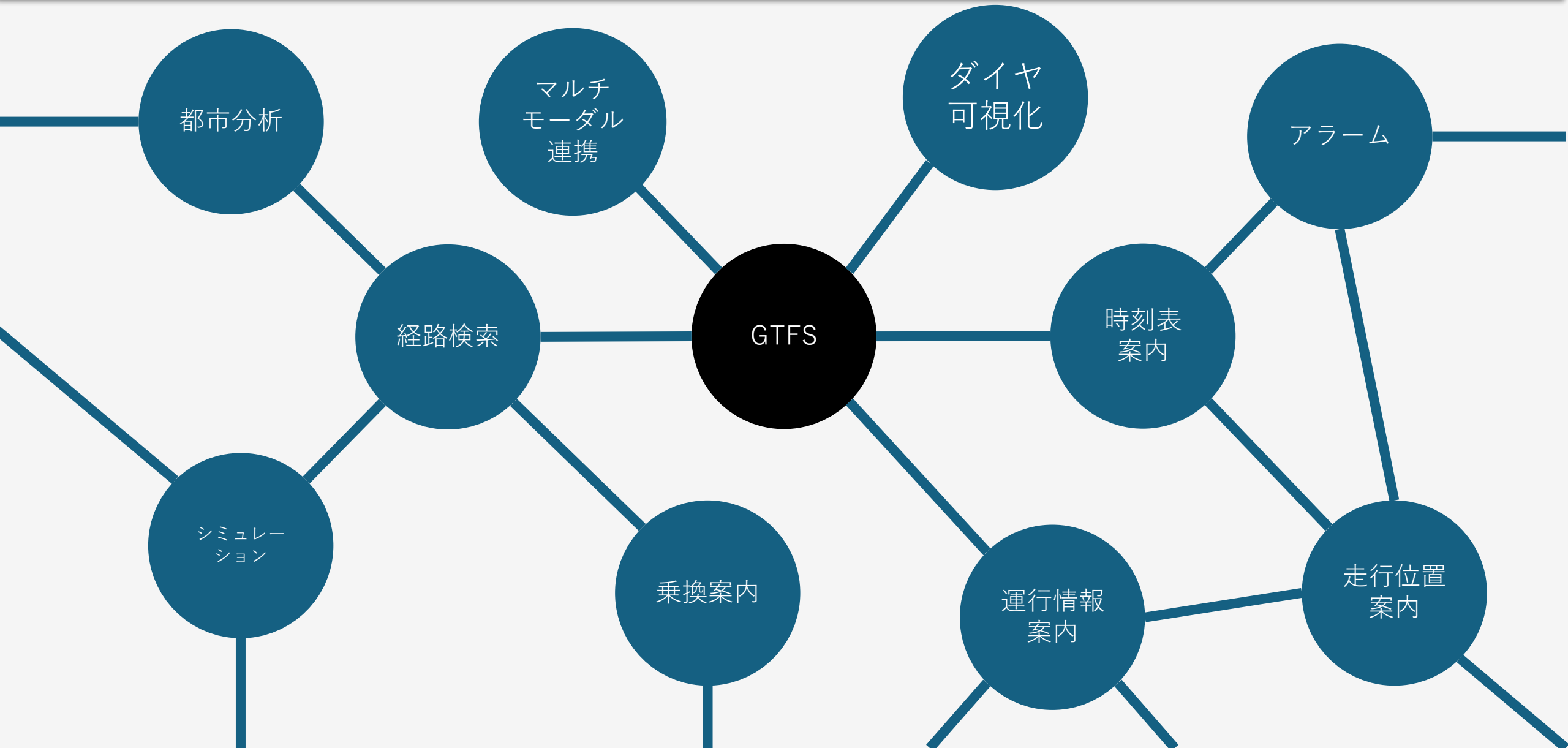
羽田野湧太、佐波哲朗



輸送障害時にどこが混雑するか

GTFSでできること

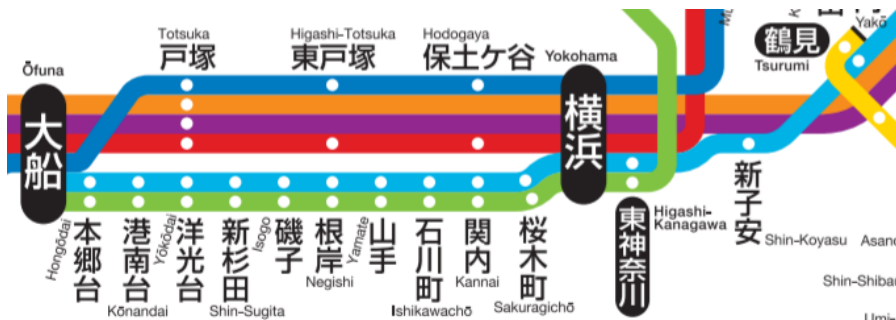
ーGTFSはワンソース・マルチユースー



経路検索を組み合わせると？

経路検索を1回するとーGTFSはワンソース・マルチユースー

9:25分に駅に着いた場合



JR東日本路線図より

情報更新：2018年10月22日 13:00

鶴見

日本語 English

北行 南行

磯子 行き
09:19発 普通 -6分後 (10分遅れ)

大船 行き
09:27発 普通 2分後 (8分遅れ)

磯子 行き
09:33発 普通 8分後

大船 行き

Worker Step

Search

鶴見

下車駅を入力 (例：新宿)

Search

詳細検索

乗車駅、下車駅は漢字で入力してください。

あ a ー ？ ！

※現在の所属先でなく学生時代に行ったものです

経路検索を60回すると —GTFSはワンソース・マルチユース—

平日12時台、東京から渋谷まで、運がいい時は **22分**、運が悪い時は **39分** かかります。



固定された経路での所要時間を可視化することができる

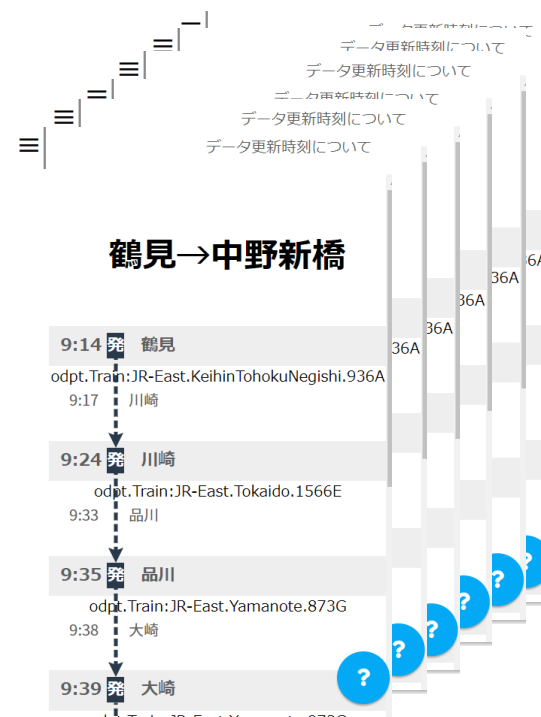
※現在の所属先でなく学生時代に行ったものです

経路検索をたくさんするとーGTFSはワンソース・マルチユースー

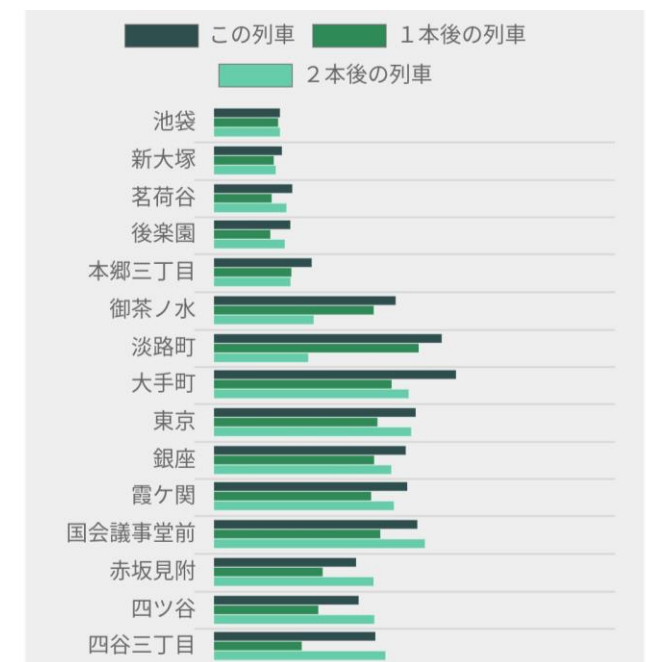
交通センサスより ODデータ作成

乗車駅	下車駅	乗車時刻
赤坂見附	中野坂上	9:30
銀座	新中野	9:42
国会議事堂前	新宿御苑前	9:27
新宿三丁目	荻窪	9:40
茗荷谷	御茶ノ水	9:35
池袋	後楽園	9:29
本郷三丁目	東京	9:32
西新宿	新高円寺	9:36
四ツ谷	四ツ谷三丁目	9:28
大手町	南阿佐ヶ谷	9:31
淡路町	霞が関	9:33

乗客の経路選択を予測



混雑状況を予測



※現在の所属先でなく学生時代に行ったものです

経路検索をたくさんすると

ーGTFSはワンソース・マルチユースー



※すべての人が最短所要時間経路を利用するという仮説の基、経路を選択しているので、実際より優等列車に混雑が集中しやすくなります。

2019年7月11日（木）@稲田堤駅 6号車 駅到着前

※現在の所属先でなく学生時代に行ったものです

GTFS/GTFS-RT × PLATEAU

PLATEAUと組み合わせると

ーGTFSはワンソース・マルチユースー

こんな経験、ありませんか？



綺麗な車窓を写真に残そうとしたら、
障害物にさえぎられてしまった…。



<https://window-grapher.app.takoyaki3.com>

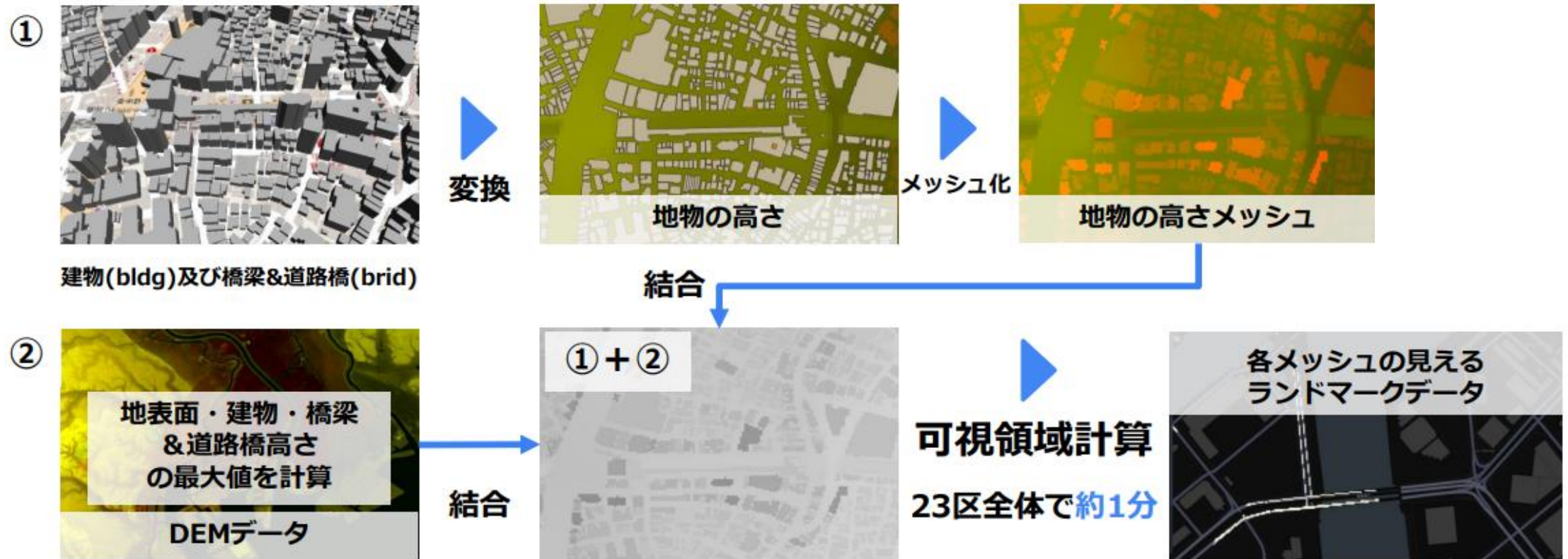
齋藤悠宇、富木菜穂、羽田野湧太、三輪哲大

※趣味でチーム開発した事例です

PLATEAUと組み合わせると

ーGTFSはワンソース・マルチユースー

PLATEAUによりランドマークが見える場所を算出



※趣味でチーム開発した事例です

PLATEAUと組み合わせると

ーGTFSはワンソース・マルチユースー

直近に通るメッシュ群に見えるメッシュが含まれば
通知を送信する

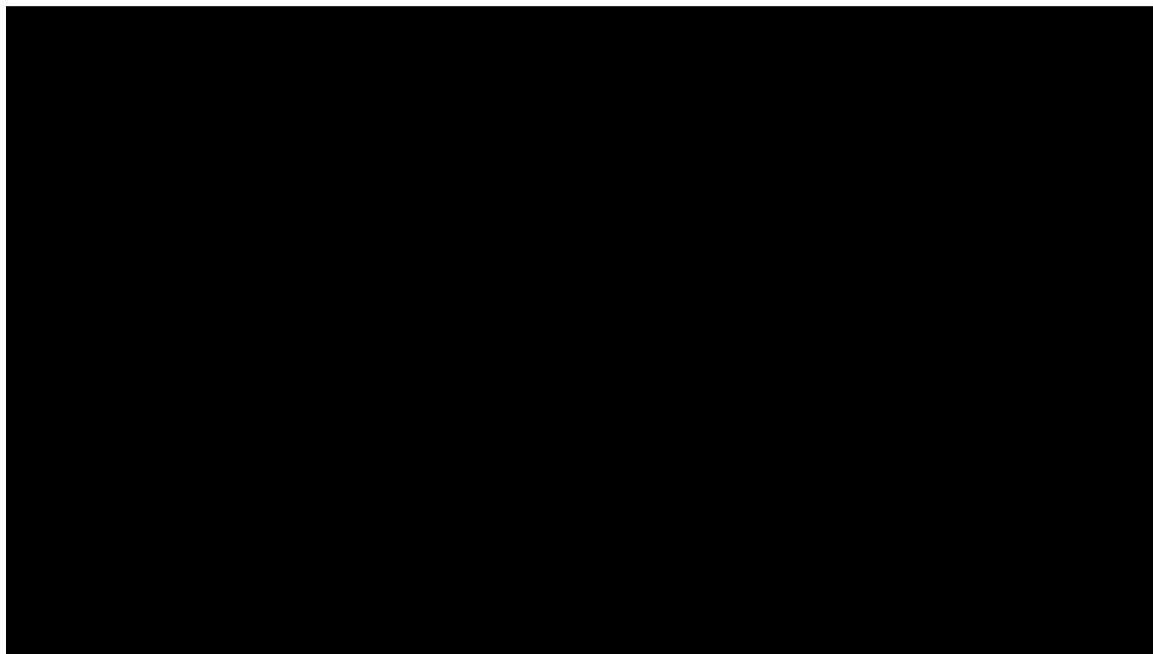


※趣味でチーム開発した事例です

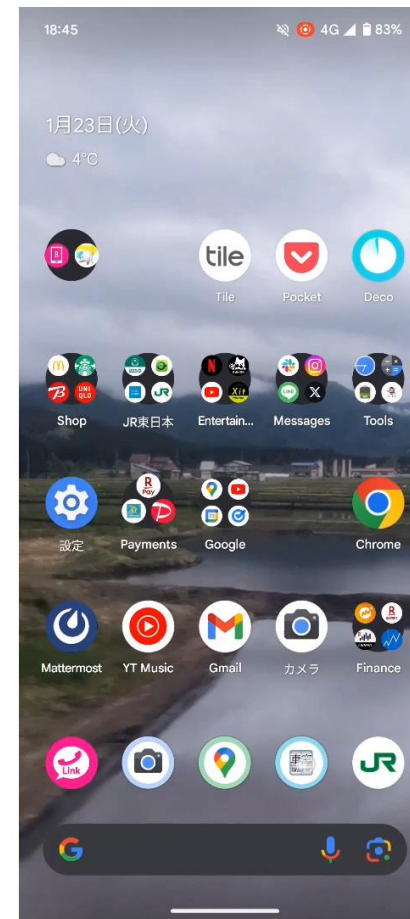
PLATEAUと組み合わせると

ーGTFSはワンソース・マルチユースー

PLATEAUによりランドマークが見える場所を算出し
GTFS-RTによりバスから見えるランドマークを通知する例



GTFS-RTによるバス位置情報をトリガーとすることで
スマホの位置情報を取得せず通知を送信できる



※趣味でチーム開発した事例です

GTFSは複数事業者に対応しやすい

複数事業者に対応しやすい

ーGTFSはワンソース・マルチユースー

BUTTER

tag-maker.butter.takoyaki3.com

ゲスト (2)

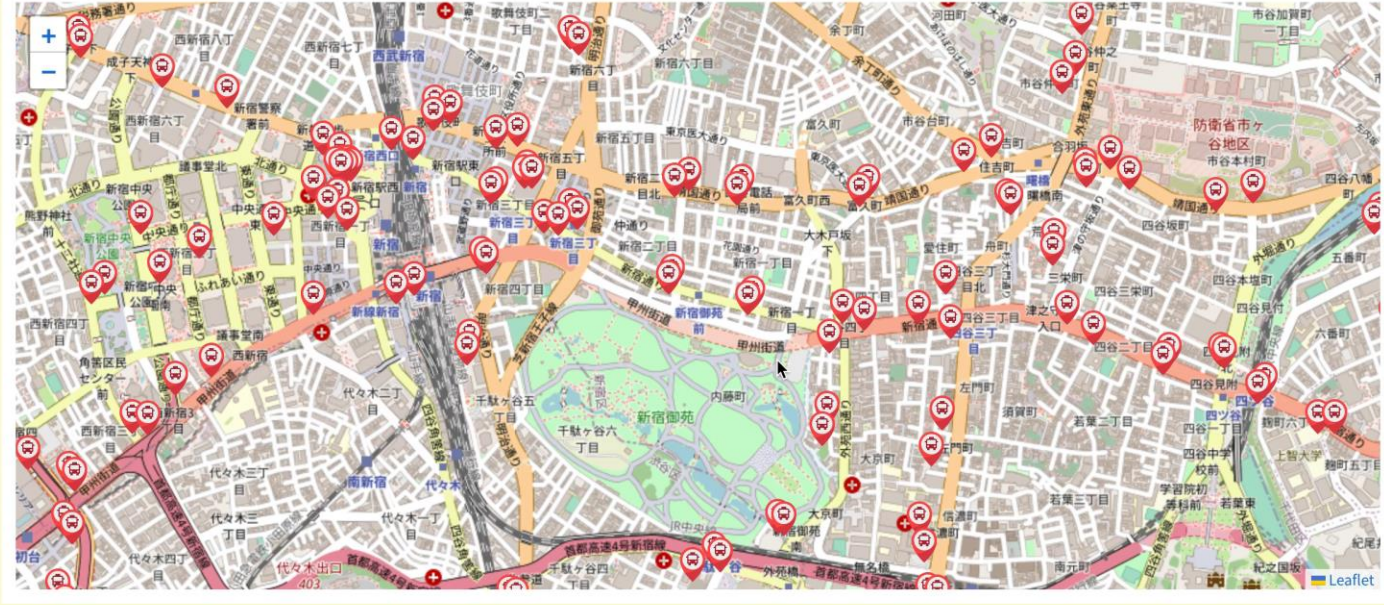
BuTTER Tag Maker

GITHUB

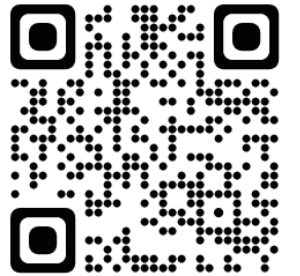
マップから選択 名前から選択

新宿御苑前

検索



プレビュー表示



<https://tag-maker.butter.takoyaki3.com>

BuTTER-Tag

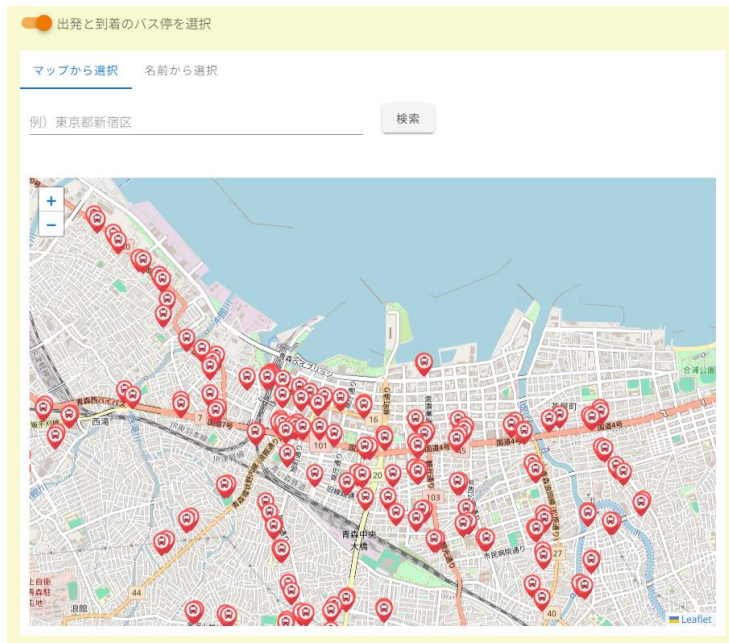
池澤隆人、伊藤尚紀、
桑原頌明、富川雄斗、
住谷祐太、羽田野湧太

※趣味でチーム開発した事例です

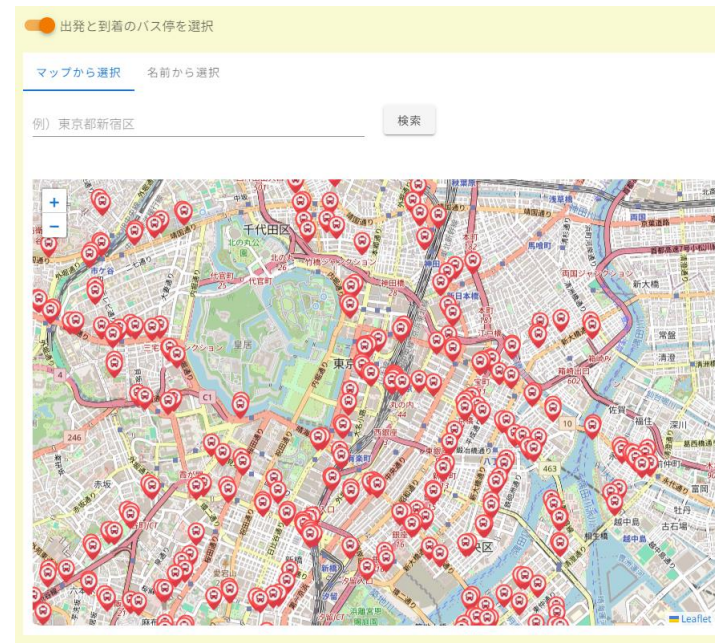
一度作れば複数事業者に対応

ーGTFSはワンソース・マルチユースー

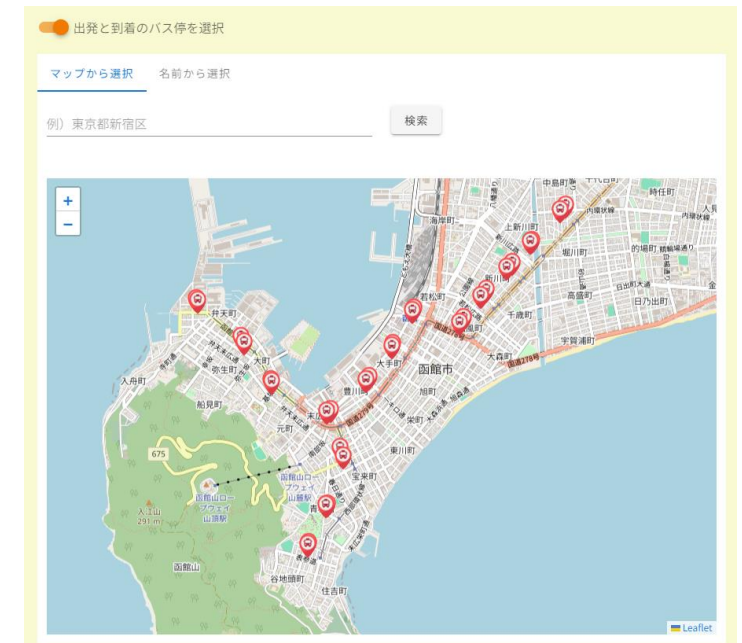
開発サービスを多地域・多事業者・マルチモーダルに展開可能



青森市営バス



都営バス



函館市電

※ODPT配信データを利用
※趣味でチーム開発した事例です

GTFSベースのメリット

—GTFSは世界的共通形式—

標準規格に基づいたデータが
入手できれば個別仕様に合わ
せた開発削減

標準規格に基づくのでドキュ
メントが豊富

OSS活用も選択肢に入る

生成AIのコーディング支援を
受けやすい

ChatGPT o1-preview

GTFSから本日運行されるバス一覧を取得するコードを書いてください



思考時間: 27 秒

GTFSデータから本日運行されるバス一覧を取得するPythonコードを以下に示します。

python

コードをコピーする

```
import pandas as pd
from datetime import datetime

# GTFSデータの読み込み
calendar = pd.read_csv('calendar.txt')
calendar_dates = pd.read_csv('calendar_dates.txt')
trips = pd.read_csv('trips.txt')
routes = pd.read_csv('routes.txt')

# 本日の日付を取得
today = datetime.today().strftime('%Y%m%d')
today_weekday = datetime.today().weekday() # 月曜日が0

# 曜日番号をGTFSの曜日名にマッピング
weekday_map = {
    0: 'monday',
    1: 'tuesday',
    2: 'wednesday',
```

ChatGPT にメッセージを送信する

GTFSベースのメリット

—GTFSは世界的共通形式—

標準規格に基づいたデータが
入手できれば個別仕様に合わ
せた開発削減

標準規格に基づくのでドキュ
メントが豊富

OSS活用も選択肢に入る

生成AIのコーディング支援を
受けやすい

リストアップされた本日運行されるバス一覧

	route_id	trip_id	service_id	route_short_name	route_long_name
0	3	00106-1-01-170-0910	01-170	市0 1	NaN
1	3	00106-1-01-170-0924	01-170	市0 1	NaN
2	3	00106-1-01-170-0941	01-170	市0 1	NaN
3	3	00106-1-01-170-0959	01-170	市0 1	NaN
4	3	00106-1-01-170-1017	01-170	市0 1	NaN
...
16036	191	74503-1-81-170-1932	81-170	陽1 2 - 2	NaN
16037	191	74503-1-81-170-2043	81-170	陽1 2 - 2	NaN
16038	191	74503-2-81-170-1926	81-170	陽1 2 - 2	NaN
16039	191	74503-2-81-170-2004	81-170	陽1 2 - 2	NaN
16040	191	74503-2-81-170-2117	81-170	陽1 2 - 2	NaN

16041 rows × 5 columns

GTFS/GTFS-RT
使ってみたくありませんか...？

ODPTのJRE GTFS で Open Trip Planner

README MIT license

公共交通オープンデータチャレンジ2024 JR東日本 OpenTripPlanner

JR東日本・JR東日本アイステーションズの運行情報とOpenStreetMapの地図情報を組み合わせ、リアルタイムな鉄道運行状況を反映した経路検索などができる OpenTripPlanner (OTP) を動かすコードです。

特徴

- Docker Composeを用いたシンプルな環境構築
- JR東日本のリアルタイム運行情報APIを活用
- OpenStreetMapの地図データと統合
- 経路検索APIを提供

インストール

必要環境

- Docker
- Docker Compose

手順

1. APIキーの取得: [ODPT](#) からAPIキーを取得してください。
2. APIキーの設定: `compose.yml` ファイルの `API_KEY` 環境変数を、取得したAPIキーに置き換えます。

```
version: '3'
services:
  downloader:
    environment:
      - API_KEY=[取得したAPIキー]
```

3. Dockerコンテナの起動: プロジェクトのルートディレクトリで以下のコマンドを実行します。



From To

35.63192 139.7440

Access

Not selected

Transit mode

Not selected

Previous page

Next page

1. 17:22 18:03

17:22 - 17:37

980 m, 14 mins 7 secs

foot [Shinagawa](#)

17:37 17:30 - 17:39 17:32

1.3 km, 2 mins

rail [null](#) [null](#), [JR東日本](#)
[Shinagawa](#) → [Osaki](#)

17:42 - 18:02

14.8 km, 20 mins

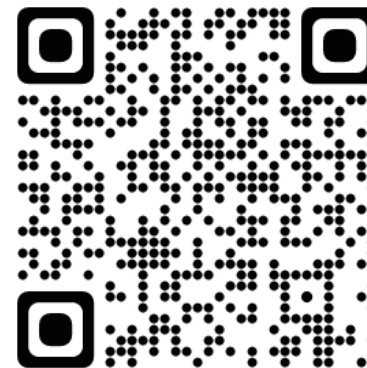
rail [null](#) [川越](#), [JR東日本](#)
[Osaki](#) → [Itabashi](#)

TECH



JR東日本のリアルタイムデータ
で乗換案内 (OTP) を動かす
-1.Docker起動編-

3日前 ♡ 3



<https://github.com/takoyaki-3/odpt-challenge2024-jre-otp-docker>

※個人的に作成したスクリプトです

交通データの輪を広げよう！

