



Accès dynamique aux données via le portail des API d'Ile-De-France Mobilités

-
Présentation des ressources
disponibles

V1.0 du 24/06/2019

A stylized blue figure on the left side of the slide, consisting of a circle for the head and a curved shape for the body and arm, pointing towards the right.

» Sommaire

- 1. Préambule**
- 2. Présentation des services disponibles**
- 3. Comment accéder aux APIs ?**
- 4. Les API pour la recherche d'itinéraire**
- 5. Les API temps réel**

01.

Préambule

-



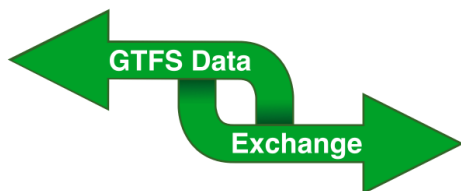
Dans le cadre de ses missions d'information, **Ile-de-France Mobilités** collecte auprès des **75 opérateurs** de transport d'Ile-de-France (RATP, SNCF et entreprises OPTILE), les données décrivant l'offre prévue de transport collectif (**lignes, parcours, arrêts, horaires, calendrier**, etc.).



A l'échelle de la Région, ces données sont considérables : c'est plus de **1 500** lignes de transport et plus de **35 000 arrêts**.

Ces informations alimentent les services d'information d'Ile-de-France Mobilités (**Vianavigo**), ceux des transporteurs, et la plateforme Open Data.

Les dispositifs de diffusion de l'offre de transport francilienne



Avec des fichiers GTFS

- Données brutes sur tous les horaires prévus OPTILE, RATP, SNCF
- Mis à jour toutes les semaines
- Données des opérateurs mise en cohérence par Ile-de-France Mobilités
- Données en licence ODBL

Sur le site Open Data
d'Ile-de-France Mobilités



En accès dynamique

- Services enrichis (itinéraires, etc.) s'appuyant sur les horaires prévus OPTILE, RATP, SNCF mis à jour toutes les semaines et données temps réel (requête unitaire ou requête globale)
- Données sous licence ODBL
- Utilisation des APIs soumise à une inscription préalable au portail et à l'acceptation des **CGU**
- Utilisation des APIs soumise à des quotas et un barème de redevance

[Sur le Portail des API](#)

02.

Présentation des services disponibles

-



Les différentes API...



L'exploration du réseau

Trouvez toutes les informations à propos du réseau de transport francilien (structure du réseau, lignes et arrêts,...)



Les horaires de passages théoriques

Accédez aux horaires des lignes de l'ensemble du réseau (horaires théoriques)



La recherche d'itinéraires

Obtenez des propositions de trajets en transports en commun sur le territoire francilien



La recherche de lieux

Recherchez des lieux grâce à l'auto-complétion et la recherche à proximité



L'info trafic en temps réel

Accédez aux messages de perturbations en temps réel



Le calcul d'isochrones

Représentez l'accessibilité d'un lieu en transports en commun depuis toute l'Île-de-France



Les prochains passages en temps réel

Les prochains passages en temps réel en requête unitaire ou requête globale



Qu'est-ce qu'une API ?

API est un acronyme pour Applications Programming Interface. Une API est une interface de programmation qui permet de se « brancher » sur une application pour échanger des données. (*définition : definitions-marketing.com*)

03.

Comment accéder aux API ?

-



Accès aux API : généralités

Principes

L'accès aux API Ile-de-France Mobilités est **gratuit** dans la limite des quotas fixés dans le barème de redevance. **Un mail vous est envoyé lorsque vous atteignez 80% de votre quota.**

L'accès aux API est **sécurisé** par une authentification préalable (protocole **OAuth2**).

L'utilisation des API est soumise à l'acceptation **des CGU** et de **la licence** sur les données

Authentification

Vous devez avoir un compte utilisateur pour accéder aux API. La création du compte se fait directement depuis le site via l'onglet « Connexion ».

Une fois identifié, et après avoir accepté les Conditions Générales d'Utilisation des API, vous aurez automatiquement **trois applications** créées.

Chacune des trois applications permet de gérer l'accès aux API selon leur nature (temps réel unitaire, temps réel globale et API du calculateur d'itinéraires).

Pas encore inscrit ?

Que vous soyez un particulier, un étudiant, une association, une administration, une start-up ou une grande entreprise, que vous consommiez en dessous des quotas ou bien de façon intensive nos API, inscrivez-vous sur le portail !

En créant votre compte, bénéficiez d'un accès à l'ensemble des API, d'un suivi précis de votre consommation, de documentations techniques et fonctionnelles complémentaires et d'un espace de discussions avec la communauté.

S'inscrire

Accès aux API : les applications (1/2)

Application 1 : Unitary-Real-Time_email@nomdedomaine.com

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


Application 2 : Global-Real-Time_email@nomdedomaine.com

- 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.

Application 3 : Journey-Planner_email@nomdedomaine.com


- Regroupe l'accès aux API Ile-de-France Mobilités (hors API Temps Réel), dont celles du calculateur d'itinéraires (Navitia), dans la limite d'un quota de 20 000 requêtes / jour.

Les noms et descriptifs d'applications peuvent être modifiés et personnalisés depuis le portail.



01-Unitary-Real-Time_██████████@gmail.com

Application regroupant l'accès aux API Temps Réel (par requêtes unitaires) dans la limite d'un quota de 1 000 000 requêtes / jour. Application guaranteeing access to real-time API (unitarian request only). The quota limit is 1 000 000 request/day. ...

 [Voir les indicateurs](#)

Accès aux API : les applications (2/2)

Préalable : Avant toute première utilisation, il est nécessaire de **lier les API à vos applications**. Pour cela, vous devez :

- aller dans l'onglet « Application » et sélectionner celle dont vous souhaitez utiliser les API,
- cliquer sur « Editer Application »,
- cocher les API à utiliser via l'application
- cliquer sur « **Appliquer les modifications** »

Choix des API

Filter les API



Nom API	Description	Sélection
Calcul d'isochrones	Représentation de l'accessibilité d'un lieu en transports en commun depuis/vers toute l'Île-de-France. / L'API s'appuie sur les données d'offres des lignes de trains, RER, métros, tramways, bus et cars franciliens. / navitia.io is ...	<input checked="" type="checkbox"/>
Calculateur Vianavigo - Accès générique (Navitia)	navitia.io is the open API for building cool stuff with mobility data. It provides the following services / * journeys computation / * line schedules / * next departures / * exploration of public transport data / search places / * and ...	<input checked="" type="checkbox"/>
Exploration du réseau	Informations à propos du réseau de transport francilien (structure du réseau, lignes et arrêts,...). / L'API s'appuie sur les données d'offres des lignes de trains, RER, métros, tramways, bus et cars franciliens. / navitia.io is the ope ...	<input checked="" type="checkbox"/>
Horaires de passages théoriques	Accès aux horaires prévus des lignes de l'ensemble du réseau. / L'API s'appuie sur les données d'offres des lignes de trains, RER, métros, tramways, bus et cars franciliens. / NB : les horaires prévus (théoriques) ne sont pas fournies ...	<input checked="" type="checkbox"/>

Appliquer les modifications

Annuler

Quelle est la différence entre une application et une API ?

L'application permet de regrouper l'accès à un groupe d'APIs selon [la grille de nos offres tarifaires](#).

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

Accès aux API : l'Access Token (1/3)

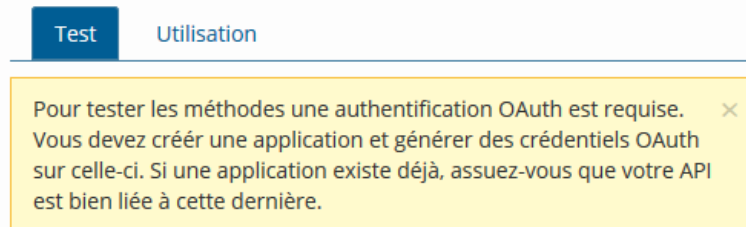
Tester une API depuis le portail :

- choisissez l'API que vous souhaitez tester dans le « Catalogue » (préalablement liée à une application),



The screenshot shows a card for the 'Messages Info Trafic (plateforme Île-de-France Mobil...)'. It features a warning icon (exclamation mark in a triangle) and a description: '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 perturbations sur le trafic (incidents, travaux prévus et non prévus...). Les requêtes et réponses sont exprimées au format SIRI Lite, vous trouverez plus d'informations sur ce format d'échange de données dans la documentation technique.' Below the description are two buttons: 'Tester cette API' (with a gear icon) and 'Voir les données' (with a document icon). At the bottom, it lists 'Type REST', 'Version 1.0.0', and 'Tags Informations - Info trafic'.

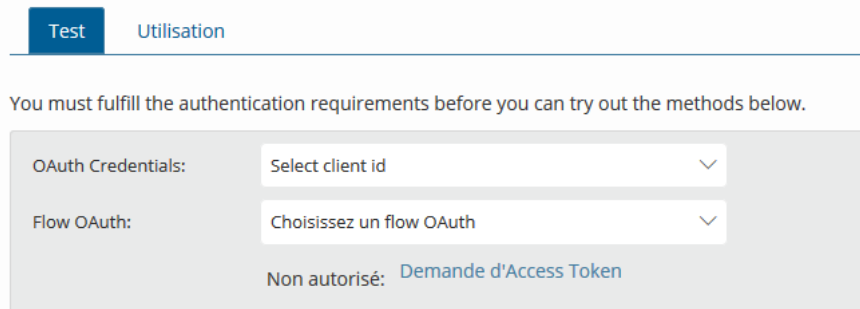
- Si l'API n'est pas liée à une application, le message d'alerte suivant apparaît :



The screenshot shows the 'Test' tab of the API interface. A yellow error box with a close icon (X) contains the text: 'Pour tester les méthodes une authentification OAuth est requise. Vous devez créer une application et générer des crédits OAuth sur celle-ci. Si une application existe déjà, assurez-vous que votre API est bien liée à cette dernière.'



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



The screenshot shows the 'Test' tab of the API interface. It displays the message: 'You must fulfill the authentication requirements before you can try out the methods below.' Below this is a form for OAuth authentication with two dropdown menus: 'OAuth Credentials:' with 'Select client id' and 'Flow OAuth:' with 'Choisissez un flow OAuth'. At the bottom, it says 'Non autorisé: Demande d'Access Token'.



Accès aux API : l'Access Token (2/3)

Authentication :

- Dans le formulaire de demande d'**Access Token** : sélectionner votre application dans « Select Client Id » puis sélectionner « **Client credentials** » comme Flow OAuth. Pour finir, cliquez sur « Demande d'Access Token »

You must fulfill the authentication requirements before you can try out the methods below.

OAuth Credentials:	Select client id
Flow OAuth:	SANDBOX_m... mobilityfr-... 2516... b2007d7044f0505c

You must fulfill the authentication requirements before you can try out the methods below.

OAuth Credentials:	SANDBOX_m... mobilityfr-... 2516... b2007d7044f0505c
Flow OAuth:	Client credentials
	Authorization code
	Client credentials

You must fulfill the authentication requirements before you can try out the methods below.

OAuth Credentials:	SANDBOX_m... mobilityfr-... 2516... b2007d7044f0505c
Flow OAuth:	Client credentials
Non autorisé:	Demande d'Access Token

Accès aux API : l'Access Token (3/3)

Utilisation du Credential :

- Suite à votre demande, il faut valider pour autoriser l'accès,

Autoriser l'accès

Axway API Portal souhaite accéder à la liste des scopes :

☒ read-data

OAuth Credentials:

01-Unitary-Real-Time_cristian.t@gmail.com -
80488e7c-ef0c-480c-b865-9b0d60774307

Flow OAuth:

Client credentials

Autorisation: gOqxMzMOMVa79Qj0eAelltf5WsljEzNsRy7axcDJ7ePDr07H25YqVW7 [Supprimer Access Token](#)

Annuler

Autoriser

- L'API peut ensuite être testé avec le credential créé (sur une durée de 1h)

Test API

Request

Content-Type:

Accept:

application/json

Query parameters

InfoChannelRef:

string

StopPointRef:

string

LineRef:

string

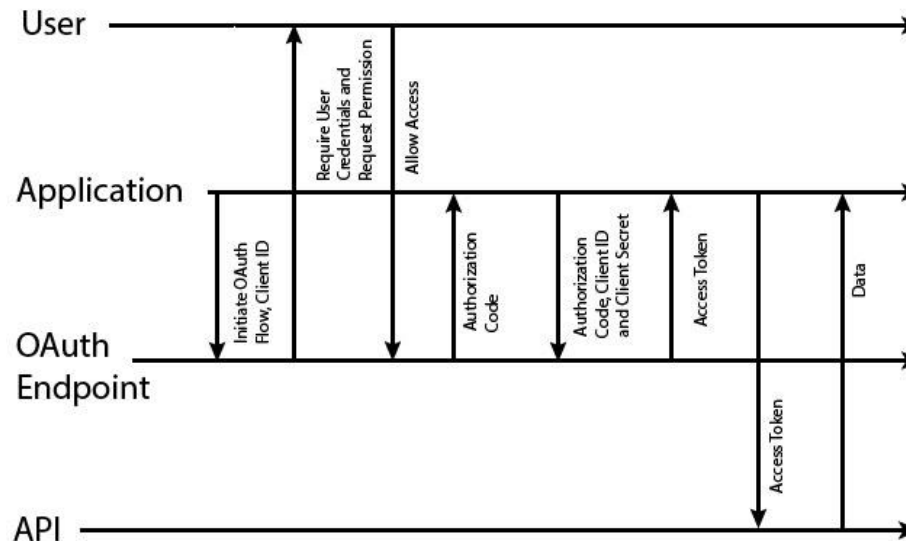
Execute

× Clear

Accès aux API : le protocole OAuth2

Principe : L'accès aux API est sécurisé par la protocole OAuth2.

L'application demande à l'utilisateur de fournir son autorisation par l'intermédiaire d'un formulaire fourni par le serveur de passerelle, qui, en cas d'acceptation, envoie un code d'autorisation à l'application. L'application envoie le code d'autorisation à l'API de fournisseur en échange d'un jeton d'accès (**Access Token**).



Plus d'information sur OAuth2 :

<https://zestedesavoir.com/articles/1616/comprendre-oauth-2-0-par-l'exemple/>

URL de demande de jeton OAUTH : <https://as.api.iledefrance-mobilites.fr/api/oauth/token>

Qu'est-ce que le client ID et le client secret ?

Ces identifiants (ou « credentials OAuth ») permettent de vous faire reconnaître lors de la demande d'un token. **Il y a un client ID et un client secret par application.** Vous pouvez donc les trouver en cliquant sur une application dans la sous-partie « Credentials OAuth ».

API sélectionnées

Nom API	Description	Actions
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 perturbations sur le trafic (incidents, travaux prévus et non prévus...). / Les requêtes et ...	▼ Actions
Prochains passages (plateforme Île-de-France Mobilités)	Horaires estimés des prochains passages aux arrêts. / Actuellement, les horaires de prochains passages à un arrêt en temps réel ne sont disponibles que pour une partie du réseau d'Île-de-France. L'ensemble des arrêts du réseau seront ...	▼ Actions
Prochains passages (source ViaNavigo)	Horaires estimés des prochains passages aux arrêts ou stations des lignes TRANSDEV, RATP et SNCF. / L'API s'appuie sur les données en temps réel fournies par les transporteurs et diffusées sur Vianavigo. / Il s'agit d'un service tempor ...	▼ Actions

▼ Clés d'API

Aucune clé d'API définie. [Éditer application](#) pour lui créer des credentials.

▼ Credentials OAuth

Client ID	Type	Javascript Origins	Redirect URLs	Créé	Actions
	Confidentiel	*		01 Jul 2019, 10:05 GMT	Voir secret

Quels sont les différents codes de réponse possible et quelle est leur signification ?

A la fin de chaque requête, vous recevez un code HTTP ayant une signification précise :

- 200 : La requête a abouti et vous recevez une réponse
- 403 : vous n'êtes pas autorisé à accéder à l'API. Nous vous conseillons de renouveler votre token et de vérifier que l'API est bien liée à votre application
- 404 : La 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 : L'API est indisponible

04.

Les API pour la recherche d'itinéraire (Navitia)

-



Application d'accès aux API de recherche d'itinéraire

Les API de recherche d'itinéraires s'appuient sur la technologie Navitia qui est utilisée sur le service **Vianavigo** (itinéraires, horaires prévus, information trafic, isochrones, ...) permettant ainsi de proposer le même périmètre fonctionnel depuis le portail d'API.

Elles sont accessibles via l'application **Journey-Planner_email@nomdedomaine.com**

En tout, six API intégrant différentes méthodes sont concernés :

- Calcul d'isochrones
- **Calculateur Vianavigo – Accès générique** (Navitia)
- Exploration du réseau
- Horaires de passages théoriques
- La recherche de lieux
- Recherche d'itinéraires

Documentation technique sur Navitia : <https://doc.navitia.io/>

Qualité des données

- ❑ La mise à jour des données est **hebdomadaire**.
- ❑ Les données couvrent les **trois prochaines semaines**.
- ❑ Les données d'offres correspondent aux **horaires théoriques**.
- ❑ 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 cars OPTILE.

Paramètres d'appels (1/2)

Chaque méthode Navitia permet de passer de nombreux paramètres lors d'un appel.

Voici quelques exemples d'identifiants d'objets :

Données	Définition	Exemple
line	Une ligne (line) est un ensemble d'itinéraires regroupés sous un même nom	line:0:100110001:1 (Métro 1) (RER A)
route	Un itinéraire (« route ») définit un enchaînement structuré d'arrêts. Une ligne simple est généralement composée de 2 itinéraires : un décrivant le sens aller, l'autre décrivant le sens retour	
vehicle_journey	Une course (« vehicle_journey ») est la déclinaison d'un itinéraire à un horaire donné. Une course attribut 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.	

Paramètres d'appels (2/2)

company	Désigne un transporteur	company:0:800 (SNCF)
stop_area	Désigne une zone regroupant plusieurs arrêts ou stop_points	stop_area:0:SA:8768600 (Gare de Lyon) (
network	Désigne un réseau commercial	network:0:442 (RATP) (bus TICE)
stop_point	Désigne un point d'arrêt	stop_point:0:SP:59620 (Chaussée d'Antin-Lafayette, ligne 9 du metro) ()

Comment utiliser l'API générique Navitia ayant comme paramètre {URI} ?

Certaines méthodes de l'API Navitia ont comme paramètre {URI}. Cela signifie que ce périmètre d'appel est intégré dans l'URL de la requête et ne se trouve pas, comme la plupart du temps, à la fin de l'URL de requête.

Il faut donc encoder ce paramètre pour qu'il reste dans l'URL.

Par exemple, pour la méthode

`GET/v1/mri/coverage/fr-idf/{uri}/stop_points`

Un paramètre d'appel pourrait être `stop_areas%2Fstop_area:0:SA :8768600` (Gare de Lyon) pour trouver tous les `stop_points` du `stop_area` de la Gare de Lyon.

Ou encore `lines%2Fline:0:100110001:1` (ligne 1 du métro) pour obtenir tous les `stop_points` de cette ligne.

05.

Les API Temps Réel

-



Application d'accès aux API temps réel

Les API temps réel sont accessibles via deux applications :

- Unitary-Real-Time_email@nomdedomaine.com

Pour les API proposant les horaires de prochains passages et l'info trafic pour un arrêt ou une ligne (espace unitaire) :

- Message info trafic (plateforme Île-de-France Mobilités)
- Prochains passages (plateforme Île-de-France Mobilités)
- Prochains passages (source ViaNavigo)

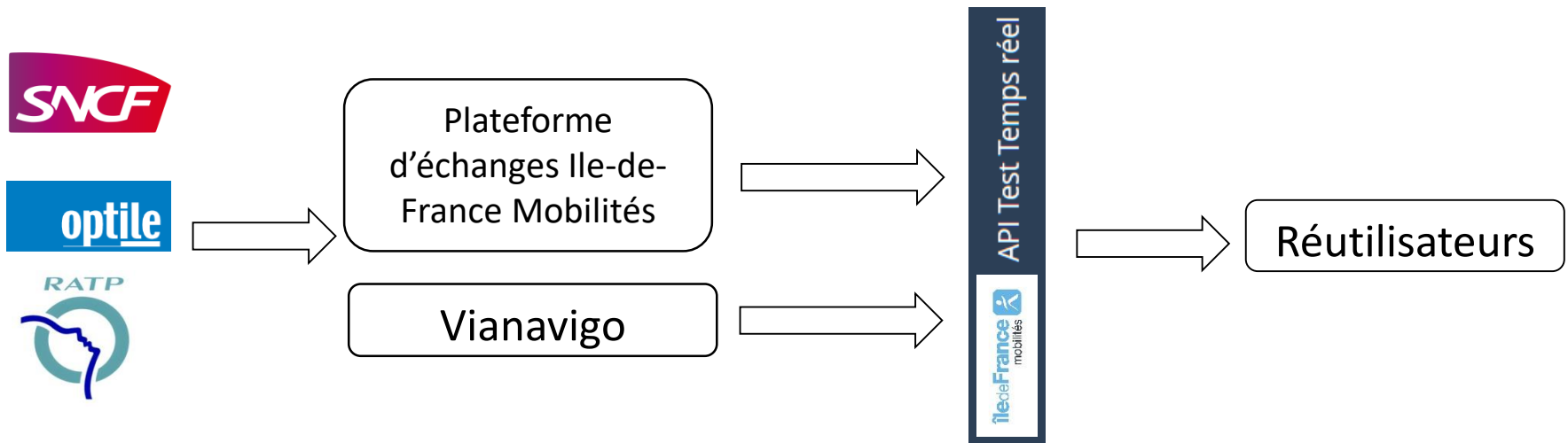
- Global-Real-Time_email@nomdedomaine.com

Pour l'API en requête globale (bulk), « Prochains passages » (requête globale).

D'où proviennent les données?

L'ensemble des données temps réel provient des transporteurs. Ces données sont acheminées via **deux canaux** :

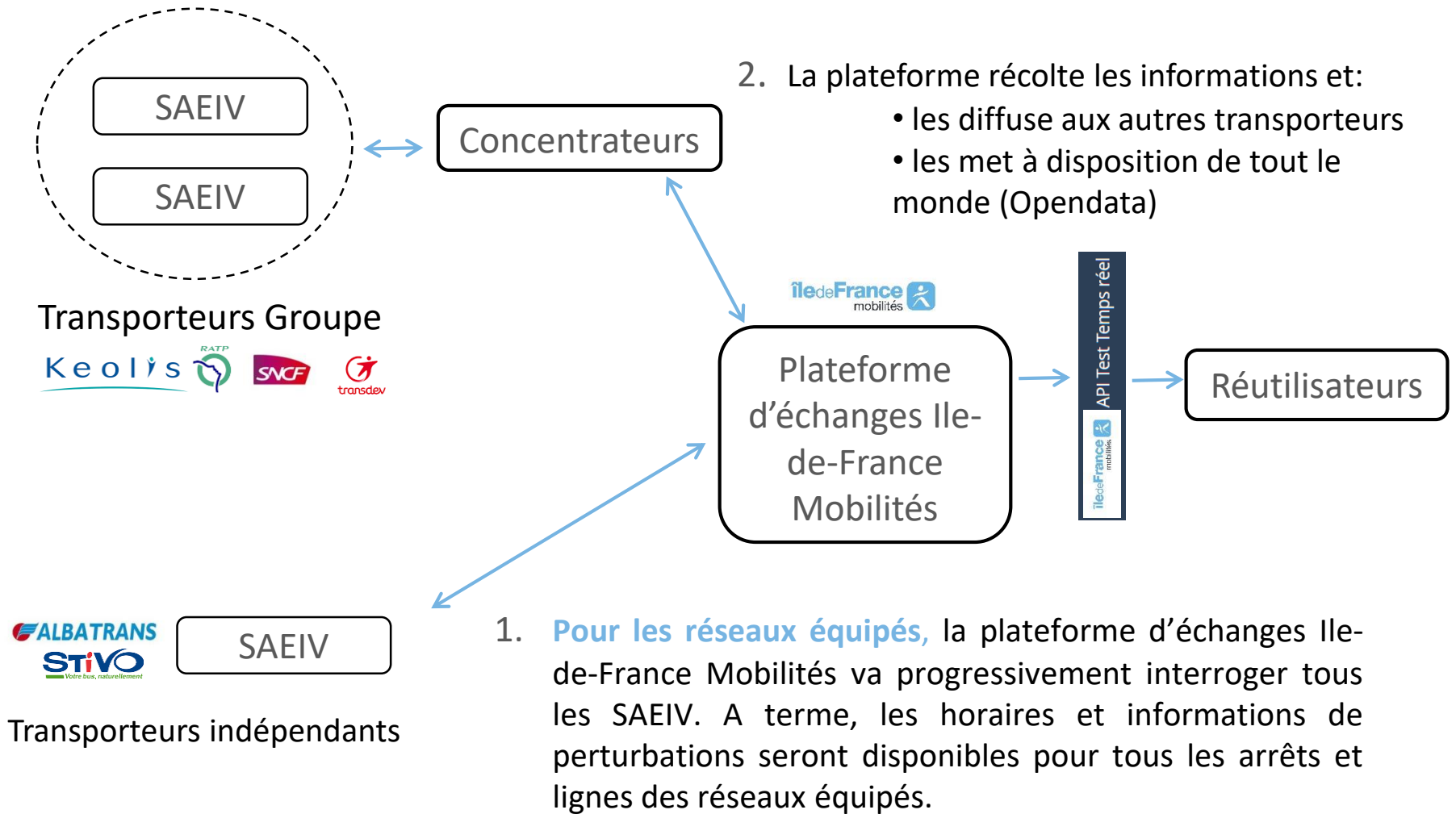
- La plateforme d'échanges Ile-de-France Mobilités
- Le service Vianavigo



Les systèmes d'aide à l'exploitation et à l'information voyageur (SAEIV) fournissent les données en temps réel. Les réseaux de transport des différents opérateurs ne sont pas tous équipés de SAEIV. La donnée sera donc entièrement disponible progressivement.

D'où proviennent les données?

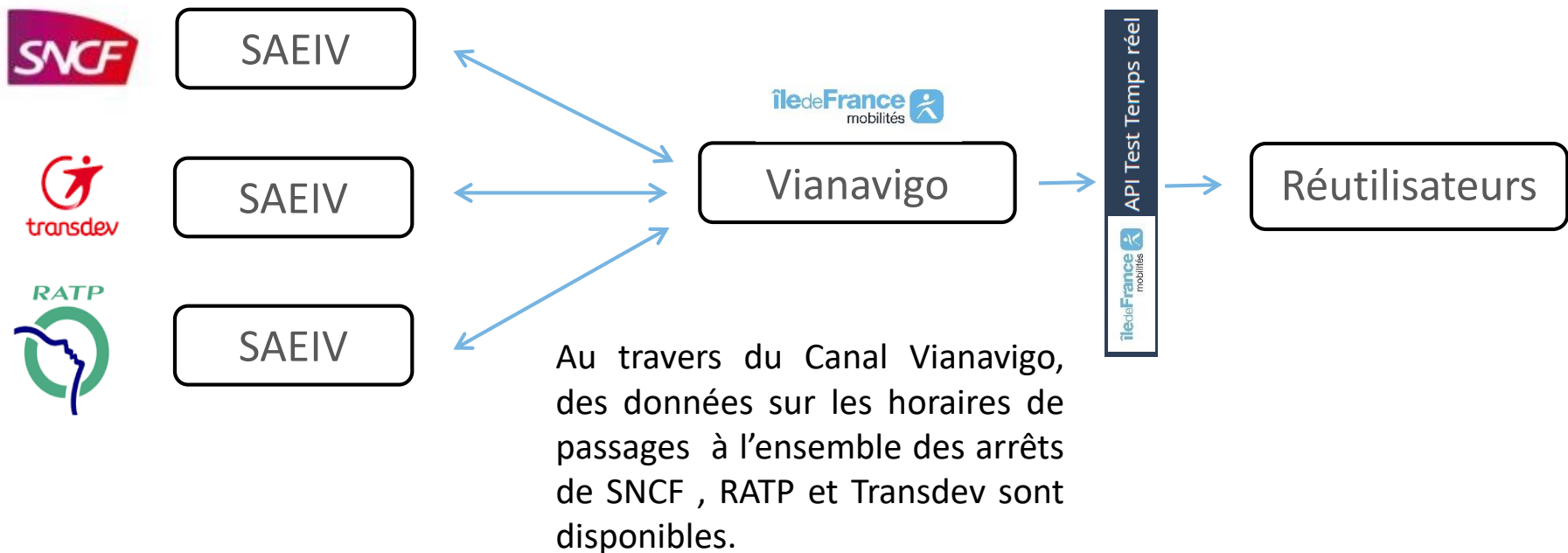
La plateforme d'échanges Ile-de-France Mobilités



D'où proviennent les données?

Le service Vianavigo

Le canal Vianavigo est utilisé en complément, le périmètre des données n'étant pas encore totalement complet via la plateforme d'échanges d'Ile-de-France Mobilités.



Quelles données disponibles?

La Plateforme d'échanges Ile-de-France Mobilités

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

La liste des données disponibles sera mise à jour toutes les 2 semaines.

Le service Vianavigo

Vianavigo 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, Apolo 7, Montesson, Rambouillet, Vulaines, Lieusaint et Vaux le Pénil).

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

Format des données

Selon le canal de distribution des données, ces dernières ne sont pas au même format pour des raisons techniques.

Canaux de données	Services	Format des données	Pérennité
Plateforme d'échange Ile-de-France Mobilités	Prochains passages pour tous les lignes disponibles Prochains passages à un arrêt Info trafic/ perturbation	SIRI Lite	Permanente
Service Vianavigo	Prochains passages à un arrêt	Json « simplifié »	Provisoire

Qualité des données

Prochains horaires : définition

Pour tous les modes, les données sont disponibles sur une profondeur de **deux heures**.

Tant que le véhicule n'a pas commencé sa course, le système fournit les horaires de passages planifiés par l'exploitant la veille de la journée en cours.

Pour les courses en circulation :

- Bus : prévision de passage à un arrêt en fonction de **la localisation du bus** en prenant compte des **circulations précédentes**.
- Mode ferré : prévision de passage à un arrêt en fonction de **la localisation du véhicule**.

Info trafic : périmètre

- 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**

Ce qui signifie que pour la SNCF et le réseau de Bus OPTILE, les API Test Temps Réel ne dispose pas des informations qui sont fournies sur les médias (site internet, twitter, ...).

Les données diffusées

Les horaires

- Les horaires fournis dans les services StopMonitoring, GeneralMessage et EstimatedTimeTable sont représentés selon la norme internationale ISO 8601 et sont exprimés en heure UTC (Temps universel Coordonné).
- En ce qui concerne les données issues du canal Vianavigo, les horaires de passages à l'arrêt sont fournis en temps d'attente exprimés en minutes.

Les perturbations

- Chaque perturbation est localisée sur des lignes ou des arrêts avec des identifiants issus des référentiels.
- Pour les messages de perturbations, trois types de textes peuvent être fournis:
 - Un texte brut (textOnly) qui est obligatoire et qui sera toujours renseigné.
 - Deux autres textes facultatifs qui ne seront pas forcément renseignés :
 - *Un message court (shortMessage)*
 - *Un message long qui peut posséder des éléments de mise en forme (longMessage)*

Précautions d'usage

Restrictions sur les données et leur interrogation

- Prévvision d'horaires de prochains passages : la précision est approximative au-delà de 20 minutes pour le bus et 30 minutes pour les modes ferrés ;
- 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 ;
- Il est recommandé d'activer dans les headers de vos requêtes l'élément « Accept-encoding : gzip, deflate » afin de diminuer la bande passante ;
- Compte tenu de sa taille très importante, la réponse à une requête globale sera transmise en mode compressé.

Précautions d'usage

Horaires théoriques et estimés

Nom	Description	Format d'échange
	Horaires théoriques fournis par les transporteurs 3 semaines à l'avance	GTFS
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. (pas toujours disponible)	SIRI
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	SIRI

Champs facultatifs

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

Identification des objets

Les principes selon le canal d'appel

- Selon le canal utilisé les données ne sont pas identifiées de la même manière.
- Les données issues de la plateforme d'échanges Ile-de-France Mobilités dépendent des référentiels des arrêts et des lignes.
- Celles issues du canal Vianavigo ont les mêmes identifiants que ceux du GTFS (*ou API Ile-de-France Mobilités*).

Pour les arrêts, la relation (de type 1..n) entre identifiants du référentiel et GTFS se trouve dans l'un des fichier du GTFS : stop_extensions.txt.

Pour les lignes, la relation entre identifiants du référentiel et GTFS se trouve dans le jeu de données « Référentiel des lignes de transport en commun d'Ile-de-France », il s'agit respectivement des champs ID_Line et ExternalCode_Line.

Identification des objets

Identifications des courses pour la plateforme Île-de-France Mobilités

- L'identifiant de la course DatedVehicleJourneyRef identifie de manière unique une course pour tous les transporteurs, la course est donc reconstituée dans une réponse à une requête globale.
- **Attention** : La RATP ne fournit pas d'identifiants de courses mais un compteur technique sans lien métier avec la notion de course. Nous obtenons alors dans la réponse à la requête globale pour une même ligne et un même sens tous les véhicules s'arrêtant à la même heure quel que soit l'arrêt. Par conséquent, dans les réponses aux requêtes globales, **les courses RATP ne sont pas correctement reconstituées**. Les prochains passages aux arrêts sont cependant tous renseignés dans la réponse.

Identification des objets

Identifications des arrêts

- Les identifiants des arrêts présents dans le GTFS sont des identifiants propres à chaque transporteur. En conséquence, un arrêt de bus unique (même poteau ou même abri) par exemple partagé par 2 transporteurs aura 2 identifiants. Par ailleurs, ces identifiants ne sont pas pérennes dans le temps. Entre 2 générations du fichier GTFS, les identifiants d'un même arrêt pour un même transporteur peuvent changer.
- Le référentiel des arrêts Reflex au niveau ZDE identifie de manière unique et pérenne un arrêt, même s'il est partagé par plusieurs transporteurs.
- Il y a donc une relation 1..n entre identifiants Reflex ZDE et identifiants GTFS. Le fichier stop_extension.txt du GTFS permet de faire la relation. Il est mis à jour à chaque génération du fichier GTFS.

Construction de l'identifiant de la ligne

Données issues de la plateforme d'échanges Ile-de-France Mobilités

Identifiant de la ligne est « *LineRef* » : **STIF:Line::CXXXXX:**

Avec XXXXX le code commercial de la ligne.

Exemples :

- Pour la ligne **B du RER**, l'identifiant commercial de la ligne est **C01743**, le pattern est donc « STIF:Line::**C01743**: »
- Pour la ligne de **bus Phébus A**, l'identifiant commercial de la ligne est **C00692**, le pattern est donc « STIF:Line::**C00692**: »

Construction de l'identifiant de la ligne

Données issues du service Vianavigo

L'identifiant de la ligne (paramètre « *Line_id* ») est le **même que celui du GTFS** (*route_id*).

Exemples :

- Pour la ligne **C du RER**, l'identifiant GTFS (*route_id*) de la ligne est le « **810:C** »
- Pour la ligne **12 du métro**, l'identifiant GTFS (*route_id*) de la ligne est le « **100110012:12** »
- Pour la ligne de **bus Transdev Mëlibus C**, l'identifiant GTFS (*route_id*) de la ligne est le « **066066022:C** »

Pour plus de détails sur les identifiants GTFS voir la documentation GTFS.

*Il est aussi possible de passer en paramètre l'identifiant utilisé par les API Ile-de-France Mobilités (*external_code*).*

Construction de l'identifiant du point d'arrêt

Données issues de la plateforme Ile-de-France Mobilités

Identifiants des arrêts physiques de bus : **STIF:StopPoint:Q:XXXXXX:**

Avec XXXXXX, l'identifiant de la zone d'embarquement (**ZDE**)

Identifiants des arrêts commerciaux de bus : **STIF:StopArea:SP:XXXXXX:**

Avec XXXXXX, l'identifiant de la zone de Lieu (**ZDL**)

Exemples « **Gare de Massy-Palaiseau** »

- Pour la granularité **ZDE** et l'arrêt sur la **ligne B du RER**, l'identifiant du référentiel est **412833**, le pattern est « **STIF:StopPoint:Q:412833:** »
- Pour la granularité **ZDL**, l'identifiant du référentiel est commun pour les **lignes B et C** et est **58774**, le pattern est « **STIF:StopArea:SP:58774:** »
- Pour la granularité **LDA** qui comprend les gares routières environnantes, l'identifiant est **63244**, le pattern est « **STIF:StopArea:SP:63244:** »

Construction de l'identifiant du point d'arrêt

Données issues du service Vianavigo

L'identifiant d'un point d'un arrêt (paramètre « *stop_point_id* ») est le **même que celui du GTFS** (*stop_id*).

Exemples :

- Pour l'arrêt Gare des **Invalides** de la ligne **RER C**, l'identifiant de l'arrêt physique est le « **stopPoint:8739303:800:C** » **même que celui du GTFS**
- des **Invalides**
- Pour l'arrêt Gare des **Invalides** de la ligne **RER C**, l'identifiant de l'arrêt physique est le « **stopPoint:8739303:800:C** »
- Pour l'arrêt **Gare de Melun** de la ligne de **bus Transdev Mèlibus C**, l'identifiant de l'arrêt physique est le « **stopPoint:27:135** »

Pour plus de détails voir la documentation GTFS.

*Il est aussi possible de passer en paramètre l'identifiant utilisé par les API Ile-de-France Mobilités (*external_code*).*

Correspondances entre les identifiants des APIs de recherche d'itinéraire et les APIs temps réel

Nom	Définition	Appellation Canal Vianavigo	Appellation Canal Plateforme d'échanges Ile-de-France Mobilités	Exemple pour les APIs temps réel plateforme IDFM	Exemple pour les APIs temps réel source ViaNavigo	Exemple pour les APIs de recherche d'itinéraire
Ligne	Une ligne est un ensemble d'itinéraires regroupés sous un même nom.	Line	Line	STIF:Line::C01371:	100110001 (Métro 1)	line:0:100110001:1 (Métro 1)
Itinéraire	Un itinéraire définit un enchaînement structuré d'arrêts. Une ligne simple est généralement composée de 2 itinéraires : un décrivant le sens aller, l'autre décrivant le sens retour.	route	route	NP	NP	NP

Correspondances entre les identifiants des APIs de recherche d'itinéraire et les APIs temps réel

Nom	Définition	Appellation Canal Vianavigo	Appellation Canal Plateforme d'échanges Ile-de-France Mobilités	Exemple pour les APIs temps réel	Exemple pour les APIs temps réel source ViaNavigo	Exemple pour les APIs de recherche d'itinéraire
Course	Une course est la déclinaison d'un itinéraire à un horaire donné. Une course attribut 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.	vehicle_journey	Monitored VehicleJourney	NP	NP	NP
Zone d'embarquement (ZDE)	Endroit où le voyageur attend, monte ou descend du véhicule	Stop_point (Arrêt physique)	stop-monitoring	STIF:StopPoint:Q:2388: (Chaussée d'Antin-Lafayette, sur la ligne 7 du métro)	Stop_point:0:SP:59568 (Chaussée d'Antin La Fayette, ligne 7 du métro)	Stop_point:0:SP:59568 (Chaussée d'Antin La Fayette, ligne 7 du métro)

Correspondances entre les identifiants des APIs de recherche d'itinéraire et les APIs temps réel

Appellation du référentiel	Définition	Appellation Canal Vianavigo	Appellation Canal Plateforme d'échanges Ile-de-France Mobilités	Exemple pour les APIs temps réel	Exemple pour les APIs temps réel source ViaNavigo	Exemple pour les APIs de recherche d'itinéraire
Zone de Lieu (ZDL)	Regroupement au sein d'un lieu d'arrêt, de zones d'embarquement portant le même nom commercial	Stop_area				NP
Lieu d'Arrêt (LDA)	Regroupement d'arrêts physiques de plusieurs transporteurs					

06.

**Exemple de code et
explication pour requêter
les API**

Comment faire une requête ? *Code / Explication*

ligne par ligne

A - Obtenir le token – A Faire une fois toutes les heures.

```
1 import requests
2 urlOAuth = 'https://as.api.iledefrance-
  mobilites.fr/api/oauth/token'
3 client_id='<your client_id>'
4 client_secret='<your client_secret>'
5
6 data =dict(
7     grant_type='client_credentials',
8     scope='read-data',
9     client_id=client_id,
10    client_secret=client_secret
11 )
12
13 response = requests.post(urlOAuth, data=data)
14 print(response.json)
15
16 if response.status_code != 200:
17     print('Status:', response.status_code, 'Erreur
18     sur la requête; fin de programme')
19     exit()
20
21 jsonData = response.json()
22 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)

> **Comment connaître mon client_id et mon client_secret ?**

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* (réponse – 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 utiliser dans la variable *token* grâce à la réponse *jsonData*

Comment faire une requête ? *Code expliqué*

B – Requête une API

```
22 url = 'https://traffic.api.iledefrance-
    mobilites.fr/v1/tr-messages-it'
23
24 params =dict(
25     LineRef='ALL'
26 )
27
28 headers = {
29     'Accept-Encoding' : 'gzip',
30     'Authorization' : 'Bearer ' + token
31 }
32
33 response = requests.get(url, params=params,
    headers=headers)
34
35 if response.status_code != 200:
36     print('Status:', response.status_code, 'Erreur
    sur la requête; fin de programme')
37     exit()
38
39 jsonData = response.json()
```

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

> *Comment trouvez 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.

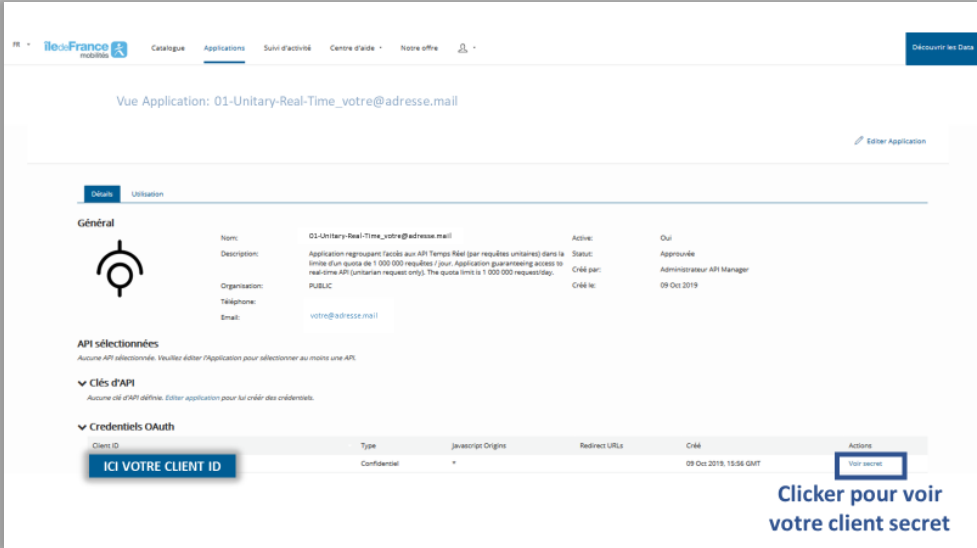
Ligne 33 – La fonction requests.get prend trois arguments en entrée l'url de l'API et les paramètres de la requête et les headers de celle-ci.

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)

Comment connaître mon client id et mon client secret ?

> Client id



Vue Application: 01-Unitary-Real-Time_votre@adresse.mail

[Éditer Application](#)

Général

Nom: 01-Unitary-Real-Time_votre@adresse.mail

Description: Application regroupant l'accès aux API Temps Réel (par requêtes unitaires) dans la limite d'un quota de 1 000 000 requêtes / jour. Application guaranteeing access to real-time API (unitary request only). The quota limit is 1 000 000 request/day.

Organisation: PUBLIC

Téléphone:

Email: votre@adresse.mail

Active: Oui

Status: Approuvée

Créé par: Administrateur API Manager

Créé le: 09 Oct 2019

API sélectionnées

Aucune API sélectionnée. Veuillez éditer l'application pour sélectionner au moins une API.

Clés d'API

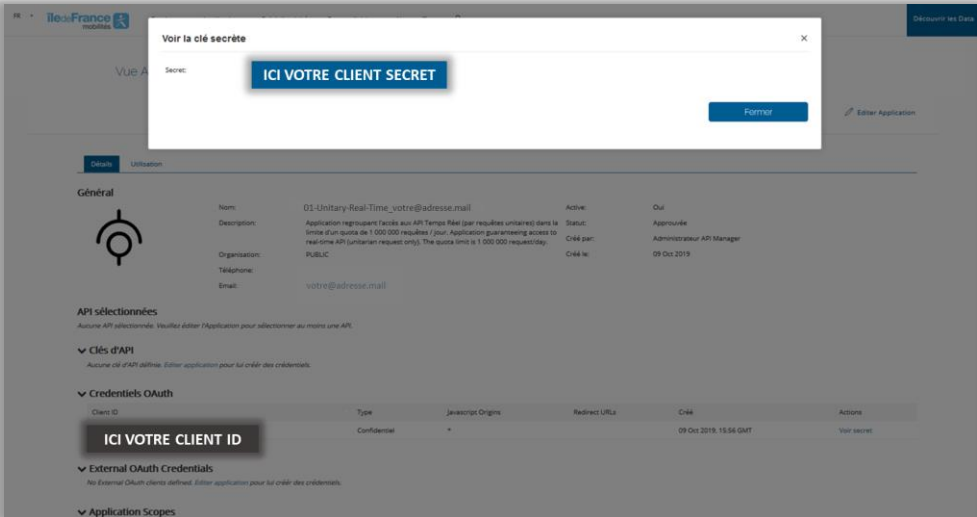
Aucune clé d'API définie. Éditer l'application pour lui créer des identifiants.

Credentials OAuth

Client ID	Type	JavaScript Origins	Redirect URIs	Créé	Actions
ICI VOTRE CLIENT ID	Confidential	*		09 Oct 2019, 15:56 GMT	Voir secret

Cliquez pour voir votre client secret

> Client secret



Vue Application: 01-Unitary-Real-Time_votre@adresse.mail

[Éditer Application](#)

Général

Nom: 01-Unitary-Real-Time_votre@adresse.mail

Description: Application regroupant l'accès aux API Temps Réel (par requêtes unitaires) dans la limite d'un quota de 1 000 000 requêtes / jour. Application guaranteeing access to real-time API (unitary request only). The quota limit is 1 000 000 request/day.

Organisation: PUBLIC

Téléphone:

Email: votre@adresse.mail

Active: Oui

Status: Approuvée

Créé par: Administrateur API Manager

Créé le: 09 Oct 2019

API sélectionnées

Aucune API sélectionnée. Veuillez éditer l'application pour sélectionner au moins une API.

Clés d'API

Aucune clé d'API définie. Éditer l'application pour lui créer des identifiants.

Credentials OAuth

Client ID	Type	JavaScript Origins	Redirect URIs	Créé	Actions
ICI VOTRE CLIENT ID	Confidential	*		09 Oct 2019, 15:56 GMT	Voir secret

External OAuth Credentials

No External OAuth clients defined. Éditer l'application pour lui créer des identifiants.

Application Scopes

Modal: Voir la clé secrète

Secret: ICI VOTRE CLIENT SECRET

[Formuler](#)

Comment trouver l'url de l'API ?

The screenshot shows the Navitia API documentation page for Ile de France Mobilités. The page includes a navigation bar with links like 'Catalogue', 'Applications', 'Suivi d'activité', 'Centre d'aide', and 'Notre offre'. A 'Découvrir les Data' button is in the top right. The main content area features a description of the API, a list of services (journeys computation, line schedules, next departures, etc.), and a gear icon labeled 'API'. Below this, there are fields for 'Version: 1.0.0', 'Host: https://traffic.api.iledefrance-mobilites.fr', 'Base path: /v1/mri', 'CORS: Enabled', and 'Type: REST'. A 'Swagger 2.0' button is also present. A 'Test' tab is active, showing authentication requirements (OAuth Credentials, Flow OAuth) and a table of API endpoints.

Method	Path	Nickname	Description
GET	/v1/mri/coverage/fr-idf/places	get_coverage_region_places	
GET	/v1/mri/coverage/fr-idf/vehicle_journeys	get_coverage_region_vehicle_journeys	
GET	/v1/mri/coverage/fr-idf/{uri}/journey_pattern_points/{id}	get_coverage_region_uri_journey_pattern_points_id	

L'url pour requêter une API est la concaténation du champs *Host* suivi du *Path* de la méthode désirée.

Ainsi, pour le *Calculateur Vianavigo – Accès générique (Navitia)* et la méthode *get_coverage__region__places* l'url est :

<https://traffic.api.iledefrance-mobilites.fr/v1/mri/coverage/fr-idf/places>

Comment connaitre les différents paramètres possibles ?

FR * **Île de France** mobilités Catalogue Notre offre Connexion Découvrir les Data

Télécharger la définition: [Swagger 2.0](#)

Test

Method	Path	Nickname	Description
GET	/v1/tr-messages-tz/general-message	getMessages	Point d'entrée pour récupérer les informations de type "Message"

Implementation Notes

Point d'entrée pour récupérer les informations de type "Message"

Description

Point d'entrée pour récupérer les informations de type "Message"

Query parameters

Field	Description
LineRef string	Identifiant Codiligne de la ligne. Ex: STIF:Line:C01727: (RER C)
StopPointRef string	Référence de l'arrêt. Ex: STIF:StopPoint:Q412844: (Gare de Saint-Rémy-Lès-Chevreuse)
InfoChannelRef string	Information OU Perturbation OU Commercial

Header parameters

Field	Description
-------	-------------

Les différents paramètres possibles d'une méthode sont définis dans Query parameters.

Dans l'exemple suivant, la méthode *getMessages* de l'application *Messages Info Trafic* (plateforme *Île-de-France Mobilités*) 1.0.0 les différents paramètres sont :

- *LineRef*
- *StopPointRef*
- *InfoChannelRef*