

# **Transports publics**

## **Projet Business Intelligence**

Promo Social Builder Data 03/2024

# ORGANISATION DE L'ÉQUIPE



**Leili**  
Data analyste



**Marion**  
Data analyste



**Sana**  
Data analyste



**Wissal**  
Data analyste



**Audrey**  
Data analyste

# S o m m a i r e



**01**  
INTRODUCTION ET RAPPEL DU CONTEXTE

**02**  
PRESENTATION DES OUTILS ET DE LA DEMARCHE

**03**  
DÉMONSTRATION DU DASHBOARD

**04**  
RECOMMANDATIONS

**05**  
CONCLUSION



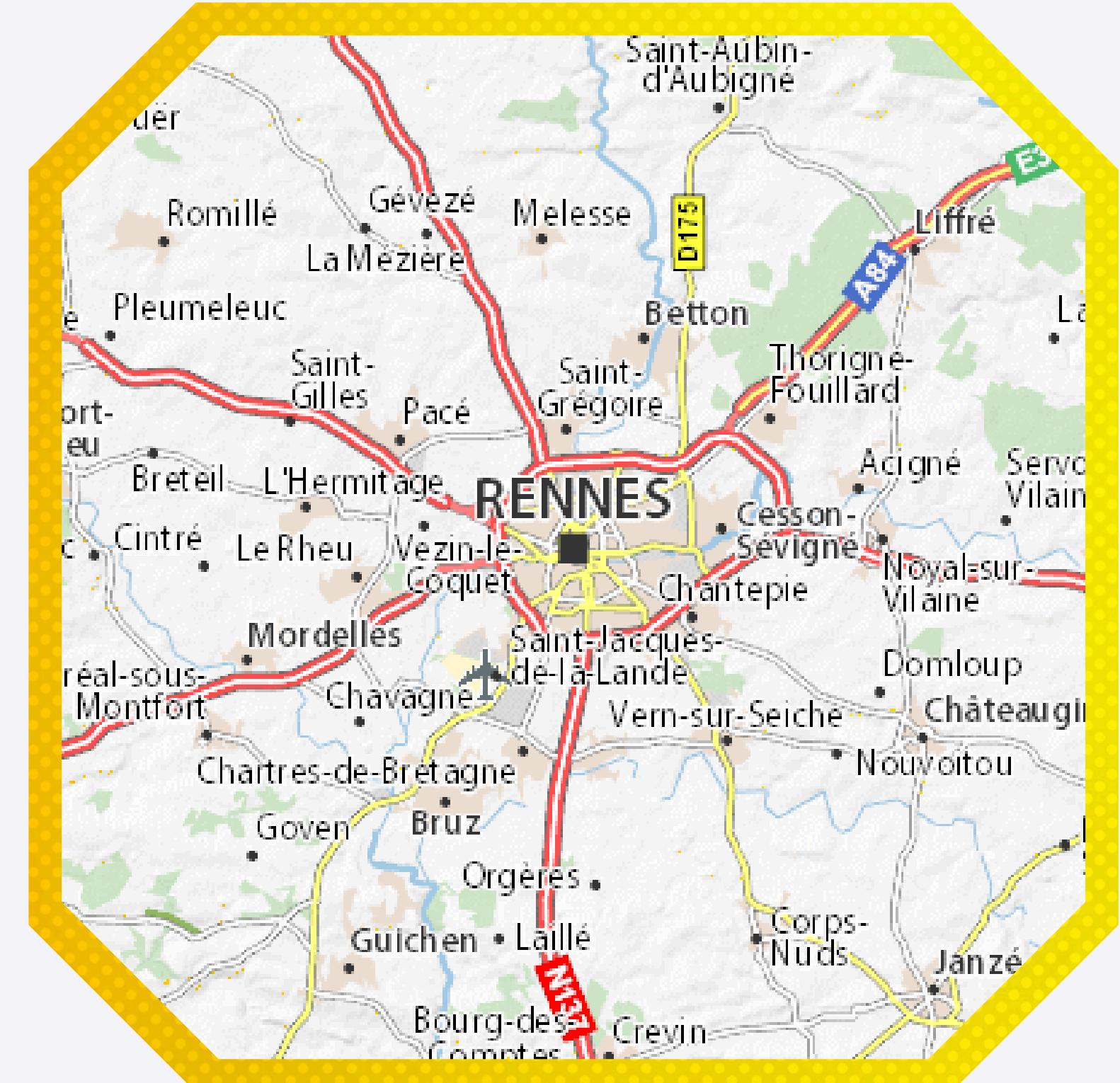
# Introduction et contexte

## • Choix de la ville de Rennes :

- Préfecture de la région Bretagne
- Nb habitants intra-muros = env. 270 000 hab.
- Nb habitants métropole rennaise = env. 475 000 hab. (regroupés dans 43 communes) (\*) => **12ème ville la plus peuplée de France**
- **Plus petite métropole d'Europe à bénéficier d'un métro mais aussi la ville la plus embouteillée de France (\*\*)**

## • Choix des moyens de transport :

- Bus
- Métro
- Réseau routier



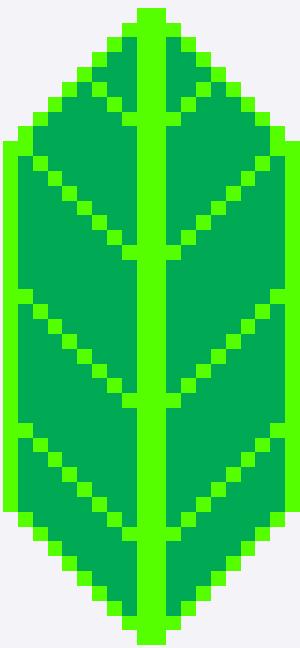
(\*) source INSEE 2021

(\*\*) source : article du Monde "Rennes, une ville-archipel à l'épreuve de la densité de sa population" de 2022

# Outils de travail



Trello



Power BI



# Trello

WCS - Projet 3 - Transports Publics Visible par l'espace de travail Tableau

The board consists of five columns:

- FAIT**: Contains tasks like "Choix du ou des types de transport", "Chercher des sources de données (ata.explore.star.fr)", "Récupération de données via API", "Automatisation toutes les 15 minutes de la collecte via API pendant 28h", "Réfléchir à des KPIs à afficher", "Exploration et Nettoyage des données", "Modifier le format des heures et des dates", "Constituer un historique de données", "Stockage de cet historique dans une base de données csv", "Supprimer les doublons", and "+ Ajouter une carte".
- EN COURS**: Contains the task "Tester la carte dynamique sur Pow BI avec API".
- A FAIRE**: Contains tasks like "Refaire un requêtage globale bus métro via API", "Récupération des données via API métro star", "Réfléchir à des KPI métro", "Finaliser les cartes", "Affichage d'une carte dynamique ou une interface web", "Présentation finale", "checker les valeurs aberrantes sur les retards", "checker si par arrêt les coordonnées géographiques sont bien uniques", and "+ Ajouter une carte".
- KPIs et Exploration de données**: Contains tasks like "l'écart entre l'arrivée réelle et l'arrivée théorique", "Nombre de courses par ligne", "Temps moyen d'attente à un arrêt", "Nombre de bus en service par période", "Fréquence des bus par ligne", "Variation du taux de ponctualité par période de la journée", "Fréquence des bus en fonction des heures de la journée", "Distribution des retards par ligne", "Temps moyen entre deux passages à un arrêt spécifique", and "Retard moyen par arrêt".
- Liste de visualisation à faire**: Contains tasks like "Afficher tous les arrêts sur une carte", "Tracer les trajets des lignes de transport en connectant les arrêts par des lignes", "Afficher les arrêts sur une carte et superposer les heures d'arrivée et de départ", "Carte qui permet de visualiser les trajets (tout les arrêts desservis, filtrer par ligne de bus)", "Afficher les heures passées et futurs de passage des bus et métros sur la même carte", and "+ Ajouter une carte".

# Open data Bus

Prochains passages des lignes de bus du réseau STAR en temps réel

Automatisation de la génération des URL pour les API bus du réseau STAR et exécution des requêtes



```
{  
    "total_count": 17568,  
    "results": [  
        {  
            "idligne": "0001",  
            "nomcourtligne": "C1",  
            "sens": 0,  
            "destination": "Chantepie",  
            "idarret": "2815",  
            "nomarret": "La Plesse",  
            "coordonnees": {  
                "lon": -1.686391,  
                "lat": 48.137081  
            },  
            "arriveetheorique": "2024-09-16T10:39:00+00:00",  
            "departtheorique": "2024-09-16T10:39:00+00:00",  
            "arrivee": "2024-09-16T10:39:00+00:00",  
            "depart": "2024-09-16T10:39:00+00:00",  
            "idcourse": "268435546",  
            "idbus": null,  
            "numerobus": null,  
            "precision": "Temps réel",  
            "visibilite": "Grand public",  
            "heureextraction": "11:35:10"  
        },  
        {  
            "idligne": "0001",  
            "nomcourtligne": "C1",  
            "sens": 0,  
            "destination": "Chantepie",  
            "idarret": "1328",  
            "nomarret": "Lycée Mendès France",  
            "coordonnees": {  
                "lon": -1.685248,  
                "lat": 48.132964  
            }  
        }  
    ]  
}
```



# Open data Metro

Prochains passages applicables des lignes de métro du réseau STAR

Automatisation de la génération des URL pour les API métro du réseau STAR et exécution des requêtes



```
{  
    "total_count": 2227,  
    "results": [  
        {  
            "idligne": "1001",  
            "nomcourtligne": "a",  
            "sens": 0,  
            "destination": "La Poterie",  
            "idarret": "5001",  
            "nomarret": "J.F. Kennedy",  
            "coordonnees": {  
                "lon": -1.71094311992153,  
                "lat": 48.1212952521027  
            },  
            "arrivee": null,  
            "depart": "2024-09-24T08:10:06+00:00",  
            "precision": [  
                "Applicable"  
            ],  
            "idrame": null  
        },  
        {  
            "idligne": "1001",  
            "nomcourtligne": "a",  
            "sens": 0,  
            "destination": "La Poterie",  
            "idarret": "5001",  
            "nomarret": "J.F. Kennedy",  
            "coordonnees": {  
                "lon": -1.71094311992153,  
                "lat": 48.1212952521027  
            },  
            "arrivee": null,  
            "depart": "2024-09-24T08:14:39+00:00",  
            "precision": [  
                "Applicable"  
            ]  
        }  
    ]  
}
```



# Open data Trafic

Etat du trafic en temps réel

Automatisation de la génération des URL pour les API routières du réseau

STAR à partir du site

data.rennesmetropole et exécution des requêtes



```
{  
    "total_count": 2859,  
    "results": [  
        {  
            "datetime": "2024-09-24T09:55:00+02:00",  
            "predefinedlocationreference": "10273_D",  
            "averagevehiclespeed": 76,  
            "traveltime": 10,  
            "traveltimereliability": 50,  
            "trafficstatus": "freeFlow",  
            "vehicleprobemeasurement": 1,  
            "geo_point_2d": {  
                "lon": -1.6502152435538264,  
                "lat": 48.044792756860865  
            },  
            "geo_shape": {  
                "type": "Feature",  
                "geometry": {  
                    "coordinates": [  
                        [  
                            [ [-1.651560961036608, 48.04488583237593],  
                                [-1.649942809885061, 48.04477391733652]  
                            ],  
                            [ [-1.648869526613244, 48.04469967350696]  
                            ]  
                        ]  
                    ]  
                }  
            }  
        }  
    ]  
},  
    "type": "MultiLineString"  
}
```

# Library Schedule

- » **### ÉTAPE 5 => IMPORT DE LA LIBRAIRIE SCHEDULE POUR AUTOMATISER LA REPETITION PERIODIQUE DES TÂCHES :**  
**### Démarche => on crée une fonction 'job' qui enchaîne les 3 fonctions précédemment créées en Etape 4.**

```
!pip install schedule
import schedule
import time

def job():
    """
    Fonction de tâche à exécuter toutes les 15 minutes.
    """

    # df = fetch_and_process_data(url)
    df_combined_bus = fetch_and_concat_data(urls_bus)
    df_combined_metro = fetch_and_concat_data(urls_metro)
    df_combined_routier = fetch_and_concat_data(urls_routier)
    save_dataframe_with_timestamp(df_combined_bus, 'csv', 'bus')
    save_dataframe_with_timestamp(df_combined_metro, 'csv', 'metro')
    save_dataframe_with_timestamp(df_combined_routier, 'csv', 'routier')

    # Planifier la tâche toutes les 15 minutes
    schedule.every(15).minutes.do(job)

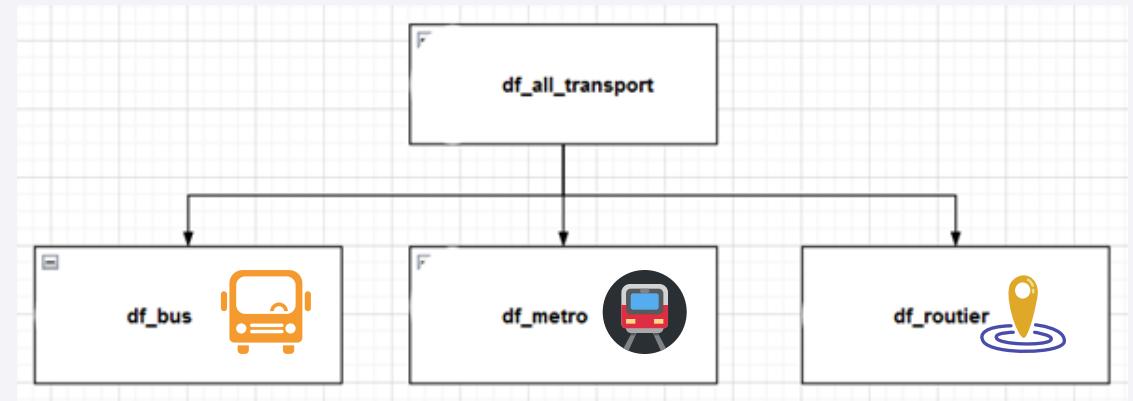
    # Boucle infinie pour exécuter le scheduler
    while True:
        schedule.run_pending()
        time.sleep(1) # Pause de 1 seconde avant de vérifier la prochaine tâche
```

# Dictionnaire des données

Colonne	df_bus	df_metro	df_routier	Description
idligne	290673 non-null, int64	16970 non-null, int64	231992 non-null, object	Identifiant unique de la ligne de transport.
nomcourtligne	290673 non-null, object	16970 non-null, object	231992 non-null, object	Nom court de la ligne.
sens	290673 non-null, int64	16970 non-null, int64	0 non-null, float64	Sens de circulation (ex : aller/retour).
destination	289749 non-null, object	16970 non-null, object	231992 non-null, object	Destination de la ligne.
idarret	290673 non-null, int64	16970 non-null, int64	0 non-null, float64	Identifiant unique de l'arrêt.
nomarret	290673 non-null, object	16970 non-null, object	0 non-null, float64	Nom de l'arrêt.
lon	290673 non-null, float64	16970 non-null, float64	231992 non-null, float64	Longitude de l'arrêt.
lat	290673 non-null, float64	16970 non-null, float64	231992 non-null, float64	Latitude de l'arrêt.
arriveetheorique	290673 non-null, object	0 non-null, float64	0 non-null, float64	Heure d'arrivée théorique.
departtheorique	290673 non-null, object	0 non-null, float64	0 non-null, float64	Heure de départ théorique.
arrivee	290673 non-null, object	15241 non-null, object	0 non-null, float64	Heure d'arrivée réelle.
depart	290673 non-null, object	12775 non-null, object	0 non-null, float64	Heure de départ réelle.
idcourse	290673 non-null, int64	0 non-null, float64	0 non-null, float64	Identifiant unique du cours (trajet).
precision	290673 non-null, object	16970 non-null, object	0 non-null, float64	Précision des données (ex : fiable, incertain).
visibilite	290673 non-null, object	0 non-null, float64	0 non-null, float64	Niveau de visibilité des données.
heureextraction	290673 non-null, object	0 non-null, float64	231992 non-null, object	Heure d'extraction des données.
trafficstatus	0 non-null, float64	0 non-null, float64	231992 non-null, object	État du trafic (aucune donnée disponible dans df_bus et df_metro).
averagevehiclespeed	0 non-null, float64	0 non-null, float64	231992 non-null, int64	Vitesse moyenne des véhicules (aucune donnée disponible dans df_bus et df_metro).
vitesse_maxi	0 non-null, float64	0 non-null, float64	231992 non-null, int64	Vitesse maximale observée (aucune donnée disponible dans df_bus et df_metro).
date_heure_requete_API	290673 non-null, object	16970 non-null, object	231992 non-null, object	Date et heure de la requête à l'API.
transport_API	290673 non-null, object	16970 non-null, object	231992 non-null, object	Type de transport (ex : bus, métro, routier).

# Nettoyage et traitement des données récoltées

Étape de Nettoyage	Problèmes Identifiés	Solution
1. Valeurs manquantes	<ul style="list-style-type: none"> <li>- destination dans df_bus a des valeurs manquantes.</li> <li>- arriveetheorique, departtheorique dans df_metro sont vides.</li> <li>- arrivée, départ dans df_routier sont entièrement vides.</li> </ul>	<ul style="list-style-type: none"> <li>- Suppression des colonnes vides.</li> <li>- Imputation des valeurs manquantes pour les colonnes importantes (ex. mode ou valeur par défaut).</li> </ul>
2. Doublons	- Possibilité de doublons dans les trois DataFrames.	- Déduplication des enregistrements pour éviter la redondance.
3. Valeurs aberrantes (outliers)	<ul style="list-style-type: none"> <li>- Latitude (lat) et Longitude (lon) peuvent contenir des valeurs hors des bornes géographiques valides.</li> <li>- Cohérence des horaires (départ, arrivée, arriveetheorique)</li> </ul>	- Filtrage des valeurs aberrantes en vérifiant que lat est entre -90 et 90, et lon entre -180 et 180.
4. Incohérences de format	- Colonnes comme arriveetheorique, departtheorique dans df_bus et df_metro sont au format texte, mais devraient être au format date/heure.	- Conversion des colonnes texte en format datetime pour un traitement correct.
5. Suppression des colonnes inutiles	<ul style="list-style-type: none"> <li>- Certaines colonnes comme trafficstatus, averagevehiclespeed, et vitesse_maxi sont vides dans df_bus et df_metro.</li> <li>- Les colonnes comme arriveetheorique, departtheorique dans df_metro et df_routier ne contiennent aucune donnée utile.</li> </ul>	<ul style="list-style-type: none"> <li>- Suppression des colonnes entièrement vides ou non pertinentes pour l'analyse.</li> </ul>
6. Imputation	- Colonnes partiellement remplies mais importantes comme destination dans df_bus.	<ul style="list-style-type: none"> <li>- Imputation de valeurs manquantes, par exemple en remplaçant les valeurs manquantes par la valeur la plus fréquente (mode).</li> </ul>
7. Fusion des DataFrames	- Création d'un DataFrame consolidé contenant les données de bus, métro et routier.	<ul style="list-style-type: none"> <li>- Utilisation de pd.concat() pour fusionner df_bus, df_metro, et df_routier en un seul DataFrame df_all_transport.</li> </ul>
8. Validation finale	- Vérification de la qualité des données après nettoyage et fusion.	<ul style="list-style-type: none"> <li>- Inspection du DataFrame df_all_transport pour rechercher d'éventuelles erreurs résiduelles et confirmer que les données sont prêtes pour l'analyse.</li> </ul>
9. Crédit à l'original	Obj. Pouvoir avoir des axes d'analyse pertinents pour les KPI donc nécessité d'agrégger à la maille idcourse (trajet) // idligne (ligne de transport) // destination (spécifique aux metros)	
10. Documentation	- Documenter les étapes de nettoyage et les choix effectués pour assurer la traçabilité et la reproductibilité des analyses.	<ul style="list-style-type: none"> <li>- Créer un rapport de nettoyage pour chaque DataFrame, consignant les colonnes supprimées, les valeurs imputées, etc. afin d'assister dans l'analyse des KPI, ainsi que dans la création de visuels et de cartes dans Power BI.</li> </ul>

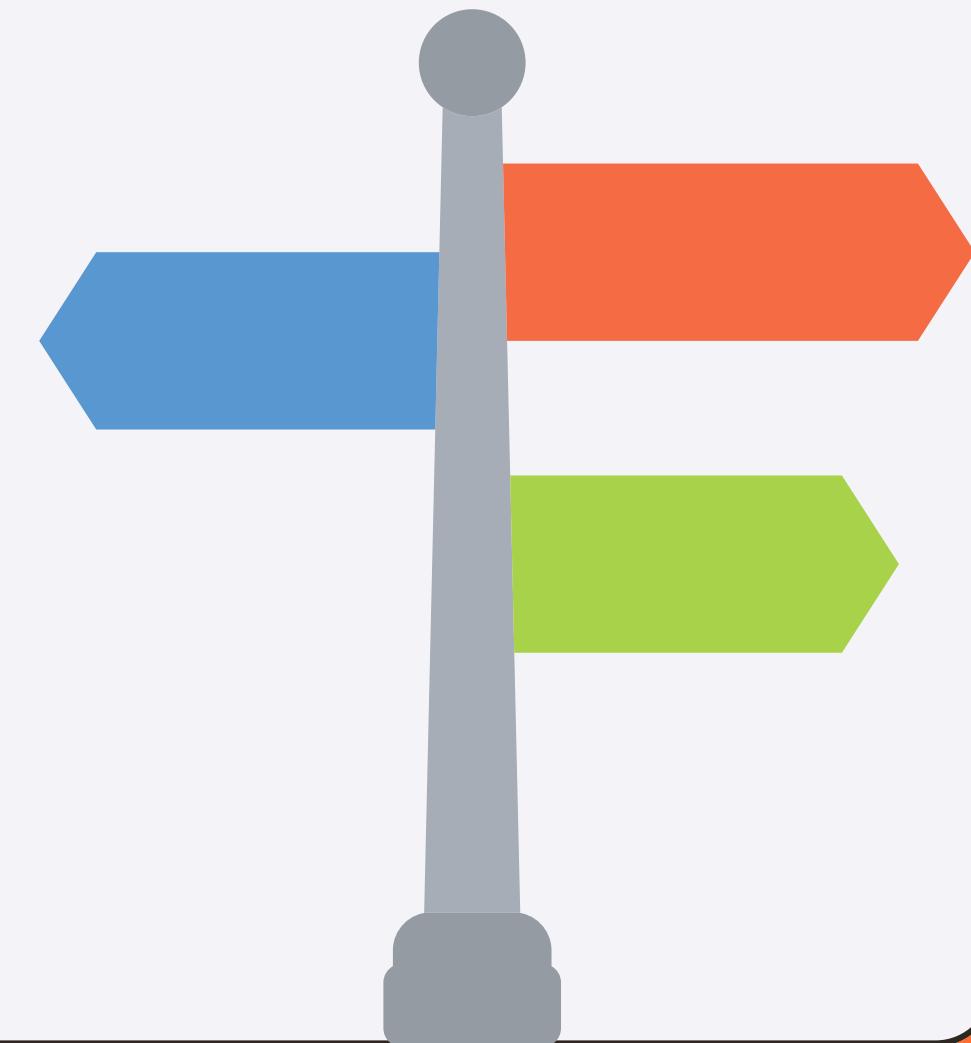


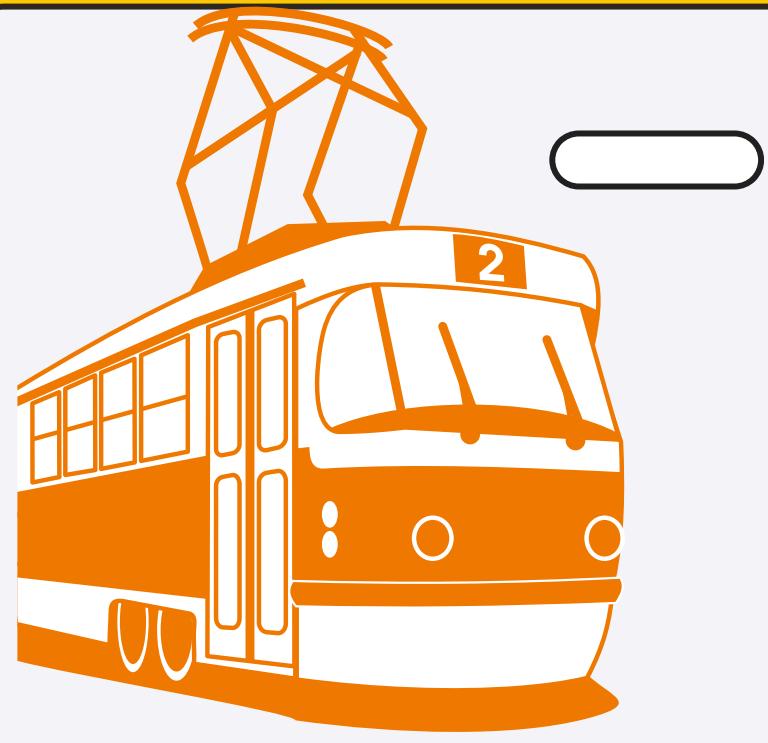
# DÉMONSTRATION



# Recommandations

-  Ajouter des trajets de bus (banlieue -> Rennes et banlieue -> banlieue)
-  La mise en place de voies réservés aux bus
-  L'ouverture des lignes Tram bus
-  Rallongement des lignes de métro





merci!

