



API de recherche d'itinéraires transports collectifs

Version 0.9 – 15/02/2012

Sommaire

1	Introduction.....	3
1.1	Contexte	3
1.2	Objectif	3
1.3	Périmètre.....	3
2	Terminologie	4
3	Définition de l'API.....	12
3.1	Principe.....	12
3.2	Identification des points de départ ou d'arrivée.....	12
3.3	Calcul d'itinéraires transport en commun.....	14
4	Présentation des différents niveaux d'interface	17
4.1	Interfaces REST / JavaScript	17
4.2	Interface HTML simple	17
4.3	Interface KML	18
4.4	Interface pour calcul d'itinéraire distribué.....	18
5	Implémentation de l'API.....	19
5.1	Service REST	19
5.2	Clef d'accès	19
5.3	Interrogation du service	19
5.4	Réponse du service	20
5.5	Définition des méthodes	21
6	Utilisation via JavaScript.....	25
6.1	La solution JSONP	25
6.2	Implémentation.....	25
7	Utilisation du résultat au format HTML.....	26
7.1	Objectif de l'interface	26
7.2	Description des éléments HTML produits	26
7.3	Personnalisation de la présentation (CSS)	26
7.4	Exemples de mises en forme	28
8	Exemples d'intégration dans une carte.....	29
8.1	Introduction	29
8.2	Affichage des points de départ ou d'arrivée	29
8.3	Affichage du résultat de la recherche d'itinéraires	30

1 Introduction

1.1 Contexte

Lors de la réunion du groupe de normalisation française sur les échanges de données de transport public (CN03/GT7) du 14 mars 2011, a été présentée une initiative associant l'UTP, le ministère des transports (DGITM), et les sociétés Cityway, Dryade et CanalTP qui se sont déclarées intéressées en vue de produire des spécifications d'interface (API) pour un service web de recherche d'itinéraire TC et Multimodal, et des documents d'accompagnement pour faciliter leur mise en œuvre.

Un document d'état des lieux et d'étude préliminaire est disponible sur le web :

<http://www.predim.org/spip.php?article3724>

Ce document et ses annexes sont le résultat de ce travail. Comme prévu initialement, ces spécifications sont publiées et soumises au groupe de travail GT7 de la CN03 lors de la réunion du 15 février 2012, en vue d'une éventuelle normalisation. Des implémentations pourront commencer dans le cadre de projets qui pourraient être lancés à l'initiative des industriels ou des maîtres d'ouvrage de Services d'Information Multimodale.

1.2 Objectif

L'objectif de cette étude, est de définir une API pour standardiser la communication avec un service de calcul d'itinéraires. Cette première version de l'API est destinée aux personnes désirant intégrer dans une application ou un site web une fonction de calcul d'itinéraires mise à disposition par une AOT.

Cette API peut être utilisée dans différents contextes, nécessitant la manipulation de techniques plus ou moins complexes. L'API supporte plusieurs formats de réponse selon le profil de l'utilisateur : développeur (format XML/JSON), ou webmaster (format HTML/KML).

1.3 Périmètre

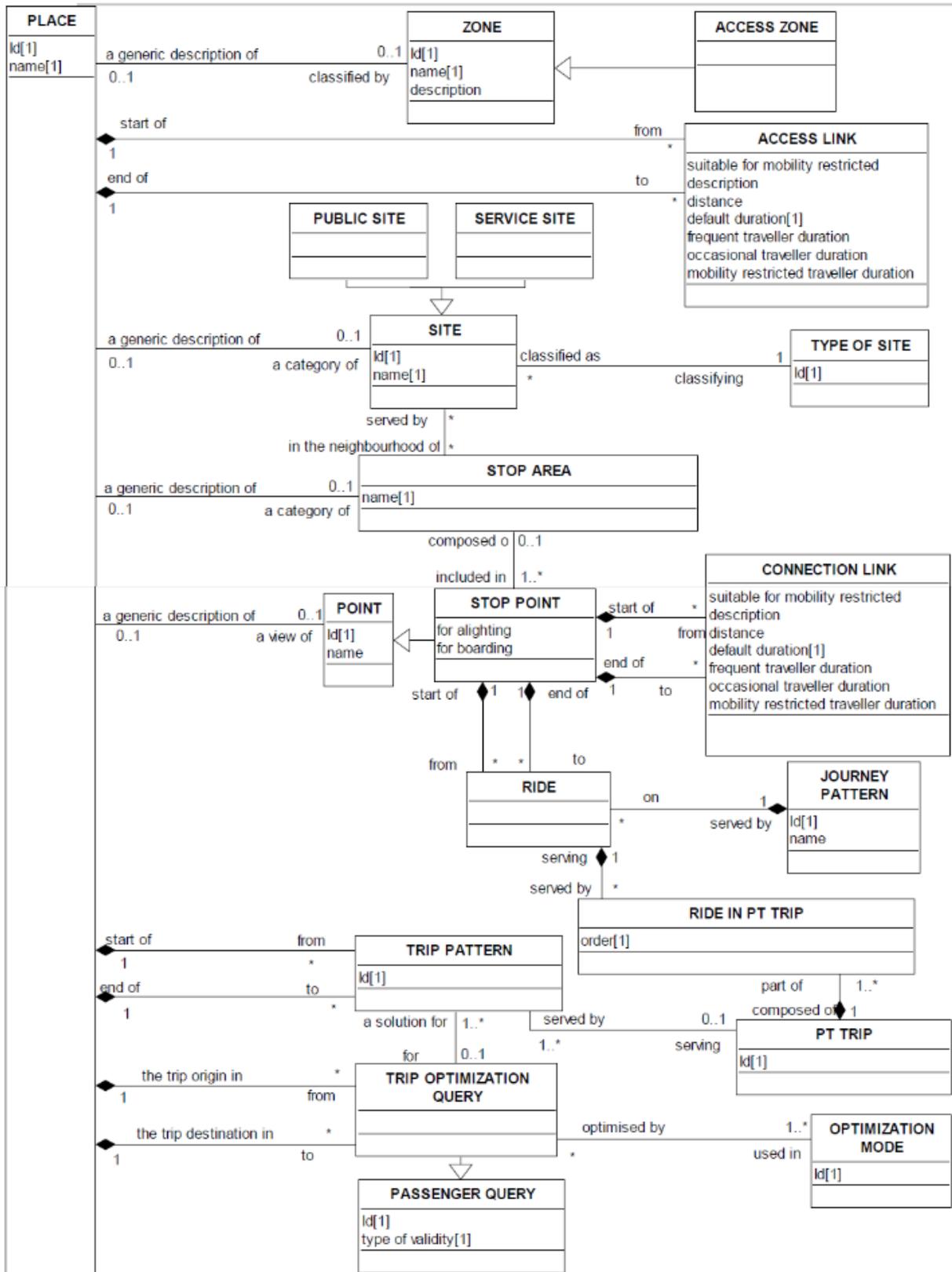
Le périmètre fonctionnel est volontairement restreint pour faciliter l'utilisation de l'API, et limiter le nombre de paramètres d'appel. Le format de la réponse proposé, a également été conçu dans cet esprit de simplicité. Néanmoins la structure de données possède des points d'extensions pour ajouter des informations complémentaires.

Bien que la notion de calcul réparti ne fasse pas partie du périmètre de cette étude, la structure de données proposée dans cette API peut être utilisée pour répondre à un calcul d'itinéraire partiel. Par contre, cette version de l'API ne définit pas les fonctionnalités spécifiques à mettre en place pour répondre à tous les besoins de ce type de calcul (ex : isochrone 1->n, n->1, n->m).

2 Terminologie

L'objectif est ici d'avoir une terminologie, que l'on retrouvera au niveau des champs de requête et de réponse, la plus normalisée possible. Transmodel est aujourd'hui la principale norme, avec IFOPT, à laquelle il convient de se référer pour la définition des concepts transport (leur nom et leur définition en particulier).

La figure ci-dessous redonne le cadre de modélisation d'une recherche d'itinéraire tel qu'il est proposé par Transmodel et permet d'introduire le glossaire qui suit.



CD TM Fig. 48 Passenger Trips

Figure 1 - Modélisation des trajet voyageurs proposée par Transmodel

Terme anglais	Terme français correspondant	Définition
ACCESS LINK	TRONÇON D'ACCÈS	"La possibilité matérielle (spatiale) pour un passager d'accéder à un système de transport public ou de le quitter. Ce trajet peut avoir lieu pendant un déplacement pour permettre au voyageur d'effectuer: - le trajet à pied d'un LIEU (origine du déplacement) vers un POINT D'ARRÊT (origine du DÉPLACEMENT SUR RÉSEAU), ou - le trajet à pied depuis un POINT D'ARRÊT (destination du DÉPLACEMENT SUR RÉSEAU) vers un LIEU (destination du déplacement)."
ACCESS ZONE	ZONE D'ACCÈS	Une ZONE pour laquelle la durée la durée de réalisation d'un TRONÇON D'ACCÈS vers un POINT D'ARRÊT donné est constante.
CALL	PASSAGE (<i>traduction non officielle</i>)	A visit to a SCHEDULED STOP POINT as part of a SERVICE JOURNEY. The Call is a view that brings together data relating to the individual visit. Passage d'un véhicule à un POINT D'ARRET PLANIFIE faisant partie d'une COURSE COMMERCIALE. Le PASSAGE une vue (et non un véritable concept) qui permet d'agréger l'ensemble des informations relative au passage d'un véhicule à l'arrêt (en particulier l'ensemble de HEURES DE PASSAGE)
CONNECTION (LINK)	TRONÇON DE CORRESPONDANCE	La possibilité physique (spatiale) d'un passager de passer d'un véhicule de transport public vers un autre dans le but de continuer son voyage. Des temps de parcours différents peuvent être nécessaires en fonction du type de passager.
DESTINATION DISPLAY	DESTINATION AFFICHÉE	Une destination d'un PARCOURS particulier, affichée au public en général sur une girouette ou sur tout autre afficheur embarqué.
ESTIMATED PASSING TIME	HEURE DE PASSAGE ESTIMÉE	"Donnée temporelle, calculée des données les plus récentes disponibles, indiquant quand un véhicule de transport passera à un POINT SUR PARCOURS pendant une COURSE DATÉE donnée. Ceux-ci sont principalement utilisés pour informer les passagers sur les heures prévisibles d'arrivée et/ou de départ, mais elles peuvent aussi être utilisées pour la surveillance et la replannification."
JOURNEY PATTERN	PARCOURS	"Une liste ordonnée de POINTs D'ARRÊT et de POINTs HORAIRES sur un unique ITINERAIRE, décrivant le plan de déplacement pour les véhicules de transport public. Un PARCOURS peut passer par le même POINT plus d'une fois. Le premier point d'un PARCOURS est l'origine. Le dernier point est la destination."
OPTIMISATION	MODE	Un type de critère d'optimisation utilisé pour

MODE	D'OPTIMISATION	sélectionner une proposition de déplacement (par exemple la durée minimum, la distance minimum, le nombre de correspondances minimum, le montant à payer minimum, etc.).
PASSENGER QUERY	REQUÊTE USAGER	Une requête pour une information spécifique sur le service de transport public, exprimée au cours d'une TRANSACTION D'INFORMATION USAGER.
PASSING TIME	HEURE DE PASSAGE	Données temporelles concernant le passage des véhicules de transport public à un POINT particulier (par exemple heure d'arrivée, heure de départ, temps d'attente).
PLACE	LIEU	Un lieu géographique de type quelconque pouvant être l'origine ou la destination d'un déplacement. Un LIEU peut être de dimension 0 (un POINT), 1 (un tronçon routier) ou 2 (une ZONE).
POINT	POINT	Un nœud de dimension 0 servant à la description spatiale du réseau. Les POINTs peuvent être localisés par la LOCALISATION dans un SYSTEME DE LOCALISATION donné.
PT TRIP	DÉPLACEMENT SUR RÉSEAU	Une partie d'un déplacement débutant lors de la première montée à bord d'un véhicule de transport public jusqu'à la dernière descente du véhicule de transport public. Un DÉPLACEMENT SUR RÉSEAU est composé d'un ou plusieurs VOYAGES et mouvements de l'usager (en général à pied) nécessaires pour des TRONÇONS DE CORRESPONDANCE.
PUBLIC SITE	SITE PUBLIC	Un sous-type de SITE sans lien particulier avec l'exploitant des transports publics.
RIDE	VOYAGE	Une partie d'un déplacement d'un utilisateur (passager, conducteur) à bord d'un seul véhicule de transport public d'un POINT D'ARRÊT à un autre sur un PARCOURS.
RIDE IN PT TRIP	VOYAGE DANS DÉPLACEMENT SUR RÉSEAU	Un VOYAGE dans un DÉPLACEMENT SUR RÉSEAU, indiquant son rang dans le DÉPLACEMENT SUR RÉSEAU considéré.
SCHEDULED STOP POINT	POINT D'ARRÊT PLANIFIÉ	Un POINT où les passagers peuvent monter à bord ou descendre des véhicules.
SERVICE JOURNEY	COURSE COMMERCIALE	
SERVICE JOURNEY PATTERN	PARCOURS COMMERCIAL (MISSION)	Un PARCOURS associé à une COURSE COMMERCIALE (transportant des passagers).
SERVICE SITE	SITE ASSOCIÉ	Un sous-type de SITE ayant un intérêt particulier pour l'exploitant (p.ex. endroit où un service ou un tarif commun est proposé).
SITE	SITE	Un LIEU bien connu du public qui peut servir de référence aux usagers pour indiquer l'origine ou la destination d'un déplacement.
STOP AREA	ARRÊT	Un regroupement de POINTs D'ARRÊT proches les uns des autres.
TIMETABLED PASSING TIME	HEURE DE PASSAGE GRAPHIQUÉE	Donnée temporelle théorique relative au passage d'un véhicule de transport public à un POINT SUR

		PARCOURS donné sur une COURSE et pour un JOUR TYPE.
TRIP OPTIMIZATION QUERY	REQUÊTE D'OPTIMISATION DE DÉPLACEMENT	Une REQUÊTE USAGER concernant une proposition de déplacement optimal en fonction d'un MODE D'OPTIMISATION spécifié.
TRIP PATTERN	SCHÉMA DE DÉPLACEMENT	La description spatiale d'un déplacement d'un passager (ou d'un autre type d'utilisateur, p.ex. conducteur) d'un LIEU d'un certain type à un autre. Le déplacement peut consister d'un DÉPLACEMENT SUR RÉSEAU et des déplacements complémentaires (en général à pied) correspondant aux TRONÇONS D'ACCÈS et TRONÇONS DE CORRESPONDANCE, ou d'un trajet à pied uniquement.
TYPE OF SITE	TYPE DE SITE	Une classification des SITES.
VEHICLE JOURNEY	COURSE	Le mouvement planifié d'un véhicule de transport public effectué un JOUR TYPE donné, depuis un point début à un point fin d'un PARCOURS sur un ITINÉRAIRE.
ZONE	ZONE	Un LIEU de dimension 2 appartenant à une zone d'exploitation d'une entreprise de transport public (zone administrative, ZONE TARIFAIRE, ZONE D'ACCÈS, etc.).

Les figures ci-dessous présentent une vue d'ensemble des différents concepts impliqués dans la recherche d'itinéraire. L'interface proposée ici n'a pas à tous les prendre en compte (elle a une vocation de simplicité et non d'exhaustivité), mais cela permet de bien poser le cadre dans lequel elle doit s'insérer.

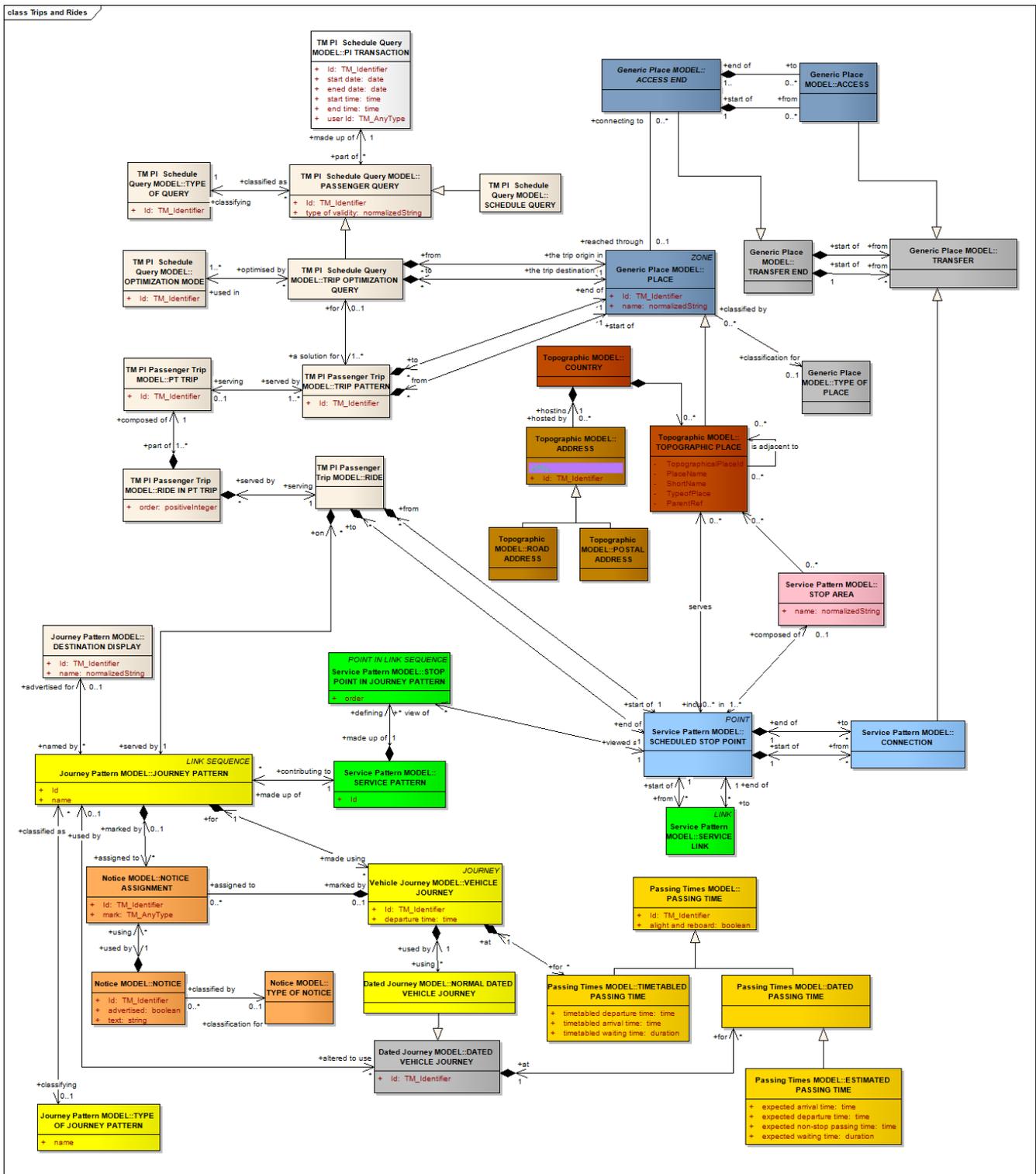


Figure 2 - Modèle conceptuel des différentes entités impliquées dans les requêtes et les réponses

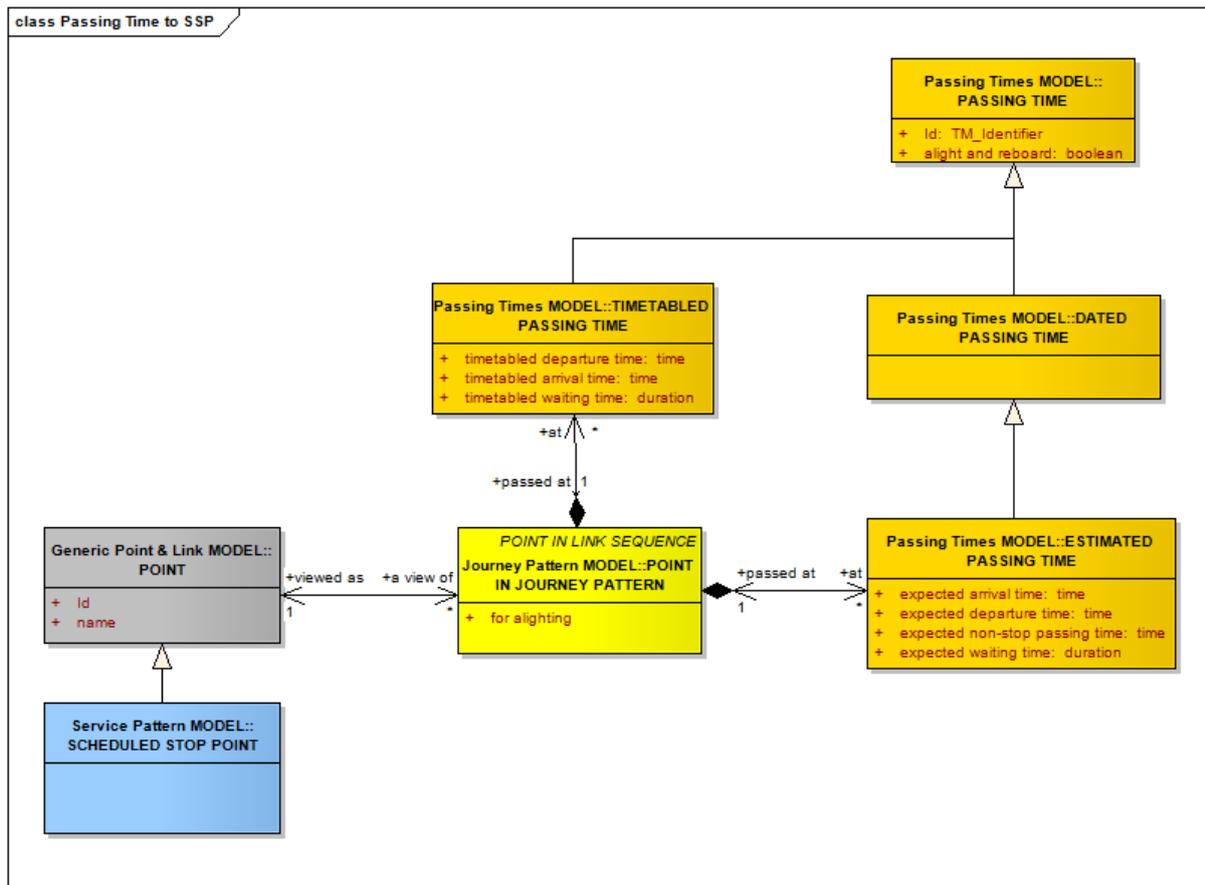


Figure 3 - Détail du modèle conceptuel liant les points d'arrêts planifiés aux heures de passage

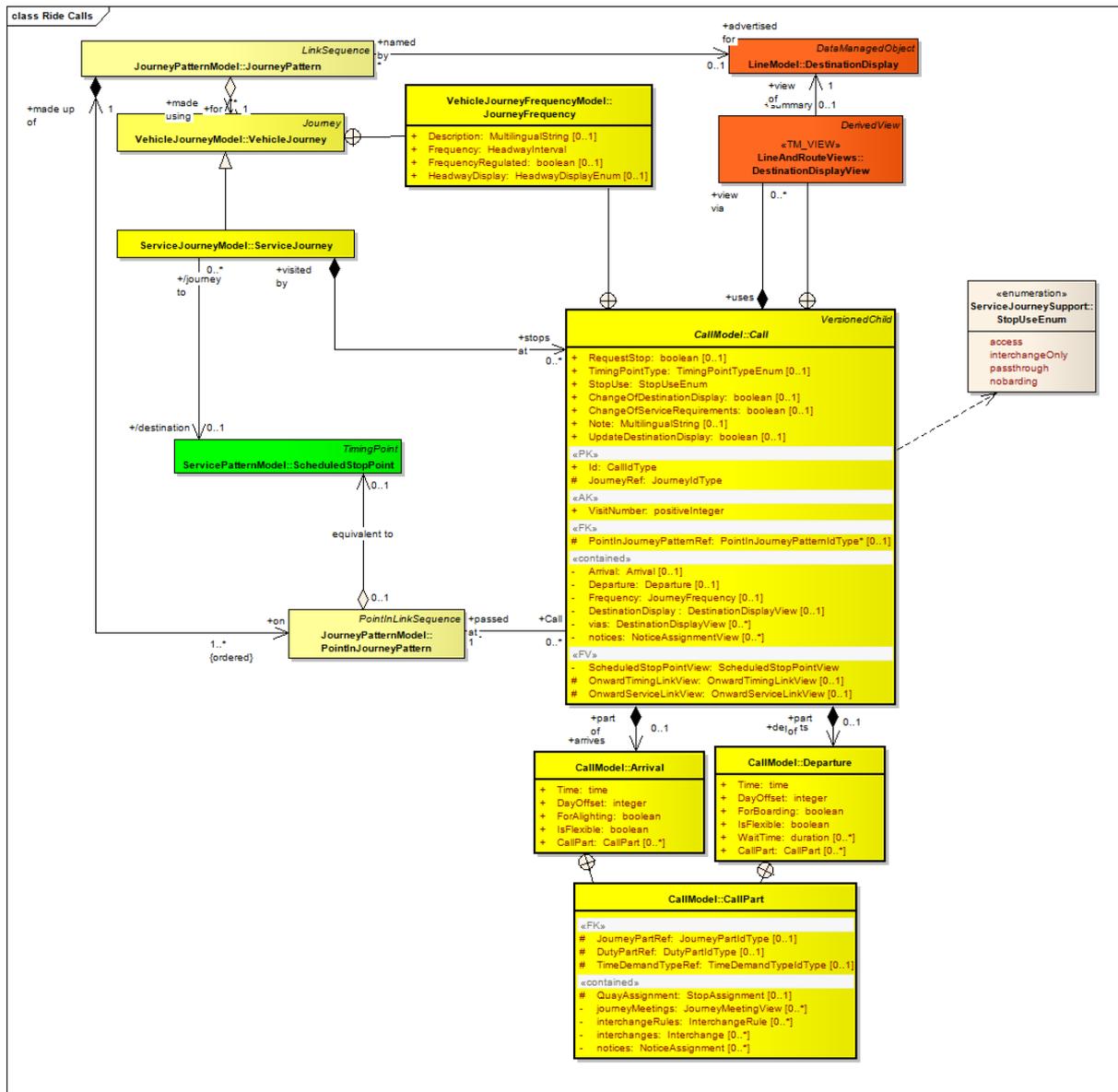


Figure 4 - Modèle physique détaillant les heures de passage (groupés au sein de l'entité Call)

3 Définition de l'API

3.1 Principe

L'API est constituée de deux méthodes permettant de :

- identifier des points de départ et d'arrivée pour la recherche ;
- réaliser un calcul d'itinéraires.

La réponse peut-être renvoyé dans différents formats (XML, JSON, HTML, ou KML). Pour le format XML et JSON, l'API s'appuie sur une structure de données définie sous la forme d'un schéma XSD. La documentation complète du schéma est disponible dans les annexes fournies avec ce document (cf. fichier « API-JourneyPlanner.xsd.pdf »).

3.2 Identification des points de départ ou d'arrivée

3.2.1 Objectif

Cette méthode permet de rechercher un arrêt, un lieu, une commune, ou une adresse à partir d'un libellé. La méthode retourne les différents éléments correspondant au libellé. Les valeurs des identifiants sont pré formatées pour être directement utilisables par la méthode de calcul d'itinéraires.

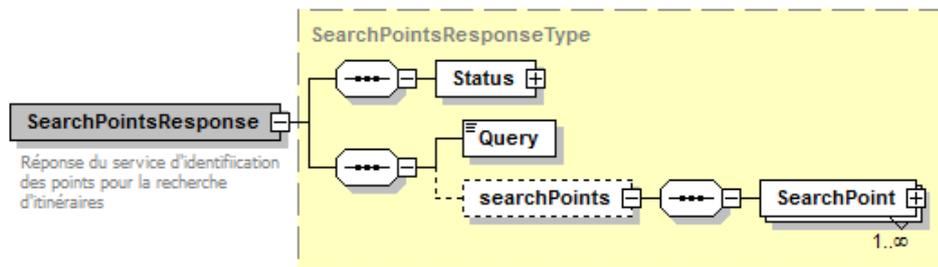
3.2.2 Description des paramètres de la requête

Query	1:1	Chaîne	Eléments de recherche. Exemple : « Charles de Gaulle ».
Type	0:1	chaîne (énumération)	Spécification des différents types de point à inclure dans la recherche. Par défaut tous les types sont inclus. Les différents types doivent être séparés par un « ». Cf. XSD définition du type « SiteTypeEnumeration ».
MaxItems	0:1	entier	Nombre maximum de propositions. Valeur par défaut selon implémentation.

3.2.3 Description de la structure de réponse

L'élément de réponse « SearchPointsResponse » s'appuie sur le type « SearchPointsResponseType » pour fournir une liste de points.

Schéma de la structure de réponse



L'élément « Status » contient le code de retour de l'exécution de la requête, ainsi qu'un commentaire en cas d'erreur.

L'élément « Query » contient la valeur fournie dans le champ « Query » de la requête.

Aperçu de la structure d'un point (type « SiteType ») :

id	1:1	chaîne	Identifiant du point structuré selon l'implémentation. Exemple avec une structuration de type Trident : « AOT:StopArea:51 ».
SiteType	1:1	chaîne (énumération)	Type de point. Cf. XSD définition du type « SiteTypeEnumération ».
Name	1:1	chaîne	Nom du point.
CityName	0:1		Nom de la commune où est situé le point.
CityCode	0:1		Identifiant de la commune (exemple : code Insee).
Position	0:1		Coordonnées géographiques du point au format WGS84.
	1:1	Décimal	
Language	0:1	chaîne	Code pays (ISO) pour indiquer la langue des éléments retournés.

Pour plus de détails sur la structure des objets voir la documentation du XSD « API-JourneyPlanner.xsd.pdf » et les exemples accompagnant ce document.

3.3 Calcul d'itinéraires transport en commun

3.3.1 Objectif

L'objectif est de fournir une méthode pour effectuer un calcul d'itinéraire en transport en commun, et d'obtenir une solution détaillée. Cette interface reste toutefois une interface simple pour la mise en place du service à l'utilisateur final. Elle a pour vocation d'être mise en œuvre dans des environnements comme des navigateurs Web, des applications de Smartphone, etc. Cela reste donc une interface d'appel simple, aussi bien pour sa mise en œuvre qu'au niveau de la richesse de ses paramètres, concentrés sur les paramètres les plus significatifs de l'interface utilisateur.

Le périmètre fonctionnel est donc volontairement restreint, afin d'en faciliter son utilisation et son implémentation :

- Les points de départ et d'arrivée sont des positions géographiques (en WGS84).
- La recherche est restreinte aux modes de transport en commun. Une recherche combinant plusieurs modes (ex : voiture + vélo + TC) complexifie la requête en raison du trop grand nombre de paramètres nécessaires.
- Les options de recherche sont limitées aux critères :
 - « plus rapide » (arriver au plus tôt pour une recherche avec une heure de départ, ou partir au plus tard pour une recherche avec une heure d'arrivée)
 - « minimum de correspondances »
 - « minimum de temps de transport »

Les modes de transport et les options de recherche, étant définis sous la forme d'énumération, il est possible d'étendre facilement les fonctionnalités selon les besoins (ex : option « le moins polluant »).

3.3.2 Description des paramètres de la requête

DepId		1:1	Chaîne	Identifiant ou coordonnées du point de départ au format WGS84.
DepLon DepLat			Décimal	
ArrId		1:1	Chaîne	Identifiant ou coordonnées du point d'arrivée au format WGS84.
ArrLon ArrLat			Décimal	
Date		0:1	chaîne	Date du trajet. Par défaut, c'est la date courante. Format : AAAA-MM-JJ
Choix	DepartureTime	1:1	chaîne	heure de départ ou d'arrivée. Format : HH-MM.
	ArrivalTime			
Algorithm		0:1	chaîne (énumération)	Option d'optimisation de la de recherche : Valeur « FASTEST » : trajet le plus rapide (par défaut). Valeur « MINCHANGES » : trajet avec le moins de changement. Valeur « SHORTEST » : trajet (TC) le plus court.
Modes		0:1	chaîne (énumération)	Spécification des modes de transport TC à utiliser (bus, car, métro, train, ...) Liste des modes séparés par un « ».

MaxTrips	0:1	entier	Nombre maximum de propositions de trajets souhaité (défaut 1).
Language	0:1	chaîne	Code pays (ISO 639-1) pour indiquer la restitution des informations dans la langue souhaitée (notamment pour les perturbations). Cf. code pays : http://www.loc.gov/standards/iso639-2/php/code_list.php

3.3.2.1 Modes de transport

La définition des modes de transport s'appuie sur Trident .

Mode	Description
AIR	Avion
TRAIN	Train
LOCAL_TRAIN	TER (SNCF), voir RER
RAPID_TRANSIT	TGV (SNCF)
LONG_DISTANCE_TRAIN	Autre Train (SNCF)
METRO	Métro
TRAMWAY	Tramway
COACH	Car
BUS	Bus
FERRY	Ferry
WATERBORNE	Navette maritime
TROLLEY_BUS	Trolley
SHUTTLE	Navette
TAXI	Taxi
OTHER	Autre mode
UNKNOWN	Non défini

Extensions possibles

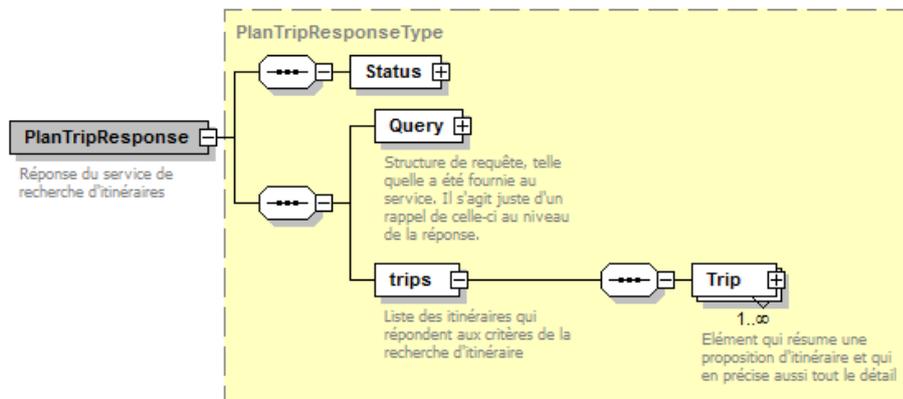
Bien que le périmètre de l'API soit restreint au transport en commun, il y a, au niveau de la requête et de la réponse, tous les éléments pour implémenter une recherche 100% voiture, vélo, ou marche à pied.

PRIVATE_VEHICLE	Voiture
WALK	Marche à pied
BICYCLE	Vélo

3.3.3 Description de la structure de réponse

L'élément de réponse « PlanTripResponse » s'appuie sur le type « PlanTripResponseType » pour fournir un ou plusieurs itinéraires.

Schéma de la structure de réponse



L'élément « Status » contient le code de retour de l'exécution de la requête, ainsi qu'un commentaire en cas d'erreur.

L'élément « Query » contient une structure représentant tous les paramètres de la requête (cf. type « PlanTripRequestType »).

Aperçu de la structure d'un itinéraire :

DepartureTime	1:1	heure	Horaire de départ de la proposition d'itinéraire.
ArrivalTime	1:1	heure	Horaire d'arrivée de la proposition d'itinéraire.
Duration	1:1	chaîne	Durée totale du trajet.
KmlOverview	0:1		URI d'une page au format KML qui trace la totalité du parcours de l'itinéraire
sections	1:1		Conteneur de sections.
choix	Section	1:n	Portion d'un itinéraire, soit en TC, soit en marche à pied (ou autres modes).
	PTRide	1:1	PTRideType Une portion de l'itinéraire qui est empruntée par un véhicule de transport en commun (TC).
	Leg	1:1	LegType Une portion de l'itinéraire réalisée sans transport en commun.

Pour plus de détails sur la structure des objets voir la documentation du XSD « API-JourneyPlanner.xsd.pdf » et les exemples accompagnant ce document.

4 Présentation des différents niveaux d'interface

Différents niveaux d'interface sont proposés, de l'intégration la plus complexe à l'intégration la plus simplifiée. En effet, il a été prévu que les profils techniques des personnes en charge de l'intégration des fonctionnalités de calcul d'itinéraire soient divers, allant de la Web Agency aguerrie à l'intégration de pages HTML aux développeurs d'applications mobiles experts dans le traitement de web-services en passant par les intégrations dans des applications web dont les traitements sont parallélisés.

4.1 Interfaces REST / JavaScript

Ce niveau d'interface s'adresse aux développeurs possédant des capacités d'intégration très fortes, sans préjuger du contexte d'intégration : il s'adresse aussi bien à des développeurs d'applications web qu'aux développeurs d'applications mobiles ou sur client « lourd ». L'API ne fournissant que le résultat, les développeurs devront construire toutes l'IHM pour proposer une fonction de recherche d'itinéraire aboutie (levée d'ambiguïté sur la saisie des points d'origine et de destination, solutions « plus tôt » / « plus tard », itinéraire retour, ...).

L'interface REST consiste à retourner un document structuré dans le langage XML, à partir d'une requête HTTP GET (ou POST) précisant les différents paramètres du calcul d'itinéraire ou de la recherche lexicographique. Le choix du XML a été guidé par le fait que la technologie se devait d'être totalement décorrélée de l'environnement d'intégration technique, et le côté universel du langage XML au-delà de tout langage et technologie permet de répondre à cette contrainte.

L'interface JavaScript consiste en la traduction dans le format de donnée JSON de la sortie XML, intégrable automatiquement et directement dans un environnement JavaScript. Cette interface aux développeurs web souhaitant intégrer les fonctionnalités d'itinéraire dans des applications web dont l'architecture implique une parallélisation forte des appels à partir de la machine cliente via la technologie AJAX.

Pour les développeurs utilisant l'API, les possibilités de personnalisation de l'intégration sont très importantes, mais demandent une compréhension métier complexe.

4.2 Interface HTML simple

Pour répondre à un besoin d'intégration d'une fonction de calcul d'itinéraires en mode « marque blanche », l'API propose une interface HTML. L'API couvre uniquement la fonction de calcul, permettant d'obtenir une solution en HTML. La construction du formulaire de recherche reste à la charge de l'intégrateur.

Ce niveau d'interface s'adresse plutôt au Web Masters, qu'aux développeurs. En effet, son utilisation ne nécessite pas de compétences techniques particulières. L'objectif est d'intégrer, à moindre coût, une fonction de calcul d'itinéraires dans un site existant. L'interface propose donc, une mise en forme préétablie pour simplifier au maximum l'intégration des fonctionnalités de recherche d'itinéraire TC.

D'un point de vue technique, cette interface retourne à partir d'une requête un résultat d'un itinéraire sous forme HTML, prêt à être représenté sous forme de cadre (frame ou iframe) dans une page web. L'effort d'intégration consiste à être en capacité de modeler la requête, et s'adresse particulièrement à des sites web qui souhaiteraient intégrer pour un coût minimal la recherche d'itinéraire en transports en commun entre deux points : aider un client à venir en direction d'une boutique, d'un lieu de spectacle, etc.

Les possibilités de personnalisation sont volontairement limitées, afin de garantir l'aspect « plug and play » de cette interface.

4.3 Interface KML

Cette interface reprend les principes de l'interface HTML, mais avec l'objectif de faire un rendu cartographique de la solution trouvée par le calculateur d'itinéraire.

Un rendu cartographique sur une page Web, passe obligatoire par l'utilisation d'API JavaScript telles que GoogleMaps ou OpenLayers. Ces API sont suffisamment documentées pour qu'un Web Master puisse facilement les utiliser (cf. chapitre 8 « Exemples d'intégration dans une carte »).

4.4 Interface pour calcul d'itinéraire distribué

Si de nombreux SIM couvrent le territoire national à différentes échelles permettant localement le calcul d'un itinéraire multimodal, il n'existe pas en France de calculateur d'itinéraires réparti entre 2 SIM, et à plus forte raison en mesure d'agréger la recherche d'itinéraires provenant de différents SIM locaux dans un cadre national. L'internaute voyageur doit alors combiner lui-même les résultats éventuellement obtenus.

L'intérêt pour le voyageur est de pouvoir disposer d'un calcul d'itinéraire d'adresse à adresse pour planifier ses trajets à l'échelle nationale, tout en permettant aux SIM « locaux » de rester gestionnaires et maîtres de leur système d'information.

Le projet de définition d'une interface pour le calcul d'itinéraire distribué n'est pas dans le périmètre de ce document. Néanmoins, rien n'interdit a priori la conception de solutions techniques pour le calcul réparti d'itinéraires s'appuyant sur les interfaces d'appels définies dans ce document.

5 Implémentation de l'API

5.1 Service REST

L'implémentation de l'API sous la forme d'un service Web de type REST, à l'avantage d'utiliser directement le protocole « http », et de bénéficier de propriétés standards pour faire évoluer le service (authentification, encodage, cache, ...).

Pour plus d'informations sur l'architecture REST :

http://fr.wikipedia.org/wiki/Representational_State_Transfer.

5.2 Clef d'accès

La mise à disposition d'une API peut générer une montée en charge non négligeable sur les serveurs d'exploitation. Pour cette raison, il est nécessaire de contrôler les accès au service, et au besoin, limiter l'utilisation de certaines fonctionnalités (notamment la recherche d'itinéraires).

Cette clef sera passée dans les paramètres de la méthode.

5.3 Interrogation du service

L'URL du service est composée selon le format suivant :

```
http://host/api/[api]/[version]/[méthode]/[format]?key=[clef utilisateur]&([param]=[valeur])*
```

La sémantique des différents composants de l'URL étant la suivante :

Composant	Description
api	Nom de l'API sollicitée
version	Version de l'API sollicitée
méthode	Méthode de l'API invoquée
format	Format de la réponse souhaité : XML, JSON, HTML, ou KML.
clef	Un identifiant unique qualifiant l'utilisateur ou l'application.
param/valeur	Les paramètres d'appel de la méthode.

Les paramètres, ainsi que la clef peuvent être fournis, soit directement dans l'URL (méthode « GET »), soit dans le contenu de la requête (méthode « POST »).

Pour l'implémentation de l'API, nous utiliserons les dénominations suivantes :

- Api : « JourneyPlanner »
- Version : « v1 »
- Méthodes :
 - o « SearchPoints » : pour la détermination des points de départ et d'arrivée.
 - o « PlanTrip » : pour le calcul d'itinéraires

5.4 Réponse du service

5.4.1 Format & Contenu

Le format de la réponse dépend du composant « format » dans l'url d'appel du service. Soit le service retourne une structure objet (format XML ou JSON), soit du contenu destiné à l'affichage (format HTML ou KML).

Pour les formats XML ou JSON, la réponse s'appuie sur deux éléments du schéma : « SearchPointsResponse » et « PlanTripResponse ». Ces deux éléments héritent de la structure de base « AbstractResponseType » qui définit le statut de la réponse.

5.4.2 Gestion du statut de la réponse

Le statut de la réponse est géré sur deux niveaux.

5.4.2.1 Statut HTTP

Le premier niveau correspond à la couche HTTP. On utilise les codes standards suivants :

200	Ok	La requête est acceptée.
403	Forbidden	Accès refusé. Clef d'accès inconnue ou absente.
404	Not Found	Api ou méthode inconnue.

5.4.2.2 Statut de la réponse

Dans l'élément « Status », on précisera un code plus spécifique à l'API.

Enumération (cf. XSD type « StatusTypeEnumeration ») :

OK	Pas d'erreur d'exécution.
INTERNAL_ERROR	Erreur lors de l'exécution (Exception).
BAD_REQUEST	Les paramètres fournis ne permettent pas la bonne exécution de la méthode.
REQUESTED_DATE_OUT_OF_SCOPE	Pas d'information pour la date demandée.
DEPARTURE_UNKNOWN	Point non identifiable.
ARRIVAL_UNKNOWN	
DEPARTURE_TOO_FAR	Les coordonnées du point ne permettent pas de trouver un point d'arrêt suffisamment proche (distance selon implémentation).
ARRIVAL_TOO_FAR	
NO_SOLUTION_FOR_REQUEST	Aucune solution trouvée.

5.5 Définition des méthodes

5.5.1 Méthode « SearchPoints »

L'URL de la méthode est composée selon le format suivant :

```
http://host/api/JourneyPlanner/v1/SearchPoints/[format]?key=[clef
utilisateur]&([param]=[valeur])*
```

La méthode « SearchPoints » accepte trois formats de sortie : XML, JSON, et KML.

5.5.1.1 Exemple d'interrogation

Recherche des points correspondant à la recherche du terme « louvre » :

```
http://host/api/JourneyPlanner/v1/SearchPoints/xml?key=DEMO&Query=louvre
```

5.5.1.2 Exemple de réponse en XML

```
<?xml version="1.0" encoding="UTF-8"?>
<SearchPointsResponse>
  <Status><Code>OK</Code></Status>
  <Query>louvre</Query>
  <searchPoints>
    <SearchPoint id="AOT:StopArea:12545">
      <Name>PALAIS ROYAL MUSEE DU LOUVRE</Name>
      <SiteType>STOP_PLACE</SiteType>
      <Position><Lat>48.86167035</Lat><Long>2.33341346</Long></Position>
      <CityName>PARIS</CityName><CityCode>75056</CityCode>
    </SearchPoint>
    <SearchPoint id="AOT:StopArea:29884">
      <Name>PONT NEUF / QUAI DU LOUVRE</Name>
      <SiteType>STOP_PLACE</SiteType>
      <Position><Lat>48.85814722</Lat><Long>2.34279825</Long></Position>
      <CityName>PARIS</CityName><CityCode>75056</CityCode>
    </SearchPoint>
    <SearchPoint id="AOT:Site:845988">
      <Name>LE CARROUSEL DU LOUVRE</Name>
      <SiteType>POI</SiteType>
      <Position><Lat>48.86144567</Lat><Long>2.33345434</Long></Position>
      <CityName>PARIS</CityName><CityCode>75056</CityCode>
    </SearchPoint>
    <SearchPoint id="AOT:City:95351">
      <Name>LOUVRES</Name>
      <SiteType>CITY</SiteType>
      <Position><Lat>49.04524402</Lat><Long>2.50931693</Long></Position>
      <CityName>LOUVRES</CityName><CityCode>95351</CityCode>
    </SearchPoint>
  </searchPoints>
</SearchPointsResponse>
```

5.5.1.3 Exemple de réponse en JSON

```
{
  "SearchPointsResponse": {
    "Status": {
      "Code": "OK",
    },
    "Query": "louvre",
    "searchPoints": {
      "SearchPoint": [
        {
          "id": "AOT:StopArea:12545",
          "Name": "PALAIS ROYAL MUSEE DU LOUVRE",
          "SiteType": "STOP_PLACE",
          "Position": {
            "Lat": "48.86167035",
            "Long": "2.33341346"
          },
          "CityName": "PARIS",
          "CityCode": "75056"
        },
        {
          "id": "AOT:StopArea:29884",
          "Name": "PONT NEUF \\/ QUAI DU LOUVRE",
          "SiteType": "STOP_PLACE",
          "Position": {
            "Lat": "48.85814722",
            "Long": "2.34279825"
          },
          "CityName": "PARIS",
          "CityCode": "75056"
        },
        {
          "id": "POI:845988",
          "Name": "LE CARROUSEL DU LOUVRE",
          "SiteType": "POI",
          "Position": {
            "Lat": "48.86144567",
            "Long": "2.33345434"
          },
          "CityName": "PARIS",
          "CityCode": "75056"
        },
        {
          "id": "CITY:95351",
          "Name": "LOUVRES",
          "SiteType": "CITY",
          "Position": {
            "Lat": "49.04524402",
            "Long": "2.50931693"
          },
          "CityName": "LOUVRES",
          "CityCode": "95351"
        }
      ]
    }
  }
}
```

5.5.1.4 Exemple de réponse en KML

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns="http://earth.google.com/kml/2.0">
  <Document>
    <Placemark>
      <name>PALAIS ROYAL MUSEE DU LOUVRE, PARIS</name>
      <Point><coordinates>2.33341346,48.86167035,0</coordinates></Point>
    </Placemark>
    <Placemark>
      <name>PONT NEUF / QUAI DU LOUVRE, PARIS</name>
      <Point><coordinates>2.34279825,48.85814722,0</coordinates></Point>
    </Placemark>
    <Placemark>
      <name>LE CARROUSEL DU LOUVRE, PARIS</name>
      <Point><coordinates>2.33345434,48.86144567,0</coordinates></Point>
    </Placemark>
    <Placemark>
      <name>LOUVRES</name>
      <Point><coordinates>2.50931693,49.04524402,0</coordinates></Point>
    </Placemark>
  </Document>
</kml>
```

5.5.2 Méthode « PlanTrip »

L'URL de la méthode est composée selon le format suivant :

```
http://host/api/JourneyPlanner/v1/PlanTrip/[format]?key=[clef utilisateur]&([param]=[valeur])*
```

La méthode « PlanTrip » accepte quatre formats de sortie : XML, JSON, HTML, et KML.

5.5.2.1 Exemple d'interrogation

Recherche d'un trajet le plus rapide entre deux points pour le 05/09/2011 en partant à 15h40 en utilisant uniquement le bus et/ou le car.

```
http://host/api/JourneyPlanner/v1/PlanTrip/xml?key=DEMO&DepLat=45.590000&DepLong=5.892900&ArrLat=45.593340&ArrLong=5.899450&DepartureTime=15-40&Date=2011-09-05&Algorithm=FASTEST&Modes=Bus|Coach&Language=FR
```

5.5.2.2 Exemples de réponse en XML et JSON

Les exemples de réponse dans ces deux formats sont disponibles dans les éléments annexes de ce document.

Voir dans :

- fichier « plantrip_result_sample.xml » : résultat au format XML.
- fichier « plantrip_result_sample.json » : résultat au format JSON.

5.5.2.3 Exemple de réponse en KML

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd=
"http://www.w3.org/2001/XMLSchema" xmlns="http://earth.google.com/kml/2.0">
<Document>
  <Style id="walk"><LineStyle><color>ffc722b</color><width>6</width></LineStyle></Style>
  <Style id="transport"><LineStyle><color>ff00ff55</color><width>6
</width></LineStyle></Style>
  <Placemark>
    <name>RUE DES MALTERIES</name>
    <Point><coordinates>7.744185,48.604953,0</coordinates></Point>
  </Placemark>
  <Placemark>
    <styleUrl>#walk</styleUrl>
    <name>Marche a pied</name>
    <LineString><coordinates>7.744185,48.604953,0 7.743656,48.603653,0
7.746339,48.603197,0 7.746108,48.602928,0 7.745935,48.602711,0 7.745657,48.602381,0
7.745436,48.602085,0 7.745436,48.602085,0 7.744915,48.601523,0
</coordinates></LineString>
  </Placemark>
  <Placemark>
    <name>STRASBOURG - Barr </name>
    <description>Monter dans la ligne 04 WOLFISHEIM / HAUTEPIERRE - REICHSTETT
</description>
    <Point><coordinates>7.744915,48.601523,0</coordinates></Point>
  </Placemark>
  <Placemark>
    <styleUrl>#transport</styleUrl>
    <name>Transport en commun</name>
    <LineString><coordinates>7.744915,48.601523,0 7.746474,48.586981,0
7.743514,48.585809,0 7.744206,48.584363,0 7.745223,48.584344,0 7.768629,48.581406,0
</coordinates></LineString>
  </Placemark>
  <Placemark>
    <name>STRASBOURG - Les Halles - Pont De Paris </name>
    <description>Correspondance avec la ligne F ELSAU - PLACE D'ISLANDE</description>
    <Point><coordinates>7.746474,48.586981,0</coordinates></Point>
  </Placemark>
  <Placemark>
    <name>STRASBOURG - Observatoire </name>
    <description>Descendre de la ligne F ELSAU - PLACE D'ISLANDE</description>
    <Point><coordinates>7.768629,48.581406,0</coordinates></Point>
  </Placemark>
  <Placemark>
    <styleUrl>#walk</styleUrl>
    <name>Itineraire marche a pied</name>
    <LineString><coordinates>7.768629,48.581406,0 7.768608,48.581335,0
7.768234,48.581397,0 7.768136,48.580843,0 7.767948,48.579824,0 7.767939,48.579347,0
7.766764,48.579536,0 7.765955,48.579204,0 7.765538,48.579241,0 7.765525,48.578989,0
7.764379,48.579078,0</coordinates></LineString>
  </Placemark>
  <Placemark>
    <name>RUE RENE DESCARTES</name>
    <Point><coordinates>7.764379,48.579078,0</coordinates></Point>
  </Placemark>
</Document>
</kml>
```

Selon l'implémentation, les styles appliqués aux différents types d'éléments, peuvent être directement définies dans le KML, ou faire référence à une Url sur le serveur hôte. Il est aussi envisageable de « customiser » l'affiche en fonction de la clef utilisateur.

Tout dépendra des conditions d'utilisation de l'API que le fournisseur (ex : AOT) aura mis en place.

6 Utilisation via JavaScript

6.1 La solution JSONP

L'appel des méthodes de l'API peut être utilisé directement par du code s'exécutant côté serveur. Par contre, lorsque l'appel provient du navigateur (via Ajax), le problème de sécurité « Cross-Domain » empêche l'exécution de la requête.

La solution la plus simple pour régler le problème de « Cross-Domain » est d'implémenter le support de JSONP dans les méthodes de l'API.

Le principe est d'ajouter la gestion d'un paramètre dont le nom est « callback ». Ce paramètre représente le nom de la fonction Javascript (côté client) appelée à l'exécution de la requête.

Voir site : <http://www.json-p.org/>

6.2 Implémentation

Côté serveur, la mise en place du support de JSONP implique quelques modifications.

Les méthodes « PlanTrip » et « SearchPoints » sont appelées avec le paramètre supplémentaire « callback ». La réponse n'est plus directement du JSON, mais du code JavaScript qui appelle la fonction définie par le paramètre « callback », avec en paramètre le résultat en JSON.

Exemple :

```
http://host/api/JourneyPlanner/v1/SearchPoints/json?key=DEMO&Query=louvre&callback=GetData
```

Réponse :

```
GetData({"SearchPointsResponse": {"Status": {"Code": "OK"}, "Query": "louvre", "searchPoints": {"SearchPoint": [{" ... } ]}}});
```

La réponse étant du JavaScript, le type MIME retourné par le serveur doit être : « application/x-javascript ».

7 Utilisation du résultat au format HTML

7.1 Objectif de l'interface

Le but de l'API présentée et définie ici est de simplifier l'intégration technique des fonctionnalités de calcul d'itinéraire. L'objectif est une solution « clé en main » permettant en un minimum d'opérations techniques d'intégrer la fonctionnalité.

Le périmètre fonctionnel est volontaire très restreint, notamment pour la requête. Le retour de cette requête est un document HTML prêt à être intégré « tel quel », les seuls éléments de personnalisation étant la feuille de style associée qui permettra une mise en forme en adéquation avec la charte graphique du site qui l'héberge.

7.2 Description des éléments HTML produits

Le code HTML produite comprend quatre sections :

1. Un résumé de l'itinéraire demandé, contenant l'adresse de départ, l'adresse d'arrivée, la date et l'heure du voyage, ainsi que les modes de transports empruntés et le mode de calcul souhaité.
2. Un résumé de l'itinéraire calculé, précisant les adresses de départ et d'arrivées trouvées, ainsi que la durée du transport et le nombre de correspondances.
3. Une carte OpenStreetMap (par exemple) affichant le tracé de l'itinéraire sur la carte.
4. Le détail de la feuille de route structuré sous forme de liste à puces.

Le code à intégrer dans la page sera de la forme :

```
<iframe width="960" height="400" frameborder="0" scrolling="no" marginheight="0" marginwidth="0" src="http://host/api/JourneyPlanner/v1/PlanTrip/xml?key=DEMO&DepLat=45.590000&DepLong=5.892900&ArrLat=45.593340&ArrLong=5.899450&DepartureTime=15-40&Date=2011-09-05&Algorithm=FASTEST&Modes=Bus|Coach&Language=FR"></iframe>
```

7.3 Personnalisation de la présentation (CSS)

Pour une intégration homogène de la réponse HTML du calculateur au site web qui l'héberge, il faut intégrer les éléments de style CSS utilisés dans la page HTML retournée à la feuille de style du site en question. A noter que l'hébergeur aura la charge d'héberger également les éléments de pictogrammes qui seraient référencés dans la feuille de style.

Cette feuille de style sera de la forme :

```

ol, ul { /* Elements de liste */ }
.clearfix:after { content: "."; display: block; clear: both; visibility: hidden; line-
height: 0; height: 0;}
.clearfix { display: inline-block; }
html[xmlns] .clearfix { display: block; }
* html .clearfix { height: 1%; }
body { /* corps de la réponse */ }
.clear {height: 0; visibility: hidden; clear: both; font-size: 0;}
.main { /* division générale de la réponse */ }
.search { /* division pour le résumé de la recherche */ }
.search h2 { /* balise h2 de la recherche */ }
.search ul li { /* éléments de liste de la recherche */ }
.search ul li span { /* éléments de liste de la recherche */ }
.search ul li strong { /* éléments de liste de la recherche */ }
h2.main_title { /* titre des division */ }
.resume ul { /* élément de liste du résumé */ }
.resume li { /* élément de liste du résumé */ }
.resume strong { /* mise en page du résumé */ }
.map h2 { /* titre de la carte */ }
#osmap { /* cadre openstreetmap */ }
.roadmap h2 { /* titre de la feuille de route */ }
.roadmap ol { /* élément de liste de la feuille de route */ }
.roadmap li { /* élément de liste de la feuille de route */ }
.roadmap li li { /* élément de liste de la feuille de route */ }
.roadmap li.roadmap_sc { /* section feuille de route */ }
.roadmap .network { /* mise en page du réseau */ }
.roadmap_sc_departure, .roadmap_sc_arrival { /* section feuille de route départ et
arrivée */ }
.roadmap_sc { /* section feuille de route */ }
.roadmap h3 { /* élment de titre de la feuille de route */ }
.roadmap .rsc_point { /* mise en page nom de station de départ */ }
.roadmap .rsc_point strong { /* mise en page nom de station de départ */ }
.roadmap .mode { /* mode de transport */ }
.roadmap .mode_walk { /* mode marche à pied */ }
.roadmap .mode_air { /* mode avion */ }
.roadmap .mode_train { /* mode train */ }
.roadmap .mode_localtrain { /* mode TER */ }
.roadmap .mode_rapidtrain { /* mode TGV */ }
.roadmap .mode_longdistancetrain { /* mode autre train (hors sncf) */ }
.roadmap .mode_tramway { /* mode tramway */ }
.roadmap .mode_metro { /* mode metro */ }
.roadmap .mode_bus { /* mode bus */ }
.roadmap .mode_car { /* mode car */ }
.roadmap .mode_ferry { /* mode ferry */ }
.roadmap .mode_waterbone { /* mode navette fluviale */ }
.roadmap .mode_trolley { /* mode trolley */ }
.roadmap .mode_shuttle { /* mode navette */ }
.roadmap .mode_taxi { /* mode taxi */ }
.roadmap .mode_other { /* mode autre mode */ }
.roadmap .mode_unknown { /* mode inconnu */ }
.roadmap .mode_velo { /* mode vélo */ }
.roadmap .network a { /* ancre pour les réseaux */ }
.roadmap .network span, .roadmap .mode span { visibility: hidden; }
.roadmap_sc_walk h4 { /* titre pour la partie marche à pied */ }
.roadmap_sc_last { /* dernière section */ }
.roadmap_sc_departure .rsc_hour, .roadmap_sc_arrival .rsc_hour { /* mise en page des
sections */ }

```

7.4 Exemples de mises en forme

Votre recherche

De : Orléans (45000) - rue de Jolie
A : Orléans (45000) - Gare SNCF et Routière
Date : Vendredi 06 Janvier 2012
Heure : Partir après 11h25
Trajet : Le plus rapide
Mode(s) : Car, Bus, Tramway, Train

DÉTAIL DE L'ITINÉRAIRE

Départ : Vendredi 06 Janvier 2012 à 11h26 - Orléans (45000) - rue de Jolie
Arrivée : Vendredi 06 Janvier 2012 à 11h37 - Orléans (45000) - Gare SNCF et Routière
Correspondance : Direct
Durée totale : 00h11

Carte de l'itinéraire

Feuille de route de l'itinéraire

Départ	11h26	rue de Jolie (Orléans)
		Départ : 11h26 rue de Jolie (Orléans) Arrivée : 11h32 Orléans - Libération Durée : 06 min
		Ligne TRAMA : FLEURY JULES VERNE - CHRO LA SOURCE Départ : 11h32 Libération Arrivée : 11h37 Gares d'Orléans Quai A Durée : 05 min
Arrivée	11h37	Gares d'Orléans Quai A (Orléans)

VOTRE RECHERCHE

De : Orléans (45000) - rue de Jolie
A : Orléans (45000) - Gare SNCF et Routière
Date : Vendredi 06 Janvier 2012
Heure : Partir après 11h25
Trajet : Le plus rapide
Mode(s) : Car, Bus, Tramway, Train

Détail de l'itinéraire

Départ : Vendredi 06 Janvier 2012 à 11h26 - Orléans (45000) - rue de Jolie
Arrivée : Vendredi 06 Janvier 2012 à 11h37 - Orléans (45000) - Gare SNCF et Routière
Correspondance : Direct
Durée totale : 00h11

Carte de l'itinéraire

Feuille de route de l'itinéraire

Départ	11h26	rue de Jolie (Orléans)
		Départ : 11h26 rue de Jolie (Orléans) Arrivée : 11h32 Orléans - Libération Durée : 06 min
		Ligne TRAMA : FLEURY JULES VERNE - CHRO LA SOURCE Départ : 11h32 Libération Arrivée : 11h37 Gares d'Orléans Quai A Durée : 05 min
Arrivée	11h37	Gares d'Orléans Quai A (Orléans)

8 Exemples d'intégration dans une carte

8.1 Introduction

Cette API a été conçue pour faciliter l'intégration d'un calcul d'itinéraires dans un site Web. A partir des deux méthodes disponibles, il est possible de réaliser simplement une page web permettant de lancer un calcul d'itinéraires, et d'avoir un rendu sur fond cartographique.

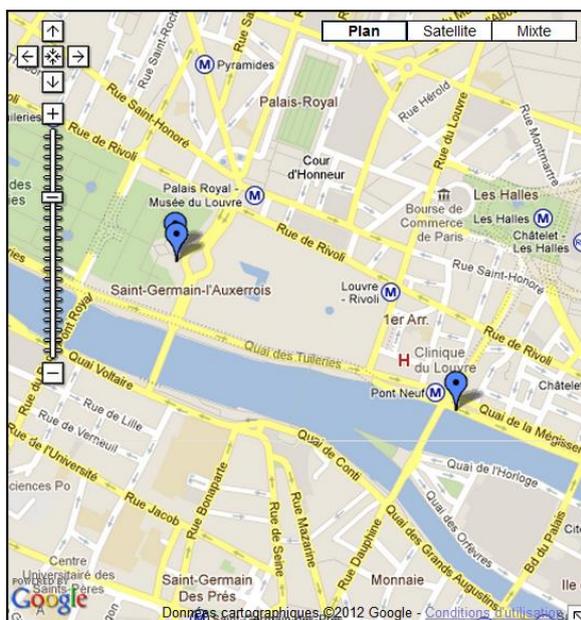
8.2 Affichage des points de départ ou d'arrivée

La méthode « SearchPoints » retourne la liste des points. En précisant le format de sortie KML, il est facile de les afficher dans une carte Google Maps.

Exemple de code Javascript pour un affichage avec Google Maps.

```
function InitMap() {  
    var centerPoint = new google.maps.LatLng(48.604953, 7.744185);  
    var myOptions = {  
        zoom: 11,  
        center: centerPoint,  
        mapTypeId: google.maps.MapTypeId.ROADMAP  
    }  
    var map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);  
}  
  
function SearchPointsLayer(query) {  
    var searchPointsLayer = new google.maps.KmlLayer('http://host/api/JourneyPlanner/v1/SearchPoints/kml?key=DEMO&Query=' + query);  
    searchPointsLayer.setMap(map);  
}
```

Rendu avec Google Maps



8.3 Affichage du résultat de la recherche d'itinéraires

La méthode « PlanTrip » accepte également un format de sortie KML.

Exemple de code JavaScript pour un affichage avec OpenLayers.

```
function InitMap() {
  var map = new OpenLayers.Map({
    div: "basicMap",
    layers: [
      new OpenLayers.Layer.OSM(),
      new OpenLayers.Layer.Vector("KML", {
        strategies: [new OpenLayers.Strategy.Fixed()],
        protocol: new OpenLayers.Protocol.HTTP({
          url:
            "http://host/api/JourneyPlanner/v1/PlanTrip/kml?key=DEMO&DepLat=48.604953&DepLong=7.744185
            &ArrLat=48.579078&ArrLong=7.764379&DepartureTime=15-40&Date=2011-09-05&Algorithm=FASTEST&M
            odes=Bus|Coach&Language=FR",
          format: new OpenLayers.Format.KML({
            extractStyles: true,
            extractAttributes: true,
            maxDepth: 2
          })
        })
      ])
    ],
    center: new OpenLayers.LonLat(7.744185,48.604953)
      .transform(
        new OpenLayers.Projection("EPSG:4326"), // transform from WGS 1984
        new OpenLayers.Projection("EPSG:900913") // to Spherical Mercator Projection
      ),
    zoom: 11
  });
}
```

Rendu avec OpenStreetMap

