

Documentation fonctionnelle de PRIM

**La Plateforme Régionale
d'Information pour la Mobilité**

V1.0

Sommaire

01 Préambule

02 Données disponibles

- Données statiques
- Données dynamiques

03 Prise en main des APIs

- Calculateur Île-de-France Mobilités – Accès générique (Navitia)
- Applications d'accès aux APIs Temps réel
- Applications d'accès aux APIs Temps réel – source plateforme Ile-de-France Mobilités
- Applications d'accès aux APIs Temps réel – source service Ile-de-France Mobilités

04 Lien entre les données

05 Support & documentation

Le rôle d'Île-de-France Mobilités

Île-de-France Mobilités est l'**Autorité Organisatrice de la Mobilité Durable (AOMD)** en Ile-de-France

Elle imagine, organise et finance les transports publics pour tous les Franciliens.

L'établissement est **également garant de la qualité de l'information voyageurs délivrée (offre de transport, horaires, tarification et informations de perturbation)**.

IDF Mobilités dispose actuellement d'un Système d'Information dédié à la collecte, au stockage et à la diffusion des données de Transport Public. Ce SI est **composé d'une quinzaine d'applications** interfacées entre elles qui proposent in fine :

- Un ensemble de services aux voyageurs au travers des sites Web & Mobile ;
- De la donnée et des API à destination des développeurs/ré-utilisateurs à travers le site prim.iledefrance-mobilites.fr

La cible métier de l'IV (Information Voyageurs) doit s'inscrire dans **une trajectoire globale** vers le **MaaS (Mobility-as-a-Service)**. Cette étape essentielle consiste à aboutir à la réalisation d'une **Plateforme Régionale d'Information pour la Mobilité (PRIM)** qui

- :
- Agrègera l'ensemble de l'offre IV en matière de services et de données ;
 - Alimentera plusieurs Front Offices dédiés aux différents publics, prioritairement ceux d'Ile-de-France Mobilités, mais éventuellement aussi à d'autres Front Offices clients.

Données disponibles

Données statiques

Le catalogue data

Sur le catalogue des données statiques, vous trouverez plusieurs jeux de données classés selon différents thèmes. Vous pouvez consulter la liste des jeux de données disponibles et accéder à leur page d'informations.

Arrêts et Lignes

25 jeux de données dont les référentiels, les fiches horaires, les couches géographique et schématiques



Travaux

2 jeux de données sur les infos trafic ainsi que sur les cartes schématiques mensuels des travaux



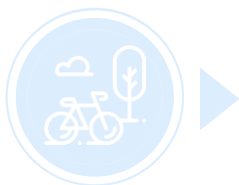
Données administratives

3 jeux de données sur les différentes informations administratives,



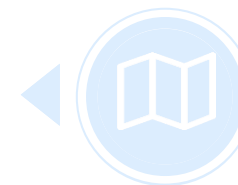
Nouvelles Mobilités

6 jeux de données, principalement des données « Vélo »



Cartes et plans

2 jeux de données sur les plans du réseau régional de transports en Île-de-France



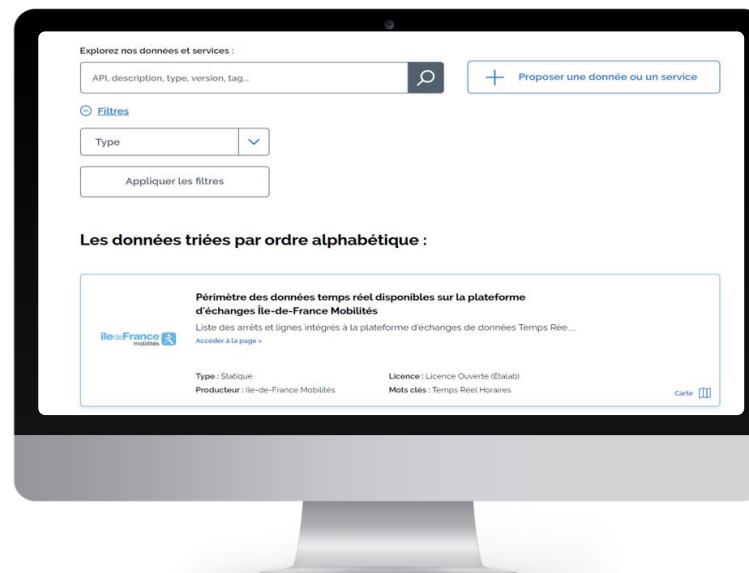
Horaires & Tarifs

3 jeux de données décrivant l'offre de transport disponible ainsi que les tarifs de transport



Fréquentations

11 jeux de données présentant les données de validations des différents réseaux



Visualisation des données

Le service des données statiques offre la fonctionnalité de visualisation des données, à travers des cartes ou des vues personnalisées.

La Data visualisation PRIM en quelques chiffres



20

Jeux de données disposant d'une carte associée



4

Jeux de données complétés par des images (des plans du réseau ou des pictogrammes)



6

Jeux de données avec des vues personnalisées développées par nos référents

Données disponibles

Données dynamiques

Les différentes APIs

Prochains passages Temps réel

Prochains Passages plateforme IDFM

(requête globale)

Prochains horaires de passages en un seul échange de l'ensemble des arrêts du réseau actuellement disponibles.

Prochains Passages plateforme IDFM

(requête unitaire)

Prochains passages à un arrêt ou sur une ligne.

Prochains Passages source Via Navigo

Prochains passages aux arrêts ou stations des lignes Transdev, RATP et SNCF.

Calcul d'itinéraires théoriques

Calculateur Ile-de-France Mobilités

(horaires, itinéraires, structure du réseau, lignes et arrêts,...)

Calcul d'itinéraires théoriques

Calculateur Ile-de-France Mobilités

Toutes les Informations trafic présentes dans Ile-de-France mobilités

Messages Info Trafic plateforme IDFM

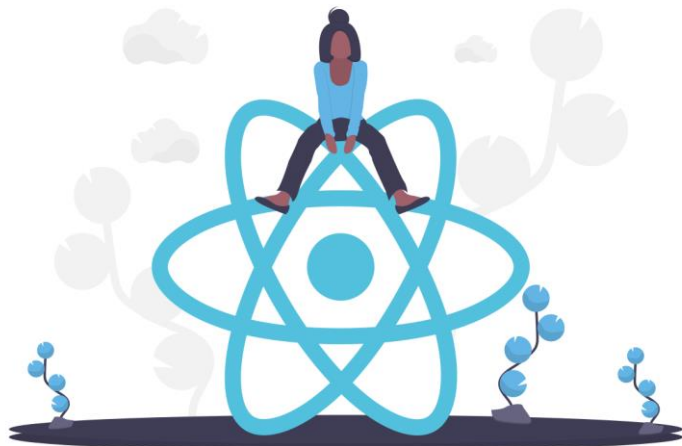
Messages de perturbations RATP, SNCF, ALBATRANS

Différence entre API et application

Application

Une application regroupe l'accès à un **groupe d'API**.

Elle permet de mettre en place un **quota** de requêtes.



API

Une API (Application Programming Interface) est une interface de programmation qui permet de se « **brancher** » sur une application pour **échanger** des données.

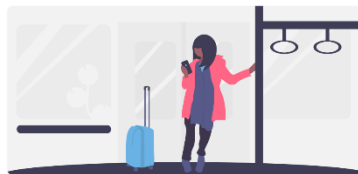
L'API permet la **consommation** de données après avoir été liée à l'application lui correspondant.

Applications

Chacune des trois applications permet de gérer l'accès aux APIs selon leur nature.

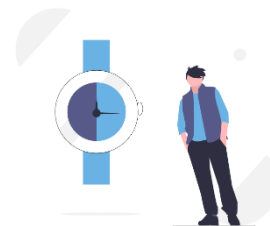
01-Unitary-Real-Time_email

regroupe l'accès aux API Temps Réel (par requêtes unitaires) dans la limite d'un quota de **1 000 000 requêtes / jour**.



02-Global-Real-Time_email

donne l'accès à l'API Temps Réel de prochains passages (par requête globale) dans la limite d'un quota de **1 000 requêtes / jour**.



03-Journey-Planner_email



donne l'accès à l'API Calculateur d'itinéraires Île-de-France Mobilités dans la limite d'un quota de **20 000 requêtes / jour**.

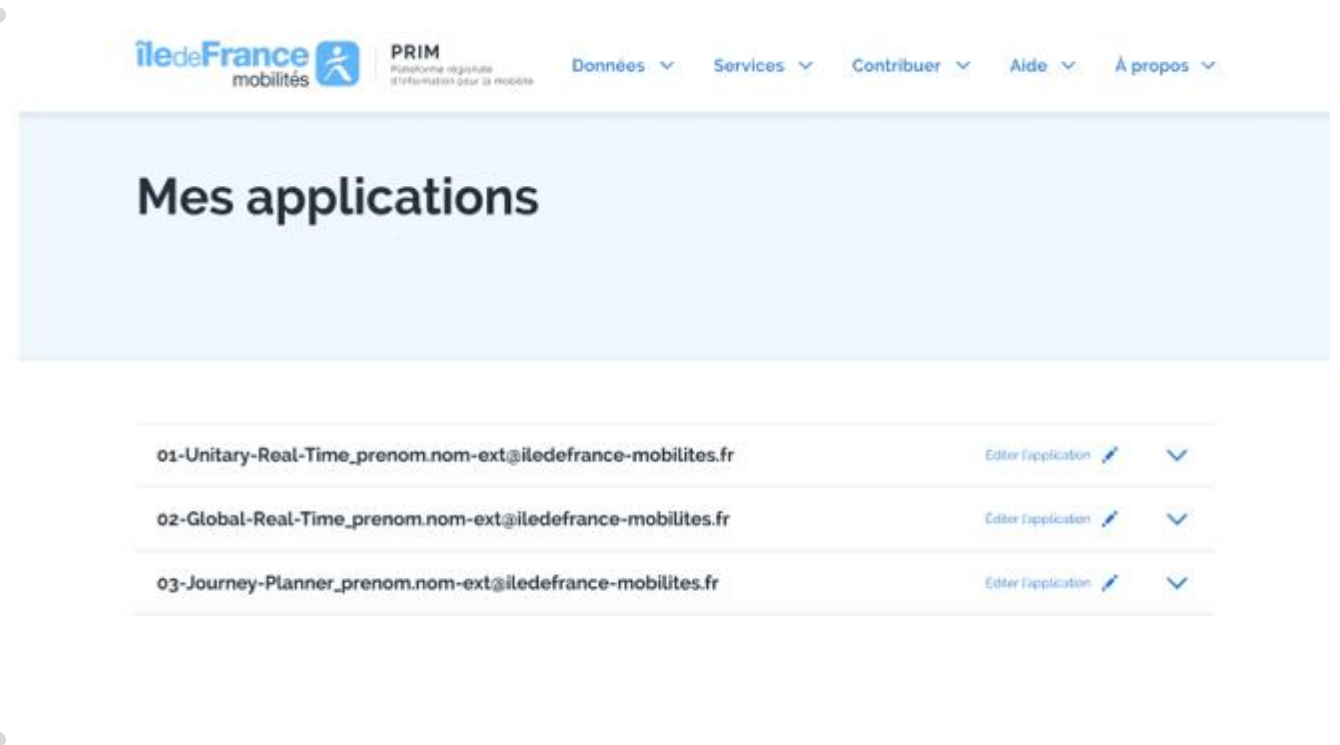


Lier les API à vos applications







Avant toute première utilisation, il est nécessaire de lier les API à vos applications.

Pour cela, vous devez :

- 1 Aller dans l'onglet « » et cliquer sur « Mes applications »
- 2 Cliquer sur [Éditer l'application](#) 
- 3 Cocher les APIs à lier à l'application
- 4 Cliquer sur 



The screenshot shows the 'Mes applications' page of the Île de France Mobilités portal. The page header includes the logo 'île de France mobilités', the text 'PRIM Plateforme régionale d'information pour la mobilité', and navigation links for 'Données', 'Services', 'Contribuer', 'Aide', and 'À propos'. The main heading is 'Mes applications'. Below this, there is a table with three rows, each representing an application and its API endpoint. Each row has an 'Éditer l'application' button with a pencil icon and a dropdown arrow.

Application	API Endpoint	Action
01-Unitary-Real-Time_prenom.nom-ext@iledefrance-mobilites.fr	01-Unitary-Real-Time_prenom.nom-ext@iledefrance-mobilites.fr	Éditer l'application  
02-Global-Real-Time_prenom.nom-ext@iledefrance-mobilites.fr	02-Global-Real-Time_prenom.nom-ext@iledefrance-mobilites.fr	Éditer l'application  
03-Journey-Planner_prenom.nom-ext@iledefrance-mobilites.fr	03-Journey-Planner_prenom.nom-ext@iledefrance-mobilites.fr	Éditer l'application  

Lier les API à vos applications

Pour savoir si une API est bien liée à une application, vous pouvez :

- 1 Aller dans l'onglet « Catalogue »
- 2 Choisir l'API que vous voulez tester
- 3 Cliquer sur « ASSISTANT REQUÊTAGE »



✓ Si l'API est **bien liée** à une application, le formulaire suivant est accessible :

Vous devez remplir les conditions d'authentification avant de pouvoir essayer les méthodes ci-dessous.

Access Token

OAuth credentials :

01-Unitary-Real-Time_Leo.welhoff-ext@iledefrance-mobilites.fr -

FLOW OAUTH

✗ Si l'API n'est **pas liée** à une application, le message d'alerte suivant apparaît :

✗ Pour tester les méthodes une authentification OAuth est requise. Vous devez créer une application et générer des credentials OAuth sur celle-ci. Si une application existe déjà, assurez-vous que votre API est bien liée à cette dernière.

Demande d'accès



Attention votre Token n'est valable qu'une heure après sa génération

Dans le formulaire de demande d'Access Token :

- 1 Sélectionner « Client credentials » comme Flow Oauth.
- 2 Cliquer sur « Demande d'Access Token »

Access Token

OAuth credentials :
01-Unitary-Real-Time_leo.welhoff-ext@iledefrance-mobilites.fr - [Client ID]

FLOW OAUTH

Sélectionner

FLOW OAUTH

- Authorization code
- Client credentials**

Access Token

OAuth credentials :
01-Unitary-Real-Time_leo.welhoff-ext@iledefrance-mobilites.fr - [Client ID]

FLOW OAUTH

Client credentials

Demander l'Access Token

- 3 Cliquer sur « Autoriser »

Autoriser l'accès

Axway API Portal souhaite accéder au scope **read-data**.

Autoriser Annuler

- 4

ACCESS TOKEN

Votre Token

Copier × Supprimer Access Token

Trouver les identifiants client

Afin d'automatiser le requêtage des APIs, nous vous proposons d'utiliser un script (ex: Python). Pour cela, il est nécessaire de récupérer d'abord les informations suivantes : vos identifiants client (**Client Id et Client secret**), l'**URL de l'API** et les paramètres nécessaires pour la requêter.

Vous trouverez : vos **identifiants client** en cliquant sur une application dans la sous-partie « General OAuth ».

API sélectionnées

Nom API	Description
Messages Info Trafic (plateforme Île-de-France Mobilités)	Messages d'information sur l'état du trafic en temps réel par ligne ou par arrêt. / Ce service donne l'ensemble des informations concernant les ...
Prochains passages (plateforme Île-de-France Mobilités) - requête unitaire	Horaires estimés des prochains passages aux arrêts. / Actuellement, les horaires de prochains passages à un arrêt en temps réel ne sont ...

General OAuth

Client ID	Type	URL d'authentification	Actions
<input type="text" value="Votre Client ID"/>	Confidentiel	https://as.api.iledefrance-mobilites.fr/api/oauth/token	Voir la clé secrète

Les identifiants client (ou « General OAuth ») permettent de vous faire reconnaître lors de la demande d'un token



Données disponibles : données dynamiques

Trouver l'URL et les paramètres d'une API

L'URL pour requêter une API est la concaténation du champs **Host** suivi du **Path** de la méthode désirée. Ainsi, pour requêter le **Calculateur Ile-de-France mobilités** en utilisant la méthode : **get_coverage_region_vehicle_journeys** :

- Le **Host** est: <https://traffic.api.iledefrance-mobilites.fr>
- L'**URL** est donc : https://traffic.api.iledefrance-mobilites.fr/v2/mri/coverage/idfm/vehicle_journeys

Method	Path	Nickname
▼ GET	/v1/tr-unitaire/stop-monitoring	getPassages
▲ GET	/v1/mri/coverage/fr-idf/vehicle_journeys	get_coverage__region__vehicle_journeys

En cliquant sur ▼ les paramètres s'affiche en-dessous de chaque méthode

Paramètres de la requête

Les différents paramètres d'une méthode sont définis dans la partie « Paramètres de la requête »

PARAMÈTRE 1	PARAMÈTRE 1
<input type="text" value="Sélectionner"/>	<input type="text"/>
PARAMÈTRE 1	PARAMÈTRE 1
<input type="text" value="exemple facultatif"/>	<input type="text" value="Sélectionner"/>

Exécuter

Réinitialiser



Exemples de code Python pour obtenir un token

```
#####  
# Récupération d'un Token  
# coding: utf8  
#####  
1 import requests  
  import json  
  import pprint  
  
2 urlOAuth = 'https://as.api.iledefrance-mobilites.fr/api/oauth/token'  
3 client_id= ""'Votre client id'"",  
4 client_secret= ""'Votre client secret'"",  
5  
6 #Paramètres de la requête de demande de token  
7 data = dict(  
8     grant_type='client_credentials',  
9     scope='read-data',  
10    client_id=client_id,  
11    client_secret=client_secret  
12 )  
13 response = requests.post(urlOAuth, data=data)  
14 print(response.json)  
15  
16 #Vérifier le code retour de la requête  
17 if response.status_code != 200:  
18     print('Status:', response.status_code, 'Erreur sur la requête; fin de programme')  
19     exit()  
20  
21 jsonData = response.json()  
   token = jsonData['access_token']
```

Ligne 1 – Importation de la bibliothèque requests

Ligne 2 – Affecter l'url du token à la variable *urlOAuth* (string)

Ligne 3 – Affecter votre client Id à la variable *client_id* (string)

Ligne 4 – Affecter votre client secret à la variable *client_secret* (string)

> [Trouver mon client id et mon client secret ? //lien](#)

Lignes 6 - 11 – Affecter à *data* un dictionnaire qui va définir comment accéder au token avec comme variables

grant_type(string) doit être défini à 'client_credentials'

scope(string) doit être défini à 'read-data' pour lire les données

client_id(string) doit être rempli par votre client ID

client_secret(string) doit être rempli par votre client secret

Ligne 13 – La fonction `requests.post` prend deux arguments en entrée l'url du token et la manière dont on souscrit à ce token défini dans *data*. Pour utiliser cette fonction la bibliothèque requests doit être installée. La réponse est stockée dans *response* (response – objet décrit par la bibliothèque Requests)

Ligne 14 – Affichage de la réponse dans le format json

Lignes 16 - 18 – Test du statut de la requête : Si *status_code* (integer) est différent de 200 (OK) alors affichage du statut et d'un message d'erreur puis fin du programme.

Ligne 20 – Si *status_code* est égal à 200 alors stockage de la réponse au format json dans *jsonData* (dictionnaire)

Ligne 21 – Stockage du token à utilisé dans la variable *token* grâce à la réponse *jsonData*

Exemples de code Python pour requêter une API

```
#####  
# Requête l'API Message Info trafic  
# coding: utf8  
#####  
22 url = 'https://traffic.api.iledefrance-mobilites.fr/v1/tr-messages-it'  
  
24 params =dict(  
    LineRef='ALL'  
)  
  
28 headers = {  
    'Accept-Encoding' : 'gzip',  
    'Authorization' : 'Bearer ' + token  
31 }  
  
33 response = requests.get(url, params=params, headers=headers)  
  
35 if response.status_code != 200:  
37     print('Status:', response.status_code, 'Erreur sur la requête; fin de programme')  
    exit()  
  
39 jsonData = response.json()
```

Ligne 22 – Affecter l'url de l'API à la variable url.

> [Trouver l'url de l'API ?](#)

Ligne 24 - 26 – Affecter à params un dictionnaire qui va définir les paramètres de la requête. Cela permet de filtrer la requête par exemple.

> [Comment connaître les différents paramètres possibles ?](#)

Ligne 28 - 31 – Affecter à headers un dictionnaire qui va définir les encodages de la requête.

Ligne 33 – La fonction requests.get prend trois arguments en entrée l'url de l'API et les paramètres de la requête et les headers de celle-ci.

Lignes 35 - 37 Test du statut de la requête : Si *status_code* (integer) est différent de 200 (OK) alors affichage du statut et d'un message d'erreur puis fin du programme.

Ligne 39 – Si *status_code* est égal à 200 alors stockage de la réponse au format json dans *jsonData* (dictionnaire)

Différents codes de réponse possibles



200

La requête a abouti et vous recevez une réponse



401

Vous n'êtes pas autorisé à accéder à l'API

403

→ Nous vous conseillons de renouveler votre token et de vérifier que l'API est bien liée à votre application

404

Votre requête contient une erreur

→ Nous vous conseillons de vérifier vos paramètres d'appel

429

Vous avez atteint votre quota

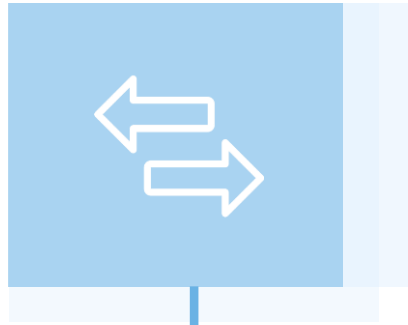
→ Vous pouvez accéder à toutes les informations sur les quotas depuis la page d'offre

500

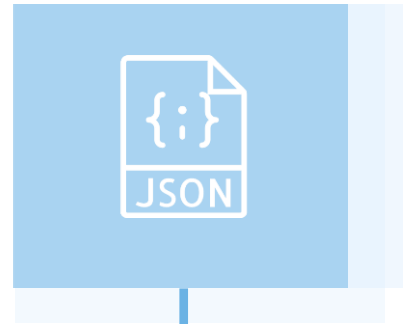
API est indisponible

→ Contactez-nous !

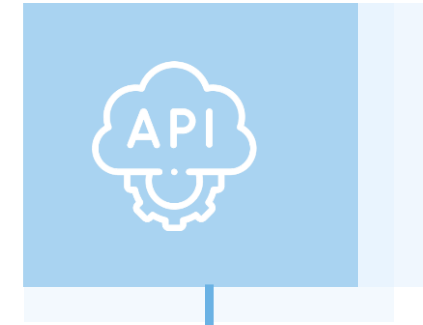
Bonnes pratiques



Les quotas d'appels sont à la journée : il est fortement recommandé de **lisser le nombre de requêtes** dans le temps afin de ne pas perturber le fonctionnement pour les autres utilisateurs.



Compte tenu de la taille très importante de la réponse à la requête globale « Prochains Passages », elle sera transmise en **mode compressé**.



Il est recommandé d'activer dans les headers de vos requêtes l'élément « **Accept-encoding : gzip, deflate** » afin d'optimiser le temps de réponse de l'API.

Prise en main des APIs

Calculateur Île-de-France Mobilités – Accès générique
(Navitia)

Calculateur Ile-de-France Mobilités – Accès générique (Navitia)



Qualité des données



La mise à jour des données est réalisée **chaque mardi et vendredi**.
Les données couvrent les **trois prochaines semaines**.

L'**intégralité des horaires** des RER A et B sont transmis par la **SNCF**.



Les horaires des **lignes SNCF incluent les adaptations faites lors de travaux prévus**, ainsi que les substitutions faites en bus (ex : un trajet en bus sera proposé si un tronçon est coupé pour cause de travaux).

Les horaires **des lignes RATP n'incluent pas** en règle générale, **les adaptations faites lors de travaux prévus**, ni les navettes de bus de substitution mises en place (ex : Fermeture d'une station de métro).

Contrairement aux fichiers issus du GTFS, **les données des API prennent en compte les Interdictions de Trafic Local (ITL)** qui existent parfois sur certaines lignes de bus OPTILE.



Paramètres d'appel



route

Une (« **route** ») est un ensemble d'itinéraires regroupés sous un même nom, cet objet fait référence à l'identifiant de la ligne disponible dans le [référentiel des lignes](#)

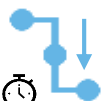
Ex : IDFM:C01371 (Métro 1)



trip

Une (« **trip** ») définit un parcours sur une ligne donnée, il s'agit d'un enchaînement structuré d'arrêts.

IDFM:RATP:9541-C01371-10_3736592_955496 (Métro 1 – Sens la Défense)



service

Une course (« **service** ») est la déclinaison d'un itinéraire à un horaire donné. Une course attribue à chaque arrêt de l'itinéraire un horaire de passage.

Sur une journée, une course est unique : deux véhicules d'une même ligne effectuent chacun une course différente.

Ex : IDFM:100071 (Métro 1 - une course qui circule que les samedis du 21 août 2021 au 28 août 2021)

Paramètres d'appel



agency

Une (« **agency** ») est un réseau commercial de transport

Ex : IDFM:Operator_100 (RATP)



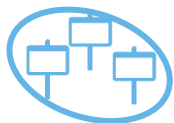
stop

Un (« **stop** ») est un objet qui décrit un type d'arrêt, il s'agit de :

1/ un point d'arrêt d'un seul et unique transporteur, pour ce cas il prend les formes suivantes :

- StopPoint Ferré (RER, Transilien): «IDFM:[ID ZDL] »
- StopPoint (reste): « IDFM:[ID ZDeR] »

Ex : IDFM:3640 (Jean de La Fontaine)



2/ un (« **stop_area** ») est un regroupement d'arrêts physiques portant le même nom dit aussi un arrêt commercial:

- StopArea : «IDFM:[ID ZDC]»

Ex : IDFM:70604 (Porte de Choisy)



stop_point est une donnée non pérenne pour le bus.

Elle est susceptible d'être changée chaque semaine à la publication du GTFS.

Les **stop_points** sont stables pour le ferré.

Utiliser le paramètre {URI} dans l'API générique Navitia

Le paramètre {URI} signifie que les paramètres d'appel sont intégrés dans l'URL de la requête et ne se trouve pas, comme la plupart du temps, à la fin de l'URL de requête.

Exemple 1

Objectif :

Récupérer tous les stop_points de la ligne du métro 1 (line:IDFM:C01373)

On utilise la méthode : `v2/mri/coverage/idfm/{uri}/stop_points/{id}`

Avec le paramètre d'appel à la place de {id}: `lines%2Fline:IDFM:C01373`

Prise en main des APIs

Applications d'accès aux APIs Temps réel

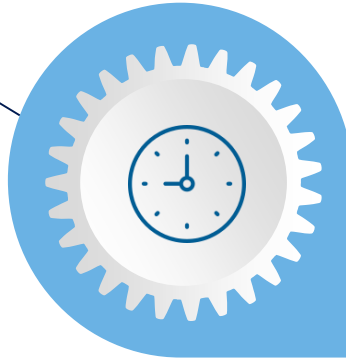
Source : I/ Plateforme d'échanges Île-de-France Mobilités

II/ Service Île-de-France Mobilités

Applications d'accès aux APIs Temps réel

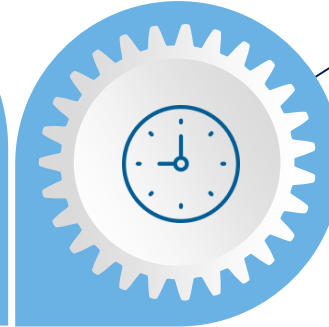
Prochains Passages : requête unitaire
(plateforme Île-de-France Mobilités)

Les prochains passages en temps réel
en requête unitaire



Prochains Passages
(Service Île-de-France Mobilités)

Les prochains passages en temps réel
en requête unitaire
accessible via l'application



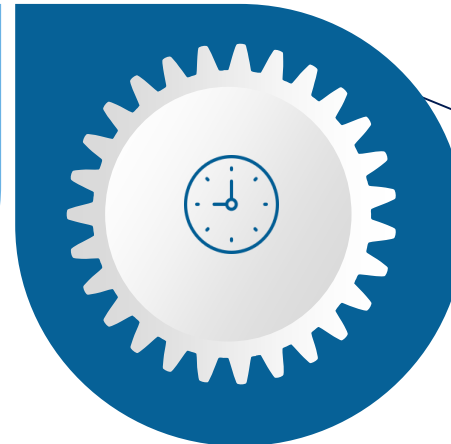
Messages Info Trafic
(plateforme Île-de-France Mobilités)

Accédez aux messages de perturbations en temps réel



Prochains Passages : requête globale
(plateforme Île-de-France Mobilités)

Les prochains passages en temps réel
En requête globale



Prise en main des APIs

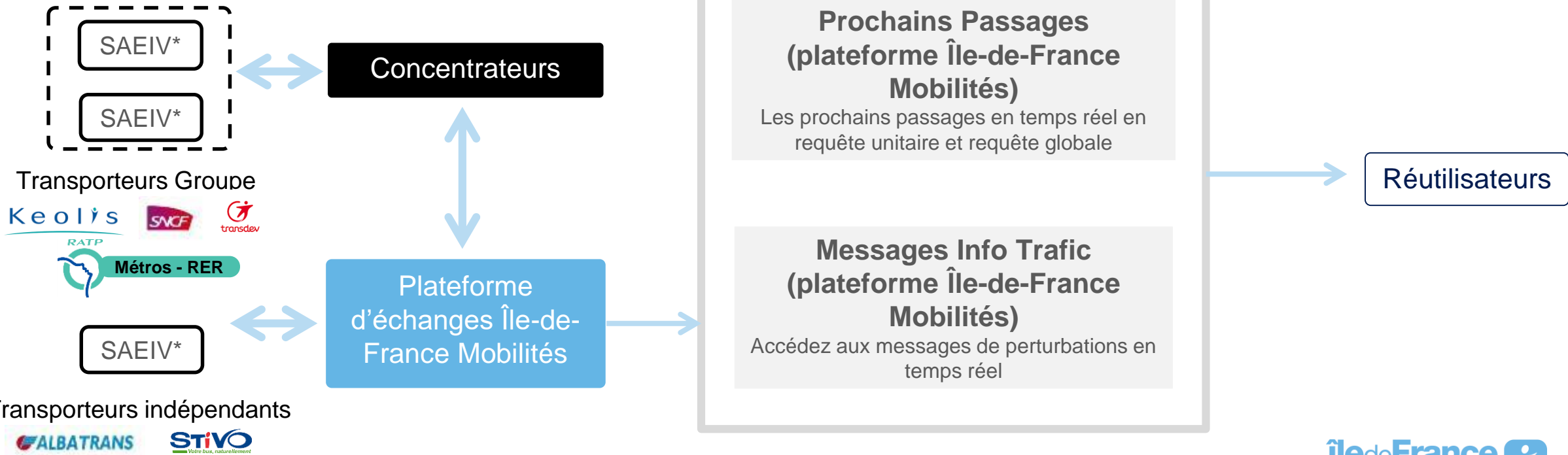
Applications d'accès aux APIs Temps réel

Source : Plateforme d'échanges Île-de-France Mobilités

Schéma de récupération des données

Plateforme d'échanges Île-de-France Mobilités

- 1 **récolte** les informations ;
- 2 les **diffuse** aux autres transporteurs ;
- 3 les **met à disposition** de tout le monde.



*SAEIV = Système d'Aide à l'Exploitation et à l'Information Voyageur

Informations sur les données diffusées

Le portail des normes pour les données d'offre du transport collectif : <http://www.normes-donnees-tc.org/>

Ces services offrent des données sur :

- Prochains passages pour toutes les lignes disponibles sur le réseau
- Prochains passages à un arrêt particulier
- Infos trafic / perturbations sur le réseau

Les données disponibles sur les APIs Prochains passages :

Pour connaître le périmètre (arrêts / lignes) des données disponibles via la plateforme d'échanges d'Île-de-France Mobilités suivez [ce lien](#).

La liste des données disponibles est mise à jour toutes les semaines.

Le format des horaires suit : La norme internationale ISO 8601

Le profil des données : est le SIRI Lite

Identification des objets



Identification des lignes

L'identifiant d'une ligne (**LineRef**) doit être passé sous la forme :

STIF:Line::CXXXXX: avec **CXXXXX** l'identifiant de la ligne dans le [Référentiel Île-de-France Mobilités](#)

Exemples :

- Pour la ligne B du RER, l'identifiant de la ligne dans le Référentiel Île-de-France Mobilités est **C01743**, le pattern est donc « **STIF:Line::C01743:** »
- Pour la ligne de bus Phébus A, l'identifiant de la ligne dans le Référentiel Île-de-France Mobilité est **C00692**, le pattern est donc « **STIF:Line::C00692:** »

Identification des objets

Identification des arrêts

L'identifiant d'un arrêt (**MonitoringRef_ZDE**) doit être passé sous la forme :

STIF:StopPoint:Q:XXXXX:
avec **XXXXX** l'identifiant du [référentiel des arrêts](#)

Exemples

- Pour l'arrêt « Gare de Massy-Palaiseau » sur la ligne B du RER, l'identifiant du référentiel est **412833**, le pattern est « **STIF:StopPoint:Q:412833:** »

Focus sur les APIs « Prochains passages » : requête globale et unitaire

AimedArrival/DepartureTime

Horaires théoriques de départ et d'arrivée établis la veille par le transporteur en prenant en compte la disponibilité des conducteurs et des véhicules. Ces horaires ne sont pas toujours disponibles.

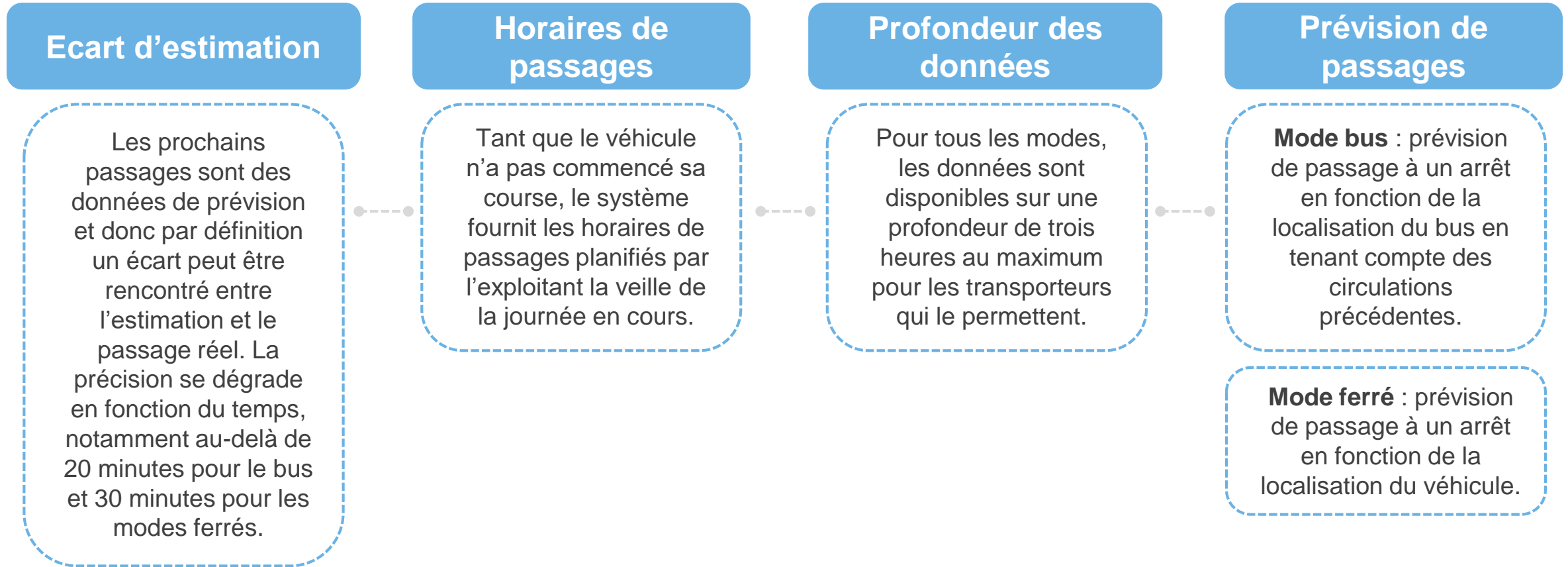
ExpectedArrival/DepartureTime

Prédictions d'horaires de prochains passages prenant en compte la position réelle du véhicule, le temps restant pour atteindre un arrêt et les temps de parcours observés sur les circulations précédentes.

DirectionRef

Il n'y a pas de référentiel partagé à l'échelle d'Île-de-France Mobilités, le champ est donc facultatif. Cependant le champ DestinationRef (terminus de la course) est systématiquement renseigné

Focus sur les APIs « Prochains passages » : requête globale et unitaire



Focus sur l'API « Messages Info Trafic »

Médias

Pour la SNCF et le réseau de Bus OPTILE, les APIs temps réel ne disposent pas des informations qui sont fournies sur les médias (site internet, Twitter, ...).

Écrans des gares

RATP : ensemble des infos trafics disponibles en gares RER et stations de métro et sur les médias RATP.

SNCF : informations disponibles sur les écrans disposés dans les gares.

Bus : informations disponibles sur les écrans disposés aux arrêts de bus.



Pour requêter l'API vous pouvez soit indiquer le champ StopPointRef soit LineRef **mais pas les deux en même temps.**

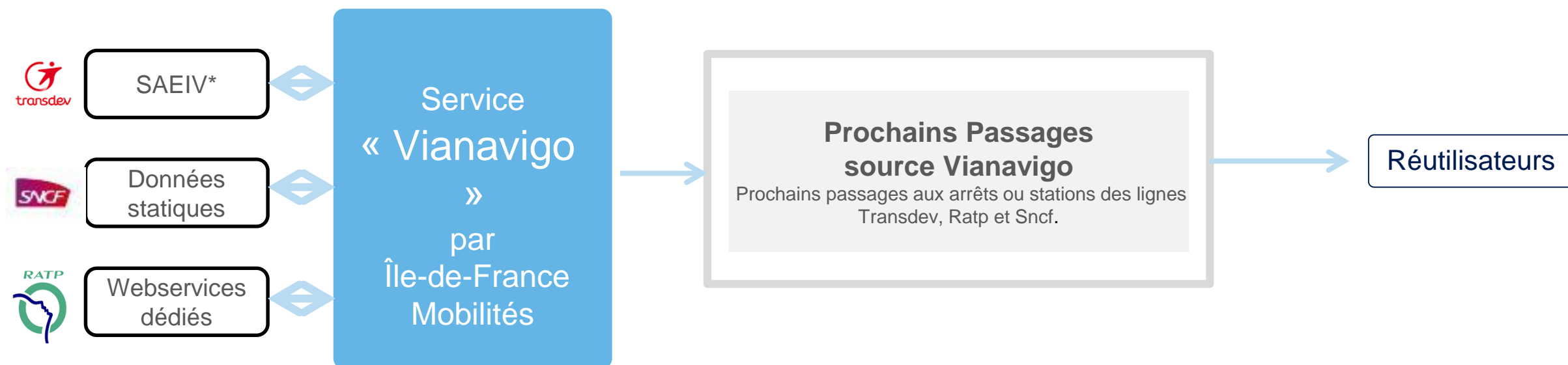
Prise en main des APIs

Applications d'accès aux APIs Temps réel

Source : Service ViaNavigo

Schéma de récupération des données

Le canal ViaNavigo est utilisé en complément, notamment sur le bus RATP et les Tramways, le périmètre des données n'étant pas encore totalement complet via la plateforme d'échanges d'Île-de-France Mobilités.



*SAEIV = Système d'Aide à l'Exploitation et à l'Information Voyageur

Informations sur les données diffusées

Ce services offre des données sur :

Les prochains passages à un arrêt.

Les données disponibles sur ces APIs :

Île-de-France Mobilités permet d'obtenir les temps d'attente des prochains passages à l'ensemble des arrêts (métro, ferré, bus...) des transporteurs suivants :

- **SNCF Transilien**, **RATP** (bus, tram, métro, RER) et **Transdev** (réseaux TRA, CSO, Apollo 7, Montesson, Rambouillet, Vulaines, Lieu saint et Vaux le Pénil).

Décommissionnement de cette API

A terme, les données disponibles via Île-de-France Mobilités vont disparaître. Toutes les données temps réel seront disponibles via la plateforme d'échanges d'Île-de-France Mobilités.

Les utilisateurs seront informés au moins un mois avant toute bascule nécessaire.

Le format des horaires :

Les horaires de passage à l'arrêt sont fournis en temps d'attente exprimé en minutes.

Identification des objets



Identification des lignes

L'identifiant d'une ligne (**Line_id**) doit être passé sous la forme :

line:IDFM:CXXXXX avec **CXXXXX** l'identifiant de la ligne dans le [Référentiel Île-de-France Mobilités](#)

Exemples :

- Pour la ligne C du RER, l'identifiant de la ligne dans le Référentiel Île-de-France Mobilités est **C01727**, le pattern est donc « **line:IDFM:C01727** »
- Pour la ligne 12 du métro, l'identifiant de la ligne dans le Référentiel Île-de-France Mobilités est **C01382**, le pattern est donc « **line:IDFM:C01382** »
- Pour la ligne de bus Grand Melun F, l'identifiant de la ligne dans le Référentiel Île-de-France Mobilité est **C00882**, le pattern est donc « **line:IDFM:C00882** »



Pour plus de détails sur les identifiants GTFS, voir la [documentation GTFS](#).

Identification des objets

Identification des arrêts

L'identifiant d'un arrêt (stop_point_id) doit être passé sous la forme :

stop_point:IDFM:XXXXXX avec **XXXXXX** l'identifiant de l'arrêt dans le [Référentiel des arrêts Île-de-France Mobilités](#) ou sur la base du [GTFS](#)

Exemples :

- Pour l'arrêt Pigalle du métro 12, l'identifiant de la ligne dans le Référentiel Île-de-France Mobilités est **22040**, le pattern est donc **stop_point:IDFM:22040**»



Pour plus de détails sur les identifiants GTFS, voir la [documentation GTFS](#).

Support & Documentation

Support & Documentation

L'équipe Support PRIM vous accompagne dans l'utilisation de ces services à travers les différentes documentations fournies, ainsi que le support par mail.

**Vous avez une question ?
Une suggestion d'amélioration ?
N'hésitez pas à nous contacter via notre
mail de support : [contact@prim.iledefrance-
mobilites.fr](mailto:contact@prim.iledefrance-mobilites.fr)**



Présentation de la plateforme Prim (PDF)