

OPEN DATA & MICRO SERVICES

DONNÉES TER & INTERCITÉS

MANUEL D'INTÉGRATION
OCTOBRE 2012



1 - INTRODUCTION

Ce service fournit les informations concernant les lignes, gares, et horaires sur l'ensemble du réseau de transport SNCF TER et Intercités.

Les informations fournies sont celles présentées dans différents sites Internet, dont ter-sncf.com, en particulier dans les résultats de calculs d'itinéraires, prochains départs et prochaines arrivées. Ces informations sont dites « théoriques » et ne comportent pas de notion de « temps réel ».

Les notions manipulées dans les pages suivantes sont celles utilisées dans le transport public et couvertes par la norme Transmodel.

1.1 - Conventions

- ▶▶ Le nœud parent de chaque flux est une liste (Liste de StopArea, de Network, etc.)
- ▶▶ Le premier nœud enfant « params » est spécifique :
 - ▶ Il reprend l'ensemble des paramètres d'appel de l'application afin de faciliter l'intégration et la recherche d'anomalie.
 - Pour des raisons de maniabilité, tous les nœuds contenant ces paramètres sont en minuscules
- ▶▶ La première lettre de chaque mot est une majuscule.
 - ▶ En cas de mot composé, agréger les deux mots (ex : StopAreaList).
 - ▶ L'index (identifiant unique non pérenne utilisé par le Webservice), l'identifiant (issu des bases de données initiales) et le nom de chaque type de données apparaissent en attribut.

1.2 - Intégration et pérennité des données

Le Webservice retourne dans chaque flux XML 3 identifiants différents pour chaque notion :

- ▶▶ **ExternalCode** (ex : StopAreaExternalCode) : Défini par la SNCF, cet identifiant est unique et pérenne, quelles que soient les modifications apportées dans la base de données.
- ▶▶ **Idx** (ex : StopAreaIdx) : correspond à l'identifiant unique du Webservice pour la base de donnée en cours. **Cet identifiant n'est pas pérenne** dès lors que l'on modifie la base de données (soit chaque semaine environ).
- ▶▶ **ID** (ex : StopAreaID) : correspond à l'identifiant issue de la base de données source. Donnée à titre indicatif, il n'est pas forcément unique et permet simplement de tracer une donnée entre la base source et le rendu final

Il est donc conseillé d'utiliser l'identifiant pérenne (ExternalCode) chaque fois que c'est possible.

2 - DEFINITIONS ET USAGE

2.1 - Définitions des types de données

Notion	Traduction	Commentaire
Idx	Index	Numéro unique de donnée : ce numéro n'est pas pérenne entre 2 publications.
ID	Identifiant initial	Identifiant issu de la base de donnée d'origine. Ce numéro n'est pas forcément unique (dépendant du format de la base initial, de la fusion entre plusieurs bases...),
ExternalCode	Code externe	Code unique et <i>pérenne</i> d'une donnée. L'administrateur de donnée a pour fonction la mise en place d'un code unique sur ses données principales : ligne (ex : code STIF en Ile de France), arrêts commerciaux (ex : code UIC des gares).
Name	Nom commercial	Intitulé commercial utilisé pour communiquer avec le client
Comment	Commentaire	Zones de remarques importantes sur une donnée. Il peut exister des commentaires sur les arrêts physiques ou sur les horaires.
Country	Pays	
City	Commune	
Alias	Alias	Intitulé commercial qui référence une autre notion (arrêt commercial ou commune). Par exemple « Châlons sur Marne » est un alias de « Châlons en Champagne » : la commune existait auparavant sous ce nom que certaines personnes continuent à utiliser.
StopArea	Zone d'arrêt ou arrêt commercial	Une zone d'arrêt, regroupement de plusieurs points d'arrêts physiques sous une même appellation.

Notion	Traduction	Commentaire
Hang	Accroche avec un arrêt physique	<p>Représente le lien entre un arrêt commercial et les points d'entrée du réseau de transport : la liste des arrêts physiques situés à proximité de coordonnées géographiques par exemple.</p> <p>« StoppointIdx » représente l'identifiant du point d'arrêt</p> <p>« Duration » indique le temps de correspondance pour rejoindre le point d'arrêt</p> <p>« ConnectionKind » indique le mode d'approche à utiliser pour rejoindre le point d'arrêt (détaillé ci-dessous).</p>
StopPoint	Point d'arrêt ou arrêt physique	Point d'arrêt du réseau de transport. Exemple : poteau de bus.
Network	Marque commerciale, réseau de transport	Exemple : réseau Paris Saint-Lazare, réseau RER E
Company	Transporteur	SNCF
Mode	Mode commercial de transport	Exemples : train, tramway, bus...
modeType	Mode générique de transport (normalisé TRIDENT)	Exemples : bus, train, rapidTransit...
Line	Ligne commerciale	<p>Regroupe les propriétés principales d'une ligne quelle que soit sa complexité (ligne avec terminus différents, lignes en boucles...)</p> <p>La définition d'une ligne commerciale dans TRANSMODEL est exprimée par le fait qu'elle représente un groupe de ligne (d'itinéraires) qui sont en général connus du public par une appellation commune (nom ou numéro)</p>

Notion	Traduction	Commentaire
Route	Parcours ou sous ligne	La définition d'un parcours à ce niveau est la plus petite entité composée d'une Origine/terminus et des arrêts entre ces deux points (cela peut correspondre à une mission pour la SNCF IdF par exemple) Une ligne commerciale est composée d'au moins 2 parcours : le sens aller et le sens retour
ValidityPattern	Régime de fonctionnement ou condition de service	Représente la liste exhaustive des dates pour lesquels la circulation est assurée. Cette représentation est de la forme suivante : à partir du <i>date de début de régime</i> 111110011111001111100
VehicleJourney	Desserte ou circulation	Suite d'horaires théorique sur un parcours de ligne
VehicleKind	Type d'équipements possibles sur les véhicules	Exemple : Bus à plancher bas, climatisation, transport de bicyclette possible...
Vehicle	Equipements disponibles sur le véhicule	Ensemble des équipements disponibles sur le véhicule. INFORMATION NON DISPONIBLE A CE JOUR DANS LES DONNEES ET RENSEIGNER A « FALSE » PAR DEFAULT
Service	Un service horaire du réseau	Décrit l'ensemble des horaires d'un réseau pour une période donnée exemple : service été des horaires SNCF
Stop	Horaire	Horaire de passage à un arrêt physique. Contient en particulier l'heure d'arrivée, l'heure de départ et l'arrêt physique.
Connection	Correspondance	Possibilité de joindre deux arrêts physiques afin de changer de véhicule lors d'un itinéraire en transport en commun.

Notion	Traduction	Commentaire
ConnectionKind	Type de correspondance	<p>3 types :</p> <p>Intra arrêt physique : on descend du véhicule et on attend un autre véhicule sans bouger. Le Webservice retourne une section (voir la notion « section » ci-dessous) de type StopPointConnection</p> <p>Intra arrêt commercial / inter arrêt physique : on descend du véhicule et on doit marcher afin de joindre un autre arrêt physique proche appartenant au même arrêt commercial. Le Webservice retourne 2 sections : LinkConnection suivie de StopPointConnection</p> <p>Inter arrêt commercial : on descend du véhicule et on doit marcher afin de joindre un autre arrêt commercial avant de se rendre à l'arrêt physique où prendre un nouveau véhicule. Le Webservice retourne 2 sections : LinkConnection suivie de StopPointConnection</p>
Route Point	Point sur itinéraire	Point d'arrêt sur un parcours. Permet de dessiner les plans de ligne par exemple.

► La valeur de ConnectionKind (mode d'approche utilisé pour rejoindre le point d'arrêt) est à récupérer depuis la liste des StopArea. A titre d'information cette valeur correspond à :

- 0 : AddressConnection □ Jonction adresse / arrêt commercial
- 1 : SiteConnection □ Jonction Lieu public / arrêt commercial
- 2 : StopAreaConnection □ Correspondance intra arrêt commercial, entre 2 arrêts physiques distincts
- 3 : StopPointConnection □ Correspondance intra-arrêt physique
- 4 : VehicleJourneyConnection □ Liaison en transport en commun
- 5 : ProlongationConnection □ Liaison en transport en commun / 1prolongement de service
- 6 : LinkConnection □ Liaison marche à pied
- 7 : WalkConnection □ Trajet a pied uniquement

- ▶ 8 : PersonnelCarConnection ☐ Liaison en transport personnel (voiture)
- ▶ 9 : Undefined ☐ La liaison est spécifiée par l'utilisateur (x, y, commune, zone d'arrêt)
- ▶ 10 : BicycleConnection ☐ Liaison en vélo
- ▶ 11 : CabConnection ☐ Liaison en taxi
- ▶ 12 : ODTConnection ☐ Liaison en transport en commun à la demande.
- ▶ 13 : DefaultConnection ☐ Le type de liaison s'adapte au type de données
- ▶ 14 : End

2.2 - Pagination des flux de réponses

Pour éviter des temps de réponses trop long, les flux de réponse ont une taille maximale de 4Mo. Afin de permettre l'exploitation de réponses plus volumineuse, un mécanisme de pagination a été mis en place.

Pour chaque flux concerné, une balise « PagerInfo » a été rajouté en fin de réponse. Cette balise a le format suivant :

- ▶▶ L'attribut « ResponseCount » contient le nombre d'éléments renvoyés effectivement.
- ▶▶ L'attribut « TotalCount » contient le nombre d'éléments total.
- ▶▶ L'attribut « PagerIndex » indique l'indice à partir duquel on a commené l'affichage. Par défaut, celui-ci vaut 0.
- ▶▶ Pour avoir la valeur «PagerIndex» pour le flux suivant, la formule est : $\text{ResponseCount} + \text{ResponseStartIndex} + 1$.

Il devient donc possible, lors de l'intégration, de récupérer l'intégralité d'une réponse, même si celle-ci est très volumineuse, en faisant un n appels successifs de l'API, en variant uniquement le paramètre «PagerIndex ».

2.3 - Liste d'objets

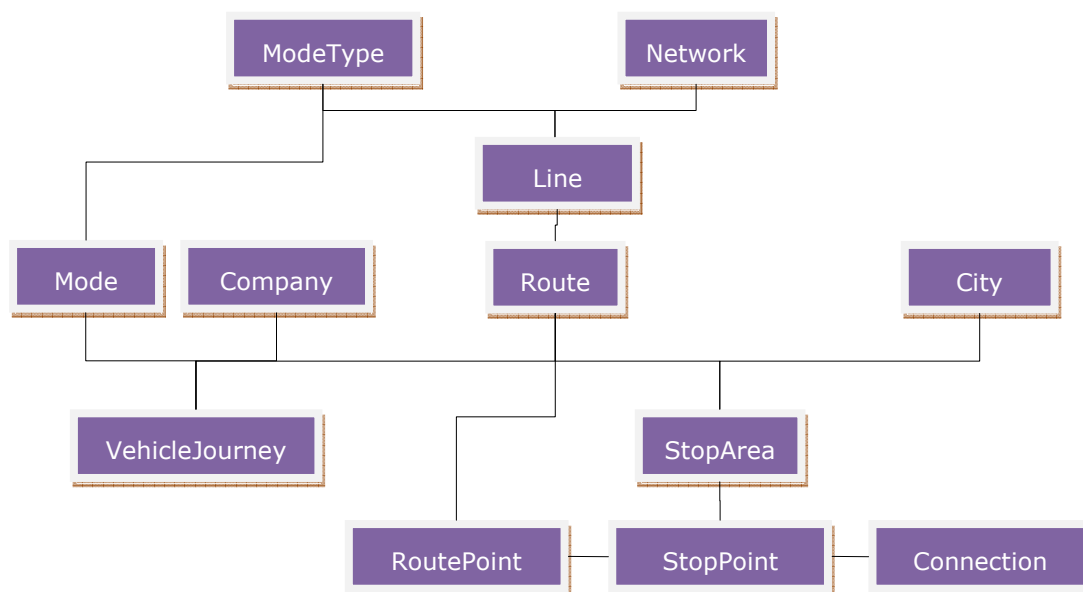
2.3.1 - Appel

« /?Action= »

2.3.2 - Description

Ce flux est une sur-couche abstraite permettant d'accéder via un seul appel paramétrable à toutes les listes d'objets liés au Transport Public.

Les paramètres sont organisés selon l'arborescence suivante :



Tout appel aux informations relatives à l'un des nœuds de cette arborescence peut être fait avec n'importe quel autre nœud de l'arborescence comme paramètre. Par exemple, on peut obtenir la liste des Route par Network et City.

- ▶▶ Chacun des nœuds de cette arborescence est à la fois un type qui peut être retourné par l'API, et un paramètre possible pour un appel
- ▶▶ Chaque paramètre peut être passé
 - ▶ avec une cardinalité unique, ou sous forme d'une liste d'éléments séparés par des points virgules
 - ▶ en inclusion ou exclusion pour chaque élément au sein d'un type d'élément. Soit par exemple :
 - « paramètre de type StopArea, valeurs SA1 **ET** SA2 »
 - « paramètre de type StopArea, valeurs SA1 **OU** SA2 »
 - Voir le chapitre suivant pour les détails.
 - ▶ Entre chaque type de paramètre, il est possible de spécifier si l'on souhaite que leur prise en compte soit inclusive ou exclusive (on peut, par exemple, obtenir la listes des routes contenues dans n Network **ET** n City ou dans n Network **OU** n City).

2.3.3 - Appel et passage de paramètres

2.4 - Type de données à retourner

L'API utilise un paramètre invariant, « Action », qui indique de quel nœud de la liste sont issus les éléments qui vont être affichés (exemple : Action=RouteList pour afficher une liste de routes).

Action peut prendre les valeurs suivantes :

- ▶ NetworkList
- ▶ ModeTypeList
- ▶ ModeList
- ▶ LineList
- ▶ RouteList
- ▶ VehicleJourneyList
- ▶ StopAreaList
- ▶ StopPointList
- ▶ RoutePointList
- ▶ CompanyList

Ces paramètres sont identifiés par code externe, avec, par défaut, la nomenclature suivante « nomNoeud »+ExternalCode (exemple : ModeTypeExternalCode, StopAreaExternalCode, ...).

Pour éviter des temps de réponses trop long, les flux de réponse ont une taille maximale de 4Mo.

2.4.1.1 Filtres

Les filtres peuvent être passés sous forme de liste, liées au type de filtre, composé de 0 à n éléments, séparés par des points virgules et contenant le type d'exclusivité.

Par exemple, pour obtenir une liste de ligne passant par l'arrêt SA1 **ET** l'arrêt SA2 :

- ▶ Action=LineList&StopareaExternalCode=SA1;SA2|and

Par exemple, pour obtenir une liste de ligne passant par l'arrêt SA1 **OU** l'arrêt SA2 :

- ▶ Action=LineList&StopareaExternalCode=SA1;SA2|or

Pour un filtre avec plusieurs éléments séparés par des points virgules et |or à la fin, le résultat contient l'union de toutes les valeurs. Au contraire avec |and à la fin, le résultat est de l'intersection de toutes les valeurs.

Note : La présence du filtre d'action avec une valeur (unique) exclu tous les autres filtres présents dans l'URL.

2.4.2 - Filtre complémentaires

Selon l'action choisie, on dispose de filtres additionnels :

- ▶ Line (Action=LineList) : on peut également filtrer par Destination (DestinationExternalCode),...

- ▶ Route (Action=RouteList) : CheckOrder, Sens
- ▶ StopArea : MainStopArea (valeur booléenne)
- ▶ ...

2.4.3 - Tableau des paramètres de l'API

Nom	Type	Format	Description	Obligatoire
Action	String		Définit le type de données à renvoyer	Oui
NetworkExternalCode	String, ou liste de string	Séparé par ; Liste finie par " and" ou " or" non obligatoire (" or" par défaut)	Renvoie les données appartenant au(x) réseau(x) passé(s) en paramètre	Non
ModeTypeExternalCode	String, ou liste de string	idem	Renvoie les données appartenant au(x) type(s) de mode passé(s) en paramètre	Non
ModeExternalCode	String, ou liste de string	idem	Renvoie les données appartenant au(x) mode(s) passé(s) en paramètre	Non
LineExternalCode	String, ou liste de string	idem	Renvoie les données appartenant au(x) ligne(s) passée(s) en paramètre	Non
RouteExternalCode	String, ou liste de string	idem	Renvoie les données appartenant au(x) route(s) passée(s) en paramètre	Non
VehicleJourneyExternalCode	String, ou liste de string	idem	Renvoie les données appartenant au(x) circulation(s) passée(s) en paramètre	Non
StopAreaExternalCode	String, ou liste de string	idem	Renvoie les données appartenant au(x) zone(s) d'arrêt passé(s) en paramètre	Non
StopPointExternalCode	String, ou liste de string	idem	Renvoie les données appartenant au(x) point(s) d'arrêt passé(s) en paramètre	Non

Nom	Type	Format	Description	Obligatoire
RoutePointExternalCode	String, ou liste de string	idem	Renvoie les données appartenant au(x) point(s) de parcours passé(s) en paramètre	Non
CityExternalCode	String, ou liste de string	idem	Renvoie les données appartenant au(x) ville(s) passée(s) en paramètre	Non
CompanyExternalCode	String	idem	Renvoie les données appartenant au(x) transporteur(s) passé(s) en paramètre	Non
LineCode	String		Filtre par code commercial de ligne	Non
Filtres complémentaires : Liés aux zones d'arrêt - StopArea				
MainStopArea	Boolean		Action=StopAreaList Ne retourne que les StopArea de type MainStopArea	Non
MultiModal	Boolean		Action=StopAreaList Ne retourne que les StopArea de type Multimodal	Non
CarPark	Boolean		Action=StopAreaList Ne retourne que les StopArea ayant un Parking	Non
Filtres complémentaires : Liés aux lignes commerciales - Line				
DestinationExternalCode	String		Action=LineList Filtre par destination, si un paramètre StopAreaExternalCode est déjà présent. Permet d'obtenir la liste des lignes passant entre 2 arrêts.	Non

Nom	Type	Format	Description	Obligatoire
Filtres complémentaires : Liés aux parcours - Route				
Sens	1 : aller -1 : retour		Action=RouteList Combiné avec un filtre sur Line (LineExternalCode), permet d'obtenir la liste des parcours d'une ligne pour un sens donné	Non
CheckOrder	Boolean		Action=RouteList Actif seulement si plus d'une zone d'arrêt est passée en paramètre. Si valorisé à 1, ne renvoie que les routes passant par les zones d'arrêts passées en paramètre DANS L'ORDRE DANS LEQUEL ELLES SONT	Non
Filtres complémentaires : Liés aux circulations - VehicleJourney				
ShowStop	Boolean		Action=VehicleJourneyList N'affiche que les propriétés d'entête des VehicleJourney, sans les détails. Permet de diminuer la taille des flux en n'affichant pas les horaires. Ainsi on peut obtenir plus de VehicleJourney par page.	Non
VehicleJourneyExternalCode	String		Action=VehicleJourneyList Attention à la volumétrie des données si ce code n'est pas renseigné.	Non

Nom	Type	Format	Description	Obligatoire
VehicleJourneyName	String		Action=VehicleJourneyList Attention à la volumétrie des données si ce code n'est pas renseigné.	Non
Date	String	YYYY MM DD	Action=VehicleJourneyList Filtre par date : propose les VehicleJourney qui circulent cette date uniquement. Peut être utilisé conjointement avec le filtre suivant EndDate.	Non
EndDate	String	YYYY MM DD	Action=VehicleJourneyList Filtre par plage de date (le paramètre Date doit être présent) : propose les VehicleJourney qui circulent entre Date et EndDate.	Non
StartTime	String	HH MM	Action=VehicleJourneyList Permet de ne retourner que les circulations qui <u>partent après</u> StartTime (dont le <u>premier horaire</u> est <u>supérieur</u> à StartTime). Rq : Le paramètre EndTime suivant est obligatoire si ce paramètre StartTime est présent.	Non

Nom	Type	Format	Description	Obligatoire
EndTime	String	HH MM	Action=VehicleJourneyList Permet de ne retourner que les circulations qui <u>partent avant</u> EndTime (dont le <u>premier horaire</u> est <u>inférieur</u> à EndTime). Le paramètre StartTime précédent est obligatoire si ce paramètre EndTime est présent.	Non

Filtres complémentaires : Liés aux points d'arrêts sur parcours - RoutePoint

Sens	1 : aller -1 : retour	Action=RoutePointList Combiné avec un filtre sur Line (LineExternalCode), permet d'obtenir la liste des RoutePoint d'une ligne pour un sens donné	Non
------	------------------------------	---	-----

2.4.4 - Exemple d'appel



Recherche de lignes commerciales

- Passant par les arrêts Lille Flandres et Armentières

<http://localhost/?action=LineList&StopAreaExternalCode=OCE87286005;OCE872861&and>



Recherche des trains TER circulant sur la ligne « Marseille-St-Charles - Toulon » et desservant la gare de Marseille-Blancarde

http://localhost/?action=VehicleJourneyList&StopAreaExternalCode=OCE87751081&ModeTypeExternalCode=OCETrain_TER&LineExternalCode=OCETrain_TER-87751008-87755009

2.4.5 - Résultats



NetworkList

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionNetworkList>
  - <Params Function="Navitia_Debug.DEBUG_VKAT/api">
    <action>networklist</action>
  </Params>
  - <NetworkList NetworkCount="3">
    <Network NetworkIdx="0" NetworkId="-1" NetworkName="" NetworkExternalCode="" />
    <Network NetworkIdx="1" NetworkId="0" NetworkName="RATP" NetworkExternalCode="RATP" />
    <Network NetworkIdx="2" NetworkId="1" NetworkName="SNCF" NetworkExternalCode="OCE1" />
  </NetworkList>
</ActionNetworkList>
```



LineList

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionLineList>
  - <Params Function="Navitia_Debug.DEBUG_VKAT/api">
    <action>linelist</action>
    <lineidx>10</lineidx>
  </Params>
  - <LineList LineCount="1">
    - <Line LineIdx="10" LineId="2687" LineName="Loudun - Chinon" LineCode="Car TER" LineExternalCode="OCECAR-87575084-87571687" LineAdditionalData=""
      SortOrder="10" LineColor="" HasAdaptedRoute="0">
        <Mode ModeIdx="2" ModeId="20" ModeName="Car TER" ModeExternalCode="OCECar TER" />
        <Network NetworkIdx="2" NetworkId="1" NetworkName="SNCF" NetworkExternalCode="OCE1" />
        - <Forward ForwardName="loudun gare sncf - chinon gare sncf">
          - <Direction>
            - <StopArea StopAreaIdx="0" StopAreaId="-1" StopAreaName="" StopAreaExternalCode="" MainStopArea="0" MultiModal="0" CarPark="0" MainConnection="0"
              AdditionalData="">
              - <City CityIdx="0" CityId="-1" CityName="" CityExternalCode="" CityCode="">
                <Country CountryIdx="0" CountryId="-1" CountryName="" CountryExternalCode="" />
              </City>
              <Coord />
            </StopArea>
            </Direction>
          </Forward>
          - <Backward BackwardName="chinon gare sncf - loudun gare sncf">
            - <Direction>
              - <StopArea StopAreaIdx="0" StopAreaId="-1" StopAreaName="" StopAreaExternalCode="" MainStopArea="0" MultiModal="0" CarPark="0" MainConnection="0"
                AdditionalData="">
                - <City CityIdx="0" CityId="-1" CityName="" CityExternalCode="" CityCode="">
                  <Country CountryIdx="0" CountryId="-1" CountryName="" CountryExternalCode="" />
                </City>
                <Coord />
              </StopArea>
              </Direction>
            </Backward>
          </Line>
        </LineList>
      </ActionLineList>
```

RouteList

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionRouteList>
- <Params Function="Navitia_Debug.DEBUG_VKAT/api">
  <action>routelist</action>
  <routeidx>10</routeidx>
</Params>
- <RouteList RouteCount="1">
- <Route RouteIdx="10" RouteId="5472" RouteName="Line 5 (Bobigny - Pablo Picasso)" RouteExternalCode="RATRoute5-2" IsForward="0" RouteLineIdx="5"
  IsFrequency="1" IsAdapted="0">
- <Line LineIdx="5" LineId="3228" LineName="Line 5" LineCode="5" LineExternalCode="RATLigne5" LineAdditionalData="" SortOrder="5" LineColor=""
  HasAdaptedRoute="0">
  <Mode ModelIdx="11" ModeId="1" ModeName="Métro" ModeExternalCode="metro" />
  <Network NetworkIdx="1" NetworkId="0" NetworkName="RATP" NetworkExternalCode="RATP" />
  <Forward ForwardName="" />
  <Backward BackwardName="" />
</Line>
</Route>
</RouteList>
</ActionRouteList>
```

VehicleJourneyList

Remarques :

- ▶ Le volume de données en sortie peut être conséquent.
- ▶ Afin de limiter ce volume, **le nombre de VehicleJourney remonté est limité à 1000**. L'utilisation de filtre est donc obligatoire pour naviguer parmi l'intégralité des VehicleJourney.

StopAreaList

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionStopAreaList>
- <Params Function="Navitia_Debug.DEBUG_VKAT/API">
  <Action>StopAreaList</Action>
  <CityIdx>122</CityIdx>
</Params>
- <StopAreaList StopAreaCount="6">
- <StopArea StopAreaIdx="445" StopAreaId="143" StopAreaName="Besneraie" StopAreaExternalCode="143" MainStopArea="0" MultiModal="0" CarPark="0"
  MainConnection="0" AdditionalData="">
- <City CityIdx="122" CityId="15" CityName="La Chapelle Thouarault" CityExternalCode="35065" CityCode="">
  <Country CountryIdx="1" CountryId="0" CountryName="France" CountryExternalCode="FRA" />
</City>
- <Coord>
  <CoordX>288376,50</CoordX>
  <CoordY>2355386,00</CoordY>
</Coord>
- <HangList>
  <Hang StopPointIdx="890" Duration="0" ConnectionKind="13" />
  <Hang StopPointIdx="889" Duration="0" ConnectionKind="13" />
</HangList>
- <ModeList ModeCount="1">
  <Mode ModelIdx="1" ModeId="1" ModeName="bus" ModeExternalCode="bus" />
</ModeList>
</StopArea>
+ <StopArea StopAreaIdx="446" StopAreaId="144" StopAreaName="Chertrrie" StopAreaExternalCode="144" MainStopArea="0" MultiModal="0" CarPark="0"
  MainConnection="0" AdditionalData="">
+ <StopArea StopAreaIdx="447" StopAreaId="146" StopAreaName="Eglise" StopAreaExternalCode="146" MainStopArea="0" MultiModal="0" CarPark="0" MainConnection="0"
  AdditionalData="">
+ <StopArea StopAreaIdx="448" StopAreaId="147" StopAreaName="Grippière" StopAreaExternalCode="147" MainStopArea="0" MultiModal="0" CarPark="0"
  MainConnection="0" AdditionalData="">
+ <StopArea StopAreaIdx="449" StopAreaId="148" StopAreaName="Pont Harel" StopAreaExternalCode="148" MainStopArea="0" MultiModal="0" CarPark="0"
  MainConnection="0" AdditionalData="">
+ <StopArea StopAreaIdx="450" StopAreaId="149" StopAreaName="Rochers" StopAreaExternalCode="149" MainStopArea="0" MultiModal="0" CarPark="0" MainConnection="0"
  AdditionalData="">
</StopAreaList>
</ActionStopAreaList>
```

StopPointList

```

- <ActionStopPointList>
- <Params Function="Navitia_Debug.DEBUG_VKAT/api">
  <action>stoppointlist</action>
</Params>
- <StopPointList StopPointCount="4407">
- <StopPoint StopPointIdx="0" StopPointId="-1" StopPointName="" StopPointExternalCode="" FareZone="0">
  <StopPointAddress StopPointAddressName="" StopPointAddressNumber="" StopPointAddressTypeName="" />
  <Equipment Sheltered="False" MIPAccess="False" Elevator="False" Escalator="False" BikeAccepted="False" BikeDepot="False" />
  <Mode ModeIdx="0" ModeId="-1" ModeName="" ModeExternalCode="" />
  <City CityIdx="0" CityId="-1" CityName="" CityExternalCode="" CityCode="" />
  <Coord />
</StopPoint>
- <StopPoint StopPointIdx="1" StopPointId="1925" StopPointName="1000 Clubs" StopPointExternalCode="TCLclu002" FareZone="0">
  <StopPointAddress StopPointAddressName="" StopPointAddressNumber="" StopPointAddressTypeName="" />
  <Equipment Sheltered="False" MIPAccess="False" Elevator="False" Escalator="False" BikeAccepted="False" BikeDepot="False" />
  <Mode ModeIdx="3" ModeId="4" ModeName="bus suburbain" ModeExternalCode="TCLbus suburbain" />
  <City CityIdx="0" CityId="-1" CityName="" CityExternalCode="" CityCode="" />
  <Coord />
</StopPoint>
- <StopPoint StopPointIdx="2" StopPointId="1976" StopPointName="1000 Clubs" StopPointExternalCode="TCLclu001" FareZone="0">
  <StopPointAddress StopPointAddressName="" StopPointAddressNumber="" StopPointAddressTypeName="" />
  <Equipment Sheltered="False" MIPAccess="False" Elevator="False" Escalator="False" BikeAccepted="False" BikeDepot="False" />
  <Mode ModeIdx="3" ModeId="4" ModeName="bus suburbain" ModeExternalCode="TCLbus suburbain" />
  <City CityIdx="0" CityId="-1" CityName="" CityExternalCode="" CityCode="" />
  <Coord />
</StopPoint>
- <StopPoint StopPointIdx="3" StopPointId="283" StopPointName="11 novembre" StopPointExternalCode="TCL11n002" FareZone="0">
  <StopPointAddress StopPointAddressName="" StopPointAddressNumber="" StopPointAddressTypeName="" />
  <Equipment Sheltered="False" MIPAccess="False" Elevator="False" Escalator="False" BikeAccepted="False" BikeDepot="False" />
  <Mode ModeIdx="3" ModeId="4" ModeName="bus suburbain" ModeExternalCode="TCLbus suburbain" />
  <City CityIdx="0" CityId="-1" CityName="" CityExternalCode="" CityCode="" />
  <Coord>
    <CoordX>654586,00</CoordX>
    <CoordY>2622245,00</CoordY>
  </Coord>
</StopPoint>
</StopPointList>
</ActionStopPointList>

```

2.5 - ProximityList

2.5.1 - Appel

« /?Action=ProximityList »

2.5.2 - Description

Recherche de proximité géographique

2.5.3 - Tableau des paramètres de l'API

Nom	Type	Format	Défaut	Description	Obligatoire
Type	StopPoint / StopArea			Type de données sur lesquelles la recherche de proximité se fera	Oui
X	réel			Longitude en Lambert II étendu	Oui
Y	réel			Latitude en Lambert II étendu	Oui
Distance	integer		1000	Rayon d'accroche (en mètre)	Non

MinCount	integer	0	Nombre d'item minimum à accrocher	Non
NbMax	Integer	100	Nombre d'item maximal à accrocher	Non
CircleFilter	booléen	1	Permet de circonscrire strictement ou non la recherche au cercle d'accroche. Si CircleFilter=faux alors les solutions sont contenues dans un carré de côté 2xDistance	Non
MainStopArea	Booléen	0	Type = StopArea Ne retourne que les StopArea de type MainStopArea	Non
MultiModal	Booléen	0	Type = StopArea Ne retourne que les StopArea de type MultiModal	Non
CarPark	Booléen	0	Type = StopArea Ne retourne que les StopArea ayant un Parking	Non
NetworkExternalCode	String		Filtrer sur le(s) réseau(x) passé(s) en paramètre. Ex : NetworkExternalCode=NW1;NW2 or (Fonctionne comme les filtres décrits précédemment)	Non
ModeExternalCode	String		Filtrer sur le(s) mode(s) passé(s) en paramètre. Ex : ModeExternalCode=M1;M2 or (Fonctionne comme les filtres décrits précédemment)	Non
ModeTypeExternalCode	String		Filtrer sur le(s) modetype(s) passé(s) en paramètre. Ex : ModeTypeExternalCode=MT1;MT2 or (Fonctionne comme les filtres décrits précédemment)	Non

2.6 - NextDeparture

2.6.1 - Appel

« /?Action=NextDeparture »

2.6.2 - Description

Ce flux contient la liste des prochains départs à l'arrêt désiré sur une ligne désiré. Ce flux se base sur les données théoriques sans tenir compte du temps-réel.

2.6.3 - Tableau des paramètres de l'API

Nom	Type	Format	Défaut	Description	Obligatoire
StopAreaExternalCode	String			Arrêt sur lequel porte la requête.	Oui
StopPointExternalCode	String			Arrêt physique sur lequel porte la requête	Oui
DestinationExternalCode	String			Filtre pour les prochains départs à « destination de ». Le filtre est actif pour toutes les gares traversées : ce n'est pas forcément la destination finale du train, mais celle de l'utilisateur	Non
LineExternalCode	String			Ligne sur laquelle porte la requête	Non
Sens	Integer			Filtre du sens de la ligne utilisée	Non
RouteExternalCode	String			Route sur laquelle porte la requête	Non
ModeTypeExternalCode	String			Filtre du type de mode utilisé	Non
ModeExternalCode	String			Filtre du mode dont le type de mode est utilisé	Non
Time	String	Hour Minute	Heure courante	Heure définissant le premier prochain départ	Non
Date	String	Year Month Day	Date courante	Date de la recherche des prochains départs	Non
NbStop	Integer		5	Nombre de départ proposés	Non

Nom	Type	Format	Défaut	Description	Obligatoire
UseAdapted	Boolean		Oui	Indique si le traitement doit utiliser les données adaptées	Non
ManageEvent	Boolean		Oui	Indique si le traitement doit tenir compte des données temps réel	Non
DateChangeTime	String	Hour Minute	00 00	Heure de changement de date (présentation en « calendrier TV »).	Non
UseTransday	Boolean		true	Permet de bloquer l'affichage des prochains départs au-delà de DateChangeTime et d'afficher par exemple « fin de service »	Non

Remarque :

- ▶ Le filtre sur « Line » est prioritaire sur le filtre « Route » si les deux sont renseignés.
- ▶ Les horaires de montée et de descente sont présentés. L'application média peut néanmoins simplifier la présentation en affichant uniquement les horaires de départ.
- ▶ La balise « Day » présente dans le flux résultat « NextDepartureList\Stop\StopTime\Day » indique un changement de jour. Si la valeur vaut 1, l'horaire présenté est sur le lendemain par rapport à la date demandée (passe-minuit)

2.6.4 - Exemple d'appel

- ▶▶ Recherche des 2 prochains trains TER et Intercités au départ de la gare d'Armentières et à destination de Lille Flandres

<http://localhost/?action=nextdeparture&StopAreaExternalCode=OCE87286146&DestinationExternalCode=OCE87286005&nbstop=2>

2.7 - NextArrival

2.7.1 - Appel

« /?Action=NextArrival »

2.7.2 - Description

Ce flux contient la liste des prochaines arrivées à l'arrêt désiré sur une ligne désirée. Ce flux se base sur les données théoriques et ne tient pas compte du temps réel.

2.7.3 - Tableau des paramètres de l'API

Ce flux a la même structure et les mêmes paramètres d'appel que l'API NextDeparture : seules des différences d'interprétation apparaissent (DestinationExternalCode par exemple)

Nom	Type	Format	Défaut	Description	Obligatoire
StopAreaExternalCode	String			Arrêt sur lequel porte la requête. Au choix avec StopPointExternalCode : l'un des 2 est obligatoire.	Oui
StopPointExternalCode	String			Arrêt physique sur lequel porte la requête. Au choix avec StopAreaExternalCode : l'un des 2 est obligatoire	Oui
DestinationExternalCode	String			Filtre pour les prochaines arrivées « ayant pour origine ». Le filtre est actif pour toutes les gares traversées : ce n'est pas forcément l'origine du train, mais celle de l'utilisateur	Non
LineExternalCode	String			Ligne sur laquelle porte la requête	Non
Sens	Integer			Filtre du sens de la ligne utilisée	Non
RouteExternalCode	String			Route sur laquelle porte la requête	Non
ModeTypeExternalCode	String			Filtre du type de mode utilisé	Non
ModeExternalCode	String			Filtre du mode dont le type de mode est utilisé	Non

Time	String	Hour Minute	Heure courante	Heure définissant la première prochaine arrivée	Non
Date	String	Year Month Day	Date courante	Date de la recherche des prochaines arrivées	Non
NbStop	Integer		5	Nombre d'arrivées présentées	Non
UseAdapted	Boolean		Oui	Indique si le traitement doit utiliser les données adaptées	Non
ManageEvent	Boolean		Oui	Indique si le traitement doit tenir compte des données temps réel	Non
DateChangeTime	String	Hour Minute	00 00	Heure de changement de date (présentation en « calendrier TV »).	Non
UseTransday	Boolean		true	Permet de bloquer l'affichage des prochaines arrivées au-delà de DateChangeTime et d'afficher par exemple « fin de service »	Non

Remarque :

- ▶ Le filtre sur « Line » est prioritaire sur le filtre « Route » si les deux sont renseignés.
- ▶ Les horaires de montée et de descente sont présentés. L'application média peut néanmoins simplifier la présentation en affichant uniquement les horaires de départ.
- ▶ La balise « Day » présente dans le flux résultat « NextArrivalList\Stop\StopTime\Day » indique un changement de jour. Si la valeur vaut 1, l'horaire présenté est sur le lendemain par rapport à la date demandée (passe-minuit)

2.7.4 - Exemple d'appel

- ▶▶ Recherche des 2 prochains trains TER et Intercités à l'arrivée de la gare d'Armentières et ayant pour origine Lille Flandres

<http://localhost/?action=nextarrival&StopAreaExternalCode=OCE87286146&DestinationExternalCode=OCE87286005&nbstop=2>

2.8 - Liste des erreurs

Code	Type	Message
101	Configuration	Le chemin du fichier de configuration est invalide
102	Configuration	Le type du fichier de configuration est invalide
201	Configuration	Le paramètre de configuration [Nom du paramètre] ne peut être vide
202	Paramètres d'appels	Le paramètre [Nom du paramètre] est obligatoire
203	Paramètres d'appels	Le paramètre [Nom du paramètre] ne peut être vide
204	Paramètres d'appels	Le paramètre [Nom du paramètre] n'est pas valide
205	Paramètres d'appels	Le paramètre [Nom du paramètre] doit avoir au moins [Nombre] caractère(s)
206	Paramètres d'appels	Le paramètre [Nom du paramètre] ne doit pas avoir plus de [Nombre] caractère(s)
301 à 401	Technique	Erreur technique
403	Format	Le type du contenu de la réponse est différent de text/xml