

SYSTEME D'INFORMATION MULTIMODAL

MANUEL D'INTEGRATION

REF: NAVITIA_MANUEL_INTEGRATION.DOC DU 17/02/2012





STATUT DU DOCUMENT

Statut	DATE	INTERVENANT	VISA
Provisoire	13/08/08	Pascal Rhod	
Approuvé	13/08/08	Stephan Simart	
Validé	11/09/09	Stephan Simart	
Date d'application			
Date de péremption		_	

HISTORIQUE

Qui	Action	Date
Krishna Adhikari	Relecture complète release 1.8	13/09/2009
Krishna Adhikari	Mise à jour mineures et ajout de l'API NextArrival	03/11/2009
Stephan Simart	Mise à jour NAViTiA 1.9	14/09/2010

Ce document est la propriété d'EFFIA - Canal TP. Toute reproduction ou communication partielle ou totale est interdite sans autorisation écrite d'EFFIA - Canal TP



> Sommaire

1 -	Introduction	6
1.1 -	- Rappel sur le fonctionnement de l'application NAViTiA	6
1.2 -	- Conventions	7
1.3 -	- Intégration et pérennité des données	7
1.4 -	- Exemples d'appel de l'application	8
2 -	Obligations et recommandations d'utilisation de NAVITIA	9
_ 2.1 -		
2.2 -		
3 -	Définitions et usage	
3.1 -		
3.2 -		
3.3 -		
3.4 -		
3.5 -		
3.6 -		
4 -	Implémentations types	
4.1 -		
4.2 - 4.3 -	S	
4.3 - 4.4 -	·	
	·	
4.5 -	·	
4.6 -	- Exemple de recherche de points d'entrée avancée	44
5 -	Description des API techniques disponibles	. 48
5.1 -	- Page de constantes « /Const »	48
5.2 -	- Page de rechargement « /Load »	51
6 -	Liste des API sur données théoriques	. 52
6.1 -	- AddressList	52
6.2 -	- AddressTypeList	53
6.3 -	- ContributorList	54
6.4 -	- DirectStopAreaList	55
6.5 -	- EndOfCourse	57
6.6 -	- EntryPoint	60

>>>>>> Introduction

6.7 - ExternalPhonetic64
6.8 - FirstLetter66
6.9 - Isochron68
6.10 - LineRouteDescription74
6.11 - LineStopAreaList74
6.12 - MakeBinaryCriteria76
6.13 - ProximityList78
6.14 - PTReferential80
6.15 - RoutePointList93
6.16 - SiteList95
6.17 - SiteTypeList96
6.18 - StreetNetwork97
6.19 - VPatternSetList101
6.20 - VPTranslator103
7 - Liste des API calculées tenant compte du temps réel 107
7.1 - DepartureBoard107
7.2 - LineSchedule114
7.3 - MultiModalSchedule123
7.4 - NextArrival
7.5 - NextDeparture
7.6 - PlanJourney
7.7 - DetailedJourney153
8 - Liste des API dédiées au temps réel 155
8.1 - Introduction
8.2 - GetImpactlist
8.3 - geteventlist159
8.4 - getimpact161
8.5 - getevent
8.6 - getmedialist162
8.7 - getmessage162
9 - Liste des API tarifaires 164
9.1 - JourneyFare164
10 - Annexes
10.1 - BookingTypeList166

Introduction

10.2 - CityL	ist	167
10.3 - Comr	monLetter	168
10.4 - Comp	oanyList	168
10.5 - Depa	rtmentlist	169
10.6 - Distri	ictList	170
10.7 - Even	tList	171
10.8 - LineL	ist	172
10.9 - Mode	eList	175
10.10 -	ModeTypeList	176
10.11 -	NetworkList	177
10.12 -	OnBoardServiceList	178
10.13 -	Phonetic	179
10.14 -	RouteList	179
10.15 -	StopAreaList	182
10.16 -	VehicleJourneyDetailList	184
10.17 -	VehicleJourneyList	189
10.18 -	Page de description des flux disponibles « /ActionList »	192
10.19 -	Page de liste des modifications / évolutions « /ChangeLog »	193



1 - INTRODUCTION

1.1 - Rappel sur le fonctionnement de l'application NAViTiA

Version d'interface

L'application est composée d'un cœur qui se charge des calculs et d'une interface, qui fabrique la réponse. Chacun de ces 2 « modules » est indépendant et possède son propre numéro de version. Canal TP est donc en mesure de changer l'application NAViTiA sans impacter le « langage d'échange » avec le média (que nous appelons « l'interface ») :

Requête média → application NAViTiA → fabrication de la réponse (interface) → retour média

La version d'interface par défaut utilisée par le web-service est rappelée dans l'API /Const. Il est néanmoins **indispensable** de choisir la version directement dans la requête d'interrogation.

A l'appel de toute requête d'interrogation NAVITIA, il est impératif d'ajouter le paramètre &interface="Numéro souhaité" afin de ne pas rendre l'intégration tributaire des évolutions de version d'interface

Exemple: http://SERVER/.../gwnavitia.dll/...&interface=1_15

Authentification

Afin de gérer au mieux les statistiques d'utilisation de chaque web-service, il est également nécessaire de s'authentifier grâce à son adresse IP, associée à un login.

- A l'appel de toute requête d'interrogation NAVITIA, il est impératif
- → d'ajouter le paramètre &login="Identifiant_fourni_par_Canal_TP"
- → de fournir l'adresse IP de la machine qui se connecte à la plateforme NAVITIA à Canal TP

afin de permettre une gestion de la sécurité et des statistiques optimales.

>> Simplification du débogage



Afin de simplifier les échanges entre les équipes qui se chargent d'intégrer NAVITIA et Canal TP, il est fortement recommander de permettre un accès simple aux requêtes NAVITIA effectués par le média.

Ainsi, l'ajout d'un paramètre dans l'URL du site permettant d'afficher les URL d'appels NAVITIA effectuée par la page permet de qualifier l'origine des bogues

Sur chaque page du site, il est impératif de proposer l'ajout d'un paramètre du type

- → &navitiadebug=1
- → ou /navitiadebug/true par exemple

afin de permettre à Canal TP l'accès direct aux URL NAVITIA appelées par la page courante du site web.

1.2 - Conventions

- Le nœud parent de chaque flux est une liste (Liste de City, de EntryPoint, 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 contenants ces paramètres sont en minuscules
- La première lettre de chaque mot est une majuscule.
 - En cas de mot composé, agréger les deux mots (ex : CityList).
 - L'index (identifiant unique non pérenne utilisé par Navitia), l'identifiant (issu des bases de données initiales) et le nom de chaque type de donné apparaissent en attribut.

1.3 - Intégration et pérennité des données

NAVITIA retourne dans chaque flux XML 3 identifiants différents pour chaque notion :

ExternalCode (ex : StopAreaExternalCode) : Définit par les transporteurs et / ou AO (Local agreement), cet identifiant doit être unique et pérenne,



quelque soient les modifications apportées dans la base de données. Contacter Canal TP afin de déterminer si ces codes seront correctement renseignés.

- Idx (ex : StopAreaIdx) : correspond à l'identifiant unique NAViTiA pour la base de donnée en cours. **Cet identifiant n'est pas pérenne** dès lors que l'on modifie la base de données alimentant NAViTiA.
- ID (ex : StopAreaID) : correspond à l'identifiant issue de la base de données source. Donnée à titre indicatif, il n'est pas forcément unique (agglomération de 2 bases via FUSiO par exemple) et permet simplement de tracer une donnée entre la base source et le rendu final NAVITIA

Il convient de bien choisir parmi ces 3 formes d'identifiants afin de valider son intégration.

1.4 - Exemples d'appel de l'application

Notre exemple utilise un site déclaré fictif « www.navitia.com »

- Pour un flux de requête « liste des constantes de paramétrage de l'application » :
 - « http://www.navitia.com/gwnavitia.dll/const »
- Pour le flux de requête « liste des modes disponibles » : « http://www.navitia.com/gwnavitia.dll/API?Action=PTReferential&requestedType=ModeTypeList » (cf. API 6.14 PTReferential)
- Pour un flux de requête « liste des arrêts commerciaux de la commune 38 » : « http://www.navitia.com/gwnavitia.dll/API?Action=PTReferential&RequestedType=StopareaList&CityExternalCode=38185 »

2 - OBLIGATIONS ET RECOMMANDATIONS D'UTILISATION DE NAVITIA

2.1 - Mention légale

Les utilisateurs (intégrateurs) du service NAViTiA s'engagent à afficher et ne pas porter atteinte, modifier et/ou retirer les notices et mentions de copyright de CANAL TP et de ses fournisseurs devant figurer sur les réponses et/ou les contenus issus (directement ou indirectement) de NAViTiA.

Toutes les pages issues (directement ou indirectement) de NAViTiA devront comporter de manière lisible, et dans la même langue que celle du reste du texte du site, les mentions de copyright concernant le service NAViTiA, et le cas échéant, de Téléatlas :

- Mention NAViTiA seul : « Solution trouvée par NAViTiA© »
- Mention NAViTiA et Téléatlas : « Solution trouvée par NAViTiA©, en partenariat avec Téléatlas© MultiNet™ 2006 ». (La mention « 2006 » correspond à l'année courante : cette mention doit être actualisée à chaque changement d'année).

Ces 2 mentions doivent être entièrement cliquables, et permettre l'accès à la page http://www.navitia.com/about/

L'ensemble des pages d'un site utilisant une ou plusieurs fonctions NAVITIA doit comporter cette mention de façon lisible et cliquable.

En outre, une page du site doit indiquer clairement les logos et descriptif de ces services. Cette page doit être accessible depuis la page d'accueil, et s'ouvrir dans la même fenêtre que celle du site.

Les logos et descriptifs à utiliser sont disponibles à l'adresse : http://www.navitia.com/about/

Le licencié doit afficher tout autre avis de copyright ou autre mention requis par Canal TP dans un délai commercialement raisonnable après avoir reçu une notification écrite émanant de Canal TP.

2.2 - Recommandations

Lors de l'intégration du service NAViTiA, nous recommandons la prise en compte des points suivants :

- Pour favoriser des listes de résolutions phonétiques courtes, nous recommandons l'utilisation de 2 zones distinctes pour dissocier la commune du nom (adresse, voie...),
- Pour accélérer la reconnaissance des communes, il est possible de proposer une fonction « d'auto-completion » reposant sur l'API FirstLetter,

Data ServiceObligations et recommandations d'utilisation de NAVITIA

- Pour éviter des recherches d'itinéraires intempestives, il est **recommandé** d'afficher une page intermédiaire (permettant d'indiquer des mentions légales, et éventuellement toute information conjoncturelle ou éditoriale),
- Les plans de quartier devront présenter les points de départ et d'arrivée de chaque liaison à pied. Le tracé du trajet (API NAVITIA StreetNetwork) entre ces points peut être un complément pertinent.
- Ces plans de quartiers peuvent également être présentés au sein de la feuille de route et non en fenêtre pop-up.
- L'utilisation de javascript doit permettre de faciliter l'utilisation du service. Le site devra néanmoins être accessible de façon moins efficace aux navigateurs dont le javascript est désactivé. (ceci constitue l'un des pré-requis pour obtenir une certification WAI ou AccessiWeb)

3 - DÉFINITIONS ET USAGE

3.1 - 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 existe des commentaires sur les arrêts physiques, sur les lieux remarquables, sur les horaires.
Country	Pays	
District	Région	
Department	Département	
City	Commune	

Notion	Traduction	Commentaire	
Alias	Alias	Intitulé commercial qui référence une autre notion (arrêt commercial, lieu remarquable, adresse 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. Utilisé principalement lors de la compréhension de la demande (API EntryPoint par exemple)	
Site	Lieu remarquable	(Point Of Interest). Un lieu remarquable est un lieu bien connu du public qui peu servir de référence aux usagers pour indiquer l'origine ou la destination d'un traje (Le Louvre par exemple)	
Address	Adresse		
SiteType	nature de lieu remarquable	lieu Permet une classification des lieux remarquables : musée, bibliothèque, restaurant	
AddressType	Type de voirie	Permet une classification des adresses : Avenue, rue, boulevard	
Hang	Accroche avec un arrêt physique	Représente le lien entre un lieu, une adresse ou une commune non géocodée et les points d'entrée du réseau de transport : la liste des arrêts physique situé à proximité d'un lieu remarquable par exemple.	
		Inutilisé si la base de données est géocodée (arrêts physique, lieux remarquables et adresses)	
StopPoint	Point d'arrêt ou arrêt physique	Point d'arrêt du réseau de transport. Exemple : poteau de bus. Utilisé principalement lors de la description d'une réponse (API PlanJourney par exemple)	

Notion	Traduction	Commentaire	
Network	Marque commerciale, réseau de transport	Exemple : réseau TER Pays de la Loire, réseau Noctilien	
Company	Transporteur	Les Autocars XXX	
Mode	Mode commercial de transport	métro, bus, bus articulé, Teoz, TGV	
modeType	Mode générique de transport (normalisé TRIDENT)	metro, bus, train, rapidTransit	
Line	Ligne commerciale	Regroupe les propriétés principales d'une ligne quelque soit sa complexité (ligne avec terminus différents, lignes en boucles)	
		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)	
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	fonctionnement ou	Représente la liste exhaustive des dates pour lesquels la circulation est assurée. Cette représentation est de la forme suivante :	
	condition de service	à partir du <i>date de début de régime</i>	
		111110011111001111100	

Notion	Traduction	Commentaire	
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	
Service	Un service horaire du	Décrit l'ensemble des horaires d'un réseau pour une période donnée	
	réseau	exemple : service été des horaires SNCF	
ODT	Transport à la demande		
EquipmentKind	Type d'équipement possible sur les points d'arrêts	Exemple : abrité, accessibles aux fauteuils roulant, escalier mécanique	
Equipment	Equipement disponible au point d'arrêt	Ensemble des équipements disponibles au point d'arrêt	
Stop	Horaire	Horaire de passage à un arrêt physique. Contient en particulier l'heure d'arrivée, l'heure de départ et l'arrêt physique.	
LineSchedule	Grille horaire de ligne	Liste de tous les horaires entre 2 arrêts physiques d'une ligne. Représente toutes les possibilités de trajet entre 2 points d'une ligne	
DepartureBoard	Horaires à l'arrêt	Liste de tous les horaires de passage à un arrêt pour une ligne	

Notion	Traduction	Commentaire	
Connection	Correspondance	Possibilité de joindre deux arrêts physiques afin de changer de véhicule lors d'un itinéraire en transport en commun.	
ConnectionKind	Type de correspondance	3 types:	
		Intra arrêt physique : on descend du véhicule et on attend un autre véhicule sans bouger. NAViTiA retourne une section (voir la notion « section » ci-dessous) de type StopPointConnection	
		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. NAViTiA retourne 2 sections : LinkConnection suivie de StopPointConnection	
		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. NAViTiA retourne 2 sections : LinkConnection suivie de StopPointConnection	
DemandPoint	Point demandé	Demande du client à interpréter afin de définir les points de départ ou de destination d'un itinéraire. Par exemple :	
		« Pl de gaulle à Paris »	
PointType	Type de point d'entrée	4 types d'entrée possible dans NAViTiA :	
	dans NAViTiA	Arrêt commercial (Stop area)	
		Lieu remarquable (Site)	
		Adresse (Address)	
		Commune (City)	

Notion	Traduction	Commentaire	
EntryPoint	Point d'entrée dans NAViTiA	Réponse de NAViTiA face à une demande de point de départ ou de destination d'un itinéraire	
JourneyResult Résultat d'itinéraire		Entre 2 entry point prédéfini à une date ou une période donnée, en utilisant des critères de recherche spécifiques ou non	
Summary Résumé d'itinéraire		Contient le résumé de'une solution d'itinéraire JourneyResult : date et heur de départ et d'arrivée, nombre de correspondance, clé unique de description de l'itinéraire	
JourneyKey	Clé d'identifiant du type de trajet	Cette clé permet de regrouper entre eux les itinéraires par « type de trajet ». Ainsi lors de l'utilisation de IAPI « multimodaleSchedule », il est possible de présenter de façon identique les itinéraires qui utilisent les mêmes modes, les mêmes correspondances	
SectionKey	Clé d'identifiant du type de segment d'itinéraire	Cette clé permet d'interfacer les API de calculs d'itinéraire (PlanJourney, DetailedJourney) avec les API de calcul de tarif (/Fare)	

Notion	Traduction	Com	mentaire
Section	Segment d'itinéraire	Détail le plus fin d'un itinéraire : il repr mode de transport.	ésente le lien entre 2 points sans changer de
		Plusieurs types possibles (cf. xsd) par e	exemple :
		type=« WalkConnection »	Liaison à pied sur l'ensemble de l'îtinéraire
		type=« VehicleJourneyConnection »	Liaison en transport en commun entre 2 points d'arrêts physique
		type=« ProlongationConnection »	Prolongement de service ou Haut le pied : temps d'attente dans le bus (généralement sur un terminus en boucle, sur une pour réguler les horaires)
		type=« ODTConnection »	Liaison en transport à la demande. Si les horaires sont estimés, lire la propriété EstimatedTime
		Type=« StopPointConnection »	Correspondance au sein d'un point d'arrêt
		Type=« Undefined »	Liaison à pied entre un point géocodé sans nature et un point d'arrêt
SectionLength	Longueur du segment	En mètre. Sur une section de type tran à la somme des longueurs à vol d'oisea	sport en commun, cette longueur correspond u entre chaque arrêt traversé.
IsDisrupt	Perturbé	IsDisrupt=1 si l'itinéraire subit des pert	urbations en temps réel

Notion	Traduction	Commentaire
EstimatedTime	Horaire estimé	EstimatedTime=1 si l'horaire n'est pas garanti. Il convient alors d'afficher une « durée estimée » du segment de trajet concerné. Présenté généralement lors de l'utilisation de Transport A la Demande zonal.
Route Point	Point sur itinéraire	Point d'arrêt sur une ligne. Permet de dessiner les plans de ligne par exemple.
ValidityPatternSet	Cible	Combinaison d'une période (plusieurs plages de date) et de jours de fonctionnement. Exemple : Jours de semaine en période scolaire Dimanche et jours férié
ValidityPatternSetDetail	Détail de cible	Ensemble de jours exacts de circulation avec un renvoi éventuel définissant les différences avec la cible. Par exemple :
		Cible : Jours de semaine en période scolaire
		Détail : « 11011001101100 » + renvoi « sauf mercredi »
Segment	Segment de voierie entre 2 intersections de	Les segments de voierie permettent d'afficher le parcours piéton sur une carte en appelant l'API streetNetwork.
	voierie. Un segment est composé d'un noed de départ, un nœud d'arrivée et des numéros de voierie pairs et impairs	Ils permettent également de positionner une adresse complète (avec numéro de voierie dans l'adresse) par rapport à son segment

Notion	Traduction	Commentaire
Node	Nœud permettant de géolocaliser le début ou la fin d'un segment	



3.2 - Modèles de données

3.2.1 - Le référentiel géographique

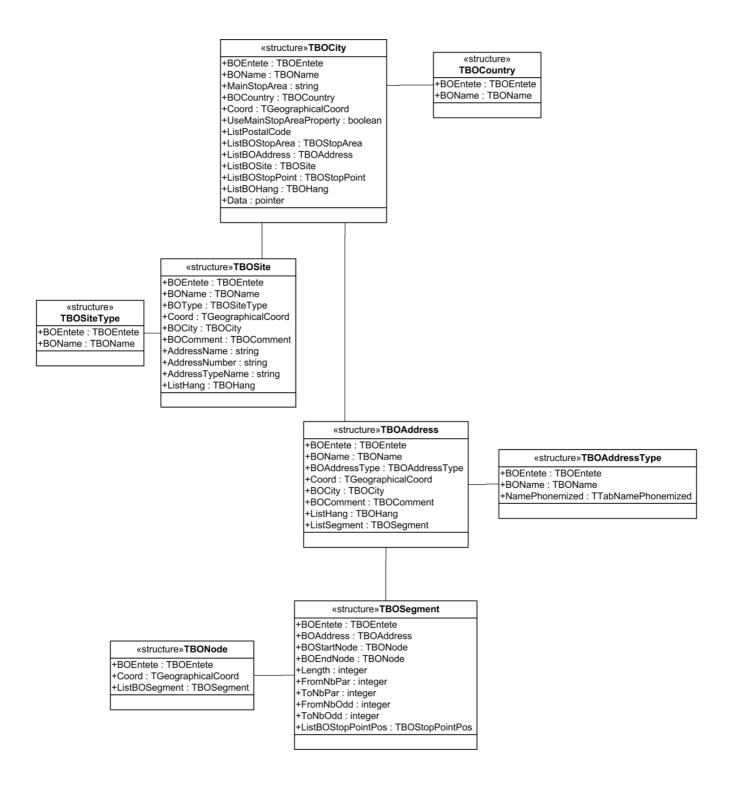
Ce référentiel stocke notamment :

- Les communes,
- Les POI (Point Of Interest), tels que les lieux remarquables, sites...
- Les adresses (voies) avec leurs intitulées, type de voie (« avenue », « rue »...), ainsi que les numéros de seuils¹ (si cette donnée existe),
- Et bien sûr, les coordonnées géographiques de ces différents « objets ».

Le schéma ci-dessous présente les différentes notions de ce domaine.

¹ Numéro dans la voie : « 109 », « 12bis », « 101ter », etc.

[©] Canal TP / NAVITIA_MANUEL_INTEGRATION - 1.9.



3.2.2 - La topologie du réseau

La topologie du réseau définit « la structure » d'une offre TC :

- les points d'arrêts,
- les zones d'arrêts,
- les parcours (regroupant ces points d'arrêts),

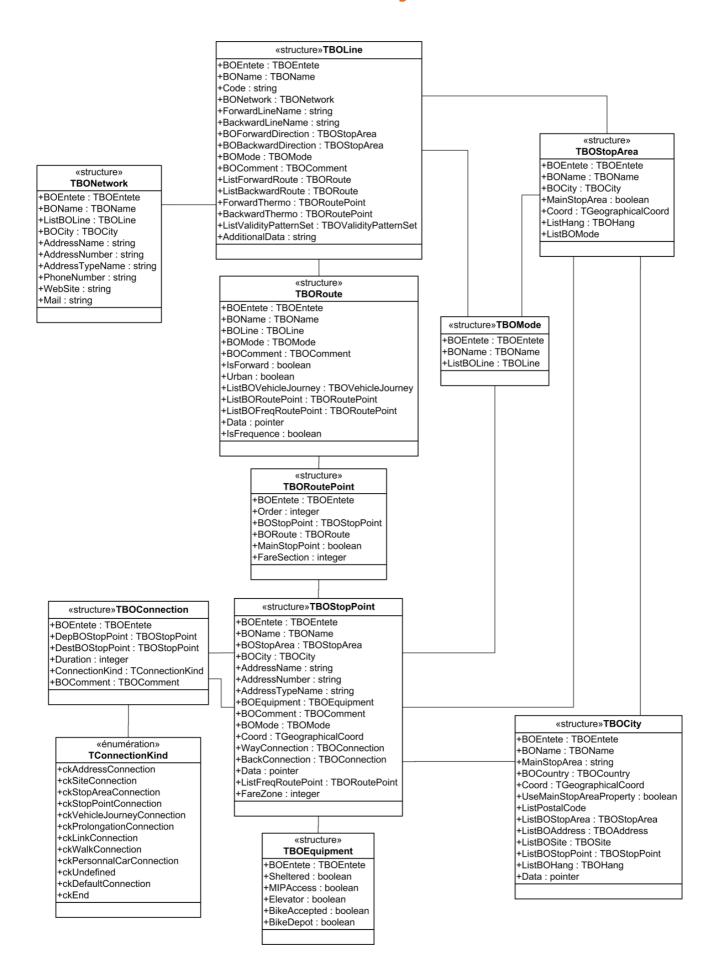
- les lignes (regroupant ces parcours selon des intitulés connus du public),
- les correspondances.

Cette topologie du réseau est « connectée » au référentiel géographique, principalement grâce à 2 propriétés « pivot » :

- les coordonnées géographiques des points d'arrêts notamment,
- la commune d'appartenance des notions point d'arrêts et zone d'arrêts.

La topologie du réseau n'introduit aucune notion liée au détail de l'offre commerciale (horaires, calendrier de circulation).

Le schéma ci-dessous présente les différentes notions de ce domaine.



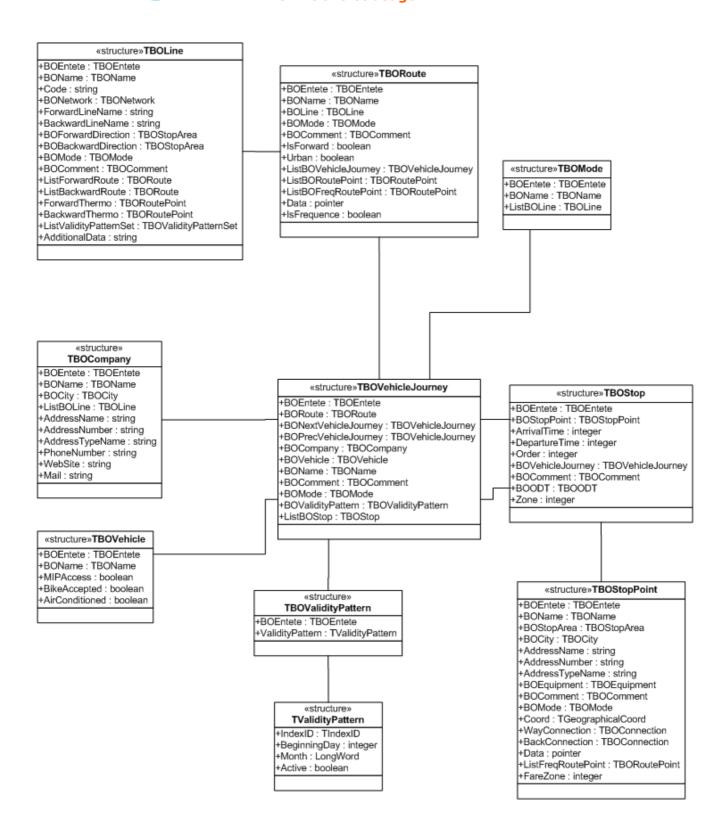
3.2.3 - L'offre commerciale, les horaires

L'offre commerciale (aussi appelée « offre horaire ») est connectée au domaine « Topologie du réseau » grâce aux notions suivantes :

- Point d'arrêt (et donc Zone d'arrêt),
- Ligne.

Ce domaine introduit véritablement le détail de l'offre commerciale : les horaires de passage aux points d'arrêts, les calendriers de circulation, etc.

Le schéma ci-dessous présente les différentes notions de ce domaine.



3.3 - Les données adaptées

Le terme données adaptées désigne des modifications « à chaud » des données NAVITIA pour y intégrer des modifications lourdes des circulations (en période de grève par exemple). Plusieurs outils existent pour alimenter NAVITIA en données adaptées. Lors de l'appel à certaines API, il est possible de demander à NAVITIA

d'utiliser soit ses données habituelles (dites « Théoriques »), soit les données adaptées que le calculateur possède. Le paramètre de requête « UseAdapted » permet de passer de l'une à l'autre de ces types de données. Si ce paramètre a la valeur « 0 », il utilisera les données théoriques, s'il a la valeur « 1 », il utilisera les données adaptées.

Par défaut, le calculateur utilise les données adaptées qu'il y en ait ou pas. De cette manière, l'ajout de données adaptées est implicitement pris en compte dans le calculateur.

3.4 - Les données perturbées

Le terme de données perturbées désigne des données modifiées « en temps réel » pour y introduire des évènements ponctuels concernant le trafic. Ces données s'appuient sur des évènements (Event) impactant différents objets (lignes, routes, circulations, arrêts). NAViTiA peut intégrer ses évènements à ses calculs, en utilisant le paramètre de requête « ManageDisrupt ». Celui-ci accepte la valeur « 0 » (par défaut), pour ne pas tenir compte des perturbations dans les calculs, ou « 1 », pour les prendre en compte.

Certaines API renvoient des informations sur les évènements impactant des objets ; il convient de se référer à la description de l'API.

3.5 - Pagination des flux de réponses

Pour éviter des temps de réponses trop long, les flux de réponse de NAVITIA 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 a 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 : ResponseCount + 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 ».



Dans le cas d'une API renvoyant plusieurs listes d'éléments (VehicleJourneyDetailList, LineSchedule, PlanJourney, ...), on ne pagine que la liste principale.

Les listes de références (CommentList, ODTList, EventList,...), placées en fin de flux, sont complètes et non paginées : elles sont identiques, quel que soit la valeur du « PagerIndex » passée en paramètre.

Exemple : dans le cas de l'API «Lineschedule», la pagination s'effectue sur le nœud «VehicleJourneyList».

3.6 - Index des API

AddressList

Liste des adresses disponibles. Les fonctions de recherche d'adresse à adresse pour Transilien par exemple, sont basées, à ce jour, sur les données TéléAtlas.

AddressTypeList

Liste des types de voirie disponibles (avenue, rue, boulevard, allée, etc.).

ContributorList

Liste des contributeurs.

DecodeBinaryCriteria

Fournit le détail des informations à partir d'un paramètre d'appel obtenue avec MakeBinaryCriteria ; le détail des équipements d'un arrêt ou d'un véhicule, ou une liste de modes.

DepartmentList

Liste des départements.

DepartureBoard

Ensemble des horaires au départ d'un arrêt donné pour une ligne commerciale ou pour une destination donnée.

DetailedJourney

Détail du trajet final en fonction des paramètres d'entrées (critères de recherche de calcul, vitesse de marche à pied, etc.).

DirectStopAreaList

Liste des arrêts commerciaux accessibles directement (sans correspondances) depuis un arrêt commercial.

EndOfCourse

Liste des horaires desservis à la suite de la montée dans une desserte.

EntryPoint

Liste des points d'arrêts / lieux remarquables / adresses phonétiquement proches du nom passé en paramètre. La recherche sur les arrêts commerciaux peut également être associée à un certain type de mode (le mode Train, par exemple, permet de rechercher les arrêts de type « gares »).

ExternalPhonetic

Permet l'utilisation de la fonction phonétique de NAViTiA sur des données externes.

Actuellement, le fonctionnement est limité. NAVITIA ne sait pas encore lire de fichier de données externe. Seul l'appel sur des données connues est donc possible. Cependant cette fonction permet d'ors et déjà de faire des recherches sur un sousensemble d'arrêts commerciaux passés en paramètre.

Une recherche phonétique parmi les arrêts accessibles en directs depuis un arrêt est ainsi possible : il suffit de demander la liste des arrêts accessibles en direct depuis un arrêt puis d'appeler « ExternalPhonetic » sur cette liste d'arrêt uniquement.

Fare

Permet de récupérer le tarif d'un segment d'itinéraire. A utiliser conjointement avec une des API de recherche d'itinéraire (PlanJourney par exemple)

FirstLetter

Liste des points d'arrêts / lieux remarquables dont un des mots commence par le nom passé en paramètre. La recherche sur les arrêts commerciaux peut également être associée à un certain type de mode (le mode Train, par exemple, permet de rechercher les arrêts de type « gares »). Cette fonctionnalité est utile en particulier pour les médias portables (téléphones, Palm, etc...)

Isochron

Calcul de tous les itinéraires au départ d'un point donné. Cette API permet de disposer d'un fichier contenant les temps d'accès et le nombre de correspondance vers tous les points d'arrêts accessibles en transport en commun depuis un point d'entrée (EntryPoint).

Cette fonctionnalité permet de faire l'analyse de l'accessibilité d'une zone grâce aux coordonnées géographiques des points d'arrêts.

LineRouteDescription



Description des parcours d'une ligne.

LineSchedule

Grille horaire d'une ligne entre 2 arrêts à une date donnée.

LineStopAreaList

Liste des arrêts commerciaux situés avant ou après un arrêt commercial pour une ligne commerciale et un sens donné. Elle est utile afin de déterminer la liste des arrêts desservis depuis un arrêt donné avant une demande de grille horaire de ligne par exemple.

L'API peut également retourner l'intégralité des arrêts de la ligne.

MakeBinaryCriteria

Fabriquer des paramètres d'appel avancés pour le lancement d'un calcul d'itinéraire. Typiquement, il s'agit d'affiner la recherche par rapport à un besoin client plus spécifique (arrêts abrités, arrêts avec accès PMT, etc.). La liste actuelle des paramètres d'appel est bien entendu évolutive.

NextArrival

Liste des prochaines arrivées (tableau d'affichage des arrivées) à l'arrêt désiré sur une ligne désiré. Le résultat fournit les données en temps réelles et/ou théorique

NextDeparture

Liste des prochains départs (tableau d'affichage des départs)à l'arrêt désiré sur une ligne désiré. Le résultat fournit les données en temps réelles et/ou théorique

RoutePointList

Liste des arrêts desservis d'une route, pour une route donnée.

PhoneticParams

Ce flux contient les règles phonétiques utilisées : la liste des mots interdits (ghostwords), puis la liste des règles phonétiques.

PlanJourney

Liste des trajets possibles en fonction des paramètres d'entrées (critères de recherche de calcul, vitesse de marche à pied, etc.).

ProximityList

Recherche de proximité géographique à partir d'un point d'arrêt ou d'un lieu remarquable. D'autres implémentations sont à l'étude.

PTReferential

Cette fonction permet de requêter les données topologiques : région, département, commune, type de mode, mode, réseau, ligne, parcours, zone d'arrêts, points d'arrêts... en combinant tous les filtres possibles

SiteList

Liste des lieux remarquables disponibles. Un lieu remarquable est un lieu bien connu du public qui peut servir de référence aux usagers pour indiquer l'origine ou la destination d'un trajet (Le louvre par exemple).

SiteTypeList

Liste des types de lieux remarquables disponibles (musée, hopital, restaurant, parcs d'attractions, etc.).

StreetNetwork

Calcul d'un itinéraire à pied uniquement en suivant le filaire de voirie.

VehicleJourneyList

Liste des circulations disponibles.

VehicleJourneyDetailList

Liste des circulations disponibles avec des précisions, et une autre présentation.

VpatternSetList

Liste des régimes de fonctionnement, sous la forme couples « périodes et jours de fonctionnement » disponibles. Il est utile en particulier lors de demande de grilles horaire de ligne ou de tableaux d'horaire à l'arrêt, que l'on peut alors grouper : horaires de semaine scolaire, horaires de samedi...

4 - IMPLÉMENTATIONS TYPES

4.1 - La recherche d'itinéraire

4.1.1 - Exemples d'appels

Formulaire de saisie d'itinéraire

ModeList pour proposer la liste des modes à utiliser à l'internaute

http://www.navitia.com/cqi-bin/qwnavitia.dll/api?action=ModeTypeList

Const pour récupérer les dates de validités de la base et afficher un message du type « données mise à jour le... »

http://www.navitia.com//cgi-bin/gwnavitia.dll/Const

Validation des paramètres saisis :

EntryPoint pour la recherche d'un point de départ

http://www.navitia.com/cgi-

<u>bin/gwnavitia.dll/api?action=entrypoint&name=my_address&cityname=my_city</u>

EntryPoint pour la recherche d'un point de destination

http://www.navitia.com/cgi-

<u>bin/gwnavitia.dll/api?action=entrypoint&name=my_destination&cityname=destination_city</u>

MakeBinaryCriteria pour fabriquer les filtres avancés de calcul d'itinéraire

http://www.navitia.com/cgi-

<u>bin/gwnavitia.dll/api?action=makebinarycriteria&modetypeexternalcode=bu</u> s;metro;tramway

Calcul d'itinéraire :

PlanJourney pour lancer le calcul d'itinéraire en fonction des paramètres cidessus

http://www.navitia.com/cgi-

bin/gwnavitia.dll/API?action=PlanJourney&Departure=Undefined|||||60135 9|2430821&Arrival=Undefined|||||599140.921|2426234.596&Sens=1&Crit eria=1&WalkSpeed=50&ForbiddenLineExtCode=&ForbiddenStopAreaExtCod



<u>e=&Date=2009|10|5&Time=12|25&HangDistance=500&HangDistanceDep=&HangDistanceArr=&ManageDisrupt=&ManageVLS=</u>

Le détail de cet appel est décomposé chapitre 7.6 - PlanJourney

4.1.2 - Identification du type de réponse

Lorsque les demandes clients sont clairement identifiées, l'appel à l'API PlanJourney permet de lancer le calcul d'itinéraire.

Dans la réponse NAVITIA, la lecture du nœud « JourneyResultList/JourneyResult/Section/Nota » permet de qualifier la réponse du moteur de calcul :

Si NAViTiA trouve une solution

- Si la solution ne comporte qu'une seule section de type "WalkConnection", alors on pourra appeler l'API StreetNetwork pour présenter le trajet à pied. Voir le détail de cette API dans le chapitre 6.18 -StreetNetwork
- Si la solution est classique, afficher celle-ci au voyageur. Préparer les liens « partir plus tôt » et « partir plus tard ». La procédure d'affichage est décrite dans le chapitre 4.1.3 - Affichage de la feuille de route
- Si NAViTiA ne trouve pas de solution (réponse composée d'une seule section de type "End", utilisation alors de la propriété "Nota") pour les raisons suivantes :
 - plan_no_solution
 - L'itinéraire n'est pas possible en transport en commun.
 - Proposer au voyageur de lancer une nouvelle recherche.
 - plan_date_out_of_bounds
 - La date de recherche est hors de la période de validité des données.
 - Proposer au voyageur de lancer une nouvelle recherche sur une date compatible avec les dates de validité de la base.
 - plan_departure_destination_too_close"
 - Les points de départ et de destination sont trop proches pour lancer un calcul d'itinéraire.
 - Proposer au voyageur de faire le trajet à pied.
 - plan_no_departure_stoppoint, plan_no_destination_stoppoint et plan_no_dep_and_dest_stoppoint
 - Ce cas se produit essentiellement lors de calcul d'itinéraire sur des zones peu desservies. NAVITIA ne trouve pas de point d'arrêt dans une zone délimité par les paramètres « HangDistanceDep » et « HangDistanceArr ». Le rabattement entre le point de départ ou

333333 Implémentations types

d'arrivée et le réseau de transport en commun ne peut donc se faire à pied. Il importe alors de proposer au voyageur un mode de transport complémentaire : un mode doux ou son véhicule particulier par exemple.

- La procédure classique à appliquer est décrite dans le chapitre 4.1.2.1 Mise en œuvre d'un rabattement.
- D'autres cas de non solution existent. La liste exhaustive est détaillée chapitre 7.6.6 Cas de non solution d'itinéraire.

4.1.2.1 Mise en œuvre d'un rabattement

Lors d'une recherche d'itinéraire multi-modale, l'internaute a la possibilité de spécifier son point de départ en indiquant :

- Une adresse,
- Un lieu,
- Une zone d'arrêt.

Pour calculer l'itinéraire « TC », le système NAViTiA doit tout d'abord « rabattre » le point de départ ou d'arrivée vers au moins un point d'arrêt du réseau TC.

En sélectionnant une zone d'arrêt, le rabattement est automatiquement effectué sur le ou les points d'arrêts constituant cette zone d'arrêt.

En revanche, pour une adresse et un lieu, NAViTiA peut rabattre en utilisant les coordonnées géographiques (du lieu ou de l'adresse). On indique alors à NAViTiA la distance maximale de rabattement que NAViTiA sera autorisé à parcourir (en suivant le filaire de voirie) pour relier le point de départ (adresse ou lieu) ou le point d'arrivée au réseau TC.

Cependant, sur certaines zones périurbaines ou sur des zones interurbaines, une distance de 1000 mètres (valeur par défaut) peut se révéler insuffisante : le système NAVITIA indiquera alors qu'il n'a pu trouver aucun arrêt TC dans la zone de recherche qui lui a été autorisée (rabattement du point de départ et/ou du point d'arrivée sur le réseau TC pour trouver au moins un point d'arrêt de départ et d'arrivée).

Compte-tenu des contraintes de rabattement décrites ci-dessus, il est nécessaire de prévoir une fonction de rabattement évoluée et modulaire, permettant de répondre efficacement à l'ensemble des recherches d'itinéraires, indépendamment de la localisation du point de départ ou d'arrivée.

Ces lieux de rabattement sont les parkings-relais ou les gares SNCF, et sont appelés « Lieux d'intermodalité TC/VP ».

Implémentations types

La solution retenue consiste à afficher les choix avancés de rabattement uniquement dans les cas où l'approche générale (marche initiale et/ou finale) ne suffit pas (en raison d'une distance trop importante entre le point d'arrêt TC et le point de départ ou de destination final).

Dans ce cas, au lieu de proposer un message du type 'aucune solution ne répond à votre demande', le système demande à l'internaute les modes d'approche qu'il souhaite utiliser pour le départ et/ou l'arrivée, parmi les choix :

- Marche à pied étendue (augmentation de la distance maximale de rabattement),
- Vélo,
- VP/Taxi (ou autre engin à moteur).

A chaque mode de transport est affiché le nombre d'arrêts accessibles. La validation de son ou ses choix relance le calcul d'itinéraire.

4.1.2.2 Mise en œuvre d'un rabattement évolué

Traitement des cas d'absence de point d'accroche

Lors de l'appel de l'API PlanJourney, le flux de retour indique si un ou plusieurs parcours ont été trouvés. Si aucun point d'accroche n'a été trouvé au départ ou à l'arrivée, le flux de retour comporte un nœud :

- En cas d'absence de point d'accroche au départ : JourneyResultList/JourneyResult/Section/Nota, notatype = plan_no_departure_stoppoint
- Sans point d'accroche à l'arrivée : JourneyResultList/JourneyResult/Section/Nota, notatype = plan_no_destination_stoppoint
- Ou les deux : JourneyResultList/JourneyResult/Section/Nota, notatype = plan_no_dep_and_dest_stoppoint
- Affichage de la prise en compte des lieux d'intermodalité TC/VP

Dans ces cas, la page de résultat contient :

- Un résumé des données saisies dans le formulaire de recherche avancé ou de précision.
- Sous le titre « Recherche infructueuse », un message « Votre point de départ n'a pu être rattaché à aucun arrêt » ou « Votre point d'arrivée n'a pu être rattaché à aucun arrêt »



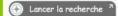
ou « Vos points de départ et d'arrivée n'ont pu être rattachés à aucun arrêt » est affiché, suivant le cas.

- Sous ce message sont indiqués le nombre d'arrêts accessibles en utilisant d'autres moyens de transport, au départ et à l'arrivée, avec un bouton pour relancer la recherche.
- Un bouton de recherche

Pour le formulaire de saisie, on peut s'inspirer du modèle suivant (il faudra tenir compte de la charte graphique du site et non celle de la maquette) :

Votre point d'arrivée n'a pu être attaché à aucun arrêt.

- O Vous souhaitez marcher 30 mn pour atteindre un des 3 arrêts accessibles vers votre point d'arrivée
- O Vous souhaitez utiliser un vélo pour parcourir jusqu'à 5 km pour atteindre un des 6 arrêts ainsi accessibles
- O Vous souhaitez utiliser un véhicule à moteur pour parcourir jusqu'à 20 km pour atteindre un des 12 arrêts ainsi accessibles



L'internaute ne peut choisir qu'un seul mode de transport pour le départ, et un seul pour l'arrivée. Les modes de transport peuvent être combinés si aucun point d'accroche n'a été trouvé au départ et à l'arrivée : un départ en voiture et une arrivée à pied par exemple.

Affichage des propositions

La recherche des points d'arrêts est effectuée avec l'API ProximityList. Cette API est lancée pour chaque mode de transport, soit entre 3 et 6 suivant s'il un seul point d'accroche n'a pas été trouvé ou les 2.

Les paramètres à transmettre sont :

- Type : StopPoint
- X et Y : les coordonnées du point de départ ou d'arrivée
- Distance : à calculer en fonction de la vitesse du moyen de transport

Pour la distance :

- Marche à pied : il faut calculer la distance parcourue en n minutes à la vitesse sélectionnée au moment de la recherche (par exemple, 33 m/min pour une vitesse lente, soit une distance de 990 mètres pour 30 minutes de marche à pied maximum)
- Pour le vélo et la voiture, les distances sont fixes : 5km pour le vélo et 20km pour la voiture par exemple.

Le temps de parcours à pied ainsi que les distances fixes pour le vélo et la voiture sont paramétrés dans un fichier.

Description Implémentations types

Le nombre d'arrêts est indiqué dans le flux retour : ProximityList/ProximityCount.

Prise en compte des choix

Un bouton « Recherche » permet de relancer la recherche avec les nouveaux points d'accroches choisi par l'internaute.

Au moment de la validation, tous les points d'accroche sont récupérés pour être injectés dans le nouveau PlanJourney. Les StopPointIdx des points d'accroche sont associés au point de départ sous forme de Hang dans le PlanJourney et le DetailedJourney (voir le chapitre 7.6 - PlanJourney pour l'utilisation des Hang).

Ces StopPointIdx devront être réutilisable en cas de demande de calcul du chemin retour. Par contre, ils ne seront plus pris en compte en cas de demande de modification de la recherche initiale.

4.1.3 - Affichage de la feuille de route

Note : les valeurs sont données à titre d'exemple (ModeExternalCode...), et doivent être validées en phase d'initialisation de projet.

4.1.3.1 Identification des concepts

L'itinéraire retourné par le calculateur est une succession de « section » chacune composée d'un départ, d'une arrivée et d'un mode pour les rejoindre. Une section « VehicleJourneyConnection » permet de définir quelle ligne de transport en commun à utiliser, entre quels arrêts et à quel heure. Les différents types de sections sont détaillés dans le chapitre 7.6 - PlanJourney.

4.1.3.2 Affichage des sections en transport en commun

Ce chapitre présente les champs à utiliser.

Dans les chapitres suivants, chaque concept sera introduit par son nom (présenté en gras ci-dessous).

Dans les traitements, ou pour toute opération nécessitant de lier un concept à une ressource externe (image, etc.), <u>seul</u> le code du concept (ou « identifiant pérenne ») doit être utilisé. De façon exceptionnelle (exemple : N° de train), on peut choisir d'utiliser un code pour affichage).

Pour chaque concept, le libellé (ou Intitulé) peut être utilisé pour afficher un libellé. En aucun cas un libellé ne doit être utilisé pour des traitements ou une opération de liaison vers une ressource externe.

- Réseau utilisé : « VehicleJourney\Route\Line\Network »
 - Intitulé du réseau utilisé : « .\@NetworkName »
 - Identifiant pérenne du réseau utilisé : « .\@NetworkExternalCode »
- Mode utilisé : « VehicleJourney\Mode\ »
 - Intitulé du mode utilisé : « .\@ModeName »
 - Identifiant pérenne du mode utilisé : « .\@ModeExternalCode »
- Ligne utilisée : « VehicleJourney\Route\Line\ »
 - Code commercial de la ligne : ».\@LineCode »
 - Intitulé de la ligne :
 - Si « VehicleJourney\Route\@IsForward= »1 », afficher
 « VehicleJourney\Route\Line\Forward\@ForwardName »
 - Sinon afficher« VehicleJourney\Route\Line\Backward\@BackwardName »
- Mission (circulation/desserte) utilisée : « VehicleJourney\ »
 - Numéro de circulation (exemple : N° de train) : « .\@VehicleJourneyExternalCode »
 - Intitulé de la mission (exemple : girouette d'un bus) : « .\@VehicleJourneyName »
- - Intitulé de la zone d'arrêt : « .\@StopAreaName »
 - Intitulé de la commune du point d'arrêt : « .\City\@CityName »

4.1.3.3 Modèle

Pour chacun des chapitres ci-dessous, on détaille les concepts qui peuvent être utilisés dans la feuille de route.

- Modèle 1 : urbain, suburbains et départementaux
 - Cible: réseaux urbains, suburbains et départementaux
 - Concepts à utiliser :
 - Réseau
 - Mode
 - Ligne, en particulier le code commercial de ligne

DDD DDD DDD Implémentations types

- Mission
- Direction
- Modèle 2 : relations ferrées ou offres sans lignes
 - Cible: liaison ferrées et offres régionales
 - Concepts à utiliser :
 - Réseau
 - Mode
 - Mission, en particulier le Numéro du train
 - Direction

(Pas de concept ligne).

4.1.3.4 Utilisation des modèles

Les intitulés sont fournis pour plus de clarté dans l'algorithme. Pour chaque concept, les codes à utiliser sont explicités au chapitre 4.1.3.5 Codes à utiliser.

SI le MODE vaut « TGV » ou « TER » ou « CARTER » ou « Train »

ALORS

Appliquer le modèle : « Modèle 2 : relations ferrées ou offres sans lignes »

SINON

Appliquer le modèle 1 : « Modèle 1 : urbain, suburbains et départementaux »

FIN SI

4.1.3.5 Codes à utiliser

Ce chapitre fournit la correspondance entre les intitulés indiqués au chapitre 4.1.3.4 Utilisation des modèles et les codes pérennes à utiliser.

NOTE: Si plusieurs codes s'appliquent pour un même intitulé, ils sont séparés par « ; » (virgule) dans la colonne « Code pérenne ».

Intitulé	Code(s) pérenne(s)
TGV	LongDistanceTrain
TER	LocalTrain

333333 Implémentations types

Intitulé	Code(s) pérenne(s)
Car TER	Coach;LocalTrain
Train	Train;LocalTrain
Avion	Air

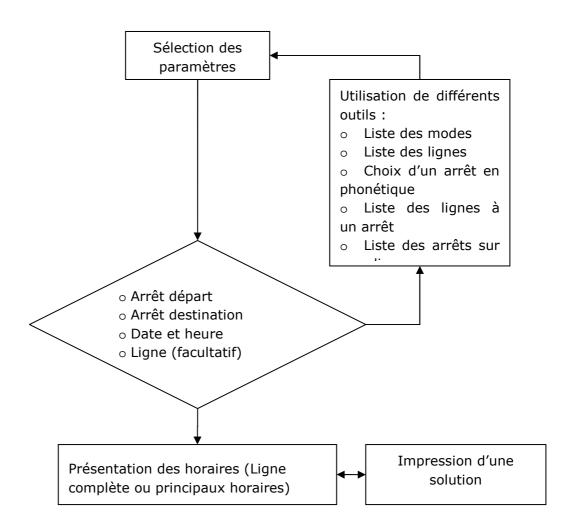
4.2 - La grille horaire entre 2 arrêts

Besoin : Consultation des horaires entre deux arrêts, à partir d'une date donnée avec possibilité de consulter le détail du parcours pour chaque horaire (abcdtrains.com sur wap par exemple qui est un site permettant de diffuser les horaires des trains du réseau transilien en période de grève).

Les API qui sont utilisées pour correspondre à ce besoin sont :

- EntryPoint pour la recherche d'un arrêt commercial de type « gare »
- DirectStopAreaList pour la recherche des gares d'arrivées à partir d'une gare de départ,
- ExternalPhonetic pour la recherche d'une gare parmi la liste reçue de DirectStopAreaList
- LineSchedule pour la récupération des horaires entre deux gares,
- EndOfCourse pour la récupération des horaires de passage d'un train et des arrêts desservis correspondant à un horaire donné (pop-up de détail par exemple).

Exemple de cinématique :



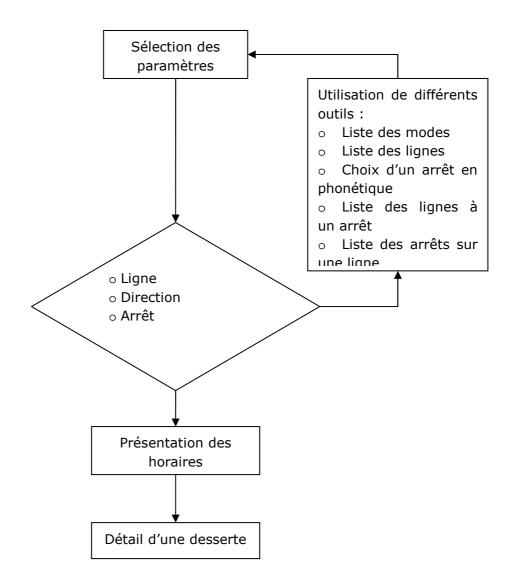
4.3 - Horaires au départ

Besoin : Consultation des horaires de passage à l'arrêt pour une date donnée avec possibilité de consulter le détail du parcours pour chaque horaire. Cette fonction est classiquement présentée pour les lignes urbaines

Les API qui sont utilisées pour correspondre à ce besoin sont :

- PTReferential sur LineList pour le choix de la ligne sur laquelle on souhaite consulter les horaires
- Affichage des 2 sens de la ligne pour sélection par l'internaute
- Affichage des arrêts desservis dans le sens choisi
 - PTReferential sur RouteList pour récupérer les différentes Route de la Line pour un sens donné
 - PTReferential sur StopPointList pour présenter la liste des arrêts desservis par la ligne
 - Etant donné une ligne à antenne LineAB, composée de RouteA1 et Route
 A2 en sens aller, et RouteB1 et RouteB2 en sens retour
 - L'appel suivant par exemple nous fournira donc les 2 Route de sens Aller: RouteA1 et RouteA2:
 - http://SERVEUR/API?action=ptreferential&requestedType=routelist&Line ExternalCode=LineAB&Sens=1&interface=1 14
 - La liste de tous les arrêts desservis en sens aller sera donc obtenue selon l'appel :
 - http://SERVEUR/API?action=PTReferential&requestedType=StopPointList&RouteExternalCode=RouteA1;RouteA2&interface=1_14
- DepartureBoard pour la récupération des horaires,
- EndOfCourse pour la récupération des horaires de passage et des arrêts desservis correspondant à un horaire donné (pop-up de détail par exemple).

Exemple de cinématique :



4.4 - Utiliser les données perturbées

Lors d'une recherche d'itinéraire, il est possible de spécifier si l'on souhaite calculer un trajet « théorique », qui ne tiendra pas compte des perturbations, ou un trajet « de contournement », qui permet, si cela est possible, d'éviter les perturbations.

Un paramètre de requête, « ManageDisrupt », permet d'effectuer ce choix. Par défaut, sa valeur est à 0, ce qui signifie que le calculateur travaillera en mode théorique. On peut passer sa valeur à 1 pour recherche un itinéraire de contournement.

333333 Implémentations types

Il existe plusieurs classes d'évènements : informations, perturbations légères, perturbations bloquantes. Le calculateur n'exclut que les itinéraires impactés par des perturbations bloquantes.

Le flux de réponse d'un calcul d'itinéraire contient la liste de tous les évènements l'impactant, ainsi qu'un lien entre les différents éléments, et leurs évènements. Un itinéraire de contournement ne contient pas de perturbations bloquantes. Il est possible qu'aucun itinéraire de contournement ne soit possible, auquel cas Navitia renvoit un message d'erreur.

Une implémentation type des données perturbées consiste à ne proposer par défaut que des calculs d'itinéraires théoriques. Lors de la mise en page des feuilles de routes, on affichera également les évènements impactant le trajet. En cas de perturbations bloquantes, on proposera un lien permettant de relancer le même calcul, en tenant compte des perturbations.

Liste des API tirant parti du temps réel :

Calcul d'itinéraire : PlanJourney et DetailedPlanJourney

Grille horaire multimodale : MultimodalSchedule

Horaires de lignes : LineSchedule

Horaires à l'arrêt : DepartureBoard

Prochains départ : NextDeparture

Prochaines arrivées : NextArrival

La mise en place des informations perturbations dans la chaine NAVITIA nécessite une Version d'interface 1.11 au minimum sur les API concernées.

4.5 - Utiliser les données adaptées

Pour utiliser les données adaptées (données de grèves, etc.) présentes dans le système, il faut spécifier le paramètre « UseAdapted=1 ». Passer le paramètre « UseAdapted=0 » permet à contrario de ne pas utiliser les données adaptées, mais les données théoriques.



Si aucun paramètre n'est spécifié, l'API de calcul d'itinéraire utilise les données adaptées (données de grèves, etc.) présentes dans le système.

4.6 - Exemple de recherche de points d'entrée avancée

4.6.1 - Besoin exprimé pour l'exemple

La résolution d'ambiguïté doit fournir pour une simple saisie d'un internaute :

- la liste des communes liées à cette saisie ET
- la liste des gares et points d'arrêts liés à cette saisie si elle existe

4.6.2 - Solution proposée

L'API NAVITIA à utiliser pour la résolution d'ambiguïté est l'API « EntryPoint » qui permet de combiner une recherche phonétique avec une recherche par faute de frappe. Cette API possède en particulier les paramètres suivants :

Nom	Туре	Description	Obligatoire
Filter	All / StopArea;Address;Site	Permet de ne rechercher que sur certains types de points de demande	Non
Name	String	intitulé du point de demande désiré	Oui
CityName	String	intitulé de la ville désiré	Non
NbMax	Integer	Nombre maximum de solution à renvoyer	Non

La solution consiste à **dédoubler les appels à NAVITIA** pour chaque type de point d'entrée, City ou StopArea. Ensuite, on ajoutera un appel de recherche de données afin d'obtenir l'exhaustivité des gares sur les communes obtenues. La réponse présentée à l'internaute sera une fusion des 2 réponses NAVITIA par le média.

Ainsi seront gérées :

- Les gares dont le nom contient un nom de commune, mais situées en dehors de la commune (gare d'Avignon TGV par exemple)
- Les communes qui permettent d'appeler un calcul d'itinéraire qui proposera automatiquement la meilleure gare à utiliser
- L'ensemble des gares d'une commune.

4.6.3 - Procédure détaillée

DDD DDD DDD Implémentations types

- Récupération de la saisie exacte de l'internaute. Initialisation de la liste de réponse finale Liste_Finale à présenter à l'internaute.
- Premier appel à l'API EntryPoint avec les paramètres suivant :
 - Name= saisie de l'internaute
 - Filter=StopArea
 - NbMax=15

L'application média stocke alors les meilleures réponses de type « Gare/arrêt » dans la liste Liste finale

- Deuxième appel à l'API EntryPoint avec les paramètres suivant :
 - Name = saisie de l'internaute
 - Filter=City
 - NbMax=15

L'application média ajoute alors les meilleures réponses de type « commune » à la suite de la liste Liste_finale

- Pour chaque commune fournie par NAViTiA, appel à l'API StopAreaList avec les paramètres suivant :
 - CityExternalCode = le code INSEE de la commune sur laquelle on souhaite obtenir la liste des arrêts.

Affichage des 20 solutions au maximum contenu dans Liste_Finale dans une liste déroulante.

4.6.4 - Exemple pour une saisie « Paris » sur les données nationales actuelle

Récupération de la liste des arrêts :

http://[ServerNational]/api?action=entrypoint&name=paris&filter=stoparea

3333333 Implémentations types

Il existe 14 gares correspondantes

Récupération de la liste des communes :

http://[ServerNational]/navitia.ashx/api?action=entrypoint&name=paris&filt
er=city

Une solution de type commune : Paris

Récupération de la liste des gares sur cette commune

http://[ServerNational]/navitia.ashx/api?action=stoparealist&cityexternalcode=75056

Affichage de la liste globale :

>>>>>> Implémentations types



5 - DESCRIPTION DES API TECHNIQUES DISPONIBLES

5.1 - Page de constantes « /Const »

5.1.1 - Appel

« http://www.navitia.com/navitia.ashx/const »

5.1.2 - Description

- L'appel de cette page permet :
 - de recharger les constantes utilisées par l'application
 - d'afficher toutes les informations utiles aux différents paramétrages (paramètres Internet, numéro de version de la base en cours d'utilisation, etc.)
- Le nœud « Server » contient : le nom de la machine sur laquelle l'application NAVITIA est *exécutée*, et l'IP du poste ayant appelé cette page.
- Ce nœud contient en outre les informations systèmes :
 - Mémoire physique totale de la machine en octet (l'affichage est limité **2147483647** si la mémoire est supérieure à 2Go)
 - Mémoire physique libre : affiche **2147483647** tant que la mémoire libre est supérieure à 2Go, affiche la mémoire libre en octet dès que la valeur passe sous la barre des 2Go
 - Mémoire libre sur le disque dur système
 - Mémoire libre sur le disque dur exécutant NAViTiA
- Le nœud « DLL » contient :

Description des API techniques disponibles

- le numéro de version de l'application en cours d'utilisation,
- le numéro de version de structure des données compatibles avec l'application NAViTiA chargée
- le numéro de version des flux xml générés (numéro identique à la version des xsd généré par NAViTiA)
- le chemin local au serveur du fichier de paramétrage de l'application,
- le statut de l'application,
- le chemin local au serveur du fichier d'enregistrement des erreurs
- le type de compilation du moteur
- le type d'enregistrement des fichiers statistiques
- L'utilisation de la fonctionnalité « Passe-minuit »

Le nœud « Thread » contient :

- <Thread>
 <MaxThread>32</MaxThread>
 <ActiveThread>1</ActiveThread>
 </Thread>
 - le nombre maximum de clients simultanés autorisés pour l'instance appelée de NAViTiA,
 - le nombre de clients actifs (en cours de traitement) au moment de l'appel a cette fonction
- Le nœud « Database» contient :

Description des API techniques disponibles

- <Database>

```
<DatabasePath>D:\Users\fusio-Midi-Pyrenees\SitePreprod\Databases\
 <DatabaseVersion>46</DatabaseVersion>
- < Production Date >
 - <StartDate>
     <Year>2005</Year>
     <Month>04</Month>
     <Day>25</Day>
   </StartDate>
 - <EndDate>
     <Year>2006</Year>
     <Month>04</Month>
     <Day>25</Day>
   </EndDate>
  </ProductionDate>
- <PublicationDate>
   <Year>2006</Year>
   <Month>02</Month>
   <Day>08</Day>
   <Hour>15</Hour>
   <Minute>55</Minute>
 </PublicationDate>
 <LastLoadDate>mardi 14/02/2006 04:01:48 PM</LastLoadDate>
 <LastLoadDuration>00:00:00</LastLoadDuration>
 <ZipFile>D:\Users\fusio-Midi-Pyrenees\SitePreprod\Databases\databases.zip</ZipFile>
 <BackupZipDir>D:\Users\fusio-Midi-Pyrenees\SitePreprod\Databases\BackUp\</BackupZipDir>
</Database>
```

- le chemin local au serveur où sont situés les fichiers binaires (fichiers de données au format NAVITIA) utilisés par l'application,
- le numéro de version de structure de ces fichiers binaires,
- les dates de production de début et de fin de ces fichiers binaires, (StartDate et EndDate)
- la date a laquelle ces fichiers ont été générés par le module de publication de NAVITIA (PublicationDate),
- la date et l'heure du dernier chargement des données,
- la durée du dernier chargement effectué
- Le chemin des données compressées à charger
- Le chemin du répertoire dans lequel historiser les anciennes données

5.1.3 - Exemple de réponse d'un appel à « /Const »

Description des API techniques disponibles

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
< const>
  <Params Function="const"/>
   <Description>Constant page - params description - Navitia ISAPI
  <Message>Const loaded</Message>
 - <Server>
    <ServerName>CTP06</ServerName>
    <YourIP>192.0.0.45</YourIP>
    <TotalPhysicalMemory unit="Mo">2047</TotalPhysicalMemory>
    <FreePhysicalMemory unit="Mo">565</FreePhysicalMemory</pre>
    <SystemHDDFreeBytes unit="Mo">6268</SystemHDDFreeBytes>
    <NavitiaHDDFreeBytes unit="Mo">0</NavitiaHDDFreeBytes>
   </Server>
 - < DLL>
    < Version > 1.4.46.524 < / Version >
    <TypeVersion>46</TypeVersion>
    <ParserVersion>1_4_46_3
    <ConfigurationFile>\\?\D:\Users\FUSiO-Midi-Pyrenees\SitePreprod\cgi-shl\navitia.txt</ConfigurationFile>
    <DLLState>Ready</DLLState>
    <ErrorLogPath>D:\Users\fusio-Midi-Pyrenees\SitePreprod\Databases\log\error.log</ErrorLogPath>
    <HandleLargeAddresses>False</HandleLargeAddresses>
    <Log TypeSaveLog="Regular" Action="False" PlanJourney="False" Error="False">Nil</Log>
  </DLL>
 < <Thread>
    <MaxThread>8</MaxThread>
    <ActiveThread>1</ActiveThread>
  </Thread>
    <DatabasePath>D:\Users\fusio-Midi-Pyrenees\SitePreprod\Databases\</DatabasePath>
    <DatabaseVersion>46</DatabaseVersion>
   - < ProductionDate >
    - <StartDate:
       <Year>2005</Year>
       <Month>04</Month>
       <Day>25</Day>
     </StartDate>
    - <EndDate>
       <Year>2006</Year>
       <Month>04</Month>
       <Day>25</Day>
     </EndDate>
    </ProductionDate>

    <PublicationDates</li>

     <Year>2006</Year>
     <Month>02</Month>
      <Day>08</Day>
     <Hour>15</Hour>
      <Minute>55</Minute>
    </PublicationDate>
    <LastLoadDate>mardi 14/02/2006 04:01:48 PM</LastLoadDate>
    <LastLoadDuration>00:00:00</LastLoadDuration>
    <ZipFile>D:\Users\fusio-Midi-Pyrenees\SitePreprod\Databases\databases.zip</ZipFile>
    <BackupZipDir>D:\Users\fusio-Midi-Pyrenees\SitePreprod\Databases\BackUp\</BackupZipDir>
   </Database>
 </const>
```

5.2 - Page de rechargement « /Load »

Cette requête pilote le rechargement d'un nouveau jeu de données pour le système. Ce rechargement s'effectue sans interruption de service

Cette requête peut être filtrée en fonction de l'IP source et du login passé en paramètre : on peut donc filtrer l'utilisation de cette API à certaines adresses IP déclarées dans le fichier de configuration de cette application.

6 - LISTE DES API SUR DONNÉES THÉORIQUES

Les flux suivants contiennent les informations de sortie de NAVITIA à destination des clients finaux (internautes, centre d'appel). Ces flux ne contiennent pas d'information perturbation.

Exemple : sur un site fictif, appel d'un point de demande sous forme phonétique :

« http://www.navitia.com/cgi-bin/gwnavitia.dll/api?action=EntryPoint&Filter=all&Name=Gare du Nord&CityName=Paris »

Le « tableau des paramètres de l'API » peut être consulté directement sur l'application en utilisant « /ActionList »

6.1 - AddressList

6.1.1 - Appel

« /API?Action=AddressList »

6.1.2 - Description

Ce flux contient la liste des adresses disponibles.

6.1.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
CityExternalCode	String			Ne renvoie que les adresses appartenant à la commune dont le code externe est passé en paramètre	Non
AddressExternalCode	String			Ne renvoie que l'adresse dont le code externe est passé en paramètre	Non

6.1.4 - Exemple

3333333 Liste des API sur données théoriques

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionAddressList>
 - <Params Function="API">
     <action>addresslist</action>
     <addressidx>1</addressidx>
   </Params>
 - <AddressList AddressCount="1">
   - <Address AddressIdx="1" AddressId="203947" AddressName="Beauvoir" AddressExternalCode="57415">
       <City CityIdx="1" CityId="1" CityName="Abbéville-la-Rivière" CityExternalCode="91001" CityCode="91150" />
       <AddressType AddressTypeIdx="0" AddressTypeId="-1" AddressTypeName="" AddressTypeExternalCode="" />
         <CoordX>589844,63</CoordX>
         <CoordY>2371226,81</CoordY>
       </Coord>
     </Address>
   </AddressList>
 </ActionAddressList>
```

6.2 - AddressTypeList

6.2.1 - Appel

« /API?Action=AddressTypeList »

6.2.2 - Description

Ce flux contient la liste des types de voirie disponibles.

6.2.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
AddressTypeExternalCo de	String			Ne renvoie que le type d'adresse dont le code externe est passé en paramètre	Non

6.2.4 - Exemple

3333333 Liste des API sur données théoriques

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionAddressTypeList>
 - <Params Function="API">
     <action>addresstypelist</action>
   </Params>
 - <AddressTypeList AddressTypeCount="24">
     <AddressType AddressTypeIdx="0" AddressTypeId="-1" AddressTypeName="" AddressTypeExternalCode="" />
     <AddressType AddressTypeIdx="1" AddressTypeId="0" AddressTypeName="allée" AddressTypeExternalCode="1" />
     <AddressType AddressTypeIdx="2" AddressTypeId="1" AddressTypeName="avenue" AddressTypeExternalCode="2" />
     <AddressType AddressTypeIdx="3" AddressTypeId="2" AddressTypeName="boulevard" AddressTypeExternalCode="3" />
     <AddressType AddressTypeIdx="4" AddressTypeId="3" AddressTypeName="chemin" AddressTypeExternalCode="4" />
     <AddressType AddressTypeIdx="5" AddressTypeId="4" AddressTypeName="cours" AddressTypeExternalCode="5" />
     <AddressType AddressTypeIdx="6" AddressTypeId="5" AddressTypeName="hameau" AddressTypeExternalCode="6" />
     <AddressType AddressTypeIdx="7" AddressTypeId="6" AddressTypeName="impasse" AddressTypeExternalCode="7" />
     <AddressType AddressTypeIdx="8" AddressTypeId="7" AddressTypeName="lieu-dit" AddressTypeExternalCode="8" />
     <AddressType AddressTypeIdx="9" AddressTypeId="8" AddressTypeName="passerelle" AddressTypeExternalCode="9" />
     <AddressType AddressTypeIdx="10" AddressTypeId="9" AddressTypeName="passage" AddressTypeExternalCode="10" />
     <AddressType AddressTypeIdx="11" AddressTypeId="10" AddressTypeName="place" AddressTypeExternalCode="11" />
     <AddressType AddressTypeIdx="12" AddressTypeId="11" AddressTypeName="pont" AddressTypeExternalCode="12" />
     <AddressType AddressTypeIdx="13" AddressTypeId="12" AddressTypeName="promenade" AddressTypeExternalCode="13" />
     <AddressType AddressTypeIdx="14" AddressTypeId="13" AddressTypeName="quai" AddressTypeExternalCode="14" />
     <AddressType AddressTypeIdx="15" AddressTypeId="14" AddressTypeName="résidence" AddressTypeExternalCode="15" />
     <AddressType AddressTypeIdx="16" AddressTypeId="15" AddressTypeName="rue" AddressTypeExternalCode="16" /
     <AddressType AddressTypeIdx="17" AddressTypeId="16" AddressTypeName="ruelle" AddressTypeExternalCode="17" />
     <AddressType AddressTypeIdx="18" AddressTypeId="17" AddressTypeName="route" AddressTypeExternalCode="18" />
     <AddressType AddressTypeIdx="19" AddressTypeId="18" AddressTypeName="sente" AddressTypeExternalCode="19" />
     <AddressType AddressTypeIdx="20" AddressTypeId="19" AddressTypeName="sentier" AddressTypeExternalCode="20" />
     <AddressType AddressTypeIdx="21" AddressTypeId="20" AddressTypeName="square" AddressTypeExternalCode="21" />
     <AddressType AddressTypeIdx="22" AddressTypeId="21" AddressTypeName="villa" AddressTypeExternalCode="22" />
     <AddressType AddressTypeIdx="23" AddressTypeId="22" AddressTypeName="voie communale" AddressTypeExternalCode="23" />
   </AddressTvpeList>
 </ActionAddressTypeList>
```

6.3 - ContributorList

6.3.1 - Appel

« /API?Action=ContributorList »

6.3.2 - Description

Ce flux contient la liste des contributeurs, ainsi que la liste des services attribués à chacun d'eux.

6.3.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
ContributorExternal Code	String			Ne renvoie que le contributeur dont le code externe est passé en paramètre	Non
ServiceID	Integer			Ne renvoie que les contributeurs ayant le service dont l'identifiant est passé en paramètre	Non

6.3.4 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ActionContributorList>
 - <Params Function="Navitia_Debug.DEBUG_VKAT/api">
            <action>contributorlist</action
        </Params>
 - <ContributorList ContributorCount="3">

    -<contributor ContributorIdx="0" ContributorId="-1" ContributorName="" ContributorExternalCode="" ContributorType="0" ContributorTypeCaption="Default">
    -<contributor ContributorIdx="1" ContributorId="0" ContributorName="Oceido" ContributorExternalCode="OCE" ContributorType="2" ContributorTypeCaption="OBiTi">
    -<contributor ContributorIdx="1" ContributorId="0" ContributorName="Oceido" ContributorExternalCode="OCE" ContributorType="2" ContributorTypeCaption="OBiTi">
    -<contributor ContributorId="0" ContributorId="0" ContributorName="Oceido" ContributorExternalCode="OCE" ContributorType="2" ContributorType="0" ContributorType=
             - <ServiceList ServiceCount="1">
                   - <Service ServiceIdx="0" ServiceId="0" ServiceFileName="" ServiceType="0">
                              <StartDate />
                                <EndDate /
                   </Service>
              </Contributor>
      </ContributorList>
 </ActionContributorList>
```

6.4 - DirectStopAreaList

6.4.1 - Appel

« /API?Action=DirectStopAreaList »

6.4.2 - Exemple d'appel

Sur un site fictif,

http://www.navitia.com/cgibin/navitia.ashx/api?Action=DirectStopAreaList&StopAreaIdx=5

6.4.3 - Description

Ce flux contient la liste des arrêts commerciaux accessibles directement (sans correspondances) depuis un arrêt commercial.

Ce flux est basé sur la liste des arrêts par ligne commerciale.

6.4.4 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
StopAreaExt ernalCode	String			Ne renvoie que les arrêts commerciaux accessibles directement depuis l'arrêt commercial dont le code externe est passé en paramètre	Oui
ModeTypeEx ternalCode	String			Ne renvoie que les arrêts commerciaux sur des lignes ayant le type de mode dont le code externe est passé en paramètre	Non

ModeExterna ICode	String		Ne renvoie que les arrêts Non commerciaux sur des lignes ayant le type de mode dont le code externe du mode est passé en paramètre
Sens	1 / -1	1	sens=1 correspond à "partir après" Non

6.4.5 - Exemple

<?xml version="1.0" encoding="ISO-8859-1" ?>

```
<ActionDirectStopAreaList>
- <Params Function="APi":
     <action>DirectStopAreaList</action>
     <StopAreaIdx>55</StopAreaIdx>
   </Params>
 - <DirectStopAreaList StopAreaCount="33">
   - <StopArea StopAreaIdx="45" StopAreaId="2458" StopAreaName="CENTRE SOCIAL" StopAreaExternalCode="CENTRE SOCIAL" MainStopArea="0" MultiModal="0"
     CarPark="0" MainConnection="0" AdditionalData="">
- <City CityIdx="5" CityId="5" CityName="Achères" CityExternalCode="78005" CityCode="78260">
           <Country CountryIdx="1" CountryId="0" CountryName="France" CountryExternalCode="FRA" />
        </City>
      - <Coord>
          <CoordX>579945.50</CoordX>
            <CoordY>2440230,50</CoordY>
        </Coord>
      - <HangList>
          <Hang StopPointIdx="60" Duration="0" ConnectionKind="13" />
           <Hang StopPointIdx="61" Duration="0" ConnectionKind="13" />
         </HangList>
      </StopArea>
   - <StopArea StopAreaIdx="46" StopAreaId="3044" StopAreaName="CHENE FEUILLU" StopAreaExternalCode="CHENE FEUILLU" MainStopArea="0" MultiModal="0"
        CarPark="0" MainConnection="0" AdditionalData=
        <City CityIdx="5" CityId="5" CityName="Achères" CityExternalCode="78005" CityCode="78260">
          <Country CountryIdx="1" CountryId="0" CountryName="France" CountryExternalCode="FRA" />
        </City>
      - <Coord>
           <CoordX>579935,56</CoordX>
           <CoordY>2439341,50</CoordY>
         </Coord>

    <HangList>

          <Hang StopPointIdx="62" Duration="0" ConnectionKind="13" />
           <Hang StopPointIdx="63" Duration="0" ConnectionKind="13" />
         </HangList>
      </StonArea>
   + <StopArea StopAreaIdx="48" StopAreaId="4002" StopAreaName="CONSTITUANTE" StopAreaExternalCode="CONSTITUANTE" MainStopArea="0" MultiModal="0" CarPark="0"
   MainConnection="0" AdditionalData="">
+ <StopArea StopArea StopAreaIdx="49" StopAreaId="4306" StopAreaName="CROIX ROMPUE" StopAreaExternalCode="CROIX ROMPUE" MainStopArea="0" MultiModal="0" CarPark="0"
        MainConnection="0" AdditionalData="">
   + <StopArea StopAreaidx="50" StopAreaid="5930" StopAreaName="FOYER DES JEUNES" StopAreaExternalCode="FOYER DES JEUNES" MainStopArea="0" MultiModal="0"
   CarPark="0" MainConnection="0" AdditionalData=""> + <StopArea StopAreaId="51" StopAreaId="6213" StopAreaName="GARE D'ACHERES VILLE" StopAreaExternalCode="8738165" MainStopArea="1" MultiModal="0" MultiM
        CarPark="0" MainConnection="0" AdditionalData
   + <StopArea StopAreaidx="52" StopAreaid="6957" StopAreaName="GEORGES BOURGOIN" StopAreaExternalCode="GEORGES BOURGOIN" MainStopArea="0" MultiModal="0"
        CarPark="0" MainConnection="0" AdditionalData
   + <StopArea StopAreaIdx="53" StopAreaId="7184" StopAreaName="Grange St Louis" StopAreaExternalCode="Grange St Louis" MainStopArea="0" MultiModal="0
        CarPark="0" MainConnection="0" AdditionalData
     + <StopArea StopAreaidx="77" StopAreaId="14777" StopAreaName="QUATORZE JUILLET" StopAreaExternalCode="QUATORZE JUILLET" MainStopArea="0" MultiModal="0"
        CarPark="0" MainConnection="0" AdditionalData
   + <StopArea StopAreaIdx="78" StopAreaId="15167" StopAreaName="ROCOURT" StopAreaExternalCode="ROCOURT" MainStopArea="0" MultiModal="0" CarPark="0"
        MainConnection="0" AdditionalData="
   + <StopArea StopAreaIdx="79" StopAreaId="16018" StopAreaName="Rue du 8 Mai 1945" StopAreaExternalCode="Rue du 8 Mai 1945" MainStopArea="0" MultiModal="0"
        CarPark="0" MainConnection="0" AdditionalData:
    <StopArea StopAreaIdx="80" StopAreaId="16939" StopAreaName="Stade Albert Leraille" StopAreaExternalCode="Stade Albert Leraill" MainStopArea="0" MultiModal="0" CarPark="0" MainConnection="0" AdditionalData="">
   + <StopArea StopAreaIdx="81" StopAreaId="17284" StopAreaName="TOUSSAIN LOUVERTURE" StopAreaExternalCode="TOUSSAIN LOUVERTURE" MainStopArea="0"
        MultiModal="0" CarPark="0" MainConnection="0" AdditionalData=""5
</DirectStopAreaList>
</ActionDirectStopAreaList>
```

6.5 - EndOfCourse

6.5.1 - Appel

« /API?Action=EndOfCourse»

6.5.2 - Description

Ce flux contient la liste des horaires desservis à la suite de la montée dans une desserte.

6.5.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
VehicleJourn eyExternalCo de	String			Code externe de la desserte	Oui
Order	Integer			Position de l'horaire de montée dans la desserte	Oui
EndOrder	Integer			Position de l'horaire de fin	Non

Remarque

- A partir de l'interface 1.8.2 les horaires de montée et de descente sont présentés. L'application média peut néanmoins simplifier la présentation en affichant par exemple :
 - L'horaire de montée sur le premier arrêt
 - Les horaires de descente sur tous les autres arrêts.
- Le paramètre « VehicleJourneyExternalCode » s'utilise de façon conjointe avec le paramètre « Order ». Ce paramètre « order » représente le numéro d'ordre de l'horaire dans la desserte. On peut le récupérer sur les objets <Stop> issus de NAViTiA. Par exemple sur une réponse à l'API « prochain départ » :

3333333 Liste des API sur données théoriques

```
- <ActionNextDepartureList>
  + < Params Function="api"></Params>
  - <NextDepartureList StopCount="5">
    -- «Stop StopIdx="301157" StopPointIdx="11376" VehicleJourneyIdx="37642" HourNumber="0" <u>Min</u>uteNumber="0"
     DestinationPos="-1" ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="0" YehicleIdx="0">
      - <StopTime>
          <TotalSeconds>49980</TotalSeconds>
          <Day>0</Day>
          <Hour>13</Hour>
          <Minute>53</Minute>
        </StopTime>
        <StopArrivalTime/>
      + < StopPoint StopPointIdx="11376" StopPointId="5237" StopPointName="gare de Toulouse-Matabiau"
        StopPointExternalCode="OCETrain TER-87611004" FareZone="0"></StopPoint>
      + < VehicleJourney VehicleJourneyIdx="37642" VehicleJourneyId="50787" VehicleJourneyName="871834"
        VehicleJourneyRouteIdx="7463" VehicleJourneyExternalCode="OCESN871834F01009" IsAdapted="0"></VehicleJourney>
    + <Stop StopIdx="303937" StopPointIdx="11382" VehicleJourneyIdx="52546" HourNumber="0" MinuteNumber="0"
     DestinationPos="-1" ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="0" VehicleIdx="0"></Stop>
    + <Stop StopIdx="310760" StopPointIdx="11376" VehicleJourneyIdx="37408" HourNumber="0" MinuteNumber="0"
     DestinationPos="-1" ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="0" VehicleIdx="0"></Stop>
    + <Stop StopIdx="316081" StopPointIdx="11380" VehicleJourneyIdx="53663" HourNumber="0" MinuteNumber="0"
     DestinationPos="-1" ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="0" VehicleIdx="0"></Stop>
    +<Stop StopIdx="317054" StopPointIdx="11376" VehicleJourneyIdx="37155" HourNumber="0" MinuteNumber="0"
     DestinationPos="-1" ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="0" VehicleIdx="0"></Stop>
    </NextDepartureList>
    <ImpactList ImpactCount="0"> </ImpactList>
 </ActionNextDepartureList>
```

6.5.4 - Exemple

```
- <EndOfCourseList>
       <Params Function="API">
            <Action>endofcourse</Action>
            <stopidx>213340</stopidx>
        </Params>
     - <StopList StopCount="12">
            <Stop StopIdx="213340" StopPointIdx="3464" VehicleJourneyIdx="434" Hour:Number="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1"</p>
            ValidityPatternSetCommentPos="0" StopOrder="1" VehicleIdx="0">
             - <StopTime>
                    <TotalSeconds>21660</TotalSeconds>
                     <Day>0</Day>
                     <Hour>06</Hour>
                    <Minute>01</Minute>
                </StopTime>
             - <StopPoint StopPointIdx="3464" StopPointId="168761" StopPointName="Paul Vaillant Couturier" StopPointExternalCode="50|5621" FareZone="5">
                     StopPointAddress StopPointAddressName=" " StopPointAddressNumber="" StopPointAddressTypeName=""/>
                      <Equipment Sheltered="False" MIPAccess="False" Elevator="False" Escalator="False" BikeAccepted="False" BikeDepot="False"/>
                    <Mode ModeIdx="3" ModeId="2" ModeName="BUS" ModeExternalCode="RATP"/>

<City CityIdx="120" CityId="125" CityName="Bois-d'Arcy" CityExternalCode="78073" CityCode="78390"> </City>
                 - <Coord>
                         <CoordX>576832,53</CoordX>
                         <CoordY>2422429,01</CoordY>
                    </Coord>
                </StopPoint>
                <StopArea StopAreaIdx="1470" StopAreaId="13231" StopAreaName="Paul Vaillant Couturier" StopAreaExternalCode="Paul Vaillant Coutur" MainStopArea="0" MultiModal="0"</p>
                CarPark="0" MainConnection="0" AdditionalData=
                  - <City CityIdx="120" CityId="125" CityName="Bois-d'Arcy" CityExternalCode="78073" CityCode="78390">
                         < Country Country Idx = "1" \ Country Id = "0" \ Country Name = "France" \ Country External Code = "FRA"/> \ Country Country
                    </City>
                 - <Coord>
                         <CoordX>576829,78</CoordX>
                         <CoordY>2422437,28</CoordY>
                    </Coord>
                </StopArea>
             </Stop>
        + <Stop StopIdx="216313" StopPointIdx="3455" VehicleJourneyIdx="434" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1"
       ValidityPatternSetCommentPos="0" StopOrder="2" VehicleIdx="0"></Stop>
+ <Stop StopIdx="256625" StopPointIdx="35508" VehicleJourneyIdx="434" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1"
           ValidityPatternSetCommentPos="0" StopOrder="7" VehicleIdx="0"></Stop>
       + <Stop StopIdx="264006" StopPointIdx="42460" VehicleJourneyIdx="434" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="8" VehicleIdx="0"></Stop> + <Stop StopIdx="267708" StopPointIdx="42601" VehicleJourneyIdx="434" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1"
           ValidityPatternSetCommentPos="0" StopOrder="9" VehicleIdx="0"></Stop>
        + <Stop StopIdx="271451" StopPointIdx="42476" VehicleJourneyIdx="434" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1"
       ValidityPatternSetCommentPos="0" StopOrder="10" VehicleIdx="0"></Stop>
+ <Stop StopIdx="275298" StopPointIdx="42637" VehicleJourneyIdx="434" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1"
           ValidityPatternSetCommentPos="0" StopOrder="11" VehicleIdx="0"></Stop>
        + <Stop StopIdx="279156" StopPointIdx="42429" VehicleJourneyIdx="434" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1"
           ValidityPatternSetCommentPos="0" StopOrder="12" VehicleIdx="0"></Stop>
       </StopList>
    · <VehicleJourneyVehicleJourneyIdx="434" VehicleJourneyId="11319655" VehicleJourneyName="11319655" VehicleJourneyRouteIdx="95" VehicleJourneyExternalCode="11319655" VehicleJourneyRouteIdx="95" VehicleJourneyExternalCode="11319655" VehicleJourneyRouteIdx="95" VehicleJourneyRouteIdx=
        - <Route RouteIdx="95" RouteId="588208" RouteName="011 - Bois d'arcy - St Cyr l'école - Versailles" RouteExternalCode="0270270110001" IsForward="1" RouteLineIdx="10"
           IsFrequence=
              · <Line LineIdx="10" LineId="307540" LineName="011 - Bois d'arcy - St Cyr l'Ecole - Versailles" LineCode="011 - Bois" LineExternalCode="027027011" LineAdditionalData=""
               SortOrder="10"
                    <Mode ModeIdx="3" ModeId="2" ModeName="BUS" ModeExternalCode="RATP"/>
                    <Network NetworkIdx="56" NetworkId="4609" NetworkName="HOURTOULE" NetworkExternalCode="027"> </Network>
                   <Forward ForwardName="011 - Bois d'arcy - St Cyr l'école - Versailles "> </Forward>
                   <Backward BackwardName="011 - Bois d'arcy - St Cyr l'école - Versailles "> </Backward>
               </Line>
           </Route>
            - <stopArea StopAreaIdx="17816" StopAreaId="6681" StopAreaName="GARE DE VERSAILLES RIVE GAUCHE" StopAreaExternalCode="8739315" MainStopArea="1"
             MultiModal="0" CarPark="0" MainConnection="0" AdditionalData="">
- <City CityIdx="1324" CityId="1325" CityName="Versailles" CityExternalCode="78646" CityCode="78000">
                       <Country CountryIdx="1" CountryId="0" CountryName="France" CountryExternalCode="FRA"/>
                   </City>
               - <Coord>
                       <CoordX>584687,65</CoordX>
                       <CoordY>2422468.22</CoordY>
                   </Coord>
               </StopArea>
           <Mode ModeIdx="3" ModeId="2" ModeName="BUS" ModeExternalCode="RATP"/>
           Company CompanyId="48" CompanyId="3057" CompanyName="CARS HOURTOULE" CompanyExternalCode="027"/>
           <Vehicle VehicleIdx="0" VehicleId="-1" VehicleName="" VehicleExternalCode="" MIPAccess="False" BikeAccepted="False" AirConditioned="False"/>
      </VehicleJourney
 </EndOfCourseList>
```

6.6 - EntryPoint

6.6.1 - Appel

« /API?Action=EntryPoint »

6.6.2 - Description

Ce flux contient la liste des points d'arrêts / lieux remarquables / adresses proches du nom passé en paramètre. Le rapprochement entre la saisie client et les données se base sur la combinaison de 2 algorithmes : phonétique et recherche par orthographe approchante.

6.6.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
Filter	All / StopArea;Address;Site			Permet de ne rechercher que sur certains types de points de demande	Non
Name	String			intitulé du point de demande désiré	Oui
CityName	String			intitulé de la ville désiré	Non
CityExternalCode	String			Code externe de la ville désiré	Non
CityPostalCode	Integer			Code postal de la ville désiré	Non
Number	String			Numéro de rue désiré (si type=Address)	Non
TypeName	String			Intitulé du type de voirie désiré (si type=Address ou Site)	Non
TypeDiscrimination	Boolean			Type de voirie / lieu remarquable discriminant pour les réponses	Non
ModeTypeExternalCode	All / ModeType1;ModeType 2;ModeType3		All	Filtre complémentaire sur les arrêts : liste des codes externes des types de modes desservis sur l'arrêt	Non
ModeExternalCode	All / Mode1;Mode2;Mode3		All	Filtre complémentaire sur les arrêts : liste des codes externes des modes dont les types de modes sont desservis sur l'arrêt	Non
NbMax	Integer			Nombre maximum de solution à renvoyer	Non

Nom	Туре	Format	Défaut	Description	Obligatoire
RawData	integer	0 / 1 / 2 / 3 / 4	1	0 : ne retourne que les meilleures réponses de même qualité (souvent sans liste de choix)	Non
				1 (par défaut et conseillé) : retourne les réponses dont la qualité est supérieure à la moyenne + l'écart type de toutes les solutions trouvées	
				2 : retourne les réponses dont la qualité est supérieure à la moyenne - l'écart type de toutes les solutions trouvées	
				3 : retourne les réponses dont la qualité est supérieure à la moyenne – 2 x l'écart type de toutes les solutions trouvées	
				4 : retourne toutes les solutions trouvées	
CityExclusive	Boolean		false	Indique si la commune est discriminante ou pas.	
QualityClass	Integer		50	Qualité des réponses phonétiques de référence	Non
UseVisible	Boolean		true	Limite la recherche aux points de demande qualifiés de « visibles dans les recherches lexicographiques »	Non

Note:

- « Filter » peut prendre les valeurs suivantes : « All », « StopArea », « Address » et « Site », ainsi qu'une combinaison de ces valeurs. Par exemple : « Address, Site ».
- Deux balises « Coord » sont présente par solution proposé : la première (« EntryPoint/Address/Coord ») correspond aux coordonnées théorique de la réponse (ex : barycentre d'une rue complète). La deuxième balise (« EntryPoint/Coord ») correspond aux coordonnées réelles demandées calculées par NAVITIA. C'est cette deuxième balise qu'il faut utiliser pour lancer un calcul d'itinéraire.
- Si le paramètre « CityPostalCode » est renseigné la recherche se limitera aux communes ayant ce code postal et le paramètre « CityName » ne sera pas pris en compte.
- QualityClass est le critère de qualité. Parmi la liste de solutions trouvées, comptabiliser il permet de dans « EntryPointList/@EntryPointInQualityCount » celles qui répondent à ce critère. Le média peut tirer parti de ce nombre afin de faire des traitements spécifiques. Ainsi dans le flux exemple, NAViTiA a trouvé deux réponses correspondant à la requête, le média peut cependant choisir de ne pas demander confirmation client. de au En effet l'attribut « EntryPointList/@EntryPointInQualityCount » nous indique une seule réponse répondant aux critères de qualité « qualityclass » demandé.
- ModeExternalCode : chaque arrêt commercial est desservi par une liste de mode. Ce filtre permet de ne renvoyer que les arrêts desservis par les modes passés en paramètre. Par exemple : «Action=EntryPoint&Name=St Lazare&City=Paris&Filter=StopArea&ModeExternalCode=Train» permet de ne rechercher que des arrêts commerciaux de type « gares ».
- CityExclusive : le paramètre peut prendre 2 valeurs ET il peut également ne pas être requêté :
 - Non fourni à NAVITIA (fonctionnement par défaut): NAVITIA fera sa recherche dans les communes fournies (dans « CityName », ou à défaut dans « Name ») et se restreindra à ces communes uniquement s'il existe des communes qui correspondent. Ce mode par défaut est conseillé dans le cadre de projets régionaux.
 - 0 : Même si une valeur de commune est reconnue (dans la demande Name ou CityName), NAViTiA recherchera dans l'ensemble de la base de données. Cette valeur est conseillée dans le cas d'un projet de type urbain sur lequel les internautes ne préciseront pas forcément le nom de la commune
 - 1 : Si « CityName » est renseigné, NAViTiA ne recherchera que parmi les communes qui correspondent à CityName, sinon NAViTiA ne recherchera que sur les communes qui correspondent à au moins un mot de la demande « Name ».

6.6.4 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ActionEntryPointList>
- <Params Function="Navitia_Debug.DEBUG_VKAT/API">
   <Action>EntryPoint</Action>
   <Name>St Lazare</Name>
   <City>Paris</City>
   <Filter>StopArea</Filter>
   <ModeExternalCode>Train</ModeExternalCode>
   <qualityclass>70</qualityclass>
   <number>5</number>
   <nbmax>500</nbmax>
  </Params>
- <EntryPointList EntryPointCount="1" EntryPointInQualityCount="0">
 - <EntryPoint EntryPointType="StopArea" EntryPointResponseQuality="11" CityName="Brest" Number="" TypeName="" EntryPointName="Gare SNCF">
   - <StopArea StopAreaIdx="108" StopAreaId="1168" StopAreaName="Gare SNCF" StopAreaExternalCode="1168" MainStopArea="0" MultiModal="0" CarPark="0"
     MainConnection="0" AdditionalData="">
- <City CityIdx="38" CityId="39" CityName="Brest" CityExternalCode="29019" CityCode=""
         <Country CountryIdx="1" CountryId="0" CountryName="France" CountryExternalCode="FRA" />
        </City>
       <Coord />
     - <ModeList ModeCount="3">
         <Mode ModeIdx="5" ModeId="21" ModeName="TER" ModeExternalCode="TER" />
         <Mode ModeIdx="6" ModeId="19" ModeName="TGV" ModeExternalCode="TGV"
         <Mode ModeIdx="7" ModeId="20" ModeName="Train" ModeExternalCode="Train" />
        </ModeList>
     </StopArea>
     <Coord />
     <HangList Odd="Yes" StartNb="0" EndNb="0">
       <Hang StopPointIdx="210" Duration="0" ConnectionKind="13" />
<Hang StopPointIdx="211" Duration="0" ConnectionKind="13" />
      </HangList>
    </EntryPoint>
  </EntryPointList>
</ActionEntryPointList>
```

6.7 - ExternalPhonetic

6.7.1 - Appel

« /API?Action=ExternalPhonetic »

6.7.2 - Description

Ce flux permet d'utiliser la fonction phonétique de NAViTiA sur des données externes.

Actuellement, le fonctionnement est limité. NAVITIA ne sait pas encore lire de fichier de données externe. Seul l'appel sur des données connues est donc possible. Cependant cette fonction permet d'ors et déjà de faire des recherches sur un sousensemble d'arrêts commerciaux passés en paramètre.

6.7.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
Filter	StopArea; ExternalFile			Permet de ne rechercher que sur certains types de données	Oui
List	Set of integer	n1;n2;n3;n4		Liste des identifiants parmi lesquels la recherche phonétique doit s'effectuer	Oui

Nom	Туре	Format	Défaut	Description	Obligatoire
Name	String			intitulé de la requête désiré	Oui
NbMax	Integer			Nombre maximum de solution à renvoyer	Non
QualityClass	Integer			Qualité des réponses phonétiques de référence	Non

Note:

- « Filter » peut prendre les valeurs suivantes : « StopArea », « ExternalFile ». Nota : Seul la recherche sur StopArea est implémentée actuellement.
- QualityClass est le critère de qualité. Parmi la liste de solutions trouvées, il permet de comptabiliser dans « EntryPointList/@EntryPointInQualityCount » celles qui répondent à ce critère. Le média peut tirer parti de ce nombre afin de faire des traitements spécifiques. Ainsi dans le flux exemple, NAViTiA a trouvé deux réponses correspondant à la requête, le média peut cependant choisir de ne pas demander de confirmation au client. En effet l'attribut « EntryPointList/@EntryPointInQualityCount » nous indique une réponse répondant aux critères de qualité « qualityclass » demandé.

6.7.4 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

    - <ActionExternalPhoneticList>

 - <params fonction="api">
     <action>entrypoint</action>
     <filter>all</filter
     t>12;23;36;52</list>
     <name>Meudon</name>
     <nbmax>500</nbmax>
     <qualityclass>85</qualityclass>
   </params>
 - <EntryPointList EntryPointCount="1" EntryPointInQualityCount="1">
    «EntryPoint EntryPointType="StopArea" EntryPointResponseQuality="100" CityName="Meudon" Number="" TypeName="" EntryPointName="Gare de Meudon">
     - <StopArea StopAreaIdx="12719" StopAreaId="310" StopAreaName="Gare de Meudon" StopAreaExternalCode="8739310" MainStopArea="1"
       - <City CityIdx="909" CityId="909" CityName="Meudon" CityExternalCode="92110" CityCode="75056"> <Country CountryIdx="0" CountryId="-1" CountryName="" CountryExternalCode="" />
         </City>
       < <Coord>
           <CoordX>599156,92</CoordX>
           <CoordY>2430791,60</CoordY>
         </Coord>
         <Comment CommentIdx="0" CommentId="-1" CommentName="" CommentExternalCode="" />
       </StopArea>
     - <Coord>
        <CoordX>602946,13</CoordX>
         <CoordY>2427062.96</CoordY>
       </Coord>
       <HangList Odd="Yes" StartNb="1" EndNb="257" />
     </EntryPoint>
   </EntryPointList:
 </ActionExternalPhoneticList>
```

DESCRIPTIONListe des API sur données théoriques

6.8 - FirstLetter

6.8.1 - Appel

« /API?Action=FirstLetter »

6.8.2 - Description

Ce flux contient la liste des points d'arrêts / lieux remarquables dont un des mots commence par le nom passé en paramètre.

Cette fonctionnalité correspond au besoin « autocompletion » de la majorité des intégrations. Elle fonctionne au mieux en limitant le nombre de paramètre et en laissant le tri des réponses tel que fourni par NAViTiA

6.8.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
Filter	All / StopArea;Si te;Address; City			Permet de ne rechercher que sur certains types de points de demande	Non
Name	String			intitulé du point de demande désiré.	Oui
				Cette propriété correspond à la saisie de l'internaute, en particulier dans la cas de « saisie en un seul champ » (1 champ départ et 1 champ arrivée, mais pas de champs commune)	
CityName	String			intitulé de la ville désiré. Ne pas remplir lorsqu'un seul champ de saisie est proposé à l'internaute	Non
CityPostalCode	Integer			Code postal de la ville désiré	Non
NbMax	Integer			Nombre maximum de solution à renvoyer	Non
QualityClass	Integer			Qualité des réponses phonétiques de référence	Non

Nom	Туре	Format	Défaut	Description	Obligatoire
TypeDiscriminati on	Boolean			Type de voirie / lieu remarquable discriminant pour les réponses	Non
ModeTypeExtern alCode	All / ModeType1 ;ModeType 2;ModeTyp e3		All	Filtre complémentaire sur les arrêts : liste des codes externes des types de modes desservis sur l'arrêt	Non
ModeExternalCo de	All / Mode1;Mod e2;Mode3		All	Filtre complémentaire sur les arrêts : liste des codes externes des modes dont les types de modes sont desservis sur l'arrêt	Non
CityExclusive	Boolean		false	Indique si la commune est discriminante ou pas.	Non

Note:

- Cette fonction est optimisée pour l'auto-completion en un seul champ : il est donc conseillé, dans le cadre d'un calcul d'itinéraire par exemple :
 - o De proposer un seul champ pour la saisie « origine »
 - o De proposer un seul champ pour la saisie « destination »
 - De définir NbMax=5 afin de ne pas présenter de liste de solution trop grande
 - De ne pas fournir d'autres paramètres que « Name » à NAViTiA pour chacune de ces saisies
 - De présenter la liste de résolution telle quelle, sans tri supplémentaire : le système NAViTiA sait prendre en compte l'importance des objets afin d'optimiser ce tri.
- « Filter » peut prendre les valeurs suivantes : « All », « StopArea » et « Site », ainsi qu'une combinaison de ces valeurs. Par exemple : « Address, Site ».
- Deux balises « Coord » sont présente par solution proposé : la première (« EntryPoint/Address/Coord ») correspond aux coordonnées théorique de la réponse (ex : barycentre d'une rue complète). La deuxième balise (« EntryPoint/Coord ») correspond aux coordonnées réelles demandées calculées par NAVITIA. C'est cette deuxième balise qu'il faut utiliser pour lancer un calcul d'itinéraire.

DESTRUIT DE LISTE DES API sur données théoriques

- QualityClass est le critère de qualité. Parmi la liste de solutions trouvées, de comptabiliser « EntryPointList/@EntryPointInQualityCount » celles qui répondent à ce critère. Le média peut tirer parti de ce nombre afin de faire des traitements spécifiques. Ainsi dans le flux exemple, NAViTiA a trouvé deux réponses correspondant à la requête, le média peut cependant choisir de ne pas au demander de confirmation client. En effet l'attribut indique « EntryPointList/@EntryPointInQualityCount » nous une seule réponse répondant aux critères de qualité « qualityclass » demandé.
- ModeExternalCode : chaque arrêt commercial est desservi par une liste de mode. Ce filtre permet de ne renvoyer que les arrêts desservis par les modes passés en paramètre. Par exemple : «Action=FirstLetter&Name=bellev&City=bres&Filter=StopArea&ModeExterna lCode=Train» permet de ne rechercher que des arrêts commerciaux de type « gares ».

6.8.4 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionFirstLetterList>
  - <Params Function="Navitia_Debug.DEBUG_VKAT/API">
      <Action>firstletter</Action:
      <filter>all</filter>
      <name>bellev</name>
      <cityname>bres</cityname>
      <nbmax>10</nbmax>
      <qualityclass>85</qualityclass>
  - <EntryPointList EntryPointCount="8" EntryPointInQualityCount="0">
    - <EntryPoint EntryPointType="StopArea" EntryPointResponseQuality="46" CityName="Redon" Number="" TypeName="" EntryPointName="Bellevue":
      - <StopArea StopAreaIdx="787" StopAreaId="1926" StopAreaName="Bellevue" StopAreaExternalCode="1926" MainStopArea="0" MultiModal="0" CarPark="0"
        MainConnection="0" AdditionalData="">
- <City CityIdx="254" CityId="76" CityName="Redon" CityExternalCode="35236" CityCode="":
             <Country CountryIdx="1" CountryId="0" CountryName="France" CountryExternalCode="FRA" />
          </City>
          <Coord />
         - <ModeList ModeCount="1">
            <Mode ModeIdx="3" ModeId="6" ModeName="car ILLENOO" ModeExternalCode="car ILLENOO" />
          </ModeList>
        </StopArea>
        <Coord />
      - <HangList Odd="Yes" StartNb="0" EndNb="0">
          <Hang StopPointIdx="1604" Duration="0" ConnectionKind="13" />
          <Hang StopPointIdx="1605" Duration="0" ConnectionKind="13" />
        </HangList>
      </EntryPoint:

<pre
    + <EntryPoint EntryPointType="StopArea" EntryPointResponseQuality="23" CityName="St Jacques de la Lande" Number="" TypeName="" EntryPointName="Bellevue">
    </EntryPointLists
  </ActionFirstLetterList>
```

6.9 - Isochron

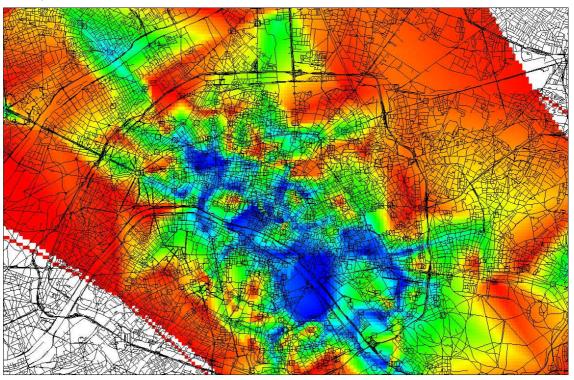
6.9.1 - Appel

« /API?Action=Isochron »

6.9.2 - Description

Ce flux contient les temps d'accès vers tous les points d'arrêts accessibles en transport en commun depuis un EntryPoint.

Cette fonctionnalité permet de faire l'analyse de l'accessibilité d'une zone. Par exemple :



6.9.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
Departure	String	TypePoint IdxPoint Name CityName Number TypeName X Y StopPointIdx ! Duration ! ConnectionKind; StopPointIdx ! Duration ! ConnectionKind; StopPointIdx ! Duration ! ConnectionKind;		Point de départ du trajet	Oui
Filter	City; MainStop Area; StopArea; Site			Filtre sur le type de données résultat : permet de ne rechercher que sur certains types de données	Oui
Sens	1 / -1		1	sens=1 correspond à "partir après"	Non
Time	String	Hour Minute		Heure de la recherche d'itinéraire	Oui
Date	String	Year Month Day		Date de la recherche d'itinéraire	Oui
VPatternSetIdx	integer		0	Index de la période de recherche d'itinéraire / si <= 0 alors calcul sur la date	Non
Criteria	1/2/3		1	critère de recherche de calcul (1: arrivée au plus tôt; 2: le moins de correspondance; 3: le moins de marche à pied)	Non
Vehicle	Integer		Tout type de véhicul e	Critère binaire de qualité de véhicule (à utiliser avec MakeBinaryCriteria)	Non

Nom	Туре	Format	Défaut	Description	Obligatoire
Equipment	Integer		Tout type d'équip ement	Critère binaire de qualité de point d'arrêt (à utiliser avec MakeBinaryCriteria)	Non
ModeType	Integer		Tout mode possible	Critère binaire de filtre sur les types de modes (à utiliser avec MakeBinaryCriteria)	Non
Mode	Integer	Conservé pour la compatibilité ascendante. Les valeurs échangées seront des types de mode	Tout mode possible	Critère binaire de filtre sur les modes (à utiliser avec MakeBinaryCriteria)	Non
WalkSpeed	Integer		50	Vitesse de marche à pied en m/min	Non
HangDistanceDep	Integer		1000	Distance maximale d'accroche en mètre sur le point de départ	Non
MaxDuration	Integer		0	Durée maximale des solutions à retourner	Non
MinDuration	Integer		0	Durée minimale des solutions à retourner	Non
NbMax	Integer		100	Nombre d'élément à afficher au maximum	Non
UseAdapted	Boolean		Oui	Utilisation des données adaptées pour le calcul	Non

Nom	Туре	Format	Défaut	Description	Obligatoire
ManageDisrupt	Boolean		Non	Indique si la recherche doit se faire en tenant compte des perturbations	Non
ViaExternalCode	Integer		Non	Code externe des arrêts que les parcours doivent emprunter	Non
ForbiddenLineExtCode	String			Liste de codes externes de lignes par lequel l'itinéraire NE DOIT PAS passer	Non

Remarques:

- Les temps de calcul de cette API peuvent être longs : ils sont directement liés au paramètre « MaxDuration ». Il convient donc de limiter au maximum ce paramètre lors de l'intégration de NAViTiA dans un média.
- Lorsque le paramètre « Filter » est vide, NAViTiA retourne la totalité des arrêts commerciaux accessibles en moins de MaxDuration. La taille du flux de réponse peut alors être considérable.
- Le point de départ est formaté de la même manière que pour l'API « PlanJourney ». Voir le chapitre 7.6.3 Descriptions des paramètres de départ et d'arrivée pour les détails.
- Si VpatternIdx a une valeur inférieure ou égale à 0 alors la recherche s'effectue sur la date. La date doit avoir le format « yyyy|mm|dd »
- Criteria : critère de recherche d'itinéraires. Les valeurs possibles sont :
 - 1 : arrivée au plus tôt
 - 2 : le moins de correspondances
 - 3 : le moins de temps de marche à pied
- Vehicle, Equipment, Mode : critères binaires a construire grâce à l'API
 « MakeBinaryCriteria »

6.9.4 - Résultat

Le flux de réponse contient une liste de données « isochron » composées de :

- La notion atteinte par le calculateur (StopArea ou City si « filter=city »)
- la durée de trajet en seconde pour rejoindre chaque notion (StopArea ou City si « filter=city ») dans l'attribut « IsochronDuration »
- le nombre de correspondance pour rejoindre chaque notion (StopArea ou City si « filter=city ») dans l'attribut « IsochronInterchange »
- L'heure d'arrivée sur la notion dans la balise « IsochronReachedTime »

6.9.5 - Exemple

3333333 Liste des API sur données théoriques

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionIsochronList>
 - <Params Function="API">
     <action>isochron</action>
     <Departure>StopArea | 12949 | NATION | Paris | | | 604393,28 | 2427808,02 | 
     <Time>12|00</Time>
     <Date>2005|9|13</Date>
     <HangDistance>2000</HangDistance>
     <Criteria>1</Criteria>
     <WalkSpeed>33</WalkSpeed>
     <Sens>1</Sens>
     <Mode>636</Mode>
     <MaxDuration>1200</MaxDuration>
   </Params>
 - <IsochronList IsochronCount="194">
   - <Isochron IsochronDuration="1200" IsochronInterchange="2"; IsAdapted="0">
     - <!sochronReachedTime>
        <TotalSeconds>44400</TotalSeconds>
        <Day>0</Day>
        <Hour>12</Hour>
        <Minute>20</Minute>

IsochronReachedTime>
     - <StopArea StopAreaIdx="12825" StopAreaId="416" StopAreaName="Lagny" StopAreaExternalCode="Lagny" MainStopArea="0">
        <City CityIdx="909" CityId="909" CityName="Paris" CityExternalCode="75056" CityCode="75056" />
          <CoordX>605464,00</CoordX>
          <CoordY>2427806,00</CoordY>
        </Coord>
     </Isochron>
   + <Isochron IsochronDuration="1142" IsochronInterchange="2" IsAdapted="0">
   + <Isochron IsochronDuration="985" IsochronInterchange="1" IsAdapted="0">
   </IsochronList>
 </ActionIsochronList>
```

6.10 - LineRouteDescription

6.10.1 - Appel

« /API?Action=LineRouteDescription »

6.10.2 - Description

Ce flux décrit les différents itinéraires possibles pour une ligne donnée. Il permet, par exemple d'afficher le trajet d'une ligne.

6.10.3 - Tableau des paramètres de l'API

Nom	Type	Format	Défaut	Description	Obligatoir e
LineExternalCode	String			Ne renvoie que les descriptions de ligne dont le code externe est passé en paramètre	Oui

6.11 - LineStopAreaList

6.11.1 - Appel

« /API?Action=LineStopAreaList»

6.11.2 - Description

Ce flux contient la liste des arrêts commerciaux situés avant ou après un arrêt commercial pour une ligne commerciale et un sens donné. Elle est utile afin de déterminer la liste des arrêts desservis depuis un arrêt donné avant une demande de grille horaire de ligne par exemple.

6.11.3 - Tableau des paramètres de l'API

Nom	Туре	Défaut	Description	Obligatoire
LineExternalCode	String		Code externe de la liste à observer	Oui
StopAreaExternalCode	String		Code externe de l'arrêt de référence	Non
Sens	1 / -1	0	sens=0 correspond à la liste des arrêts complète.	Non
			sens=1 correspond à la liste des arrêts sur les parcours en sens aller	
			sens=-1 correspond à la liste des arrêts sur les parcours en sens retour	
BeforeStopArea	1 / -1	1	Renvoie la liste des arrêts suivant (BeforeStopArea=1) ou précédent (BeforeStopArea=-1) l'arrêt de référence sur la ligne	Non
MainStopArea	1 / -1	-1	Indique si l'on ne renvoie que les arrêts remarquables	

Note : Si « StopAreaIdx » n'est pas renseigné, la fonction renverra l'intégralité des arrêts de la ligne.

6.11.4 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionLineStopAreaList>
 - <params fonction="api">
     <action>stoparealistbefore</action>
     lineidx>5</lineidx>
     <sens>1</sens>
     <arrivalidx>2</arrivalidx>
   </params>
 - <LineStopAreaList StopAreaCount="1">
   - <StopArea StopAreaIdx="1" StopAreaId="4081" StopAreaName="Aeroport Lyon Saint Exupery - Gare SNCF" StopAreaExternalCode="">
     - <City CityIdx="0" CityId="-1" CityName="" CityExternalCode="">
        <Country CountryIdx="0" CountryId="-1" CountryName="" CountryExternalCode="" />
     - <Coord>
        <CoordX>0,00</CoordX>
        <CoordY>0,00</CoordY>
       </Coord>
     </StopArea>
   </LineStopAreaList>
 </ActionLineStopAreaList>
```

6.12 - MakeBinaryCriteria

6.12.1 - Appel

« /API?Action=MakeBinaryCriteria »

6.12.2 - Description

Ce flux permet de fabriquer les paramètres d'appel avancés pour le lancement d'un calcul d'itinéraire.

6.12.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
VKMIPAccess	Boolean		false	Les véhicules utilisés doivent être dotés d'un accès PMR	Non
VKBikeAccepted	Boolean		false	Les véhicules utilisés doivent accepter le transport des bicyclettes	Non
VKAirConditioned	Boolean		false	Les véhicules utilisés doivent être dotés d'une climatisation	Non
VKVisualAnnouncemen t	Boolean		false	Les véhicules utilisés doivent être dotés de signal visuel	Non
VKAudibleAnnouncem ent	Boolean		False	Les véhicules utilisés doivent être dotés de signal sonore	Non
SPSheltered	Boolean		false	Les arrêts utilisés doivent être abrités	Non

Nom	Туре	Format	Défaut	Description	Obligatoire
SPMIPAccess	Boolean		false	Les arrêts utilisés doivent être dotés d'un accès PMR	Non
SPElevator	Boolean		false	Les arrêts utilisés doivent être dotés d'un ascenseur	Non
SPEscalator	Boolean		false	Les arrêts utilisés doivent être dotés d'un escalier mécanique	Non
SPBikeAccepted	Boolean		false	Les arrêts utilisés doivent accepter le passage des bicyclettes	Non
SPBikeDepot	Boolean		false	L'arrêt de départ doit accepter le stockage de bicyclettes	Non
SPVisualAnnouncemen t	Boolean		false	L'arrêt de départ doit être doté de signal visuel	Non
SPAudibleAnnounceme nt	Boolean		false	L'arrêt de départ doit être doté de signal sonore	Non
ModeTypeExternalCod e	string	All/ Mode1; Mode2; Mode3	All	Liste des codes externes des types de modes utilisables lors du trajet	Non
ModeExternalCode	string	All / Mode1 ; Mode2 ; Mode3	All	Liste des codes externes des modes dont les types de modes sont utilisables lors du trajet	Non

Notes : le paramètre « ModeExternalCode » et « ModeTypeExternalCode » contiennent les codes externes **des types de modes de transports** utilisables séparés par des « ; ».

Par exemple: « ModeTypeExternalCode=bus;metro;tramway ».

Si tous les modes doivent être disponibles, il suffit de passer « all » à l'un ou l'autre de ces paramètres.

Exemple d'appel:

http://SERVER/cgi-

 $\frac{bin/navitia.ashx/API?Action=MakeBinaryCriteria\&vkmipaccess=1\&spmipacc}{ess=1\&modeexternalcode=all}$

Pour des raisons de compatibilité, les interfaces 1_8_2 et inferieurs renverront toujours une balise « Mode » avec le contenu de la donnée type de mode, mais les interfaces de version > 1_9 renvoient une balise « ModeType »

6.13 - ProximityList

6.13.1 - Appel

« /API?Action=ProximityList »

6.13.2 - Description

Recherche de proximité géographique

6.13.3 - Tableau des paramètres de l'API

Nom	Type Format	Défaut	Description	Obligatoire
Туре	Site / StopPoint / StopArea / MainStop Area		Type de données sur lesquelles la recherche de proximité se fera (seulement la recherche sur « Site » ou « StopPoint » sont implémentés actuellement	Oui
X	réel		Longitude dans le système utilisé par la base de données NAViTiA	Oui

Y	réel		Latitude dans le système utilisé par la base de données NAViTiA	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
Filter	String		Filtre sur les Type de site (est un restaurant, est une mairie)	Non
MainStopArea	a Booléen	0	Type = StopArea Ne retourne que les StopArea de type MainStopArea	Non
MultiModal	Booléen	0	Type = StopArea ou MainStopArea Ne retourne que les StopArea de type MultiModal	
CarPark	Booléen	0	Type = StopArea ou MainStopArea Ne retourne que les StopArea ayant un Parking	

6.13.4 - Exemple

3333333 Liste des API sur données théoriques

```
- <ActionProximityList>
 - < ProximityList ProximityCount="3">
   - <StopPoint StopPointIdx="27945" StopPointId="736662" StopPointName="TOLBIAC-NATIONALE" StopPointExternalCode="59|78522">
      <StopPointAddress StopPointAddressName="" StopPointAddressNumber="" StopPointAddressTypeName="" />
      <City CityIdx="1046" CityId="30791" CityName="Paris" CityExternalCode="75056" />
     - <StopArea StopAreaIdx="12906" StopAreaId="871" StopAreaName="Tolbiac / Nationale" StopAreaExternalCode="Tolbiac / Nationale">
      - <Coord>
          <CoordX>602253,00</CoordX>
          <CoordY>2425417,00</CoordY>
         </Coord>
      </StopArea>
     - <Coord>
        <CoordX>602267,00</CoordX>
        <CoordY>2425437,00</CoordY>
       </Coord>
   - <StopPoint StopPointIdx="27944" StopPointId="736652" StopPointName="TOLBIAC-NATIONALE" StopPointExternalCode="59|79544">
      <StopPointAddress StopPointAddressName="" StopPointAddressNumber="" StopPointAddressTypeName="" />
      <City CityIdx="1046" CityId="30791" CityName="Paris" CityExternalCode="75056" />
     - <StopArea StopAreaIdx="12906" StopAreaId="871" StopAreaName="Tolbiac / Nationale" StopAreaExternalCode="Tolbiac / Nationale">
          <CoordX>602253,00</CoordX>
          <CoordY>2425417,00</CoordY>
         </Coord>
      </StopArea>
     - <Coord>
        <CoordX>602239.00</CoordX>
        <CoordY>2425397,00</CoordY>
       </Coord>
     </StopPoint>
   - <StopPoint StopPointIdx="27013" StopPointId="736653" StopPointName="PONSCARME" StopPointExternalCode="59|79545">
      <StopPointAddress StopPointAddressName="" StopPointAddressNumber="" StopPointAddressTypeName="" />
      <City CityIdx="1046" CityId="30791" CityName="Paris" CityExternalCode="75056" />
     - <StopArea StopAreaIdx="12676" StopAreaId="642" StopAreaName="PONSCARME" StopAreaExternalCode="PONSCARME">
      - <Coord>
          <CoordX>602313,00</CoordX>
          <CoordY>2425307,00</CoordY>
         </Coord>
      </StopArea>
     - <Coord>
        <CoordX>602364,00</CoordX>
        <CoordY>2425377,00</CoordY>
       </Coord>
     </StopPoint>
   </ProximityList>
 </ActionProximityList>
```

6.14 - PTReferential

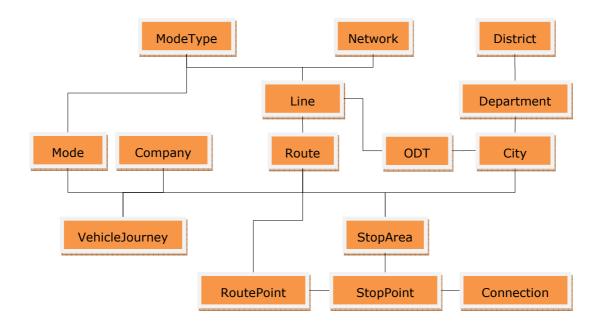
6.14.1 - Appel

« /API?Action=PTReferential »

6.14.2 - 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).

6.14.3 - Appel et passage de paramètres

6.14.3.1 Type de données à retourner

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

RequestedType peut prendre les valeurs suivantes :

- NetworkList
- ModeTypeList
- ModeList
- LineList
- RouteList
- VehicleJourneyList
- StopAreaList
- StopPointList
- RoutePointList
- CityList
- DepartmentList
- DistrictList
- CompanyList
- ConnectionList
- ODTList

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

6.14.3.2 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 :

RequestedType=LineList&StopareaExternalCode=SA1;SA2|and

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

RequestedType=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.

Pour une recherche combinant plusieurs filtres de types différents, il est également possible de choisir une combinaison inclusive ou exclusive.

Par exemple, pour obtenir la liste des lignes passant par 2 arrêts (SA1 ET SA2) ET dont le transporteur est T1 ou T2, la requête sera la suivante :

RequestedType=LineList&StopareaExternalCode=SA1;SA2|and&company ExternalCode=T1;T2|or&isinclusive=0

Note : La présence du filtre de RequestedType avec une valeur (unique) exclu tous les autres filtres présents dans le URL.

6.14.4 - Tris

Globalement, les listes sont triées dans un autre fonctionnellement pertinent, mais figé :

Récupération d'une liste de StopPoint : NAVITIA fourni cette liste triée par ordre de passage dans la ligne si un filtre par ligne est demandé.

Si les tris fournis ne conviennent pas à l'affichage souhaité, il est nécessaire d'appliquer un nouveau tri dans l'application qui intègre NAVITIA.

6.14.5 - Filtre complémentaires

Selon le requestedType choisi, on dispose de filtres additionnels :

- Line (RequestedType=LineList) : on peut également filtrer par Destination (DestinationExternalCode), IsAdapted...
- Route (RequestedType=RouteList) : CheckOrder, IsAdapted (valeur booléenne), Sens
- StopArea : MainStopArea (valeur booléenne)
- ...

6.14.6 - Tableau des paramètres de l'API

Nom	Туре	Format	Description	Obligatoire
RequestedType	String		Définit le type de données à renvoyer	Oui
IsInclusive	boolean	1/0 (0 par défaut)	Combiner les différents filtres de façon exclusive (0) ou inclusive (1)	Non
NetworkExternalCode	String, ou liste de string	Séparé par ; Liste finie par " and" ou " or" non obligatoire (" or" par défaut)	Renvoi les données appartenant au(x) réseau(x) passé(s) en paramètre	Non
ModeTypeExternalCode	String, ou liste de string	idem	Renvoi les données appartenant au(x) type(s) de mode passé(s) en paramètre	Non
ModeExternalCode	String, ou liste de string	idem	Renvoi les données appartenant au(x) mode(s) passé(s) en paramètre	Non
LineExternalCode	String, ou liste de string	idem	Renvoi les données appartenant au(x) ligne(s) passée(s) en paramètre	Non
RouteExternalCode	String, ou liste de string	idem	Renvoi les données appartenant au(x) route(s) passée(s) en paramètre	Non
VehicleJourneyExternalCode	String, ou liste de string	idem	Renvoi les données appartenant au(x) circulation(s) passée(s) en paramètre	Non
StopAreaExternalCode	String, ou liste de string	idem	Renvoi les données appartenant au(x) zone(s) d'arrêt passé(s) en paramètre	Non

Nom	Туре	Format	Description	Obligatoire
StopPointExternalCode	String, ou liste de string	idem	Renvoi les données appartenant au(x) point(s) d'arrêt passé(s) en paramètre	Non
RoutePointExternalCode	String, ou liste de string	idem	Renvoi les données appartenant au(x) point(s) de parcours passé(s) en paramètre	Non
CityExternalCode	String, ou liste de string	idem	Renvoi les données appartenant au(x) ville(s) passée(s) en paramètre	Non
DistrictExternalCode	String, ou liste de string	idem	Renvoi les données appartenant au(x) région(s) passée(s) en paramètre	Non
DepartmentExternalCode	String	idem	Renvoi les données appartenant au(x) département(s) passé(s) en paramètre	Non
CompanyExternalCode	String	idem	Renvoi les données appartenant au(x) transporteur(s) passé(s) en paramètre	Non
LineCode	String		Filtre par code commercial de ligne	Non
ODTExternalCode	String	Idem	Renvoi les données appartenant au(x) ODT(s) passé(s) en paramètre	Non
	Filtres	complémentaires : Liés	aux zones d'arrêt - StopArea	
MainStopArea	Boolean		RequestedType=StopAreaList Ne retourne que les StopArea de type MainStopArea	Non
MultiModal	Boolean		RequestedType=StopAreaList Ne retourne que les StopArea de type Multimodal	Non

Nom	Туре	Format	Description	Obligatoire
CarPark	Boolean		RequestedType=StopAreaList	Non
			Ne retourne que les StopArea ayant un Parking	
	Filtres	complémentaires : Liés	aux lignes commerciales - Line	
DestinationExternalCode	String		RequestedType=LineList	Non
			Filtre par destination, si un paramètre StopAreaExternalCode est déjà présent. Permet d'obtenir la liste des lignes passant entre 2 arrêts.	
IsAdapted	Boolean	1/0	RequestedType=LineList	Non
			Filtre par ligne appartenant au plan de transport adaptée	
	Fil	tres complémentaires :	Liés aux parcours - Route	
Sens	1 : aller		RequestedType=RouteList	Non
	-1 : retour		Combiné avec un filtre sur Line (LineExternalCode), permet d'obtenir la liste des parcours d'une ligne pour un sens donné	
CheckOrder	Boolean		RequestedType=RouteList	Non
			Actif seulement si plus d'une zone d'arrêt est passée en paramètre.	
			Ne renvoie que les routes passant par ces zones d'arrêts DANS L'ORDRE DANS LEQUELS ELLES SONT PASSES si valorisé à 1	

Nom	Туре	Format	Description	Obligatoire
IsAdapted	Boolean		RequestedType=RouteList	Non
			Filtre par parcours appartenant au plan de transport adapté	
	Filtres cor	nplémentaires : Liés au	x circulations - VehicleJourney	
ShowStop	Boolean		RequestedType=VehicleJourneyList	Non
			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.	
IsAdapted	Boolean	1/0	RequestedType=VehicleJourneyList	Non
			Ne retourne que les lignes ou parcours ayant des données adaptées	
VehicleJourneyExternalCode	String		RequestedType=VehicleJourneyList	Non
			Attention à la volumétrie des données si ce code n'est pas renseigné.	
VehicleJourneyName	String		RequestedType=VehicleJourneyList	Non
			Attention à la volumétrie des données si ce code n'est pas renseigné.	

Nom	Туре	Format	Description	Obligatoire
Date	String	YYYY MM DD	RequestedType=VehicleJourneyList	Non
			Filtre par date : propose les VehicleJourney qui circulent cette date uniquement.	
			Peut être utilisé conjointement avec le filtre suivant EndDate.	
EndDate	String	YYYY MM DD	RequestedType=VehicleJourneyList	Non
			Filtre par plage de date (le paramètre Date doit être présent) : propose les VehicleJourney qui circulent entre Date et EndDate.	
StartTime	String	НН ММ	RequestedType=VehicleJourneyList	Non
			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.	
EndTime	String	НН ММ	RequestedType=VehicleJourneyList	Non
			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.	

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

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

6.14.7 - Exemple d'appel



http://localhost:8081/Navitia_Debug.DEBUG_VKAT/api?action=PTReferential&RequestedType=linelist&StopAreaExternalCode=CHC153;CHC112;CHC156|and&RouteExternalCode=CHCNav5;CHCNav11;CHMNav1|or&isinclusive=false&CityExternalCode=51108&interface=112

VehicleJourneyList

http://localhost:8081/Navitia Debug.DEBUG_VKAT/api?action=PTReferential&RequestedType=vehiclejourneylist&ModeTypeExternalCode=Bus&LineExternalCode=CHCNav6&StopAreaExternalCode=CHC153&isinclusive=true&interface=1 12

6.14.8 - Résultats

NetworkList

LineList

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionLineList>
       - <Params Function="Navitia_Debug.DEBUG_VKAT/api">
<action>linelist</action>
              lineidx>10</lineidx>
        SortOrder="10" LineColor=" HasAdaptedRoute="0">
SortOrder="10" LineColor=" HasAdaptedRoute="0">
<Mode Modeldx="2" Modeld="20" ModeMame="Car TER" ModeExternalCode="OCECar TER"

*\text{Network NetworkIdx="2"} NetworkId="1" NetworkName="SNOF" NetworkExternalCode="OCI

- \text{Forward ForwardName="loudun gare sncf} - chinon gare sncf}

**The Color of the Col
                  - <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>
                 </
                       - <StonArea StonAreaIdx="0" StonAreaId="-1" StonAreaName="" StonAreaExternalCode="" MainStonArea="0" MultiModal="0" CarPark="0" MainConnection="0"
                           AdditionalData="">
- <City CityIdx="0" CityId="-1" CityName=" CityExternalCode="" CityCode="">
- <Country CountryIdx="0" CountryId="-1" CountryName="" CountryExternalCode=""/>
                                </City>
                        </Direction>
                  </Backward>
    </ActionLineList>
  RouteList
    <?xml version="1.0" encoding="ISO-8859-1" ?>
    - <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" IsFrequence="1" IsAdapted="0">
— <Line LineIdx="5" LineId="3228" LineName="Line 5" LineCode="5" LineExternalCode="RATLigne5" LineAdditionalData="" SortOrder="5" LineColor=""
```

>>

VehicleJourneyList

HasAdaptedRoute="0">

Remarques:

</Route>
</RouteList>
</ActionRouteList>

- Le volume de données en sortie peut être conséquent. Il convient donc de se rapprocher de Canal TP afin de valider que les filtres utilisés sont suffisamment discriminant pour éviter une surcharge réseau.
- 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

DESTRUIT DE LISTE DES API sur données théoriques

```
/*Xmi version= 1.0 encoding= 150-8659-1* />

<actionStopAreaList>
<action> StopAreaList</action>

<action> StopAreaList</action>

                <CityIdx>122</CityIdx>
           </Params>
        <StopAreaList StopAreaCount="6">
          <CoordX>288376,50</CoordX
                          <CoordY>2355386,00</CoordY>
                   </Coord>

</Coord>

                     </ModeList>
       </ModeList>

<
        MainConnection="0" AdditionalData=""> + <StopAreaId="148" StopAreaId="148" StopAreaExternalCode="148" MainStopArea="0" MultiModal="0" CarPark="0" CarP
        MainConnection="0" AdditionalData=""> + <StopAreaId="49" StopAreaId="149" StopAreaId="0" CarPark="0" MainConnection="0" MultiModal="0" CarPark="0" MainConnection="0" MultiModal="0" CarPark="0" MainConnection="0" MainConnec
                     AdditionalData="">
          </StopAreaList:
   </ActionStopAreaList>
                        StopPointList
- <ActionStopPointList>
       - <Params Function="Navitia_Debug.DEBUG_VKAT/api">
                     <action>stoppointlist</action>
       - <StopPointList StopPointCount="4407">
            - <StopPointCust StopPointCustre 440 / >
- <StopPoint StopPointIds="0" StopPointId="-1" StopPointName="" StopPointExternalCode="" FareZone="0" >
- <StopPointAddress StopPointAddressName="" StopPointAddressNumber="" StopPointAddressTypeName="" / >
- <StopPointAddress StopPointAddressName="" StopPointAddressTypeName="" / >
- <StopPointAddress StopPointAddressNumber="" StopPointAddressTypeName="" / >
- <StopPointAddress StopPointAddressNumber=" StopPointAddressTypeName="" / >
- <StopPointAddress StopPointAddressNumber="" StopPointAddressTypeName="" / >
- <StopPointAddress StopPointAddressTypeName=" / >
- <StopPointAddress StopPointAddress StopPointAddressTypeName=" / >
- <StopPointAddress StopPointAddress StopPointAddress StopPointAddress StopPointAddress StopPointAddress StopPointAddress StopPointAddress StopPointAddress StopPointAddress StopPo
                              <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:
               <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>
```

CityList

DESTRUIT DE LISTE DES API sur données théoriques

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionCityList>
 - <Params Function="Navitia_Debug.DEBUG_VKAT/api">
     <action>citylist</action>
     <cityidx>12</cityidx>
 - <CityList CityCount="1">
   - <City CityIdx="12" CityId="23695" CityName="Abbecourt" CityExternalCode="60002" CityCode="60430">
     - <Coord>
        <CoordX>586794.04</CoordX>
        <CoordY>2484516,38</CoordY>
       </Coord>
       <Department DepartmentIdx="0" DepartmentId="-1" DepartmentName="" DepartmentExternalCode="" DepartmentCode="" />
       <District DistrictIdx="0" DistrictId="-1" DistrictName="" DistrictExternalCode="" DistrictCode="" />
       <Country CountryIdx="1" CountryId="0" CountryName="France" CountryExternalCode="FRA" />
     </City>
   </CityLists
 </ActionCityList>
```

DepartmentList

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionDepartmentList>
 - <Params Function="Navitia_Debug.DEBUG_VKAT/api">
     <action>departmentlist</action
    </Params>
 - <DepartmentList DepartmentCount="4">
   - <Department DepartmentIdx="0" DepartmentId="-1" DepartmentName="" DepartmentExternalCode="" DepartmentCode=""
        <District DistrictIdx="0" DistrictId="-1" DistrictName="" DistrictExternalCode="" DistrictCode="" />
      </Department>
    - <Department DepartmentIdx="1" DepartmentId="0" DepartmentName="Ain" DepartmentExternalCode="01" DepartmentCode="">
       <District DistrictIdx="1" DistrictId="0" DistrictName="Pays-de-la-Loire" DistrictExternalCode="05" DistrictCode="" /</pre>
      </Department>
    - <Department DepartmentIdx="2" DepartmentId="1" DepartmentName="Aisne" DepartmentExternalCode="02" DepartmentCode="">
       <District DistrictIdx="1" DistrictId="0" DistrictName="Pays-de-la-Loire" DistrictExternalCode="05" DistrictCode="" ,</pre>
      </Department>
   - <Department DepartmentIdx="3" DepartmentId="2" DepartmentName="Loire-Atlantique" DepartmentExternalCode="44" DepartmentCode="">
        <District DistrictIdx="1" DistrictId="0" DistrictName="Pays-de-la-Loire" DistrictExternalCode="05" DistrictCode="" />
      </Department>
    </DepartmentLists
  </ActionDepartmentList>
```

DistrictList

6.15 - RoutePointList

6.15.1 - Appel

« /API?Action=RoutePointList »

6.15.2 - Description

Ce flux contient la liste des arrêts desservis d'une route, pour une ligne et un sens donnée.

6.15.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
RoutePointExternalCod e	String			Ne renvoie que le RoutePoint dont le code externe est passé en paramètre	Non
LineExternalCode	String			Code Externe de la ligne à observer. Obligatoire si pas de	Oui
				RoutePoint renseigné	
Sens	1 / -1		1	sens=1 correspond l'ensemble des parcours rattachés à la "ligne commerciale aller"	Non

6.15.4 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionRoutePointList>
 - <Params Function="api">
     <action>routepointlist</action>
     <routepointExternalCode>TCL54800|TCL2154</routepointExternalCode>
     <interface>1_10</interface>
   </Params>
 - <RoutePointList RoutePointCount="1">
   - <RoutePoint RoutePointExternalCode="TCL54800|TCL2154" RoutePointIdx="10" RoutePointId="0" RouteIdx="1"
      MainStopPoint="1" FareSection="0">
     - <StopPoint StopPointIdx="3144" StopPointId="1786" StopPointName="Oullins Mairie" StopPointExternalCode="TCL2154"
        FareZone="0">
         <StopPointAddress StopPointAddressName="GRANDE" StopPointAddressNumber="132"
          StopPointAddressTypeName="RUE" />
         <Equipment Sheltered="False" MIPAccess="False" Elevator="False" Escalator="False" BikeAccepted="False"
          BikeDepot="False" />
         <Mode ModeIdx="1" ModeId="1" ModeName="Bus" ModeExternalCode="Bus" ModeTypeExternalCode="Bus" />
         <City CityIdx="50" CityId="29" CityName="Oullins" CityExternalCode="69149" CityCode="69600" />
       - «StopArea StopAreaIdx="1391" StopAreaId="757" StopAreaName="Oullins Mairie" StopAreaExternalCode="TCL11035"
          MainStopArea="0" MultiModal="0" CarPark="0" MainConnection="0" AdditionalData="" ResaRailCode="">
            <CoordX>792314,40</CoordX>
            <CoordY>2082349,20</CoordY>
           </Coord>
         </StopArea>
       - <Coord>
          <CoordX>792294,00</CoordX>
          <CoordY>2082342,00</CoordY>
         </Coord>
       </StopPoint>
     </RoutePoint>
   </RoutePointList>
   <PagerInfo ResponseCount="1" ResponseStartIndex="0" TotalCount="1" />
 </ActionRoutePointList>
```

6.16 - SiteList

6.16.1 - Appel

« /API?Action=SiteList »

6.16.2 - Description

Ce flux contient la liste des lieux remarquables disponibles.

6.16.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
CityExternalCode	String			Ne renvoie que les lieux appartenant a la commune dont le code externe est passé en paramètre	Non
SiteExternalCode	String			Ne renvoie que le lieu dont le code externe est passé en paramètre	Non
SiteTypeExternalCo de	String			Ne renvoie que les lieux dont le type est passé en paramètre	Non
StopAreaExternalCo de	String			Renvoi tous les lieux remarquables accrochés à au moins un arrêt physique en commun avec l'arrêt commercial passé en paramètre	Non

6.16.4 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionSiteList>
 - <Params Function="API">
     <action>sitelist</action>
     <siteidx>1</siteidx>
   </Params>
 - <SiteList SiteCount="1">
   - <Site SiteIdx="1" SiteId="11722" SiteName="Beaumont-du-Gâtinais" SiteExternalCode="11722">
       <City CityIdx="0" CityId="-1" CityName="" CityExternalCode="" CityCode="" />
       <SiteType SiteTypeIdx="11" SiteTypeId="10" SiteTypeName="Lieu de culte" SiteTypeExternalCode="7339||" />
         <CoordX>610301,68</CoordX>
         <CoordY>2348550,01</CoordY>
       </Coord>
       <SiteAddress SiteAddressName="Place de l'Eglise" SiteAddressNumber="" SiteAddressTypeName="" />
     </Site>
   </SiteList>
 </ActionSiteList>
```

6.17 - SiteTypeList

6.17.1 - Appel

« /API?Action=SiteTypeList »

6.17.2 - Description

Ce flux contient la liste des types de lieux remarquables disponibles.

6.17.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
SiteTypeExternal Code	String			Ne renvoie que le type de site dont le code externe est passé en paramètre	Non

6.17.4 - Exemple

Liste des API sur données théoriques

```
<?xml version="1.0" encodina="ISO-8859-1" ?>
- <ActionSiteTypeList>
 - <Params Function="API">
     <action>sitetypelist</action>
   </Params>
 - <SiteTypeList SiteTypeCount="20">
     <SiteType SiteTypeIdx="0" SiteTypeId="-1" SiteTypeName="" SiteTypeExternalCode="" />
     <SiteType SiteTypeIdx="1" SiteTypeId="0" SiteTypeName="Office du tourisme/Syndicat d'initiative" SiteTypeExternalCode="7316||" />
     <SiteTypeSiteTypeIdx="2" SiteTypeId="1" SiteTypeName="Musée" SiteTypeExternalCode="7317||" /:
     <SiteType SiteTypeIdx="3" SiteTypeId="2" SiteTypeName="Théatre" SiteTypeExternalCode="7318|| "/>
     <SiteType SiteTypeIdx="4" SiteTypeId="3" SiteTypeName="Centre culturel" SiteTypeExternalCode="7319||" />
     <SiteType SiteTypeIdx="5" SiteTypeId="4" SiteTypeName="Centre sportif" SiteTypeExternalCode="7320||" />
     <SiteType SiteTypeIdx="6" SiteTypeId="5" SiteTypeName="Hôpital/Polyclinique" SiteTypeExternalCode="7321||"/>
     <SiteType SiteTypeIdx="7" SiteTypeId="6" SiteTypeName="Commisariat de police/Gendarmerie" SiteTypeExternalCode="7322||" />
     <SiteType SiteTypeIdx="8" SiteTypeId="7" SiteTypeName="bureau de poste" SiteTypeExternalCode="7324||" />
     <SiteType SiteTypeIdx="9" SiteTypeId="8" SiteTypeName="Pharmacie" SiteTypeExternalCode="7326||" />
     <SiteType SiteTypeIdx="10" SiteTypeId="9" SiteTypeName="Piscine" SiteTypeExternalCode="7338||" /
     <SiteType SiteTypeIdx="11" SiteTypeId="10" SiteTypeName="Lieu de culte" SiteTypeExternalCode="7339||" />
     <SiteType SiteTypeIdx="12" SiteTypeId="11" SiteTypeName="Batiment officiel" SiteTypeExternalCode="7367||" />
     <SiteType SiteTypeIdx="13" SiteTypeId="12" SiteTypeName="Préfecture/Conseil Général" SiteTypeExternalCode="7367|7|9G" />
     <SiteType SiteTypeIdx="14" SiteTypeId="13" SiteTypeName="Mairie" SiteTypeExternalCode="7367|8|9G" />
<SiteType SiteTypeIdx="15" SiteTypeId="14" SiteTypeName="Ecole" SiteTypeExternalCode="7372||" />
     <SiteType SiteTypeIdx="16" SiteTypeId="15" SiteTypeName="Stade" SiteTypeExternalCode="7374||" />
     <SiteType SiteTypeIdx="17" SiteTypeId="16" SiteTypeName="lieu touristique" SiteTypeExternalCode="7376||" />
     <SiteType SiteTypeIdx="18" SiteTypeId="17" SiteTypeName="college/université" SiteTypeExternalCode="7377||" />
     <SiteType SiteTypeIdx="19" SiteTypeId="18" SiteTypeName="Opéra" SiteTypeExternalCode="9365||" />
   </SiteTypeList>
 </ActionSiteTypeList>
```

6.18 - StreetNetwork

6.18.1 - Appel

« /API?Action=StreetNetwork »

6.18.2 - Exemple d'appel

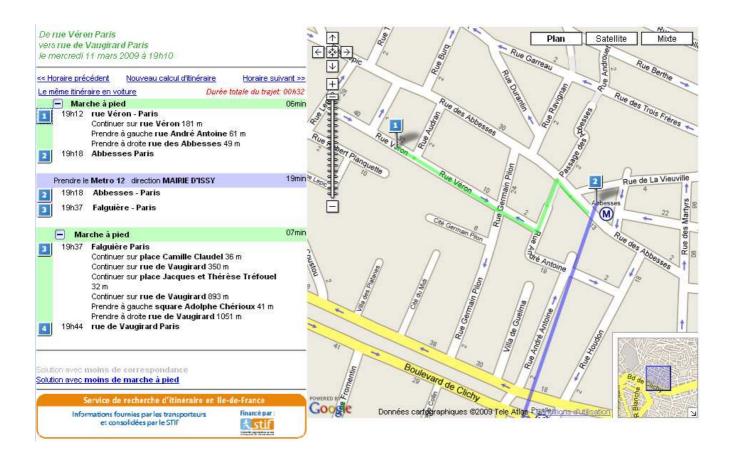
Sur un site fictif,

http://www.navitia.com/cgi-bin/navitia.ashx/api?action=streetnetwork&StartCoordX=599815,00&StartCoordY=2429233,50&EndCoordX=599890,54&EndCoordY=2429084,36&Walk Speed=50

6.18.3 - Description

Ce flux permet de calculer un itinéraire à pied uniquement en suivant le filaire de voierie. Le flux de réponse obtenu contient tous les segments de voierie utilisés, les coordonnées de leurs nœuds de début et de fin.

Le résultat de ce flux permet d'afficher le filaire de voierie détaillé sur une carte. Il permet également de générer des aides au déplacement comme par exemple dans les sections vertes ci-dessous :



6.18.4 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
StartCoordX	Réel			Coordonnée X de départ dans le système utilisé par les bases	Oui
StartCoordY	Réel			Coordonnée Y de départ dans le système utilisé par les bases	Oui
EndCoordX	Réel			Coordonnée X de destination dans le système utilisé par les bases	Oui
EndCoordY	Réel			Coordonnée Y de destination dans le système utilisé par les bases	Oui
WalkSpeed	intege r		50m/min	Vitesse de marche à pied	Non
HangDistance Dep	Intege r		1000	Distance maximale d'accroche autour des coordonnées de	Non

			départ en mètre	
HangDistance Arr	Intege r	1000	Distance maximale d'accroche autour des coordonnées d'arrivée en mètre	Non

6.18.5 - Sortie

Le flux de sortie est composé d'une liste de segment de voierie à emprunter. Chaque segment est composé de

- Une adresse (nœud standard adresse, avec commune, coordonnées du barycentre...)
- Un nœud de début de segment composé de Coordonnées
- Un nœud de fin de segment composé de Coordonnées

Attention:

Les nœuds de début et de fin de segment **ne sont pas forcément ordonnés entre 2 segments qui se suivent**. Il est donc conseillé de tracer chaque segment de façon autonome et non en utilisant uniquement les nœuds de départ de chaque segment par exemple.

Les balises spécifiques à un segment sont définies ci-dessous :

Balise	Description	Туре
Length	Longueur du segment	Integer (en mètre)
FromNbPar	Numéro paire de la voirie en face du début du segment. Peut être nul ou <= 0 : le segment porte alors sur une rue sans numéro du côté pair.	Integer
	Sur le premier segment, contient le numéro de la rue le plus proche de la coordonnée de départ.	
	Sur le dernier segment, contient le numéro de la rue le plus proche de la coordonnée d'arrivée.	
ToNbPar	Numéro paire de la voirie en face de la fin du segment. Peut être nul ou <= 0 : le segment porte alors sur une rue sans numéro du côté pair.	Integer
	Sur le premier segment, contient le numéro de la rue le plus proche de la coordonnée de départ.	
	Sur le dernier segment, contient le numéro	

Balise	Description	Туре
	de la rue le plus proche de la coordonnée d'arrivée.	
FromNbOdd	Numéro impaire de la voirie en face du début du segment. Peut être nul ou <= 0 : le segment porte alors sur une rue sans numéro du côté impair.	Integer
	Sur le premier segment, contient le numéro de la rue le plus proche de la coordonnée de départ.	
	Sur le dernier segment, contient le numéro de la rue le plus proche de la coordonnée d'arrivée.	
ToNbOdd	Numéro impaire de la voirie en face de la fin du segment. Peut être nul ou <= 0 : le segment porte alors sur une rue sans numéro du côté impair.	Integer
	Sur le premier segment, contient le numéro de la rue le plus proche de la coordonnée de départ.	
	Sur le dernier segment, contient le numéro de la rue le plus proche de la coordonnée d'arrivée.	
Duration	Arrondi de la durée de parcours du segment en minute en fonction de walkSpeed	

6.18.6 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionStreetNetwork>
 - <Params Function="APi">
     <action>streetnetwork</action>
     <StartCoordX>599815,00</StartCoordX>
     <StartCoordY>2429233,50</StartCoordY>
     <EndCoordX>599890,54</EndCoordX>
     <EndCoordY>2429084,36</EndCoordY>
     <WalkSpeed>50</WalkSpeed>
   </Params>
 - <SegmentList SegmentCount="6">
   - <Segment Length="42" FromNbPar="-1" ToNbPar="-1" FromNbOdd="7" ToNbOdd="0" Duration="1">
     - <Address AddressIdx="78430" AddressId="7905" AddressName="Jardin du Carrousel" AddressExternalCode="1881">
        <City CityIdx="1011" CityId="1011" CityName="Paris" CityExternalCode="75056" CityCode="75001" />
        <AddressType AddressTypeIdx="0" AddressTypeId="-1" AddressTypeName="" AddressTypeExternalCode="" />
      - <Coord>
          <CoordX>599791,83</CoordX>
          <CoordY>2429246,51</CoordY>
        </Coord>
      </Address>
     - <StartNode>
      < <Coord>
          <CoordX>599813,93</CoordX>
          <CoordY>2429233,85</CoordY>
        </Coord>
      </StartNode>
     - <EndNode>
      - <Coord>
          <CoordX>599804,55</CoordX>
          <CoordY>2429204,93</CoordY>
        </Coord>
      </EndNode>
     </Seament>
   + <Segment Length="20" FromNbPar="-1" ToNbPar="-1" FromNbOdd="-1" ToNbOdd="-1" Duration="0">
   + <Segment Length="83" FromNbPar="-1" ToNbPar="-1" FromNbOdd="-1" ToNbOdd="-1" Duration="1">
   + <Segment Length="18" FromNbPar="-1" ToNbPar="-1" FromNbOdd="-1" ToNbOdd="-1" Duration="0">
   + <Segment Length="8" FromNbPar="-1" ToNbPar="-1" FromNbOdd="-1" ToNbOdd="-1" Duration="0">
   + <Segment Length="162" FromNbPar="-1" ToNbPar="-1" FromNbOdd="-1" ToNbOdd="0" Duration="4">
   </SeamentList>
 </ActionStreetNetwork>
```

6.19 - VPatternSetList

6.19.1 - Appel

« /API?Action=VPatternSetList »

6.19.2 - Description

Ce flux contient la liste des couples « périodes et jours de fonctionnement » disponibles. Il est utile en particulier lors de demande de grilles horaire de ligne ou de tableaux d'horaire à l'arrêt, que l'on peut alors groupés : horaires de semaine scolaire, horaires de samedi...

6.19.3 - Tableau des paramètres de l'API

Nom Type Format Défaut Description Obligatoire

Nom	Туре	Format	Défaut	Description	Obligatoire
VPatternSetExternal Code	String			Ne renvoie que la période dont le code externe est passé en paramètre	Non
VPatternSetIdx	integer			Ne renvoie que la période dont l'index est passé en paramètre	Non
LineExternalCode	String			Ne renvoie que les périodes référencées par la ligne dont le code externe est passé en paramètre	Non

6.19.4 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ActionValidityPatternSetList>
- <params fonction="api">
       <action>vpatternsetlist</action>
       <vpatternsetidx>1</vpatternsetidx>
- <ValidityPatternSetList ValidityPatternSetCount="1">
   - <ValidityPatternSet ValidityPatternSetIdx="1" ValidityPatternSetId="3" ValidityPatternSetName="Dimanche et fêtes" ValidityPatternExternalCode="">
       - <ValidityPatternSetDetail ValidityPatternSetDetailId="1" ValidityPatternSetDetailId="10" Val
             < ValidityPattern
                 </ValidityPatternSetDetail>
       - <ValidityPatternSetDetail ValidityPatternSetDetailIdx="2" ValidityPatternSetDetailId="23" ValidityPatternSetDetailNota="Pendant les vacances scolaires">
             < ValidityPattern
                 </ValidityPatternSetDetail>
       - <LineList>
          - <Line LineIdx="1" LineId="1" LineName="Scolaire Collège de Marcq" LineCode="40r" LineExternalCode="1">
                 <Mode ModeIdx="3" ModeId="3" ModeName="bus urbain" ModeExternalCode="bus urbain" />
                 <Network NetworkIdx="9" NetworkId="47" NetworkName="Transpole" NetworkExternalCode="47" />
             - <Forward ForwardName="":
                 - <Direction>
                     - <StopArea StopAreaIdx="2191" StopAreaId="1067" StopAreaName="collége" StopAreaExternalCode="1067">
                           <City CityIdx="1072" CityId="28" CityName="Marcq en Baroeul" CityExternalCode="28" />
                        - <Coord>
                              <CoordX>654357,00</CoordX>
                              <CoordY>2632397,00</CoordY>
                           </Coord>
                        </StopArea>
                     </Direction>
                 </Forward>
             - <Backward BackwardName="Scolaire Collège de Marcg">
                 - <Direction>
                     - <StopArea StopAreaIdx="0" StopAreaId="-1" StopAreaName="" StopAreaExternalCode="">
                           <City CityIdx="0" CityId="-1" CityName="" CityExternalCode="" />
                           <Coord />
                        </StopArea>
                    </Direction>
                 </Backward>
             </Line>
          </LineList>
       </ValidityPatternSet>
    </ValidityPatternSetList>
</ActionValidityPatternSetList>
```

6.20 - VPTranslator

6.20.1 - Appel

« /API?Action=VPTranslator »

6.20.2 - Description

Ce flux permet de traduire une liste de dates en une phrase intelligible. Cette API est utile :

en sorti de fiche horaire, chaque desserte présente en effet son « tableau de marche » ou « condition de service » :

En sorti de calcul d'itinéraire (ou fiche horaire multimodale), NAVITIA présentant alors le « tableau de marche multimodal » ou les « conditions de service multimodales », intersection des tableaux de marche de chacune des dessertes utilisées :

```
<JourneyResultList JourneyResultCount="1">
– <JourneyResult SectionCount="9" Criteria="AsSoonAsPossible" IsCriteriaFound="0" IsBest="1" JourneyResultPosition="1" IsFirstSoluce="0" IsLastSoluce="0" IsDisrupt="0"
 IsAdapted="0":
 + <Section Type="Undefined"></Section>
 + < Section Type="VehicleJourneyConnection"> </ Section>
 + < Section Type="LinkConnection"> </ Section>
 + < Section Type="StopPointConnection"> </ Section>
 + < Section Type="VehicleJourneyConnection"> </ Section>
 + < Section Type="LinkConnection"> </ Section>
 + < Section Type="StopPointConnection"> </ Section>
 + < Section Type="VehicleJourneyConnection"> </ Section>
  + <Section Type="Undefined"></Section>

- <ValidityPattern ValidityPatternBeginningDay="2009|09|01">

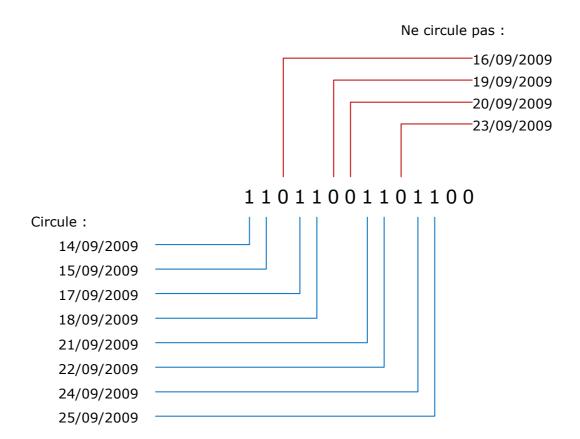
    </ValidityPattern>
 </JourneyResult>
```

6.20.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
Date	String	Year Month Day	Jour courant	Date de début de fonctionnement. Issue de la « ValidityPatternBeginningDa y » des résultats de cacul d'itinéraire	Oui
Pattern	String	111100 Chaine composée exclusivement de "0" et de "1"		Liste des jours actifs à partir de BeginningDay	Oui

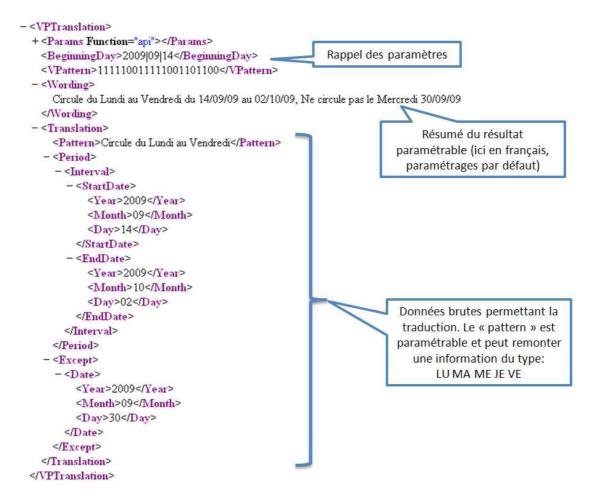
6.20.4 - Description par l'exemple :

- Si « Date=14/09/2009 », et que « Pattern=11011001101100 »
- Alors la demande porte sur : traduire la liste des jours actifs suivant :
 - o 14/09/2009
 - 0 15/09/2009
 - 0 17/09/2009
 - 0 18/09/2009
 - o 21/09/2009
 - o 22/09/2009
 - 0 24/09/2009
 - 0 25/09/2009
- En effet :



La réponse NAViTiA sera du type « circule du Lundi au vendredi, sauf mercredi, du 14/09/2009 au 25/09/2009

6.20.5 - Exemple



7 - LISTE DES API CALCULÉES TENANT COMPTE DU TEMPS RÉEL

Ces API utilisent AlerteTrafic afin de tirer parti des perturbations en temps réel. Ainsi il est possible :

- de demander à NAViTiA de calculer un itinéraire,
- de recevoir une réponse du type « voici l'itinéraire attendu. Attention cet itinéraire est perturbé »
- de demander à NAViTiA de calculer une solution de contournement

7.1 - DepartureBoard

7.1.1 - Appel

« /API?Action=DepartureBoard»

7.1.2 - Description

Ce flux contient l'ensemble des horaires au départ d'un arrêt donné pour une ligne commerciale donnée.

7.1.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obliga toire
StopAreaExternalCode	string			Code Externe de la zone d'arrêt (StopArea) de la fiche horaire. Au choix avec StopPointExternalCode : l'un des 2 est obligatoire	Oui
StopPointExternalCode	string			Code Externe du point d'arrêt (StopPoint) de la fiche horaire. Au choix avec StopAreaExternalCode : l'un des 2 est obligatoire	Oui
LineExternalCode	String			Code Externe de la ligne à observer.	Oui
				Obligatoire si pas de zone d'arrêt de destination renseignée	
DestinationExternalCode	string			Filtre pour présenter uniquement les horaires à destination d'une zone d'arrêt. Code externe de cette zone d'arrêt.	Non
				obligatoire si pas de ligne renseignée, sauf si le paramètre DestinationStopPointExternalCode est renseigné	
DestinationStopPointExternalCode	string			Filtre pour présenter uniquement les horaires à destination d'un point d'arrêt. Code externe de ce point d'arrêt.	Non
				obligatoire si pas de ligne renseignée, sauf si le paramètre DestinationExternalCode est renseigné	

Nom	Туре	Format	Défaut	Description		
Sens	1 / -1		1	sens=1 correspond l'ensemble des parcours rattachés à la "ligne commerciale aller"		
RouteExternalCode	String			permet de filtrer les horaires pour un parcours uniquement. Actif si la ligne est renseignée uniquement		
ModeTypeExternalCode	String			permet de filtrer les horaires sur une liste de types de mode des circulations.	Non	
				Actif si la ligne n'est PAS renseignée uniquement		
ModeExternalCode	String			permet de filtrer les horaires sur une liste de types de mode des circulations dont les modes sont spécifiés.	Non	
				Actif si la ligne n'est PAS renseignée uniquement		
Date	String	Year Month Day		Date des départs désirés	Oui	
VPatternSetIdx	integer		0	Index de la période des départs désirés / si <= 0 alors calcul sur la date		
NetworkExternalCode	List			Liste de codes des réseaux par lesquels doivent passer les circulations à observer	Non	
DateChangeTime	String	Hour Minute	00 00	Heure de changement de date (présentation en « calendrier TV »)		
UseAdapted	Boolean		True	Indique si le calcul doit utiliser des données adaptées	Non	

Remarque:

- Si VpatternIdx a une valeur inférieure ou égale à 0 alors la recherche s'effectue sur la date. La date doit avoir le format « yyyy|mm|dd »
- L'attribut « StopList/Stop/@DestinationPos » contient l'index du StopArea de destination s'il est différent de l'arrêt principal de destination de la ligne, dans la liste « DestinationList ».
- L'attribut « StopList/Stop/@ValidityPatternSetCommentPos » contient l'index du régime exact utilisé dans la liste « ValidityPattenSet ».
- A partir de l'interface 1.8.2 les horaires de montée et de descente sont présentés. L'application média peut néanmoins simplifier la présentation en affichant uniquement les horaires de montée dans le cas d'une fiche horaire à l'arrêt.

Cause de non réponses : détaillées dans le chapitre 7.2.4 - Cause de réponse sans horaires

Pour des raisons de compatibilité, les paramètres « mode » fournis à l'API seront interprétés comme des « ModeType »

7.1.4 - Gestion des notes horaires et des destinations principales / secondaires

Dans le flux de réponse, chaque élément <Stop> possède un attribut @DestinationPos.

- Si la valeur de cet attribut est égale à -1
 - alors l'arrêt terminus est identique à l'arrêt habituel.
- Sinon
 - La valeur de @ DestinationPos indique une position dans la liste DestinationList (dernière liste du flux de réponse) tout en sachant que le premier élément de cette liste est d'indice 0.
 - Exemple:
 - Horaire à afficher:

```
<Stop StopIdx="60268" StopPointIdx="2145" VehicleJourneyIdx="808" HourNumber="3" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="12" VehicleIdx="2">.
```

Liste finale des destination:

```
<DestinationList DestinationCount="1">

<StopArea StopAreaIdx="1057" StopAreaId="406" StopAreaName="...>

<City CityIdx="242" CityId="26" CityName="Rennes" ...>

</City>

<Coord>

<CoordX>304096,89</CoordX>

<CoordY>2352481,95</CoordY>

</Coord>

</StopArea>
```

- </DestinationList>
- Alors dans ce cas l'arrêt de destination à afficher pour l'horaire 60268 est celui indiqué par l'élément StopArea
- Dans le principe, il faut créer une note pour chaque élément StopArea de la liste destinationList et dans les horaires la référence à cette note est créé suivant le schéma décisionnel expliqué si dessus.

7.1.5 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<DepartureBoardList MaxHourNumber="13" MaxMinuteNumber="3">
- <Params Function="API">
       <Action>departureboard</Action>
       lineidx>10</lineidx>
       <stopareaidx>1470</stopareaidx>
       <date>2007|11|07</date>
       <vpatternsetidx>-1/vpatternsetidx>
- <StopList StopCount="6" Nota="">
  + <Stop StopIdx="213340" StopPointIdx="3464" VehicleJourneyIdx="434" HourNumber="0" MinuteNumber="0" DestinationPos="-1"
          ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="1" VehicleIdx="0">
   + <Stop StopIdx="325690" StopPointIdx="3464" VehicleJourneyIdx="433" HourNumber="0" MinuteNumber="1" DestinationPos="-1"
  ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="1" VehicleIdx="0"> + <Stop StopIdx="402293" StopPointIdx="3464" VehicleJourneyIdx="451" HourNumber="0" MinuteNumber="2" DestinationPos="-1"
          ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="1" VehicleIdx="0">
  + <Stop StopIdx="492219" StopPointIdx="3464" VehicleJourneyIdx="438" HourNumber="1" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="1" VehicleIdx="0">
   + <Stop StopIdx="596243" StopPointIdx="3464" VehicleJourneyIdx="432" HourNumber="1" MinuteNumber="1" DestinationPos="-1"
   ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="1" VehicleIdx="0"> - <Stop StopIdx="6127506" StopPointIdx="3464" VehicleJourneyIdx="453" HourNumber="13" MinuteNumber="1" DestinationPos="-1" ODTIdx="0" ODTPos="-1"
         ValidityPatternSetCommentPos="0" StopOrder="1" VehicleIdx="0">
       < <StopTime>
             <TotalSeconds>70260</TotalSeconds>
             <Dav>0</Dav>
             <Hour>19</Hour>
             <Minute>31</Minute>
          </StopTime:
          <Route RouteIdx="95" RouteId="588208" RouteName="011 - Bois d'arcy - St Cyr l'école - Versailles" RouteExternalCode="0270270110001" IsForward="1"</p>
             RouteLineIdx="10" IsFrequence="0" /
       </Stop:
   </StopList>
 <StopPointList StopPointCount="12">
  + <StopPoint StopPointIdx="3464" StopPointId="168761" StopPointName="Paul Vaillant Couturier" StopPointExternalCode="50|5621" FareZone="5">
  + <StopPoint StopPointId="3455" StopPointId="168750" StopPointName="Le Puits" StopPointExternalCode="50|5627" FareZone="5">
+ <StopPoint StopPointId="3455" StopPointId="168750" StopPointName="Mairie" StopPointExternalCode="50|5610" FareZone="5">
+ <StopPoint StopPointId="3450" StopPointId="168733" StopPointName="C.C. Leclerc" StopPointExternalCode="50|5593" FareZone="5">
+ <StopPoint StopPointId="3430" StopPointId="168733" StopPointName="C.C. Leclerc" StopPointExternalCode="50|5593" FareZone="5">
+ <StopPoint StopPointId="3450" StopPointId="168733" StopPointName="C.C. Leclerc" StopPointExternalCode="50|5593" FareZone="5">
+ <StopPoint StopPointId="168733" StopPointName="C.C. Leclerc" StopPointExternalCode="50|5593" FareZone="5">
+ <StopPoint StopPointId="168733" StopPointName="C.C. Leclerc" StopPointExternalCode="50|5593" FareZone="5">
+ < StopPoint StopPointId="168733" StopPointName="C.C. Leclerc" StopPointExternalCode="50|5593" FareZone="5">
+ < StopPoint 
  + <StopPoint StopPointIdx="35598" StopPointId="170818" StopPointName="Terriers Blancs" StopPointExternalCode="50|7678" FareZone="5">
 + <StopPoint StopPointId="35590" StopPointId="170816" StopPointName="Les 2 Ponts" StopPointExternalCode="50|7676" FareZone="5">
+ <StopPoint StopPointId="35508" StopPointId="170816" StopPointName="Les 2 Ponts" StopPointExternalCode="50|7674" FareZone="5">
+ <StopPoint StopPointId="35508" StopPointId="170814" StopPointName="République" StopPointExternalCode="50|7674" FareZone="5">
+ <StopPoint StopPointId="42460" StopPointId="171304" StopPointName="INRA" StopPointExternalCode="50|8164" FareZone="4">
+ <StopPoint StopPointId="42476" StopPointId="171314" StopPointName="Les Matelots" StopPointExternalCode="50|8168" FareZone="4">
+ <StopPoint StopPointId="42476" StopPointId="171308" StopPointName="L'Orangerie" StopPointExternalCode="50|8168" FareZone="4">
 <Mode ModeIdx="3" ModeId="2" ModeName="BUS" ModeExternalCode="RATP" />
          <City CityIdx="1324" CityId="1325" CityName="Versailles" CityExternalCode="78646" CityCode="78000" /
       - <StopArea StopAreaIdx="17816" StopAreaId="6681" StopAreaName="GARE DE VERSAILLES RIVE GAUCHE" StopAreaExternalCode="8739315" MainStopArea="1" MultiModal="0" CarPark="0" MainConnection="0" AdditionalData="">
           < Coord>
                 <CoordX>584687,65</CoordX>
                 <CoordY>2422468,22</CoordY>
             </Coord>
          </StopArea>
              CoordY > 504600 09 < /CoordY >
```

>>>>>> Liste des API calculées tenant compte du temps réel

```
<CoordY>2422452.26</CoordY>
    </Coord>
  </StopPoint>
</StopPointList>
<LineList LineCount="1">
- <Line LineIdx="10" LineId="307540" LineName="011 - Bois d'arcy - St Cyr l'Ecole - Versailles" LineCode="011 - Bois" LineExternalCode="027027011"
    LineAdditionalData="" SortOrder="10">
<Mode ModeIdx="3" ModeId="2" ModeName="BUS" ModeExternalCode="RATP" />
  - <Network NetworkIdx="56" NetworkId="4609" NetworkName="HOURTOULE" NetworkExternalCode="027">
      <EventPosList EventPosCount="0" />
    </Network>
  - <Forward ForwardName="011 - Bois d'arcy - St Cyr l'école - Versailles">
    - <Direction>
      - <StopAreaIdx="17816" StopAreaId="6681" StopAreaName="GARE DE VERSAILLES RIVE GAUCHE" StopAreaExternalCode="8739315"
        MainStopArea="1" MultiModal="0" CarPark="0" MainConnection="0" AdditionalData=""> - <City CityIdx="1324" CityId="1325" CityName="Versailles" CityExternalCode="78646" CityCode="78000">
            <Country CountryIdx="1" CountryId="0" CountryName="France" CountryExternalCode="FRA" />
          </City>
        - <Coord>
            <CoordX>584687,65</CoordX>
            <CoordY>2422468,22</CoordY>
          </Coord>
         </StopArea>
       </Direction>
    </Forward>
  - <Backward BackwardName="011 - Bois d'arcy - St Cyr l'école - Versailles">
    - <Direction>
      - <StopArea StopAreaIdx="1460" StopAreaId="4246" StopAreaName="Croix Blanche" StopAreaExternalCode="Croix Blanche" MainStopArea="0" MultiModal="0"
        CarPark="0" MainConnection="0" AdditionalData="">
- <City CityIdx="120" CityId="125" CityName="Bois-d'Arcy" CityExternalCode="78073" CityCode="78390">
            <Country CountryIdx="1" CountryId="0" CountryName="France" CountryExternalCode="FRA" />
          </City>
        - <Coord>
            <CoordX>576460,29</CoordX>
            <CoordY>2422246,18</CoordY>
          </Coord>
         </StopArea>
       </Direction>
    </Backward>
    <EventPosList EventPosCount="0" />
  </Line>
 </LineList>
<RouteList RouteCount="1">
- <Route RouteId="95" RouteId="588208" RouteName="011 - Bois d'arcy - St Cyr l'école - Versailles" RouteExternalCode="0270270110001" IsForward="1"</pre>
    RouteLineIdx="10" IsFrequence="0">
  - <Line LineIdx="10" LineId="307540" LineName="011 - Bois d'arcy - St Cyr l'Ecole - Versailles" LineCode="011 - Bois" LineExternalCode="027027011" LineAdditionalData="" SortOrder="10"> < Mode ModeIdx="3" ModeId="2" ModeName="BUS" ModeExternalCode="RATP" />
      <Network NetworkIdx="56" NetworkId="4609" NetworkName="HOURTOULE" NetworkExternalCode="027" />
      <Forward ForwardName="011 - Bois d'arcy - St Cyr l'école - Versailles"
      <Backward BackwardName="011 - Bois d'arcy - St Cyr l'école - Versailles" />
    </Line>
  </Route>
</RouteList>
<DestinationList DestinationCount="0" />
<CommentList CommentCount="0" />
<ODTList ODTCount="0" />
<ValidityPatternSet ValidityPatternSetIdx="0" ValidityPatternSetId="-1" ValidityPatternSetName="" ValidityPatternExternalCode="" ValidityPatternAdditionalData="">
  <ValidityPatternSetDetailList VPatternSetDetailCount="1" />
 </ValidityPatternSet>
< VehicleList VehicleCount="1":
  <Vehicle VehicleIdx="0" VehicleId="-1" VehicleName="" VehicleExternalCode="" MIPAccess="False" BikeAccepted="False" AirConditioned="False" />
 </VehicleList>
'DepartureBoardList:
```

7.2 - LineSchedule

7.2.1 - Appel

« /API?Action=LineSchedule»

7.2.2 - Description

Ce flux contient la grille horaire d'une ligne entre 2 arrêts à une date donnée.

7.2.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
LineExternalCode	String			Code Externe de la ligne à observer.	Oui
				N'est plus obligatoire si DepartureExternalCode et ArrivalExternalCode sont renseignés	
DepartureExternalCode	string			Code externe de la zone d'arrêt (StopArea) de départ	Non
				Obligatoire si pas de ligne renseignée, sauf si le paramètre DepartureStopPointExternalCode est renseigné	
DepartureStopPointExternalCode				Code externe du point d'arrêt (StopPoint) de départ	Non
				Obligatoire si pas de ligne renseignée, sauf si le paramètre DepartureExternalCode est renseigné	
ArrivalExternalCode	string			Code externe de la zone d'arrêt (StopArea) de destination.	Non
				Obligatoire si pas de ligne renseignée, sauf si le paramètre ArrivalStopPointExternalCode est renseigné	
ArrivalStopPointExternalCode				Code externe du point d'arrêt (StopPoint) de départ	Non
				Obligatoire si pas de ligne renseignée, sauf si le paramètre ArrivalExternalCode est renseigné	
Sens	1 / -1		1	sens=1 correspond l'ensemble des parcours rattachés à la "ligne commerciale aller"	Non

Nom	Туре	Format	Défaut	Description	Obligatoire
RouteExternalCode	String			permet de filtrer les horaires pour un parcours uniquement. Actif si la ligne est renseignée uniquement	Non
ModeTypeExternalCode	String			permet de filtrer les horaires sur une liste de types de mode des circulations.	Non
				Actif si la ligne n'est PAS renseignée uniquement	
ModeExternalCode	String			permet de filtrer les horaires sur une liste de modes des circulations dont les modes sont spécifiés.	Non
				Actif si la ligne n'est PAS renseignée uniquement	
Date	String	Year Month Day		Date des départs désirés	Oui
VPatternSetIdx	integer		0	Index de la période des départs désirés / si <= 0 alors calcul sur la date	Non
NetworkExternalCode	List			Liste de codes des réseaux par lesquels doivent passer les circulations à observer	Non
DateChangeTime	String	Hour Minute	00 00	Heure de changement de date (présentation de type « calendrier TV »)	Non
UseAdapted	Boolean		True	Indique si le calcul doit utiliser des données adaptées	Non

Remarque:

- Si VpatternIdx a une valeur inférieure ou égale à 0 alors la recherche s'effectue sur la date.
- La date doit avoir le format « yyyy|mm|dd »

- Si LineExternalCode a une valeur nulle ou inconnue alors la recherche s'effectuera sur toutes les lignes entre les points de départ et d'arrivée, filtrés éventuellement par le mode renseigné (ModeTypeExternalCode ou ModeExternalCode).
- A partir de l'interface 1.8.2 les horaires de montée et de descente sont présentés. L'application média peut néanmoins simplifier la présentation en affichant par exemple :
 - L'horaire de montée sur le premier arrêt
 - Les horaires de descente sur tous les autres arrêts.

Pour des raisons de compatibilité, les paramètres « mode » fournis à l'API seront interprétés comme des « ModeType »

7.2.4 - Cause de réponse sans horaires

"lineschedule date out of bound"

La date de recherche est en dehors de la période de validité des données.

"lineschedule_terminus"

La demande est faite sur un arrêt terminus.

"lineschedule_no_departure_stoppoint_this_date"

L'arrêt d'origine n'est pas desservi ce jour.

"lineschedule_no_active_circulation_this_date"

Aucune circulation active sur cette ligne le jour demandé.

"lineschedule_no_active_mode_this_date"

Aucune circulation utilisant le mode n'est active entre l'arrêt d'origine et l'arrêt de destination le jour demandé

"lineschedule_no_direct_route_this_date"

Aucune circulation directe entre l'arrêt d'origine et l'arrêt de destination n'est active le jour demandé

"departureboard_no_departure_stoppoint"

Aucun point d'arrêt n'est sélectionné: si par exemple le StopAreaExternalCode passé en parameter n'est pas valide.

7.2.5 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <LineScheduleList>
  - <Params Function="Navitia_Debug.DEBUG_VKAT/api">
       <action>lineschedule</action
      lineidx>112</lineidx>
       <date>2007|11|01</date>
    </Params:
  - <StonList StonCount="12" Nota="">
    + <Stop StopIdx="104884" StopPointIdx="3509" VehicleJourneyIdx="657" HourNumber="0" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1"
    ValidityPatternSetCommentPos="0" StopOrder="0" VehicleIdx="1"> + <Stop StopIdx="110741" StopPointIdx="3550" VehicleJourneyIdx="657" HourNumber="0" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1"
    ValidityPatternSetCommentPos="0" StopOrder="1" VehicleIdx="1">

ValidityPatternSetCommentPos="0" StopOrder="1" VehicleIdx="1">

+ <Stop StopIdx="112021" StopPointIdx="4933" VehicleJourneyIdx="657" HourNumber="0" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="2" VehicleIdx="1">
    + <Stop StopIdx="115316" StopPointIdx="2312" VehicleJourneyIdx="657" HourNumber="0" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1"
         ValidityPatternSetCommentPos="0" StopOrder="3" VehicleIdx="1"
    + <Stop StopIdx="120442" StopPointIdx="3259" VehicleJourneyIdx="657" HourNumber="0" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1"
    ValidityPatternSetCommentPos="0" StopOrder="4" VehicleIdx="1"> + <Stop StopIdx="122450" StopPointIdx="4629" VehicleJourneyIdx="657" HourNumber="0" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1"
    ValidityPatternSetCommentPos="0" StopOrder="5" VehicleIdx="1"> + <Stop StopIdx="125594" StopPointIdx="4242" VehicleJourneyIdx="657" HourNumber="0" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1"
         ValidityPatternSetCommentPos="0" StopOrder="6" VehicleIdx="1"
    + <Stop StopIdx="128895" StopPointIdx="1985" VehicleJourneyIdx="657" HourNumber="0" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="7" VehicleIdx="1">
      <Stop StopIdx="132794" StopPointIdx="2361" VehicleJourneyIdx="657" HourNumber="0" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1"
        ValidityPatternSetCommentPos="0" StonOrder="8" VehicleIdx="1"
      <Stop StopIdx="138171" StopPointIdx="3145" VehicleJourneyIdx="657" HourNumber="0" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1"
    ValidityPatternSetCommentPos="0" StopOrder="9" VehicleIdx="1"> + <Stop StopIdx="144826" StopPointIdx="265" VehicleJourneyIdx="657" HourNumber="0" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1"
     ValidityPatternSetCommentPos="0" StopOrder="10" VehicleIdx="1" 

< Stop StopIdx="150279" StopPointIdx="4060" VehicleJourneyIdx="657" HourNumber="0" MinuteNumber="0" DestinationPos="0" ODTIdx="0" ODTPos="-1"</p>
        ValidityPatternSetCommentPos="0" StopOrder="11" VehicleIdx="1"
       – <StopŤime>
           <TotalSeconds>39300</TotalSeconds>
           <Day>0</Day>
           <Hour>10</Hour>
           <Minute>55</Minute>
         </StopTime>
       </Stop:
    </StopList>
    <FreqSettingList FreqSettingCount="0" />

        <Mode ModeIdx="22" ModeId="2" ModeName="Train TER" ModeExternalCode="OCETrain TER" />
      <City CityIdx="23922" CityId="29577" CityName="Paray-le-Monial" CityExternalCode="71342" CityCode="71600" />
- <StopArea StopAreaIdx="4060" StopAreaId="4171" StopAreaName="gare de Paray-le-Monial" StopAreaExternalCode="OCE87694687" MainStopArea="1"
    MultiModal="0" CarPark="0" MainConnection="0" AdditionalData="">
          <Coord>
             <CoordX>736520,00</CoordX>
             <CoordY>2162320,00</CoordY>
           </Coord>
        </StopArea>
       + <Coord>
      </StopPoint>
    </StopPointList>
 - <VehicleJourneyList VehicleJourneyCount="1">
      < VehicleJourney VehicleJourneyIdx="657" VehicleJourneyId="41823" VehicleJourneyName="C39502" VehicleJourneyRouteIdx="186"
        VehicleJourneyExternalCode="OCE4583-3" IsAdapted="0"> < Mode ModeIdx="2" ModeId="20" ModeName="Car TER" ModeExternalCode="OCECar TER" />
        <
      </VehicleJourney>
    </VehicleJourneyList>
   <RouteList RouteCount="1">
- <Route RouteIdx="186" RouteId="4576" RouteName="Lozanne vers Paray Le Monial" RouteExternalCode="OCECAR-87721423-87694687" IsForward="1"</pre>
      RouteLineIdx="112" IsFrequence="0" IsAdapted="0">
- <Line LineIdx="112" LineId="2719" LineName="Lozanne - Paray Le Monial" LineCode="Car TER" LineExternalCode="OCECAR-87721423-87694687"
LineAdditionalData="" SortOrder="112" 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="lozanne gare sncf - paray le monial gare sncf",</pre>
           <Backward BackwardName="paray le monial gare sncf - lozanne gare sncf" />
        </Line>
      </Route>
    </RouteList>
    <CommentList CommentCount="0" />
    <ODTList ODTCount="0" />
    <ValidityPatternSet ValidityPatternSetIdx="0" ValidityPatternSetId="-1" ValidityPatternSetName="" ValidityPatternExternalCode="" ValidityPatternAdditionalData="">
      <ValidityPatternSetDetailList VPatternSetDetailCount="1" />
    </ValidityPatternSet>
  </LineScheduleList>
```

7.2.6 - Affichage de la grille horaire

La réponse fournie par NAViTiA contient 2 listes principales : la liste des horaires et la liste des arrêts.

- -<LineScheduleList>
 - + <Params Function="api"></Params>
- + <StopList StopCount="384" Nota=""></StopList>
 - <FreqSettingList FreqSettingCount="0"> </FreqSettingList>
- +<StopPointList StopPointCount="37"></StopPointList>
- + <VehicleJourneyList VehicleJourneyCount="11"></VehicleJourneyList>
- + < RouteList RouteCount="1"></RouteList>
- + < CommentList CommentCount="1"></CommentList>
- <ODTList ODTCount="0"> </ODTList>
- + <ValidityPatternSetValidityPatternSetIdx="0" ValidityPatternSetId="-1" ValidityPatternSetName="" ValidityPatternExternalCode="" ValidityPatternAdditionalData=""> </ValidityPatternSet>
- </LineScheduleList>

A partir de ces 2 listes, le média devra fabriquer un tableau avec par exemple la liste des arrêts dans la première colonne et les horaires dans les colonnes suivantes :

Rezé - PIRMIL	08:05	09:05	10:35	12:37	16:05		16:37	17:09		17:39
Nantes - Les Bourdonnières Lycée									17:15	
Rezé - JEAN PERRIN						16:45				
Le Bignon - Maison Rouge	08:20	09:20	10:50	12:52	16:27	17:00	17:04	17:29		18:05
Le Bignon - La Jarrie	08:25	09:25	10:55	12:57	16:30	17:02	17:05	17:32		18:07
Le Bignon - Les Aires	08:29	09:29	10:59	13:01	16:32	17:05	17:07	17:34		18:09
Le Bignon - Centre	08:30	09:30	11:00	13:02	16:33	17:06	17:08	17:35	17:35	18:10
Le Bignon - La Masure	08:32	09:32	11:02	13:04	16:35		17:10	17:37	17:37	18:13
Montbert - Les Chaises	08:33	09:33	11:03	13:05	16:36		17:11	17:38	17:38	18:14
Montbert - L'Hommeau	08:34	09:34	11:04	13:06	16:37		17:12	17:39	17:39	18:15
Montbert - La Croix des Prés	08:36	09:35	11:05	13:07	16:39	17:10	17:13	17:40	17:40	18:16
Montbert - Centre	08:37	09:37	11:07	13:10	16:40	17:12	17:14	17:42	17:42	18:17
Montbert - La Poste				13:12				17:43		18:18
La Planche - Nonnaire				13:15				17:45		18:21
La Planche - Le Noyer				13:17				17:48		18:24
La Planche - Centre				13:19				17:50		18:25
La Planche - La Moune				13:21				17:52		18:27
La Planche - Cinq Routes				13:23				17:54		18:28
Vieillevigne - L'Audonnière				13:25				17:56		18:29
Vieillevigne - Centre				13:27				18:00		18:32
Vieillevigne - Terrain des Sports				13:30				18:02		18:34
Geneston - Centre			11:11		16:45	17:17	17:19		17:47	

Cet affichage peut être compliqué, en particulier pour les cas suivants :

- Dans les cas de ligne en boucle : un même arrêt apparait alors 2 fois dans la liste et les horaires sont différents à chacun des passages
- Dans le cas de ligne dont les arrêts ne sont pas desservis dans le même sens : l'affichage de la grille devient alors très compliqué, avec des horaires pas toujours affichés dans le bon ordre

La restitution avec anomalie d'affichage classique est la suivante :

>>>>>> Liste des API calculées tenant compte du temps réel

Fontenay-le-Comte - Porteau	07:22	08:22	09:22
Fontenay-le-Comte - Cerisaie		07:28	08:28
Fontenay-le-Comte - Boidé		07:29	08:29
Fontenay-le-Comte - Gaillardon		07:31	08:31
Fontenay-le-Comte - Bédouard		07:32	08:32
Fontenay-le-Comte - Viète - Pt St Laurent		07:34	08:34
Fontenay-le-Comte - Mairie	07:15	08:15	09:15

Afin de simplifier l'intégration et de permettre au média d'éviter simplement cette anomalie, NAViTiA remonte 2 informations : le numéro de colonne et le numéro de ligne pour chaque horaire.

Voici par exemple la liste des arrêts d'une ligne en boucle :

```
+ <StopPoint StopPointIdx="14280" StopPointId="21638" StopPointName="Porteau" StopPointExternalCode="FONNav1" FareZone="0"></StopPoint>
 +<StopPointStopPointIdx="14241" StopPointId="21604" StopPointName="Cerisaie" StopPointExternalCode="FONNav2" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14237" StopPointId="21600" StopPointName="Boidé" StopPointExternalCode="FONNav3" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14255" StopPointId="21616" StopPointName="Gaillardon" StopPointExternalCode="FONNav4" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14238" StopPointId="21601" StopPointName="Bédouard" StopPointExternalCode="FONNav5" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14291" StopPointId="21649" StopPointName="Viète - Pt St Laurent" StopPointExternalCode="FONNav6" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14269" StopPointId="21628" StopPointName="Mairie" StopPointExternalCode="FONNav26" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14281" StopPointId="21639" StopPointName="Poste" StopPointExternalCode="FONNav8" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14284" StopPointId="21642" StopPointName="Seillot" StopPointExternalCode="FONNav9" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14235" StopPointId="21598" StopPointName="Biossais" StopPointExternalCode="FONNav10" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14272" StopPointId="21630" StopPointName="Moriceau" StopPointExternalCode="FONNav11" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14285" StopPointId="21643" StopPointName="St Médard" StopPointExternalCode="FONNav12" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14268" StopPointId="21626" StopPointName="Maingueneau" StopPointExternalCode="FONNav13" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14288" StopPointId="21646" StopPointName="Vinet" StopPointExternalCode="FONNav14" FareZone="0"><</stopPoint>
 + <StopPoint StopPointIdx="14243" StopPointId="21606" StopPointName="Champs Marot" StopPointExternalCode="FONNav15" FareZone="0"><</StopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPo
 + <StopPoint StopPointIdx="14266" StopPointId="21624" StopPointName="Libération" StopPointExternalCode="FONNav16" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14268" StopPointId="21626" StopPointName="Maingueneau" StopPointExternalCode="FONNav13" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14285" StopPointId="21643" StopPointName="St Médard" StopPointExternalCode="FONNav12" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14272" StopPointId="21630" StopPointName="Moricean" StopPointExternalCode="FONNav11" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14235" StopPointId="21598" StopPointName="Biossais" StopPointExternalCode="FONNav10" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14284" StopPointId="21642" StopPointName="Seillot" StopPointExternalCode="FONNav9" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14239" StopPointId="21602" StopPointName="Caserne" StopPointExternalCode="FONNav52" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14251" StopPointId="21614" StopPointName="Espace Culture!" StopPointExternalCode="FONNav18" FareZone="0"></StopPoint>
 +<StopPointStopPointIdx="14286" StopPointId="21644" StopPointName="Stade" StopPointExternalCode="FONNav19" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14278" StopPointId="21636" StopPointName="Pommeraie" StopPointExternalCode="FONNav20" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14275" StopPointId="21633" StopPointName="Mérité" StopPointExternalCode="FONNav21" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14283" StopPointId="21641" StopPointName="Sablière" StopPointExternalCode="FONNav22" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14264" StopPointId="21622" StopPointName="Les Horts" StopPointExternalCode="FONNav23" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14265" StopPointId="21623" StopPointName="Les Loges" StopPointExternalCode="FONNav24" FareZone="0"></StopPoint>
 + <StopPointStopPointIdx="14236" StopPointId="21599" StopPointName="Blossac" StopPointExternalCode="FONNav25" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14269" StopPointId="21628" StopPointName="Mairie" StopPointExternalCode="FONNav26" FareZone="0"></StopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStopPointStop
 + <StopPoint StopPointIdx="14292" StopPointId="21650" StopPointName="Viète Musée" StopPointExternalCode="FONNav27" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14252" StopPointId="21615" StopPointName="Espierre" StopPointExternalCode="FONNav28" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14246" StopPointId="21609" StopPointName="Château D'eau" StopPointExternalCode="FONNav29" FareZone="0"></StopPoint>
 + <StopPointStopPointIdx="14271" StopPointId="21629" StopPointName="Marchoux" StopPointExternalCode="FONNav30" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14260" StopPointId="21618" StopPointName="Guinefolle" StopPointExternalCode="FONNav31" FareZone="0"></StopPoint>
 + <StopPoint StopPointIdx="14280" StopPointId="21638" StopPointName="Porteau" StopPointExternalCode="FONNav1" FareZone="0"></StopPoint>
 </StopPointList>
```

L'arrêt de départ et le terminus sont identiques : la ligne est en boucle. Un extrait de la liste des horaires nous donne :

DESTRUMListe des API calculées tenant compte du temps réel

- <stoplist nota="" stopcount="384"></stoplist>
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="23" odtidx="0" odtpos="-1" stopidx="185660" stoppointidx="14286" td="" vehiclejourneyidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="0" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="24" odtidx="0" odtpos="-1" stopidx="190367" stoppointidx="14278" td="" vehiclejourneyidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="1" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="25" odtidx="0" odtpos="-1" stopidx="192605" stoppointidx="14275" td="" vehiclejourneyidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="2" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="26" odtidx="0" odtpos="-1" stopidx="195270" stoppointidx="14283" td="" vehiclejourneyidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="3" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="27" odtidx="0" odtpos="-1" stopidx="197639" stoppointidx="14264" td="" vehiclejourneyidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="4" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="28" odtidx="0" odtpos="-1" stopidx="200116" stoppointidx="14265" td="" vehiclejourneyidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="5" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="29" odtidx="0" odtpos="-1" stopidx="202492" stoppointidx="14236" td="" vehiclejourneyidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="6" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="30" odtidx="0" odtpos="-1" stopidx="204937" stoppointidx="14269" td="" vehiclejourneyidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="7" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="31" odtidx="0" odtpos="-1" stopidx="210059" stoppointidx="14292" td="" vehiclejourneyidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="8" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="32" odtidx="0" odtpos="-1" stopidx="212507" stoppointidx="14252" td="" vehiclejourneyidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="9" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="33" odtidx="0" odtpos="-1" stopidx="214870" stoppointidx="14246" td="" vehiclejourneyidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="10" VehicleIdx="10">
+ <stop destinationpos="-1" hournumber="0" minutenumber="34" odtidx="0" odtpos="-1" stopidx="217336" stoporder="11" stoppointidx="14271" validitypatternsetcommentpos="0" vehicleidx="10" vehiclejourneyidx="3273"></stop>
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="35" odtidx="0" odtpos="-1" stopidx="220089" stoppointidx="14260" td="" vehiclejournevidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="12" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="0" minutenumber="36" odtidx="0" odtpos="-1" stopidx="222559" stoppointidx="14280" td="" vehiclejournevidx="3273"></stop>
ValidityPatternSetCommentPos="0" StopOrder="13" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="1" minutenumber="0" odtidx="0" odtpos="-1" stopidx="235331" stoppointidx="14280" td="" vehiclejourneyidx="3274"></stop>
ValidityPatternSetCommentPos="0" StopOrder="0" VehicleIdx="10">
+ <stop <="" destinationpos="-1" hournumber="1" minutenumber="1" odtidx="0" odtpos="-1" stopidx="237885" stoppointidx="14241" td="" vehiclejourneyidx="3274"></stop>
ValidityPatternSetCommentPos="0" StopOrder="1" VehicleIdx="10">
+ Stop StopIdx="240392" StopPointIdx="14237" VehicleJourneyIdx="3274" HourNumber="1" MinuteNumber= "2" DestinationPos="-1" ODTIdx="0" ODTPos="-1"
ValidityPatternSetCommentPos="0" StopOrder="2" VehicleIdx="10">
•

Apparaissent en vert les horaires de la première desserte, à afficher dans la première colonne ; en orange les horaires de la deuxième desserte à afficher dans la deuxième colonne.

On retrouve en rouge le même arrêt « 14280 » qui est le dernier arrêt de la première desserte et également le premier de la seconde desserte.

- La propriété « HourNumber » indique directement le numéro de la colonne qui contient la desserte
 - « 0 » pour la première desserte
 - « 1 » poura la seconde desserte
 - ..
- La propriété « MinuteNumber » indique directement le numéro de la ligne de l'horaire :
 - Ainsi l'horaire de l'arrêt « 14280 », pour la première desserte (colonne « HourNumber="0" ») doit s'afficher en ligne « 36 » (« MinuteNumber="36" »)
 - Ainsi l'horaire de l'arrêt « 14280 », pour la deuxième desserte (colonne « HourNumber="1" ») doit s'afficher en ligne « 1 » (« MinuteNumber="0" »)

7.3 - MultiModalSchedule

7.3.1 - Appel

« /API?Action=MultiModalSchedule »

7.3.2 - Description

Ce flux contient la liste des trajets possibles entre 2 points pour une plage horaire donnée. Cette plage horaire peut concerner la journée complète, l'API correspondant alors à une grille horaire multimodale.

7.3.3 - Tableau des paramètres de l'API :

Les paramètres sont identiques aux paramètres de l'API PlanJourney, sauf :

Nom	Туре	Format	Défaut Paramètres princip	Description paux	Obligatoire
Time				Ce paramètre de l'API PLanJourney n'est pas pris en compte	-
StartTime	String	Hour Minute	00h00	Première heure de départ de la grille horaire	Non
EndTime	string	Hour Minute	24h00	Dernière heure d'arrivée de la grille horaire. Cet horaire peut dépasser 24h (par exemple 25h00 pour rechercher une grille horaire qui dépasse minuit)	Non
			Paramètres compléme	entaires	
Fuse	Integer		100	Nombre maximum d'horaires retournés	Non
DateChangeTime				Ce paramètre de l'API PLanJourney n'est pas pris en compte	-
ExtrapoleData				Ce paramètre de l'API PLanJourney n'est pas pris en compte	-
VPatternSetIdx				Ce paramètre de l'API PLanJourney n'est pas pris en compte	-

Le flux résultat est identique à celui de l'API PlanJourney.

Limitations actuelles:

- Aujourd'hui la gestion des « via » n'est pas compatible avec la gestion d'une journée sur plus de 24h00 (EndTime > 23h59)
- La plage horaire est limitée à 47h59

7.4 - NextArrival

7.4.1 - Appel

« /API?Action=NextArrival »

7.4.2 - Description

Ce flux contient la liste des prochaines arrivées à l'arrêt désiré sur une ligne désiré. Ce flux se base sur les données théoriques et adaptées, et tient compte du temps réel s'il est disponible.

7.4.3 - 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	Туре	Format	Défaut	Description	Obligatoire
StopAreaExternalCode	String			Arrêt sur lequel porte la requête.	Oui
DestinationExternalCode	String			Filtre pour les prochains départs « ayant pour origine ».	Non
				Le filtre est actif pour toutes les gares traversées : ce n'est pas forcément l'origine du train, mais celle de l'usager	
LineExternalCode	String			Ligne sur laquelle porte la requête	Oui
Sens	Integer			Filtre du sens de la ligne utilisée	Non
RouteExternalCode	String			Route sur laquelle porte la requête	Oui
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 arrivée 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.
- A partir de l'interface 1.8.2 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)

7.5 - NextDeparture

7.5.1 - Appel

« /API?Action=NextDeparture »

7.5.2 - 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 et adaptées, et tient compte du temps réel s'il est disponible.

7.5.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
StopAreaExternalCode	String			Arrêt sur lequel porte la requête.	Oui
DestinationExternalCode	String			Filtre pour les prochains départs à « destination de ».	Non
				Le filtre est actif pour toutes les gares traversées : ce n'est pas forcément la destination finale du train, mais celle de l'usager	
LineExternalCode	String			Ligne sur laquelle porte la requête	Oui
Sens	Integer			Filtre du sens de la ligne utilisée	Non
RouteExternalCode	String			Route sur laquelle porte la requête	Oui
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	Туре	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.
- A partir de l'interface 1.8.2 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)

7.5.4 - Exemple

Exemple d'appel : sur un site fictif,

http://www.navitia.com/cgi-

bin/navitia.ashx/api?action=nextdeparture&lineidx=273&date=2006|05|02 &time=12|00&nbstop=4&stopareaidx=12952

>>>>>> Liste des API calculées tenant compte du temps réel

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionNextDepartureList:</p>
 - <Params Function="Navitia_Debug.DEBUG_VKAT/api">
     <action>nextdeparture</act
     </p
     <date>2007|11|05</date>
     <time>15|00</time>
     <stopareaidx>252</stopareaidx>
     <useadapted>false</useadapted>
   </Params>
   + <Stop StopIdx="250148" StopPointIdx="252" VehicleJourneyIdx="12730" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1"
   ValidityPatternSetCommentPos="0" StopOrder="3" VehicleIdx="1">
- <Stop StopIdx="255379" StopPointIdx="252" VehicleJourneyIdx="12733" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1" ValidityPatternSetCommentPos="0" StopOrder="3" VehicleIdx="1">
     - <StopTime>
        <TotalSeconds>56040</TotalSeconds>
         <Day>0</Day>
         <Hour>15</Hour>
         <Minute>34</Minute>
       </StopTime:
     <Coord />
       </StopPoint:
       <StopAreaIdx="252" StopAreaIdx="4900" StopAreaName="champigny" StopAreaExternalCode="OCE87758177" 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>
<VehicleJourneyIdx="12733" VehicleJourneyId="7547" VehicleJourneyName="81194" VehicleJourneyRouteIdx="2342"
        VehicleJourneyExternalCode="OCE1018-2" IsAdapted="0":
       - <Destination>
        - <StopArea StopAreaIdx="779" StopAreaId="4884" StopAreaName="la grande arche" StopAreaExternalCode="0CE87758011" 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>
         <Mode ModeIdx="14" ModeId="13" ModeName="RER A" ModeExternalCode="OCERER A" />
        <Company CompanyIdx="0" CompanyId="-1" CompanyName="" CompanyExternalCode="" />
<Vehicle VehicleIdx="1" VehicleId="0" VehicleName="" VehicleExternalCode="OCE1" MIPAccess="False" BikeAccepted="False" AirConditioned="False" />
         <EventPosList EventPosCount="0" />
       </VehicleJourney>
     </Stop>
   </NextDepartureList>
   <EventList EventCount="0" />
 </ActionNextDepartureList>
```

7.6 - PlanJourney

7.6.1 - Appel

« /API?Action=PlanJourney »

7.6.2 - Description

Ce flux contient la liste des trajets possibles en fonction des paramètres. Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
		Paramètres	principaux		
Departure	String	TypePoint IdxPoint Name CityName Number TypeName X Y StopPointIdx!Duration!Connect ionKind; StopPointIdx!Duration!Connect ionKind; StopPointIdx!Duration!Connect ionKind;		Point d'origine du trajet. (Cf 7.6.3 - Descriptions des paramètres de départ et d'arrivée)	Oui
DepartureExternalCode	String			Point d'origine du trajet sous forme de CodeExterne de zone d'arrêt. Utilisé uniquement si le paramètre Departure n'est pas renseigné.	Non
Arrival	String	TypePoint IdxPoint Name CityName Number TypeName X Y StopPointIdx!Duration!Connect ionKind; StopPointIdx!Duration!Connect ionKind; StopPointIdx!Duration!Connect ionKind;		Point de destination du trajet. (Cf 7.6.3 - Descriptions des paramètres de départ et d'arrivée)	Oui
ArrivalExternalCode	String			Point de destination du trajet sous forme de CodeExterne de zone d'arrêt. Utilisé uniquement si le paramètre Arrival n'est pas renseigné	Non

Nom	Туре	Format	Défaut	Description	Obligatoire
Sens	1 / -1		1	sens=1 correspond à "partir après"	Non
				sens=-1 correspond à "arriver avant"	
Time	String	Hour Minute	00 00	Heure de la recherche d'itinéraire	Non
Date	String	Year Month Day	Jour courant	Date de la recherche d'itinéraire	Non
Criteria	1/2/3		1	critère de recherche de calcul :	Non
				1: arrivée au plus tôt;	
				2: le moins de correspondance;	
				3: le moins de marche à pied	
NbBefore	Integer		0	Nombre de solutions à présenter avant la meilleure	Non
NbAfter	Integer		0	Nombre de solutions à présenter après la meilleure	Non
		Paramètres	de rabattement pi	éton	
WalkSpeed	Integer		50	Vitesse de marche à pied en m/min	Non
HangDistance	Integer		1000	Distance maximale d'accroche en mètre autour des points de départ et d'arrivée pour la marche à pied. Inutile si les 2 paramètres HangDistanceDep et HangDistanceArr sont renseignés	Non

Nom	Туре	Format	Défaut	Description	Obligatoire
HangDistanceDep	Integer		HangDistan ce ou 1000	Distance maximale d'accroche en mètre autour du point de départ pour la marche à pied	Non
HangDistanceArr	Integer		HangDistan ce ou 1000	Distance maximale d'accroche en mètre autour du point de destination pour la marche à pied	Non
		Paramètres de rabattemer	nt en Vélo Libre	e Service (VLS)	
ManageVLS	Integer	Integer En mètre : par exemple ManageVLS=2000 autorise 2km de VLS en rabattement au départ et à l'arrivée.	-1	-1 : l'utilisation de l'utilisation de Vélo en Libre Service (VLS) est désactivée	non
				>0 : Autorise NAViTiA à proposer des solution utilisant du VLS en début ou en fin de trajet dans un rayon limité à la valeur de « ManageVLS »	
TakeVLSDuration	Integer	En seconde	120	Temps de prise du VLS en station de départ	Non
DropVLSDuration	Integer	En seconde	60	Temps de dépose du VLS en station de d'arrivée	Non
BicycleSpeed	Integer	En m/minutes : par exemple 200 correspond à 12km/h		Vitesse moyenne à bicyclette	Non
		Paramètres de ra	abattement en	TAD	

Nom	Туре	Format	Défaut	Description	Obligatoire
UseODT	1 : TAD sur lignes virtuelles autorisé 2 : TAD sur lignes virtuelles	0 : pas de TAD 1 : TAD sur lignes virtuelles	1	Si 0, NAVITIA ne proposera pas d'itinéraire utilisant les transports à la demande (TAD).	Non
		autorisé 2 : TAD sur lignes virtuelles + TAD zonal autorisés		Si 1 : NAViTiA proposera des itinéraires utilisant des lignes virtuelles (d'arrêt à arrêt, avec horaires estimés ou garantis)	
	TAD zonal autorisés			Si 2 : NAViTiA pourra proposer des itinéraires utilisant du TAD zonal ou de rabattement sans horaires	
UseSchoolVehicle	1 : tout type de v	0 : pas de bus scolaire 1	1	Si 0 : NAViTiA ne proposera pas d'itinéraire utilisant les véhicules scolaires	Non
		1 : tout type de véhicule2 : Bus scolaire exclusivement		Si 1 : NAViTiA proposera des itinéraires utilisant les véhicules scolaires s'ils sont pertinents.	
				Si 2 : NAViTiA proposera des itinéraires n'utilisant que des véhicules scolaires (trajet avec carte étudiant par exemple)	
ODTTakeDuration	Integer	En seconde	120	Temps de prise en charge pour monter dans le véhicule qui effectue le TAD	non
ODTSpeed	Integer	En m/minutes : par exemple 1000 correspond à 60km/h	500	Valeur moyenne de la vitesse du véhicule qui effectue le TAD (souvent un taxi)	non
		Paramètres	d'accessibilité		

Nom	Туре	Format	Défaut	Description	Obligatoire
ModeType	Integer		, ,	Critère binaire de filtre sur les types de modes. La valeur à renseignée provient de la réponse à l'API :	Non
				Action=MakeBinaryCriteria&modetypeextern alcode=bus;metro;tramway par exemple	
Vehicle	Integer		Tout type de véhicule	Critère binaire de qualité de véhicule (à utiliser avec MakeBinaryCriteria)	Non
Equipment	Integer			Critère binaire de qualité de point d'arrêt (à utiliser avec MakeBinaryCriteria)	Non
ForbiddenStopAreaExtCode	String	Liste de codes externe de zones d'arrêts séparés par « ; »		Personnalisation des itinéraires : liste de codes externes de zone d'arrêt par lequel l'itinéraire NE DOIT PAS passer.	Non
				Permet de recalculer une feuille de route en évitant de passer par un arrêt ou une correspondance utilisant cet arrêt.	
ForbiddenLineExtCode	String	Liste de codes externe de lignes séparés par « ; »		Personnalisation des itinéraires : liste de codes externes de lignes par lequel l'itinéraire NE DOIT PAS passer	Non
				Permet de recalculer une feuille de route en évitant d'utiliser une ou plusieurs lignes	
ViaExternalCode	String			ExternalCode de la zone d'arrêt par lequel le résultat DOIT passer (trajet VIA)	Non

Nom	Туре	Format	Défaut	Description	Obligatoire
ViaConnectionDuration	Integer	Nombre de seconde		Durée de correspondance à appliquer sur le via. Si non précisée, NAVITIA applique les durées de correspondance définies dans les données sur l'arrêt de via	Non
ViaHangDistance	Integer		1000	Distance maximale de correspondance à pied sur l'arrêt de via.	Non
		Paramètres de	gestion des pertur	bations	
ManageDisrupt	Boolean		False	Indique si le calcul d'itinéraire doit tenir compte des évènements bloquants en indiquant un itinéraire de contournement	Non
UseAdapted	Boolean		True	Indique si le calcul doit utiliser des données adaptées	Non
		Paramètr	res complémentaire	es	
DateChangeTime	String	Hour Minute	00 00	Heure de changement de date (présentation en « calendrier TV »). Attention UseTransday doit être à vrai pour permettre le fonctionnement de ce paramètre (voir le Chef de projet Canal TP)	Non
MaxInterchange	integer		infini	Nombre maximal de correspondance à retourner. Par défaut NAViTiA est en mesure de calculer tout itinéraire sans limitation en nombre de correspondance.	Non
				0 : trajet en direct uniquement	
				1 : trajet en une correspondance	

Nom	Туре	Format	Défaut	Description	Obligatoire
ExtrapoleData	Boolean	0 : pas d'extrapolation1 : Extrapolation de recherche	False	Autorise NAViTiA à rechercher un itinéraire « type » même si la demande est en dehors des dates de production	Non
VPatternSetIdx	integer		0	Index de la période de recherche M d'itinéraire. Si <= 0 alors calcul sur la date	

7.6.3 - Descriptions des paramètres de départ et d'arrivée

7.6.3.1 Généralités

Les points de départ et d'arrivée sont des chaînes au format suivant :

- « TypePoint|IdxPoint|Name|CityName|Number|TypeName|X|Y|StopPointIdx!Duration!ConnectionKind;StopPointIdx!Duration!ConnectionKind;StopPointIdx!Duration!ConnectionKind; »
 - Les premiers champs représentent l'objet de départ ou de destination. Ces champs se trouvent, par exemple, dans la première balise du flux de réponse à un appel EntryPoint :

<EntryPoint EntryPointType="StopArea" EntryPointResponseQuality="100" CityName="Nantes" Number="" TypeName="" EntryPointName="COMMERCE">

<StopArea StopAreaIdx="2636" ...

...

</EntryPoint>

Chaine de paramètre d'appel à PlanJourney	Flux EntryPoint	Description	Obligatoire
TypePoint	EntryPointType	Type de point d'entrée: StopArea Site Address City Undefined	Oui Undefined permet d'omettre les paramètres suivants.
IdxPoint	StopAreaIdx, CityIdx, AddressIdx, SiteIdx selon le type de réponse	Identifiant unique du point d'entrée	Non si TypePoint=Undefined
Name	EntryPointName	Nom du point de d'entrée	Non si TypePoint=Undefined
CityName	CityName	Nom de la commune du point d'entrée	Non si TypePoint=Undefined
Number	Number	Numéro de voierie du point d'entrée	Non
TypeName	TypeName	Type de voierie du point d'entrée	Non

Chaine de paramètre d'appel à PlanJourney	Flux EntryPoint	Description	Obligatoire
X	Coord/CoordX	Coordonnées X du point d'entrée	Oui
Υ	Coord/CoordY	Coordonnées Y du point d'entrée	Oui

7.6.3.2 Accroche des point de départ et d'arrivée en X,Y

Si les champs X et Y sont renseignés, le calcul se fera en partant de ces coordonnées géographiques → c'est le mode de fonctionnement conseillé. La fabrication de la chaine d'appel est alors très simple :

departure=TypePoint|IdxPoint|Name|CityName|Number|TypeName|X|Y

Il est possible d'omettre les premiers champs en indiquant le type "undefined" à NAVITIA:

departure=Undefined||nom libre|commune libre|||X|Y

Il est également possible de forcer NAViTiA à proposer une première section typée.

departure=Undefined|||||X|Y

NAVITIA proposera alors le point des coordonnées X,Y demandées comme le point de départ et/ou arrivé.

departure=Address|||||X|Y

NAViTiA proposera alors la rue la plus proche des coordonnées X,Y demandées

departure=StopArea|||||X|Y

- NAVITIA proposera alors la zone d'arrêt la plus proche des coordonnées X,Y demandées
- Le paramètre "HangDistance" des API de calcul d'itinéraire correspond à la distance maximale de marche à pied au départ (ou à l'arrivée). A partir de cette distance, NAVITIA récupère donc l'ensemble des points d'arrêts accessible en moins de "hangdistance" mètre en suivant le filaire de voirie. Cependant, sur ces arrêts NAVITIA ajoute l'ensemble des correspondances afin de simplifier la présentation des trajets du type :
 - « Marche à pied, puis correspondance, puis Bus »
 - Qui devient « Marche à pied puis Bus ».

En conséquence, la marche à pied au départ (ou à l'arrivée) peut parfois dépasser la Hangdistance.

7.6.3.3 Cas particulier de la recherche depuis ou vers une commune

Remarque : si la recherche est faite depuis ou vers une commune :

- NAVITIA utilisera en priorité les arrêts rattachés à cette commune, la première section utilisant alors le mode de rattachement défini en base.
- Si aucun arrêt n'est rattaché à cette commune, NAViTiA utilisera alors les arrêts principaux de cette commune
- Si aucun arrêt n'est défini comme principal sur cette commune, NAViTiA utilisera les coordonnées X,Y de la commune

Dans les 2 derniers cas (liaison floue depuis une commune vers le premier point d'arrêt), le type de la première section sera « LinkConnection ».

7.6.3.4 Cas particulier d'un appel en forçant les points d'arrêts de départ ou de destination

Si les coordonnées X Y ne sont pas renseignées, le calculateur passe en mode « Hang » :

- le calculateur va utiliser le dernier champ multiple des paramètres Departure ou Arrival : « StopPointIdx!Duration!ConnectionKind; ».
- Ce champ est composé pour le point de départ : de chaque « arrêt physique potentiel de départ ! durée d'accroche avec le point théorique de départ ! Mode d'approche utilisé » chaque trigramme étant séparé par « ; » du couple suivant.
- La valeur de « Duration » est en seconde.
- La valeur de ConnectionKind (mode d'approche utilisé pour rejoindre le point d'arrêt) est à récupérer depuis l'API EntryPoint. 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

DESTRUITATIONListe des API calculées tenant compte du temps réel

- 8 : PersonnalCarConnection → 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. Voir chapitre 7.6.7 - Gestion des transports à la demande (ODT : OnDemandTransport)
- 13 : DefaultConnection → Le type de liaison s'adapte au type de données
- 14 : End
- Lorsque l'on appelle PlanJourney avec un mode d'approche au départ et/ou à l'arrivée (en indiquant pour chaque « hang » le mode d'approche via le paramètre « ConnectionKind »), la réponse retourne les informations suivantes :
 - La <u>première</u> section du voyage utilise le ConnectionKind spécifié dans le « hang » du point de départ lors de l'appel à PlanJourney
 - La <u>dernière</u> section du voyage utilise le ConnectionKind spécifié dans le « hang » du point d'arrivée lors de l'appel à PlanJourney

7.6.4 - Autres paramètres

7.6.4.1 VpatternIdx

Si VpatternIdx a une valeur inférieure ou égale à 0 ou n'est pas renseigné alors la recherche s'effectue sur la date. La date doit avoir le format « yyyy|mm|dd »

7.6.4.2 Criteria

Critère de recherche d'itinéraires. Les valeurs possibles sont :

- 1 : arrivée au plus tôt
- 2 : le moins de correspondances
- 3 : le moins de temps de marche à pied

7.6.4.3 Autres critères

Vehicle, Equipment, ModeType: critères binaires à construire grâce à l'API « MakeBinaryCriteria »

7.6.5 - Génération des appels « détail de la feuille de route », « Horaires suivants/précédents »

La fabrication des URL de calcul suivant et précédant (« horaire précédent » et « horaire suivant ») se fait grâce au contenu de la balise « Summary/Call ». Il est important de prendre en compte tous les critères fournis : Sens / Year / Month / Day / Hour / Minute / Criteria afin de laisser NAViTiA gérer ces calculs. Ne pas utiliser ces balises expose l'intégration à des anomalies de calculs :

- autour de minuit
- pour les calculs dont le critère (arrivée au plus tôt, moins de correspondance...) n'a pas été trouvé

Afin de générer

- l'appel aux horaires précédents, il suffit d'utiliser la balise Summary/Call/Before.
- Afin de générer l'appel aux horaires suivants, il faut utiliser le contenu de la balise Summary/Call/After
- Afin de rappeler l'itinéraire courant (cas du détail d'un itinéraire par exemple), utiliser le contenu de la balise Summary/Call/This

Remarque:

En sortie de calcul, les balises IsFirstSoluce et IsLastSoluce peuvent être mal renseignées par NAViTiA :

- IsFirstSoluce n'est pas bien renseigné si la solution trouvée est la première de la journée sans aucune violation de paramètre. Alors un calcul sur « horaire précédent » permettra de retrouver cette solution avec la balise IsFirstSoluce bien renseignée.
- IsLastSoluce n'est pas bien renseigné si la solution trouvée est la dernière de la journée sans aucune violation de paramètre. Alors un calcul sur « horaire suivant » permettra de retrouver cette même solution avec la balise IsLastSoluce bien renseignée.

7.6.6 - Cas de non solution d'itinéraire

La liste des causes de non solution d'itinéraire est détaillée dans les xsd liées au format des flux utilisés. Il est impératif de se référer au chapitre 4.1.2 - Identification du type de réponse afin de présenter une réponse correcte au client.

Voici la liste des principales causes :

- plan_no_solution
 - L'itinéraire n'est pas possible en transport en commun.
 - Proposer au voyageur de lancer une nouvelle recherche.
- plan_date_out_of_bounds
 - La date de recherche est hors de la période de validité des données.

- Proposer au voyageur de lancer une nouvelle recherche sur une date compatible avec les dates de validité de la base.
- "plan_departure_destination_too_close"
 - Les points de départ et de destination sont trop proches pour lancer un calcul d'itinéraire.
 - Proposer au voyageur de faire le trajet à pied.
- plan_no_solution_limited_interchange
 - L'appel à NAVITIA a utilisé le paramètre MaxInterchange (nombre maximum de correspondance) en entrée de calcul d'itinéraire. L'itinéraire n'est pas possible avec cette limitation, cependant NAVITIA a détecté que l'itinéraire est faisable en plus de correspondance.
- plan_no_departurecode
 - L'appel à NAVITIA a utilisé le paramètre departureCode en entrée. NAVITIA n'a pas trouvé de zone d'arrêt correspondante.
- plan_no_arrivalcode
 - L'appel à NAViTiA a utilisé le paramètre arrivalCode en entrée. NAViTiA n'a pas trouvé de zone d'arrêt correspondante.
- plan_no_dep_and_dest_code
 - L'appel à NAVITIA a utilisé le paramètre arrivalCode et departureCode en entrée. NAVITIA n'a pas trouvé les zones d'arrêts correspondantes.
- plan_connection_problem,
 plan_reader_error
 - Cas d'erreur technique ou d'incohérence de données NAViTiA
 - Proposer de changer les paramètres de données ou de renouveler la demande ultérieurement
- plan_no_departure_stoppoint,
 plan_no_destination_stoppoint
 et plan_no_dep_and_dest_stoppoint
 - Ce cas se produit essentiellement lors de calcul d'itinéraire sur des zones peu desservies. NAVITIA ne trouve pas de point d'arrêt dans une zone délimité par les paramètres « HangDistanceDep » et « HangDistanceArr ». Le rabattement entre le point de départ ou d'arrivée et le réseau de transport en commun ne peut donc se faire à pied.

7.6.7 - Gestion des transports à la demande (ODT: OnDemandTransport)

Lorsque l'utilisation du paramètre « &UseODT » vaut 2, NAVITIA peut remonter des itinéraires utilisant des transports à la demande. Ces itinéraires peuvent avoir des structures particulières, il faut donc tenir compte des règles suivantes :

- Une section « ODTConnection » définit un segment de trajet en transport à la demande
- L'attribut « ODTPos » des nœuds « Departure » et « Arrival » permet de récupérer des informations complémentaires dans la liste « ODTList » en fin de flux. La valeur de ODTPos représente la position du service TAD dans cette liste à partir de « 0 »
- Les horaires peuvent être estimés (attribut EstimatedTime à 1) : dans ce cas il est préférable d'indiquer des durées de trajet et non les horaires
- Les points de montée et de descente peuvent être des StopPoint ou des **adresses** selon le type de TAD :
 - TAD de desserte locale : le véhicule prend en charge depuis une adresse vers une adresse. La structure du flux de réponse NAViTiA est alors la suivante :
 - Une section unique de voyage de type ODTConnection
 - « Departure » est de type « Address » et l'attribut EstimatedTime vaut 1
 - o « Arrival » est de type « Address » et l'attribut EstimatedTime vaut 1
 - La ligne est détaillée dans le nœud « Line ». Des informations complémentaires sont présentes dans les informations ODT en fin de flux
 - TAD de rabattement depuis ou vers une adresse : le véhicule prend en charge depuis une adresse. La structure du flux de réponse NAViTiA est alors la suivante :
 - Une section de voyage de type ODTConnection en début de voyage (convergence)
 - « Departure » est de type « Address » et l'attribut EstimatedTime vaut 1
 - « Arrival » est de type « StopPoint » et l'attribut EstimatedTime vaut
 - La ligne est détaillée dans le nœud « Line ». Des informations complémentaires sont présentes dans les informations ODT en fin de flux
 - La suite du trajet est standard

- TAD de rabattement depuis ou vers un arrêt : le véhicule prend en charge depuis un arrêt dit « virtuel ». La structure du flux de réponse NAViTiA est alors la suivante :
 - Une section de voyage standard pour rejoindre l'arrêt virtuel à pied (LinkConnection, AddressConnection...)
 - Une section de voyage de type ODTConnection en début de voyage (convergence)
 - « Departure » est de type « Address » et l'attribut EstimatedTime vaut 1
 - « Arrival » est de type « StopPoint » et l'attribut EstimatedTime vaut
 0 ou 1 selon que l'horaire d'arrivée est garanti.
 - La desserte et la ligne sont détaillées dans le nœud « Line ». Des informations complémentaires sont présentes dans les informations ODT en fin de flux
 - La suite du trajet est standard

7.6.8 - Exemple

Pour une meilleure lisibilité, le flux d'exemple est tronçonné. Les nœuds précédés d'un « + » ont été repliés.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ActionJourneyResultList
   - <Params Function="Navitia_Debug.DEBUG_VKAT/API">
         <action>planjourney<
         <departure>StopArea|108|Gare SNCF|Brest|||0,00|0,00|210!0!13;211!0!13;</departure>
          <arrival>StopArea|642|Gare SNCF|Nantes|||0,00|0,00|1333!0!13;1332!0!13;</arrival>
         <date>2007[6]2</date>
          <time>15 | 43</time:
         <VPatternSetIdx>0</VPatternSetIdx>
<sens>1</sens>
          <WalkSpeed>50</WalkSpeed>
         <criteria>1</criteria>
<Vehicle>0</Vehicle>
          <nbbefore>1</nbbefore:
          <nbafter>1</nbafter>
          <Equipment>0</Equipment>
         <Mode>254</Mode:
         <HangDistance>1000</HangDistance>
          <ManageDisrupt>0</ManageDisrupt>
     </Params>
 - <JourneyResultList JourneyResultCount="3">
    + <_JourneyResult SectionCount="3" Criteria="AsSoonAsPossible" IsCriteriaFound="1" IsBest="0" JourneyResultPosition="1" IsFirstSoluce="0" IsLastSoluce="1" IsDisrupt="0"
             IsAdapted="1":
     + <Journey/Result SectionCount="3" Criteria="AsSoonAsPossible" IsCriteriaFound="1" IsBest="1" Journey/ResultPosition="2" IsFirstSoluce="1" IsLastSoluce="0" IsDisrupt="0"
    IsAdapted="1"> + <JourneyResult SectionCount="3" Criteria="AsSoonAsPossible" IsCriteriaFound="1" IsBest="0" JourneyResultPosition="3" IsFirstSoluce="0" IsLastSoluce="0" IslastS
             IsAdapted="1":
     </JournevResultList>
     <CommentList CommentCount="0" />
     <ODTList ODTCount="0" /
     <EventList EventCount="0" />
 </ActionJourneyResultList>
```

Développement du nœud "JourneyResult" :

>>>>>> Liste des API calculées tenant compte du temps réel

Développement du nœud "Summary" :

Note : les structures des nœuds « DepartureTime » et « ArrivalTime » sont identiques, de mêmes pour les nœuds « Before », « This » et « After ».

```
- <Summary Interchange="1" JourneyKey="Nav5507;3;TER;Nav5451;Nav5451;3;TER;Nav5518;">
 + <DepartureDate>
 + <DepartureTime>
 + <ArrivalDate>
 + <ArrivalTime>
  - <Duration>
     <TotalSeconds>18660</TotalSeconds>
     <Day>0</Day>
     <Hour>05</Hour>
     <Minute>11</Minute>
   </Duration>
 - <TotalLinkTime>
     <TotalSeconds>0</TotalSeconds>
     <Day>0</Day>
     <Hour>00</Hour>
     <Minute>00</Minute>
   </TotalLinkTime>
  - <TotalWaitTime>
     <TotalSeconds>4860</TotalSeconds>
     <Day>0</Day>
     <Hour>01</Hour>
     <Minute>21</Minute>
   </TotalWaitTime>
 - <Call>
   + <Before>
   - <This>
       <Sens>-1</Sens>
       <Year>2007</Year>
       <Month>06</Month>
       <Day>02</Day>
       <Hour>19</Hour>
       <Minute>50</Minute>
       <Criteria>1</Criteria>
     </This>
   + <After>
   </Call>
  </Summary>
```

Développement du nœud "Section" :

Note : le nœud « Duration » a la même structure que celui ce trouvant dans le nœud « Summary » ci-dessus. Les nœuds « Departure » et « Arrival » ont aussi la même structure.

Tous les nœuds « Section » dont l'attribut « type » est différent de «VehicleJourneyConnection» ou « ProlongationConnection » ne contiennent plus de nœud «VehicleJourney» à partir de l'Interface 1_15. Ce nœud n'était pas utile dans ce cas, il a été retiré afin d'alléger le flux XML.

Les types de section possibles sont :

AddressConnection

Marche à pied entre un point d'arrêt du réseau de transport en commun et une adresse.

SiteConnection

Marche à pied entre un point d'arrêt du réseau de transport en commun et un lieu remarquable.

StopAreaConnection

Marche à pied entre un point d'arrêt du réseau de transport en commun et une zone d'arrêt.

VehicleJourneyConnection

Utilisation d'un véhicule de transport en commun.

LinkConnection

Marche à pied entre deux arrêts physique.

StopPointConnection

Correspondance au sein d'un point d'arrêt.

WalkConnection

Cette section est particulière : elle compose à elle seule l'ensemble d'un itinéraire. L'itinéraire est alors entièrement en marche à pied. Attention à **ne pas appeler le détail d'un itinéraire** dans ce cas. Le résultat serait potentiellement différent. Le détail de marche à pied pourra être obtenu via l'API StreetNetwork.

ProlongationConnection

Attente d'un véhicule de transport en commun. Le passager doit attendre dans ce véhicule

PersonnalCarConnection

Utilisation d'un véhicule particulier.

Undefined

Section entre des coordonnées géographiques sans références à des données NAVITIA et le premier point d'arrêt.

Section entre le dernier point d'arrêt et des coordonnées géographique sans références à des données NAViTiA

BicycleConnection

Utilisation d'un mode doux : bicyclette, roller....

CabConnection

Utilisation d'un taxi

ODTConnection

Utilisation d'un Transport à la demande

DefaultConnection

Test NAViTiA

End

Test NAViTiA

>> Longueur de trajet et émissions carbone

A partir de l'interface 1.16, chaque section contient l'attribut SectionLength correspondant à la longueur en mètres de la section parcourue.

Il est possible d'utiliser cette valeur pour calculer les émissions de carbone liées à un trajet. Cependant les émissions changent selon le mode, le type de carburant, l'exploitation et le remplissage des véhicules. A titre d'information, voici les chiffres moyens publiés par l'ADEME dans son « Guide des facteurs d'émission » version 6.1 :

• Voiture: 189gCO₂/km/voiture

Autobus Île-de-France : 67gCO₂/km/passager

• Autobus province: 85gCO₂/km/passager

Autocar: 34gCO₂/km/passager

TGV: 22gCO₂/km/passager

• TER: 59gCO₂/km/passager

Métro, RER, Tramway : 6gCO₂/km/passager

NB : les émissions sont parfois données en équivalent carbone $(g_{eq}C)$ ou en équivalent CO_2 . Ces deux unités ne doivent pas être confondues : $1g_{eq}C=3,67g_{eq}CO_2$

DESTRUITATIONListe des API calculées tenant compte du temps réel

```
- <Section Type="VehicleJourneyConnection">
 + < Duration>
 - <Departure Type="StopPoint" ODTIdx="0" ODTPos="-1" StopIdx="127467" CommentPos="-1">
   + <Date>
   + <Time>
     <Coord />
   + <StopPoint StopPointIdx="211" StopPointId="5507" StopPointName="Gare SNCF" StopPointExternalCode="Nav5507" FareZone="0">
   </Departure>
 + <Arrival Type="StopPoint" ODTIdx="0" ODTPos="-1" StopIdx="163964" CommentPos="-1">
 + <VehicleJourney VehicleJourneyIdx="11529" VehicleJourneyId="15" VehicleJourneyName="855826" VehicleJourneyRouteIdx="245"
     VehicleJourneyExternalCode="1031-15" IsAdapted="0">
   <Nota NotaType=""/>
 - <FareSectionList FareSectionCount="1">
     <FareSection>0</FareSection>
   </FareSectionList>
  - <FareZoneList FareZoneCount="1">
     <FareZone>0</FareZone>
   </FareZoneList>
 </Section>
```

Développement du nœud "StopPoint" :

Note : le nœud « City » a déjà été développé dans le flus « CityList ». Le nœud « Comment » a déjà été développé dans d'autres flux (par exemple : « SiteList »)

Développement du nœud "VehicleJourney" :

Note: les nœuds « Line » et « StopArea » ont été développés respectivement dans « LineList » et « StopAreaList ». La propriété «IsExtrapolated=0 ou 1» ajoutée dans le nœud "VehicleJourney" à partir de l'Interface 1_15, permet de qualifier la qualité d'un itinéraire. En effet, cette propriété permet de savoir si les données de cette desserte sont extrapolées, ou validées par le transporteur.

DESTRUITATIONListe des API calculées tenant compte du temps réel

```
< VehicleJourney VehicleJourneyIdx="11529" VehicleJourneyId="15" VehicleJourneyName="855826" VehicleJourneyRouteIdx="245"
  VehicleJourneyExternalCode="1031-15" IsAdapted="0">
 <Route RouteIdx="245" RouteId="1031" RouteName="Brest vers Rennes" RouteExternalCode="TER-87474007-87471003" IsForward="0" RouteLineIdx="127"
  IsFrequence="0" IsAdapted="0"> - <Line LineIdx="127" LineId="668" LineName="Rennes - Brest" LineCode="TER" LineExternalCode="TER-87471003-87" LineAdditionalData=""
      SortOrder="127" LineColor="" HasAdaptedRoute="0">
      <Mode ModeIdx="5" ModeId="21" ModeName="TER" ModeExternalCode="TER" />
    - <Network NetworkIdx="5" NetworkId="3" NetworkName="TER" NetworkExternalCode="3">
        <EventPosList EventPosCount="0" />
      </Network>
      <Forward ForwardName="rennes gare sncf - brest gare sncf" />
      <Backward BackwardName="brest gare sncf - rennes gare sncf" />
      <EventPosList EventPosCount="0" />
    </Line>
  </Route>
- < Destination>
  - <StopArea StopAreaIdx="886" StopAreaId="1172" StopAreaName="Gare SNCF" StopAreaExternalCode="1172" MainStopArea="0" MultiModal="0"
    CarPark="0" MainConnection="0" AdditionalData="">
- <City CityIdx="256" CityId="27" CityName="Rennes" CityExternalCode="35238" CityCode="">
        <Country CountryIdx="1" CountryId="0" CountryName="France" CountryExternalCode="FRA" />
      </City>
      <Coord />
    </StopArea>
  </Destination>
  </mode ModeIdx="5" ModeId="21" ModeName="TER" ModeExternalCode="TER" />
<Company CompanyIdx="0" CompanyId="-1" CompanyName="" CompanyExternalCode="" />
<Vehicle VehicleIdx="1" VehicleId="1" VehicleName="" VehicleExternalCode="1" MIPAccess="False" BikeAccepted="False" AirConditioned="False" />
  <EventPosList EventPosCount="0" />
</VehicleJourney>
```

7.7 - DetailedJourney

Les paramètres d'appels sont identiques, hormis les paramètres « NbBefore » et « NbAfter » supprimés.

Le flux résultat diffère :

- sur le fait qu'il n'y a qu'un seul résultat
- sur le fait que le calcul itinéraire à pied est désactivée : DetailedJourney propose uniquement des itinéraires utilisant au moins une section de transport en commun. Ainsi :
 - Si un itinéraire de l'API « PlanJourney » est du type « marche à pied » uniquement (WalkConnection), alors il convient de ne pas appeler l'API DetailedJourney. On pourra dans le détail d'un tel itinéraire
 - soit afficher « rejoindre à pied »,
 - soit faire un appel à l'API 97 StreetNetwork
 - Si un itinéraire de l'API « PlanJourney » est du type « vélo » ou « VLS » uniquement (BicycleConnection, entouré éventuellement de LinkConnection dans le cas d'un VLS), alors il convient de ne pas appeler l'API DetailedJourney. On pourra dans le détail d'un tel itinéraire :
 - soit afficher les points de départ et d'arrivée sur la carte,
 - soit se brancher sur le service d'un prestataire dédié à l'information vélo

DESTRUCTIONListe des API calculées tenant compte du temps réel

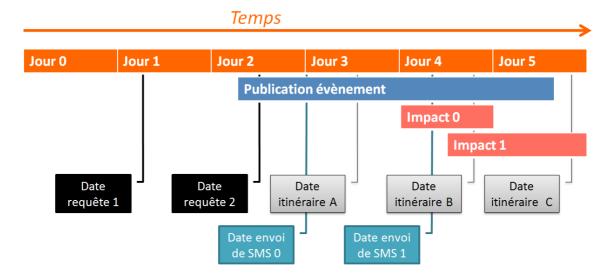
```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionJourneyResultList>
  - <Params Function="Navitia_Debug.DEBUG_VKAT/API">
       <action>detailedjourney</action>
<departure>StopArea|108|Gare SNCF|Brest|||0,00|0,00|210!0!13;211!0!13;</departure>
<arrival>StopArea|642|Gare SNCF|Nantes|||0,00|0,00|1333!0!13;1332!0!13;</arrival>
       <date>2007|6|2</date>
       <time>15 | 43</time>
       <\!\!\mathsf{VPatternSetIdx}\!\!>\!\!\mathbf{0}\!<\!\!/\mathsf{VPatternSetIdx}\!\!>
       <sens>1</sens>
<WalkSpeed>50</WalkSpeed>
       <criteria>1</criteria>
<Vehicle>0</Vehicle>
       <Equipment>0</Equipment>
       <Mode>254</Mode>
<HangDistance>1000</HangDistance>
       <ManageDisrupt>0</ManageDisrupt>
  </Params>
- <JourneyResultList JourneyResultCount="1">
    + <JourneyResult SectionCount="3" Criteria="AsSoonAsPossible" IsCriteriaFound="1" IsBest="1" JourneyResultPosition="1" IsFirstSoluce="0" IsLastSoluce="0" IsDisrupt="0" IsAdapted="1"> </JourneyResultList>
    <CommentList CommentCount="0" />
<ODTList ODTCount="0" />
<EventList EventCount="0" />
  </ActionJourneyResultList>
```

8 - LISTE DES API DÉDIÉES AU TEMPS RÉEL

8.1 - Introduction

8.1.1 - Utilisation des différentes dates disponibles

Un évènement possède des dates de publication qui permettent de contrôler la « date à partir de » et la « date jusqu'à laquelle » on désire communiquer autour de cet évènement. Ainsi, une perturbation prévue peut être renseignée par le système dès que le réseau de transport en a pris connaissance.



A partir du diagramme ci-dessus, voici les différents comportements du système :

Date de la requête	Date de l'itinéraire	Information NAViTiA	Itinéraire de contournement
Date requête 1	Date itinéraire A, B ou C	NAViTiA n'informe pas sur l'évènement	n/a
Date requête 2	Date itinéraire A	NAViTiA n'informe pas sur l'évènement puisqu'aucun impact n'est actif	n/a
Date requête 2	Date itinéraire B	NAViTiA informe sur l'impact 0 et sur l'impact 1, tous deux actifs	Gestion de l'ensemble des impacts actifs à t=date itinéraire B
Date requête 2	Date itinéraire C	NAViTiA n'informe plus sur l'évènement	n/a

Les dates d'envoi de Mail/SMS sont également paramétrables par impact afin de communiquer au plus près de la perturbation

A partir du diagramme ci-dessus, voici les différents comportements du système pour l'API GetEventList:

Date de la requête	Paramètre "Date consultée"	Information NAViTiA
Date requête 1	Pas de date	NAViTiA n'informe pas sur l'évènement (l'évènement n'est pas publiable)
Date requête 2	Pas de date	NAViTiA informe sur l'évènement parce que l'évènement est actif à la date requête 2.
Date requête 2	PublicationStartDate=Date itinéraire A	NAViTiA informe sur l'évènement parce que PublicationEndDate de l'évènement est supérieur à la date itinéraire A
Date requête 2	PublicationEndDate=Date itinéraire A	NAViTiA informe sur l'évènement parce que PublicationStartDate de l'évènement est inférieur à la date itinéraire A
Date requête 2	PublicationStartDate=Date itinéraire A PublicationEndDate=Date itinéraire B	NAViTiA informe sur l'évènement parce que PublicationStartDate de l'évènement est inférieur à la date itinéraire B et PublicationEndDate est supérieur à la date itinéraire A.

A partir du diagramme ci-dessus, voici les différents comportements du système pour l'API GetImpactList:

Date de la requête	Paramètre "Date consultée"	Information NAViTiA
Date requête 1	Pas de date	NAViTiA n'informe pas sur l'évènement et ses impacts parce que l'évènement n'est pas publiable.
Date requête 2	Pas de date	NAViTiA informe sur l'évènement et ses impacts parce que l'évènement est actif à la date requête 2.
Date requête 2	StartDate=Date itinéraire A	NAViTiA informe sur tous les impacts de l'évènement parce que ImpactEndDate est supérieur à la date itinéraire A.

Date de la requête	Paramètre "Date consultée"	Information NAViTiA
Date requête 2	StartDate=Date itinéraire A	NAViTiA informe sur tous les impacts parce qu'ils sont actif
	EndDate=Date itinéraire B	entre les date A et B.

Rappel: les flux ci-dessous sont accessibles en requêtant NAViTiA à l'aide d'une URI du type : http://SERVEUR/Navitia.dll/Event?action=xxx.

Pour la nomenclature de ces API, se reporter à la documentation de référence d'AlerteTrafic. Ces API ne fonctionnent qu'à partir de l'Interface 1_11

8.1.2 - Limitations

Etant donnée la fréquence de mise à jour et la volumétrie de ces données, il n'est pas prévu de « pager » sur ces données.

- Il convient de choisir convenablement ses filtres afin préciser les données à afficher en amont.
- Plutôt que de demander l'ensemble des évènements recensés par le système, il vaut demander les impacts sur un certain type de données, ou pour une plage de date ou d'horaire donnée

8.2 - GetImpactlist

8.2.1 - Appel

« /Event?Action=GetImpactList »

8.2.2 - Exemple d'appel

8.2.3 - Description

Ce flux liste les impacts « temps réels » sur les objets de transport en commun référencés par NAVITIA.

8.2.4 - Tableau des paramètres de l'API

Nom	Type	Format	Défaut	Description	Obligatoire

DescriptionListe des API dédiées au temps réel

Nom	Туре	Format	Défaut	Description	Obligatoire
StartDate	string	yyyy mm dd hh nn ss		Renvoie les impacts actifs à partir de la date passée en paramètre	Non
				Si EndDate est précisé, NAViTiA renvoie tous les évènements dont une partie au moins est active entre ces 2 dates	
EndDate	String	yyyy mm dd hh nn ss		Renvoie les impacts actifs avant la date passée en paramètre	Non
				Si StartDate est précisé, NAViTiA renvoie tous les évènements dont une partie au moins est active entre ces 2 dates	
State	énuméré	1 parmi:	All	Permet filtrer les réponses sur les états d'impacts	
		DisruptWarning		Paramètre sous forme de	
		Informatio		liste séparé par «;»	
		n			
		Delay			
IsActive	Boolean	0 ou 1	0	IsActive=1 permet de ne retourner que les impacts actifs entre StartDate et EndDate, en tenant compte des propriétés :	
				• DailyStartTime	
				 DailyEndTime 	
				 ImpactActiveDays 	
				 et dont l'état est actif (state différent de none) 	

Nom	Туре	Format	Défaut	Description	Obligatoire
TCOType	énuméré	1 parmi: StopArea VehicleJou rney Line Network StopPoint RoutePoin t	All	Permet filtrer les réponses sur les types d'objets impactés	Non
TCOExtCo de	Liste séparé par « ; »			Permet filtrer les réponses sur une liste d'objet TC Attention le paramètre	Non
				TCOType est obligatoire lors de l'utilisation du paramètre TCOExtCode, sinon ce dernier est ignoré	

8.2.5 - Exemple de l'utilisation des filtres sur date.

- Appel avec StartDate uniquement : un appel avec le paramètre « &startdate=2010|02|28|10|30|15 » renvoie tous les impacts dont ImpactEndDate est supérieur ou égal à la date et heure passée en paramètre.
- Appel avec EndDate uniquement : un appel avec le paramètre « &enddate=2010|03|01|20|30|15 » renvoie tous les impacts dont ImpactStartDate est inférieur ou égal à la date et heure passée en paramètre.
- Appel avec StartDate et EndDate conjointement : un appel avec les paramètres « &startdate=2010|02|28|10|30|15 &enddate=2010|03|01|20|30|15 » renvoie tous les impacts qui sont valides au moins une fois entre ces deux dates (les impacts dont ImpactStartDate est inférieur ou égal à EndDate passé en paramètre et ImpactEndDate supérieur ou égal à la StartDate du paramètre).
- Utilisation de IsActive :
 - si IsActive=1, les filtres sont plus précis (gestion des DailyStartTime, DailyEndTime, ImpactActiveDays et dont l'état est actif (state différent de none)
 - Par exemple: &startdate=2010|02|28|10|30|15 &enddate=2010|03|01|20|30|15&IsActive=1 » renvoie tous les impacts qui sont **actifs** entre startDate et EndDate, en particulier dont la plage horaire de [dailyStartTime, DailyEndTime] superpose une partie de la

plage date/horaire définie par [startdate, EndDate] et dont au moins un jour de l'impact est actif dans cette plage de date/horaire

Le mode d'utilisation habituel de cette API (Type page « état du trafic » : permettant de retourner tous les impacts actifs à l'heure courante) est le suivant :

- &startdate= heure courante
- &enddate= heure courante
- &IsActive=1

8.3 - Eventuellement un filtre sur les TCOTypegeteventlist

8.3.1 - Appel

« /Event?Action=GetEventList »

8.3.2 - Exemple d'appel

http://serveur/navitia.dll/event?action=geteventlist&publicationstartdate=2 009%7C05%7C30&publicationenddate=&interface=1 11

8.3.3 - Description

Ce flux liste les évènements « temps réels » référencés par NAViTiA.

8.3.4 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
Publicatio nStartDat e	string	yyyy mm dd hh nn ss		Renvoie les évènements actifs à partir de la date passée en paramètre	Non
				Si PublicationEndDate est précisé, NAViTiA renvoi tous les évènements actifs compris entre ces 2 dates	

Nom	Туре	Format	Défaut	Description	Obligatoire
Publicatio nEndDate	string	yyyy mm dd hh nn ss		Renvoie les impacts actifs avant la date passée en paramètre	Non
				Si PublicationStartDate est renseigné, NAViTiA renvoi les impacts actifs sur la période comprise entre PublicationStartDate et PublicationEndDate	

8.4 - getimpact

8.4.1 - Appel

« /Event?Action=GetImpact »

8.4.2 - Exemple d'appel

http://serveur/navitia.dll/event?action=getimpact&impactid=20&interface= 1 11

8.4.3 - Description

Ce flux détaille les propriétés d'un impact sur un objet de transport en commun référencé par NAVITIA.

8.4.4 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
ImpactID	integer			Renvoie que l'impact don identifiant est passé er paramètre et actif à ce jour	

8.5 - getevent

8.5.1 - Appel

« /Event?Action=GetEvent »

8.5.2 - Exemple d'appel

http://serveur/navitia.dll/event?action=getevent&Eventid=4&interface=1 1 1

8.5.3 - Description

Ce flux détaille les propriétés d'un évènement « temps réels » référencé par NAVITIA.

8.5.4 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
EventID	integer			Renvoie que l'évènement dont identifiant est passé en paramètre et actif à ce jour	Oui

8.6 - getmedialist

8.6.1 - Appel

« /Event?Action=GetMediaList »

8.6.2 - Exemple d'appel

http://serveur/navitia.dll/event?action=getmedialist&interface=1 11

8.6.3 - Description

Ce flux liste les medias de diffusion connus dans la base d'AlerteTrafic.

8.7 - getmessage

8.7.1 - Appel

« /Event?Action=GetMessage »

8.7.2 - Exemple d'appel

http://serveur/navitia.dll/event?action=getmessage&impactid=16&mediaid =1&interface=1 11

8.7.3 - Description

Ce flux permet d'obtenir le message d'un impact associé à un média de la base AlerteTrafic. Ce message ne peut faire plus de 1024 caractères.

8.7.4 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
ImpactID	Integer			Renvoie que le message de l'impact dont identifiant est passé en paramètre	Oui
MediaID	integer			Renvoie que le message De l'impact avec l'identifiant du media passé en paramètre	Oui

9 - LISTE DES API TARIFAIRES

9.1 - JourneyFare

9.1.1 - Appel

« /Fare?Action=JourneyFare »

9.1.2 - Description

Ce flux retourne les tarifs qui composent un itinéraire multimodal. Attention, l'itinéraire multimodal est composé de plusieurs segment de voyages, et certain tarif sont applicables sur plusieurs segment, par exemple :

- Premier segment de voyage urbain
- Correspondance intra-urbaine
- Deuxième segment de voyage urbain
- Correspondance multimodale sur la gare
- Troisième segment de voyage inter-urbain

Le tarif quand à lui regroupera probablement les 2 premiers segment de voyage urbain, souvent possible avec un seul ticket :

- Premier tarif de voyage urbain, fusion des 2 segments de voyage
- Deuxième tarif interurbain

La présentation devra tenir compte de ces contraintes.

9.1.3 - Tableau des paramètres de l'API

Rappel: PlanJourney retourne une JourneyKey globale pour chaque itinéraire ainsi que des SectionKey pour chaque segment d'itinéraire. Ces SectionKey serviront de paramètre à l'API JourneyFare

Nom	Туре	Format	Défaut	Description	Obligatoire
Section1	String			Issu de la balise « SectionKey » de la première section d'un itinéraire fourni par l'API PlanJourney	Oui
Section[n]	String			Le paramètre « Section » est répété autant de fois qu'il y a de section de	

Nom	Туре	Format	Défaut	Description		Obligatoire	
				transport	en	commur	ı
				(section=VehicleJourney)		dans ur	l
				itinéraire.			

Remarques

- Nb: les sections doivent être numérotées **suivant l'ordre des sections** de l'api PlanJourney.
- Si l'élément « cost » a une valeur égale à -1 cela indique que le tarif n'est pas renseigné (tarif inconnu).
- La clé repose sur les données suivantes:
 - Réseau | StopPoint Dep | Ligne | StopPoint Arr | Heure Dep | Heure Arr
- Cet API n'est disponible qu'à partir de l'interface 1_15

9.1.4 - Exemple

../fare.dll/API?action=JourneyFare§ion1=Tub;TUBNav965;TUB22;TUBNav877;1 6:41:00;16:52:00§ion2=Tub;TUBNav1066;TUB39;TUBNav936;16:59:00;17:3 8:00

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <FareList FareCount="2">
 - <Fare>
     <Network>Tub</Network>
     <Cost Money="Euro">1,10</Cost>
   - <SectionList SectionCount="1">
       <Section SectionKey="Tub;TUBNav965;TUB22;TUBNav877;16:41:00;16:52:00" />
     </SectionList>
   </Fare>
 - <Fare>
     <Network>Tub</Network>
     <Cost Money="Euro">1,10</Cost>
   - <SectionList SectionCount="1">
       <Section SectionKey="Tub;TUBNav1066;TUB39;TUBNav936;16:59:00;17:38:00" />
     </SectionList>
   </Fare>
 </FareList>
```



10 - ANNEXES

Vous trouverez ici la liste des API dépréciées et/ou non maintenues. Il est conseillé d'utiliser l'API standard PTReferential en remplacement.

10.1 - BookingTypeList

10.1.1 - Appel

« /API?Action=BookingTypeList »

10.1.2 - Description

Ce flux contient la liste des types de réservations disponibles.

10.1.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
BookingTypeExternalCode	String			Ne renvoie que le type de réservation dont le code externe est passé en paramètre	Non
BookingTypeIdx	integer			Ne renvoie que le type de réservation dont l'index est passé en paramètre	Non

10.1.4 - Exemple

<BookingTypeList BookingTypeCount="3">



10.2 - CityList

10.2.1 - Appel

« /API?Action=CityList »

10.2.2 - Exemple d'appel

Sur un site fictif,

http://www.navitia.com/navitia.ashx/api?Action=CityList

10.2.3 - Description

Ce flux contient la liste des communes disponibles. Chaque commune est identifiée par son index dans la base affiché en attribut. Chaque nœud propose de plus le nom de la commune et son pays associé.

10.2.4 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
CityIdx	Integer			Permet de ne renvoyer que la City dont l'index est renseigné	Non
CityExternalCode	String			Permet de ne renvoyer que la City dont le code externe est renseigné	Non
DistrictIdx	Integer			Permet de ne renvoyer que la liste des City se trouvant dans la région passée en paramètre	Non
DistrictExternalCo de	String			Permet de ne renvoyer que la liste des City se trouvant dans la région passée en paramètre	Non
DepartmentIdx	Integer			Permet de ne renvoyer que la liste des City se trouvant dans le département passé en paramètre	Non
DepartmentExtern alCode	String			Permet de ne renvoyer que la liste des City se trouvant dans le département passé en paramètre	Non



10.2.5 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionCityList>
 - <Params Function="Navitia_Debug.DEBUG_VKAT/api">
    <action>citylist</action>
     <cityidx>12</cityidx>
   </Params>
 - <CityList CityCount="1">
   - <City CityIdx="12" CityId="23695" CityName="Abbecourt" CityExternalCode="60002" CityCode="60430">
        <CoordX>586794,04</CoordX>
        <CoordY>2484516,38</CoordY>
       </Coord>
       <Department DepartmentIdx="0" DepartmentId="-1" DepartmentName="" DepartmentExternalCode="" DepartmentCode="" />
       <District DistrictIdx="0" DistrictId="-1" DistrictName="" DistrictExternalCode="" DistrictCode="" />
       <Country CountryIdx="1" CountryId="0" CountryName="France" CountryExternalCode="FRA" />
   </CityList>
 </ActionCityList>
```

10.3 - CommonLetter

10.3.1 - Appel

« /API?Action=CommonLetter »

10.3.2 - Description

Ce flux contient la liste des points d'arrêts / lieux remarquables / adresses dont l'orthographe est proche du nom passé en paramètre. Cette API est un extrati de l'API EntryPoint et ne traitera pas les erreurs de type « Gare de Pary Lion » en lieu de « Gare de Paris Lyon »

10.3.3 - Tableau des paramètres de l'API

Les paramètres et la structure du flux sont identique à l'API « EntryPoint »

10.4 - CompanyList

10.4.1 - Appel

« /API?Action=CompanyList »

10.4.2 - Description

Ce flux contient la liste des entreprises de transport disponibles.

10.4.3 - Tableau des paramètres de l'API

Nom Ty	ype Format	Défaut	Description	Obligatoire
--------	------------	--------	-------------	-------------



Nom	Туре	Format	Défaut	Description	Obligatoire
CompanyExternal Code	String			Ne renvoie que le transporteur dont le code externe est passé en paramètre	Non

10.4.4 - Exemple

10.5 - Departmentlist

10.5.1 - Appel

« /API?Action=DepartmentList »

10.5.2 - Description

Ce flux propose la liste des départements disponibles.

10.5.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
DepartmentIdx	Integer			Ne renvoie que le département dont l'index est passé en paramètre	Non
DepartementExter nalCode	String			Ne renvoie que le département dont le code externe est passé en paramètre	Non

10.5.4 - Exemple



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionDepartmentList>
 - <Params Function="Navitia_Debug.DEBUG_VKAT/api">
     <action>departmentlist</action>
   </Params>
 - <DepartmentList DepartmentCount="4">
   - <Department DepartmentId="0" DepartmentId="1" DepartmentName="" DepartmentExternalCode="" DepartmentCode="">
       <District DistrictIdx="0" DistrictId="-1" DistrictName="" DistrictExternalCode="" DistrictCode="" />
     </Department>
   -<\!\!\text{Department DepartmentIdx="1" DepartmentId="0" DepartmentName="Ain" DepartmentExternalCode="01" DepartmentCode="">>
       <District DistrictIdx="1" DistrictId="0" DistrictName="Pays-de-la-Loire" DistrictExternalCode="05" DistrictCode="" />
   - <Department DepartmentIdx="2" DepartmentId="1" DepartmentName="Aisne" DepartmentExternalCode="02" DepartmentCode="">
       <District DistrictIdx="1" DistrictId="0" DistrictName="Pays-de-la-Loire" DistrictExternalCode="05" DistrictCode="" />
   - <Department DepartmentIdx="3" DepartmentId="2" DepartmentName="Loire-Atlantique" DepartmentExternalCode="44" DepartmentCode="">
       <District DistrictIdx="1" DistrictId="0" DistrictName="Pays-de-la-Loire" DistrictExternalCode="05" DistrictCode="" />
     </Department>
   </DepartmentList>
 </ActionDepartmentList>
```

10.6 - DistrictList

10.6.1 - Appel

« /API?Action=DistrictList »

10.6.2 - Description

Ce flux propose la liste des régions disponibles.

10.6.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
DistrictIdx	Integer			Ne renvoie que le district dont l'index est passé en paramètre	Non
DistrictExternalCo de	String			Ne renvoie que le district dont le code externe est passé en paramètre	Non

10.6.4 - Exemple



10.7 - EventList

Cette API est dépréciée. Cette API fonctionne jusqu'à l'Interface 1_10.

10.7.1 - Appel

« /Event?Action=EventList »

10.7.2 - Exemple d'appel

http://serveur/navitia.dll/event?action=eventlist&filter=Network&interface= 1 10

10.7.3 - Description

Ce flux liste les événements « temps réels » référencés par NAViTiA.

10.7.4 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
StartDate	string	Year Month Day		Renvoie les évenements actifs à partir de la date passée en paramètre	Non
				Si EndDate est précisé, NAViTiA renvoit tous les évènements actifs compris entre ces 2 dates	
EndDate	string	Year Month Day		Renvoie les évenements actifs avant la date passée en paramètre	Non
				Si StartDate est renseigné, NAViTiA renvoit les évenements actifs sur la période comprise entre StartDate et EndDate	



Nom	Туре	Format	Défaut	Description	Obligatoire
Filter	combination	None / StopArea; VehicleJourney; Line; Network; StopPoint	All	Permet filtrer les réponses sur les types d'évenement	Non

10.8 - LineList

10.8.1 - Appel

« /API?Action=LineList »

10.8.2 - Description

Ce flux contient la liste des lignes commerciales disponibles.

10.8.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
LineIdx	Integer			Ne renvoie que la ligne commerciale dont l'index est passé en paramètre	Non
LineCode	String			Ne renvoie que les lignes commerciales correspondant au code commercial passé en paramètre	Non
LineExternalCode	String			Ne renvoie que la ligne commerciale dont le code externe est passé en paramètre	Non
NetworkIdx	Integer			Ne renvoie que les lignes commerciales appartenant au réseau dont l'index est passé en paramètre	Non
NetworkExternalCode	String			Ne renvoie que les lignes commerciales appartenant au réseau dont le code externe est	Non



		passé en paramètre
ModeIdx	Integer	Ne renvoie que les Non lignes commerciales appartenant au type de mode dont l'index du mode est passé en paramètre
ModeExternalCode	String	Ne renvoie que les Non lignes commerciales appartenant au type de mode dont le code externe du mode est passé en paramètre
ModeTypeIdx	Integer	Ne renvoie que les lignes commerciales appartenant au type de mode dont l'index est passé en paramètre
ModeTypeExternalCode	String	Ne renvoie que les lignes commerciales appartenant au Type de mode dont le code externe est passé en paramètre
ContributorIdx	Integer	Ne renvoie que les Non lignes commerciales appartenant au contributeur dont l'index est passé en paramètre
ContributorExternalCode	String	Ne renvoie que les Non lignes commerciales appartenant au contributeur dont le code externe est passé en paramètre
StopAreaIdx	Integer	Ne renvoie que les Non lignes commerciales passant par la zone d'arrêt dont l'index est passé en paramètre
StopAreaExternalCode	String	Ne renvoie que les Non lignes commerciales



			passant par la zone d'arrêt dont le code externe est passé en paramètre	
DestinationIdx	Integer		Ne renvoie que les lignes commerciales desservant par la zone d'arrêt dont l'index est passé en paramètre depuis le StopAreaIdx initial	Non
DestinationExternalCode	String		Ne renvoie que les lignes commerciales desservant par la zone d'arrêt dont le code externe est passé en paramètre depuis le StopAreaExternalCode initial	Non
UseAdapted	Boolean	True	Indique si les traitements doivent utiliser des données adaptées	Non
DepartmentIdx	List		Ne renvoie que les lignes passant par les départements dont les index sont passés en paramètre	Non
DepartmentExternalCode	List		Ne renvoie que les lignes passant par les départements dont les codes externes sont passés en paramètre	Non
DistrictIdx	List		Ne renvoie que les lignes passant par les régions dont les index sont passés en paramètre	Non
DistrictExternalCode	List		Ne renvoie que les lignes passant par les régions dont les codes externes sont passés en paramètre	Non



10.8.4 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ActionLineList>
- <Params Function="Navitia_Debug.DEBUG_VKAT/api">
    <action>linelist</action>
    </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>
            <Coord />
          </StopArea>
        </Direction>
      </Backward>
    </Line>
  </LineList>
</ActionLineList>
```

10.9 - ModeList

10.9.1 - Appel

« /API?Action=ModeList »

10.9.2 - Description

Ce flux contient la liste des modes disponibles.

10.9.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
ModeIdx	Integer			Ne renvoie que le mode dont l'index est passé en paramètre	Non
ModeExternalCode	String			Ne renvoie que le mode dont le code externe est passé en paramètre	Non



ModeTypeIdx	Integer	Ne renvoie que les modes dont le type de mode associé correspond à l'index passé en paramètre	Non
ModeTypeExternalC ode	String	Ne renvoie que les modes dont le type de mode associé a le code externe passé en paramètre	Non

10.9.4 - Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionModeList>
 - <Params Function="Navitia_Debuq.DEBUG_VKAT/api">
     <action>modelist</action>
   </Params>
 - <ModeList ModeCount="24">
     <Mode ModeIdx="0" ModeId="-1" ModeName="" ModeExternalCode="" />
     <Mode ModeIdx="1" ModeId="4" ModeName="Artesia" ModeExternalCode="OCEArtesia" />
     <Mode ModeIdx="2" ModeId="20" ModeName="Car TER" ModeExternalCode="OCECar TER" />
     <Mode ModeIdx="3" ModeId="3" ModeName="Corail" ModeExternalCode="OCECorail" />
     <Mode ModeIdx="4" ModeId="8" ModeName="Corail Intercité" ModeExternalCode="OCECorail Intercité" />
     <Mode ModeIdx="5" ModeId="10" ModeName="Corail Lunéa" ModeExternalCode="OCECorail Lunéa" />
     <Mode ModeIdx="6" ModeId="9" ModeName="Corail Téoz" ModeExternalCode="OCECorail Téoz" />
     <Mode ModeIdx="7" ModeId="6" ModeName="Elipsos" ModeExternalCode="OCEElipsos" />
     <Mode ModeIdx="8" ModeId="21" ModeName="Eurostar" ModeExternalCode="OCEEurostar" />
     <Mode ModeIdx="9" ModeId="0" ModeName="Indéfini" ModeExternalCode="Indéfini" />
     <Mode ModeIdx="10" ModeId="11" ModeName="Lyria" ModeExternalCode="OCELyria" />
     <Mode ModeIdx="11" ModeId="1" ModeName="Métro" ModeExternalCode="metro" />
     <Mode ModeIdx="12" ModeId="12" ModeName="Navette" ModeExternalCode="OCENavette" />
     <Mode ModeIdx="13" ModeId="19" ModeName="Noctilien" ModeExternalCode="OCENoctilien" />
     <Mode ModeIdx="14" ModeId="13" ModeName="RER A" ModeExternalCode="OCERER A" />
     <Mode ModeIdx="15" ModeId="14" ModeName="RER B" ModeExternalCode="OCERER B" />
     <Mode ModeIdx="16" ModeId="18" ModeName="RER C" ModeExternalCode="OCERER C" />
     <mode ModeIdx="17" ModeId="17" ModeName="RER D" ModeExternalCode="OCERER D" />
     <Mode ModeIdx="18" ModeId="16" ModeName="RER E" ModeExternalCode="OCERER E" />
     <Mode ModeIdx="19" ModeId="7" ModeName="TGV" ModeExternalCode="OCETGV" />
     <Mode ModeIdx="20" ModeId="22" ModeName="Thalys" ModeExternalCode="OCEThalys" />
     <Mode ModeIdx="21" ModeId="5" ModeName="Train" ModeExternalCode="OCETrain" />
     <Mode ModeIdx="22" ModeId="2" ModeName="Train TER" ModeExternalCode="OCETrain TER" />
     <Mode ModeIdx="23" ModeId="15" ModeName="Transilien" ModeExternalCode="OCETransilien" />
   </ModeList>
 </ActionModeList>
```

10.10 - ModeTypeList

10.10.1 - Appel



« /API?Action=ModeTypeList »

10.10.2 - Description

Ce flux contient la liste des types de modes disponibles.

10.10.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
ModeTypeExternalCode	String			Ne renvoie que le type de mode dont le code externe est passé en paramètre	Non
ModeTypeIdx	integer			Ne renvoie que le type de mode dont l'index est passé en paramètre	Non

10.10.4 - Exemple

<ModeTypeList ModeTypeCount="3">

ModeTypeExternalCode="Air"

10.11 - NetworkList

10.11.1 - Appel

« /API?Action=NetworkList »

10.11.2 - Description

Ce flux contient la liste des réseaux disponibles.

10.11.3 - Tableau des paramètres de l'API



NetworkIdx	Integer	Ne renvoie que le réseau dont l'index est passé en paramètre	Non
NetworkExternalCode	String	Ne renvoie que le réseau dont le code externe est passé en paramètre	Non

10.11.4 - Exemple

10.12 - OnBoardServiceList

10.12.1 - Appel

« /API?Action=OnBoardServiceList »

10.12.2 - Description

Ce flux contient la liste des services à bord disponibles dans les données. Elles n'ont pas de régime de données associées car il s'agit uniquement de la liste globale.

Les nuancements de la SNCF sont traités comme des services à bord.

10.12.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
OnBoardServiceExternalCo de	String			Ne renvoie que le service à bord dont le code externe est passé en paramètre	Non



OnBoardServiceIdx integer Ne renvoie que Non le service à bord dont l'index est passé en paramètre

10.12.4 - Exemple

Mettre un exemple de flux de type <OnBoardServiceList OnBoardServiceCount="3">

<OnBoardService OnBoardServiceIdx="1" OnBoardServiceExternalCode="M"
 OnBoardServiceName="vélo payant" OnBoardServiceType="OnboardService" />
</OnBoardServiceList>

10.13 - Phonetic

10.13.1 - Appel

« /API?Action=Phonetic »

10.13.2 - Description

Ce flux contient la liste des points d'arrêts / lieux remarquables / adresses phonétiquement proches du nom passé en paramètre. Cette API est un extrait de l'API EntryPoint et ne traitera pas les erreurs de type « Gzre de Paris Lyno » en lieu de « Gare de Paris Lyon »

10.13.3 - Tableau des paramètres de l'API

Les paramètres et la structure du flux sont identique à l'API « EntryPoint »

10.14 - RouteList

10.14.1 - Appel

« /API?Action=RouteList »

10.14.2 - Description

Ce flux contient la liste des types de courses disponibles

10.14.3 - Tableau des paramètres de l'API



Nom	Туре	Format	Défaut	Description	Obligatoire
RouteIdx	Integer			Ne renvoie que le type de course dont l'index est passé en paramètre	Non
RouteExternalCode	String			Ne renvoie que le type de course dont le code externe est passé en paramètre	Non
LineIdx	Integer			Ne renvoie que les types de course appartenant à la ligne dont l'index est passé en paramètre	Non
LineExternalCode	String			Ne renvoie que les types de course appartenant à la ligne dont le code externe est passé en paramètre	Non
StopAreaIdxList	List			Ne renvoie que les courses passant par les arrêts commerciaux dont les index sont passés en paramètres	Non
StopAreaExternalCo deList	List			Ne renvoie que les courses passant par les arrêts commerciaux dont les codes externes sont passés en paramètres	Non
StopPointExternalCo de	String			Ne renvoie que les types de courses passant par l'arrêt physique dont le code externe est passé en paramètre	Non
UseAdapted	Boolean		True	Indique si les traitements doivent	Non



			utiliser des données adaptées	
DepartmentIdx	List		Ne renvoie que les lignes passant par les départements dont les index sont passés en paramètre	Non
DepartmentExternal Code	List		Ne renvoie que les lignes passant par les départements dont les codes externes sont passés en paramètre	Non
DistrictIdx	List		Ne renvoie que les lignes passant par les régions dont les index sont passés en paramètre	Non
DistrictExternalCode	List		Ne renvoie que les lignes passant par les régions dont les codes externes sont passés en paramètre	Non
CheckOrder	-1 / 1	Non	Si plus de un arrêt commercial est passé en paramètre, ne renvoie que les routes passant par ces arrêts commerciaux DANS L'ORDRE DANS LEQUELS ILS SONT PASSES si valorisé à 1	Non

10.14.4 - Exemple



```
<?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"
        IsFrequence="1" IsAdapted="0">
       - <Line LineIdx="5" LineId="3228" LineName="Line 5" LineCode="5" LineExternalCode="RATLigne5" LineAdditionalData="" SortOrder="5" LineColor=""
            | HasAdaptedRoute="0">
| AmadaptedRoute="0">
| AmadaptedRoute="1" | ModeId="1" | ModeName="Métro" | ModeExternalCode="metro" | />
| AmadaptedRoute="1" | ModeId="1" | ModeName="Métro" | ModeExternalCode="metro" | />
| AmadaptedRoute="0" | ModeId="0" | ModeExternalCode="metro" | />
| AmadaptedRoute="0" | ModeId="0" | ModeExternalCode="metro" | />
            <Forward ForwardName="" />
            <Backward BackwardName="" />
         </Line>
       </Route>
     </RouteList>
  </ActionRouteList>
```

10.15 - StopAreaList

10.15.1 - Appel

« /API?Action=StopAreaList »

10.15.2 - Exemple d'appel

Sur un site fictif,

http://localhost:8081/Navitia Debug.DEBUG VKAT/API?Action=StopAreaList&CityIdx=122

10.15.3 - Description

Ce flux contient la liste des arrêts commerciaux disponibles.

10.15.4 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoir e
CityIdx	Integer			Ne renvoie que les arrêts commerciaux appartenant à la commune dont l'index est passé en paramètre	Non
StopAreaI dx	Integer			Ne renvoie que l'arrêt commercial dont l'index est passé en paramètre	Non
StopArea ExternalC ode	String			Ne renvoie que l'arrêt commercial dont le code externe est passé en paramètre	Non



Nom	Туре	Format	Défaut	Description	Obligatoir e
LineIdx	Integer			Ne renvoie que les arrêts commerciaux appartenant à la ligne dont l'index est passé en paramètre. Non cumulable avec StopAreaExternalCode ou StopAreaIdx. Renvoie l'intégralité des Arrêts commerciaux de la ligne	Non
MainStop Area	Boolean			Ne renvoi que les arrêts commerciaux « remarquables » (MainStop)	Non
ModeIdx	integer	Idx1 ; Idx2 ; Idx3		Liste des index des modes utilisables lors du trajet	Non
ModeTyp eIdx	Integer	Idx1; Idx2; Idx3		Liste des index des types de mode utilisables lors du trajet	Non

NOTE : Le filtre « LineIdx » est prioritaire sur les autres filtres. Si ce Filtre est renseigné, le flux contiendra la totalité des arrêts commerciaux de la ligne.

10.15.5 - Exemple

```
<?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 ModeIdx="1" ModeId="1" ModeName="bus" ModeExternalCode="bus" />
     </ModeList>
    </StopArea>
 + <StopArea StopAreaIdx="446" StopAreaId="144" StopAreaName="Chertrie" StopAreaExternalCode="144" MainStopArea="0" MultiModal="0" CarPark="0"
     MainConnection="0" AdditionalData=""
 + <StopAreaId="1447" StopAreaId="146" StopAreaId="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 Hare|" 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>
```



10.16 - VehicleJourneyDetailList

10.16.1 - Appel

« /API?Action=VehicleJourneyDetailList »

10.16.2 - Description

Ce flux contient la liste des circulations disponibles. Il s'agit d'une réimplémentation du flux VehicleJourneyList. Il comprend une nouvelle présentation des données renvoyées, et d'autres possibilités de filtres.

Remarque : le volume de données en sortie peut être conséquent. Il convient donc de se rapprocher de Canal TP afin de valider que les filtres utilisés sont suffisamment discriminant pour éviter une surcharge réseau.



10.16.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
VehicleJourneyIdx	Integer			Ne renvoie que la circulation dont l'index est passé en paramètre	Non
VehicleJourneyExternalC ode	String			Ne renvoie que la circulation dont le code externe est passé en paramètre	Non
VehicleJourneyName	String			Ne renvoie que les circulations dont l'intitulé correspond au nom passé en paramètre (plusieurs circulations possibles)	Non
Date	String	Year Month Day		Ne renvoie que les circulations actives sur la date passée en paramètre	Non
VPatternSetIdx	Integer			Ne renvoie que les circulations actives sur le régime passé en paramètre	Non
RouteIdx	Integer			Ne renvoie que les circulations appartenant au type de course dont l'index est passé en paramètre	Non
RouteExternalCode	String			Ne renvoie que les circulations appartenant au type de course dont le code externe est passé en paramètre	Non
StartTime	String	Hour Minute		Ne renvoie que les circulations dont les heures de départ sont supérieures à l'heure passée en	Non



		paramètre
EndTime	String Hour Minute	Ne renvoie que les les circulations dont les Non heures de départ sont inférieures à l'heure passée en paramètre
StopAreaExternalCode	String	Ne renvoie que les circulations dont la route Non passe par la zone d'arrêt
StopAreaIdx	Integer	Ne renvoie que les circulations dont la route Non passe par les zones d'arrêts
CityIdx	Integer	Ne renvoie que les circulations dont la route Non passe par les villes
CityExternalCode	String	Ne renvoie que les circulations dont la route Non passe par les villes
DepartementIdx	Integer	Ne renvoie que les circulations dont la route Non passe par les départements
DepartmentExternalCod e	String	Ne renvoie que les circulations dont la route Non passe par les départements
DistrictIdx	Integer	Ne renvoie que les circulations dont la route Non passe par les régions
DistrictExternalCode	String	Ne renvoie que les circulations dont la route passe par les régions



NetworkIdx	Integer	Ne renvoie que les circulations dont la route appartient aux réseaux
NetworkExternalCode	String	Ne renvoie que les circulations dont la route appartient aux réseaux
ModeIdx	Integer	Ne renvoie que les circulations concernées par les modes passés en paramètres
ModeExternalCode	String	Ne renvoie que les circulations concernées par les modes passés en paramètres
ModeTypeIdx	Integer	Ne renvoie que les circulations concernées par les types de mode passés en paramètres
ModeTypeExternalCode	String	Ne renvoie que les circulations concernées par les types de modes passés en paramètres
IsAdapted	Boolea n	Indique si le calcul doit utiliser des données Non adaptées
ReformTrain	Boolea Fai	Indique si le résultat doit être transformé pour Non montrer les trains au lieu des tranches



10.16.4 - Exemple

Ce flux propose la liste des VehicleJourney avec toutes les informations s'y rattachant. Viennent ensuite en fin de flux les informations sur les TAD (ODTList), commentaires puis les informations temps réels.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ActionVehicleJourneyDetailList>
  <Params Function="api">
     <action>vehiclejourneydetaillist</action>
      <routeidx>1</routeidx>
      <interface>1_11</interface>
   </Params>
  <VehicleJourneyList VehicleJourneyCount="3">
   - <VehicleJourney VehicleJourneyIdx="1" VehicleJourneyId="33702" VehicleJourneyName="839455" VehicleJourneyRouteIdx="1"
        Vehicle lourneyExternalCode ="OCESN839455E04002" IsAdapted = "0"
      - «Route Routeidx="1" Routeid="7857" RouteName="Bar-le-Duc vers Nancy-Ville" RouteExternalCode="OCETrain TER-87175042-87141002-4" IsForward="1"
        RouteLineIdx="1" IsFrequence="0" IsAdapted="0"> - <Line LineIdx="1" LineId="2651" LineName="Bar-le-Duc - Nancy-Ville" LineCode="" LineExternalCode="OCETrain TER-87175042-87141002" LineAdditionalData=""
              SortOrder="1" LineColor="" HasAdaptedRoute="0">
           <ModeType ModeTypeIdx="11" ModeTypeExternalCode="LocalTrain" ModeTypeName="Train local" />
- <Network NetworkIdx="1" NetworkId="0" NetworkName="SNCF" NetworkExternalCode="OCE1">
                 <ImpactPosList ImpactPosCount="0" />
              </Network>
              <Forward ForwardName="Bar-le-Duc vers Nancy-Ville" />
              <Backward BackwardName="Nancy-Ville vers Bar-le-Duc" />
<ImpactPosList ImpactPosCount="0" />
            </Line>
            <ImpactPosList ImpactPosCount="0" />
         </Route>
         <mode ModeId="18" ModeId="17" ModeName="Train TER" ModeExternalCode="Train TER" ModeTypeExternalCode="LocalTrain" />
         Company CompanyIdx="26" CompanyIdx="25" CompanyName="SNCF CRS" CompanyExternalCode="OCESN" /> 
<Vehicle VehicleIdx="0" VehicleId="-1" VehicleName="" VehicleExternalCode="" MIPAccess="False" BikeAccepted="False" AirConditioned="False" /> 
<ValidityPattern ValidityPatternBeginningDay="2009|07|12" IsAdapted="0"</p>
            <ValidityPattern ValidityPatternBeginningDay="2009|07|12" IsAdapted="1</p>
           <StopList StopCount="4":
+ <Stop StopIdx="209191" StopPointIdx="3143" VehicleJourneyIdx="1" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1"
      ValidityPatternSetCommentPos="0" StopOrder="0" VehicleIdx="0">
   <Stop StopIdx="217539" StopPointIdx="4662" VehicleJourneyIdx="1" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1"</p>
  ValidityPatternSetCommentPos="0" StopOrder="1" VehicleIdx="0"> < Stop StopIdx="223697" StopPointIdx="11323" VehicleJourneyIdx="1" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1"
      \label{lem:validityPatternSetCommentPos="0"} ValidityPatternSetCommentPos="0" StopOrder="2" VehicleIdx="0" StopOrder="2" VehicleIdx="2" VehicleI
- <Stop StopIdx="1" StopPointIdx="8439" VehicleJourneyIdx="1" HourNumber="0" MinuteNumber="0" DestinationPos="-1" ODTIdx="0" ODTPos="-1"
      ValidityPatternSetCommentPos="0" StopOrder="3" VehicleIdx="0">
      <StopTime />
   - <StopArrivalTime>
         <TotalSeconds>48120</TotalSeconds>
         <Day>0</Day>
         <Hour>13</Hour:
          <Minute>22</Minute>
       </StopArrivalTime:</pre>
      <StopPoint StopPointIdx="8439" StopPointId="7292" StopPointName="gare de Nancy-Ville" StopPointExternalCode="0CETrain TER-87141002"</p>
         FareZone="0">
          <StopPointAddress StopPointAddressName="" StopPointAddressNumber="" StopPointAddressTypeName="" />
         <Equipment Sheltered="False" MIPAccess="False" Elevator="False" Escalator="False" BikeAccepted="False" BikeDepot="False" />
         Mode ModeIdx="18" ModeId="17" ModeName="Train TER" ModeExternalCode="Train TER" ModeTypeExternalCode="LocalTrain" />
         <City CityIdx="2589" CityId="2071" CityName="Nancy" CityExternalCode="54395" CityCode="54000" />
      - <Coord>
             <CoordX>882537.00</CoordX>
            <CoordY>2417058,00</CoordY>
         </Coord>
         <ImpactPosList ImpactPosCount="0" />
      </StopPoint>
   </Stop>
</StopList>
```



10.17 - VehicleJourneyList

10.17.1 - Appel

« /API?Action=VehicleJourneyList »

10.17.2 - Description

Ce flux contient la liste des circulations disponibles.

Remarque : le volume de données en sortie peut être conséquent. Il convient donc de se raprocher de Canal TP afin de valider que les filtres utilisés sont suffisamment discriminant pour éviter une surcharge réseau.

10.17.3 - Tableau des paramètres de l'API

Nom	Туре	Format	Défaut	Description	Obligatoire
VehicleJourneyIdx	Integer			Ne renvoie que la circulation dont l'index est passé en paramètre	Non
VehicleJourneyExternalC ode	String			Ne renvoie que la circulation dont le code externe est passé en paramètre	Non
VehicleJourneyName	String			Ne renvoie que les circulations	Non



		dont l'intitulé correspond au nom passé en paramètre (plusieurs circulations possibles)	
Date	String Year Month Day	Ne renvoie que les circulations actives sur la date passée en paramètre	Non
VPatternSetIdx	Integer	Ne renvoie que les circulations actives sur le régime passé en paramètre	Non
RouteIdx	Integer	Ne renvoie que les circulations appartenant au type de course dont l'index est passé en paramètre	Non
RouteExternalCode	String	Ne renvoie que les circulations appartenant au type de course dont le code externe est passé en paramètre	Non
LineIdx	Integer	Ne renvoie que les circulations appartenant au type de ligne dont l'index est passé en paramètre	Non
LineExternalCode	String	Ne renvoie que les circulations appartenant au type de ligne dont le code	Non



				externe est passé en paramètre	
StopAreaIdx	Integer			Ne renvoie que les circulations passant par l'arrêt commercial dont l'index est passé en paramètre	Non
StopAreaExternalCode	String			Ne renvoie que les circulations passant par l'arrêt commercial dont le code externe est passé en paramètre	Non
StartTime	String	Hour Minute		Ne renvoie que les circulations dont les heures de départ sont supérieures à l'heure passée en paramètre	Non
EndTime	String	Hour Minute		Ne renvoie que les les circulations dont les heures de départ sont inférieures à l'heure passée en paramètre	Non
ShowStop	Boolea n		-1	Indique si le flux de sortie doit afficher la liste des arrêts (stop)	Non

Remarque:

- Si VehicleJourneyIdx est passé en paramètre, tous les autres filtres seront ignorés.
- « VehicleJourneyExternalCode » et « VehicleJourneyName » sont combinables avec « Date » ou « VPatternSetIdx »



Si ni « VehicleJourneyIdx », ni « VehicleJourneyIdx », ni « VehicleJourneyName » ne sont renseignés, alors les paramètres « RouteIdx », « StartTime » et « EndTime » sont obligatoires.

10.18 - Page de description des flux disponibles « /ActionList »

Cette API est dépréciée et remplacée par le présent document.

Cette page énumère tous les flux disponibles sur l'application interrogée ainsi que leurs paramètres d'appels (type de données, et caractère obligatoire ou non).

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ActionList>
   <params fonction="actionlist" />
 - <Action>
     <Name>Const</Name>
     <Description>Liste paramètres
   </Action>
 - <Action>
     <Name>Load</Name>
     <Description>Charges les données en mémoire
   </Action>
 - <Action>
    <Name>ActionList</Name>
     <Description>Liste des actions disponibles
   </Action>
 - <Action>
    <Name>API</Name>
    <Description>Accès aux fonctions de l'application
   + <API Name="CityList">
   + <API Name="StopAreaList">
   + <API Name="SiteList">
   + <API Name="AddressList">
   + <API Name="SiteTypeList">
   + <API Name="AddressTypeList">
   + <API Name="MakeBinaryCriteria">
   + <API Name="EntryPoint">
   + <API Name="PlanJourney">
   + <API Name="DetailedJourney">
   + <API Name="ModeList">
   + <API Name="NetworkList">
   + <API Name="CompanyList">
   + <API Name="LineList">
   + <API Name="VPatternSetList">
   + <API Name="LineStopAreaList">
   + <API Name="DepartureBoard">
   + <API Name="LineSchedule">
   + <API Name="EndOfCourse">
   </Action>
 </ActionList>
```



Exemple du nœud de présentation de l'API CityList :

10.19 - Page de liste des modifications / évolutions « /ChangeLog »

Cette API est dépréciée et remplacé par le fichier changeLog.txt disponible auprès de Canal TP.

Cette page rappelle l'historique des corrections et évolutions du web-service depuis la première version stable.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <ChangeLogList>
   <Params Function="changelog" />
  <ChangeLog:
    <Date>07/02/2008
    <Version>1.6.54.175</Version>
    <Frequence>Activée</prequence>
   </ChangeLog>
 - <ChangeLog>
    <Date>05/12/2007</Date>
     <Version>1.6.54.173</Version>
       <Correction>[EVENT] Anomalie sur API EventList quand le nombre d'événements est trop grand.
     </CorrectionList>
   </ChangeLog>
 + <ChangeLog>
 + <ChangeLog>
 + <ChangeLog>
 + <ChangeLog>
 + <ChangeLog>
 + <ChangeLog>
 - <ChangeLog>
    <Date>15/05/2007</Date>
     <Version>1.6.54.157 </Version>
   - <EvolutionList>
      <Evolution>[EVENT]Lecture du paramètre "delay" dans l'URL et du champ "EVT_DELAY" en base de données</Evolution>
     </EvolutionList

    CorrectionLists

      <Correction>[Parser 1.4.48.1] Modification de l'indice de TList utilisé dans la réponse de l'API "PhoneticList" </Correction>
       <Correction>[Parser 1.5] Modification de la liste de référence d'événements utilisée par chaque solution d'itinéraire.
     </CorrectionList>
   </ChangeLog>
  <ChangeLog
     <Date>09/05/2007
     <Version>1.6.54.155/Version>
   - < EvolutionList >
       <Evolution>Gestion des MDI</Evolution>
      <Evolution>[Parser 1.7]Gestion du paramètre "DateChangeTime" pour API "DepartureBoard" et "LineScheduled". Gestion du calendrier "programme TV" </Evolution>
       <Evolution>[Event] Accroche d'une liste d'événements à chaque solution d'itinéraire. </Evolution>
     </EvolutionList>
   - <CorrectionList:
```